https://doi.org/10.1016/j.sigpro.2017.05.010

A Variable Forgetting Factor Diffusion Recursive Least Squares Algorithm for Distributed Estimation

Y. J. Chu and C. M. Mak*

Department of Building Services Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

*corresponding author.
E-mail address: cheuk-ming.mak@polyu.edu.hk (C.M. Mak).

***Abstract*—Distributed recursive least squares (RLS) algorithms have superior convergence properties compared to the least mean squares (LMS) counterpart. However, with a fixed forgetting factor (FF), they are not suitable for tracking time-varying (TV) parameters. This paper proposes a novel diffusion variable FF RLS (Diff-VFF-RLS) algorithm based on a local polynomial modeling (LPM) of the unknown TV system. The diffusion RLS solution is derived analytically such that the estimation deviation from the true value is investigated. Based on the analysis and the LPM of the TV system, a new optimal VFF formula that tries to minimize the estimation deviation is obtained. Simulations are conducted to verify the theoretical analysis in terms of the steady-state mean square deviation (MSD) and the VFF formula. Results also show that the convergence and tracking performance of the proposed algorithm compares favorably with conventional ones.**

***Keywords*—Adaptive networks, diffusion RLS, MSD analysis, and VFF.**

I. INTRODUCTION

Distributed estimation over ad hoc adaptive networks is an attractive and challenging problem, where a collection of networked nodes can interact locally and adapt to track parameters of interest in a collaborative way. Much attention has been paid to distributed strategies and cooperation policies (combination weights) of adaptive networks [1]. Three frequently used strategies are based on the incremental [2][3], consensus [4], and diffusion [5]-[9] techniques. Diffusion strategies are particularly prominent due to their improved robustness and enlarged stable region [1]. We hence focus on this class of strategies in the rest of the paper.

Algorithms based on diffusion strategies update the estimate at each node in the network mainly in two steps [1]:

adaptation and combination. According to the order of the two steps, there are two different diffusion strategies called the adapt-then-combine (ATC) or the combine-then-adapt (CTA) diffusion. Current research has been focusing on the combination part that aims to improve estimation accuracy. A common choice for the combination weights is Metropolis rules [4]. A pioneer work [5] for the diffusion recursive least squares (RLS) algorithm seeks combination weights that minimize the estimation deviation through convex optimization. A similar rule for the diffusion least mean squares (LMS) algorithm [7] puts more weights to nodes that are well connected and less affected by noises [10].

As an alternative to the research on combination policy, research on adaptation develops techniques that accelerate the convergence and tracking capability for a given network. A variety of diffusion variable step-size (VSS) LMS-type algorithms have been proposed [11]-[13] that employ larger step-sizes to have fast rate when the algorithm is far from convergence while smaller ones when the algorithm is close to convergence. They can be derived from different optimization criteria and differ from the measure of convergence used. For the RLS-type algorithms [14]-[16], they usually achieve fast convergence and small mean square error (MSE) in stationary environment if a large forgetting factor (FF) is used; while in nonstationary environment with time-varying (TV) system parameters, a relatively small FF is required to facilitate fast parameter tracking. Thus, the FF needs to be made variable for TV systems. A number of variable FF (VFF) techniques [17]-[20] have been proposed for the single node estimation problems. For distributed RLS algorithms, however, only a few work that focuses on adaptation schemes has been conducted. It is probably due to the difficulty in understanding how the topology of adaptive networks that allows nodes to interact within the neighborhood affects the performance of the distributed adaptive algorithms. In stationary environments, a detailed analysis in terms of mean and mean square convergence performance has been carried out and a diffusion VFF RLS (Diff-VFF-RLS) algorithm, called the ATC-LCT-RLS algorithm, has been derived in a conference paper [21].

This paper aims to derive a new VFF version of the Diff-RLS algorithm, which has not been widely-discussed in the current literature. To this end, we start from finding an analytical expression for the solution to the Diff-RLS algorithm and discuss conditions for the solution to be unbiased from the true value in Section II.B. Based on the solution and a local polynomial (LP) modeling of the unknown system [20], the mean square deviation (MSD) of the Diff-RLS algorithm in TV environments is derived in Section II.C. An analytical formula of an optimal FF is hence obtained by balancing the bias and variance terms in MSD. The underlying mechanism of the combination strategy that improves the estimation performance of adaptive networks is illustrated using a simplified case at the end of Section II.C. The validity of the theoretical analysis and performance of the proposed ATC-VFF-RLS algorithm is examined in Section III and conclusions are drawn in Section IV. A summary of important symbols has been listed in Table I.

Overall, the main contribution of this paper is the proposed ATC-VFF-RLS algorithm with improved convergence and tracking capability, which is derived from the MSD analysis of the Diff-RLS algorithm in TV environments. The algorithm is different from the classical work [5] in that it deals with the tracking problems of the conventional Diff-RLS algorithm in [5]. Although similar assumptions have been used, the performance analysis for Diff-RLS algorithms in TV environments is new and different from [5]. The main differences include: 1) an analytical solution to the Diff-RLS algorithm is obtained instead of the mean convergence analysis in [5]; 2) the MSD in TV environments is derived from the analytical solution directly rather than the mean square performance analysis for constant systems [5]; and 3) conditions for an unbiased RLS solution and the optimal FF formula are not clarified in [5]. The current work is also different from our previous paper [20] which involves the LP modeling, since the adaptive network allows gradients and estimates combination within the neighborhood, which makes the performance analysis completely different from and much more complicated than that for the single node RLS algorithm in [20].

II. THE DIFF-VFF-RLS ALGORITHM

A. *Review of the ATC-RLS Algorithm*

We consider a network of K connected nodes, labeled k = 1, 2 … K. The neighborhood of node k is denoted by $\mathcal{N}_k$ and is connected to k by an edge. Any two neighboring nodes have the ability to share information through the edge. The kth node collects data { $d_k(n)$, $x_k(n)$ } that satisfy the linear model with unknown system vector $\boldsymbol{h}(n)$ of length L:

$$d_k(n) = \boldsymbol{h}^T(n)\boldsymbol{x}_k(n) + \eta_k(n) \tag{1}$$

where $\boldsymbol{x}_k(n) = [x_k(n),...,x_k(n-L+1)]^T$ is the input vector, and $\eta_k(n)$ is the additive noise independent of { $x_k(n)$ } and $\boldsymbol{h}(n)$. The objective of the network is to estimate the TV system $\boldsymbol{h}(n)$ by minimizing the LS cost function

$$\arg\min_{\boldsymbol{w}} \sum\nolimits_{i=L-1}^{n} \lambda_{n-i}(n) \sum\nolimits_{k=1}^{K} e_k^2(i) \tag{2}$$

where $e_k(i) = d_k(i) - \boldsymbol{w}^T \boldsymbol{x}_k(i)$, and $\lambda_{n-i}(n)$ is the weight (at time index n) of square errors (at time index i). $\lambda_{n-i}(n)$ serves as an exponential window since it decreases exponentially towards past data, and is calculated recursively by using a FF λ satisfying $0 << \lambda < 1$, i.e. $\lambda_{n-i}(n) = \lambda\lambda_{n-i-1}(n-1)$ with $\lambda_0(n) = 1$. λ can be made variable at each time index n, i.e. the VFF $\lambda(n)$, for a better tracking. To address problems in (2), an ATC-type strategy is developed [5] and takes the form as listed in Table II. The local estimate $\boldsymbol{\psi}_k(n)$ at node k shares the neighboring gradients through a positive weighting coefficient c_{lk} from node $l \in \mathcal{N}_k$ as shown in (3a) in Table II. The cluster averaged estimate, $\boldsymbol{w}_k(n)$, combines the local solutions over $l \in \mathcal{N}_k$ using weights a_{lk} as shown in (4) in Table II. The nonnegative scalars { c_{lk}, a_{lk} } are selected such that $\boldsymbol{C} = \{c_{lk}\}$ is right stochastic and $\boldsymbol{A} = \{a_{lk}\}$ is left stochastic [1].

B. Solution to the ATC-RLS Algorithm

To make the analysis tractable, we use $\lim_{n\to\infty} E[\boldsymbol{R}_{X\_k}(n)] = \lim_{n\to\infty} \sum_{i=L-1}^{n} \lambda_{n-i}(n) E[\boldsymbol{x}_k(i)\boldsymbol{x}_k^T(i)] = \frac{1-\lambda^{n-L+2}}{1-\lambda} \boldsymbol{R}_{x\_k} \approx \frac{1}{1-\lambda} \boldsymbol{R}_{x\_k}$ [22], which is based on the fact that the input signal is stationary and $E[\boldsymbol{x}_k(i)\boldsymbol{x}_k^T(i)] = \boldsymbol{R}_{x\_k}$. Then the assumption holds for large n

$$\lim_{n\to\infty} E[\boldsymbol{P}_k(n)] = \lim_{n\to\infty} E[\left(\boldsymbol{\mathcal{X}}^T(n)\boldsymbol{\Lambda}_k(n)\boldsymbol{\mathcal{X}}(n)\right)^{-1}] = (\textstyle\sum_{j=1}^{K} c_{jk} \boldsymbol{R}_{X\_j})^{-1} = (1-\lambda)(\sum_{j=1}^{K} c_{jk} \boldsymbol{R}_{x\_j})^{-1} = (1-\lambda)\boldsymbol{\mathcal{R}}_k^{-1} \tag{5}$$

where $\boldsymbol{P}_k(n) = \left(\boldsymbol{\mathcal{X}}^T(n)\boldsymbol{\Lambda}_k(n)\boldsymbol{\mathcal{X}}(n)\right)^{-1}$ is a concise expression for the inverse of the cluster averaged covariance matrix, and the notations such as $\boldsymbol{\mathcal{X}}(n)$ and $\boldsymbol{\Lambda}_k(n)$ have been defined in Table I. Eq. (5) is in consistency with the result in [5].

Under the assumption (5), the update for $\boldsymbol{P}_k(n)$ in (3b) is unnecessary such that (3a) can be expressed as $\boldsymbol{\psi}_k^{(l)}(n-1) = \boldsymbol{\psi}_k^{(l-1)}(n-1) + c_{lk}\boldsymbol{P}_k(n)\boldsymbol{x}_l(n)[d_l(n) - \boldsymbol{x}_l^T(n)\boldsymbol{\psi}_k^{(l-1)}(n-1)]$ for l =1,…,K, where $\boldsymbol{\psi}_k^{(l)}(n-1)$ is the adaptive filter for the lth loop during the incremental update with $\boldsymbol{\psi}_k^{(0)}(n-1) = \boldsymbol{w}_k(n-1)$ and $\boldsymbol{\psi}_k(n) = \boldsymbol{\psi}_k^{(K)}(n-1)$. Summing (3a) over l, we have

$$\boldsymbol{\psi}_k(n) = \boldsymbol{w}_k(n-1) + \boldsymbol{P}_k(n)\boldsymbol{X}^T(n)\boldsymbol{C}_k\boldsymbol{e}_k(n) \tag{6}$$

where the lth element of $\boldsymbol{e}_k(n)$ (K×1) reads $e_{k,l}(n) = d_l(n) - \boldsymbol{x}_l^T(n)\boldsymbol{\psi}_k^{(l-1)}(n-1)$. To further relates the cluster averaged estimate $\boldsymbol{w}_k(n-1)$ to the update of the local estimate $\boldsymbol{\psi}_k(n)$, we use $\boldsymbol{\psi}_k^{(l-1)}(n-1) = \boldsymbol{w}_k(n-1)$ for l = 1,…, K. Hence

$$\boldsymbol{\psi}_k(n) = \boldsymbol{w}_k(n-1) + \boldsymbol{P}_k(n)\boldsymbol{X}^T(n)\boldsymbol{C}_k[\boldsymbol{d}(n) - \boldsymbol{X}(n)\boldsymbol{w}_k(n-1)] = \lambda\boldsymbol{w}_k(n-1) + \boldsymbol{P}_k(n)\boldsymbol{X}^T(n)\boldsymbol{C}_k\boldsymbol{d}(n) \tag{7}$$

where $\boldsymbol{d}(n) = [d_1(n),..., d_K(n)]^T$ (K×1), $\boldsymbol{X}(n) = [\boldsymbol{x}_1(n) ... \boldsymbol{x}_K(n)]^T$ (K×L), and we have used the fact $\boldsymbol{P}_k^{-1}(n) = \lambda\boldsymbol{P}_k^{-1}(n-1) + \boldsymbol{X}^T(n)\boldsymbol{C}_k\boldsymbol{X}(n)$ [5] such that under the assumption (5) $\boldsymbol{P}_k(n)\boldsymbol{P}_k^{-1}(n)\boldsymbol{w}_k(n-1) = [\lambda + \boldsymbol{P}_k(n)\boldsymbol{X}^T(n)\boldsymbol{C}_k\boldsymbol{X}(n)]\boldsymbol{w}_k(n-1)$.

Substituting (7) into (4), the update for $\boldsymbol{w}_k(n)$ becomes

$$\begin{aligned}\boldsymbol{w}_k(n) &= \textstyle\sum_{l=1}^{K} a_{lk}[\lambda\boldsymbol{w}_l(n-1) + \boldsymbol{P}_l(n)\boldsymbol{X}^T(n)\boldsymbol{C}_l\boldsymbol{d}(n)] \\ &= \textstyle\sum_{l=1}^{K} a_{lk}[\sum_{j=1}^{K} \lambda^2 a_{jl}\boldsymbol{w}_j(n-2) + \sum_{j=1}^{K} \lambda a_{jl}\boldsymbol{P}_j(n)\boldsymbol{X}^T(n-1)\boldsymbol{C}_j\boldsymbol{d}(n-1) + \boldsymbol{P}_l(n)\boldsymbol{X}^T(n)\boldsymbol{C}_l\boldsymbol{d}(n)].\end{aligned} \tag{8}$$

It can be seen from (8) that $\boldsymbol{w}_k(n)$ combines $\lambda_{n-i}(n)[\boldsymbol{A}^{n+1-i}]_{lk}\boldsymbol{P}_l(n)\boldsymbol{X}^T(i)\boldsymbol{C}_l\boldsymbol{d}(i)$ over $l \in \mathcal{N}_k$ for $i = L-1$,…,n, where $[\cdot]_{lk}$ indicates the (lk)th entry of a matrix, and the other terms vanish as $n \to \infty$. Then, the solution to the Diff-RLS finds

$$\boldsymbol{w}_k(n)=\sum_{l=1}^{K}\boldsymbol{P}_l(n)\boldsymbol{\mathcal{X}}^T(n)\boldsymbol{\Lambda}_{lk}(n)\boldsymbol{\mathcal{d}}(n) \tag{9}$$

where $\boldsymbol{\Lambda}_{lk}(n)$ is a diagonal matrix of order $(n-L+2)K$, i.e., $\boldsymbol{\Lambda}_{lk}(n)=\text{diag}\{[\boldsymbol{A}]_{lk}\boldsymbol{C}_l,\lambda_1(n)[\boldsymbol{A}^2]_{lk}\boldsymbol{C}_l,\dots,\lambda_{n-L+1}(n)[\boldsymbol{A}^{n-L+2}]_{lk}\boldsymbol{C}_l\}$ and $\boldsymbol{\mathcal{d}}(n)=\text{col}\{\boldsymbol{d}(n),\dots,\boldsymbol{d}(L-1)\}$ $((n-L+2)K\times 1)$ is the output vector.

Note the nonnegative combination matrix $\boldsymbol{A}$, which satisfies $\boldsymbol{1}^T\boldsymbol{A}=\boldsymbol{1}^T$ with $\boldsymbol{1}$ a vector of unity entries [1], is irreducible and aperiodic such that $\boldsymbol{A}$ converges to a unique matrix as time progresses, i.e.

$$\lim_{n\to\infty}E[\boldsymbol{A}^n]=\boldsymbol{B}\text{, where }\boldsymbol{1}^T\boldsymbol{B}=\boldsymbol{1}^T. \tag{10}$$

Under the condition (10), which is in consistency with that for Diff-LMS algorithms [23], (9) becomes

$$\boldsymbol{w}_k(n)\overset{n\to\infty}{=}\sum_{l=1}^{K}[\boldsymbol{B}]_{lk}\boldsymbol{P}_l(n)\boldsymbol{\mathcal{X}}^T(n)\boldsymbol{\Lambda}_l(n)\boldsymbol{\mathcal{d}}(n). \tag{11}$$

Eq. (11) shows that the Diff-RLS solution is unbiased to a time-invariant system $\boldsymbol{h}(n)=\boldsymbol{h}_0$, where the desired signal can be written as $\boldsymbol{\mathcal{d}}(n)=\boldsymbol{\mathcal{X}}(n)\boldsymbol{h}_0$. In the rest of the paper, we carry out the MSD analysis based on (11).

C. Estimation Deviation Analysis

To start with, we assume that the system vector of length L is continuous and differentiable. It then admits a local first order polynomial expansion at time t_n [20], which reads

$$\boldsymbol{h}(t_m)=\boldsymbol{h}(t_n)+\tfrac{1}{1!}\boldsymbol{h}^{(1)}(t_n)(t_m-t_n)+\boldsymbol{r}(t_m-t_n) \tag{12}$$

where t_m belongs to a closed neighborhood of t_n, $\boldsymbol{h}^{(1)}(t_n)$ is the derivative of $\boldsymbol{h}(t)$ at t_n, and $\boldsymbol{r}(t_m-t_n)$ is the remainder of $o(t_m-t_n)$. $\boldsymbol{h}(t)$ is deterministic while both $\boldsymbol{h}^{(1)}(t_n)$ and $\boldsymbol{r}(t_m-t_n)$ are Lth-order random vectors and are assumed to be wide-sense stationary processes inside the neighborhood. Here, the channel coefficients are modeled locally as a first-order polynomial with additional stochastic variations. Substituting (12) into (1) with $t_n=nT_s$, we have

$$\begin{aligned}d_k(m)&=\boldsymbol{x}_k^T(m)\boldsymbol{h}(m)+\eta_k(m)\\&=\boldsymbol{x}_k^T(m)(\boldsymbol{h}(n)+\boldsymbol{h}^{(1)}(n)(m-n))+\eta_k(m)+\xi_k(m)\end{aligned} \tag{13}$$

where the time index indicates digital signals, e.g. $\boldsymbol{h}(n)$ is short for $\boldsymbol{h}(nT_s)$ with T_s the sampling period, and $\xi_k(m)=\boldsymbol{x}_k^T(m)\boldsymbol{r}(m-n)$. The random vector $\boldsymbol{h}^{(1)}(n)=\overline{\boldsymbol{h}^{(1)}}(n)+\delta\boldsymbol{h}^{(1)}(n)$ such that $\overline{\boldsymbol{h}^{(1)}}(n)=E[\boldsymbol{h}^{(1)}(n)]$ and $\delta\boldsymbol{h}^{(1)}(n)$ are, respectively, the mean and variance of $\boldsymbol{h}^{(1)}(n)$. Then, the observed signal vector can be expressed as

$$\boldsymbol{\mathcal{d}}(n)=\boldsymbol{\mathcal{X}}(n)\boldsymbol{h}(n)+\boldsymbol{D}_\tau(n)\boldsymbol{\mathcal{X}}(n)\boldsymbol{h}^{(1)}(n)+\boldsymbol{\xi}(n)+\boldsymbol{\eta}(n) \tag{14}$$

where $\boldsymbol{D}_\tau(n)=\text{diag}\{0\boldsymbol{I}_K,-1\boldsymbol{I}_K,\dots,-(n-L+1)\boldsymbol{I}_K\}$ is a diagonal matrix with $\boldsymbol{I}_K$ the identity matrix of order K,

$\boldsymbol{\xi}(n) = \mathrm{col}\{\xi(n),...,\xi(L-1)\}$ and $\boldsymbol{\eta}(n) = \mathrm{col}\{\boldsymbol{\eta}(n),...,\boldsymbol{\eta}(L-1)\}$ are, respectively, the residue and background noise vectors with $\boldsymbol{\xi}(n) = [\xi_1(n),...,\xi_K(n)]^T$ (K×1) and $\boldsymbol{\eta}(n) = [\eta_1(n),...,\eta_K(n)]^T$ (K×1).

Substituting (14) into (11) leads to a Diff-RLS solution:

$$\boldsymbol{w}_k(n) = \boldsymbol{h}(n) + \sum_{l=1}^{K}[\boldsymbol{B}]_{lk}\left(\boldsymbol{P}_l(n)\boldsymbol{R}_{\tau\_l}(n)\boldsymbol{h}^{(1)}(n) + \boldsymbol{P}_l(n)\boldsymbol{\mathcal{X}}^T(n)\boldsymbol{\Lambda}_l(n)(\boldsymbol{\xi}(n)+\boldsymbol{\eta}(n))\right) \tag{15}$$

where $\boldsymbol{R}_{\tau\_l}(n) = \boldsymbol{\mathcal{X}}^T(n)\boldsymbol{\Lambda}_l(n)\boldsymbol{D}_\tau(n)\boldsymbol{\mathcal{X}}(n)$. Since the remainder $\boldsymbol{r}(m-n)$ is assumed to be zero-mean and independent of the input [20], the expectation between $\boldsymbol{\mathcal{X}}(n)$ and $\boldsymbol{\xi}(n)$ is zero. Consequently,

$$E[\boldsymbol{w}_k(n)] = \boldsymbol{h}(n) + \sum_{l=1}^{K}[\boldsymbol{B}]_{lk}\boldsymbol{P}_l(n)\boldsymbol{R}_{\tau\_l}(n)\overline{\boldsymbol{h}^{(1)}}(n)\ . \tag{16}$$

It can be seen that if $\overline{\boldsymbol{h}^{(1)}}(n) = 0$, the optimal LS solution is identical to the system coefficients due to the property of the combination matrix (10). Then, we analyze the deviation of $\boldsymbol{w}_k(n)$ from $\boldsymbol{h}(n)$, i.e.

$$\boldsymbol{w}_k(n) - \boldsymbol{h}(n) = \{E[\boldsymbol{w}_k(n)] - \boldsymbol{h}(n)\} + \{\boldsymbol{w}_k(n) - E[\boldsymbol{w}_k(n)]\}\ . \tag{17}$$

The term in the first brackets corresponds to bias while the latter corresponds to variance. From (17), the MSD finds

$$J_{MSD\_k} = E[\|\boldsymbol{w}_k(n) - \boldsymbol{h}(n)\|_2^2] = \|E[\boldsymbol{w}_k(n)] - \boldsymbol{h}(n)\|_2^2 + E[\|\boldsymbol{w}_k(n) - E[\boldsymbol{w}_k(n)]\|_2^2]. \tag{18}$$

Using (15), (16), $\boldsymbol{R}_{\tau\_l}(n) = -(1-\lambda)^{-2}\boldsymbol{\mathcal{R}}_l$ according to Appendix A of [22], and the assumption in (5), we have

$$E[\boldsymbol{w}_k(n)] - \boldsymbol{h}(n) = -\tfrac{1}{1-\lambda}\sum_{l=1}^{K}[\boldsymbol{B}]_{lk}\overline{\boldsymbol{h}^{(1)}}(n) = -\tfrac{1}{1-\lambda}\overline{\boldsymbol{h}^{(1)}}(n) \tag{19}$$

$$\boldsymbol{w}_k(n) - E[\boldsymbol{w}_k(n)] = -\tfrac{1}{1-\lambda}\delta\boldsymbol{h}^{(1)}(n) + \sum_{l=1}^{K}[\boldsymbol{B}]_{lk}\boldsymbol{P}_l(n)\boldsymbol{\mathcal{X}}^T(n)\boldsymbol{\Lambda}_l(n)(\boldsymbol{\xi}(n)+\boldsymbol{\eta}(n)) \tag{20}$$

For the variance term (20), we further have $\mathrm{var}(\boldsymbol{w}_k(n) - E[\boldsymbol{w}_k(n)]) = \sigma_{\delta h}^2/(1-\lambda)^2 + Tr(\sum_{i=1}^{K}\sum_{j=1}^{K}[\boldsymbol{B}]_{ik}[\boldsymbol{B}]_{jk}\boldsymbol{P}_i(n)\boldsymbol{\mathcal{R}}_{ij}(n)\boldsymbol{P}_j(n))$, where $\sigma_{\delta h}^2 = E[\|\delta\boldsymbol{h}^{(1)}[n]\|^2]$, $\boldsymbol{\mathcal{R}}_{ij} = \sum_{l=1}^{K}(\sigma_{\eta l}^2 + \sigma_{\xi l}^2)c_{li}c_{lj}\hat{\boldsymbol{R}}_{X\_l}$ with $\hat{\boldsymbol{R}}_{X\_l} = \lim_{n\to\infty}\sum_{m=L-1}^{n}\lambda_{n-m}^2(n)\boldsymbol{x}_l(m)\boldsymbol{x}_l^T(m) = \tfrac{1}{1-\lambda^2}\boldsymbol{R}_{x\_l}$, and $\sigma_{\eta l}^2 = E[\eta_l^2(n)]$ and $\sigma_{\xi l}^2 = E[\xi_l^2(n)]$ are the variances of system and model noises at the lth node. Consequently,

$$J_{MSD\_k}(n) = \tfrac{1}{(1-\lambda)^2}\sigma_h^2(n) + \tfrac{1-\lambda}{1+\lambda}T_k \tag{21}$$

where $T_k = Tr(\sum_{i=1}^{K}\sum_{j=1}^{K}[\boldsymbol{B}]_{ik}[\boldsymbol{B}]_{jk}\boldsymbol{\mathcal{R}}_i^{-1}(\sum_{l=1}^{K}\sigma_{\Sigma l}^2 c_{li}c_{lj}\boldsymbol{R}_{x\_l})\boldsymbol{\mathcal{R}}_j^{-1})$ with $\sigma_{\Sigma l}^2 = \sigma_{\eta l}^2 + \sigma_{\xi l}^2$ provides information on signal to noise ratio (SNR) and is irrelative to FF, and $\sigma_h^2(n) = \|\overline{\boldsymbol{h}}^{(1)}(n)\|^2 + \sigma_{\delta h}^2$ is the system variance. To minimize (2), one takes the derivative of $J_{MSD} = \frac{1}{K}\sum_{k=1}^{K}J_{MSD\_k}(n)$ and let it equal to zero to get

$$\frac{\overline{T}}{(1+\lambda)^2} = \frac{\sigma_h^2(n)}{(1-\lambda)^3} \tag{22}$$

where $\overline{T} = \frac{1}{K}\sum_{k=1}^{K}T_k$. To proceed further, we let $\mu = \frac{1+\lambda}{1-\lambda}$. Then, (22) reduces to $\mu^2(\mu+1) = 2\overline{T}/\sigma_h^2(n)$. For moderate and large values of λ, say, 0.5< λ <1, μ satisfies μ >>1 and we can use the assumption $\mu+1 \approx \mu$ for computational efficiency [20]. Under this assumption, $\mu = \left(2\overline{T} / \sigma_h^2(n)\right)^{\frac{1}{3}}$ and the optimal FF is determined as

$$\lambda_{opt} = (\mu - 1)/(\mu + 1), \text{ if } \lambda_{opt} > 0 . \tag{23}$$

Then, a VFF $\lambda(n)$ can be calculated from (23) at each time index n, i.e.

$$\lambda(n) = [\left(2\overline{T} / \sigma_h^2(n)\right)^{\frac{1}{3}} - 1]/[\left(2\overline{T} / \sigma_h^2(n)\right)^{\frac{1}{3}} + 1], \text{ if } \lambda(n) > 0 . \tag{24}$$

In (24), the estimation of T_k is computationally consuming and should be simplified. We hence assume that the weighted noise variance $\sigma_{\Sigma l}^2 c_{li}$ at neighboring nodes is close to each other [10] and equal to $\overline{\sigma}_{W\_i}^2 = \frac{1}{K}\sum_{l=1}^{K} \sigma_{\Sigma l}^2 c_{li}$. Then, $T_k = \sum_{j=1}^{K} [\boldsymbol{B}]_{jk} Tr(\sum_{i=1}^{K} \overline{\sigma}_{W\_i}^2 [\boldsymbol{B}]_{ik} \boldsymbol{\mathcal{R}}_i^{-1})$, where $\boldsymbol{\mathcal{R}}_i^{-1}$ can be estimated by means of $\boldsymbol{P}_i(n)$, i.e. $\boldsymbol{\mathcal{R}}_i^{-1} \cong \boldsymbol{P}_i(n) \sum_{i=0}^{n} \lambda(n)$. This process needs $O(KL^2)$ times of multiplications. Since $\sigma_h^2(n)$ and $\sigma_{\Sigma l}^2(n)$ can be estimated by using a fixed FF [20] that is computationally efficient compared to the calculation of T_k, the multiplication required by (24) is $O(K^2L^2)$, which is in the same order of the Diff-RLS algorithm. The arithmetic complexity of the propose algorithm is listed in Table III. It should be mentioned that although (24) is derived from modeling the differentiable system as a LP, the proposed VFF scheme can also accelerate the tracking speed of Diff-RLS algorithms significantly for sudden change systems due to the on-line estimation methods for noise variances. It has been shown in simulation results.

We now discuss a special case when both noise variances and input covariance matrices are the same for each node:

$$\sigma_{\eta l}^2 = \sigma_\eta^2 ,\ \sigma_{\xi l}^2 = \sigma_\xi^2 ,\ \boldsymbol{R}_{x\_l} = \boldsymbol{R}_x , \text{ for } l = 1,2,\ldots,K. \tag{25}$$

In this case, T_k reduces to $T_k = \rho_k \sigma_\Sigma^2 Tr(\boldsymbol{R}_x^{-1})$ and $\rho_k = \sum_{i=1}^{K}\sum_{j=1}^{K} [\boldsymbol{B}]_{ik} [\boldsymbol{B}]_{jk} (\sum_{l=1}^{K} c_{li} c_{lj})$. It can be seen that ρ_k is a constant smaller than 1, indicating that SNR at the kth node is increased by using the diffusion strategy. This also explains the improved performance of distributed estimation.

III. Simulation Results

In this section, we evaluate the proposed ATC-VFF-RLS algorithm and its performance analysis. All results are obtained by averaging 50 Monte-Carlo simulations if not specified.

A. Evaluation of VFF Formula for Random Walk Model

The proposed VFF formula in (23) is evaluated by the identification of a random walk system: $\boldsymbol{h}(n+1) = \boldsymbol{h}(n) + \boldsymbol{\delta}(n)$, where $\boldsymbol{\delta}(n)$ is a zero-mean white Gaussian random vector with covariance matrix $4\times10^{-5}\boldsymbol{I}_{10}$ and the initial value of the channel is $\boldsymbol{h}_0 = [-1,1,-1,\ldots,1]$ of length L = 10. The measurements are generated according to (1), where $x_k(n)$ is a first-order auto-regressive (AR) process $x_k(n+1)=0.9x_k(n)+g_k(n)$ with $g_k(n)$ a zero-mean Gaussian process. The network has a total of K = 5 nodes. The variances of { $x_k(n)$ } at each node are set to 1, 1, 0.5, 0.5 and 2. Metropolis weights [4] are used for both the selection matrix $\boldsymbol{C}$ and the combination matrix $\boldsymbol{A}$. The variance of noises are selected so as to achieve an averaged SNR of 0, 10 and 20 dB.

ATC-RLS algorithms with different but fixed FF values in the range [0.5, 0.999] are examined. The simulated MSD curves are compared with theoretical predictions in Fig. 1. It can be seen that the simulated and theoretical results for MSD are in good agreement. It also shows that MSD is slightly overestimated for large FFs since the approximations for covariance matrices are used in (5), but it does not affect the selection of the optimal FF significantly. Both simulated and theoretical results illustrate that the optimal FF decreases slightly with the SNR. It indicates how the FF balances between the tracking speed and estimation accuracy in noisy environments. Next, we examine the performance of the VFF formula with noise variance mismatches, where the true variance ratio $b=\sigma_h^2/\overline{T}$ is replaced by the estimated values $\hat{b}=\hat{\sigma}_h^2/\hat{T}$. The predicted MSDs at the variance ratios $\hat{b}=10b$, $0.1b$ are marked by 'Δ' in Fig. 1. The results show that the FF formula is not particularly sensitive to variance mismatches. If noise variance information is not exactly known in practical applications, (24) can provide a good reference for FF selection.

B. Evaluation of the ATC-VFF-RLS Algorithm

In this experiment, a larger network with K = 20 nodes is considered. Metropolis weights are also used for the selection and the combination matrices. The system to be identified also follows a random walk model with the initial value $\boldsymbol{h}_0$, and it has a sudden jump to $\boldsymbol{h}_1=[1,1,\ldots,1]$ of the same length at the 800th sample. The covariance matrix for this random walk process is also $4\times10^{-5}\boldsymbol{I}_{10}$. The averaged SNR is set to 0, 10 and 20 dB. AR sequences are also used as inputs to excite the system, i.e. $x_k(n+1)=0.5x_k(n)+g_k(n)$ and the input variances at each node rang from 0.5 to 2. The algorithms under test include ATC-LMS in [6], ATC-VSS-LMS in [13], which outperforms other VSS Diff-LMS algorithms, ATC-RLS in [5], and ATC-LCT-RLS in [21].

The step-size for ATC-LMS is set to 0.02 while the FF for ATC-RLS is set to 0.98. Since it is difficult to choose parameters so as to let the two algorithms converge to a similar steady-state MSD in TV environment, we just follow a selection rule for time-invariant systems. The user parameters for ATC-LCT-RLS are selected as suggested in [21] except that the upper and lower bounds of the FF are tuned so as to provide the best performance at SNR = 0 dB. For the proposed ATC-VFF-RLS, we use the estimated input and noise variances as suggested in [20] for the calculation of the FF (24). The simulation results are shown in Fig. 2. It can be seen that ATC-RLS has a much faster convergence than ATC-LMS algorithm. However, the tracking capability of ATC-RLS is even worse than its LMS counterpart. The ATC-VSS-LMS algorithm has significantly improved convergence and tracking performance over its fixed step-size version, especially at higher SNRs. The performance of the ATC-LCT-RLS algorithm is comparable with ATC-VFF-RLS at SNR = 0 dB, but is slightly affected by the change of noise variances. The proposed ATC-VFF-RLS algorithm in each case converges faster to a much lower steady-state MSD than the ATC-LCT-RLS at the cost of a higher complexity for the calculation of FFs, a comparison of which is presented in Table III. To further examine the VFF strategies of the two VFF diffusion RLS algorithms, the FF curves for ATC-LCT-RLS and ATC-VFF-RLS are shown in Fig. 3. It can be seen that, in time-varying systems, the proposed VFF formula (24) can converge quickly to an appropriate value so that faster convergence rate and smaller estimation deviation can be achieved.

IV. Conclusion

A new VFF diffusion RLS algorithm has been presented that is derived from the MSD performance analysis of the Diff-RLS algorithm for channels whose coefficients are modeled by LP. Simulations show that the theoretical and experimental results are in good agreement with each other. Comparison with other diffusion algorithms illustrates the improved convergence and tracking performance of the proposed algorithm.

Acknowledgment

The work described in this study was fully supported by a grant from the Hong Kong Polytechnic University (The Hong Kong Polytechnic University Postdoctoral Fellowships Scheme, G-YW0L). The first author is indebted to Prof. S. C. Chan for his inspiration for writing this paper.

References

[1] A. H. Sayed, “Adaptive networks,” *Proc. IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.

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

[3] Y. Liu and K. S. Tang, “Enhanced incremental LMS with norm constraints for distributed in-network estimation,” *Signal Process*., vol. 90, no. 8, pp. 2621–2627, Aug. 2010.

[4] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” *Syst. Control Lett*., vol. 53, no. 1, pp. 65–78, Sep. 2004.

[5] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, “Diffusion recursive least-squares for distributed estimation over adaptive networks,” *IEEE*

Trans. Signal Process., vol. 56, no. 5, pp. 1865–1877, May 2008.

[6] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Process*., vol. 56, no. 7, pp. 3122–3136, Jul. 2008.

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

[8] A. Bertrand and M. Moonen, "Distributed signal estimation in sensor networks where nodes have different interests," *Signal Process.*, vol. 92, no. 7, pp. 1679–1690, Jul. 2013.

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

[10] J. Chan, C. Richard, and A. H. Sayed, "Diffusion LMS over multitask networks," *IEEE Trans. Signal Process.*, vol. 63, no. 11, pp. 2733–2748, Jun. 2015.

[11] M. O. B. Saeed and A. Zerguine, "A new variable step-size strategy for adaptive networks," in *Proc. Asilamar Conf. Signals, Syst., Comput*., Pacific Grove, CA, Nov. 2011, pp. 312–315.

[12] A. Khalili, A. Rastegarnia, J. A. Chambers, and W. M. Bazzi, "An optimum step-size assignment for incremental LMS adaptive networks based on average convergence rate constraint," *AEU—Int. Electron. Commun.*, vol. 67, no. 3, pp. 263–268, Mar. 2013.

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

[14] H. C. So, "A comparative study of three recursive least-squares algorithms for single-tone frequency tracking," *Signal Process.*, vol. 83, no. 9, pp. 2059–2062, Sep. 2003.

[15] A. H. Sayed and N. J. Hoboken, *Adaptive Filters*, P John Wiley & Sons, NJ, 2008.

[16] Y. J. Chu, and C. M. Mak, "A new QR decomposition-based RLS algorithm using the Split Bregman method for L1-regularized problems," *Signal Process.*, vol. 128, pp. 303–308, Nov. 2016.

[17] C. F. So, S. C. Ng, and S. H. Leung, "Gradient based variable forgetting factor RLS algorithm," *Signal Process.*, vol. 83, no. 6, pp. 1163–1175, Jun. 2003.

[18] C. Paleologu, J. Benesty, and S. Ciochina, "A robust variable forgetting factor recursive least-squares algorithm for system identification," *IEEE Signal Process. Lett.*, vol. 15, pp. 597–600, 2008.

[19] B. Qin, Y. Cai, B. Champagne, R. C. de Lamare, M. Zhao, "A low-complexity variable forgetting factor constant modulus RLS algorithm for blind adaptive beamforming," *Signal Process.*, vol. 105, pp. 277–282, 2014.

[20] Y. J. Chu and S. C. Chan, "A new local polynomial modeling-based variable forgetting factor RLS algorithm and its acoustic applications," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 11, pp. 2059–2069, Nov. 2015.

[21] L. Zhang, Y. Cai, C. Li, R. C. de Lamare, and M. Zhao, "Low-complexity correlated time-averaged variable forgetting factor mechanism for diffusion RLS algorithm in sensor networks," in *Proc. IEEE SAM 2016*, Rio de Janeiro, Brazil, 10-13 Jul. 2016, pp. 1–5.

[22] S. C. Chan, Y. J. Chu, Z. G. Zhang, and K. M. Tsui, "A new variable regularized QR decomposition-based recursive least M-estimate algorithm—performance analysis and acoustic applications," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 5, pp. 907–922, May 2013.

[23] X. Zhao and A. H. Sayed, "Performance limits for distributed estimation over LMS adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5107–5113, Oct. 2012.

TABLES

TABLE I SUMMARY OF IMPORTANT SYMBOLS (IN ORDER OF APPEARANCE IN THE PAPER)

Eq. (1):	K	Number of nodes
	$\{d_k(n), x_k(n), \eta_k(n)\}$	Desired, input, and noise signals received at the kth node
	$\boldsymbol{h}(n)$	Unknown system vector of length L
	L	Length of $\boldsymbol{h}(n)$ and adaptive filters
	$\boldsymbol{x}_k(n)$	$=[x_k(n),\ldots,x_k(n-L+1)]^T$, input vector for the kth node of length L
Eq. (2):	λ or $\lambda(n)$	Forgetting factor or variable forgetting factor
	$\lambda_{n-i}(n)$	$=\lambda\lambda_{n-i-1}(n-1)$, weight (at time index n) of square errors (at time index i)
Eq. (3):	$\boldsymbol{\psi}_k(n)$	Local estimate of the unknown system of length L
	$\boldsymbol{C}$	$=\{c_{lk}\}$ $(K\times K)$, selection matrix which is right stochastic
Eq. (4):	$\boldsymbol{w}_k(n)$	Cluster averaged estimate of the unknown system of length L
	$\boldsymbol{A}$	$=\{a_{lk}\}$ $(K\times K)$, combination matrix which is left stochastic
Eq. (5):	$\boldsymbol{R}_{x\_k}(n)$	$=\sum_{i=L-1}^{n}\lambda_{n-i}(n)E[\boldsymbol{x}_k(i)\boldsymbol{x}_k^T(i)]$, weighted input covariance matrix at node k
	$\boldsymbol{R}_{x\_k}$	$=E[\boldsymbol{x}_k(i)\boldsymbol{x}_k^T(i)]$, input covariance matrix at node k
	$\boldsymbol{\mathcal{R}}_k$	$=\sum_{j=1}^{K}c_{jk}\boldsymbol{R}_{x\_j}$, cluster averaged input covariance matrix at node k
	$\boldsymbol{P}_k(n)$	$=\left(\boldsymbol{\mathcal{X}}^T(n)\boldsymbol{\Lambda}_k(n)\boldsymbol{\mathcal{X}}(n)\right)^{-1}$, the inverse of cluster averaged input covariance matrix at k
	col$\{\boldsymbol{U}_1,\ldots,\boldsymbol{U}_N\}$	A column of N elements with the nth element equal to $\boldsymbol{U}_n$
	$\boldsymbol{X}(n)$	$=[\boldsymbol{x}_1(n)\ldots\boldsymbol{x}_K(n)]^T$ $(K\times L)$, input vector of all nodes at time index n
	$\boldsymbol{\mathcal{X}}(n)$	$=\mathrm{col}\{\boldsymbol{X}(n),\ldots,\boldsymbol{X}(L-1)\}$ $((n-L+2)K\times L)$, input signal matrix
	$\boldsymbol{C}_k$	$=\mathrm{diag}\{c_{1k},c_{2k},\ldots,c_{Kk}\}$, a diagonal matrix of order K
	$\boldsymbol{\Lambda}_k(n)$	$=\mathrm{diag}\{\boldsymbol{C}_k,\lambda_1(n)\boldsymbol{C}_k,\ldots,\lambda_{n-L+1}(n)\boldsymbol{C}_k\}$, a diagonal matrix of order $(n-L+2)K$
Eq. (7):	$\boldsymbol{d}(n)$	$=[d_1(n),\ldots,d_K(n)]^T$ $(K\times 1)$, desired signal vector at time index n
Eq. (9):	$\boldsymbol{\mathscr{d}}(n)$	$=\mathrm{col}\{\boldsymbol{d}(n),\ldots,\boldsymbol{d}(L-1)\}$ $((n-L+2)K)\times 1$, desired signal vector
	$\boldsymbol{\Lambda}_{lk}(n)$	$=\mathrm{diag}\{[\boldsymbol{A}]_{lk}\boldsymbol{C}_l,\lambda_1(n)[\boldsymbol{A}^2]_{lk}\boldsymbol{C}_l,\ldots,\lambda_{n-L+1}(n)[\boldsymbol{A}^{n-L+2}]_{lk}\boldsymbol{C}_l\}$, a diagonal matrix of order $(n-L+2)K$
Eq. (12) or (13):	$\boldsymbol{h}^{(1)}(t_n)$ or $\boldsymbol{h}^{(1)}(n)$	$=\overline{\boldsymbol{h}}^{(1)}(n)+\delta\boldsymbol{h}^{(1)}(n)$, first order derivative of the unknown system with $\overline{\boldsymbol{h}}^{(1)}(n)=E[\boldsymbol{h}^{(1)}(n)]$ and $\delta\boldsymbol{h}^{(1)}(n)$, respectively, the mean and variance of $\boldsymbol{h}^{(1)}(n)$
Eq. (14):	$\boldsymbol{D}_\tau(n)$	$=\mathrm{diag}\{0\boldsymbol{I}_K,-1\boldsymbol{I}_K,\ldots,-(n-L+1)\boldsymbol{I}_K\}$, a diagonal matrix of order $(n-L+2)K$
	$\boldsymbol{\xi}(n)$	$=\mathrm{col}\{\xi(n),\ldots,\xi(L-1)\}$ $((n-L+2)K)\times 1$, modeling residue vector with $\xi(n)=[\xi_1(n),\ldots,\xi_K(n)]^T$
	$\boldsymbol{\eta}(n)$	$=\mathrm{col}\{\boldsymbol{\eta}(n),\ldots,\boldsymbol{\eta}(L-1)\}$ $((n-L+2)K)\times 1$, background noise vector with $\boldsymbol{\eta}(n)=[\eta_1(n),\ldots,\eta_K(n)]^T$
Eq. (15):	$\boldsymbol{R}_{\tau\_l}(n)$	$=\boldsymbol{\mathcal{X}}^T(n)\boldsymbol{\Lambda}_l(n)\boldsymbol{D}_\tau(n)\boldsymbol{\mathcal{X}}(n)$ $(L\times L)$
Eq. (20):	$\hat{\boldsymbol{R}}_{x\_l}$	$=\lim_{n\to\infty}\sum_{m=L-1}^{n}\lambda_{n-m}^2(n)\boldsymbol{x}_l(m)\boldsymbol{x}_l^T(m)=\frac{1}{1-\lambda^2}\boldsymbol{R}_{x\_l}$
	$\boldsymbol{\mathcal{R}}_{ij}$	$=\sum_{l=1}^{K}(\sigma_{\eta l}^2+\sigma_{\xi l}^2)c_{li}c_{lj}\hat{\boldsymbol{R}}_{x\_l}$
	$\sigma_{\delta h}^2$	$=E[\Vert\delta\boldsymbol{h}^{(1)}[n]\Vert^2]$, variance of $\boldsymbol{h}^{(1)}(n)$
	$\sigma_{\eta l}^2$	$=E[\eta_l^2(n)]$, variance of the background noise
	$\sigma_{\xi l}^2$	$=E[\xi_l^2(n)]$, variance of the modeling residue
Eq. (21):	$\sigma_h^2(n)$	$=\Vert\overline{\boldsymbol{h}}^{(1)}(n)\Vert^2+\sigma_{\delta h}^2$, variance of the unknown system
	$\sigma_{\Sigma l}^2$	$=\sigma_{\eta l}^2+\sigma_{\xi l}^2$

TABLE II DIFFUSION ATC-RLS STRATEGY

Initialization for node *k*:
$\boldsymbol{P}_k(0) = \delta \boldsymbol{I}_L$, with δ a small positive constant;
$\boldsymbol{w}_k(0) = \boldsymbol{0}$ is a null vector.
Update:
Given $\boldsymbol{\psi}_k(n) = \boldsymbol{w}_k(n-1)$, $\boldsymbol{P}_k(n) = \lambda(n)^{-1}\boldsymbol{P}_k(n-1)$
Adaptation at time *n* for $l \in \mathcal{N}_k$:
$\boldsymbol{\psi}_k(n) \leftarrow \boldsymbol{\psi}_k(n) + \frac{c_{lk}\boldsymbol{P}_k(n)\boldsymbol{x}_l(n)}{1 + c_{lk}\boldsymbol{x}_l^T(n)\boldsymbol{P}_k(n)\boldsymbol{x}_l(n)}(d_l(n) - \boldsymbol{x}_l^T(n)\boldsymbol{\psi}_k(n))$ (3a)
$\boldsymbol{P}_k(n) \leftarrow \boldsymbol{P}_k(n) - \frac{c_{lk}\boldsymbol{P}_k(n)\boldsymbol{x}_l(n)\boldsymbol{x}_l^T(n)\boldsymbol{P}_k(n)}{1 + c_{lk}\boldsymbol{x}_l^T(n)\boldsymbol{P}_k(n)\boldsymbol{x}_l(n)}$ (3b)
End of *l*
Combination:
$\boldsymbol{w}_k(n) = \sum_{l \in \mathcal{N}_k} a_{lk}\boldsymbol{\psi}_l(n)$ (4)

TABLE III ARITHMETIC COMPLEXITIES OF TWO DIFF-VFF-RLS ALGORITHMS

	ATC-VFF-RLS		ATC-LCT-RLS
$\boldsymbol{w}_k(n)$	$O(K^2L^2)$		$O(K^2L^2)$
$\lambda(n)$	$\overline{T}$	$O(K^2L^2)$	$6K$
	$\overline{\sigma}^2_{w\_i}$	$O(K^2)$	
	$\hat{\sigma}^2_h$	3	

K: number of nodes; *L*: filter length.

TABLE IV ACCURACY ANALYSIS OF THE THEORETICAL PREDICTIONS FOR THE ATC-RLS ALGORITHM IN FIG. 1

SNR	Point 1	Point 2	Point 3	Point 4	Point 5
0 dB	2.4	2.3	<u>5.2</u>	7.5	>10
10 dB	2.3	2.2	1.9	<u>6.6</u>	8.9
20 dB	1.7	1.9	2.2	3.0	<u>5.8</u>

Point 1 to Point 5 correspond to the simulated results in Fig. 1 (from left to right)

Figure Caption

Fig. 1. Simulated and theoretical results of the steady-state MSDs for the ATC-RLS algorithm using different FFs with the colored input at SNR = (a) 0dB (b) 10dB (c) 20dB. $K = 5$, $L = 10$.

Fig. 2. The MSD curves of different ATC algorithms with the colored input at SNR = (a) 0dB (b) 10dB and (c) 20dB. $K = 20$, $L = 10$.

Fig. 3. The FF curves of ATC-LCT-RLS and ATC-VFF-RLS algorithms with the colored input at SNR = (a) 0dB (b) 10dB and (c) 20dB. $K = 20$, $L = 10$.

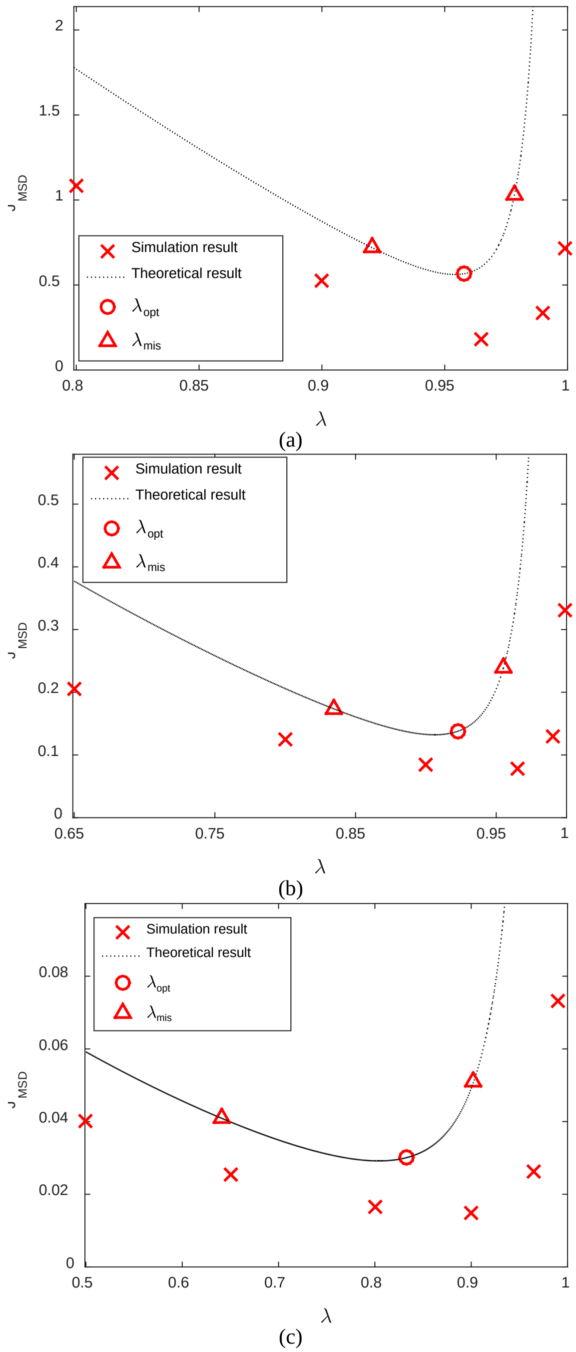
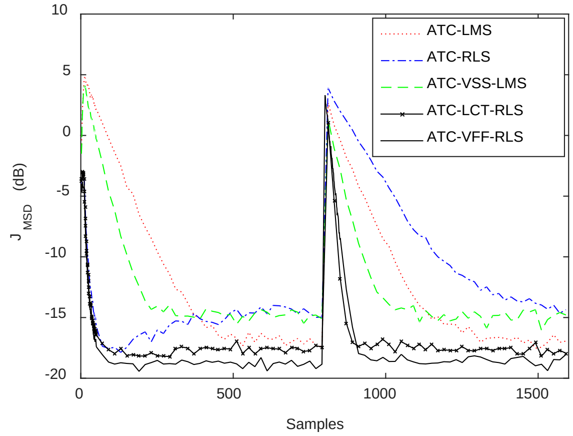
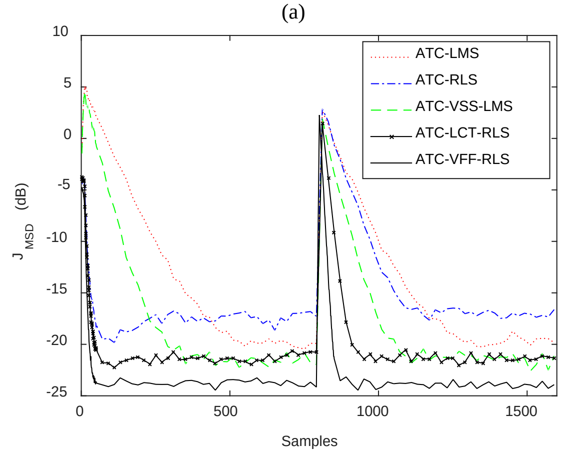


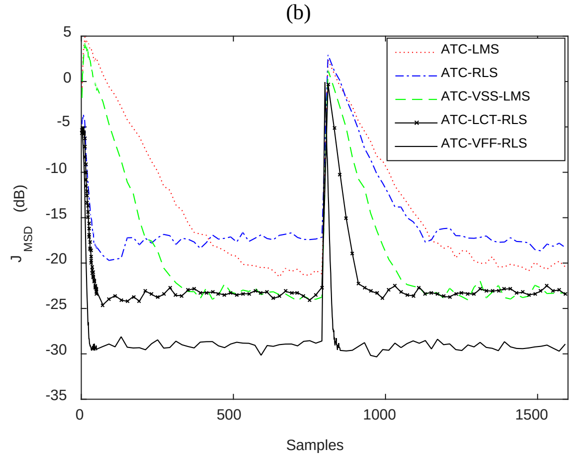
Fig. 1. Simulated and theoretical results of the steady-state MSDs for the ATC-RLS algorithm using different FFs with the colored input at SNR = (a) 0dB (b) 10dB (c) 20dB. K = 5, L = 10.



(a)



(b)



(c)

Fig. 2. The MSD curves of different ATC algorithms with the colored input at SNR = (a) 0dB (b) 10dB and (c) 20dB. K = 20, L = 10.

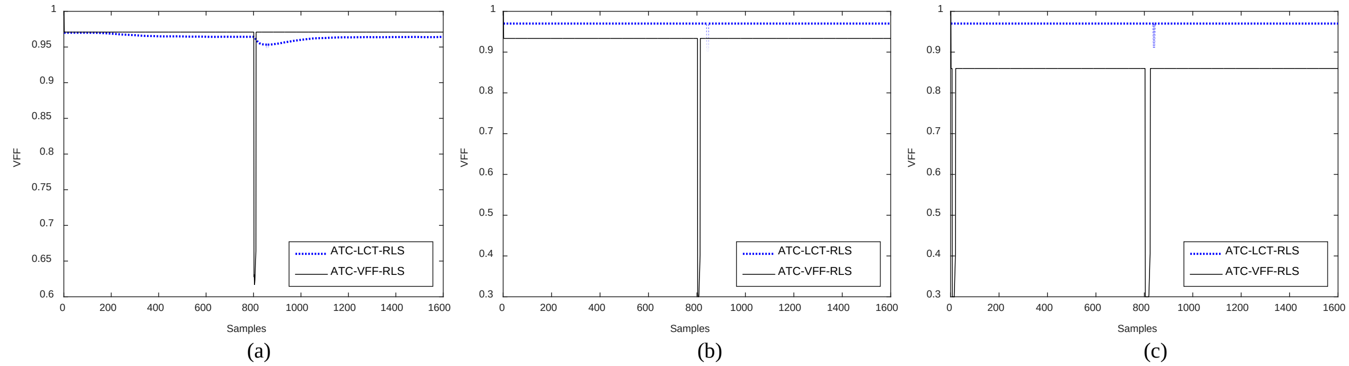


Fig. 3. The FF curves of ATC-LCT-RLS and ATC-VFF-RLS algorithms with the colored input at SNR = (a) 0dB (b) 10dB and (c) 20dB. $K = 20$, $L = 10$.