Zhang YG, Wang CC, Zhao L, Chambers JA.

**A Spatial Diffusion Strategy for Tap-Length Estimation Over Adaptive
Networks.**

***IEEE Transactions on Signal Processing* 2015, 63(17), 4487-4501**

Copyright:

© 2015 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

DOI link to article:

<http://dx.doi.org/10.1109/TSP.2015.2440182>

Date deposited:

08/02/2017

Choose a building block.

A Spatial Diffusion Strategy for Tap-Length Estimation Over Adaptive Networks

Yonggang Zhang, Chengcheng Wang, Lin Zhao and Jonathon A. Chambers

Abstract—We consider the distributed estimation problem, where a set of nodes is required to collectively estimate some parameter vector of interest with unknown or variable tap-length. In practice, a sufficiently large filter length is utilized in such contexts to avoid a large excess mean square error at steady state, while resulting in slower convergence rate and increased computations. In this work we motivate and propose a new diffusion variable tap-length algorithm, which is able to track tap-length changes during the convergence process. Theoretical analyses are provided in terms of steady-state performance and convergence performance, which are verified by simulation results. Some general criteria for parameter selections are also given according to the performance analyses. Numerical simulations demonstrate the efficiency of the proposed algorithm as compared with existing techniques, and great robustness to parameter settings provided the parameter choice guidelines are satisfied.

Keywords—Adaptive networks, variable tap-length algorithm, diffusion algorithm, distributed estimation

I. INTRODUCTION

Distributed algorithms have been proposed for a wide variety of applications, ranging from precision agriculture to environmental monitoring, military surveillance, and the modeling of self-organization in biological networks, since they remove the need for a fusion center and reduce the amount of communication compared to centralized solutions [1].

In the problem of distributed estimation over adaptive networks, a set of nodes is required to collectively estimate some parameters of interest from noisy measurements. Each node in the network is allowed to share information locally with its neighbors, and processing is distributed among all nodes in the network [1], [2]. Distributed processing schemes were firstly proposed in [3], [4], [5] based on incremental strategies. Several variations have been developed afterwards, including incremental least mean square (LMS) [6] and incremental recursive least squares (RLS) [7]. However, the incremental approach has the disadvantage of being non-robust to node and link failure, since data are transmitted and processed in

a cyclic manner [8]. In a diffusion protocol, every node has access to the measurements and estimates from a subset of adjacent nodes, which is called the neighborhood of the node, and assigns different weights to information received from the nodes in the neighborhood. As a result, the estimate at each node is a function of both its temporal data as well as the spatial data across the neighbors, and furthermore an adaptive network is obtained that is able to respond in real-time to the temporal and spatial variations in the statistical profile of the data [9]. Diffusion-based distributed algorithms include diffusion LMS [2], [9], [10], diffusion RLS [11], and diffusion Kalman filters and smoothers [12]. Other research works focus on the performance analysis of diffusion strategies [8], [13], [14], [15] and the study of optimal combination weights [16]. Moreover, in schemes [1] and [17], a temporal dimension is added to the processing at the nodes in addition to the spatial dimension. This results in a lower steady-state error, but additional computations and extra memory slots are required [1][17]. All of these distributed algorithms use a common assumption that the tap-length or the dimension of the unknown parameter vector is known a priori.

However, in some contexts, the optimal tap-length is always unknown or variable. For example, in the application of system identification, for a given set of channel conditions, no rules exist to exactly predict how many taps the filter should have to attain the lowest possible mean square error (MSE) level, particularly with a time-varying system [18]. Thus in most practical designs, unfortunately, the tap-length is usually fixed at a sufficiently large value to avoid a large excess mean square error (EMSE) arising from under-modeling. However, that may cause slower convergence rate and increased computations. Therefore, an optimum tap-length is required to best balance the steady-state error and the convergence rate. This problem has been previously referred to in the stand-alone adaptive strategies [18], [19], [20], [21]. Among these, the fractional tap-length normalized LMS (FTNLMS) algorithm in [19] is of particular interest, in that the concept of FT facilitates tap-length update and that good performance is confirmed for stationary input signals [21]. Several related work has further been proposed based on the FTNLMS algorithm [22], [23], [24], [25], [26]. Recently, tap-length estimation problem has been studied in the context of a distributed network in [27] and [28], where an incremental protocol and a diffusion protocol are utilized respectively. Unfortunately, in [28] only a global cost function is proposed to define the optimal tap-length estimate. In addition, a combine-then-adapt (CTA) policy is used in the diffusion strategy, where during the combination step, tap-length estimates are combined directly

This work was supported by the National Natural Science Foundation of China (61001154, 61201409 and 61371173), China Postdoctoral Science Foundation (2013M530147 and 2014T70309), Heilongjiang Postdoctoral Fund (LBH-Z13052), China Scholarship Council, and Fundamental Research Funds for the Central Universities of Harbin Engineering University (HEUCFX41307).

Yonggang Zhang, Chengcheng Wang and Lin Zhao are with the College of Automation, Harbin Engineering University, Harbin, Heilongjiang 150001 China (e-mail: zhangyg@hrbeu.edu.cn; wangchengcheng@hrbeu.edu.cn; zhaolin@hrbeu.edu.cn).

J. A. Chambers is with the ComS²IP Group, School of Electrical and Electronic Engineering, Newcastle University, Newcastle upon Tyne, NE1 7RU, UK (e-mail: J.A.Chambers@lboro.ac.uk).

in the neighborhood, but there is no cost function indicating that; during the adaptation step, only data from the node itself are exploited to update the tap weights and tap-length. Finally, comparisons are needed to select the largest tap-length within the neighborhood, then the tap-weight vectors will be combined and updated based on that tap-length, therefore, large computation complexity is required.

The purpose of this paper is therefore to motivate and develop a new distributed strategy that is able to seek the optimal tap-length estimator in real-time through local interactions and rely on a single time-scale (namely, the observation time-scale). Thus, consider a set of nodes that are distributed over a certain region in space, as is shown in Fig.1 [13]. The set of nodes that is connected to node k (including node k itself) is called the neighborhood of node k and is denoted by $\mathcal{N}_k$. At time instants $\{i\}$, each node k is assumed to acquire a scalar measurement $d_{k,i}$, a regression vector $\mathbf{u}_k(i)$ and an intermediate estimate $\psi_{k,N_{k,i}}(i)$ for the unknown parameter vector based on the data $\{d_{l,i}, \mathbf{u}_l(i)\}$ from its neighborhood. The objective is for each node k in the network to estimate a length L_{opt} parameter vector $\mathbf{w}_{opt}$ of interest, where the tap-length L_{opt} is also unknown to us, by exploiting the data and estimates $\{d_{l,i}, \mathbf{u}_l(i), \psi_{l,N_{l,i}}(i)\}$ from its neighboring nodes.

Fig. 1. Representation of the diffusion strategy for distributed estimation^[13].

In this work, we propose a global as well as a distributed scheme to collectively estimate the unknown parameter vector with unknown tap-length over adaptive networks. The former is theoretically optimal by exploiting the data across the entire network, while the latter is practically effective by only utilizing the data from the neighbors at each node. In the proposed distributed strategy, based on the cost function of minimum distributed MSE, the tap-length estimate is adapted in real-time in the form of FT, by following the adapt-then-combine (ATC) policy. That is, during the adaptation step, each node collects measurements and regression vectors from its neighborhood and uses them to update its current filter tap-length, then the tap-weight vector adapts with the improved tap-length; during the combination step, each node collects tap-weight vectors from its neighbors and fuses deviations between its own weight estimate and its neighbors' to further renew its tap-length. Finally, update tap-weight vectors

with the new filter tap-lengths following the ATC diffusion normalized LMS (NLMS) algorithm in [2]. Thus it is evident that our work is much different from the previous diffusion version in [28]. Furthermore, both steady-state expressions for the mean square deviation (MSD) as well as tap-length, and mean convergence conditions for the tap-length update process are derived and formulated in this paper and match well with simulation results. Parameter choice guidelines are also provided to facilitate practical applications. As it will be confirmed in simulations, the learning curves of the proposed spatial diffusion variable tap-length algorithm converge to those of the global approach. Additionally, the proposed diffusion version outperforms the diffusion algorithm with fixed filter tap-length in [2] and the existing diffusion variable tap-length algorithm in [28] when the unknown tap-length is variable as well. Ultimately, theoretical findings and simulation results verify great robustness of the proposed schemes to parameter selections provided the parameter choice criteria are satisfied.

The remainder of the paper is organized as follows: In Section II we formulate the cost function for tap-length update. In Section III we motivate and derive the spatial diffusion FT variable tap-length NLMS (SDFTNLMS) algorithm and the global FT variable tap-length NLMS (GFTNLMS) algorithm. These algorithms are analyzed in Section IV and Section V respectively from the aspects of tap-weight and tap-length convergence processes. General guidelines for parameter selections are provided in Section VI. Section VII presents simulation results.

Notation: For convenience of description and analysis, all the variables used in this paper are assumed to be real. Boldface small and capital letters are used for vectors and matrices; normal font is utilized for scalars. All the vectors in this paper are deemed as column vectors by default. $|\cdot|$ and $\|\cdot\|$ refer to the absolute and Euclidean norm operation. $E[\cdot]$ denotes expectation and $\text{Tr}[\cdot]$ denotes trace of a matrix. The superscript $(\cdot)^T$ represents the transpose of a matrix or a vector. The notations $\text{col}\{\cdots\}$ and $\text{diag}\{\cdots\}$ denote a column vector and a diagonal matrix respectively. Operation $\lfloor \cdot \rfloor$ rounds the embraced value to the nearest integer, $\otimes$ denotes the Kronecker product, and $\circ$ is the Hadamard product. The operator $\text{vec}(\cdot)$ stacks the columns of a matrix argument on top of each other, and $\|\mathbf{x}\|_{\Sigma}^2 = \mathbf{x}^T \Sigma \mathbf{x}$ represents a weighted vector norm for any positive-definite matrix Σ . For a vector, the first subscript refers to the node, the second to the size, and (i) to the time index by default, e.g., $\mathbf{u}_{k,N}(i)$. For a variable, the first subscript denotes the node, and the second denotes the time index by default, e.g., $d_{k,i}$.

II. COST FUNCTION FORMULATION

Assume that there are P nodes in the network. At each time instant i and node k , the data are assumed to satisfy a linear regression model of the form [2]:

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

where $d_{k,i}$, $\mathbf{u}_{k,L_{opt}}(i)$ and $v_{k,i}$ are the desired signal, regression vector and measurement noise respectively. $\mathbf{w}_{opt,L_{opt}}$ denotes

the optimal tap-weight vector of size L_{opt} , which is same for all nodes. We further define the estimate error $e_{k,i}$ based on the weight estimate $\mathbf{w}$ and tap-length estimate N as

$$e_{k,i}(\mathbf{w}, N) \triangleq d_{k,i} - \mathbf{u}_{k,N}^T(i) \mathbf{w}_N. \quad (2)$$

As is shown in Fig.1, in practical networks, node k may have access only to the data $\{d_{l,i}, \mathbf{u}_l(i)\}$ from its neighbors $\{l \in \mathcal{N}_k\}$. In that case, the local MSE at node k is defined as the weighted sum of the MSEs at its neighbors [2]:

$$MSE_{k,i}^{loc}(\mathbf{w}_k, N_k) \triangleq \sum_{l \in \mathcal{N}_k} c_{l,k} E[e_{l,i}^2(\mathbf{w}_k, N_k)] \quad (3)$$

where $\mathbf{w}_k$ and N_k respectively are the weight estimate and tap-length estimate at node k obtained by exploiting data $\{d_{l,i}, \mathbf{u}_l(i)\}$ from its neighbors. The non-negative real coefficients $\{c_{l,k}\}$ give different weights to the data from the neighbors of node k . The adaptation matrix $\mathbf{C}$ with entries $\{c_{l,k}\}$ satisfies:

$$c_{l,k} = 0, \text{ if } l \notin \mathcal{N}_k, \mathbf{C}\mathbf{1} = \mathbf{1}, \text{ and } \mathbf{1}^T \mathbf{C} = \mathbf{1}^T \quad (4)$$

in which $\mathbf{1}$ denotes the P -vector with unit entries. Furthermore, the Metropolis rule is generally used to calculate coefficients $\{c_{l,k}\}$ [9]:

$$c_{l,k} = \begin{cases} \frac{1}{\max(\mathfrak{N}_k, \mathfrak{N}_l)}, & \text{if } k \neq l \text{ are linked} \\ 0, & \text{if } k \text{ and } l \text{ are not linked} \\ 1 - \sum_{s \in \mathcal{N}_k \setminus \{k\}} c_{s,k}, & \text{if } k = l \end{cases} \quad (5)$$

where $\mathfrak{N}_k, \mathfrak{N}_l$ respectively denote the cardinality of the nodes in networks $\mathcal{N}_k$ and $\mathcal{N}_l$; $\mathcal{N}_k \setminus \{k\}$ represents all nodes in the network $\mathcal{N}_k$ except for node k .

Before we proceed, several assumptions are made for convenience of analysis:

Assumption 1: The processes $d_{k,i}$ and $\mathbf{u}_k(i)$ are assumed to be jointly wide-sense stationary (WSS).

Assumption 2: Both the regressors $\{\mathbf{u}_{k,L_{opt}}(i)\}$ and measurement noises $\{v_{k,i}\}$ are assumed to be zero-mean white over time and independent over space with variances $\{\sigma_{u,k}^2\}$ and $\{\sigma_{v,k}^2\}$ respectively.

Assumption 3: The noise process $v_{k,i}$ and the regression data $\mathbf{u}_l(j)$ are assumed to be independent of each other for all k, l, i, j .

By utilizing Assumption 1, the optimal local estimate at node k for $\mathbf{w}_{opt, L_{opt}}$ is denoted as follows when the filter tap-length is N_k

$$\mathbf{w}_{k,N_k}^{loc} = \{\mathbf{w}_k | \partial MSE_{k,i}^{loc}(\mathbf{w}_k, N_k) / \partial \mathbf{w}_k = 0\}. \quad (6)$$

Note that the local estimates $\{\mathbf{w}_{k,N_k}^{loc}\}$ may differ from node to node, in that only measurements from each neighborhood are utilized.

We further define the distributed MSE at node k as the sum of local MSEs at all nodes in the network $\mathcal{N}_k$, based on Assumptions 2-3 and equations (3) and (6) [2]:

$$MSE_{k,i}^{dist}(\mathbf{w}_k, N_k) \triangleq \sum_{l \in \mathcal{N}_k} c_{l,k} E[e_{l,i}^2(\mathbf{w}_k, N_k)] + \sum_{l \in \mathcal{N}_k \setminus \{k\}} a_{l,k} \|\mathbf{w}_{k,N_k} - \mathbf{w}_{l,N_k}^{loc}\|^2 \quad (7)$$

in which the first item on the right-hand side (RHS) represents the local MSE at node k defined in (3), and the second denotes the sum of approximate local MSEs at the other nodes $\{l \in \mathcal{N}_k \setminus \{k\}\}$, with the weighted sum of measurement noise variances at the neighbors being omitted. Coefficients $\{a_{l,k}\}$ give different weights to the estimate deviations at the neighbors, and the diffusion matrix $\mathbf{A}$ with entries $\{a_{l,k}\}$ satisfies

$$a_{l,k} = 0, \text{ if } l \notin \mathcal{N}_k, \text{ and } \mathbf{1}^T \mathbf{A} = \mathbf{1}^T. \quad (8)$$

The relative-degree rule is generally utilized to calculate $\{a_{l,k}\}$, that is [11]

$$a_{l,k} = \begin{cases} \frac{\mathfrak{N}_l}{\sum_{s \in \mathcal{N}_k} \mathfrak{N}_s}, & \text{if } l \in \mathcal{N}_k \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

In the definition in (7), not only the data, but also the estimates from the neighbors are exploited, which means that at node k the data from all the neighboring nodes of its neighbors are utilized, thus accuracy is guaranteed if we seek estimates based on definition (7).

We continue to define the optimal tap-length estimate at node k based on the definition of the distributed MSE:

$$N_k^{opt} \triangleq \min \left\{ N_k \mid MSE_{k,\infty}^{dist}(\mathbf{w}_k, \mathbf{1} : N_k - \Delta | N_k) - MSE_{k,\infty}^{dist}(\mathbf{w}_k, N_k) \leq \xi_k, N_k > \Delta \right\} \quad (10)$$

where $MSE_{k,\infty}^{dist}$ is the steady-state distributed MSE at node k , namely as $i \rightarrow \infty$. Notation $\mathbf{1} : N_k - \Delta | N_k$ represents the first $N_k - \Delta$ entries of an N_k -vector, and we use $N_k - \Delta$ for convenience below. Positive integer Δ avoids suboptimal tap-length estimates and ξ_k is a small positive value that is predetermined according to system requirements [19]. In the following sections, we will provide choice guidelines for parameters Δ and ξ_k in adaptive networks, which facilitate practical use. Note that the tap-length $N_k, k = 1, 2, \dots, P$ is constrained to be not less than a lower floor value N_{min} , where $N_{min} > \Delta$, during its evolution. It is necessary since $N_k - \Delta$ is used as a tap-length [24].

Remark 1: It is verified that the minimum MSE (MMSE) is a monotonically non-increasing function of the tap-length. Moreover, a constant level of misalignment can be obtained under different scenarios by utilizing the NLMS algorithm [19]. Thus, we can conclude that the steady-state MSE of a single node as well as the steady-state distributed MSE defined in (7) are monotonically non-increasing relative to the filter length. By following the cost function in (10), we intend to find the filter length N_k^{opt} at node k that corresponds to the minimum steady-state distributed MSE. To be explicit, decrease of the steady-state distributed MSE at node k becomes negligible (depending on the parameter ξ_k) when a larger tap length is utilized.

In addition, the cost function defined in (10) is more suitable for the case where the unknown parameter vector is non-sparse or only suffers from a low degree of sparseness, otherwise the resulting tap-length estimates $\{N_k^{opt}\}$ can be affected by local minima of the distributed MSE.

Next, a spatial diffusion variable tap-length algorithm is proposed based on the cost function (10).

III. SPATIAL DIFFUSION FT VARIABLE TAP-LENGTH NLMS ALGORITHM

Based on the cost function (10), we proceed to update the tap-length in a style similar to that used in the leaky LMS algorithm [29, p. 314]. A leakage factor α_k is introduced in the adaptation rule at node k , $k = 1, 2, \dots, P$, thus, the concept of FT [19] is also adopted for application needs:

$$m_{k,i} = m_{k,i-1} - \alpha_k - \gamma_k \left\{ MS E_{k,i}^{dist}(\mathbf{w}_k(i-1), M_{k,i-1}) - MS E_{k,i}^{dist}(\mathbf{w}_k(i-1), M_{k,i-1} - \Delta) \right\} \quad (11)$$

and

$$M_{k,i} = \begin{cases} \lfloor m_{k,i} \rfloor, & \text{if } |m_{k,i} - M_{k,i-1}| \geq \delta_k \\ M_{k,i-1}, & \text{otherwise} \end{cases} \quad (12)$$

where $m_{k,i} > 0$ is the FT; positive integer $M_{k,i}$ is the tap-length estimate; we have $m_{k,0} = M_{k,0}$. The leakage factor $\alpha_k > 0$ aims to prevent the tap-length from wandering at a certain value; $\gamma_k > 0$ is the step-size for the tap-length adaptation and satisfies $\alpha_k \ll \gamma_k$; parameter $\delta_k > 0$ is a small integer [19], [24]. Replacing the estimates $\mathbf{w}_k$ and N_k in (10) by the current estimates $\mathbf{w}_k(i-1)$ and $M_{k,i-1}$ at node k , we arrive at recursion (11) based on (10), given the choice

$$\frac{\alpha_k}{\gamma_k} = \xi_k. \quad (13)$$

Furthermore, by exploiting definition (7) we obtain:

$$\begin{aligned} & MS E_{k,i}^{dist}(\mathbf{w}_k(i-1), M_{k,i-1}) - MS E_{k,i}^{dist}(\mathbf{w}_k(i-1), M_{k,i-1} - \Delta) \\ &= \sum_{l \in \mathcal{N}_k} c_{l,k} E \left[e_{l,i}^2(\mathbf{w}_k(i-1), M_{k,i-1}) - e_{l,i}^2(\mathbf{w}_k(i-1), M_{k,i-1} - \Delta) \right] \\ &+ \sum_{l \in \mathcal{N}_k \setminus \{k\}} a_{l,k} \left\{ \left\| \mathbf{w}_{k,M_{k,i-1}}(i-1) - \mathbf{w}_{l,M_{k,i-1}}^{loc} \right\|^2 \right. \\ &\quad \left. - \left\| \mathbf{w}_{k,M_{k,i-1}-\Delta}(i-1) - \mathbf{w}_{l,M_{k,i-1}-\Delta}^{loc} \right\|^2 \right\} \end{aligned} \quad (14)$$

where the last term on the RHS of (14) can be simplified as:

$$\sum_{l \in \mathcal{N}_k \setminus \{k\}} a_{l,k} \left\| \mathbf{w}_{k,\Delta|M_{k,i-1}}(i-1) - \mathbf{w}_{l,\Delta|M_{k,i-1}}^{loc} \right\|^2 \quad (15)$$

in which subscript $\Delta|M_{k,i-1}$ represents the last Δ entries of an $M_{k,i-1}$ -vector. Combining (14) and (15), the difference in (14) is the sum of two terms, namely,

$$\sum_{l \in \mathcal{N}_k} c_{l,k} E \left[e_{l,i}^2(\mathbf{w}_k(i-1), M_{k,i-1}) - e_{l,i}^2(\mathbf{w}_k(i-1), M_{k,i-1} - \Delta) \right],$$

$$\text{and } \sum_{l \in \mathcal{N}_k \setminus \{k\}} a_{l,k} \left\| \mathbf{w}_{k,\Delta|M_{k,i-1}}(i-1) - \mathbf{w}_{l,\Delta|M_{k,i-1}}^{loc} \right\|^2$$

in which the first term is named the adaptation term and the second is called the combination term. If we split the update equation in (11) into two steps, with only one term above being utilized in each step, then by ordering the manner in which these terms are combined to evaluate the difference in (14), we can motivate two combinations of adaptive diffusion strategies, namely the ATC policy and the CTA policy. Generally, the ATC strategy performs better than the other version in that more data $\{d_{k,i}, \mathbf{u}_k(i)\}$ in the network are fused [2], thus we

proceed in that manner. In the recursion from $m_{k,i-1}$ to $m_{k,i}$ in (11), the following two steps are performed in succession by generating an intermediate estimate $n_{k,i}$:

$$n_{k,i} = m_{k,i-1} - \alpha_k - \gamma_k \sum_{l \in \mathcal{N}_k} c_{l,k} \left[e_{l,i}^2(\mathbf{w}_k(i-1), M_{k,i-1}) - e_{l,i}^2(\mathbf{w}_k(i-1), M_{k,i-1} - \Delta) \right] \quad (16)$$

$$m_{k,i} = n_{k,i} - \gamma_k \sum_{l \in \mathcal{N}_k \setminus \{k\}} a_{l,k} \left\| \mathbf{w}_{k,\Delta|M_{k,i-1}}(i-1) - \mathbf{w}_{l,\Delta|M_{k,i-1}}^{loc} \right\|^2 \quad (17)$$

where the local MSE is replaced by its instantaneous approximation in (16). One problem that remains in (17) is that the estimates $\{\mathbf{w}_{l,\Delta|M_{k,i-1}}^{loc}\}$ are unknown a priori. Equation (6) is not a good solution, since node k is required to have access to the statistical information of the data at the neighbors. Note that the weight estimate is updated as well following the ATC diffusion NLMS algorithm at each node l as the measurements and regressors $\{d_{s,i}, \mathbf{u}_s(i)\}$ from its neighborhood arrive, by generating the intermediate estimate $\psi_{l,M_{l,i-1}}(i)$ [2]:

$$\begin{aligned} \psi_{l,M_{l,i-1}}(i) &= \mathbf{w}_{l,M_{l,i-1}}(i-1) + \mu_l \sum_{s \in \mathcal{N}_l} \frac{c_{s,l}}{\left\| \mathbf{u}_{s,M_{l,i-1}}(i) \right\|^2 + \varepsilon} \times \\ &\quad \mathbf{u}_{s,M_{l,i-1}}(i) \left[d_{s,i} - \mathbf{u}_{s,M_{l,i-1}}^T(i) \mathbf{w}_{l,M_{l,i-1}}(i-1) \right] \end{aligned} \quad (18)$$

where $\mu_l > 0$ is the step-size for the weight adaptation, $0 < \varepsilon \ll 1$ ensures that the denominator is not zero. Note that in (18) the NLMS algorithm is utilized to keep the system stable when the tap-length varies [19]. We replace thus estimates $\{\mathbf{w}_{l,\Delta|M_{k,i-1}}^{loc}\}$ in (17) with $\{\psi_{l,\Delta|M_{k,i-1}}(i)\}$ for nodes $\{l \in \mathcal{N}_k\}$, since they are available at node k . The quantity $\psi_{l,M_{l,i-1}}(i)$ represents the first $M_{l,i-1}$ entries in $\psi_{l,M_{l,i-1}}(i)$. Note that, if $M_{k,i-1} > M_{l,i-1}$, $\psi_{l,M_{k,i-1}}(i)$ is obtained by padding $\psi_{l,M_{l,i-1}}(i)$ with zeros. The same method is utilized below. We also replace $\mathbf{w}_{k,\Delta|M_{k,i-1}}(i-1)$ in (17) by the improved intermediate estimate $\psi_{k,\Delta|M_{k,i-1}}(i)$. As such, we can rewrite (17) as

$$m_{k,i} = n_{k,i} - \gamma_k \sum_{l \in \mathcal{N}_k \setminus \{k\}} a_{l,k} \left\| \psi_{k,\Delta|M_{k,i-1}}(i) - \psi_{l,\Delta|M_{k,i-1}}(i) \right\|^2. \quad (19)$$

After adaptation fusion (16), a better estimate for the tap-length of interest is obtained as follows:

$$N_{k,i} = \begin{cases} \lfloor n_{k,i} \rfloor, & \text{if } |n_{k,i} - M_{k,i-1}| \geq \delta_k \\ M_{k,i-1}, & \text{otherwise.} \end{cases} \quad (20)$$

We therefore replace tap-lengths $M_{l,i-1}$ and $M_{k,i-1}$ in recursions (18) and (19) with $N_{l,i}$ and $N_{k,i}$ respectively, namely, (19) is rewritten as

$$m_{k,i} = n_{k,i} - \gamma_k \sum_{l \in \mathcal{N}_k \setminus \{k\}} a_{l,k} \left\| \psi_{k,\Delta|N_{k,i}}(i) - \psi_{l,\Delta|N_{k,i}}(i) \right\|^2. \quad (21)$$

The filter tap-length at time i is given by

$$M_{k,i} = \begin{cases} \lfloor m_{k,i} \rfloor, & \text{if } |m_{k,i} - N_{k,i}| \geq \delta_k \\ N_{k,i}, & \text{otherwise.} \end{cases} \quad (22)$$

Finally, the ATC diffusion NLMS algorithm in [2] is applied with the new filter tap-length at each node to offer an estimate of the parameter vector of interest. Fig.2 shows schematically the proposed SDFTNLMS algorithm at node k , $k = 1, 2, \dots, P$.

Fig. 2. Schematic diagram of the SDFTNLMS algorithm at node k for any time index $i > 0$, which starts with measurements and regressors exchange in the neighborhood.

Similarly, we can obtain the GFTNLMS algorithm by defining the entire network as the neighborhood, namely, in the GFTNLMS algorithm, we choose the adaptation matrix $\mathbf{C} = 1/P\mathbf{1}\mathbf{1}^T$, and the diffusion matrix $\mathbf{A} = \mathbf{I}_P$ [2], where $\mathbf{I}_P$ is a $P \times P$ unit matrix. That is because in the global approach the measurements and regressors from all nodes in the network are conveyed to a central processor, where equal weights are given to the data from different nodes to estimate the unknown parameter vector. After that, the estimate information is transmitted back to each node from the central processor. We proceed further to take this algorithm as a basis to evaluate the SDFTNLMS algorithm, since the information from the entire network is fused directly in the GFTNLMS algorithm.

IV. STEADY-STATE MEAN-SQUARE PERFORMANCE ANALYSIS

In this section, we will provide steady-state mean-square performance analysis of the proposed SDFTNLMS algorithm, and formulate the steady-state MSD at each node k in the network. Compared to the performance analysis in the fixed tap-length case in [2], the main difference in a variable tap-length case is that the steady-state filter length differs from each other across the nodes in the network. In the subsequent analysis, we will primarily deal with the varying filter lengths.

Before we perform the steady-state analysis, we shall assume that:

Assumption 4: The term $\|\mathbf{u}_{l,M}(i)\|^2$ can be approximated by $M\sigma_{u,l}^2$, $l = 1, 2, \dots, P$, and $M\sigma_{u,l}^2 \gg \varepsilon$, with M being the tap-length estimate.

Assumption 5: At steady state, namely $i \rightarrow \infty$, the tap-length estimate at each node k will converge in the mean to a deterministic value $M_{k,\infty}$. We show in Section V.C the mean convergence conditions for the tap-length update process.

Assumption 6: The regressors $\{\mathbf{u}_{k,L_{opt}}(i)\}$ are Gaussian vectors. This assumption is made to further simplify analysis results.

For convenience of analysis, we use a vector $\mathbf{w}_{opt,N}$ to denote the unknown parameter vector by padding $\mathbf{w}_{opt,L_{opt}}$ with zeros, where N is an integer larger than both the optimal tap-length L_{opt} and the maximum value of the variable tap-length sequence $\{M_{k,i}\}$ for all k and i . In the same way, we get length

N tap-weight vectors at all nodes in the time sequence. Thus the weight adaption in the SDFTNLMS algorithm at node k becomes:

$$\psi_{k,N}(i) = \mathbf{w}_{k,N}(i-1) + \mu_k \mathbf{I}_{M_{k,i}}^{aug} \sum_{l \in \mathcal{N}_k} \frac{c_{l,k}}{\|\mathbf{u}_{l,M_{k,i}}(i)\|^2 + \varepsilon} \times \mathbf{u}_{l,N}(i) [d_{l,i} - \mathbf{u}_{l,N}^T(i) \mathbf{w}_{k,N}(i-1)] \quad (23)$$

and

$$\mathbf{w}_{k,N}(i) = \sum_{l \in \mathcal{N}_k} a_{l,k} \mathbf{I}_{M_{k,i}}^{aug} \psi_{l,N}(i) \quad (24)$$

where $\mathbf{I}_{M_{k,i}}^{aug}$ is an $N \times N$ matrix obtained by padding $\mathbf{I}_{M_{k,i}}$ with zeros.

By utilizing Assumption 4, we then define matrix $\mathbf{C}^{norm}$ with entries $\{c_{l,k}/(M_{k,i}\sigma_{u,l}^2)\}$ as the normalized adaptation matrix. Next deviation quantities at node k are defined as:

$$\tilde{\mathbf{w}}_{k,N}(i) = \mathbf{w}_{opt,N} - \mathbf{w}_{k,N}(i), \quad \tilde{\psi}_{k,N}(i) = \mathbf{w}_{opt,N} - \psi_{k,N}(i) \quad (25)$$

and the deviations are given by

$$\tilde{\mathbf{w}}(i) = \begin{bmatrix} \tilde{\mathbf{w}}_{1,N}(i) \\ \vdots \\ \tilde{\mathbf{w}}_{P,N}(i) \end{bmatrix}, \quad \tilde{\psi}(i) = \begin{bmatrix} \tilde{\psi}_{1,N}(i) \\ \vdots \\ \tilde{\psi}_{P,N}(i) \end{bmatrix}. \quad (26)$$

The steady-state MSD at node k is defined as [30]:

$$MSD_k = \lim_{i \rightarrow \infty} E \left\{ \|\tilde{\mathbf{w}}_{k,N}(i)\|^2 \right\}. \quad (27)$$

Remark 2: As is shown in our previous work in [24], the approach that all the vectors are padded with zeros to a sufficiently large size of N facilitates further analysis significantly, since all the vectors in question are of an identical and constant size. Rewriting the tap-weights update equations for a variable tap-length case as in equations (23) and (24), we can follow the performance analysis for the fixed tap-length case in [2] by only redefining some of the following matrices.

We first introduce the diagonal matrix

$$\mathcal{M}_i = \text{diag} \{ \mu_1 \mathbf{I}_{M_{1,i}}^{aug}, \dots, \mu_P \mathbf{I}_{M_{P,i}}^{aug} \} \quad (28)$$

and the extended weighting matrices

$$\mathbf{C} = \mathbf{C} \otimes \mathbf{I}_N, \quad \mathcal{A}_i = (\mathbf{A} \otimes \mathbf{I}_N) \text{diag} \{ \mathbf{I}_{M_{1,i}}^{aug}, \dots, \mathbf{I}_{M_{P,i}}^{aug} \} \quad (29)$$

$$\mathcal{D}_i = \text{diag} \left\{ \sum_{l=1}^P c_{l,1}^{norm} \mathbf{u}_{l,N}(i) \mathbf{u}_{l,N}^T(i), \dots, \sum_{l=1}^P c_{l,P}^{norm} \mathbf{u}_{l,N}(i) \mathbf{u}_{l,N}^T(i) \right\} \quad (30)$$

$$\mathcal{G}_i = (\mathbf{C}^{norm})^T \text{col} \{ \mathbf{u}_{1,N}(i) v_{1,i}, \dots, \mathbf{u}_{P,N}(i) v_{P,i} \}. \quad (31)$$

We also define the following based on Assumptions 2 and 5:

$$\mathcal{M} \triangleq E[\mathcal{M}_\infty] = \text{diag} \{ \mu_1 \mathbf{I}_{M_{1,\infty}}^{aug}, \dots, \mu_P \mathbf{I}_{M_{P,\infty}}^{aug} \} \quad (32)$$

$$\mathcal{A} \triangleq E[\mathcal{A}_\infty] = (\mathbf{A} \otimes \mathbf{I}_N) \text{diag} \{ \mathbf{I}_{M_{1,\infty}}^{aug}, \dots, \mathbf{I}_{M_{P,\infty}}^{aug} \} \quad (33)$$

$$\mathcal{D} \triangleq E[\mathcal{D}_\infty] = \text{diag} \left\{ \frac{1}{M_{1,\infty}}, \dots, \frac{1}{M_{P,\infty}} \right\} \otimes \mathbf{I}_N \quad (34)$$

where the property in (4) is utilized in equation (33) for simplification, and

$$\begin{aligned} \mathcal{G} &\triangleq \mathbb{E}[\mathcal{G}_\infty \mathcal{G}_\infty^T] \\ &= \mathcal{D} C^T \text{diag} \left\{ \frac{\sigma_{v,1}^2}{\sigma_{u,1}^2} \mathbf{I}_N, \dots, \frac{\sigma_{v,P}^2}{\sigma_{u,P}^2} \mathbf{I}_N \right\} C \mathcal{D}. \end{aligned} \quad (35)$$

Theorem 1: Assume Assumptions 1-6 hold and the following defined matrix $(\mathbf{I} - \mathbf{F})$ is invertible. The steady-state MSD at each node k is

$$MSD_k = \left[\text{vec}(\mathcal{A}^T \mathcal{M} \mathcal{G} \mathcal{M} \mathcal{A}) \right]^T (\mathbf{I} - \mathbf{F})^{-1} s_k \quad (36)$$

where

$$s_k \triangleq \text{vec}(\text{diag}\{\mathbf{e}_{k|P}\} \otimes \mathbf{I}_N) \quad (37)$$

with $\mathbf{e}_{k|P}$ being a P -vector with a unit entry at position k and zeros elsewhere. Moreover,

$$\begin{aligned} \mathbf{F} &= \left[(\mathbf{I} - \mathcal{D}\mathcal{M}) \otimes (\mathbf{I} - \mathcal{D}\mathcal{M}) + \right. \\ &\quad \left. \sum_{m=1}^P (\mathcal{D} \otimes \mathcal{D})(C_m^T \otimes C_m)(\mathcal{L}\mathbf{E} + \mathbf{I})(\mathcal{M} \otimes \mathcal{M}) \right] (\mathcal{A} \otimes \mathcal{A}) \end{aligned} \quad (38)$$

where $\mathbf{I}$ is a unit matrix of comparable size, and

$$C_m = \text{diag}\{c_{m,1}\mathbf{I}_N, \dots, c_{m,P}\mathbf{I}_N\} = \text{diag}\{\mathbf{e}_{m|P}^T \mathbf{C}\} \otimes \mathbf{I}_N \quad (39)$$

$$\mathcal{L} = \mathbf{I}_P \otimes \begin{bmatrix} \mathbf{I}_P \otimes \mathbf{e}_1 \\ \vdots \\ \mathbf{I}_P \otimes \mathbf{e}_N \end{bmatrix} \otimes (\text{vec}(\mathbf{I}_N))^T. \quad (40)$$

Let

$$\mathbf{e}_{kl} \triangleq (\mathbf{e}_{l|P}^T \otimes \mathbf{I}_N) \otimes (\mathbf{e}_{k|P}^T \otimes \mathbf{I}_N)$$

then we define

$$\mathbf{E} \triangleq \text{col}\{\mathbf{e}_{11}, \dots, \mathbf{e}_{P1}, \dots, \mathbf{e}_{1P}, \dots, \mathbf{e}_{PP}\}. \quad (41)$$

Proof: See Section IV¹ in [2] for reference. ■

Note that mean and mean square convergence conditions for the tap-weights update process can be obtained if one refers to [2] in Sections IV.B and IV.E.3 respectively, based on the results in this section. We proceed further to analyse the update process of the tap-length estimate.

V. PERFORMANCE ANALYSIS OF TAP-LENGTH UPDATE PROCESS

In this section, we will first formulate the steady-state expression for the tap-length estimate based on some convergence assumptions, then we will provide a sufficient mean convergence condition for the tap-length update process.

¹There is a typo in the second expression in Appendix II of [2]; the correct expression appears in expression (60) of [9]. Also, in expression (54) of [2], the notation $\text{bvec}(\cdot)$ should be used instead of $\text{vec}(\cdot)$ as done in (58) of [9] or alternatively by using the matrix $\mathbf{E}$ in equation (41) of this paper, with the relation that $\text{bvec}(\mathbf{J}) = \mathbf{E} \text{vec}(\mathbf{J})$ for any block matrix $\mathbf{J}$.

A. Problem Formulation

At each node k , following the similar technique in [24], we split the unknown parameter vector $\mathbf{w}_{opt,N}$ into three parts:

$$\begin{pmatrix} \mathbf{w}'_{opt,M_{k,i-1}-\Delta} \\ \mathbf{w}''_{opt,\Delta} \\ \mathbf{w}'''_{opt,N-M_{k,i-1}} \end{pmatrix}$$

where $\mathbf{w}'_{opt,M_{k,i-1}-\Delta}$ is the part modeled by $\mathbf{w}_{k,1:M_{k,i-1}-\Delta|M_{k,i-1}}(i-1)$, $\mathbf{w}''_{opt,\Delta}$ is the part modeled by $\mathbf{w}_{k,M_{k,i-1}-\Delta+1:M_{k,i-1}|M_{k,i-1}}(i-1)$, and $\mathbf{w}'''_{opt,N-M_{k,i-1}}$ is the under-modeled part.

For convenience of description, at node k the regression vectors $\{\mathbf{u}_{l,N}(i)\}$ from nodes $\{l \in \mathcal{N}_k\}$ and deviation quantity $\tilde{\mathbf{w}}_{k,N}(i-1)$ are split similarly to that of $\mathbf{w}_{opt,N}$. With the above notations, at node k the terms $\{e_{l,i}^2(\mathbf{w}_k(i-1), M_{k,i-1}) - e_{l,i}^2(\mathbf{w}_k(i-1), M_{k,i-1} - \Delta)\}_{l \in \mathcal{N}_k}$ in (16) can be written as:

$$\begin{aligned} &e_{l,i}^2(\mathbf{w}_k(i-1), M_{k,i-1}) - e_{l,i}^2(\mathbf{w}_k(i-1), M_{k,i-1} - \Delta) \\ &= \underbrace{2v_{l,i} \mathbf{u}_{l,\Delta}^{\prime T}(i) \tilde{\mathbf{w}}_{k,\Delta}''(i-1)}_a - \underbrace{2v_{l,i} \mathbf{u}_{l,\Delta}^{\prime T}(i) \mathbf{w}_{opt,\Delta}''}_{b} \\ &\quad + \underbrace{2\mathbf{u}_{l,M_{k,i-1}-\Delta}^{\prime T}(i) \tilde{\mathbf{w}}_{k,M_{k,i-1}-\Delta}'(i-1) \mathbf{u}_{l,\Delta}^{\prime T}(i) \tilde{\mathbf{w}}_{k,\Delta}''(i-1)}_c \\ &\quad - \underbrace{2\mathbf{u}_{l,M_{k,i-1}-\Delta}^{\prime T}(i) \tilde{\mathbf{w}}_{k,M_{k,i-1}-\Delta}'(i-1) \mathbf{u}_{l,\Delta}^{\prime T}(i) \mathbf{w}_{opt,\Delta}''}_{d} \\ &\quad + \underbrace{\left[\mathbf{u}_{l,\Delta}^{\prime T}(i) \tilde{\mathbf{w}}_{k,\Delta}''(i-1) \right]^2}_e - \underbrace{\left[\mathbf{u}_{l,\Delta}^{\prime T}(i) \mathbf{w}_{opt,\Delta}'' \right]^2}_f \\ &\quad + \underbrace{2\mathbf{u}_{l,N-M_{k,i-1}}^{\prime T}(i) \mathbf{w}_{opt,N-M_{k,i-1}}''' \mathbf{u}_{l,\Delta}^{\prime T}(i) \tilde{\mathbf{w}}_{k,\Delta}''(i-1)}_g \\ &\quad - \underbrace{2\mathbf{u}_{l,N-M_{k,i-1}}^{\prime T}(i) \mathbf{w}_{opt,N-M_{k,i-1}}''' \mathbf{u}_{l,\Delta}^{\prime T}(i) \mathbf{w}_{opt,\Delta}''}_{h}. \end{aligned} \quad (42)$$

Next the terms $\left\{ \left\| \psi_{k,\Delta|N_{k,i}}(i) - \psi_{l,\Delta|N_{k,i}}(i) \right\|^2 \right\}_{l \in \mathcal{N}_k}$ in (21) will be analyzed for each node k , $k = 1, 2, \dots, P$. Using definitions of Euclidean norm and deviation quantity in (25), we get

$$\begin{aligned} &\underbrace{\left\| \psi_{k,\Delta|N_{k,i}}(i) - \psi_{l,\Delta|N_{k,i}}(i) \right\|^2}_n \\ &= \left\| \tilde{\psi}_{l,\Delta|N_{k,i}}(i) \right\|^2 + \left\| \tilde{\psi}_{k,\Delta|N_{k,i}}(i) \right\|^2 \\ &\quad - \tilde{\psi}_{l,\Delta|N_{k,i}}^T(i) \tilde{\psi}_{k,\Delta|N_{k,i}}(i) - \tilde{\psi}_{k,\Delta|N_{k,i}}^T(i) \tilde{\psi}_{l,\Delta|N_{k,i}}(i) \\ &= \left\| \tilde{\psi}(i) \right\|_{\delta_{ll,i} + \delta_{kk,i} - \delta_{lk,i} - \delta_{kl,i}}^2. \end{aligned} \quad (43)$$

in which

$$\delta_{lk,i} \triangleq \text{vec}[(\mathbf{e}_{l|P} \otimes \Delta_i)(\mathbf{e}_{k|P}^T \otimes \Delta_i)], \quad (44)$$

$$\Delta_i \triangleq \text{diag}\{\mathbf{0}_{N_{k,i}-\Delta}, \mathbf{I}_\Delta, \mathbf{0}_{N-N_{k,i}}\}. \quad (45)$$

Substituting (16), (42) and (43) into (21), we have

$$\begin{aligned} m_{k,i} &= m_{k,i-1} - \alpha_k - \\ &\quad \gamma_k \sum_{l \in \mathcal{N}_k} [c_{l,k}(a - b + c - d + e - f + g - h) + a_{l,k}n]. \end{aligned} \quad (46)$$

Next, a steady-state performance analysis of tap-length will be given based on this update equation.

Remark 3: Different from the steady-state tap-length analysis for single-node FTLNLS algorithm in our previous work [24], the diffusion step of tap-length recursion results in a diffusion term n in the proposed SFTNLMS algorithm, which does not exist in the single-node version.

B. Steady-State Performance Analysis

To simplify the analysis, we make several further assumptions:

Assumption 7: At steady-state, $E[n_{k,i}]$ and $E[m_{k,i}]$ tend to be constants for each node k , as $i \rightarrow \infty$.²

Assumption 8: At steady state, we assume $M_{k,\infty} > L_{opt}$ for all k . See Theorem 3 in Section V.C for conditions that ensure the justification.

Assumption 9: The tail elements of the unknown parameter vector $\mathbf{w}_{opt,L_{opt}}$ can be deemed to be drawn from a random white sequence with zero mean and variance σ_w^2 . This assumption is justified and discussed in [24].

Assumption 10: At steady state, vector $\tilde{\mathbf{w}}_{k,N}(i-1)$ is independent of $\mathbf{u}_{l,N}(i)$, for all k, l . The justification of this assumption is that the update of $\tilde{\mathbf{w}}_{k,N}(i-1)$ only depends on the past regression vectors $\{\mathbf{u}_l(j), l = 1, 2, \dots, P, j \leq i-1\}$ as is shown in (23) and (24).

Theorem 2: Assume that Assumptions 1-10 hold. The tap-length estimate at each node k converges in the mean to

$$M_{k,\infty} = L_{opt} + \Delta - E \left[\left\| \tilde{\mathbf{w}}''_{k,\Delta}(\infty) \right\|^2 \right] / \sigma_w^2 - \left(\alpha_k / \gamma_k + E \left[\left\| \tilde{\psi}(\infty) \right\|_{\mathbf{b}_{l,k}}^2 \right] \right) / \left(\overline{\sigma_{u,k}^2} \sigma_w^2 \right) \quad (47)$$

where

$$E \left[\left\| \tilde{\mathbf{w}}''_{k,\Delta}(\infty) \right\|^2 \right] = \left[\text{vec} \left(\mathcal{A}^T \mathcal{M} \mathcal{G} \mathcal{M} \mathcal{A} \right) \right]^T (I - F)^{-1} \delta_{kk,\infty}. \quad (48)$$

with $\delta_{kk,\infty}$ being defined in (44) as $i \rightarrow \infty$, and

$$\overline{\sigma_{u,k}^2} \triangleq \sum_{l \in N_k} c_{l,k} \sigma_{u,l}^2. \quad (49)$$

Let

$$\mathbf{b}_{l,k} \triangleq \sum_{l \in N_k} a_{l,k} (\delta_{ll,\infty} + \delta_{kk,\infty} - \delta_{lk,\infty} - \delta_{kl,\infty}) \quad (50)$$

and

$$\mathbf{H} = \mathbf{F}|_{\mathcal{A}=\mathbf{I}} \quad (51)$$

which means that $\mathbf{H}$ equals to $\mathbf{F}$ when we replace $\mathcal{A}$ with a unit matrix of comparable size in (38), then we define

$$E \left[\left\| \tilde{\psi}(\infty) \right\|_{\mathbf{b}_{l,k}}^2 \right] = [\text{vec}(\mathcal{M} \mathcal{G} \mathcal{M})]^T \mathbf{b}_{l,k} + [\text{vec}(\mathcal{A}^T \mathcal{M} \mathcal{G} \mathcal{M} \mathcal{A})]^T (I - F)^{-1} \mathbf{H} \mathbf{b}_{l,k}. \quad (52)$$

Proof: Taking expectation of both sides of (46), we have the following under Assumption 7

$$\sum_{l \in N_k} \{c_{l,k} E[a - b + c - d + e - f + g - h] + a_{l,k} E[n]\} = -\frac{\alpha_k}{\gamma_k}. \quad (53)$$

²This assumption may not always hold in practice, however, it is required to derive a closed-form expression for the steady-state tap-length estimate.

Based on our previous study on the single-node FTLNLS algorithm in [24], the left hand side of equation (53) can be simplified as

$$\sum_{l \in N_k} \{c_{l,k} E[a - b + c - d + e - f + g - h] + a_{l,k} E[n]\} = \overline{\sigma_{u,k}^2} \left(E \left[\left\| \tilde{\mathbf{w}}''_{k,\Delta}(\infty) \right\|^2 \right] - E \left[\left\| \mathbf{w}_{opt,\Delta}' \right\|^2 \right] \right) + E \left[\left\| \tilde{\psi}(\infty) \right\|_{\mathbf{b}_{l,k}}^2 \right] \quad (54)$$

where

$$E \left[\left\| \mathbf{w}_{opt,\Delta}' \right\|^2 \right] = (L_{opt} + \Delta - M_{k,\infty}) \sigma_w^2 \quad (55)$$

and

$$L_{opt} < M_{k,\infty} \leq L_{opt} + \Delta. \quad (56)$$

In a manner similar to (43), the term $E \left[\left\| \tilde{\mathbf{w}}''_{k,\Delta}(\infty) \right\|^2 \right]$ becomes:

$$E \left[\left\| \tilde{\mathbf{w}}''_{k,\Delta}(\infty) \right\|^2 \right] = E \left[\left\| \tilde{\mathbf{w}}(\infty) \right\|_{\delta_{kk,\infty}}^2 \right]. \quad (57)$$

Recall the following steady-state relation in [2]

$$E \left[\left\| \tilde{\mathbf{w}}(\infty) \right\|_{(I-F)\sigma}^2 \right] = [\text{vec}(\mathcal{A}^T \mathcal{M} \mathcal{G} \mathcal{M} \mathcal{A})]^T \sigma \quad (58)$$

where $\sigma = \text{vec}(\Sigma)$ for any positive-definite matrix Σ and we are free to choose. Then the term $E \left[\left\| \tilde{\mathbf{w}}(\infty) \right\|_{\delta_{kk,\infty}}^2 \right]$ can be obtained by choosing $\sigma = (I - F)^{-1} \delta_{kk,\infty}$ in (58), and equation (48) follows.

Based on (23) as $i \rightarrow \infty$, the weighted norm of $\tilde{\psi}(\infty)$ for any positive-definite matrix Φ is obtained under Assumptions 2-3, 10:

$$E \left[\left\| \tilde{\psi}(\infty) \right\|_{\Phi}^2 \right] = E \left[\left\| \tilde{\mathbf{w}}(\infty) \right\|_{\Phi}^2 \right] + \text{Tr}[\Phi \mathcal{M} \mathcal{G} \mathcal{M}] \quad (59)$$

where

$$\Phi' \triangleq E \left[(\mathbf{I} - \mathcal{D}_{\infty}^T \mathcal{M}) \Phi (\mathbf{I} - \mathcal{M} \mathcal{D}_{\infty}) \right]. \quad (60)$$

Let

$$\varphi \triangleq \text{vec}(\Phi), \text{ and } \varphi' \triangleq \text{vec}(\Phi') \triangleq \mathbf{H} \varphi$$

and comparing the variance relation (39) in [2] with relation (59), we obtain equation (51). Using the property

$$\text{Tr}[\Sigma \mathbf{X}] = \text{vec}(\mathbf{X}^T)^T \text{vec}(\Sigma)$$

then we rewrite equation (59) as

$$E \left[\left\| \tilde{\psi}(\infty) \right\|_{\varphi}^2 \right] = E \left[\left\| \tilde{\mathbf{w}}(\infty) \right\|_{\mathbf{H}\varphi}^2 \right] + [\text{vec}(\mathcal{M} \mathcal{G} \mathcal{M})]^T \varphi. \quad (61)$$

We proceed to replace φ in (61) with the defined $\mathbf{b}_{l,k}$ in (50), and combine equations (58) and (61) by choosing σ as $(\mathbf{I} - F)^{-1} \mathbf{H} \mathbf{b}_{l,k}$ in (58), then equation (52) follows. The steady-state expression for the tap-length estimate at node k in (47) finally comes out from (53), (54) and (55). ■

Remark 4: Recall the result of the single-node FTLNLS algorithm, where the steady-state expression of the tap-length estimate is as follows [24]:

$$M_{\infty} = L_{opt} + \Delta - \frac{\Delta \mu \sigma_v^2}{(2 - \mu) L_{opt} \sigma_u^2 \sigma_w^2} - \frac{\alpha}{\gamma \sigma_u^2 \sigma_w^2} \quad (62)$$

in which the first two and the last terms are much similar to those in equation (47) respectively, and the third one is caused by the deviation from the last Δ weights of the adaptive filter,

as is modelled by the term $E[\|\tilde{\mathbf{w}}'_{k,\Delta}(\infty)\|^2]/\sigma_w^2$ in equation (47). Therefore, the only difference between these two steady-state expressions is in the last term $E[\|\tilde{\psi}(\infty)\|_{\mathbf{b}_{l,k}}^2]/(\sigma_{u,k}^2\sigma_w^2)$ in equation (47), which is caused by the diffusion step of the proposed algorithm. Intuitively the numerator $E[\|\tilde{\psi}(\infty)\|_{\mathbf{b}_{l,k}}^2]$ denotes the steady-state deviation between estimates in the neighborhood on the last Δ tap-weights, thus the value of $E[\|\tilde{\psi}(\infty)\|_{\mathbf{b}_{l,k}}^2]/(\sigma_{u,k}^2\sigma_w^2)$ is relatively small compared with $L_{opt} + \Delta$.

Consequently, similar to the single-node case in [24], the mean value of the steady-state tap-length estimate at each node in the network will be close to the value of $L_{opt} + \Delta$. This indicates that we can set $\{M_{k,\infty} = L_{opt} + \Delta\}$ and $N = L_{opt} + \Delta$ on the RHS of (36) to evaluate the steady-state MSDs across the network.

Moreover, expression (47) also implies the robustness of the proposed algorithms to the parameter selections of $\{\alpha_k\}$ and $\{\gamma_k\}$, resulting from their slight influence on the steady-state performance of the proposed algorithms, provided that the convergence conditions and parameter choice guidelines in the subsequent sections are satisfied. That will be also shown in the following simulations.

C. Mean Convergence Analysis

In view of Theorem 2, we know that in the proposed algorithms the mean value of the steady-state tap-length at each node k is close to $L_{opt} + \Delta$. In this section, we will provide sufficient conditions for the parameter selections to ensure theoretical mean convergence of the tap-length update process, under which the tap-length estimate at each node k in the proposed algorithms converges in the mean to a value close to $L_{opt} + \Delta$.

For the convenience of analysis, we assume that

Assumption 11: The value of Δ is larger than the width of the suboptimum tap-length, which is defined as the number of successive suboptimum tap-lengths [19]. See Section VI for specific parameter choice guidelines.

The following lemma guarantees the tap-length estimate converges in the mean to a value close to the optimal tap-length estimate defined in cost function (10).

Lemma 1: Assume that Assumption 11 and the relation in (13) hold. Then the tap-length at each node k converges in the mean to within a range of $(N_k^{opt} - \delta_k, N_k^{opt} + \delta_k)$, where N_k^{opt} is defined in cost function (10) [19].

In view of Lemma 1, we proceed to seek conditions on parameter selections in the proposed algorithms under which the optimal tap-length estimate at each node k defined in cost function (10) is exactly $L_{opt} + \Delta$.

Let

$$dev_{I,k} \triangleq MSE_{k,\infty}^{dist}(\mathbf{w}_k(\infty), L_{opt}-1) - MSE_{k,\infty}^{dist}(\mathbf{w}_k(\infty), L_{opt}+\Delta-1) \quad (63)$$

and

$$dev_{II,k} \triangleq MSE_{k,\infty}^{dist}(\mathbf{w}_k(\infty), L_{opt}) - MSE_{k,\infty}^{dist}(\mathbf{w}_k(\infty), L_{opt} + \Delta) \quad (64)$$

where the subscript k denotes a specific node, then we have

$$dev_{I,k} > 0 \quad (65)$$

$$dev_{II,k} < 0 \quad (66)$$

where inequality (65) holds in that the unknown parameter vector $\mathbf{w}_{opt,L_{opt}}$ is under-modeled by $\mathbf{w}_{k,L_{opt}-1}(\infty)$, and is fully modeled by $\mathbf{w}_{k,L_{opt}+\Delta-1}(\infty)$ for any $\Delta \geq 1$, thus the difference between the steady-state MSEs on the RHS will be positive. The inequality (66) holds due to the existence of adaptation noise.

Lemma 2: Assume that Assumption 11 and the relation in (13) hold. Then $N_k^{opt} = L_{opt} + \Delta$ is the optimal tap-length estimate at node k defined in cost function (10) if, and only if

$$\gamma_k > \frac{\alpha_k}{dev_{I,k}}. \quad (67)$$

Proof: See Appendix A. ■

The following theorem finally guarantees that the tap-length estimate at each node k in the proposed algorithms converges in the mean to a value close to $L_{opt} + \Delta$.

Theorem 3: Assume that Assumption 11 and the relation in (13) hold. Then the tap-length at each node k converges in the mean to within a range of $(L_{opt} + \Delta - \delta_k, L_{opt} + \Delta + \delta_k)$ if

$$\gamma_k > \frac{\alpha_k}{dev_{I,k}}. \quad (68)$$

Proof: In view of Lemma 1 and Lemma 2, Theorem 3 follows. ■

Theorem 3 ensures the mean convergence assumptions of the tap-length update process in Assumptions 5 and 8 hold, by further setting $\{\Delta \geq \delta_k\}$. Assume that all other assumptions made in Theorem 2 hold, then Theorem 3 actually guarantees the tap-length estimate at each node k converges in the mean to the value formulated by (47) in Theorem 2.

In the following section, we will provide some practical choice guidelines for the parameter settings in the proposed algorithms, which satisfy the mean convergence conditions in Theorem 3.

VI. PARAMETER CHOICE GUIDELINES

There are four extra parameters at each node k , $k = 1, 2, \dots, P$ introduced by the tap-length estimation as compared with the diffusion NLMS algorithm with fixed tap-length, namely, α_k , γ_k (as well as ξ_k due to the relationship of $\xi_k = \alpha_k/\gamma_k$), Δ and δ_k . In this section, we will provide some general guidelines for parameter choices of γ_k and Δ , which are different from those designed for single-node tap-length adaptation in [24], while for selections of the other parameters one can refer to [24]. To be specific, the parameters can be determined as follows:

1. The parameter ξ_k is a very small positive number, and its value should be chosen based on the measurement noise level. Practically, ξ_k can be set to the steady-state EMSE at node k , which is defined as [30]:

$$EMSE_{k,\infty} = E\left\{\left[\mathbf{u}_{k,N}^T(\infty)\tilde{\mathbf{w}}_{k,N}(\infty)\right]^2\right\}. \quad (69)$$

To this end, we can weight $E\|\tilde{\mathbf{w}}(\infty)\|^2$ in (58) with a block matrix that has $\sigma_{u,k}^2 \mathbf{I}_N$ at block (k, k) and zeros elsewhere [2], that is to select $\sigma = (\mathbf{I} - \mathbf{F})^{-1} \sigma_{u,k}^2 s_k$ in (58), with s_k being defined in (37). Specifically, ξ_k is chosen as follows:

$$\xi_k = \left[\text{vec}(\mathcal{A}^T \mathcal{M} \mathcal{G} \mathcal{M} \mathcal{A}) \right]^T \times \left[\mathbf{I} - (\mathbf{I} - \mathcal{D} \mathcal{M} \otimes \mathbf{I} - \mathbf{I} \otimes \mathcal{D} \mathcal{M})(\mathcal{A} \otimes \mathcal{A}) \right]^{-1} \sigma_{u,k}^2 s_k \quad (70)$$

with the other terms in $\mathbf{F}$ defined in (38) being neglected, since they will depend on $\{\mu_k^2 / M_{k,\infty}^2\}$. In practice, in order to estimate ξ_k , we only need estimates of the input variance as well as the measurement noise variance, which are practically available or can be estimated [31] and obtained at each node through network transmission. Note that the values of $\{M_{k,\infty}\}$ in (70) will have slight influence on the value of ξ_k , since we know that the steady-state EMSE of the NLMS algorithm will mainly depend on the measurement noise variance and the value of the step-size μ , if the steady-state filter length is larger than L_{opt} [30]. Namely, in practice we can choose a sufficiently large value for $\{M_{k,\infty}\}$ to calculate $\{\xi_k\}$, from which $\{\gamma_k = \alpha_k / \xi_k\}$ can be determined. We show in Appendix B that such selected values of $\{\gamma_k\}$ satisfy the mean convergence condition in Theorem 3.

2. The parameter Δ must be larger than the width of the suboptimum tap-length, so that the length adaptation can escape from local minima [19]. Whereas, Δ should be much smaller than the estimate of L_{opt} , so that the steady-state tap-length formulated in (47) will not be significantly biased from L_{opt} . Generally, an integer around $0.1L_{opt}$ will be a good choice [24]. However, since L_{opt} is unknown a priori and may be varying, in practice, we can set Δ to an integer that is close to 10 percent of the current tap-length estimate, with a minimum value chosen according to the estimate for the order of the magnitude of L_{opt} . Better performance can be achieved by adapting the parameter Δ during the length adaptation [25], in which the value of Δ is large initially, contributing to a fast convergence rate, and then decreases gradually to ensure a small bias from the optimal tap-length at steady state.

Remark 5: The parameter choice guidelines for $\{\gamma_k\}$ and Δ are different from those designed for the single-node FTLNLS algorithm in [24], and more suitable for network adaptive filter application. In this work, we calculate step-sizes $\{\gamma_k\}$ for tap-length adaptation from the relation $\{\gamma_k = \alpha_k / \xi_k\}$ with $\{\xi_k\}$ being set to steady-state $\{EMSE_k\}$, which are obtained from our theoretical results. On the contrary, in our previous version for the single-node strategy, we provide lower and upper bounds for setting the parameter γ , by limiting the expectation and variance of the steady-state filter length respectively. As for the parameter Δ , we set it to an integer that is close to 10 percent of the current tap-length estimate, with a minimum value chosen according to the estimate for the order of the magnitude of the optimum tap-length, which is more practical than our previous guideline that Δ is set to be an integer around 10 percent of the optimal tap-length L_{opt} , which is unknown a priori.

Note that only the steady-state mean value of the parameter Δ is involved in the performance analyses of the tap-weights and tap-length update processes in Sections IV and

V. Moreover, at steady state, the mean value of the parameter Δ will be constant as the tap-length estimate converges in the mean to a constant. Consequently, the time-varying value of the parameter Δ will not affect the previous theoretical results. Furthermore, the steady-state expression of the tap-length estimate $M_{k,\infty} \approx L_{opt} + \Delta$ indicates that the tap-length will be slightly overestimated compared to the value of L_{opt} , provided the above parameter choice guidelines are satisfied.

To sum up, the proposed SDFTNLMS algorithm is shown in Table I at the top of the next page.

VII. SIMULATION RESULTS

In order to illustrate the performance of the proposed algorithms, we perform four simulations in this section. The steady-state performance is tested in the first simulation, followed by the test for mean convergence performance of the tap-length update process. The last two experiments are conducted to test tracking performance, where the optimum tap-length varies unpredictably under a low-noise condition and a high-noise condition respectively. The robustness to parameter settings is also tested in the last three simulations, according to the provided parameter choice guidelines.

Fig. 3. Network topology (top), regressor variances (bottom, left) and noise variances (bottom, right) for $P = 7$ nodes.

A. Steady-State Performance

The setup of the simulation is as follows. The regressors and measurement noises across the network are uncorrelated zero-mean Gaussian, independent in time and space with variances denoted by $\{\sigma_{u,k}^2\}$ and $\{\sigma_{v,k}^2\}$ respectively. Fig.3 depicts the network topology with $P = 7$ nodes, together with the network statistical profile showing how the signal power varies across the nodes. This network topology was firstly discussed in [9]. The unknown parameters, of size $L_{opt} = 45$, are drawn from a white Gaussian sequence with zero mean and variance of 0.01. Matrices $\mathbf{C}$ and $\mathbf{A}$ are calculated by (5) and (9) respectively in the SDFTNLMS algorithm, while for the GFTNLMS

TABLE I
THE PROPOSED SDFTNLMS ALGORITHM.

Start with $\{m_{k,0} = M_{k,0}, \mathbf{w}_{k,M_{k,0}}(0) = \mathbf{0}\}$, $\Delta = \Delta_{min}$ for all k . Given non-negative real coefficients $\{c_{l,k}\}$ and $\{a_{l,k}\}$ calculated from (5) and (9) respectively, step-sizes $\{\mu_k\}$, parameters $\{\alpha_k, \delta_k\}$, Δ_{min} , step-sizes $\{\gamma_k = \alpha_k/\xi_k\}$ with $\{\xi_k\}$ being calculated from equation (70), for each time $i \geq 1$ and for each node k , repeat:

(i) Error calculations from different segments

$$e_{k,i}(\mathbf{w}_k(i-1), M_{k,i-1}) = d_{k,i} - \mathbf{u}_{k,M_{k,i-1}}^T(i) \mathbf{w}_{k,M_{k,i-1}}(i-1)$$

$$e_{k,i}(\mathbf{w}_k(i-1), M_{k,i-1} - \Delta) = d_{k,i} - \mathbf{u}_{k,M_{k,i-1}-\Delta}^T(i) \mathbf{w}_{k,M_{k,i-1}-\Delta}(i-1)$$

(ii) Measurements and regressors exchange among the neighborhood, tap-length adapts following equations (16) and (20)

(iii) Tap-weights adapt using updated tap-length

$$\psi_{k,N_{k,i}}(i) = \mathbf{w}_{k,N_{k,i}}(i-1) + \mu_k \sum_{l \in N_k} c_{l,k} \left(\left\| \mathbf{u}_{l,N_{k,i}}(i) \right\|^2 + \varepsilon \right) \mathbf{u}_{l,N_{k,i}}(i) \left[d_{l,i} - \mathbf{u}_{l,N_{k,i}}^T(i) \mathbf{w}_{k,N_{k,i}}(i-1) \right]$$

(iv) Estimates exchange among the neighborhood, tap-length combines following equations (21) and (22)

(v) Tap-weights update following diffusion NLMS algorithm based on new tap-length

$$\psi_{k,M_{k,i}}(i) = \mathbf{w}_{k,M_{k,i}}(i-1) + \mu_k \sum_{l \in N_k} c_{l,k} \left(\left\| \mathbf{u}_{l,M_{k,i}}(i) \right\|^2 + \varepsilon \right) \mathbf{u}_{l,M_{k,i}}(i) \left[d_{l,i} - \mathbf{u}_{l,M_{k,i}}^T(i) \mathbf{w}_{k,M_{k,i}}(i-1) \right]$$

$$\mathbf{w}_{k,M_{k,i}}(i) = \sum_{l \in N_k} a_{l,k} \psi_{l,M_{k,i}}(i)$$

(vi) Parameters update based on choice guidelines

$$\Delta = \max\{0.1M_{k,i}, \Delta_{min}\}$$

Note that tap-weight vectors are padded with zeros if necessary.

algorithm, $\mathbf{C} = 1/711\mathbf{I}$, $\mathbf{A} = \mathbf{I}_7$. Moreover, in order to verify our theoretical analysis results in equations (36) and (47), in both the GFTNLMS and SDFTNLMS algorithms, Δ is fixed at 6 during the convergence process, δ is 2, $M_{min} = m_{k,0} = 10$, $\mu = 0.1$, $\varepsilon = 10^{-4}$, and we calculate step-sizes $\{\gamma_k\}$ through $\{\alpha_k/\xi_k\}$, where we set parameters $\{\alpha_k\}$ to 0.008, and select $\{\xi_k\}$ according to equation (70) with a sufficiently large tap-length of 100. The simulation values are obtained over 6000 time samples (which ensures that the algorithms arrive at steady state) in each independent experiment.

Table II and Table III show the simulated steady-state tap-lengths and MSDs respectively for both the SDFTNLMS and GFTNLMS algorithms, together with the theoretical results from expressions (47) and (36) where we set $\{M_{k,\infty} = L_{opt} + \Delta\}$ and $N = L_{opt} + \Delta$. In both Table II and Table III, the steady-state values of the diffusion algorithm are presented on the left, while those of the global version are on the right. Note that the simulation results are averaged over 100 Monte Carlo runs, thus an integral tap-length is not guaranteed; and that in the GFTNLMS algorithm, we use each node k , $k = 1, 2, \dots, 7$ to serve as the fusion center respectively, thus different signal variances, resulting in different parameter settings $\{\xi_k, \gamma_k\}$, lead to different steady-state tap-lengths in Table II; slightly different MSDs are obtained in Table III due to different runs.

TABLE II
THE STEADY-STATE TAP-LENGTHS FOR SDFTNLMS AND GFTNLMS, SIMULATION VERSUS THEORY.

diff. node	theo. diff. tap-length (47)	simu. diff. tap-length	theo. glob. tap-length (47)	simu. glob. tap-length
1	50.97	52.18	50.98	52.21
2	50.95	51.25	50.94	51.21
3	50.96	51.63	50.96	51.39
4	50.92	51.18	50.92	51.25
5	50.94	51.26	50.93	51.18
6	50.93	51.29	50.95	51.39
7	50.95	51.29	50.95	51.40

Several conclusions can be drawn from Table II and Table III: firstly, theoretical results match well with simulation values in terms of MSD and tap-length, with the steady-state tap-lengths being all around $L_{opt} + \Delta$; subsequently, it is proven

TABLE III
THE STEADY-STATE MSDs FOR SDFTNLMS AND GFTNLMS, SIMULATION VERSUS THEORY.

diff. node	theo. diff. MSD (dB) (36)	simu. diff. MSD (dB)	theo. glob. MSD (dB) (36)	simu. glob. MSD (dB)
1	-33.03	-33.02	-33.34	-33.32
2	-33.04	-33.03	-33.34	-33.29
3	-33.03	-33.02	-33.34	-33.30
4	-33.04	-33.03	-33.34	-33.29
5	-33.04	-33.03	-33.34	-33.28
6	-33.02	-33.01	-33.34	-33.28
7	-33.03	-33.02	-33.34	-33.30

theoretically and practically that the SDFTNLMS algorithm drives the tap-lengths and the MSDs to converge to a similar value respectively across the nodes; ultimately, the tap-lengths and MSDs from the SDFTNLMS algorithm converge to those of the GFTNLMS algorithm at steady state, which indicates a similar steady-state performance of both algorithms, but the distributed scheme requires lower computations.

B. Mean Convergence Performance

In this section, we will verify the theoretical results presented in Theorem 3, by using the same test environment as is shown in the steady-state performance simulation. Let $\{M_{k,\infty} = L_{opt} + \Delta - 1\}$ and $N = L_{opt} + \Delta - 1$ in equation (75) (in Appendix B), we can calculate the values of $\{dev_{l,k}\}$ in Theorem 3 as is shown in Table IV. Setting $\{\xi_k\}$ according to equation (70) with a sufficiently large tap-length of 100, we can see from Table IV that $\{dev_{l,k} > \xi_k\}$. Let $\{\gamma_k = \alpha_k/\xi_k\}$, then condition (68) in Theorem 3 is satisfied at each node k for any $\{\alpha_k > 0\}$. Moreover, Table IV also shows the values of $\{\gamma_{k,min} = \alpha_k/dev_{l,k}\}$ and $\{\gamma_k = \alpha_k/\xi_k\}$ in the case when $\{\alpha_k = 0.001\}$, which is the lower limit value for $\{\alpha_k\}$ in [24]. Clearly we have $\{\gamma_k > \gamma_{k,min}\}$.

In this simulation, in the proposed SDFTNLMS algorithm we set step-sizes $\{\gamma_k\}$ to $\{0.1\gamma_{k,min}\}$, $\{\gamma_{k,min}\}$ and $\{\alpha_k/\xi_k\}$ for $\{\alpha_k = 0.001\}$ respectively, and for robustness performance test we set $\{\alpha_k = 0.01\}$, which is the upper limit value for $\{\alpha_k\}$ in [24], and $\{\gamma_k = \alpha_k/\xi_k\}$. Table V shows steady-state network tap-lengths of the SDFTNLMS algorithms with

TABLE IV
VALUES FOR TERMS IN THEOREM 3.

diff. node	$\{\text{dev}_{l,k}\}$ ($\times 10^{-3}$) (75)	$\{\xi_k\}$ ($\times 10^{-4}$) (70)	$\{\gamma_{k,min}\},$ $\{\alpha_k =$ 0.001 (68)	$\{\gamma_k =$ $\alpha_k/\xi_k\},$ $\{\alpha_k =$ 0.001
1	4.65	0.90	0.22	11.15
2	7.28	3.48	0.14	2.88
3	5.62	1.94	0.18	5.15
4	5.78	4.32	0.17	2.31
5	7.35	4.22	0.14	2.37
6	4.61	2.99	0.22	3.34
7	5.37	2.49	0.19	4.02

these four sets of $\{\alpha_k, \gamma_k\}$, where the values are averaged over 100 independent experiments, with 100,000 iterations in each experiment to ensure mean convergence of the tap-length estimates. The steady-state network tap-length is defined as:

$$M_{\infty}^{Network} = \frac{1}{P} \sum_{k=1}^P M_{k,\infty}. \quad (71)$$

TABLE V
STEADY-STATE NETWORK TAP-LENGTHS FOR SDFTNLMS WITH DIFFERENT
PARAMETER SETTINGS.

optimal tap-length estimate	$\{\alpha_k =$ 0.001, $\{\gamma_k =$ $0.1\gamma_{k,min}\}$	$\{\alpha_k =$ 0.001, $\{\gamma_k =$ $\gamma_{k,min}\}$	$\{\alpha_k =$ 0.001, $\{\gamma_k =$ $\alpha_k/\xi_k\}$	$\{\alpha_k =$ 0.01, $\{\gamma_k =$ $\alpha_k/\xi_k\}$
51	10.21	44.55	51.31	51.43

It can be observed from Table V that by setting $\{\alpha_k = 0.001, \gamma_k = \alpha_k/\xi_k\}$ and $\{\alpha_k = 0.01, \gamma_k = \alpha_k/\xi_k\}$ in the proposed SDFTNLMS algorithm, which satisfy condition (68) in Theorem 3, the tap-length estimates across the network will converge in the mean to a value close to $L_{opt} + \Delta$. However, the tap-length will be underestimated, on average, to some extent at steady state compared to $L_{opt} + \Delta$, if the condition (68) is not met. This is because the ratios of $\{\alpha_k/(0.1\gamma_{k,min})\}$ and $\{\alpha_k/\gamma_{k,min}\}$ in the first two parameter settings in Table V will be higher than the values of $\{\xi_k\}$, which are calculated from equation (70), thus the values of the optimal tap-length estimates under these parameter settings will be lower than $L_{opt} + \Delta$.

C. Tracking Performance in a Low-Noise Case

The set up of this simulation is as follows. The unknown parameters are drawn from a white Gaussian sequence with zero mean and variance of 0.05, with a variable tap-length $L_{opt} = 300$ for $i \geq 12000$ otherwise $L_{opt} = 100$. The regressors and measurement noises are the same as those in the previous simulations, except that the measurement noise variances $\{\sigma_{v,k}^2\}$ are scaled to achieve high signal-to-noise ratios (SNRs) as is depicted in Fig.4(a).

In both the SDFTNLMS and GFTNLMS algorithms, at each node k the tap-length updates with an initial value of $M_{min} = m_{k,0} = 11$; the parameter Δ is adjusted to be 10 percent of the current filter length, with a minimum $\Delta_{min} = 10$. Moreover, to test the robustness of the proposed algorithms, we set parameters $\{\alpha_k\}$ to 0.01 and 0.001 (upper and lower

Fig. 4. The SNRs across the nodes in the network, where (a) is for the test of a low-noise case and (b) is for the test of a high-noise case.

limit values respectively as is shown in [24]). Parameters $\{\xi_k\}$ are calculated from (70) by using a sufficiently large tap-length of 500, and step-sizes $\{\gamma_k\}$ are set to $\{\alpha_k/\xi_k\}$, which are different across the nodes due to different SNRs. We checked that convergence condition (68) was met. In the diffusion FTNLMS (DFTNLMS) algorithm proposed in [28], leakage factors $\{\alpha_k\}$ are set to 0.001 to ensure a fast convergence rate of the tap-length, step-sizes $\{\gamma_k = 2\}$, parameters $\{\delta_k = 2\}$, with the parameter Δ as well as the initial tap-length value being the same as those in the SDFTNLMS and GFTNLMS algorithms. We choose step-size $\mu = 0.4$ for DFTNLMS algorithm, while $\mu = 0.2$ for all the other tested algorithms to obtain similar steady-state EMSEs in the first stage. The other parameters are unchanged from those in the previous simulations.

Fig.5 shows the evolution curves for different NLMS algorithms over adaptive networks, including the SDFTNLMS, GFTNLMS algorithms with varying leakage factors of $\{\alpha_k = 0.01, 0.001\}$, ATC diffusion NLMS algorithms [2] with fixed tap-lengths of 100 and 300, and DFTNLMS algorithm [28]. The results are averaged over 100 independent experiments. The network EMSE at time instant i is defined as [2]:

$$EMSE_i^{Network} = \frac{1}{P} \sum_{k=1}^P EMSE_{k,i} \quad (72)$$

and the definition for the network tap-length follows that in equation (68).

It is clear to see from Fig.5 that the proposed SDFTNLMS algorithm provides a good trade-off between the convergence rate and the steady-state performance. The ATC diffusion NLMS algorithm with a filter tap-length of 100 suffers from a large steady-state EMSE when the optimal tap-length becomes 300. The ATC diffusion NLMS algorithm with a fixed tap-length of 300 has a significantly slower convergence rate when the optimal tap-length is smaller than its set tap-length. Compared with the diffusion variable tap-length algorithm in [28], the proposed SDFTNLMS algorithms also show better tracking performance in the tested case, with a faster convergence rate in the first stage, and a smaller steady-state error in the second stage. Furthermore, the learning curves of the SDFTNLMS algorithms converge to those of the GFTNLMS

Fig. 5. Transient network tap-length (top) and EMSE (bottom) under a low-noise condition for SDFTNLMS, GFTNLMS algorithms with varying leakage factors of $\{\alpha_k = 0.01, 0.001\}$, and for comparison, ATC diffusion NLMS algorithms with fixed tap-lengths of 100 and 300, and DFTNLMS algorithm.

algorithms for all the tested parameters.

Fig. 5 also demonstrates great robustness performance of the proposed algorithms to leakage factors $\{\alpha_k\}$ (as well as step-sizes $\{\gamma_k\}$, since $\{\gamma_k = \alpha_k / \xi_k\}$). As is analyzed above, varying values of $\{\alpha_k\}$ (or $\{\gamma_k\}$) have slight influence on the steady-state performance of the proposed algorithms, but will affect the transient performance to some extent. Small values for $\{\alpha_k\}$ and $\{\gamma_k\}$ are more suitable in the low-noise case as is shown in Fig. 5, which avoid the tap-length overshoot, thus accelerate its convergence rate. However, convergence rates of network EMSEs are vary similar with different parameter settings of $\{\alpha_k, \gamma_k\}$.

Fig. 6. Transient network tap-length (top), and EMSE (bottom) under a high-noise condition for SDFTNLMS, GFTNLMS algorithms with varying leakage factors of $\{\alpha_k = 0.01, 0.001\}$, and for comparison, ATC diffusion NLMS algorithms with fixed tap-lengths of 100 and 300, and DFTNLMS algorithm.

D. Tracking Performance in a High-Noise Case

In this simulation, a high-noise environment is utilized. The unknown parameters are the same as those in the low-noise case, which are drawn from a white Gaussian sequence with zero mean and variance of 0.05, but the tap-length L_{opt} changes at the 20000th time instant from 100 to 300. Measurement noise variances are scaled to achieve low SNRs as is depicted in Fig. 4(b).

In both the SDFTNLMS and GFTNLMS algorithms, leakage factors $\{\alpha_k\}$ are set to 0.01 and 0.001 respectively to test the robustness to parameter settings. We also calculate parameters $\{\xi_k\}$ based on (70) by using a sufficiently large tap-length of 500, and step-sizes $\{\gamma_k\}$ are calculated from $\{\alpha_k / \xi_k\}$, which satisfy the convergence condition in equation (68). In the DFTNLMS algorithm, step-sizes $\{\gamma_k\}$ are 1, leakage factors $\{\alpha_k\}$ are 0.001 to ensure a fast convergence rate during the tap-length update process. Moreover, we set the step-size $\mu = 0.2$ in the DFTNLMS algorithm, while $\mu = 0.08$ in all the other tested algorithms, with the other parameters being unchanged.

Fig. 6 shows the evolution curves in this case for the SDFTNLMS, GFTNLMS algorithms with varying leakage factors of $\{\alpha_k = 0.01, 0.001\}$, ATC diffusion NLMS algorithms with fixed tap-lengths of 100 and 300, and DFTNLMS algorithm. The results are averaged over 100 independent experiments, and definitions of network tap-length and network EMSE follow those in equations (71) and (72) respectively.

Again from Fig. 6, the proposed SDFTNLMS algorithm provides a good trade-off between the convergence rate and the steady-state performance. The ATC diffusion NLMS algorithm with a filter tap-length of 100 still suffers from a large steady-state EMSE when the optimal tap-length becomes 300, resulting from a deficient tap-length. The ATC diffusion NLMS algorithm with a fixed tap-length of 300, on the other hand, suffers from a slow convergence rate at the first stage due to a too large filter tap-length. In addition, the DFTNLMS algorithm can still track the change of the unknown tap-length, but will suffer from inferior performance as is compared with the proposed SDFTNLMS algorithm with leakage factors $\{\alpha_k = 0.01\}$, since less measurements are exploited. Similar learning curves are obtained in the proposed SDFTNLMS algorithms and the GFTNLMS algorithms.

Furthermore, Fig. 6 also illustrates that the proposed variable tap-length algorithms, in the high-noise case, still show great robustness performance to leakage factors $\{\alpha_k\}$ (as well as step-sizes $\{\gamma_k\}$). Varying values of $\{\alpha_k\}$ and $\{\gamma_k\}$ still have slight influence on the steady-state performance of the proposed schemes. However, in the high-noise case, a fast convergence rate of network tap-length is obtained if we choose large values for $\{\gamma_k\}$, but the convergence rates of network EMSEs won't change significantly.

VIII. CONCLUSION

We formulated two variable tap-length algorithms over adaptive networks. One of them was the GFTNLMS algorithm, which is theoretically optimal. The other was the SDFTNLMS algorithm that allows information to exchange spatially in the neighborhood and is more practical. In the proposed

SDFTNLMS algorithm, we followed the concept of pseudo fractional tap-length and adapt-then-combine diffusion policy. Steady-state performance in terms of MSD and tap-length was analyzed. Moreover, we derived the sufficient condition for tap-length convergence in the mean. These analyses matched well with simulation results. General guidelines for parameter selections were also provided to facilitate practical use. Tracking performance simulations demonstrate that the proposed strategies outperform the ATC diffusion NLMS algorithm with a fixed tap-length as well as the existing diffusion variable tap-length algorithm, and that learning curves of the SDFTNLMS algorithm converge to those of the GFTNLMS algorithm at steady state. In addition, the proposed strategies show great robustness to parameter selections provided the parameter choice guidelines are satisfied.

APPENDIX A PROOF OF LEMMA 2

Assume that $N_k^{opt} = L_{opt} + \Delta$ is the optimal tap-length estimate at node k in cost function (10). Then N_k^{opt} is the smallest value that satisfies the inequality on the RHS of (10) by definition, we can conclude that

$$dev_{I,k} > \xi_k \quad (73)$$

$$dev_{II,k} \leq \xi_k \quad (74)$$

where $dev_{I,k}$ and $dev_{II,k}$ are defined in equations (63) and (64) respectively.

Recall that $\xi_k = \alpha_k/\gamma_k$ for any positive parameters ξ_k , α_k and γ_k , then from equation (73) we obtain relation (67). In view of relation (66), inequality (74) always holds for any $\xi_k > 0$.

On the other hand, assume that $\gamma_k > \alpha_k/dev_{I,k}$ for positive parameters α_k and γ_k . Let $\xi_k = \alpha_k/\gamma_k$, then $dev_{I,k} > \xi_k$, and $dev_{II,k} < \xi_k$ by definition. Noting that the steady-state MSE (as well as the steady-state distributed MSE, $MS E_{k,\infty}^{dist}(\mathbf{w}_k, N_k)$, $k = 1, 2, \dots, P$ that is defined in equation (7)) is basically a convex function of the tap-length [19], and that Δ is larger than the width of the suboptimum tap-length by assumption, we can conclude that $N_k^{opt} = L_{opt} + \Delta$ is the optimal tap-length estimate at node k in cost function (10).

APPENDIX B

Let $M_{k,\infty} = L_{opt} + \Delta - 1$ in equation (54), the RHS gives the value of $-dev_{I,k}$, by comparing equations (11), (46), (54) and (63). Combining equations (55) and (54), we get

$$dev_{I,k} = \overline{\sigma_{u,k}^2} \sigma_w^2 - \overline{\sigma_{u,k}^2} \mathbb{E} \left[\left\| \tilde{\mathbf{w}}_{k,\Delta}'(\infty) \right\|^2 \right] - \mathbb{E} \left[\left\| \tilde{\psi}(\infty) \right\|_{b_{I,k}}^2 \right] \quad (75)$$

where values of the two right-most terms will be relatively small compared to the value of $\overline{\sigma_{u,k}^2} \sigma_w^2$. Then the term $dev_{I,k}$ can be approximately deemed as the EMSE in the case when one tail element of the unknown parameter vector is not modeled by the tap-weight vector. On the other hand, the term $EMSE_{k,\infty}$ defined in (69) refers to the case when the unknown parameter vector is fully modeled. Thus intuitively we have $\{dev_{I,k} > EMSE_{k,\infty}\}$. Let $\{\xi_k = EMSE_{k,\infty}\}$ and $\{\gamma_k = \alpha_k/\xi_k\}$, then Theorem 3 will be satisfied for any $\{\alpha_k \geq 0\}$.

ACKNOWLEDGEMENT

The authors would like to thank all the reviewers and the associate editor for greatly improving the quality of this manuscript.

REFERENCES

- [1] J.-W. Lee, S.-E. Kim, W.-J. Song, and A. H. Sayed, "Spatio-temporal diffusion strategies for estimation and detection over networks," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4017–4034, Aug. 2012.
- [2] 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.
- [3] D. P. Bertsekas, "A new class of incremental gradient methods for least squares problems," *SIAM J. Optim.*, vol. 7, no. 4, pp. 913–926, Nov. 1997.
- [4] A. Nedic and D. P. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," *SIAM J. Optim.*, vol. 12, no. 1, pp. 109–138, 2001.
- [5] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 798–808, Apr. 2005.
- [6] 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.
- [7] A. H. Sayed and C. G. Lopes, "Adaptive processing over distributed networks," *IEICE Trans. Fund. Electron., Commun. Comput. Sci.*, vol. E90-A, no. 8, pp. 1504–1510, Aug. 2007.
- [8] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4289–4305, Aug. 2012.
- [9] 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.
- [10] N. Takahashi, I. Yamada, and A. H. Sayed, "Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 8, no. 9, pp. 4795–4810, Sep. 2010.
- [11] 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.
- [12] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2069–2084, Sep. 2010.
- [13] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6217–6234, Dec. 2012.
- [14] X. Zhao, S.-Y. Tu, and A. H. Sayed, "Diffusion adaptation over networks under imperfect information exchange and non-stationary data," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3460–3475, Jul. 2012.
- [15] A. Khalili, M. A. Tinati, A. Rastegarnia, and J. A. Chambers, "Steady-state analysis of diffusion LMS adaptive networks with noisy links," *IEEE Trans. Signal Process.*, vol. 60, no. 2, pp. 974–979, Feb. 2012.
- [16] X. Zhao and A. H. Sayed, "Performance limits for distributed estimation over LMS adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5107–5124, Oct. 2012.
- [17] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Adaptive robust distributed learning in diffusion sensor networks," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4692–4707, Oct. 2011.
- [18] F. Riera-Palou, J. M. Noras and D. G. M. Cruickshank, "Linear equalisers with dynamic and automatic length selection," *Electronics Letters*, vol. 37, no. 25, pp. 1553–1554, 2001.
- [19] Y. Gong and C. F. N. Cowan, "An LMS style variable tap-length algorithm for structure adaptation," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2400–2407, Jul. 2005.
- [20] Y. Gu, K. Tang, H. Cui, and W. Du, "LMS algorithm with gradient descent filter length," *IEEE Signal Process. Lett.*, vol. 11, no. 3, pp. 305–307, Mar. 2004.
- [21] C. Schüldt, F. Lindstrom, H. Li and I. Claesson, "Adaptive filter length selection for acoustic echo cancellation," *Signal Process.*, vol. 89, no. 6, pp. 1185–1194, 2009.
- [22] Y. Zhang and J. A. Chambers, "Convex combination of adaptive filters for a variable tap-length LMS algorithm," *IEEE Signal Process. Lett.*, vol. 13, no. 10, pp. 628–631, Oct. 2006.
- [23] Y. Zhang, J. A. Chambers, S. Sane, P. Kendrick, and T. J. Cox, "A new variable tap-length LMS algorithm to model an exponential decay impulse response," *IEEE Signal Process. Lett.*, vol. 14, no. 4, pp. 263–266, Apr. 2007.
- [24] Y. Zhang, N. Li, J. A. Chambers and A. H. Sayed, "Steady-state performance analysis of a variable tap-length LMS algorithm," *IEEE Trans. Signal Process.*, vol. 56, no. 2, pp. 839–845, Feb. 2008.

- [25] N. Li, Y. Zhang, Y. Hao and Y. Zhao, "A new variable tap-length LMS algorithm with variable error width," in *Signal Process., 2008. ICSP 2008. 9th International Conference on*, 2008, pp. 276–279.
- [26] Z. Liu, "Variable tap-length linear equaliser with variable tap-length adaptation step-size," *Electronics Letters*, vol. 50, no. 8, pp. 587–589, 2014.
- [27] L. Li, Y. Zhang, and J. A. Chambers, "Variable length adaptive filtering within incremental learning algorithms for distributed networks," in *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*. IEEE, 2008, pp. 225–229.
- [28] L. Li, J. Feng, J. He, and J. A. Chambers, "A distributed variable tap-length algorithm within diffusion adaptive networks," *AASRI Procedia*, vol. 5, pp. 77–84, 2013.
- [29] S. Haykin, *Adaptive Filter Theory*, 4th ed. New Jersey: Prentice-Hall, 2002.
- [30] A. H. Sayed, *Fundamentals of Adaptive Filtering*. New York: Wiley, 2003.
- [31] H. C. Huang and J. Lee, "A new variable step-size NLMS algorithm and its performance analysis," *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 2055–2060, 2012.