VARIABLE LENGTH ADAPTIVE FILTERING WITHIN INCREMENTAL LEARNING ALGORITHMS FOR DISTRIBUTED NETWORKS

Leilei Li, Yonggang Zhang and Jonathon A. Chambers

Advanced Signal Processing Group, Department of Electronic & Electrical Engineering
Loughborough University, Loughborough LE11 3TU, UK.
(Email: l.li2@lboro.ac.uk, Y.Zhang5@lboro.ac.uk, j.a.chambers@lboro.ac.uk)

ABSTRACT

In this paper we propose the use of variable length adaptive filtering within the context of incremental learning for distributed networks. Algorithms for such incremental learning strategies must have low computational complexity and require minimal communication between nodes as compared to centralized networks. To match the dynamics of the data across the network we optimize the length of the adaptive filters used within each node by exploiting the statistics of the local signals to each node. In particular, we use a fractional tap-length solution to determine the length of the adaptive filter within each node, the coefficients of which are adapted with an incremental-learning learning algorithm. Simulation studies are presented to confirm the convergence properties of the scheme and these are verified by theoretical analysis of excess mean square error and mean square deviation.

***Index Terms*—** variable tap-length, distributed processing, adaptive filters, incremental algorithm

1. INTRODUCTION

Distributed solutions, only exploiting local data exchanges and communications between immediate neighboring nodes, have been proposed [1, 2, 3] as a consequence of their reduced processing and communications requirements as compared to central solutions. The applications of such distributed adaptive networks range from sensor networks to environmental monitoring and factory instrumentation [4, 5]. However, in many applications of such distributed solutions the tap-length of the adaptive filters is assumed fixed, which is not appropriate for certain situations where the optimal tap-length is unknown or variable.

As a key parameter, tap-length plays an important role in the design of adaptive filters. It is well known that the selection of tap-length significantly influences the performance of adaptive filters: deficient tap-length is likely to result in increase of the mean-square-error (MSE); whereas the computational cost and the excess mean-square-error (EMSE) may become too high if the tap-length is too large. Since the concept of variable tap-length in adaptive filters was initially proposed in [6], various related results [7, 8, 9, 10] have been reported for the single adaptive filter case. As compared with other methods, Gong and Cowan [10] introduce a low-complexity and robust fractional tap-length (FT) algorithm based on instantaneous errors, which obtains improved convergence properties. The steady-state performance analysis of the FT algorithm are provided in [10, 11, 12], which also provide a guideline for parameter selection in the FT algorithm.

Motivated by both the ideas of distributed estimation and variable tap-length, this paper proposes an adaptive learning algorithm which solves the parameter estimation problem in a distributed network where the tap-length of the optimal filter is not known. The steady-state performance of the new algorithm for Gaussian data is studied using weighted spatial-temporal energy conservation arguments. In particular, we derive theoretical expressions for the mean-square deviation (MSD), the EMSE and the MSE for each node within the network. Simulation studies are presented to confirm the convergence properties of the scheme and to verify the theoretical results.

The remainder of this paper is organized as follows. The proposed algorithm and its motivation are introduced in Section 2. The analysis of the fractional tap-length function is described in Section 3. Simulation results that confirm the analysis of the presented algorithm are given in section 4. Section 5 offers conclusions.

The following notations are used in this paper: boldface small and capital letters are used for random complex vectors or scalars and matrices; normal font is employed for deterministic quantities; $(\cdot)^{\mathrm{T}}$ and $(\cdot)^*$ denote transposition and complex-conjugate transposition respectively; $|\cdot|^2$ and $\|\cdot\|^2$ denote the absolute squared operation and squared Euclidean norm operation.

2. ESTIMATION PROBLEM AND FORMULATION

Consider an N-node network, where we collect data and seek an unknown vector w^o, whose tap weights and tap-length are desirable to be estimated. We can decouple the adaptation rules for the tap weights and tap-length, which means that the selection of one does not depend on the other. Assuming the tap-length is L, which is estimated by the tap-length search solution that will be discussed later, each node k obtains the time observations $\{d_k(i),\, u_{k,i}\}$ of zero-mean complex spatial data $\{\mathbf{d}_k,\, \mathbf{u}_k\}$ at time instant i. Each $\mathbf{d}_k$ is a scalar value and each $\mathbf{u}_k$ is a $1 \times L$ row regression vector. In order to seek w, we formulated the linear minimum mean-square estimation problem:

$$\min_{w} J_L(w) \quad \text{and} \quad J_L(w) = E\|\mathbf{d} - \mathbf{U}w\|^2 \qquad (1)$$

where two global matrices of the desired response and regression data are given by

$$\mathbf{d} \triangleq \mathrm{col}\{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N\}, \quad (N \times 1) \tag{2}$$

$$\mathbf{U} \triangleq \mathrm{col}\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N\}, \quad (N \times L). \tag{3}$$

Let w_i be an estimate for w^o at time instant i and $\psi_k(i)$ be a local estimate for w^o at node k at time instant i. We introduce the incremental steepest-descent solution [2], which estimates the optimal solution w^o by iterating $\psi_0(i)$ through an incremental network in the following manner:

$$\psi_k(i) = \psi_{k-1}(i) + \mu_k u_{k,i}^*(d_k(i) - u_{k,i}w_{i-1}) \quad k = 1, \ldots N \tag{4}$$

where we start from the initial condition $\psi_0(i) = w_{i-1}$ at node 1.

With the assumption of the tap-length L, we define the segmented cost function as

$$J_M(w) \triangleq E\|\mathbf{d} - \mathbf{U}_M w_M\|^2 \tag{5}$$

where $1 \le M \le L$, w_M and $\mathbf{U}_M$ consist of the initial M elements of w and the initial M column vectors of $\mathbf{U}$, respectively, as

$$w_M \triangleq \mathrm{col}\{w(1), \ldots, w(M)\}, \quad (1 \times M) \tag{6}$$

$$\mathbf{U}_M \triangleq \mathrm{col}\{\mathbf{u}_1(1:M), \ldots, \mathbf{u}_N(1:M)\}, \ (N \times M). \tag{7}$$

We pose the minimum difference of mean-square errors estimation problem to seek the optimal tap-length L^o

$$\min\left\{L \middle| J_{L-\triangle}(w) - J_L(w) \le \varepsilon\right\} \tag{8}$$

where ε, predetermined by system requirements, is a small positive value and $\triangle$ is an integer value to avoid the suboptimum tap-lengths. The segmented mean-square estimation error and the mean-square estimation error at time instant i are respectively given by,

$$J_{L(i)-\triangle}(w_{i-1}) = e^2_{L(i)-\triangle}(w_{i-1}) \tag{9}$$

$$J_{L(i)}(w_{i-1}) = e^2_{L(i)}(w_{i-1}). \tag{10}$$

We start from the standard pseudo fractional tap-length implement as in [10]

$$L_f(i+1) = (L_f(i) - \alpha) + \beta \cdot [e^2_{L(i)-\triangle}(w_{i-1}) - e^2_{L(i)}(w_{i-1})] \tag{11}$$

where α and β are small positive values, α is the leakage factor used to prevent $L_f(i)$ from increasing to an undesirably large value and β is the step-size for $L_f(i)$ adaptation. Then the integer tap-length $L(i)$ is adjusted according to

$$L(i+1) = \begin{cases} \lfloor L_f(i) \rfloor & \text{if} |L(i) - L_f(i+1)| \ge \nu \\ L(i) & \text{otherwise} \end{cases} \tag{12}$$

where $\lfloor \cdot \rfloor$ rounds the embraced value to the nearest integer and the step-size parameter ν is a small integer.

Distributed networks motivate us to decompose the segment cost function and cost function as

$$J_{L-\triangle}(w) = \sum_{k=1}^{N} J_{k,L-\triangle}(w) \tag{13}$$

$$J_L(w) = \sum_{k=1}^{N} J_{k,L}(w) \tag{14}$$

where

$$J_{k,L-\triangle}(w) = E|\mathbf{d}_k - \mathbf{u}_k(1:M)w(1:M)|^2, \tag{15}$$

and

$$J_{k,L}(w) = E|\mathbf{d}_k - \mathbf{u}_k w|^2. \tag{16}$$

Such results allow us to rewrite (11) as

$$L_f(i+1) = (L_f(i) - \alpha) + \beta \cdot \sum_{k=1}^{N} \gamma_k(w_{i-1}) \tag{17}$$

with $\gamma_k(w_{i-1}) = e^2_{k,L(i)-\triangle}(w_{i-1}) - e^2_{k,L(i)}(w_{i-1})$. With proper choice of α and β, we will have $L(i) \to L^o$ as $i \to \infty$ for any initial condition, where L^o is an optimal tap-length and always larger than the true tap-length of w^o. Let $\ell_{k,f}(i)$ denote the local estimate of the fractional tap-length at node k at time i. α and β can be decomposed to $\alpha = \sum_{k=1}^{N} \alpha_k$ and $\beta = \sum_{k=1}^{N} \beta_k$ respectively, where α_k indicates the local leakage factor and β_k denotes the local step-size for $\ell_{k,f}(i)$ adaptation at node k. In the defined cycle, node k received the estimated fractional tap-length $\ell_{k-1,f}(i)$ from the node $k-1$. At each time instant i, we start with the initial condition $\ell_{0,f}(i) = L_f(i)$ at node 1 ($L_f(i)$ is the current global estimation for L^o). At the end of the cycle, the local estimation $\ell_{N,f}(i)$ is employed as the global estimation $L_f(i+1)$ for the next time $i+1$ and the integer tap-length $L(i+1)$ is also evaluated by (12). Such implementation of a centralized solution for tap-length adaptation is described as follows:

For each time instant $i \ge 0$ repeat:
$\ell_{0,f}(i) = L_f(i)$
For k=1,… N
$\quad \ell_{k,f}(i) = \ell_{k-1,f}(i) - \alpha_k + \beta_k \cdot \gamma_k(w_{i-1})$
end
$L_f(i+1) = \ell_{N,f}(i)$
$L(i+1) = \begin{cases} \lfloor L_f(i+1) \rfloor & if |L(i) - L_f(i+1)| \ge \nu \\ L(i) & \text{otherwise} \end{cases}$
where $\gamma_k(w_{i-1}) = e^2_{k,L(i)-\triangle}(w_{i-1}) - e^2_{k,L(i)}(w_{i-1})$

where we always hold $\ell_{k,f}(i) \ge L_{\min}$, which is the minimum tap-length. This method also requires all nodes to access the global information w_{i-1} and only adapts the integer tap-length at the end of a cycle. A fully distributed solution can be achieved by evaluating the segmented mean-square error and mean-square error from its local estimate $\psi^{(i)}_{k-1}$. This approach leads to distributed adaptation for both tap weights and tap-length. For the estimation of tap weights, a distributed version of algorithm of (4) is presented in [2] as

$$\psi_k(i) = \psi_{k-1}(i) + \mu_k u^*_{k,i}(d_k(i) - u_{k,i}\psi_{k-1}(i)) \quad k = 1, \ldots N \tag{18}$$

Let $L_k(i)$ denote the local estimate of L^o at node k at time i. A distributed solution for tap-length adaptation is summarized below:

For each time instant $i \ge 0$ repeat:
$\ell_{0,f}(i) = L_f(i)$
For k=1,… N
$\quad \ell_{k,f}(i) = \ell_{k-1,f}(i) - \alpha_k + \beta_k \cdot \gamma_k(\psi^{(i)}_{k-1})$
$\quad L_{k+1}(i) = \begin{cases} \lfloor l_k(i) \rfloor & if |L_k(i) - \ell_{k,f}(i)| \ge \nu_k \\ L_k(i) & \text{otherwise} \end{cases}$
end
$L_f(i+1) = \ell_{N,f}(i)$
where $\gamma_k(\psi^{(i)}_{k-1}) = e^2_{k,L_k(i)-\triangle}(\psi^{(i)}_{k-1}) - e^2_{k,L_k(i)}(\psi^{(i)}_{k-1})$

where

$$e_{k,M(i)}(\psi_{k-1}^{(i)}) = d_k(i) - u_{k,i}(1:M(i))\psi_{k-1}^{(i)}(1:M(i)) \quad (19)$$

$$e_{k,L_k(i)}(\psi_{k-1}^{(i)}) = d_k(i) - u_{k,i}\psi_{k-1}^{(i)} \quad (20)$$

with $M(i) = L_k(i) - \triangle$. As discussed in [2], Fig. 1 illustrates that in optimization theory the distributed solution can outperform the centralized solution. The simulated curves are obtained by averaging 500 independent Monte Carlo runs with $\mu_k = 0.05$. The network utilized in the experiment has 12 nodes and seeks an unknown filter with variable tap-length $M = 10$ for $i \geq 140$ otherwise $M = 19$. The input signal is Gaussian data with $R_{u,k} = I$ and the background noise is zero mean real white Gaussian with $\sigma_{v,k}^2 = 0.001$. For both algorithms, we choose the parameters $\nu_k = \nu = 1$, $\alpha_k = 0.03$, $\beta_k = 1$, $\triangle = 4$ and $L_{\min} = L_f(0) = 6$, where the selection of parameters follows the rules in [10]-[12]. Note that, in theory, we do not need to set the upper bound for the value of fractional tap-length. However, since the length estimation uses instantaneous errors rather than averaged errors, the fractional tap-length may be, at certain time instants, at an undesired large value, which leads instantaneously to high computational and memory cost. Therefore, in practice, the upper bound is required to avoid such situation.

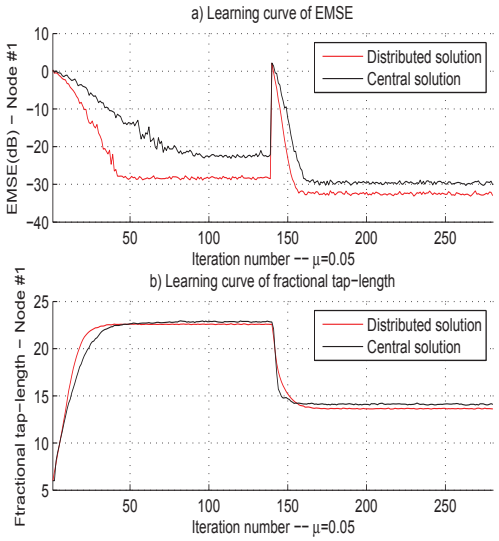

Fig. 1. Evolution curves for the distributed solution and the centralized solution at node 1: a) EMSE performance b) Fractional tap-length performance.

3. PERFORMANCE ANALYSIS

In this section, we evaluate the steady-state performance of the distributed algorithm using weighted spatial-temporal energy conservation arguments [2]. Due to space constraints, only the main procedures are given. The following assumptions are utilized

A1) In order to simplify the analysis, the estimated tap-length is assumed to be fixed at the steady-state stage.

A2) The unknown system vector w^o and $\{\mathbf{d}_k(i), \mathbf{u}_{k,i}\}$ construct:

$$\mathbf{d}_k(i) = \mathbf{u}_{k,i}w^o + \mathbf{v}_k(i) \quad (21)$$

where $\mathbf{v}_k(i)$ is a temporally and spatially white noise sequence with variance $\sigma_{v,k}^2$ and independent of $\mathbf{d}_l(j)$ for $k \neq l$ or $i \neq j$ and $\mathbf{u}_{l,j}$ for all l and j;

A3) $\mathbf{u}_{k,i}$ is spatially and temporally independent, namely $\mathbf{u}_{k,i}$ is independent of $\mathbf{u}_{l,i}$ and $\mathbf{u}_{k,j}$ for $k \neq l$ or $i \neq j$

The following local error signals defined as in [2] are introduced to perform the evaluation:

$$\tilde{\boldsymbol{\psi}}_{k-1}^{(i)} \triangleq w^o - \boldsymbol{\psi}_{k-1}^{(i)}, \quad \tilde{\boldsymbol{\psi}}_{k}^{(i)} \triangleq w^o - \boldsymbol{\psi}_{k}^{(i)} \quad (22)$$

$$\mathbf{e}_{a,k}(i) \triangleq \mathbf{u}_{k,i}\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}, \quad \mathbf{e}_{p,k}(i) \triangleq \mathbf{u}_{k,i}\tilde{\boldsymbol{\psi}}_{k}^{(i)} \quad (23)$$

Introduce further the weighted error signals:

$$\boxed{\mathbf{e}_{p,k}^{\Sigma_k,(i)} \triangleq \mathbf{u}_{k,i}\Sigma_k\tilde{\boldsymbol{\psi}}_{k}^{(i)}, \quad \mathbf{e}_{a,k}^{\Sigma_k,(i)} \triangleq \mathbf{u}_{k,i}\Sigma_k\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}} \quad (24)$$

where Σ_k is a Hermitian positive-definite weighting matrix that we are free to choose at each node k. Note that the output error $\mathbf{e}_k(i) = \mathbf{e}_{a,k}(i) + \mathbf{v}_k(i)$. As a result, the steady-state quantities for each node are formed as

$$\eta_k \triangleq E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(\infty)}\|^2 \quad \text{(MSD)} \quad (25)$$

$$\zeta_k \triangleq E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(\infty)}\|_{R_{u,k}}^2 \quad \text{(EMSE)} \quad (26)$$

$$\xi_k \triangleq \zeta_k + \sigma_{v,k}^2 \quad \text{(MSE)}. \quad (27)$$

where we use the weighted norm notation $\|x\|_\Sigma^2 = x^*\Sigma x$ for a vector x and Hermitian positive-definite $\Sigma > 0$.

As presented in [2], the spatial-temporal energy conservation relation between two successive nodes is given by

$$\boxed{E\|\tilde{\boldsymbol{\psi}}_k\|_{\Sigma_k}^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}\|_{\Sigma_k'}^2 + \mu_k^2\sigma_{v,k}^2 E\|\mathbf{u}_k\|_{\Sigma_k}^2} \quad (28)$$

where Σ_k' is given by

$$\boxed{\Sigma_k' = \Sigma_k - \mu_k E(\mathbf{u}_k^*\mathbf{u}_k\Sigma_k + \Sigma_k\mathbf{u}_k^*\mathbf{u}_k) + \mu_k^2 E(\|\mathbf{u}_k\|_{\Sigma_k}^2\mathbf{u}_k^*\mathbf{u}_k)}. \quad (29)$$

We assume the regressors are from a circular Gaussian distribution and introduce the eigendecomposition $R_{u,k} = Q_k\Lambda_k Q_k^*$, where Q_k is unitary and Λ_k is a diagonal matrix with the eigenvalues of $R_{u,k}$. The transformed quantities are defined as

$$\bar{\boldsymbol{\psi}}_k \triangleq Q_k^*\tilde{\boldsymbol{\psi}}_k,\ \bar{\boldsymbol{\psi}}_{k-1} \triangleq Q_k^*\tilde{\boldsymbol{\psi}}_{k-1},\ \bar{\mathbf{u}}_k \triangleq \mathbf{u}_k Q_k$$
$$\bar{\Sigma}_k \triangleq Q_k^*\Sigma_k Q_k,\ \bar{\Sigma}_k' \triangleq Q_k^*\bar{\Sigma}_k Q_k$$

Since Q_k is unitary, we can have $E\|\tilde{\boldsymbol{\psi}}_{k-1}\|_{\Sigma_k}^2 = E\|\bar{\boldsymbol{\psi}}_{k-1}\|_{\bar{\Sigma}_k}^2$ and $E\|\mathbf{u}_k\|_\Sigma^2 = E\|\bar{\mathbf{u}}_k\|_{\bar{\Sigma}}^2$. Let $Tr\{A\}$ denote the trace of a matrix A. Using the results for Gaussian data [13], we obtain the transformed expressions from (28) and (29)

$$\boxed{E\|\bar{\boldsymbol{\psi}}_k\|_{\bar{\Sigma}_k}^2 = E\|\bar{\boldsymbol{\psi}}_{k-1}\|_{\bar{\Sigma}_k'}^2 + \mu_k^2\sigma_{v,k}^2 Tr(\Lambda_k\bar{\Sigma}_k)} \quad (30)$$

where $\bar{\Sigma}_k'$ is given by

$$\boxed{\bar{\Sigma}_k' = \bar{\Sigma}_k - \mu_k X_k + \mu_k^2 Y_k} \quad (31)$$

where $X_k = \Lambda_k\bar{\Sigma}_k + \bar{\Sigma}_k\Lambda_k$ and $Y_k = \Lambda_k Tr(\bar{\Sigma}_k\Lambda_k) + \tau\Lambda_k\bar{\Sigma}_k\Lambda_k$ with $\tau = 1$ for complex data and $\tau = 2$ for real data.

Therefore, we proceed as in [13, 14] by introducing the $M^2 \times 1$ vectors:

$$\delta_k = \mathrm{vec}\{\bar{\Sigma}_k\}\ , \delta'_k = \mathrm{vec}\{\bar{\Sigma}'_k\} \text{ and } \lambda_k = \mathrm{vec}\{\Lambda_k\} \tag{32}$$

where we use the $\mathrm{vec}\{\cdot\}$ notation in two ways: $\delta = \mathrm{vec}\{\Sigma\}$ denotes an $M^2 \times 1$ column vector whose entries are formed by stacking the successive columns of an $M \times M$ matrix on top of each other, and $\Sigma = \mathrm{vec}\{\delta\}$ indicates a matrix whose entries are recovered from δ. We exploit the following useful property for the $\mathrm{vec}\{\cdot\}$ notation when working with Kronecker products: for any matrices $\{P, \Sigma, Q\}$ of compatible dimensions, it holds that

$$\mathrm{vec}\{P\Sigma Q\} = (Q^{\mathrm{T}} \otimes P)\mathrm{vec}\{\Sigma\}. \tag{33}$$

The choice of Σ_k can make both $\bar{\Sigma}_k$ and $\bar{\Sigma}'_k$ become diagonal in (31). Applying the $\mathrm{vec}\{\cdot\}$ operation to both sides of (31), a linear relation between the corresponding vectors $\{\delta'_k, \delta_k\}$ is obtained, namely,

$$\delta'_k = F_k\delta_k \tag{34}$$

where F_k is an $M^2 \times M^2$ matrix and given by

$$F_k = I - 2\mu_k(I \otimes \Lambda_k) + \mu_k^2(\tau(\Lambda_k \otimes \Lambda_k) + \lambda_k\lambda_k^T). \tag{35}$$

Therefore, expression (30) becomes

$$E\|\bar{\psi}_k^{(i)}\|^2_{\mathrm{vec}\{\delta_k\}} = E\|\bar{\psi}_{k-1}^{(i)}\|^2_{\mathrm{vec}\{F_k\delta_k\}} + \mu_k^2\sigma_{v,k}^2(\lambda_k^T\delta_k) \tag{36}$$

where we reuse the time index i for clarity. For simplicity of notation, we drop the $\mathrm{vec}\{\cdot\}$ notation from the subscripts in (36):

$$\boxed{E\|\bar{\psi}_k^{(i)}\|^2_{\delta_k} = E\|\bar{\psi}_{k-1}^{(i)}\|^2_{F_k\delta_k} + \mu_k^2\sigma_{v,k}^2(\lambda_k^T\delta_k)}. \tag{37}$$

Let $\boldsymbol{\rho}_k = \bar{\psi}_k^{(\infty)}$, then

$$E\|\boldsymbol{\rho}_k\|^2_{\delta_k} = E\|\boldsymbol{\rho}_{k-1}\|^2_{F_k\delta_k} + \mu_k^2\sigma_{v,k}^2(\lambda^T\delta_k). \tag{38}$$

By iterating (38) over one cycle, N coupled equations are obtained:

$$\begin{aligned}
E\|\boldsymbol{\rho}_1\|^2_{\delta_1} &= E\|\boldsymbol{\rho}_N\|^2_{\mathrm{F}_1\delta_1} + g_1\delta_1 \\
E\|\boldsymbol{\rho}_2\|^2_{\delta_2} &= E\|\boldsymbol{\rho}_1\|^2_{\mathrm{F}_2\delta_2} + g_2\delta_2 \\
&\vdots \\
E\|\boldsymbol{\rho}_{k-1}\|^2_{\delta_{k-1}} &= E\|\boldsymbol{\rho}_{k-2}\|^2_{\mathrm{F}_{k-1}\delta_{k-1}} + g_{k-1}\delta_{k-1} \qquad (39)\\
E\|\boldsymbol{\rho}_k\|^2_{\delta_k} &= E\|\boldsymbol{\rho}_{k-1}\|^2_{\mathrm{F}_k\delta_k} + g_k\delta_k \qquad (40)\\
&\vdots \\
E\|\boldsymbol{\rho}_N\|^2_{\delta_N} &= E\|\boldsymbol{\rho}_{N-1}\|^2_{\mathrm{F}_N\delta_N} + g_N\delta_N
\end{aligned}$$

with $g_k = \mu_k^2\sigma_{v,k}^2\lambda^T$. Choose the free parameters δ_k and δ_{k-1} such that $\delta_{k-1} = F_k\delta_k$ and combine (39) and (40), then we iterate this procedure across the cycle to obtain

$$\begin{aligned}
E\|\boldsymbol{\rho}_{k-1}\|^2_{\delta_{k-1}} = E\|\boldsymbol{\rho}_{k-1}\|^2_{\mathrm{F}_k\cdots\mathrm{F}_N\mathrm{F}_1\cdots\mathrm{F}_{k-1}\delta_{k-1}} \\
+g_k\mathrm{F}_{k+1}\cdots\mathrm{F}_N\mathrm{F}_1\cdots\mathrm{F}_{k-1}\delta_{k-1} \\
+g_{k+1}\mathrm{F}_{k+2}\cdots\mathrm{F}_N\mathrm{F}_1\cdots\mathrm{F}_{k-1}\delta_{k-1} \\
\cdots + g_{k-2}\mathrm{F}_{k-1}\delta_{k-1} + g_{k-1}\delta_{k-1}. \qquad (41)
\end{aligned}$$

Let

$$\Pi_{k-1,l} = \mathrm{F}_{k+l-1}\cdots\mathrm{F}_N\mathrm{F}_1\cdots\mathrm{F}_{k-1},\ \ l = 1, 2, \ldots, N \tag{42}$$

$$a_{k-1} = g_k\Pi_{k-1,2} + \cdots + g_{k-2}\Pi_{k-1,N} + g_{k-1} \tag{43}$$

then

$$\boxed{E\|\boldsymbol{\rho}_{k-1}\|^2_{(I-\Pi_{k-1,1})\delta_{k-1}} = a_{k-1}\delta_{k-1}}. \tag{44}$$

Since we are free to select the weight vector δ_{k-1} in (44), choosing $\delta_{k-1} = (I - \Pi_{k-1,1})^{-1}q$ or $\delta_{k-1} = (I - \Pi_{k-1,1})^{-1}\lambda_k$ results in the expressions for the steady-state MSD, EMSE and MSE at node k:

$$\eta_k = E\|\boldsymbol{\rho}_{k-1}\|^2_q = a_{k-1}(I - \Pi_{k-1,1})^{-1}q \quad \text{(MSD)} \tag{45}$$

$$\zeta_k = E\|\boldsymbol{\rho}_{k-1}\|^2_{\lambda_k} = a_{k-1}(I - \Pi_{k-1,1})^{-1}\lambda_k \quad \text{(EMSE)} \tag{46}$$

$$\xi_k = \zeta_k + \sigma_{v,k}^2 \quad \text{(MSE)} \tag{47}$$

where $q = \mathrm{vec}\{I\}$ and $\lambda_k = \mathrm{vec}\{\Lambda_k\}$.

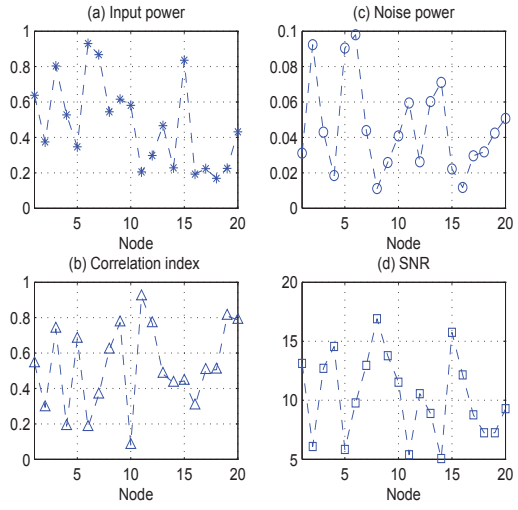

Fig. 2. Node profile: a) Input power b) Noise power c) Correlation index d) SNR.

4. SIMULATIONS

In this section, we compare the theoretical performance with the computer simulations in a system identification scenario. All simulations results are averaged over 100 independent Monte Carlo runs. The steady-state curves are obtained by averaging the last 2000 instantaneous samples of $20,000$ iterations. We considered a network with 20 nodes seeking an unknown filter with $M = 10$ taps. A correlated Gaussian signal is used to generate the inputs at each node k which satisfies the recursion

$$u_k(i) = a_k u_k(i-1) + b_k \cdot c_k(i). \tag{48}$$

Expression (48) produces a first-order autoregressive (AR) process with a pole at a_k; c_k is a white, zero-mean, Gaussian random sequence with unity variance, $a_k \in (0, 1]$ and $b_k = \sqrt{\sigma_{u,k}^2 \cdot (1 - a_k^2)}$.

In this way, the covariance matrix $R_{u,k}$ of the regressor $\mathbf{u}_{k,i}$ is a 10×10 Toeplitz matrix with entries $r_k(m) = \sigma_{u,k}^2 a_k^{|m|}$, $m = 0, \ldots M-1$ with $\sigma_{u,k}^2 \in [0, 1)$. The background noise has variance $\sigma_{v,k}^2 \in [0, 0.1)$ across the network. The statistical profiles are illustrated in Fig. 2. Fig. 3 and Fig. 4 show that the theoretical results for MSE match well with the simulated results, where $\nu_k = 1$, $\alpha_k = 0.05$, $\beta_k = 1$, $\triangle = 3$, and $L_{\min} = L_f(0) = 4$. Similar results for EMSE and MSD have been obtained but are not included due to space limitations.

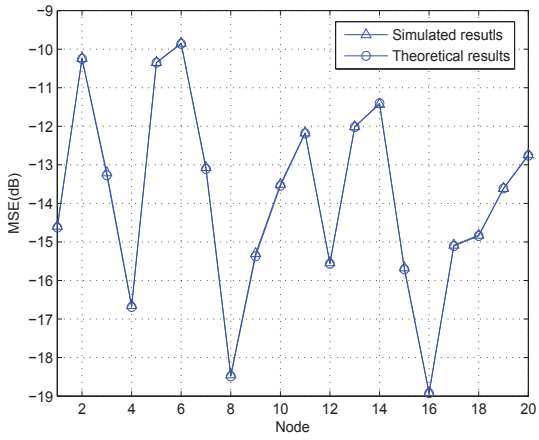

Fig. 3. MSE versus node k - $\mu_k = 0.02$.

5. CONCLUSION

In this paper, a new algorithm has been proposed for structure adaptation of adaptive filters in an incremental distributed network. Under the assumptions A1, A2 and A3, weighted spatial-temporal energy conservation argument are used to analyze steady-state mean square performance in the Gaussian case. Numerical simulations show that there is a good match between simulated results and theoretical results. Future work includes more sophisticated distributed estimation in a cooperative fashion.

6. REFERENCES

[1] A. H. Sayed and C. G. Lopes, "Distributed recursive least-squares strategies over adaptive networks," *Proc. 40th Asilomar Conference on Signals, Systems and Computers*, pp. 223–237, Oct. 2006.

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

[3] L. Li and J. A. Chambers, "A new incremental affine projection based adaptive algorithm for distributed networks," *Fast Communication in Signal Processing*, vol. 88, no. 10, pp. 2599–2603, 2008.

[4] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," *Proc. ICASSP*, pp. 2033–2036, May. 2001.

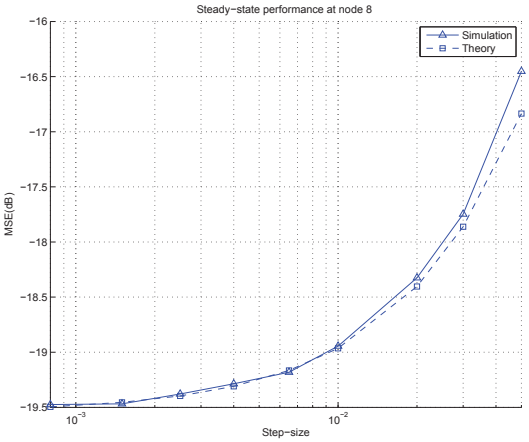

Fig. 4. MSE versus step-size at node 8.

[5] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for sistributed optimization," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 798–808, Apr. 2007.

[6] Z. Pritzker and A. Feuer, "Variable length stochastic gradient algorithm," *IEEE Trans. Signal Process*, vol. 39, no. 4, pp. 977–1001, Apr. 1991.

[7] Y. K. Won, R. H. Park, J. H. Park, and B. U. Lee, "Variable LMS algorithm using the time constant concept," *IEEE Trans. Consumer Electron*, vol. 40, no. 3, pp. 655–661, 1994.

[8] F. Riera-Palou, J. M. Norsa, and D. J. M. Cruickshank, "Linear equalizsers with dynamic and automatic length selection," *Electron. Lett.*, vol. 37, no. 25, pp. 1553–1554, Dec. 2001.

[9] 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.

[10] Y. Gong and C. F. N. Cowan, "An LMS style variable tap-length algorithm for structure adaptation," *IEEE Tran. Signal Processing*, vol. 53, no. 7, pp. 2400–2407, 2005.

[11] Y. Zhang, N. Li, J. Chambers, and A. H. Sayed, "Steady state performance analysis of a variable tap-length LMS algorithm," *IEEE Transactions on Signal Processing*, vol. 56, no. 2, pp. 839–845, 2008.

[12] L. Li and J. A. Chambers, "A novel adaptive leakage factor scheme for enhancement of a variable tap-length learning algorithm," *ICASSP 2008*, pp. 2837–2840, Apr. 2008.

[13] A. H. Sayed, *Fundamentals of Adaptive Filtering*, John Wiley & Sons, Inc., Hoboken, NJ, 2003.

[14] L. Li, Y. Zhang, and J. A. Chambers, "Steady-state performance of incremental learning over distributed networks for non-Gaussian data," to appear in *ICSP 2008*, Oct. 2008.