

RESEARCH

Open Access



Variable forgetting factor mechanisms for diffusion recursive least squares algorithm in sensor networks

Ling Zhang¹, Yunlong Cai^{1*}, Chunguang Li¹ and Rodrigo C. de Lamare²

Abstract

In this work, we present low-complexity variable forgetting factor (VFF) techniques for diffusion recursive least squares (DRLS) algorithms. Particularly, we propose low-complexity VFF-DRLS algorithms for distributed parameter and spectrum estimation in sensor networks. For the proposed algorithms, they can adjust the forgetting factor automatically according to the posteriori error signal. We develop detailed analyses in terms of mean and mean square performance for the proposed algorithms and derive mathematical expressions for the mean square deviation (MSD) and the excess mean square error (EMSE). The simulation results show that the proposed low-complexity VFF-DRLS algorithms achieve superior performance to the existing DRLS algorithm with fixed forgetting factor when applied to scenarios of distributed parameter and spectrum estimation. Besides, the simulation results also demonstrate a good match for our proposed analytical expressions.

Keywords: Sensor networks, Distributed parameter estimation, Distributed spectrum estimation, Diffusion recursive least-squares, Variable forgetting factor

1 Introduction

Distributed estimation is commonly utilized for distributed data processing over sensor networks, which exhibits increased robustness, flexibility, and system efficiency compared to centralized processing. Owing to these merits, distributed estimation has received more and more attention and been widely used in applications ranging from environmental monitoring [1], medical data collecting for healthcare [2], animal tracking in agriculture [1], monitoring physical phenomena [3], localizing moving mobile terminals [4, 5] to national security. Particularly, distributed estimation technique relies on the cooperation among geographically spread sensor nodes to process locally collected data. With different cooperation strategies employed, distributed estimation algorithms can be classified into the incremental type and the diffusion type. Note that we consider the diffusion cooperation strategy in this paper since the incremental strategy requires the definition of a path through the

network and may be not suitable for large networks or dynamic configurations [6, 7]. Many distributed estimation algorithms with the diffusion strategy have been put forward recently, such as diffusion least-mean squares (LMS) [8, 9], diffusion sparse LMS [10–12], variable step size diffusion LMS (VSS-DLMS) [13, 14], diffusion recursive least squares (RLS) [6, 7], distributed sparse RLS [15], distributed sparse total least squares (LS) [16], diffusion information theoretic learning (ITL) [17], and the diffusion-based algorithm for distributed censor regression [18]. Among assorted distributed estimation algorithms, the RLS-based algorithms achieve superior performance to the LMS-based ones by inheriting the advantages of fast convergence and low steady-state misadjustment from the RLS technique. Thus, the distributed estimation algorithms based on the diffusion strategy and the RLS adaptive technique are investigated in this paper.

However, the existing RLS-based distributed estimation algorithms provide a fixed forgetting factor, which has some drawbacks. With a fixed forgetting factor, the algorithm fails to keep up with real-time variations in environment, such as variations in sensor network topology. Moreover, it is expected to adjust the forgetting

*Correspondence: ylcai@zju.edu.cn

¹College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, People's Republic of China

Full list of author information is available at the end of the article

factors automatically according to the estimation errors rather than choose appropriate values for them through simulations. There have been several studies on variable forgetting factor (VFF) methods. Specifically, the classic gradient-based VFF (GVFF) mechanism was proposed in [19], and most of the existing VFF mechanisms are extensions of this method [20–24]. Nevertheless, the GVFF mechanism requires a large amount of computation. In order to reduce the computational complexity, the improved low-complexity VFF mechanisms have been reported in [25, 26]. To the best of our knowledge, the existing VFF mechanisms are mostly employed in a centralized context and have not been considered in the field of distributed estimation yet.

In this work, the previously reported VFF mechanisms [25, 26] are employed to the diffusion RLS algorithms for distributed signal processing applications, by simplifying the inverse relation between the forgetting factor and the adaptation component to provide lower computational complexity. The resulting algorithms are referred to as low-complexity time-averaged VFF diffusion RLS (LTVFF-DRLS) algorithm and low-complexity correlated time-averaged VFF diffusion RLS (LCTVFF-DRLS) algorithm, respectively. Compared with the GVFF mechanisms, the proposed LTVFF and LCTVFF mechanisms can reduce the computational complexity significantly [25, 26]. Then, we carry out the analysis for the proposed algorithms in terms of the mean and mean square error performance. Finally, we provide simulation results to verify the effectiveness of the proposed algorithms when applied in distributed parameter estimation and distributed spectrum estimation.

Our main contributions are summarized as follows:

- 1) We propose the low-complexity VFF-DRLS algorithms for distributed estimation in sensor networks. To the best of our knowledge, the VFF mechanisms have not been considered in the distributed estimation algorithms yet.
- 2) We study the mean and mean square performance for the proposed algorithms in a general case, and provide the transient analysis for a specialized case. Specifically, for the general case, in terms of the mean performance, we show that the mean value of the weight error vector approaches zero as iteration numbers go to infinity, which implies the asymptotical convergence of the proposed algorithms; from the perspective of mean square performance, we derive the mathematical expressions for the steady-state MSD and EMSE values. In the specialized case, we study the transient analysis by focusing on the learning curve and prove that the proposed algorithms are convergent and the convergence rate is related to the varying forgetting factors.

- 3) We perform simulations to evaluate the performance of the proposed algorithms when applied to distributed parameter estimation and distributed spectrum estimation tasks. The simulation results indicate that the proposed algorithms exhibit remarkable improvements in convergence and steady-state performance when compared with the DRLS algorithm that has a fixed forgetting factor. Besides, effectiveness of our analytical expressions for calculating the steady-state MSD and EMSE is verified by the simulation results. In addition, we also provided detailed simulation results regarding the choice of the parameters in the proposed algorithms to help with the parameter selection in practice.

This paper is organized as follows. Section 2 provides the system model for the distributed estimation over sensor networks. Besides, the DRLS algorithm with the fixed forgetting factor is described briefly. In Section 3, two low-complexity VFF mechanisms are presented, followed by the analyses for the variable forgetting factor in terms of steady-state statistical properties. Besides, the proposed LTVFF-DRLS algorithm and the LCTVFF-DRLS algorithm are presented. In the last part of this section, the computational complexity of the VFF mechanisms as well as the proposed algorithms is analyzed. In Section 4, detailed analyses based on mean and mean-square performance for the proposed algorithms are carried out and analytical expressions to compute MSD and EMSE are derived. In addition, transient analysis for a specialized case is provided in the last part of Section 4. In Section 5, simulation results are presented for distributed parameter estimation and distributed spectrum estimation. Section 6 draws the conclusions.

Notation: Boldface letters are used for vectors or matrices, while normal font for scalar quantities. Matrices are denoted by capital letters and small letters are used for vectors. We use the operator $\text{row}\{\cdot\}$ to denote a row vector, $\text{col}\{\cdot\}$ to denote a column vector, and $\text{diag}\{\cdot\}$ to denote a diagonal matrix. The operator $E[\cdot]$ stands for the expectation of some quantity, and $\text{Tr}\{\cdot\}$ represents the trace of a matrix. We use $(\cdot)^T$ and $(\cdot)^{-1}$ to denote the transpose and inverse operator, respectively, and $(\cdot)^*$ for complex conjugate-transposition. We also use the symbol \mathbf{I}_n to represent an identity matrix of size n and $\mathbf{1}$ to denote a vector of appropriate size with all elements equal to one.

2 System model and diffusion-based DRLS algorithm

In this section, we first illustrate the system model for the distributed estimation over sensor networks. Following this, we review the conventional DRLS algorithm with the fixed forgetting factor briefly.

2.1 System model

Let us consider a sensor network consisting of N sensor nodes which are spatially distributed over a geographical area. The set of nodes connected to node k including itself are called the neighbor nodes of node k , denoted by \mathcal{N}_k . The number of the nodes linked to node k is the degree of k , denoted by n_k . The system model for the distributed estimation over sensor networks is presented in Fig. 1.

At each time instant i , each node k has access to complex valued time realizations $\{d_{k,i}, \mathbf{u}_{k,i}\}$, $k = 1, 2, \dots, N$, $i = 1, 2, \dots$, with $d_{k,i}$ a scalar measurement and $\mathbf{u}_{k,i}$ a $M \times 1$ input vector. The relation between the measurement $d_{k,i}$ and the input vector $\mathbf{u}_{k,i}$ can be characterized as

$$d_{k,i} = \mathbf{u}_{k,i}^* \mathbf{w}^o + v_{k,i} \quad (1)$$

where \mathbf{w}^o is the unknown optimal weight vector of size $M \times 1$, and $v_{k,i}$ is zero-mean additive white Gaussian noise with variance $\sigma_{v,k}^2$. Particularly, we assume that the noise variance has been determined in advance somehow. We also assume that the noise samples $v_{k,i}$, $k = 1, 2, \dots, N$, $i = 1, 2, \dots$, are independent of each other as well as the input vectors $\mathbf{u}_{k,i}$. We aim to estimate the unknown optimal weight vector \mathbf{w}^o in a distributed manner. That is, each sensor node k obtains a local estimate $\mathbf{w}_{k,i}$ of size $M \times 1$ to approach the optimal weight vector \mathbf{w}^o as much as possible. To this end, each node k not only uses its local measurement $d_{k,i}$ and input vector $\mathbf{u}_{k,i}$ but also cooperates with its closest neighbors for updating its local estimate $\mathbf{w}_{k,i}$. Specifically, by cooperation, each node k has access to its neighbors' data $\{d_{l,i}, \mathbf{u}_{l,i}\}$ and estimates $\mathbf{w}_{l,i}$ at each

time instant i where $l \in \mathcal{N}_k$, and then, each node k fuses all the available information to update its local estimate $\psi_{k,i}$.

Let us first introduce some vectors and matrices. At each time instant i , by collecting all nodes' measurements into vector \mathbf{y}_i , noise samples into vector \mathbf{v}_i (both of length N), and input vectors into the matrix \mathbf{H}_i of size $N \times M$, we obtain

$$\begin{aligned} \mathbf{y}_i &= \text{col}\{d_{1,i} \dots d_{N,i}\} \\ \mathbf{H}_i &= \text{col}\{\mathbf{u}_{1,i}^* \dots \mathbf{u}_{N,i}^*\} \\ \mathbf{v}_i &= \text{col}\{v_{1,i} \dots v_{N,i}\}. \end{aligned} \quad (2)$$

Following this, we define the covariance matrix of the noise vector \mathbf{v}_i as

$$\mathbf{R}_v = E[\mathbf{v}_i \mathbf{v}_i^*] = \text{diag}\{\sigma_{v,1}^2, \sigma_{v,2}^2, \dots, \sigma_{v,N}^2\}. \quad (3)$$

Next, we stack \mathbf{y}_i , \mathbf{v}_i and \mathbf{H}_i from time instant 0 to time instant i into matrices respectively, which are given by

$$\begin{aligned} \mathcal{Y}_i &= \text{col}\{\mathbf{y}_1 \dots \mathbf{y}_i\} \\ \mathcal{H}_i &= \text{col}\{\mathbf{H}_1 \dots \mathbf{H}_i\} \\ \mathcal{V}_i &= \text{col}\{\mathbf{v}_1 \dots \mathbf{v}_i\}. \end{aligned} \quad (4)$$

Besides, we define $\mathcal{R}_{v,i} = E[\mathcal{V}_i \mathcal{V}_i^*]$.

2.2 Brief review of diffusion-based DRLS algorithm

In this part, we give a brief introduction to the diffusion-based DRLS algorithm [6, 7].

For the diffusion-based DRLS algorithm, the local optimization problem to estimate the optimal weight vector \mathbf{w}^o at each node k can be formulated as follows:

$$\psi_{k,i} = \arg \min_{\mathbf{w}} \left\{ \|\mathbf{w}\|_{\Pi_i}^2 + \|\mathcal{Y}_i - \mathcal{H}_i \mathbf{w}\|_{\mathcal{W}_{k,i}}^2 \right\} \quad (5)$$

Note that the notation $\|\mathbf{a}\|_{\Sigma}^2 = \mathbf{a}^* \Sigma \mathbf{a}$ represents the weighted vector norm of any positive definite Hermitian matrix Σ . Besides, the matrix Π_i is given by $\Pi_i = \lambda^{i+1} \Pi$ where $0 \ll \lambda < 1$ representing the forgetting factor and $\Pi = \delta^{-1} \mathbf{I}_M$ with $\delta > 0$. Furthermore, the matrix $\mathcal{W}_{k,i}$ can be expressed as $\mathcal{W}_{k,i} = \mathcal{R}_{v,i}^{-1} \Lambda_i \text{diag}\{\mathbf{C}_k, \mathbf{C}_k, \dots, \mathbf{C}_k\}$, where $\Lambda_i = \text{diag}\{I_N, \lambda I_N, \dots, \lambda^i I_N\}$ and \mathbf{C}_k is a diagonal matrix. It is worth noting that the main diagonal elements of the matrix \mathbf{C}_k is composed of the k th column of matrix \mathbf{C} . Particularly, the matrix \mathbf{C} is the adaptation matrix for the diffusion-based DRLS algorithm and satisfies $\mathbb{I}^T \mathbf{C} = \mathbb{I}$ and $\mathbf{C} \mathbb{I} = \mathbb{I}$ [6]. Also, note that the matrix \mathbf{C} is a doubly stochastic matrix, that is, both a left stochastic matrix and a right stochastic matrix.

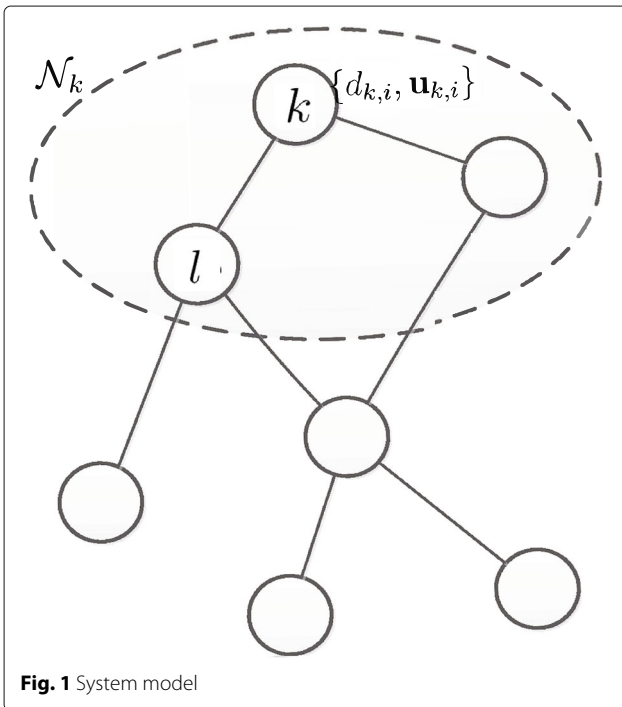


Fig. 1 System model

The optimization problem (5) can be rewritten as follows [6]:

$$\psi_{k,i} = \arg \min_{\mathbf{w}} \left\{ \lambda^{i+1} \|\mathbf{w}\|_{\Pi}^2 + \sum_{j=0}^i \lambda^{i-j} \sum_{l=1}^N \frac{C_{l,k} |d_{l,j} - \mathbf{u}_{l,j}^* \mathbf{w}|^2}{\sigma_{v,l}^2} \right\} \quad (6)$$

where $C_{l,k}$ represents the (l, k) th element of the matrix \mathbf{C} . The closed-form solution to (6) is given by [6, 7]

$$\psi_{k,i} = \mathbf{P}_{k,i} \mathcal{H}_i^* \mathcal{W}_{k,i} \mathcal{Y}_i \quad (7)$$

where $\mathbf{P}_{k,i}$ can be expressed as

$$\mathbf{P}_{k,i} = [\lambda^{i+1} \Pi + \mathcal{H}_i^* \mathcal{W}_{k,i} \mathcal{H}_i]^{-1}. \quad (8)$$

However, the closed-form solution in (7) is obtained via calculating the inversion of matrices, which requires large computation. Instead, the diffusion-based DRLS algorithm provides a recursive approach to solve (6), which can be implemented by the following two steps.

Step 1: Let us take the updates at time instant i for example. Note that we denote the iteration number at time instant i as the superscript $(\cdot)^l$ with $l = 0$ representing the initial value. At the very start, we initialize the intermediate local estimate $\psi_{k,i}$ and the inverse matrix $\mathbf{P}_{k,i}$ for each node k by utilizing the updated results from time instant $i - 1$, that is

$$\begin{aligned} \psi_{k,i}^0 &= \mathbf{w}_{k,i-1} \\ \mathbf{P}_{k,i}^0 &= \lambda^{-1} \mathbf{P}_{k,i-1} \end{aligned} \quad (9)$$

Then, for each node k , its data is updated incrementally among its neighbors, which is given by

$$\psi_{k,i}^l \leftarrow \psi_{k,i}^{l-1} + \frac{C_{l,k} \mathbf{P}_{k,i}^{l-1} \mathbf{u}_{l,i} [d_{l,i} - \mathbf{u}_{l,i}^* \psi_{k,i}^{l-1}]}{\sigma_{v,l}^2 + C_{l,k} \mathbf{u}_{l,i}^* \mathbf{P}_{k,i}^{l-1} \mathbf{u}_{l,i}} \quad (10)$$

$$\mathbf{P}_{k,i}^l \leftarrow \mathbf{P}_{k,i}^{l-1} - \frac{C_{l,k} \mathbf{P}_{k,i}^{l-1} \mathbf{u}_{l,i} \mathbf{u}_{l,i}^* \mathbf{P}_{k,i}^{l-1}}{\sigma_{v,l}^2 + C_{l,k} \mathbf{u}_{l,i}^* \mathbf{P}_{k,i}^{l-1} \mathbf{u}_{l,i}} \quad (11)$$

where the left arrow denotes the operation of assignment. Finally, each node k obtains its ultimate intermediate local estimate $\psi_{k,i}$ which can be expressed as

$$\psi_{k,i} \leftarrow \psi_{k,i}^{|\mathcal{N}_k|} \quad (12)$$

Step 2: Each node k combines the ultimate intermediate local estimate of its own, i.e., $\psi_{k,i}$, obtained in step 1 with that of its neighbors, i.e., $\psi_{l,i}$, $l \in \mathcal{N}_k$ by performing the following diffusion to obtain the local estimate $\mathbf{w}_{k,i}$:

$$\mathbf{w}_{k,i} = \sum_{l=1}^N A_{l,k} \psi_{l,i} \quad (13)$$

where $A_{l,k}$ denotes the (l, k) th element of the matrix \mathbf{A} . Particularly, the matrix \mathbf{A} is the combination matrix for

the diffusion-based DRLS algorithm and is chosen such that $\mathbb{I}^T \mathbf{A} = \mathbb{I}$ [6].

Note that the steps (9)–(13) constitute the diffusion-based DRLS algorithm [6, 7].

3 Low-complexity variable forgetting factor mechanisms

In this section, we introduce the LTVFF mechanism and the LCTVFF mechanism that are employed by our proposed algorithms. Particularly, the analyses for the variable forgetting factor in terms of the steady-state properties of the first-order statistics are presented, and the LTVFF-DRLS algorithm that employs the LTVFF mechanism as well as the LCTVFF-DRLS algorithm that applies the LCTVFF mechanism are proposed. In the last part of this section, we analyze the computational complexity for these two VFF mechanisms as well as the proposed algorithms.

3.1 LTVFF mechanism

Motivated by the VSS mechanism [13, 14] for the diffusion LMS algorithm, the low-complexity VFF mechanisms are designed such that smaller forgetting factors are employed when the estimation errors are large in order to obtain a faster convergence speed, whereas the forgetting factor increases when the estimation errors become small so as to yield better steady-state performance. Based on the above idea, an effective rule to adapt the forgetting factor can be formulated as

$$\lambda_k(i) = [1 - \zeta_k(i)]_{\lambda_-}^{\lambda_+} \quad (14)$$

where the quantity $\zeta_k(i)$ is related to the estimation errors and varies in an inverse way to the forgetting factor, which is referred to as the adaptation component. The operator $[\cdot]_{\lambda_-}^{\lambda_+}$ denotes the truncation of the forgetting factor to the limits of the range $[\lambda_+, \lambda_-]$.

For the LTVFF mechanism, the adaptation component is given by

$$\zeta_k(i) = \alpha \zeta_k(i-1) + \beta |e_k(i)|^2 \quad (15)$$

with parameters $0 < \alpha < 1$ and $\beta > 0$. Besides, α is chosen close to 1 and β is set to be a small value. The quantity $e_k(i)$ denotes the priori estimation error [19] of each node for the DRLS algorithm, which can be expressed as

$$e_k(i) = d_{k,i} - \mathbf{u}_{k,i}^* \mathbf{w}_{k,i-1}. \quad (16)$$

That is to say, in the LTVFF mechanism, the adaptation component is updated based on the instantaneous estimation error.

The LTVFF mechanism is given by (14) and (15). The value of the forgetting factor $\lambda_k(i)$ is controlled by the parameters α and β . Particularly, the effects of α and β on the performance of our proposed algorithms are investigated in Section 5. As can be seen from (14) and (15),

large estimation errors will cause an increase in the adaptation component $\zeta_k(i)$, which yields a smaller forgetting factor and provides a faster tracking speed. Conversely, small estimation errors will lead to the decrease of the adaptation component $\zeta_k(i)$, and thus, the forgetting factor $\lambda_k(i)$ will be increased to yield smaller steady-state misadjustment.

Next, we study the steady-state statistical properties of the adaptation component $\zeta_k(i)$ and the forgetting factor $\lambda_k(i)$. Based on (15), it is reasonable to assume that $\zeta_k(i)$ and $\zeta_k(i-1)$ are approximately equivalent when $i \rightarrow \infty$. By taking expectations on both sides of (15) and letting i goes to infinity, we can obtain $E[\zeta_k(\infty)]$

$$E[\zeta_k(\infty)] = \frac{\beta}{1-\alpha} E[|e_k(\infty)|^2]. \quad (17)$$

Then, we compute the quantity of $E[|e_k(\infty)|^2]$. Let us define the weight error vector for node k as

$$\tilde{\mathbf{w}}_{k,i} = \mathbf{w}_{k,i} - \mathbf{w}^o. \quad (18)$$

According to (16) and (18), we can rewrite $E[|e_k(i)|^2]$ as

$$\begin{aligned} E[|e_k(i)|^2] &= E[|d_{k,i} - \mathbf{u}_{k,i}^* (\tilde{\mathbf{w}}_{k,i-1} + \mathbf{w}^o)|^2] \\ &= E[|v_{k,i} - \mathbf{u}_{k,i}^* \tilde{\mathbf{w}}_{k,i-1}|^2] \\ &= \sigma_{v,k}^2 + E[|\mathbf{u}_{k,i}^* \tilde{\mathbf{w}}_{k,i-1}|^2] \end{aligned} \quad (19)$$

where the term $E[|\mathbf{u}_{k,i}^* \tilde{\mathbf{w}}_{k,i-1}|^2]$ denotes the excess error. Since it is sufficiently small when $i \rightarrow \infty$ compared with the variance of noise, it can be neglected. As a consequence, the following approximation holds

$$E[|e_k(\infty)|^2] \approx \varepsilon_{\min} \quad (20)$$

where ε_{\min} denotes the minimum mean-square error and can be expressed as

$$\varepsilon_{\min} = E[|d_{k,i} - \mathbf{u}_{k,i}^* \mathbf{w}^o|^2] = \sigma_{v,k}^2. \quad (21)$$

Subsequently, by substituting (20) into (17), we can approximately write

$$E[\zeta_k(\infty)] \approx \frac{\beta}{1-\alpha} \varepsilon_{\min}. \quad (22)$$

According to (14), we can deduce

$$E[\lambda_k(\infty)] = 1 - E[\zeta_k(\infty)]. \quad (23)$$

By substituting (22) into (23), we can obtain the first-order statistics of the forgetting factor for the LTVFF mechanism:

$$E[\lambda_k(\infty)] = 1 - \frac{\beta}{1-\alpha} \varepsilon_{\min}. \quad (24)$$

By applying the LTVFF mechanism to the diffusion-based DRLS algorithm, we propose the LTVFF-DRLS algorithm, which is exhibited in the left column of Table 1.

3.2 LCTVFF mechanism

For the LCTVFF mechanism, the forgetting factor can be calculated through (14) while the adaptation component $\zeta_k(i)$ can be adjusted according to an alternative rule, that is, the time-averaged estimation of the correlation of two consecutive estimation errors is employed to the updating equation of the adaptation component $\zeta_k(i)$. Therefore, the rule to update the adaptation component can be described as

$$\zeta_k(i) = \alpha \zeta_k(i-1) + \beta |\rho_k(i)|^2 \quad (25)$$

where $0 < \alpha < 1$ and $\beta > 0$. Particularly, α is set close to 1 and β is chosen to be slightly larger than 0. The quantity $\rho_k(i)$ denotes the time-averaged estimation of the correlation of two consecutive estimation errors, which is defined by

$$\rho_k(i) = \gamma \rho_k(i-1) + (1-\gamma) |e_k(i-1)| |e_k(i)| \quad (26)$$

where $0 < \gamma < 1$ and γ is slightly smaller than 1. Note that the LCTVFF mechanism is given by (14), (25), and (26).

Next, we consider the steady-state statistical properties of $\rho_k(i)$, $\zeta_k(i)$, and $\lambda_k(i)$ for the LCTVFF mechanism. As we will see in simulation results, the proposed algorithm converges to the steady-state in numerous iterations, and thus, the values of $\rho_k(i-1)$ and $\rho_k(i)$ can be assumed to be approximately equivalent, respectively, when i is large enough. Thus, we can obtain $E[|e_k(i-1)| |e_k(i)|] \approx E[|e_k(i)|^2]$ and $\rho_k(i-1) \approx \rho_k(i)$ when $i \rightarrow \infty$. Then, by taking expectations on both sides of (26) and letting i go to infinity, we can obtain the first-order statistical properties of $\rho_k(i)$:

$$E[\rho_k(\infty)] \approx \varepsilon_{\min}. \quad (27)$$

To study the second-order statistical properties of $\rho_k(i)$, we consider the square of (26), which is given by

$$\begin{aligned} \rho_k^2(i) &= \gamma^2 \rho_k^2(i-1) + (1-\gamma)^2 |e_k(i-1)|^2 |e_k(i)|^2 \\ &\quad + 2\gamma(1-\gamma) \rho_k(i-1) |e_k(i-1)| |e_k(i)|. \end{aligned} \quad (28)$$

Recall that $|e_k(i-1)|$ and $|e_k(i)|$ can be considered equivalent when $i \rightarrow \infty$, and thus, we can rewrite (28) as

$$\begin{aligned} \rho_k^2(i) &\approx \gamma^2 \rho_k^2(i-1) + (1-\gamma)^2 |e_k(i)|^4 \\ &\quad + 2\gamma(1-\gamma) \rho_k(i-1) |e_k(i)|^2. \end{aligned} \quad (29)$$

Since $(1-\gamma)^2 |e_k(i)|^4$ is sufficiently small when compared with other terms in (29), it can be neglected. Therefore, we can obtain

$$\rho_k^2(i) \approx \gamma^2 \rho_k^2(i-1) + 2\gamma(1-\gamma) \rho_k(i-1) |e_k(i)|^2. \quad (30)$$

According to (16) and (26), the quantities of $\rho_k(i-1)$ and $|e_k(i)|^2$ can be considered uncorrelated at steady state, that is to say, $E[\rho_k(i-1) |e_k(i)|^2] \approx E[\rho_k(i-1)] E[|e_k(i)|^2]$. Note that the detailed derivation is presented in Appendix A.

Table 1 LTVFF-DRLS and LCTVFF-DRLS algorithms

LTVFF-DRLS Algorithm		LCTVFF-DRLS Algorithm	
1	For each node $k = 1, 2, \dots, N$ do	1	For each node $k = 1, 2, \dots, N$ do
2	Initialize $\mathbf{w}_{k-1} = \mathbf{0}$, $\mathbf{P}_{k-1} = \mathbf{\Pi}^{-1}$.	2	Initialize $\mathbf{w}_{k-1} = \mathbf{0}$, $\mathbf{P}_{k-1} = \mathbf{\Pi}^{-1}$.
3	For time instant $i = 1, 2, \dots$ do	3	For time instant $i = 1, 2, \dots$ do
4	$\lambda_k(i) = [1 - \zeta_k(i)]_{\lambda_-}^{\lambda_+}$.	4	$\lambda_k(i) = [1 - \zeta_k(i)]_{\lambda_-}^{\lambda_+}$.
5	$\zeta_k(i) = \alpha \zeta_k(i-1) + \beta e_k(i) ^2$.	5	$\zeta_k(i) = \alpha \zeta_k(i-1) + \beta \rho_k(i) ^2$.
6	Set $\boldsymbol{\psi}_{k,i} = \mathbf{w}_{k,i-1}$, $\mathbf{P}_{k,i} = \lambda_k^{-1}(i) \mathbf{P}_{k,i-1}$.	6	$\rho_k(i) = \gamma \rho_k(i-1) + (1 - \gamma) e_k(i-1) e_k(i) $.
7	For $l \in \mathcal{N}_k$ do	7	Set $\boldsymbol{\psi}_{k,i} = \mathbf{w}_{k,i-1}$, $\mathbf{P}_{k,i} = \lambda_k^{-1}(i) \mathbf{P}_{k,i-1}$.
8	$\boldsymbol{\psi}_{k,i} \leftarrow \boldsymbol{\psi}_{k,i} + \frac{C_{l,k} \mathbf{P}_{k,i} \mathbf{u}_{l,i} [d_{l,i} - \mathbf{u}_{l,i}^* \boldsymbol{\psi}_{k,i}]}{\sigma_{v,i}^2 + C_{l,k} \mathbf{u}_{l,i}^* \mathbf{P}_{k,i} \mathbf{u}_{l,i}}$.	8	For $l \in \mathcal{N}_k$ do
9	$\mathbf{P}_{k,i} \leftarrow \mathbf{P}_{k,i} - \frac{C_{l,k} \mathbf{P}_{k,i} \mathbf{u}_{l,i} \mathbf{u}_{l,i}^* \mathbf{P}_{k,i}}{\sigma_{v,i}^2 + C_{l,k} \mathbf{u}_{l,i}^* \mathbf{P}_{k,i} \mathbf{u}_{l,i}}$.	9	$\boldsymbol{\psi}_{k,i} \leftarrow \boldsymbol{\psi}_{k,i} + \frac{C_{l,k} \mathbf{P}_{k,i} \mathbf{u}_{l,i} [d_{l,i} - \mathbf{u}_{l,i}^* \boldsymbol{\psi}_{k,i}]}{\sigma_{v,i}^2 + C_{l,k} \mathbf{u}_{l,i}^* \mathbf{P}_{k,i} \mathbf{u}_{l,i}}$.
10	End	10	$\mathbf{P}_{k,i} \leftarrow \mathbf{P}_{k,i} - \frac{C_{l,k} \mathbf{P}_{k,i} \mathbf{u}_{l,i} \mathbf{u}_{l,i}^* \mathbf{P}_{k,i}}{\sigma_{v,i}^2 + C_{l,k} \mathbf{u}_{l,i}^* \mathbf{P}_{k,i} \mathbf{u}_{l,i}}$.
11	Generate the final estimate $\mathbf{w}_{k,i} = \sum_{l \in \mathcal{N}_k} A_{l,k} \boldsymbol{\psi}_{l,i}$.	11	End
12	End	12	Generate the final estimate $\mathbf{w}_{k,i} = \sum_{l \in \mathcal{N}_k} A_{l,k} \boldsymbol{\psi}_{l,i}$.
13	End	13	End
		14	End

Then, by taking expectations on both sides of (30), we can obtain the following result:

$$E[\rho_k^2(\infty)] = \frac{2\gamma}{1+\gamma} E[\rho_k(\infty)] E[|e_k(\infty)|^2]. \quad (31)$$

Substituting (20) and (27) into (31) results in

$$E[\rho_k^2(\infty)] \approx \frac{2\gamma}{1+\gamma} \varepsilon_{\min}^2. \quad (32)$$

To calculate the first-order statistics of the adaptation component $\zeta_k(i)$, we take expectations on both sides of (25) and let i goes to infinity, as a result, we obtain

$$E[\zeta_k(\infty)] = \frac{\beta}{1-\alpha} E[\rho_k^2(\infty)]. \quad (33)$$

Substituting (32) into (33) leads to

$$E[\zeta_k(\infty)] = \frac{2\gamma\beta}{(1+\gamma)(1-\alpha)} \varepsilon_{\min}^2. \quad (34)$$

Consequently, we have the first-order steady-state statistics of the forgetting factor for the LCTVFF mechanism as follows:

$$E[\lambda_k(\infty)] = 1 - \frac{2\gamma\beta}{(1+\gamma)(1-\alpha)} \varepsilon_{\min}^2. \quad (35)$$

By employing the LCTVFF mechanism to the diffusion-based DRLS algorithm, we propose the LCTVFF-DRLS algorithm, which is presented in the right column of Table 1.

3.3 Computational complexity

In this part, we study the computational complexity of the proposed LTVFF and LCTVFF mechanisms in comparison with the GVFF mechanism. Generally, we evaluate the number of arithmetic operations in terms of complex

additions and multiplications for each node at each iteration. The results have been shown in Tables 2 and 3. From Table 3, the additional computational complexity of the proposed LTVFF and LCTVFF mechanisms is evaluated by fixed small values for each node at each iteration. However, for the GVFF mechanism, the additional computational complexity increases with the size of the sensor network for each node at each iteration. The result in Table 3 clearly reveals that the proposed LTVFF and LCTVFF mechanisms greatly reduce computational cost when compared to the GVFF mechanism.

4 Performance analysis

In this section, we carry out the analyses in terms of mean and mean square error performance for the proposed LTVFF-DRLS and LCTVFF-DRLS algorithms. In particular, we derive mathematical expressions to describe the steady-state behavior based on MSD and EMSE. In addition, we also perform transient analysis in a specialized case for the proposed algorithms in the last part of this section. To proceed with the analysis, we first introduce several assumptions, which have been widely adopted in the analysis for the RLS-type algorithms and have been verified by simulations [7, 27].

Assumption 1 To facilitate analytical studies, we assume that all the input vectors $\mathbf{u}_{k,i}$, $\forall k, i$ are independent of each other and the correlation matrix of the input vector $\mathbf{u}_{k,i}$ is invariant over time, which is defined as

$$E[\mathbf{u}_{k,i} \mathbf{u}_{k,i}^*] \mathbf{R}_{u_k}. \quad (36)$$

Assumption 2 For the proposed LTVFF and LCTVFF mechanisms, when i becomes large, we assume that there

Table 2 Computational complexity of the DRLS algorithm

	Number of operations for each node at each iteration	
	Multiplications	Additions
DRLS (fixed forgetting factor)	$M^2 + N^2(4M^2 + 5M + 1) + M$	$4N^2M^2 + M - 1$

exists a positive number N_i , when $i > N_i$, for which we have that the forgetting factor $\lambda_k(i)$ varies slowly around its mean value, that is

$$E\{\lambda_k(N_i)\} \simeq E\{\lambda_k(N_i + 1)\} \simeq \dots \simeq E\{\lambda_k(i)\} \simeq E\{\lambda_k(\infty)\}. \quad (37)$$

For the RLS-type algorithms with the fixed forgetting factor, we have the ergodicity assumption for $\mathbf{P}_{k,i}$ [6, 7, 27], that is, the time average of a sequence of random variables can be replaced by its expected value so as to make the analysis for the performance of these algorithms tractable. Similarly, for the RLS-type algorithms with variable forgetting factors, we still have the ergodicity assumption:

Assumption 3 We assume that there exists a number $N_i > 0$, when $i > N_i$, for which we can replace $\mathbf{P}_{k,i}^{-1}$ by its expected value $E[\mathbf{P}_{k,i}^{-1}]$, which can be represented as

$$\lim_{i \rightarrow \infty} \mathbf{P}_{k,i}^{-1} \approx \lim_{i \rightarrow \infty} E[\mathbf{P}_{k,i}^{-1}] \quad (38)$$

where $\lim_{i \rightarrow \infty} E[\mathbf{P}_{k,i}^{-1}]$ can be calculated through

$$\lim_{i \rightarrow \infty} E[\mathbf{P}_{k,i}^{-1}] = \frac{1}{1 - E[\lambda_k(\infty)]} \sum_{l=1}^N \frac{C_{l,k}}{\sigma_{v,l}^2} \mathbf{R}_{u_l}. \quad (39)$$

The derivation is presented in Appendix B. Since $\lim_{i \rightarrow \infty} E[\mathbf{P}_{k,i}^{-1}]$ is independent of i , we can denote it by \mathbf{P}_k^{-1} . Moreover, based on the ergodicity assumption, it is also common in the analysis of the performance of the RLS-type algorithms to replace the random matrix $\mathbf{P}_{k,i}$ by \mathbf{P}_k when i is large enough.

4.1 Mean performance

In light of (1) and (13), the following relation holds [7] after the incremental update of $\psi_{l,i}$ is complete:

$$\mathbf{P}_{l,i}^{-1} \psi_{l,i} = \lambda_l(i) \mathbf{P}_{l,i-1}^{-1} \mathbf{w}_{l,i-1} + \sum_{m=1}^N \frac{C_{m,l}}{\sigma_{v,m}^2} \mathbf{u}_{m,i} d_{m,i}. \quad (40)$$

By substituting (1) and (18) into (40), we obtain the following equation:

$$\mathbf{P}_{l,i}^{-1} (\psi_{l,i} - \mathbf{w}^o) = \lambda_l(i) \mathbf{P}_{l,i-1}^{-1} \tilde{\mathbf{w}}_{l,i-1} + \sum_{m=1}^N \frac{C_{m,l}}{\sigma_{v,m}^2} \mathbf{u}_{m,i} v_{m,i}. \quad (41)$$

Next, let us define the intermediate weight error vector $\tilde{\psi}_{k,i}$ for node k as

$$\tilde{\psi}_{k,i} = \psi_{k,i} - \mathbf{w}^o. \quad (42)$$

Substituting (42) into (41) results in the following result:

$$\tilde{\psi}_{l,i} = \lambda_l(i) \mathbf{P}_{l,i} \mathbf{P}_{l,i-1}^{-1} \tilde{\mathbf{w}}_{l,i-1} + \mathbf{P}_{l,i} \sum_{m=1}^N \frac{C_{m,l}}{\sigma_{v,m}^2} \mathbf{u}_{m,i} v_{m,i}. \quad (43)$$

Then, we construct $\tilde{\mathbf{w}}_{k,i}$ from $\tilde{\psi}_{l,i}$ based on (13) and obtain

$$\tilde{\mathbf{w}}_{k,i} = \sum_{l=1}^N A_{l,k} \left[\lambda_l(i) \mathbf{P}_{l,i} \mathbf{P}_{l,i-1}^{-1} \tilde{\mathbf{w}}_{l,i-1} + \mathbf{P}_{l,i} \sum_{m=1}^N \frac{C_{m,l}}{\sigma_{v,m}^2} \mathbf{u}_{m,i} v_{m,i} \right]. \quad (44)$$

Note that $\mathbf{P}_{k,i}$ can be replaced by \mathbf{P}_k when i is large enough (cf. Assumption 3), and thus, it is reasonable to assume that $\mathbf{P}_{k,i}$ converges as $i \rightarrow \infty$. Therefore, we can approximately have

$$\mathbf{P}_{k,i} \approx E[\mathbf{P}_{k,i}]. \quad (45)$$

Besides, in view of Assumption 3 and the Eq. (39), we can obtain

$$\mathbf{P}_{k,i} = (\mathbf{P}_{k,i}^{-1})^{-1} \approx \{E[\mathbf{P}_{k,i}^{-1}]\}^{-1} \approx (1 - E[\lambda_k(\infty)]) \left(\sum_{l=1}^N \frac{C_{l,k}}{\sigma_{v,l}^2} \mathbf{R}_{u_l} \right)^{-1}. \quad (46)$$

By combining (45) and (46), we have the following approximation:

$$\mathbf{P}_{k,i} \mathbf{P}_{k,i-1}^{-1} \approx E^{-1}[\mathbf{P}_{k,i}^{-1}] E[\mathbf{P}_{k,i}^{-1}] = \mathbf{I}_M. \quad (47)$$

Table 3 Additional computational complexity of the analyzed VFF mechanisms

	Number of operations for each node at each iteration	
	Multiplications	Additions
GVFF	$M^2 + N^2(9M^2 + 4M) + 2M + 1$	$M^2 + N^2(9M^2 - 3M - 1) + 2M - 2$
LTVFF	3	2
LCTVFF	6	3

Then, substituting (47) into (44) yields the following result when i is sufficiently large:

$$\tilde{\mathbf{w}}_{k,i} = \sum_{l=1}^N A_{l,k} \left[\lambda_l(i) \tilde{\mathbf{w}}_{l,i-1} + \mathbf{P}_{l,i} \sum_{m=1}^N \frac{C_{m,l}}{\sigma_{v,m}^2} \mathbf{u}_{m,i} v_{m,i} \right]. \quad (48)$$

Following this, two global matrices $\tilde{\mathbf{W}}_i$ and \mathcal{P} are built in the following form in order to collect the weight error vectors $\tilde{\mathbf{w}}_{k,i}$, $k = 1, \dots, N$ and matrices \mathbf{P}_k , $k = 1, \dots, N$, respectively:

$$\begin{aligned} \tilde{\mathbf{W}}_i &= \text{row}\{\tilde{\mathbf{w}}_{1,i}, \tilde{\mathbf{w}}_{2,i}, \dots, \tilde{\mathbf{w}}_{N,i}\} \\ \mathcal{P} &= \text{row}\{\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_N\}. \end{aligned} \quad (49)$$

In addition, we introduce a global diagonal matrix $\mathbf{D}(i)$ to collect the forgetting factors of all nodes at time instant i , which is given by

$$\mathbf{\Lambda}_i = \text{diag}\{\lambda_1(i), \lambda_2(i), \dots, \lambda_N(i)\}. \quad (50)$$

Using the vectors in (2), the term $\sum_{m=1}^N \frac{C_{m,l}}{\sigma_{v,m}^2} \mathbf{u}_{m,i} v_{m,i}$ in (44) can be rewritten as $\mathbf{H}_i^* \mathbf{C}_l \mathbf{R}_v^{-1} \mathbf{v}_i$. By collecting the vectors $\mathbf{H}_i^* \mathbf{C}_l \mathbf{R}_v^{-1} \mathbf{v}_i$, $l = 1, 2, \dots, N$, into a block diagonal matrix \mathbf{G}_i , we obtain

$$\mathbf{G}_i = \text{diag}\{\mathbf{H}_i^* \mathbf{C}_1 \mathbf{R}_v^{-1} \mathbf{v}_i, \mathbf{H}_i^* \mathbf{C}_2 \mathbf{R}_v^{-1} \mathbf{v}_i, \dots, \mathbf{H}_i^* \mathbf{C}_N \mathbf{R}_v^{-1} \mathbf{v}_i\}. \quad (51)$$

To separate the noise vectors, we can rewrite (51) as

$$\mathbf{G}_i = \text{diag}\{\mathbf{H}_i^* \mathbf{C}_1 \mathbf{R}_v^{-1}, \mathbf{H}_i^* \mathbf{C}_2 \mathbf{R}_v^{-1}, \dots, \mathbf{H}_i^* \mathbf{C}_N \mathbf{R}_v^{-1}\} (\mathbf{I}_N \otimes \mathbf{v}_i). \quad (52)$$

where \otimes denotes the Kronecker product of two matrices [28]. Subsequently, we express (48) in a more compact way, which leads to the following updating equation for the global matrix $\tilde{\mathbf{W}}_i$:

$$\tilde{\mathbf{W}}_i = \tilde{\mathbf{W}}_{i-1} \mathbf{\Lambda}_i \mathbf{A} + \mathcal{P} \mathbf{G}_i \mathbf{A}. \quad (53)$$

In order to simplify the notation $\mathbf{\Lambda}_i \mathbf{A}$, we denote it as $\mathbf{F}(i)$, and thus, we can rewrite (53) as

$$\tilde{\mathbf{W}}_i = \tilde{\mathbf{W}}_{i-1} \mathbf{F}(i) + \mathcal{P} \mathbf{G}_i \mathbf{A}. \quad (54)$$

In order to facilitate analysis, we assume that $\tilde{\mathbf{W}}_{i-1}$ and $\mathbf{F}(i)$ can be considered uncorrelated, that is, $E[\tilde{\mathbf{W}}_{i-1} \mathbf{F}(i)] \approx E[\tilde{\mathbf{W}}_{i-1}] E[\mathbf{F}(i)]$. As we will see in simulation results, this assumption works well for theoretical analysis, which matches numerical results perfectly. By taking expectations on both sides of (54), we obtain the following result:

$$E[\tilde{\mathbf{W}}_i] = E[\tilde{\mathbf{W}}_{i-1}] E[\mathbf{F}(i)] + \mathcal{P} E[\mathbf{G}_i] \mathbf{A}. \quad (55)$$

Recall (52), since the noise samples \mathbf{v}_i have zero mean, $E[\mathbf{G}_i]$ equals to zero; therefore, we can obtain

$$E[\tilde{\mathbf{W}}_i] = E[\tilde{\mathbf{W}}_{i-1}] E[\mathbf{F}(i)]. \quad (56)$$

Following this, we assume that there exists a number $N_i > 0$ and iterate (56) starting from the time instant i to N_i , as a result, we obtain

$$E[\tilde{\mathbf{W}}_i] = E[\tilde{\mathbf{W}}_{N_i}] \prod_{j=N_i+1}^i E[\mathbf{F}(j)]. \quad (57)$$

Recalling that $\mathbf{F}(i) = \mathbf{\Lambda}_i \mathbf{A}$, with $\mathbf{\Lambda}_i$ a diagonal matrix, we have the following relation for each element in $\mathbf{F}(i)$:

$$\mathbf{F}_{m,n}(i) = \lambda_m(i) \mathbf{A}_{m,n}(i), \forall m, n \in \{1, 2, \dots, N\} \quad (58)$$

where the subscript m, n represents the (m, n) -th element in the matrix. Given that the elements of \mathbf{A} are all between 0 and 1 and each element in the diagonal matrix $\mathbf{\Lambda}_i$ does not exceed the upper bound λ_+ , which is smaller than unity, we have

$$\mathbf{F}_{m,n}(i) = \lambda_m(i) \mathbf{A}_{m,n}(i) < \lambda_+ \mathbf{A}_{m,n}(i) < 1, \forall m, n \in \{1, 2, \dots, N\} \quad (59)$$

Each element in the product $\prod_{j=N_i+1}^i E[\mathbf{F}(j)]$ can be viewed as a polynomial of $\mathbf{F}_{1,1}(i), \mathbf{F}_{1,2}(i), \dots$, with an order of $i - N_i + 1$. When $i \rightarrow \infty$, each element of this product approaches zero since $\mathbf{F}_{1,1}(i), \mathbf{F}_{1,2}(i), \dots$ are all smaller than unity. Now, assuming that all the elements of $E[\tilde{\mathbf{W}}_{N_i}]$ are bounded in absolute value by some finite constant, therefore, all the elements of $E[\tilde{\mathbf{W}}_i]$ converge to zero when $i \rightarrow \infty$. As a result, we can conclude that the proposed LTVFF-DRLS and LCTVFF-DRLS algorithms converge asymptotically when $i \rightarrow \infty$.

4.2 Mean-square error and deviation performances

In this part, we perform analyses for the proposed LTVFF-DRLS and LCTVFF-DRLS algorithms based on mean square performance and derive expressions for the steady-state MSD and EMSE, which are defined as

$$\begin{aligned} \text{MSD}_k^{ss} &= \lim_{i \rightarrow \infty} E[\|\tilde{\mathbf{w}}_{k,i}\|^2] \\ \text{EMSE}_k^{ss} &= \lim_{i \rightarrow \infty} E[|\mathbf{u}_{k,i}^* \tilde{\mathbf{w}}_{k,i-1}|^2]. \end{aligned} \quad (60)$$

We start with (54) and then operate recursively from time instant N_i , which yields

$$\tilde{\mathbf{W}}_i = \tilde{\mathbf{W}}_{N_i} \prod_{j=N_i+1}^i \mathbf{F}(j) + \mathcal{P} \sum_{t=N_i+1}^i \mathbf{G}_t \mathbf{A} \prod_{j=t+1}^i \mathbf{F}(j). \quad (61)$$

Then, the k th column of $\tilde{\mathbf{W}}_i$ is given by

$$\tilde{\mathbf{w}}_{k,i} = \tilde{\mathbf{w}}_{k,N_i} \prod_{j=N_i+1}^i \mathbf{F}(j) \mathbf{e}_k + \mathcal{P} \sum_{t=N_i+1}^i \mathbf{G}_t \mathbf{A} \prod_{j=t+1}^i \mathbf{F}(j) \mathbf{e}_k \quad (62)$$

where \mathbf{e}_k is a column vector of length N with unity for the k th element and zero for the others. Next, we write the Euclidean norm of the weight error vector $\tilde{\mathbf{w}}_{k,i}$, that is, $\|\tilde{\mathbf{w}}_{k,i}\|^2$, or equivalently, $\text{Tr}\{\tilde{\mathbf{w}}_{k,i}\tilde{\mathbf{w}}_{k,i}^*\}$.

Since the elements of $\mathbf{F}(i)$ are all bounded by zero and one, $\prod_{j=N_i+1}^i \mathbf{F}(j)$ vanishes when $i \rightarrow \infty$, which leads to the expectation of the first term becoming zero. Moreover, seeing that the cross terms incorporate the zero-mean vectors \mathbf{v}_i , their expectations also become zero. As a result, we have the following expression:

$$E[\|\tilde{\mathbf{w}}_{k,i}\|^2] = E\left[\left\|\mathcal{P} \sum_{t=N_i+1}^i \mathbf{G}_t \mathbf{A} \prod_{j=t+1}^i \mathbf{F}(j) \mathbf{e}_k\right\|^2\right] \quad (63)$$

which can be rewritten as

$$E[\|\tilde{\mathbf{w}}_{k,i}\|^2] = E\left[\text{Tr}\left\{\mathcal{P} \sum_{t=N_i+1}^i \mathbf{G}_t \mathbf{A} \prod_{j=t+1}^i \mathbf{F}(j) \mathbf{e}_k \mathbf{e}_k^* \sum_{l=N_i+1}^i \left(\prod_{j=l+1}^i \mathbf{F}(j)\right)^* \mathbf{A}^* \mathbf{G}_l^* \mathcal{P}^*\right\}\right]. \quad (64)$$

For simplicity, we have the following notation:

$$\mathbf{J}^{t,l}(i) = \mathbf{A} \prod_{j=t+1}^i \mathbf{F}(j) \mathbf{e}_k \mathbf{e}_k^* \left(\prod_{j=l+1}^i \mathbf{F}(j)\right)^* \mathbf{A}^* \quad (65)$$

where $\mathbf{J}^{t,l}(i)$ is a matrix of size $N \times N$. By combining (52), (64), and (65), let us first compute $(\mathbf{I}_N \otimes \mathbf{v}_t) \mathbf{J}^{t,l}(i) (\mathbf{I}_N \otimes \mathbf{v}_l^*)$. According to the properties of the Kronecker product, we have the following equality:

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}. \quad (66)$$

Therefore, $(\mathbf{I}_N \otimes \mathbf{v}_t) \mathbf{J}^{t,l}(i) (\mathbf{I}_N \otimes \mathbf{v}_l^*)$ can be expressed as

$$\begin{aligned} (\mathbf{I}_N \otimes \mathbf{v}_t) \mathbf{J}^{t,l}(i) (\mathbf{I}_N \otimes \mathbf{v}_l^*) &= (\mathbf{I}_N \otimes \mathbf{v}_t) \left(\mathbf{J}^{t,l}(i) \otimes \mathbf{1}\right) (\mathbf{I}_N \otimes \mathbf{v}_l^*) \\ &= \mathbf{J}^{t,l}(i) \otimes (\mathbf{v}_t \mathbf{v}_l^*). \end{aligned} \quad (67)$$

Note that, in light of (65), the matrix $\mathbf{J}^t(i)$ and the covariance matrix of noise \mathbf{R}_v can be considered uncorrelated. Then, by taking expectations on both sides of (67), we have the following results:

$$\begin{aligned} E\left[(\mathbf{I}_N \otimes \mathbf{v}_t) \mathbf{J}^{t,l}(i) (\mathbf{I}_N \otimes \mathbf{v}_l^*)\right] &= E\left[\mathbf{J}^{t,l}(i)\right] \otimes E\left[(\mathbf{v}_t \mathbf{v}_l^*)\right] \\ &= \begin{cases} E[\mathbf{J}^t(i)] \otimes \mathbf{R}_v & t = l, \\ 0 & t \neq l, \end{cases} \end{aligned} \quad (68)$$

where we drop the index and denote $\mathbf{J}^{t,t}(i)$ as $\mathbf{J}^t(i)$. By substituting (68) into (64), we can obtain

$$E[\|\tilde{\mathbf{w}}_{k,i}\|^2] = E\left[\text{Tr}\left\{\mathcal{P} \sum_{t=N_i+1}^i \mathbf{G}_t \mathbf{J}^t(i) \mathbf{G}_t^* \mathcal{P}^*\right\}\right]. \quad (69)$$

Note that $\mathbf{P}_k, k = 1, 2, \dots, N$ is Hermitian; therefore, we have the following expression:

$$E[\|\tilde{\mathbf{w}}_{k,i}\|^2] = E\left[\text{Tr}\left\{\mathcal{P} \sum_{t=N_i+1}^i \mathbf{G}_t \mathbf{J}^t(i) \mathbf{G}_t^* \mathcal{P}^T\right\}\right] \quad (70)$$

where $\mathbf{G}_t \mathbf{J}^t(i) \mathbf{G}_t^*$ can be represented as a block matrix $\mathbf{K}^t(i)$, which can be decomposed into $N \times N$ blocks of size $M \times M$ each. The (m, l) th block is given by

$$\mathbf{K}_{m,l}^t(i) = \mathbf{H}_t^* \mathbf{C}_m \mathbf{R}_v^{-1} \mathbf{J}_{m,l}^t(i) \mathbf{v}_t \mathbf{v}_t^* \mathbf{C}_l^* \mathbf{R}_v^{-1} \mathbf{H}_t. \quad (71)$$

By taking expectations on both sides of (71), we obtain the following equality:

$$E[\mathbf{K}_{m,l}^t(i)] = E[\mathbf{J}_{m,l}^t(i)] \sum_{n=1}^N \frac{C_{n,m} C_{n,l}}{\sigma_{v,n}^2} \mathbf{R}_{u_n}. \quad (72)$$

Substituting (65) and (72) into (70) yields the following result:

$$\begin{aligned} E[\|\tilde{\mathbf{w}}_{k,i}\|^2] &= \text{Tr}\left\{\sum_{t=N_i+1}^i \sum_{l=1}^N \sum_{m=1}^N \mathbf{P}_m E[\mathbf{K}_{m,l}^t(i)] \mathbf{P}_l\right\} \\ &= \sum_{t=N_i+1}^i \sum_{l=1}^N \sum_{m=1}^N \sum_{n=1}^N \text{Tr}\{\mathbf{P}_m \mathbf{R}_{u_n} \mathbf{P}_l\} \frac{C_{n,m} C_{n,l}}{\sigma_{v,n}^2} \\ &\quad \times \left\{\mathbf{A} \prod_{j=t+1}^i E[\mathbf{F}(j)]\right\}_{m,k} \left\{\mathbf{A} \prod_{j=l+1}^i E[\mathbf{F}(j)]\right\}_{l,k}. \end{aligned} \quad (73)$$

In view of Assumption 2, we can verify that there exists a number $N_i > 0$, when $i > N_i$, for which $\mathbf{F}(i)$ satisfies

$$E[\mathbf{F}(N_i)] \simeq E[\mathbf{F}(N_i + 1)] \simeq \dots \simeq E[\mathbf{F}(i)] \simeq E[\mathbf{F}(\infty)]. \quad (74)$$

Therefore, we replace $E[\mathbf{F}(i)]$ with $E[\mathbf{F}(\infty)]$ when $i > N_i$ and then reformulate (73) as

$$\begin{aligned} E[\|\tilde{\mathbf{w}}_{k,i}\|^2] &\approx \sum_{t=N_i+1}^i \sum_{l=1}^N \sum_{m=1}^N \sum_{n=1}^N \text{Tr}\{\mathbf{P}_m \mathbf{R}_{u_n} \mathbf{P}_l\} \\ &\quad \times \frac{C_{n,m} C_{n,l}}{\sigma_{v,n}^2} \left\{\mathbf{A} E^{i-t}[\mathbf{F}(\infty)]\right\}_{m,k} \left\{\mathbf{A} E^{i-l}[\mathbf{F}(\infty)]\right\}_{l,k}. \end{aligned} \quad (75)$$

Subsequently, we replace $i - t$ with t in (75) and then let i goes to infinity. As a result, we can obtain the expression of the steady-state MSD for node k :

$$\begin{aligned} MSD_k^{ss} &= \lim_{i \rightarrow \infty} E[\|\tilde{\mathbf{w}}_{k,i}\|^2] \\ &= \lim_{i \rightarrow \infty} \sum_{t=0}^i \sum_{l=1}^N \sum_{m=1}^N \sum_{n=1}^N Tr\{\mathbf{P}_m \mathbf{R}_{u_n} \mathbf{P}_l\} \\ &\quad \times \frac{C_{n,m} C_{n,l}}{\sigma_{v,n}^2} \{A E^t[\mathbf{F}(\infty)]\}_{m,k} \{A E^t[\mathbf{F}(\infty)]\}_{l,k}. \end{aligned} \quad (76)$$

Next, we calculate the steady-state EMSE for node k . According to (60), the EMSE for node k can be expressed as follows

$$\begin{aligned} E[|\mathbf{u}_{k,i}^* \tilde{\mathbf{w}}_{k,i-1}|^2] &= E[Tr\{\tilde{\mathbf{w}}_{k,i-1}^* \mathbf{u}_{k,i} \mathbf{u}_{k,i}^* \tilde{\mathbf{w}}_{k,i-1}\}] \\ &= E[Tr\{\mathbf{u}_{k,i} \mathbf{u}_{k,i}^* \tilde{\mathbf{w}}_{k,i-1} \tilde{\mathbf{w}}_{k,i-1}^*\}] \\ &= Tr\{\mathbf{R}_{u_k} E[\tilde{\mathbf{w}}_{k,i-1} \tilde{\mathbf{w}}_{k,i-1}^*]\}. \end{aligned} \quad (77)$$

Note that $\mathbf{u}_{k,i}$ is independent of $\tilde{\mathbf{w}}_{k,i-1}$. By substituting (76) into (77), we can obtain the expression of the steady-state EMSE for node k :

$$\begin{aligned} EMSE_k^{ss} &= \lim_{i \rightarrow \infty} \sum_{t=0}^i \sum_{l=1}^N \sum_{m=1}^N \sum_{n=1}^N Tr\{\mathbf{R}_{u_k} \mathbf{P}_m \mathbf{R}_{u_n} \mathbf{P}_l\} \\ &\quad \times \frac{C_{n,m} C_{n,l}}{\sigma_{v,n}^2} \{A E^t[\mathbf{F}(\infty)]\}_{m,k} \{A E^t[\mathbf{F}(\infty)]\}_{l,k}. \end{aligned} \quad (78)$$

Expressions (76) and (78) describe the steady-state behavior of the proposed LTVFF-DRLS and LCTVFF-DRLS algorithms. By comparing the expressions (76) and (78) with the analytical results in [7], it is clear that the fixed matrix $\lambda^2 \mathbf{A}$ in the expressions for the conventional DRLS algorithms has been replaced by the matrix $\mathbf{F}(i)$ in the expressions (76) and (78), which is weighted by the matrix $\mathbf{\Lambda}_i$. Since $\mathbf{\Lambda}_i$ varies from one iteration to the next, $\mathbf{F}(i)$ varies for each iteration as well, which improves the tracking performance of the resulting algorithms. Furthermore, since all the elements in $\mathbf{F}(i)$ are bounded by zero and unity, the values of the steady-state MSD and EMSE given by (76) and (78) are both very small values when i is large enough. Thus, we can verify that the proposed LTVFF-DRLS and LCTVFF-DRLS algorithms both converge in the mean-square sense.

4.3 Transient analysis under spatial invariance assumption

In this subsection, we consider a specialized case that the noise variances and input vector covariance matrices are the same for all the sensor nodes, and provide

transient analysis for this specific case. Particularly, we assume spatial invariance:

$$\sigma_{v_1}^2 = \sigma_{v_2}^2 = \dots = \sigma_{v_N}^2 = \sigma_v^2 \quad (79)$$

$$\mathbf{R}_{u_1} = \mathbf{R}_{u_2} = \dots = \mathbf{R}_{u_N} = \mathbf{R}_u. \quad (80)$$

In addition, to facilitate analysis, we assume that all elements of the adaptation matrix \mathbf{C} are equal to $\frac{1}{N}$.

We study the transient analysis through focusing on the learning curve, which is obtained by depicting the squared priori estimation error, i.e., $E[|\mathbf{u}_{k,i}^* (\mathbf{w}_{k,i} - \mathbf{w}^o)|^2]$ [29, 30], as a function of the iteration number i . We first rewrite this squared priori estimation error in a more compact form:

$$\begin{aligned} &E[|\mathbf{u}_{k,i}^* (\mathbf{w}_{k,i} - \mathbf{w}^o)|^2] \\ &= E[|\mathbf{u}_{k,i}^* \tilde{\mathbf{w}}_{k,i}|^2] \\ &= E[\tilde{\mathbf{w}}_{k,i}^* \mathbf{u}_{k,i} \mathbf{u}_{k,i}^* \tilde{\mathbf{w}}_{k,i}] \\ &= E[\tilde{\mathbf{w}}_{k,i}^* \mathbf{R}_u \tilde{\mathbf{w}}_{k,i}] \\ &= E[\|\tilde{\mathbf{w}}_{k,i}\|_{\mathbf{R}_u}^2] \end{aligned} \quad (81)$$

where we use the representation $\|\mathbf{t}\|_{\mathbf{A}}^2 = \mathbf{t}^* \mathbf{A} \mathbf{t}$ in the last equality.

Then, we use the spatial invariance assumption to simplify (39) and (48). Particularly, by taking advantage of the assumption that the input vector covariance matrix is the same over all sensor nodes, we can derive the following expression from (39), when i is large enough:

$$\mathbf{P}_{k,i} \approx E[\mathbf{P}_{k,i}^{-1}]^{-1} \approx (1 - E[\lambda_k(i)]) \sigma_v^2 \mathbf{R}_u^{-1} \approx (1 - \lambda_k(i)) \sigma_v^2 \mathbf{R}_u^{-1}. \quad (82)$$

By substituting (82) into (48), we can arrive at

$$\begin{aligned} \tilde{\mathbf{w}}_{k,i} &= \sum_{l=1}^N A_{l,k} \left[\lambda_l(i) \tilde{\mathbf{w}}_{l,i} + (1 - \lambda_l(i)) \mathbf{R}_u^{-1} \sum_{m=1}^N \mathbf{u}_{m,i} v_{m,i} \right] \\ &= \sum_{l=1}^N A_{l,k} \lambda_l(i) \tilde{\mathbf{w}}_{l,i} + \sum_{l=1}^N A_{l,k} (1 - \lambda_l(i)) \mathbf{R}_u^{-1} \mathbf{H}_i^* \mathbf{v}_i \\ &= \sum_{l=1}^N A_{l,k} \lambda_l(i) \tilde{\mathbf{w}}_{l,i} + \sum_{l=1}^N A_{l,k} (1 - \lambda_l(i)) \mathbf{s}_i \\ &= \sum_{l=1}^N A_{l,k} \lambda_l(i) \tilde{\mathbf{w}}_{l,i} + \left(1 - \sum_{l=1}^N A_{l,k} \lambda_l(i) \right) \mathbf{s}_i \end{aligned} \quad (83)$$

where we use the column vector \mathbf{s}_i to denote the quantity $\mathbf{R}_u^{-1} \mathbf{H}_i^* \mathbf{v}_i$ in the third equality, and we use the property of the combination matrix, i.e., $\sum_{l=1}^N A_{l,k} = 1, \forall k \in \{1, 2, \dots, N\}$, to arrive at the fourth equality. Let us define

$$\begin{aligned} \tilde{\mathbf{W}}_i &= \text{col}\{\tilde{\mathbf{w}}_{1,i}, \tilde{\mathbf{w}}_{2,i}, \dots, \tilde{\mathbf{w}}_{N,i}\} \\ \lambda_i &= \text{col}\{\lambda_1(i), \lambda_2(i), \dots, \lambda_N(i)\}. \end{aligned} \quad (84)$$

Note that $\Lambda_i = \text{diag}\{\lambda_i\}$. Then, we can write the recursive equation of type (83) for all sensor nodes in a more compact form as follows:

$$\begin{aligned}\tilde{\mathbf{W}}_i &= \mathbf{A}^T \Lambda_i \tilde{\mathbf{W}}_{i-1} + (\mathbb{I} - \mathbf{A}^T \Lambda_i) \otimes \mathbf{s}_i \\ &= \mathbf{A}^T \Lambda_i \tilde{\mathbf{W}}_{i-1} + \mathbf{f}(i) \otimes \mathbf{s}_i\end{aligned}\quad (85)$$

where $\mathbf{f}(i) = \mathbb{I} - \mathbf{A}^T \Lambda_i$ in the second equality. Then, we have the following global squared priori estimation error for all sensor nodes by using the last equality in (85):

$$\begin{aligned}E[\|\tilde{\mathbf{W}}_i\|_{\mathbf{R}_u}^2] &= E[\tilde{\mathbf{W}}_i^* \mathbf{R}_u \tilde{\mathbf{W}}_i] \\ &= E[\tilde{\mathbf{W}}_{i-1}^* \Lambda_i \mathbf{A} \mathbf{R}_u \mathbf{A}^T \Lambda_i \tilde{\mathbf{W}}_{i-1} + (\mathbf{f}(i)^T \otimes \mathbf{s}_i^*) \mathbf{R}_u (\mathbf{f}(i) \otimes \mathbf{s}_i)] \\ &= E[\|\tilde{\mathbf{W}}_{i-1}\|_{\Sigma}^2] + E[(\mathbf{f}(i)^T \otimes \mathbf{s}_i^*) (\mathbf{R}_u \otimes \mathbb{I}) (\mathbf{f}(i) \otimes \mathbf{s}_i)] \\ &= E[\|\tilde{\mathbf{W}}_{i-1}\|_{\Sigma}^2] + E[(\mathbf{f}(i)^T \mathbf{R}_u) \otimes \mathbf{s}_i^* (\mathbf{f}(i) \otimes \mathbf{s}_i)] \\ &= E[\|\tilde{\mathbf{W}}_{i-1}\|_{\Sigma}^2] + E[(\mathbf{f}(i)^T \mathbf{R}_u \mathbf{f}(i)) \otimes (\mathbf{s}_i^* \mathbf{s}_i)] \\ &= E[\|\tilde{\mathbf{W}}_{i-1}\|_{\Sigma}^2] + E[\mathbf{f}(i)^T \mathbf{R}_u \mathbf{f}(i)] E[\mathbf{s}_i^* \mathbf{s}_i]\end{aligned}\quad (86)$$

where $\Sigma = \Lambda_i \mathbf{A} \mathbf{R}_u \mathbf{A}^T \Lambda_i$, and we use the property of the Kronecker product, i.e., (66), in the fourth and fifth equalities, and the fact that both quantities of $\mathbf{f}(i)^T \mathbf{R}_u \mathbf{f}(i)$ and $\mathbf{s}_i^* \mathbf{s}_i$ are scalar and they are independent to arrive at the last equality. Particularly, $E[\mathbf{s}_i^* \mathbf{s}_i]$ can be rewritten as

$$\begin{aligned}E[\mathbf{s}_i^* \mathbf{s}_i] &= E[\text{Tr}(\mathbf{v}_i^* \mathbf{H}_i (\mathbf{R}_u^{-1})^* \mathbf{R}_u^{-1} \mathbf{H}_i^* \mathbf{v}_i)] \\ &= E[\text{Tr}(\mathbf{H}_i (\mathbf{R}_u^{-1})^* \mathbf{R}_u^{-1} \mathbf{H}_i^* \mathbf{v}_i \mathbf{v}_i^*)] \\ &= \sigma_v^2 E[\text{Tr}((\mathbf{R}_u^{-1})^* \mathbf{R}_u^{-1} \mathbf{H}_i^* \mathbf{H}_i)] \\ &= N \sigma_v^2 \text{Tr}((\mathbf{R}_u^{-1})^* \mathbf{R}_u^{-1} \mathbf{R}_u) \\ &= N \sigma_v^2 \text{Tr}(\mathbf{R}_u^{-1})\end{aligned}\quad (87)$$

where we use the spatial invariance assumption, i.e., $\mathbf{v}_i \mathbf{v}_i^* = \text{diag}\{\sigma_v^2, \sigma_v^2, \dots, \sigma_v^2\} = \sigma_v^2 \mathbf{I}_N$ and $\mathbf{H}_i^* \mathbf{H}_i = \sum_{m=1}^N \mathbf{u}_{m,i} \mathbf{u}_{m,i}^* = \sum_{m=1}^N \mathbf{R}_{u_m} = N \mathbf{R}_u$, to arrive at the third and fourth equalities, respectively, and the symmetry of the input vector covariance matrix in the last equality. By plugging (87) back into (86), we have

$$\begin{aligned}E[\|\tilde{\mathbf{W}}_i\|_{\mathbf{R}_u}^2] &= E[\|\tilde{\mathbf{W}}_{i-1}\|_{\Sigma}^2] + N \sigma_v^2 E[\text{Tr}(\mathbf{f}(i)^T \mathbf{R}_u \mathbf{f}(i) \mathbf{R}_u^{-1})] \\ &= E[\|\tilde{\mathbf{W}}_{i-1}\|_{E[\Sigma]}^2] + N \sigma_v^2 E[\text{Tr}(\mathbf{f}(i)^T \mathbf{R}_u \mathbf{f}(i) \mathbf{R}_u^{-1})].\end{aligned}\quad (88)$$

For convenience, we use the notation $\|\mathbf{t}\|_{\text{vec}\{\mathbf{A}\}}^2$ to denote the weighted norm $\|\mathbf{t}\|_{\mathbf{A}}^2$, where the symbol $\text{vec}\{\mathbf{A}\}$ represents the vectorization of a matrix. Particularly, by using

the equality $\text{vec}\{\mathbf{ABC}\} = (\mathbf{C}^T \otimes \mathbf{A}) \text{vec}\{\mathbf{B}\}$, we can vectorize the matrix $\Sigma = \Lambda_i \mathbf{A} \mathbf{R}_u \mathbf{A}^T \Lambda_i$ as follows

$$\begin{aligned}\text{vec}\{\Sigma\} &= \text{vec}\{\Lambda_i \mathbf{A} \mathbf{R}_u \mathbf{A}^T \Lambda_i\} \\ &= ((\mathbf{A}^T \Lambda_i) \otimes (\Lambda_i \mathbf{A})) \text{vec}\{\mathbf{R}_u\} \\ &= ((\Lambda_i \mathbf{A}) \otimes (\Lambda_i \mathbf{A})) \text{vec}\{\mathbf{R}_u\} \\ &= (\mathbf{F}(i) \otimes \mathbf{F}(i)) \text{vec}\{\mathbf{R}_u\} \\ &= \mathcal{F}_i \boldsymbol{\gamma}\end{aligned}\quad (89)$$

where $\mathbf{F}(i) = \Lambda_i \mathbf{A}$, $\mathcal{F}_i = \mathbf{F}(i) \otimes \mathbf{F}(i)$ and $\boldsymbol{\gamma} = \text{vec}\{\mathbf{R}_u\}$. Ultimately, we have

$$\begin{aligned}E[\|\tilde{\mathbf{W}}_i\|_{\boldsymbol{\gamma}}^2] &= E[\|\tilde{\mathbf{W}}_{i-1}\|_{E[\mathcal{F}_i] \boldsymbol{\gamma}}^2] + N \sigma_v^2 E[\text{Tr}(\mathbf{f}(i)^T \mathbf{R}_u \mathbf{f}(i) \mathbf{R}_u^{-1})].\end{aligned}\quad (90)$$

This recursive equation is stable and convergent if $E[\mathcal{F}_i]$ is stable [31].

Particularly, the quantity \mathcal{F}_i has a spectral radius smaller than unity and thus is stable. This can be proved as follows: If we replace each element in Λ_i by its upper bound λ_+ , then we have \mathcal{F}_i replaced by $\lambda_+^2 \mathbf{A} \otimes \mathbf{A}$. Note that \mathbf{A} satisfies $\mathbb{I}^T \mathbf{A} = \mathbb{I}$, and then, we can readily verify that each column of $\mathbf{A} \otimes \mathbf{A}$ sums up to unity. Hence, the quantity $\lambda_+^2 \mathbf{A} \otimes \mathbf{A}$ has the spectral radius λ_+^2 that is smaller than one. Given that each element in Λ_i does not exceed λ_+ , the spectral radius of \mathcal{F}_i is smaller than λ_+^2 and surely is smaller than unity. Therefore, for this specialized case, it can be verified theoretically that the proposed LTVFF-DRLS and LCTVFF-DRLS algorithms are convergent in terms of the learning curve and the convergence rate is related to the varying forgetting factors.

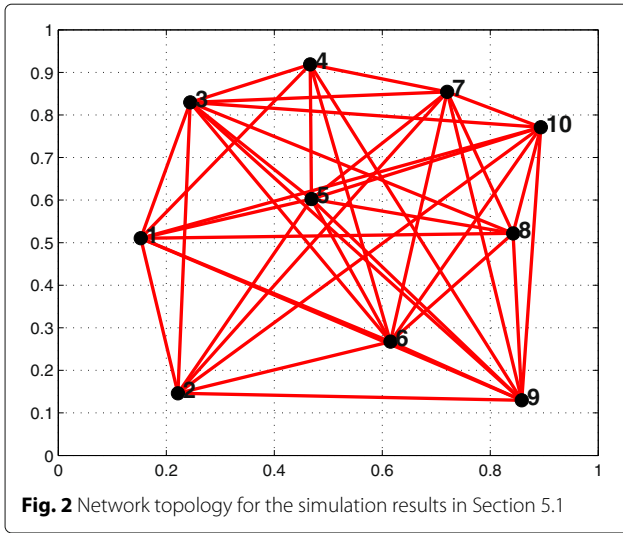
Also note that, since the convergence performance of the adaptive algorithms does not depend on the outside environment but rely on the network topology and the design of algorithms, the analytical results in this specialized case also apply to the general case.

5 Simulation results

In this section, we present the simulation results for the proposed LTVFF-DRLS and LCTVFF-DRLS algorithms when applied in two applications, that is, distributed parameter estimation and distributed spectrum estimation over sensor networks.

5.1 Distributed parameter estimation

In this part, we evaluate the performance of the proposed LTVFF-DRLS and LCTVFF-DRLS algorithms when applied to distributed parameter estimation in comparison with the DRLS algorithm with the fixed forgetting



factor and the GVFF-DRLS algorithm. In addition, we also verify the effectiveness of the proposed analytical expressions in (76) and (78) based on simulations.

We assume that there are 10 nodes in the sensor network and the length of the unknown weight vector is $M = 5$. The input vectors $\mathbf{u}_{k,i}$, $k = 1, 2, \dots, N$ are assumed to be Gaussian with zero means and variances $\{\sigma_{u,k}^2\}$ chosen randomly between 1 and 2 for each node. The Gaussian noise samples $v_{k,i}$, $k = 1, 2, \dots, N$ have variances $\{\sigma_{v,k}^2\}$ that are chosen randomly between 0.1 and 0.2 for each node. We generate the measurements $\{d_{k,i}\}$ according to (1). Simulation results are averaged over 100 experiments. The adaptation matrix \mathbf{C} is governed by the Metropolis

rule, while the choice of the diffusion matrix \mathbf{A} follows the relative-degree rule [8]. The network topology used for the simulations is shown in Fig. 2.

5.1.1 Effects of α , β , and γ

In this subsection, we study the effects of the parameters α , β , and γ on the performance of the proposed LTVFF and LCTVFF mechanisms. For the LTVFF mechanism, we investigate the steady-state MSD values versus α for $\beta = 0.0015, 0.002, 0.0025, 0.005$. The simulation results are shown in Fig. 3. For the LCTVFF mechanism, we first depict the steady-state MSD values versus α for $\beta = 0.0025, 0.005, 0.0075, 0.01$ in Fig. 4. Then, the effects of γ are illustrated in Fig. 5 by investigating the steady-state MSD values against γ for different pairs of α and β .

As can be seen from Figs. 3 and 4 for both the LTVFF and LCTVFF mechanisms, the optimal choice of α and β is not unique. Specifically, different pairs of α and β can yield the same steady-state MSD value. For example, for the LTVFF mechanism, the pairs $\alpha = 0.91, \beta = 0.0015$, $\alpha = 0.89, \beta = 0.002$, and $\alpha = 0.87, \beta = 0.0025$ provide almost the same steady-state MSD performance. For the LCTVFF mechanism, when $\gamma = 0.95$, the pairs $\alpha = 0.93, \beta = 0.0025$, $\alpha = 0.90, \beta = 0.005$, $\alpha = 0.85, \beta = 0.0075$, and $\alpha = 0.80, \beta = 0.01$ yield almost the same steady-state MSD value. In addition, it can also be observed that with the decreasing of α and β , the steady-state performance degrades. Furthermore, the result in Fig. 5 reveals that the steady-state MSD performance of the LCTVFF mechanism does not change so much as γ varies for different pairs of α and β .

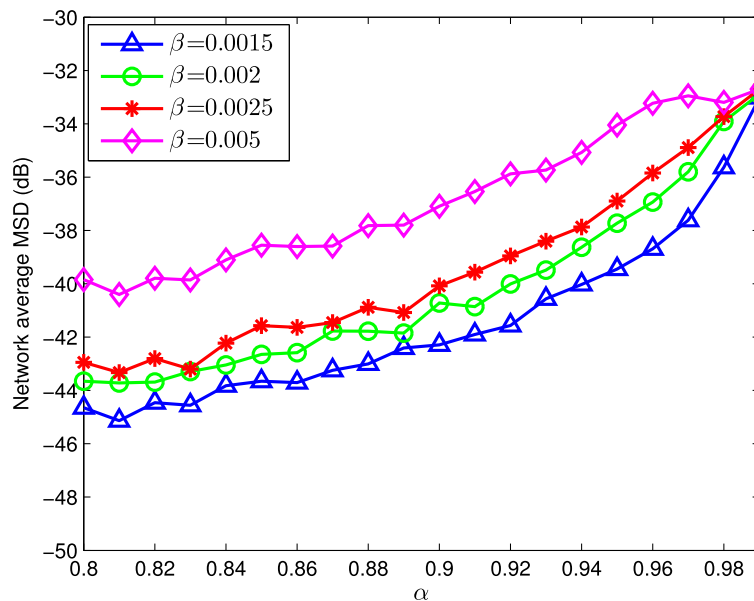


Fig. 3 Steady-state MSD versus α for different values of β for the LTVFF mechanism

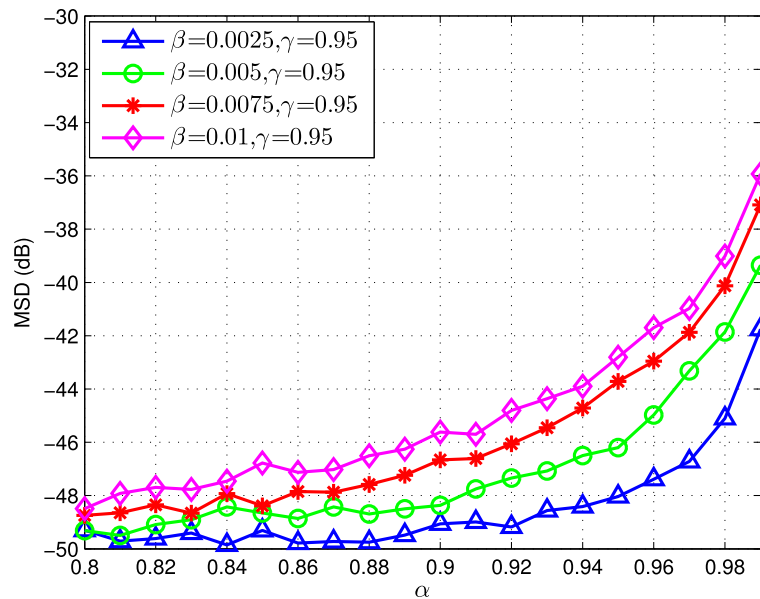


Fig. 4 Steady-state MSD versus α for different values of β for the LCTVFF mechanism when $\gamma = 0.95$

However, when we choose appropriate values for α , β , and γ , only considering the effects on the steady-state behaviors is not enough. This is because that the convergence speed is closely connected to the steady-state MSD values. That is to say, when the algorithm assumes a faster convergence speed, the steady-state error floor rises; if the convergence speed is controlled to be slower, the steady-state performance improves.

Figures 6 and 7 show the trade-off between convergence speed and steady-state performance by depicting learning curves against different values of α and β for LTVFF-DRLS and LCTVFF-DRLS algorithms, respectively. Therefore, we need to keep a good balance between the steady-state behaviors and the convergence speed in order to ensure good performance. In practical applications, the optimized values of α , β , and γ should be

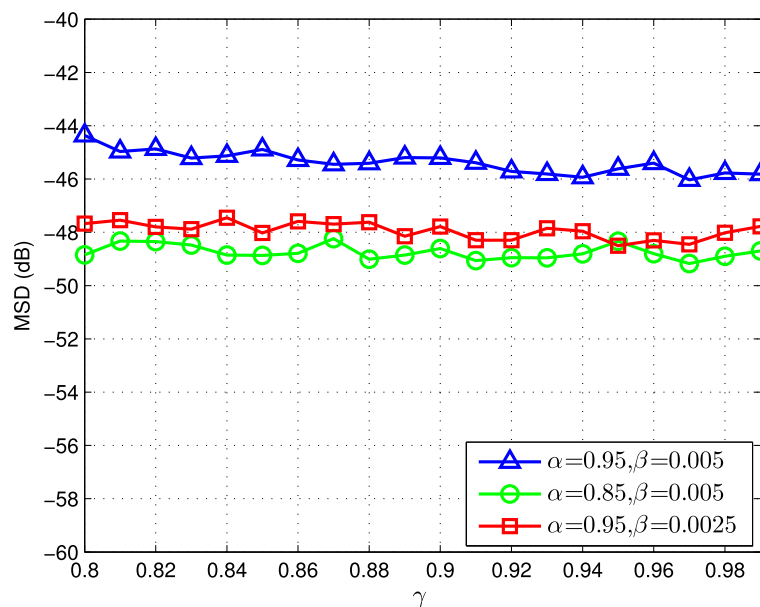


Fig. 5 Steady-state MSD versus γ for different values of α and β for the LCTVFF mechanism

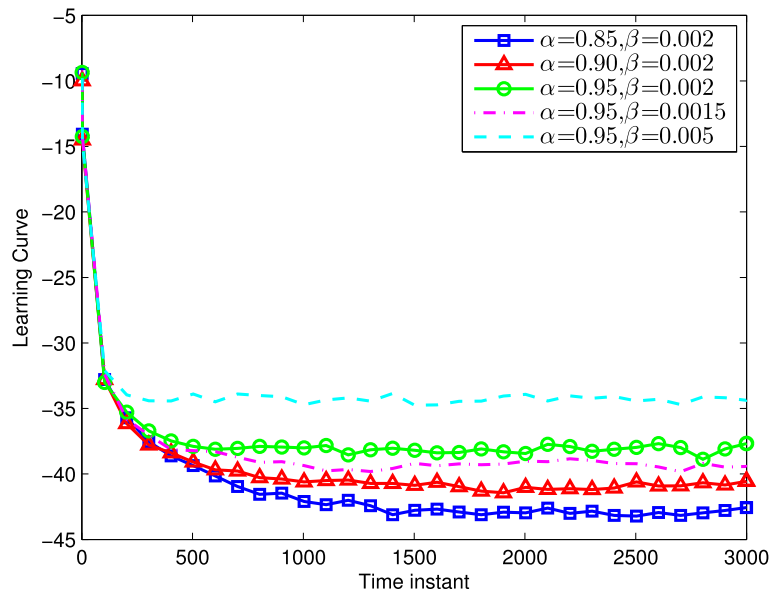


Fig. 6 Learning curves against different values of α and β for LTVFF-DRLS algorithm

obtained through experiments and then stored for the future use.

5.1.2 MSD and EMSE performance

Figures 8 and 9 show the MSD curves against the number of iterations for the LTVFF-DRLS and LCTVFF-DRLS algorithms with different initial values for the forgetting factor in comparison with the conventional DRLS algorithm and the GVFF-DRLS algorithm, respectively. The parameters of the considered algorithms are listed in

Table 4. From the results, the LTVFF-DRLS algorithm converges to almost the same error floor in two scenarios where the variable forgetting factor is initialized to be small or large. This is also true for the LCTVFF-DRLS algorithm, which has lower error floor and faster convergence speed than the LTVFF-DRLS algorithm. However, as shown in Fig. 8, for the conventional DRLS algorithm, its convergence speed and steady-state error floor both have obvious changes when the fixed forgetting factors increases. Specifically, when the fixed forgetting factor is

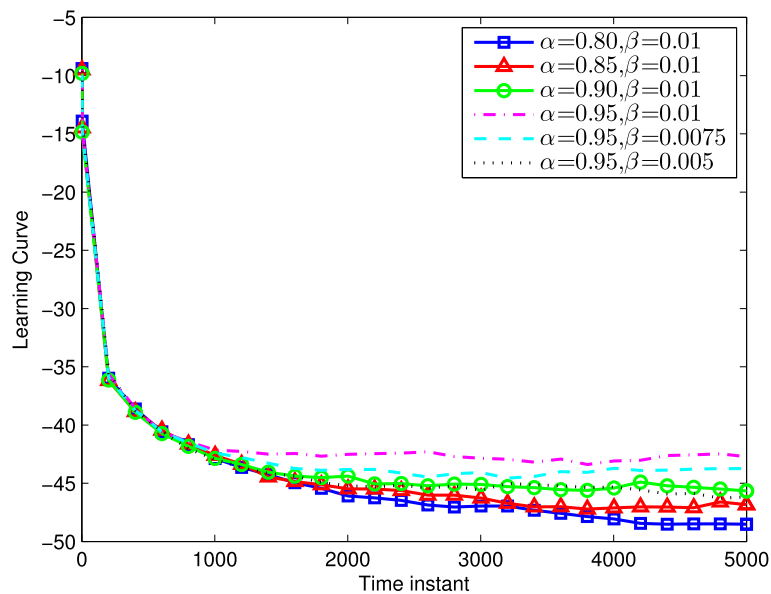


Fig. 7 Learning curves against different values of α and β for LCTVFF-DRLS algorithm when $\gamma = 0.95$

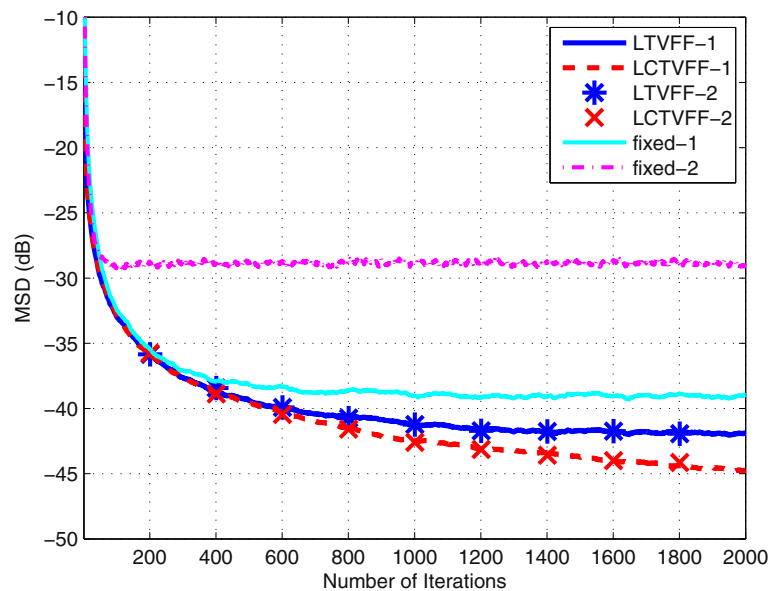


Fig. 8 MSD performance against iterations for the proposed algorithms with different initial values for the forgetting factor compared with the DRLS algorithm with the fixed forgetting factor

small, the conventional DRLS algorithm converges faster but has a higher error floor than the LTVFF-DRLS algorithm; however, as the fixed forgetting factors increase, it converges to a lower error floor (not as good as the LTVFF-DRLS algorithm) but has slower convergence speed. Besides, from Fig. 9, the MSD performance of the proposed LTVFF-DRLS and LCTVFF-DRLS algorithms are

less sensitive to the initial values for the forgetting factor than that of the GVFF-DRLS algorithm. Therefore, by employing the LTVFF and LCTVFF mechanisms, the proposed algorithms can track the optimal performance regardless of the initial values for the forgetting factor and greatly reduce the difficulty in choosing the appropriate value for the forgetting factor.

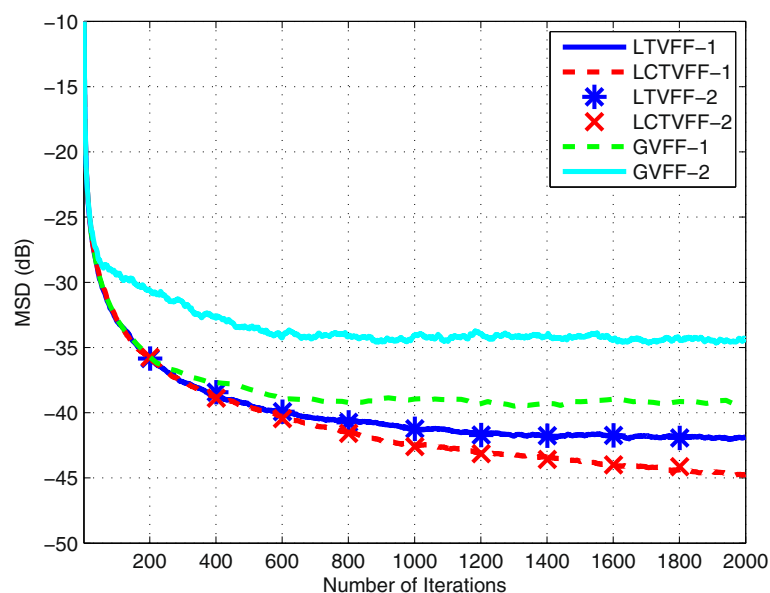


Fig. 9 MSD performance against iterations for the proposed algorithms with different initial values for the forgetting factor compared with the GVFF-DRLS algorithm

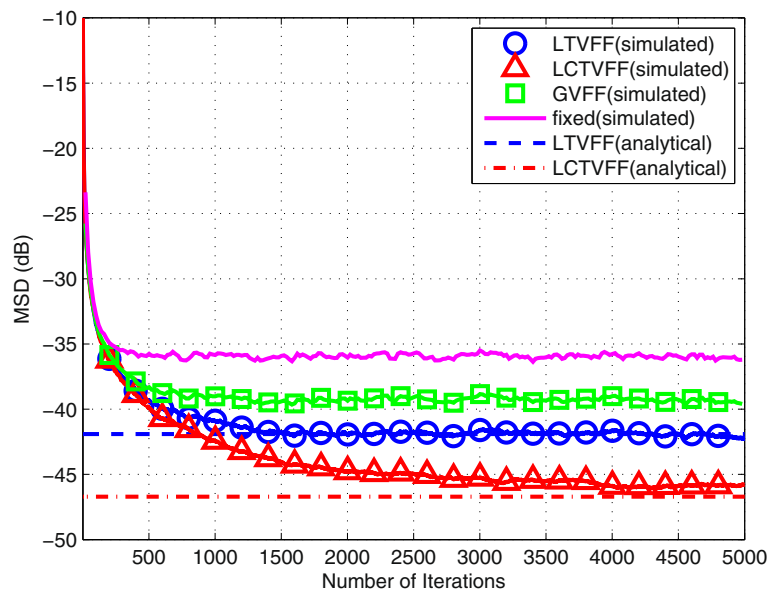
Table 4 Optimized parameters for different algorithms considered in Figs. 8 and 9

LTVFF-1	$\alpha = 0.91, \beta = 0.0015$ $\lambda_0 = 0.995, \lambda_+ = 0.9998, \lambda_- = 0.980$
LTVFF-2	$\alpha = 0.91, \beta = 0.0015$ $\lambda_0 = 0.950, \lambda_+ = 0.9998, \lambda_- = 0.950$
LCTVFF-1	$\alpha = 0.95, \beta = 0.005, \gamma = 0.95$ $\lambda_0 = 0.995, \lambda_+ = 0.9998, \lambda_- = 0.950$
LCTVFF-2	$\alpha = 0.95, \beta = 0.005, \gamma = 0.95$ $\lambda_0 = 0.950, \lambda_+ = 0.9998, \lambda_- = 0.950$
GVFF-1	$\lambda_0 = 0.995, \mu = 0.005, \lambda_+ = 0.9998, \lambda_- = 0.990$
GVFF-2	$\lambda_0 = 0.950, \mu = 0.005, \lambda_+ = 0.9998, \lambda_- = 0.950$
Fixed-1	$\lambda = 0.998$
Fixed-2	$\lambda = 0.995$
Fixed-3	$\lambda = 0.950$

In Figs. 10, 11, 12, and 13, we evaluate the performance of the proposed LTVFF-DRLS and LCTVFF-DRLS algorithms based on MSD and EMSE behaviors in comparison with that of the conventional DRLS with the fixed forgetting factor and the GVFF-DRLS algorithms. Specifically, the MSD and EMSE curves against the number of iterations for the analyzed algorithms are depicted in Figs. 10 and 11, respectively, while the steady-state MSD and EMSE values for each node are shown in Figs. 12 and 13, respectively. As can be seen from these results, both the LTVFF-DRLS and LCTVFF-DRLS algorithms converge

after a number of iterations and achieve lower steady-state MSD and EMSE values compared to the DRLS algorithm with the fixed forgetting factor and the GVFF-DRLS algorithm. Besides, we also depict the analytical results which are calculated through expressions (76) and (78) in Figs. 10, 11, 12, and 13. From these results, it is clear that analytical expressions corroborate the simulated results very well. The parameters of the considered algorithms are shown in Table 5, which are tuned through experiments by referring to the investigation in Section 5.1.1.

In Fig. 14, we test the performance of different algorithms considered in a non-stationary environment. Specifically, in order to simulate the non-stationary environment, we consider the scenario where the topology of the sensor network varies over time, that is, the total number of sensor nodes is set to 40 at the start, and then, we switch off half of the nodes after 100 iterations and another 10 nodes after 800 iterations. The MSD curves against the number of iterations for the proposed algorithms in comparison with the conventional DRLS algorithm with the fixed forgetting factor and the GVFF-DRLS algorithm in the non-stationary environment are depicted in Fig. 14. As can be observed, the switching off of some sensor nodes results in the degradation of the performance for all the algorithms. However, the proposed LTVFF-DRLS and LCTVFF-DRLS algorithms still outperform the other two existing algorithms in MSD performance. Besides, they exhibit better tracking properties by showing smaller and smoother variations in the MSD curves at the time of switching sensor nodes.

**Fig. 10** MSD curve against number of iterations for the proposed and existing algorithms

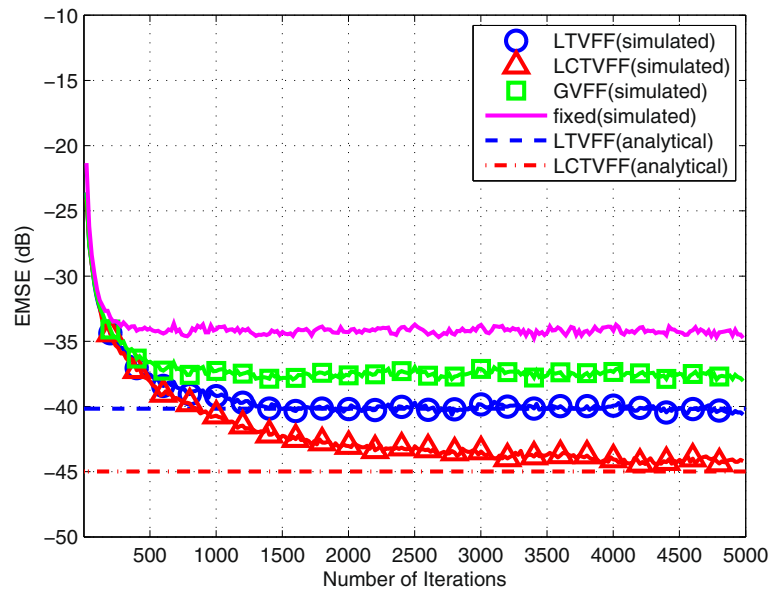


Fig. 11 EMSE curve against number of iterations for the proposed and existing algorithms

Next, we elaborate the numerical stability of the proposed LTVFF-DRLS and LCTVFF-DRLS algorithms. Through tuning the parameters α , β , γ , λ_+ , λ_- to different values, the proposed LTVFF-DRLS and LCTVFF-DRLS algorithms can have different convergence speed and steady-state performance, but their MSD and EMSE curves always decrease to the steady-state. Indeed, after a number of experiments, we have not encountered the case where they diverge. Hence, the proposed LTVFF and

LCTVFF mechanisms do not make the numerical stability of the DRLS algorithm worse. Besides, the simulation results in Fig. 14 show that, after switching some nodes in the network, the proposed LTVFF-DRLS and LCTVFF-DRLS algorithms still achieve superior performance to the conventional DRLS algorithm, and they exhibit smoother MSD curves at the time of switching nodes, especially the LCTVFF-DRLS algorithm. This further verifies that the proposed algorithms improve instead of impair the

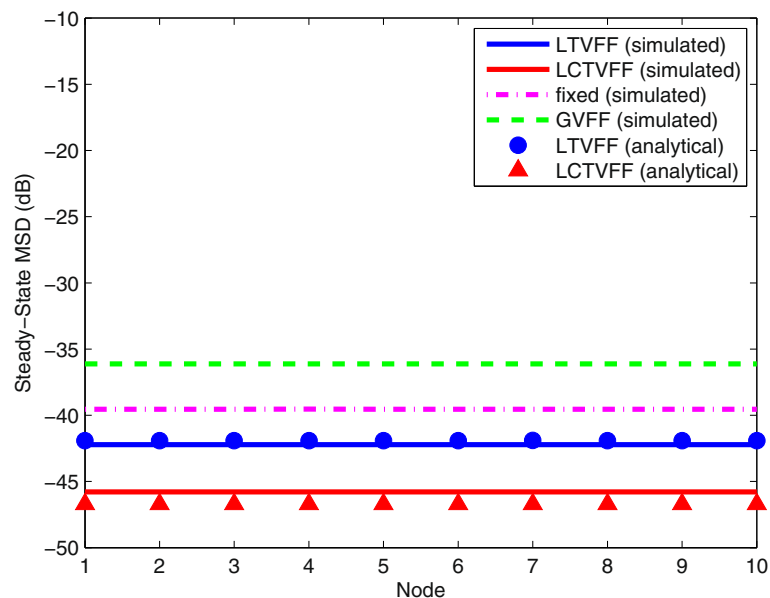


Fig. 12 Steady-state MSD value versus node for the proposed and existing algorithms

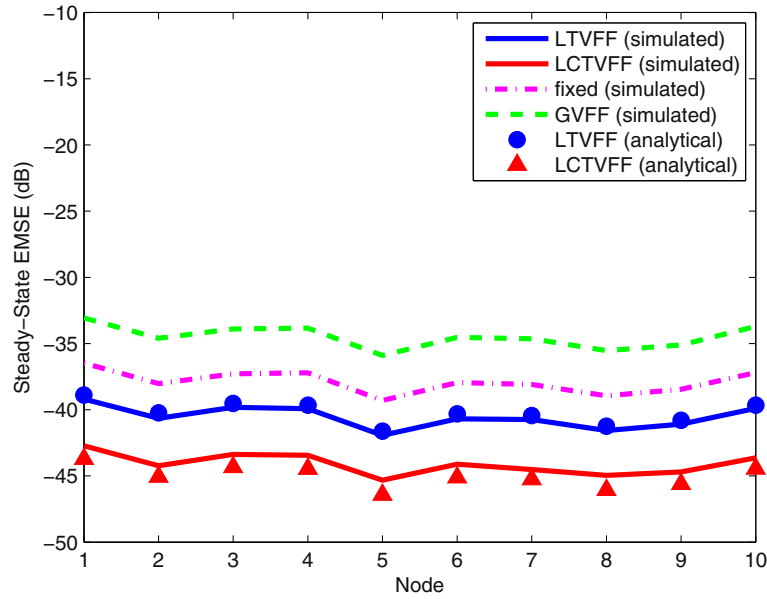


Fig. 13 Steady-state EMSE value versus node for the proposed and existing algorithms

numerical stability of the DRLS algorithm by keeping better tracking of the variations.

5.2 Distributed spectrum estimation

In this part, we extend the proposed LTVFF-DRLS and LCTVFF-DRLS algorithms to the application of distributed spectrum estimation, for which we focus on estimating the parameter \mathbf{w}_0 that is relevant to the unknown spectrum of a transmitted signal s . First of all, we characterize the system model of distributed spectrum estimation.

We denote the power spectral density (PSD) of the unknown spectrum of the transmitted signal s by $\Phi_s(f)$, which can be well approximated by the following basis expansion model [32] with N_b sufficiently large:

$$\Phi_s(f) = \sum_{m=1}^{N_b} b_m(f) w_{0m} = \mathbf{b}_0^T(f) \mathbf{w}_0 \quad (91)$$

Table 5 Optimized parameters for different algorithms considered in Figs. 10, 11, 12, and 13

Fixed scheme	$\lambda = 0.990$
GVFF	$\mu = 0.005, \lambda_0 = 0.990, \lambda_+ = 0.9998, \lambda_- = 0.990$
LTVFF	$\alpha = 0.91, \beta = 0.0015, \lambda_0 = 0.990$ $\lambda_+ = 0.9998, \lambda_- = 0.980$
LCTVFF	$\alpha = 0.95, \beta = 0.005, \gamma = 0.95$ $\lambda_0 = 0.990, \lambda_+ = 0.9998, \lambda_- = 0.950$

where $\mathbf{b}_0(f) = \text{col}\{b_1(f), b_2(f), \dots, b_{N_b}(f)\}$ is the vector of basis functions [33, 34], $\mathbf{w}_0 = \text{col}\{w_{01}, w_{02}, \dots, w_{0N_b}\}$ is the expansion parameter to be estimated and represents the power that transmits the signal s over each basis, and N_b is the number of basis functions.

We assume $H_k(f, i)$ to be the channel transfer function between the source emitting the signal s and the receiver node k at time instant i . Based on (91), the PSD of the signal received by node k can be represented as

$$\begin{aligned} \Phi_r(f) &= |H_k(f, i)|^2 \Phi_s(f) + \sigma_{r,k}^2 \\ &= \sum_{m=1}^{N_b} |H_k(f, i)|^2 b_m(f) w_{0m} + \sigma_{r,k}^2 \\ &= \mathbf{b}_{k,i}^T(f) \mathbf{w}_0 + \sigma_{r,k}^2 \end{aligned} \quad (92)$$

where $\mathbf{b}_{k,i}(f) = [|H_k(f, i)|^2 b_m(f)]_{m=1}^{N_b} \in \mathbb{R}^{N_b}$ and $\sigma_{r,k}^2$ denotes the receiver noise power at node k .

At each time instant i , by observing the received PSD described in (92) over N_c frequency samples $f_j = f_{\min} : (f_{\max} - f_{\min})/N_c : f_{\max}$, for $j = 1, 2, \dots, N_c$, each node k takes measurements according to the following model:

$$d_{k,i}^j = \mathbf{b}_{k,i}^T(f_j) \mathbf{w}_0 + \sigma_{r,k}^2 + v_{k,i}^j \quad (93)$$

where $v_{k,i}^j$ denotes the sampling noise at frequency f_j with zero mean and variance $\sigma_{n,j}^2$. The receiver noise power $\sigma_{r,k}^2$ can be estimated with high accuracy preliminarily

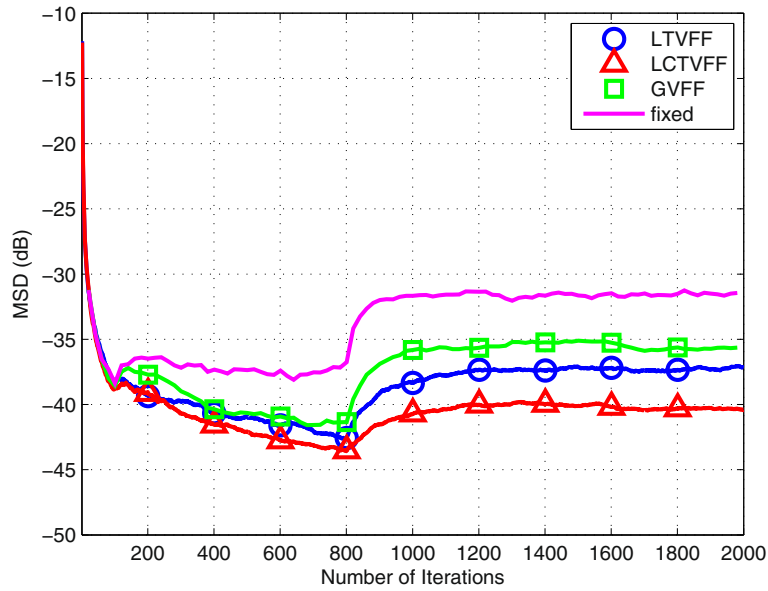


Fig. 14 MSD performance against number of iterations for the proposed and existing algorithms in a nonstationary environment

and then subtracted from (93) [35, 36]. Therefore, we can obtain

$$d_{k,i}^j = \mathbf{b}_{k,i}^T(f_j) \mathbf{w}_0 + v_{k,i}^j. \quad (94)$$

By collecting the measurements over N_c frequencies into a column vector $\mathbf{d}_{k,i}$, we obtain the following system model of distributed spectrum estimation:

$$\mathbf{d}_{k,i} = \mathbf{B}_{k,i} \mathbf{w}_0 + \mathbf{v}_{k,i}. \quad (95)$$

where $\mathbf{d}_{k,i} = [d_{k,i}^j]_{j=1}^{N_c} \in \mathbb{R}^{N_c}$, $\mathbf{B}_{k,i} = [\mathbf{b}_{k,i}^T(f_j)]_{j=1}^{N_c} \in \mathbb{R}^{N_c \times N_b}$, with $N_c > N_b$, and $\mathbf{v}_{k,i} = [v_{k,i}^j]_{j=1}^{N_c} \in \mathbb{R}^{N_c}$.

Next, we carry out simulations to show the performance of the proposed algorithms when applied to distributed spectrum estimation. We consider a sensor network composed of $N = 20$ nodes in order to estimate the unknown expansion parameter \mathbf{w}_0 . We use $N_b = 50$ non-overlapping rectangular basis functions with amplitude

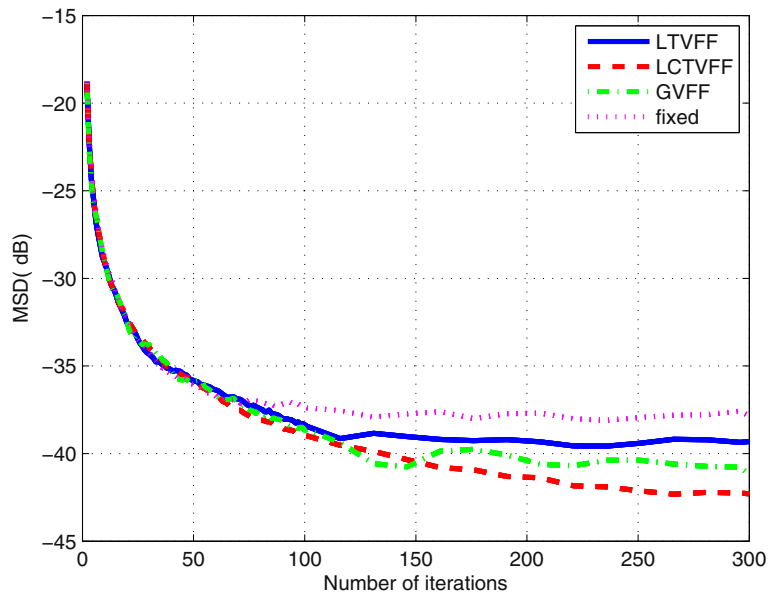


Fig. 15 MSD performance for different algorithms applied in distributed spectrum estimation

Table 6 Simulation time of running different algorithms in Figs. 15 and 16

	Simulation time (seconds)
LTVFF	61.11
LCTVFF	66.35
GVFF	176
fixed	73

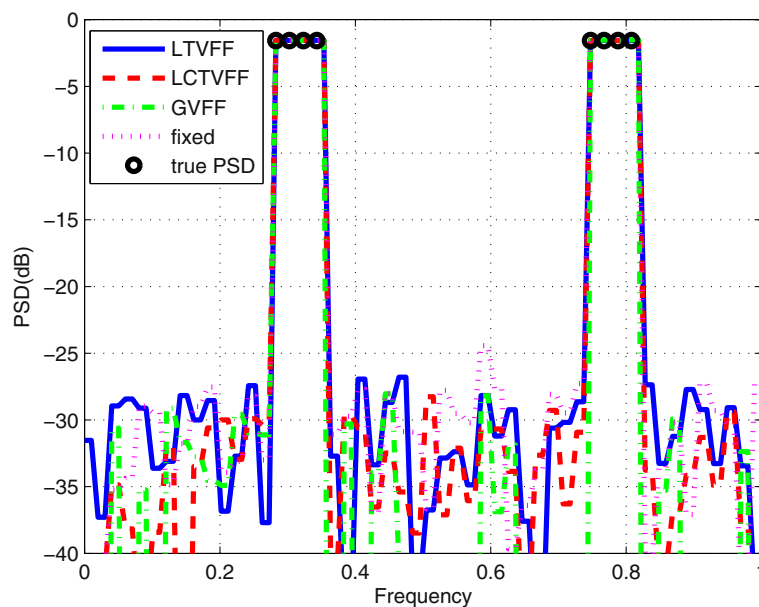
equal to one to approximate the PSD of the unknown spectrum. The nodes can scan $N_c = 100$ frequencies over the frequency axis, which is normalized between 0 and 1. In particular, we assume that only 8 entries of \mathbf{w}_0 are non-zero, which implies that the unknown spectrum is transmitted over 8 basis functions. Thus, the sparsity ratio equals to 8/50. We set the power transmitted over each basis function to be 0.7 and the variance of the sampling noise to be 0.004.

In Fig. 15, we compare the performance of different algorithms for the distributed spectrum estimation in terms of MSD. As can be depicted, the proposed LTVFF-DRLS and LCTVFF-DRLS algorithms still outperform the conventional DRLS algorithm in steady-state performance. By tuning parameters, the GVFF-DRLS algorithm can achieve similar performance to the proposed algorithms in the convergence speed and steady-state MSD values but at huge computational cost. We have listed the simulation time of running each algorithm for 600 iterations and 1 Monte Carlo experiment in Table 6.

As can be observed, the simulation time of running the GVFF-DRLS algorithm is almost 3 times of that for running the other algorithms. In Fig. 16, we take node 1 as an example to investigate the performance of different algorithms in estimating the true PSD. From the results, although different algorithms obtain similar estimates of the true PSD, the proposed LCTVFF-DRLS algorithm obviously leads to smaller side lobes in the PSD curve than the other three.

6 Conclusions

In this paper, we have proposed two low-complexity VFF-DRLS algorithms for distributed estimation including the LTVFF-DRLS and LCTVFF-DRLS algorithms. For the LTVFF-DRLS algorithm, the forgetting factor is adjusted by the time-averaged cost function, while for the LCTVFF-DRLS algorithm, the forgetting factor is adjusted by the time-averaged of the correlation of two successive estimation errors. We also have investigated the computational complexity of the low-complexity VFF mechanisms as well as the proposed VFF-DRLS algorithms. In addition, we have carried out the convergence and steady-state analysis for the proposed algorithms. Moreover, we also have derived analytical expressions for the steady-state MSD and EMSE. The simulation results have shown the superiority of the proposed algorithms to the conventional DRLS and GVFF-DRLS algorithms in applications of distributed parameter estimation and distributed spectrum estimation and have verified the effectiveness of our proposed analytical expressions for the steady-state MSD and EMSE.

**Fig. 16** PSD performance for different algorithms applied in distributed spectrum estimation

Appendices

A: Proof of the uncorrelation of $\rho_k(i-1)$ and $|e_k(i)|^2$ in the steady state

By multiplying both sides of (26) by $|e_k(i)|^2$ and taking expectations, we have the following equation:

$$\begin{aligned} E[\rho_k(i-1)|e_k(i)|^2] &= \gamma E[\rho_k(i-2)|e_k(i)|^2] \\ &\quad + (1-\gamma)E[|e_k(i-2)||e_k(i-1)||e_k(i)|^2]. \end{aligned} \quad (96)$$

Recall that the values of $e_k(i-1)$ and $e_k(i)$ and the values of $\rho_k(i-1)$ and $\rho_k(i)$ can be considered approximately equivalent when $i \rightarrow \infty$; therefore, we have the following results:

$$\begin{aligned} E[\rho_k(i-1)|e_k(i)|^2] &\approx \gamma E[\rho_k(i-2)|e_k(i)|^2] \\ &\quad + (1-\gamma)E[|e_k(i-1)|^2]E[|e_k(i)|^2] \\ &\approx \gamma E[\rho_k(i-1)|e_k(i)|^2] \\ &\quad + (1-\gamma)\varepsilon_{\min}^2. \end{aligned} \quad (97)$$

By recalling (27), we can obtain

$$E[\rho_k(i-1)|e_k(i)|^2] \approx \varepsilon_{\min}^2 \approx E[\rho_k(i-1)]E[|e_k(i)|^2] \quad (98)$$

That is, we can conclude that $\rho_k(i-1)$ and $|e_k(i)|^2$ are uncorrelated in the steady state.

B: Proof of (39)

According to (8), we can obtain the following equation:

$$\mathbf{P}_{k,i}^{-1} = \prod_{j=0}^i \lambda_k(j) \mathbf{\Pi} + \mathcal{H}_i^* \mathcal{W}_{k,i} \mathcal{H}_i \quad (99)$$

where the matrices \mathcal{H}_i and $\mathcal{W}_{k,i}$ can be expressed as follows

$$\begin{aligned} \mathcal{H}_i &= \begin{bmatrix} \mathbf{H}_i \\ \mathcal{H}_{i-1} \end{bmatrix} \\ \mathcal{W}_{k,i} &= \begin{bmatrix} \mathbf{R}_v^{-1} \mathbf{C}_k & \\ & \lambda_k(i) \mathcal{W}_{k,i-1} \end{bmatrix}. \end{aligned} \quad (100)$$

Therefore, (99) can be reformulate as

$$\begin{aligned} \mathbf{P}_{k,i}^{-1} &= \lambda_k(i) \left(\prod_{j=0}^{i-1} \lambda_k(j) \mathbf{\Pi} + \mathcal{H}_{i-1}^* \mathcal{W}_{k,i-1} \mathcal{H}_{i-1} \right) \\ &\quad + \mathbf{H}_i^* \mathbf{R}_v^{-1} \mathbf{C}_k \mathbf{H}_i. \end{aligned} \quad (101)$$

Substituting (2) into (101) yields the following recursion:

$$\mathbf{P}_{k,i}^{-1} = \lambda_k(i) \mathbf{P}_{k,i-1}^{-1} + \sum_{m=1}^N \frac{C_{m,k}}{\sigma_{v,m}^2} \mathbf{u}_{m,i} \mathbf{u}_{m,i}^*. \quad (102)$$

By employing the iterative Eq. (102), we can write

$$\begin{aligned} \mathbf{P}_{k,i}^{-1} &= \sum_{l=1}^N \frac{C_{l,k}}{\sigma_{v,l}^2} \mathbf{u}_{l,i} \mathbf{u}_{l,i}^* + \lambda_k(i) \sum_{l=1}^N \frac{C_{l,k}}{\sigma_{v,l}^2} \mathbf{u}_{l,i-1} \mathbf{u}_{l,i-1}^* \\ &\quad + \lambda_k(i) \lambda_k(i-1) \sum_{l=1}^N \frac{C_{l,k}}{\sigma_{v,l}^2} \mathbf{u}_{l,i-2} \mathbf{u}_{l,i-2}^* + \dots \\ &\quad + \prod_{j=i}^1 \lambda_k(j) \sum_{l=1}^N \frac{C_{l,k}}{\sigma_{v,l}^2} \mathbf{u}_{l,0} \mathbf{u}_{l,0}^* + \prod_{j=i}^0 \lambda_k(j) \mathbf{\Pi}. \end{aligned} \quad (103)$$

Recalling Assumption 1, we know that the correlation matrix of the input vector is invariant over time, as a result, the correlation matrix $\mathbf{R}_{u_{l,i}}$ can be represented as \mathbf{R}_{u_l} . Therefore, by taking expectations on both sides of (103), we obtain the following result

$$\begin{aligned} E[\mathbf{P}_{k,i}^{-1}] &= \sum_{l=1}^N \frac{C_{l,k}}{\sigma_{v,l}^2} \mathbf{R}_{u_l} + E[\lambda_k(i)] \sum_{l=1}^N \frac{C_{l,k}}{\sigma_{v,l}^2} \mathbf{R}_{u_l} + E[\lambda_k(i)] \\ &\quad \times \lambda_k(i-1) \sum_{l=1}^N \frac{C_{l,k}}{\sigma_{v,l}^2} \mathbf{R}_{u_l} + \dots + E \left[\prod_{j=i}^1 \lambda_k(j) \right] \sum_{l=1}^N \frac{C_{l,k}}{\sigma_{v,l}^2} \mathbf{R}_{u_l} \\ &\quad + E \left[\prod_{j=i}^0 \lambda_k(j) \right] \mathbf{\Pi}. \end{aligned} \quad (104)$$

In view of Assumption 2, (104) can be approximately rewritten as

$$\begin{aligned} E[\mathbf{P}_{k,i}^{-1}] &\approx (1 + E[\lambda_k(i)] + \dots + E[\lambda_k(i)]^{i-N_i+1}) \sum_{l=1}^N \frac{C_{l,k}}{\sigma_{v,l}^2} \mathbf{R}_{u_l} \\ &\quad + E[\lambda_k(i) \lambda_k(i-1) \dots \lambda_k(N_i)] E \left[\lambda_k(N_i-1) \right. \\ &\quad \left. + \lambda_k(N_i-1) \lambda_k(N_i-2) + \dots + \prod_{j=N_i-1}^1 \lambda_k(j) \right] \sum_{l=1}^N \frac{C_{l,k}}{\sigma_{v,l}^2} \mathbf{R}_{u_l} \\ &\quad + E \left[\prod_{j=i}^{N_i} \lambda_k(j) \right] E \left[\prod_{j=N_i-1}^0 \lambda_k(j) \right] \mathbf{\Pi} \\ &\approx (1 + E[\lambda_k(i)] + \dots + E[\lambda_k(i)]^{i-N_i+1}) \sum_{l=1}^N \frac{C_{l,k}}{\sigma_{v,l}^2} \mathbf{R}_{u_l} \\ &\quad + E[\lambda_k(i)]^{i-N_i+1} (\xi + \chi) \end{aligned} \quad (105)$$

where ξ and χ can be expressed as follows, respectively:

$$\begin{aligned} \xi &= E \left[\lambda_k(N_i-1) + \lambda_k(N_i-1) \lambda_k(N_i-2) \right. \\ &\quad \left. + \dots + \prod_{j=N_i-1}^1 \lambda_k(j) \right] \sum_{l=1}^N \frac{C_{l,k}}{\sigma_{v,l}^2} \mathbf{R}_{u_l} \end{aligned} \quad (106)$$

and

$$\chi = E \left[\prod_{j=N_i-1}^0 \lambda_k(j) \right] \mathbf{\Pi}. \quad (107)$$

Since n_i is a finite positive number, ξ and χ are two deterministic values. In addition, note that $\lambda_k(i)$ does not exceed its upper bound λ_+ , which is smaller than but close to unity. Therefore, we have $0 < E[\lambda_k(i)] < \lambda_+ < 1$, and $E[\lambda_k(i)]^{i-N_i+1} < \lambda_+^{i-N_i+1}$. When i is large enough, $\lambda_+^{i-N_i+1}$ approaches zero, and, of course, $E[\lambda_k(i)]^{i-N_i+1}$ also approaches zero. As a result, the last term in (105) vanishes. Then, we obtain the following result:

$$\lim_{i \rightarrow \infty} E[\mathbf{P}_{k,i}^{-1}] = \frac{1}{1 - E[\lambda_k(\infty)]} \sum_{l=1}^N \frac{C_{l,k}}{\sigma_{v,l}^2} \mathbf{R}_{u_l} \quad (108)$$

where the values of $\lambda_k(\infty)$ is given in (24) for the LTVFF mechanism and in (35) for the LCTVFF mechanism, respectively. Hence, we obtain (39). Note that, by setting appropriate truncation bounds for $\lambda_k(i)$, the steady-state forgetting factor value will not be influenced by the truncation. Hence, the result (39) always holds true despite the truncation employed to the VFF mechanisms. Indeed, the truncation mechanism only plays a role during the process of converging. Once the algorithms reach the steady state, the values of the forgetting factor are not affected by the truncation mechanism any longer.

Funding

This work was supported in part by the National Natural Science Foundation of China under Grant 61471319, the Scientific Research Project of Zhejiang Provincial Education Department under Grant Y201122655, and the Fundamental Research Funds for the Central Universities.

Authors' contributions

YC and RCDL proposed the original idea. LZ carried out the experiment. In addition, LZ and YC wrote the paper. CL and RCDL supervised and reviewed the manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

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

Author details

¹College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, People's Republic of China. ²CETUC-PUC-Rio, Rio de Janeiro, Brazil.

Received: 9 November 2016 Accepted: 6 July 2017

Published online: 15 August 2017

References

1. P. Corke, T. Wark, R. Jurdak, W. Hu, P. Valencia, D. Moore, Environmental wireless sensor networks. *Proc. IEEE*. **98**(11), 1903–1917 (2010)
2. JG Ko, C. Lu, MB Srivastava, JA Stankovic, A Terzis, M. Welsh, Wireless sensor networks for healthcare. *Proc. IEEE*. **98**(11), 1947–1960 (2010)
3. R. Abdolee, B. Champagne, AH Sayed, in *Proc. IEEE Statistical Signal Processing Workshop*. Diffusion LMS for Source and Process Estimation in Sensor Networks (IEEE, Ann Arbor, 2012)
4. R. Abdolee, B. Champagne, AH Sayed, in *Proc. IEEE ICASSP*. Diffusion LMS Localization and Tracking Algorithm for Wireless Cellular Networks (IEEE, Vancouver, 2013)
5. R. Abdolee, B. Champagne, AH Sayed, Diffusion adaptation over multi-agent networks with wireless link impairments. *IEEE Trans. Mob. Comput.* **15**(6), 1362–1376 (2016)
6. FS Cattivelli, CG Lopes, AH Sayed, in *Proc. IEEE Workshop Signal Process. Advances Wireless Commun. (SPAWC)*. A Diffusion RLS Scheme for Distributed Estimation over Adaptive Networks (IEEE, Helsinki, 2007), pp. 1–5
7. FS Cattivelli, CG Lopes, AH Sayed, Diffusion recursive least-squares for distributed estimation over adaptive networks. *IEEE Trans. Signal Process.* **56**(5), 1865–1877 (2008)
8. FS Cattivelli, AH Sayed, Diffusion LMS strategies for distributed estimation. *IEEE Trans. Signal Process.* **58**(3), 1035–1048 (2010)
9. CG Lopes, AH Sayed, Diffusion least-mean squares over distributed networks: formulation and performance analysis. *IEEE Trans. Signal Process.* **56**(7), 3122–3136 (2008)
10. Y. Liu, C. Li, Z. Zhang, Diffusion sparse least-mean squares over networks. *IEEE Trans. Signal Process.* **60**(8), 4480–4485 (2012)
11. S. Xu, RC de Lamare, HV Poor, in *Proc. IEEE ICASSP*. Adaptive link selection strategies for distributed estimation in diffusion wireless networks (IEEE, Vancouver, 2013)
12. S. Xu, RC de Lamare, HV Poor, Distributed compressed estimation based on compressive sensing. *IEEE Signal Process. Lett.* **22**(9), 1311–1315 (2015)
13. MOB Saeed, A. Zerguine, SA Zummo, A variable step-size strategy for distributed estimation over adaptive networks. *EURASIP J. Adv. Signal Process.* **2013**(1), 1–14 (2013)
14. H. Lee, S. Kim, J. Lee, W. Song, A variable step-size diffusion LMS algorithm for distributed estimation. *IEEE Trans. Signal Process.* **63**(7), 1808–1820 (2015)
15. Z. Liu, Y. Liu, C. Li, Distributed sparse recursive least-squares over networks. *IEEE Trans. Signal Process.* **62**(6), 1386–1395 (2014)
16. S. Huang, C. Li, Distributed sparse total least-squares over networks. *IEEE Trans. Signal Process.* **63**(11), 2986–2998 (2015)
17. C. Li, P. Shen, Y. Liu, Z. Zhang, Diffusion information theoretic learning for distributed estimation over network. *IEEE Trans. Signal Process.* **61**(16), 4011–4024 (2013)
18. Z. Liu, C. Li, Y. Liu, Distributed censored regression over networks. *IEEE Trans. Signal Process.* **63**(20), 5437–5449 (2015)
19. S. Haykin, *Adaptive Filter Theory*, 4th edn. (Prentice-Hall, Englewood cliffs, 2000)
20. S. Leung, CF So, Gradient-based variable forgetting factor RLS algorithm in time-varying environments. *IEEE Trans. Signal Process.* **53**(8), 3141–3150 (2005)
21. CF So, SH Leung, Variable forgetting factor RLS algorithm based on dynamic equation of gradient of mean square error. *Electron. Lett.* **37**(3), 202–203 (2011)
22. S. Song, J. Lim, S. Baek, K. Sung, Gauss Newton variable forgetting factor recursive least squares for time varying parameter tracking. *Electron. Lett.* **36**(11), 988–990 (2000)
23. S. Song, J. Lim, SJ Baek, K. Sung, Variable forgetting factor linear least squares algorithm for frequency selective fading channel estimation. *IEEE Trans. Vehi. Technol.* **51**(3), 613–616 (2002)
24. F. Albu, in *Proc. of ICARCV 2012*. Improved Variable Forgetting Factor Recursive Least Square Algorithm (IEEE, Guangzhou, 2012)
25. Y. Cai, RC de Lamare, M. Zhao, J. Zhong, Low-complexity variable forgetting factor mechanisms for blind adaptive constrained constant modulus algorithms. *IEEE Trans. Signal Process.* **60**(8), 3988–4002 (2012)
26. L. Qiu, Y. Cai, M. Zhao, Low-complexity variable forgetting factor mechanisms for adaptive linearly constrained minimum variance beamforming algorithms. *IET Signal Process.* **9**(2), 154–165 (2015)
27. R. Arablouei, K. Dogancay, S. Werner, Y. Huang, Adaptive distributed estimation based on recursive least-squares and partial diffusion. *IET Signal Process.* **62**(14), 1198–1208 (2014)
28. DS Tracy, RP Singh, A new matrix product and its applications in partitioned matrix differentiation. *Statistica Neerlandica*. **51**(3), 639–652 (2003)
29. H. Shin, AH Sayed, in *Proc. IEEE ICASSP*. Transient Behavior of Affine Projection Algorithms (IEEE, Hong Kong, 2003)
30. JH Husoy, MSE Abadi, in *IEEE MELECON 2004*. A Common Framework for Transient Analysis of Adaptive Filters (IEEE, Dubrovnik, 2004)

31. AH Sayed, *Adaptive filters*. (Wiley, 2011)
32. JA Bazerque, GB Giannakis, Distributed spectrum sensing for cognitive radio networks by exploiting sparsity. *IEEE Trans. Signal Process.* **58**(3), 1847–1862 (2010)
33. S Chen, DL Donoho, MA Saunders, Atomic decomposition by basis pursuit. *SIAM J. Sci Comput.* **20**, 33–61 (1998)
34. Y Zakharov, T Tozer, J Adlard, Polynomial splines-approximation of Clarke's model. *IEEE Trans. Signal Process.* **52**(5), 1198–128 (2004)
35. PD Lorenzo, S Barbarossa, A Sayed, Distributed spectrum estimation for small cell networks based on sparse diffusion adaptation. *IEEE Signal Process. Lett.* **20**(123), 1261–1265 (2013)
36. ID Schizas, G Mateos, GB Giannakis, Distributed LMS for consensus-based in-network adaptive processing. *IEEE Trans. Signal Process.* **57**(6), 2365–2382 (2009)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com