Distributed Estimation Over an Adaptive Incremental Network Based on the Affine Projection Algorithm

Leilei Li, *Member, IEEE*, Jonathon A. Chambers, *Senior Member, IEEE*, Cassio G. Lopes, *Member, IEEE*, and Ali H. Sayed, *Fellow, IEEE*

Abstract—We study the problem of distributed estimation based on the affine projection algorithm (APA), which is developed from Newton's method for minimizing a cost function. The proposed solution is formulated to ameliorate the limited convergence properties of least-mean-square (LMS) type distributed adaptive filters with colored inputs. The analysis of transient and steady-state performances at each individual node within the network is developed by using a weighted spatial-temporal energy conservation relation and confirmed by computer simulations. The simulation results also verify that the proposed algorithm provides not only a faster convergence rate but also an improved steady-state performance as compared to an LMS-based scheme. In addition, the new approach attains an acceptable misadjustment performance with lower computational and memory cost, provided the number of regressor vectors and filter length parameters are appropriately chosen, as compared to a distributed recursive-least-squares (RLS) based method.

Index Terms—Adaptive filters, affine projection algorithm, distributed estimation, energy conversation.

I. INTRODUCTION

IN order to reduce the requirement of a powerful central processor and extensive amount of communications in a traditional centralized solution, a distributed solution is developed relying only on local data exchange and interactions between immediate neighborhood nodes, whilst retaining the estimation accuracy of a centralized solution [1], [2]. Such distributed adaptive networks could therefore find a wide range of potential applications, such as in precision agriculture, environmental monitoring, military surveillance, transportation and factory instrumentation [1]–[3]. One of their particular merits is that a distributed adaptive network has abilities to collaborate and adapt as explained in [9]–[12]. The individual nodes within a distributed network share the computational

burden so that communications are reduced as compared to a centralized network, and power and bandwidth usage are also thereby reduced [1], [4]. In addition, the ability to track not only the variations of the environment but also the topology of the network is also achieved due to their adaptive capability.

Distributed solutions which exploit consensus implementation presented in [6]–[8] require two time scales: during the initial period of time each node makes an individual estimation and then through consensus iterations the nodes combine estimations to reach the desired estimate. This approach relies on particular conditions for the coefficients and network topology. Recent investigations [9]–[12] have therefore developed incremental learning algorithms over a distributed network, which have a cyclic pattern of cooperation with minimum power and communications. In such a network, each node cooperates only with one adjacent node to exploit the spatial dimension, whilst performing local computations in the time dimension. The incremental algorithms thereby approximate the global estimation in a defined cyclic learning framework. In addition, this approach reduces communications between nodes and improves the network autonomy as compared to a centralized solution. In practical wireless sensor networks, it should be highlighted, however, that it may become more difficult to establish a Hamiltonian cycle as required in the incremental mode of cooperation as the number of sensors increases. Moreover, such incremental distributed processing schemes may not scale well for very large networks. In this work, we therefore assume the network size is sufficiently small, typically less than one hundred, so that incremental schemes can be used. In the future, we plan to study adaptive algorithms of the diffusion type, which remove the requirement of a Hamiltonian cycle at the expense of a slightly reduced mean-square performance [24]–[26]. In diffusive networks, the nodes share and process information in real-time and a Hamiltonian cycle is not needed. The incremental approach studied in our paper can be viewed as a reference point against which other algorithms can be measured; this is because incremental approaches provide one of the best performances when cycles are permitted [9]. It is well known that in the case of a single adaptive filter, one major drawback of the LMS algorithm is its slow convergence rate for colored input signals and the APA algorithm is a better alternative to LMS in such an environment [13]–[15]. For distributed networks, highly correlated inputs also deteriorate the performance of the distributed LMS (dLMS) algorithm [5]. In this paper, we therefore focus on a new APA-based distributed learning scheme for such a network which was proposed to obtain a good compromise between convergence performance and computational cost, rather than considering a precise application. The term

"affine projection" represents linear projection type mapping onto the subspace defined by the columns of a low column-rank input matrix. In fact, as in [17] where further details can be found, the precise term would be ϵ-APA, where ϵ represents the weighting on a regularizing identity matrix in the algorithm formulation; however, in this work, as in [17] the ϵ is dropped for notational convenience. A key contribution of this paper is using the weighted spatial-temporal energy conservation relation to evaluate the performance of the resulting ring network of nodes, which incorporates the space-time structure of the data. These theoretical results are found to agree well with the simulation results for both Gaussian and uniform distributed input signals for sufficiently small step-sizes.

The remainder of this paper is organized as follows. In Section II, a distributed estimation problem is defined and the derivation of the proposed APA-based algorithm is given. In Section III, we analyze the performance of the spatial and temporal adaptive algorithm and provide the closed-form expressions for the transient and steady-state performances of the incremental algorithm. Simulation results are compared with the theoretical results in Section IV.

Throughout the paper, the following notations are adopted: boldface small and capital letters are used for random complex vectors or scalars and matrices respectively; normal font is employed for deterministic complex quantities. For ease of reference, the main symbols used in this paper are listed:

$\mathbb{R}^+$	set of positive real numbers;
$(\cdot)^{\mathrm{T}}$	transposition;
$(\cdot)^*$	complex – conjugate transposition;
$|\cdot|^2$	absolute squared operation;
$\|\cdot\|^2$	squared Euclidean norm operation;
$\nabla(\cdot)$	differentiation;
Ea	statistical expectation of a;
$\min(\cdot)$	minimum value of its argument;
$\lambda(\cdot)$	eigenvalues of a matrix;
$\lambda_{\min}(\cdot)$	smallest eigenvalue of a matrix;
$\lambda_{\max}(\cdot)$	largest eigenvalue of a matrix;
$\mathrm{col}\{a\}$	column vector with entries a;
$Tr(A)$	trace of the matrix A;
$A \otimes B$	Kronecker product of two matrices A and B.

II. ESTIMATION PROBLEM

A motivational example for distributed adaptive networks is to examine an application in the context of measuring some quantity such as temperature or humidity across a spatial field. Consider that a network with N sensors is deployed to observe this physical phenomenon and related events in a specified environment. At time i, the sensor at node k collects a measurement $d_k(i)$, where i denotes the discrete time index and k indicates the node index, and assuming an autoregressive (AR) model is adopted to represent these measurements as follows:

$$d_k(i) = \sum_{m=1}^{M} \beta_m d_k(i-m) + v_k(i) \qquad (1)$$

where $v_k(i)$ is additive zero-mean noise and the coefficients $\{\beta_m\}$ are the parameters of the underlying model. If we define the $M \times 1$ parameter vector

$$w^o = \mathrm{col}\{\beta_1, \beta_2, \ldots, \beta_M\}, (M \times 1) \qquad (2)$$

and the $1 \times M$ regression vector

$$u_{k,i} = [d_k(i-1)\, d_k(i-2)\, \ldots\, d_k(i-M)] \qquad (3)$$

then (1) at each node k can be rewritten as an equivalent linear measurement model

$$d_k(i) = u_{k,i} w^o + v_k(i). \qquad (4)$$

The objective becomes to estimate the model parameter vector w^o from the measurements $d_k(i)$ and $u_{k,i}$ over the network and thereby has the form of a system identification problem. Thus, in order to find the $M \times 1$ vector w^o, we formulate the linear space-time least-mean-square estimation problem as

$$\min_w J(w) \quad \text{and} \quad J(w) = E\|\mathbf{d} - \mathbf{U}w\|^2 \qquad (5)$$

where the global desired response vector and input matrix are

$$\mathbf{d} = \mathrm{col}\{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N\}, \quad (N \times 1) \qquad (6)$$

$$\mathbf{U} = \mathrm{col}\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N\}, \quad (N \times M) \qquad (7)$$

where the $\{d_k(i), u_{k,i}\}$ are realizations of $\{\mathbf{d}_k, \mathbf{u}_k\}$. Thus, the optimal minimum mean-square error (MMSE) solution w^o is calculated, for which the normal equations [17] are satisfied,

$$R_{du} = R_u w^o \qquad (8)$$

where $R_u = E\mathbf{U}^*\mathbf{U}$ and $R_{du} = E\mathbf{U}^*\mathbf{d}$.

When a multitude of nodes in the network has access to data, in order to take advantage of node cooperation, we can introduce a distributed network with incremental learning, where at least one cyclic path can be established across the network. In such a network, information should be transferred from one node to its immediate neighborhood node in a cyclic manner to return to the initial node (see Fig. 1). As in [9], the cost function can be decomposed into

$$J(w) = \sum_{k=1}^{N} J_k(w) \quad \text{and} \quad J_k(w) = E|\mathbf{d}_k - \mathbf{u}_k w|^2. \qquad (9)$$

In previous works [9]–[12], the incremental distributed LMS-based and distributed RLS-based schemes have been introduced. These algorithms exploit a cyclic estimation strategy, which at each discrete time entails visiting every node within the whole network only once, i.e., a Hamiltonian cycle. Our contribution aims at improving upon the convergence performance of the dLMS algorithm with colored input signals whilst reducing the complexity of the low communications distributed RLS (Lc-dRLS) algorithm and the distributed RLS (dRLS) algorithm of [11].

Let $\psi_k^{(i)}$ denote the local estimation of the desired optimal weight vector at node k and time instant i and let w_i indicate the

global estimation at time instant i. Consider a Newton's search based approach to solving (9) for incremental learning within a distributed network. The optimal tap weight w^o is estimated via [17]

$$w_i = w_{i-1} - \sum_{k=1}^{N} \mu_k [\epsilon I + \nabla^2 J_k(\psi_{k-1}^{(i)})]^{-1} \nabla [J_k(\psi_{k-1}^{(i)})]^*$$

$$= w_{i-1} + \sum_{k=1}^{N} \mu_k (\epsilon I + R_{u,k})^{-1} [R_{du,k} - R_{u,k}\psi_{k-1}^{(i)}] \quad (10)$$

where $R_{u,k} = E\mathbf{u}_k^*\mathbf{u}_k$, $R_{du,k} = E\mathbf{d}_k\mathbf{u}_k^*$, ϵ denotes a regularization parameter with small positive value, μ_k indicates an appropriately chosen step-size, which is evaluated in Section III-E, and the scheme is initialized with an $M \times 1$ vector $w_{-1} = \text{col}\{0, \ldots, 0\}$.

For a practical scheme to realize (10), and utilizing the correlation of the input signal at each node, we replace $\{R_{u,k}, R_{du,k}\}$ by the following sample sliding-window estimates:

$$\hat{R}_{u,k} = \frac{1}{T} \sum_{j=i-T+1}^{i} u_{k,j}^* u_{k,j} \quad (11)$$

$$\hat{R}_{du,k} = \frac{1}{T} \sum_{j=i-T+1}^{i} d_k(j) u_{k,j}^* \quad (12)$$

with T equal to the number of recent regressors of each node whilst $u_{k,j}$ and $d_k(j)$ denote the corresponding input vector and desired response at instant time j for the kth node. Hence, using the matrix inversion formula, recursion (10) becomes,

$$w_i = w_{i-1} + \sum_{k=1}^{N} \mu_k U_{k,i}^* (\epsilon I + U_{k,i} U_{k,i}^*)^{-1} [d_{k,i} - U_{k,i}\psi_{k-1}^{(i)}] \quad (13)$$

where the local $T \times M$ block data matrix and $T \times 1$ data vector are

$$U_{k,i} = \begin{pmatrix} u_{k,i} \\ u_{k,i-1} \\ \vdots \\ u_{k,i-T+1} \end{pmatrix}, \ d_{k,i} = \begin{pmatrix} d_k(i) \\ d_k(i-1) \\ \vdots \\ d_k(i-T+1) \end{pmatrix} \quad (14)$$

and ϵ is employed to avoid the inversion of a rank deficient matrix $U_{k,i}U_{k,i}^*$. As such, recursion (13) is the distributed APA (dAPA) learning algorithm in an incremental network, the operation of which is shown in Fig. 1. At each time instant i, each node utilizes local data $\{d_{k,i}, U_{k,i}\}$ and $\psi_{k-1}^{(i)}$ received from its previous node $k-1$ in the cycle to update the local estimation. At the end of the cycle, the local estimation $\psi_N^{(i)}$ is employed as the global estimation w_i and the initial local estimation $\psi_1^{(i+1)}$ for the next discrete time instant $i+1$. The final weight vector shown at the bottom of Fig. 1 can either be used to generate a filter output vector term of the form $u_{k,i}\psi_k^{(i)}$ or the vector $\psi_k^{(i)}$ itself can then be used for system identification or equalization. The pseudo-code implementation of dAPA is described in Table I. In addition, dAPA has intermediate computational and memory cost between dLMS and dRLS, for certain regressor length, which is verified in the Appendix.
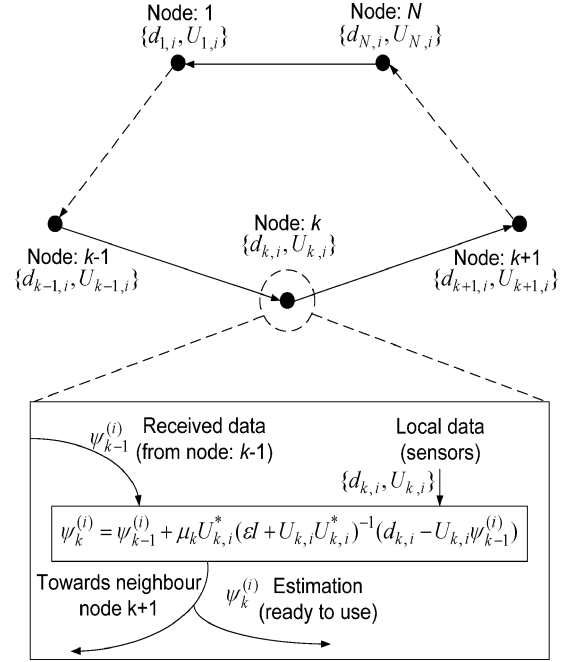

Fig. 1. Data processing of the dAPA algorithm in an incremental network.

TABLE I
PSEUDO-CODE IMPLEMENTATION OF DAPA

For each time instant $i \geq 0$ repeat:
k=1,... N
$\quad \psi_0^{(i)} = w_{i-1}$
$\quad \psi_k^{(i)} = \psi_{k-1}^{(i)} + \mu_k U_{k,i}^* (\epsilon I + U_{k,i} U_{k,i}^*)^{-1} [d_{k,i} - U_{k,i}\psi_{k-1}^{(i)}]$
end
$w_i = \psi_N^{(i)}$
where $\{d_{k,i}, U_{k,i}\}$ are defined by (10).

III. PERFORMANCE ANALYSIS

The convergence behaviors of classical APA-based algorithms are studied in [15]–[19], exploiting arguments based on a single adaptive filter. In order to study the performance of the dAPA algorithm, we extend the weighted energy conservation approach for the APA-based algorithms of [18], [19] to the case of a distributed incremental network, which involves both the space dimension and the time dimension. However, due to the energy flow across the interconnected filter, some of the simplifications for a single filter case cannot be adopted. A set of weighting matrices is particularly chosen to decouple a set of equations and we evaluate the transient and steady-state performances at each individual node in terms of mean-square deviation (MSD), excess mean-square-error (EMSE) and mean-square error (MSE). The closed-form expressions for the theoretical results are formed under some simplifying assumptions described below.

A. Data Model and Assumption

As defined earlier, we use boldface letters as the random quantities and assume the same model as in [9] to carry out the performance analysis:

A1) The relation between the unknown system vector w^o and $\{\mathbf{d}_k(i), \mathbf{u}_{k,i}\}$ takes the form:

$$\mathbf{d}_k(i) = \mathbf{u}_{k,i} w^o + \mathbf{v}_k(i) \quad (15)$$

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

A2) $\mathbf{u}_{k,i}$ is spatially independent 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 $i \neq j$.

We highlight that assumption A2) is an extension of that made for time-only adaptive filtering to space-time adaptive filtering. We emphasize that the spatial independence assumption is generally more likely to hold than the temporal independence assumption in the context of a distributed network, as a consequence of the different locations of the nodes. The temporal independence assumption is necessary for analysis purposes and is commonly used to provide useful performance measures in adaptive signal processing [27]. In terms of analysis in this paper, we study only the stationary case due to space limitations, i.e., where the system vector is fixed and statistics of the various input and noise signals are time-invariant, but the following analysis could be extended to a non-stationary model, such as the random-walk model (see [20]–[22]).

B. Weighted Spatial-Temporal Energy Conservation Relation

Using the following error vectors:

$$\tilde{\boldsymbol{\psi}}_k^{(i)} \triangleq w^o - \boldsymbol{\psi}_k^{(i)} \quad \text{(weight-error vector)} \tag{16}$$

$$\mathbf{e}_{k,i} \triangleq \mathbf{d}_{k,i} - \mathbf{U}_{k,i}\boldsymbol{\psi}_{k-1}^{(i)} \quad \text{(error ,vector)} \tag{17}$$

to represent the update tap weights expression in dAPA, we obtain

$$\tilde{\boldsymbol{\psi}}_k^{(i)} = \tilde{\boldsymbol{\psi}}_{k-1}^{(i)} - \mu_k \mathbf{U}_{k,i}^*(\epsilon I + \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*)^{-1}\mathbf{e}_{k,i}. \tag{18}$$

Multiplying both sides of (18) by $\mathbf{U}_{k,i}$ from the left, (18) becomes

$$\mathbf{U}_{k,i}\tilde{\boldsymbol{\psi}}_k^{(i)} = \mathbf{U}_{k,i}\tilde{\boldsymbol{\psi}}_{k-1}^{(i)} - \mu_k \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*(\epsilon I + \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*)^{-1}\mathbf{e}_{k,i}. \tag{19}$$

The *a posteriori* and *a priori* error vectors $\{\mathbf{e}_{p,k}, \mathbf{e}_{a,k}\}$ are introduced

$$\mathbf{e}_{p,k}^{(i)} \triangleq \mathbf{U}_{k,i}\tilde{\boldsymbol{\psi}}_k^{(i)} \quad \text{and} \quad \mathbf{e}_{a,k}^{(i)} \triangleq \mathbf{U}_{k,i}\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}.$$

As a result, we have

$$\mathbf{e}_{p,k}^{(i)} = \mathbf{e}_{a,k}^{(i)} - \mu_k \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*(\epsilon I + \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*)^{-1}\mathbf{e}_{k,i}. \tag{20}$$

Note that the error vector is given by

$$\mathbf{e}_{k,i} = \mathbf{U}_{k,i}\tilde{\boldsymbol{\psi}}_{k-1}^{(i)} + \mathbf{v}_{k,i} = \mathbf{e}_{a,k}^{(i)} + \mathbf{v}_{k,i} \tag{21}$$

where

$$\mathbf{v}_{k,i} = \text{col}\{\mathbf{v}_k(i), \mathbf{v}_k(i-1), \ldots, \mathbf{v}_k(i-T+1)\}. \tag{22}$$

We are interested in evaluating the following performance measure at each node k:

$$\eta_k \triangleq E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_I^2 \quad \text{(MSD)} \tag{23}$$

$$\zeta_k \triangleq E|\mathbf{u}_k\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}|^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{R_{u,k}}^2 \quad \text{(EMSE)} \tag{24}$$

$$\xi_k \triangleq \zeta_k + \sigma_{v,k}^2 \quad \text{(MSE)} \tag{25}$$

where under the assumed data conditions we introduce the weighted norm notation $\|x\|_\Sigma^2 \triangleq x^*\Sigma x$ with a vector x and a Hermitian positive definite matrix $\Sigma > 0$. In order to study the performance behavior of the dAPA algorithm for incremental networks, the method of weighted energy conservation described in [9], [17] is used in this paper. As a consequence, we firstly define the weighted *a posteriori* and *a priori* error vectors at node k,

$$\mathbf{e}_{p,k}^{\Sigma_k,(i)} \triangleq \mathbf{U}_{k,i}\Sigma_k\tilde{\boldsymbol{\psi}}_k^{(i)} \tag{26}$$

$$\mathbf{e}_{a,k}^{\Sigma_k,(i)} \triangleq \mathbf{U}_{k,i}\Sigma_k\tilde{\boldsymbol{\psi}}_{k-1}^{(i)} \tag{27}$$

where Σ_k is a Hermitian positive-definite weighting matrix and free to choose for each node k. Using the weighted definitions (26) and (27), we can expand (20) in terms of weighted error vectors and the regressor data as follows:

$$\mathbf{e}_{p,k}^{\Sigma_k,(i)} = \mathbf{e}_{a,k}^{\Sigma_k,(i)} - \mu_k \mathbf{U}_{k,i}\Sigma_k\mathbf{U}_{k,i}^*(\epsilon I + \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*)^{-1}\mathbf{e}_{k,i}. \tag{28}$$

If we choose the special case $\Sigma_k = I$, (28) is simplified to (20). With the assumption that $\mathbf{U}_{k,i}\Sigma_k\mathbf{U}_{k,i}^*$ is invertible, (28) can be used to replace $\mathbf{e}_{k,i}$ in (18). After rearrangement, we obtain

$$\tilde{\boldsymbol{\psi}}_k^{(i)} + \mathbf{U}_{k,i}^*(\mathbf{U}_{k,i}\Sigma_k\mathbf{U}_{k,i}^*)^{-1}\mathbf{e}_{a,k}^{\Sigma_k,(i)}$$
$$= \tilde{\boldsymbol{\psi}}_{k-1}^{(i)} + \mathbf{U}_{k,i}^*(\mathbf{U}_{k,i}\Sigma_k\mathbf{U}_{k,i}^*)^{-1}\mathbf{e}_{p,k}^{\Sigma_k,(i)}. \tag{29}$$

If we equate the weighted energies of both sides of (29), we can establish the space-time version of the weighted energy-conservation relation for dAPA:

$$\|\tilde{\boldsymbol{\psi}}_k^{(i)}\|_{\Sigma_k}^2 + \mathbf{e}_{a,k}^{\Sigma_k,(i)*}(\mathbf{U}_{k,i}\Sigma_k\mathbf{U}_{k,i}^*)^{-1}\mathbf{e}_{a,k}^{\Sigma_k,(i)}$$
$$= \|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{\Sigma_k}^2 + \mathbf{e}_{p,k}^{\Sigma_k,(i)*}(\mathbf{U}_{k,i}\Sigma_k\mathbf{U}_{k,i}^*)^{-1}\mathbf{e}_{p,k}^{\Sigma_k,(i)}. \tag{30}$$

Then, substituting (28) into (30) and rearranging the result, we obtain

$$\|\tilde{\boldsymbol{\psi}}_k^{(i)}\|_{\Sigma_k}^2 = \|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{\Sigma_k}^2 - \mu_k[\mathbf{e}_{a,k}^{\Sigma_k,(i)*}\mathbf{A}_k\mathbf{e}_{k,i}]$$
$$- \mu_k[\mathbf{e}_{k,i}^*\mathbf{A}_k\mathbf{e}_{a,k}^{\Sigma_k,(i)*}] + \mu_k^2[\mathbf{e}_{k,i}^*\mathbf{B}_k\mathbf{e}_{k,i}] \tag{31}$$

with

$$\mathbf{A}_k = (\epsilon I + \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*)^{-1} \tag{32}$$

$$\mathbf{B}_k^{\Sigma_k} = (\epsilon I + \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*)^{-1}\mathbf{U}_{k,i}\Sigma_k\mathbf{U}_{k,i}^*(\epsilon I + \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*)^{-1}. \tag{33}$$

By using $\mathbf{e}_{k,i} = \mathbf{U}_{k,i}\tilde{\boldsymbol{\psi}}_{k-1}^{(i)} + \mathbf{v}_{k,i}$, taking expectations of both sides and ignoring dependence between $\mathbf{v}_{k,i}$ and $\{\mathbf{e}_{a,k}^{(i)}, \mathbf{e}_{a,k}^{\Sigma_k,(i)}\}$ due to assumption A1), expression (31) becomes

$$E\|\tilde{\boldsymbol{\psi}}_k^{(i)}\|_{\Sigma_k}^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{\Sigma_k'}^2 + \mu_k^2 E[\mathbf{v}_{k,i}^*\mathbf{B}_k^{\Sigma_k}\mathbf{v}_{k,i}] \tag{34}$$

where Σ_k' is a stochastic weighting matrix

$$\Sigma_k' = \Sigma_k - \mu_k\Sigma_k\mathbf{C}_k - \mu_k\mathbf{C}_k\Sigma_k + \mu_k^2\mathbf{D}_k^{\Sigma_k} \tag{35}$$

with

$$\mathbf{C}_k = \mathbf{U}_{k,i}^*\mathbf{A}_k\mathbf{U}_{k,i} \quad \text{and} \quad \mathbf{D}_k^{\Sigma_k} = \mathbf{U}_{k,i}^*\mathbf{B}_k^{\Sigma_k}\mathbf{U}_{k,i}. \tag{36}$$

Equation (34) is difficult to evaluate due to the dependence of $\boldsymbol{\psi}_{k-1}^{(i)}$ on previous regressors of $\mathbf{U}_{k,i}$ and Σ_k' on $\mathbf{U}_{k,i}$. In order to resolve this problem in terms of analysis, we introduce an independence assumption on the regressor sequence $\mathbf{U}_{k,i}$, namely:

A3) The matrix sequence $\mathbf{U}_{k,i}$ is independent of $\mathbf{U}_{k,i-1}$. which guarantees that $\boldsymbol{\psi}_{k-1}^{(i)}$ is independent of both $\mathbf{U}_{k,i}$ and Σ_k'. As compared to A2), A3) is a strong assumption. However, we can introduce a weaker assumption:

A3') $\boldsymbol{\psi}_{k-1}^{(i)}$ is independent of $\mathbf{U}_{k,i}^*(\epsilon I + \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*)^{-1}\mathbf{U}_{k,i}$ which is derived from (35) for Σ_k' to satisfy our purpose. We highlight that these assumptions violate the dAPA updating recursions, and hence they are not entirely realistic but they are necessary to facilitate analysis similar to those in [18], [19] used for the original non-distributed APA algorithm. When the step-size is sufficiently small, we can find a good match between theoretical and practical results, which is verified by simulations. For compactness of notation, the index i is dropped. Using this assumption, we can separate the expectation $E\|\tilde{\boldsymbol{\psi}}_{k-1}\|_{\Sigma_k'}^2$ into

$$E\|\tilde{\boldsymbol{\psi}}_{k-1}\|_{\Sigma_k'}^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}\|_{E\Sigma_k'}^2 \tag{37}$$

where the mean of the weighted matrix Σ_k' is given by $E\Sigma_k' = \Sigma_k'$:

$$\Sigma_k' = \Sigma_k - \mu_k \Sigma_k E\mathbf{C}_k - \mu_k E\mathbf{C}_k \cdot \Sigma_k + \mu_k^2 E\mathbf{D}_k^{\Sigma_k} \tag{38}$$

where Σ_k' is now a deterministic matrix. In this manner, expression (34) is replaced by

$$E\|\tilde{\boldsymbol{\psi}}_k\|_{\Sigma_k}^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}\|_{\Sigma_k'}^2 + \mu_k^2 E[\mathbf{v}_k^*\mathbf{B}_k^{\Sigma_k}\mathbf{v}_k]. \tag{39}$$

For studying the behavior of the distributed learning algorithm, we need to evaluate the following three moments:

$$E\mathbf{C}_k = E[\mathbf{U}_k^*\mathbf{A}_k\mathbf{U}_k] \tag{40}$$

$$E\mathbf{D}_k^{\Sigma_k} = E[\mathbf{U}_k^*\mathbf{A}_k\mathbf{U}_k\Sigma_k\mathbf{U}_k^*\mathbf{A}_k\mathbf{U}_k] \tag{41}$$

$$E[\mathbf{v}_k^*\mathbf{B}_k^{\Sigma_k}\mathbf{v}_k] = E[\mathbf{v}_k^*\mathbf{A}_k\mathbf{U}_k\Sigma_k\mathbf{U}_k^*\mathbf{A}_k\mathbf{v}_k]. \tag{42}$$

The terms in (41) and (42) are difficult to calculate, even the eigen-decomposition and diagonalization methods used for Gaussian data in [9] are not available to express (38) and (39) in a compact manner and thereby closed-forms of the mean-square quantities can not be obtained. To proceed, we need to extract Σ_k from the right-hand side expressions (41) and (42). This is achieved by vectorization and exploiting the property of Kronecker products.

C. Weighted Variance Relation

In order to evaluate $E\|\tilde{\boldsymbol{\psi}}_k\|_{\Sigma_k}^2$, we now introduce $M^2 \times 1$ column vectors as in [17]:

$$\sigma_k = \text{vec}\{\Sigma_k\}, \quad \sigma_k' = \text{vec}\{\Sigma_k'\} \tag{43}$$

where the $\text{vec}\{\cdot\}$ notation is used in two ways: $\sigma = \text{vec}\{\Sigma\}$ is 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 $\Sigma = \text{vec}\{\sigma\}$, as will be used below, is a matrix whose entries are recovered from σ. One useful property for

the $\text{vec}\{\cdot\}$ notation when working with Kronecker products is the following. For any matrices $\{P, \Sigma, Q\}$ of compatible dimensions, it holds

$$\text{vec}\{P\Sigma Q\} = (Q^{\mathrm{T}} \otimes P)\text{vec}\{\Sigma\}. \tag{44}$$

By applying (44) to express some items in (38), we find that

$$\text{vec}\{\Sigma_k E\mathbf{C}_k\} = (E\mathbf{C}_k^{\mathrm{T}} \otimes I)\sigma_k \tag{45}$$

$$\text{vec}\{E\mathbf{C}_k \cdot \Sigma_k\} = (I \otimes E\mathbf{C}_k)\sigma_k \tag{46}$$

$$\text{vec}\{E\mathbf{D}_k^{\Sigma_k}\} = E[\mathbf{C}_k^{\mathrm{T}} \otimes \mathbf{C}_k]\sigma_k. \tag{47}$$

Therefore, using (44) in (38), we obtain a linear relation between the corresponding vectors $\{\sigma_k', \sigma_k\}$, namely

$$\sigma_k' = \mathrm{F}_k \sigma_k \tag{48}$$

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

$$\mathrm{F}_k \triangleq I - \mu_k(E\mathbf{C}_k^{\mathrm{T}} \otimes I) - \mu_k(I \otimes E\mathbf{C}_k) + \mu_k^2 E[\mathbf{C}_k^{\mathrm{T}} \otimes \mathbf{C}_k]. \tag{49}$$

As a result, expression (39) becomes

$$E\|\tilde{\boldsymbol{\psi}}_k\|_{\text{vec}\{\sigma_k\}}^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}\|_{\text{vec}\{\mathrm{F}_k\sigma_k\}}^2 + \mu_k^2 E[\mathbf{v}_k^*\mathbf{B}_k^{\Sigma_k}\mathbf{v}_k]. \tag{50}$$

For the sake of clarity, we reintroduce the time index i but drop the $\text{vec}\{\cdot\}$ notation from the subscripts in (50) for compactness. Expression (50) becomes

$$E\|\tilde{\boldsymbol{\psi}}_k^{(i)}\|_{\sigma_k}^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{\mathrm{F}_k\sigma_k}^2 + \mu_k^2 E[\mathbf{v}_{k,i}^*\mathbf{B}_k^{\Sigma_k}\mathbf{v}_{k,i}]. \tag{51}$$

Due to the assumption that $\mathbf{v}_{k,i}$ is independent from $\mathbf{U}_{k,i}$, the last item (51) can be written as

$$\mu_k^2 E[\mathbf{v}_{k,i}^*\mathbf{B}_k^{\Sigma_k}\mathbf{v}_{k,i}] = \mu_k^2 \delta_{v,k}^2 Tr\{E\boldsymbol{\Phi}_k \cdot \Sigma_k\} = \mu_k^2 \delta_{v,k}^2 \gamma_k^{\mathrm{T}}\sigma_k \tag{52}$$

with $\boldsymbol{\Phi}_k = \mathbf{U}_{k,i}^*(\epsilon I + \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*)^{-2}\mathbf{U}_{k,i}$ and $\gamma_k = \text{vec}\{E\boldsymbol{\Phi}_k\}$. Therefore, expression (51) can be written in a compact manner as

$$E\|\tilde{\boldsymbol{\psi}}_k^{(i)}\|_{\sigma_k}^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{\mathrm{F}_k\sigma_k}^2 + \mu_k^2 \delta_{v,k}^2 \gamma_k^{\mathrm{T}}\sigma_k. \tag{53}$$

D. Learning Curves

Expression (53) involves spatially local information from two nodes, namely, $\tilde{\boldsymbol{\psi}}_k^{(i)}$ and $\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}$. The ring topology with the weighting matrices enables us to resolve this problem. By iterating (53), N coupled equalities are obtained:

$$E\|\tilde{\boldsymbol{\psi}}_1^{(i)}\|_{\sigma_1}^2 = E\|\tilde{\boldsymbol{\psi}}_N^{(i-1)}\|_{\mathrm{F}_1\sigma_1}^2 + g_1\sigma_1$$

$$E\|\tilde{\boldsymbol{\psi}}_2^{(i)}\|_{\sigma_2}^2 = E\|\tilde{\boldsymbol{\psi}}_1^{(i)}\|_{\mathrm{F}_2\sigma_2}^2 + g_2\sigma_2$$

$$\vdots$$

$$E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{\sigma_{k-1}}^2 = E\|\tilde{\boldsymbol{\psi}}_{k-2}^{(i)}\|_{\mathrm{F}_{k-1}\sigma_{k-1}}^2 + g_{k-1}\sigma_{k-1} \tag{54}$$

$$E\|\tilde{\boldsymbol{\psi}}_k^{(i)}\|_{\sigma_k}^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{\mathrm{F}_k\sigma_k}^2 + g_k\sigma_k$$

$$\vdots$$

$$E\|\tilde{\boldsymbol{\psi}}_N^{(i)}\|_{\sigma_N}^2 = E\|\tilde{\boldsymbol{\psi}}_{N-1}^{(i)}\|_{\mathrm{F}_N\sigma_N}^2 + g_N\sigma_N \tag{55}$$

with $g_k = \mu_k^2 \delta_{v,k}^2 \gamma_k^{\mathrm{T}}$. In order to describe the energy flow through the nodes it is necessary to connect the free parameters σ_k and σ_{k-1}. Following the approach in [9], we adopt a linear relation as $\sigma_{k-1} = \mathrm{F}_k \sigma_k$ and thereby combine (54) and (55) to obtain

$$E\|\tilde{\boldsymbol{\psi}}_k^{(i)}\|_{\sigma_k}^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{\sigma_{k-1}}^2 + g_k \sigma_k$$
$$= E\|\tilde{\boldsymbol{\psi}}_{k-2}^{(i)}\|_{\mathrm{F}_{k-1}\mathrm{F}_k\sigma_k}^2 + g_{k-1}\mathrm{F}_k\sigma_k + g_k\sigma_k \quad (56)$$

where F_k is as in (49) and includes statistics of local data. The matrix F_k turns out to determine the dynamics of propagation of the energy through the network. Conditions to ensure stability and convergence end up depending on the matrix F_k. A detailed discussion on such energy relations appears in [27]. Iterating in this way, we can obtain an equality involving only $\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}$ and $\tilde{\boldsymbol{\psi}}_{k-1}^{(i-1)}$, namely

$$E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{\sigma_{k-1}}^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i-1)}\|_{\mathrm{F}_k\cdots\mathrm{F}_N\mathrm{F}_1\cdots\mathrm{F}_{k-1}\sigma_{k-1}}^2$$
$$+ g_k \mathrm{F}_{k+1}\cdots\mathrm{F}_N\mathrm{F}_1\cdots\mathrm{F}_{k-1}\sigma_{k-1}$$
$$+ g_{k+1}\mathrm{F}_{k+2}\cdots\mathrm{F}_N\mathrm{F}_1\cdots\mathrm{F}_{k-1}\sigma_{k-1}.$$
$$\cdots + g_{k-2}\mathrm{F}_{k-1}\sigma_{k-1} + g_{k-1}\sigma_{k-1}. \quad (57)$$

We should note that the *a posteriori* and *a priori* error vectors $\{\mathbf{e}_{p,k}^{(i)}, \mathbf{e}_{a,k}^{(i)}\}$ have spatial connotation, which is different from the traditional terminology as in [17]. By choosing $\sigma_{k-1} = q = \mathrm{vec}(I)$, we formulate the closed-form expression for the MSD learning curve at node k:

$$E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_q^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(-1)}\|_{(\Pi_{k-1,1})^i q}^2$$
$$+ a_{k-1}(I + \cdots + (\Pi_{k-1,1})^{i-1})q \quad (58)$$

where the product of F_k matrices for each node, $\Pi_{k-1,l}$ and the $1 \times M^2$ row vector a_{k-1} are defined by

$$\Pi_{k-1,l} = \mathrm{F}_{k+l-1}\mathrm{F}_{k+l}\cdots\mathrm{F}_N\mathrm{F}_1\cdots\mathrm{F}_{k-1}, \, l = 1, 2, \ldots, N \quad (59)$$
and
$$a_{k-1} = g_k\Pi_{k-1,2} + g_{k+1}\Pi_{k-1,3}\cdots + g_{k-2}\Pi_{k-1,N} + g_{k-1}. \quad (60)$$

Therefore, the theoretical transient performance of the MSD of node k is formulated as

$$\boxed{E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_q^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(-1)}\|_{f_{i-1}}^2 + a_{k-1}\beta_{i-1}} \quad (61)$$

where the vectors f_{i-1} and β_{i-1} are given respectively by

$$f_{i-1} = \Pi_{k-1,1}f_{i-2}, \quad f_{-1} = q, \quad \text{and}$$
$$\beta_{i-1} = \beta_{i-2} + f_{i-1}, \quad \beta_{-1} = M^2 \times 1 \text{ null vector.}$$

Let $E|\mathbf{e}_{a,k}(i)|^2$ and $E|\mathbf{e}_k(i)|^2$ respectively denote the EMSE and MSE learning curves of the set of adaptive filters, in terms of time iterations, indexed for $k = 1, 2, \ldots, N$. Under the assumption that $\mathbf{u}_{k,i}$ is i.i.d., we obtain

$$E|\mathbf{e}_{a,k}(i)|^2 = E|\mathbf{u}_{k,i}\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}|^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{R_{u,k}}^2 \quad (62)$$

$$E|\mathbf{e}_k(i)|^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{R_{u,k}}^2 + \delta_{v,k}^2 \quad (63)$$

which are used to evaluate the learning curves. Thus, the selection of $\sigma_{k-1} = r_k = \mathrm{vec}(R_{u,k})$ leads to the closed-form expressions for EMSE and MSE learning curves at node k:

$$E|\mathbf{e}_{a,k}(i)|^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(-1)}\|_{f'_{i-1}}^2 + a_{k-1}\beta'_{i-1} \quad (64)$$

$$E|\mathbf{e}_k(i)|^2 = E|\mathbf{e}_{a,k}(i)|^2 + \delta_{v,k}^2 \quad (65)$$

where the vectors f'_{i-1} and β'_{i-1} are formulated by

$$f'_{i-1} = \Pi_{k-1,1}f'_{i-2}, \quad f'_{-1} = r_k \quad \text{and}$$
$$\beta'_{i-1} = \beta'_{i-2} + f'_{i-1}, \quad \beta'_{-1} = M^2 \times 1 \text{ null vector.}$$

E. Mean and Mean-Square Stability

Substitute $\mathbf{e}_{k,i}$ in (18) by (21), we obtain the relationship expression between $\tilde{\boldsymbol{\psi}}_k^{(i)}$ and $\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}$ as

$$\tilde{\boldsymbol{\psi}}_k^{(i)} = (I - \mu_k\mathbf{U}_{k,i}^*(\epsilon I + \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*)^{-1}\mathbf{U}_{k,i})\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}$$
$$- \mu_k\mathbf{U}_{k,i}^*(\epsilon I + \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*)^{-1}\mathbf{v}_{k,i}. \quad (66)$$

Then, taking expectation of both sides, we have the corresponding result for the evolution of the mean of the weight-error vector:

$$\boxed{E\tilde{\boldsymbol{\psi}}_k^{(i)} = (I - \mu_k E\mathbf{C}_k)E\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}}. \quad (67)$$

To guarantee convergence in the mean of dAPA, the step-size μ_k should satisfy the condition

$$\mu_k < \frac{2}{\lambda_{\max}(E\mathbf{C}_k)}, \quad k = 1, \ldots N. \quad (68)$$

Note, it is not necessary for there to be an infinite number of nodes for this to hold, as through the incremental learning the repeated visits of the nodes at each discrete time i will achieve convergence with only a finite number of nodes. In the same way, dAPA will be said to be mean-square stable if the eigenvalues of F_k from (53) satisfy $-1 < \lambda(\mathrm{F}_k) < 1$ as in [17]. Therefore, we express the matrix F_k in (49) as

$$\mathrm{F}_k = I - \mu_k\mathrm{X}_k + \mu_k^2\mathrm{Y}_k, \quad k = 1, \ldots N \quad (69)$$

with $\mathrm{X}_k = E\mathbf{C}_k^{\mathrm{T}} \otimes I + I \otimes E\mathbf{C}_k$ and $\mathrm{Y}_k = E[\mathbf{C}_k^{\mathrm{T}} \otimes \mathbf{C}_k]$. Exploiting the approach as in [17], the convergence of the mean-square error of dAPA will be achieved for values of μ_k in the range

$$\boxed{0 < \mu_k < \min\left\{\frac{1}{\lambda_{\max}(\mathrm{X}_k^{-1}\mathrm{Y}_k)}, \frac{1}{\lambda_{\max}(\mathrm{H}_k)\in\mathbb{R}^+}\right\}} \quad (70)$$

where the second condition is the reciprocal value of the largest positive eigenvalue of

$$\mathrm{H}_k = \begin{pmatrix} \frac{1}{2}\mathrm{X}_k & -\frac{1}{2}\mathrm{Y}_k \\ I & 0 \end{pmatrix}. \quad (71)$$

F. Steady-State Behavior

In the above, the variance relation (53) is used to evaluate the transient behavior of dAPA. The same variance relation can be also used to characterize the steady-state mean-square performance of dAPA. For the convenience of studying the steady-state performance of dAPA, letting $\mathbf{p}_k = \tilde{\boldsymbol{\psi}}_k^{(\infty)}$, we can rewrite (53) as

$$\boxed{E\|\mathbf{p}_k\|_{\sigma_k}^2 = E\|\mathbf{p}_{k-1}\|_{\mathrm{F}_k \sigma_k}^2 + g_k \sigma_k} \qquad (72)$$

where the step-size μ_k of $g_k = \mu_k^2 \delta_{v,k}^2 r_k^{\mathrm{T}}$ is chosen to guarantee stability of the filter. Iterating (72) for an incremental network and choosing the proper weighting vectors σ_k for $k = 1, 2, \ldots, N$, we obtain an expression only containing $\mathbf{p}_{k-1}$, given by

$$\begin{aligned} E\|\mathbf{p}_{k-1}\|_{\sigma_{k-1}}^2 = {} & E\|\mathbf{p}_{k-1}\|_{\mathrm{F}_k \cdots \mathrm{F}_N \mathrm{F}_1 \cdots \mathrm{F}_{k-1} \sigma_{k-1}}^2 \\ & + g_k \mathrm{F}_{k+1} \cdots \mathrm{F}_N \mathrm{F}_1 \cdots \mathrm{F}_{k-1} \sigma_{k-1} \\ & + g_{k+1} \mathrm{F}_{k+2} \cdots \mathrm{F}_N \mathrm{F}_1 \cdots \mathrm{F}_{k-1} \sigma_{k-1} \\ & \cdots + g_{k-2} \mathrm{F}_{k-1} \sigma_{k-1} + g_{k-1} \sigma_{k-1}. \end{aligned} \qquad (73)$$

Using (59) and (60) to simplify this expression, we get

$$E\|\mathbf{p}_{k-1}\|_{(I - \Pi_{k-1,1})\sigma_{k-1}}^2 = a_{k-1} \sigma_{k-1}. \qquad (74)$$

We can use expression (74) to evaluate the performance measures at node k, as follows:

$$\begin{aligned} \eta_k &= E\|\mathbf{p}_{k-1}\|_q^2, & q &= \mathrm{vec}\{I\} & \text{(MSD)} & \quad (75) \\ \zeta_k &= E\|\mathbf{p}_{k-1}\|_{r_k}^2, & r_k &= \mathrm{vec}\{R_{u,k}\} & \text{(EMSE)} & \quad (76) \\ \xi_k &= \zeta_k + \delta_{v,k}^2 & & & \text{(MSE)}. & \quad (77) \end{aligned}$$

Since we are free to choose the weighting vector σ_{k-1}, we can exploit it to calculate the steady-state performance of each node. When we select the weighting vector σ_{k-1} as the solution of the linear equation $(I - \Pi_{k-1,1})\sigma_{k-1} = q$ or $(I - \Pi_{k-1,1})\sigma_{k-1} = r_k$, the desired MSD, EMSE and MSE at node k are obtained as

$$\begin{aligned} \eta_k &= a_{k-1}(I - \Pi_{k-1,1})^{-1} q & \text{(MSD)} & \quad (78) \\ \zeta_k &= a_{k-1}(I - \Pi_{k-1,1})^{-1} r_k & \text{(EMSE)} & \quad (79) \\ \xi_k &= \zeta_k + \delta_{v,k}^2 & \text{(MSE)}. & \quad (80) \end{aligned}$$

The matrix $\Pi_{k-1,l}$ can be regarded as the transition matrix for the weighting vector σ_{k-1} and the vector a_{k-1} can be interpreted as the effect of combining the transformed noise and local data statistics from all the nodes in the ring topology.

The distributed normalized LMS (dNLMS) algorithm can be regarded as a special case for dAPA with the number of recent regressors $T = 1$. Since the autocorrelation matrix $R_{u,k}$ of the input at each node is a Hermitian matrix, we can exploit unitary G_k and a diagonal matrix Λ_k with the eigenvalues of $R_{u,k}$ to decompose the autocorrelation matrix into $R_{u,k} = G_k \Lambda_k G_k^*$. Using the following transformed quantities

$$\bar{\mathbf{p}}_k \triangleq G_k^* \mathbf{p}_k, \quad \bar{\mathbf{u}}_k \triangleq \mathbf{u}_k G_k \bar{\Sigma}_k \triangleq G_k^* \Sigma_k G_k, \quad \bar{\Sigma}_k' \triangleq G_k^* \Sigma_k' G_k$$

we can obtain that $\|\mathbf{p}_{k-1}\|_{\Sigma_k}^2 = \|\bar{\mathbf{p}}_{k-1}\|_{\bar{\Sigma}_k}^2$ and $\|\mathbf{u}_k\|_{\Sigma_k}^2 = \|\bar{\mathbf{u}}_k\|_{\bar{\Sigma}_k}^2$. In addition, at the steady-state stage, we can rewrite (40) for dNLMS as

$$E\bar{\mathbf{C}}_k \approx \frac{\Lambda_k}{\epsilon + Tr(R_{u,k})} \qquad (81)$$

which is a diagonal matrix. For a small step-size, $\bar{\mathrm{F}}_k \approx I - \mu_k(E\bar{\mathbf{C}}_k \otimes I) - \mu_k(I \otimes E\bar{\mathbf{C}}_k)$, namely, $\bar{\mathrm{F}}_k$ also becomes a diagonal matrix. Therefore, matrix $\Pi_{k-1,1} = \Pi = \mathrm{F}_1 \mathrm{F}_2 \cdots \mathrm{F}_N$ will be diagonal as well and $a_{k-1} \approx \sum_{k=1}^N g_k$. Using $E\|\mathbf{p}_{k-1}\|^2 = E\|\bar{\mathbf{p}}_{k-1}\|^2$, we can rewrite (78) as

$$\eta_k \approx \sum_{k=1}^N g_k(I - \Pi)^{-1} q \qquad (82)$$

which clearly explains why there is an equalization effect on the MSD of dNLMS throughout the network, and is verified in Figs. 7 and 10 even for a large step-size. We now proceed to confirm these values through Monte Carlo simulations.

IV. SIMULATIONS

The above analysis is based on the independence assumptions A2) and A3), but simulations presented in this paper were carried out under independence assumption A2) or a more realistic situation where the input regressors have shift structure and are generated by feeding data $\mathrm{u}_k(i)$ into a tapped delay line as

$$\mathrm{u}_{k,i} = \mathrm{col}\{\mathrm{u}_k(i), \mathrm{u}_k(i-1) \ldots, \mathrm{u}_k(i - M + 1)\}. \qquad (83)$$

As mentioned in [23], the regularization parameter ϵ plays a role in the convergence behavior of APA-based algorithms.

A large regularization parameter results in a small step-size, which implies that an APA-based solution has a slow convergence rate but a small misadjustment error in the steady-state, while a small ϵ provides a large step-size, which causes a poor steady-state performance but a fast convergence rate during learning. In all the experiments, the regularization parameter $\epsilon = 0.001$ is set as a small value, whose influence on the step-size μ_k of the APA-based solutions is very small and can be neglected. All the coefficients of the adaptive filters within the network are initialized to zeros.

A. Comparison of Distributed Algorithms

In the first experiment, we evaluate the proposed algorithm with shift structure in the inputs for an incremental network in a system identification scenario, which corresponds to identifying the parameter vector w^o in (4). Consider a network with $N = 50$ nodes in order to seek the two unknown filters with $M = 40$, whose z-domain transfer functions are given by,

$$W_1(z) = \sum_{n=0}^{19} z^{-n} - \sum_{n=21}^{M-1} z^{-n} \quad \text{and} \quad W_2(z) = -\sum_{n=0}^{M-1} z^{-n} \qquad (84)$$

where $w^o(z) = W_1(z)$ for $i \leq 4000$ and $w^o(z) = W_2(z)$ for $4000 < i \leq 8000$. The elements of the vector $u_k(i)$ in (4) within this simulation are generated by passing a unit-variance
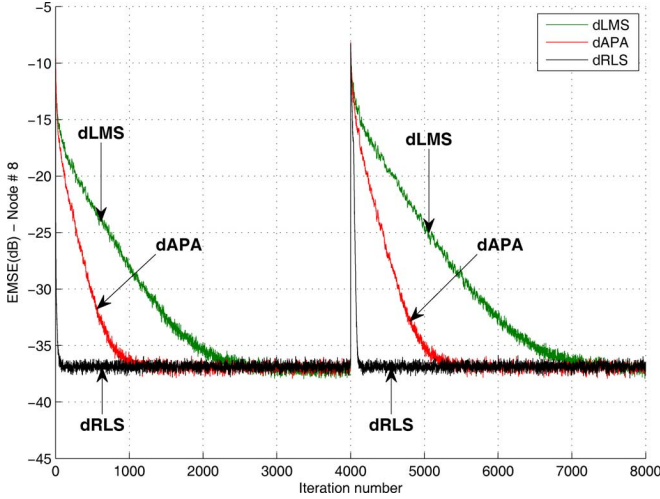
Fig. 2. Comparison of simulated EMSE learning curves at node 8 for the dLMS, dNLMS, dAPA, and dRLS algorithms in a time-varying environment.
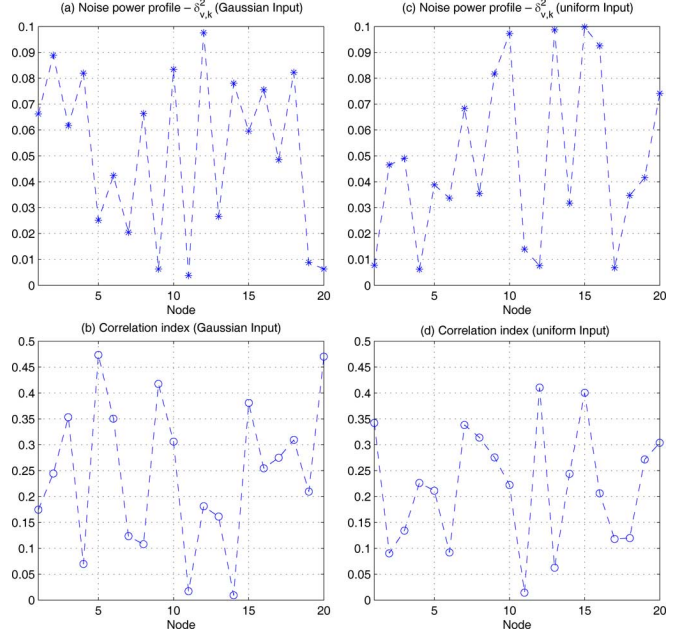

Fig. 3. Node profile: a) Noise power for the Gaussian data network b) Correlation index for the Gaussian data network c) Noise power for the uniform data network d) Correlation index for the uniform data network.

white Gaussian sequence through a coloring filter (proposed in [21]) with the system function as

$$H(z) = 0.1 - 0.2z^{-1} - 0.3z^{-2} + 0.4z^{-3}$$
$$+ 0.4z^{-4} - 0.2z^{-5} - 0.1z^{-6} \quad (85)$$

which results in the input correlation matrix of each node having an eigenvalue spread $\lambda_{\max}/\lambda_{\min} \approx 91.85$. The noise term in (4) is a white additive uncorrelated noise component with variance $\delta_{v,k}^2 = 0.01$. As seen in [10], when the forgetting factor τ_k reaches close to unity, both dRLS and Lc-dRLS enable the networks to have similar steady-state performance. However, Lc-dRLS has slow convergence rate during the initial learning stage due to the matrices $P_{k,i}$ evolving locally. As such, dRLS is chosen in the comparison experiment. Each node within the incremental network is trained by exploiting either dLMS with step-size $\mu_k = 0.0011$, or dAPA with step-size $\mu_k = 0.0125$, or dRLS, for which we select the forgetting factor $\tau_k = 0.95$, the initial value of $P_0 = \epsilon^{-1}I$ and the spatial weighting factor $\varrho_k = 1$. Since the spatial weighting factor ϱ_k in [11] is not considered by other distributed algorithms, $\varrho_k = 1$ ensures a fair comparison. The selection of the parameters for the different algorithms allows dLMS, dAPA and dRLS to converge to a similar steady-state EMSE. On the other hand, as shown in Fig. 19, the setting $T = 3$ enables dAPA to have approximately one-third of the computational cost of dRLS per node. For dRLS, the small value of τ_k gives more relevance to recent data in the estimation process so that changes in the input can be better tracked. We therefore select $\tau_k = 0.95$, which corresponds to a window length of $\approx (1-\tau_k)^{-1} = 20$, to ensure dRLS has the fastest performance over other algorithms in the time-varying w^o environment. The curves are shown in Fig. 2 by running 8000 iterations and averaging 500 Monte Carlo runs, where iteration stands for discrete time. This figure shows that the dRLS algorithm obtains the fastest convergence rate with the largest computational complexity, while the dLMS algorithm has the slowest convergence

rate with the smallest computational complexity. The dAPA algorithm provides good compromise between convergence behavior and computational complexity, namely, dAPA achieves the improved convergence rate with reasonable computational cost. Since for each learning scheme the performances are similar at different nodes, Fig. 2 illustrates a comparison of the EMSE learning curves of the different distributed schemes only at node 8.

In the following examples, computer simulations are performed to compare the experimental results with the theoretical values obtained by using the theoretical expressions. Consider a network with $N = 20$ nodes in order to seek an unknown parameter vector with $M = 10$ as in (4), whose elements are generated randomly from a standardized Gaussian or uniform distribution. The correlated elements of $u_{k,i}$ as in (4) at each node are obtained by passing a white Gaussian or uniform random process with variance $\delta_{u,k}^2 = 1$ through a first-order autoregressive (AR) model filter with z-domain transfer function $\sqrt{1-\alpha_k^2}/(1-\alpha_k z^{-1})$ and $\alpha_k \in [0, 0.5]$. It follows that the auto-correlation sequence of the resulting process $\mathbf{u}_k(i)$ is

$$\mathrm{r}_k(m) = E\mathbf{u}_k(i)\mathbf{u}_k(i-m) = \delta_{u,k}^2 \alpha_k^{|m|} \quad (86)$$

for all integer values m. In this way, the covariance matrix $R_{u,k}$ of the regressor vector $\mathbf{u}_{k,i}$ is a 10×10 Toeplitz matrix with entries $\mathrm{r}_k(m) = \delta_{u,k}^2 \alpha_k^{|m|}$, $m = 0, \ldots, M-1$. The noise term in (4) of each node is a white Gaussian process with variance $\delta_{v,k}^2 \in (0, 0.1]$. Fig. 3 illustrates respectively the node profiles of $\delta_{v,k}^2$ and α_k for both colored Gaussian network inputs and colored uniform data network inputs. Furthermore, as seen in Fig. 3, since node 5 and node 12 have the most highly correlated inputs within the corresponding networks, we draw a comparison of their performance between theory and practice in the following examples.
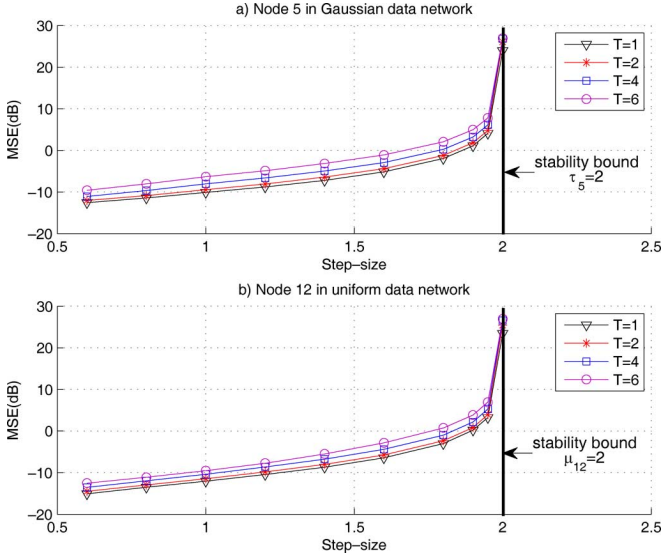
Fig. 4. Simulated MSE curves of dAPA as a function of the step-size: a) Node 5 in the colored Gaussian data network; b) node 12 in the colored uniform data network.

Fig. 5. Learning MSE curves of dAPA using $\mu_k = 0.01$ for node 5 in the colored Gaussian data network.

TABLE II
STABILITY BOUNDS OF STEP-SIZE FOR dAPA
(AT NODE 5 IN THE GAUSSIAN DATA NETWORK)

	$\frac{2}{\lambda_{\max}(E\mathbf{C}_5)}$	$\frac{1}{\lambda_{\max}(\mathbf{X}_5^{-1}\mathbf{Y}_5)}$	$\frac{1}{\lambda_{\max}(\mathbf{H}_5)\in\mathbb{R}^+}$	$\mu_{\max}$
T=1	9.1204	2.0003	5.0117	2.0003
T=2	6.5406	2.0004	3.9605	2.0004
T=4	3.9959	2.0005	3.4544	2.0005
T=6	2.8628	2.0008	2.8974	2.0008

TABLE III
STABILITY BOUNDS OF STEP-SIZE FOR dAPA (AT NODE 12 IN THE
UNIFORM DATA NETWORK)

	$\frac{2}{\lambda_{\max}(E\mathbf{C}_{12})}$	$\frac{1}{\lambda_{\max}(\mathbf{X}_{12}^{-1}\mathbf{Y}_{12})}$	$\frac{1}{\lambda_{\max}(\mathbf{H}_{12})\in\mathbb{R}^+}$	$\mu_{\max}$
T=1	10.3941	2.0002	5.5589	2.0002
T=2	7.1882	2.0003	4.1698	2.0003
T=4	4.1641	2.0004	3.4132	2.0004
T=6	2.9241	2.0006	2.9498	2.0006

B. Mean and Mean-Square Stability

The experimental values are obtained by running dAPA for 10 000 iterations and then averaged over 100 independent experiments to generate the ensemble-average curves. Using expressions (68) and (70), we evaluate the step-size bounds for dAPA with shift structure at the two corresponding nodes in Tables II and III, which verify that the stability bounds on μ_k are approximately $0 < \mu_k < 2$. This fact is further confirmed in Fig. 4, where steady-state MSE curves are plotted as a function of the step-size. $\mathbf{X}_5$, $\mathbf{Y}_5$, $\mathbf{X}_{12}$, and $\mathbf{Y}_{12}$ which are involved in evaluating the expectations for the bounds of step-size, are calculated via ensemble averaging.

C. Transient Performance

Figs. 5 and 6 illustrate the transient performance curves of dAPA with shift structure during the initial 180 samples. Since we choose the same step-size as $\mu_k = 0.01$ for dAPA with different T, increasing T results in faster convergence rate but poor
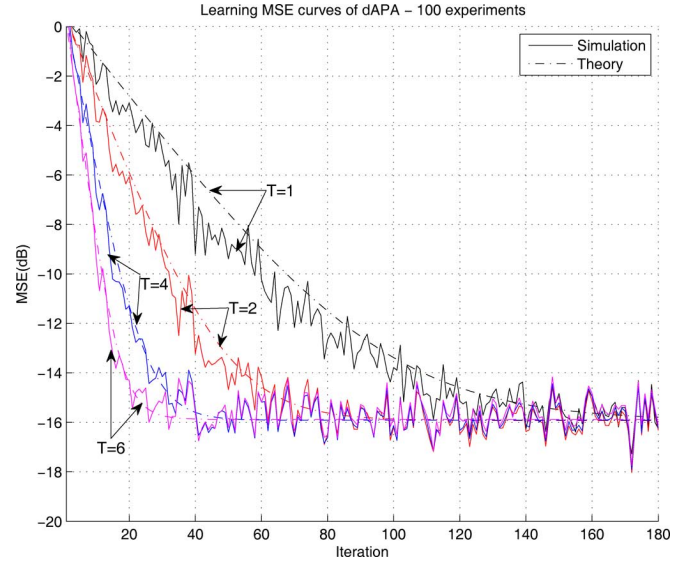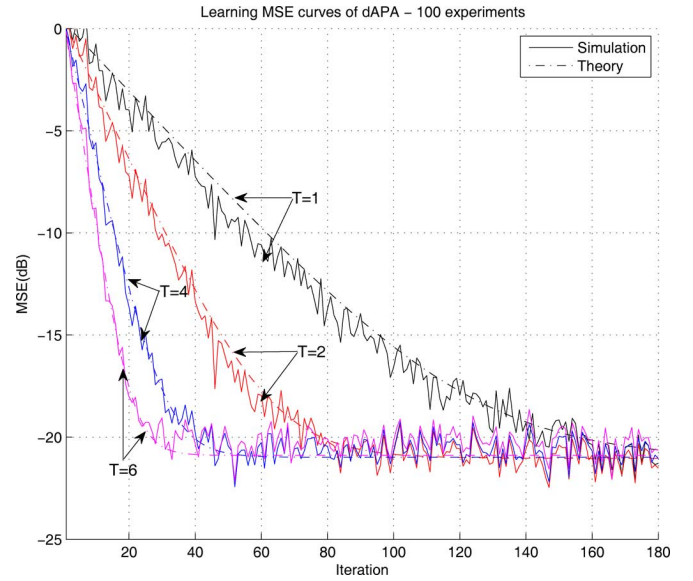

Fig. 6. Learning MSE curves of dAPA using $\mu_k = 0.01$ for node 12 in the colored uniform data network.

misadjustment expected in the steady-state, namely, dNLMS obtains the best steady-state performance, which can be clearly seen in Fig. 4. For the transient performance, the simulation results present good agreement with the theoretical results using (65), where F_k and a_k are calculated by ensemble averaging.

D. Steady-State Performance

In these simulations, we aim at satisfy independence assumption A2). Therefore, we perform the following simulations with temporally independent Gaussian or uniform inputs, i.e., the elements of $u_{k,i}$ in (4) satisfy the temporal assumption in A2). At each node the regressors are generated as independent realizations, so that the spatial assumption in A2) is satisfied. The sample temporal correlation indexes for those inputs are as shown in Fig. 3. This figure also illustrates the noise power
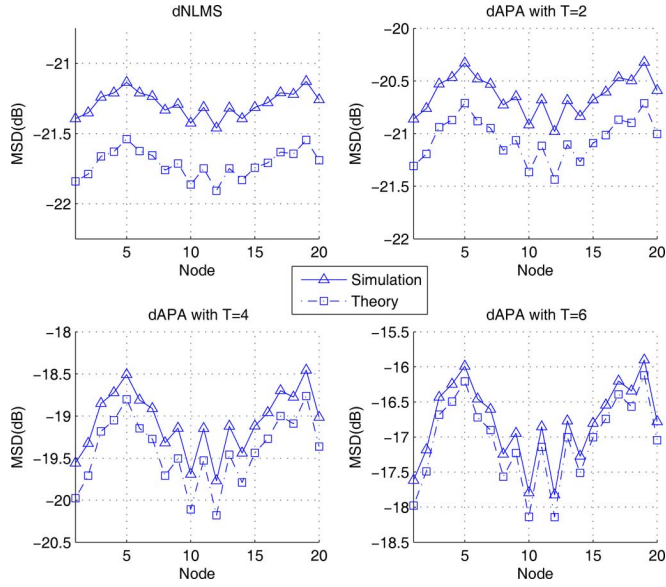
Fig. 7. Steady-state MSD for dNLMS and dAPA using $\mu_k = 0.2$ for the colored Gaussian data network.
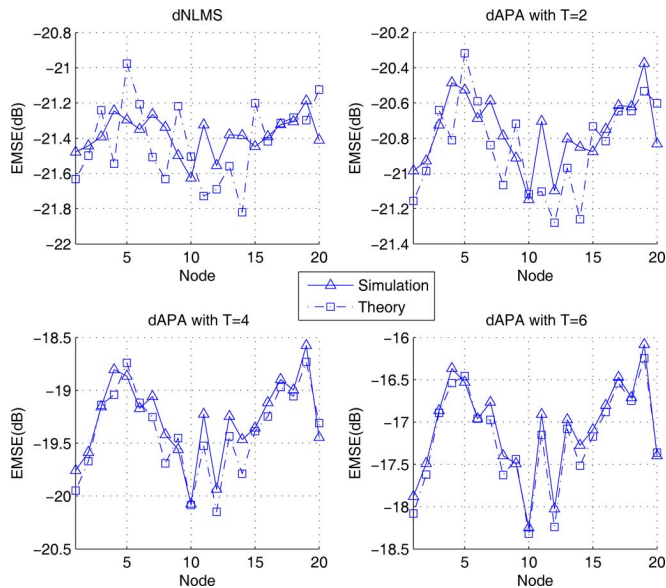

Fig. 8. Steady-state EMSE for dNLMS and dAPA using $\mu_k = 0.2$ for the colored Gaussian data network.
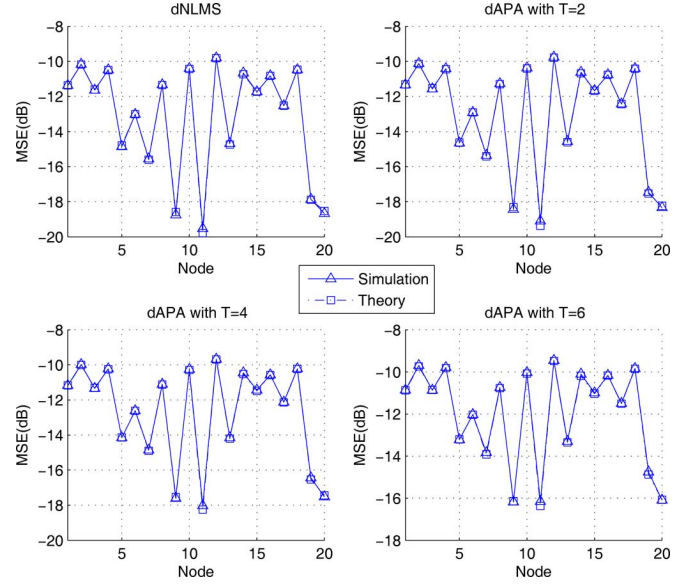

Fig. 9. Steady-state MSE for dNLMS and dAPA using $\mu_k = 0.2$ for the colored Gaussian data network.
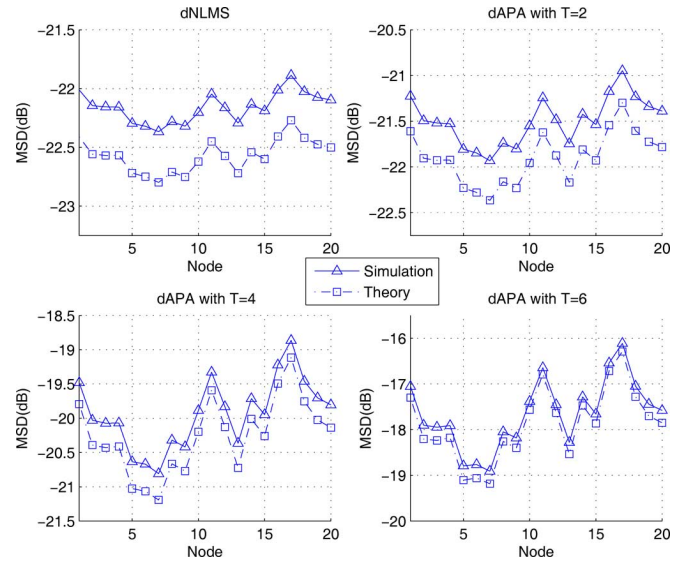

Fig. 10. Steady-state MSD for dNLMS and dAPA using $\mu_k = 0.2$ for the colored uniform data network.

at each node for the corresponding networks. In Figs. 7–18, it is clear to see a good match between theory and simulation. The simulated curves are obtained by averaging the last 1000 instantaneous samples of 10 000 iterations and then averaging 100 independent trials. The theoretical results are calculated by using expressions (78)–(80). Figs. 7–12 show the steady-state MSD, EMSE and MSE curves of dAPA with different T using a particular choice of the step-size $\mu_k = 0.2$ for both the colored Gaussian input data network and the colored uniform input data network. These quantities combine the transformed noise and local statistics from the whole network. As expression (82) claims, even for a large step-size $\mu_k = 0.2$, the MSD curves of dNLMS in Figs. 7 and 10, are approximately flat throughout the

networks. Increasing the rank T of dAPA leads to large fluctuations of the MSD curves. Compared to the MSD, the EMSE and the MSE are more sensitive to local statistics. Since the theoretical MSD of dNLMS is roughly even over the network (82), (79) enables the EMSE curves of dNLMS to have a similar shape as the correlation index. As can be seen in Figs. 8 and 11, there seems a better fit between theory and practice for the steady state EMSE of dAPA with ranks $T = 2, 4, 6$, which is mostly due to the steady-state EMSE of dNLMS having the smallest absolute value around -21.4 dB for the Gaussian input network or around -22.4 dB for the uniform inputs network. For each node, for the difference between theory and simulation, dNLMS has smaller absolute value than dAPA, for example, as shown in Fig. 11 for node 13 the absolute gap value of dNLMS between -22.25 dB and -22.65 dB is about 0.00052 while that
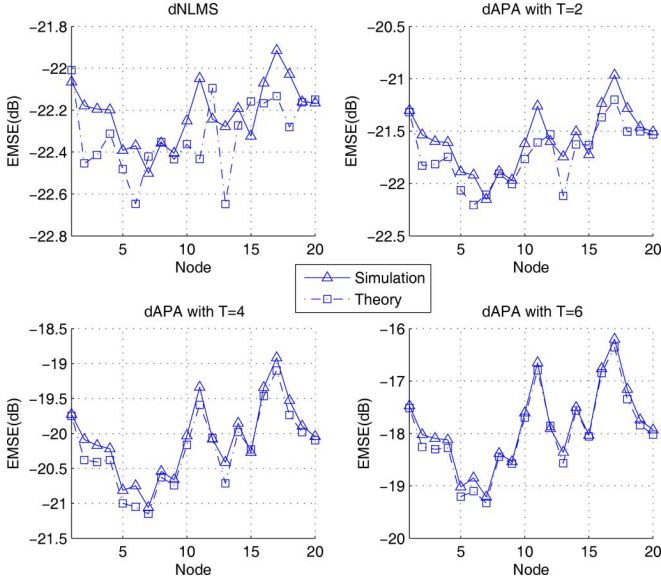
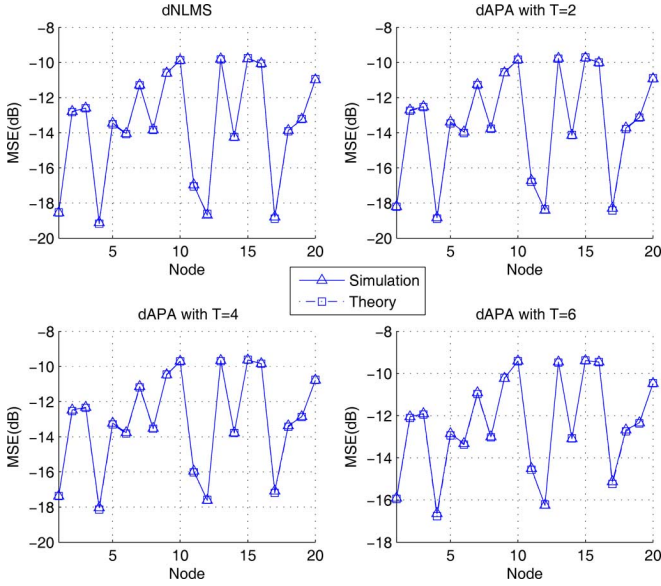Fig. 11. Steady-state EMSE for dNLMS and dAPA using $\mu_k = 0.2$ for the colored uniform data network.

Fig. 12. Steady-state MSE for dNLMS and dAPA using $\mu_k = 0.2$ for the colored uniform data network.
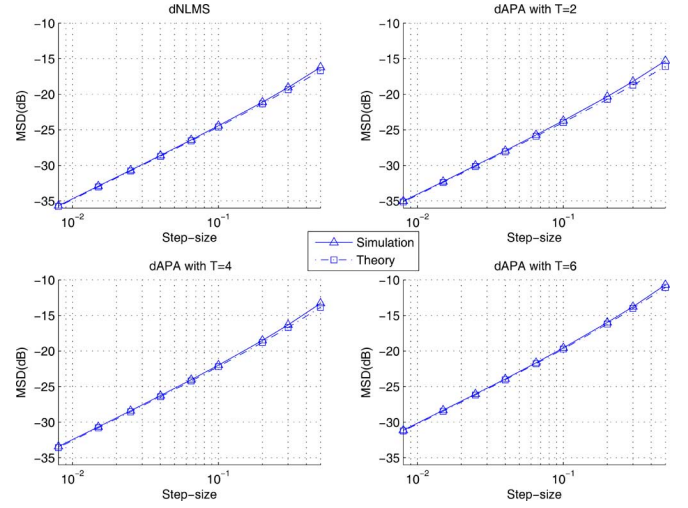

Fig. 13. Steady-state MSD curves of dNLMS and dAPA at node 5 in the colored Gaussian data network as a function of the step-size.
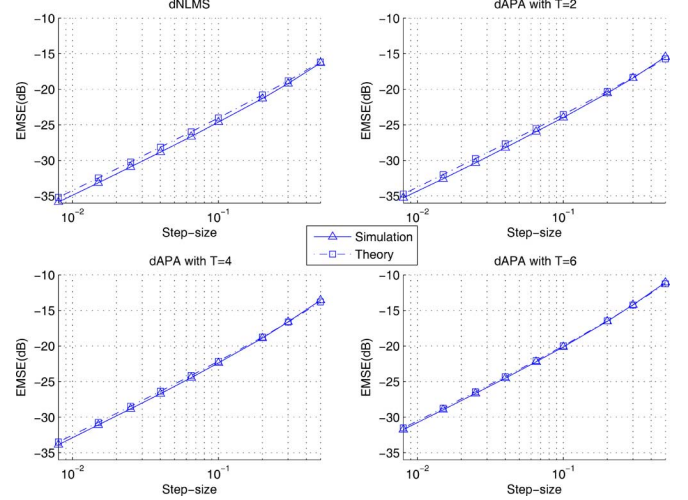

Fig. 14. Steady-state EMSE curves of dNLMS and dAPA at node 5 in the colored Gaussian data network as a function of the step-size.
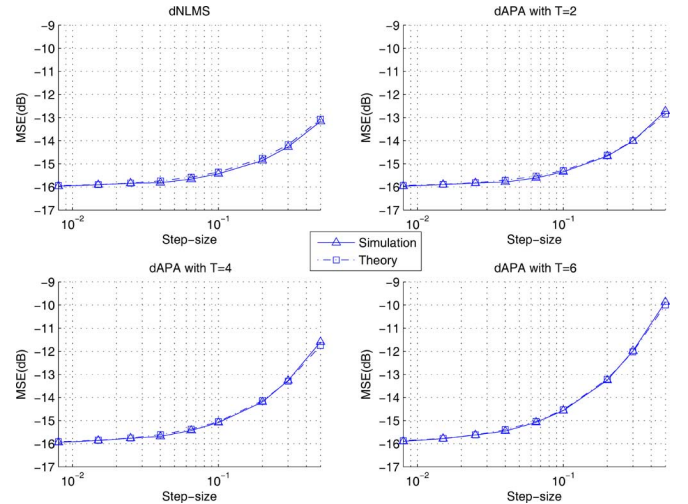

Fig. 15. Steady-state MSE curves of dNLMS and dAPA at node 5 in the colored Gaussian data network as a function of the step-size.

of dAPA with $T = 4$ between -20.4 dB and -20.7 dB is also about 0.0006. In addition, the theoretical results of MSE evaluated by expressions (80) are very close to the simulated results even with the large step-size $\mu_k = 0.2$ as shown in Figs. 9 and 12. It should be highlighted that the assumptions in A3) and A3') underlie the mismatch between the theoretical and simulated results.

Moreover, as shown in Figs. 9 and 12, the MSE curves roughly coincide with the noise power, which indicates that when the proper step-sizes μ_k are chosen, the adaptive filter of each node has good performance and w^o can be well estimated by $\psi_k^{(\infty)}$, namely, the residual error $e_k(\infty)$ is close to the background noise. In addition, we obtain similar finding as in [9] if the whole network is required to have an equalized

performance, Figs. 9 and 12 confirm that the spatial diversity of the adaptive networks can be used to design the step-size for
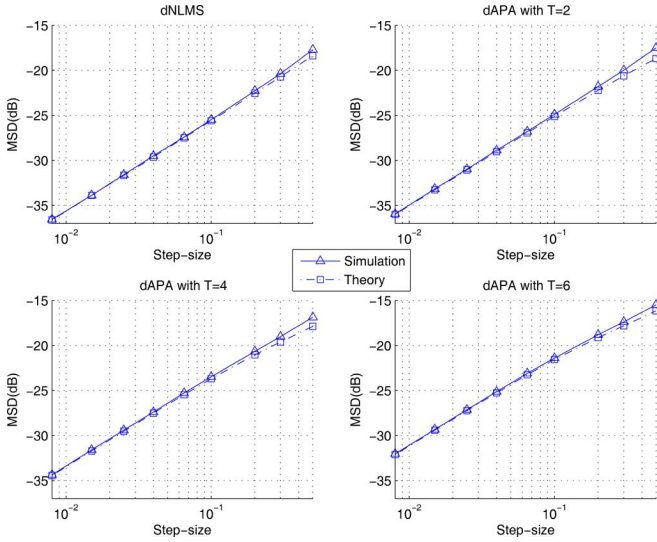
Fig. 16. Steady-state MSD curves of dNLMS and dAPA at node 5 in the colored uniform data network as a function of the step-size.
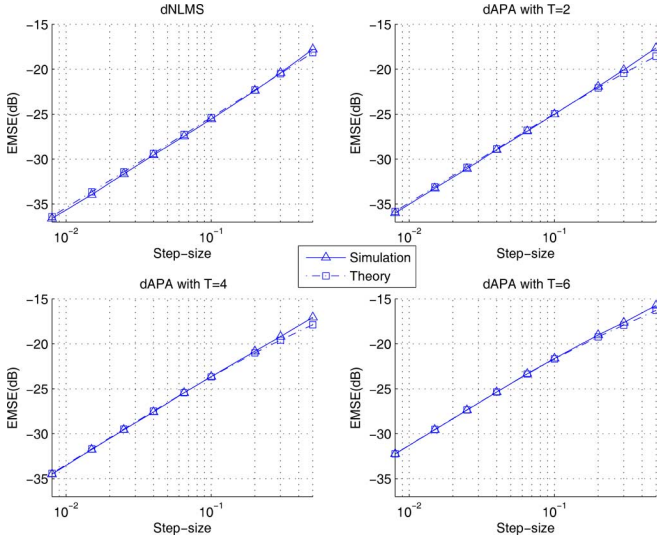

Fig. 18. Steady-state MSE curves of dNLMS and dAPA at node 5 in the colored uniform data network as a function of the step-size.

Fig. 17. Steady-state EMSE curves of dNLMS and dAPA at node 5 in the colored uniform data network as a function of the step-size.

and presented detailed performance analysis based on the weighted space-time energy conservation approach of Lopes and Sayed [9] under assumptions A1), A2), and A3'). The theoretical expressions for MSD, EMSE and MSE, derived in this paper, do not restrict the distribution of the inputs to being Gaussian or white. Both stationary environments and one simple non-stationary environment were considered to test the proposed algorithm. Compared to the dRLS algorithm for certain regressor lengths, the dAPA algorithm at each node involves less computational cost and reduced inter-node communications and memory cost whilst retaining an acceptable steady-state performance. In addition, this algorithm has obtained improved performance as compared to dLMS in the highly correlated input case. The bounds of the step-size for mean and mean-square stability of dAPA have also been evaluated, which have been employed consistently within the simulation experiments. Furthermore, we have compared the theoretical expressions with the simulated results, for both the transient and steady-state performance of dAPA in both Gaussian data network and uniform data network. In addition, we can clearly see that the theoretical results of the steady-state performance have a good match with the experimental results for small step-size.

In the problem of incremental estimation, a Hamiltonian cyclic path through the network is required to be defined and nodes communicate with neighbors within this path. However, this strategy limits the mode of node collaboration in the network and the ability of the network to respond to a topological change, which is likely to be problematic in practical wireless sensor networks. Thus, if more communication and computation resources are available, more sophisticated cooperative modes (rather than the incremental type) for the APA-based algorithm can also be pursued, e.g., a diffusion mode of the form used in [24]–[26], where each node cooperates with a subset of neighboring nodes, but this will be very much dependent upon the network size and topology. These topics will be addressed in future work.

each node. Nodes with poor performance, or high noise level, can be tuned with properly small step-sizes to guarantee a good level of performance equalization throughout the network. In certain cases, they are just relay nodes. Figs. 13–18 illustrate the steady-state MSD, EMSE and MSE curves of dNLMS and dAPA at node 5 for the colored Gaussian input data network and at node 5 for the colored uniform input data network for different choices of the step-size in the range [0.008 0.5], which guarantees the stability of the scheme as mentioned before. The experimental values match well with the theoretical values for small step-size but deviate from the theoretical ones for a larger step-size and larger T.

V. CONCLUSION AND FUTURE WORK

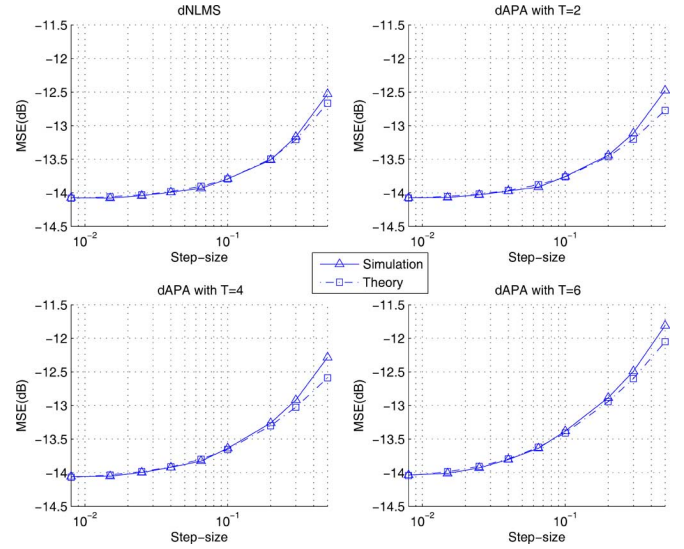In this paper, we have considered new incremental adaptive learning algorithms based on APA for a distributed network

TABLE IV
COMPARISON OF THE ESTIMATED COMPLEXITY PER ITERATION PER NODE FOR
VARIOUS ALGORITHMS FOR THE CASE OF REAL-VALUED DATA

Solution	$\times$	$+$	$\div$
dLMS	$2M+1$	$2M$	
dNLMS	$3M+1$	$3M$	1
dAPA	T^2M+2TM $+M+T^3+T$	T^2M+2TM $+T^3+T^2$	
dRLS	M^2+5M+1	M^2+3M	1

TABLE V
COMPARISON OF THE ESTIMATED COMPLEXITY PER ITERATION PER
NODE FOR VARIOUS INCREMENTAL ALGORITHMS FOR THE CASE OF
COMPLEX-VALUED DATA

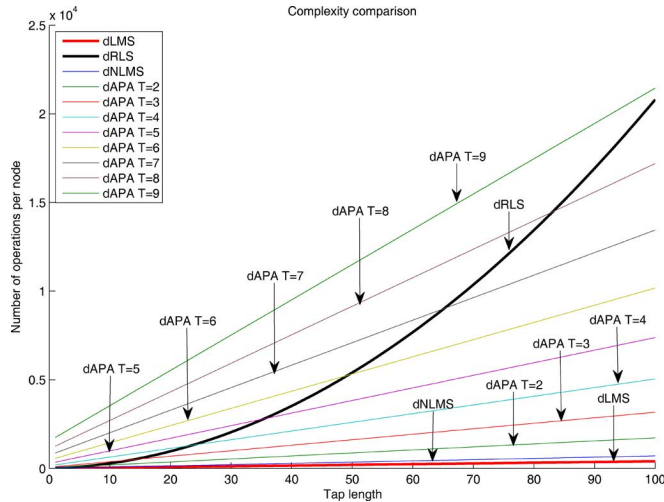| Solution	$\times$	$+$	$\div$
dLMS	$8M+2$	$8M$	
dNLMS	$10M+2$	$10M$	1
dAPA	$4T^2M+8TM$ $+2M+4T^3+4T$	$4T^2M+8TM$ $+4T^3+4T^2$	
dRLS	$4M^2+16M+1$	$4M^2+12M-1$	1

Fig. 19. Complexity comparison in terms of operations per node for various algorithms (dLMS, dNLMS, dAPA and dRLS).

APPENDIX

COMPARISON OF COMPLEXITY,
MEMORY AND TRANSMISSION COSTS

The computational costs for the different schemes in the same style as the presented complexity formulas in [17] are compared in this appendix. Tables IV and V show the total number of operations of various algorithms per node per time instant for real-value data and complex-valued data, respectively. The complexity comparison for each update of the different algorithms per node is depicted in Fig. 19, where we note that the increase of the tap length M enables the maximum T of dAPA, which has lower complexity than dRLS, to become larger, e.g., at $M = 25$, dAPA has less complexity than dRLS when $T < 5$; whereas at $M = 90$, $T < 9$. In addition, as presented in [5], dAPA requires intermediate memory cost between dLMS and dRLS, and the same $O(M)$ transmission complexity as dLMS.

REFERENCES

[1] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proc. IEEE Int. Conf. Acoutstics, Speech and Signal Processing (ICASSP)*, Salt Lake City, UT, May 2001, pp. 2033–2036.
[2] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE J. Sel. Areas Commun.*, vol. 23, pp. 798–808, Apr. 2007.
[3] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–49, Aug. 2004.
[4] J. B. Predd, S. B. Kulkarni, and H. V. Poor, "Distributed learning in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 23, no. 4, pp. 56–69, Jul. 2006.
[5] L. Li and J. A. Chambers, "A new incremental affine projection based adaptive learning scheme for distributed networks," *Fast Commun., Eur. Signal Process. J.*, vol. 88, no. 10, pp. 2599–2603, Oct. 2008.
[6] R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *Proc. 44th IEEE Conf. Decision Control*, Dec. 2005, pp. 6698–6703.
[7] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. 4th Int. Symp. Information Processing in Sensor Networks*, Los Angeles, CA, Apr. 2005, pp. 63–70.
[8] R. Olfati-Saber, "Distributed Kalman filter with embedded consensus filters," in *CDC-ECC*, Dec. 2005, pp. 8179–8184.
[9] 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.
[10] A. H. Sayed and C. G. Lopes, "Adaptive processing over distributed networks," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E90-A, no. 8, pp. 1504–1510, 2007.
[11] A. H. Sayed and C. G. Lopes, "Distributed recursive least-squares strategies over adaptive networks," in *Proc. 40th Asilomar Conf. Signals, Systems, Computers*, Pacific Grove, CA, Oct. 2006, pp. 223–237.
[12] C. G. Lopes and A. H. Sayed, "Distributed adaptive incremental strategies: Formulation and performance analysis," in *Proc. IEEE Int. Conf. Acoutstics, Speech, Signal Processing (ICASSP)*, Toulouse, France, May 2006, pp. 584–587.
[13] S. Shimauchi and S. Makino, "Stereo projection echo canceler with true echo path estimation," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Detroit, MI, May 1995, pp. 3059–3062.
[14] Y. Kaneda, M. Tanaka, and J. Kojima, "An adaptive algorithm with fast convergence for multi-input sound control," in *Proc. Active*, Newport Beach, CA, Jul. 1995, pp. 993–1004.
[15] S. G. Sankaran and A. A. Beex, "Stereophonic acoustic echo cancellation using NLMS with orthogonal correction factors," in *Proc. Int. Workshop Acoust. Echo Noise Control*, Pocono Manor, PA, Sep. 1999, pp. 40–43.
[16] J. Apolinario Jr., M. L. R. Campos, and P. S. Diniz, "Convergence analysis of the binormalized data-reusing LMS algorithm," *IEEE Trans. Signal Process.*, vol. 48, no. 11, pp. 3235–3242, Nov. 2003.
[17] A. H. Sayed, *Fundamentals of Adaptive Filtering*. New York: Wiley, 2003.
[18] H.-C. Shin and A. H. Sayed, "Transient behavior of affine projection algorithms," in *Proc. IEEE Int. Conf. Acoutstics, Speech, Signal Processing (ICASSP)*, Hong Kong, Apr. 2003, vol. 6, pp. 353–356.
[19] H.-C. Shin and A. H. Sayed, "Mean-square performance of a family of affine projection algorithms," *IEEE Trans Signal Process.*, vol. 52, no. 1, pp. 90–102, Jan. 2004.
[20] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.
[21] S. Haykin, *Adaptive Filter Theory*, Second ed. Englewood Cliffs, NJ: Prentice-Hall, 1996.
[22] N. R. Yousef and A. H. Sayed, "A unified approach to the steady-state and tracking analyzes of adaptive filters," *IEEE Trans. Signal Process.*, vol. 49, no. 2, pp. 314–324, Feb. 2001.
[23] D. P. Mandic, "A generalized normalized gradient descent algorithm," *IEEE Signal Process. Lett.*, vol. 11, no. 1, pp. 115–118, Jan. 2004.
[24] 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.
[25] F. 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.
[26] C. Lopes and A. H. Sayed, "Diffusion adaptive networks with changing topologies," in *Proc. IEEE Int. Conf. Acoutstics, Speech, Signal Processing (ICASSP)*, Las Vegas, NV, Apr. 2008, pp. 3285–3288.
[27] A. H. Sayed, *Adaptive Filters*. New York: Wiley, 2008.
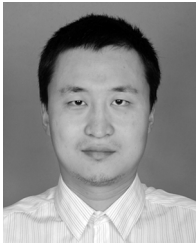
Leilei Li (S'08–M'09) received the B.Sc. degree in electrical and electronic engineering from Communication University of China, Beijing, China, the M.Sc. degree in electrical and electronic engineering from Edinburgh University, Edinburgh, U.K., in 2002 and 2005, respectively, and the Ph.D. degree from Loughborough University, Loughborough, U.K., in 2009.

From 2002 to 2004, he worked in the China Centre Television (CCTV) Station and joined the technology design team for the CCTV new building, as an Engineer. He is currently with the Academy of Broadcasting Planning in the State Administration of Radio Film and Television (SARFT), Beijing, China, working on wireless communications. His research interests include signal processing, distributed networks, and wireless communications.

Jonathon A. Chambers (S'83–M'90–SM'98) was born in Peterborough, U.K., in 1960. He received the B.Sc. (Hons.) degree in electrical and electronic engineering from the Polytechnic of Central London, London, U.K., in 1985, and the Ph.D. degree in digital signal processing from the University of London, London, U.K., in 1990, after postgraduate research at Peterhouse, Cambridge University, Cambridge, U.K., and Imperial College, London, U.K.

From 1979 to 1982, he was an Artificer Apprentice in Action, Data and Control in the Royal Navy. He has held academic and industrial positions at Bath University, Cardiff University, Imperial College London, King's College London, and Schlumberger Cambridge Research, all in the United Kingdom. In July 2007, he joined the Department of Electronic and Electrical Engineering, Loughborough University, U.K., as Professor of Communications and Signal Processing. He leads the Advanced Signal Processing Group and is the Director of Research (Deputy-Head) within the Department. He has authored or coauthored more than 300 research outputs, including two monographs and more than 100 journal articles. He has mentored 47 researchers through to Ph.D. graduation.

Dr. Chambers is a member of the IEEE Technical Committee on Signal Processing Theory and Methods, was the Technical Programme Chair for the IEEE Workshop on Statistical Signal Processing 2009, Cardiff, U.K., and is the Co-Technical Programme Chair for the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2011, Prague, Czech Republic. He has served as an Associate Editor for IEEE publications, including the IEEE TRANSACTIONS ON SIGNAL PROCESSING for more than ten years. Most recently, he was awarded the first QinetiQ Visiting Fellowship in 2007 for his contributions to adaptive signal processing and successful partnership with the internationally renowned defense systems company QinetiQ.

Cassio G. Lopes (S'06–S'08) received the B.S. degree and the M.S. degree in electrical engineering from the Federal University of Santa Catarina, Brazil, in 1997 and 1999, respectively, and the M.S. degree and the Ph.D. degree in electrical engineering from the University of California, Los Angeles (UCLA) in 2004 and 2008, respectively.

From 1999 to 2002, he worked with hardware design for biomedical signals acquisition and transmission over the Internet. He also taught undergraduate-level courses in computer science and electrical engineering. From 2005 to 2007, he worked in cooperation with the NASA Jet Propulsion Laboratory to develop frequency tracking algorithms to support direct-to-Earth Mars communications during entry, descent, and landing. In 2008, he worked at the Instituto Tecnologico de Aeronautica (ITA), Sao Jose dos Campos, Brazil, as a Postdoctoral Researcher, developing distributed inertial navigation systems for aerospace applications. He is currently with the Department of Electronic Systems of The University of Sao Paulo (USP), Escola Politecnica, where he is an Assistant Professor of electrical engineering. His current research interests are theory and methods for adaptive and statistical signal processing, distributed adaptive estimation and data fusion, as well as frequency tracking algorithms for space communications.

Ali. H. Sayed (S'90–M'92–SM'99–F'01) received the Ph.D. degree from Stanford University, Stanford, CA, in 1992.

He is Professor and Chairman of Electrical Engineering at the University of California, Los Angeles (UCLA) and Principal Investigator of the Adaptive Systems Laboratory. He has published widely, with over 300 articles and five books, in the areas of statistical signal processing, estimation theory, adaptive filtering, signal processing for communications and wireless networking, and fast algorithms for large structured problems. He is coauthor of the textbook *Linear Estimation* (Prentice-Hall, 2000), of the research monograph *Indefinite Quadratic Estimation and Control* (SIAM, 1999), and co-editor of Fast Algorithms for Matrices with Structure (SIAM, 1999). He is also the author of the textbooks *Fundamentals of Adaptive Filtering* (Wiley, 2003) and *Adaptive Filters* (Wiley, 2008). He has contributed several encyclopedia and handbook articles.

Dr. Sayed has served on the editorial boards of the *IEEE Signal Processing Magazine*, the *European Signal Processing Journal*, the *International Journal on Adaptive Control and Signal Processing*, and the *SIAM Journal on Matrix Analysis and Applications*. He also served as the Editor-in-Chief of the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 2003 to 2005 and the *EURASIP Journal on Advances in Signal Processing* from 2006 to 2007. He is a member of the Signal Processing for Communications and the Signal Processing Theory and Methods technical committees of the IEEE Signal Processing Society. He has served on the Publications (2003–2005), Awards (2005), and Conference (2007–present) Boards of the IEEE Signal Processing Society. He served on the Board of Governors (2007–2008) of the same Society and is now serving as Vice-President of Publications (2009–present). His work has received several recognitions, including the 1996 IEEE Donald G. Fink Award, the 2002 Best Paper Award from the IEEE Signal Processing Society, the 2003 Kuwait Prize in Basic Sciences, the 2005 Terman Award, the 2005 Young Author Best Paper Award from the IEEE Signal Processing Society, and two Best Student Paper Awards at international meetings (1999 and 2001). He has served as a 2005 Distinguished Lecturer of the IEEE Signal Processing Society and as General Chairman of IEEE International Conference on Acoustics, Speech and Signal Processing (IEEE Int. Conf. Acoutstics, Speech and Signal Processing (ICASSP)) 2008.