Adaptive Algorithms and Variable Structures for Distributed Estimation

Thesis submitted to Loughborough University in candidature for the degree of Doctor of Philosophy.

Leilei Li

Advanced Signal Processing Group
Loughborough University
2009

ABSTRACT

The analysis and design of new non-centralized learning algorithms for potential application in distributed adaptive estimation is the focus of this thesis. Such algorithms should be designed to have low processing requirement and to need minimal communication between the nodes which would form a distributed network. They ought, moreover, to have acceptable performance when the nodal input measurements are coloured and the environment is dynamic.

Least mean square (LMS) and recursive least squares (RLS) type incremental distributed adaptive learning algorithms are first introduced on the basis of a Hamiltonian cycle through all of the nodes of a distributed network. These schemes require each node to communicate only with one of its neighbours during the learning process. An original steady-steady performance analysis of the incremental LMS algorithm is performed by exploiting a weighted spatial-temporal energy conservation formulation. This analysis confirms that the effect of varying signal-to-noise ratio (SNR) in the measurements at the nodes within the network is equalized by the learning algorithm.

A novel incremental affine projection algorithm (APA) is then proposed to ameliorate the problem of ill-convergence in adaptive filters with coloured inputs which are controlled by the incremental LMS algorithm. The computational and memory costs of this incremental

APA algorithm are shown for a range of filter lengths to be lower than those of an incremental RLS algorithm. The transient and steady-state performances of the incremental APA algorithm are evaluated in detail through analytical and simulation studies. The nature of the inter-node collaboration within the incremental APA algorithm is further enhanced through the adoption of a diffusion-based cooperation protocol.

The concept of variable tap-length (VT) adaptive filtering is next introduced to facilitate structural change during learning. The monotonically non-increasing nature of the converged difference between the segmented mean square error (MSE) of a filter formed from a number of the initial coefficients of an adaptive filter and the MSE of the full adaptive filter, as a function of the tap-length of the adaptive filter, is confirmed through analysis. An innovative strategy for adaptation of the leakage factor, a key parameter in the fractional tap-length (FT) learning algorithm, is proposed to ensure the converged tap-length can be used to calculate the true length of the unknown system for a range of initial tap-lengths. For sub-Gaussian noise conditions, a VT adaptive filtering algorithm which exploits both second and fourth order statistics is also presented.

Finally, VT adaptive filters are introduced for the first time into distributed adaptive estimation. In particular, an FT learning algorithm is used to determine the length of the adaptive filter within each node in parallel with the scheme to calculate the coefficients of the filter. The efficacy of this technique is confirmed through analytical and simulation studies of the steady-state performance.

To my loving wife

and to my parents

CONTENTS

ACKNOWLEDGEMENTS

I would like to give my great thanks to my supervisor Prof. Jonathon A. Chambers for providing me with the chance to start my scientific career by offering me a PhD position. It has been an excellent opportunity and experience, and allowed my dream to come true. Since the beginning of my study, Prof. Jonathon A. Chambers has kindly spent much time consulting with me regarding the problems of my research. When writing papers, his advice and suggestions have very much helped me to clarify the focus of my study and to sharpen my analysis. Every progress step I have made includes a great contribution from him. He also provided considerable encouragement during my entire study and gave me the chance to serve as a reviewer for some journals, which has not only enriched my theoretical knowledge but has also broadened my prospective in understanding the subject I was studying. His advice and comments on my thesis drafts have been invaluable. Without them, I would not have finished this thesis.

I wish to give special thanks to Prof. Ali H. Sayed and Dr. Cassio G. Lopes from the University of California Los Angeles (UCLA) and the University of Sao Paulo respectively, who have given me valuable comments on my work and been co-authors of my transactions paper.

Appreciation is also give to Mr. Cheng Shen, a research student, during my experience in Cardiff, for kind support and useful informa-

tion given as I just started my PhD study at Cardiff University. I would like to thank my friends in Cardiff for making it such a nice experience during the time I spent studying and living there.

My experience in Loughborough University has been wonderful, where I have worked with members of the advanced signal processing group (ASPG). I thank them for taking the time to discuss my research with me and provided me with much needed information. Special thanks to Dr. Yonggang Zhang from the ASPG for giving me a chance to discuss my research findings with whom I have co-authored two conference papers.

I would like to thank my beloved wife, Lei, who has offered moral support and unconditional love during my entire study. Distance made us closer and has showed me the true meaning of love. Last but not least, I extend my thanks to my parents and friends in China, who provided endless support and encouragement to me at all times.

STATEMENT OF ORIGINALITY

The contributions of this thesis concentrate on the design and analysis of new distributed adaptive algorithms, which take advantage of cooperation among the individual adaptive nodes and thereby exploit the space-time data. The contributions are supported by five published papers (including one journal paper and four conference papers), one submitted transactions paper and one submitted conference paper. The summary of contributions are given as follows:

In Chapter 2, a steady-state performance analysis of the incremental LMS algorithm is performed for an incremental network with non-Gaussian wide sense stationary data. In addition, the closed-form equations describing mean-square performance are also derived by utilizing a weighted spatial-temporal energy conservation formulation. The work of this chapter has been presented in:

- **L. Li**, Y. Zhang, J. A. Chambers and A. H. Sayed, "Steady-state performance of incremental learning over distributed networks for non-Gaussian data," in *Proc. 9th Int. Conf. on Signal Processing (ICSP)*, Beijing, China, vol. 1, pp. 227 - 230, October 2008.

In Chapter 3, motivated by Newton's method, a novel incremental

algorithm relying on the APA rule is proposed to improve convergence performance with correlated inputs as compared with the incremental LMS algorithm, and to obtain acceptable misadjustment performance together with reduced computational and memory cost for certain tap-lengths as compared with the incremental RLS algorithm. The results have been published in *Signal Processing* and submitted to *IEEE Trans. on Signal Processing*:

- **L. Li** and J. A. Chambers, "A new incremental affine projection based adaptive algorithm for distributed networks," (*Fast Communications in) Signal Processing*, vol. 88, no. 10, pp. 2599 - 2603, October 2008.

- **L. Li**, J. A. Chambers, C. G. Lopes and A. H. Sayed, "Distributed estimation over an adaptive incremental network based on the affine projection algorithm," *IEEE Trans. on Signal Processing*, in review.

In Chapter 4, the adaptive APA-based method in incremental networks is extended to more general topology networks, namely, diffusion-based cooperation protocols. Moreover, the proposed algorithm is also studied in dynamic topology networks. The work of this chapter has been submitted in:

- **L. Li** and J. A. Chambers, "Adaptive distributed estimation based on the APA algorithm over a diffusion network with changing topology," submitted in *Proc. Int. Conf. on Statistical Signal Processing (SSP'09)*, Cardiff, UK.

In Chapter 5, two novel VT adaptive schemes are proposed to search for a good estimate of the optimal tap-length and to obtain improved

convergence performance as compared with the FT algorithm. The research results have been published respectively in:

- **L. Li** and J. A. Chambers, "A novel adaptive leakage factor scheme for enhancement of a variable tap-length learning algorithm," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Las Vegas, USA, pp. 3837 - 3840, April 2008.

- Y. Zhang, **L. Li** and J. A. Chambers, "Variable tap-length adaptive algorithm which exploits both second and fourth order statistics," in *Proc. 9th Int. Conf. on Signal Processing (ICSP)*, Beijing, China, vol. 1, pp. 223 - 226, October 2008.

In Chapter 6, it is the first time to introduce the concept of VT into distributed adaptive estimation. A novel incremental VT algorithm is therefore developed for distributed networks in certain situations, where the optimal tap-length of the filters within the corresponding network is unknown or time-varying. The steady-state performance of the proposed algorithm is studied and verified by simulation results. The work of this chapter has been presented in:

- **L. Li**, Y. Zhang and J. A. Chambers, "Variable length adaptive filtering within incremental learning algorithms for distributed networks," in *Proc. Asilomar Conf. on Signals, Systems and Computers*, California, USA, October 2008.

LIST OF ACRONYMS

APA	Affine Projection Algorithm
AR	Autoregressive
dAPA	Distributed APA
dLMS	Distributed LMS
dNLMS	Distributed NLMS
dRLS	Distributed RLS
DTL	Deceasing Tap-Length
EMSE	Excess Mean Square Error
FIR	Finite Impulse Response
FT	Fractional Tap-Length
HOS	Higher Order Statistics
ITL	Increasing Tap-Length
Lc-dRLS	Low Communication dRLS
LMF	Least Mean Fourth
LMS	Least Mean Square

MMSE	Minimum Mean Square Error
MSD	Mean Square Deviation
MSE	Mean Square Error
NLMS	Normalized LMS
nLMS	Noncooperative LMS
RLS	Recursive Least Squares
sLMS	Stochastic Single Global Weight LMS
SNR	Signal-to-Noise Ratio
SOS	Second Order Statistics
VTLMS	Variable Tap-Length LMS
VT	Variable Tap-Length

LIST OF SYMBOLS

$\|.\|^2$	Absolute squared operator
E	Statistical expectation
$(\cdot)^{\mathrm{T}}$,$(\cdot)^*$	Transposition and complex-conjugate transposition
∇	Differentiation operator
$\|\|.\|\|^2$	Squared Euclidean norm operator
$\mathrm{Rd}(\cdot)$	Nearest integer operator
$\mathrm{col}\{a\}$	Column vector with entries a
$Tr(A)$	Trace of the matrix A
$\mathrm{vec}\{\cdot\}$	Vectorization operator in columns
$A \otimes B$	Kronecker product of A and B
$\lambda(A)$	All eigenvalues of A
$\lambda_{\max}(A)$	Largest eigenvalue of A
$\mathrm{diag}\{\cdot\}$	Block diagonalization operator
$A \odot B$	Block Kronecker product of A and B
$\mathrm{bvec}\{\cdot\}$	Block vectorization operator

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Stochastic-gradient algorithms

In the literature, the linear estimation problem shown in Figure 1.1 is widely discussed [3], [4], [5], [6], [7] in the context of the development of various adaptive algorithms, which is based upon a convex cost function of the form,

$$\min_{w} J(w) = \min_{w} E|\mathbf{d} - \mathbf{u}w|^2 \tag{1.1.1}$$

where the desired output $\mathbf{d}$ is a zero-mean random complex value with variance σ_d^2, the input $\mathbf{u}$ denotes a $1 \times M$ row vector with zero-mean, $|\cdot|^2$ indicates absolute squared operation and E is statistical expectation. Both $\mathbf{d}$ and $\mathbf{u}$ are assumed jointly wide-sense stationary processes. The solution $w^o = R_u^{-1} R_{du}$ is the optimal weight vector, and the resulting minimum mean square error (MMSE) cost is given by,

$$\text{MMSE} = \sigma_d^2 - R_{ud} R_u^{-1} R_{du} \tag{1.1.2}$$

where $R_u = E\mathbf{u}^*\mathbf{u}$ is a positive definite covariance matrix, $R_{du} = E\mathbf{d}\mathbf{u}^* = R_{ud}^*$ is a cross correlation vector and $(\cdot)^*$ denotes complex-conjugate transposition. Such least mean square problems (1.1.1) are quadratic in w and have a unique global minimum at w^o, provided R_u

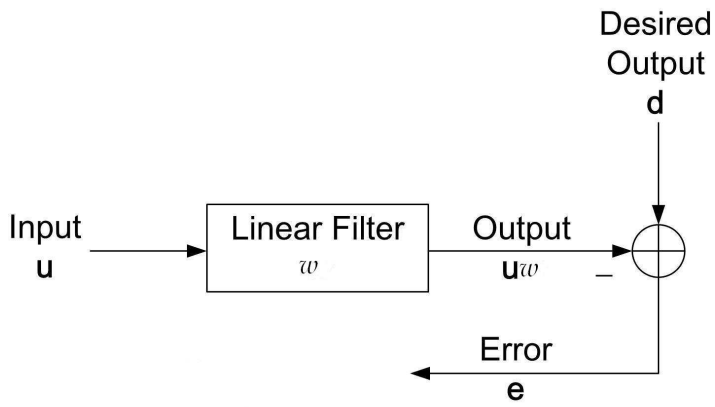

Figure 1.1. Linear estimation problem.

has full column rank. Throughout the thesis, 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.

The steepest-descent algorithm provides an iterative procedure to approximate the solution w^o, namely, as in [6],

$$\begin{aligned} w_i &= w_{i-1} - \mu[\nabla_w J(w_{i-1})]^* \\ &= w_{i-1} + \mu[R_{du} - R_u w_{i-1}], \quad i \geq 0 \quad w_{-1} = \text{initial value} \end{aligned} \tag{1.1.3}$$

where the positive scalar μ is the step-size, i denotes the iteration index and $\nabla_w J(w_{i-1})$ is the gradient vector of $J(w)$ at $w = w_{i-1}$. Lack of the exact signal statistics in practice leads to the development of stochastic-gradient algorithms, which can usefully also track the variations in the signal statistics. Let $\{d(i), u_i\}$ denote the observations of the random variables $\{\mathbf{d}, \mathbf{u}\}$ in (1.1.1). According to its simplicity and robustness,

the LMS algorithm is developed on the basis of the steepest-descent algorithm, where the statistical quantities $\{R_{du}, R_u\}$ are replaced by the instantaneous approximations $\{d(i)u_i^*, u_i^*u_i\}$ (Widrow 1959/60) [3] and the corresponding recursion (1.1.3) therefore becomes,

$$w_i = w_{i-1} + \mu u_i^*[d(i) - u_i w_{i-1}] \tag{1.1.4}$$

In addition, LMS is also the exact solution to the localized constrained optimization problem as [6],

$$\min_{w_i} \|w_i - w_{i-1}\|^2, \quad \text{subject to } r(i) = (1 - \mu\|u_i\|^2)e(i) \tag{1.1.5}$$

where $\|\cdot\|^2$ indicates squared Euclidean norm operation and two estimation errors are defined by:

$$e(i) \triangleq d(i) - u_i w_{i-1} \quad \text{(a priori output estimation error)} \tag{1.1.6}$$

$$r(i) \triangleq d(i) - u_i w_i \quad \text{(a posteriori output estimation error)}. \tag{1.1.7}$$

When for all i the condition $0 < \mu\|u_i\|^2 < 2$ holds, it is always the case that $|r(i)| < |e(i)|$, namely, $u_i w_i$ yields a better estimate for $d(i)$ than $u_i w_{i-1}$ (except for the case of $e(i) = r(i) = 0$). Using $\delta w = w_i - w_{i-1}$ and the constraint in (1.1.5), the optimization problem (1.1.5) is equivalent to,

$$\min_{\delta w} \|\delta w\|^2, \quad \text{subject to } u_i \delta w = \mu\|u_i\|^2 e(i) \tag{1.1.8}$$

In the case of $\|u_i\|^2 \neq 0$, $\delta w^o = \mu u_i^* e(i)$ has the smallest Euclidean

norm [6] and with $\delta w^o = w_i - w_{i-1}$ the LMS recursion for w_i is obtained,

$$w_i = w_{i-1} + \mu u_i^* [d(i) - u_i w_{i-1}]. \tag{1.1.9}$$

This therefore verifies that LMS is the exact solution to the localized constrained optimization problem (1.1.5).

Regularized Newton's method is developed to obtain the optimal solution w^o and the resulting recursive form is

$$\begin{aligned} w_i &= w_{i-1} + \mu[\epsilon I + \nabla_w^2 J(w_{i-1})]^{-1} [\nabla_w J(w_{i-1})]^* \\ &= w_{i-1} + \mu[\epsilon I + R_u]^{-1} [R_{du} - R_u w_{i-1}] \end{aligned} \tag{1.1.10}$$

where the regularization parameter ϵ is a small positive value to avoid the inversion of a rank deficient matrix R_u and $\nabla_w^2 J(w_{i-1})$ is the further differentiation of $\nabla_w J(w_{i-1})$ with respect to w_{i-1}^* and called the Hessian Matrix. One well-known property of Newton's method (ϵI is dropped in (1.1.10)) is that the choice $\mu = 1$ guarantees convergence in a single iteration to the minimizing argument of a quadratic cost function $J(w)$. By replacing the quantities $\{R_{du}, R_u\}$ by different instantaneous approximations [6], several stochastic-gradient algorithm

variations based on (1.1.10) have been devised, for example:

$$\text{NLMS}: w_i = w_{i-1} + \frac{\mu}{\epsilon + \|u_i\|^2} u_i^* [d(i) - u_i w_{i-1}] \tag{1.1.11}$$

$$\text{APA}: U_i = \begin{pmatrix} u_i \\ u_{i-1} \\ \vdots \\ u_{i-T+1} \end{pmatrix} \text{ and } d_i = \begin{pmatrix} d(i) \\ d(i-1) \\ \vdots \\ d(i-T+1) \end{pmatrix}$$

$$w_i = w_{i-1} + \mu U_i^* (\epsilon I + U_i U_i^*)^{-1} [d_i - U_i w_{i-1}] \tag{1.1.12}$$

$$\text{RLS}: P_i = \lambda^{-1} (P_{i-1} - \frac{\lambda^{-1} P_{i-1} u_i^* u_i P_{i-1}}{1 + \lambda^{-1} u_i P_{i-1} u_i^*})$$

$$w_i = w_{i-1} + P_i u_i^* [d(i) - u_i w_{i-1}] \tag{1.1.13}$$

where in APA T denotes the number of recent regressor samples, and in RLS the initial condition $P_{-1} = \epsilon^{-1} I$ is generally used and $0 \ll \lambda < 1$ is the weighting or forgetting factor. In normalized LMS (NLMS) the adaptive gain is a scalar but it is time-varying, and ϵ is a constant to avoid divide by zero and associated gain amplification. The term "affine projection" in APA represents linear projection type mapping onto the subspace defined by the columns of a low column-rank input matrix. In fact, as in [6] 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 thesis, as in [6] the ϵ is dropped for notational convenience. Although the RLS algorithm is derived as an exact solution to a least-squares estimation problem, it can also be motivated as a stochastic-gradient method (details shown in [6]). These algorithms generally trade performance with computational complexity. Employing adaptive filters within the nodes of a network motivates the development of adaptive

estimation over distributed networks, which allows applications to obtain improved performance by exploiting data in both the spatial and temporal dimensions as compared to individual filters without cooperative implementation.

1.2 Distributed adaptive estimation

Compared with a traditional centralized solution that requires a powerful central processor and extensive amount of communications, the objective of a distributed solution is to reduce significantly the amount of processing and communications between nodes relying only on local data exchange and interactions between their immediate neighborhood nodes whilst retaining the accuracy of a centralized solution [8], [9], [10]. Distributed adaptive networks can therefore find potential application in a wide number of fields, such as precision agriculture, environmental monitoring, defence, transportation and factory instrumentation [8]. Particular merits of a distributed adaptive estimation solution are collaboration and adaptation. The computational burden is shared over the individual nodes so that communications are reduced as compared to a centralized network, and power and bandwidth usage are also thereby reduced [1]. On the other hand, the adaptive capability enables tracking of not only the variations of the environment but also the topology of the network.

As shown in [1], three modes of distributed cooperative network are illustrated in Figure 1.2. An incremental network tends to require the least amount of communications and power, and exploits a cyclic cooperative pattern, generally a Hamiltonian cycle through the network, where the information flows from one node to the immediate neighbour

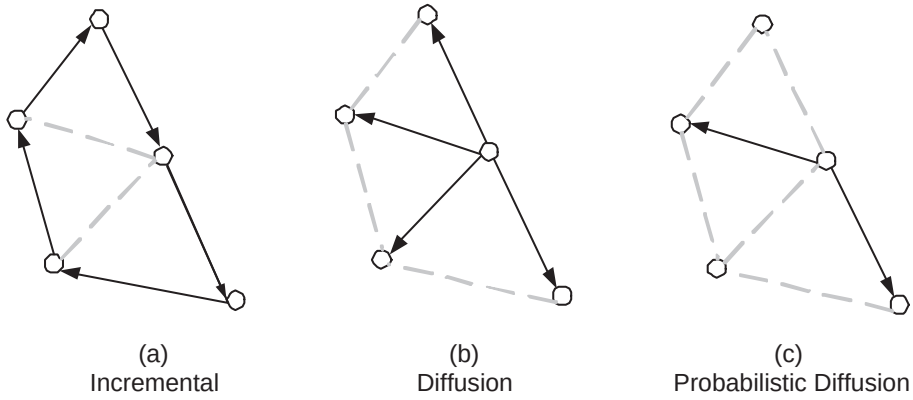

Figure 1.2. Three modes of cooperation between nodes in a distributed estimation environment [1].

nodes. On the other hand, each node in a diffusion network communicates more than in an incremental fashion; namely, it interacts with all its neighbours and obtains more data. A probabilistic diffusion on the other hand selects the nodes with which it communicates in a probabilistic fashion. Novel adaptive algorithms are required to perform learning in such cooperative environments the development of which is the focus of the research of this thesis.

In all previous work, such distributed networks have been assumed to use a fixed tap-length of the adaptive filter at each node. When the tap-length is chosen too long or sometimes too short in some typical scenarios, the convergence performance and computational cost will be significantly impacted. According to the conflicting requirements of performance and complexity, it is desirable to derive for the first time distributed adaptive algorithms that can automatically find the optimum tap-length for distributed adaptive estimation.

1.3 Variable tap-length LMS (VTLMS) algorithm

As a key parameter, tap-length plays an important role in the design of adaptive filters based on the LMS algorithm, which has been utilized in a wide range of applications [4], [5], [6], as a consequence of its simplicity and robustness. However, in many applications the tap-length of the adaptive filter is for simplicity assumed to be fixed, which is not suitable for certain situations where the optimal tap-length of the system filer is unknown or variable. Furthermore, 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 MMSE; 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 [11], many related works [12], [13], [14], [15], [16], [17], [18] have been reported in the literature on the basis of this concept. Utilizing instantaneous errors for the tap-length adaptation, the FT algorithm has been proposed to arrive at improved convergence properties over other methods and is formulated as follows [17]:

$$e^{(L(i))}_{L(i)-\Delta}(i)=d(i)-u_i(1:L(i)-\Delta)w_{i-1}(1:L(i)-\Delta) \quad (1.3.1)$$

$$l_f(i+1)=l_f(i)-\alpha+\beta[(e^{(L(i))}_{L(i)-\Delta}(i))^2-(e^{(L(i))}_{L(i)}(i))^2] \quad (1.3.2)$$

where α is the leakage factor, β is the step-size of the fractional tap-length update and $L(i)$ is the integer value of $l_f(i)$ calculated by

$$L(i+1)=\begin{cases} \mathrm{Rd}\,(l_f(i)) & if|L(i)-l_f(i+1)|\geq\nu \\ L(i) & \text{otherwise} \end{cases} \quad (1.3.3)$$

where Rd(·) rounds the embraced value to the nearest integer and the parameter ν is a small integer. The FT algorithm is therefore extended for the first time to distributed adaptive estimation to obtain the estimate of the optimal tap-length of adaptive filters within distributed adaptive estimation schemes.

1.4 Organization of the thesis

The remainder of this thesis is organized as follows:

Chapter 2 provides a brief review of the development of incremental learning algorithms based on LMS or RLS methods over distributed networks. In addition, a steady-state performance analysis of the dLMS algorithm is provided for an incremental network with non-Gaussian data which is supported by simulation.

In Chapter 3, a new incremental adaptive learning scheme based on APA is proposed to overcome efficiently the ill-convergence in LMS-type adaptive filters with coloured inputs for incremental distributed networks and provides an improved convergence rate as compared to an LMS based scheme. As compared to RLS for a range of filter lengths, the new scheme requires less computational cost and communications to obtain an acceptable misadjustment at the steady-state stage.

Chapter 4 presents a novel diffusion affine projection algorithm for adaptive estimation over distributed networks, where the network topology exploits more communication resources and computational costs than the incremental type to increase the cooperation among nodes. Such a strategy is robust to node and link failures and utilizes the network connectivity to obtain improved performance over the non-cooperative method. In addition, the proposed algorithm is also

extended to networks with dynamic topology, where the links between nodes are random due to link failures or time delays.

Chapter 5 provides an analysis of the converged difference between the segmented MSE of a filter formed from a number of the initial coefficients of an adaptive filter, and the MSE of the full adaptive filter, which is confirmed as a function of the tap-length of the adaptive filter to be monotonically non-increasing. Based on this analysis, a new strategy for adaptation of the leakage factor is therefore developed to systematically choose the key parameters in the FT learning scheme to ensure convergence to the desired range, which can be used to compute the true tap-length of the unknown filter. In addition, a novel VTLMS algorithm exploiting both second and fourth order statistics of the errors is presented to improve the convergence performance of the fractional tap-length function for the sub-Gaussian noise case.

In Chapter 6, motivated by both ideas of tap-length adaptation and distributed adaptive estimation, a variable tap-length adaptive filtering algorithm within the context of incremental learning for distributed networks is presented for situations where the optimal tap-length of the filters within the corresponding network is unknown or variable with respect to time.

Chapter 7, finally, summarizes the content of this thesis and provides the overall conclusions and suggestions for future work.

Chapter 2

INCREMENTAL LEARNING SCHEMES FOR DISTRIBUTED ADAPTIVE ESTIMATION

2.1 Introduction

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 a physical phenomenum and events in a specified environment. At time i, each 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 (shown in Figure 2.1) is adopted to represent these measurements as follows [19]:

$$d_k(i) = \sum_{m=1}^{M} \beta_m d_k(i-m) + v_k(i) \tag{2.1.1}$$

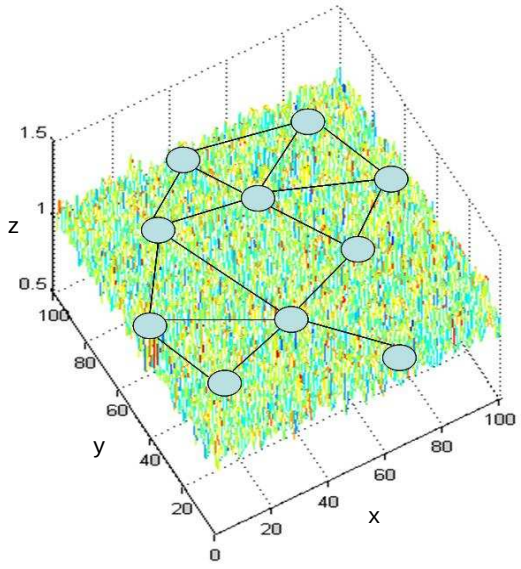

Figure 2.1. Modelling application: AR field [2].

where $v_k(i)$ is additive zero-mean noise and the random selected coefficients $\{\beta_m\}$ are the parameters of the underlying model. Define the $1 \times M$ regression vector

$$u_{k,i} = [d_k(i-1)\, d_k(i-2)\, \ldots\, d_k(i-M)] \tag{2.1.2}$$

and the $M \times 1$ parameter vector

$$w^o = \text{col}\{\beta_1, \beta_2, \ldots, \beta_m\}, (M \times 1) \tag{2.1.3}$$

where col$\{a\}$ indicates column vector with entries a, then the measurement equation (2.1.1) at each node k can be rewritten as an equivalent linear measurement model

$$d_k(i) = u_{k,i} w^o + v_k(i). \tag{2.1.4}$$

The objective is 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, the linear space-time least mean square estimation problem is posed as:

$$\min_{w} J(w) \quad \text{and} \quad J(w) = E\|\mathbf{d} - \mathbf{U}w\|^2 \tag{2.1.5}$$

where the global desired response vector and input matrix are

$$\mathbf{d}=\text{col}\{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N\}, \quad (N \times 1) \tag{2.1.6}$$

$$\mathbf{U}=\text{col}\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N\}, \quad (N \times M). \tag{2.1.7}$$

where the $\{d_k(i), u_{k,i}\}$ are realizations of $\{\mathbf{d}_k, \mathbf{u}_k\}$.

One solution is a non-distributed implementation, which allows an individual adaptive filter at each node k to solely respond to its local data $\{d_k(i), u_{k,i}\}$ and estimate w^o independently of the other nodes. As a result, the quality of the local estimate at each node relies on the statistical properties of its own data. When a multitude of nodes in the network has access to data, in order to take advantage of node cooperation, it is useful to seek solutions that utilize the assumed collaboration among the nodes to arrive at the parameters of interest.

The following desirable features should be required for an adaptive estimation approach over distributed networks:

- *Adaptive implementation*: The variance in the statistical profile of the data can be promptly tracked by the solution, which allows each node to combine the received information and local data to update its local estimate of the parameter of interest.

- *Convergence performance*: The local estimates are shared in a cooperative fashion according to the network topology. Each node in the network should end up with an estimate that is as accurate as it could be obtained if the space-time dimension of the data were fully exploited.

- *Energy consumption and complexity*: Communications and local processing at the nodes should be minimized.

Distributed adaptive solutions are therefore derived to satisfy these requirements, where local processing at every node allows an application to obtain an estimate in a collaborative manner, thereby saving communications and network resources. Such cooperative adaptive solutions are expected to obtain improved performance over the non-cooperative individual filters. Several developments of the LMS and RLS types based on distributed adaptive estimation have been proposed in both incremental and diffusion network topologies [1], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29]. Due to space constraint, the following section will give a brief introduction to incremental distributed solutions. These solutions allow the local information of a certain node to create a ripple effect through the network and influence the performance behaviour of other nodes.

2.2 Incremental adaptive solutions over distributed networks

Inspired by the earlier studies on incremental methods for distributed optimization problems [9], [30], [31], [32], Lopes and Sayed extended these incremental methods in the context of adaptive estimation over a distributed network, where at least one cyclic path can be established

across the network. In this distributed network with incremental learning, a local estimation at each node is computed via combining the received information from its immediate neighbourhood node and the local information from its own sensor. Let $\psi_k^{(i)}$ denote a local estimation at node k and time i and w_i denote a global estimation at time i. The steepest-descent solution is therefore introduced to compute the optimal tap weight w^o, via,

$$\psi_k^{(i)} = \psi_{k-1}^{(i)} - \mu[\nabla J_k(w_{i-1})]^*, \quad k = 1, 2, \ldots, N \tag{2.2.1}$$

At time instant i, the first node starts with the initial local tap vector $\psi_0^{(i)} = w_{i-1}$, which is regarded as the current global estimation obtained cyclically across the network. At the last node, the local estimation $\psi_N^{(i)}$ is given to the next global estimation at time $i+1$, namely $w_i = \psi_N^{(i)}$. Using the instantaneous approximations, the implementation of the stochastic single global weight scheme is formed as,

$$\begin{cases} \psi_0^{(i)} = w_{i-1} \\ \psi_k^{(i)} = \psi_{k-1}^{(i)} + \mu_k u_{k,i}^*(d_k(i) - u_{k,i}w_{i-1}), \quad k = 1, \ldots, N \\ w_i = \psi_N^{(i)} \end{cases} \tag{2.2.2}$$

For a truly distributed solution to avoid transfer of the global estimate w_{i-1} through the network, the evaluation of $\nabla J_k(w_{i-1})$ is replaced by $\nabla J_k(\psi_{k-1}^{(i)})$. Thus, the implementation of (2.2.2) becomes,

Table 2.1. Pseudo-code implementation of dLMS.

For each time instant $i \geq 0$ repeat:
$\psi_0^{(i)} = w_{i-1}$
For k=1, ... N
$\psi_k^{(i)} = \psi_{k-1}^{(i)} + \mu_k u_{k,i}^*[d_{k,i} - u_{k,i}\psi_{k-1}^{(i)}]$
end
$w_i = \psi_N^{(i)}$

where the received information $\psi_{k-1}^{(i)}$ is obtained from node $k-1$. The simulated results of noncooperative LMS (nLMS), stochastic single global weight LMS (sLMS) [1] and distributed LMS (dLMS) solutions are compared in Figure 2.2, where a 20-node network seeks a 10×1 unknown filter. The input signal is from an independent Gaussian sequence with $R_k = I$ and the background noise is from zero-mean real white Gaussian data with variance $\sigma_v^2 = 0.001$. The simulated curves are obtained by averaging 100 independent Monte Carlo runs and the global EMSE evolution curves are obtained by averaging all the local EMSE_k results of nodes in the network, namely,

$$\text{global EMSE} = \frac{1}{N}\sum_{k=1}^{N} \text{EMSE}_k \tag{2.2.3}$$

In simulation, both dLMS and sLMS solutions have the same step-size as $\mu_k = 0.05$. As the analysis in [1], the dLMS algorithm has improved steady-state performance and faster convergence rate than the sLMS algorithm. Figure 2.2 shows the converged global EMSE of dLMS at approximately 34.5dB after 10 iterations, on the other hand, the converged global EMSE of sLMS is at approximately 29.5dB but

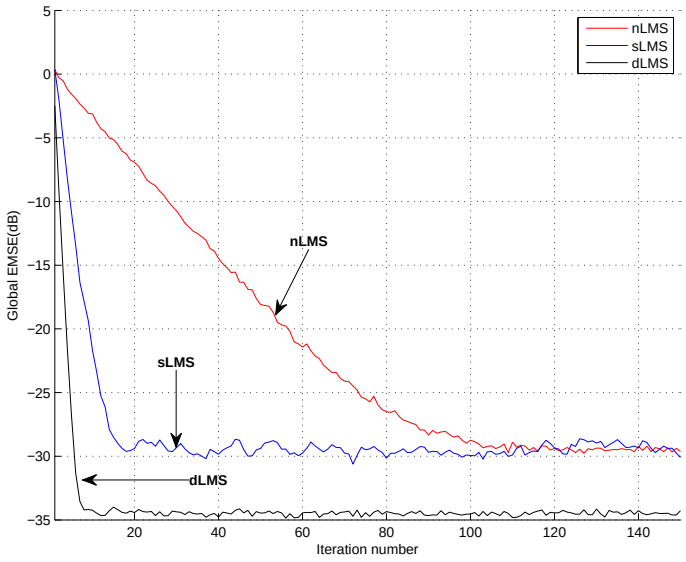

Figure 2.2. Global EMSE of optimization algorithms within an incremental network.

only after 15 iterations. However, for the nLMS solution, the selection of $\mu_k = 0.095$ allows the global EMSE to converge after 100 iterations at almost the same misajustment as the sLMS solution but the converged global EMSE is still 5dB above dLMS. Compared with other methods, the dLMS algorithm therefore has the best performance.

With the development of hardware manufacturing, the computational capability of the processor is continuously increased but the cost is reduced. Therefore, more complicated adaptation rules can be considered and an incremental RLS-based implementation [27] is introduced to obtain better performance at the price of computational complexity and memory cost. The implementation of such an algorithm is constructed as,

Table 2.2. Pseudo-code implementation of dRLS.

For each time instant $i \geq 0$ repeat:

$\psi_0^{(i)} = w_{i-1}$ and $P_{0,i} = \lambda^{-1} P_{i-1}$

For k=1, ... N

$\psi_k^{(i)} = \psi_{k-1}^{(i)} + \frac{P_{k-1,i}}{\gamma_k^{-1} + u_{k,i} P_{k-1,i} u_{k,i}^*} u_{k,i}^* [d_k(i) - u_{k,i} \psi_{k-1}^{(i)}]$

$P_{k,i} = P_{k-1,i} - \frac{P_{k-1,i} u_{k,i}^* u_{k,i} P_{k-1,i}}{\gamma_k^{-1} + u_{k,i} P_{k-1,i} u_{k,i}^*}$

end

$w_i = \psi_N^{(i)}$ and $P_{N,i} = P_i$

where $\gamma_k > 0$ indicates a spatial weighting factor and $0 \ll \lambda < 1$ is a forgetting factor. To reduce the communications between nodes, a low communication distributed RLS (Lc-dRLS) adaptation that requires the same amount of communication as other algorithms is also proposed in [27], as,

Table 2.3. Pseudo-code implementation of Lc-dRLS.

For each time instant $i \geq 0$ repeat:

$\psi_0^{(i)} = w_{i-1}$

For k=1, ... N

$\psi_k^{(i)} = \psi_{k-1}^{(i)} + \frac{P_{k,i-1}}{\gamma_k^{-1} + u_{k,i} P_{k,i-1} u_{k,i}^*} u_{k,i}^* [d_k(i) - u_{k,i} \psi_{k-1}^{(i)}]$

$P_{k,i} = P_{k,i-1} - \frac{P_{k,i-1} u_{k,i}^* u_{k,i} P_{k,i-1}}{\gamma_k^{-1} + u_{k,i} P_{k,i-1} u_{k,i}^*}$

end

$w_i = \psi_N^{(i)}$

where matrix $P_{k,i}$ evolves locally at node k to reduce the transmission complexity from $O(M^2)$ to $O(M)$. The performance analyses of adaptive estimation algorithms based on LMS and RLS type learning over incremental networks for Gaussian data have been studied in [1], [27].

For the non-Gaussian case, the steady-state performance analysis of dLMS is studied for the first time in the following section. This is an important practical consideration as in many sensor network-type applications Gaussianity of the data can not be guaranteed.

2.3 Performance analysis for non-Gaussian data

To evaluate the performance of dLMS, the following assumptions as in [1] are utilized

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

$$\mathbf{d}_k(i) = \mathbf{u}_{k,i} w^o + \mathbf{v}_k(i) \tag{2.3.1}$$

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

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 $i \neq j$

These two assumptions are utilized to build a spatial-temporal independent model. In addition, with the purpose to facilitate analysis, the dLMS algorithm is assumed to be used in a stationary environment, where the statistics of various input and noise signals, and the system vector w^o are assumed to be fixed. It is important to point out that the algorithm derivation is carried out without regard to any independence assumptions: the distributed algorithm does work, as the error curves show; the algorithm operation/formulation is not related to the assumptions required to analyze its (mean-square) performance.

The following local error signals defined as in [1] are introduced to carry out the evaluation:

$$\tilde{\boldsymbol{\psi}}_{k-1}^{(i)} \triangleq w^o - \boldsymbol{\psi}_{k-1}^{(i)}, \quad \tilde{\boldsymbol{\psi}}_{k}^{(i)} \triangleq w^o - \boldsymbol{\psi}_{k}^{(i)} \tag{2.3.2}$$

$$\mathbf{e}_{a,k}(i) \triangleq \mathbf{u}_{k,i}\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}, \quad \mathbf{e}_{p,k}(i) \triangleq \mathbf{u}_{k,i}\tilde{\boldsymbol{\psi}}_{k}^{(i)} \tag{2.3.3}$$

Note that the output error is given by

$$\mathbf{e}_k(i) = \mathbf{e}_{a,k}(i) + \mathbf{v}_k(i). \tag{2.3.4}$$

The objective is to evaluate the following steady-state mean square deviation (MSD), EMSE and MSE measures at each node k:

$$\eta_k \triangleq E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(\infty)}\|^2 \qquad \text{(MSD)} \tag{2.3.5}$$

$$\zeta_k \triangleq E|\mathbf{e}_{a,k}(\infty)|^2 \qquad \text{(EMSE)} \tag{2.3.6}$$

$$\xi_k \triangleq \zeta_k + \sigma_{v,k}^2 \qquad \text{(MSE)} \tag{2.3.7}$$

Introduce further the weighted error signals:

$$\boxed{\mathbf{e}_{p,k}^{\Sigma_k}(i) \triangleq \mathbf{u}_{k,i}\Sigma_k\tilde{\boldsymbol{\psi}}_{k}^{(i)}, \quad \mathbf{e}_{a,k}^{\Sigma_k}(i) \triangleq \mathbf{u}_{k,i}\Sigma_k\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}} \tag{2.3.8}$$

where Σ_k is a Hermitian positive-definite weighting matrix that can be chosen arbitrarily at each node k. Also exploit the weighted norm notation $\|x\|_{\Sigma}^2 = x^*\Sigma x$ for a vector x and Hermitian positive-definite $\Sigma > 0$. After the same manipulations as in [1], the expression relating between two neighboring nodes is formulated as:

$$\tilde{\boldsymbol{\psi}}_{k}^{(i)} + \frac{\mathbf{u}_{k,i}^* \mathbf{e}_{a,k}^{\Sigma_k}(i)}{\|\mathbf{u}_{k,i}\|_{\Sigma_k}^2} = \tilde{\boldsymbol{\psi}}_{k-1}^{(i)} + \frac{\mathbf{u}_{k,i}^* \mathbf{e}_{p,k}^{\Sigma_k}(i)}{\|\mathbf{u}_{k,i}\|_{\Sigma_k}^2} \tag{2.3.9}$$

By calculating the energies of both sides of (2.3.9), a spatial-temporal energy relation is obtained as

$$\|\tilde{\boldsymbol{\psi}}_k^{(i)}\|_{\Sigma_k}^2 + \frac{|\mathbf{e}_{a,k}^{\Sigma_k}(i)|^2}{\|\mathbf{u}_{k,i}\|_{\Sigma_k}^2} = \|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{\Sigma_k}^2 + \frac{|\mathbf{e}_{p,k}^{\Sigma_k}(i)|^2}{\|\mathbf{u}_{k,i}\|_{\Sigma_k}^2} \tag{2.3.10}$$

which is an exact energy relation between two adjacent nodes in space and time, and is derived without any approximations. For simplicity, the time index i is dropped. Applying the expectation operation to both sides of (2.3.10), it becomes

$$\boxed{E\|\tilde{\boldsymbol{\psi}}_k\|_{\Sigma_k}^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}\|_{\Sigma_k'}^2 + \mu_k^2\sigma_{v,k}^2 E\|\mathbf{u}_k\|_{\Sigma_k}^2} \tag{2.3.11}$$

where Σ_k' is given by

$$\boxed{\Sigma_k' = \Sigma_k + \mu_k E(\mathbf{u}_k^*\mathbf{u}_k\Sigma_k + \Sigma_k\mathbf{u}_k^*\mathbf{u}_k) + \mu_k^2 E(\|\mathbf{u}_k\|_{\Sigma_k}^2\mathbf{u}_k^*\mathbf{u}_k)}. \tag{2.3.12}$$

In order to evaluate the performance of the learning algorithm, the following three moments must be examined:

$$E\mathbf{u}_k^*\mathbf{u}_k = R_{u,k} \tag{2.3.13}$$

$$E\|\mathbf{u}_k\|_{\Sigma_k}^2 = Tr(R_{u,k}\Sigma_k) \tag{2.3.14}$$

$$E(\|\mathbf{u}_k\|_{\Sigma_k}^2\mathbf{u}_k^*\mathbf{u}_k) \tag{2.3.15}$$

where $Tr(A)$ indicates trace of the matrix A.

In [1], with the assumption of Gaussian regressors, eigendecomposition and diagonalization methods are exploited to derive the closed-form equations for (2.3.11) and (2.3.12). In the non-Gaussian wide sense stationary case (with finite variance), vectorization and a prop-

erty of Kronecker products are introduced to achieve the theoretical mean-square quantities of dLMS. First, the $M^2 \times 1$ vectors are introduced, given by:

$$\delta_k = \text{vec}\{\Sigma_k\} \text{ and } \delta'_k = \text{vec}\{\Sigma'_k\} \tag{2.3.16}$$

which allow these relations (2.3.11) and (2.3.12) to be expressed by using a convenient vector notation [6], [33]. The $\text{vec}\{\cdot\}$ notation is used in two ways: $\delta = \text{vec}\{\Sigma\}$ 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 $\Sigma = \text{vec}\{\delta\}$ indicates a matrix whose entries are recovered from δ. The following useful property for the $\text{vec}\{\cdot\}$ notation when working with Kronecker products [34] is also introduced: for any matrices $\{P, \Sigma, Q\}$ of compatible dimensions, it holds that

$$\text{vec}\{P\Sigma Q\} = (Q^{\text{T}} \otimes P)\text{vec}\{\Sigma\} \tag{2.3.17}$$

where $(\cdot)^{\text{T}}$ indicates matrix transposition and $A \otimes B$ is a Kronecker product of A and B. Using (2.3.17) to express some items in the right side of (2.3.12), these moments become

$$\text{vec}\{E(\mathbf{u}_k^* \mathbf{u}_k)\Sigma_k\} = (I \otimes R_{u,k})\delta_k \tag{2.3.18}$$

$$\text{vec}\{\Sigma_k E(\mathbf{u}_k^* \mathbf{u}_k)\} = (R_{u,k}^{\text{T}} \otimes I)\delta_k \tag{2.3.19}$$

$$\text{vec}\{E(\|\mathbf{u}_k\|^2_{\Sigma_k} \mathbf{u}_k^* \mathbf{u}_k)\} = E[(\mathbf{u}_k^* \mathbf{u}_k)^{\text{T}} \otimes (\mathbf{u}_k^* \mathbf{u}_k)]\delta_k. \tag{2.3.20}$$

The vec$\{\cdot\}$ operation is applied to both sides of (2.3.12) to obtain a linear relation between the corresponding vectors $\{\delta'_k, \delta_k\}$, namely,

$$\delta'_k = F_k \delta_k \tag{2.3.21}$$

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

$$\boxed{F_k = I - \mu_k X_k + \mu_k^2 Y_k} \tag{2.3.22}$$

with $X_k = (R_{u,k}^{\mathrm{T}} \otimes I) + (I \otimes R_{u,k})$ and $Y_k = E[(\mathbf{u}_k^* \mathbf{u}_k)^{\mathrm{T}} \otimes (\mathbf{u}_k^* \mathbf{u}_k)]$. For clarity, recall the time index i. Therefore, expression (2.3.11) becomes

$$E\|\tilde{\boldsymbol{\psi}}_k^{(i)}\|^2_{\mathrm{vec}\{\delta_k\}} = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|^2_{\mathrm{vec}\{F_k \delta_k\}} + \mu_k^2 \sigma_{v,k}^2 (\acute{r}_k^{\mathrm{T}} \delta_k) \tag{2.3.23}$$

where it exploits the fact that $E\|\mathbf{u}_{k,i}\|^2_{\Sigma_k} = Tr(R_{u,k}\Sigma_k) = \acute{r}_k^{\mathrm{T}} \delta_k$ with $\acute{r}_k = \mathrm{vec}\{R_{u,k}^{\mathrm{T}}\}$. For simplicity of notation, the vec$\{\cdot\}$ notation is dropped from the subscripts in (2.3.23):

$$\boxed{E\|\tilde{\boldsymbol{\psi}}_k^{(i)}\|^2_{\delta_k} = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|^2_{F_k \delta_k} + \mu_k^2 \sigma_{v,k}^2 (\acute{r}_k^{\mathrm{T}} \delta_k)}. \tag{2.3.24}$$

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

$$E\|\boldsymbol{\rho}_k\|^2_{\delta_k} = E\|\boldsymbol{\rho}_{k-1}\|^2_{F_k \delta_k} + \mu_k^2 \sigma_{v,k}^2 (\acute{r}_k^{\mathrm{T}} \delta_k). \tag{2.3.25}$$

By iterating (2.3.25) over one cycle, N coupled equations are obtained:

$$\begin{aligned}
E\|\boldsymbol{\rho}_1\|^2_{\delta_1} &= E\|\boldsymbol{\rho}_N\|^2_{\mathrm{F}_1\delta_1} + g_1\delta_1 \\
E\|\boldsymbol{\rho}_2\|^2_{\delta_2} &= E\|\boldsymbol{\rho}_1\|^2_{\mathrm{F}_2\delta_2} + g_2\delta_2 \\
&\vdots \\
E\|\boldsymbol{\rho}_{k-1}\|^2_{\delta_{k-1}} &= E\|\boldsymbol{\rho}_{k-2}\|^2_{\mathrm{F}_{k-1}\delta_{k-1}} + g_{k-1}\delta_{k-1} \qquad (2.3.26) \\
E\|\boldsymbol{\rho}_k\|^2_{\delta_k} &= E\|\boldsymbol{\rho}_{k-1}\|^2_{\mathrm{F}_k\delta_k} + g_k\delta_k \qquad (2.3.27) \\
&\vdots \\
E\|\boldsymbol{\rho}_N\|^2_{\delta_N} &= E\|\boldsymbol{\rho}_{N-1}\|^2_{\mathrm{F}_N\delta_N} + g_N\delta_N
\end{aligned}$$

with $g_k = \mu_k^2\sigma_{v,k}^2 r_k^{\mathrm{T}}$. By choosing the free parameters $\{\delta_k, \delta_{k-1}\}$ such that $\delta_{k-1} = F_k\delta_k$, (2.3.26) and (2.3.27) are combined to obtain

$$\begin{aligned}
E\|\boldsymbol{\rho}_k\|^2_{\delta_k} &= E\|\boldsymbol{\rho}_{k-1}\|^2_{\delta_{k-1}} + g_k\delta_k \\
&= E\|\boldsymbol{\rho}_{k-2}\|^2_{F_{k-1}F_k\delta_k} + g_{k-1}F_k\delta_k + g_k\delta_k. \qquad (2.3.28)
\end{aligned}$$

Next, iterate across the cycle to arrive at

$$\begin{aligned}
E\|\boldsymbol{\rho}_{k-1}\|^2_{\delta_{k-1}} = E\|\boldsymbol{\rho}_{k-1}\|^2_{\mathrm{F}_k\cdots\mathrm{F}_N\mathrm{F}_1\cdots\mathrm{F}_{k-1}\delta_{k-1}} \\
+g_k\mathrm{F}_{k+1}\cdots\mathrm{F}_N\mathrm{F}_1\cdots\mathrm{F}_{k-1}\delta_{k-1} \\
+g_{k+1}\mathrm{F}_{k+2}\cdots\mathrm{F}_N\mathrm{F}_1\cdots\mathrm{F}_{k-1}\delta_{k-1} \\
\cdots + g_{k-2}\mathrm{F}_{k-1}\delta_{k-1} + g_{k-1}\delta_{k-1}. \qquad (2.3.29)
\end{aligned}$$

and let

$$\Pi_{k-1,l} = \mathrm{F}_{k+l-1} + \cdots \mathrm{F}_N\mathrm{F}_1\cdots\mathrm{F}_{k-1}, \;\; l = 1, 2, \ldots, N \qquad (2.3.30)$$

$$a_{k-1} = g_k\Pi_{k-1,2} + \cdots + g_{k-2}\Pi_{k-1,N} + g_{k-1} \qquad (2.3.31)$$

so that

$$\boxed{E\|\boldsymbol{\rho}_{k-1}\|^2_{(I-\Pi_{k-1,1})\delta_{k-1}} = a_{k-1}\delta_{k-1}}. \tag{2.3.32}$$

Expression (2.3.32) can be exploited to evaluate the desired performance measures at node k, as follows:

$$\eta_k = E\|\boldsymbol{\rho}_{k-1}\|^2_q, \quad q = \text{vec}\{I\} \quad \text{(MSD)} \tag{2.3.33}$$

$$\zeta_k = E\|\boldsymbol{\rho}_{k-1}\|^2_{r_k}, \quad r_k = \text{vec}\{R_{u,k}\} \quad \text{(EMSE)} \tag{2.3.34}$$

$$\xi_k = \zeta_k + \sigma^2_{v,k} \quad \text{(MSE)}. \tag{2.3.35}$$

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

$$\eta_k = a_{k-1}(I - \Pi_{k-1,1})^{-1}q \quad \text{(MSD)} \tag{2.3.36}$$

$$\zeta_k = a_{k-1}(I - \Pi_{k-1,1})^{-1}r_k \quad \text{(EMSE)} \tag{2.3.37}$$

$$\xi_k = \zeta_k + \sigma^2_{v,k} \quad \text{(MSE)}. \tag{2.3.38}$$

These expressions are next verified by simulation study.

2.4 Simulations

In this section, the theoretical performance is compared with that obtained from computer simulations in a system identification scenario. All simulation 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. Consider a network with 20 nodes seeking an unknown filter with $M = 10$ taps, coefficients of which are selected randomly. A quasi-uniformly distributed signal is

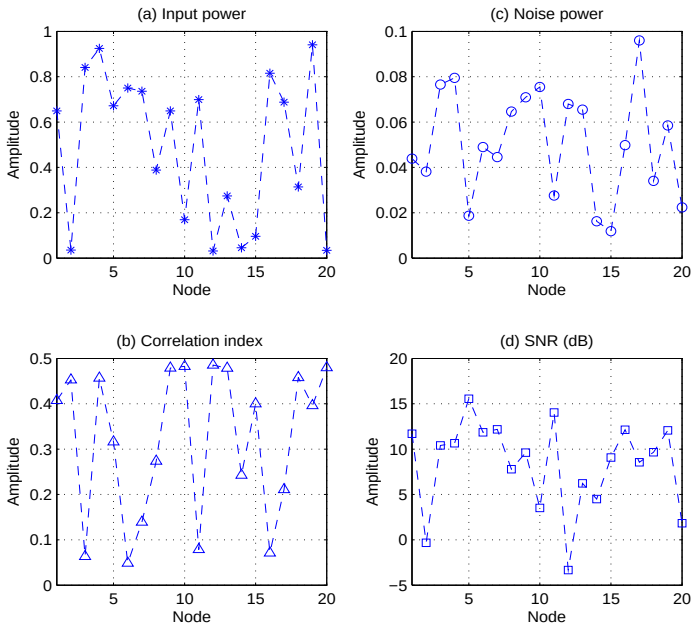

Figure 2.3. Statistical property per node over the quasi-uniform data network.

used to generate the inputs at each node k according to the recursion

$$u_k(i) = \alpha_k u_k(i-1) + \beta_k \cdot \tau_k(i). \tag{2.4.1}$$

Expression (2.4.1) describes a first-order AR process with a pole at α_k; $\tau_k(i)$ is a white, zero-mean, uniformly distributed random sequence with unity variance, $\alpha_k \in (0, 0.5]$ and $\beta_k = \sqrt{\sigma_{u,k}^2 \cdot (1-\alpha_k^2)}$. In this way, the covariance matrix $R_{u,k}$ of the regressor $\mathbf{u}_{k,i}$ is a 10×10 Toeplitz matrix with entries $r_k(m) = \sigma_{u,k}^2 \alpha_k^{|m|}$, $m = 0, \ldots M-1$ with $\sigma_{u,k}^2 \in (0, 1]$. One should note that (2.4.1) can be transformed approximately to a finite impulse response (FIR) process

$$u_k(i) = \sum_{j=0}^{n} a_k(i-j)\tau_k(i-j). \tag{2.4.2}$$

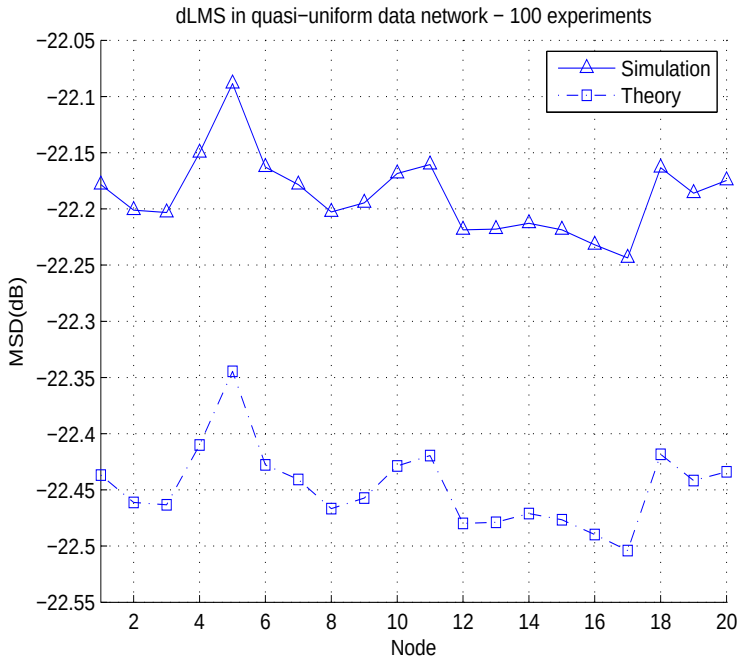

Figure 2.4. Steady-state MSD using $\mu_k = 0.02$ per node for quasi-uniformly distributed inputs.

where $\{a_k(i-j)\}$ are calculated parameters and n denotes the order of the delayed inputs. According to the central limit theorem [35], as $n \to \infty$, the distribution of $\mathbf{u}_k$ will approximate Gaussian despite the uniformly distributed sequence $\boldsymbol{\tau}_k$. As a consequence, the choice of $\alpha_k \in (0, 0.5]$ allows the FIR process transformed from (2.4.1) to likely have a limited number of n. From a strict definition, the distribution of $\mathbf{u}_k$ in (2.4.1) is not Gaussian or uniform. Due to simplification, $\mathbf{u}_k$ generated by (2.4.1) is termed as correlated quasi-uniform data. The background noise per node is generated from Gaussian distributed data with variance $\sigma^2_{v,k} \in (0, 0.1]$. These statistical profiles for uniformly distributed data are illustrated in Figure 2.3.

Comparison curves between the theoretical and simulated results are illustrated in Figures 2.4-2.9. Figures 2.4-2.6 illustrate the steady-

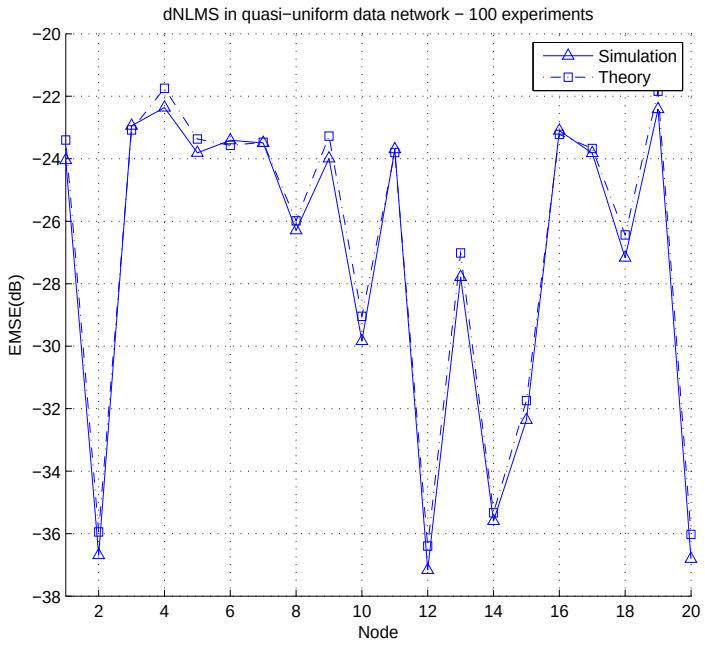

Figure 2.5. Steady-state EMSE using $\mu_k = 0.02$ per node for quasi-uniformly distributed inputs.

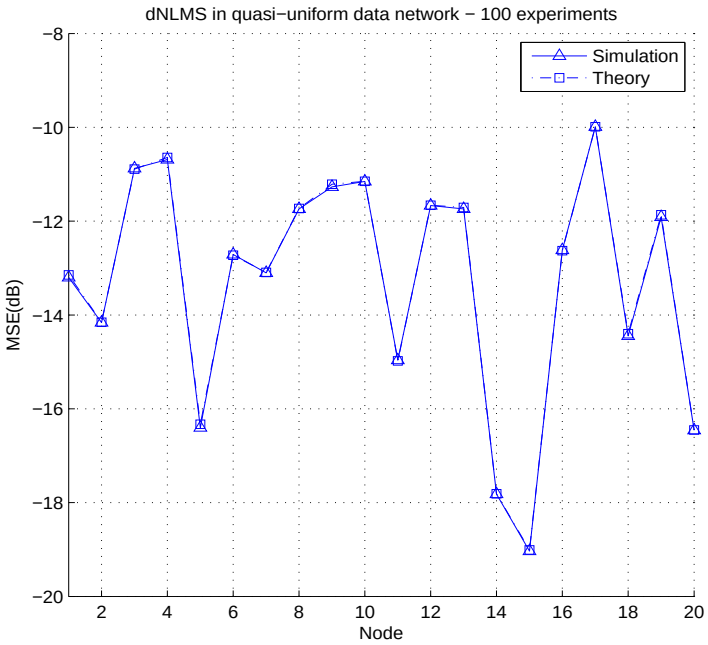

Figure 2.6. Steady-state MSE using $\mu_k = 0.02$ per node for quasi-uniformly distributed inputs.

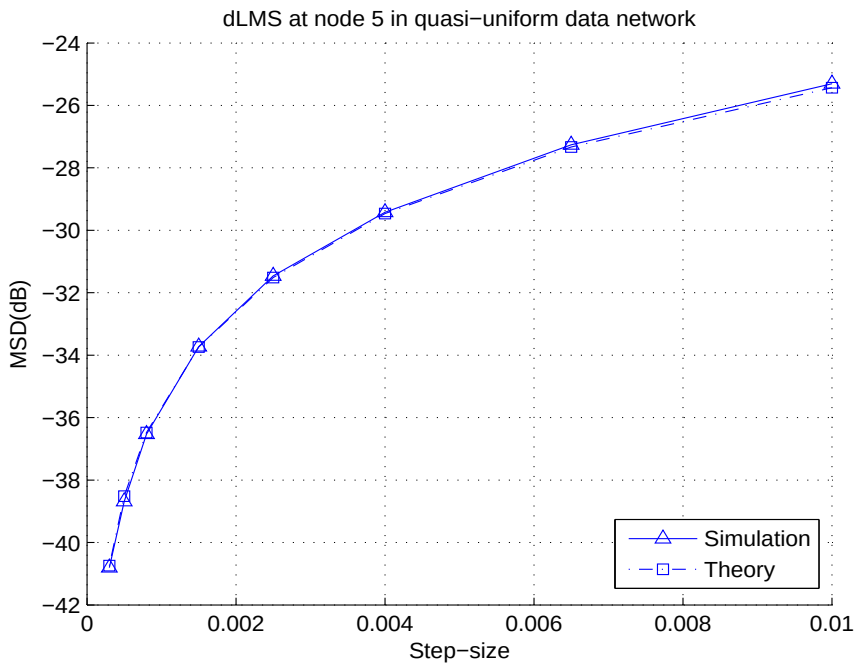

Figure 2.7. Steady-state MSD versus μ_k for quasi-uniformly distribution data at node 5.

state performance of dLMS using $\mu_k = 0.02$. For the steady-state MSD, both the theoretical and simulation curves shown in Figure 2.4 are almost flat through the network, namely, the simulated MSD fluctuates at approximately -22.20dB with ± 0.12dB and the theoretical MSD at approximately -22.40dB with ± 0.12dB. One can see in Figures 2.5-2.6 that the steady-state EMSE and MSE are more sensitive to the node statistics and the theoretical results for the steady-state MSE match well with the simulated results. The small differences are a consequence of independence assumption **A2** but these differences reduce in amplitude as the step-size reduces. Even though there are some small discrepancies for the steady-state EMSE between simulation and theory at certain nodes, but these discrepancies can be reduced by decreasing the step-size as shown in Figure 2.8.

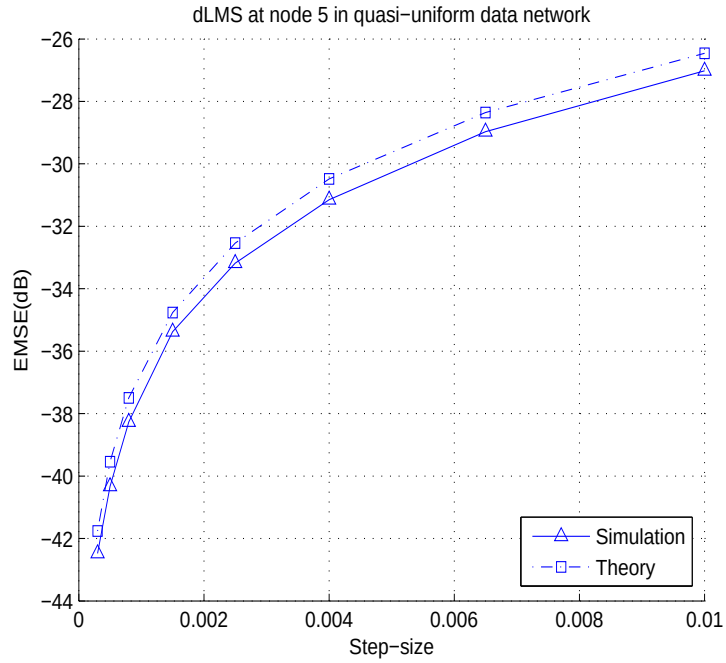

Figure 2.8. Steady-state EMSE versus μ_k for quasi-uniformly distribution data at node 5.

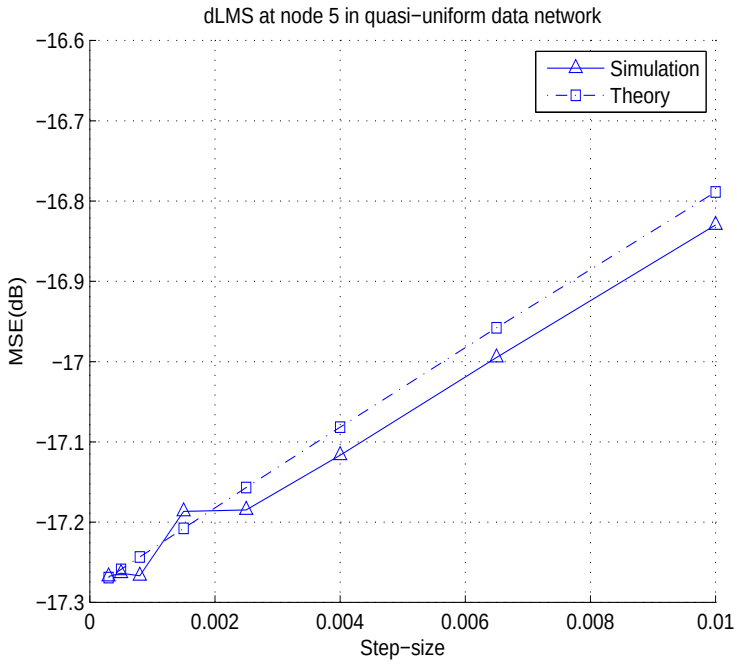

Figure 2.9. Steady-state MSE versus μ_k for quasi-uniformly distribution data at node 5.

Figures 2.7-2.9 illustrate the steady-state performance of node 5 within the range of the step-size [0.0008, 0.01]. One can clearly see that as the step-size decreases the discrepancy between theoretical and simulated results decreases. This is due to the simplifying assumptions used in the analysis, which will be degraded by large step-sizes. For the steady-state MSD and MSE, there is a good match between theory and practice shown in Figure 2.7 and Figure 2.9. Although it is clearly shown in Figure 2.8 that EMSE almost has the same discrepancy in dB for the different step-sizes, one can find that a decrease of the step-size leads to a decrease of the discrepancy in absolute value. This simulation study confirms the theoretical expressions (2.3.36)-(2.3.38) for the non-Gaussian wide sense stationary case.

2.5 Conclusions

In this chapter, the steady-state mean-square performance evaluation of dLMS is carried out under the assumptions **A1** and **A2** for the non-Gaussian wide sense stationary case. Using weighted spatial-temporal energy conservation arguments, the key contribution of this chapter is obtained, namely, to derive expressions for the steady-state MSD, EMSE, MSE without restricting the distribution of the inputs. This work is a useful extension for the dLMS study. However, different learning rules can be applied in the context of a distributed network with incremental topology. Chapter 3 therefore illustrates a novel incremental algorithm based on the APA algorithm.

Chapter 3

DISTRIBUTED ESTIMATION OVER AN ADAPTIVE INCREMENTAL NETWORK BASED ON THE APA ALGORITHM

3.1 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 to obtain significant reduction in the amount of processing and communications between nodes within a network, relying only on local data exchange and interactions between immediate neighbourhood nodes, whilst retaining the estimation accuracy of a centralized solution [8], [9]. Distributed solutions which exploit consensus implementation presented in [36], [37], [38] 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 [1], [21], [26], [27] have therefore focused on incremental learning over a distributed network, which has 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 obtain 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, the network size is therefore assumed to be sufficiently small, typically less than one hundred, so that incremental schemes can be used. In the next chapter, adaptive algorithms of the diffusion type will be studied, which removes the requirement of a Hamiltonian cycle at the expense of a slightly reduced mean-square performance [2]. The incremental approach in this chapter 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. 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 coloured input signals and the APA algorithm is a better alterna-

tive to LMS in such an environment [39], [40], [41]. For distributed networks, highly correlated inputs also deteriorate the performance of the dLMS algorithm. This chapter therefore describes a new APA-based distributed learning scheme for an incremental network. This distributed solution is proposed to obtain a good compromise between convergence performance and computational cost, rather than considering a precise application. A key contribution of this chapter is using the weighted spatial-temporal energy conservation relation to reveal the nature of the energy flow through the network and 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.

3.2 Estimation problem

Firstly, consider an N node network, as in [1], where each node has a separate random complex valued desired response $\mathbf{d}_k$ with zero-mean and a $1 \times M$ spatially distinct row input vector $\mathbf{u}_k$ with zero-mean and random elements; $\mathbf{d}_k$ and $\mathbf{u}_k$ are jointly wide-sense stationary, and k indicates the node index. Each node obtains time realizations $\{d_k(i), u_{k,i}\}$ of $\{\mathbf{d}_k, \mathbf{u}_k\}$, where i denotes the discrete time index. Thus, the linear space-time least mean square estimation problem is formed as:

$$\min_{w} J(w) \quad \text{and} \quad J(w) = E\|\mathbf{d} - \mathbf{U}w\|^2 \tag{3.2.1}$$

where the global desired response vector and input matrix are

$$\mathbf{d}=\text{col}\{\mathbf{d}_1,\, \mathbf{d}_2,\, \ldots,\, \mathbf{d}_N\}, \quad (N \times 1) \tag{3.2.2}$$

$$\mathbf{U}=\text{col}\{\mathbf{u}_1,\, \mathbf{u}_2,\, \ldots,\, \mathbf{u}_N\}, \quad (N \times M). \tag{3.2.3}$$

and the data are obtained from N spatially distinct nodes and exploited to estimate the $M \times 1$ vector w. Thus, the optimal MMSE solution w^o is calculated, for which the normal equations [6] are satisfied,

$$R_{du} = R_u w^o \tag{3.2.4}$$

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

When a multitude of nodes in the network have access to data, in order to take advantage of node cooperation, an incremental adaptive method is introduced to a distributed network, where at least one cyclic path can be established around 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 Figure 3.1). As in [1], the cost function can be decomposed into

$$J(w) = \sum_{k=1}^{N} J_k(w) \quad \text{and} \quad J_k(w) = E|\mathbf{d}_k - \mathbf{u}_k w|^2. \tag{3.2.5}$$

In previous works [1], [21], [26], [27], 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. The contribution of this chapter aims at improving upon the convergence performance of the dLMS algorithm

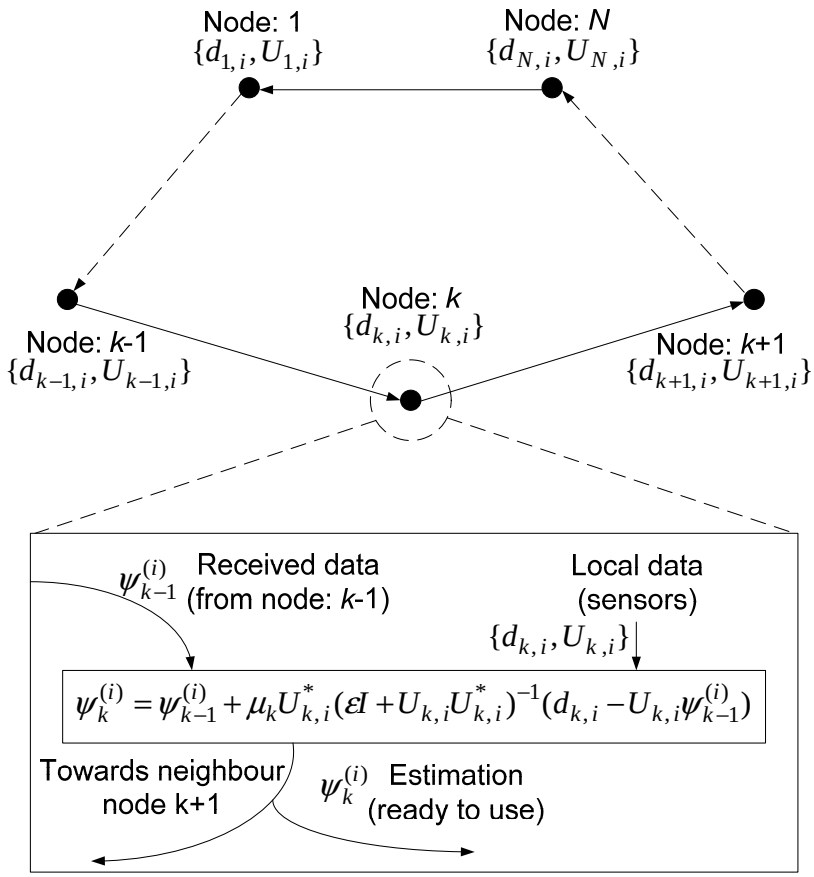

Figure 3.1. Data processing of the dAPA algorithm in an incremental network.

with coloured input signals whilst reducing the complexity of the Lc-dRLS algorithm and dRLS algorithm of [27].

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 regularized Newton's search based approach to solving (3.2.5) for incremental learning within a distributed

network. The optimal tap weight w^o is estimated via [6]

$$\begin{aligned} w_i &= w_{i-1} - \sum_{k=1}^{N} \mu_k [\epsilon I + \nabla^2 J_k(\psi_{k-1}^{(i)})]^{-1} [\nabla J_k(\psi_{k-1}^{(i)})]^* \\ &= w_{i-1} + \sum_{k=1}^{N} \mu_k (\epsilon I + R_{u,k})^{-1} [R_{du,k} - R_{u,k}\psi_{k-1}^{(i)}] \end{aligned} \tag{3.2.6}$$

where $R_{u,k} = E\mathbf{u}_k^*\mathbf{u}_k$, $R_{du,k} = E\mathbf{d}_k\mathbf{u}_k^*$ and ϵ denotes a regularization parameter with small positive value. The parameter μ_k indicates an appropriately chosen step-size, which is evaluated in Section 3.3.5, and the scheme is initialized with an $M \times 1$ vector $w_{-1} = \text{col}\{0, \ldots, 0\}$.

For a practical scheme to realize (3.2.6), and utilizing the correlation of the input signal at each node, $\{R_{u,k}, R_{du,k}\}$ is replaced by the following sample sliding-window estimates,

$$\begin{aligned} \hat{R}_{u,k} &= \frac{1}{T} \sum_{j=i-T+1}^{i} u_{k,j}^* u_{k,j} \\ \hat{R}_{du,k} &= \frac{1}{T} \sum_{j=i-T+1}^{i} u_{k,j}^* d_k(j) \end{aligned} \tag{3.2.7}$$

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 k-th node. Hence, using the matrix inversion formula, recursion (3.2.6) becomes,

$$w_i = w_{i-1} + \sum_{k=1}^{N} \mu_k U_{k,i}^* (\epsilon I + U_{k,i} U_{k,i}^*)^{-1} [d_{k,i} - U_{k,i}\psi_{k-1}^{(i)}] \tag{3.2.8}$$

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

$$U_{k,i} = \begin{pmatrix} u_{k,i} \\ u_{k,i-1} \\ \vdots \\ u_{k,i-T+1} \end{pmatrix}, \; d_{k,i} = \begin{pmatrix} d_k(i) \\ d_k(i-1) \\ \vdots \\ d_k(i-T+1) \end{pmatrix} \tag{3.2.9}$$

and ϵ is employed to avoid the inversion of a rank deficient matrix $U_{k,i}U_{k,i}^*$. As such, recursion (3.2.9) is the dAPA learning algorithm in an incremental network, the operation of which is shown in Figure 3.1. At each time instant i, each node utilizes local data $\{d_{k,i}, U_{k,i}\}$ and $\psi_{k-1}^{(i)}$ is 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 Figure 3.1 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 3.1. In addition, dAPA has intermediate computational and memory cost between dLMS and dRLS, for certain regressor lengths, which is verified in Appendix A.

3.3 Performance analysis

The convergence behaviours of classical APA-based algorithms are studied in [41], [42], [43] exploiting arguments based on a single adaptive filter. In order to study the performance of the dAPA algorithm, the weighted energy conservation approach for the APA-based algorithms of [42], [43] is extended to the case of a distributed incremental network,

Table 3.1. Pseudo-code implementation of dAPA.

For each time instant $i \geq 0$ repeat:
$\psi_0^{(i)} = w_{i-1}$
k=1, ... N
$\psi_k^{(i)} = \psi_{k-1}^{(i)} + \mu_k U_{k,i}^*(\epsilon I + U_{k,i}U_{k,i}^*)^{-1}[d_{k,i} - U_{k,i}\psi_{k-1}^{(i)}]$
end
$w_i = \psi_N^{(i)}$
where $\{d_{k,i}, U_{k,i}\}$ are defined by (3.2.9).

which involves both the space and the time dimensions. 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 the transient and steady-state performances at each individual node are evaluated in terms of MSD, EMSE and MSE. The closed-form expressions for the theoretical results are formed under some simplifying assumptions described below.

3.3.1 Data model and assumption

As defined earlier, boldface letters are used as the random quantities and the same model as in the previous chapter is assumed 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:

$$\mathbf{d}_k(i) = \mathbf{u}_{k,i}w^o + \mathbf{v}_k(i) \tag{3.3.1}$$

where $\mathbf{v}_k(i)$ is a temporally and spatially independent noise sequence with variance $\sigma_{v,k}^2$ 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 $i \neq j$

It is highlighted that assumption **A2** is an extension of that made for time-only adaptive filtering to space-time filtering. The spatial independence assumption is generally more likely to hold than the temporal independence assumption, as a consequence of the different locations of the nodes. The temporal independence assumption is necessary for analysis purpose and is commonly used to provide useful performance measures in adaptive signal processing [7]. In terms of analysis in this chapter, only the stationary case is studied, where the system vector is fixed and the 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 [3], [4], [44]).

3.3.2 Weighted spatial-temporal energy conservation relation

Using the following error vectors:

$$\tilde{\boldsymbol{\psi}}_k^{(i)} \triangleq w^o - \boldsymbol{\psi}_k^{(i)} \qquad \text{(weight} - \text{error vector)} \tag{3.3.2}$$

$$\mathbf{e}_{k,i} \triangleq \mathbf{d}_{k,i} - \mathbf{U}_{k,i}\boldsymbol{\psi}_{k-1}^{(i)} \quad \text{(error vector)} \tag{3.3.3}$$

to represent the update tap weights expression in dAPA, the relation expression of $\{\tilde{\boldsymbol{\psi}}_k^{(i)}, \tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\}$ is therefore obtained,

$$\tilde{\boldsymbol{\psi}}_k^{(i)} = \tilde{\boldsymbol{\psi}}_{k-1}^{(i)} - \mu_k \mathbf{U}_{k,i}^*(\epsilon I + \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*)^{-1}\mathbf{e}_{k,i}. \tag{3.3.4}$$

Multiplying both sides of (3.3.4) by $\mathbf{U}_{k,i}$ from the left, equation (3.3.4) becomes

$$\mathbf{U}_{k,i}\tilde{\boldsymbol{\psi}}_k^{(i)} = \mathbf{U}_{k,i}\tilde{\boldsymbol{\psi}}_{k-1}^{(i)} - \mu_k \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*(\epsilon I + \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*)^{-1}\mathbf{e}_{k,i}. \tag{3.3.5}$$

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, expression (3.3.5) becomes,

$$\mathbf{e}_{p,k}^{(i)} = \mathbf{e}_{a,k}^{(i)} - \mu_k \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*(\epsilon I + \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*)^{-1}\mathbf{e}_{k,i}. \tag{3.3.6}$$

Note that the error vector is given by

$$\mathbf{e}_{k,i} = \mathbf{U}_{k,i}\tilde{\boldsymbol{\psi}}_{k-1}^{(i)} + \mathbf{v}_{k,i} = \mathbf{e}_{a,k}^{(i)} + \mathbf{v}_{k,i} \tag{3.3.7}$$

where

$$\mathbf{v}_{k,i} = \mathrm{col}\{\mathbf{v}_k(i), \mathbf{v}_k(i-1), \ldots, \mathbf{v}_k(i-T+1)\}. \tag{3.3.8}$$

The following performance measures at each node k must be evaluated:

$$\eta_k(i) \triangleq E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_I^2 \qquad \text{(MSD)} \tag{3.3.9}$$

$$\zeta_k(i) \triangleq E|\mathbf{u}_{k,i}\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}|^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{R_{u,k}}^2 \qquad \text{(EMSE)} \tag{3.3.10}$$

$$\xi_k(i) \triangleq \zeta_k(i) + \sigma_{v,k}^2 \qquad \text{(MSE)} \tag{3.3.11}$$

where under the assumed data conditions the weighted norm notation $\|x\|_\Sigma^2 \triangleq x^*\Sigma x$ is introduced with a vector x and a Hermitian positive

definite matrix $\Sigma > 0$. In order to study the performance behaviour of the dAPA algorithm for incremental networks, the method of weighted energy conservation described in [6], [1] is used in this section. As a consequence, the weighted a posteriori and a priori error vectors at node k are defined by,

$$\mathbf{e}_{p,k}^{\Sigma_k,(i)} \triangleq \mathbf{U}_{k,i}\Sigma_k\tilde{\boldsymbol{\psi}}_k^{(i)} \tag{3.3.12}$$

$$\mathbf{e}_{a,k}^{\Sigma_k,(i)} \triangleq \mathbf{U}_{k,i}\Sigma_k\tilde{\boldsymbol{\psi}}_{k-1}^{(i)} \tag{3.3.13}$$

where Σ_k is a Hermitian positive-definite weighting matrix and is free to choose for each node k. The weighted definitions (3.3.12) and (3.3.13) are used to expand (3.3.6) in terms of weighted error vectors and the regressor data as follows:

$$\mathbf{e}_{p,k}^{\Sigma_k,(i)} = \mathbf{e}_{a,k}^{\Sigma_k,(i)} - \mu_k\mathbf{U}_{k,i}\Sigma_k\mathbf{U}_{k,i}^*(\epsilon I + \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*)^{-1}\mathbf{e}_{k,i} \tag{3.3.14}$$

If the special case $\Sigma_k = I$ is chosen, equation (3.3.14) is simplified to (3.3.6). With the assumption that $\mathbf{U}_{k,i}\Sigma_k\mathbf{U}_{k,i}^*$ is invertible, (3.3.14) can be used to replace $\mathbf{e}_{k,i}$ in (3.3.4). After rearrangement, the expression

$$\begin{aligned}\tilde{\boldsymbol{\psi}}_k^{(i)} + \mathbf{U}_{k,i}^*(\mathbf{U}_{k,i}\Sigma_k\mathbf{U}_{k,i}^*)^{-1}\mathbf{e}_{a,k}^{\Sigma_k,(i)} = \\ \tilde{\boldsymbol{\psi}}_{k-1}^{(i)} + \mathbf{U}_{k,i}^*(\mathbf{U}_{k,i}\Sigma_k\mathbf{U}_{k,i}^*)^{-1}\mathbf{e}_{p,k}^{\Sigma_k,(i)}\end{aligned} \tag{3.3.15}$$

is obtained. If the weighted energies of both sides of (3.3.15) are equated, the space-time version of the weighted energy-conservation

relation for dAPA is established:

$$\begin{aligned}\|\tilde{\boldsymbol{\psi}}_k^{(i)}\|_{\Sigma_k}^2 + \mathbf{e}_{a,k}^{\Sigma_k,(i)*}(\mathbf{U}_{k,i}\Sigma_k\mathbf{U}_{k,i}^*)^{-1}\mathbf{e}_{a,k}^{\Sigma_k,(i)} \\ = \|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{\Sigma_k}^2 + \mathbf{e}_{p,k}^{\Sigma_k,(i)*}(\mathbf{U}_{k,i}\Sigma_k\mathbf{U}_{k,i}^*)^{-1}\mathbf{e}_{p,k}^{\Sigma_k,(i)}\end{aligned} \tag{3.3.16}$$

Then, substituting (3.3.14) into (3.3.16) and rearranging the result, (3.3.16) becomes

$$\begin{aligned}\|\tilde{\boldsymbol{\psi}}_k^{(i)}\|_{\Sigma_k}^2 = \|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{\Sigma_k}^2 - \mu_k[\mathbf{e}_{a,k}^{\Sigma_k,(i)*}\mathbf{A}_k\mathbf{e}_{k,i}] \\ - \mu_k[\mathbf{e}_{k,i}^*\mathbf{A}_k\mathbf{e}_{a,k}^{\Sigma_k,(i)*}] + \mu_k^2[\mathbf{e}_{k,i}^*\mathbf{B}_k\mathbf{e}_{k,i}]\end{aligned} \tag{3.3.17}$$

with

$$\mathbf{A}_k = (\epsilon I + \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*)^{-1} \tag{3.3.18}$$

$$\mathbf{B}_k^{\Sigma_k} = (\epsilon I + \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*)^{-1}\mathbf{U}_{k,i}\Sigma_k\mathbf{U}_{k,i}^*(\epsilon I + \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*)^{-1}. \tag{3.3.19}$$

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 (3.3.17) becomes

$$E\|\tilde{\boldsymbol{\psi}}_k^{(i)}\|_{\Sigma_k}^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{\boldsymbol{\Sigma}_k'}^2 + \mu_k^2 E[\mathbf{v}_{k,i}^*\mathbf{B}_k^{\Sigma_k}\mathbf{v}_{k,i}] \tag{3.3.20}$$

where $\boldsymbol{\Sigma}_k'$ is a stochastic weighting matrix

$$\boldsymbol{\Sigma}_k' = \Sigma_k - \mu_k\Sigma_k\mathbf{C}_k - \mu_k\mathbf{C}_k\Sigma_k + \mu_k^2\mathbf{D}_k^{\Sigma_k} \tag{3.3.21}$$

with

$$\mathbf{C}_k = \mathbf{U}_{k,i}^*\mathbf{A}_k\mathbf{U}_{k,i} \quad \text{and} \quad \mathbf{D}_k^{\Sigma_k} = \mathbf{U}_{k,i}^*\mathbf{B}_k^{\Sigma_k}\mathbf{U}_{k,i}. \tag{3.3.22}$$

Equation (3.3.20) is difficult to evaluate due to the dependence of $\boldsymbol{\psi}_{k-1}^{(i)}$ on previous regressors of $\mathbf{U}_{k,i}$ and $\boldsymbol{\Sigma}_k'$ on $\mathbf{U}_{k,i}$. In order to resolve this problem in terms of analysis, an independence assumption is introduced 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 $\boldsymbol{\Sigma}_k'$. As compared to **A2**, **A3** is a strong assumption. However, a weaker assumption can be made:

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 (3.3.21) for $\boldsymbol{\Sigma}_k'$ to satisfy the requirement. It is highlighted that these are only assumptions necessary to facilitate analysis similar to those in [42], [43] used for the original non-distributed APA algorithm, but such assumptions will be verified by simulations to yield useful performance measures when the step-size is sufficiently small. For compactness of notation, the index i is dropped. Using this assumption, the expectation $E\|\tilde{\boldsymbol{\psi}}_{k-1}\|_{\boldsymbol{\Sigma}_k'}^2$ is separated into

$$\boxed{E\|\tilde{\boldsymbol{\psi}}_{k-1}\|_{\boldsymbol{\Sigma}_k'}^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}\|_{E\boldsymbol{\Sigma}_k'}^2} \tag{3.3.23}$$

where the mean of the weighted matrix $\boldsymbol{\Sigma}_k'$ is given by $E\boldsymbol{\Sigma}_k' = \Sigma_k'$:

$$\boxed{\Sigma_k' = \Sigma_k - \mu_k\Sigma_k E\mathbf{C}_k - \mu_k E\mathbf{C}_k \cdot \Sigma_k + \mu_k^2 E\mathbf{D}_k^{\Sigma_k}} \tag{3.3.24}$$

and Σ_k' is now a deterministic matrix. In this manner, expression (3.3.20) is replaced by

$$E\|\tilde{\boldsymbol{\psi}}_k\|_{\Sigma_k}^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}\|_{\Sigma_k'}^2 + \mu_k^2 E[\mathbf{v}_k^*\mathbf{B}_k^{\Sigma_k}\mathbf{v}_k]. \tag{3.3.25}$$

For studying the behaviour of the distributed learning algorithm, the following three moments must be evaluated:

$$E\mathbf{C}_k = E[\mathbf{U}_k^* \mathbf{A}_k \mathbf{U}_k] \tag{3.3.26}$$

$$E\mathbf{D}_k^{\Sigma_k} = E[\mathbf{U}_k^* \mathbf{A}_k \mathbf{U}_k \Sigma_k \mathbf{U}_k^* \mathbf{A}_k \mathbf{U}_k] \tag{3.3.27}$$

$$E[\mathbf{v}_k^* \mathbf{B}_k^{\Sigma_k} \mathbf{v}_k] = E[\mathbf{v}_k^* \mathbf{A}_k \mathbf{U}_k \Sigma_k \mathbf{U}_k^* \mathbf{A}_k \mathbf{v}_k]. \tag{3.3.28}$$

The terms in (3.3.27) and (3.3.28) are difficult to calculate, even the eigendecomposition and diagonalization methods used for Gaussian data in [1] are not available to express (3.3.24) and (3.3.25) 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 side of expressions (3.3.27) and (3.3.28). This is achieved by vectorization and exploiting the property of Kronecker products.

3.3.3 Weighted variance relation

With the purpose to evaluate $E\|\tilde{\boldsymbol{\psi}}_k\|_{\Sigma_k}^2$, $M^2 \times 1$ column vectors as in Chapter 2 are introduced:

$$\sigma_k = \text{vec}\{\Sigma_k\}, \quad \sigma_k' = \text{vec}\{\Sigma_k'\}. \tag{3.3.29}$$

When working with Kronecker products, the vec$\{\cdot\}$ notation has the following property,

$$\text{vec}\{P\Sigma Q\} = (Q^{\mathrm{T}} \otimes P)\text{vec}\{\Sigma\} \tag{3.3.30}$$

where matrices $\{P, \Sigma, Q\}$ have compatible dimensions.

By applying (3.3.30) to express some items in (3.3.24), the following expressions are obtained

$$\text{vec}\{\Sigma_k E\mathbf{C}_k\} = (E\mathbf{C}_k^{\mathrm{T}} \otimes I)\sigma_k \tag{3.3.31}$$

$$\text{vec}\{E\mathbf{C}_k \cdot \Sigma_k\} = (I \otimes E\mathbf{C}_k)\sigma_k \tag{3.3.32}$$

$$\text{vec}\{E\mathbf{D}_k^{\Sigma_k}\} = E[\mathbf{C}_k^{\mathrm{T}} \otimes \mathbf{C}_k]\sigma_k. \tag{3.3.33}$$

Therefore, a linear relation between the corresponding vectors $\{\sigma_k', \sigma_k\}$ is formed, namely

$$\sigma_k' = \mathrm{F}_k \sigma_k \tag{3.3.34}$$

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

$$\boxed{\mathrm{F}_k \triangleq I - \mu_k(E\mathbf{C}_k^{\mathrm{T}} \otimes I) - \mu_k(I \otimes E\mathbf{C}_k) + \mu_k^2 E[\mathbf{C}_k^{\mathrm{T}} \otimes \mathbf{C}_k]}. \tag{3.3.35}$$

As a result, expression (3.3.25) becomes

$$E\|\tilde{\boldsymbol{\psi}}_k\|^2_{\text{vec}\{\sigma_k\}} = E\|\tilde{\boldsymbol{\psi}}_{k-1}\|^2_{\text{vec}\{\mathrm{F}_k\sigma_k\}} + \mu_k^2 E[\mathbf{v}_k^* \mathbf{B}_k^{\Sigma_k} \mathbf{v}_k]. \tag{3.3.36}$$

For the sake of clarity, the time index i is reintroduced but the $\text{vec}\{\cdot\}$ notation is dropped from the subscripts in (3.3.36) for compactness. Expression (3.3.36) becomes

$$E\|\tilde{\boldsymbol{\psi}}_k^{(i)}\|^2_{\sigma_k} = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|^2_{\mathrm{F}_k\sigma_k} + \mu_k^2 E[\mathbf{v}_{k,i}^* \mathbf{B}_k^{\Sigma_k} \mathbf{v}_{k,i}] \tag{3.3.37}$$

Due to the assumption that $\mathbf{v}_{k,i}$ is independent from $\mathbf{U}_{k,i}$, the last item

in (3.3.37) can be written as

$$\begin{aligned}\mu_k^2 E[\mathbf{v}_{k,i}^* \mathbf{B}_k^{\Sigma_k} \mathbf{v}_{k,i}] &= \mu_k^2 \delta_{v,k}^2 Tr\{E\boldsymbol{\Phi}_k \cdot \Sigma_k\} \\ &= \mu_k^2 \sigma_{v,k}^2 \gamma_k^{\mathrm{T}} \sigma_k \end{aligned} \tag{3.3.38}$$

with $\boldsymbol{\Phi}_k = \mathbf{U}_{k,i}^*(\epsilon I + \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*)^{-2}\mathbf{U}_{k,i}$ and $\gamma_k = \mathrm{vec}\{E\boldsymbol{\Phi}_k\}$. Therefore, expression (3.3.37) can be written as

$$\boxed{E\|\tilde{\boldsymbol{\psi}}_k^{(i)}\|_{\sigma_k}^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{\mathrm{F}_k\sigma_k}^2 + \mu_k^2\sigma_{v,k}^2\gamma_k^{\mathrm{T}}\sigma_k}. \tag{3.3.39}$$

3.3.4 Learning curves

Expression (3.3.39) 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 allows this problem to be resolved. By iterating (3.3.39), N coupled equalities are obtained,

$$\begin{aligned} E\|\tilde{\boldsymbol{\psi}}_1^{(i)}\|_{\sigma_1}^2 &= E\|\tilde{\boldsymbol{\psi}}_N^{(i-1)}\|_{\mathrm{F}_1\sigma_1}^2 + g_1\sigma_1 \\ E\|\tilde{\boldsymbol{\psi}}_2^{(i)}\|_{\sigma_2}^2 &= E\|\tilde{\boldsymbol{\psi}}_1^{(i)}\|_{\mathrm{F}_2\sigma_2}^2 + g_2\sigma_2 \\ &\vdots \\ E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{\sigma_{k-1}}^2 &= E\|\tilde{\boldsymbol{\psi}}_{k-2}^{(i)}\|_{\mathrm{F}_{k-1}\sigma_{k-1}}^2 + g_{k-1}\sigma_{k-1} \qquad (3.3.40) \\ E\|\tilde{\boldsymbol{\psi}}_k^{(i)}\|_{\sigma_k}^2 &= E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{\mathrm{F}_k\sigma_k}^2 + g_k\sigma_k \qquad (3.3.41) \\ &\vdots \\ E\|\tilde{\boldsymbol{\psi}}_N^{(i)}\|_{\sigma_N}^2 &= E\|\tilde{\boldsymbol{\psi}}_{N-1}^{(i)}\|_{\mathrm{F}_N\sigma_N}^2 + g_N\sigma_N \end{aligned}$$

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

is adopted and (3.3.40) and (3.3.41) are thereby combined to obtain

$$\begin{aligned} E\|\tilde{\boldsymbol{\psi}}_k^{(i)}\|_{\sigma_k}^2 &= E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{\sigma_{k-1}}^2 + g_k\sigma_k \\ &= E\|\tilde{\boldsymbol{\psi}}_{k-2}^{(i)}\|_{\mathrm{F}_{k-1}\mathrm{F}_k\sigma_k}^2 + g_{k-1}\mathrm{F}_k\sigma_k + g_k\sigma_k \end{aligned} \tag{3.3.42}$$

where F_k is as in (3.3.35) 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 [7]. Iterating in this way, an equality involving only $\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}$ and $\tilde{\boldsymbol{\psi}}_{k-1}^{(i-1)}$ is obtained, namely

$$\begin{aligned} E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{\sigma_{k-1}}^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i-1)}\|_{\mathrm{F}_k\cdots\mathrm{F}_N\mathrm{F}_1\cdots\mathrm{F}_{k-1}\sigma_{k-1}}^2 \\ + g_k\mathrm{F}_{k+1}\cdots\mathrm{F}_N\mathrm{F}_1\cdots\mathrm{F}_{k-1}\sigma_{k-1} \\ + g_{k+1}\mathrm{F}_{k+2}\cdots\mathrm{F}_N\mathrm{F}_1\cdots\mathrm{F}_{k-1}\sigma_{k-1}. \\ \cdots + g_{k-2}\mathrm{F}_{k-1}\sigma_{k-1} + g_{k-1}\sigma_{k-1} \end{aligned} \tag{3.3.43}$$

One 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 [6]. Since the expression $\sigma_{k-1} = F_k\sigma_k$ denotes the relationship of two free parameters $\{\sigma_{k-1}, \sigma_k\}$, repeating this manner through the network, N relation equations are obtained, where $\sigma_N = F_1\sigma_1$ at the end of the cycle. To begin with, note that there are N degrees of freedom since N different σ_ks are introduced. Moreover, the σ_ks are dummy variables, and this flexibility is used to decouple the set of N equations that arise from the cyclic structure of the algorithm. In this sense, the equations are only rewritten in an equivalent form, using the

set $\{\sigma_k\}$ for "pivoting". Then, the flexibility introduced by the dummy variables can be explored to explicitly assign to the σ_ks quantities that will allow closed-form expressions to be recovered for the quantities of interest. By choosing $\sigma_{k-1} = q = \mathrm{vec}(I)$, the closed-form expression is formulated for the MSD learning curve at node k:

$$\begin{aligned} E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_q^2 = & E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(-1)}\|_{(\Pi_{k-1,1})^i q}^2 \\ & + a_{k-1}(I + \ldots + (\Pi_{k-1,1})^{i-1})q \end{aligned} \tag{3.3.44}$$

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

$$\Pi_{k-1,l} = \mathrm{F}_{k+l-1}\mathrm{F}_{k+l}\cdots\mathrm{F}_N\mathrm{F}_1\cdots\mathrm{F}_{k-1},\ l = 1, 2, \ldots, N \tag{3.3.45}$$

and

$$a_{k-1} = g_k\Pi_{k-1,2} + g_{k+1}\Pi_{k-1,3}\cdots + g_{k-2}\Pi_{k-1,N} + g_{k-1} \tag{3.3.46}$$

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

$$\boxed{E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_q^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(-1)}\|_{f_{i-1}}^2 + a_{k-1}\beta_{i-1}} \tag{3.3.47}$$

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

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

Let $E|\mathbf{e}_{a,k}(i)|^2$ and $E|\mathbf{e}_k(i)|^2$ denote respectively 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., the EMSE and MSE expressions are formulated as

$$E|\mathbf{e}_{a,k}(i)|^2 = E|\mathbf{u}_{k,i}\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}|^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{R_{u,k}}^2 \tag{3.3.48}$$

$$E|\mathbf{e}_k(i)|^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{R_{u,k}}^2 + \sigma_{v,k}^2 \tag{3.3.49}$$

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:

$$E|\mathbf{e}_{a,k}(i)|^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(-1)}\|_{f'_{i-1}}^2 + a_{k-1}\beta'_{i-1} \tag{3.3.50}$$

$$E|\mathbf{e}_k(i)|^2 = E|\mathbf{e}_{a,k}(i)|^2 + \sigma_{v,k}^2 \tag{3.3.51}$$

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

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

3.3.5 Mean and mean-square stability

Substitute $\mathbf{e}_{k,i}$ in (3.3.4) by (3.3.7) to obtain the following relationship between $\tilde{\boldsymbol{\psi}}_k^{(i)}$ and $\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}$ as

$$\begin{aligned} \tilde{\boldsymbol{\psi}}_k^{(i)} = (I - \mu_k \mathbf{U}_{k,i}^*(\epsilon I + \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*)^{-1}\mathbf{U}_{k,i})\tilde{\boldsymbol{\psi}}_{k-1}^{(i)} \\ - \mu_k \mathbf{U}_{k,i}^*(\epsilon I + \mathbf{U}_{k,i}\mathbf{U}_{k,i}^*)^{-1}\mathbf{v}_{k,i} \end{aligned} \tag{3.3.52}$$

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

$$\boxed{E\tilde{\boldsymbol{\psi}}_k^{(i)} = (I - \mu_k E\mathbf{C}_k)E\tilde{\boldsymbol{\psi}}_{k-1}^{(i)}}. \tag{3.3.53}$$

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

$$\mu_k < \frac{2}{\lambda_{\max}(E\mathbf{C}_k)} \tag{3.3.54}$$

where $\lambda_{\max}(A)$ is defined as the largest eigenvalue of A. 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 all eigenvalues of F_k from (3.3.39) satisfy $-1 < \lambda(\mathrm{F}_k) < 1$ as in [6]. Therefore, the matrix F_k is described in (3.3.35) as

$$\mathrm{F}_k = I - \mu_k \mathrm{X}_k + \mu_k^2 \mathrm{Y}_k \tag{3.3.55}$$

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 [6], the convergence of the mean-square error of dAPA will be achieved for values of μ_k in the range

$$\boxed{0 < \mu_k < \min\Big\{\tfrac{1}{\lambda_{\max}(\mathrm{X}_k^{-1}\mathrm{Y}_k)},\ \tfrac{1}{\lambda_{\max}(\mathrm{H}_k)\in\mathbb{R}^+}\Big\}} \tag{3.3.56}$$

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

$$\mathrm{H}_k = \begin{pmatrix} \frac{1}{2}\mathrm{X}_k & -\frac{1}{2}\mathrm{Y}_k \\ I & 0 \end{pmatrix}. \tag{3.3.57}$$

3.3.6 Steady-state behaviour

In the above, the variance relation (3.3.39) 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 $\mathbf{p}_k = \tilde{\boldsymbol{\psi}}_k^{(\infty)}$, (3.3.39) is rewritten as

$$\boxed{E\|\mathbf{p}_k\|_{\sigma_k}^2 = E\|\mathbf{p}_{k-1}\|_{\mathrm{F}_k\sigma_k}^2 + g_k\sigma_k} \tag{3.3.58}$$

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 filters. Iterating (3.3.58) for an incremental network and choosing the proper weighting vectors σ_k for $k = 1, 2, \ldots, N$, an expression only containing $\mathbf{p}_{k-1}$ is obtained, given by

$$\begin{aligned} E\|\mathbf{p}_{k-1}\|_{\sigma_{k-1}}^2 = & E\|\mathbf{p}_{k-1}\|_{\mathrm{F}_k\cdots\mathrm{F}_N\mathrm{F}_1\cdots\mathrm{F}_{k-1}\sigma_{k-1}}^2 \\ & + g_k\mathrm{F}_{k+1}\cdots\mathrm{F}_N\mathrm{F}_1\cdots\mathrm{F}_{k-1}\sigma_{k-1} \\ & + g_{k+1}\mathrm{F}_{k+2}\cdots\mathrm{F}_N\mathrm{F}_1\cdots\mathrm{F}_{k-1}\sigma_{k-1} \\ & \cdots + g_{k-2}\mathrm{F}_{k-1}\sigma_{k-1} + g_{k-1}\sigma_{k-1}. \end{aligned} \tag{3.3.59}$$

Use (6.3.22) and (3.3.46) to simplify this expression, which becomes

$$E\|\mathbf{p}_{k-1}\|_{(I-\Pi_{k-1,1})\sigma_{k-1}}^2 = a_{k-1}\sigma_{k-1} \tag{3.3.60}$$

Expression (3.3.60) is used to evaluate the steady-state performance measures at node k, as follows:

$$\eta_k(\infty) = E\|\mathbf{p}_{k-1}\|_q^2, \quad q = \text{vec}\{I\} \quad \text{(MSD)} \tag{3.3.61}$$

$$\zeta_k(\infty) = E\|\mathbf{p}_{k-1}\|_{r_k}^2, \quad r_k = \text{vec}\{R_{u,k}\} \quad \text{(EMSE)} \tag{3.3.62}$$

$$\xi_k(\infty) = \zeta_k(\infty) + \sigma_{v,k}^2 \quad \text{(MSE)}. \tag{3.3.63}$$

Since the weighting vector σ_{k-1} can be freely chosen, it can be exploited to calculate the steady-state performance of each node. Selecting 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

$$\eta_k(\infty) = a_{k-1}(I - \Pi_{k-1,1})^{-1}q \quad \text{(MSD)} \tag{3.3.64}$$

$$\zeta_k(\infty) = a_{k-1}(I - \Pi_{k-1,1})^{-1}r_k \quad \text{(EMSE)} \tag{3.3.65}$$

$$\xi_k(\infty) = \zeta_k(\infty) + \sigma_{v,k}^2 \quad \text{(MSE)}. \tag{3.3.66}$$

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 NLMS (dNLMS) algorithm can be regarded as a special case of 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, then a unitary G_k and a diagonal matrix Λ_k with the eigenvalues of $R_{u,k}$ are utilized to decompose the autocorrelation

matrix into $R_{u,k} = G_k \Lambda_k G_k^*$. The following transformed quantities

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

are then used to obtain $\|\mathbf{p}_{k-1}\|^2_{\Sigma_k} = \|\bar{\mathbf{p}}_{k-1}\|^2_{\bar{\Sigma}_k}$ and $\|\mathbf{u}_k\|^2_{\Sigma_k} = \|\bar{\mathbf{u}}_k\|^2_{\bar{\Sigma}_k}$. Then, the transformed quantities are used to obtain

$$\begin{aligned}
\bar{\Sigma}'_k =& G_k^* \Sigma'_k G_k \\
=& \bar{\Sigma}_k - \mu_k G_k^* \Sigma_k E\mathbf{C}_k G_k - \mu_k G_k^* E\mathbf{C}_k \cdot \Sigma_k G_k + \mu_k^2 G_k^* E\mathbf{D}_k^{\Sigma_k} G_k \\
=& \bar{\Sigma}_k + \mu_k \bar{\Sigma}_k E\bar{\mathbf{C}}_k - \mu_k E\bar{\mathbf{C}}_k \cdot \bar{\Sigma}_k + \mu_k^2 E\bar{\mathbf{D}}_k^{\bar{\Sigma}_k}
\end{aligned} \tag{3.3.67}$$

where $E\bar{\mathbf{C}}_k = G_k^* E\mathbf{C}_k G_k$ and $E\bar{\mathbf{D}}_k^{\Sigma_k} = G_k^* E\mathbf{D}_k^{\Sigma_k} G_k$. At the steady-state stage, (3.3.26) is rewritten for dNLMS as

$$\begin{aligned}
E\bar{\mathbf{C}}_k =& E\Big[\frac{G_k^* \mathbf{u}_k^* \mathbf{u}_k G_k}{\epsilon + \|\mathbf{u}_k\|^2}\Big] \\
\approx& \frac{\Lambda_k}{\epsilon + Tr(R_{u,k})}
\end{aligned}$$

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 $E\|\mathbf{p}_{k-1}\|^2 = E\|\bar{\mathbf{p}}_{k-1}\|^2$, expression (3.3.64) can be rewritten as

$$\eta_k(\infty) \approx \sum_{k=1}^N g_k (I - \Pi)^{-1} q \tag{3.3.68}$$

which means that the steady-state MSD at each node is approximately the same value. This clearly explains why there is an equalization effect on the MSD of dNLMS throughout the network, and is verified

in Figures 3.7 and 3.10 even for a large step-size. However, for the case of the rank $T \neq 1$ in dAPA, some simplifying approximations in (3.3.68) can not be used, which means that dAPA with $T \neq 1$ can not obtain the equalized performance of MSD throughout the network. Now it is required to be proceed to confirm these values through Monte Carlo simulations.

3.4 Simulations

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

$$\mathrm{u}_{k,i} = \mathrm{col}\{\mathrm{u}_k(i), \mathrm{u}_k(i-1) \cdots, \mathrm{u}_k(i-M+1)\}. \tag{3.4.1}$$

As mentioned in [45], the regularization parameter ϵ plays an important 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 solution is very small and can be neglected. All the coefficients of the adaptive filters within the network are initialized to zeros.

3.4.1 Comparison of distributed algorithms

In the first experiment, the proposed algorithm is performed under shift structure for an incremental network in a system identification scenario, which corresponds to identifying the parameter vector w^o in (2.1.4). 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,

$$W_1(z) = \sum_{n=0}^{19} z^{-n} - \sum_{n=21}^{M-1} z^{-n} \quad \text{and} \quad W_2(z) = -\sum_{n=0}^{M-1} z^{-n} \tag{3.4.2}$$

where $w^o(z) = W_1(z)$ for $i \leq 4000$ and $w^o(z) = W_2(z)$ for $4000 < i \leq 8000$. At each node, the input $u_k(i)$ is generated by passing a unit-variance white Gaussian sequence through a colouring filter (proposed in [4]) with the system function as

$$\begin{aligned} H(z) = 0.1 - 0.2z^{-1} - 0.3z^{-2} + 0.4z^{-3} \\ + 0.4z^{-4} - 0.2z^{-5} - 0.1z^{-6} \end{aligned} \tag{3.4.3}$$

which results in the input correlation matrix of each node having an eigenvalue spread $\frac{\lambda_{\max}}{\lambda_{\min}} \approx 91.85$. The background noise at each node is a white additive uncorrelated noise component with variance $\sigma_{v,k}^2 = 0.01$. As seen in [26], when the forgetting factor τ_k reaches close to unity, both dRLS and Lc-dRLS enable the distributed adaptive estimation 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.

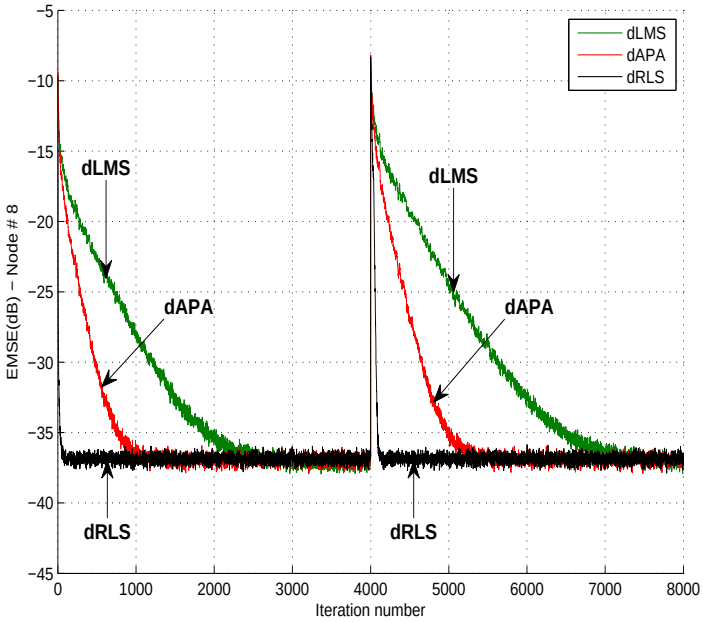

Figure 3.2. Comparison of simulated EMSE learning curves at node 8 for the dLMS, dAPA, and dRLS algorithms in a time-varying environment.

Each node within the incremental network is trained by exploiting either dLMS with step-size $\mu_k = 0.0011$, or dAPA with step-size $\mu_k = 0.0125$ and $T = 3$, or dRLS, for which the forgetting factor is selected as $\tau_k = 0.95$, 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 [27] 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 approximating 37dB. In addition, as shown in Figure 3.19 in Appendix A of this chapter, the setting $T = 3$ enables dAPA to have approximately one-third of the computational cost of dRLS per iteration 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. The selection $\tau_k = 0.95$, which corresponds to a window length of $(1-\tau_k)^{-1} \approx 20$, is therefore used to ensure dRLS has the best convergence performance over other algorithms in the time-varying w^o environment. The curves are shown in Figure 3.2 by running 8000 iterations and averaging 500 Monte Carlo runs, where iteration number stands for discrete time index. 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 behaviour 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, Figure 3.2 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 filter with $M = 10$, whose tap weights are generated randomly from a standardized Gaussian distribution. The correlated input $u_k(i)$ at each node is obtained by passing a white Gaussian or quasi-uniform random process with variance $\sigma_{u,k}^2 = 1$ through a first-order AR model filter with z-domain transfer function $\sqrt{1-\alpha_k^2}/(1-\alpha_k z^{-1})$ and $\alpha_k \in (0, 0.5]$ as in Chapter 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 $\mathrm{r}_k(m) = \sigma_{u,k}^2 \alpha_k^{|m|}$, $m = 0, \cdots, 9$. The background noise of each node is a white Gaussian process with variance $\sigma_{v,k}^2 \in (0, 0.1]$. Fig-

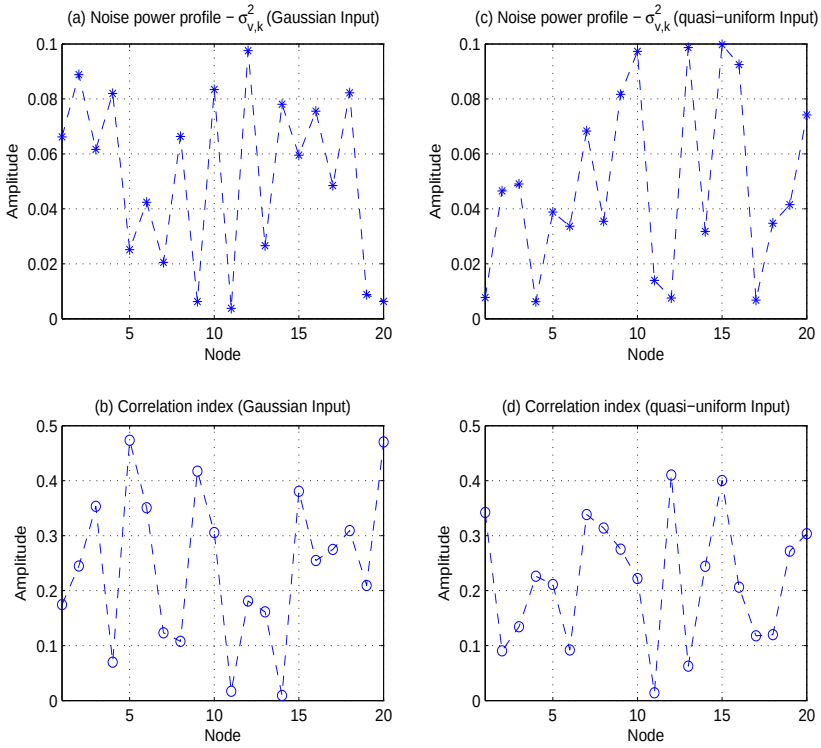

Figure 3.3. Node profile: a) Noise power for the Gaussian data network b) Correlation index for the Gaussian data network c) Noise power for the quasi-uniform data network d) Correlation index for the quasi-uniform data network.

ure 3.3 illustrates respectively the node profiles of $\sigma^2_{v,k}$ and α_k for both coloured Gaussian network inputs and coloured quasi-uniform data network inputs. Furthermore, as seen in Figure 3.3, since node 5 and node 12 have the most highly correlated inputs within the corresponding networks, a comparison of their performance between theory and practice is provided in the following example.

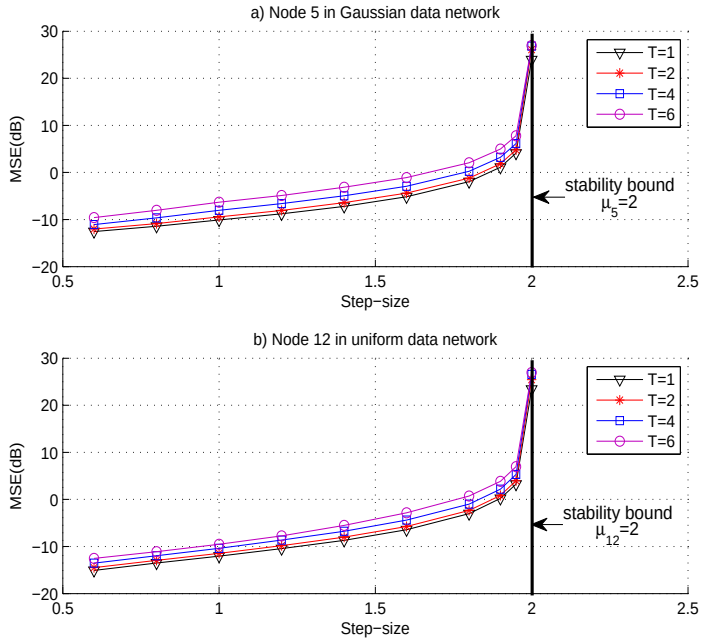

Figure 3.4. Simulated MSE curves of dAPA as a function of the step-size: a) Node 5 in the coloured Gaussian data network b) Node 12 in the coloured quasi-uniform data network.

Table 3.2. Stability bounds of step-size for dAPA at node 5 in the Gaussian data network (4 decimal place precision is used as this corresponds to the resolution in the changes of $\mu_{\max}$).

	$\frac{2}{\lambda_{\max}(E\mathbf{C}_5)}$	$\frac{1}{\lambda_{\max}(\mathrm{X}_5^{-1}\mathrm{Y}_5)}$	$\frac{1}{\lambda_{\max}(\mathrm{H}_5)\in\mathbb{R}^+}$	$\mu_{\max}$
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 3.3. Stability bounds of step-size for dAPA (at node 12 in the quasi-uniform data network).

	$\frac{2}{\lambda_{\max}(E\mathbf{C}_{12})}$	$\frac{1}{\lambda_{\max}(\mathrm{X}_{12}^{-1}\mathrm{Y}_{12})}$	$\frac{1}{\lambda_{\max}(\mathrm{H}_{12})\in\mathbb{R}^+}$	$\mu_{\max}$
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

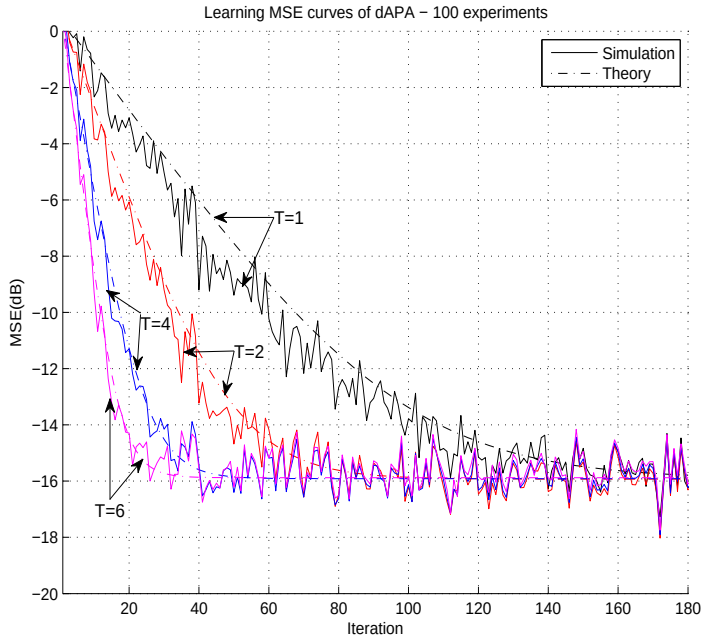

Figure 3.5. Learning MSE curves of dAPA using $\mu_k = 0.01$ for Node 5 in the coloured Gaussian data network.

3.4.2 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 (3.3.54) and (3.3.56), the step-size bounds for dAPA with shift structure are evaluated at the two corresponding nodes in Table 3.2 and Table 3.3, which verify that the stability bounds on μ_k are approximately $0 < \mu_k < 2$. This fact is further confirmed in Figure 3.4, where steady-state MSE curves are plotted as a function of the step-size. X_5, Y_5, X_{12} and Y_{12} which are involved in evaluating the expectations for the bounds of step-size, are calculated via ensemble averaging.

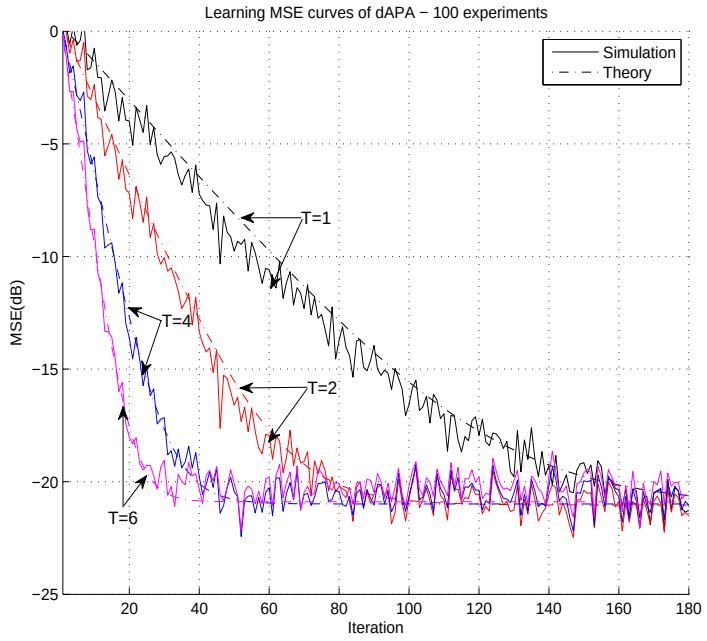

Figure 3.6. Learning MSE curves of dAPA using $\mu_k = 0.01$ for Node 12 in the coloured quasi-uniform data network.

3.4.3 Transient performance

Figures 3.5 and 3.6 illustrate the transient performance curves of dAPA with shift structure during the initial 180 samples. Since the same step-size as $\mu_k = 0.01$ is chosen for dNLMS and dAPA with different T, increasing T results in faster convergence rate but poorer expected misadjustment in the steady-state, namely, dNLMS obtains the best steady-state performance, which can be clearly seen in Figure 3.9. For the transient performance, the simulation results present good agreement with the theoretical results using equation (3.3.51), where F_k and a_k are calculated by ensemble averaging.

3.4.4 Steady-State performance

In these simulation, the aim is to satisfy independence assumption **A2**. Therefore, the following simulations are performed with temporally independent Gaussian or uniform inputs, i.e. the elements of $u_{k,i}$ 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 indices for those inputs are shown in Figure 3.3, which also illustrates the noise power at each node for the corresponding networks. In Figures 3.7-3.18, 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 equations (3.3.64)-(3.3.66). Figure 3.7-3.12 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 coloured Gaussian input data network and the coloured uniform input data network. These quantities combine the transformed noise and local statistics from the whole network. As equation (3.3.68) claims, even for a large step-size $\mu_k = 0.2$, the MSD curves of dNLMS in Figures 3.7 and 3.10, are approximately flat throughout the networks. 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 due to equation (3.3.68), (3.3.65) enables the EMSE curves of dNLMS to have a similar shape as the correlation index.

As can be seen in Figures 3.8 and 3.11, 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

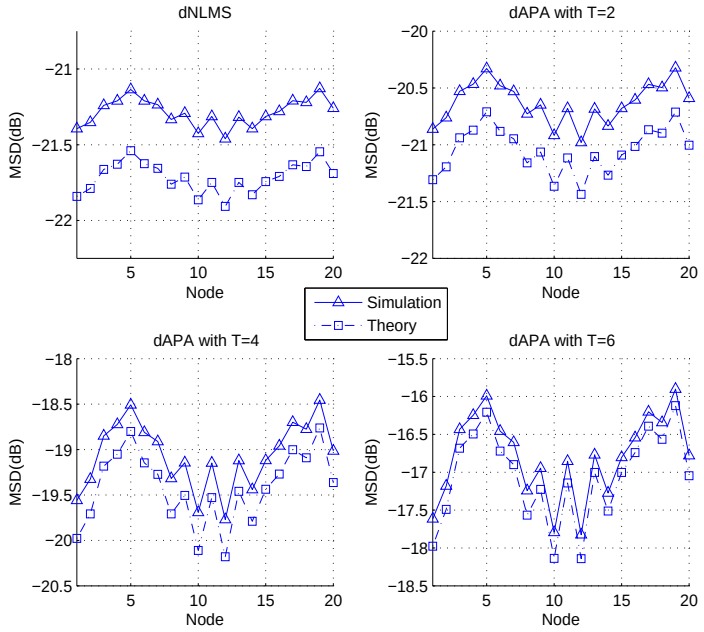

Figure 3.7. Steady-state MSD for dNLMS and dAPA using $\mu_k = 0.2$ for the coloured Gaussian data network.

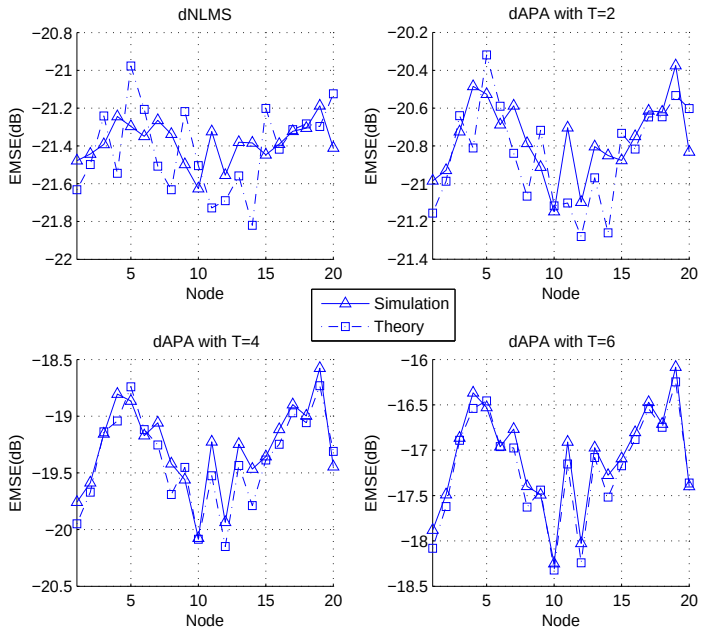

Figure 3.8. Steady-state EMSE for dNLMS and dAPA using $\mu_k = 0.2$ for the coloured Gaussian data network.

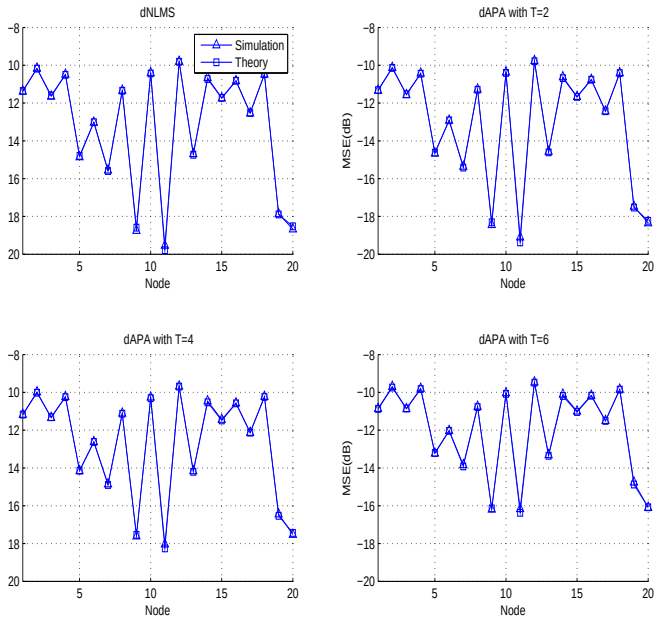

Figure 3.9. Steady-state MSE for dNLMS and dAPA using $\mu_k = 0.2$ for the coloured Gaussian data network.

dNLMS having the smallest absolute value approximately -21.4dB for the Gaussian input network or approximately -22.4dB for the uniform inputs network. For the absolute value of the difference between theory and simulation, dNLMS has smaller absolute value than dAPA, for example, as shown in Figure 3.11 for node 13 the absolute gap value of dNLMS between -22.25dB and -22.65dB is approximately 0.00052 while that of dAPA with $T = 4$ between -20.4dB and -20.7dB is also approximately 0.0006. In addition, the theoretical results of MSE evaluated by equation (3.3.66) are very close to the simulated results even with the large step-size $\mu_k = 0.2$ as shown in Figures 3.9 and 3.12. It should be highlighted that the assumptions in **A3** and **A3'** underlies the mismatch between the theoretical and simulated results. Moreover, as shown in Figures 3.9 and 3.12, the MSE curves roughly coincide

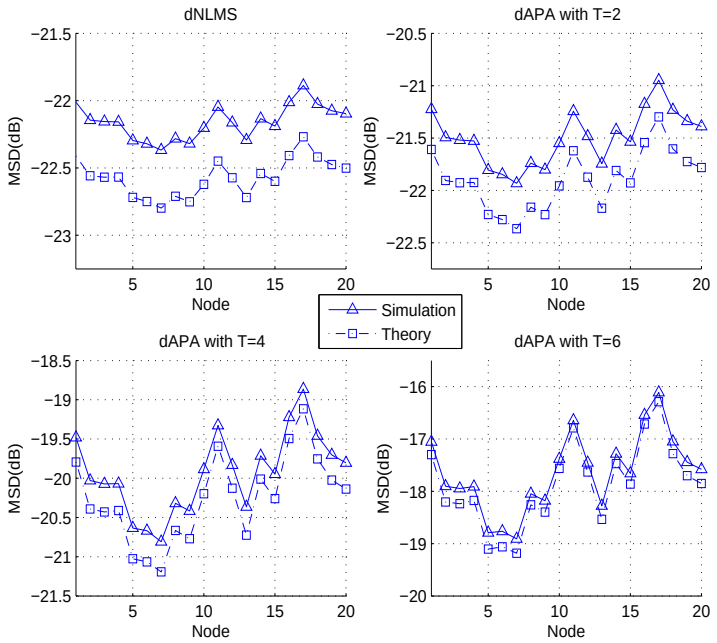

Figure 3.10. Steady-state MSD for dNLMS and dAPA using $\mu_k = 0.2$ for the coloured quasi-uniform data network.

with the noise power, which indicates that when the proper step-size μ_k is 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, a similar finding as in [1] is obtained if the whole network is required to have an equalized performance, Figures 3.9 and 3.12 confirm that the spatial diversity of the adaptive networks can be used to design the step-size for 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.

Figures 3.13-3.18 illustrate the steady-state MSD, EMSE and MSE curves of dNLMS and dAPA at node 5 for the coloured Gaussian in-

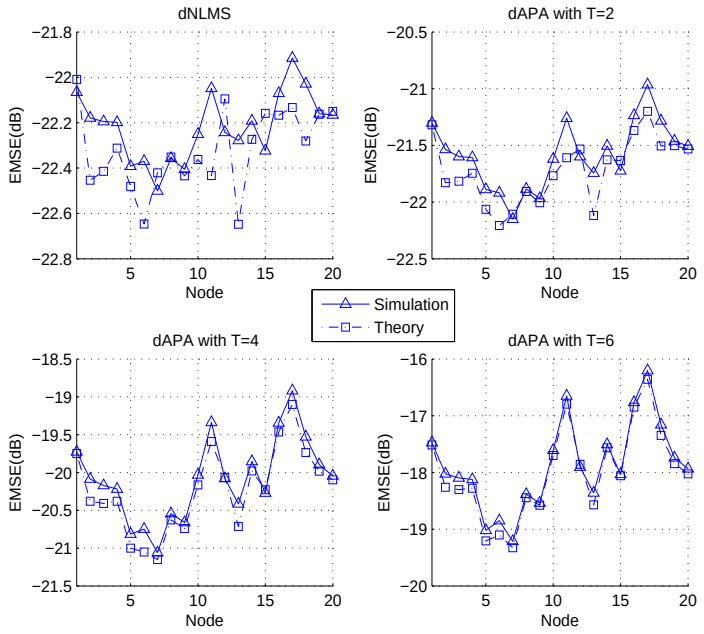

Figure 3.11. Steady-state EMSE for dNLMS and dAPA using $\mu_k = 0.2$ for the coloured quasi-uniform data network.

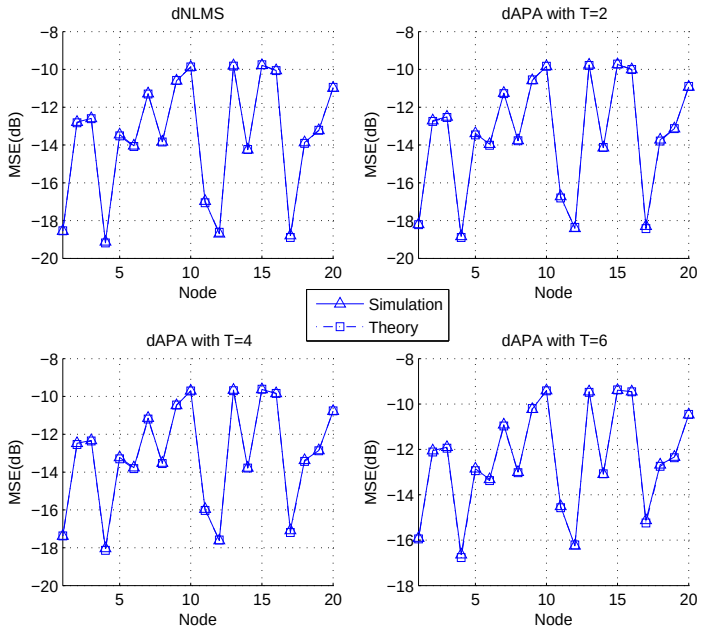

Figure 3.12. Steady-state MSE for dNLMS and dAPA using $\mu_k = 0.2$ for the coloured quasi-uniform data network.

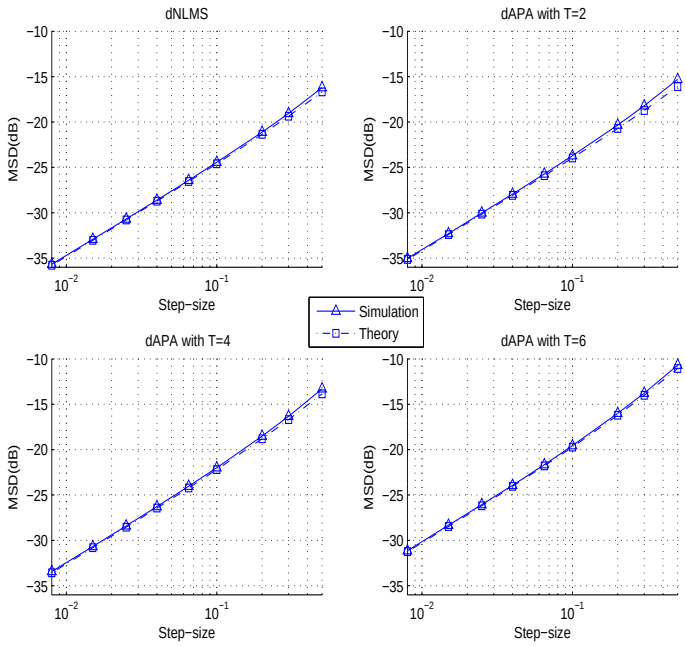

Figure 3.13. Steady-state MSD curves of dNLMS and dAPA at node 5 in the coloured Gaussian data network as a function of the step-size.

put data network and at node 5 for the coloured uniform input data network as a function of the step-size. For both corresponding networks, the step-size varies in the range [0.008 0.5], which guarantees the stability of the scheme as mentioned before. Equations (3.3.64)-(3.3.66) are used to calculate the theoretical results. In order to obtain the instantaneous mean square error measurements in the steady-state, the simulation results are obtained by averaging the last 2000 sample in 20000 iteration numbers and then averaging 100 independent trails. 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. As is expected, when the step-size is the same, dNLMS has the smaller absolute value of the discrepancy between theory and practice than dAPA, namely, increase of T results in an increase of discrep-

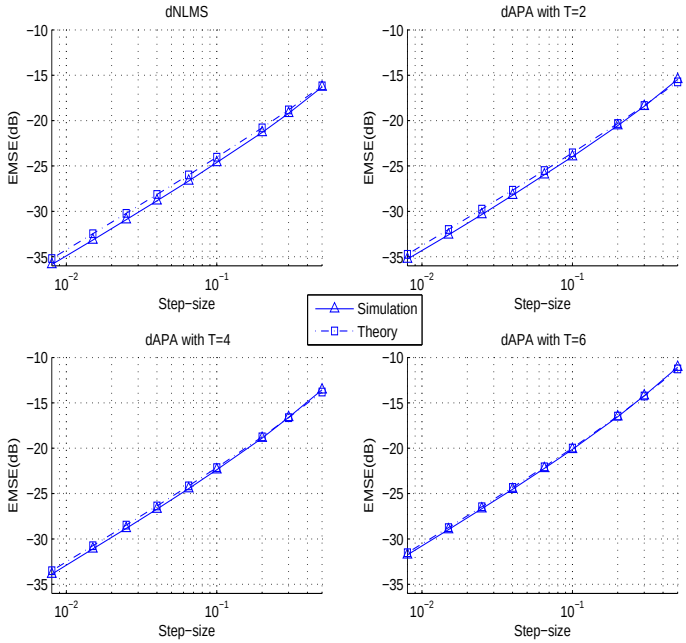

Figure 3.14. Steady-state EMSE curves of dNLMS and dAPA at node 5 in the coloured Gaussian data network as a function of the step-size.

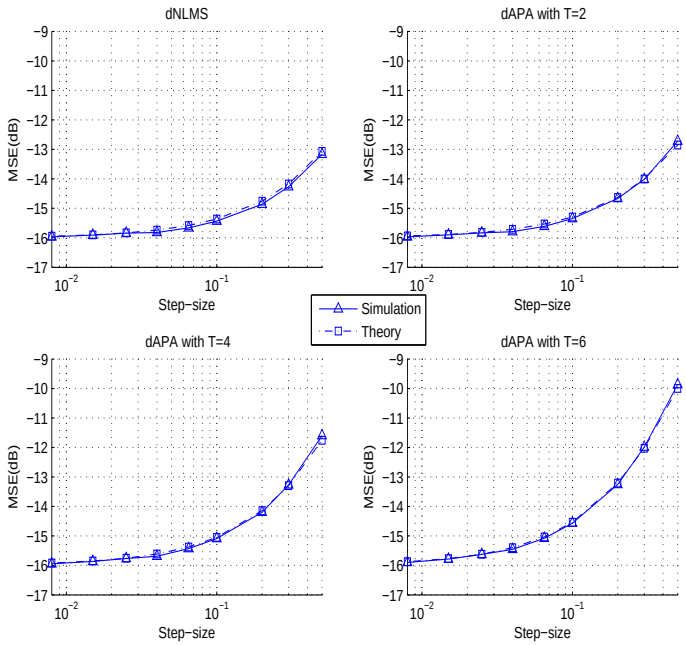

Figure 3.15. Steady-state MSE curves of dNLMS and dAPA at node 5 in the coloured Gaussian data network as a function of the step-size.

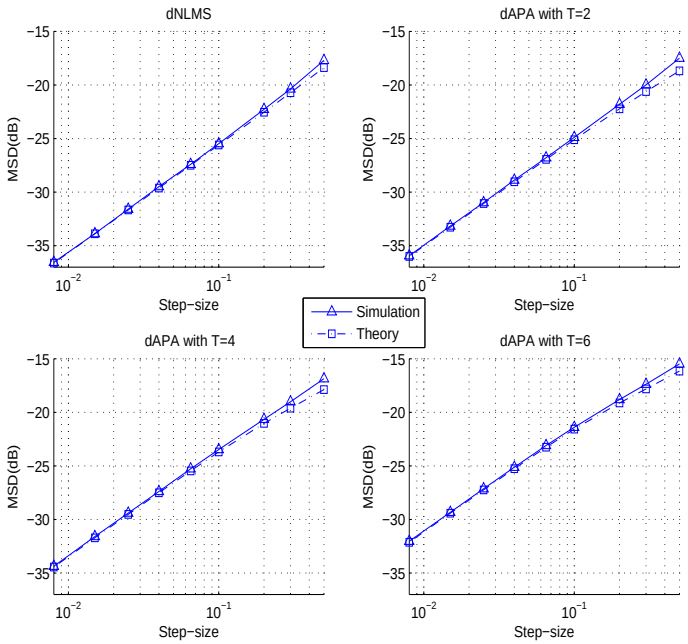

Figure 3.16. Steady-state MSD curves of dNLMS and dAPA at node 5 in the coloured quasi-uniform data network as a function of the step-size.

ancy. This is because larger T enables more local input regressors to be involved in the $w_{k,i}$ update, which will degrade the simplifying assumptions adopted in the analysis. On the other hand, the use of large step-sizes usually does not satisfy these simplifying assumptions, which will lead to large deviations between theory and simulation. Therefore, the range of the step-size is chosen in small values with the purpose of comparison. One can clearly seen that as the step-size decreases, the discrepancy decreases.

3.5 Conclusions

This chapter described a new incremental adaptive learning algorithm based on APA for a distributed network and presented detailed per-

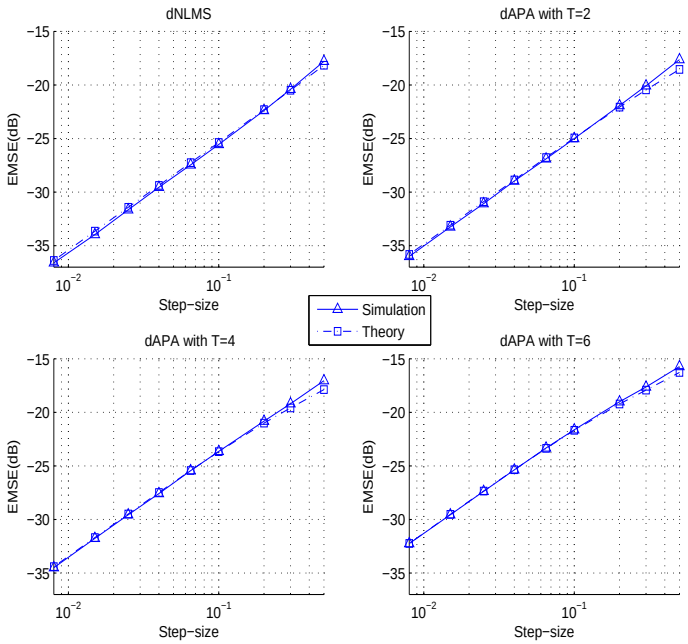

Figure 3.17. Steady-state EMSE curves of dNLMS and dAPA at node 5 in the coloured quasi-uniform data network as a function of the step-size.

formance analysis based on the weighted space-time energy conservation approach of Lopes and Sayed [1] under assumptions **A1**, **A2**, **A3** and **A3'**. The theoretical expressions for MSD, EMSE and MSE derived in this chapter do not restrict the distribution of the inputs to being Gaussian or white. Both stationary environments and a single 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

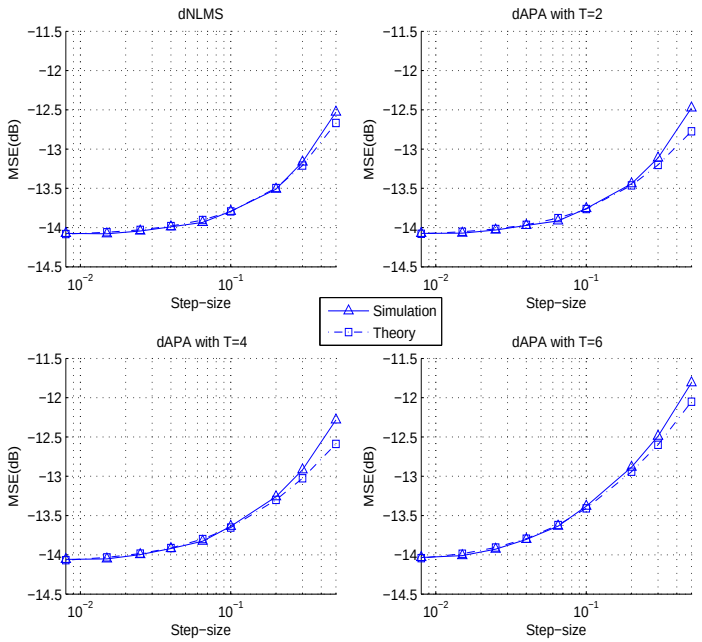

Figure 3.18. Steady-state MSE curves of dNLMS and dAPA at node 5 in the coloured quasi-uniform data network as a function of the step-size.

employed consistently within the simulation experiments. Furthermore, the theoretical expressions are compared with the simulated results, for both the transient and steady-state performance of dAPA in both the Gaussian data network and quasi-uniform data network. In addition, it can be clearly seen 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 across the network is required to be defined and nodes communicate with neighbours within this path. However, this manner 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 [23], [24], where each node cooperates with a subset of neighbouring nodes, but this will be very much dependent upon the network size and topolgy. These topics will be addressed in the following chapter.

3.6 Appendix A: Comparison of complexity, memory and transmission costs

The computational costs for the different schemes in the same style as the presented complexity formulas in [6] are compared in this appendix. Since a complex addition includes two real additions, and a complex multiplication is assumed to be computed with four real multiplications and two real additions, the estimated number of real multiplications, real additions and real divisions, which are required for the case of real-valued data for the various algorithms per node per time instant, is depicted in Table 3.4, and the total number of operations for complex-valued data is shown in Table 3.5. In order to ensure that dAPA has intermediate complexity between dLMS and dRLS in the real-valued data case, a complexity condition should hold which ensures that the total number of computational operations in dRLS is the largest, namely $2M^2+8M+2 > 2T^2M+4TM+M+2T^3+T^2+T > 4M+1$, which corresponds to the region below the solid quadratic curve in Fig. 3.19 which shows the complexity comparison for each update of the different algorithms per node. On the other hand, the complexity condition for complex data is $8M^2+28M+1 > 8T^2M+16TM+2M+8T^2+2T^2+4T > 16M+2$. Note that dNLMS always has higher complexity than dLMS,

Table 3.4. 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 3.5. 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

which is illustrated in Figure 3.19. 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 $M = 90$, $T < 9$.

For communications between nodes, since the entire inverse sample input correlation matrix requires to be transferred between nodes, dRLS of [1] at each node requires $O(M^2)$ transmission complexity. On the other hand, the requirement of other algorithms is $O(M)$ as essentially only the tap vector is transferred between nodes. A simplified distributed RLS algorithm presented in [27] can reduce the communications between nodes to $O(M)$ at the expense of reduced steady-state performance and tends to the same performance as dRLS implementation when the forgetting factor λ_k approaches unity. On the other hand, for real data the estimated memory costs of dLMS, dAPA and dRLS are respectively $5M+4$, $5T^2+4M+3T+2TM+2$, $5T^2+3M+4T+3TM+1$

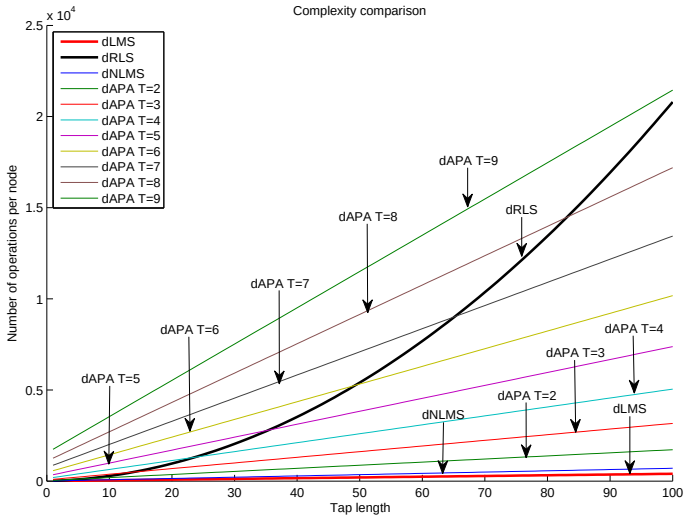

Figure 3.19. Complexity comparison in terms of operations per iteration per node for various algorithms (dLMS, dNLMS, dAPA and dRLS).

and $6M^2 + 10M + 6$ at each node. Since $T < M$ always holds for APA based methods, dAPA has intermediate memory cost between dLMS and dRLS. The above memory computations rely on the assumption that all the scalars, vectors and matrices involved during the computing process are stored. However, in practice only unique values of vectors and matrices are required to be stored and some storage of intermediate results can be removed after computations, which cuts down the memory storage.

Chapter 4

DISTRIBUTED ADAPTIVE ESTIMATION BASED ON THE APA ALGORITHM OVER A DIFFUSION NETWORK WITH CHANGING TOPOLOGY

4.1 Introduction

In a network of nodes spreading over a physical area, the objective is to estimate the parameters of interest by exploiting observations of temporal data collected from nodes with different spatial locations, statistical profiles of which are possibly different. However, applications in some environments suffer limited capabilities of communications and complexity due to tight energy and bandwidth constraints, especially in wireless sensor networks. Such constraints lead to the development of distributed adaptive algorithms based on LMS and RLS rules for

incremental networks as in [1], [27]. This type of topology reduces the level of communication and computational resources as compared with networks having other topologies. It is well known that an incremental topology requires to establish a Hamiltonian cycle to connect all the nodes in the network at every iteration. This mode of cooperation is only likely to be suitable for small size networks and limits the autonomy of the network topology. An ideal cooperation strategy should be capable of dealing with different network topologies, possibly dynamic changing. Therefore, peer-to-peer distributed algorithms based on diffusion protocols have been derived and studied in [22], [25], [28], [29], which take advantage of the cooperation among the individual adaptive nodes. In these types of diffusion distributed algorithms, each node in a network only communicates with the nodes within its neighbourhood, but also experiences the effect of the entire network. The applications of these schemes could be found useful to solve the problem of estimation and event detection by using multiple nodes to collect space-time data [8], [46], [47], [48], [49].

In the case of a single adaptive filter, the advantage of the APA algorithm is well known: it achieves an improved convergence performance as compared with the LMS algorithm for some coloured input signals but obtains a reasonable convergence performance by using less computational cost than the RLS algorithm for certain tap-lengths. In the former chapter, a new incremental distributed algorithm based on an APA rule was developed with the purpose to obtain a good balance between computational cost and convergence performance. As a consequence, a new diffusion type learning algorithm relying on the APA rule is derived in this chapter to solve the distributed estimation

problem, where the cooperation strategy between nodes is a peer-to-peer diffusion protocol and nodes communicates only with their peer neighbours at every iteration. Simulations confirm that the proposed algorithm achieves a greatly improved performance as compared with a noncooperative scheme.

4.2 Estimation problem

Consider the distributed estimation problem as in Chapter 2: the objective is to seek an unknown $M \times 1$ vector w over an $N-$node network by solving

$$\min_{w} E\|\mathbf{d} - \mathbf{U}w\|^2 \tag{4.2.1}$$

where two global data matrices

$$\mathbf{d}=\text{col}\{\mathbf{d}_1,\, \mathbf{d}_2,\, \ldots,\, \mathbf{d}_N\}, \quad (N \times 1) \tag{4.2.2}$$

$$\mathbf{U}=\text{col}\{\mathbf{u}_1,\, \mathbf{u}_2,\, \ldots,\, \mathbf{u}_N\}, \quad (N \times M) \tag{4.2.3}$$

are formulated by the zero-mean random measurement data $\mathbf{d}_k$ and regression data $\mathbf{u}_k$, the time realizations of which are denoted by $\{d_{k,i}, u_{k,i}\}$, $k = 1, \ldots, N$ with time index i. The optimal solution w^o satisfying the normal equations [6] is therefore obtained by solving

$$R_{du} = R_u w^o \tag{4.2.4}$$

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

The purpose is to design a distributed adaptive scheme relying on the APA rule to achieve a good estimate approximating the solution w^o of (4.2.4) at every node within the network. As mentioned in the

introduction, diffusion topology can be used as a node cooperation strategy, where at node k the adaptive filter fuses the local estimates from its neighbourhood and exploits the resulting aggregated estimate to update the local estimate. Let $\psi_k^{(i-1)}$ denote the local estimate at node k at time $i-1$. Node k therefore obtains a set of unbiased estimates $\{\psi_l^{(i-1)}\}_{l\in\mathcal{N}_{k,i-1}}$ from its neighborhood $\mathcal{N}_{k,i-1}$, which is defined as the set of all nodes connecting to node k at any given time $i-1$. At node k, a local combining function f_k is used to fuse these local estimates, yielding the aggregate estimate $\phi_k^{(i-1)}$,

$$\phi_k^{(i-1)} = f_k\left(\psi_l^{i-1}; l \in \mathcal{N}_{k,i-1}\right) \tag{4.2.5}$$

where $\mathcal{N}_{k,i-1}$ is time dependent and includes node k itself. The time-dependent neighbourhood provides a robust framework when the link failures are possible. Due to every node having a diffusion linking neighbourhood, one should note that this aggregate step combines information from the whole network. The APA-based adaptive filter updates the local estimate $\psi_k^{(i)}$ by using the resulting aggregate estimate ϕ_k^{i-1}. As mentioned in [25], the combiner function f_k may be nonlinear, to respond to the time-varying conditions. In this work, linear combiners are used and f_k is therefore replaced by some weighted combination, yielding

$$\phi_k^{(i-1)} = \sum_{l\in\mathcal{N}_{k,i-1}} c_{k,l}(i-1)\psi_l^{(i-1)} \tag{4.2.6}$$

with weighted coefficients $c_{k,l}(i-1) \geq 0$ and $\sum_{l\in\mathcal{N}_{k,i-1}} c_{k,l}(i-1) = 1$. Let $n_{k,i-1}$ and $n_{l,i-1}$ denotes the degree of the neighbourhoods, i.e. $n_{k,i-1} = |\mathcal{N}_{k,i-1}|$. One choice for the combiner is the Metropolis rule,

as in [37],

$$c_{k,l}(i-1)=\begin{cases}\frac{1}{\max(n_{k,i-1},n_{l,i-1})}, & \text{if } k \text{ links } l \text{ but } k\neq l\\ 1-\sum_{l\in\mathcal{N}_{k,i-1},l\neq k} c_{k,l}(i-1), & k=l\\ 0, & \text{if } k \text{ and } l \text{ are not linked.}\end{cases} \tag{4.2.7}$$

Other combiners are the Laplacian and the nearest neighbour rules [36], [50].

Recall the update recursion of the diffusion LMS strategy as in [25], which can be rewritten in theory as follows:

$$\psi_k^{(i-1)} = \phi_k^{(i-1)} - \mu_k[\nabla J_k(\phi_k^{(i-1)})]^* \text{ for node } k \tag{4.2.8}$$

where μ_k denotes the local step-size. With the purpose of improving the convergence performance, a regularized Newton's search based approach [6] is therefore adopted in (4.2.8), yielding,

$$\begin{aligned}\psi_k^{(i)}&=\phi_k^{(i-1)} - \mu_k[\epsilon I + \nabla^2 J_k(\phi_{k-1}^{(i)})]^{-1}[\nabla J_k(\phi_{k-1}^{(i)})]^*\\ &=\phi_k^{(i-1)} + \mu_k(\epsilon I + R_{u,k})^{-1}[R_{du,k} - R_{u,k}\phi_{k-1}^{(i)}]\end{aligned} \tag{4.2.9}$$

where $R_{u,k} = E\mathbf{u}_k^*\mathbf{u}_k$, $R_{du,k} = E\mathbf{d}_k\mathbf{u}_k^*$ and ϵ denotes a regularization parameter with small positive value. One method to realize (4.2.9) in practice is to replace $\{R_{u,k}, R_{du,k}\}$ by the following sample sliding-window estimates,

$$\begin{aligned}\hat{R}_{u,k}&=\frac{1}{T}\sum_{j=i-T+1}^{i} u_{k,j}^* u_{k,j}\\ \hat{R}_{du,k}&=\frac{1}{T}\sum_{j=i-T+1}^{i} u_{k,j}^* d_k(j)\end{aligned} \tag{4.2.10}$$

where T indicates the rank of recent regressors. Hence, recursion (4.2.9) becomes,

$$\psi_k^{(i)} = \phi_k^{(i-1)} + \mu_k U_{k,i}^*(\epsilon I + U_{k,i}U_{k,i}^*)^{-1}[d_{k,i} - U_{k,i}\phi_k^{(i-1)}] \qquad (4.2.11)$$

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

$$U_{k,i} = \begin{pmatrix} u_{k,i} \\ u_{k,i-1} \\ \vdots \\ u_{k,i-T+1} \end{pmatrix}, \; d_{k,i} = \begin{pmatrix} d_k(i) \\ d_k(i-1) \\ \vdots \\ d_k(i-T+1) \end{pmatrix} \qquad (4.2.12)$$

and ϵ is employed to avoid the inversion of a rank deficient matrix $U_{k,i}U_{k,i}^*$. Therefore, a linear combiner model and APA-type local adaptive rule are employed to result in the diffusion APA algorithm as summarized in Table 4.1. The convergence performance of the proposed algorithm will be studied in the following sections.

Table 4.1. Pseudo-code implementation of diffusion APA.

For each time instant $i \geq 0$ repeat:
Initialization $\phi_k^{(-1)} = \mathbf{0}$
k=1, ... N
$\phi_k^{(i-1)} = \sum_{l \in \mathcal{N}_{k,i-1}} c_{k,l}(i-1)\psi_l^{(i-1)}$
$\psi_k^{(i)} = \phi_k^{(i-1)} + \mu_k U_{k,i}^*(\epsilon I + U_{k,i}U_{k,i}^*)^{-1}[d_{k,i} - U_{k,i}\phi_k^{(i-1)}]$
end
where $\{d_{k,i}, U_{k,i}\}$ are defined by (4.2.12).

4.3 Network global model

The space-time structure of the algorithm leads to challenge in the performance analysis. To proceed, first, the global representations, in terms of the stochastic quantities, are formulated to gain the insights into the effects of cooperation strategy and network model on system performance and are defined as,

$$\boldsymbol{\psi}_c^{i-1} = \mathrm{col}\{\boldsymbol{\psi}_1^{(i-1)}, \ldots, \boldsymbol{\psi}_N^{(i-1)}\}, \quad \boldsymbol{\phi}_c^{i-1} = \mathrm{col}\{\boldsymbol{\phi}_1^{(i-1)}, \ldots, \boldsymbol{\phi}_N^{(i-1)}\}$$
$$\mathbf{U}_c^i = \mathrm{diag}\{\mathbf{U}_{1,i}, \ldots, \mathbf{U}_{N,i}\}, \qquad \mathbf{d}_c^i = \mathrm{col}\{\mathbf{d}_{1,i}, \ldots, \mathbf{d}_{N,i}\}$$

where $\mathbf{U}_c^i$ is an $NT \times NM$ block diagonal matrix. An $NM \times NM$ diagonal matrix D is defined by

$$D = \mathrm{diag}\{\mu_1 I_M, \ldots, \mu_N I_M\} \tag{4.3.1}$$

to collect the local step-sizes. The measurements for the APA rule can be expressed in a traditional manner, given by,

$$\mathbf{d}_{k,i} = \mathbf{U}_{k,i} w^o + \mathbf{v}_{k,i} \tag{4.3.2}$$

where $\mathbf{v}_{k,i}$ is obtained by

$$\mathbf{v}_{k,i} = \mathrm{col}\{\mathbf{v}_k(i), \ldots, \mathbf{v}_k(i-T+1)\}, \quad (T \times 1) \tag{4.3.3}$$

and $\mathbf{v}_k(i)$ is background noise, spatially and temporally independent with variance $\sigma_{v,k}^2$. From the linear model of the form (4.3.2), the

global model for diffusion APA is obtained

$$\mathbf{d}_c^i = \mathbf{U}_c^i w_c^o + \mathbf{v}_c^i \tag{4.3.4}$$

where $w_c^o = Qw^o$ with an $NM \times M$ transition matrix

$$Q = \text{col}\{I_M, \ldots, I_M\} \tag{4.3.5}$$

and an $NT \times 1$ global error vector

$$\mathbf{v}_c^i = \text{col}\{\mathbf{v}_{1,i}, \ldots, \mathbf{v}_{N,i}\}. \tag{4.3.6}$$

To facilitate analysis, the network topology is assumed to be static (i.e. $c_{k,l}(i) = c_{k,l}$). It should be highlighted that this assumption does not compromise the algorithm derivation or its operation, and is used for analysis only. Using the above expressions, the global model of diffusion APA is therefore formulated as follows:

$$\boldsymbol{\psi}_c^i = G\boldsymbol{\psi}_c^{i-1} + D\mathbf{U}_c^{i*}(\epsilon I_{TN} + \mathbf{U}_c^i\mathbf{U}_c^{i*})^{-1}(\mathbf{d}_c^i - \mathbf{U}_c^i G\boldsymbol{\psi}_c^{i-1}) \tag{4.3.7}$$

where $G = C \otimes I_M$ is the $NM \times NM$ network topology matrix and the symmetric combining matrix C, is formed by $\{c_{k,l}\}$. Now the objective is to study the performance behavior of cooperative systems governed by the form (4.3.7).

4.4 Performance analysis

Since the spatial independence assumption of $\{\mathbf{U}_{k,i}\}$ is likely to hold, the independence assumptions in Chapter 3 can therefore be extended

to the global matrix $\mathbf{U}_c^i$. These types of assumptions are very common in the adaptive signal processing literature, and are frequently necessary to permit analysis even for the simplest of adaptive schemes. For the later reference, the local error vector is defined as

$$\mathbf{e}_{k,i} = \mathbf{d}_{k,i} - \mathbf{U}_{k,i}\boldsymbol{\phi}_k^{i-1}, \quad (T \times 1) \tag{4.4.1}$$

which is related to the global error vector $\mathbf{e}_{c,i}$

$$\mathbf{e}_{c,i} = \mathrm{col}\{\mathbf{e}_{1,i}, \ldots, \mathbf{e}_{N,i}\}, \quad (NT \times 1). \tag{4.4.2}$$

Let $\tilde{\boldsymbol{\psi}}_c^i$ denote the global weight error vector, given by,

$$\tilde{\boldsymbol{\psi}}_c^i = w_c^o - \boldsymbol{\psi}_c^i, \quad (NM \times 1). \tag{4.4.3}$$

As a consequence, (4.4.2) can be rewritten as

$$\mathbf{e}_{c,i} = \mathbf{d}_c^i - \mathbf{U}_c^i G\boldsymbol{\psi}_c^{i-1} = \mathbf{U}_c^i G\tilde{\boldsymbol{\psi}}_c^{\,i-1} + \mathbf{v}_c^i = \mathbf{e}_{a,i}^G + \mathbf{v}_c^i \tag{4.4.4}$$

where

$$\mathbf{e}_{a,i}^G = \mathbf{U}_c^i G\tilde{\boldsymbol{\psi}}_c^{\,i-1} \tag{4.4.5}$$

and

$$\tilde{\boldsymbol{\psi}}_c^i = G\tilde{\boldsymbol{\psi}}_c^{i-1} - D\mathbf{U}_c^{i*}(\epsilon I_{TN} + \mathbf{U}_c^i\mathbf{U}_c^{i*})^{-1}\mathbf{e}_{c,i}. \tag{4.4.6}$$

Let Σ denote an $NM \times NM$ arbitrary positive matrix. Pre-multiply $\mathbf{U}_c^i D\Sigma$ to both sides of (4.4.6) to yield,

$$\mathbf{U}_c^i D\Sigma\tilde{\boldsymbol{\psi}}_c^i = \mathbf{U}_c^i D\Sigma G\tilde{\boldsymbol{\psi}}_c^{i-1} - \mathbf{U}_c^i D\Sigma D\mathbf{U}_c^{i*}(\epsilon I_{TN} + \mathbf{U}_c^i\mathbf{U}_c^{i*})^{-1}\mathbf{e}_{c,i} \tag{4.4.7}$$

$$\mathbf{e}_{p,i}^{D\Sigma} = \mathbf{e}_{a,i}^{D\Sigma G} - \mathbf{U}_c^i D\Sigma D\mathbf{U}_c^{i*}(\epsilon I_{TN} + \mathbf{U}_c^i\mathbf{U}_c^{i*})^{-1}\mathbf{e}_{c,i} \tag{4.4.8}$$

where the global a priori and a posteriori weighted estimation errors are given by:

$$\mathbf{e}_{a,i}^{D\Sigma G} = \mathbf{U}_c^i D\Sigma G \tilde{\boldsymbol{\psi}}_c^{i-1} \tag{4.4.9}$$

$$\mathbf{e}_{p,i}^{D\Sigma} = \mathbf{U}_c^i D\Sigma \tilde{\boldsymbol{\psi}}_c^{i}. \tag{4.4.10}$$

Replacing $\mathbf{e}_{c,i}$ in (4.4.6) by $\{\mathbf{e}_{a,i}^{D\Sigma G}, \mathbf{e}_{p,i}^{D\Sigma}\}$ from (4.4.8) leads to

$$\tilde{\boldsymbol{\psi}}_c^i + D\mathbf{U}_c^{i*}(\mathbf{U}_c^i D\Sigma D\mathbf{U}_c^{i*})^{-1}\mathbf{e}_{a,i}^{D\Sigma G} = G\tilde{\boldsymbol{\psi}}_c^{i-1} + D\mathbf{U}_c^{i*}(\mathbf{U}_c^i D\Sigma D\mathbf{U}_c^{i*})^{-1}\mathbf{e}_{p,i}^{D\Sigma} \tag{4.4.11}$$

with the assumption that $\mathbf{U}_c^i D\Sigma D\mathbf{U}_c^{i*}$ is invertible. Weight energy balance is performed on both side of (4.4.11) to yield

$$\begin{aligned}\|\tilde{\boldsymbol{\psi}}_c^i\|_\Sigma^2 + \mathbf{e}_{a,i}^{D\Sigma G*}(\mathbf{U}_c^i D\Sigma D\mathbf{U}_c^{i*})^{-1}\mathbf{e}_{a,i}^{D\Sigma G} \\ = \|\tilde{\boldsymbol{\psi}}_c^{i-1}\|_{G^*\Sigma G}^2 + \mathbf{e}_{p,i}^{D\Sigma *}(\mathbf{U}_c^i D\Sigma D\mathbf{U}_c^{i*})^{-1}\mathbf{e}_{p,i}^{D\Sigma}\end{aligned} \tag{4.4.12}$$

Then, substituting (4.4.8) in (4.4.12) and rearranging the result, the following equation is obtained,

$$\|\tilde{\boldsymbol{\psi}}_c^i\|_\Sigma^2 = \|\tilde{\boldsymbol{\psi}}_c^{i-1}\|_{G^*\Sigma G}^2 - \mathbf{e}_{a,i}^{D\Sigma G*}\mathbf{X}\mathbf{e}_{c,i} - \mathbf{e}_{c,i}^*\mathbf{X}\mathbf{e}_{a,i}^{D\Sigma G} + \mathbf{e}_{c,i}^*\mathbf{Y}\mathbf{e}_{c,i} \tag{4.4.13}$$

where $\{\mathbf{X}, \mathbf{Y}\}$ are denoted by

$$\mathbf{X} = (\epsilon I_{TN} + \mathbf{U}_c^i \mathbf{U}_c^{i*})^{-1} \tag{4.4.14}$$

$$\mathbf{Y}^\Sigma = (\epsilon I_{TN} + \mathbf{U}_c^i \mathbf{U}_c^{i*})^{-1}\mathbf{U}_c^i D\Sigma D\mathbf{U}_c^{i*}(\epsilon I_{TN} + \mathbf{U}_c^i \mathbf{U}_c^{i*})^{-1} \tag{4.4.15}$$

Recall that $\mathbf{e}_{c,i} = \mathbf{U}_c^i G\tilde{\boldsymbol{\psi}}_c^{i-1} + \mathbf{v}_c^i$. Taking expectation of both sides of (4.4.14) and using the independence assumption for the noise signals,

the expression relating to the terms of $\{\tilde{\boldsymbol{\psi}}_c^i, \tilde{\boldsymbol{\psi}}_c^{i-1}\}$ can be obtained

$$E\|\tilde{\boldsymbol{\psi}}_c^i\|_\Sigma^2 = E\|\tilde{\boldsymbol{\psi}}_c^{i-1}\|_{\boldsymbol{\Sigma}'}^2 + E[\mathbf{v}_c^{i*}\mathbf{Y}^\Sigma\mathbf{v}_c^i] \tag{4.4.16}$$

$$\begin{aligned}\boldsymbol{\Sigma}' = &G^*\Sigma G - G^*\Sigma D\mathbf{U}_c^{i*}\mathbf{X}\mathbf{U}_c^i G \\ &-G^*\mathbf{U}_c^{i*}\mathbf{X}\mathbf{U}_c^i D\Sigma G + G^*\mathbf{U}_c^{i*}\mathbf{Y}^\Sigma\mathbf{U}_c^i G\end{aligned} \tag{4.4.17}$$

Evaluating expression (4.4.16) is challenging since the weighting matrix $\boldsymbol{\Sigma}'$, dependent on $\mathbf{U}_c^i$, is a random quantity. Recall the independence assumption of $\mathbf{U}_c^i$, which allows $\tilde{\boldsymbol{\psi}}_c^{i-1}$ to be independent of $\mathbf{U}_c^i$. In this way, (4.4.16) can be expressed as

$$E\|\tilde{\boldsymbol{\psi}}_c^i\|_\Sigma^2 = E\|\tilde{\boldsymbol{\psi}}_c^{i-1}\|_{E\boldsymbol{\Sigma}'}^2 + E[\mathbf{v}_c^{i*}\mathbf{Y}^\Sigma\mathbf{v}_c^i] \tag{4.4.18}$$

where the mean of the weighted matrix $\boldsymbol{\Sigma}'$ is given by $E\boldsymbol{\Sigma}' = \Sigma'$:

$$\begin{aligned}\Sigma' = &G^*\Sigma G - G^*\Sigma D E[\mathbf{U}_c^{i*}\mathbf{X}\mathbf{U}_c^i] G \\ &-G^* E[\mathbf{U}_c^{i*}\mathbf{X}\mathbf{U}_c^i] D\Sigma G + G^* E[\mathbf{U}_c^{i*}\mathbf{Y}^\Sigma\mathbf{U}_c^i] G\end{aligned} \tag{4.4.19}$$

In order to study the behaviour of the diffusion APA algorithm, the following moments in (4.4.18) and (4.4.19) must be evaluated:

$$E[\mathbf{U}_c^{i*}(\epsilon I_{TN} + \mathbf{U}_c^i\mathbf{U}_c^{i*})^{-1}\mathbf{U}_c^i] \tag{4.4.20}$$

$$E[\mathbf{U}_c^{i*}(\epsilon I_{TN} + \mathbf{U}_c^i\mathbf{U}_c^{i*})^{-1}\mathbf{U}_c^i D\Sigma D\mathbf{U}_c^{i*}(\epsilon I_{TN} + \mathbf{U}_c^i\mathbf{U}_c^{i*})^{-1}\mathbf{U}_c^i] \tag{4.4.21}$$

$$E[\mathbf{v}_c^{i*}(\epsilon I_{TN} + \mathbf{U}_c^i\mathbf{U}_c^{i*})^{-1}\mathbf{U}_c^i D\Sigma D\mathbf{U}_c^{i*}(\epsilon I_{TN} + \mathbf{U}_c^i\mathbf{U}_c^{i*})^{-1}\mathbf{v}_c^i] \tag{4.4.22}$$

To extract the matrix Σ from the expectation terms, a weighted variance relation is introduced by using $N^2M^2 \times 1$ column vectors:

$$\sigma = \text{bvec}\{\Sigma\} \text{ and } \sigma' = \text{bvec}\{\Sigma'\} \tag{4.4.23}$$

where bvec$\{\cdot\}$ denotes the block vector operator and Σ is an $NM \times NM$ block matrix, formed by

$$\Sigma = \begin{bmatrix} \Sigma_{11} \ldots \Sigma_{1n} \ldots \Sigma_{1N} \\ \Sigma_{21} \ldots \Sigma_{2n} \ldots \Sigma_{2N} \\ \vdots \quad \ddots \quad \vdots \quad \ddots \quad \vdots \\ \Sigma_{N1} \ldots \Sigma_{Nn} \ldots \Sigma_{NN} \end{bmatrix} \tag{4.4.24}$$

where each Σ_{mn} is an $M \times M$ matrix for $m, n = 1, \ldots, N$. A more detailed discussion of the block vectorization is presented in Appendix A. In addition, $\Sigma = \text{bvec}\{\sigma\}$ is also used to recover the original matrix Σ from σ. One property of the bvec$\{\cdot\}$ operator when working with the block Kronecker product [51], [52] is used in this work, namely,

$$\text{bvec}\{Q\Sigma P^{\text{T}}\} = (P \odot Q)\sigma \tag{4.4.25}$$

where $P \odot Q$ denotes the block Kronecker product of two block matrices P and Q and its mn-block is defined as

$$[P \odot Q]_{mn} = \begin{bmatrix} P_{mn} \otimes Q_{11} \ldots Q_{mn} \otimes Q_{1N} \\ \vdots \quad \ddots \quad \vdots \\ P_{mn} \otimes Q_{N1} \ldots P_{mn} \otimes Q_{NN} \end{bmatrix} \quad (NM^2 \times NM^2) \tag{4.4.26}$$

where $\{P_{mn}, Q_{mn}\}$ are $M \times M$ matrices for $m, n = 1, \ldots, N$. Using (4.4.25) to (4.4.19) after block vectorization, the following terms on the right side of (4.4.19) are given by

$$\text{bvec}\{G^*\Sigma G\} = (G^{\mathrm{T}} \odot G^*)\sigma \tag{4.4.27}$$

$$\begin{aligned}\text{bvec}\{G^*\Sigma D \cdot E\mathbf{Z} \cdot G\} &= (G^{\mathrm{T}} \odot G^*)\text{bvec}\{\Sigma D Z\} \\ &= (G^{\mathrm{T}} \odot G^*)(Z^{\mathrm{T}} \odot I_{NM})(D^{\mathrm{T}} \odot I_{NM})\sigma \end{aligned} \tag{4.4.28}$$

$$\begin{aligned}\text{bvec}\{G^* \cdot E\mathbf{Z} \cdot D\Sigma G\} &= (G^{\mathrm{T}} \odot G^*)\text{bvec}\{Z D \Sigma\} \\ &= (G^{\mathrm{T}} \odot G^*)(I_{NM} \odot Z)(I_{NM} \odot D)\sigma \end{aligned} \tag{4.4.29}$$

$$\begin{aligned}\text{bvec}\{G^* E[\mathbf{Z}D\Sigma D\mathbf{Z}]G\} &= (G^{\mathrm{T}} \odot G^*)\text{bvec}\{E[\mathbf{Z}D\Sigma D\mathbf{Z}]\} \\ &= (G^{\mathrm{T}} \odot G^*)E(\mathbf{Z}^{\mathrm{T}} \odot \mathbf{Z})(D^{\mathrm{T}} \odot D)\sigma \end{aligned} \tag{4.4.30}$$

where $\mathbf{Z} = \mathbf{U}_c^{i*}(\epsilon I_{TN} + \mathbf{U}_c^i \mathbf{U}_c^{i*})^{-1}\mathbf{U}_c^i$ and $Z = E\mathbf{Z}$. Therefore, a linear relation between the corresponding vectors $\{\sigma, \sigma'\}$ is formulated by

$$\sigma' = \mathrm{F}\sigma \tag{4.4.31}$$

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

$$\begin{aligned}F = (G^{\mathrm{T}} \odot G^*)\big[&I_{N^2M^2} - (Z^{\mathrm{T}} \odot I_{NM})(D^{\mathrm{T}} \odot I_{NM}) \\ &-(I_{NM} \odot Z)(I_{NM} \odot D) + \Pi \cdot (D^{\mathrm{T}} \odot D)\big]\end{aligned} \tag{4.4.32}$$

with $\Pi = E\big[(\mathbf{Z}^{\mathrm{T}} \odot \mathbf{Z})\big]$. For the sake of compactness, the bvec$\{\cdot\}$ notation is dropped from the subscripts and (4.4.18) becomes

$$E\|\tilde{\boldsymbol{\psi}}_c^i\|_\sigma^2 = E\|\tilde{\boldsymbol{\psi}}_c^{i-1}\|_{F\sigma}^2 + E[\mathbf{v}_c^{i*}\mathbf{Y}^{\Sigma}\mathbf{v}_c^i] \tag{4.4.33}$$

Let $\Lambda_v = E[\mathbf{v}_c^i \mathbf{v}_c^{i*}]$ denote a $TN \times TN$ diagonal matrix, whose entries are the noise variances $\{\sigma_{v,k}^2\}$ for $k = 1, \ldots, N$, and given by

$$\Lambda_v = \mathrm{diag}\{\sigma_{v,1}^2 I_{\mathrm{T}}, \ldots, \sigma_{v,N}^2 I_{\mathrm{T}}\}. \tag{4.4.34}$$

Using the independence assumption of noise signals, the last item in (4.4.33) can be written as

$$\begin{aligned} E[\mathbf{v}_c^{i*} \mathbf{Y}^{\Sigma} \mathbf{v}_c^i] &= \mathrm{Tr}(D \cdot E\boldsymbol{\Phi} \cdot D \cdot \Sigma) \\ &= \gamma^{\mathrm{T}} \sigma \end{aligned} \tag{4.4.35}$$

where $\boldsymbol{\Phi} = \mathbf{U}_c^{i*}(\epsilon I_{TN} + \mathbf{U}_c^i \mathbf{U}_c^{i*})^{-1} \Lambda_v (\epsilon I_{TN} + \mathbf{U}_c^i \mathbf{U}_c^{i*})^{-1} \mathbf{U}_c^i$ and

$$\begin{aligned} \gamma &= \mathrm{vec}\{D \cdot E\boldsymbol{\Phi} \cdot D\} \\ &= (D^{\mathrm{T}} \otimes D) \cdot \mathrm{vec}\{E\big[\mathbf{W}^* \Lambda_v \mathbf{W}\big]\} \\ &= (D^{\mathrm{T}} \otimes D) \cdot E\big[(\mathbf{W}^T \otimes \mathbf{W}^*)\big] \cdot \gamma_v. \end{aligned} \tag{4.4.36}$$

with $\mathbf{W} = (\epsilon I_{TN} + \mathbf{U}_c^i \mathbf{U}_c^{i*})^{-1} \mathbf{U}_c^i$ and $\gamma_v = \mathrm{vec}\{\Lambda_v\}$. Therefore, the mean-square behaviour of the diffusion APA algorithm is summarized by the following expressions:

$$E\|\tilde{\boldsymbol{\psi}}_c^i\|_\sigma^2 = E\|\tilde{\boldsymbol{\psi}}_c^{i-1}\|_{F\sigma}^2 + \gamma^{\mathrm{T}} \sigma \tag{4.4.37}$$

$$\begin{aligned} F = (G^{\mathrm{T}} \odot G^*)\big[I_{N^2M^2} - (Z^{\mathrm{T}} \odot I_{NM})(D^{\mathrm{T}} \odot I_{NM}) \\ -(I_{NM} \odot Z)(I_{NM} \odot D) + \Pi \cdot (D^{\mathrm{T}} \odot D)\big]. \end{aligned} \tag{4.4.38}$$

The global transient behaviour of the adaptive network is shown in $\{\tilde{\boldsymbol{\psi}}_c^{i-1}, \tilde{\boldsymbol{\psi}}_c^i\}$ by expressions (4.4.37) and (4.4.38), which can be used below to study the mean-square behavior of diffusion APA.

4.4.1 Mean and mean-square stability analysis

This section presents a discussion on the mean and mean-square stability of diffusion APA. According to $Gw_c^o = w_c^o$, the global data model (4.3.4) is used and w_c^o is abstracted from both sides of (4.3.7) to obtain

$$\begin{aligned}\tilde{\boldsymbol{\psi}}_c^i =& Gw_c^o - G\boldsymbol{\psi}_c^{i-1} \\ &-D\mathbf{U}_c^{i*}(\epsilon I_{TN} + \mathbf{U}_c^i\mathbf{U}_c^{i*})^{-1}[\mathbf{U}_c^i w_c^o + \mathbf{v}_c^i - \mathbf{U}_c^i G\boldsymbol{\psi}_c^{i-1}] \\ =&(I_{NM} - D\mathbf{U}_c^{i*}(\epsilon I_{TN} + \mathbf{U}_c^i\mathbf{U}_c^{i*})^{-1}\mathbf{U}_c^i)G\tilde{\boldsymbol{\psi}}_c^{i-1} \\ &+D\mathbf{U}_c^{i*}(\epsilon I_{TN} + \mathbf{U}_c^i\mathbf{U}_c^{i*})^{-1}\mathbf{v}_c^i. \end{aligned} \tag{4.4.39}$$

Then, taking expectation in both sides of (4.4.39) and using the independence assumption of $\mathbf{U}_c^i$, leads to

$$\boxed{E\tilde{\boldsymbol{\psi}}_c^i = [I_{NM} - DE(\mathbf{U}_c^{i*}(\epsilon I_{TN} + \mathbf{U}_c^i\mathbf{U}_c^{i*})^{-1}\mathbf{U}_c^i)]GE\tilde{\boldsymbol{\psi}}_c^{i-1}} \tag{4.4.40}$$

In order to guarantee the stability in the mean, all eigenvalues of BG should satisfy

$$\boxed{-1 < \lambda(BG) < 1} \tag{4.4.41}$$

where $B = [I_{NM} - DE(\mathbf{U}_c^{i*}(\epsilon I_{TN} + \mathbf{U}_c^i\mathbf{U}_c^{i*})^{-1}\mathbf{U}_c^i)]$ and $\lambda(A)$ denotes all eigenvalues of a matrix A. Therefore, the convergence of diffusion APA depends on the data moment $E(\mathbf{U}_c^{i*}(\epsilon I_{TN} + \mathbf{U}_c^i\mathbf{U}_c^{i*})^{-1}\mathbf{U}_c^i)$ and the topology matrix G. In the noncooperative algorithm, the mean evolution of the global weight error vector is formed by

$$E\tilde{\boldsymbol{\psi}}_c^i = B \cdot E\tilde{\boldsymbol{\psi}}_c^{i-1} \tag{4.4.42}$$

where the topology matrix G disappears due to $G = I_{NM}$ in a non-cooperation network. Using matrix 2-norms, the following result can be obtained,

$$|\lambda_{\max}(BG)| \leq |\lambda_{\max}(B)| \tag{4.4.43}$$

where $\lambda_{\max}(A)$ denotes the largest eigenvalue of matrix A. The derivation of (4.4.43) is presented in Appendix B. Moreover, with the purpose to show the convergence in the mean-square sense, F in (4.4.37) can be rewritten as

$$F = (G^{\mathrm{T}} \odot G^{*})H \tag{4.4.44}$$

where H is an Hermitian matrix, given by

$$H = I_{N^2M^2} - (Z^{\mathrm{T}} \odot I_{NM})(D^{\mathrm{T}} \odot I_{NM}) - (I_{NM} \odot Z)(I_{NM} \odot D) + \Pi \cdot (D^{\mathrm{T}} \odot D) \tag{4.4.45}$$

so that F should satisfy

$$\boxed{-1 < \lambda(F) < 1} \tag{4.4.46}$$

to guarantee the mean-square stability. In other words, the spectrum of F must be strictly inside the unit disc. For both the mean and the mean-square stability, the selection of μ_k and the network topology G must satisfy (4.4.41) and (4.4.46). In the similar way, the following result is obtained

$$|\lambda_{\max}((G^{\mathrm{T}} \odot G^{*})H)| \leq |\lambda_{\max}(H)|. \tag{4.4.47}$$

which is proved in Appendix B. Thus, the diffusion cooperation strategy leads to a stabilizing effect on the network. In addition, cooperation

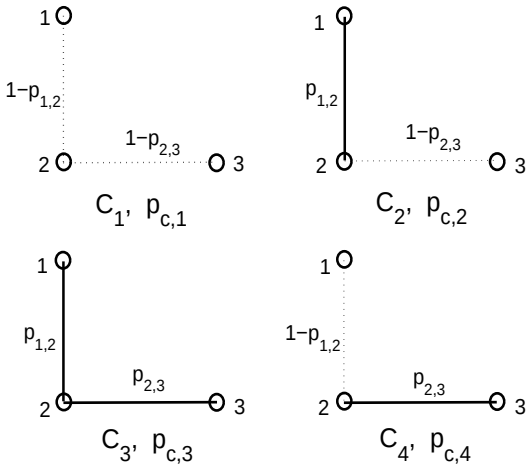

Figure 4.1. The subnetworks C_j with corresponding probability $p_{c,j}$

reduces the eigenmode of both the mean and mean-square weight error evolutions as compared with its noncooperative counterpart, which will be verified by simulation results.

4.5 Dynamic network topology

In the dynamic network topology model (i.e. probabilistic diffusion network as in [23]) it is assumed that the nature of links between nodes is determined randomly due to link failures or time delays. At time i, when the connection between undirected nodes k and l is established with probability $p_{k,l}$, the value of $\mathbf{c}_{k,l}(i)$ is set to be equal to $c_{k,l}$, where $c_{k,l}$ denotes the link weighted coefficient as in (4.2.6). Otherwise, $\mathbf{c}_{k,l}(i)$ is zero with probability $1 - p_{k,l}$. Therefore, for undirected nodes k and l, the elements of the corresponding combining matrix $\mathbf{C}_i$ are formed

by, for $k, l = 1, \ldots, N$:

$$\mathbf{c}_{k,l}(i) = \begin{cases} c_{k,l} & \text{with} \quad p_{k,l} \\ 0 & \text{with} \quad 1 - p_{k,l} \end{cases} \tag{4.5.1}$$

where $c_{k,l} = c_{l,k}$ and $p_{k,l} = p_{l,k}$. In an N-node network, n_l is defined as the maximum number of links and C_j is a subnetwork matrix for $j = 1, \ldots, 2^{n_l}$. A simple example is shown in Figure 4.1, which describes a 3-node network with $n_l = 2$. The probability $p_{c,j}$ of C_j depends on $\{p_{k,l}\}$, for instance, $p_{c,3} = p_{1,2}p_{2,3}$ is the probability of the subnetwork C_3. In this manner, the mean topology matrices $G = E\mathbf{G}_i$ and $G_c = E(\mathbf{G}_i^{\mathrm{T}} \odot \mathbf{G}_i^*)$ for the dynamic network topology are given by,

$$G = \sum_{j=1}^{2^{n_l}} p_{c,j} G_j \quad \text{and} \quad G_c = \sum_{j=1}^{2^{n_l}} p_{c,j} (G_j^{\mathrm{T}} \odot G_j^*) \tag{4.5.2}$$

which are therefore used to replace the corresponding terms in the above equations (4.4.40) and (4.4.44) to obtain the similar analysis of the mean and mean-square stability.

4.6 Simulations

In this section, computer simulations are carried out in a system identification scenario. In the following examples, a correlated input signal at a local node is generated as a Gaussian or uniform first-order Markov process, which allows the local covariance matrix $R_{u,k}$ to be a Toeplitz matrix with entries $r_k(m) = \sigma_{u,k}^2 \alpha_k^{|m|}$ for $m = 0, \ldots, M-1$, where α_k denotes the correlation index and the variance of the local input signal is set as $\sigma_{u,k}^2 = 1$. In addition, for all APA-based schemes the regularization parameter $\epsilon = 0.001$ is chosen as a small value to reduce its

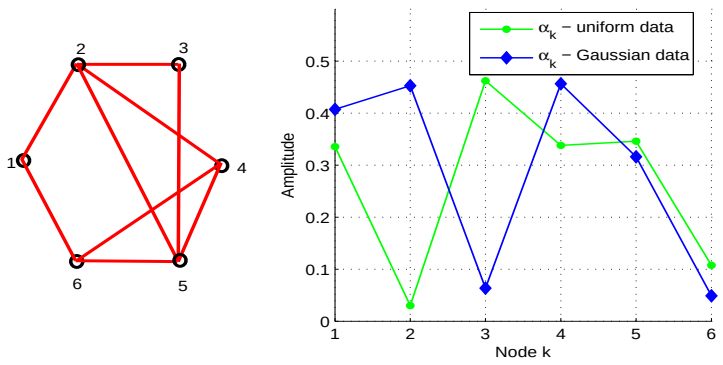

Figure 4.2. Example 1: Network topology (left) and node profile of the statistical setting for both Gaussian and Uniform data. The parameter α_k denotes the correlation index.

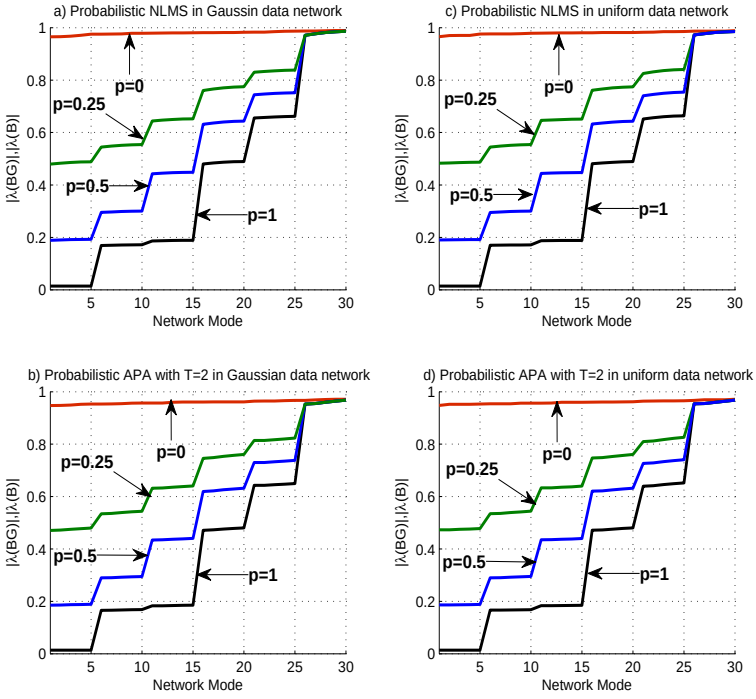

Figure 4.3. Network mode for Example 1: a) NLMS-based schemes in Gaussian data network; b) APA-based schemes with $T = 2$ in Gaussian data network; c) NLMS-based schemes in uniform data network; d) APA-based schemes with $T = 2$ in uniform data network. The value $p_{k,l} = p$ denotes the probability of the link between nodes k and l.

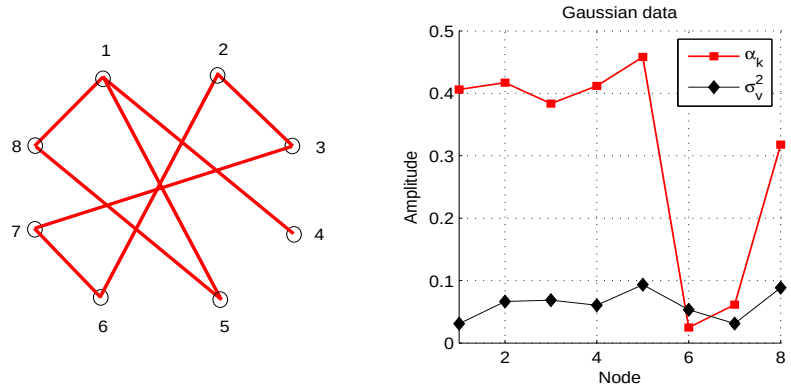

Figure 4.4. Example 2: network topology (left) and node profile of statistical setting for Gaussian data. The parameter α_k denotes the correlation index.

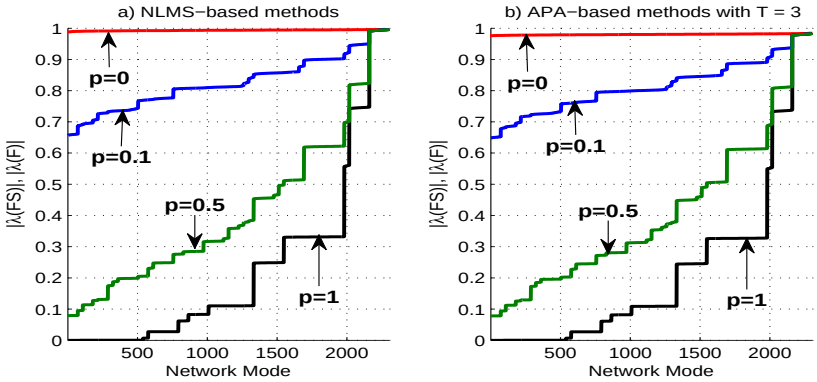

Figure 4.5. The N^2M^2 modes of F for Example 2: a) NLMS-based schemes; b) APA-based schemes with $T = 3$. The value $p_{k,l} = p$ denotes the probability of the link between nodes k and l.

effect on step-sizes. For the theoretical evaluations, all the expected terms $\{Z, \Pi\}$ are calculated by ensemble averaging. Moreover, all the coefficients of the adaptive filters within the network are initialized to zeros. An NLMS-based strategy can be regarded as a special case of an APA-based scheme with $T = 1$. All the simulated results in this work are obtained by averaging 100 Monte Carlo runs.

In Example 1, consider a 6-node network to seek a 5×1 unknown vector w^o by using $\mu_k = 0.1$. The background noise is set as $\sigma_{v,k}^2 =$

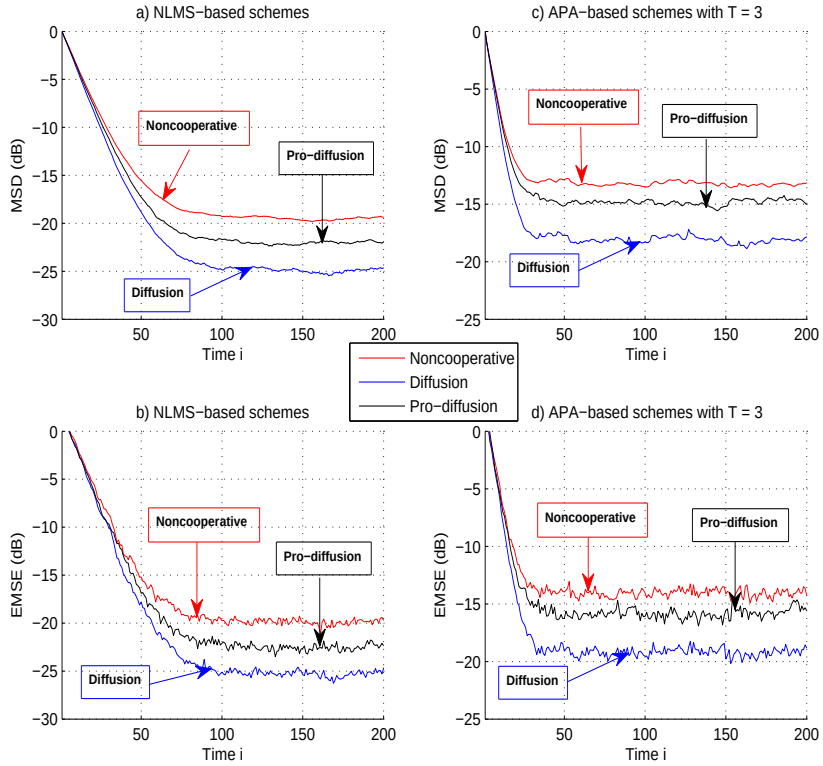

Figure 4.6. Global transient performance: a) MSD for NLMS-based schemes; b) EMSE for NLMS-based schemes; c) MSD for APA-based schemes with T = 3; d) EMSE for APA-based schemes with T = 3. Pro-diffusion denotes the probability diffusion algorithm.

0.001. In addition, $p_{k,l} = p$ denotes the probability of the link between nodes k and l. The ranks of APA-based schemes are set as $T = 1, 2$. Noncooperative APA, diffusion APA and probabilistic diffusion are implemented in this network. In addition, noncooperative and diffusion APA algorithms can be also regarded as the probabilistic diffusion APA with $p = 0$ and $p = 1$ respectively. On the left of Figure 4.2, it illustrates the network topology. In this network, coloured Gaussian or uniform data are used as input signals respectively. The statistical settings of these data are also shown in Figure 4.2. For the probabilistic APA algorithms, the probabilities $p = 0.1, 0.5$ are chosen. In Figure

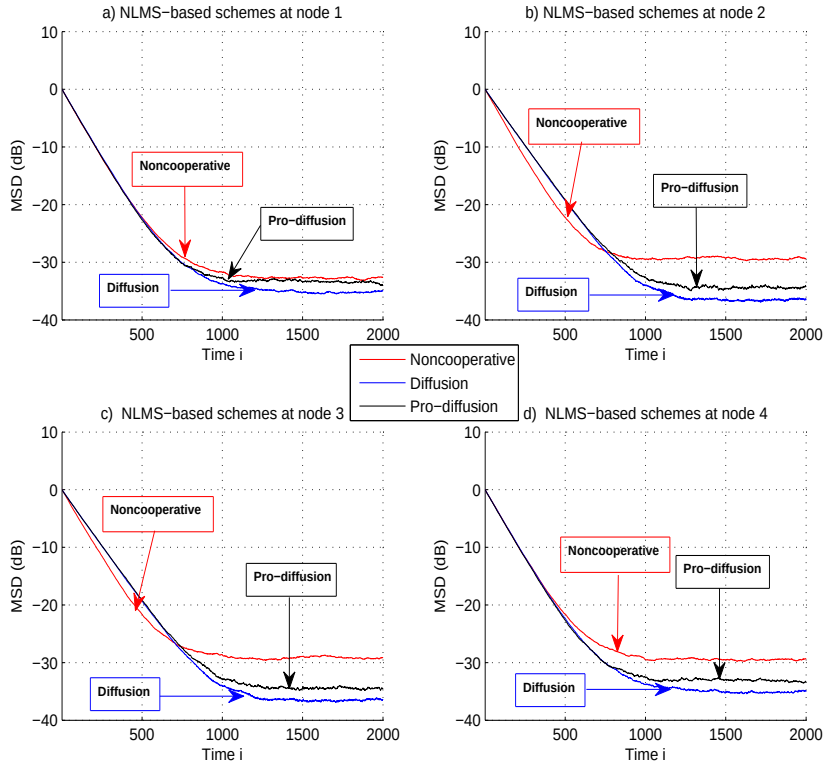

Figure 4.7. Local transient performance for various NLMS-based algorithms: a) MSD for NLMS-based schemes at node 1; b) MSD for NLMS-based schemes at node 2; c) MSD for NLMS-based schemes at node 3; d) MSD for NLMS-based schemes at node 4.

4.3, one can clearly see that the mode of diffusion cooperation reduces the eigenmodes of the mean weight error evolution, as compared with the noncooperative APA. In addition, as shown in Figure 4.3, a large probability decreases greatly the eigenmodes of mean weight error evolution. Moreover, the mean-square stability of diffusion APA is verified by the following example.

Figure 4.4 presents the network topology of Example 2, where various algorithms are performed to estimate a 6×1 unknown vector w^o in an 8-node network by using $\mu_k = 0.2$. The corresponding statis-

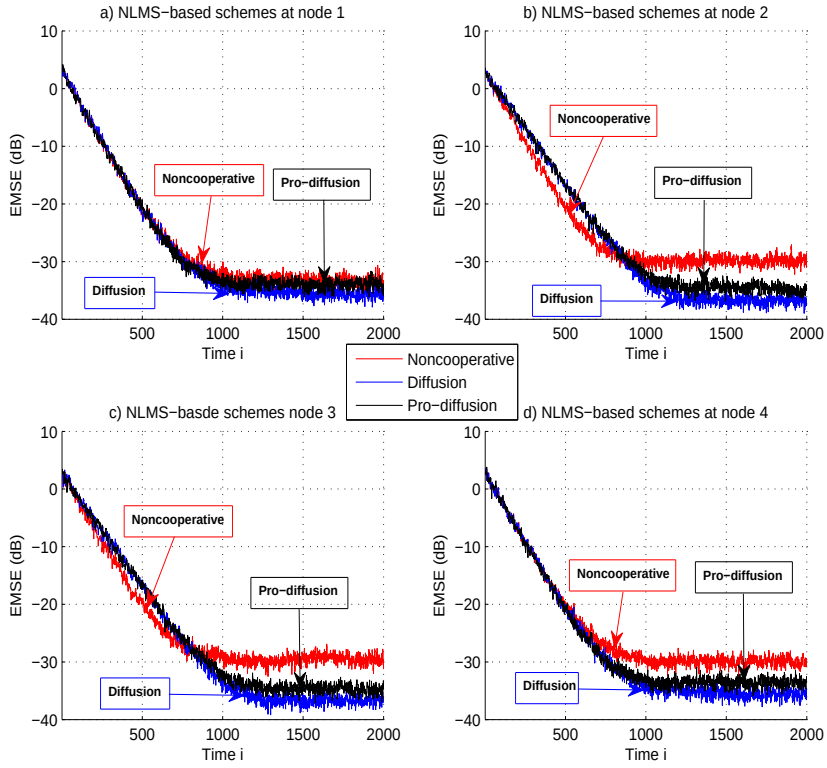

Figure 4.8. Local transient performance for various NLMS-based algorithms: a) EMSE for NLMS-based schemes at node 1; b) EMSE for NLMS-based schemes at node 2; c) EMSE for NLMS-based schemes at node 3; d) EMSE for NLMS-based schemes at node 4.

tical settings of Gaussian input and noise signals are plotted on the right of Figure 4.4. Figure 4.5 presents the mean-square eigenmodes for Example 2 in Figure 4.4. For the probabilistic APA algorithms, the probabilities $p = 0.1, 0.5$ are chosen. It is clear to see that cooperation also decreases the eigenmodes of the mean-square weight error evolution, which is verified by Figure 4.6. Figure 4.6 illustrates the global transient performance of various algorithms in 200 time samples. For the probabilistic APA algorithm, the probability $p = 0.1$ is chosen. One can clearly see that cooperation results in improvement of mean-square performance.

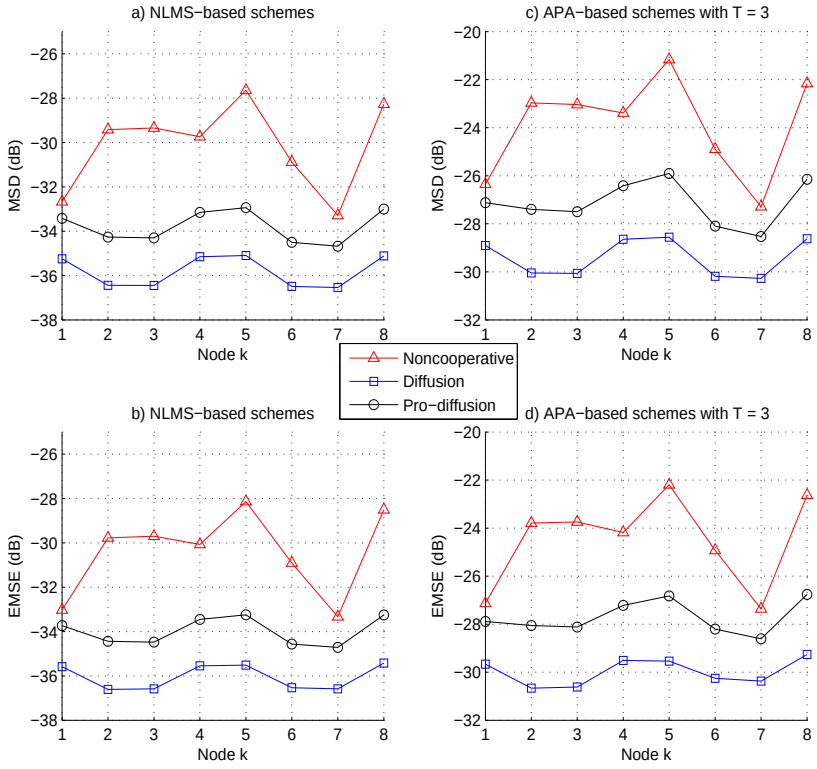

Figure 4.9. Local steady-state performance for various APA-based algorithms over the network by using $\mu_k = 0.02$: a) MSD for NLMS-based schemes; b) EMSE for NLMS-based schemes; c) MSD for APA-based schemes with T = 3; d) EMSE for APA-based schemes with T = 3.

Since the result of the comparison of various APA-based schemes is similar as that of various NLMS-based schemes, only the local transient performances of NLMS-based schemes are shown at nodes $k = 1, 2, 3, 4$ by using $\mu_k = 0.02$ in Figure 4.7-4.8. It should be noted that due to the cooperation strategy, the nodes present a strikingly similar mean-square performance. Figure 4.9 presents the local steady-state performance over the network by using step-size $\mu_k = 0.02$. Figure 4.10 shows the global steady-state performance as a function of the step-size in the range [0.004 0.6]. The results shown in both of these figures also

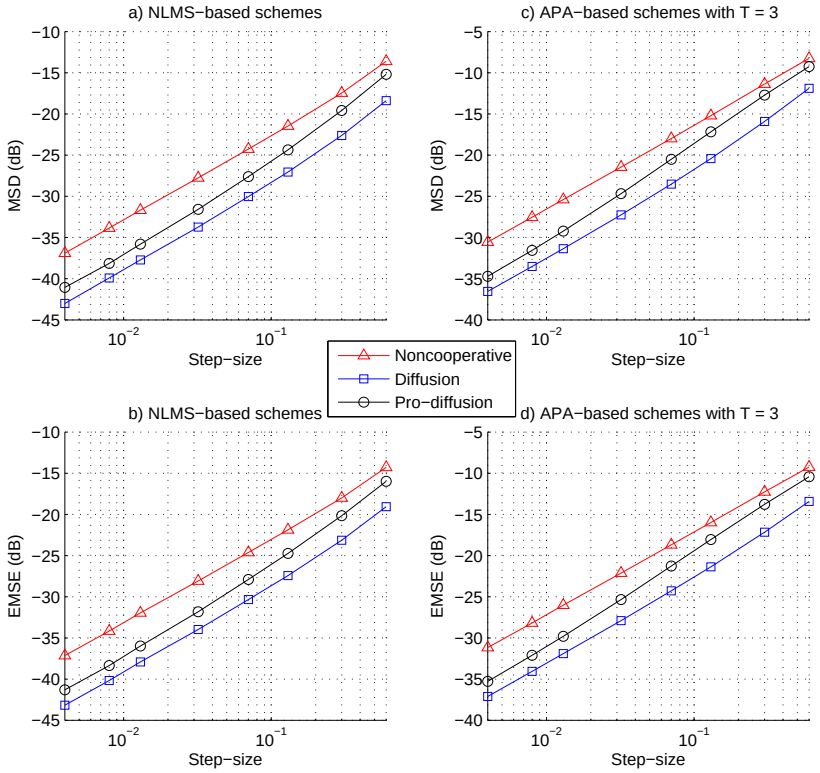

Figure 4.10. Global steady-state performance for various APA-based algorithms as a function of the step-size: a) MSD for NLMS-based schemes; b) EMSE for NLMS-based schemes; c) MSD for APA-based schemes with T = 3; d) EMSE for APA-based schemes with T = 3.

confirm the improvement in performance over the noncooperative case.

Another example with a larger network is also provided to carry out various algorithms by using $\mu_k = 0.1$. The topology of the network with 21 nodes is shown in Figure 4.11. Figure 4.12 a) presents the statistical setting of the network. In addition, the instantaneous network traffic is plotted in Figure 4.12 b), which illustrates the network mean usage is somewhat below its maximum capacity. The network usage denotes the number of available links at time i. Figure 4.13 shows that even with a very low probability, i.e. $p = 0.01$, the probabilistic diffusion APA

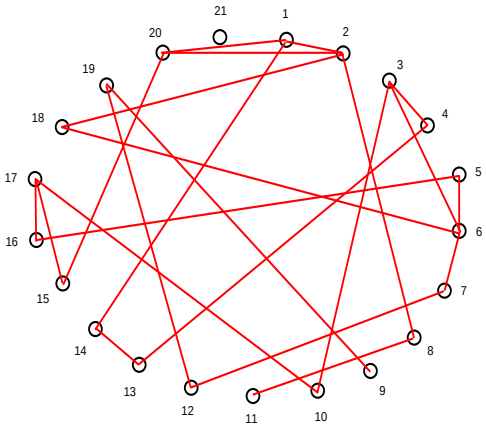

Figure 4.11. Network topology of Example 3.

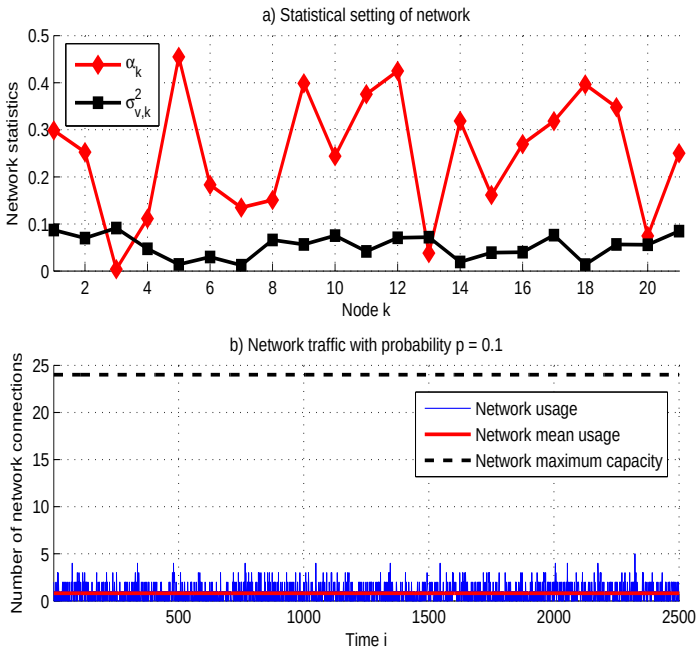

Figure 4.12. Example 3: a) Statistical setting of the network b) Network traffic;.

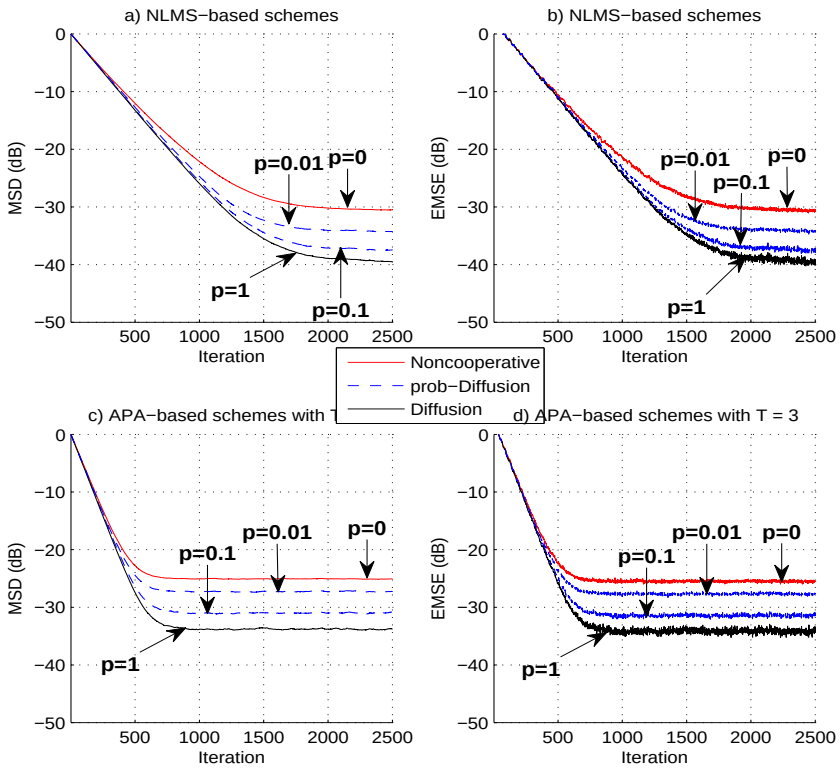

Figure 4.13. Example 3: Global transient performance comparison for Example 3: a) MSD for NLMS-based schemes b) EMSE for NLMS-based schemes c) MSD for APA-based schemes with T = 3 d) EMSE for APA-based scehemes with T = 3.

algorithm can achieve a reasonable improved performance as compared with the noncooperative scheme.

4.7 Conclusions

This chapter described a new diffusion adaptive learning algorithm based on APA for a distributed network and presented detailed performance analysis based on the weighted space-time energy conservation approach of Lopes and Sayed [25] under independence assumptions. This approach yields insight into the energy flow between nodes. One

main contribution of this chapter is to study the mean and mean-square stabilities of diffusion APA, which is consistent with simulations. Compared with the non-cooperative APA scheme, diffusion APA achieves a great improvement in terms of not only convergence rate but also steady-state performance.

4.8 Appendix A: block vectorization

Recall Σ is an $NM \times NM$ block matrix, given by,

$$\Sigma = \begin{bmatrix} \Sigma_{11} \ldots \Sigma_{1n} \ldots \Sigma_{1N} \\ \Sigma_{21} \ldots \Sigma_{2n} \ldots \Sigma_{2N} \\ \vdots \quad \ddots \quad \vdots \quad \ddots \quad \vdots \\ \Sigma_{N1} \ldots \Sigma_{Nn} \ldots \Sigma_{NN} \end{bmatrix} \tag{4.8.1}$$

with the block element matrix $\Sigma_{m,n}$ for $m, n = 1, \ldots, N$. The bvec$\{\cdot\}$ notation works in the following steps: firstly $\sigma_{mn} = \text{vec}\{\Sigma_{mn}\}$ forms an $M^2 \times 1$ column vector by stacking the successive columns of Σ_{mn} on top of each other; then the block columns are stacked on top of each other to obtain

$$\Sigma_n = \text{col}\{\Sigma_{1n}, \ldots, \Sigma_{Nn}\}, \quad (NM \times M) \tag{4.8.2}$$

and this is vectorized to obtain the corresponding column vector

$$\sigma_n = \text{col}\{\sigma_{\sigma_{1n},\ldots,\sigma_{Nn}}\}, \quad (NM^2) \tag{4.8.3}$$

then the $N^2M \times M$ stacked matrix Σ_v is formed by $\{\Sigma_n\}$,

$$\Sigma_v = \begin{bmatrix} \Sigma_1 \\ \Sigma_2 \\ \vdots \\ \Sigma_N \end{bmatrix} \tag{4.8.4}$$

which yields the final column vector

$$\sigma = \mathrm{col}\{\sigma_1, \ldots, \sigma_N\}, \quad (N^2M^2 \times 1) \tag{4.8.5}$$

4.9 Appendix B: mean and mean-square eigenmodes of diffusion APA

Using matrix 2-norms in (4.4.41), it has

$$\|BG\|_2 \leq \|B\|_2 \|G\|_2 \tag{4.9.1}$$

Given the block structure of $\mathbf{U}_c^i$, B can be regarded as a Hermitian matrix. Using $G = C \otimes I_M$, (4.9.1) becomes

$$|\lambda_{\max}(BG)| \leq \|C\|_2 \cdot |\lambda_{\max}(B)|. \tag{4.9.2}$$

where $\lambda_{\max}(A)$ is defined as the largest eigenvalue of A. The proper choice of stochastic and symmetric matrix C leads to $\|C\|_2 \leq 1$. For $\|C\|_2 = 1$, the result is obtained,

$$|\lambda_{\max}(BG)| \leq |\lambda_{\max}(B)| \tag{4.9.3}$$

which describes that the robustness of the cooperative scheme is greater than that of the noncooperative scheme. In a similar way as in [25], recall matrix 2-norms for (4.4.46), yielding,

$$\begin{aligned}\|F\|_2 &= \|(G^{\mathrm{T}} \odot G^*)H\|_2 \\ &\leq \|G^{\mathrm{T}} \odot G^*\|_2 \cdot \|H\|_2 \\ &= \|\Omega^{\mathrm{T}}(G^{\mathrm{T}} \otimes G^*)\Omega\|_2 \cdot \|H\|_2 \\ &\leq \|G^{\mathrm{T}} \otimes G^*\|_2 \cdot \|H\|_2 \\ &= \|G^{\mathrm{T}}\|_2 \cdot \|G^*\|_2 \cdot \|H\|_2 \end{aligned} \tag{4.9.4}$$

where the standard Kronecker product is used to express $(G^{\mathrm{T}} \odot G^*)$ with some permutation matrix Ω [53]. Since H is Hermitian and $G = C \otimes I_M$, the result is obtained

$$|\lambda_{\max}((G^{\mathrm{T}} \odot G^*)H)| \leq \|C\|_2^2 \cdot |\lambda_{\max}(H)| \tag{4.9.5}$$

which illustrates that the mean-square evolution of the overall system depends on the data matrix H and the chosen topology matrix C. Therefore, choosing $\|C\|_2^2 \leq 1$ can guarantee the robustness of the cooperative scheme over that of the noncooperative scheme. With $\|C\|_2^2 = 1$ as a Metropolis rule is used in the combiner, then (4.9.5) becomes

$$|\lambda_{\max}((G^{\mathrm{T}} \odot G^*)H)| \leq |\lambda_{\max}(H)|. \tag{4.9.6}$$

Chapter 5

TAP-LENGTH ADAPTATION WITHIN LMS LEARNING ALGORITHMS

5.1 Overview of VTLMS algorithms

The LMS adaptive algorithm has been extensively used as a consequence of its simplicity and robustness [4], [5], [6]. In many applications of LMS type algorithms, the tap-length of the adaptive filter is kept fixed. However, in certain situations the tap-length of the optimal filter is unknown or variable. According to the analysis in [54], [55], the MSE of the adaptive filter is likely to increase if the tap-length is under-modelled. To avoid such a situation, a sufficiently large tap-length is required to be chosen in the steady-state. However, the computational cost and the EMSE of the LMS algorithm will increase if the tap-length is too large, thus a VTLMS algorithm is needed to find a proper choice of the tap-length. Several algorithms based on the concept of variable tap-length [11] have been studied in recent years [12], [14], [16], [17], [18], [56], [57]. The algorithm in [11] compares the current MSE of a deficient tap-length adaptive filter to

the pre-estimated minimum MSE for a specific tap-length to improve convergence rate. In [12], a variable tap-length algorithm with a pre-calculated time constant is proposed. However, both algorithms were initial attempts at enhancing the convergence behavior of the MSE in an environment where the tap-length of the system is known. In time-varying scenarios, where the system to be identified has changing length, Riera-Palou et al. presented an algorithm which relies on the concept of partitioned segments, the number of which must be carefully chosen dependent on application [13]. During learning in variable tap-length adaptive filters, when the adaptation noise is low, the "wandering" problem is encountered, that is, the tap-length wanders within a range that is always greater than the optimum tap-length [14]; this issue is also addressed in Section 5.3 through careful selection of the leakage factor for the FT algorithm. The algorithms in [16] were presented to make the estimated tap-length converge to the optimum tap-length in the mean. However, both algorithms suffer from slow tap-length convergence in certain scenarios.

Gong and Cowan [17] introduce a low-complexity FT algorithm based on instantaneous errors, which obtains improved convergence properties. A convex combination structure of the FT algorithm has been proposed in [56] to establish the optimal tap-length in high noise conditions, in which two filters are updated simultaneously with different parameters, so that the overall filter can obtain both a rapid convergence rate from the fast filter and a smooth curve for the steady-state tap-length from the slow filter. In [18], a variable tap-length natural gradient blind equalization algorithm based on the FT algorithm is proposed, which gives a good compromise between steady-state per-

formance and computational complexity. A steady-state performance analysis of the FT algorithm is provided in [57], which also gives a guideline for the parameter choice of the FT algorithm. As analyzed in [17], the FT algorithm is more robust and has lower computational complexity when compared with other methods. For expressing the analysis clearly, when the initial tap-length of the adaptive filter is smaller than the converged tap-length of the adaptive filter, it is termed as an increasing tap-length (ITL) estimation; when the initial tap-length is larger than the converged tap-length, it is named as a decreasing tap-length (DTL) estimation. The next section therefore will give a brief introduction of the FT algorithm.

5.2 The FT algorithm

The FT algorithm based on the tap-length adaptation and LMS-type rule is designed to find the optimal tap-length of the adaptive filter. In agreement with most approaches used to derive algorithms for adaptive filtering the design problem is related to the optimization of a certain criterion that is dependent on the tap-length. For convenience, the LMS algorithm is formed within a system identification framework, in which the unknown filter w^o has an unknown tap-length M which is to be identified. In this case, the desired signal $d(i)$ is observed from the model:

$$d(i) = u_i w^o + v(i) \tag{5.2.1}$$

where u_i is a $1 \times M$ row input vector, $v(i)$ indicates a zero mean additive noise term uncorrelated with the input u_i and i denotes the discrete time index. All quantities are assumed to be real valued for convenience

of development but extension to complex values is straightforward. It is well known in [4], [5], [6] that the traditional LMS algorithm computes w_i via

$$w_i = w_{i-1} + \mu u_i^{\mathrm{T}}(d(i) - u_i w_{i-1}) \tag{5.2.2}$$

where μ is the step-size of the tap weight update.

In order to calculate the unknown tap-length of the adaptive filter, variable tap-length schemes can be employed. But, as mentioned in [16], MSE-based variable tap-length schemes can lead to under estimation of the tap-length in certain applications. In order to overcome this problem, a parameter Δ, which is a positive integer and sufficiently large to avoid suboptimum tap-lengths, has been introduced in [16]. Suppose the estimated length of the variable tap-length adaptive filter is a fixed value and denoted by N, which satisfies $N > \Delta$. Thus, the segmented error is constructed, as in [17] by,

$$e_{N-\Delta}^{(N)}(i) = d(i) - u_i(1 : N - \Delta)w_{i-1}(1 : N - \Delta) \tag{5.2.3}$$

where $u_i(1 : N - \Delta)$ and $w_{i-1}(1 : N - \Delta)$ are vectors consisting of the first $N - \Delta$ coefficients of u_i and w_{i-1} respectively. In addition, let $e_{N-\Delta}^{(N)}(i)$ denote the realization of random quantity $\mathbf{e}_{N-\Delta}^{(N)}(i)$. As $i \to \infty$, the steady-state segmented MSE is further defined by,

$$J_{N-\Delta}^{(N)} = E\left[(\mathbf{e}_{N-\Delta}^{(N)}(\infty))^2\right]. \tag{5.2.4}$$

The cost function to search for the optimum tap-length is described as:

$$\min\{N | J_{N-\Delta}^{(N)} - J_N^{(N)} \leq \varepsilon\} \tag{5.2.5}$$

where ε is a predetermined small positive value selected according to the requirements of the adaptive filter. The minimum N that satisfies (5.2.5) is then chosen as the optimum tap-length for the adaptive filter. A detailed description of this criterion and another similar criterion can be found in [17].

Gradient-based methods can be used to estimate the optimum tap-length on the basis of (5.2.5). However, the tap-length that should be used in the adaptive filter structure must be an integer, and this constrains the adaptation of the tap-length. Different approaches have been applied to solve this problem [13], [14], [15], [16] , [17]. In [17], the concept of "pseudo fractional tap-length", denoted by $l_f(i)$, is utilized to make instantaneous tap-length adaptation possible. The update of the fractional tap-length is as follows:

$$l_f(i+1) = l_f(i) - \alpha + \beta \cdot \eta_{L(i)}(i) \tag{5.2.6}$$

where α is the leakage factor, β is the step-size for the fractional tap-length update [17] and $\eta_{L(i)}(i)$ is the difference between the segmented MSE and the full MSE, given by,

$$\eta_{L(i)}(i) = \left[e^{(L(i))}_{L(i)-\Delta}(i)\right]^2 - \left[e^{(L(i))}_{L(i)}(i)\right]^2 \tag{5.2.7}$$

where $L(i)$ is constrained to be not less than $L\text{min}$ with $L\text{min} > \Delta$, since any tap-length below it cannot be calculated in (5.2.7). One should note that $l_f(i)$ is no longer constrained to be an integer, and the tap-length $L(i+1)$, which will be used in the adaptation of the filter weights in the next iteration, is obtained from the fractional tap-length

$l_f(i)$ as (1.3.3):

$$L(i+1) = \begin{cases} \mathrm{Rd}\left(l_f(n)\right) & \text{if } |L(i) - l_f(i+1)| > \nu \\ L(i) & \text{otherwise.} \end{cases} \tag{5.2.8}$$

where $\mathrm{Rd}(\cdot)$ rounds the embraced value to the nearest integer and the parameter ν is a small integer. Similar to the step-size in the LMS algorithm, a large parameter β will speed up the convergence rate of the tap-length, but will result in a large fluctuation of the steady-state tap-length. Once the tap-length fluctuates under the optimal tap-length, extra error will be introduced. This is named as the under-modelling phenomenon for VT algorithms. On the other hand, a small parameter β can obtain a small fluctuation of the steady-state tap-length, but leads to a slow convergence rate of both the tap-length and EMSE. Compared with the leakage factor used in the LMS algorithm [58], [59], [60], the objective of the leakage factor α introduced in the FT algorithm is to prevent the adaptive tap-length from increasing to an undesirable large value. A large α leads to a slow convenience rate for the ITL estimation, but a small α may result in the "wandering" problem for the DTL estimation.

Motivated by the FT algorithm, two new VTLMS algorithms are therefore developed to obtain the improved performance over the original FT algorithm.

5.3 A novel adaptive leakage FT algorithm

Through analysis provided in the following subsection, the converged difference between the segmented MSE of a filter formed from a number

of the initial coefficients of an adaptive filter, and the MSE of the full adaptive filter, is confirmed as a function of the tap-length of the adaptive filter to be monotonically non-increasing under the assumption that the final element of the unknown filter is significantly different from zero. This analysis also provides a systematic way to select the key parameters in the FT learning algorithm, first proposed by Gong and Cowan, to ensure convergence to permit calculation of the true tap-length of the unknown system and motivates the need for adaptation in the leakage factor during learning.

5.3.1 Analysis of the tap-length update function

As defined earlier, boldface letters are used to denote random variables. Let $\{d(i), u_i, v(i)\}$ denote realizations of the real random quantities $\{\mathbf{d}(i), \mathbf{u}_i, \mathbf{v}(i)\}$. To simplify the analysis, three assumptions are therefore made:

A1) The input $\mathbf{u}(i)$ is a zero-mean stationary white signal with variance σ_u^2 and the input vector $\mathbf{u}_i$ is uncorrelated with $\mathbf{u}_j$ for $i \neq j$.

A2) The background noise $\mathbf{v}(i)$ is also a zero-mean stationary white signal with variance σ_v^2 and uncorrelated with $\mathbf{v}_j$ for $i \neq j$ and $\mathbf{u}(j)$ for all j.

A3) The final optimum weight vector coefficient is sufficiently different from zero, $w^o(M) \neq 0$.

After the initial convergence, assuming close proximity to the optimal adaptive filter length and small misadjustment, the converged tap-length $E[\mathbf{L}(\infty)]$ should be designed to vary within $[M+\Delta-1,\ M+\Delta]$, from which the true tap-length of the unknown system M can be found.

When $E[\mathbf{L}(\infty)]$ equals to $M+\Delta-1$, the fractional tap-length function should be increased towards $M+\Delta$, namely $E[\boldsymbol{\eta}_{M+\Delta-1}(\infty)] > \alpha/\beta$. On the other hand, when $E[\mathbf{L}(i)] = M+\Delta$, the fractional tap-length should be decreased, namely $E[\boldsymbol{\eta}_{M+\Delta-1}(\infty)] < \alpha/\beta$. Therefore, the problem becomes how to select the parameters α and β in equation (5.2.6) to satisfy the above requirements. In order to make the appropriate selection of α/β, the performance of $\boldsymbol{\eta}_{M+\Delta-1}(i)$ should be evaluated, which is,

$$\boldsymbol{\eta}_{M+\Delta-1}(i) = \left(\mathbf{e}_{M-1}^{(M+\Delta-1)}(i)\right)^2 - \left(\mathbf{e}_{M+\Delta-1}^{(M+\Delta-1)}(i)\right)^2 \tag{5.3.1}$$

Taking statistical expectation of both sides, expression (5.3.1) becomes

$$E[\boldsymbol{\eta}_{M+\Delta-1}(i)] = E\left[\left(\mathbf{e}_{M-1}^{(M+\Delta-1)}(i)\right)^2\right] - E\left[\left(\mathbf{e}_{M+\Delta-1}^{(M+\Delta-1)}(i)\right)^2\right] \tag{5.3.2}$$

Due to the assumption **A2**, the following expression is obtained

$$E[\boldsymbol{\eta}_{M+\Delta-1}(i)] = A + \sigma_u^2 w^o(M)^2 - B \tag{5.3.3}$$

where $w^o(M)$ is the final coefficient of the tap weight in the unknown system, B is formulated by,

$$B = E\left[(\mathbf{u}_i(1:M+\Delta-1)[w^o(1:M+\Delta-1) - \mathbf{w}_i(1:M+\Delta-1)])^2\right] \tag{5.3.4}$$

and A is constructed as,

$$\begin{aligned} A=&E\left[(\mathbf{u}_i(1:M-1)[w^o(1:M-1)-\mathbf{w}_{i-1}(1:M-1)])^2\right] \\ &-2w^o(M)E\left[\mathbf{u}_i(M)\mathbf{u}_i(1:M-1)\mathbf{w}_{i-1}(1:M-1)\right] \\ &+2w^o(M)E\left[\mathbf{u}_i(M)\mathbf{u}_i(1:M-1)\right]w^o(1:M-1) \end{aligned} \tag{5.3.5}$$

According to the assumption **A1**, $\mathbf{u}_i(M)$ is uncorrelated with $\mathbf{u}_i(1:M-1)$ and $\mathbf{w}_{i-1}(1:M-1)$. Thus, the result of the last two items on the right side of equation (5.3.5) is zero. Therefore, as $i \to \infty$, equation (5.3.3) becomes

$$E[\boldsymbol{\eta}_{M+\Delta-1}(\infty)]=J_{M-1,\ excess}^{(M+\Delta-1)}-J_{M+\Delta-1,\ excess}^{(M+\Delta-1)}+\sigma_u^2 w^o(M)^2 \tag{5.3.6}$$

where $J_{M-1,\ excess}^{(M+\Delta-1)}$ and $J_{M+\Delta-1,\ excess}^{(M+\Delta-1)}$ are used to indicate the steady-state EMSE, namely,

$$J_{M-1,excess}^{(M+\Delta-1)}=E\left[(\mathbf{u}_i(1:M-1)\bar{\mathbf{w}}_i(1:M-1))^2\right] \tag{5.3.7}$$

$$J_{M+\Delta-1,excess}^{(M+\Delta-1)}=E\left[(\mathbf{u}_i(1:M+\Delta-1)\bar{\mathbf{w}}_i(1:M+\Delta-1))^2\right] \tag{5.3.8}$$

where the error vectors $\{\bar{\mathbf{w}}_i(1:M-1), \bar{\mathbf{w}}_i(1:M+\Delta-1)\}$ are defined by

$$\bar{\mathbf{w}}_i(1:M-1)=w^o(1:M-1)-\mathbf{w}_{i-1}(1:M-1) \tag{5.3.9}$$

$$\bar{\mathbf{w}}_i(1:M+\Delta-1)=w^o(1:M+\Delta-1)-\mathbf{w}_{i-1}(1:M+\Delta-1). \tag{5.3.10}$$

For $E[\mathbf{L}(i)] = M+\Delta$, after the same manipulations as in (5.3.2)-(5.3.6), the converged $E[\boldsymbol{\eta}_{M+\Delta}(\infty)]$ can be obtained as,

$$\begin{aligned} E[\boldsymbol{\eta}_{M+\Delta}(\infty)] &= J^{(M+\Delta)}_{M,\ excess} - J^{(M+\Delta)}_{M+\Delta,\ excess} \\ &= -\left[\mathbf{u}_i(M+1:M+\Delta)\mathbf{w}_i(M+1:M+\Delta)\right]^2 \end{aligned} \quad (5.3.11)$$

the result of which is a small negative value. Therefore, given that the leakage parameter is positive, $E[\boldsymbol{\eta}_{M+\Delta-1}(\infty)]$ should be bigger than zero, namely $w^o(M)^2 > J^{(M+\Delta-1)}_{M-1,\ excess} - J^{(M+\Delta-1)}_{M+\Delta-1,\ excess}$ as required in **A3**. Only when the ratio of the optimum parameters α/β is chosen bigger than zero and smaller than the value of $E[\boldsymbol{\eta}_{M+\Delta-1}(\infty)]$, can the resulting steady-state tap-length of the FT algorithm be used to find the true tap-length of the system.

However, in the FT algorithm $E[\mathbf{L}(\infty)]$ may converge to a value within $(M,\ M+\Delta]$ when the parameters α and β are not chosen appropriately. Thus, the analysis for this case is developed. The expectation of the MSE difference, $E[\boldsymbol{\eta}_{L(\infty)}]$ is given by,

$$E[\boldsymbol{\eta}_L(i)] = E\left[\left(\mathbf{e}^{(L)}_{L-\Delta}(i)\right)^2\right] - E\left[\left(\mathbf{e}^{(L)}_{L}(i)\right)^2\right] \quad (5.3.12)$$

where $M < L \leq M+\Delta$. Applying the same manipulations as in the earlier analysis, as $i \to \infty$, expression (5.3.10) becomes

$$\begin{aligned} E[\boldsymbol{\eta}_L(\infty)] &= J^{(L)}_{L-\Delta,\ excess} - J^{(L)}_{L,\ excess} \\ &\quad +(M+\Delta-L)||w^o(L-\Delta+1:M)||^2. \end{aligned} \quad (5.3.13)$$

The steady-state EMSE of the LMS algorithm in [6] is formulated as

$$J_{LMS,\ excess} = \frac{\mu L \sigma_u^2 \sigma_v^2}{2 - \mu L \sigma_u^2} \tag{5.3.14}$$

According to the stability condition of the LMS algorithm (see [4]), the value of $\mu L \sigma_u^2$ is chosen significantly smaller than 2. As a result, the EMSE of the LMS algorithm (5.3.14) becomes

$$J_{LMS,\ excess} \approx \frac{\mu L \sigma_u^2 \sigma_v^2}{2} \tag{5.3.15}$$

In order to facilitate the analysis, it is assumed that with a small μ the difference between the segmented EMSE and the full length EMSE is trivial compared to $(M + \Delta - L)\sigma_u^2 ||w^o(L - \Delta + 1 : M)||^2$. Thus, equation (5.3.13) is approximated as

$$E[\boldsymbol{\eta}_L(\infty)] \approx (M + \Delta - L)\sigma_u^2 ||w^o(L - \Delta + 1 : M)||^2 \tag{5.3.16}$$

From the above analysis, it is clear that $E[\boldsymbol{\eta}_L(\infty)]$ is a monotonic non-increasing function with respect to the tap-length L, which is within the range $(M,\ M + \Delta]$. From the above analysis, when the value of α/β is close to zero the converged $E[L(\infty)]$ will vary within $[M + \Delta - 1, M + \Delta]$, which can be utilized to calculate the true tap-length of the unknown system. However, when α/β is chosen too small, as observed in [14] the "wandering" problem occurs in the DTL estimation, which results in a very slow convergence rate for the adaptation of the fractional tap-length of the adaptive filter. Careful choice of α/β is therefore crucial for successful operation of the FT algorithm, and this motivates the work of the development of a new FT strategy with

leakage factor adaptation to satisfy these requirements in both DTL and ITL estimations. In the proposed algorithm, a variable leakage factor based on the squared smoothed error is used to improve the convergence behavior of the fractional tap-length as compared to the original FT algorithm.

5.3.2 Proposed novel algorithm and performance analysis

Due to the analysis in the above subsection, a novel algorithm based on the FT algorithm [17] is therefore proposed to control leakage factor adaptation. The objective is to allow the leakage factor $\alpha(i)$ to be big enough during the learning stage to avoid the "wandering" problem but to approach the desired value in the steady-state to limit the converged range of $E[L(\infty)]$. To achieve this objective, the use of smoothed errors [61], [62], [63] is introduced in the proposed algorithm, which exploits an estimate of $\mathbf{e}(i)$ to control leakage factor update. To facilitate analysis, the full MSE $\mathbf{e}_{L(i)}^{(L(i))}(i)$ is replaced by $\mathbf{e}(i)$. Let $\{e(i)\}$ denote the realizations of $\{\mathbf{e}(i)\}$. Such an estimate is a time average of $e(i)$, formulated as,

$$p(i) = \rho p(i-1) + (1-\rho)e(i) \tag{5.3.17}$$

where ρ ($0 \ll \rho < 1$) is a forgetting factor and the initial value is chosen as $p(0) = 0$. The positive parameter ρ governs the time averaging window to reduce the effect of the distant past and adapt to the current statistics. In the steady-state, as the error autocorrelation approaches zero, the resulting $\alpha(i)$ decreases to the desired value. Therefore, the

proposed leakage factor update is given by

$$\alpha(i)=\begin{cases}\frac{p^2(i)}{p^2(i)+\delta} & \text{if } \alpha(i)<\alpha_{\max}\\ \alpha_{\max} & \text{otherwise}\end{cases} \tag{5.3.18}$$

$$l_f(i+1)=l_f(i)-\alpha(i)+\beta\cdot\eta_{L(i)}(i) \tag{5.3.19}$$

where δ indicates a positive constant and $\eta_{L(i)}(i)$ is the difference of the errors as defined in (5.3.12). As a result of the averaging operation, the instantaneous behaviour of the leakage factor will be smoother.

For convenience of analysis, the following assumption is established:

A4) In the steady-state, if the EMSE is much smaller than the MSE, the full error signal $e(i)$ is approximately equal to the noise signal $v(i)$ in the system identification case.

Since the squared norm of the smoothed errors $p^2(i)$ plays a key role for the proposed algorithm, a steady-state performance analysis of this term is firstly developed. From (5.3.17), the recursive expression of $p(i)$ can be formed as

$$p(i)=(1-\rho)\sum_{j=1}^{i}\rho^{i-j}e(j) \tag{5.3.20}$$

with $p(0)=0$ and $e(0)=1$. Let $p(i)$ denote the realization of $\mathbf{p}(i)$. The expected performance of the squared norm of $\mathbf{p}(i)$ can be obtained

$$E[\mathbf{p}^2(i)]=(1-\rho)^2\sum_{j=1}^{i}\sum_{k=1}^{i}C_{j,k} \tag{5.3.21}$$

where $C_{j,k}$ is defined as

$$C(j,k)=E\left[\rho^{i-j}\rho^{i-k}\mathbf{e}(j)\mathbf{e}(k)\right] \tag{5.3.22}$$

The following analysis will only discuss the item in the steady-state, i.e. j and k are both steady-state time indices. The MSE is given by [64]

$$E[\mathbf{e}^2(i)] = \epsilon_{\min} + \epsilon_{ex}(i) \tag{5.3.23}$$

where $\epsilon_{\min}$ indicates the minimum value of the MSE and $\epsilon_{ex}(i)$ denotes the EMSE. In the steady-state, when $j = k$, (5.3.22) therefore becomes

$$C(j,j) \approx \rho^{2i-2j}(\epsilon_{\min} + \epsilon_{ex}(\infty)) \tag{5.3.24}$$

From assumption **A4**, with $\epsilon_{ex}(\infty) \ll E[\mathbf{e}^2(\infty)]$ and $\epsilon_{\min} = \sigma_v^2$, (5.3.24) becomes

$$C(j,j) \approx \rho^{2i-2j}\sigma_v^2 \tag{5.3.25}$$

where σ_v^2 is the variance of the noise signal. When $j \neq k$, the samples of errors can be assumed uncorrelated, i.e., $E[\mathbf{e}(j)\mathbf{e}(k)] = 0$, which yields,

$$C(j,k) = 0 \tag{5.3.26}$$

Substitute (5.3.25) and (5.3.26) into (5.3.21) to obtain

$$\lim_{i\to\infty} E[\mathbf{p}^2(\infty)] \approx (1-\rho)^2 \sum_{j=n}^{i} \rho^{2(i-j)}\sigma_v^2 \tag{5.3.27}$$

where n is the time index when the system is assumed in steady-state. Since $0 \ll \rho < 1$ holds, expression (5.3.27) is simplified as

$$E[\mathbf{p}^2(\infty)] = \frac{1-\rho}{1+\rho}\sigma_v^2 \tag{5.3.28}$$

Therefore, when $\delta \gg\ p^2(i)$ is chosen as $i \to \infty$, according to (5.3.18),

the converged variable leakage factor is obtained

$$\alpha(\infty) \approx \frac{(1-\rho)\sigma_v^2}{(1+\rho)\delta} \tag{5.3.29}$$

In addition, when δ is smaller than the square of the instantaneous errors and $\alpha_{\max}$ is big enough, (5.3.18) leads to $\alpha(i+1) = \alpha_{\max}$ during the initial learning. Therefore, proper selection of ρ and δ enables the variable leakage factor to become large initially but to converge to the desired value in the steady-state.

It is clear that as compared with the original FT learning algorithm, two additional update equations (5.3.17) and (5.3.18) are involved in the proposed algorithm. The added complexity is therefore one division, two additions and three multiplications per iteration.

The following simulation study is presented in order to verify the advantage of the proposed algorithm.

5.3.3 Simulations

This section shows results of the computer simulations which compare the performances of the original FT learning algorithm and the novel algorithm. Experiments are performed in a system identification model. Although the performance analysis is studied for the stationary environment, the simulations are carried out in the nonstationary environment, where the system weight vector abruptly changes at a certain time. The input signal $u(i)$ is a zero-mean white-noise Gaussian sequence with $\sigma_u^2 = 1$ and the background noise is also a zero mean white-noise Gaussian sequence with $\sigma_v^2 = 0.001$. The coefficients of the adaptive filter were initialized with zeros. Simulated results are

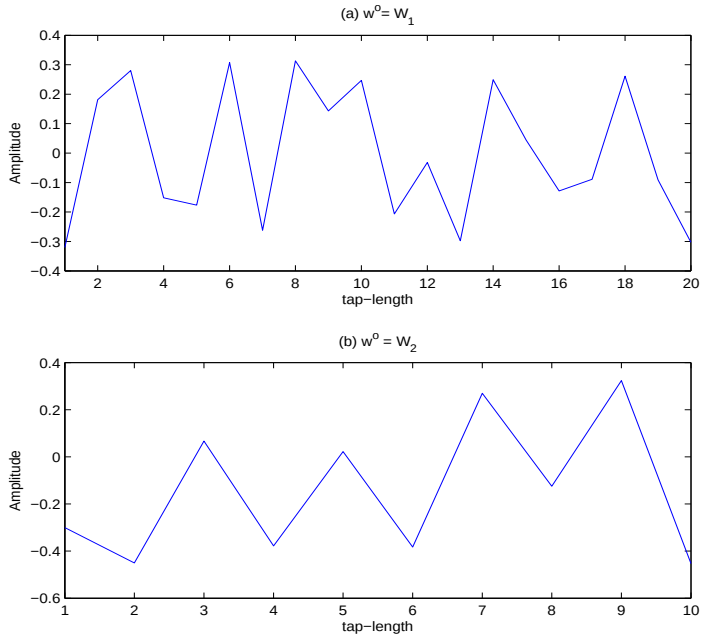

Figure 5.1. Two unknown systems: a) W_1; b) W_2.

obtained by averaging 200 independent runs.

There are two unknown systems W_1 and W_2 adopted to be tested. The coefficients of these filters are chosen from a zero-mean uniformly distributed random sequence and the tap-lengths are 20 and 10 respectively, shown in Figures 5.1(a) and 5.1(b). To satisfy **A3**, both variances of the final coefficients $W_1(M)^2$ and $W_2(M)^2$ are always bigger than 0.01. Initially, the unknown system is assumed to be W_1, a 20 tap-length filter; then after 2000 instants, it is replaced by W_2, which has a 10 tap-length; then after 4000 instants, W_1 is recalled to substitute W_2 to be the unknown filter. The choice of parameters follows the guidelines of the corresponding publications [17], [18]. For both algorithms, the parameter $\Delta = 6$ is selected and the initial tap-length $L(0)$ equals to the minimum tap-length $L\text{min} = \Delta + 1 = 7$. And the step-size β for the fractional tap-length update is set as unity. It is well

known that, for the case of uncorrelated Gaussian data, the resulting bound of the step-size is obtained as follows [65]:

$$0 < \mu < \frac{2}{3Tr(R_u)} \tag{5.3.30}$$

In this example, the step-size of the adaptive filter is therefore chosen as $\mu = 0.005$, which not only ensures the stability of LMS but satisfies the condition $w^o(M)^2 > J^{(M+\Delta-1)}_{M-1,\ excess} - J^{(M+\Delta-1)}_{M+\Delta-1,\ excess}$ as described in Subsection 5.3.1. The leakage factors of the FT algorithm are chosen as 0.1 and 0.0005 respectively. For the proposed algorithm, $\alpha_{\max}$ is chosen as 0.1 and $\delta = 0.01$ is used to control the adaptive leakage factor. And the smoothing factor of the error is chosen as $\rho = 0.99$, which results in the converged smoothed MSE of the novel algorithm having approximately the value of $0.005\sigma_v^2$, as in equation (5.3.19).

Figure 5.2(a) illustrates the evolution curves of the fractional tap-length adaptation. Figure 5.2(b) plots the learning curve of the adaptive leakage factor. According to equations (5.3.16) and (5.3.18), the converged leakage factor can be estimated and the value is approximately 0.0005. As expected, Figure 5.2(a) shows that the proposed scheme not only avoids "wandering" in high value areas, which the FT algorithm with $\alpha = 0.0005$ encounters during the period $[2001,\ 4000]$, but also obtains an improvement that its fractional tap-length better calculates the true tap-length as compared to the FT algorithm with $\alpha = 0.1$ for example during periods $[1000,\ 2000]$, $[2500,\ 4000]$ and $[5000,\ 6000]$.

In addition, it gives a good compromise for the fractional tap-length convergence rate in both the ITL estimation and the DTL estimation.

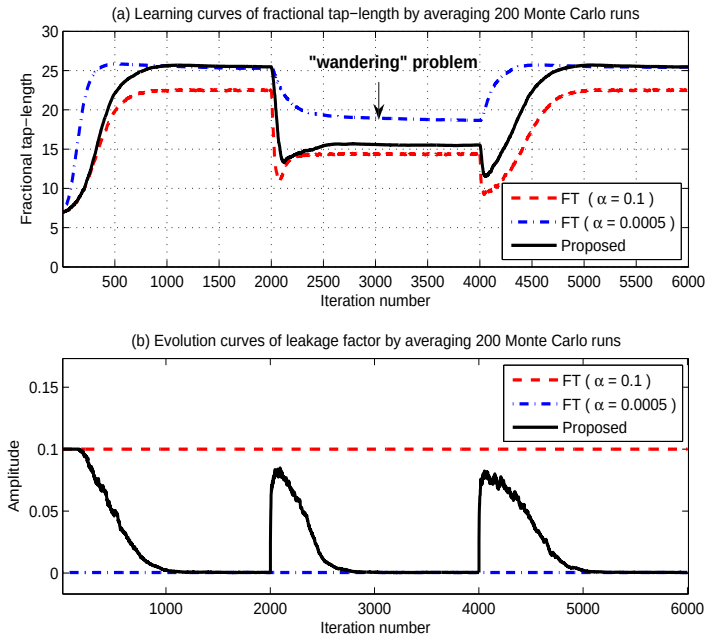

Figure 5.2. Learning curves of the FT algorithm and the proposed algorithm for Gaussian data by averaging 200 Monte Carlo runs in 30dB SNR: a) fractional tap-length, b) leakage factor.

The learning curve in Figure 5.2(b) shows the fluctuation of the variable leakage factor in the time varying scenario which confirms the tracking ability of the variable leakage factor. Therefore, it is clear that when the value of the leakage factor of the novel scheme for the true tap-length is located in the approximate range $(0.0005,\ 0.1)$, the proposed algorithm can be utilized to search for the true tap-length without the "wandering" problem.

5.4 A new VT algorithm with second and fourth order statistics

Several VTLMS algorithms have been proposed in recent years [13], [14], [15], [16], [54], [56], [66]; a summary of these works is given in [17]. All the above work on the VTLMS algorithms on the update of both the adaptive filter coefficients and the tap-length of the LMS algorithm are

based on second order statistics (SOS). It is well known that algorithms based on higher order statistics (HOS) potentially work more efficiently for sub-Gaussian noise environments since they utilize the information contained in higher order moments, which yields a better approximation of the actual distribution of the signal. A typical algorithm is the least mean fourth (LMF) algorithm, which has been proved to have a faster convergence rate as compared with the LMS algorithm [67] in sub-Gaussian noise environments. However, it suffers from a stability condition. If the LMF algorithm is initialized far from the optimal filter coefficients, it may be unstable [68].

Motivated by the FT algorithm, the next subsection will describe a new VTLMS algorithm, in which the update of the tap-length is controlled by fourth order statistics whilst the coefficient update retains as conventional LMS form. Such an approach utilizes the good properties of both SOS and HOS, i.e., good stability and quick convergence. As will be shown by simulations, the proposed approach has a faster convergence rate as compared with the FT variable tap-length LMS algorithm in sub-Gaussian noise environments.

5.4.1 Proposed algorithm

It is clear to see from (5.2.6) that in the FT algorithm, the tap-length of the adaptive filter is updated based on the information provided by SOS, i.e., squared value of instantaneous errors, and no HOS information is utilized. In the proposed algorithm, the update of the coefficients of the adaptive filters still utilizes the SOS, i.e., the LMS algorithm, to have a good stability property, but the update of the tap-length is driven by HOS to obtain a faster convergence rate. The

update of the tap-length of the proposed algorithm is as follows

$$l_f(i+1) = (l_f(i) - \alpha) - \beta \left[(e_{L(i)}^{(L(i))}(i))^4 - (e_{L(i)-\Delta}^{(L(i))}(i))^4 \right] \qquad (5.4.1)$$

It is clear to see that the update equation (5.4.1) is very similar to (5.2.6), and the fourth moments of the instantaneous errors are utilized. In order to speed up the convergence rate of the tap weight adaptation in the proposed algorithm, the step-size is made variable rather than fixed, according to the range of μ described in [14]:

$$\mu(i) = \mu' / \left[(L(i)+2)\sigma_u^2 \right] \qquad (5.4.2)$$

where μ' is a constant and σ_u^2 is the variance of the input. The coefficients of the adaptive filter are then updated according to the LMS algorithm by using $L(i)$ and $\mu(i)$. As analyzed in [57] the parameter β controls the adaptation process of the variable tap-length. Similar to that in [17] the tap-length $L(i+1)$ which will be used in the adaptation of the filter weights in the LMS algorithm is obtained from the fractional tap-length $l_f(i+1)$ according to (5.4.1), and the step-size is chosen according to (5.4.2), followed by the update of the adaptive filter coefficients according to the LMS algorithm.

By decomposing $\beta \left[(e_{L(i)}^{(L(i))}(i))^4 - (e_{L(i)-\Delta}^{(L(i))}(i))^4 \right]$ in expressing (5.4.1) into $\beta \left[(e_{L(i)}^{(L(i))}(i))^2 + (e_{L(i)-\Delta}^{(L(i))}(i))^2 \right] \left[(e_{L(i)}^{(L(i))}(i))^2 - (e_{L(i)-\Delta}^{(L(i))}(i))^2 \right]$, one can find that the proposed algorithm can be deemed as the FT algorithm with a variable parameter $\beta \left[(e_{L(i)}^{(L(i))}(i))^2 + (e_{L(i)-\Delta}^{(L(i))}(i))^2 \right]$. As compared with a fixed parameter β, it is large initially and small in the steady-state. Thus as compared with the FT algorithm, the proposed algorithm will have a smaller variance of the tap-length but a similar

convergence rate of the tap-length, which results in a quick convergence rate of the EMSE, but also reduces or avoids the under-modeling phenomenon. As will also be shown in the simulations in the next subsection, the proposed algorithm has a better performance as compared with the FT algorithm in sub-Gaussian noise environments.

5.4.2 Simulations

Two simulations are performed in this section to show the advantages of the proposed algorithm as compared with the FT algorithm in sub-Gaussian noise environments. The setup of the first simulation is as follows. The impulse response sequence of the unknown filter is a random sequence with zero mean and variance 0.01. The tap-length L_{opt} is set to 200. The input signal is a white Gaussian sequence with zero mean and unit variance. The noise signal is a white uniformly distributed sequence with zero mean and scaled to make the SNR 0dB. The parameter ν in (5.2.8) is set to 2. The step size μ' in (5.4.2) is set to 0.5. The leakage parameter α is set to 0.01, and Δ is set to 20.

To show the advantages of the proposed algorithm, the original FT algorithm is performed with different values $\beta = 1$ and $\beta = 5$ respectively. The parameter β for the proposed algorithm is set to 0.2.

In Figure 5.3, one run of the simulation is plotted according to the above set up. It is clear to see in Figure 1(a) that the tap-length of the proposed algorithm has a similar convergence rate as that of the FT algorithm with a parameter $\beta = 5$, but has a much smaller steady state variance. For the FT algorithm with a large parameter $\beta = 5$, the under-modeling phenomenon appears, which results in an increase of the EMSE, as can be seen in Figure 1(b). The proposed algorithm

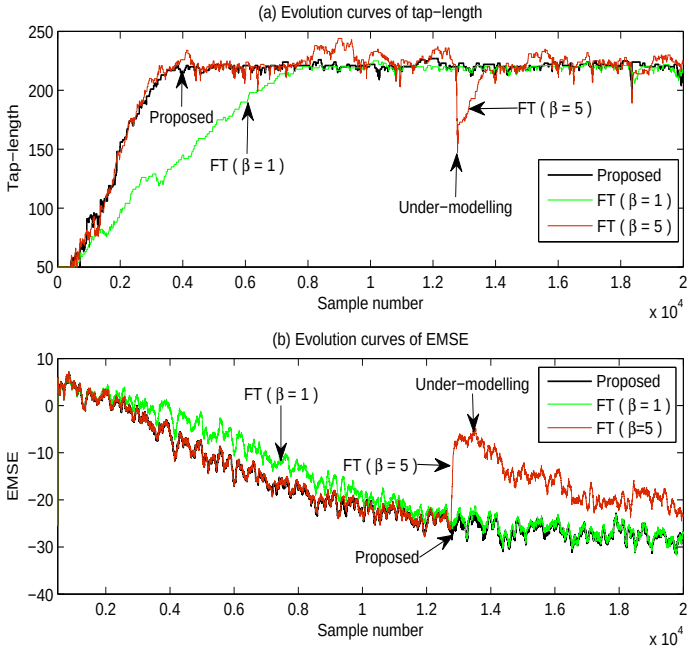

Figure 5.3. The evolution curves of the tap-length and EMSE for both the proposed algorithm and the FT algorithm with a uniformly distributed noise and SNR=0dB.

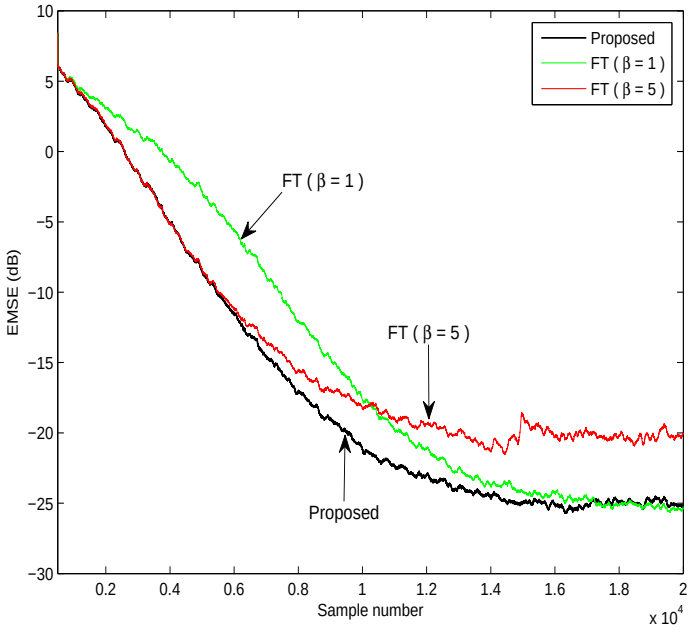

Figure 5.4. The evolution curves of the EMSE for both the proposed algorithm and the FT algorithm with a uniformly distributed noise and SNR=0dB, obtained by averaging 100 independent runs.

has both a quicker convergence of the tap-length and the EMSE as compared with that of the FT algorithm with $\beta = 1$.

Figure 5.4 shows the evolution curves of the EMSE of both algorithms by averaging the results of 100 independent runs. It is clear to see from this figure that the proposed algorithm has a similar convergence rate of the EMSE with that of the FT algorithm with $\beta = 5$, but approximately 8dB EMSE improvement. It has a similar steady-state EMSE with that of the FT algorithm with $\beta = 1$ but a quicker convergence. Based on the above simulation results it can be concluded that the proposed algorithm outperforms the FT algorithm in this uniform distributed noise environment.

In the second simulation, the set up is the same as that of the first simulation, but the noise is a binary sequence and scaled to make the SNR 0dB. Similar to that of the first simulation, Figure 5.5 shows the evolution curves of the tap-length and EMSE obtained from a single run of both the proposed algorithm and the FT algorithm. In Figure 5.6, the evolution curves of both algorithms are obtained by averaging the results of 100 independent runs. Again, it is clear to see from both Figure 5.5 and Figure 5.6 that the proposed algorithm outperforms the FT algorithm for the binary noise environment.

5.5 Conclusions

This chapter presented a novel adaptive leakage factor variable tap-length learning algorithm based on the FT algorithm together with an analysis of the converged difference between the segmented MSE of a filter formed from a number of initial coefficients of an adaptive filter, and the MSE of the full adaptive filter, which motivated the novel

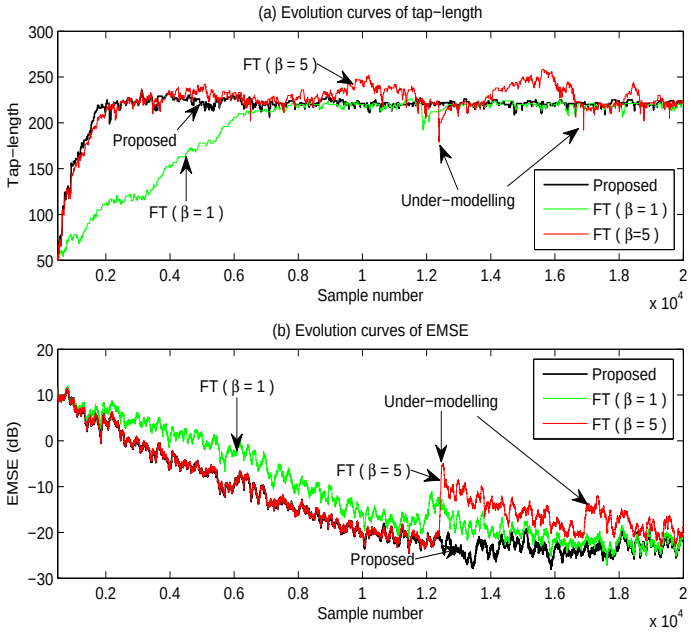

Figure 5.5. The evolution curves of the tap-length and EMSE for both the proposed algorithm and the FT algorithm with a binary noise sequence and SNR=0dB.

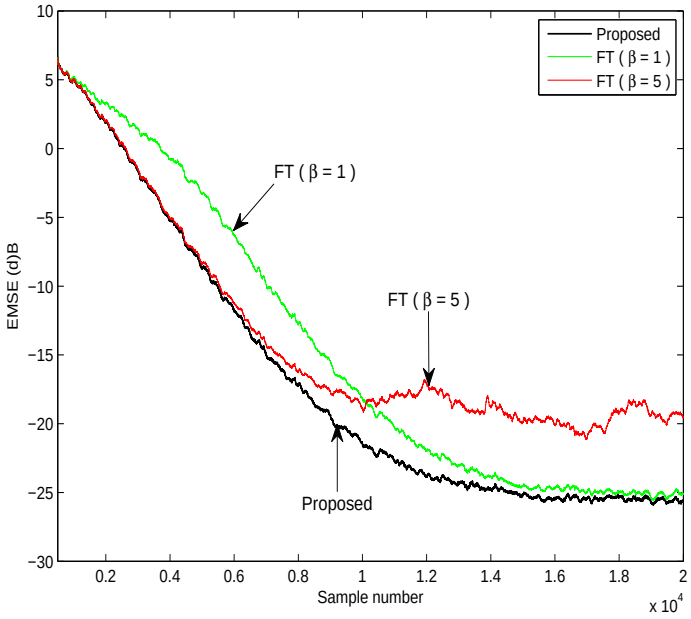

Figure 5.6. The evolution curves of the EMSE for both the proposed algorithm and the FT algorithm with a binary noise sequence and SNR=0dB, obtained by averaging 100 independent runs.

scheme. The simulation results have confirmed the advantages of the presented algorithm over the original FT algorithm in terms of the performance behavior of the fractional tap-length to ensure convergence to permit calculation of the true tap-length in both the ITL estimation and DTL estimation cases. Although this chapter focuses on the uncorrelated white-noise input, further improvements can be expected by extending it to general coloured input. In addition, a new VT adaptive algorithm based on the FT method is also proposed in this chapter. In this algorithm, the update of the tap-length is controlled by fourth order statistics. As have been shown by simulation results the proposed algorithm has a better performance as compared with the original FT algorithm in sub-Gaussian noise environments, and can be potentially utilized in many applications. The work of this chapter provided a deeper understanding of VTLMS algorithms based on the FT method, which motivates the further study of tap-length adjustment within distributed adaptive estimation.

Chapter 6

VARIABLE LENGTH FILTERING WITHIN INCREMENTAL LEARNING ALGORITHMS FOR DISTRIBUTED ADAPTIVE ESTIMATION

6.1 Introduction

Distributed solutions, only exploiting local data exchanges and communications between immediate neighboring nodes, have been proposed for adaptive networks [1] and [27], with the purpose of reducing processing and communications requirements as compared to centralized solutions. The applications of such distributed adaptive networks range from sensor networks to environmental monitoring and factory instrumentation [8] and [9]. However, in many applications of such distributed adaptive estimation 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.

The concept of tap-length adaptation is therefore introduced in the design of the structure of the adaptive filter. With low-complexity and robustness, the FT learning algorithm based on squared instantaneous errors has been proposed in [17] to obtain improved convergence performance, as compared with other methods. The steady-state performance analysis of the FT algorithm is provided in [17] and [18], which also provides a guideline for parameter selection in the FT algorithm.

Motivated by both the ideas of distributed adaptive estimation and variable tap-length, this chapter 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 [1], [21]. In particular, theoretical expressions are derived for the MSD, EMSE and MSE of each node within the network. Simulation studies are presented to confirm the convergence properties of the scheme and to verify the theoretical results.

6.2 Estimation problem and formulation

Consider an $N-$node network, where data are observed and collected to seek an unknown system vector w^o, whose tap weights and tap-length are desirable to be estimated. Note that the adaptation rules can be decoupled into the tap weights and tap-length adjustments respectively, which means that the selection of one does not explicitly 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 the unknown coefficients of w, which is a $1\times L$ vector, the linear minimum mean-square estimation problem is formulated as:

$$\min_{w}\; J_L^{(L)}(w) \quad \text{and} \quad J_L^{(L)}(w) = E\|\mathbf{d}-\mathbf{U}w\|^2 \tag{6.2.1}$$

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

$$\mathbf{d} \triangleq \text{col}\{\mathbf{d}_1,\, \mathbf{d}_2,\, \ldots,\, \mathbf{d}_N\}, \quad (N\times 1) \tag{6.2.2}$$

$$\mathbf{U} \triangleq \text{col}\{\mathbf{u}_1,\, \mathbf{u}_2,\, \ldots,\, \mathbf{u}_N\}, \quad (N\times L). \tag{6.2.3}$$

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. The incremental steepest-descent solution [1] is introduced to estimate the optimal solution w^o by iterating $\psi_0(i)$ through an incremental network in the following manner:

$$\psi_k(i) = \psi_{k-1}(i) + \mu_k u_{k,i}^*(d_k(i) - u_{k,i}w_{i-1}) \quad k = 1,\ldots N \tag{6.2.4}$$

which begins with the setup of the initial condition $\psi_0(i) = w_{i-1}$ at node 1.

With the assumption of the tap-length L, the segmented cost function is defined

$$J_M^{(L)}(w) \triangleq E\|\mathbf{d}-\mathbf{U}_M w_M\|^2 \tag{6.2.5}$$

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

$$w_M \triangleq \text{col}\{w(1), \ldots, w(M)\}, \qquad (1 \times M) \tag{6.2.6}$$

$$\mathbf{U}_M \triangleq \text{col}\{\mathbf{u}_1(1:M), \ldots, \mathbf{u}_N(1:M)\}, \ (N \times M). \tag{6.2.7}$$

Thus, the minimum difference of mean-square errors estimation problem is posed to seek the optimal tap-length L^o for the adaptive filters

$$\min\left\{L \Big| J^{(L)}_{L-\triangle}(w) - J^{(L)}_{L}(w) \leq \varepsilon\right\} \tag{6.2.8}$$

where ε, predetermined by system requirements, is a small positive value and $\triangle$ is an integer value to avoid the suboptimum tap-lengths.

Motivated by distributed adaptive estimation, the segment cost function and full cost function can be decomposed as

$$J^{(L)}_{L-\triangle}(w) = \sum_{k=1}^{N} J^{(L)}_{k,L-\triangle}(w) \tag{6.2.9}$$

$$J^{(L)}_{L}(w) = \sum_{k=1}^{N} J^{(L)}_{k,L}(w) \tag{6.2.10}$$

where

$$J_{k,L-\triangle}(w) = E|\mathbf{d}_k - \mathbf{u}_k(1:L-\triangle)w(1:L-\triangle)|^2 \tag{6.2.11}$$

$$J_{k,L}(w) = E|\mathbf{d}_k - \mathbf{u}_k w|^2. \tag{6.2.12}$$

Therefore, (6.2.8) for distributed adaptive estimation is formulated as

$$\min\left\{L \Big| \sum_{k=1}^{N} J^{(L)}_{k,L-\triangle}(w) - \sum_{k=1}^{N} J^{(L)}_{k,L}(w) \leq \varepsilon\right\} \tag{6.2.13}$$

Such results lead to the fractional tap-length adaptation function (5.2.7)

being rewritten as

$$L_f(i+1) = (L_f(i) - \alpha) + \beta \cdot \sum_{k=1}^{N} \left(e^{(L)}_{k,L(i)-\triangle}(w_{i-1}) \right)^2 - \left(e^{(L)}_{k,L(i)}(w_{i-1}) \right)^2 \tag{6.2.14}$$

where

$$e^{(L_k(i))}_{k,L_k(i)-\Delta}(w_{i-1}) = d_k(i) - u_{k,i}(1 : L_k(i) - \Delta) w_{i-1}(1 : L_k(i) - \Delta) \tag{6.2.15}$$

$$e^{(L_k(i))}_{k,L_k(i)}(w_{i-1}) = d_k(i) - u_{k,i} w_{i-1} \tag{6.2.16}$$

With proper choice of α and β, such a distributed algorithm can obtain $L(i) \to L^o$ 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. The parameters 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, it starts with the initial condition $\ell_{0,f}(i) = L_f(i)$ at node 1 ($L_f(i)$ is the current global estimation). 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 (5.2.8). Such implementation of a centralized solution for tap-length adaptation is described as follows:

Table 6.1. Pseudo-code implementation of centralized solution.

For each time instant $i \geq 0$ repeat:

$\ell_{0,f}(i) = L_f(i)$

For k=1,... N

$\ell_{k,f}(i) = \ell_{k-1,f}(i) - \alpha_k + \beta_k \cdot \gamma_k(w_{i-1})$

end

$L_f(i+1) = \ell_{N,f}(i)$

$$L(i+1) = \begin{cases} \mathrm{Rd}\left[L_f(i+1)\right] if |L(i) - L_f(i+1)| \geq \nu \\ L(i) \qquad\qquad \text{otherwise} \end{cases}$$

where $\gamma_k(w_{i-1}) = \left(e^{(L(i))}_{k,L(i)-\triangle}(w_{i-1})\right)^2 - \left(e^{(L(i))}_{k,L(i)}(w_{i-1})\right)^2$

where it always holds that $\ell_{k,f}(i) \geq L_{\min}$, 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 (6.2.4) is presented in [1] as

$$\psi_k(i) = \psi_{k-1}(i) + \mu_k u^*_{k,i}(d_k(i) - u_{k,i}\psi_{k-1}(i)) \quad k = 1, \ldots N \qquad (6.2.17)$$

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

Table 6.2. Pseudo-code implementation of distributed solution.

For each time instant $i \geq 0$ repeat:

$\ell_{0,f}(i) = L_f(i)$

For k=1,... N

$$\ell_{k,f}(i) = \ell_{k-1,f}(i) - \alpha_k + \beta_k \cdot \gamma_k(\psi_{k-1}^{(i)})$$

$$L_{k+1}(i) = \begin{cases} \mathrm{Rd}\left[\ell_k(i)\right] if |L_k(i) - \ell_{k,f}(i)| \geq \nu_k \\ L_k(i) \qquad \text{otherwise} \end{cases}$$

end

$L_f(i+1) = \ell_{N,f}(i)$

where $\gamma_k(\psi_{k-1}^{(i)}) = \left(e_{k,L_k(i)-\triangle}^{(L_k(i))}(\psi_{k-1}^{(i)})\right)^2 - \left(e_{k,L_k(i)}^{(L_k(i))}(\psi_{k-1}^{(i)})\right)^2$

where

$$e_{k,L_k(i)-\Delta}^{(L_k(i))}(\psi_{k-1}^{(i)}) = d_k(i) - u_{k,i}(1:L_k(i)-\Delta)\psi_{k-1}^{(i)}(1:L_k(i)-\Delta) \quad (6.2.18)$$

$$e_{k,L_k(i)}^{(L_k(i))}(\psi_{k-1}^{(i)}) = d_k(i) - u_{k,i}\psi_{k-1}^{(i)} \quad (6.2.19)$$

Figure 6.1 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, the selection of the parameters is set as: $\nu_k = \nu = 1$, $\alpha_k = 0.03$, $\beta_k = 1$, $\triangle = 4$ and $L_{\min} = L_f(0) = 6$; such selection follows the rules in [17], [18]. Note that, in theory, the upper bound for the value of fractional tap-length is not necessary to be chosen. However, since the length estimation uses instantaneous

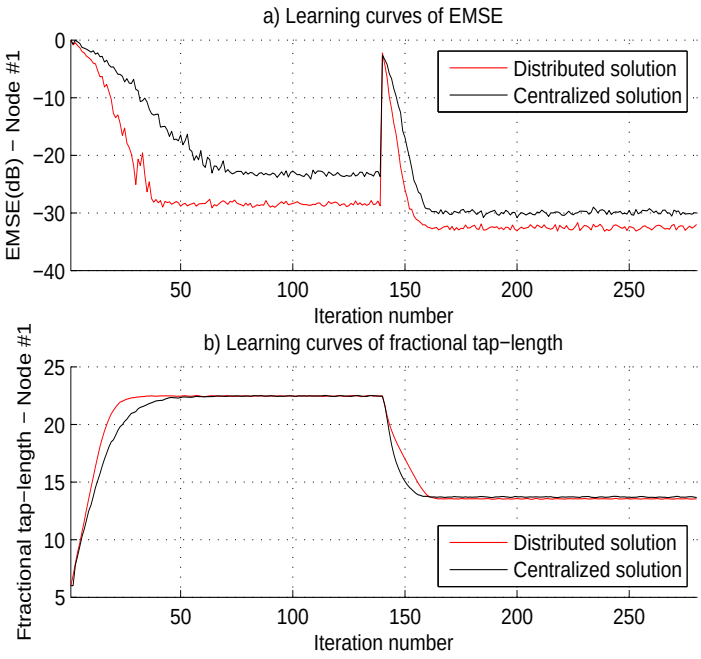

Figure 6.1. Evolution curves for the distributed solution and the centralized solution at node 1: a) EMSE performance b) Fractional tap-length performance.

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 a situation. The performance of the distributed solution for an incremental network is studied in the subsequent section.

6.3 Performance analysis

In this section, weighted spatial-temporal energy conservation arguments [1] are used to evaluate the steady-state performance of the distributed algorithm. The following assumptions are utilized

A1) In order to simplify the analysis, the estimated tap-length is as-

sumed to be fixed in the steady-state.

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

$$\mathbf{d}_k(i) = \mathbf{u}_{k,i} w^o + \mathbf{v}_k(i) \tag{6.3.1}$$

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 [1] are introduced to perform the evaluation:

$$\tilde{\boldsymbol{\psi}}_{k-1}^{(i)} \triangleq w^o - \boldsymbol{\psi}_{k-1}^{(i)}, \quad \tilde{\boldsymbol{\psi}}_{k}^{(i)} \triangleq w^o - \boldsymbol{\psi}_{k}^{(i)} \tag{6.3.2}$$

$$\mathbf{e}_{a,k}(i) \triangleq \mathbf{u}_{k,i} \tilde{\boldsymbol{\psi}}_{k-1}^{(i)}, \quad \mathbf{e}_{p,k}(i) \triangleq \mathbf{u}_{k,i} \tilde{\boldsymbol{\psi}}_{k}^{(i)} \tag{6.3.3}$$

Introduce further the weighted error signals:

$$\boxed{\mathbf{e}_{p,k}^{\Sigma_k}(i) \triangleq \mathbf{u}_{k,i} \Sigma_k \tilde{\boldsymbol{\psi}}_{k}^{(i)}, \quad \mathbf{e}_{a,k}^{\Sigma_k}(i) \triangleq \mathbf{u}_{k,i} \Sigma_k \tilde{\boldsymbol{\psi}}_{k-1}^{(i)}} \tag{6.3.4}$$

where Σ_k is an arbitrary Hermitian positive-definite weighting matrix 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

$$\eta_k \triangleq E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(\infty)}\|^2 \qquad \text{(MSD)} \tag{6.3.5}$$

$$\zeta_k \triangleq E\|\tilde{\boldsymbol{\psi}}_{k-1}^{(\infty)}\|_{R_{u,k}}^2 \quad \text{(EMSE)} \tag{6.3.6}$$

$$\xi_k \triangleq \zeta_k + \sigma_{v,k}^2 \qquad \text{(MSE)}. \tag{6.3.7}$$

where the weighted norm notation $\|x\|_{\Sigma}^2 = x^*\Sigma x$ is used with a vector x and Hermitian positive-definite matrix $\Sigma > 0$.

As presented in [1], the spatial-temporal energy conservation relation between two successive nodes is given by

$$\boxed{E\|\tilde{\boldsymbol{\psi}}_k\|_{\Sigma_k}^2 = E\|\tilde{\boldsymbol{\psi}}_{k-1}\|_{\Sigma_k'}^2 + \mu_k^2\sigma_{v,k}^2 E\|\mathbf{u}_k\|_{\Sigma_k}^2} \tag{6.3.8}$$

where Σ_k' is given by

$$\boxed{\Sigma_k' = \Sigma_k - \mu_k E(\mathbf{u}_k^*\mathbf{u}_k\Sigma_k + \Sigma_k\mathbf{u}_k^*\mathbf{u}_k) + \mu_k^2 E(\|\mathbf{u}_k\|_{\Sigma_k}^2\mathbf{u}_k^*\mathbf{u}_k)}. \tag{6.3.9}$$

Assume that 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, it is easy to derive $E\|\tilde{\boldsymbol{\psi}}_{k-1}\|_{\Sigma_k}^2 = E\|\bar{\boldsymbol{\psi}}_{k-1}\|_{\bar{\Sigma}_k}^2$ 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 [6], the transformed expressions from (6.3.8) and (6.3.9) are obtained

$$\boxed{E\|\bar{\boldsymbol{\psi}}_k\|_{\bar{\Sigma}_k}^2 = E\|\bar{\boldsymbol{\psi}}_{k-1}\|_{\bar{\Sigma}_k'}^2 + \mu_k^2\sigma_{v,k}^2 Tr(\Lambda_k\bar{\Sigma}_k)} \tag{6.3.10}$$

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

$$\boxed{\bar{\Sigma}'_k = \bar{\Sigma}_k - \mu_k X_k + \mu_k^2 Y_k} \tag{6.3.11}$$

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.

A different method for the analysis as compared with that in [1] is therefore used for originality, the $M^2 \times 1$ vectors as in [6] are therefore introduced

$$\delta_k = \text{vec}\{\bar{\Sigma}_k\} \ , \delta'_k = \text{vec}\{\bar{\Sigma}'_k\} \text{ and } \lambda_k = \text{vec}\{\Lambda_k\} \tag{6.3.12}$$

to develop expressions (6.3.10) and (6.3.11). One can exploit the following useful property for the vec$\{\cdot\}$ notation when working with Kronecker products as in Chapter 2 and 3: for any matrices $\{P, \Sigma, Q\}$ of compatible dimensions, it holds that

$$\text{vec}\{P\Sigma Q\} = (Q^{\mathrm{T}} \otimes P)\text{vec}\{\Sigma\}. \tag{6.3.13}$$

The choice of Σ_k can make both $\bar{\Sigma}_k$ and $\bar{\Sigma}'_k$ become diagonal in (6.3.11). Applying the vec$\{\cdot\}$ operation to both sides of (6.3.11), a linear relation between the corresponding vectors $\{\delta'_k, \delta_k\}$ is obtained, namely,

$$\delta'_k = F_k \delta_k \tag{6.3.14}$$

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

$$F_k = I - 2\mu_k (I \otimes \Lambda_k) + \mu_k^2 (\tau(\Lambda_k \otimes \Lambda_k) + \lambda_k \lambda_k^T). \tag{6.3.15}$$

Therefore, expression (6.3.10) becomes

$$E\|\bar{\boldsymbol{\psi}}_k^{(i)}\|^2_{\mathrm{vec}\{\delta_k\}} = E\|\bar{\boldsymbol{\psi}}_{k-1}^{(i)}\|^2_{\mathrm{vec}\{F_k\delta_k\}} + \mu_k^2\sigma_{v,k}^2(\lambda_k^T\delta_k) \tag{6.3.16}$$

where the time index i is recalled for clarity. For simplicity of notation, the vec$\{\cdot\}$ notation is dropped from the subscripts in (6.3.16):

$$\boxed{E\|\bar{\boldsymbol{\psi}}_k^{(i)}\|^2_{\delta_k} = E\|\bar{\boldsymbol{\psi}}_{k-1}^{(i)}\|^2_{F_k\delta_k} + \mu_k^2\sigma_{v,k}^2(\lambda_k^T\delta_k)}\,. \tag{6.3.17}$$

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

$$E\|\boldsymbol{\rho}_k\|^2_{\delta_k} = E\|\boldsymbol{\rho}_{k-1}\|^2_{F_k\delta_k} + \mu_k^2\sigma_{v,k}^2(\lambda^T\delta_k). \tag{6.3.18}$$

By iterating (6.3.18) over one cycle, N coupled equations are obtained:

$$\begin{aligned}
E\|\boldsymbol{\rho}_1\|^2_{\delta_1} &= E\|\boldsymbol{\rho}_N\|^2_{\mathrm{F}_1\delta_1} + g_1\delta_1 \\
E\|\boldsymbol{\rho}_2\|^2_{\delta_2} &= E\|\boldsymbol{\rho}_1\|^2_{\mathrm{F}_2\delta_2} + g_2\delta_2 \\
&\vdots \\
E\|\boldsymbol{\rho}_{k-1}\|^2_{\delta_{k-1}} &= E\|\boldsymbol{\rho}_{k-2}\|^2_{\mathrm{F}_{k-1}\delta_{k-1}} + g_{k-1}\delta_{k-1} \qquad (6.3.19)\\
E\|\boldsymbol{\rho}_k\|^2_{\delta_k} &= E\|\boldsymbol{\rho}_{k-1}\|^2_{\mathrm{F}_k\delta_k} + g_k\delta_k \qquad (6.3.20)\\
&\vdots \\
E\|\boldsymbol{\rho}_N\|^2_{\delta_N} &= E\|\boldsymbol{\rho}_{N-1}\|^2_{\mathrm{F}_N\delta_N} + g_N\delta_N
\end{aligned}$$

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 (6.3.19) and (6.3.20), then iterate this

procedure across the cycle to obtain

$$\begin{aligned} E\|\boldsymbol{\rho}_{k-1}\|^2_{\delta_{k-1}} = & E\|\boldsymbol{\rho}_{k-1}\|^2_{\mathrm{F}_k\cdots\mathrm{F}_N\mathrm{F}_1\cdots\mathrm{F}_{k-1}\delta_{k-1}} \\ & + g_k\mathrm{F}_{k+1}\cdots\mathrm{F}_N\mathrm{F}_1\cdots\mathrm{F}_{k-1}\delta_{k-1} \\ & + g_{k+1}\mathrm{F}_{k+2}\cdots\mathrm{F}_N\mathrm{F}_1\cdots\mathrm{F}_{k-1}\delta_{k-1} \\ & \cdots + g_{k-2}\mathrm{F}_{k-1}\delta_{k-1} + g_{k-1}\delta_{k-1}. \end{aligned} \tag{6.3.21}$$

Let

$$\Pi_{k-1,l}=\mathrm{F}_{k+l-1}\cdots\mathrm{F}_N\mathrm{F}_1\cdots\mathrm{F}_{k-1},\ \ l=1,2,\ldots,N \tag{6.3.22}$$

$$a_{k-1}=g_k\Pi_{k-1,2}+\cdots+g_{k-2}\Pi_{k-1,N}+g_{k-1} \tag{6.3.23}$$

then

$$\boxed{E\|\boldsymbol{\rho}_{k-1}\|^2_{(I-\Pi_{k-1,1})\delta_{k-1}} = a_{k-1}\delta_{k-1}}. \tag{6.3.24}$$

Since the weight vector δ_{k-1} in (6.3.24) is selected arbitrarily due to $\delta_{k-1} = \mathrm{vec}\Sigma_{k-1}$, 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:

$$\eta_k = E\|\boldsymbol{\rho}_{k-1}\|^2_q = a_{k-1}(I-\Pi_{k-1,1})^{-1}q \quad \text{(MSD)} \tag{6.3.25}$$

$$\zeta_k = E\|\boldsymbol{\rho}_{k-1}\|^2_{\lambda_k} = a_{k-1}(I-\Pi_{k-1,1})^{-1}\lambda_k \quad \text{(EMSE)} \tag{6.3.26}$$

$$\xi_k = \zeta_k + \sigma^2_{v,k} \quad \text{(MSE)} \tag{6.3.27}$$

where $q = \mathrm{vec}\{I\}$ and $\lambda_k = \mathrm{vec}\{\Lambda_k\}$. In the subsequent section, simulation study verifies the theoretical expressions (6.3.25)-(6.3.27).

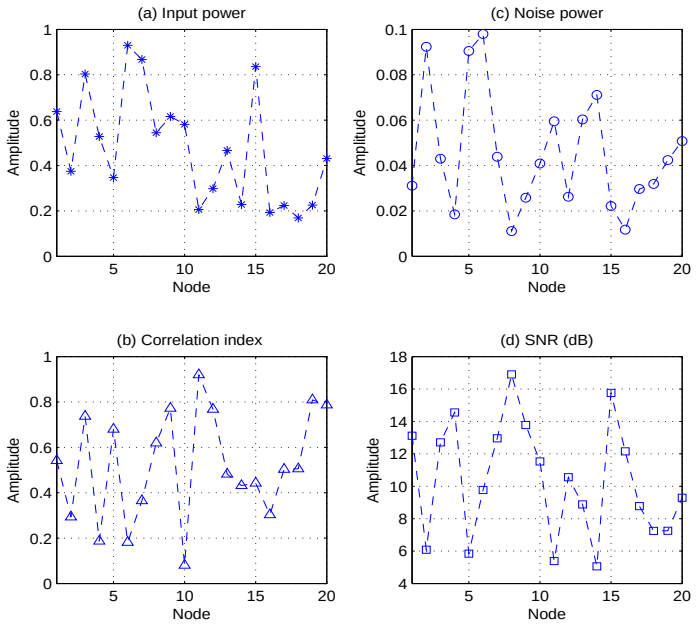

Figure 6.2. Node profile throughout the network: a) Input power; b) Noise power; c) Correlation index; d) SNR.

6.4 Simulations

A system identification model is used in this section in order to demonstrate the performance of the proposed algorithm, whose theoretical performance and computer results are compared in the simulations. Although the analysis is developed based on the independence assumptions, regressors with shift structure in all the simulations are used in order to approach real-time scenarios. All simulated results are averaged by using 100 independent Monte Carlo runs to generate the performance curves. The steady-state curves (MSD, EMSE and MSE) are obtained by averaging the last 5000 instantaneous samples of $50,000$ iterations. Consider a network with 20 nodes to seek an unknown filter with $M = 10$ taps, whose coefficients are selected from a uniform distributed sequence. A correlated Gaussian signal is used to generate

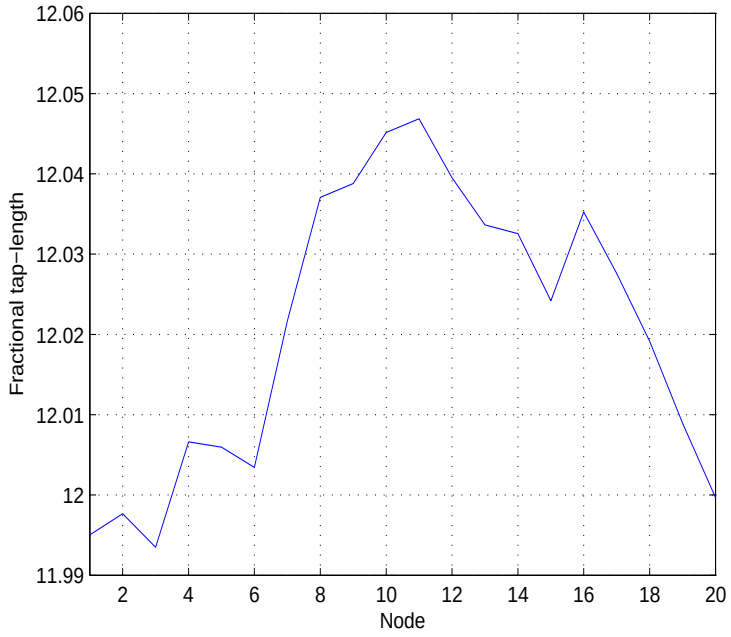

Figure 6.3. Fractional tap-length versus node k - $\mu_k = 0.02$.

the inputs at each node k which satisfies the recursion

$$u_k(i) = a_k u_k(i-1) + b_k \cdot c_k(i). \tag{6.4.1}$$

Expression (6.4.1) 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 an $M \times M$ 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 Figure 6.2. For the proposed algorithm, the selection of parameters is $\nu_k = 1$, $\alpha_k = 0.05$, $\beta_k = 1$, $\triangle = 3$, and $L_{\min} = L_f(0) = 4$. In addition, the initial global estimate of the unknown system is assumed as an $L_{\min} \times 1$ column vector with zero

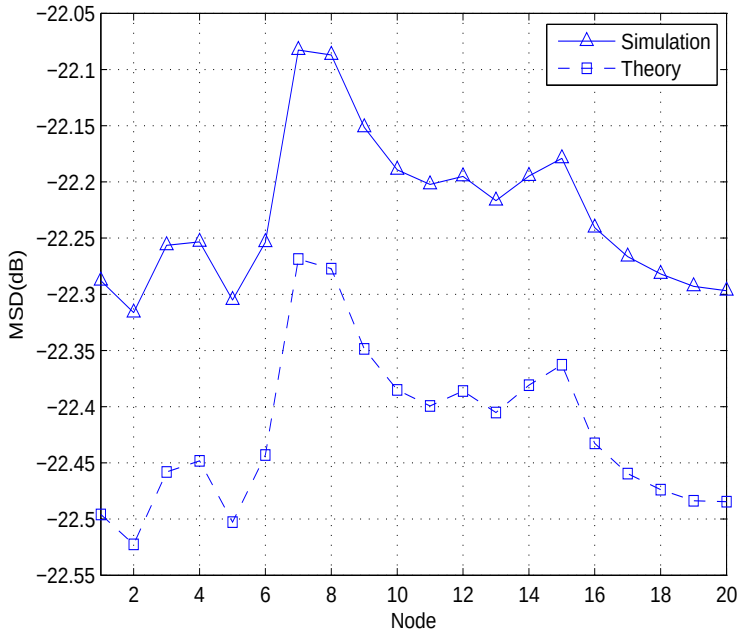

Figure 6.4. Steady-state MSD versus node k - $\mu_k = 0.02$.

elements. It is expected that the use of large step-sizes usually does not satisfy the simplifying assumptions adopted in the analysis, which will lead to large deviations between theory and simulation. Therefore, the step-size is chosen as a small value in all simulations for comparison.

In the first example, the step-size of the proposed algorithm is selected as $\mu_k = 0.02$, which leads to a good quality for the theoretical model. Figure 6.3 illustrates the simulated results of the converged fractional tap-length throughout the network, with a deviation of approximate ± 0.26 at 12.02. Therefore, as it is expected, in the steady-state the proposed algorithm can obtain a good estimate of the tap-length. One can see that the steady-state results, both in theory and in experiment, are plotted for comparison in Figures 6.4-6.7, which show that they have a good match. Figures 6.4-6.6 describe the steady-state performances of interest, namely, MSD, EMSE and MSE, as a function

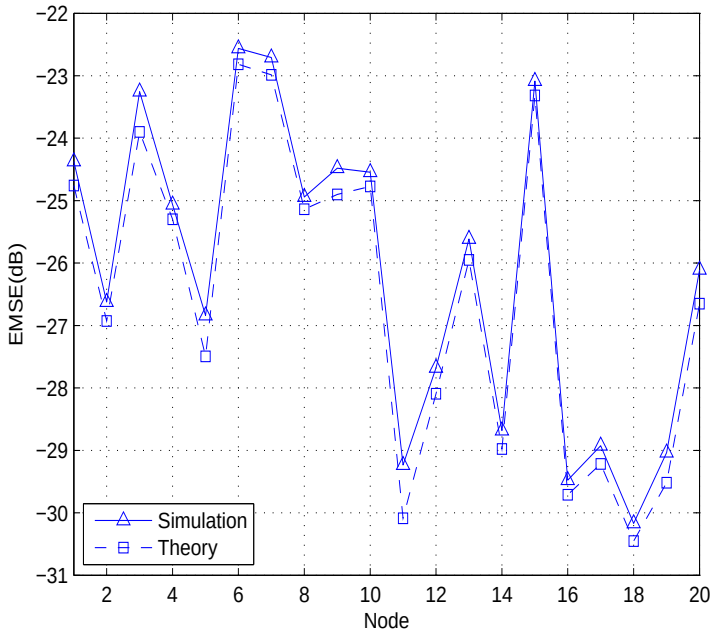

Figure 6.5. Steady-state EMSE versus node k - $\mu_k = 0.02$.

of the node k using a particular choice of the step size $\mu_k = 0.02$. The MSD shown in Figure 6.3 is roughly flat over the network, with a deviation about 0.2dB between theory and simulation. Figure 6.4 shows that the EMSE is more sensitive to the node profiles. The selection of parameter allows the proposed algorithm to have a good estimate for w^o in the steady-state, which means $e_{k,L_k(\infty)}(\infty)$ is close to the background noise. In Figure 6.4, one can therefore see that the MSE curve is similar to the noise power. It is clearly shown in these curves how node profiles affect the filters in the network, which provides a good guideline for the setup of the step-size μ_k at certain nodes in the network. Therefore, as a result of the proper selection of the step-size at each node, a performance equalization in the EMSE is expected to be achieved throughout the network. In the second example, the steady-state MSE is depicted as a function of the step-size in the range $[0.0008, 0.04]$ for a particular

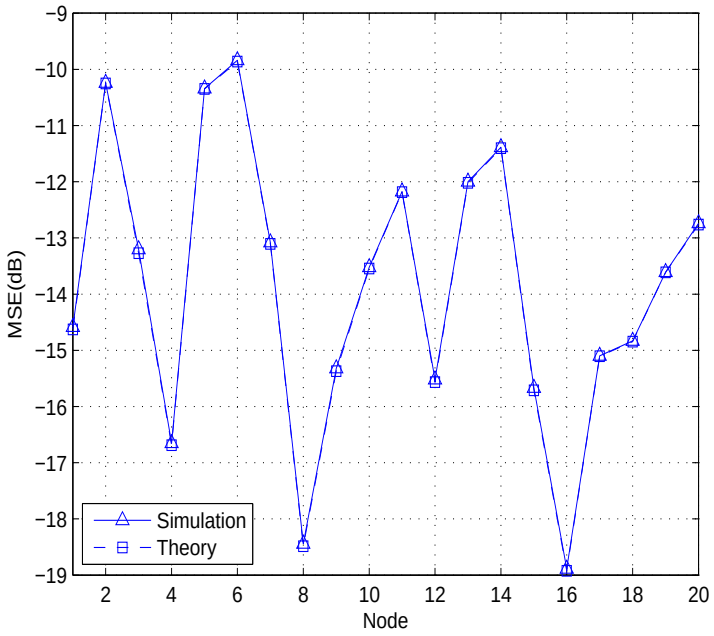

Figure 6.6. Steady-state MSE versus node k - $\mu_k = 0.02$.

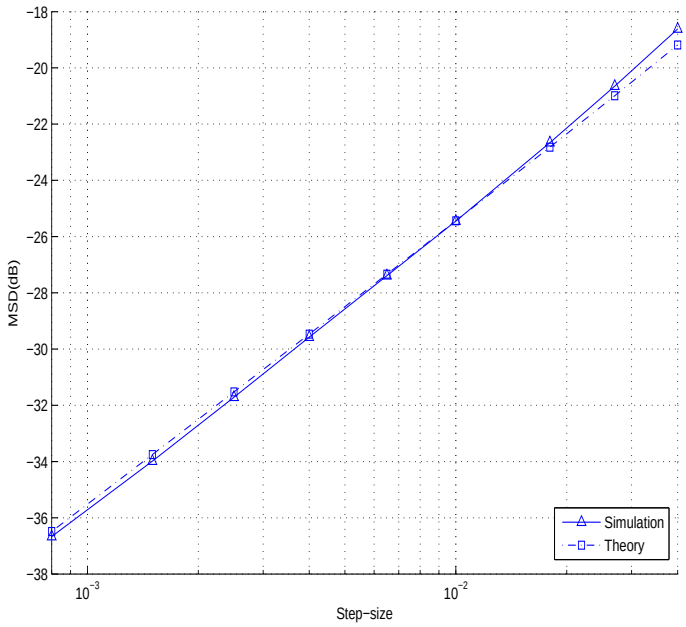

Figure 6.7. Steady-state MSD versus step-size at node 8.

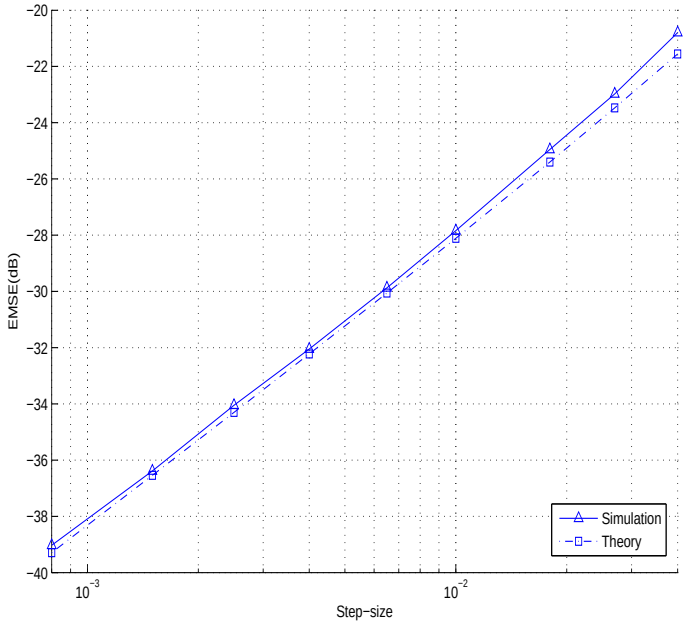

Figure 6.8. Steady-state EMSE versus step-size at node 8.

node $k = 8$ in Figure 6.9, which shows a good match between simulated results and theoretical results. As it is expected, the decrease of step-size leads to the improvement of the steady-state performance and reduces the deviation between theory and simulation. This is because small step-sizes strongly support the simplifying assumptions used in the analysis. Similar results for MSD and EMSE can be obtained and are therefore shown in Figures 6.7-6.8.

6.5 Conclusions

Several research studies appear in the literature to develop distributed estimation algorithms based on an incremental adaptive network. However, the adaptive filters involved in such algorithms are assumed to have fixed structures, namely, the tap-length is known due to the sys-

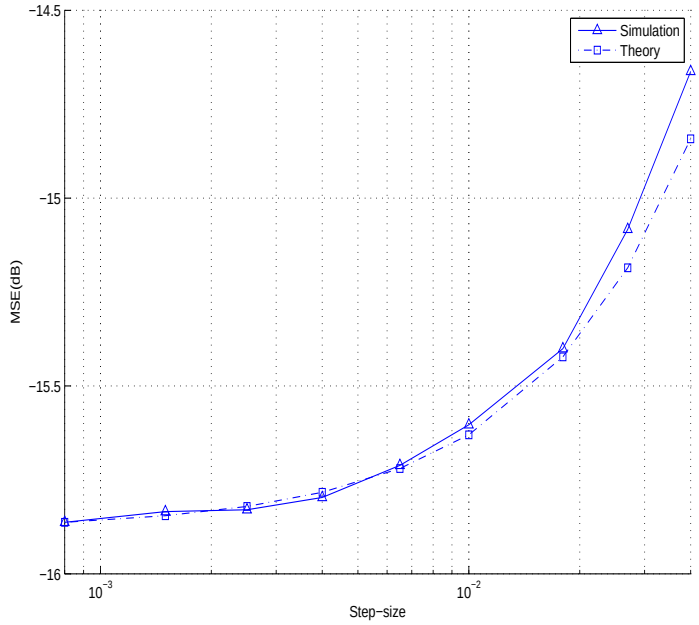

Figure 6.9. Steady-state MSE versus step-size at node 8.

tem requirement. In a more general vain, variable tap-length filters are more flexible in terms of an unknown vector estimation, where both tap weights and tap-length are unknown or variable. In this chapter, a new algorithm therefore has been proposed for structure adaptation of adaptive filters for an incremental distributed estimation. Under the assumptions **A1**, **A2** and **A3**, weighted spatial-temporal energy conservation arguments 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.

Chapter 7

CONCLUSION

This final chapter presents a summary of the work and contributions in the previous chapters of this thesis. In addition, overall conclusions of this study and recommendations for future research are also provided.

7.1 Summary of the thesis

This thesis concerns distributed adaptive estimation, which is posed to resolve the problem of using adaptive filters scattered in a physical area to estimate some parameters of interests. Compared to traditional methods, such as centralized and consensus schemes, distributed adaptive strategies have lower requirements of communication and complexity, and could therefore find a wide field of applications. Different adaptive methods employed at node filters, accompanied with different cooperation strategies, lead to various distributed adaptive schemes with various complexities. Thus, new approaches are required to improve the distributed adaptive estimation within different topological networks.

In this thesis, a background introduction to adaptive techniques, i.e. the least mean square algorithm (LMS), the affine projection algorithm (APA), the recursive least squares scheme (RLS) and the fractional tap-length algorithm (FT), was firstly presented. A brief review was

also given to describe the incremental adaptive schemes based on LMS and RLS methods in distributed networks. By utilizing the space-time weighted energy conservation arguments, a useful extension of the analysis for distributed LMS (dLMS) was developed to obtain the closed-form equations of the mean-square quantities in the non-Gaussian wide sense stationary data case.

Then, a new distributed APA algorithm (dAPA) was proposed to be adopted in incremental adaptive networks. The motivation of the proposed algorithm was to improve the poor convergence performance of the LMS-based distributed algorithm due to coloured input signals. In addition, it was shown to achieve a reasonable performance for certain tap-lengths with lower computational cost as compared with the corresponding RLS-based scheme. A key contribution of this work was to provide the analysis of its mean-square performance. One should note that studying the mean-square performance of the APA algorithm is demanding even for standard filters (without networks); when networking is added, the complexity is compounded due to the spatial and temporal interconnectedness of the data. For an incremental method, a Hamiltonian cycle must be established at every time in the network, which limits applications of dAPA. As a consequence, a new diffusion APA adaptive algorithm without topological constraints was developed and its performance analysis was also provided.

It is clear to see in some applications, the optimal tap-length for the adaptive filter is unknown. In order to obtain a good estimate of the steady-state adaptive filter tap-length, the variable tap-length LMS (VTLMS) algorithm is required. New research results have been achieved for the VTLMS algorithms, i.e., a new VTLMS algorithm

based on the fractional tap-length (FT) scheme and a new VTLMS algorithm which exploited both high order statistics (HOS) and second order statistics (SOS). All the analysis and proposed algorithms were verified and supported by simulations. Although these schemes can not be adopted directly in the field of distributed adaptive estimation due to the nonstationarity of the statistical spatial-temporal data found in networks, these research results provide a deeper understanding of the VTLMS algorithms, and may be potentially used in other applications.

Within this thesis it was the first time that the idea of variable tap-length was introduced into the distributed adaptive estimation research area, in particular for incremental adaptive networks. A novel variable tap-length dLMS algorithm based on the FT method was proposed to search for a good choice of the tap-length of the adaptive filters within the network. A steady-sate performance analysis was also presented to achieve the closed-form equations of mean-square quantities, which provide good performance measures. As shown in simulation results, the proposed algorithm can converge to a tap-length, which provided a good trade-off between the steady-state performance and computational complexity. Due to the similarity of the optimal tap-length model for the LMS algorithm and the dLMS algorithm, more research is required for variable tap-length distributed algorithms. The idea of variable tap-length can be potentially extended to distributed networks with different topologies.

7.2 Overall conclusions and recommendations of future research

The study of steady-state performance of dLMS revealed new insights into the energy flow through an incremental network for non-Gaussian

data. The work on VTLMS algorithms provided new tap-length adjustment techniques for applications where the optimal tap-length of the adaptive filter is unknown. The research results on incremental APA and diffusion APA were very useful in several problems involving distributed networks. The analyses of both algorithms also provided the insights into their mean-square performance. The work of variable tap-length dLMS showed that the proposed algorithm achieves a good performance in both tap-length and tap weight estimation in an incremental network. To make the proposed variable tap-length dLMS estimation approach more practical in reality, future research is suggested:

1. The concept of variable tap-length can be potentially used with various adaptive rules in networks with various topologies, including diffusion networks and networks with dynamic topology. The improvement of the performance of distributed adaptive networks with adjusting the structures of the filters is then the most valuable work for further research.

2. In order to choose reasonable parameters of variable tap-length dLMS, future work should include analysis for the deficient tap-length case of adaptive filters in distributed adaptive networks. More theoretical research is needed to investigate the relationship between the performance of distributed adaptive networks with different topologies and the tap-length order.

3. The existing two VTLMS algorithms, including the research results shown in this thesis are not suitable for applications where both input and noise signals are statistical nonstationarity. In further work, new VTLMS algorithms, robust to such nonstationarity, should there-

fore be pursued and could be extended to distributed adaptive estimation. In addition, future work should also include evaluating the performance of the FT algorithm with coloured signals to provide a good guide for parameter selection.

REFERENCES

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

[2] F. Cattivelli and A. H. Sayed, "Diffusion LMS algorithms with information exchange," in *Proc. Asilomar Conf. on Signals, Systems and Computers*, Los Angeles, USA, October 2008.

[3] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, 1985.

[4] S. Haykin, *Adaptive Filter Theory*, Second Edition, Prentice Hall, Englewood Cliffs, NJ, 1996.

[5] B. Farhang-Boroujeny, *Adaptive Filters: Theory and Applications*, Wiley, NJ, 1998.

[6] A. H. Sayed, *Fundamentals of Adaptive Filters*, Wiley, NJ, 2003.

[7] A. H. Sayed, *Adaptive Filters*, Wiley, NJ, 2008.

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

[9] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optmization," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 798 – 808, April 2005.

[10] M. Wax and T. Kailath, "Decentralized processing in sensor arrays," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 33, no. 5, pp. 1123 – 1129, October 1985.

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

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

[13] F. Riera-Palou, J. M. Norsa and D. G. M. Cruickshank, "Linear equalisers with dynamic and automatic length seletion," *Electronics Letters*, vol. 37, no. 25, pp. 1553 – 1554, 2001.

[14] Y. Gu, K. Tang, H. Cui, and W. Du, "LMS algorithm with gradient descent filter length," *IEEE Signal Processing Letters*, vol. 11, no. 3, pp. 305 – 307, March 2004.

[15] Y. Gong and C. F. N. Cowan, "A novel variable tap-length algorithm for linear adaptive filter," in *Proc. IEEE Int. Cof. Acoustic, Speech, Signal Processing (ICASSP)*, Montreal, Quebec, Canada, May 2004, pp. 825 – 828.

[16] Y. Gong and C. F. N. Cowan, "Structure adaptation of linear MMSE adaptive filters," *Proc. Inst. Elect. Eng.-Vision, Image, Signal Process.*, vol. 151, no. 4, pp. 271 – 277, August 2004.

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

[18] Y. Zhang and J. A. Chambers, "A variable tap-length natural gradient blind deconvolution/equalization algorithm," *Electronic Letters*, vol. 43, no. 14, 2007.

[19] A. H. Sayed and F. Cattivelli, *Distributed adaptive learning mechanisms*, Handbook on Array Processing and Sensor Networks, S. Haykin and K. J. Ray Liu, editor, Wiley, NJ, 2009.

[20] C. G. Lopes and A. H. Sayed, "Distributed processing over adaptive networks," in *Proc. Adaptive Sensor Array Processing Workshop*, MIT Lincoln Laboratory, MA, June 2006, pp. 917 – 920.

[21] C. G. Lopes and A. H. Sayed, "Distributed adaptive incremental strategies: formulation and performance analysis," in *Proc. IEEE Int. Cof. Acoustic, Speech, Signal Processing (ICASSP)*, Toulouse, France, May 2006, pp. 584 – 587.

[22] C. G. Lopes and A. H. Sayed, "Diffusion least-mean-squares over adaptive networks," in *Proc. IEEE Int. Cof. Acoustic, Speech, Signal Processing (ICASSP)*, Honolulu, Hawaii, April 2007, pp. 917 – 920.

[23] C. G. Lopes and A. H. Sayed, "Diffusion adaptive networks with changing topologies," in *Proc. IEEE Int. Cof. Acoustic, Speech, Signal Processing (ICASSP)*, Las Vegas, April 2008, pp. 3285 – 3288.

[24] C. G. Lopes and A. H. Sayed, "Steady-state performance of adaptive diffusion least-mean squares," in *Proc. IEEE Workshop on Sta-*

tistical Signal Processing (SSP), Madison, WI, August 2007, pp. 136 – 140.

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

[26] A. H. Sayed and C. G. Lopes, "Adaptive estimation algorithms over distributed networks," in *Proc. 21st IEICE Signal Processing Symposium*, Kyoto, Japan, November 2006.

[27] A. H. Sayed and C. G. Lopes, "Distributed recursive least-squares strategies over adaptive networks," in *Proc. Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, October-November 2006, pp. 233 – 237.

[28] F. Cattivelli, C. G. Lopes, and A. H. Sayed, "A diffusion RLS scheme for distributed estimation over adaptive networks," in *Proc. IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Helsinki, Finland, June 2007, pp. 1 – 5.

[29] F. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Trans. on Signal Processing*, vol. 56, no. 5, pp. 1865 – 1877, May 2008.

[30] D. Bertsekas, "A new class of incremental gradient methods for least squares problems," *SIAM J. Optim.*, vol. 7, no. 4, pp. 913 – 926, November 1997.

[31] A. Nedic and D. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," *SIAM J. on Optimization*, vol. 12, no. 1, pp. 109 – 138, 2001.

[32] M. G. Rabbat and T. D. Nowak, "Decentralized source localization and tracking," in *Proc. IEEE Int. Cof. Acoustic, Speech, Signal Processing (ICASSP)*, Montreal, QC, Canada, May 2004, pp. 921 – 924.

[33] T. Y. A. Naffouri and A. H. Sayed, "Transient analysis of data-normalized adaptive filters," *IEEE Trans. on Signal Processing*, vol. 51, no. 3, pp. 639 – 652, March 2003.

[34] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Third Edition, The Johns Hopkins University Press, 1996.

[35] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, New York, 2002.

[36] R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *Proc. 44th IEEE Conference on Decision and Control*, December 2005, pp. 6698 – 6703.

[37] 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, USA, April 2005, pp. 63 – 70.

[38] R. Olfati-Saber, "Distributed Kalman Filter with Embedded Consensus Filters," in *CDC-ECC'05*, December 2005, pp. 8179 – 8184.

[39] S. Shimauchi and S. Makino, "Stereo projection echo canceler with true echo path estimation," in *Proc. IEEE Int. Cof. Acoustic, Speech, Signal Processing (ICASSP)*, Detroit, MI, May 1995, pp. 3059 – 3062.

[40] Y. Kaneda, M. Tanaka and J. Kojima, "An adaptive algorithm with fast convergence for multi-input sound control," in *Proc. Active*, Newport Beach, CA, July 1995, pp. 993 – 1004.

[41] S. G. Sankaran and A. A. Beex, "Stereophonic acoustic echo cancellation using NLMS with orthogonal correction factors," in *Proc. Int. Workshop Acoust. Echo Noise Contr.*, Pocono Manor, PA, September 1999, pp. 40 – 43.

[42] H-C. Shin and A. H. Sayed, "Transient behavior of affine projecion algorithms," in *Proc. IEEE Int. Cof. Acoustic, Speech, Signal Processing (ICASSP)*, Hong Kong, April 2003, pp. 353 – 356.

[43] H-C. Shin and A. H. Sayed, "Mean-square performance of a family of affine projection algorithms," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 52, no. 1, pp. 90 – 102, January 2004.

[44] N. R. Yousef and A. H. Sayed, "A unified approach to the steady-state and tracking analyzes of adaptive filters," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 49, no. 2, pp. 314 – 324, February 2001.

[45] D. P. Mandic, "A generalized normalized gradient descent algorithm," *IEEE Signal Processing Letters*, vol. 11, no. 2, pp. 115 – 118, January 2004.

[46] L. A. Rossi, B. Krishnamachari and C. C. J. Kuo, "Distributed parameter estimation for monitoring diffusion phenomena using physical models," in *Proc. 1st IEEE Conf. Sensor Ad Hoc Communications Networks*, Santa Clara, CA, October 2004, pp. 460 – 469.

[47] D. Li, K. D. Wong, Y. H. Yu and A. M. Sayeed, "Detection, classification and tracking of targets," *IEEE Commun. Mag.*, vol. 19, no. 2, pp. 17 – 29, March 2007.

[48] I. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "A surey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102 – 114, August 2002.

[49] D. Spanos, R. Olfati-Saber and R. Murray, "Distributed sensor fusion using dynamic consensus," presented at the 16th IFAC World Congr., Prague, Czech Republic, July 2005.

[50] A. Jadbabaie, J. Lin and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. on Autom. Control*, vol. 48, no. 6, pp. 988 – 1001, June 2003.

[51] R. H. Koning, H. Neudecker and T. Wansbeek, "Block Kronecker products and the vecb operator," Economics Dept., Institute of Economics Research, Univ of Groningen, Groningen, The Netherlands, Research Memo. No. 351, 1990.

[52] D. S. Tracy and R. P. Singh, "A new matrix product and its applications in partitioned matrix differentiation," *Statistica Neerlandica*, vol. 26, no. NR.4, pp. 143 – 157, 1972.

[53] Y. Wei and F. Zhang, "Equivalence of a matrix product to the Kronecker product," *Hadronic J. Supp.*, vol. 15, no. 3, pp. 327 – 331, August 2000.

[54] Y. Gu, K. Tang, H. Cui, and W. Du, "Convergence analysis of a deficient-length LMS filter and optimal-length sequence to model expo-

nential decay impulse response," *IEEE Signal Processing Letters*, vol. 10, no. 1, pp. 4 – 7, January 2003.

[55] K. Mayyas, "Performance analysis of the deficient length LMS adaptive algorithm," *IEEE Trans. on Signal Processing*, vol. 53, no. 8, pp. 2727 – 2734, August 2005.

[56] Y. Zhang and J. A. Chambers, "Convex combination of adaptive filters for variable tap-length LMS algorithm," *IEEE Signal Processing Letters*, vol. 13, no. 10, pp. 628–631, October 2006.

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

[58] J. M. Cioffi, "Limited-precision effects in adaptive filtering," *IEEE Trans. on Circuits Systems*, vol. 34, no. 7, pp. 821 – 833, July 1987.

[59] M. Kamenetsky and B. Widrow, "A variable leaky LMS adaptive algorithm," in *Proc. Asilomar Conf. on Signals, Systems and Computers* , Los Angeles, USA, November 2004, pp. 125 – 128.

[60] A. Segalen and G. Demoment, "Constrained LMS adaptive algorithm," *Electronic Letters*, vol. 18, no. 5, pp. 226 – 227, May 1982.

[61] T. Aboulnasr and K. Mayyas, "A robust variable step-size LMS-type algorithm: analysis and simulations," *IEEE Trans. on Signal Processing*, vol. 45, no. 3, pp. 631 – 639, March 1997.

[62] H-C. Shin, A. H. Sayed and W. J. Song, "Variable step-size NLMS and affine projection algorithms," *IEEE Signal Processing Letters*, vol. 11, no. 2, pp. 132 – 135, February 2004.

[63] J. E. Greenberg, "Modified LMS algorithm for speech processing with an adaptive noise canceller," *IEEE Trans. on Signal Processing*, vol. 6, no. 4, pp. 338 – 351, July 1998.

[64] B. Widrow *et al.*, "Stationary and nonstationary learing characteristics of the LMS adaptive filer," *Proceedings of the IEEE*, vol. 64, no. 8, pp. 1151 – 1162, February 1976.

[65] A. Feuer and E. Weinstein, "Convergence analysis of LMS filters with uncorrelated Gaussian data," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 33, no. 1, pp. 222 – 229, February 1985.

[66] Y. Zhang, J. A. Chambers, S. Sanei, P. Kendrick and T. J. Cox, "A new variable tap-length LMS algorithm to model an exponential decay impulse response," *IEEE Signal Processing Letters*, vol. 14, no. 4, pp. 263–266, April 2007.

[67] E. Walach and B. Widrow, "The least mean fourth (LMF) adaptive algorithm and its family," *IEEE Trans. Information Theory*, vol. 30, pp. 275–283, July 1984.

[68] P. I. Hbscher, J. C. M. Bermudez and V. H. Nascimento, "A mean-square stability analysis of the least mean forth adaptive algorithm," *IEEE Trans. on Signal Processing*, vol. 55, no. 8, pp. 4018 – 4028, August 2007.