Low-Complexity Adaptive Filtering Algorithms Based on the Minimum L_∞-norm Method

Abhishek Tandon

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science (Electrical and Computer Engineering) at
Concordia University,
Montreal, Quebec, Canada

September 2004

Your file Votre référence
ISBN: 0-612-94711-4
Our file Notre référence
ISBN: 0-612-94711-4

ABSTRACT

Low-Complexity Adaptive Filtering Algorithms Based on
the Minimum L_∞-norm Method

Abhishek Tandon

The complexity of an adaptive filtering algorithm is proportional to the tap length of the filter and hence, may become computationally probhitive for applications requiring a long filter tap. In this thesis, we provide a framework for developing low-complexity adaptive filter algorithms by utilizing the concept of partial-updating along with the technique of finding the gradient vector in the hyperplane based on the L_∞-norm criterion. The resulting algorithm should have low-complexity not only because of the updating of only a subset of the filter coefficients at each time step, but also from the fact that updating a filter coefficient using the algorithm based on L_∞-norm requires less number of operations compared to the L_2-norm algorithm. Two specific coefficient selection techniques, namely the sequential and M-Max coefficient selection techniques, are considered in this thesis. Statistical analyses of these two algorithms are carried out to derive the evolution equations for the mean and mean-square of the filter coefficient misalignment as well as to obtain stability bounds on the step-size of the two algorithms. Further, these analyses are used to show that the algorithm employing the M-Max coefficient selection technique can achieve a convergence rate that is closest to the full update algorithm. As a consequence, even though there are various ways of selecting a subset of the filter coefficients, the study of the other techniques becomes redundant. Simulations are carried out to validate the results obtained from the statistical analyses of the algorithms. The concept of developing algorithms based on the partial-updating and L_∞-norm is extended to proportionate adaptive filtering. Finally, the performance of the proposed adaptive filtering algorithms as well as that of the existing ones is studied in echo cancellation.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my thesis supervisors Dr. M. Omair Ahmad and Dr. M. N. S. Swamy for their interest and constructive criticism of my work. Their advice, encouragement and close support have been invaluable.

I sincerely thank Dr. William Lynch, Dr. Eugene Plotkin and Dr. Weiping Zhu for insightful and timely discussions on research related problems at one point or another during the course of this thesis.

It is my great pleasure to thank my friends and colleagues at the Center for Signal Processing and Communications for the many fruitful discussions which we have had during the course of this work. I owe special thanks to my family for their encouragement and support, without which it would have been impossible to finish this work.

Abhishek Tandon, September 2004

To My Loving Family

Contents

List of Figures

List of Tables

List of Symbols

$\mathbf{A}(k)$	Coefficient selection matrix at time k
$\triangle(k)$	Variable vector
δ	Small positive number to prevent the division by zero
$e(k)$	Error in predicting output of the unknown system
$E\{.\}$	Expectation operator
$\mathbf{G}(k)$	Gain distribution matrix
λ	Lagrange multiplier
M	Number of filter coefficients updated during each time step
N	Length of the filter coefficients vector
$\sigma_{\tilde{w}}(k)$	Trace of the filter coefficient misalignment covariance matrix
σ_x^2	Variance of the input signal $x(k)$
σ_n^2	Variance of the noise $n(k)$
$sign\{.\}$	Sign operator
$\mathbf{V}(k)$	Filter coefficient misalignment covariance matrix
τ	Convergence rate of the evolution equation for the mean-square of the filter coefficient misalignment
$\mathbf{W}$	Optimal (Wiener) filter solution

$\mathbf{W}(k)$	Filter coefficient vector at time k
$\mathbf{X}(\mathbf{k})$	Input regressor vector
$y(k)$	Output of the unknown system
$\widehat{y}(k)$	Output of the adaptive filter
$\|.\|_p$	L_p-norm operator
$\widetilde{\mathbf{W}}(k)$	Filter coefficient misalignment vector

List of Abbreviations

A/D	Analog to digital
D/A	Digital to analog
dB	Decibel
DTD	Double-talk-detector
FCM	Filter coefficient misalignment
FIR	Finite impulse response
IIR	Infinite impulse response
LMS	Least-mean-square
NLMS	Normalized least-mean-square
NSLMS	Normalized signed least-mean-square
PDF	Probability density function
PU	Partial-update
PCU	Predetermined coefficient updating
PU-NSLMS	Partial-update normalized signed least-mean-square
PU-PNSLMS	Partial-update proportionate normalized signed least-mean-square
PNLMS	Proportionate
SCU	Selective coefficient updating
SPU-NLMS	Selective-partial-update normalized least-mean-square
SNR	Signal to noise ratio

Chapter 1

Introduction

1.1 Adaptive Filter

Filters are devices used for removing undesired components from a given input while not affecting the desired ones. The very fact that they are supposed to extract the useful information out of a noisy set of data makes them useful in different fields of engineering such as sonar, communications, and speech processing. Depending on the kind of operators used, the filters can be classified as linear or nonlinear. A linear filter uses linear operators on the noisy input signal such that the effect of noise is minimized at the output based on some criterion. A useful criterion that is often employed in designing filters is to minimize the mean-square value of the error between the desired response and the actual output of the filter. The use of this criterion leads to a mean-square optimal filter, known as the Wiener filter [1].

In deriving the Wiener filter, it is assumed that certain statistical parameters of the noisy input signal are available. This might not be the case, when there is no *a priori* knowledge of the signal statistics or when the system statistics are nonstationary. In such cases, it becomes difficult to obtain a Wiener filter. One way of solving this problem is to estimate

the statistical parameters of the input signal and then to adjust the internal parameters of the filter accordingly. This can be done using an *adaptive filter* [1], [2].

Adaptive filters have the ability to operate and adapt themselves to an unknown environment. To start with, the coefficients of these self-tuning filters are set to some initial values based on whatever knowledge we have about the unknown system. For a stationary environment, the coefficients converge to the Wiener solution after a number of iterations. For a nonstationary environment, the adaptive filters have the tracking capability, i.e., the coefficients can adjust themselves to the changes provided these changes are sufficiently slow. Adaptive filter algorithms are recursive, which means the internal parameters of the filter can be updated from one iteration to the next. Since the coefficients are time varying and can be updated using a particular recursive adaptive algorithm, adaptive filters belong to the category of nonlinear filters. Hence, properties such as additivity and homogeneity are not applicable to such filters. In addition, these filters are data dependent. Therefore, it is also very difficult to analyze the statistical behavior of such filters as opposed to those of time-invariant filters.

The coefficients of an adaptive filter are obtained using *a posterior* data that never yields a complete picture of the input signal statistics. Also, in most applications of adaptive filters, the unknown systems are nonstationary. Therefore, it is next to impossible to obtain an optimal model of the unknown system using an adaptive filter. For these reasons, we have to settle with suboptimal models of the unknown systems. Depending on the application, one may select or develop an adaptive filtering algorithm. The choice of an adaptive filtering algorithm is dependent on one or more of the following factors:

- *Computational cost* - Amount of computational resources consumed within a processor while implementing the algorithm.

- *Convergence rate* - Number of iterations required to obtain a solution that is close to

the Weiner solution for a stationary environment. A fast convergence rate is always desirable provided all the other constraints are also met.

- *Tracking rate* - Capability of an adaptive filter algorithm to adjust itself in case of a nonstationary environment.

- *Steady-state error* (Filter coefficient misalignment)- Steady-state error while obtaining the Weiner solution.

- *Robustness* - Capability of the algorithm for not to become divergent in the presence of small disturbances.

The ability of an adaptive filter to change its internal parameters to adapt itself to the changes in the environment makes it a powerful device for signal processing. Adaptive filters find applications in diverse fields such as sonar, biomedical engineering, communications and speech processing. Depending on the overall structure that is used for updating the filter coefficients, various applications of an adaptive filter are classified into the following four categories [1]:

1. *System-identification* - Modeling an unknown system such that for the same input to the mathematical model and the unknown system, the error between their outputs is minimized for a specific criterion.

2. *Inverse-modeling* - Developing a best-fitting inverse mathematical model of the unknown system.

3. *Prediction* - Developing a mathematical model such that it best predicts the present value of the random variables based on their past values.

4. *Interference* - Removal of an undesired interference signal from the input signal.

1.2 System-Identification using Adaptive Filters

Identifying nonstationary unknown systems, such as the impulse response of a communication channel, is one of the most common applications of an adaptive filter (see Fig. 1.1). The performance requirement of an adaptive filter for the system-identification application is typically set by the desired response, which is the output of the unknown system. The objective of the adaptive filter algorithm is to provide a linear model of the unknown system such that for same input, the difference between the outputs of the mathematical model of the unknown system and the unknown system itself is minimized for some criterion. Let us assume that we have an unknown system whose output is $y(k)$ for the an input sequence $x(k)$ at time k. The output of the mathematical model $\widehat{y}(k)$ using an adaptive filter is given by

$$\widehat{y}(k) = W^T(k) X(k) \tag{1.1}$$

where $W(k) = [w_1(k), w_2(k), \ldots, w_N(k)]^T_{N\times 1}$ is the filter coefficient vector, and $X(k) = [x(k), x(k-1), \ldots, x(k-N+1)]^T_{N\times 1}$ is the input regressor vector. Using some adaptive algorithm, the coefficients of the adaptive filter are varied such that the error

$$e(k) = y(k) - \widehat{y}(k) \tag{1.2}$$

is minimized. In order to update the filter coefficients, the error $e(k)$ is fed back to the adaptive filter. When the model of the unknown system is a true representation of the unknown system, then $e(k)$ becomes zero and the updating of the filter coefficients stops. However, in most cases, the internal parameters of the unknown system are nonstationary. In such cases, $e(k)$ never becomes zero and the adaptive filter keeps on tracking the variations in the unknown system.

Adaptive filtering consists of two major operations.

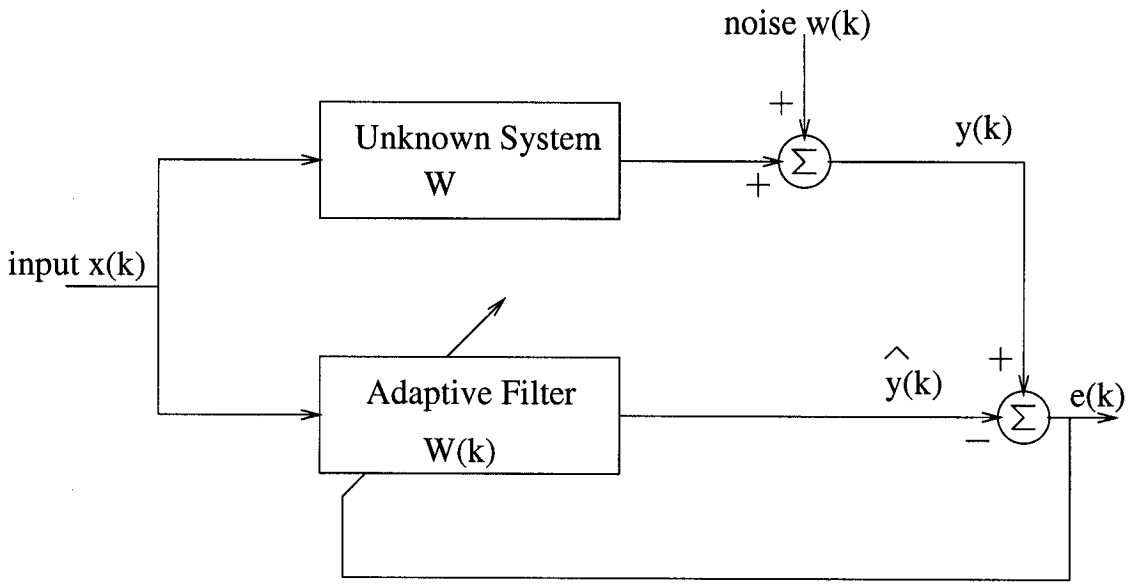

Figure 1.1. Modeling an unknown system using an adaptive filter

- Transversal filtering
- Coefficient updating

Transversal filtering involves computing the output of the modeled system. This is done by multiplying the filter coefficient vector, $W(k)$, with the input regressor vector $X(k)$. For a filter of length N, transversal filtering requires N multiplications and $N-1$ additions. Since the input regressor vector changes at each time step, the number of operations required for transversal filtering is always constant in terms of additions and multiplications. The other adaptive filtering operation, namely, the coefficient updating, involves updating the filter coefficient vector using some adaptive algorithm. Its complexity, which is usually $O(N)$ operations, depends on the adaptive algorithm used for updating the coefficients. Therefore, the overall complexity of the adaptive filter algorithm is proportional to the tap length.

1.3 Scope and Objective of the Thesis

System-identification applications such as network echo cancellation, require large number of filter taps for modeling an unknown system with sufficient accuracy. As mentioned earlier, the complexity of an adaptive filter algorithm is proportional to the tap length of the filter. Therefore, an algorithm may become computationally probhitive for applications requiring a long filter tap, especially for real-time implementations. Furthermore, in applications such as network echo cancellation, the challenge is to increase the convergence rate of the algorithms, as well as to increase the number of channels that can be handled by the available real-time resources. For these reasons, there is an ever increasing demand of low complexity, fast converging adaptive filter algorithms.

The choice of algorithms range from the complex fast converging least-square-based (e.g. recursive-least-square) algorithms to the relatively simple slow converging stochastic-gradient-based (e.g. least-mean square) algorithms. The use of complex least-square-based algorithms can sometime be computationally probhitive for real-time applications, in which case we resort to the simpler stochastic-gradient-based algorithms (e.g. NLMS algorithm). However, requirements such as the large number of filter coefficients needed to model the unknown system with sufficient accuracy and the possibility of making maximum use of the available real-time resources can impair the possibility of employing the NLMS algorithm. One approach for reducing the complexity of the algorithms is to update only a subset of the filter coefficients at each time step [3]-[13]. Since the complexity for updating the filter coefficients is proportional to the length of the filter coefficients, the reduction in the complexity of the partial-update (PU) algorithms is proportional to the reduction in the number of filter coefficients updated. Another efficient approach for reducing the complexity of algorithms is to employ quantization in the filter coefficient updates, such as in the case of the simplified NLMS algorithm due to Nagumo and Noda [14], [15], which

is based on the concept of finding the minimum gradient vector in the hyperplane using the L_∞-norm criterion. It requires less number of computational operations for updating the filter coefficients compared to the NLMS algorithm, which is based on the L_2-norm.

The objective of this thesis is to provide a framework for developing low complexity adaptive filtering algorithms by combining the concepts of partial updating and the technique of finding the gradient vector in the hyperplane based on the L_∞-norm criterion. The complexity of the resulting algorithms is reduced not only because of the updating of only a subset of the filter coefficients, but also from the fact that compared to algorithms based on L_2-norm algorithm (for example, NLMS algorithm), algorithms based on the L_∞-norm method require less number of operations for updating each filter coefficient. Due to the significant reduction in the computational complexity, the proposed algorithms are well suited for applications requiring a large number of filter taps in order to model an unknown system with sufficient accuracy.

The organization of the thesis is as follows. In Chapter 2, a brief introduction to the different approaches used for developing adaptive filtering algorithms is given. Some of the low-complexity stochastic-gradient algorithms are discussed in detail. The derivation of these algorithms using constrained optimization method are also presented.

In Chapter 3, we provide a framework for developing low-complexity adaptive filtering algorithms by combining the concepts of partial updating and the technique of finding the gradient vector in the hyperplane based on the L_∞-norm criterion. The resulting algorithm, referred to as partial-update normalized sign LMS (PU-NSLMS) algorithm, updates a subset of the filter coefficients of the simplified NLMS algorithm due to Nagumo and Noda [14], [15]. The PU-NSLMS algorithm is derived by employing the error control procedure as suggested in [15]. Then, two specific cases of the general PU-NSLMS algorithm, namely, sequential and M-Max PU-NSLMS are proposed.

In Chapter 4, statistical analysis of the two proposed PU-NSLMS algorithms are carried

out. It is shown that for a selected value of the step-size, the evolution equation for the mean of the *filter coefficient misalignment* (FCM) converges to zero. The evolution equation for the mean-square of the filter coefficient misalignment as well as the stability bounds on the step-size of the algorithms are also obtained. The M-Max PU-NSLMS algorithm is shown to have a convergence rate that is closest to that of the full update algorithm. Finally, simulations are carried out to validate the theoretical results.

In Chapter 5, we first review the proportionate-stochastic-gradient algorithms. Next, the notion of partial updating is extended to the proportionate algorithms based on the L_∞-norm to develop a low-complexity proportionate algorithm whose convergence rate is comparable to that of the full update algorithm.

In chapter 6, the application of adaptive filters in echo cancellation is considered. The performance of the proposed filtering algorithms as well as that of the existing ones is studied in echo cancellation.

In chapter 7, we conclude the thesis by summarizing the results obtained and discussing the scope for future work.

Chapter 2

Adaptive Filter Algorithms

In this chapter, a brief introduction to the different approaches used for developing the adaptive filtering algorithms is first given. Then, some of the low-complexity adaptive filter algorithms based on the stochastic-gradient approach are discussed. The derivation of these algorithms using constraint optimization methods is also presented.

2.1 Overview

There are a number of approaches that exist for developing adaptive filter algorithms and these may be broadly classified into two categories [1].

1. **Stochastic-gradient** - The procedure of deriving the stochastic-gradient based adaptive filter algorithms consists of two steps. The first step involves developing a recursive gradient-based algorithm, which after successive iterations, converges to the Wiener filter solution. These recursive equations for computing the gradient vector during each iteration require the input correlation matrix and the cross-correlation vector between the input signal and the desired response. In the second step, the two correlations are replaced by their respective stochastic estimates to yield the

stochastic-gradient algorithm. Replacing the correlations in the *steepest-decent* algorithm by their respective stochastic estimates leads to the well known least-mean-square (LMS) algorithm [16]. The LMS algorithm was first proposed by Widrow and Hoff in 1960, and since then, a great deal of work has been carried out to analyze the performance of this simple, yet effective, adaptive filter algorithm [18]-[23]. Stochastic-gradient algorithms, such as the LMS algorithm, have simple architectures and are well suited for applications requiring low-complexity algorithms.

2. **Least-square** - Algorithms based on least-square estimation are more complex compared to those based on the stochastic-gradient approach. The least-square method involves the minimization of the weighted error square cost function, where the error is the difference between the output of the mathematical model of the unknown system and the unknown system itself for the same input. Compared to the stochastic-gradient algorithms, the least-square based algorithms have faster convergence rates in a stationary environment. However, high complexity and numerical instability of the least-square based algorithms place a limitation on their practical applications. It should also be noted that the performance of the least-square based algorithms are dependent on the mathematical model of the unknown system. Therefore, their tracking rate can be inferior to that of the stochastic-gradient algorithms, which are model independent.

2.2 NLMS Algorithm

The LMS algorithm having a simple architecture, and yet the ability to give a satisfactory performance, is one of the most popular and widely used stochastic-gradient algorithms.

The coefficient-update equation of the LMS algorithm is given by

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \mu e(k)\mathbf{X}(k) \tag{2.1}$$

$$e(k) = y(k) - \mathbf{W}^T(k)\mathbf{X}(k) \tag{2.2}$$

where μ is the step-size, $y(k)$ is the desired response of the adaptive filter, $\mathbf{W}(k) = [w_1(k), w_2(k), \ldots, w_N(k)]^T_{N\times 1}$ is the filter coefficient vector, and $X(k) = [x(k), x(k-1), \ldots, x(k-N+1)]^T_{N\times 1}$ is the input regressor vector.

The convergence rate of the LMS algorithm is dependent on the relationship between the step-size and the input signal statistics [20]-[23]. The choice of a step-size that is good for a given environment might lead to a poor performance or even a divergence of the LMS algorithm once the environment changes. Therefore, the user of the LMS algorithm may have to choose overly conservative values of the step-size to avoid the divergence of the algorithm. This might ultimately cause an undesirably slow convergence rate of the LMS algorithm. In order to overcome this problem, the normalized LMS (NLMS) algorithm has been proposed [14], [24]. The NLMS algorithm can be viewed as the solution of the following constrained optimization problem [17]:

$$\begin{aligned} &\min_{\mathbf{W}(k+1)} \|(\mathbf{W}(k+1) - \mathbf{W}(k))\|_2^2 \\ &\text{subject to } \mathbf{W}^T(k+1)\mathbf{X}(k) = y(k) \end{aligned} \tag{2.3}$$

The L_2-norm constrained optimization problem in (2.3) is solved using the Lagrange multipliers. The cost function to be minimized is given by

$$J(k) = \|\mathbf{W}(k+1) - \mathbf{W}(k)\|_2^2 + \lambda\left(y(k) - \mathbf{W}^T(k+1)\mathbf{X}(k)\right) \tag{2.4}$$

where λ is the Lagrange multiplier. Solving $(\partial J/\partial\lambda) = 0$ and $(\partial J/\partial w_i(k+1)) = 0$ gives

$$y(k) = \mathbf{W}^T(k+1)\mathbf{X}(k) \tag{2.5}$$

and

$$w_i(k+1) - w_i(k) = \frac{\lambda}{2}x(k-i+1), \ i \in (1,\ldots,N) \tag{2.6}$$

From (2.5), (2.6), and (2.2), we have

$$\begin{aligned} \frac{\lambda}{2} &= \frac{y(k) - \mathbf{W}^T(k)\mathbf{X}(k)}{\mathbf{X}^T(k)\mathbf{X}(k)} \\ &= \frac{e(k)}{\mathbf{X}^T(k)\mathbf{X}(k)} \end{aligned} \tag{2.7}$$

Substituting (2.7) in (2.6), and introducing a small step-size μ, gives the NLMS algorithm:

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \frac{\mu e(k)\mathbf{X}(k)}{\mathbf{X}^T(k)\mathbf{X}(k) + \delta} \tag{2.8}$$

where δ is a small positive number added to the denominator to prevent the division by zero.

The normalizing factor of the NLMS algorithm ensures that the adaptation gain of the algorithm is independent of the changes in the environment statistics, contrary to that of the LMS algorithm. Therefore, users of the NLMS algorithm need not be overly conservative while choosing the step-size and can still achieve a faster convergence rate compared to the LMS algorithm in a changing environment. The normalizing factor of the NLMS algorithm at time k is given by

$$\begin{aligned} \sigma_x^2(k) &= \Sigma_{i=1}^{N} x^2(k-i+1) + \delta \\ &= \sigma_x^2(k-1) + x^2(k) - x^2(k-N) \end{aligned} \tag{2.9}$$

The above equation shows that the normalizing factor at time k can be computed using the normalizing factor at time $(k-1)$, and this would require only two additions and one multiplication. The step-size of the NLMS algorithm is divided by the normalizing factor to obtain the adaptation gain of the algorithm. Then, using this normalized adaptation gain, the NLMS algorithm updates its filter coefficients. The NLMS algorithm requires 2N+2 additions, 2N+2 multiplications, and one division for updating the N filter coefficients at each time step. It should be noted that the NLMS algorithm has an overhead of only two additions, one multiplication, and one division over the LMS algorithm.

Since the development of the NLMS algorithm, a great deal of study has been done regarding its statistical behavior [25]-[29]. It has been shown that the mean of the filter coefficients converge to the optimal (Wiener) solution in a stationary environment. For non-stationary environments, the NLMS algorithm shows a good tracking capability. The stability bounds on the step-size of the NLMS algorithm for input signals generated from stationary, zero-mean, white Gaussian process are given by

$$0 < \mu < 2 \tag{2.10}$$

which reaffirms the fact that the bounds of the step-size of the NLMS algorithm are independent of the changes in the environment statistics.

2.3 Partial-Update NLMS Algorithm

Requirements such as the large number of filter taps needed to model the unknown system with sufficient accuracy, and maximizing the number of implementations that can be handled within the available real-time resources may diminish the possibility of employing the NLMS algorithm. Since the computational complexity of the adaptive filter algorithms

depends on the number of filters taps, one may choose to reduce the complexity by updating only a subset of the filter coefficients at each time step [3]-[13]; this process known as partial-updating.

A partial-update NLMS (PU-NLMS) algorithm can be viewed as the solution of the following constraint optimization problem:

$$\min_{\mathbf{W}(k+1)} \|\mathbf{A}(k)(\mathbf{W}(k+1)-\mathbf{W}(k))\|_2^2 \quad \text{subject to } \mathbf{W}^T(k+1)\mathbf{X}(k)=y(k) \tag{2.11}$$

where $\mathbf{A}(k) = diag\{a_1(k), a_2(k), \ldots, a_N(k)\}_{N\times N}$ is a diagonal matrix whose elements are +1 or 0. Solving (2.11) in a fashion similar to that used in solving (2.3) gives the coefficient-update equation of the PU-NLMS algorithm:

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \frac{\mu e(k)\mathbf{A}(k)\mathbf{X}(k)}{\mathbf{X}^T(k)\mathbf{A}(k)\mathbf{X}(k)+\delta} \tag{2.12}$$

From the above equation, we see that $A(k)$, referred to as the coefficient selection matrix, is used to determine the subset of the filter coefficients that are updated at time k. It should be noted that $\mathbf{A}(k)$ has no unique solution, when $trace(\mathbf{A}(\mathbf{k})) = M$ is a constant less than N. In such cases, there are various ways of selecting a subset of the filter coefficients, and these may be broadly classified into the following two categories:

- Predetermined coefficient updating (PCU)
- Selective coefficient updating (SCU)

In the case of PCU, the subset of the filter coefficients to be updated at each time step is predetermined. Therefore, this method does not require any operation for obtaining the subset of the filter coefficients to be updated at a particular time step. However, such is not the case for SCU, which requires a selection process to be run to obtain the subset of the

filter coefficients to be updated. Therefore, a partial-update algorithm employing SCU is more complex compared to a partial-update algorithm employing PCU.

2.3.1 Sequential NLMS Algorithm

This section presents the PU-NLMS algorithm when the partial updating is carried out through a specific PCU technique, namely, the sequential updating [11]. The resulting algorithm is termed sequential NLMS algorithm. The coefficient-update equation of this algorithm is given by

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \frac{\mu e(k)\mathbf{A}(k)\mathbf{X}(k)}{\mathbf{X}^T(k)\mathbf{A}(k)\mathbf{X}(k)+\delta} \tag{2.13}$$

where

$$\mathbf{A}(k) = diag\{a_1(k), a_2(k), \ldots, a_N(k)\}$$

$$a_i(k) = \begin{cases} 1 & if\ (k+i)\ mod\ K = 0 \\ 0 & if\ (k+i)\ mod\ K \neq 0 \end{cases}, i \in (1, \ldots, N)$$

$$K = (N/M), K \in I$$

This algorithm sequentially updates $M = (N/K)$ filter coefficients at each time step. In order to do so, the sequential NLMS algorithm uses every K^{th} element of the input regressor vector at each time step and each element of the input regressor vector after K time steps. For $K = 1$, the sequential NLMS algorithm reduces to the NLMS algorithm. The sequential NLMS algorithm requires N+M+2 additions, N+M+2 multiplications, and one division for updating the M filter coefficients at each time step.

There is an inevitable reduction in the convergence rate of an adaptive filter algorithm, when only a subset of the filter coefficients are updated at each time step. In the case of PCU, the reduction in the convergence rate of the algorithm is proportional to the reduction

in the number of filter coefficients updated. This can be a major drawback, especially for applications requiring a long filter tap.

2.3.2 Selective-PU-NLMS Algorithm

This section presents the PU-NLMS algorithm when the partial updating is carried out through a specific SCU technique, namely, the M-Max updating [6]. The resulting algorithm is termed selective-PU-NLMS (SPU-NLMS) algorithm [4], [5]. This algorithm runs a selection process to determine the subset of the filter coefficients to be updated at a particular time step. The selection of the subset is carried out such that the Euclidean norm (L_2-norm) of the resulting coefficient-update vector is minimum.

Let the solution to the coefficient selection matrix $\mathbf{A}(k)$, for which the Euclidean norm of the coefficient-update vector is minimum, be denoted by $\overline{\mathbf{A}}(k)$. Then, the coefficient-update vector of the SPU-NLMS algorithm is given by

$$\mathbf{W}(k+1) - \mathbf{W}(k) = \frac{\mu e(k)\overline{\mathbf{A}}(k)\mathbf{X}(k)}{\mathbf{X}^T(k)\overline{\mathbf{A}}(k)\mathbf{X}(k)} \tag{2.14}$$

In the above equation, for given $e(k)$ and μ, the Euclidean norm of the coefficient-update vector $\overline{\mathbf{A}}(k)\Delta(k)$ is minimum, when $\left\|\overline{\mathbf{A}}(k)\mathbf{X}(k)\right\|_1$ is maximum. Therefore, the SPU-NLMS algorithm updates the filter coefficients corresponding to the M largest elements of the vector $|\mathbf{X}(k)|$. The coefficient-update equation of the SPU-NLMS algorithm is given by

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \frac{\mu e(k)\mathbf{A}(k)\mathbf{X}(k)}{\mathbf{X}^T(k)\mathbf{A}(k)\mathbf{X}(k) + \delta} \tag{2.15}$$

where

$$\begin{aligned}\mathbf{A}(k) &= \overline{\mathbf{A}}(k) \\ &= diag\{\overline{a}_1(k), \overline{a}_2(k), \ldots, \overline{a}_N(k)\}\end{aligned}$$

$$\overline{a}_i(k) = \begin{cases} 1 & if\, i = arg\, M\, maxima\, of\, |x(k-i+1)| \\ 0 & otherwise \end{cases}, i \in (1, \ldots, N) \qquad (2.16)$$

Since the SPU-NLMS algorithm runs a sorting procedure for selecting the subset of the filter coefficients at each time step, it is more complex compared to the sequential NLMS algorithm.

2.4 Simplified NLMS Algorithm due to Nagumo and Noda

Another efficient approach for reducing the complexity of the adaptive filter algorithm is to employ quantization in the filter coefficient updates, such as in the case of the simplified NLMS algorithm due to Nagumo and Noda [14], [15]. The simplified NLMS algorithm due to Nagumo and Noda is based on the concept of finding the gradient vector in the hyperplane using the L_∞-norm criterion and employs sign-data to update the filter coefficients.

2.4.1 Minimum L_∞-norm Solution

In order to be able to solve for the coefficient-update vector of the simplified NLMS algorithm due to Nagumo and Noda, we need the following lemma [15].

Lemma *Let* $\mathbf{A}$ *be a non-zero vector in a vector space* $\Re^N$*, and* b *a scalar quantity. Then,* $\mathbf{z} = \frac{\mathbf{b}}{\|\mathbf{A}\|_1} sign\{\mathbf{A}\}$ *is the minimum* L_∞*-norm solution of the linear equation* $\mathbf{A}^T\mathbf{Z} = b$*, where* $\|.\|_1$ *and* $sign\{.\}$ *are the* L_1*-norm and the sign operator, respectively.*

Proof: Let a_i and z_i denote the denote the i^{th} element of the vectors $\mathbf{A}$ and $\mathbf{Z}$, respectively.

Then,

$$\begin{aligned} |b| &= \left| \sum_{i=1}^{N} a_i z_i \right| \\ &\leq \max_{1 \leq 1 \leq N} \left| \sum_{i=1}^{N} a_i z_i \right| \\ &= \|\mathbf{Z}\|_\infty \|\mathbf{A}\|_1 \end{aligned} \tag{2.17}$$

Thus, for a non-zero $\mathbf{A}$

$$\|\mathbf{Z}\|_\infty \leq \frac{|b|}{\|\mathbf{A}\|_1} \tag{2.18}$$

Let $\overline{\mathbf{Z}}$ be the solution vector to the equation $\mathbf{A}^T\mathbf{Z} = b$ such that it satisfies

$$\left\|\overline{\mathbf{Z}}\right\|_\infty \geq \min_{\mathbf{A}^T\mathbf{Z}=b} \|\mathbf{Z}\|_\infty \tag{2.19}$$

It should be noted that the solution vector $\overline{\mathbf{Z}}$ is not unique. Assume that

$$\overline{\mathbf{Z}} = \frac{|b|}{\|\mathbf{A}\|_1} sign\{\mathbf{A}\} \tag{2.20}$$

Then,

$$\left\|\overline{\mathbf{Z}}\right\|_\infty = \frac{|b|}{\|\mathbf{A}\|_1} \geq \min_{\mathbf{A}^T\mathbf{Z}=b} \|\mathbf{Z}\|_\infty \tag{2.21}$$

From (2.18) and (2.21), we get

$$\min_{\mathbf{A}^T\mathbf{Z}=b} \|\mathbf{Z}\|_\infty = \left\|\overline{\mathbf{Z}}\right\|_\infty \tag{2.22}$$

Therefore,

$$\mathbf{Z} = \frac{|b|}{\|\mathbf{A}\|_1} sign\{\mathbf{A}\} \tag{2.23}$$

is the minimum L_∞-norm solution to the linear equation $\mathbf{A}^T\mathbf{Z} = b$. ■

2.4.2 Derivation of the Algorithm

For an adaptive filter, $e(k)$, the error in predicting the output of the unknown system is given by

$$e(k) = y(k) - \mathbf{W}^T(k)\mathbf{X}(k) \tag{2.24}$$

Let $e(k)$ be minimized by adjusting the filter coefficient vector $\mathbf{W}(k)$ using the vector $\Delta(k)$ to give

$$(1-\mu)e(k) = y(k) - \left[\mathbf{W}^T(k) + \Delta^T(k)\right]\mathbf{X}(k) \tag{2.25}$$

where $\Delta(k) \in \Re^N$ is the coefficient-update vector at time k and μ is the step-size. Substituting (2.24) in (2.25) yields

$$\mu e(k) = \Delta^T(k)\mathbf{X}(k) \tag{2.26}$$

Using the lemma given in Section 2.4.1, the minimum L_∞-norm solution of $\Delta(k)$ is given by

$$\Delta(k) = \frac{\mu e(k)}{\|\mathbf{X}(k)\|_1} sign\{\mathbf{X}(k)\} \tag{2.27}$$

Thus, the coefficient-update equation of the simplified NLMS algorithm due to Nagumo and Noda is given by

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \frac{\mu e(k)}{\|\mathbf{X}(k)\|_1 + \delta} sign\{\mathbf{X}(k)\} \tag{2.28}$$

where δ is a small positive number added to the denominator to prevent the division by zero.

The simplified NLMS algorithm of Nagumo requires 2N+2 additions, N+2 multiplications, and one division. Thus, the computational complexity of this algorithm is less than that of the NLMS algorithm. In terms of performance, the convergence rate of the simplified NLMS due to Nagumo and Noda is comparable to that of the NLMS algorithm.

For input signals generated from stationary, zero-mean, white Gaussian process, the stability bounds on the step-size of the simplified NLMS algorithm due to Nagumo and Noda algorithm are given by

$$0 < \mu < 2\left(\frac{2}{\pi}\right) \tag{2.29}$$

It should be noted that the above bound on the step-size is independent of the environment statistics and slightly tighter compared to that of the NLMS algorithm.

2.5 Summary

In this chapter, stochastic-gradient- and least-square-based approaches for developing the adaptive filter algorithms have been briefly discussed. The derivation of the NLMS algorithm by posing the adaptation problem as a L_2-norm constraint optimization problem and then solving it using the Lagrange multipliers have been explained. The derivation of the low-complexity partial-update NLMS algorithm, by introducing the coefficient selection matrix into the constraint optimization equations of the NLMS algorithm, has been presented. Two specific partial-update NLMS algorithms, based on the sequential and M-Max coefficient updating have been briefly discussed. Finally, the derivation of the simplified NLMS algorithm due to Nagumo and Noda [14],[15] using the minimum L_∞-norm method has been given.

Chapter 3

PU-NSLMS Algorithm

3.1 Introduction

The choice of algorithms range from the complex least-square-based (e.g. recursive-least-square) algorithms to the relatively simpler stochastic-gradient-based (e.g. least-mean-square) algorithms. In applications, where use of the computationally intensive least-square-based algorithms are probhitive, we resort to the simpler stochastic-gradient-based algorithms, such as the NLMS algorithm. However, requirements such as the large number of filter coefficients needed to model an unknown system with sufficient accuracy may diminish the possibility of employing the NLMS algorithm. The complexity of the adaptive filter algorithm can be reduced by updating only a subset of the filter coefficients at each time step [3]-[13]. The reduction in the complexity of the algorithm is proportional to the reduction in the number of filter coefficients updated at each time step. Another efficient approach for reducing the complexity of the algorithm is to employ quantization in filter coefficient updates, such as in the case of the simplified NLMS algorithm due to Nagumo and Noda [14], [15].

In this chapter, we propose a framework for developing low-complexity adaptive filtering algorithms by combining the concepts of partial updating and finding the gradient

vector in the hyperplane based on the L_∞-norm criterion. The resulting algorithm, referred to as the PU-NSLMS algorithm, can be expected to have a reduced complexity not only because of the updating of only a subset of the filter coefficients, but also from the fact that compared to an L_2-norm algorithm (for example, NLMS algorithm), algorithms based on the L_∞-norm require a smaller number of operations for updating each filter coefficient. Using the framework proposed in this chapter, two specific algorithms, namely sequential and M-Max PU-NSLMS algorithms, are proposed.

The organization of this chapter is as follows. Section 3.2 presents the derivation of the general PU-NSLMS algorithm. Sections 3.3 and 3.4 present two specific cases of the PU-NSLMS algorithm by employing the sequential [11] and M-Max [6] coefficient updating. In Section 3.5, complexity of the two proposed PU-NSLMS algorithms are discussed.

3.2 PU-NSLMS Algorithm

This section presents the derivation of the PU-NSLMS algorithm by employing the error control procedure suggested in [15]. Following notations are used while deriving and explaining the algorithm:

$y(k)$: Desired response of the adaptive filter at time k

$x(k)$: Input signal

$e(k)$: The error in predicting the desired response

$\mathbf{X}(k) = [x(k), x(k-1), \ldots, x(k-N+1)]^T_{N\times 1}$: Input regressor vector

$\mathbf{W}(k) = [w_1(k), w_2(k), \ldots, w_N(k)]^T_{N\times 1}$: Filter coefficient vector

$\mathbf{A}(k) = diag\{a_1(k), a_2(k), \ldots, a_N(k)\}_{N\times N}$: Coefficient selection matrix

having diagonal elements as +1 or 0

For an adaptive filter, $e(k)$ is given by

$$e(k) = y(k) - \mathbf{W}^T(k)\mathbf{X}(k). \tag{3.1}$$

The error is minimized by adjusting the filter coefficient vector $\mathbf{W}(k)$ using the vector $\mathbf{A}(k)\Delta(k)$ to obtain

$$(1-\mu)e(k) = y(k) - \left[\mathbf{W}^T(k) + \Delta^T(k)\mathbf{A}(k)\right]\mathbf{X}(k) \tag{3.2}$$

where $\Delta(k) \in \Re^N$ is a variable vector and μ is the step-size. From (3.1) and (3.2), we have

$$\mu e(k) = \Delta^T(k)\mathbf{A}(k)\mathbf{X}(k). \tag{3.3}$$

The minimum L_∞-norm solution of $\Delta(k)$ is given by [15]

$$\Delta(k) = \frac{\mu e(k)}{\|\mathbf{A}(k)\mathbf{X}(k)\|_1} sign\{\mathbf{A}(k)\mathbf{X}(k)\} \tag{3.4}$$

where $\|.\|_1$ and $sign\{.\}$ are the L_1-norm and sign operators, respectively. Thus, the coefficient-update equation of the PU-NSLMS algorithm is given by

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \frac{\mu e(k)}{\|\mathbf{A}(k)\mathbf{X}(k)\|_1 + \delta} sign\{\mathbf{A}(k)\mathbf{X}(k)\} \tag{3.5}$$

where δ is a small positive number added to the denominator to prevent the division by zero.

As discussed in Chapter 2, the coefficient selection matrix $\mathbf{A}(k)$ has no unique solution, when $trace(\mathbf{A}(k)) = M$ is a constant is less than N. In such a case, there are various ways of determining the subset of the filter coefficients to be updated at each time step, and these may be broadly classified as the PCU and SCU.

3.3 Sequential PU-NSLMS Algorithm

This section presents the PU-NSLMS algorithm when the partial updating is carried out through a specific PCU technique, namely, the sequential updating [11]. The resulting algorithm is termed sequential PU-NSLMS algorithm. The coefficient-update equation of this algorithm is given by

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \frac{\mu e(k)}{\|\mathbf{A}(k)\mathbf{X}(k)\|_1 + \delta} sign\{\mathbf{A}(k)\mathbf{X}(k)\} \tag{3.6}$$

where

$$\mathbf{A}(k) = diag\{a_1(k), a_2(k), \ldots, a_N(k)\}$$

$$a_i(k) = \begin{cases} 1 & if\ (k+i)\ mod\ K = 0 \\ 0 & if\ (k+i)\ mod\ K \neq 0 \end{cases}, i \in (1, \ldots, N) \tag{3.7}$$

and $K, M \in I$. This algorithm sequentially updates $M = (N/K)$ filter coefficients at each time step. In order to do so, the sequential PU-NSLMS algorithm uses every K^{th} element of the input regressor vector at each time step and each element of the input regressor vector after K time steps. For $K = 1$, the sequential PU-NSLMS algorithm reduces to the simplified NLMS algorithm due to Nagumo and Noda [14], [15].

There is an inevitable reduction in the convergence rate of an adaptive filter algorithm, when only a subset of the filter coefficients are updated at each time step. In the case of PCU, the reduction in the convergence rate of the algorithm is proportional to the reduction in the number of filter coefficients updated. This can be a major drawback, especially for applications requiring a long filter tap.

3.4 *M*-Max PU-NSLMS Algorithm

This section presents the PU-NSLMS algorithm when the partial updating is carried out through a specific SCU technique, namely the *M*-Max updating [6]. The resulting algorithm is termed *M*-Max PU-NSLMS algorithm. This algorithm runs a selection process to determine the subset of the filter coefficients to be updated at a particular time step. The selection of the subset is carried out such that the Euclidean norm (L_2-norm) of the resulting coefficient-update vector is minimum.

Let us assume that the solution to the coefficient selection matrix $\mathbf{A}(k)$ for which the Euclidean norm of the coefficient-update vector is minimum is denoted by $\overline{\mathbf{A}}(k)$. Then, the coefficient-update vector of the *M*-Max PU-NSLMS algorithm is given by

$$\begin{aligned} \overline{\mathbf{A}}(k)\Delta(k) &= \mathbf{W}(k+1)-\mathbf{W}(k) \\ &= \frac{\mu e(k)\, sign\{\overline{\mathbf{A}}(k)\mathbf{X}(k)\}}{\left\|\overline{\mathbf{A}}(k)\mathbf{X}(k)\right\|_1} \end{aligned} \tag{3.8}$$

In the above equation, for given $e(k)$ and μ, the Euclidean norm of the coefficient updated vector $\overline{\mathbf{A}}(k)\Delta(k)$ is minimum, when $\left\|\overline{\mathbf{A}}(k)\mathbf{X}(k)\right\|_1$ is maximum. Therefore, the *M*-Max PU-NSLMS algorithm updates the filter coefficients corresponding to the *M* largest elements of the vector $|\mathbf{X}(k)|$. The coefficient-update equation of the sequential PU-NSLMS algorithm is given by

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \frac{\mu e(k)}{\|\mathbf{A}(k)\mathbf{X}(k)\|_1 + \delta} sign\{\mathbf{A}(k)\mathbf{X}(k)\} \tag{3.9}$$

where

$$\begin{aligned} \mathbf{A}(k) &= \overline{\mathbf{A}}(k) \\ &= diag\{\overline{a}_1(k), \overline{a}_2(k), \ldots, \overline{a}_N(k)\} \end{aligned}$$

$$\overline{a}_i(k) = \begin{cases} 1 \; if \; i = arg M maxima \; of \; |x(k-i+1)| \\ 0 \qquad otherwise \end{cases} , i \in (1, \ldots, N) \tag{3.10}$$

The *M-Max* PU-NSLMS algorithm obtains the subset of the filter coefficients to be updated at each time step in a fashion similar to that used in the *M*-Max NLMS algorithm [6]. Since the *M*-Max PU-NSLMS algorithm runs a sorting procedure before updating the filter coefficients, its complexity is higher than that of the sequential PU-NSLMS algorithm.

3.5 Complexity

In this section, the computational complexities of the proposed sequential and *M*-Max PU-NSLMS algorithms are discussed. The sequential PU-NSLMS algorithm requires N+M+2 additions, N+2 multiplications, and one division to update M filter coefficients at each time step. Since the subset of the filter coefficients to be updated at each time step is predetermined, the sequential PU-NSLMS algorithm does not require additional operations to determine the coefficients. This is not true in the case of *M*-Max PU-NSLMS algorithm, where the selection of the subset of the filter coefficients is carried out at each time step. The elements of $|\mathbf{X}(k)|$ are sorted in an ascending order, and then, the filter coefficients corresponding to the M largest elements of $|\mathbf{X}(k)|$ are updated. Fast sorting algorithms such as the one in [30] require $2log_2(N)+2$ comparisons for sorting. Therefore, *M*-Max PU-NSLMS algorithm requires $2log_2(N)+2$ additional operations over the sequential PU-NSLMS algorithm. Furthermore, additional memory is required for storing the pointers of the sorted vector of $|\mathbf{X}(k)|$. The memory requirements can be reduced using the selective block coefficient updating [9].

3.6 Summary

In this chapter, we have provided a framework for developing low-complexity adaptive filtering algorithms by combining the concepts of partial updating and finding the gradient vector in the hyperplane based on the L_∞-norm criterion. The resulting PU-NSLMS algorithms have a reduced complexity not only because of the updating of only a subset of the filter coefficients, but also from the fact that compared to an L_2-norm algorithm, algorithms based on the L_∞-norm require less number of operations for updating each filter coefficient. Using the framework proposed in this chapter, two specific algorithms, namely, sequential and M-Max PU-NSLMS algorithms have been obtained by using the sequential and M-Max coefficient updating techniques for partial updating. It has been shown that the M-Max PU-NSLMS algorithm is computationally more intensive than the sequential PU-NSLMS algorithm.

Chapter 4

Statistical Analysis

4.1 Introduction

In adaptive signal processing, statistical analyses are generally carried out to better understand the behavior of the algorithms, such as the convergence to the Wiener solution, stability bounds on the step-size, among other.

In this chapter, the statistical analyses of the sequential PU-NSLMS and M-Max PU-NSLMS algorithms are carried out. In Section 4.2, assumptions made in order to make the analyses of the two PU-NSLMS algorithms tractable are discussed. In Section 4.3, evolution equations for the mean and mean-square of the filter coefficient misalignment, and stability bounds on the step-size of the sequential PU-NSLMS algorithm are obtained. In Section 4.4, results of the statistical analysis of the sequential PU-NSLMS algorithm are suitably modified to obtain the corresponding results of the M-Max PU-NSLMS algorithm. In Section 4.5, it is shown that the M-Max PU-NSLMS algorithm can achieve a convergence rate that is closest to that of the full update algorithm. In Section 4.6, simulations are carried out to validate the theoretical results obtained in Sections 4.3, 4.4 and 4.5.

4.2 Analysis of the Sequential PU-NSLMS Algorithm

An adaptive filter is a nonlinear filter, whose internal parameters are dependent on the input signal, and as a consequence, it is difficult to carry out the statistical analysis of such a filter without making certain assumptions. We make the following assumptions about the statistics of the input data and desired response of such a filter

- Vectors $\left[y(k)\,\mathbf{X}^T(k)\right]^T$ and $\left[y(l)\,\mathbf{X}^T(l)\right]^T$ are independent, when $l \neq k$. This assumption is also known as the *independence assumption* [31], [1], and it has been shown to produce theoretical results that can accurately predict the simulations.

- Input signal is generated from a stationary, zero-mean, white Gaussian process with a variance σ_x^2.

- Desired response of the adaptive filter $y(k)$ is given by

$$y(k) = \mathbf{W}^T\mathbf{X}(k) + n(k) \tag{4.1}$$

  where $\mathbf{W}$ is the optimum filter response, and $n(k)$ a stationary, zero-mean, white noise with a variance σ_n^2. The filter coefficient misalignment (FCM) vector $\widetilde{\mathbf{W}}(k)$ is given by

$$\widetilde{\mathbf{W}}(k) = \mathbf{W} - \mathbf{W}(k) \tag{4.2}$$

In this section, we derive the evolution equations for the mean and mean-square of the filter coefficients of the sequential PU-NSLMS algorithm. By taking the expectation of both sides of (3.9), we obtain

$$\begin{aligned} E\{\mathbf{W}(k+1)\} &= E\{\mathbf{W}(k)\} \\ &+ E\left\{\frac{\mu e(k)}{\|\mathbf{A}(k)X(k)\|_1} sign\{\mathbf{A}(k)\mathbf{X}(k)\}\right\} \end{aligned} \tag{4.3}$$

where $E\{.\}$ is the expectation operator. Assuming $\|\mathbf{A}(k)\mathbf{X}(k)\|_1$ to be uncorrelated with the $[e(k)\,sign\{\mathbf{A}(k)\mathbf{X}(k)\}]$, we have

$$\begin{aligned} E\{\mathbf{W}(k+1)\} &\approx E\{\mathbf{W}(k)\} \\ &+ \mu E\left\{\frac{1}{\|\mathbf{A}(k)\mathbf{X}(k)\|_1}\right\} E\{e(k)\,sign\{\mathbf{A}(k)\mathbf{X}(k)\}\} \end{aligned} \tag{4.4}$$

In order to compute the second expectation term on the right hand side of (4.4), $E\left\{\frac{1}{\|\mathbf{A}(k)\mathbf{X}(k)\|_1}\right\}$ is approximated as

$$E\left\{\frac{1}{\|\mathbf{A}(k)\mathbf{X}(k)\|_1}\right\} \approx \frac{1}{E\{\|\mathbf{A}(k)\mathbf{X}(k)\|_1\}}. \tag{4.5}$$

The expectation $E\{\|\mathbf{A}(k)\mathbf{X}(k)\|_1\}$ can be expressed as

$$\begin{aligned} E\{\|\mathbf{A}(k)\mathbf{X}(k)\|_1\} &= E\{\|\mathbf{X}(k)\|_1\}\frac{E\{\|\mathbf{A}(k)\mathbf{X}(k)\|_1\}}{E\{\|\mathbf{X}(k)\|_1\}} \\ &= N\sqrt{\frac{2}{\pi}}\sigma_x\frac{E\{\|\mathbf{A}(k)\mathbf{X}(k)\|_1\}}{E\{\|\mathbf{X}(k)\|_1\}} \\ &= M\sqrt{\frac{2}{\pi}}\sigma_x\zeta_1 \end{aligned} \tag{4.6}$$

where

$$\zeta_1 = \frac{NE\{\|\mathbf{A}(k)\mathbf{X}(k)\|_1\}}{ME\{\|\mathbf{X}(k)\|_1\}} \tag{4.7}$$

It should be noted that the value of ζ_1 is dependent on the coefficient selection process employed for updating the subset of the filter coefficients. For the sequential PU-NSLMS algorithm, $[E\{\|\mathbf{A}(k)\mathbf{X}(k)\|_1\}/E\{\|\mathbf{X}(k)\|_1\}]$ can be shown to be (M/N). Hence, $\zeta_1 = 1$ for the sequential PU-NSLMS algorithm.

From (3.1), (4.1), (4.2), (4.4), (4.5) and (4.6), we have

$$\begin{aligned} E\left\{\widetilde{\mathbf{W}}(k+1)\right\} &= E\left\{\widetilde{\mathbf{W}}(k)\right\} \\ &\quad -\widehat{\mu}E\left\{sign\{|\mathbf{A}(k)|\mathbf{X}(k)\}\left[\mathbf{X}^T(k)\widetilde{\mathbf{W}}(k)+n(k)\right]\right\} \end{aligned} \tag{4.8}$$

where

$$\widehat{\mu} = \frac{\mu}{M\sqrt{\frac{2}{\pi}}\sigma_x\zeta_1} \tag{4.9}$$

Solving (4.8) based on the independence assumption yields

$$E\left\{\widetilde{\mathbf{W}}(k+1)\right\} = \left[I-\widehat{\mu}\frac{M}{N}\widetilde{\mathbf{R}}_{xx}(k)\right]E\left\{\widetilde{\mathbf{W}}(k)\right\} \tag{4.10}$$

where

$$\begin{aligned} \widetilde{\mathbf{R}}_{xx}(k) &= diag\{r_1,\ldots,r_N\}, \\ r_i &= \sqrt{\frac{2}{\pi}}\sigma_x\zeta_1,\ i\in(1,\ldots,N) \end{aligned} \tag{4.11}$$

From (4.10), we can see that for a selected value of the step-size μ, the evolution equation for the mean of the FCM converges to zero.

Next, an equation characterizing the mean-square behavior of the sequential PU-NSLMS algorithm is derived. The FCM covariance matrix $E\{\mathbf{V}(k)\}$ is defined as

$$E\{\mathbf{V}(k)\} = E\left\{\widetilde{\mathbf{W}}(k)\widetilde{\mathbf{W}}^T(k)\right\} \tag{4.12}$$

Using the independence assumption, the evolution equation for the covariance matrix of the FCM is given by

$$
\begin{aligned}
E\{\mathbf{V}(k+1)\} &= E\{\mathbf{V}(k)\} \\
&\quad -\widehat{\mu}E\left\{sign\{\mathbf{A}(k)\mathbf{X}(k)\}\mathbf{X}^T(k)\right\}E\{\mathbf{V}(k)\} \\
&\quad -\widehat{\mu}E\{\mathbf{V}(k)\}E\left\{sign\{\mathbf{A}(k)\mathbf{X}(k)\}\mathbf{X}^T(k)\right\} \\
&\quad +\widehat{\mu}^2E\left\{sign\{\mathbf{A}(k)\mathbf{X}(k)\}\mathbf{X}^T(k)\mathbf{V}(k)\mathbf{X}(k)sign\{\mathbf{X}^T(k)\}\right\} \\
&\quad +\widehat{\mu}^2E\left\{sign\{\mathbf{A}(k)\mathbf{X}(k)\}sign\{\mathbf{X}^T(k)\mathbf{A}(k)\}\right\}\sigma_n^2
\end{aligned}
\tag{4.13}
$$

Taking the *trace* of the matrices on both sides of (4.13) yields the evolution equation for the mean-square of the FCM (for details, see Appendix A)

$$
\sigma_{\widetilde{w}}^2(k+1) = \left[1-2\widehat{\mu}\left(\frac{M}{N}\sqrt{\frac{2}{\pi}}\zeta_1\sigma_x\right)+\widehat{\mu}^2\left(\frac{M}{N}(N+\zeta_2-1)\sigma_x^2\right)\right]\sigma_{\widetilde{w}}^2(k)+\widehat{\mu}^2\frac{M}{N}N\sigma_n^2 \tag{4.14}
$$

where

$$
\sigma_{\widetilde{w}}^2(k+1) = trace\left(E\{\mathbf{V}(k+1)\}\right) \tag{4.15}
$$

$$
\zeta_2 = \frac{NE\left\{\left\|\mathbf{A}(k)\mathbf{X}^2(k)\right\|_1\right\}}{ME\{\|\mathbf{X}^2(k)\|_1\}} \tag{4.16}
$$

Similar to ζ_1, ζ_2 is also dependent on the coefficient selection process employed for updating the subset of the filter coefficients. For the sequential PU-NSLMS algorithm, the expression $\left[E\left\{\left\|\mathbf{A}(k)\mathbf{X}^2(k)\right\|_1\right\}/E\left\{\left\|\mathbf{X}^2(k)\right\|_1\right\}\right]$ can be shown to be (M/N). Hence, $\zeta_2 = 1$ for the sequential PU-NSLMS algorithm. From (4.7), (4.9), (4.16) and (4.14), we have a simplified expression of the evolution equation for the mean-square of the FCM:

$$
\sigma_{\widetilde{w}}^2(k+1) = \left[1-2\frac{\mu}{N}+\frac{\mu^2}{M\left(\frac{2}{\pi}\right)}\right]\sigma_{\widetilde{w}}^2(k)+\frac{\mu^2}{M\left(\frac{2}{\pi}\right)}\frac{\sigma_n^2}{\sigma_x^2} \tag{4.17}
$$

Letting $k \to \infty$ in (4.17) gives the steady-state FCM, $\sigma_{\widetilde{w}}^2(\infty)$:

$$\sigma_{\tilde{w}}^2(\infty) = \left[\frac{\mu N}{2M\left(\frac{2}{\pi}\right) - \mu N}\right] \frac{\sigma_n^2}{\sigma_x^2} \tag{4.18}$$

Stability of the evolution equation for the mean-square of the FCM is guaranteed, if

$$\left|1 - 2\frac{\mu}{N} + \frac{\mu^2}{M\left(\frac{2}{\pi}\right)}\right| < 1 \tag{4.19}$$

that is,

$$0 < \mu < \frac{2M}{N}\left(\frac{2}{\pi}\right) \tag{4.20}$$

4.3 Analysis of the *M*-Max PU-NSLMS Algorithm

The only difference between the architectures of the sequential PU-NSLMS and *M*-Max PU-NSLMS algorithms is the coefficient selection process employed to determine the subset of the filter coefficients to be updated at a particular time step. Therefore, instead of carrying out a separate statistical analysis of the *M*-Max PU-NSLMS algorithm, one can appropriately modify the results of the sequential PU-NSLMS algorithm to obtain the corresponding results of the *M*-Max PU-NSLMS algorithm.

While carrying out the statistical analysis of the sequential PU-NSLMS algorithm, only the values of $\zeta_1$and ζ_2 are dependent on the coefficient selection process employed. Thus, modifying the values of these terms accordingly would provide us with the statistical results of the *M*-Max PU-NSLMS algorithm.

4.3.1 Estimating Sum of *M*-Max

It is difficult to obtain the values of $\zeta_1 = \frac{NE\{\|\mathbf{A}(k)\mathbf{X}(k)\|_1\}}{ME\{\|\mathbf{X}(k)\|_1\}}$ and $\zeta_2 = \frac{NE\{\|\mathbf{A}(k)\mathbf{X}^2(k)\|_1\}}{ME\{\|\mathbf{X}^2(k)\|_1\}}$ in the case of SCU, since obtaining $E\{\|\mathbf{A}(k)\mathbf{X}(k)\|_1\}$ and $E\{\|\mathbf{A}(k)\mathbf{X}^2(k)\|_1\}$ requires *ordered statistics* [32], [33].

The problem of ordered statistics has been well discussed in [34]-[36], where recursion formulae and tables have been generated for the expected values and moments of the ordered statistics. Werner, de Campos and Diniz [3] use numerical methods for obtaining $E\left\{\left\|\mathbf{A}(k)\mathbf{X}^2(k)\right\|_1\right\}$. In this section, instead of using numerical methods, we derive approximate closed form expressions for ζ_1 and ζ_2. Using simulations, it is shown that the derived expressions provide fairly accurate results for sufficiently large values of M and N. In order to derive the expressions, we introduce the following variable

$\underline{\mathbf{Z}} = [z_1\ z_2\ \ldots\ z_N]_{N\times 1}$: Contains all the elements of the vector $|\mathbf{X}(k)|$ such that

$$z_1 \leq z_2 \leq \ldots \leq z_j \leq \ldots \leq z_N$$

Using ordered statistics, the probability density function (PDF) $f_j(z)$ of the j^{th}element of $\underline{\mathbf{Z}}$ is given by [32]

$$f_j(z) = \frac{N!}{(j-1)!\,(N-j)!} F_X^{j-1}(z)\left[1-F_X(z)\right]^{N-j} f_X(z) \tag{4.21}$$

where $f_X(.)$ and $F_X(.)$ are respectively the PDF and the cumulative distribution function of the elements of $|\mathbf{X}(k)|$. It can be shown that

$$E\left\{\left\|\mathbf{A}(k)\mathbf{X}(k)\right\|_1\right\} = \int_{-\infty}^{\infty} z\, g_X(z)\, dz \tag{4.22}$$

$$E\left\{\left\|\mathbf{A}(k)\mathbf{X}^2(k)\right\|_1\right\} = \int_{-\infty}^{\infty} z^2\, g_X(z)\, dz \tag{4.23}$$

where

$$\begin{aligned} g_X(z) &= \sum_{t=N-M+1}^{N} f_t(z) \\ &= \sum_{t=N-M+1}^{N} \frac{N!}{(t-1)!\,(N-t)!} F_X^{t-1}(z)\left[1-F_X(z)\right]^{N-t} f_X(z) \end{aligned} \tag{4.24}$$

The above equation can also be written as

$$g_X(z) = N f_X(z)\,\xi \tag{4.25}$$

where

$$\xi = \left[\sum_{t=N-M+1}^{N} \frac{(N-1)!}{(t-1)!\,(N-t)!} p^{t-1} q^{N-t}\right] \tag{4.26}$$

$$p = F_X(z) \tag{4.27}$$

$$q = [1 - F_X(z)] \tag{4.28}$$

Using the De Moivre-Laplace theorem [32] , ξ in (4.26) is approximated as

$$\xi \approx \int_{\left(1-\frac{M-1}{N-1}\right)}^{1} \frac{1}{\sqrt{2\pi}\sigma_m} exp\left(-\frac{(m-p)^2}{2\sigma_m^2}\right) dm \tag{4.29}$$

where

$$m = \frac{l}{N-1} \tag{4.30}$$

$$\sigma_m^2 = \frac{pq}{N-1} \tag{4.31}$$

For large values of N,

$$\frac{1}{\sqrt{2\pi}\sigma_m} exp\left(-\frac{(m-p)^2}{2\sigma_m^2}\right) \approx \delta(m-p) \tag{4.32}$$

where $\delta(m-p)$ is a Dirac delta function with unit area under the curve. From (4.27), (4.29) and (4.32), we obtain

$$\xi \approx \int_{\left(1-\frac{M-1}{N-1}\right)}^{1} \delta(m - F_X(z))\, dm \tag{4.33}$$

The solution of the above equation is given by

$$\xi \approx \begin{cases} 0 & z < F_X^{-1}\left(1-\frac{M-1}{N-1}\right) \\ 1 & z > F_X^{-1}\left(1-\frac{M-1}{N-1}\right) \end{cases} \tag{4.34}$$

where $F_X^{-1}(.)$ being the inverse of $F_X(.)$. The above equation can also be written as

$$\xi \approx U(z-z_0) \tag{4.35}$$

$$z_0 = F_X^{-1}\left(1-\frac{M-1}{N-1}\right) \tag{4.36}$$

where $U(.)$ is a unit step function. From (4.25) and (4.35), we have

$$g_X(z) \approx N f_X(z) U(z-z_0) \tag{4.37}$$

Substituting (4.37) in (4.22) and (4.23) gives

$$E\{\|\mathbf{A}(k)\mathbf{X}(k)\|_1\} \approx N\int_{-\infty}^{\infty} z f_X(z) U(z-z_0)\, dz \tag{4.38}$$

$$E\{\|\mathbf{A}(k)\mathbf{X}^2(k)\|_1\} \approx N\int_{-\infty}^{\infty} z^2 f_X(z) U(z-z_0)\, dz \tag{4.39}$$

If elements of $\mathbf{X}(k)$ are generated from a stationary, zero-mean Gaussian process with a variance σ_x^2, then $E\{\|X(k)\|_1\} = N\sqrt{\frac{2}{\pi}}\sigma_x$ and $E\{\|X^2(k)\|_1\} = N\sigma_x^2$. Using (4.38), (4.39), one can obtain the the approximate expressions for ζ_1 and ζ_2 as

$$\zeta_1 = \frac{NE\{\|A(k)X(k)\|_1\}}{ME\{\|X(k)\|_1\}} \approx \frac{exp\left(\frac{-z_0^2}{2\sigma_x^2}\right)}{\left(\frac{M}{N}\right)} \tag{4.40}$$

and

$$\begin{aligned}\zeta_2 &= \frac{NE\{\|A(k)X^2(k)\|_1\}}{ME\{\|X^2(k)\|_1\}} \\ &\approx \sqrt{\frac{2}{\pi}}\frac{\left[\frac{z_0}{\sigma_x}exp\left(\frac{-z_0^2}{2\sigma_x^2}\right)+\sqrt{2\pi}\frac{(M-1)}{2(N-1)}\right]}{\left(\frac{M}{N}\right)}\end{aligned} \tag{4.41}$$

where z_0 satisfies (4.36) and $F_X^{-1}(.)$ is the inverse of cumulative distribution function of the random variable generated from stationary, zero-mean, white Gaussian process with a variance σ_x^2. Table 4.1 compares the values of ζ_1 and ζ_2 obtained experimentally using MATLAB and using the closed form expressions in (4.40) and (4.41), respectively, for the case of variance $\sigma_x^2 = 1$. It is seen from Table 4.1 that the theoretical and experimental estimates of ζ_1 and ζ_2 are close, and that the estimation error decreases with increasing M.

TABLE 4.1

COMPARISON OF THE EXPERIMENTAL ESTIMATES OF ζ_1 AND ζ_2 WITH THEIR THEORETICAL ESTIMATES IN (4.40) AND (4.41), RESPECTIVELY FOR THE CASE OF $N = 256$ AND DIFFERENT VALUES OF M

M	ζ_1(Th)	ζ_1(Exp)	% error of ζ_1	ζ_2(Th)	ζ_2(Exp)	% error of ζ_2
16	2.6844	2.8075	-4.5863	4.9882	5.1465	-3.1728
32	2.4132	2.4614	-1.9966	3.9536	4.007	-1.1913
64	2.0470	2.0615	-0.7079	2.8788	2.8883	-0.3304
96	1.7919	1.7977	-0.3237	2.2683	2.2704	-0.0909
128	1.5898	1.5920	-0.1410	1.8556	1.8557	-0.0068
Th: Theoretical results, Exp: Experimental results, and % error = $\frac{Th-Exp}{Th} \times 100$						

4.3.2 Analysis

The evolution equation for the mean of the FCM of the M-Max PU-NSLMS algorithm is given by

$$E\left\{\widetilde{\mathbf{W}}(k+1)\right\} = \left[I - \widehat{\mu}\frac{M}{N}\widetilde{\mathbf{R}}_{xx}(k)\right] E\left\{\widetilde{\mathbf{W}}(k)\right\} \tag{4.42}$$

where

$$\begin{aligned} \widetilde{\mathbf{R}}_{xx}(k) &= diag\{r_1,\ldots,r_N\}, && (4.43)\\ r_i &= \sqrt{\frac{2}{\pi}}\sigma_x\zeta_1,\ i\in(1,\ldots,N) \\ \widehat{\mu} &= \frac{\mu}{M\sqrt{\frac{2}{\pi}}\sigma_x\zeta_1} && (4.44) \end{aligned}$$

The approximate closed-form expression for estimating the value of ζ_1 is given for the M-Max PU-NSLMS algorithm in (4.40) (Appendix B). From (4.42), we see that for a selected value of the step-size μ, the evolution equation for the mean of the FCM converges to zero.

Also, the evolution equation for the mean-square of the FCM of the M-Max PU-NSLMS algorithm is given by

$$\sigma^2_{\widetilde{w}}(k+1) = \left[1 - 2\widehat{\mu}\left(\frac{M}{N}\sqrt{\frac{2}{\pi}}\zeta_1\sigma_x\right) + \widehat{\mu}^2\left(\frac{M}{N}(N+\zeta_2-1)\sigma_x^2\right)\right]\sigma^2_{\widetilde{w}}(k) + \widehat{\mu}^2\frac{M}{N}N\sigma_n^2 \tag{4.45}$$

The approximate closed-form expression for estimating the value of ζ_2 is given by (4.41) (Appendix B). From (4.44) and (4.45), we have

$$\sigma^2_{\widetilde{w}}(k+1) = \left[1 - 2\frac{\mu}{N} + \frac{\mu^2(N+\zeta_2-1)}{MN\left(\frac{2}{\pi}\zeta_1^2\right)}\right]\sigma^2_{\widetilde{w}}(k) + \frac{\mu^2}{M\left(\frac{2}{\pi}\zeta_1^2\right)}\frac{\sigma_n^2}{\sigma_x^2} \tag{4.46}$$

Letting $k \to \infty$ in (4.46) gives, $\sigma^2_{\widetilde{w}}(\infty)$, the steady-state FCM of the M-Max PU-NSLMS

algorithm:

$$\sigma_{\tilde{w}}^2(\infty) = \left[\frac{\mu N}{2M\left(\frac{2}{\pi}\zeta_1^2\right) - \mu(N+\zeta_2-1)}\right]\frac{\sigma_n^2}{\sigma_x^2} \tag{4.47}$$

Stability of the evolution equation for the mean-square of the FCM of the M-Max PU-NSLMS algorithm is guaranteed, if

$$\left|1 - 2\frac{\mu}{N} + \frac{\mu^2(N+\zeta_2-1)}{MN\left(\frac{2}{\pi}\zeta_1^2\right)}\right| < 1 \tag{4.48}$$

that is,

$$0 < \mu < \frac{2M}{(N+\zeta_2-1)}\left(\frac{2}{\pi}\zeta_1^2\right) \tag{4.49}$$

For large values of N, (4.49) can be approximated as

$$0 < \mu < \frac{2M}{N}\left(\frac{2}{\pi}\zeta_1^2\right) \tag{4.50}$$

4.4 Convergence Rate

The general evolution equation for the mean-square of the FCM of the PU-NSLMS algorithm can be written as

$$\sigma_{\tilde{w}}^2(k+1) = [1-\tau]\,\sigma_{\tilde{w}}^2(k) + \widehat{\mu}^2\frac{M}{N}N\sigma_n^2 \tag{4.51}$$

where

$$\tau = \frac{M}{N}\left[2\widehat{\mu}\left(\sqrt{\frac{2}{\pi}}\zeta_1\sigma_x\right) - \widehat{\mu}^2(N+\zeta_2-1)\,\sigma_x^2\right] \tag{4.52}$$

$$\widehat{\mu} = \frac{\mu}{M\sqrt{\frac{2}{\pi}}\sigma_x\zeta_1} \tag{4.53}$$

It should be noted that the above equations are independent of the coefficient selection process employed, and only the values of ζ_1 and ζ_2 change depending on the coefficient selection process employed. The evolution equation for the mean-square of the FCM is convergent for $|1-\tau| < 1$, that is, $0 < \tau < 2$. For $\tau > 1$, neither the convergence rate of the evolution equation for the mean-square of the FCM increases nor the steady-state FCM decreases. Hence, only $0 < \tau \leq 1$ is of practical interest. In this range, the convergence rate of the evolution equation for the mean-square of the FCM improves with increasing value of τ.

From (4.52) and (4.53), we have

$$\tau = \frac{M}{N}\left[\frac{2}{M}\mu - \frac{\mu^2}{M^2\left(\frac{2}{\pi}\right)}\frac{(N+\zeta_2-1)}{\zeta_1^2}\right] \tag{4.54}$$

For large values of N, $N+\zeta_2-1 \approx N$. Therefore, (4.54) can be approximated as

$$\tau \approx \frac{M}{N}\left[\frac{2}{M}\mu - \frac{N\mu^2}{M^2\left(\frac{2}{\pi}\right)\zeta_1^2}\right] \tag{4.55}$$

In the above equation, τ and consequently, the convergence rate of the evolution equation for the mean-square of the FCM increase with increasing values of ζ_1. Thus, the coefficient selection process having the largest value of ζ_1 will provide the best convergence rate for the PU-NSLMS algorithm.

Since at any instant k, the coefficient selection process of the M-Max PU-NSLMS algorithm maximizes $\|\mathbf{A}(k)\mathbf{X}(k)\|_1$, the value of $\zeta_1 = [E\{\|\mathbf{A}(k)\mathbf{X}(k)\|_1\}/E\{\|\mathbf{X}(k)\|_1\}]$ is maximum for the M-Max PU-NSLMS algorithm. Thus, the convergence rate of the M-Max PU-NSLMS algorithm is better than that of the PU-NSLMS algorithm employing any other coefficient selection process, and M-Max PU-NSLMS algorithm can achieve a convergence rate that is closest to the full update algorithm.

4.5 Simulations

In this section, we study the performance of the proposed PU-NSLMS algorithms. In order to do so, a system-identification problem is considered. Fig. 4.1 shows the impulse response of an unknown system characterized by an FIR filter with 256 taps. The input signal is generated from a stationary, zero-mean, white Gaussian process with a variance $\sigma_x^2 = 1$. The signal to noise ratio (SNR) is 40 dB. The adaptive filter used to model the unknown system is also assumed to have a filter tap length $N = 256$. The L_2-norm of the FCM, $||\mathbf{W}(k) - \mathbf{W}||_2^2$, is used as the performance index.

The stability bounds on the step-size of the sequential PU-NSLMS and M-Max PU-NSLMS algorithms are given by (4.20) and (4.49), respectively. Even though the actual bounds on the step-size of these algorithms are somewhat different from these expressions, they can still be used as guidelines towards a good choice of the step-size. Fig. 4.2 shows the theoretical upper bounds on the step-size of the sequential PU-NSLMS and M-Max PU-NSLMS algorithms for different values of $K = (N/M)$. The upper bound on the step-size of the M-Max PU-NSLMS algorithm is higher than that of the sequential PU-NSLMS algorithm. Therefore, by opting for a larger value of the step-size, the M-Max PU-NSLMS algorithm can achieve a faster convergence rate than that of the sequential PU-NSLMS algorithm.

Figs. 4.3 and 4.4 show the FCM curves of the sequential PU-NSLMS and M-Max PU-NSLMS algorithms for $M = 64$ and different values of μ. The FCM curves were obtained through averaging 50 trials. They also show the corresponding theoretical FCM curves obtained from (4.17) and (4.45), respectively. Figs. 4.5 and 4.6 show the steady-state FCM obtained from the simulations as well as from the theoretical analyses of the sequential PU-NSLMS and M-Max PU-NSLMS algorithms given by (4.18) and (4.47), respectively. Figs. 4.3, 4.4, 4.5 and 4.6 show an excellent agreement between the theoretical and experimental results.

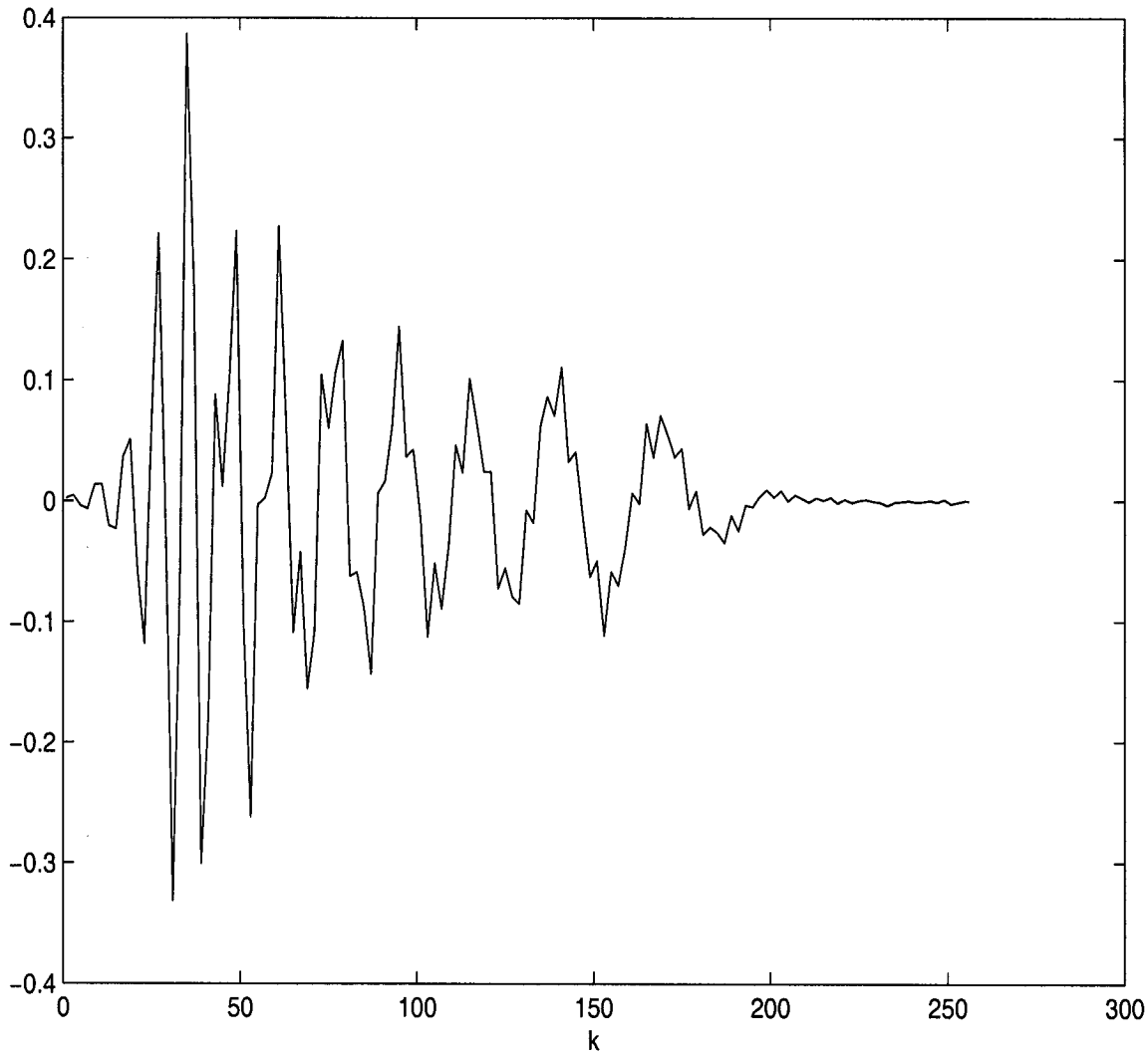

Figure 4.1. Impulse response of an unknown system.

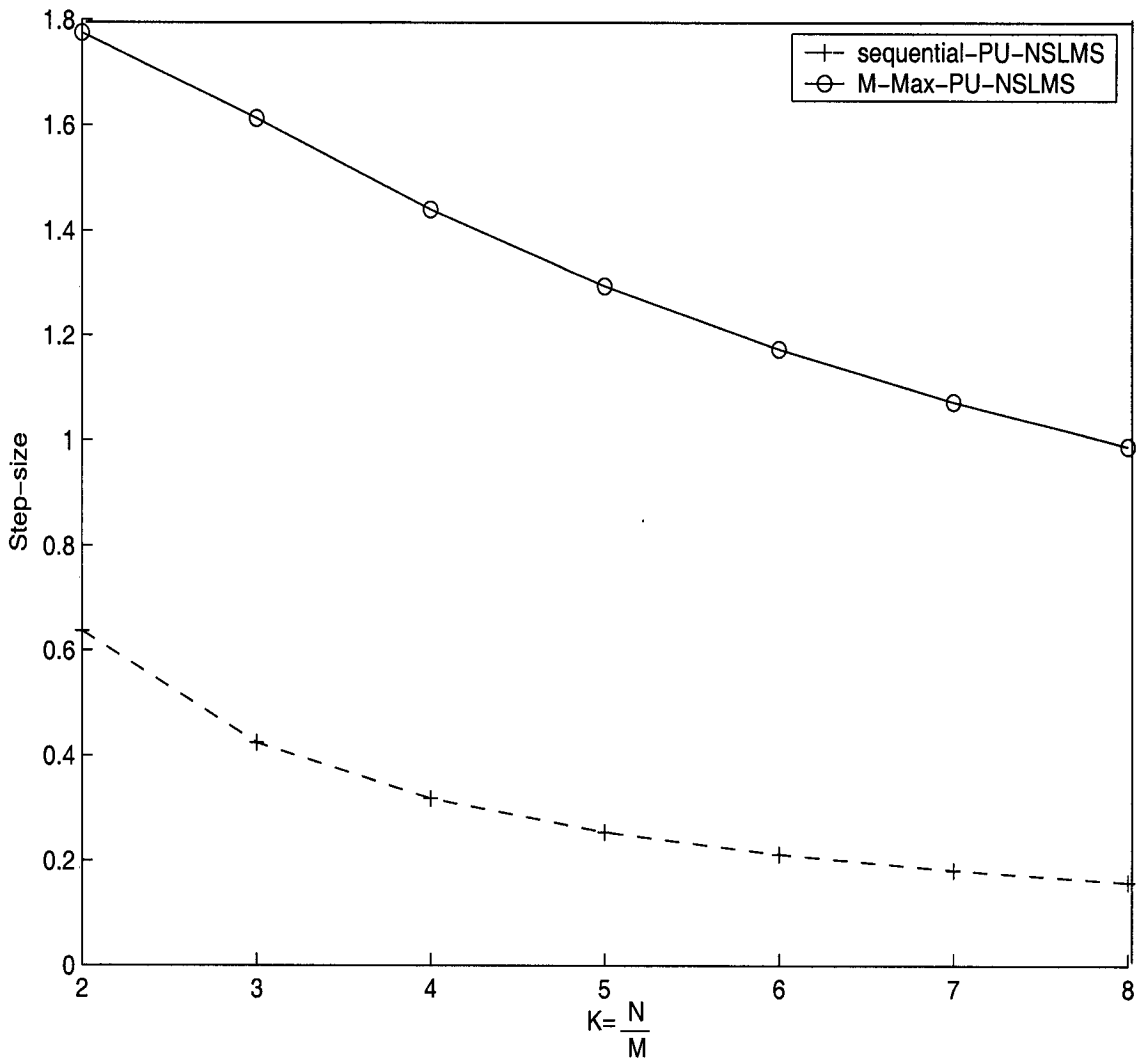

Figure 4.2. Upper bounds on the step-size of the sequential PU-NSLMS and M-Max PU-NSLMS algorithms for the case of $K = \frac{N}{M} = 2, 3, \ldots, 8$.

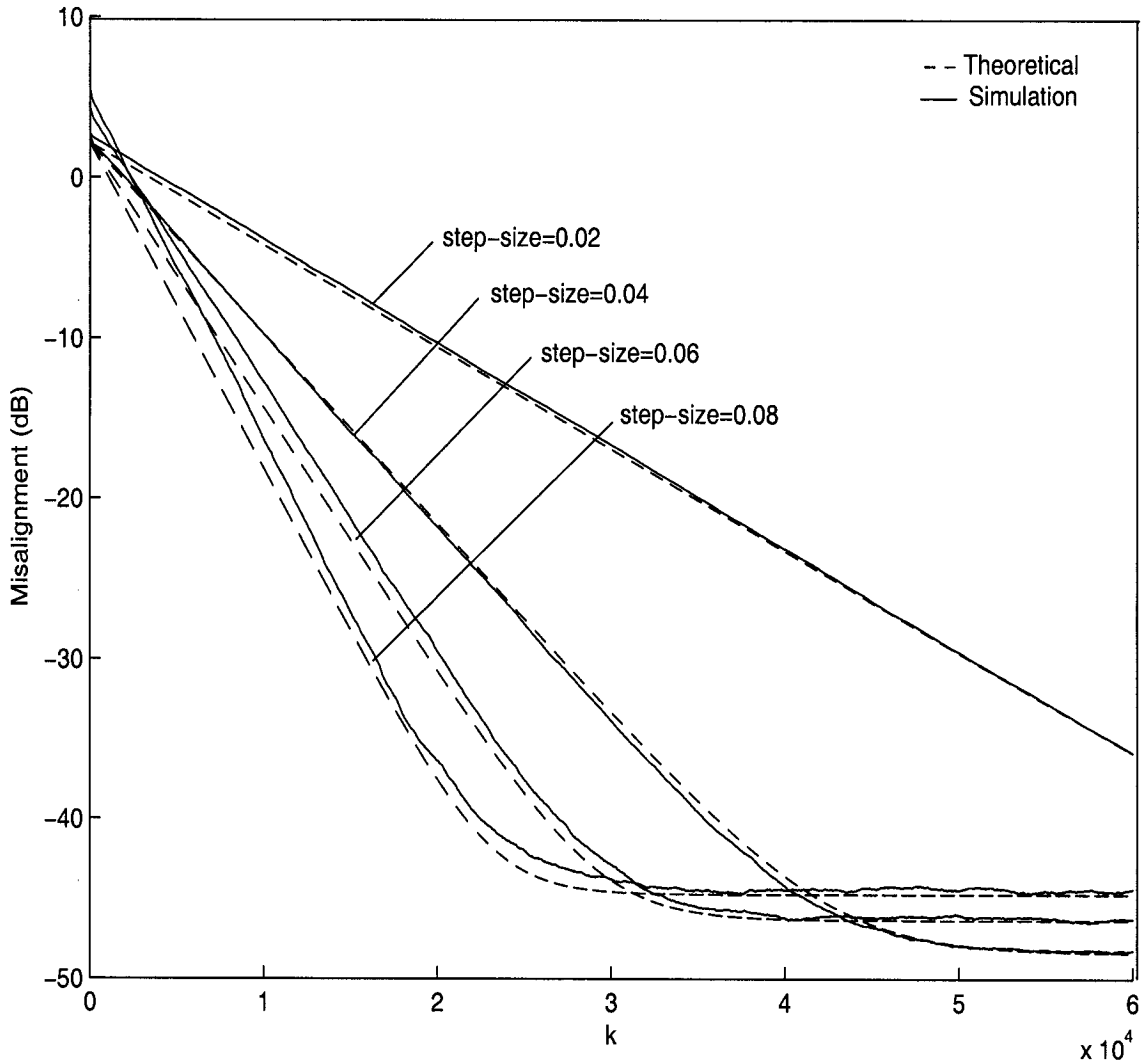

Figure 4.3. Simulated and theoretical FCM curves of the sequential PU-NSLMS algorithm for the case of $N = 256$, $M = 64$ and $\mu = 0.02, 0.04, 0.06, 0.08$.

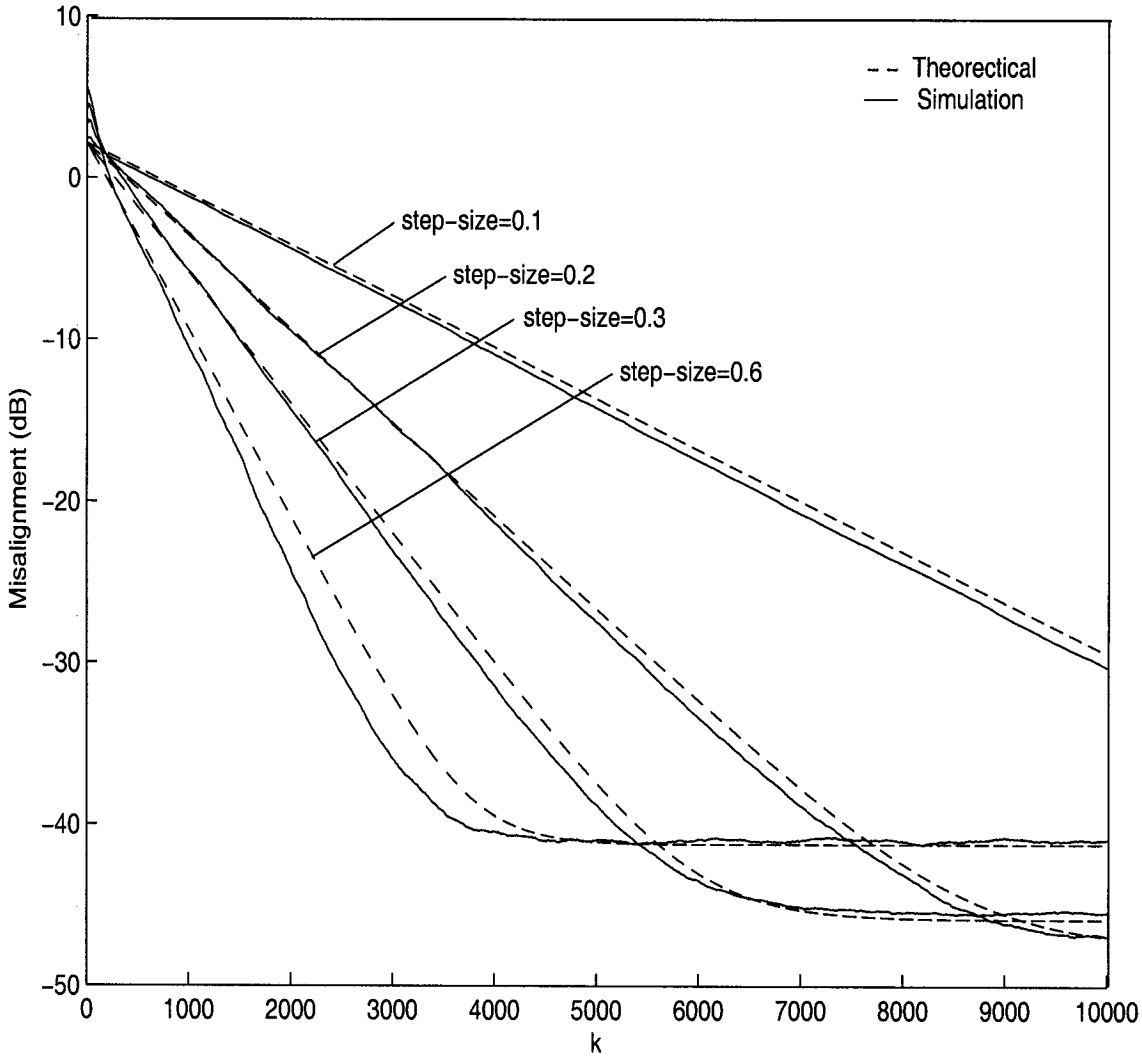

Figure 4.4. Simulated and theoretical FCM curves of the M-Max PU-NSLMS algorithm for the case of $N = 256$, $M = 64$ and $\mu = 0.1, 0.2, 0.3, 0.6$.

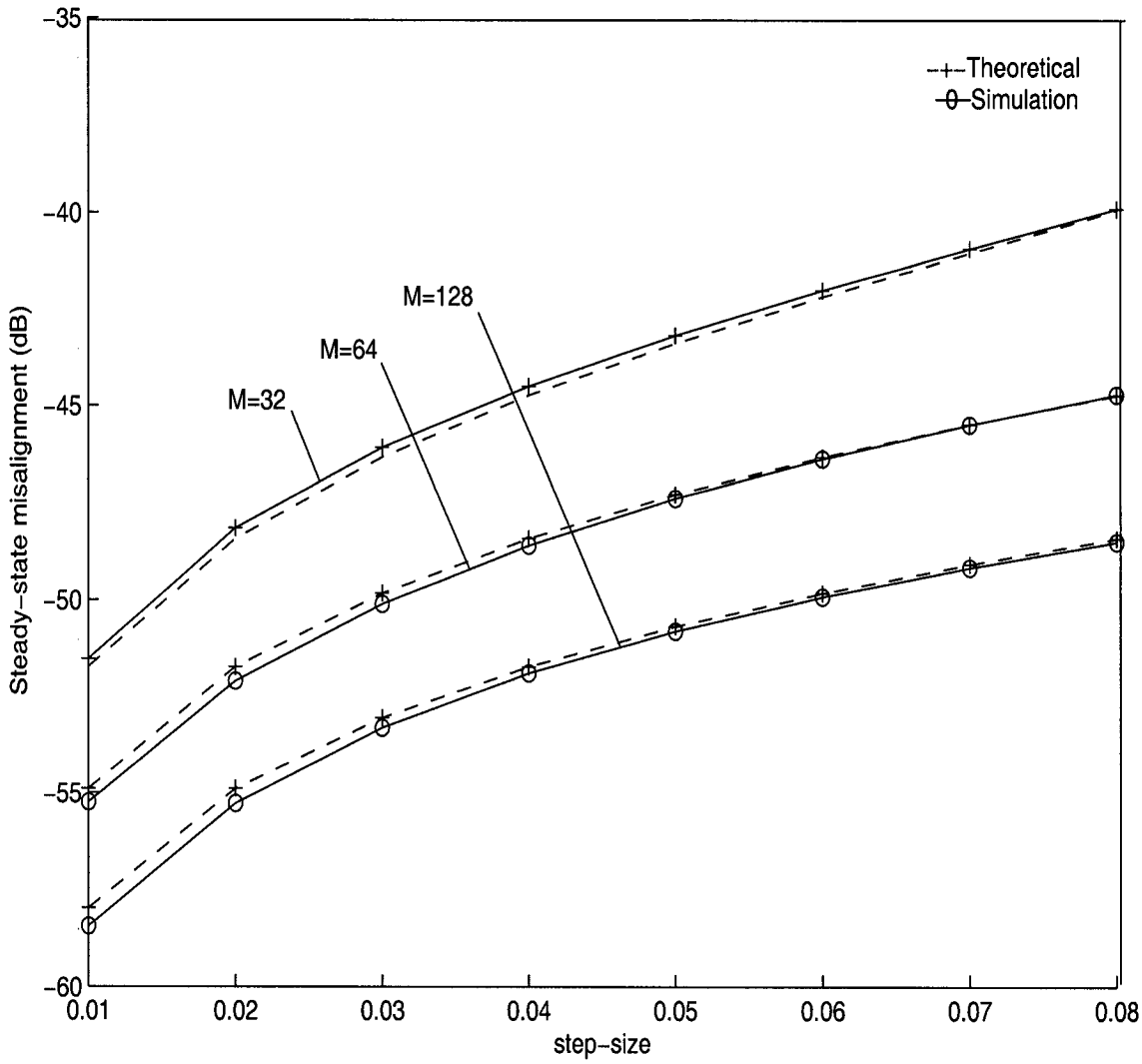

Figure 4.5. Steady-state FCM of the sequential PU-NSLMS algorithm for the case of $N = 256, M = 32, 64, 128$ and $\mu = 0.01, 0.02, \ldots, 0.08$.

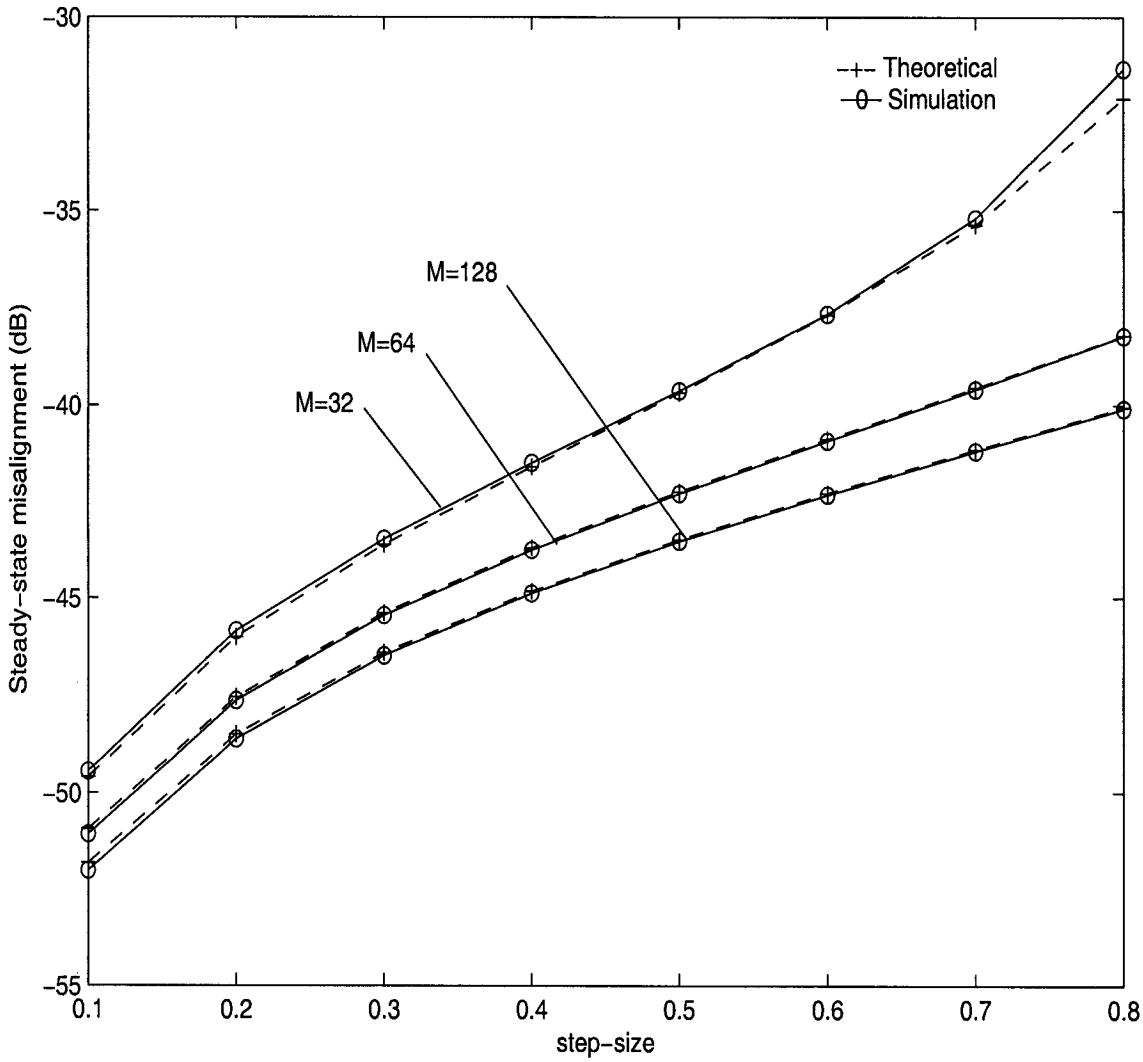

Figure 4.6. Steady-state FCM of the M-Max PU-NSLMS algorithm for the case of $N = 256$, $M = 32, 64, 128$ and $\mu = 0.1, 0.2, \ldots, 0.8$.

Fig. 4.7 shows the FCM curves of the PU-NSLMS algorithm ($M = N$), and for the sequential PU-NSLMS and M-Max PU-NSLMS algorithms for the case of $M = 64$. The step-size for all the algorithms is chosen such that the algorithms achieve approximately the same steady-state FCM. It is seen from Fig. 4.7 that the convergence rate of the sequential PU-NSLMS algorithm is very much lower than that of the M-Max PU-NSLMS algorithm. The M-Max PU-NSLMS algorithm shows only a small degradation in the convergence rate compared to that of the full update algorithm, and has a convergence rate comparable to that of the SPU-NLMS algorithm. The results of Fig. 4.7 reaffirm that the M-Max PU-NSLMS algorithm can achieve a convergence rate that is closest to that of the full update algorithm.

4.6 Summary

In this chapter, we have carried out the statistical analyses of the sequential and M-Max PU-NSLMS algorithms. Evolution equations for the mean and mean-square of the filter coefficient misalignment, and stability bounds on the step-size of the two algorithms have been obtained. It has been shown that the M-Max PU-NSLMS algorithm has a convergence rate that is closest to that of the full update algorithm. The theoretical results obtained from the statistical analyses of the two algorithms have been validated using simulations.

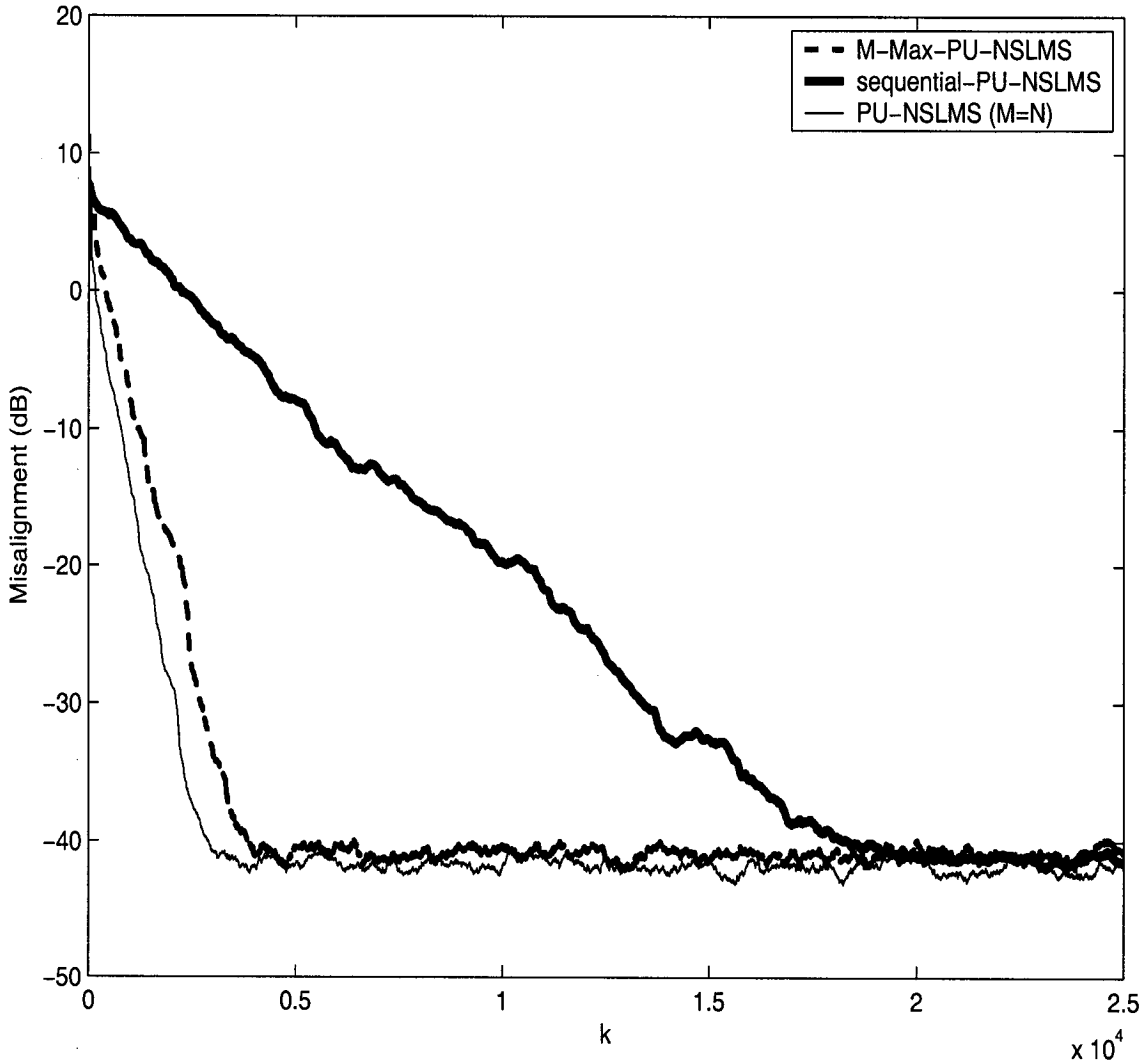

Figure 4.7 Convergence rates of the PU-NSLMS (M=N), sequential PU-NSLMS, and M-Max PU-NSLMS algorithms for the case of N=256 and M=64.

Chapter 5

Proportionate Algorithms

5.1 Introduction

With its simple architecture and ability to give robust performance, the NLMS algorithm is one of the most popular and widely-used adaptive filter algorithms. However, its convergence rate is inversely proportional to its filter tap length, thus making the rate undesirably slow for applications requiring a long filter tap. One approach for increasing the rate of the adaptive filter algorithm is to incorporate *a priori* information about the unknown system into its design, as in the case of adaptive filter algorithms having nonuniform gain distribution [37]-[47]. An algorithm having a nonuniform gain distribution allocates more adaptation energy to the filter taps with larger FCM. Thus, by making the filter coefficients with larger FCM to converge faster, such an algorithm can achieve a faster convergence rate compared to the one having a uniform gain distribution (for example, the NLMS algorithm).

In this chapter, we discuss proportionate algorithms, wherein different adaptation gains are allocated to the various filter coefficients using some criterion. In Section 5.2, a brief introduction to the different proportionate algorithms is provided. The derivation of the recently developed proportionate NLMS (PNLMS) algorithm using a constraint optimization

method [40] is also presented. In Section 5.3, a low-complexity partial-update proportionate algorithm based on the L_∞-norm criterion is proposed.

5.2 PNLMS Algorithm

It is known that in the field of wireless communication, echo cancellation, and underwater acoustic communication, the impulse response of a transmission channel is often sparse, that is, the finite-impulse-response (FIR) filter representing the transmission channel has only a few large taps . Martin *et al.* [44], [45] have shown that one can achieve a substantial improvement in the performance of an adaptive filter, by incorporating into the design of the filter, *a priori* information such as the impulse response of the unknown system. Based on this fact, various adaptive filter algorithms have been proposed exploiting the "sparseness" of the impulse response of the unknown system. These algorithms allocate more adaptation energy to the filter coefficients having larger FCM, and thus, force them to converge faster than the filter coefficients having relatively smaller FCM. Since the filter coefficients having larger FCM are more important in terms of the convergence rate, an algorithm having a nonuniform gain distribution can achieve a faster convergence rate compared to the one having a uniform gain distribution.

Makino *et al.* [41] have proposed an adaptive filter algorithm for acoustic echo cancellation, wherein different adaptation gains are allocated to the various filter coefficients, thus allowing individual learning rates for the filter taps. This algorithm, referred to as the exponentially weighted step-size (ES) algorithm, has an architecture similar to that of the NLMS algorithm except for the exponential distribution (rather than uniform distribution) of the adaptation energy over the filter taps. The motivation for the ES algorithm is that the envelop of the room impulse response decays exponentially over the filter coefficients, and therefore, by having a gain distribution similar to the envelop of the room impulse response, it can achieve a faster convergence rate compared to that of the NLMS algorithm. It

should be noted that in this algorithm, the nonuniform distribution of the adaptation energy over the filter taps is time-invariant and is set by the user. This is undesirable, since the adaptation gains of the various filter taps cannot be varied in accordance with the changes in the acoustic impulse response of the room.

If the idea of allocating less gain to the filter taps having relatively smaller FCM is stretched to the extreme, then it leads to algorithms that try to determine the flat regions in the impulse response using some criterion, and then allocate zero gains to the filter taps in those regions (for example, [42], [43]). However, the problem with such algorithms is in the determination of the flat regions in the impulse response.

In order to overcome these problems, a new variant of the NLMS algorithm, namely, the PNLMS algorithm [37], [38] has been proposed recently. The novelty of the PNLMS algorithm lies in the fact that the learning rate assigned to each of the filter coefficients is adaptive. This means that the PNLMS algorithm can automatically determine the filter coefficients that are important in terms of the convergence rate and accordingly allocates more adaptation energy to these coefficients. The PNLMS algorithm is employed to identify an unknown system having a sparse impulse response, such as in the case of network echo cancellation. A network echo path is usually 64 ms to 128 ms long. However, the active region of a network echo path, where most part of its energy is concentrated, is only about 4 ms to 8 ms long. The rest of the region in the network echo path is relatively flat and can be taken to have zero value. The PNLMS algorithm exploits the "sparseness" of the network echo path by allocating more adaptation energy to the filter coefficients in the active region compared to those in the relatively flat region of the network echo path. Since the filter coefficients in the active region have larger FCM, thus forcing these coefficients to converge faster compared to those in the relatively flat region, the PNLMS algorithm can achieve a faster convergence rate compared to that of the NLMS algorithm. For the PNLMS algorithm, the learning rate of the individual filter taps are taken to be proportional to the

magnitude of their respective tap-weights in the preceding iteration. Therefore, the allocation of the adaptation energy among the filter taps of the PNLMS algorithm is automatic, and not user controlled.

The PNLMS or any other L_2-norm algorithm having a nonuniform gain distribution can be viewed as the solution of the constraint optimization problem

$$\min_{\mathbf{W}(k+1)} \left\| \mathbf{G}^{\frac{-1}{2}}(k) \left(\mathbf{W}(k+1) - \mathbf{W}(k) \right) \right\|_2^2 \qquad (5.1)$$
$$\text{subject to } \mathbf{W}^T(k+1)\mathbf{X}(k) = y(k)$$

where $\mathbf{G}(k) = diag\{g_1(k), g_2(k), \ldots, g_N(k)\}_{N\times N}$ is the gain distribution matrix. Solving (5.1) in a manner similar to that used in solving (2.3) gives the coefficient-update equation

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \frac{\mu e(k)\mathbf{G}(k)\mathbf{X}(k)}{\mathbf{X}^T(k)\mathbf{G}(k)\mathbf{X}(k) + \delta} \qquad (5.2)$$

where δ is a small positive number added to the denominator to prevent the division by zero. The matrix $\mathbf{G}(k)$ controls the allocation of the adaptation energy over the filter taps. The expression for the diagonal elements of $\mathbf{G}(k)$ in the case of PNLMS algorithm is given by [37]

$$g_i(k) = max\left\{ \rho l'_\infty(k), |w_i(k)| \right\}, i \in (1, \ldots, N) \qquad (5.3)$$

where

$$\begin{aligned} l'_\infty(k) &= max\{l_\infty(k), \delta\} \\ l_\infty(k) &= max\{|w_1(k)|, \ldots, |w_N(k)|\} \end{aligned}$$

and ρ and δ have small positive values.

For an unknown system having a sparse impulse response, the PNLMS algorithm can achieve a faster convergence rate compared to that of the NLMS algorithm. However, since

the PNLMS algorithm needs to compute the individual learning rates of the filter taps, it is computationally more complex than the NLMS algorithm. The high computational complexity of the PNLMS algorithm can impair its use in real-time applications, especially those requiring a long filter tap. The complexity of the PNLMS algorithm can be reduced in a number of ways. These include: (1) reducing the number of comparisons required for computing the individual learning rates of the filter taps [39], (2) reducing the number of filter coefficients updated at each time step [40], and (3) finding the gradient vector in the hyperplane based on the L_∞-norm criterion [47].

5.3 PU-PNSLMS Algorithm

In this section, a low-complexity proportionate adaptive filter algorithm is developed by combining the concepts of partial-updating and proportionate updating based on the L_∞-norm criterion. The proposed algorithm is derived by modifying the error control equations employed for developing the PU-NSLMS algorithm.

For an adaptive filter, the error $e(k)$ is given by

$$e(k) = y(k) - \mathbf{W}^T(k)\mathbf{X}(k) \tag{5.4}$$

This error is minimized by adjusting the filter coefficient vector $\mathbf{W}(k)$ using the vector $\mathbf{A}(k)\mathbf{G}(k)\Delta(k)$ to obtain

$$(1-\mu)e(k) = y(k) - \left[\mathbf{W}^T(k) + \Delta^T(k)\mathbf{G}(k)\mathbf{A}(k)\right]\mathbf{X}(k) \tag{5.5}$$

where $\mathbf{G}(k) = diag\{g_1(k), g_2(k), \ldots, g_N(k)\}$ is the gain distribution matrix, $\mathbf{A}(k)$ is the coefficient selection matrix, $\Delta(k) \in \Re^N$ is a variable vector, and μ is the step-size. From

(5.4) and (5.5), we have

$$\mu e(k) = \Delta^T(k)\,\mathbf{G}(k)\,\mathbf{A}(k)\,\mathbf{X}(k) \tag{5.6}$$

Using the lemma given in Section 2.4.1, the minimum L_∞-norm solution of $\Delta(k)$ is given by

$$\Delta(k) = \frac{\mu e(k)}{\|\mathbf{G}(k)\,\mathbf{A}(k)\,\mathbf{X}(k)\|_1} sign\{\mathbf{G}(k)\,\mathbf{A}(k)\,\mathbf{X}(k)\} \tag{5.7}$$

The coefficient-update equation of the PU-PNSLMS algorithm is given by

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \frac{\mu e(k)\,|\mathbf{G}(k)\,\mathbf{A}(k)|}{\|\mathbf{G}(k)\,\mathbf{A}(k)\,\mathbf{X}(k)\|_1} sign\{\mathbf{X}(k)\} \tag{5.8}$$

For $M = N$, the PU-PNSLMS algorithm reduces to the proportionate normalized sign LMS (PNSLMS) algorithm [47]

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \frac{\mu e(k)\,|\mathbf{G}(k)|}{\|\mathbf{G}(k)\,\mathbf{X}(k)\|_1} sign\{\mathbf{X}(k)\} \tag{5.9}$$

For $M < N$, there are various ways of updating a subset of the filter coefficients at each time step. One approach is to select the filter coefficients such that the Euclidean norm of the vector $\mathbf{G}^{-1}(k)\,[\mathbf{W}(k+1) - \mathbf{W}(k)]$ is minimum at each time step. In (5.8), for given $e(k)$ and μ, the Euclidean norm of the vector $\mathbf{G}^{-1}(k)\,[\mathbf{W}(k+1) - \mathbf{W}(k)]$ is minimum, when $\|\mathbf{A}(k)\,\mathbf{G}(k)\,\mathbf{X}(k)\|_1$ is maximum. Therefore, the filter coefficients corresponding to the M largest elements of $\|\mathbf{G}(k)\,\mathbf{X}(k)\|_1$ are updated at each time step. The resulting algorithm is referred to as the M-Max PU-PNSLMS algorithm and its coefficient-update equation is given by

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \frac{\mu e(k)\,|\mathbf{G}(k)\,\mathbf{A}(k)|}{\|\mathbf{G}(k)\,\mathbf{A}(k)\,\mathbf{X}(k)\|_1} sign\{\mathbf{X}(k)\} \tag{5.10}$$

where

$$\mathbf{A}(k) = diag\{\bar{a}_1(k), \bar{a}_2(k), \ldots, \bar{a}_N(k)\}$$

$$\overline{a}_i(k) = \begin{cases} 1 & if\, i = arg\, M\, maxima\, of\, |g_i(k)x(k-i+1)| \\ 0 & otherwise \end{cases}, i \in (1,\ldots,N) \quad (5.11)$$

In order to reduce the operations required to compute the individual learning rate of the filter taps, the elements of the gain distribution matrix of the PU-PNSLMS algorithm are defined as

$$g_i(k) = \frac{|w_i(k)|}{\sum_{l=1}^{N}|w_l(k)|+\gamma} + \delta,\ i \in (1,\ldots,N) \quad (5.12)$$

where δ and γ are small positive numbers. The M-Max PU-PNSLMS algorithm requires N+2M additions, 2N+M+2 multiplications, one division and $log_2(N)+2$ comparisons per time step, whereas the PNLMS algorithm requires 3N-2 additions, 4N+4 multiplications and one division.

5.4 Summary

In this chapter, first, a brief introduction to the adaptive filter algorithms having nonuniform gain distributions and motivations for developing such algorithms have been discussed. Next, a low-complexity proportionate partial-update algorithm has been developed by combining the concept of updating a subset of the filter coefficients and the technique of finding the gradient vector in the hyperplane based on the L_∞-norm criterion. Also, a coefficient selection technique for the low-complexity proportionate partial-update algorithm based on minimizing the length of the coefficient-update vector has been proposed. It has been shown that the complexity of the proposed algorithm is less compared to that of the PNLMS algorithm.

Chapter 6

Echo Cancellation

Echo cancellation is one of the most important applications of the adaptive filter algorithms. In this chapter, a brief introduction to the problem of echo cancellation is first given. Then, the performance of the proposed adaptive filter algorithms in the problem of echo cancellation is studied and compared with that of some of the existing ones.

6.1 Echo Canceler

Echo is generated due to the reflection of signals at points along the transmission medium. It remains unnoticed if the time delay of the echo path is small. As the time delay of the echo path increases, say above a few tens of a millisecond, the echo becomes more apparent and annoying. The problem of echo arises in a telecommunication channel due to the impedance mismatch between the hybrid and customer loop, and is referred to as the network echo. Since the development of the first echo canceler by the Bell Labs [48] to alleviate the problem of network echos, a lot of work has been done in this field. An echo canceler generates the replica of the echo path by applying the reference signal to an adaptive filter algorithm. If the transfer function of the adaptive filter is identical to that of

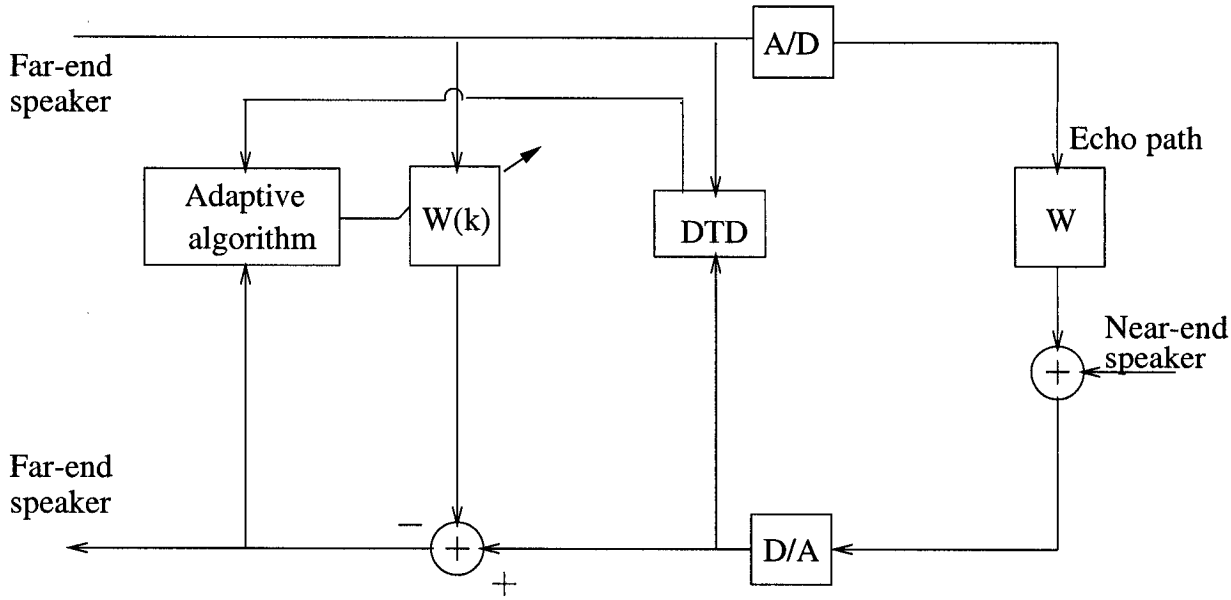

Figure 6.1. Echo-canceler configuration.

the echo path, then it is possible to achieve a total cancellation of the echo. However, in most of the cases, total echo cancellation is impossible and the difference between the echo and the output of the adaptive filter algorithm is fed back to the adaptive filter so as to update the filter coefficients. Fig. 6.1 shows the architecture of an echo canceler. In addition to the adaptive filter algorithm, an echo canceler also consists of the double-talk-detector (DTD), analog to digital (A/D) converter and digital to analog (D/A) converter. During a double talk, the signal generating from the near-end speaker acts as an uncorrelated noise of large amplitude to the adaptive filter algorithm and can cause the adaptive filter algorithm to diverge. In order to overcome this problem during a double talk, filter coefficient updating is stopped whenever the DTD detects a double talk. Adaptive filter is a digital filter and the speech an analog signal. Therefore, in order to overcome the problem of compatibility between the digital filter and the analog signal, A/D and D/A converters are required. The A/D converter transforms the speech signal into a digital signal. This signal is fed into the adaptive filter for processing. The output of the adaptive filter is converted into an analog signal using the D/A converter. During a double talk, if the model of the network echo path

is accurate, then the output of the adaptive filter is equal to the signal generated from the near-end speaker.

The impulse response of an FIR filter is more stable compared to that of an infinite-impulse-response (IIR) filter. Therefore, FIR filters are preferred over IIR filters for the purpose of echo cancellation. Ideally, the tap length of an FIR filter should be equal to the length of the echo path. For most practical cases, it has been been observed that the network echo paths can be modeled accurately by employing FIR filters with a tap length less than or equal to 128 ms.

6.2 Performance

In this section, we discuss the performance of the proposed algorithms, namely, the M-Max PU-NSLMS and M-Max PU-PNSLMS algorithms and compare with that of the SPU-NLMS and PNLMS algorithms, respectively, both from the point of the complexity and the convergence rate.

6.2.1 Complexity

Tables 6.1 and 6.2 show the number of multiplications, additions and comparisons required by the proposed M-Max PU-NSLMS and M-Max PU-PNSLMS algorithms as well as the SPU-NLMS and PNLMS algorithms for different values of M and N. It is observed that the M-Max PU-NSLMS algorithm requires the least number of operations compared to the other three algorithms. The SPU-NLMS algorithm requires more number of multiplications and the same number additions and comparisons compared to the M-Max PU-NSLMS algorithm. It is also observed that the proposed M-Max PU-PNSLMS algorithm requires much less number of operations compared to the PNLMS algorithm.

TABLE 6.1

COMPUTATIONAL COMPLEXITIES OF THE M-MAX PU-NSLMS, SPU-NLMS, M-MAX PU-PNSLMS, AND PNLMS ALGORITHMS FOR THE CASE OF $N = 256$

Algorithm	M	Multiplications (m)	Additions (a)	Comparisons (c)	Complexity (m+a+c)
M-Max PU-NSLMS	32	258	290	10	558
	64	258	322	10	590
	128	258	386	10	654
SPU-NLMS	32	290	290	10	590
	64	322	322	10	654
	128	386	386	10	782
M-Max PU-PNSLMS	32	546	320	10	876
	64	578	384	10	974
	128	642	512	10	1164
PNLMS	256	1024	766	-	1790

TABLE 6.2

COMPUTATIONAL COMPLEXITIES OF THE M-MAX PU-NSLMS, SPU-NLMS, M-MAX PU-PNSLMS, AND PNLMS ALGORITHMS FOR THE CASE OF $N = 512$

Algorithm	M	Multiplications (m)	Additions (a)	Comparisons (c)	Complexity (m+a+c)
M-Max PU-NSLMS	32	514	546	11	1071
	64	514	578	11	1103
	128	514	642	11	1167
	256	514	770	11	1295
SPU-NLMS	32	546	546	11	1103
	64	578	578	11	1167
	128	642	642	11	1295
	256	770	770	11	1551
M-Max PU-PNSLMS	32	1058	576	11	1645
	64	1090	640	11	1742
	128	1154	768	11	1933
	256	1282	1024	11	2317
PNLMS	512	2052	1534	-	3586

6.2.2 Convergence rate

In this section, a study of the convergence rate of the proposed M-Max PU-NSLMS and M-Max PU-PNSLMS algorithms as well as that of the SPU-NLMS and PNLMS algorithms is carried out. Figs 6.2 and 6.3 respectively show the sparse and dispersive echo paths, which are used to study the convergence rate of the algorithms. In order to compare the convergence rate, the step-size for each of the algorithms is chosen so that all the algorithms achieve approximately the same steady-state FCM.

Fig. 6.4 shows the convergence rates of the M-Max PU-NSLMS and SPU-NLMS algorithms for the case of $M = 32$ using the sparse and dispersive echo paths of Figs. 6.2 and 6.3, respectively. Figs. 6.5 and 6.6 show similar convergence rates of the two algorithms for the case of $M = 64$ and 128, respectively. From Figs. 6.4, 6.5 and 6.6, it is observed that the convergence rate of the M-Max PU-NSLMS algorithm is comparable to that of the SPU-NLMS algorithm both for the sparse and dispersive echo paths. Further, it is observed that the convergence rates of the two algorithms are independent of the nature of the echo path.

Fig. 6.7 shows the convergence rates of the PNLMS algorithm as well as that of the proposed M-Max PU-PNSLMS algorithm for the case of $M = 32$, 64 and 128 using the sparse echo of Fig. 6.2. It is observed that the convergence rate of the M-Max PU-PNSLMS algorithm ($M = 128$) is comparable to that of the PNLMS algorithm. Further, the M-Max PU-PNSLMS algorithm shows a minimal degradation in the convergence rate as the number of filter coefficients updated at each time step decreases.

Fig. 6.8 shows the convergence rates of the M-Max PU-PNSLMS and M-Max PU-NSLMS algorithms for the case of $M = 32$ using the sparse and dispersive echo paths of Fig 6.2 and Fig 6.3, respectively. Figs. 6.9 and 6.10 show similar convergence rates of the two algorithms for the case of $M = 64$ and 128, respectively. It is observed that in the case of a sparse echo path, the M-Max PU-PNSLMS algorithm has a faster convergence rate

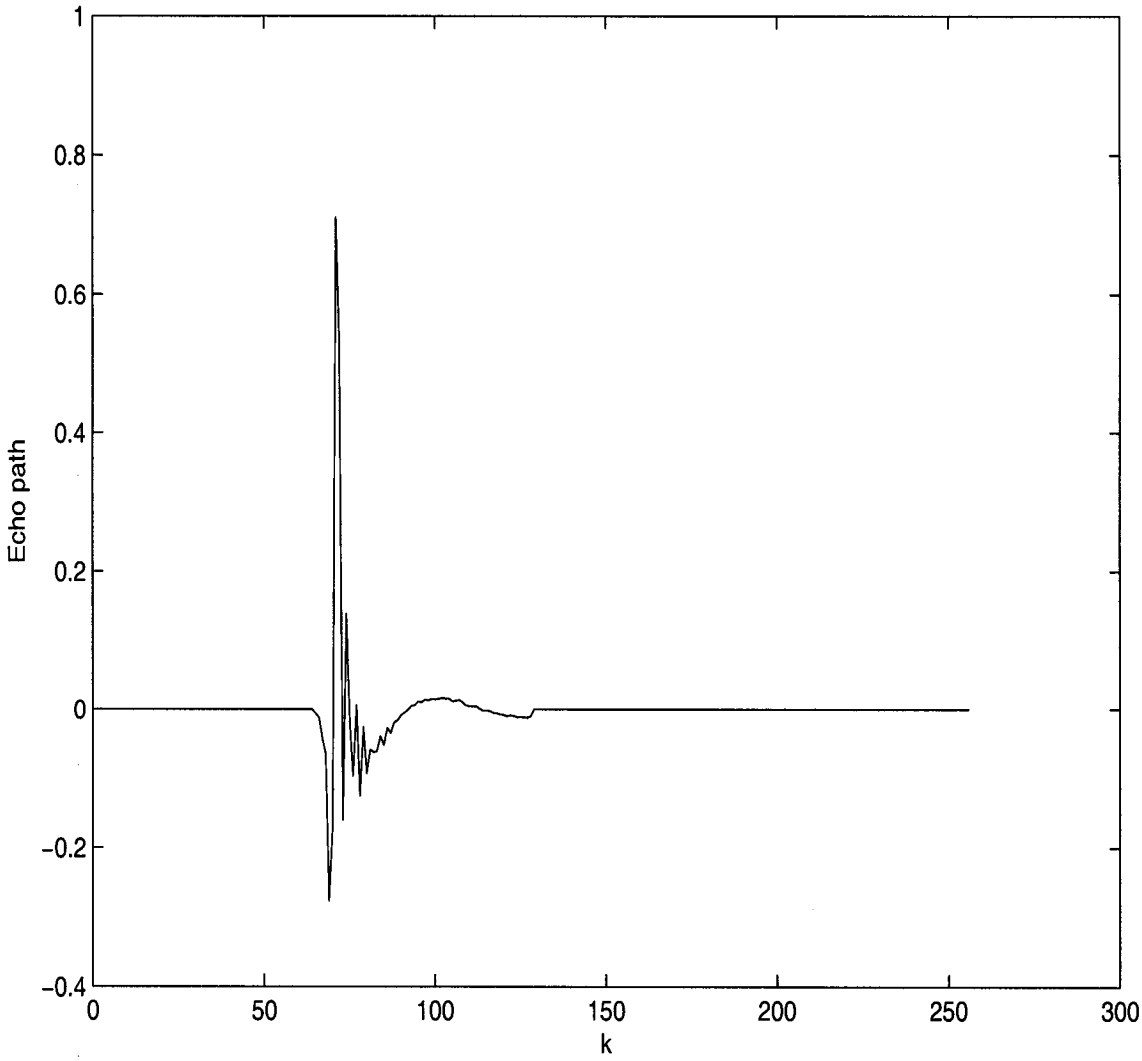

Figure 6.2. Sparse echo path.

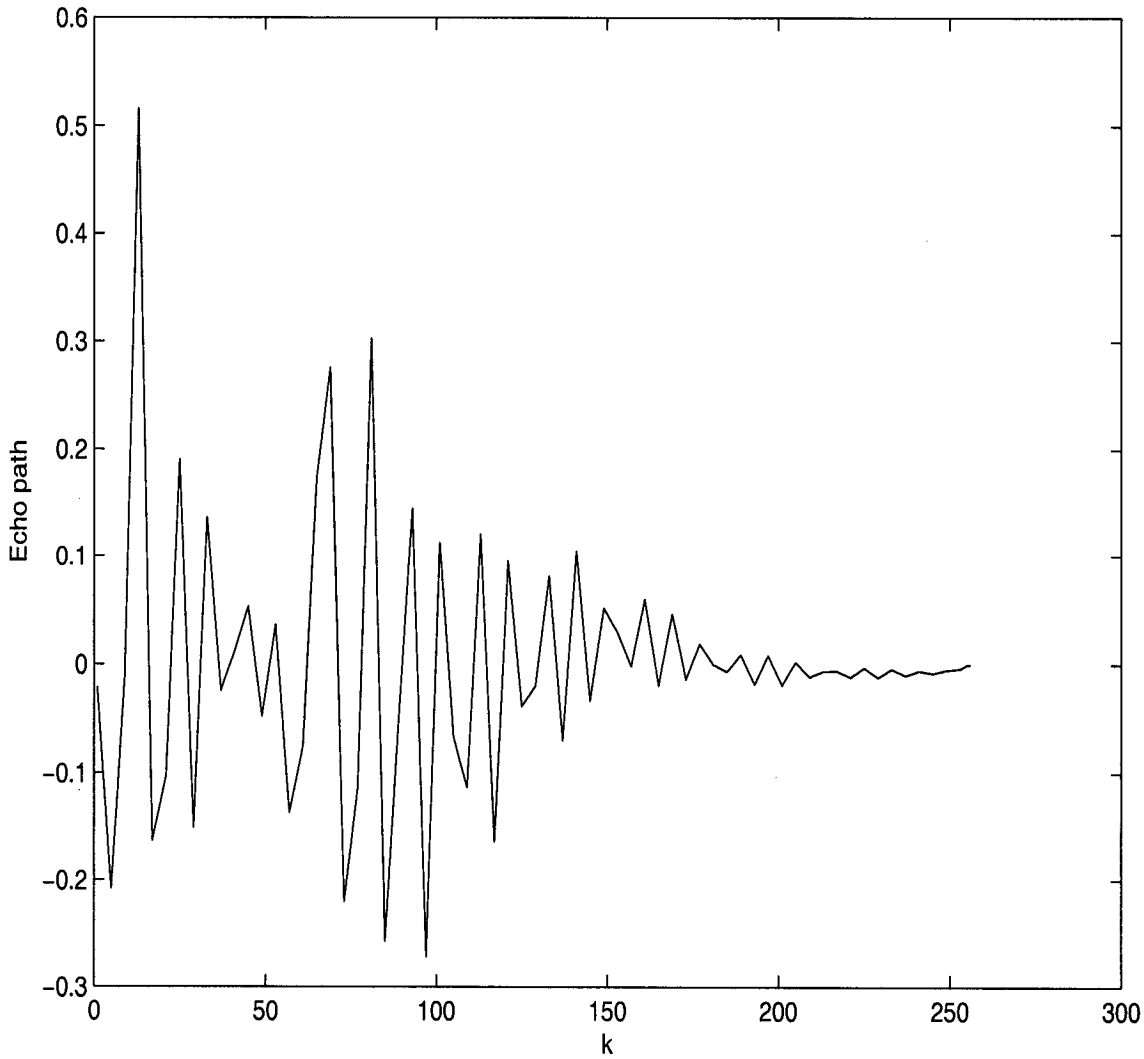

Figure 6.3. Dispersive echo path.

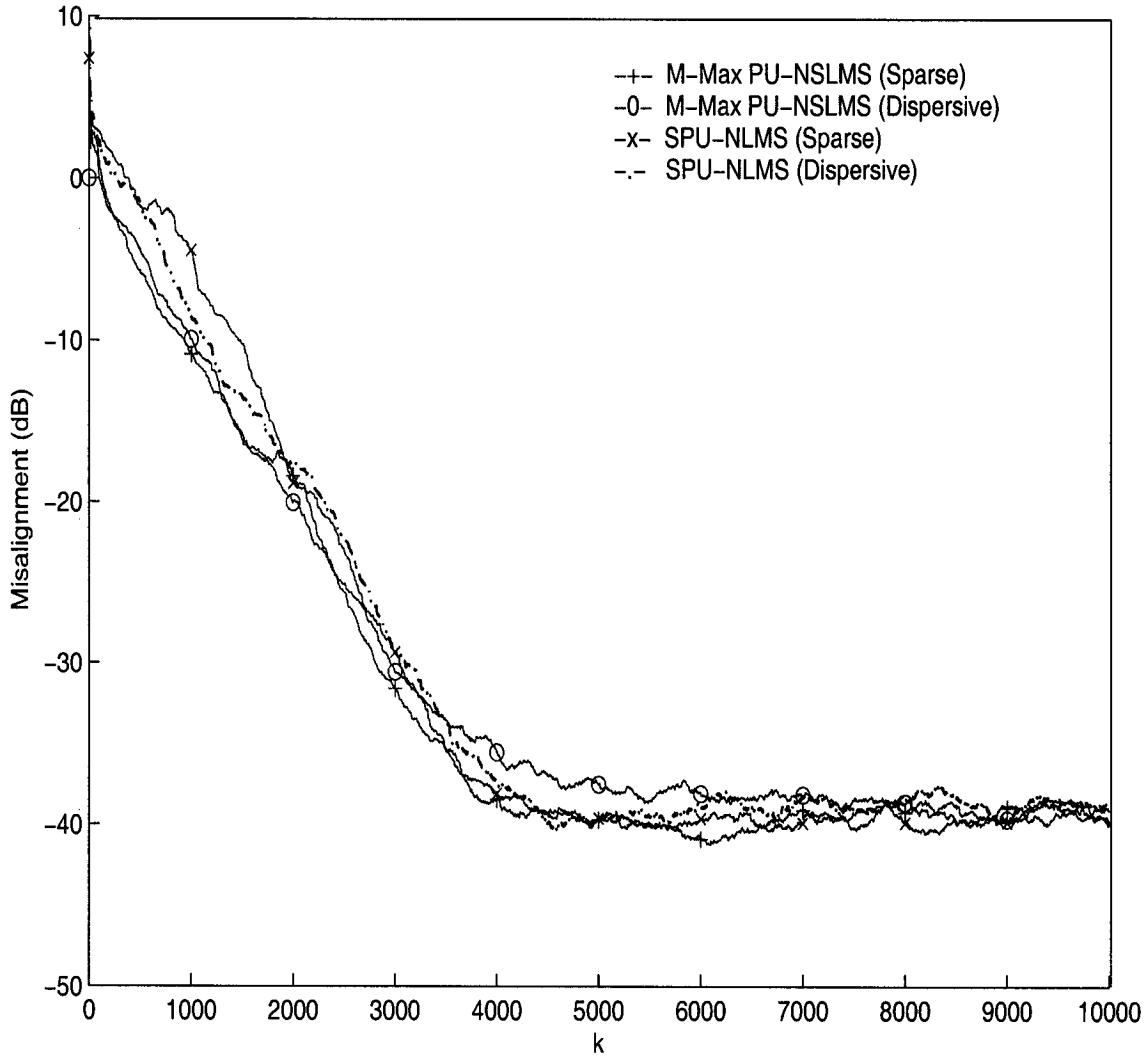

Figure 6.4. Convergence rates of the M-Max PU-NSLMS and SPU-NLMS algorithms for the case of $N = 256$ and $M = 32$ using the sparse and dispersive echo paths of Figs. 6.2 and 6.3, respectively.

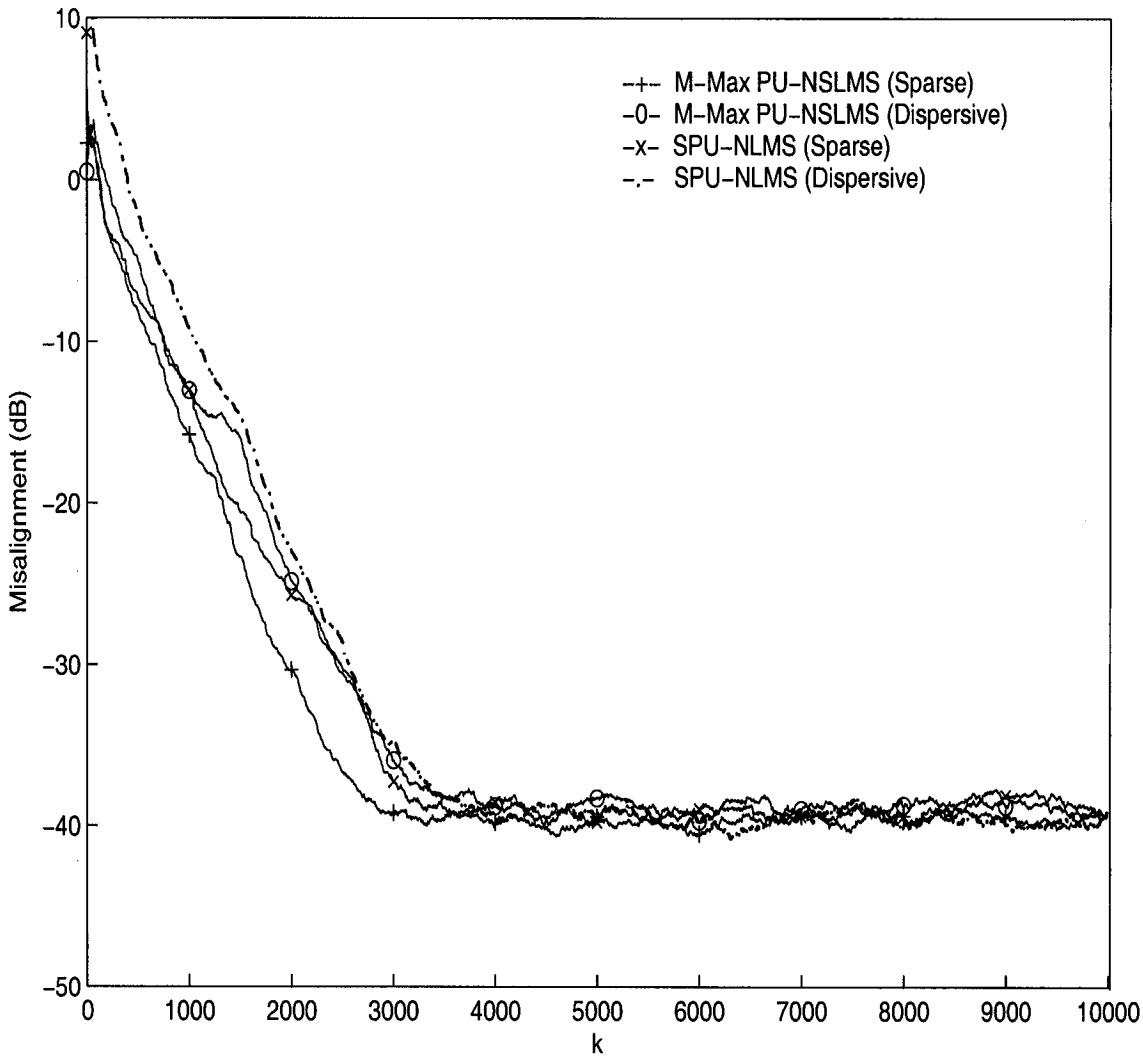

Figure 6.5. Convergence rates of the M-Max PU-NSLMS and SPU-NLMS algorithms for the case of $N = 256$ and $M = 64$ using the sparse and dispersive echo paths of Figs. 6.2 and 6.3, respectively.

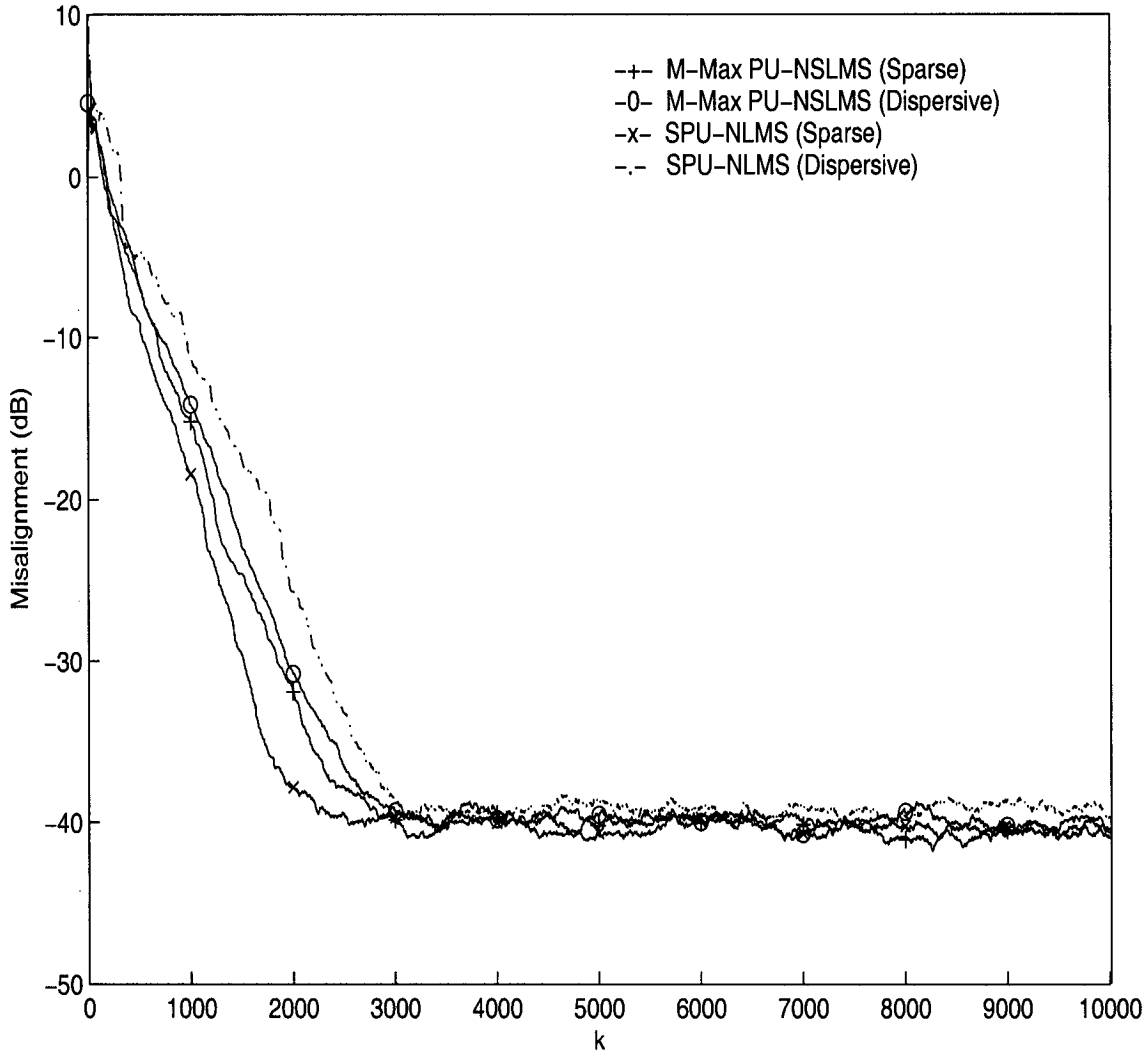

Figure 6.6. Convergence rates of the M-Max PU-NSLMS and SPU-NLMS algorithms for the case of $N = 256$ and $M = 128$ using the sparse and dispersive echo paths of Figs. 6.2 and 6.3, respectively.

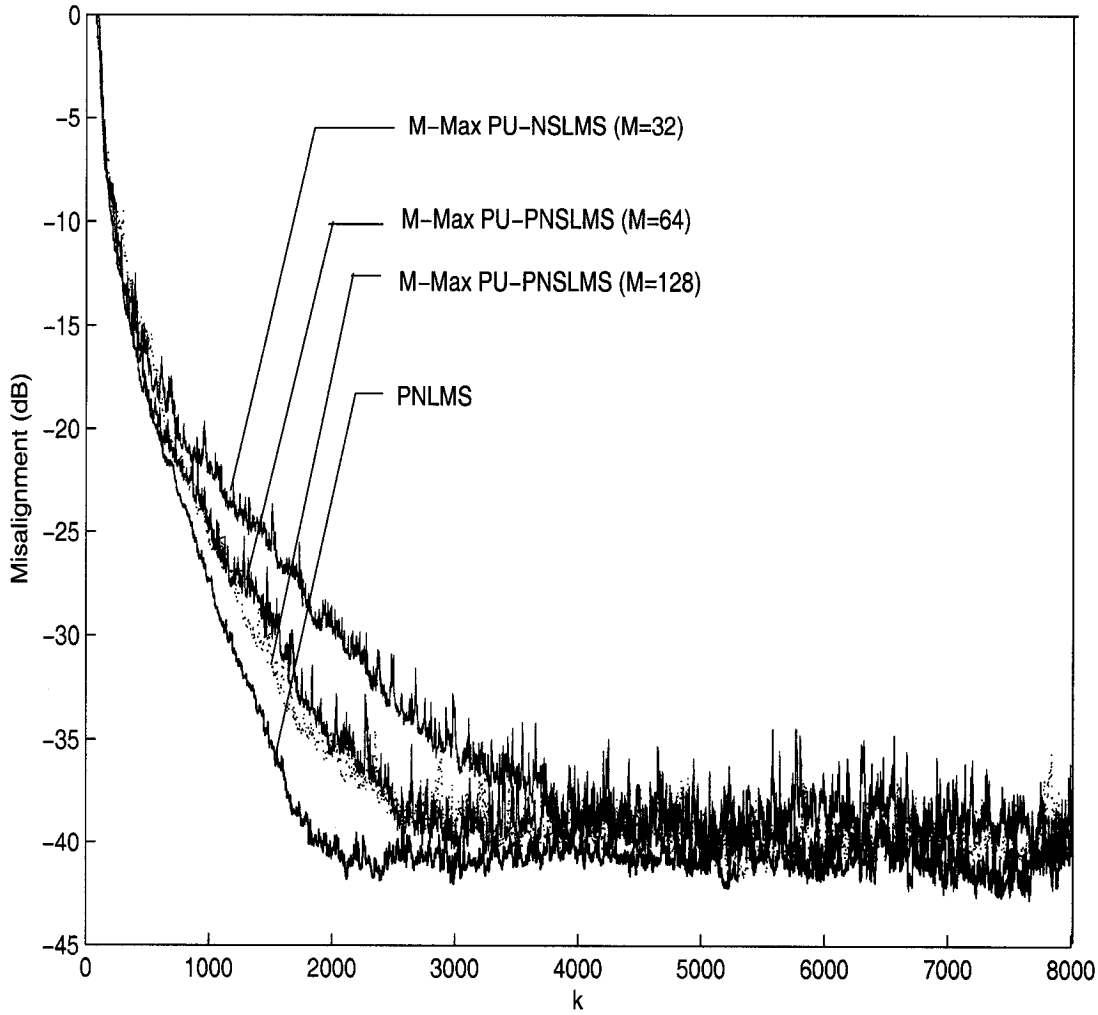

Figure 6.7. Convergence rates of the PNLMS algorithm ($N = M = 256$) and the proposed M-Max PU-PNSLMS algorithm ($N = 256$ and $M = 32, 64, 128$) using the sparse echo path of Fig. 6.2.

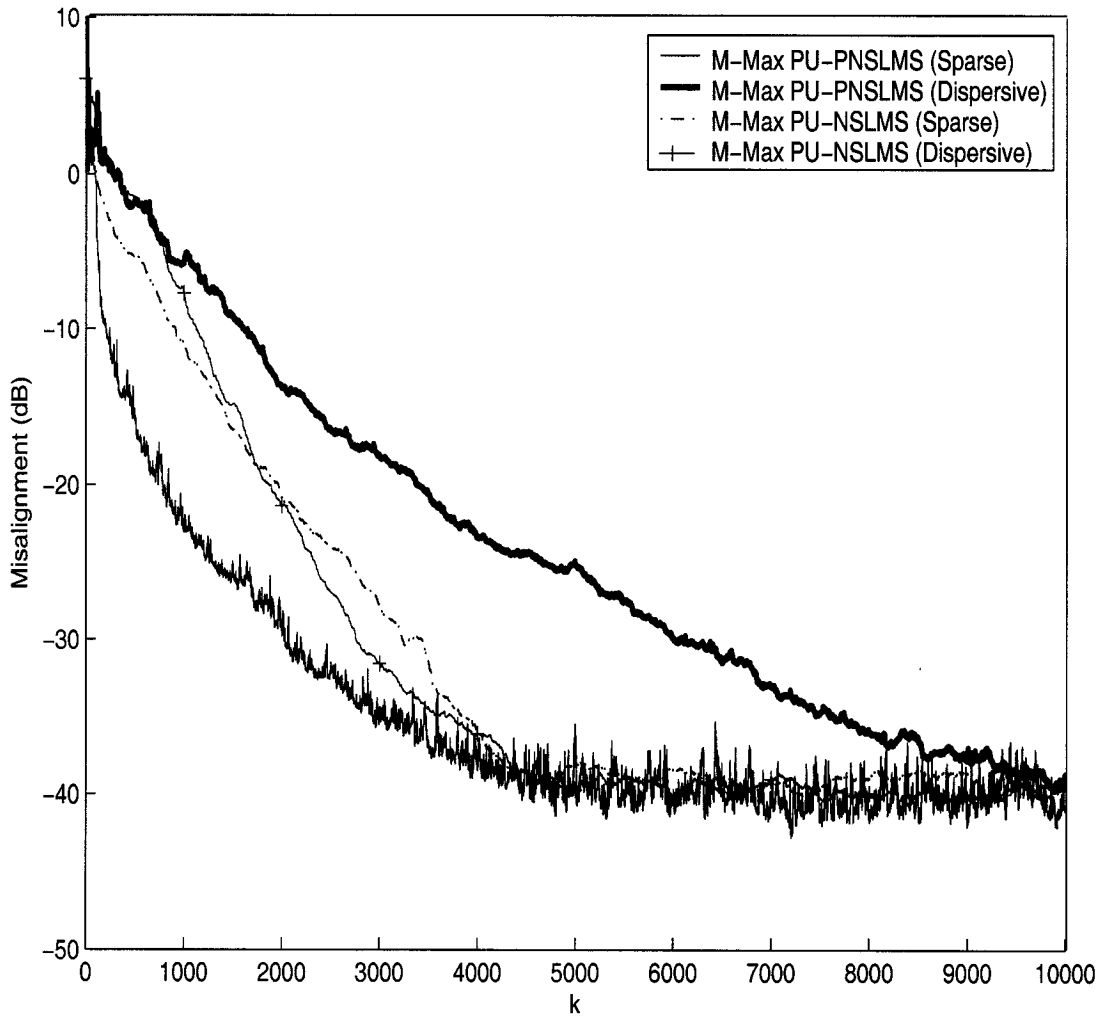

Figure 6.8. Convergence rates of the M-Max PU-PNSLMS and M-Max PU-NSLMS algorithms for the case of $N = 256$ and $M = 32$ using the sparse and dispersive echo paths of Figs. 6.2 and 6.3, respectively.

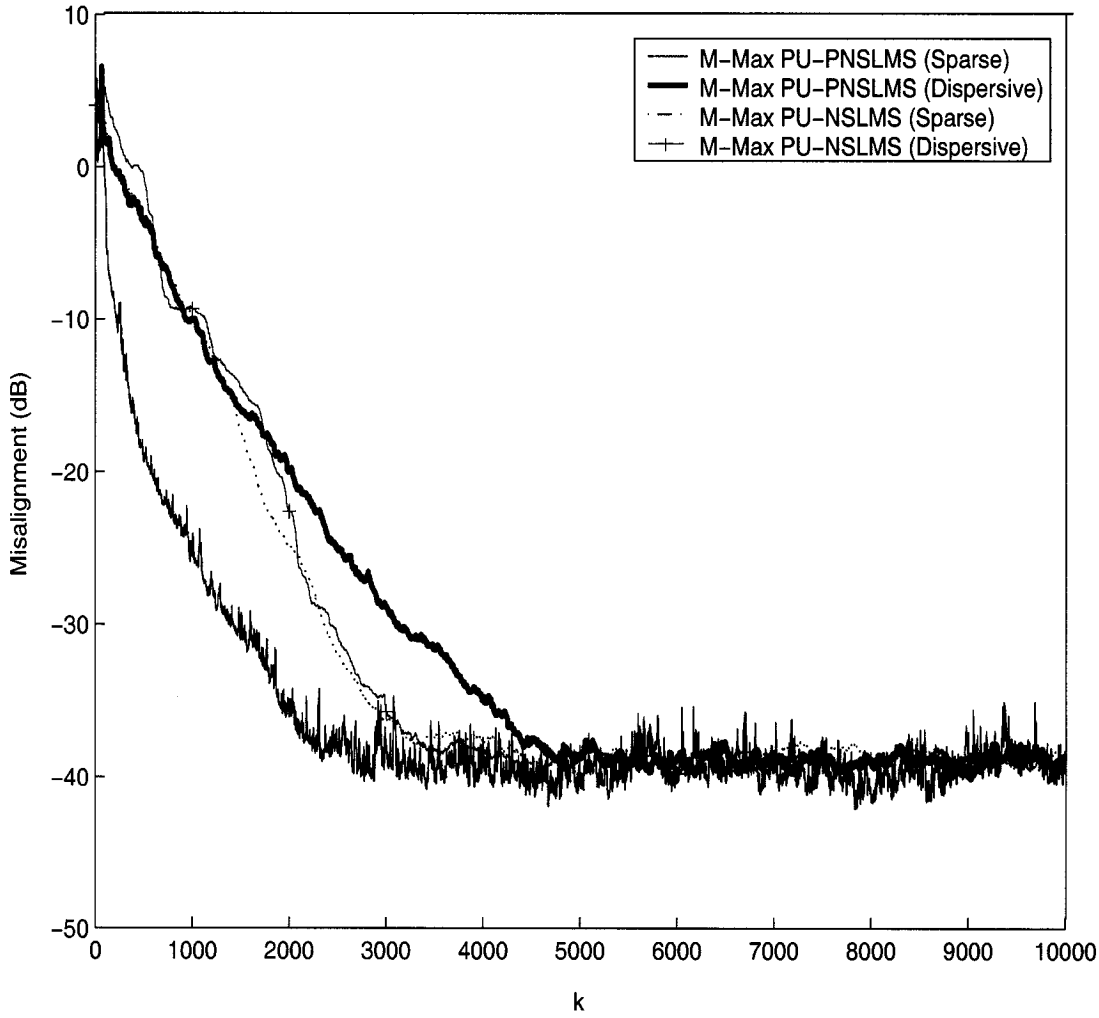

Figure 6.9. Convergence rates of the M-Max PU-PNSLMS and M-Max PU-NSLMS algorithms for the case of $N = 256$ and $M = 64$ using the sparse and dispersive echo paths of Figs. 6.2 and 6.3, respectively.

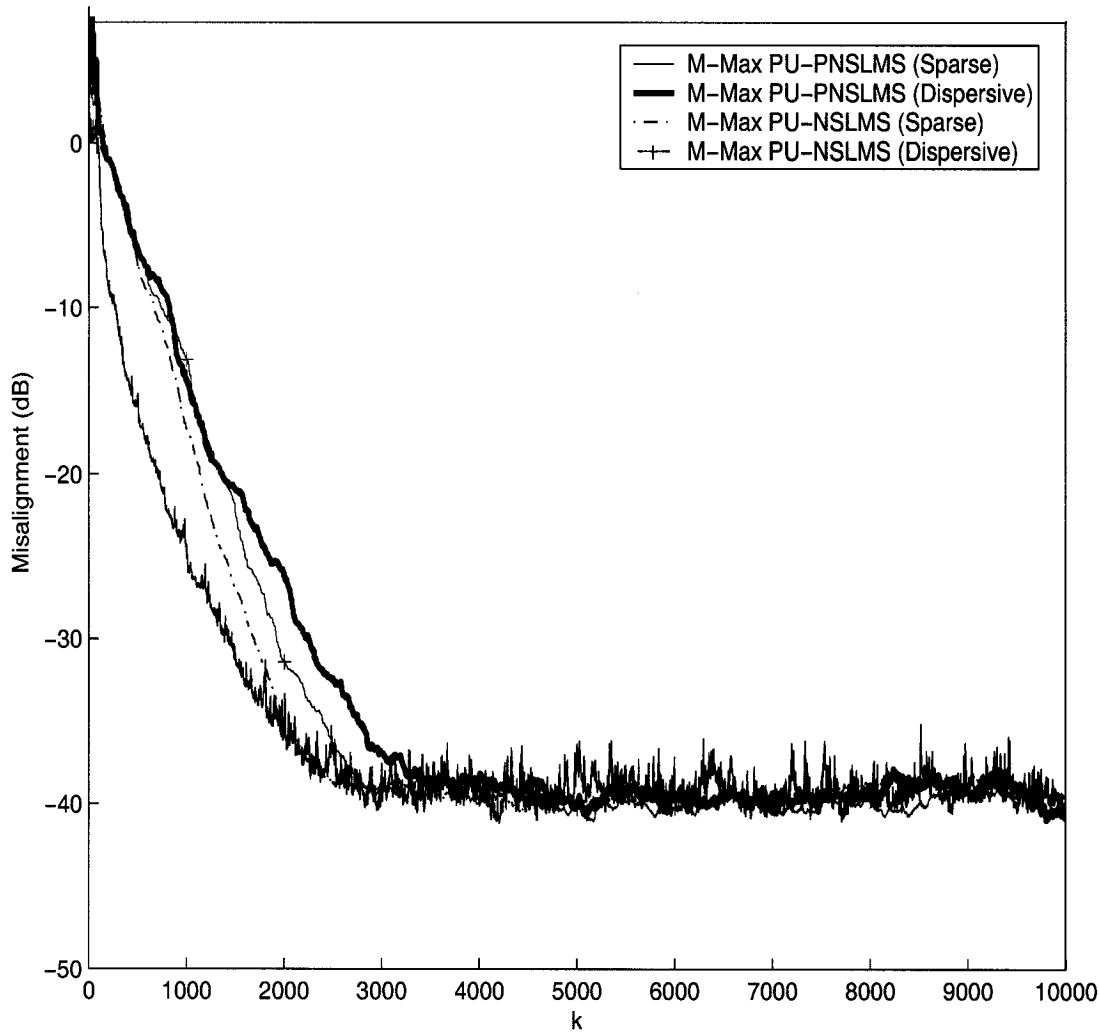

Figure 6.10. Convergence rates of the M-Max PU-PNSLMS and M-Max PU-NSLMS algorithms for the case of $N = 256$ and $M = 128$ using the sparse and dispersive echo paths of Figs. 6.2 and 6.3, respectively.

compared to the M-Max PU-NSLMS algorithm. However, for a dispersive echo path, where the energy of the impulse response is spread over a wide region, the convergence rate of the M-Max PU-PNSLMS algorithm is slower compared to that of the M-Max PU-NSLMS algorithm.

6.3 Summary

In this chapter, first, a brief introduction to the problem of echo cancellation has been given. Next, the performance of the proposed algorithms, namely, the M-Max PU-NSLMS and M-Max PU-PNSLMS algorithms have been studied and compared with that of the SPU-NLMS and PNLMS algorithms, respectively, both from the point of the complexity and the convergence rate. The complexities of these algorithms have been compared in terms of the number of multiplications, additions and comparisons required to update the filter coefficients at each time step. It has been observed that the proposed M-Max PU-NSLMS and M-Max PU-PNSLMS algorithms are computationally less intensive compared to the SPU-NLMS and PNLMS algorithms, respectively. For comparing the convergence rates of the algorithms, different echo paths have been used. It has been shown that the convergence rate of the M-Max PU-NSLMS algorithm is comparable to that of the SPU-NLMS algorithm both for the sparse and dispersive echo paths. Further, it has been observed that the convergence rates of the two algorithms are independent of the nature of the echo path. It has been shown that the convergence rates of the M-Max PU-PNSLMS and PNLMS algorithms are comparable. Also, it has been observed that in the case of a sparse echo path, the M-Max PU-PNSLMS algorithm shows a minimal degradation in the convergence rate as the number of filter coefficients being updated at each time step decreases. The convergence rates of the M-Max PU-NSLMS and M-Max PU-PNSLMS algorithms have been compared using the sparse and dispersive echo paths. It has been observed that in

the case of a sparse echo path, the convergence rate of the M-Max PU-PNSLMS algorithm is faster than that of the M-Max PU-NSLMS algorithm. Also, in the case of a dispersive echo path, the convergence rate of the M-Max PU-PNSLMS algorithm deteriorates and is slower compared to that of the M-Max PU-NSLMS algorithm. As a consequence, the proposed algorithms have lower complexities compared to the existing algorithms, with the former algorithms providing a performance comparable to that of the later in terms of the convergence rate.

Chapter 7

Conclusion

The ability of an adaptive filter to adapt itself to the changes in an unknown environment makes it a powerful tool for signal processing. One of the main applications of adaptive filters is in the identification of unknown system. The complexity of an adaptive filter algorithm is proportional to the tap length of the filter, and therefore, such an algorithm can be computationally probhitive for applications requiring a long filter tap.

This thesis has been concerned with the development of low-complexity adaptive filter algorithms by combining the concept of partial-updating and the technique of finding the gradient vector in the hyperplane based on the L_∞-norm criterion. The resulting algorithms, referred to as the PU-NSLMS algorithms, have a reduced complexity not only because of the updating of only a subset of the filter coefficients at each time step, but also from the fact that compared to the L_2-norm algorithm (for example, the NLMS algorithm), algorithms based on L_∞-norm require less number of operations to update a filter coefficient. Two specific algorithms, namely, the sequential and M-Max PU-NSLMS algorithms have been obtained by employing the sequential and M-Max coefficient selection techniques for updating a subset of the filter coefficients. The statistical analyses of the two algorithms have been carried out under the independence assumption. Evolution equations for the

mean and mean-square of the filter coefficient misalignment as well as the stability bounds on the step-size of the two algorithms have been obtained. It has been shown that the M-Max PU-NSLMS algorithm has a convergence rate that is closest to that of the full update algorithm. Simulations have been carried out to validate the theoretical results obtained from the statistical analyses of the algorithms. The concept of partial-updating based on the L_∞-norm criterion was extended to the proportionate adaptive filtering to develop the PU-PNSLMS algorithm.

The performance of the proposed algorithms as well as that of the existing ones have been studied in echo cancellation. The complexities of these algorithms have been compared for different values of, N, the tap length of the filter and M, the number of filter coefficients updated at each time step. Simulations using the sparse and dispersive echo paths have been carried out to study the convergence rates of the algorithms. It has been shown that the convergence rate of the proposed M-Max PU-NSLMS algorithm is comparable to that of the SPU-NLMS algorithm both for the sparse and dispersive echo paths. Further, it has been observed that the convergence rates of the two algorithms are independent of the echo paths. It has been shown that the convergence rate of the M-Max PU-PNSLMS algorithm is comparable to that of the PNLMS algorithm. Also, it has been observed that in the case of a sparse echo path, the M-Max PU-PNSLMS algorithm shows a minimal degradation in the convergence rate as the number of filter coefficients being updated at each time step decreases. The convergence rates of the M-Max PU-NSLMS and M-Max PU-PNSLMS algorithms have been compared for the sparse and dispersive echo paths. It has been shown that in the case of a sparse echo path, the M-Max PU-PNSLMS algorithm converges much faster compared to the M-Max PU-NSLMS algorithm. However, in the case of a dispersive echo path, the convergence rate of the M-Max PU-PNSLMS algorithm deteriorates and is slower compared to that of the M-Max PU-NSLMS algorithm.

One of the major contributions of this thesis is the statistical analyses of the sequential

and M-Max PU-NSLMS algorithms. Results obtained from these analyses have helped us to better understand the behavior of the two proposed algorithms. A similar analysis for the M-Max PU-PNSLMS algorithm needs to be carried out. Also, the performance of the proposed algorithms have been studied in the case of a stationary, zero-mean, white Gaussian input signal. Real life signals are neither stationary nor generated from a white Gaussian process. Therefore, the performance of the proposed algorithms need to be studied for nonstationary and correlated signals. Also, a detailed study needs to be carried out to understand the behavior of the proposed algorithms in the case of a finite word length environment, such as in the case of a fixed-point processor.

Appendix A

In this appendix, the derivation of (4.14) is given. Let $[.]_{m,m}$ refer to the $(m,m)^{th}$ element of the the matrix [.]. Solving the diagonal terms in (4.13) separately, we have

$$\begin{aligned}
\Psi_1 &= \left[-\widehat{\mu}E\left\{sign\left\{\mathbf{A}(k)\mathbf{X}(k)\right\}\mathbf{X}^T(k)\right\}E\left\{\mathbf{V}(k)\right\}\right]_{m,m} \\
&= \left[-\widehat{\mu}E\left\{sign\left\{\mathbf{A}(k)\mathbf{X}(k)\right\}\mathbf{X}^T(k)\right\}\right]_{m,m} E\left\{\mathbf{V}(k)\right\}_{m,m} \\
&= -\widehat{\mu}\sqrt{\frac{2}{\pi}}\sigma_x\zeta_1 E\left\{\mathbf{V}(k)\right\}_{m,m} \qquad (7.1)
\end{aligned}$$

$$\begin{aligned}
\Psi_2 &= \left[-\widehat{\mu}E\left\{\mathbf{V}(k)\right\}E\left\{sign\left\{\mathbf{A}(k)\mathbf{X}(k)\right\}\mathbf{X}^T(k)\right\}\right]_{m,m} \\
&= E\left\{\mathbf{V}(k)\right\}_{m,m}\left[-\widehat{\mu}E\left\{sign\left\{\mathbf{A}(k)\mathbf{X}(k)\right\}\mathbf{X}^T(k)\right\}\right]_{m,m} \\
&= -\widehat{\mu}\sqrt{\frac{2}{\pi}}\sigma_x\zeta_1 E\left\{\mathbf{V}(k)\right\}_{m,m} \qquad (7.2)
\end{aligned}$$

$$\begin{aligned}
\Psi_3 &= \left[\widehat{\mu}^2E\left\{sign\left\{\mathbf{A}(k)\mathbf{X}(k)\right\}\mathbf{X}^T(k)\mathbf{V}(k)\mathbf{X}(k)sign\left\{\mathbf{X}^T(k)\right\}\right\}\right] \\
&= \widehat{\mu}^2\frac{M}{N}\sigma_x^2\left[\sigma_{\tilde{w}}^2(k)+(\zeta_2-1)E\left\{\mathbf{V}(k)\right\}_{m,m}\right] \qquad (7.3)
\end{aligned}$$

$$\begin{aligned}
\Psi_4 &= \left[\widehat{\mu}^2E\left\{sign\left\{\mathbf{A}(k)\mathbf{X}(k)\right\}sign\left\{\mathbf{X}^T(k)\mathbf{A}(k)\right\}\right\}\sigma_n^2\right]_{m,m} \\
&= \widehat{\mu}^2\frac{M}{N}\sigma_n^2 \qquad (7.4)
\end{aligned}$$

where

$$\zeta_1 = \frac{NE\{\|\mathbf{A}(k)\mathbf{X}(k)\|_1\}}{ME\{\|\mathbf{X}(k)\|_1\}}$$

and

$$\zeta_2 = \frac{NE\{\|\mathbf{A}(k)\mathbf{X}^2(k)\|_1\}}{ME\{\|\mathbf{X}^2(k)\|_1\}}$$

The use of (7.1), (7.2), (7.3), and (7.4) in (4.13) results in (4.14).

References

[1] P. S. R. Diniz, *Adaptive Filtering-Algorithms and Practical Implementation,* 2nd ed. Boston, MA: Kluwer, 2002.

[2] Stefan Werner, Marcello L. R. de Campos, and Paulo S. R. Diniz, "Partial-update NLMS algorithms with data-selective updating," *IEEE Transactions on Signal Processing*, vol. 52, No. 4, pp. 938-949, April 2004.

[3] K. Dogancay and O. Tanrikulu, " Adaptive Filtering algorithms with selective partial update," *IEEE Transactions on Circuits and Systems-II*, vol. 48, pp. 762-769, Aug. 2001.

[4] K. Dogancay and O. Tanrikulu, "Selective-partial-update NLMS and affine projection algorithms for acoustic echo cancellation," *Proc. IEEE International Conference on Acoustic, Speech and Signal Processing,* vol. 1, pp. 448-451, Istanbul, Turkey, June 2000.

[5] T. Aboulnasr and K. Mayyas, "Complexity reduction of the NLMS algorithm via selective coefficient update," *IEEE Transactions on Signal Processing*, vol. 47, pp. 1421-1427, May 1999.

[6] M. I. Haddad, K. Mayyas and M. A. Khasawneh, "Analytical development of the MMAXNLMS algorithm," *Proc. IEEE International Conference on Acoustic, Speech and Signal Processing,* vol. 4, pp. 1853-1856, Phoenix, AZ, March 1999.

[7] G. A. Clark, S. K. Mitra, and S. R. parker, " Block implementation of adaptive digital filters," *IEEE Transactions on Acoustic, Speech and Signal Processing,* vol. ASSP-29, pp. 744-752, June 1981.

[8] T. Schertler, "Selective block update of NLMS type alorithms," *Proc. IEEE International Conference on Acoustic, Speech and Signal Processing,* vol. 3, pp. 1717-1720, Seattle, USA, May 1998.

[9] S. M. Kuo, and J. Chen, "Multiple microphone acoustic echo cancellation with the partial adaptive process," *Digital Signal processing*, Vol. 3, pp. 54-63, 1993.

[10] S. C. Douglas, "Adaptive filters employing partial updates," *IEEE Transactions on Circuits and Systems-II*, vol. 44, pp. 209-216, March 1997.

[11] S. C. Douglas, " A Family of Normalized LMS Algorithms," *IEEE Signal Processing Letters*, vol. 1 , No. 3, pp. 49-51, March 1994.

[12] S. C. Douglas, "Analysis and implementation of the max-NLMS filter," *Proc. Twenty-Ninth Asilomar Conf. Signals, Syst. Comput.*, vol. 1, pp. 659-663, Pacific Grove, CA, Oct. 1995.

[13] J. Nagumo and A. Noda, "A learning method for system identification," *IEEE Transactions on Automat. Contr.*, vol. AC-12, no. 3, pp. 283-287, June 1967.

[14] S. H. Cho, Y. S. Kim, and J. A. Cadzow, "Adaptive FIR filtering based on the L_∞-norm," *Proc. IEEE Pacific Rim Conference Commun. Comput. Signal Processing*, pp. 643-646, vol. 2, May 1991.

[15] B. Widrow and S. D. Stearns, *Adaptive Signal Processing,* Prentice Hall, Englewood Cliffs, New Jersey, 1985.

[16] G.C. Goodwin and K. S. Sin, *Adaptive Filtering, Prediction* and *Control.* Englewood Cliffs, NJ: Prentice-Hall, 1984.

[17] B. Widrow, et al., "Adaptive noise cancelling: Principles and applications," *Proc. of IEEE*, Vol. 63, pp. 1692-1716, Dec. 1975.

[18] B. Widrow, et al., "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proc. of IEEE*, Vol. 64, pp. 1151-1162, Aug. 1976.

[19] L.L. Horowitz and K.D. Senne, "Performance advantage of complex LMS for controlling narrow-band adaptive arrays," *IEEE Transactions on Acoustic, Speech and Signal Processing*,, Vol. 29, pp. 722-736, June 1981.

[20] W. A. Gardner, "Learning characteristics of stochastic-gradient-descent algorithms: A general study, analysis, and critique, " *Signal Processing,* Vol. 6, No. 2, pp. 113-133, April 1989.

[21] A. Feuer and E. Weinstein, "Convergence analysis of LMS filters with uncorrelated Gaussian data," *IEEE Transactions on Acoustic, Speech, Signal Processing*, vol. ASSP-33, –. 222-230, Feb. 1985.

[22] V. Solo, "The limiting behavior of the LMS," *IEEE Transactions on Acoustic , Speech and Signal Processing,* vol. 37, pp. 1909-1922, 1989.

[23] Albert A.E. and L.S. Gardner Jr., *Stochastic Approximation and Nonlinear Regression*, MIT press, Cambridge, Mass.

[24] M. Tarrab and A. Feuer, "Convergence and performance analysis of the Normalized LMS Algorithm with Uncorrelated Gaussian data," *IEEE Transactions on Info. Theory,* vol. 34, No. 4, July 1988.

[25] R. R. Bitmead and B.D.O. Anderson, "Performance of adaptive estimation algorithms in dependent random environment," *IEEE Transactions on Automat. Contr.*, vol AC-25, pp. 788-793, Aug. 1980.

[26] N. J. Bershad, "Analysis of the normalized LMS algorithm with Gaussian inputs," *IEEE Transactions on Acoustic, Speech, Signal Processing*, vol. ASSP-34, pp. 793-806, Aug. 1986.

[27] T.C. Hsia, "Convergence analysis of LMS and NLMS adaptive algorithms," *Proc. IEEE International Conference on Acoustic, Speech and Signal Processing,* pp. 667-670, 1983.

[28] D.T.M. Slock, "On the convergence behavior of the LMS and the normalized LMS algorithms," *IEEE Transactions on Signal processing*, vol. 4, Sept. 1993.

[29] I. Pitas, "Fast algorithms for running ordering and max/min calculation" *IEEE Transactions on Circuits and Systems-II,* vol. 36, pp. 795-804, June 1 1989.

[30] J. E. Mazo, "On the independence theory of the equalizer convergence," *Bell Sys. Tech. J.,* vol. 58, pp. 963-993, May-June 1979.

[31] A. Papoulis, *Probability, Random Variables, and Stochastic Processes,* McGraw-Hill, 2001.

[32] H. A. David, Order statistics, Wiley, 1981.

[33] H. J. Godvin, "Some low moments of order statistics," *Ann. Math. Statist.,* vol. 20, pp. 279-285, June 1949.

[34] H. L. Harter, "Expected values of normal order-statistics," *Biometrika*, vol. 48, pp. 151-165, June 1961.

[35] Z. Govindarajalu, "Exact moments of order statistics in samples from chi-distribution (1 d.f)," *Ann. Math. Statist.,* vol. 33, pp. 1292-1305, Dec. 1963.

[36] D. L. Duttweiler,"Proportionate normalized least mean square adaptation in echo cancelers," *IEEE Transactions on Speech Audio Processing,* vol. 8, pp. 508-518, Sept. 2000.

[37] S. L. Gay, "An efficient, fast converging adaptive filter for network echo cancellation," *Proc. Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, 1998.

[38] J. Benesty, and S. L. Gay,"An Improved PNLMS algorithm," *IEEE International Conference on Acoustic, Speech and Signal Processing*, pp. 1881-1884, vol. 2, May 2002.

[39] O. Tanrikulu, and K. Dogancay, "Selective-partial-update proportionate normalized least-mean-squares algorithm for network echo cancellation," *IEEE Conference Acoustics, Speech, Signal Processing,* vol. 2, 1889-1892, 2002.

[40] S. Makino, Y. Kaneda and N. Koizumi, "Exponentially weighted step-size adaptive filter based on the statistics of a room impulse response," *IEEE Transactions on Speech Audio Processing*, vol. 1, Jan. 1993.

[41] A. Sugiyama, H. Sato, A. Hirano, and S. Ikeda, "A fast convergence algorithm for adaptive FIR filters under computational constraint for adaptive tap-position control," *IEEE Transactions on Circuits and Systems-II*, vol. 43, no. CAS-9, Sept. 1966.

[42] J. Homer, I. Mareels, R.R. Bitmead, B. Wahlberg, and F. Gustafsson, "LMS estimation via structural detection," *IEEE Transactions on Signal Processing,* vol. 46, Oct. 1998.

[43] R. K. Martin and C. R. Johnson Jr., "NSLMS: A proportional weight algorithm for sparse adaptive filters," *Proc. Asilomar Conference on Signals, Systems,* Pacific Grove, CA, 2001.

[44] R. K. Martin, W. A. Sethares, R. C. Williamson, and C. R. Johnson Jr., "Exploiting sparsity in adaptive filters," *IEEE Transactions on Signal Processing*, vol. 50, pp. 1883-1894, Aug. 2002.

[45] T. Gansler, S. L. Gay, M. M. Sondhi,and J. Benesty, "Double talk robust fast converging algorithms for network echo cancellation," *IEEE Transactions on Speech Audio Processing,* vol. 8, pp. 656-663, Nov. 2000.

[46] A. Maleki, K. Nayebi,"A new efficient PNLMS based algorithm for adaptive line echo cancellation," *Proc. IEEE International Symposium on Signal processing and its applications*, pp. 555-558, July 2003.

[47] M. M. Sondhi, "An adaptive echo canceller," *Bell Systems Technical Journal,* 1967.