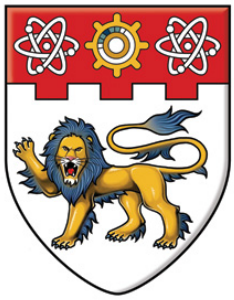

NANYANG TECHNOLOGICAL UNIVERSITY

Performance Analysis and Enhancements of Adaptive Algorithms and Their Applications

SHENGKUI ZHAO

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the Degree of
Doctor of Philosophy

2009

Abstract

Adaptive filters that self-adjust their transfer functions according to optimizing algorithms are powerful adaptive systems with numerous applications in the fields of signal processing, communications, radar, sonar, seismology, navigation systems and biomedical engineering. An adaptive signal processing algorithm, e.g., the least mean squares (LMS) algorithm and the recursive least square (RLS) algorithm, is used to deal with adaptation of adaptive filters. The adaptive algorithms are expected to be computationally simple, numerically robust, fast convergent and low fluctuant. Unfortunately, none of the adaptive algorithms developed so far perfectly fulfils these requirements. The stability and convergence performance of the widely-used adaptive algorithms also haven't been fully explored. This work aims to deal with performance analysis and enhancements for the adaptive algorithms and their applications.

We first develop a new variable step-size adjustment scheme for the LMS algorithm using a quotient form of filtered quadratic output errors. Compared to the existing approaches, the proposed scheme reduces the convergence sensitivity to the power of the measurement noise and improves the steady-state performance and tracking capability for comparable transient behavior, with negligible increase in the computational costs.

We then develop variable step-size approaches for the normalized least mean squares (NLMS) algorithm. We derive the optimal step-size which minimizes the mean square deviation at each iteration, and propose four approximated step-sizes according to the correlation properties of the additive noise and the variations of the input excitation.

We next analyze the stability and performance of the transform-domain LMS algorithms which preprocess the inputs with a fixed data-independent orthogonal transform such as

the discrete Fourier transform (DFT), discrete Hartley transform (DHT), discrete cosine transform (DCT), discrete sine transform (DST) and power normalization. For the particular second-order autoregressive inputs, we derive the eigenvalue distributions of the autocorrelation matrices obtained after the transformations of DFT, DHT, DCT, DST and power normalization, and we make comparisons on the convergence performance of the transform-domain LMS algorithms.

Finally, we enhance the performance of the conjugate gradient (CG) algorithms by presenting a generalized data windowing scheme. We analyze that the misadjustment of the new CG algorithms can reduce to zero, which is comparable to that of the RLS algorithm.

In addition, we demonstrate the performance of the enhanced algorithms and verify the analytical results by computer simulations in system identification, noise cancelation and channel equalization.

Acknowledgments

My deepest gratitude goes to my supervisor and mentor, Prof. Man Zhihong, for his passionate guidance, support, advices and encouragements throughout my studies at Nanyang Technological University (NTU). His kindness, patience, and warmth make it a pleasant journey to accomplish my Ph.D study. I will always remember his teaching.

I gratefully thank Assoc/Prof. Cai Jianfei for his supervision, help, support in the final year of my Ph.D candidate.

I am also very grateful to all the members in C^2I lab for their friendship and support, especially my colleague Khoo Suiyang for his cooperation and help. I also thank the lab technicians, Mr Tan Swee Huat and Mr. Lau Boon Chee, for their help of setting up my working station.

I would like to thank all my family members for their loves, life cares and encouragements.

I acknowledge the help of the School of Computer Engineering for offering me an opportunity and financial support to study at NTU.

Contents

List of Figures

List of Tables

List of Abbreviation

LMS	Least Mean Squares
NLMS	Normalized Least Mean Squares
RLS	Recursive Least Squares
CG	Conjugate Gradient
DFT	Discrete Fourier Transform
DHT	Discrete Hartley Transfrom
DCT	Discrete Cosine Transfrom
DST	Discrete Sine Transfrom
DWT	Discrete Wavelet Transfrom
WHT	Walsh-Hadamard Transform
FFT	Fast Fourier Transform
IFFT	Inverse Fast Fourier Transform
MSE	Mean Square Error
EMSE	Excess Mean Square Error
MSD	Mean Square Deviation
AR	Autoregressive
MV	Moving-Average
ARMA	Autoregressive-Moving-Average

List of Notation

$*$	Convolution
x_n	Input signal of adaptive filter at time index n
d_n	Desired signal of adaptive filter at time index n
y_n	Output signal of adaptive filter at time index n
e_n	Filter prediction error at time index n
e_n^o	Measurement noise at time index n
μ	Step size of adaptive algorithm
$\mathbf{x}_n$	Input vector of adaptive filter at time index n
$\mathbf{w}_n$	Weight vector of adaptive filter at time index n
$\mathbf{w}^o$	Optimal Wiener solution
$(\cdot)^T$	Matrix or vector transpose
$(\cdot)^H$	Matrix or vector Hermitian transpose
$(\cdot)^\star$	Matrix or vector conjugate
ξ_{min}	Minimum mean square error
ξ_{ex}	Steady-state value of the excess mean square error
$\mathbf{R}$	Input autocorrelation matrix
$\mathbf{p}$	Input cross-correlation vector
$\mathbf{R}_n$	Estimate of input autocorrelation matrix at time index n
$\mathbf{p}_n$	Estimate of input cross-correlation vector at time index n
$\mathbf{Q}$	Eigenvector matrix of the input autocorrelation matrix, $\mathbf{R}$
$\mathbf{I}$	Identity matrix
$\mathbf{\Lambda}$	Diagonal eigenvalue matrix of the input autocorrelation matrix, $\mathbf{R}$
$\boldsymbol{\lambda}$	Column vector of eigenvalues of the input autocorrelation matrix, $\mathbf{R}$
λ_{max}	The maximum eigenvalue of the input autocorrelation matrix, $\mathbf{R}$
λ_{min}	The minimum eigenvalue of the input autocorrelation matrix, $\mathbf{R}$
$E[\cdot]$	Expectation operation
Tr$[\cdot]$	Trace of matrix
det$[\cdot]$	Determinant of matrix

List of Author's Publications

[1] S. Zhao, Z. Man and S. Khoo, "A generalized data windowing scheme for adaptive conjugate gradient algorithms", Signal Processing, vol. 89, no. 5, pp. 894–900, May 2009.

[2] S. Zhao, Z. Man, S. Khoo and H. R. Wu, "Stability and convergence analysis of transform-domain LMS adaptive filters with second-order autoregressive process," IEEE Trans. on Signal Processing, vol. 57, no. 1, pp. 119–130, January 2009.

[3] S. Zhao, Z. Man, S. Khoo and H. R. Wu, "Variable step-size LMS algorithm with a quotient form", Signal Processing, vol. 89, no. 1, pp. 67–76, January 2009.

[4] Shengkui Zhao, Zhihong Man and Suiyang Khoo, "A class of modified variable step-size NLMS algorithms for system identification", Proceedings of the 4th IEEE Conference on Industrial Electronics and Applications, Xi'an of China, pp. 2987–2991, May 2009.

[5] Zhao Shengkui, Man Zhihong and Khoo Suiyang, "On performance of Transform Domain Adaptive Filters with Markov-2 Inputs", Proceedings of IEEE 3rd International Conference on Industrial Electronics and Applications, Singapore, pp.989–994, June 2008. ***(Selected as session chair)***

[6] Zhao Shengkui, Man Zhihong and Khoo Suiyang, "Conjugate gradient algorithm design with RLS normal equation", Proceedings of the 6th International Conference on Information, Communications, Signal Processing, Singapore, pp. 1–5, December 2007.

[7] Zhao Shengkui, Man Zhihong and Khoo Suiyang, "A Fast Variable Step-Size LMS Algorithm with System Identification", Proceedings of the 2nd IEEE International Conference on Industrial Electronics and Applications, Harbin of China, pp. 2340–2345, May 2007.

[8] Zhao Shengkui, Man Zhihong and Khoo Suiyang, "A New Variable Step-Size Transform Domain LMS Algorithm with System Identification", Proceedings of IEEE International Conference on Control and Automation, Guangzhou of China, pp. 2721–2724, May 2007.

[9] Zhao Shengkui, Man Zhihong and Khoo Suiyang, "Modified LMS and NLMS Algorithms with a New Variable Step Size", Proceedings of the 9th International Conference on Control, Automation, Robotics and Vision, Singapore, pp. 1–6, December 2006.

Chapter 1

Introduction

1.1 Motivation

This thesis deals with adaptive filtering problem. That is, given an input signal and a target signal, the filter parameters are designed iteratively to "best" map the input signal to the target signal. The estimation error arises from the difference between the target signal and the filter output signal. The mechanism of iteratively updating filter parameters comes from minimizing a cost function of the estimation errors. And the commonly used cost function is mean-square value or least-square value of the error signal.

For stationary input and desired signals, minimizing the mean-square error results in the well-known Wiener solution [1], which is said to be optimum in the mean-square error sense. The resulting filter is referred to as the Wiener filter. It has been shown that the Wiener filter is capable of handling the tasks of smoothing, estimation and prediction in continuous time as well as discrete time as long as they are involved with linear estimation. However, directly implementing the Wiener filter consists of solving for the Wiener solution, which is a function of the autocorrelation of the input signal and the cross-correlation between the input and the desired signals. That is, the Wiener filter requires a priori knowledge of the statistics of the underlying signals. Strictly speaking, reliably estimating these statistic requires large number of realizations of the underlying signal sequences. This procedure is not feasible in practice since we usually have only one realization for each of the signal sequences. By ergodic property of the underlying

signals, this can be done by computing the time averages of one realization. However, in most practical applications, these second order statistics can only be estimated from the data the system has seen so far. Therefore, the Wiener filters of being determined once forever in a preliminary design phase are fixed, non-adaptive filters, which are suited to the situations where the statistics of the intervening signals are time invariant. And they are not effective to reflect changes that occur in the statistics of the signals or parameters of time-varying systems.

The adaptive filters that can track the changes in the statistics of the signals or parameters of time-varying systems, differ from the non-adaptive Wiener filter in that their impulse response (a set of parameters for discrete-time filters) are adjusted iteratively as data flow through the filters. Over the last four decades, adaptive filters have found application in many areas. They have been used in noise cancelation [2–4], channel equalization [5–8], system identification [9–14], echo cancelation [15–18], antenna arrays [19–23], line enhancement [24–29], control [30–33], and many other applications. The reasons for solving the problems with adaptive filters instead of non-adaptive filters could be summarized into four points. Firstly, the non-adaptive filter requires direct computation of the necessary time averages based on large amount of signal samples, which requires significant amount of memory. The adaptive filter, on the other hand, updates its parameters each time the new signal sample is available, thereby saving memory. Secondly, the accumulating process of signal samples and their post processing to generate the filter output, as required in non-adaptive filters, introduces a large delay in the filter output. This is unacceptable in many real-time applications such as acoustic echo cancellation, dynamic system identification, channel equalization etc. The adaptive filters, on the contrary, do not introduce any significant delay in the filter output since the filter output is usually estimated for each new input signal sample comes in (note that there are adaptive filters with block processing which introduce moderate delay). Thirdly, another important property of the adaptive filters is the tracking of the variations of signal statistics or time-varying systems. The non-adaptive filters can not adapt to the new statistics once they have been designed. Fourthly, an adaptive, in general, is much simpler to code in software or to implement in hardware than the non-adaptive counterpart.

As follows, some examples are to be given to explain that why an adaptive filter, instead of a non-adaptive filter, is more attractive. In the application of acoustic echo

cancellation, taking one example of the scenario that arises in teleconferencing applications. The speech signal from a far-end speaker, received through a communication channel, is broadcast by a loudspeaker in a room and picked up by a local microphone. The filter to cancel the effect of echo is to be designed to model the room transfer function. The non-adaptive filters estimate the room transfer function by accumulating a large amount of speech signal samples from the far-end speaker and the reflected signals received by the microphone. It will require a large amount of memory and the post processing also introduces delay. The main problem is that the speech signal is nonstationary and the statistics of input signals are changing. In addition, the room transfer function may be time-varying due to the movements of human or objects in the room. All these factors will make the non-adaptive filters ineffective. The adaptive filters, on the other hand, update their parameters by minimizing cost functions of the estimation error which reflects the mismatch between the filter output signal and the desired signal due to the changes. In this way, the adaptive filters can track the changes in the nonstationary speech statistics as well as the variations of the room transfer function. In the class of applications dealing with identification, an unknown plant is modeled by a linear filter. The filter and the plant are driven by the same input. It is to be designed that the filter output can best resemble the plant output. For the plant which is dynamic in nature, the model will be time varying. However, a non-adaptive filter only provides a fixed model, which is, therefore, not suited to identify the dynamic plants. An adaptive filter can self-design its adjustable coefficients based on the estimation error. Therefore, it is applicable to model the dynamic plant. Let us consider the case of noise cancellation. In this application, a signal is corrupted by a noise uncorrelated with the signal. The combined signal and noise form the primary input to the canceller. A second input to the canceller is also a noise which is uncorrelated with the signal but correlated in some unknown way with the noise added in the signal. The canceller filters the second noise input to produce an output that is as close a replica as possible to the noise added in the signal. This output is subtracted from the primary input. If one knew the characteristics of the channels over which the noise was transmitted to the forms in the primary input and reference input, it would theoretically be possible to design a fixed non-adaptive filter capable of changing the second reference noise input into the noise in the primary input.

The filter output could then be subtracted from the primary input, and the system output would be signal alone. Since, however, the characteristics of the transmission paths are as a rule unknown or known only approximately and are seldom of a fixed nature, the use of a fixed filter is non feasible. An adaptive filter can automatically adjusts its own parameters responding to an error signal dependent on the filter output. Therefore, it is capable of removing the noise satisfactorily [4]. The ability of an adaptive filter to operate satisfactorily in an unknown environment and track time variations of input statistics makes the adaptive filter a powerful device. By the manner in which the desired signal is extracted, the adaptive filtering applications may be classified as four classes: identification, inverse modeling, prediction and interference cancellation [1].

The parameters (usually refer to filter coefficients) of an adaptive filter are usually randomly initialized or based on the a priori knowledge available to the system, and they are refined each time the new input sample is available. The mechanism for adjusting the parameters of an adaptive filter is referred to as an adaptive algorithm. In an adaptive algorithm, the refining procedure is designed based on some criteria, and the most popular criteria are the minimum mean square error (MMSE) and least square error. One such adaptive algorithm is the least mean squares (LMS) algorithm, which was invented by Widrow and Hoff in the early 1960's [34]. The LMS algorithm is extremely simple since it minimizes the instantaneous square error instead of the mean square error, using a simple gradient-based optimization method. This algorithm achieves the Wiener solution in mean sense. Because of its simplicity, the LMS algorithm shows many robust properties, but also displays slow convergence, specially for color inputs. Another important issue of the LMS algorithm is that the precision of its steady-state solution is directly proportional to the learning rate (or step-size). Reducing the learning rate, as a result, improves the precision of the steady-state solution, but has the direct consequence of slowing down the convergence.

Another famous adaptive algorithm is the recursive least squares (RLS) algorithm [1]. The RLS algorithm achieves the least square error at each iteration by finding the best approximation of the Wiener solution based on all the data the system has seen so far. As a result, the RLS algorithm is able to achieve convergence quickly. However, it suffers from the problems of high computational cost, and poor numerical stability for ill input conditions.

The convergence speed of an adaptive algorithm measures the number of iterations to be taken for the algorithm to reach a desired state. Usually some iterations are required for convergence of an adaptive algorithm. In most of the applications, a fast convergence speed is required. For example, in the application of acoustic echo cancellation, before the convergence of an adaptive algorithm the estimated room transfer function can be quite different from the actual transfer function. During the convergence time, applying the echo subtraction from the speech source will produce high distortion, which will make difficulty of comprehension for the far end. On the other hand, the stability of an adaptive algorithm measures its working reliability. Poor stability may eventually make the systems involved in the real applications totally fail. That means the software or hardware is totally useless. For example, in the application of channel equalization, an adaptive filter is used to model the inverse transfer function of a communication channel. The receiver is expected to receive the correct signals transmitted from the sending part. If the adaptive algorithm is unstable, it may never be converged and the estimated transfer function of the communication channel will be wrong. The receiver will never recover the sending information.

As consequences, the increasing requirements of fast convergence, low computational cost, good robustness and high numerical stability from new applications and existing systems are motivating new methods for adaptive algorithms to enhance or replace the LMS and RLS algorithms. Therefore, this thesis is to deal with analyzing and enhancing the performance of the adaptive algorithms. Specifically, the variable step-size LMS and normalized LMS (NLMS) algorithms, transform-domain LMS algorithms, and conjugate gradient (CG) algorithms are concentrated.

1.2 Author's Contributions

The work described in this thesis involves developments of variable step-size adjustment schemes for the LMS and NLMS algorithms. Their convergence and stability issues are also examined. Then, the transform-domain LMS algorithms with various orthogonal transforms are analyzed and asymptotic eigenvalue spreads of the first and second-order autoregressive inputs preprocessed by orthogonal transforms and power normalization are

computed. Lastly, the performance of the conjugate gradient algorithms are enhanced by proposing a generalized data windowing scheme. It will be shown that the misadjustment of the new enhanced CG algorithms can be reduced to zero. In addition, the applications of the enhanced adaptive algorithms on system identification, noise cancellation and channel equalization are demonstrated. The simulation results are shown to support the theoretical analysis well.

The first part of the thesis deals with the developments of robust variable step-size adjustment schemes for the LMS and NLMS algorithms. It was shown that design of the variable step-size adjustment scheme for the LMS-type adaptive algorithms should be based on the desirable criteria as follows. The LMS-type algorithms with the variable step-size can achieve good tradeoff between convergence speed and steady-state misadjustment. The algorithms can fast track the statistic changes of the inputs or the system variations. Initialization of parameters involved in the variable step-size adjustment schemes is easy and affected as less as possible by measurement noise. In addition, the increase of computation cost due to the adaptation of the step-size should be moderate compared to the implementation of the algorithms with fixed step-size. According to these criterions, the existing approaches for adjusting variable step-size still need to improve since they can only perform well in some of the criteria. The fact is that designing a variable step-size scheme to addressing all these criteria is a challenge. In this work, we propose a variable step-size adjustment scheme of a quotient form motivated by its interesting features. The performance analysis and experiments show that the LMS algorithm with the new proposed scheme outperforms the existing approaches in steady-state performance and tracking time-varying systems.

Although the variable step-size adjustment schemes designed for the LMS algorithms can be applied to the NLMS algorithm. Analysis showed that the optimal step-size of the NLMS algorithm can be derived and approximated for a better performance. In this work, we approximate the optimal step-size of minimizing the mean-square deviation by using the properties of the additive noise and input excitation, which leads to a new class of variable step-size approaches for the NLMS algorithm. Experiments show that this new class of variable step-size NLMS algorithms give more favorite features compared to the existing approaches.

We then investigate another important class of the LMS-based adaptive algorithms, i.e., the transform-domain LMS algorithms. Previous analysis and experiments showed that the LMS-type algorithms perform well with the whitened input signals. In the real applications, the input signals are usually correlated and the simple LMS-type algorithms converge very slowly. To overcome this problem, preprocessing colored input signals with a fixed, data-independent orthogonal transform was shown to improve the convergence. Many transforms have been considered, such as the discrete Fourier transform (DFT), the discrete Hartley transform (DHT), the discrete cosine transform (DCT), the discrete sine transform (DST), discrete Wavelet transform (DWT), Walsh-Hadamard transform (WHT), etc. The resulting algorithms are generally referred to as DFT-LMS, DHT-LMS, DCT-LMS, DST-LMS, etc. Although these transform-domain LMS algorithms are promising, their convergence properties are shown to relay on the class of input process and the choice of transforms. A comparison on the convergence performance of these algorithms are desired. In following part of the thesis, we examine the stability and convergence performance of these transform-domain LMS algorithms with the second-order autoregressive (AR) input process. We derive the asymptotic results of the preprocessed input eigenvalue spreads for different transforms. Convergence performance are made based on the asymptotic results. Our analysis shows that in general DCT-LMS and DST-LMS provide better convergence performance than DFT-LMS and DHT-LMS. DCT-LMS provides the best convergence performance with the lowpass second-order AR input signals, and DST-LMS provides the best convergence performance with the bandpass and highpass second-order AR input signals. DFT-LMS and DHT-LMS always show same performance for such inputs. The challenge of extending the methodology to higher order AR processes are also discussed. Although our analysis is limited to the second order. Computer simulations show that it is also true for the high order cases and other processes like moving-average (MA) process and autoregressive-moving-average (ARMA) process.

The last part of the thesis deals with the tradeoff of conjugate gradient (CG) algorithms. It is known that the conjugate gradient (CG) algorithms designed for adaptive filtering problem based on the CG method of optimization have convergence properties better than the LMS-type algorithms and have more robust numerical properties compared to the RLS algorithm. However, the current data windowing schemes used for

the estimation of the autocorrelation matrix and cross-correlation vector lose the convergence or increase the misadjustment. To address these issues, we implement the CG algorithms with a generalized data windowing scheme. The convergence and steady-state performance of the new developed CG algorithms are shown to be comparable with those of the RLS algorithms with little increase of computational cost. The convergence behavior of the CG algorithms in norm-2 sense is also analyzed.

The major contributions of this thesis can be summarized as follows:

- Performance analysis and robustness enhancement of variable step-size approach for the LMS algorithm by proposing a quotient form of variable step-size.
- Simplification and performance enhancement of variable step-size approach for the NLMS algorithm by analyzing the properties of the additive noise and input excitation.
- Stability and convergence performance analysis of transform-domain LMS algorithms including the DFT-LMS, DHT-LMS, DCT-LMS and DST-LMS by deriving and comparing the asymptotic results of the preprocessed input eigenvalue spreads with second-order autoregressive inputs.
- Convergence improvements of the CG algorithms by applying a generalized data windowing scheme and analysis of convergence behaviors of the CG algorithms in norm-2 sense.
- Verification of those proposed schemes and analytic results by simulation examples.

1.3 Thesis Outline

In Chapter 2, the important families of the LMS and RLS algorithms are reviewed. Their derivations and major properties are briefly summarized.

In Chapter 3, the issues of the existing variable step-size approaches for the LMS algorithm such as high computational complexity, sensitive to power of measurement noise and slow tracking are addressed. A more robust variable step-size LMS (MRVSS)

algorithm with a quotient form is proposed. Studies on the mean and mean square convergence, steady-state performance and mean step-size behavior of the MRVSS algorithm are made. Simulation results with system identification and noise cancelation are presented to demonstrate the improved performance.

In Chapter 4, a preferred new class of variable step-size NLMS algorithms are formulated based on the characteristics of input excitation and measurement noise. Improved filter performance are demonstrated by computer simulations.

In Chapter 5, the stability of the transform-domain LMS algorithms including the DFT-LMS, DHT-LMS, DCT-LMS and DST-LMS are addressed first, and then their convergence performance for second-order autoregressive inputs are investigated and compared. The asymptotic eigenvalue spread of the input autocorrelation matrix of the algorithms are derived. Finally, computer simulation are demonstrated to corroborate the analytic results.

In Chapter 6, a generalized data windowing scheme is proposed and modifications of the CG algorithms with the new scheme are made. Analysis for the convergence behavior of the CG algorithms in normal-2 sense is made. Computer simulations for system identification and channel equalization are demonstrated.

In Chapter 7, conclusions are drawn for the thesis and the interesting number of areas are summarized.

Chapter 2

The Families of the LMS and RLS Algorithms

In this chapter, we first introduce the motivation of adaptive algorithm and the principles of adaptive transversal filter. Then, the derivations and properties of the famous families of the LMS and RLS algorithms are summarized. Lastly, a discussion on the limitations of the classic LMS and RLS algorithms are given. Based on the review, we are directed to the main research work of the thesis.

2.1 Motivation of Adaptive Algorithm

Let us demonstrate the needs of adaptive algorithm with a simple application: system identification. In practice, it is necessary to identify a physical process or system in the sense of creating a model for it that can describe the system's input-output behavior. Consider a discrete-time unknown system. The input signal to the system is denoted by x_n, and d_n is the system output. We would like to identify the impulse response $\mathbf{w}$ (a set of coefficients) of the system. That is, we would like to build a model for the system such that the model produces a response which resembles the system's output when applying the same input. Under the assumptions that the system input x_n is a wide-sense stationary random process with the autocorrelation matrix of $\mathbf{R}$ and the

cross-correlation vector $\mathbf{p}$ between the input signal x_n and the system output d_n, and the model has the structure of an FIR filter, the solution $\mathbf{w}^o$ of the Wiener-Hopf equation of $\mathbf{R}\mathbf{w}^o = \mathbf{p}$ is optimal in the sense of minimizing the mean square difference between the system's output and the model's response. However, in practice, the statistics of the input process are normally unknown, and the input-output characteristics of the system may evolve with time. In such cases, the method of building the model in one-time phase may not be applicable. Even though the statistics of the input process can be estimated, the Wiener-Hopf equation has to be computed each time the input statistics change.

A simple structure for building the model using an adaptive filter is shown in Figure 2.1. The input to the system is passed through the adaptive filter of N variable coefficients or weights. The filter output y_n is compared with the noisy observations of the system output d_n. The coefficients of the adaptive filter are adjustable with the modeling error e_n to produce the best possible match between the system output and filter output. The adaptive algorithms provide the filter with an algorithmic solution which allows for the tracking of time variations in the input statistics without having to solve the Wiener-Hopf equation. Typically, adaptive filters work as follows. The filter weights are initially set to zero or to arbitrary values. Then, at each iteration, the weights are adjusted by using an optimization method so as to improve the solution. The important point to note is that, under the appropriate conditions, the filter weights converge to the solution of the Wiener-Hopf equation. In this thesis, we concentrate on the adaptive algorithms for discrete-time adaptive FIR filter due to the advanced developments of digital signal processing and the inherent stability of FIR filter. In the next section, we will describe the transversal structure of the discrete-time FIR adaptive filter, which serves as the fundamentals towards the introduction of the important adaptive algorithms.

2.2 Principles of Adaptive Transversal Filter

A discrete-time adaptive FIR transversal filter [1] of length N is shown in Figure 2.2. The input N signal samples $x_n, x_{n-1}, \cdots, x_{n-N+1}$ to the filter come from a tap-delay line, and are assumed to be drawn from a wide-sense stationary stochastic process of zero-mean. The filter coefficients, $w_n(0), w_n(1), \cdots, w_n(N-1)$, are referred to as the weights

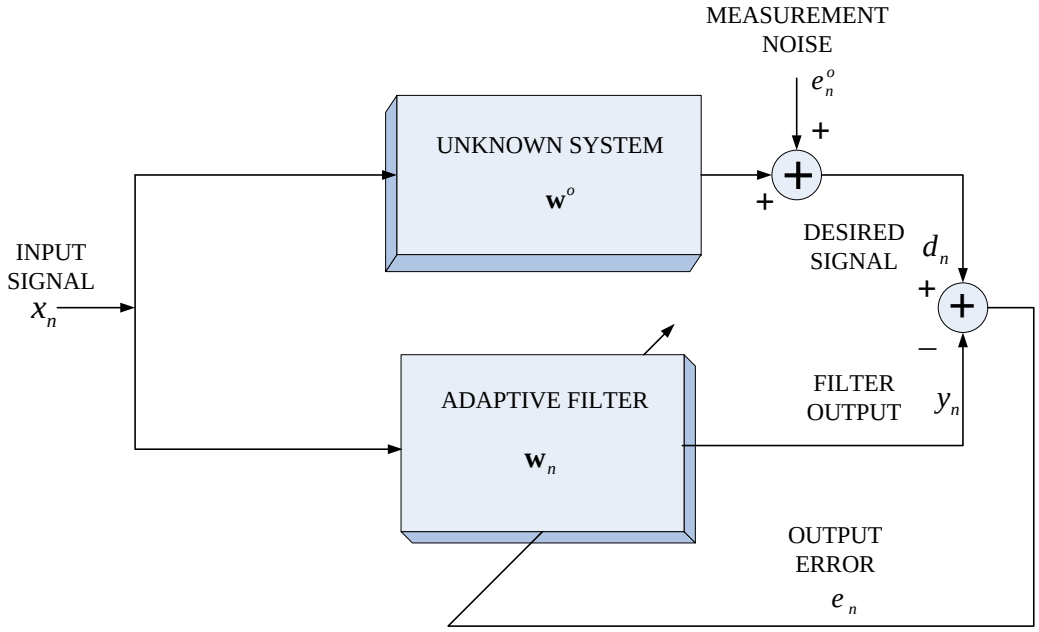

Figure 2.1: System identification using an adaptive filter.

of the filter. The filter is also supplied with a desired response d_n that supervises the adjustment of the filter weights to make the filter output y_n resemble d_n. The estimation error e_n measures the distance between the filter output y_n and the desired response d_n. In another words,

$$\begin{aligned} e_n &= d_n - y_n \\ &= d_n - \mathbf{w}_n^T \mathbf{x}_n, \end{aligned} \tag{2.1}$$

where $\mathbf{w}_n = [w_n(0), w_n(1), \cdots, w_n(N-1)]^T$ is the weight vector, and the input vector is given by $\mathbf{x}_n = [x_n, x_{n-1}, \cdots, x_{n-N+1}]^T$. Here, the output signal y_n is expressed as the inner product of the weight and input vectors. The superscript T denotes the matrix transpose.

If the input vector $\mathbf{x}_n$ and the desired response d_n are jointly stationary, the mean

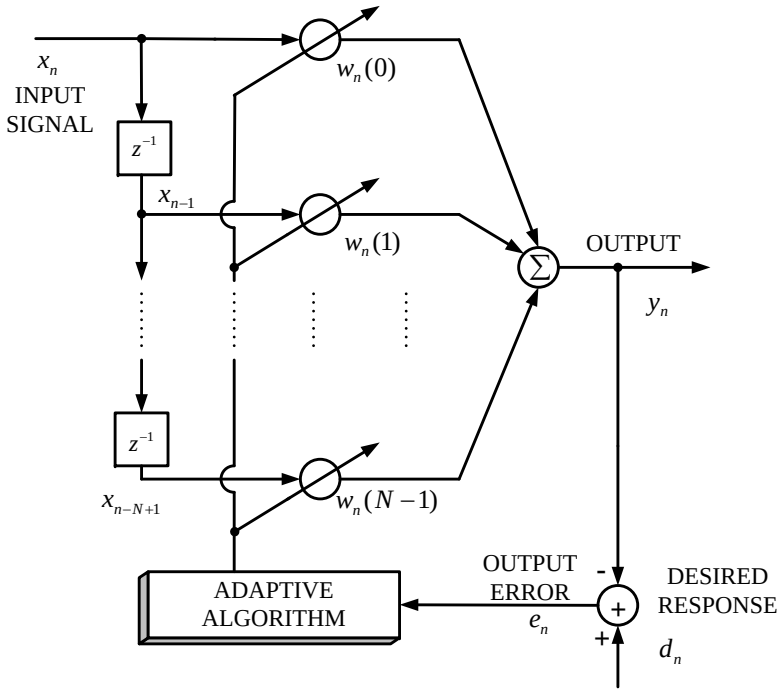

Figure 2.2: Structure of adaptive transversal filter.

square error (MSE) criterion with a fixed weight vector $\mathbf{w}$ can be expressed as

$$\begin{aligned} \xi(\mathbf{w}) &= E\left[e_n^2\right] \\ &= E\left[d_n^2\right] - 2\mathbf{w}^T\mathbf{p} + \mathbf{w}^T\mathbf{R}\mathbf{w}, \end{aligned} \tag{2.2}$$

where $\mathbf{R} \triangleq E\left[\mathbf{x}_n\mathbf{x}_n^T\right]$ and $\mathbf{p} \triangleq E\left[d_n\mathbf{x}_n\right]$. When the input is stationary and tap-delayed, the autocorrelation matrix $\mathbf{R}$ is Hermitian, Toeplitz and nonnegative definite [1].

The MSE function $\xi(\mathbf{w})$ is a quadratic function of the weight vector, and the MSE surface $\xi(\mathbf{w})$ is a hyperparaboloid for $N > 2$ and a paraboloid for $N = 2$. An example of a two-dimensional surface with contours is illustrated in Figure 2.3. The convexity of the MSE surface $\xi(\mathbf{w})$ means that the MSE is minimized at some point of the weight space. The mathematical method of finding such a point denoted by $\mathbf{w}^o$ is to take the derivative of $\xi(\mathbf{w})$ with respect to $\mathbf{w}$, and then set it to zero. This leads to the Wiener-Hopf equation: $\mathbf{R}\mathbf{w}^o = \mathbf{p}$. The solution, which is called the Wiener solution, is equal

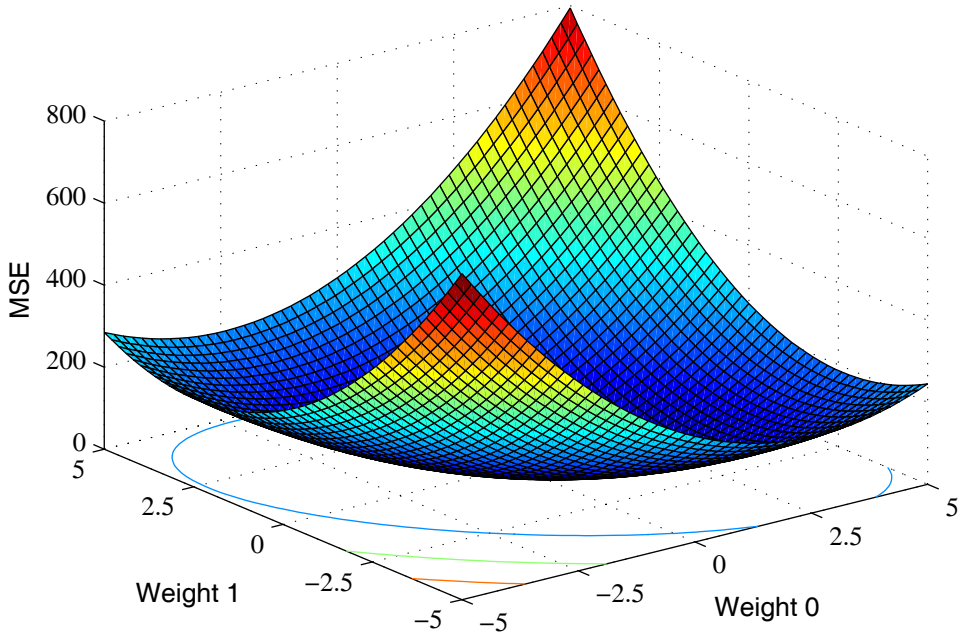

Figure 2.3: MSE surface for 2-D case.

to

$$\mathbf{w}^o = \mathbf{R}^{-1}\mathbf{p}. \tag{2.3}$$

The adaptive filter is said to be optimal with such solution. Substituting this solution into (2.2), the minimum achievable MSE is given by

$$\xi_{min} = \xi(\mathbf{w}^o) = \sigma_d^2 - \mathbf{p}^T\mathbf{R}^{-1}\mathbf{p}. \tag{2.4}$$

where σ_d^2 denotes the variance of the desired response d_n. Eq. (2.4) indicates that the optimal weight vector removes part of the power of the desired signal through cross-correlation between $\mathbf{x}_n$ and d_n. If the reference signal and the input signal are orthogonal, it is easily checked that the optimal coefficients are equal to zero and the minimum MSE is $\xi_{min} = E[d_n^2]$.

2.3 Properties of the Wiener Solution

We now introduce some important properties of the Wiener solution in (2.3). The error-performance surface is further analyzed as

$$\frac{\partial E[e_n^2]}{\partial \mathbf{w}_n} = E\left[2e_n \frac{\partial e_n}{\mathbf{w}_n}\right] = -E[2e_n \mathbf{x}_n]. \tag{2.5}$$

When the optimal weight vector is used, the gradient vector (2.5) goes to zero, which implies that

$$E[e_n^o \mathbf{x}_n] = 0 \tag{2.6}$$

or

$$E[e_n^o x_{n-i}] = 0 \tag{2.7}$$

for $i = 0, 1, \cdots, N-1$, where e_n^o denotes the special value of the estimation error when the adaptive filter operates in its optimal condition. It consist of the measurement noise, system noise or other ambient noise in practical applications. Eq. (2.7) indicates that the optimal error resulted from the Wiener solution is orthogonal to the elements of the current input vector, and it was called as the *principle of orthogonality* [1]. This principle serves as the necessary and sufficient condition for the cost function $\xi(\mathbf{w}_n)$ to obtain its minimum value. When the principle of orthogonality can be met, the adaptive filter is said to perform the best way in mean-square sense.

The orthogonality can also be extended to the cross-correlation between the optimal filter output y_n^o and the estimation error e_n^o. That is,

$$E[y_n^o e_n^o] = E[\mathbf{w}_n^{oT} \mathbf{x}_n e_n^o] = \mathbf{w}_n^{oT} E[e_n^o \mathbf{x}_n] = 0. \tag{2.8}$$

When the filter operates in its optimal condition, the filter output and the corresponding estimation error are also orthogonal.

In real applications, instead of calculating the inverse of the autocorrelation matrix to obtain the Wiener solution, a recursive adaptive algorithm is preferred to iteratively search the Wiener solution using the available input data. Therefore, an adaptive filter involves an adaptation process, which is the automatic adjustment of the weight vector in accordance with the estimation error. To implement this adaptation process, an adaptive

algorithm is used to update the weight vector iteratively along the MSE surface from an initial point to its minimum point or a vicinity of its minimum point for stochastic algorithms. In the following sections, we will review two representative families of adaptive algorithms: the LMS algorithms and the RLS algorithms.

2.4 The LMS Algorithm

The LMS algorithm was invented by Widrow and Hoff [1, 34]. It is extremely simple and numerically robust, and is an important member of the family of stochastic gradient algorithms. Before we introduce the LMS algorithm, let's first briefly introduce the famous steepest descent optimization method.

From the discussion in the previous section, we know that the MSE surface is convex. Suppose that the MSE function $\xi(\mathbf{w})$ is continuously differentiable. The gradient vector $\nabla\xi(\mathbf{w})$ of $\xi(\mathbf{w})$ will always point to the inverse direction to the minimum MSE. Increasing the weights along the opposite direction of the gradient vector by a small amount will push the weights close to the minimum point in the weight space. Mathematically, it is described as

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \mu\frac{\partial\xi(\mathbf{w}_n)}{\partial\mathbf{w}_n}, \tag{2.9}$$

where n denotes the iteration number, μ is a positive constant called step-size parameter. Using $\mathbf{w}_n$ into the MSE function (2.2), and finding its derivative yields

$$\frac{\partial\xi(\mathbf{w}_n)}{\partial\mathbf{w}_n} = 2\mathbf{R}\mathbf{w}_n - 2\mathbf{p}. \tag{2.10}$$

Substituting the gradient vector into (2.9) leads to

$$\mathbf{w}_{n+1} = \mathbf{w}_n - 2\mu\left(\mathbf{R}\mathbf{w}_n - \mathbf{p}\right). \tag{2.11}$$

This is known as the steepest descent algorithm. The operation of this algorithm with two weight taps is illustrated by a numerical example in Figure 2.4. The algorithm starts at an initial state of the weight vector $(w_0(0), w_0(1))$ as positioned in the contour of the MSE performance surface, and then the next weight vector $(w_1(0), w_1(1))$ is updated

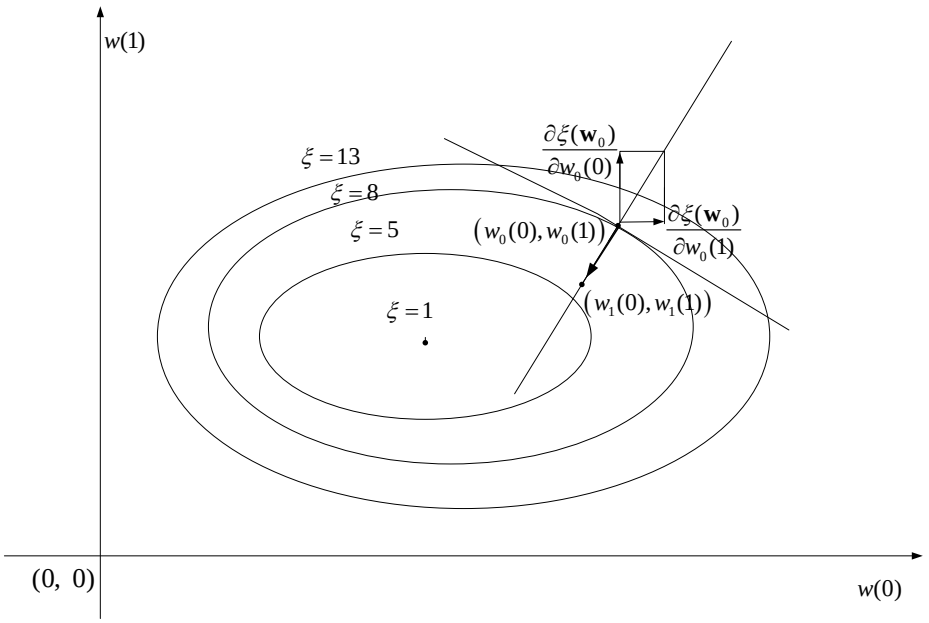

Figure 2.4: Weight vector update of the steepest descent algorithm.

in the inverse direction of the true gradient of MSE function $\xi(\mathbf{w}_0)$ to lead the value of $\xi(\mathbf{w}_1)$ closer to the optimal value ξ_{min}. By n steps in this way, the value of $\xi(\mathbf{w}_n)$ will approach to ξ_{min} and the weight vector $\mathbf{w}_n$ will be converging to Wiener solution $\mathbf{w}^o$.

Let λ_{max} be the maximum eigenvalue of $\mathbf{R}$. It can easily be checked that the necessary and sufficient condition for the convergence or stability of the steepest decent algorithm is given by $0 < \mu < \frac{1}{\lambda_{max}}$ [1]. When choosing the step-size μ under this condition, the steepest descent algorithm will finally converge to the Wiener solution.

2.4.1 Updating Forms of the LMS Algorithm

The LMS algorithm is considered as a stochastic estimation of the steepest descent algorithm. Since applying the steepest descent algorithm requires the exact measurement of the gradient vector, it might not be possible in reality. The gradient vector has to

be estimated based on the available data. One way is to simply use the instantaneous gradient vector. It is given by

$$\begin{aligned} \frac{\partial \hat{\xi}(\mathbf{w}_n)}{\partial \mathbf{w}_n} &= \frac{\partial e_n^2}{\partial \mathbf{w}_n} \\ &= \frac{\partial \big(d_n - \mathbf{w}_n^T \mathbf{x}_n\big)^2}{\partial \mathbf{w}_n} \\ &= -2e_n \mathbf{x}_n. \end{aligned} \tag{2.12}$$

Using this stochastic gradient vector in the weight update rule (2.9) yields

$$\mathbf{w}_{n+1} = \mathbf{w}_n + 2\mu e_n \mathbf{x}_n. \tag{2.13}$$

This is the updating form of the famous LMS algorithm. The initial weight vector $\mathbf{w}_0$ is set based on the priori knowledge of the system. Typically it is set to zero when there is an absence of such knowledge. For each iteration, the estimation error is computed, the next weight vector is obtained by adding the term $2\mu e_n \mathbf{x}_n$ to the current weight estimate.

Note that in the form of Eq. (2.13), we assume that the input data $\mathbf{x}_n$ and d_n are real signals. To deal with the complex input signals, Widrow *et al* in [4] introduced the complex form of the LMS algorithm as follows:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + 2\mu e_n \mathbf{x}_n^*. \tag{2.14}$$

where $\mathbf{x}_n^*$ is the complex conjugate of $\mathbf{x}_n$, and the output error is defined as $e_n = d_n - \mathbf{w}_n^H \mathbf{x}_n$, where the superscript H denotes the hermitian transpose [1]. The working manners and properties of complex LMS algorithm are very similar to those of real LMS algorithm.

In the literature, the LMS algorithm is also extended such that the weight update takes place every L iterations. That is, the weight vector is not updated until the input vectors are buffered to a certain number L. Such updating mechanism is called the block LMS algorithm with the updating form given by

$$\mathbf{w}_{n+1} = \mathbf{w}_n + 2\mu \mathbf{X}_n \mathbf{e}_n, \tag{2.15}$$

where $\mathbf{X}_n = [\mathbf{x}_n, \mathbf{x}_{n-1}, \cdots, \mathbf{x}_{n-L+1}]$ and $\mathbf{e}_n = [e_n, e_{n-1}, \cdots, e_{n-L+1}]$. The advantage of block LMS algorithm is that the computation of $\mathbf{X}_n \mathbf{e}_n$ can be efficiently implemented

by using the fast Fourier transform (FFT), and the inverse FFT (IFFT) can be used to transform the results back to the original form [35, 36]. However, block LMS algorithm may introduce significant time delay.

One major advantage of the LMS algorithms is its simplicity and ease of implementation. The computational complexity of a regular LMS algorithm is $\mathcal{O}(N)$, which is much more efficient than the steepest descent algorithm or the way of computing the average gradient vector over the whole input space at each iteration. However, the stochastic nature of the LMS algorithm makes it very difficult to analyze its stability and convergence behavior in terms of a single realization. Next we will summarize the main methods of analyzing properties of the LMS algorithm and the results that have been reached.

2.4.2 Independent Theory for LMS Analysis

The update form of the LMS algorithm seems quite simple, but it has a highly nonlinear nature. Assuming that the initial condition is $\mathbf{w}_0 = \mathbf{0}$ and rewriting the weight update equation (2.13) yields

$$\mathbf{w}_n = 2\mu \sum_{i=0}^{n-1} e(i)\mathbf{x}(i). \tag{2.16}$$

It is clearly seen that the current weight estimate depends on all the past $n-1$ input samples. Generally speaking, it is hard to analyze the convergence behavior of the weight estimate $\mathbf{w}_n$ without any assumption on the stochastic input process x_n. To make the analysis possible, one basic assumption for the input process is that it is wide-sense stationary. That means the first and second statistical properties of the input process are invariant to a time shift. In addition, the ensemble averages of a stochastic process are not possible in reality. We expect that the ensemble averages can be replaced by the long-term sample averages or time averages. This requires the input process to be mean and correlation ergodic (see [1, 37]). Based on these assumptions, we then are able to proceed the convergence analysis of the LMS algorithm in average. Taking expectations on both sides of (2.13), we have

$$\begin{aligned} E\left[\mathbf{w}_{n+1}\right] &= E\left[\mathbf{w}_n\right] + 2\mu E\left[e_n\mathbf{x}_n\right] \\ &= E\left[\mathbf{w}_n\right] - 2\mu\left(E\left[\mathbf{x}_n\mathbf{x}_n^T\mathbf{w}_n\right] - E\left[d_n\mathbf{x}_n\right]\right). \end{aligned} \tag{2.17}$$

Since the estimate weight vector $\mathbf{w}_n$ depends on all the past $n-1$ input vectors, the analysis of the third-order expectation in Eq. (2.17) is difficult without further assumptions. To make the problem tractable, a standard simplifying assumption is known as the *independent assumption*[23]. By the independent assumption, we assume the input vector $\mathbf{x}_n$ and the sample of the desired response d_n are statistically independent of all previous input data $\mathbf{x}_{n-1}, \mathbf{x}_{n-2}, \cdots, \mathbf{x}_0$ and $d_{n-1}, d_{n-2}, \cdots, d_0$, but the sample of the desired response d_n can be dependent on the corresponding input vector x_n. Clearly, this assumption may not be satisfied by many practical problems. However, experience with the LMS algorithm has shown the theoretical results obtained based on the independent assumption match very well the simulated results [1, 38, 39]. The statistical analysis of the LMS algorithm based on the independent assumption is called as the *independent theory*. Using the independent assumption for Eq. (2.17), we obtain

$$E\left[\mathbf{w}_{n+1}\right] = E\left[\mathbf{w}_n\right] - 2\mu\left(\mathbf{R}E\left[\mathbf{w}_n\right] - \mathbf{p}\right). \tag{2.18}$$

Compared to Eq. (2.11), it is seen from Eq. (2.18) that the average convergence behavior of the LMS algorithm is equivalent to the steepest descent algorithm.

2.4.3 Stability of the LMS Algorithm

Stabilizing the LMS algorithm is very crucial when it is implemented. In this section, we will see the conditions for the stability of the LMS algorithm. Note that some research work on its stability are still carrying on. For ease of description, let the expectation of the weight error vector be $\mathbf{v}_n \triangleq E\left[\mathbf{w}^o - \mathbf{w}_n\right]$, and the transformed $\mathbf{v}_n$ by $\mathbf{Q}$ be $\mathbf{v}'_n \triangleq \mathbf{Q}^T\mathbf{v}_n$, where the unitary matrix $\mathbf{Q}$ is the eigenvector matrix of $\mathbf{R} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$, with $\mathbf{\Lambda}$ the diagonal eigenvalue matrix. Eq. (2.18) implies that

$$\mathbf{v}'_n = (\mathbf{I} - 2\mu\mathbf{\Lambda})^n\mathbf{v}'_0, \tag{2.19}$$

where $\mathbf{v}'_0$ is the initial value. For each eigenvalue, λ_i, of $\mathbf{R}$, there is a corresponding mode

$$v'_n(i) = (1 - 2\mu\lambda_i)^n v'_n(0). \tag{2.20}$$

Therefore, there are as many modes as the number of unique eigenvalues of $\mathbf{R}$. As long as the step-size parameter μ fulfills the condition $0 < \mu < \frac{1}{\lambda_{max}}$, all the elements, $v'_n(i)$, relax to zero. Thus, the LMS algorithm converges to the Wiener solution $\mathbf{w}^o$ in mean.

However, proving convergence in the mean is not sufficient since the variance of the weight estimate can be unbounded due to the stochastic property of the LMS algorithm. Therefore, the mean-square convergence of the algorithm has also been studied using the independent assumption and white Gaussian input data in [40]. It was shown that a sufficient condition is given by

$$0 < \mu < \frac{1}{3\mathrm{Tr}(\mathbf{R})}. \tag{2.21}$$

Though it is enough to stabilize the LMS algorithm in practice with this sufficient condition, for scientific study it is expected to find the necessary and sufficient conditions for convergence of the algorithm. Efforts have been made by many researchers for many years and likely to remain so for years to come. Necessary and sufficient conditions for the exponential stability of the algorithm with probability one have been studied by Bitmead [41, 42], Guo et al. [43] and Solo [44]. However, these researchers derive only very general statements in the sense "for a sufficient low step-size parameter μ, the LMS algorithm is convergent," but do not present concrete bounds on μ. From a filter design perspective, a concrete bound on μ is highly desirable. When the LMS algorithm uses a number of weights that tends towards infinity, the study carried out by Butterweck [45] first rigorously derived result for a necessary stability bound on the step-size parameter μ. That is, for stationary inputs and desired response the condition

$$0 < \mu < \frac{1}{NS_{max}} \tag{2.22}$$

defines a necessary condition for the mean square stability of the LMS algorithm. To this day, finding a general set of necessary and sufficient conditions for mean square convergence of the LMS algorithm is still an open problem [1].

2.4.4 Transient Behavior of the LMS Algorithm

If the step-size parameter μ is chosen such that the LMS algorithm is mean square stable, the transient behavior of the algorithm can be described by using the ensemble-average

learning curves, usually the mean-square error (MSE) learning curve or the mean-square deviation (MSD) learning curve. Next, we describe the analytic results on the MSE learning curve.

Assume the input data $\{\mathbf{x}_n, d_n\}$ fit the multiple linear regression model

$$d_n = \mathbf{x}_n^T \mathbf{w}^o + e_n^o \tag{2.23}$$

where e_n^o accounts for measurement noise, which is zero mean and the variance of σ_v^2, and is assumed to be statistically independent of $\mathbf{x}_n$. Thus the filter output error e_n is expressed as

$$\begin{aligned} e_n &= d_n - \mathbf{w}_n^T \mathbf{x}_n \\ &= \mathbf{x}_n^T \mathbf{w}^o + e_n^o - \mathbf{w}_n^T \mathbf{x}_n \\ &= e_n^o + \boldsymbol{\epsilon}_n^T \mathbf{x}_n \end{aligned} \tag{2.24}$$

where $\boldsymbol{\epsilon}_n = \mathbf{w}^o - \mathbf{w}_n$ denotes the weight error vector. Using Eq. (2.24) into the MSE as described in Eq. (2.2) yields

$$\begin{aligned} \xi_n &= E\left[(e_n^o + \boldsymbol{\epsilon}_n^T \mathbf{x}_n)^2\right] \\ &= E[(e_n^o)^2)] + 2E[e_n^o \boldsymbol{\epsilon}_n^T \mathbf{x}_n] + E[\boldsymbol{\epsilon}_n^T \mathbf{x}_n \mathbf{x}_n^T \boldsymbol{\epsilon}_n] \\ &= \xi_{min} + \mathrm{Tr}\left\{E[\boldsymbol{\epsilon}_n \boldsymbol{\epsilon}_n^T]\mathbf{R}\right\} \end{aligned} \tag{2.25}$$

where ξ_{min} is the minimum achievable MSE of the LMS algorithm, which is equal to variance of the measurement noise. Note that in the derivation of Eq. (2.25), the matrix property $\mathrm{Tr}[\mathbf{A} \cdot \mathbf{B}] = \mathrm{Tr}[\mathbf{B} \cdot \mathbf{A}]$ and the *principle of orthogonality* [1] have been applied.

It is seen from Eq. (2.25) that finding the MSE learning curve involves analyzing covariance of the weight-error vector $\boldsymbol{\epsilon}_n$. Subtracting the optimal weight vector $\mathbf{w}^o$ from both sides of Eq. (2.13) yields the following recursive form of the weight-error vector

$$\boldsymbol{\epsilon}_{n+1} = [\mathbf{I} - 2\mu \mathbf{x}_n \mathbf{x}_n^T]\boldsymbol{\epsilon}_n - 2\mu e_n^o \mathbf{x}_n. \tag{2.26}$$

Eq. (2.26) is a *stochastic difference equation* of the weight-error vector $\boldsymbol{\epsilon}_n$, where the characteristic feature of the system matrix is $[\mathbf{I} - 2\mu \mathbf{x}_n \mathbf{x}_n^T]$. To study the MSE convergence behavior of such a stochastic algorithm, the *independent theory* or the *small-step-size*

theory [1] with Butterweck's iterative procedure have been proposed. These two theories lead to the same steady-state solution of the LMS algorithm. By comparison, the *small-step-size theory* uses loose assumption on the input statistics. We next briefly introduce derivation of the MSE of the LMS algorithm based on these two theories.

Using the independent assumption and Eq. (2.26), the covariance of the weight-error vector, $\text{cov}[\boldsymbol{\epsilon}_{n+1}] = E[\boldsymbol{\epsilon}_{n+1}\boldsymbol{\epsilon}_{n+1}^T]$, is given by

$$\begin{aligned}\text{cov}[\boldsymbol{\epsilon}_{n+1}] &= E\{[\mathbf{I}-2\mu\mathbf{x}_n\mathbf{x}_n^T]\boldsymbol{\epsilon}_n\boldsymbol{\epsilon}_n^T[\mathbf{I}-2\mu\mathbf{x}_n\mathbf{x}_n^T] \\ &\quad -2[\mathbf{I}-2\mu\mathbf{x}_n\mathbf{x}_n^T]\boldsymbol{\epsilon}_n\mu e_n^o\mathbf{x}_n^T \\ &\quad -2\mu e_n^o\mathbf{x}_n\boldsymbol{\epsilon}_n^T[\mathbf{I}-2\mu\mathbf{x}_n\mathbf{x}_n^T]^T \\ &\quad +4\mu^2(e_n^o)^2\mathbf{x}_n\mathbf{x}_n^T\}. \end{aligned} \tag{2.27}$$

By considering e_n^o independent of $\boldsymbol{\epsilon}_n$ and orthogonal to $\mathbf{x}_n$, the second and third terms of the right-hand side of Eq. (2.27) can be eliminated. In this case, Eq. (2.27) can be rewritten as

$$\begin{aligned}\text{cov}[\boldsymbol{\epsilon}_{n+1}] &= \text{cov}[\boldsymbol{\epsilon}_n]-2\mu\mathbf{R}\text{cov}[\boldsymbol{\epsilon}_n]-2\mu\text{cov}[\boldsymbol{\epsilon}_n]\mathbf{R} \\ &\quad +4\mu^2E\{\mathbf{x}_n\mathbf{x}_n^T\boldsymbol{\epsilon}_n\boldsymbol{\epsilon}_n^T\mathbf{x}_n\mathbf{x}_n^T\}+4\mu^2\sigma_v^2\mathbf{R}. \end{aligned} \tag{2.28}$$

Using the Gaussian moment factoring theorem[1] to simplify the fourth order term in Eq. (2.28) yields

$$\begin{aligned}\text{cov}[\boldsymbol{\epsilon}_{n+1}] &= \text{cov}[\boldsymbol{\epsilon}_n]-2\mu\mathbf{R}\text{cov}[\boldsymbol{\epsilon}_n]-2\mu\text{cov}[\boldsymbol{\epsilon}_n]\mathbf{R} \\ &\quad +8\mu^2\mathbf{R}\text{cov}[\boldsymbol{\epsilon}_n]\mathbf{R}+4\mu^2\mathbf{R}\text{Tr}\{\mathbf{R}\text{cov}[\boldsymbol{\epsilon}_n]\}+4\mu^2\sigma_v^2\mathbf{R}. \end{aligned} \tag{2.29}$$

Further simplifying Eq. (2.29) by premultiplying and postmultiplying it by $\mathbf{Q}^T$ and $\mathbf{Q}$, respectively, yields

$$\begin{aligned}\text{cov}[\boldsymbol{\epsilon}'_{n+1}] &= \text{cov}[\boldsymbol{\epsilon}'_n]-2\mu\boldsymbol{\Lambda}\text{cov}[\boldsymbol{\epsilon}'_n]-2\mu\text{cov}[\boldsymbol{\epsilon}'_n]\boldsymbol{\Lambda} \\ &\quad +8\mu^2\boldsymbol{\Lambda}\text{cov}[\boldsymbol{\epsilon}'_n]\boldsymbol{\Lambda}+4\mu^2\boldsymbol{\Lambda}\text{Tr}\{\boldsymbol{\Lambda}\text{cov}[\boldsymbol{\epsilon}'_n]\}+4\mu^2\sigma_v^2\boldsymbol{\Lambda}. \end{aligned} \tag{2.30}$$

[1]Gaussian moment factoring theorem: If the zero-mean random variables $\mathbf{X}_i, i = 1, 2, 3, 4$ are jointly Gaussian, then

$$E[\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3\mathbf{X}_4] = E[\mathbf{X}_1\mathbf{X}_2]E[\mathbf{X}_3\mathbf{X}_4]+E[\mathbf{X}_1\mathbf{X}_3]E[\mathbf{X}_2\mathbf{X}_4]+E[\mathbf{X}_1\mathbf{X}_4]E[\mathbf{X}_2\mathbf{X}_3].$$

The proof is given in [46].

where $\text{cov}[\boldsymbol{\epsilon}'_n] = \mathbf{Q}^T \text{cov}[\boldsymbol{\epsilon}_n]\mathbf{Q}$. Let $\mathbf{v}_n$ be the vector consisting of the diagonal elements of $\text{cov}[\boldsymbol{\epsilon}'_n]$ and $\boldsymbol{\lambda}$ be the vector consisting of the eigenvalues of $\mathbf{R}$, from Eq. (2.29) it follows

$$\mathbf{v}_{n+1} = (\mathbf{I} - 4\mu\boldsymbol{\Lambda} + 8\mu^2\boldsymbol{\Lambda}^2 + 4\mu^2\boldsymbol{\lambda}\boldsymbol{\lambda}^T)\mathbf{v}_n + 4\mu^2\sigma_v^2\boldsymbol{\lambda}. \tag{2.31}$$

With Eq. (2.31), the MSE of Eq. (2.25) can be rewritten as

$$\xi_n = \xi_{min} + \mathbf{v}_n^T\boldsymbol{\Lambda}\mathbf{v}_n. \tag{2.32}$$

Writing the MSE of Eq. (2.32) in terms of the elements $v_n(i)$ and λ_i for $i = 1, 2, \cdots, N$, one finds

$$\xi_n = \xi_{min} + \sum_{i=1}^{N} |v_n(i)|^2\lambda_i. \tag{2.33}$$

where $v_n(i)$ is derived from (2.31) as follows

$$v_{n+1}(i) = (1 - 4\mu\lambda_i + 8\mu^2\lambda_i^2)v_n(i) + 4\mu^2\lambda_i\sum_{j=1}^{N}\lambda_j v_n(j) + 4\mu^2\sigma_v^2\lambda_i. \tag{2.34}$$

The MSE learning curve of LMS is defined by Eq. (2.33) and Eq. (2.34) based on the *independent theory*. Unfortunately, it is not a close form. Using the *small-step-size theory* for derivation the MSE learning curve leads to a much simpler problem.

The Butterweck's iterative procedure expresses the solution of Eq. (2.26) as the sum of partial functions

$$\boldsymbol{\epsilon}_n = \boldsymbol{\epsilon}_n(0) + \boldsymbol{\epsilon}_n(1) + \boldsymbol{\epsilon}_n(2) + \cdots, \tag{2.35}$$

where $\boldsymbol{\epsilon}_n(0)$ is called as zero-order solution of Eq. (2.26) as μ tends towards zero, and $\boldsymbol{\epsilon}_n(1), \boldsymbol{\epsilon}_n(2), \ldots$ are higher order corrections to $\boldsymbol{\epsilon}_n(0)$ for $\mu > 0$. Substituting Eq. (2.35) into Eq. (2.26) yields

$$\begin{aligned}\boldsymbol{\epsilon}_{n+1} &= (\mathbf{I} - 2\mu\mathbf{R})[\boldsymbol{\epsilon}_n(0) + \boldsymbol{\epsilon}_n(1) + \boldsymbol{\epsilon}_n(2) + \cdots] \\ &\quad -2\mu(\mathbf{x}_n\mathbf{x}_n^T - \mathbf{R})[\boldsymbol{\epsilon}_n(0) + \boldsymbol{\epsilon}_n(1) + \boldsymbol{\epsilon}_n(2) + \cdots] - \mu e_n^o\mathbf{x}_n,\end{aligned} \tag{2.36}$$

from which the following coupled difference equations can be deduced

$$\boldsymbol{\epsilon}_{n+1}(i) = (\mathbf{I} - 2\mu\mathbf{R})\boldsymbol{\epsilon}_n(i) + \mathbf{f}_n(i), \;\; i = 0, 1, 2, \ldots, \tag{2.37}$$

where the driving force $\mathbf{f}_n(i)$ for the difference equation (2.37) is defined by

$$\mathbf{f}_n(i) = \begin{cases} -\mu e_n^o \mathbf{x}_n, & i = 0 \\ -\mu(\mathbf{x}_n\mathbf{x}_n^T - \mathbf{R})\boldsymbol{\epsilon}_n(i-1), & i = 1, 2, \ldots \end{cases} \tag{2.38}$$

Then the covariance matrix of the weight-error vector $\boldsymbol{\epsilon}_n$ can be expressed as follows

$$\mathbf{M}_n = \mathbf{M}_n(0) + 2\mu\mathbf{M}_n(1) + 4\mu^2\mathbf{M}_n(2) + \cdots, \tag{2.39}$$

where $\mathbf{M}_n = \text{cov}[\boldsymbol{\epsilon}_n]$, and the various matrix coefficients are defined as follows

$$2^j\mu^j\mathbf{M}_n(j) = \left\{ \begin{array}{ll} E[\boldsymbol{\epsilon}_n(0)\boldsymbol{\epsilon}_n(0)^T], & j = 0 \\ \sum_i \sum_k E[\boldsymbol{\epsilon}_n(i)\boldsymbol{\epsilon}_n(k)^T], & \text{for all } (i,k) \geq 0 \\ & \text{such that } i + k = 2j - 1, 2j \end{array} \right\}. \tag{2.40}$$

The *small-step-size theory* assumes that the step-size parameter μ is very small, so the LMS filter acts as a lowpass filter with a low cutoff frequency. Under this assumption, the terms $\boldsymbol{\epsilon}_n(0)$ and $\mathbf{M}_n(0)$ can be used to approximate the actual $\boldsymbol{\epsilon}_n$ and $\mathbf{M}_n$. Butterweck's iterative procedure reduces to the following equation

$$\boldsymbol{\epsilon}_{n+1}(0) = (\mathbf{I} - 2\mu\mathbf{R})\boldsymbol{\epsilon}_n(0) - 2\mu e_n^o \mathbf{x}_n. \tag{2.41}$$

Define the transformed $\boldsymbol{\epsilon}_n(0)$ by $\mathbf{Q}$ as $\mathbf{v}_n = \mathbf{Q}^T\boldsymbol{\epsilon}_n(0)$, Eq. (2.41) is transformed into the form

$$\mathbf{v}_{n+1} = (\mathbf{I} - 2\mu\boldsymbol{\Lambda})\mathbf{v}_n - \boldsymbol{\phi}_n. \tag{2.42}$$

where $\boldsymbol{\phi}_n \triangleq 2\mu e_n^o \mathbf{Q}^T\mathbf{x}_n$. In particular, for the ith natural mode of the LMS algorithm, we have

$$v_{n+1}(i) = (\mathbf{I} - 2\mu\lambda_i)v_n(i) - \phi_n(i). \tag{2.43}$$

Eq. (2.43) defines *Brownian motion* of the LMS algorithm. Under the assumption that the measurement noise e_n^o is white noise process and statistically independent of the input vector $\mathbf{x}_n$, one can easily finds

$$E[\boldsymbol{\phi}_n\boldsymbol{\phi}_n^T] = 4\mu^2\xi_{min}\boldsymbol{\Lambda}. \tag{2.44}$$

Let $v_0(i)$ denote the initial value of $v_n(i)$. Eq. (2.43) can be solved to get

$$v_n(i) = (1 - 2\mu\lambda_i)^n v_0(i) + \sum_{k=0}^{n-1}(1 - 2\mu\lambda_i)^{n-1-k}\phi_i(n) \tag{2.45}$$

Since the MSE function ξ_n can be approximated as

$$\begin{aligned}\xi_n &\approx \xi_{min} + \mathrm{Tr}[\mathbf{R}\mathbf{M}_0] \\ &= \xi_{min} + \mathrm{Tr}[\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T\mathbf{M}_0] \\ &= \xi_{min} + \mathrm{Tr}[\mathbf{\Lambda}\mathbf{Q}^T\mathbf{M}_0\mathbf{Q}] \\ &= \xi_{min} + E[\mathbf{v}_n^T\mathbf{\Lambda}\mathbf{v}_n] \\ &= \xi_{min} + \sum_{i=1}^{N}\lambda_i E[v_n^2(i)]. \end{aligned} \tag{2.46}$$

Substituting Eq. (2.45) into Eq. (2.46) and using Eq. (2.44) yields

$$\xi_n = \xi_{min} + \mu\xi_{min}\sum_{i=0}^{N}\frac{\lambda_i}{1-\mu\lambda_i} + \sum_{i=0}^{N}\lambda_i\left(v_i^2(0) - \frac{\mu\xi_{min}}{1-\mu\lambda_i}\right)(1-2\mu\lambda_i)^{2n} \tag{2.47}$$

Eq. (2.46) is a close form for the MSE learning curve of the LMS algorithm for small μ. The MSE does not exactly converge to ξ_{min} due to the oscillations. To characterize this property in steady state, a new measurement called as misadjustment is defined as

$$\mathcal{M} = \frac{\xi_{ex}(\infty)}{\xi_{min}} \tag{2.48}$$

where $\xi_{ex}(\infty) = \xi(\infty) - \xi_{min}$ denotes the steady-state value of the excess mean-square error. The misadjustment defined as a ratio form provides a measure of how close an adaptive algorithm is to optimality in the mean-square-error sense. The smaller the misadjustment, the more accurate of the steady-state solution the algorithm. For small μ, we can assume $\mu\lambda_i \ll 1$. As k tends towards infinity, using Eq. (2.33) or Eq. (2.47) yields

$$\xi(\infty) \approx \xi_{min} + \mu\xi_{min}\mathrm{Tr}(\mathbf{R}). \tag{2.49}$$

The misadjustment of the LMS algorithm can thus be written for small μ as

$$\mathcal{M} = \mu\mathrm{Tr}(\mathbf{R}). \tag{2.50}$$

2.4.5 Properties of the LMS Algorithm

In this section, some of the main positive and negative properties of the LMS algorithm are summarized. The first property is that its convergence speed is highly affected by the

eigenvalue spread [2] of the input autocorrelation matrix, $\mathbf{R}$. Eq. (2.47) shows that the evolution of the mean-square error is governed by the exponential factor $(1-2\mu\lambda_i)^{2n}$ for all natural modes $i = 1, 2, \cdots, N$. If the time constant τ_i is used to define the number of iterations required for the amplitude of the ith natural mode $\xi_n(i)$ to decay to $1/e$ of its initial value $\xi_i(0)$, where e is the base of the natural logarithm, we can approximate τ_i for small μ as

$$\tau_i \approx \frac{1}{4\mu\lambda_i}. \tag{2.51}$$

Therefore, the LMS algorithm has N time constants corresponding to its N eigenvalues. The exponential factors corresponding to large eigenvalues decrease to zero fast, whereas small eigenvalues slow down the overall convergence. Input signals with high eigenvalue spread will therefore results in poor convergence performance except when the algorithm starts at the initial weight vector that lies along the eigenvector corresponding to the largest eigenvalue, which is the only one mode with very fast convergence speed. In reality, it is impossible to find such initial weight vector without knowledge of the second order statistics of the input process.

The second property is that the step-size parameter, μ, governs the convergence speed of the LMS algorithm as seen from Eq. (2.51). Large μ's speed up the convergence of the algorithm, but large μ's also lower the precision of the steady-state solution of the algorithm as seen from Eq. (2.50). Therefore, the algorithm has a tradeoff between the convergence speed and misadjustment. Choosing the right constant value for μ is an important and difficult task. In Chapter 3 of the thesis, we will deal with this tradeoff problem by using a variable step-size approach. Moreover, the algorithm is sensitive to the scaling of its input when choosing a step-size μ to guarantee stability as seen from Eq. (2.21) and Eq. (2.22). Normalizing the step-size parameter μ with power of the input can significantly reduce such sensitivity, which results in the NLMS algorithm. We will discuss this algorithm in Chapter 4.

Besides the *negative* properties mentioned above, the LMS algorithm does have many *positive* properties, which make the algorithm popular. One is its simplicity of updating form and ease of implementation. Its weight update requires only $\mathcal{O}(N)$ computations

[2] The eigenvalue spread of a matrix is defined as a ratio of its maximum eigenvalue to its minimum eigenvalue.

per iteration. The LMS algorithm has no memory. It therefore can more easily track a time-varying solution than the accumulative algorithms like the RLS-type algorithms. Moreover, it is robust to error propagation in limited precision implementations [47] and numerical stability problem. In fact, it has been shown that the LMS algorithm is H^∞ optimal, i.e., that it minimizes the maximum energy gain from the disturbances to the predicted errors [48]. In [49], the performance of the LMS algorithm was compared with that of the LMS/Newton algorithm [50] in terms of the *excess error energy*, which is defined as the summation of the total excess MSE. It has been shown that the excess error energies of the LMS algorithm and LMS/Newton algorithm are the same when averaged over random initial conditions. By this definition, this implies that the LMS algorithm and LMS/Newton algorithm have the same convergence time when averaged over random initial conditions.

2.5 The RLS Algorithm

Although the LMS algorithm has many advantageous properties, the major complaint about its slow convergence may limit the uses in applications where fast convergence is critical. With this respective, the exact least squares algorithms will be more attractive.

2.5.1 Updating Form of the RLS Algorithm

Unlike the LMS algorithm minimizes the instantaneous square error per iteration only based on the current input data, the RLS algorithm finds an exact least squares solution per iteration based on all past data[1]. To see this, the cost function of the least squares error is shown as follows

$$J_n = \frac{1}{2}\sum_{k=1}^{n} \gamma^{n-k} e_k^2. \tag{2.52}$$

where $\gamma \in [0, 1]$ is the exponential weighting factor, e_k is the output error. Minimizing the cost function of Eq. (2.52) is equivalent to finding the derivative of Eq. (2.52) with respect to $\mathbf{w}_n$. Proceeding the derivative yields

$$\frac{\partial J_n}{\partial \mathbf{w}_n} = \sum_{k=1}^{n} \gamma^{n-k} \mathbf{x}_k \mathbf{x}_k^T \mathbf{w}_n - \sum_{k=1}^{n} \gamma^{n-k} \mathbf{x}_k d_k. \tag{2.53}$$

The exponentially decaying data window scheme of estimating the autocorrelation matrix $\mathbf{R}$ and the cross-correlation $\mathbf{p}$ leads to the estimated $\mathbf{R}_n$ and $\mathbf{p}_n$ as

$$\mathbf{R}_n = \sum_{k=1}^{n} \gamma^{n-k} \mathbf{x}_k \mathbf{x}_k^T \tag{2.54}$$

$$\mathbf{p}_n = \sum_{k=1}^{n} \gamma^{n-k} \mathbf{x}_k d_k. \tag{2.55}$$

Substituting the estimated $\mathbf{R}_n$ given in Eq. (2.54) and $\mathbf{p}_n$ in Eq. (2.55) into Eq. (2.53) and setting the result to zero, the resulting equation is known as the normal equation with the matrix form given by

$$\mathbf{R}_n \mathbf{w}_n = \mathbf{p}_n. \tag{2.56}$$

To guarantee the nonsingularity of $\mathbf{R}_n$ in finding the solution of the normal equation at the initial stages, a regularizing term $\delta\gamma^n \|\mathbf{w}_n\|^2$ is commonly added into the cost function of Eq. (2.52), where δ is the positive regularization parameter. As the effect, the estimated $\mathbf{R}_n$ is added with the term $\delta\gamma^n \mathbf{I}$, where $\mathbf{I}$ is the $N \times N$ identity matrix. Solving the normal equation gives the least squares solution $\mathbf{w}_n = \mathbf{R}_n^{-1}\mathbf{p}_n$, which is thus the best to-date approximation to the Wiener solution. To develop a recursive equation for updating the least squares solution, the estimated $\mathbf{R}_n$ and $\mathbf{p}_n$ are expressed in recursion forms as follows

$$\mathbf{R}_n = \gamma \mathbf{R}_{n-1} + \mathbf{x}_n \mathbf{x}_n^T \tag{2.57}$$

$$\mathbf{p}_n = \gamma \mathbf{p}_{n-1} + \mathbf{x}_n d_n. \tag{2.58}$$

Substituting the recursion forms of $\mathbf{R}_n$ and $\mathbf{p}_n$ into the least squares solution $\mathbf{w}_n = \mathbf{R}_n^{-1}\mathbf{p}_n$ yields

$$\mathbf{w}_n = (\gamma \mathbf{R}_{n-1} + \mathbf{x}_n \mathbf{x}_n^T)^{-1}(\gamma \mathbf{p}_{n-1} + \mathbf{x}_n d_n). \tag{2.59}$$

Applying the *matrix inversion lemma*[3] to Eq. (2.59), we have

$$\begin{aligned} \mathbf{w}_n &= \left(\gamma^{-1}\mathbf{R}_{n-1}^{-1} - \frac{\gamma^{-2}\mathbf{R}_{n-1}^{-1}\mathbf{x}_n\mathbf{x}_n^T\mathbf{R}_{n-1}^{-1}}{1+\gamma^{-1}\mathbf{x}_n^T\mathbf{R}_{n-1}^{-1}\mathbf{x}_n}\right)(\gamma\mathbf{p}_{n-1} + \mathbf{x}_n d_n) \\ &= \mathbf{R}_{n-1}^{-1}\mathbf{p}_{n-1} + \frac{\gamma^{-1}\mathbf{R}_{n-1}^{-1}\mathbf{x}_n(d_n - \mathbf{x}_n^T\mathbf{R}_{n-1}^{-1}\mathbf{p}_{n-1})}{1+\gamma^{-1}\mathbf{x}_n^T\mathbf{R}_{n-1}^{-1}\mathbf{x}_n}. \end{aligned} \tag{2.60}$$

[3]Matrix Inversion Lemma [51]: Let $\mathbf{A}$ and $\mathbf{B}$ be two positive-definite $N \times N$ matrices, $\mathbf{C}$ a $N \times M$ matrix, and $\mathbf{D}$ a positive definite $M \times M$ matrix. If they are related by

$$\mathbf{A} = \mathbf{B} + \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^T,$$

Note that $\mathbf{w}_{n-1} = \mathbf{R}_{n-1}^{-1}\mathbf{p}_{n-1}$. Defining the priori error as $e'_n = d_n - \mathbf{x}_n^T\mathbf{w}_{n-1}$, it follows from Eq. (2.60) that

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mathbf{G}_n\mathbf{x}_n e'_n. \tag{2.61}$$

where the gain matrix $\mathbf{G}_n$ is defined as

$$\mathbf{G}_n = \frac{\gamma^{-1}\mathbf{R}_{n-1}^{-1}}{1 + \gamma^{-1}\mathbf{x}_n^T\mathbf{R}_{n-1}^{-1}\mathbf{x}_n}. \tag{2.62}$$

Eq. (2.61) and Eq. (2.62) are the updating forms of the RLS algorithm. The initial weight vector $\mathbf{w}(0)$ are typically set to zero.

2.5.2 Properties of the RLS Algorithm

Comparing with the LMS algorithm, the RLS algorithm uses a matrix gain, $\mathbf{G}_n$, such that each convergence mode has a its own learning rate. The effectiveness of this matrix gain is evidenced from that in the updating process the input data are decorrelated by the data-dependent matrix $\mathbf{R}_n^{-1}$. To see this, the updating form of the RLS algorithm is reformed as

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu_n\mathbf{R}_{n-1}^{-1}e'_n\mathbf{x}_n, \tag{2.63}$$

with

$$\mu_n = \frac{\gamma^{-1}}{1 + \gamma^{-1}\mathbf{x}_n^T\mathbf{R}_{n-1}^{-1}\mathbf{x}_n}. \tag{2.64}$$

Intuitively speaking, this decorrelation reduces eigenvalue spread of the signals that are applied for the adaptation of the weight vector. From the analysis of the LMS algorithm, we have known that a smaller input eigenvalue spread means a faster convergence speed. This may justify the fast convergence property of the RLS algorithm. However, the operation of multiplying $\mathbf{x}_n$ by $\mathbf{R}_{n-1}^{-1}$ can unfortunately suffer from the numerical stability problem when the filter uses more weights than necessary, where the matrix $\mathbf{R}_{n-1}$ becomes ill-conditioned.

---

then the inverse of the matrix $\mathbf{A}$ is

$$\mathbf{A}^{-1} = \mathbf{B}^{-1} - \mathbf{B}^{-1}\mathbf{C}(\mathbf{D} + \mathbf{C}^T\mathbf{B}^{-1}\mathbf{C})^{-1}\mathbf{C}^T\mathbf{B}^{-1}.$$

Another aspect of comparing with the LMS algorithm is that the RLS algorithm theoretically converges to the Wiener solution if γ is chosen as one. That is, the misadjustment in RLS can asymptotically converge to zero. This is due to that the RLS algorithm obtains the solution of the normal equation which has the same form as the Wiener solution. As the statistical estimates $\mathbf{R}_n$ and $\mathbf{p}_n$ asymptotically converge to their true values, the solution of the RLS algorithm also asymptotically converges to the Wiener solution. Note that the convergence speed of the RLS algorithm is roughly independent of the exponential factor γ.

Because the RLS algorithm uses the accumulative data, it has poor tracking capabilities in time-varying environments where the input signal has statistic changes [1]. The past history of the input data has high effects on the estimates of the current statistics. As a counterpart, the LMS algorithm has a better tracking property due to its stochastic nature. Applying the second order statistics of the input data has a significant increase in the computational complexity of the RLS algorithm. The computational complexity of the RLS algorithm is $\mathcal{O}(N^2)$ for each weight update. Note that to reduce the computational complexity of the RLS algorithm, many fast implementations have been studied in the literature. The so-called *lattice structure* has been successfully used to implement fast RLS algorithm[52, 53]. The nature of the lattice structure allows the modified RLS algorithm to achieve the computational cost of $\mathbf{O}(N)$ equivalent to the LMS algorithm while maintaining fast convergence property. Moreover, the stability of the RLS algorithm can be monitored with this structure. It is, however, less robust in terms of the finite arithmetic operation as well as complication of the implementation compared with the LMS algorithm.

2.6 Concluding Remarks

In this chapter, two major families of adaptive algorithms, i.e., the LMS algorithms and the RLS algorithms, have been introduced. Their derivations are considered from the unconstraint minimization problem, i.e., the minimization of the instantaneous square error or exponentially accumulated square error. Properties of the LMS algorithm reveals its simplicity, robustness and good tracking capabilities as well as its slow convergence

and sensitivity to the step-size parameter. Properties of the RLS algorithm show its fast convergence independent of the input eigenvalue spread, but also show its high computational complexity, numerical instability and poor tracking capabilities. The LMS algorithm and the RLS algorithm are extreme implementations in terms of the simplicity and convergence. The techniques of improving the convergence of the LMS algorithm while maintaining the low computational complexity or with moderate increase, and of reducing the computational complexity and improving the numerical stability of the classic RLS algorithm while maintaining its fast convergence are to be studied in the following chapters.

Chapter 3

Variable Step-Size LMS Algorithms[1]

3.1 Limitations of Existing Variable Step-size Algorithms

From the discussion of Chapter 2, it has been shown that the performance of the fixed step-size LMS algorithm is highly dependent on the choice of the step-size. That is, the parameter of step-size governs the convergence speed and steady-state estimation error or the excess mean-square error (EMSE). Specifically, the rate of convergence is proportional to the step-size, whereas the steady-state EMSE is inversely proportional to the step-size [1, 54]. When the fixed step-size LMS algorithm is used, the tradeoff between the rate of convergence and the steady-state EMSE must be considered. This problem has motivated many researchers to use a time-varying step-size approach, which speeds up the convergence using large step-size at the initial stage, and reduces misadjustment with small step-size in steady-state.

Numerous variable step-size adjustment schemes have been proposed [9, 10, 40, 55–66] to meet the conflicting requirements of fast convergence speed and low steady-state EMSE or misadjustment. All these schemes are based on a specific criterion. Different

[1]The results of this chapter were first published in Signal Processing, vol. 89, no. 1, pp. 67-76, January 2009.

criteria usually lead to distinct performance improvements and limitations. For instance, the schemes based on the squared instantaneous error [57], on the time-averaged estimate of the error autocorrelation at adjacent time [58], on the sign of the instantaneous error [61], and on the estimate of gradient power [56] have low computational costs, but their steady-state performance are highly dependent on the power of measurement noise. As a result, these algorithms suffer from the applications where low signal-to-noise ratios (SNR) present. The schemes introduced in [63, 64] were shown to be largely insensitive to the power of measurement noise, but the increases in computational cost are significant compared to the fixed step-size LMS algorithm for real-time applications. Recently, it was demonstrated in [67] that the variable step-size (VSS) LMS algorithm developed by Kwong and Johnston in [57] is probably the best low-complexity variable step-size LMS algorithm available in the literature if well designed, except for its limited robustness to the measurement noise power. More recently, a modified version of the VSS algorithm called noise resilient variable step-size (NRVSS) LMS algorithm was presented in [65, 66]. Compared with the VSS algorithm, the NRVSS algorithm was shown to achieve the same transient behavior and less sensitivity to the measurement noise for small increase in the computational cost. Since the optimal design of the NRVSS algorithm requires the reference signal power, it is specially indicated for the applications where estimation of the input signal power is possible.

In this chapter, we propose to use the quotient form of filtered versions of the error statistics, where the error arises from the mismatches between the desired response and the filter output. The filtered estimates of the error are based on exponential windows, applying different decaying factors for the estimations in the numerator and denominator. Compared to the existing modified variable step-size LMS algorithms, the new developed algorithm is preferred in terms of the following four merits: 1) it achieves lower steady-state EMSE or misadjustment, 2) the performance of the new algorithm is less affected by the measurement noise power, 3) it tracks faster in a time-varying system, 4) it converges faster when applied for noise cancellation.

In following sections, we will formulate the new proposed algorithm, study the mean and mean-square convergence, the steady-state performance and the mean step-size behavior of the new algorithm under a slow time-varying system model. We also demonstrate the applications of system identification and noise cancellation.

3.2 Formulation of the Proposed Variable Step-Size LMS Algorithm

In this section, the VSS algorithm [57] and the NRVSS algorithm [65, 66] are introduced first, and then the proposed more noise resilient variable step-size algorithm (MRVSS) is formulated.

For ease of description, we use a general adaptive filtering problem of system identification as depicted in Figure 2.1, where x_n is input signal, $\mathbf{w}^o$ is the vector consisting of the coefficients of the unknown system, $\mathbf{w}_n$ is the vector consisting of the coefficients of the adaptive filter that models the unknown system, e_n^o denotes the measurement noise, which is assumed to be zero mean and statistically independent of x_n, and d_n represents the desired response consisting of system output plus the measurement noise. A variable step-size LMS algorithm uses the following weight vector update equation:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu_n e_n \mathbf{x}_n, \tag{3.1}$$

where μ_n is a scalar variable step-size, e_n is the instantaneous estimation error given by $e_n = d_n - \mathbf{w}_n^T \mathbf{x}_n$, the superscript T denotes matrix transposition, and $\mathbf{x}_n$ is the tap-delayed input vector.

3.2.1 Description of VSS and NRVSS Algorithms

The variable step-size adjustment schemes presented in [57] is shown as follows:

$$\text{VSS:} \quad \mu_{n+1} = \alpha_{VSS}\mu_n + \gamma_{VSS} e_n^2, \tag{3.2}$$

where α_{VSS} and γ_{VSS} are the constant parameters, which are chosen as $0 < \alpha_{VSS} < 1$ and $\gamma_{VSS} > 0$. To ensure the stability and desired steady-state performance of the algorithm, the variable step-size is usually bounded as $\mu_{min} \leq \mu_n \leq \mu_{max}$. The extensive studies in [57, 65–67] show that the VSS algorithm can achieve the mean step-size behavior that is the closest to the optimal step-size behavior for stationary environments. The main complaint about the VSS algorithm goes to its intrinsic high sensitivity to the measurement noise power. This high sensitivity could be explained by the analytic results

of mean step-size and steady-state misadjustment derived in [57] where n is assumed to equal to infinity, which are listed as follows

$$\text{VSS:} \quad E[\mu_\infty] = \frac{\gamma_{VSS}\left(\xi_{min} + \xi_{ex}(\infty)\right)}{1-\alpha_{VSS}}, \tag{3.3}$$

$$\mathcal{M}_{VSS} \approx \frac{\gamma_{VSS}(3-\alpha_{VSS})\left(\xi_{min} + \xi_{ex}(\infty)\right)}{1-\alpha_{VSS}^2}\text{tr}(\mathbf{R}), \tag{3.4}$$

where ξ_{min} denotes the minimum MSE, which is equal to the power of measurement noise, $\xi_{ex}(\infty)$ denotes the steady-state EMSE, which is given by $\xi_{ex}(\infty) = \text{tr}\{\mathbf{R}\mathbf{K}_{VSS}(\infty)\}$ with $\mathbf{K}_{VSS}(n) = E[\boldsymbol{\epsilon}_n\boldsymbol{\epsilon}_n^T]$ and $\boldsymbol{\epsilon}_n = \mathbf{w}^o - \mathbf{w}_n$ the weight error vector, and $\text{tr}(\mathbf{R})$ the trace of the input autocorrelation matrix $\mathbf{R} \triangleq E[\mathbf{x}_n\mathbf{x}_n^T]$. We would like to write $\mathbf{R}$ as $\mathbf{R} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^T$ with $\mathbf{Q}$ the unitary eigenvector matrix and $\boldsymbol{\Lambda}$ the diagonal eigenvalue matrix, which has been mentioned in Chapter 2. It is observed from (3.3) and (3.4) that the mean step-size and steady-state misadjustment of VSS algorithm are proportional to the power of measurement noise. It indicates that the VSS algorithm will produce large steady-state EMSE or misadjustment in the applications where the SNR is low. When the power of measurement noise increases, the mean step-size of the VSS algorithm also increases, which may diverge the algorithm for a large increase in the noise power.

The work presented in [65, 66] addressed this problem with the following step-size adjustment scheme,

$$\text{NRVSS:} \quad \mu_{n+1} = \alpha_{NRVSS}\mu_n + \gamma_{NRVSS}[k\mathbf{x}_n^T\mathbf{x}_n - 1]e_n^2, \tag{3.5}$$

where the constant parameter settings, $\alpha_{NRVSS} = \alpha_{VSS}$ and $\gamma_{NRVSS} = N\gamma_{VSS}/2$, are provided in [65, 66], and the newly added parameter k is optimally designed as $k = 1/(r_xN)$ with r_x the input power and N the filter length. The absolute value of μ_n in (3.5) is used for the weight vector update. It was shown in [65, 66] that the NRVSS algorithm can achieve the same initial transient behavior and is less sensitive to the noise power when compared to VSS algorithm.

3.2.2 The New Proposed MRVSS Algorithm

It was known that the value of the squared prediction error, e_n^2, is a good indicator of the convergence status of the LMS algorithm. Usually when the squared prediction error

is large, the LMS algorithm is far from the steady state and needs to converge fast with large step size and when the squared prediction error goes to small, the LMS algorithm is close to the steady state, and a small step size is needed to maintain small variations of the tap weights, thereby achieving low misadjustment. From the VSS and NRVSS algorithms, it is observed that the step size is adjusted by the following common form:

$$\mu_{n+1} = \alpha\mu_n + \gamma f(e_n^2), \tag{3.6}$$

where $f(e_n^2)$ represents a function of the squared prediction error, e_n^2. In both VSS and NRVSS algorithms, the scaled versions of e_n^2 were adopted. Note that the instantaneous squared prediction error is very sensitive to the disturbance noise that may occur in the input signals. The weighted sum of the squared prediction errors can reduce this sensitivity. Therefore, we propose to use the weighted sum of the squared prediction errors instead of the instantaneous squared prediction error. We define that

$$A_n = \sum_{i=0}^{n} a^i e_{n-i}^2. \tag{3.7}$$

where a is a decaying factor and bounded as $0 < a < 1$. It can be shown that for very large n, the expectation of A_n is

$$\lim_{n\to\infty} E[A_n] \approx \frac{1}{1-a} \lim_{n\to\infty} E[e_n^2]. \tag{3.8}$$

If the term e_n^2 in VSS is directly replaced by A_n, from the analysis in [66], it is known that a bias corresponding to the variance of measurement noise will be added to the steady-state mean step size. It will increase the steady-state misadjustment. Now we consider another quantity B_n which has the same form as A_n but uses a different weighting factor $0 < b < 1$. By constructing a quotient form of A_n/B_n, the steady-state misadjustment will not affected by the prediction error, thereby being independent of the measurement noise power. The variable step-size adjustment scheme of quotient form is described as follows:

$$\text{MRVSS:} \quad \mu_{n+1} = \alpha\mu_n + \gamma\theta_n, \tag{3.9}$$

$$\theta_n = \frac{\sum_{i=0}^{n} a^i e_{n-i}^2}{\sum_{j=0}^{n} b^j e_{n-j}^2}, \tag{3.10}$$

The constant parameter settings, $\alpha = \alpha_{VSS}$ and $\gamma = \gamma_{VSS}/N$, should be recommended such that the MRVSS algorithm achieves an almost optimal mean step-size behavior as the VSS algorithm for stationary environments according to the design guidelines of [67]. For nonstationary enviornments, we shall demonstrate that the parameter γ in (3.9) could be optimally designed and small α should be chosen for good tracking performance.

The use of a quotient form in (3.9) and (3.10) is twofold. First, the quotient is expected for a smoothing decrease of the step-size, where the transient behavior of the proposed variable step-size in stationary environment may be described by a reformulation of (3.10) as follows

$$\begin{aligned} \theta_n &= \frac{A_n}{B_n} = \frac{aA_{n-1} + e_n^2}{bB_{n-1} + e_n^2} \\ &\approx \frac{a}{b}\theta_{n-1} + \frac{e_n^2}{bB_{n-1}}. \end{aligned} \tag{3.11}$$

Note that in derivation of (3.11), we have neglected the value of e_n^2 in the denominator, since compared to e_n^2 the error cumulant bB_{n-1} becomes much larger during the adaptation because the decaying factor b is very close to one, which will be discussed in Section 3.4. Assuming that the initial step-size μ_0 is set to be μ_{max} for initial fast transient behavior, since the term e_n^2/bB_{n-1} in (3.11) decreases due to the decrease of e_n^2, the variable step-size μ_n will also decrease from its maximum value with the decrease of θ_n. Second, as will be shown in the analysis given in the latter sections, in steady-state, the EMSE of the algorithm should be much smaller compared to the power of measurement noise. This implies that the measurement noise dominates the numerator and denominator of (3.10). In statistical sense, the power of measurement noise in the numerator and denominator could be canceled out, leaving the steady-state mean step-size determined only by the constant parameters. Therefore, the decaying factors a and b could be designed beforehand for a desired steady-state mean step-size level.

3.2.3 Complexity Comparison between the MRVSS and NRVSS Algorithms

Comparing the computational complexities of the NRVSS and MRVSS algorithms, we find that the MRVSS algorithm requires the same number of multiplications/divisions

and additions per iteration by using the recursive forms of A_n and B_n in (3.11). Note that there is a division computation in the MRVSS algorithm, which usually requires some more processing time compared to additions or multiplications. Both of the algorithms are considered to be low complexity and can be implemented in real time. Comparing design simplicity for the parameter of NRVSS and MRVSS added on the VSS algorithm, the NRVSS algorithm adds one new design parameter k, which could be optimally designed with the knowledge of the reference signal power, and the MRVSS algorithm adds two new decaying factors a and b, which could provide a more flexible design as discussed in latter sections and demonstrated in simulations.

In the next sections, we shall study the convergence and performance of the MRVSS algorithm in both stationary and nonstationary environments, and the design guidelines of the decaying factors a and b.

3.3 Convergence and Performance Analysis of the MRVSS Algorithm

3.3.1 Convergence in Mean Sense

For the problem of system identification being considered, we assume that the desired response of the system arises from the model

$$d_n = \mathbf{w}^{oT}\mathbf{x}_n + e_n^o. \tag{3.12}$$

We would like to consider an unknown plant whose weight vector follows a random walk process [1]:

$$\mathbf{w}_{n+1}^o = \mathbf{w}_n^o + \mathbf{r}_n, \tag{3.13}$$

where $\mathbf{r}_n$ represents a driving vector of zero-mean white sequence with a diagonal autocorrelation matrix $\sigma_r^2\mathbf{I}$, and $\mathbf{I}$ is an identity matrix. It is known that for $\sigma_r^2 = 0$, the unknown plant is considered as time invariant and the adaptive filters do not need to

perform a tracking task. Here, we consider a general case of $\sigma_r^2 \neq 0$. Using (3.1) and (3.13), the transient behavior of the weight vector error, $\boldsymbol{\epsilon}_n$, can be modeled as

$$\boldsymbol{\epsilon}_{n+1} = \boldsymbol{\epsilon}_n - \mu_n e_n \mathbf{x}_n + \mathbf{r}_n. \tag{3.14}$$

Consider that the output error can be reformed as

$$e_n = e_n^o + \mathbf{x}_n^T \boldsymbol{\epsilon}_n. \tag{3.15}$$

Substituting (3.15) into (3.14) and applying the unitary transformation $\mathbf{Q}^T$ on both sides of (3.14) yields

$$\mathbf{v}_{n+1} = \left[\mathbf{I} - \mu_n \hat{\mathbf{x}}_n \hat{\mathbf{x}}_n^T\right] \mathbf{v}_n - \mu_n e_n^o \hat{\mathbf{x}}_n + \hat{\mathbf{r}}_n, \tag{3.16}$$

where $\hat{\mathbf{r}}_n$, $\hat{\mathbf{x}}_n$ and $\mathbf{v}_n$ are the transformed versions of $\mathbf{r}_n$, $\mathbf{x}_n$, and $\boldsymbol{\epsilon}_n$ by $\mathbf{Q}^T$, respectively.

To proceed further, we introduce the independence assumption. That is, the input data vectors $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$ are mutually independent, and $\mathbf{x}_n$ is statistically independent of the previous samples of the system response d_n. We also require the following averaging principle commonly used in [40, 57, 58, 68]

$$E\left[\mu_n \hat{\mathbf{x}}_n \hat{\mathbf{x}}_n^T\right] \approx E[\mu_n] E\left[\hat{\mathbf{x}}_n \hat{\mathbf{x}}_n^T\right]. \tag{3.17}$$

This is approximately true since the step-size μ_n varies slowly around its mean value when compared to e_n and $\mathbf{x}_n$ for small γ. Therefore, this justifies the independence assumption of μ_n and μ_n^2 with e_n, $\mathbf{x}_n$ and $\mathbf{w}_n$. Based on the above assumptions, Eq. (3.16) is stable in the mean sense if and only if the eigenvalues of the matrix $\mathbf{D}_n \triangleq E\left[\mathbf{I} - \mu_n \hat{\mathbf{x}}_n \hat{\mathbf{x}}_n^T\right]$ are all within the unit circle. This may be solved as follows

$$\begin{aligned} \det[\mathbf{D}_n - \eta \mathbf{I}] &= \det\{\mathbf{I} - E[\mu_n]\boldsymbol{\Lambda} - \eta \mathbf{I}\} \\ &= \prod_{i=1}^{N} (1 - \eta - E[\mu_n]\lambda_i) = 0. \end{aligned} \tag{3.18}$$

where η denotes the eigenvalues of $\mathbf{D}_n$, λ_i is the ith eigenvalue of the input correlation matrix $\mathbf{R}$. To ensure all the solutions of (3.18) lie within the unit circle, we only need to find the conditions to make sure both the smallest and the largest solutions to be within the unit circle. That is

$$-1 < 1 - E[\mu_n]\lambda_{max} < 1 - E[\mu_n]\lambda_{min} < 1, \tag{3.19}$$

where λ_{max} and λ_{min} are the largest and smallest eigenvalues of $\mathbf{R}$, respectively. From (3.19) we can easily find the following bound on the mean step-size

$$0 < E[\mu_n] < \frac{2}{\lambda_{max}}. \tag{3.20}$$

The condition in (3.20) guarantees the stability of the MRVSS algorithm in the mean sense for nonstationary environments. For the special case of $\sigma_r^2 = 0$, the algorithm is guaranteed to be convergent in the mean sense. Alternative derivations of the condition (3.20) could be found in [57, 58]. For the case of a constant step-size μ_n, the condition (3.20) is the same as the condition for mean convergence of the fixed step-size LMS algorithm as discussed in Chapter 2 of the thesis and in [40, 68].

3.3.2 Convergence in Mean-Square Sense

In this section, we study the more strict condition for mean-square convergence of the MRVSS algorithm. We shall use the following generalized condition derived in Eq. (35) of [57], which provides a sufficient condition for the convergence of the MSE of a variable step-size LMS algorithm

$$0 < \frac{E[\mu_\infty^2]}{E[\mu_\infty]} \leq \frac{2}{3\text{tr}(\mathbf{R})}, \tag{3.21}$$

where it is assumed that $\lim_{n\to\infty} E[\mu_n] = E[\mu_\infty]$ and $\lim_{n\to\infty} E[\mu_n^2] = E[\mu_\infty^2]$, which are the steady-state mean step-size and mean-square step-size. It is seen that only the steady-state statistics of the step-size are required to establish the mean-square convergence of the MRVSS algorithm. Next, we shall find the forms for $E[\mu_\infty]$ and $E[\mu_\infty^2]$, respectively, to establish the sufficient condition in (3.21).

Taking expectations on both sides of (3.9) and assuming that n approaches to infinity yields

$$\lim_{n\to\infty} E[\mu_n] = \alpha \lim_{n\to\infty} E[\mu_{n-1}] + \gamma \lim_{n\to\infty} E[\theta_n]. \tag{3.22}$$

From (3.22), we can approximate the steady-state mean step-size as follows

$$E[\mu_\infty] \approx \frac{\gamma \lim_{n\to\infty} E[\theta_n]}{1-\alpha}. \tag{3.23}$$

The term $\lim_{n\to\infty} E[\theta_n]$ in (3.23) is given by

$$\lim_{n\to\infty} E[\theta_n] = \lim_{n\to\infty} E\left[\frac{A_n}{B_n}\right]. \tag{3.24}$$

Note that A_n and B_n are with recursive forms as shown in (3.11). Assuming that n approaches to infinity and $\lim_{n\to\infty} E[e_n^2]$ is slowly varying, we have the following approximations

$$\lim_{n\to\infty} E[A_n] \approx \frac{1}{1-a} \lim_{n\to\infty} E[e_n^2], \tag{3.25}$$

$$\lim_{n\to\infty} E[B_n] \approx \frac{1}{1-b} \lim_{n\to\infty} E[e_n^2]. \tag{3.26}$$

To proceed (3.24), we assume that

$$\lim_{n\to\infty} E\left[\frac{A_n}{B_n}\right] \approx \lim_{n\to\infty} \frac{E[A_n]}{E[B_n]}. \tag{3.27}$$

This cannot really hold for the algorithm, but approximately true since that $\lim_{n\to\infty} A_n$ and $\lim_{n\to\infty} B_n$ will vary slowly. Using the results of (3.25) and (3.26), we can approximate (3.24) as

$$\lim_{n\to\infty} E[\theta_n] = E[\theta_\infty] \approx \frac{1-b}{1-a}. \tag{3.28}$$

Substituting the result of (3.28) into (3.23) yields

$$E[\mu_\infty] \approx \frac{\gamma(1-b)}{(1-\alpha)(1-a)}. \tag{3.29}$$

Here, we would like to drop a point that comparing to the steady-state mean VSS step-size shown in (3.3), Eq. (3.29) shows that the steady-state mean MRVSS step-size is now free from the steady-state MSE, that is, $\xi_{min} + \xi_{ex}(\infty)$. Since the assumption in (3.27) is approximately true, we may conclude that the MRVSS algorithm should be much less sensitive to the measurement noise power when compared to the VSS algorithm. This will be demonstrated in the simulation section.

Next we derive the steady-state mean-square MRVSS step-size. Squaring Eq. (3.10) and taking expectation of it, we yield

$$\lim_{n\to\infty} E[\mu_n^2] = \lim_{n\to\infty} \{\alpha^2 E[\mu_{n-1}^2] + 2\alpha\gamma E[\mu_n] E[\theta_n] + \gamma^2 E[\theta_n^2]\}. \tag{3.30}$$

For small γ, the last term in (3.30) involving γ^2 is negligible compared with the other terms. So after eliminating this small term, we can approximate as

$$\lim_{n\to\infty} E[\mu_n^2] \approx \frac{2\alpha\gamma E[\mu_\infty]E[\theta_\infty]}{1-\alpha^2}. \tag{3.31}$$

Substituting the results of (3.28) and (3.29) into (3.31) yields

$$E[\mu_\infty^2] \approx \frac{2\alpha\gamma^2(1-b)^2}{(1-\alpha^2)(1-\alpha)(1-a)^2}. \tag{3.32}$$

Eq. (3.32) gives the steady-state mean-square MRVSS step-size. The sufficient condition on α, γ, a, and b can be now established by substituting (3.29) and (3.32) into (3.21), which is given by

$$0 < \frac{\alpha\gamma(1-b)}{(1-\alpha^2)(1-a)} \leq \frac{1}{3\mathrm{tr}(\mathbf{R})}. \tag{3.33}$$

Eq. (3.33) therefore guarantees the mean-square convergence of the MRVSS algorithm. Since the parameters α, γ of MRVSS are set following the guidelines for VSS given in [67] for good convergence performance, the choices of a and b should satisfy the condition of (3.33) for stability. Eq. (3.29) already shows that the values of a and b determine the steady-state mean MRVSS step-size, thereby determine the steady-state performance too. It can be easily observed that choosing the small ratio of $(1-b)/(1-a)$ can satisfy the condition of (3.33). This also means that a small steady-state mean MRVSS step-size is produced, which implies a small steady-state EMSE or misadjustment. To show this point clearly, we provide the analytic result for the steady-state performance of the MRVSS algorithm in the next section.

3.3.3 Analysis of Steady-State Performance

We analyze the steady-state performance of the proposed algorithm by deriving the misadjustment, where the independence assumptions are applied. We are considering the random walk model of the nonstationary system shown in (3.13) and the weight adaptation law (3.1). To proceed further, besides the independent assumption we need to further assume that the input signal x_n and the desired response d_n are jointly Gaussian.

Let the transformed covariance of the weight vector be $\mathbf{C}_n = E[\mathbf{v}_n\mathbf{v}_n^T]$ where $\mathbf{v}_n$ is given in (3.16). Postmultiplying both sides of (3.16) by $\mathbf{v}_{n+1}^T$ and then taking the expectations yields the transient behavior of the transformed covariance $\mathbf{C}_n$ as follows:

$$\begin{aligned}\mathbf{C}_{n+1} &= \mathbf{C}_n - E[\mu_n]\mathbf{C}_n\Lambda - E[\mu_n]\Lambda\mathbf{C}_n \\ &\quad +2E[\mu_n^2]\Lambda\mathbf{C}_n\Lambda + E[\mu_n^2]\Lambda\mathrm{tr}(\Lambda\mathbf{C}_n) \\ &\quad +E[\mu_n^2]\xi_{min}\Lambda + \sigma_r^2\mathbf{I}. \end{aligned} \tag{3.34}$$

Note that in (3.34), the Gaussian factoring theorem is used to simplify the expression $E[\hat{\mathbf{x}}_n\hat{\mathbf{x}}_n^T\mathbf{v}_n\mathbf{v}_n^T\hat{\mathbf{x}}_n\hat{\mathbf{x}}_n^T]$ into a sum of second-order moments. Let $\mathbf{c}_n$ be the vector with the entries of the diagonal elements of $\mathbf{C}_n$, let $\boldsymbol{\lambda}$ be the column vector of eigenvalues of $\mathbf{R}$, and let $\mathbf{1}$ be a column vector of 1's with the same length as $\boldsymbol{\lambda}$. Following the analyzed results in [40, 57], from (3.34) we obtain

$$\mathbf{c}_{n+1} = \mathbf{L}_n\mathbf{c}_n + E[\mu_n^2]\xi_{min}\boldsymbol{\lambda} + \sigma_r^2\mathbf{1}, \tag{3.35}$$

$$\mathbf{L}_n = diag[\rho_1, \rho_2, \cdots, \rho_N] + E[\mu_n^2]\boldsymbol{\lambda}\boldsymbol{\lambda}^T, \tag{3.36}$$

$$\rho_i = 1 - 2E[\mu_n]\lambda_i + 2E[\mu_n^2]\lambda_i^2. \tag{3.37}$$

In the steady-state of the algorithm, we can assume $\lim_{n\to\infty}\mathbf{c}_{n+1} = \lim_{n\to\infty}\mathbf{c}_n$. Then we rewrite (3.35) as

$$\mathbf{c}_\infty = [\mathbf{I} - \mathbf{L}_\infty]^{-1}\{E[\mu_\infty^2]\xi_{min}\boldsymbol{\lambda} + \sigma_r^2\mathbf{1}\}. \tag{3.38}$$

Using (3.38) into the steady-state excess MSE given by $\xi_{ex}(\infty) = \boldsymbol{\lambda}^T\mathbf{c}_\infty$, we obtain

$$\xi_{ex}(\infty) = \boldsymbol{\lambda}^T[\mathbf{I} - \mathbf{L}_\infty]^{-1}\{E[\mu_\infty^2]\xi_{min}\boldsymbol{\lambda} + \sigma_r^2\mathbf{1}\}. \tag{3.39}$$

Let $\boldsymbol{\rho} = diag[1-\rho_1, 1-\rho_2, \cdots, 1-\rho_N]$. Using (3.36), (3.37) in steady-state, and the matrix inversion lemma, we can rewrite (3.39) as

$$\begin{aligned}\xi_{ex}(\infty) &= \boldsymbol{\lambda}^T\left\{\boldsymbol{\rho} - \boldsymbol{\lambda}E[\mu_\infty^2]\boldsymbol{\lambda}^T\right\}^{-1}\left\{E[\mu_\infty^2]\xi_{min}\boldsymbol{\lambda} + \sigma_r^2\mathbf{1}\right\} \\ &= \boldsymbol{\lambda}^T\left\{\boldsymbol{\rho}^{-1} + \boldsymbol{\rho}^{-1}\boldsymbol{\lambda}\left(E[\mu_\infty^2]^{-1} - \boldsymbol{\lambda}^T\boldsymbol{\rho}^{-1}\boldsymbol{\lambda}\right)^{-1}\boldsymbol{\lambda}^T\boldsymbol{\rho}^{-1}\right\} \\ &\quad \cdot\left\{E[\mu_\infty^2]\xi_{min}\boldsymbol{\lambda} + \sigma_r^2\mathbf{1}\right\} \\ &= \left\{1 - E[\mu_\infty^2]\boldsymbol{\lambda}^T\boldsymbol{\rho}^{-1}\boldsymbol{\lambda}\right\}^{-1} \\ &\quad \cdot\left\{E[\mu_\infty^2]\xi_{min}\boldsymbol{\lambda}^T\boldsymbol{\rho}^{-1}\boldsymbol{\lambda} + \sigma_r^2\boldsymbol{\lambda}^T\boldsymbol{\rho}^{-1}\mathbf{1}\right\}. \end{aligned} \tag{3.40}$$

It is easy to obtain the terms in (3.40) as follows

$$\boldsymbol{\lambda}^T\boldsymbol{\rho}^{-1}\boldsymbol{\lambda} = \sum_{i=1}^{N}\frac{\lambda_i}{2E[\mu_\infty]-2E[\mu_\infty^2]\lambda_i}, \tag{3.41}$$

$$\boldsymbol{\lambda}^T\boldsymbol{\rho}^{-1}\mathbf{1} = \sum_{i=1}^{N}\frac{1}{2E[\mu_\infty]-2E[\mu_\infty^2]\lambda_i}. \tag{3.42}$$

Substituting (3.41) and (3.42) into (3.40), we finally obtain

$$\begin{aligned}\xi_{ex}(\infty) = &\left\{\frac{\xi_{min}\sum_{i=1}^{N}\frac{E[\mu_\infty^2]\lambda_i}{2E[\mu_\infty]-2E[\mu_\infty^2]\lambda_i}}{1-\sum_{i=1}^{N}\frac{E[\mu_\infty^2]\lambda_i}{2E[\mu_\infty]-2E[\mu_\infty^2]\lambda_i}}\right\}\\ &+\left\{\frac{\sum_{i=1}^{N}\frac{\sigma_r^2}{2E[\mu_\infty]-2E[\mu_\infty^2]\lambda_i}}{1-\sum_{i=1}^{N}\frac{E[\mu_\infty^2]\lambda_i}{2E[\mu_\infty]-2E[\mu_\infty^2]\lambda_i}}\right\}.\end{aligned} \tag{3.43}$$

Note that in the denominator of (3.43), compared with the first term, the second term involved with $E[\mu_\infty^2]$ is very small in the event of small values of misadjustment. Neglecting the second term in the denominator of (3.43), we have

$$\xi_{ex}(\infty) \approx \xi_{min}\mathrm{tr}(\mathbf{R})\frac{E[\mu_\infty^2]}{2E[\mu_\infty]}+\frac{N\sigma_r^2}{2E[\mu_\infty]}. \tag{3.44}$$

Substituting the result of μ_∞ (3.29) and μ_∞^2 (3.32) into (3.44) and using the definition of $\mathcal{M} = \xi_{ex}(\infty)/\xi_{min}$ yields the misadjustment of the MRVSS algorithm:

$$\mathcal{M} \approx \frac{\alpha\gamma(1-b)}{(1-\alpha^2)(1-a)}\mathrm{tr}(\mathbf{R})+\frac{(1-\alpha)(1-a)N\sigma_r^2}{2\gamma(1-b)\xi_{min}}. \tag{3.45}$$

Since the first term in (3.45) is directly proportional to γ and the second term is inversely proportional to γ, we can optimize the choice for γ given α, a and b to achieve the minimum misadjustment. When both terms are equal, the optimal choice of γ is given by

$$\gamma^* = \sqrt{\frac{(1-\alpha)(1-\alpha^2)(1-a)^2N\sigma_r^2}{2\alpha(1-b)^2\xi_{min}\mathrm{tr}(\mathbf{R})}}. \tag{3.46}$$

And the result of minimum misadjustment follows

$$\mathcal{M}_{min} = \sqrt{\frac{2\alpha N \sigma_r^2 \mathrm{tr}(\mathbf{R})}{(1+\alpha)\xi_{min}}}. \tag{3.47}$$

We can observe from (3.47) that smaller α should provide better tracking performance. We will demonstrate this point in the simulation section.

Compared to the minimum misadjustment of the fixed LMS algorithm [68], we have

$$\mathcal{M}_{min} = M_{min}^{LMS} \sqrt{\frac{2\alpha}{(1+\alpha)}}. \tag{3.48}$$

It is seen from (3.48) that for the parameter α of very close to one, the misadjustment of the new algorithm can be very close to that of the fixed LMS algorithm.

In stationary environment, i.e., $\sigma_r^2 = 0$, the misadjustment is then given by

$$\mathcal{M} \approx \frac{\alpha\gamma(1-b)}{(1-\alpha^2)(1-a)} \mathrm{tr}(\mathbf{R}). \tag{3.49}$$

We can observe from (3.49) that the steady-state misadjustment of the MRVSS algorithm is proportional to the ratio of $(1-b)/(1-a)$, which is consistent with the result of the steady-state mean MRVSS step-size shown in (3.29). For the case that α is very close to one, we can approximate (3.49) as

$$\mathcal{M} \approx \frac{1}{2} E[\mu_\infty] \mathrm{tr}(\mathbf{R}). \tag{3.50}$$

The result of (3.50) is then consistent with the result for the fixed LMS algorithm as shown in [1].

3.3.4 Mean Behavior of the Proposed Variable Step-Size

In this section, we investigate the mean behavior of the MRVSS step-size in a general nonstationary environment where the plant varies following the model of (3.13).

Taking the expectations of (3.9) we obtain

$$E[\mu_{n+1}] = \alpha E[\mu_n] + \gamma E[\theta_n]. \tag{3.51}$$

Next, we need to evaluate the mean value of θ_n, which is given by

$$E[\theta_n] = E\left[\frac{A_n}{B_n}\right]. \tag{3.52}$$

To proceed further, we make the following assumption

$$E\left[\frac{A_n}{B_n}\right] \approx \frac{E[A_n]}{E[B_n]}. \tag{3.53}$$

This is approximately true for the decaying factors of a and b close to one. Using (3.53) in (3.52), we then only need to evaluate $E[A_n]$ and $E[B_n]$, which are given by

$$E[A_n] = aE[A_{n-1}] + E[e_n^2], \tag{3.54}$$

$$E[B_n] = bE[B_{n-1}] + E[e_n^2]. \tag{3.55}$$

Using the form of the output error $e_n = e_n^o + \hat{\mathbf{x}}_n^T \mathbf{v}_n$, we proceed $E[e_n^2]$ as

$$\begin{aligned} E[e_n^2] &= E[(e_n^o + \hat{\mathbf{x}}_n^T \mathbf{v}_n)^2] \\ &= E[(e_n^o)^2] + 2E[e_n^o \hat{\mathbf{x}}_n^T \mathbf{v}_n \mathbf{v}_n^T \hat{\mathbf{x}}_n] + E[\hat{\mathbf{x}}_n^T \mathbf{v}_n \mathbf{v}_n^T \hat{\mathbf{x}}_n] \\ &\approx (\xi_{min} + \boldsymbol{\lambda}^T \mathbf{c}_n). \end{aligned} \tag{3.56}$$

where $\mathbf{c}_n$ is given in (3.35) and $\boldsymbol{\lambda}$ is the column vector of the eigenvalues of $\mathbf{R}$. Note that during the derivation of (3.56) we have used the independent assumption and the assumptions that the input signal x_n and desired response d_n are jointly Gaussian and the measurement noise e_n^o is zero-mean white Gaussian and statistically independent of x_n, which justifies the Gaussian moment factoring theorem and the simplification. Using (3.52) to (3.56), we have

$$E[\theta_n] \approx \frac{aE[A_{n-1}] + (\xi_{min} + \boldsymbol{\lambda}^T \mathbf{c}_n)}{bE[B_{n-1}] + (\xi_{min} + \boldsymbol{\lambda}^T \mathbf{c}_n)}. \tag{3.57}$$

Since $\mathbf{c}_n$ requires the value of $E[\mu_{n-1}^2]$, we need to establish the transient behavior for $E[\mu_n^2]$. Taking the expectations of the squared (3.10) we obtain

$$E[\mu_n^2] \approx \alpha^2 E[\mu_{n-1}^2] + 2\alpha\gamma E[\mu_{n-1}]E[\theta_{n-1}] + \gamma^2 E[\theta_{n-1}^2]. \tag{3.58}$$

Note that we have used $E[\mu_n \theta_n] \approx E[\mu_n]E[\theta_n]$ based on the independence assumption. Neglecting the very small term involving γ^2, we can approximate (3.58) as follows

$$E[\mu_n^2] \approx \alpha^2 E[\mu_{n-1}^2] + 2\alpha\gamma E[\mu_{n-1}]E[\theta_{n-1}]. \tag{3.59}$$

Now the mean behavior of the MRVSS step-size is completed described by (3.51), (3.54), (3.55), (3.57), (3.59) and (3.35)–(3.37).

3.4 Issues on the Design of Constant Parameters

From above discussion, we are clear that the mean and mean square convergence of the MRVSS algorithm are guaranteed by conditions (3.20) and (3.33), respectively. Therefore, the choices of α, γ, a and b should first satisfy such conditions. We have recommended that the choices of α and γ are $\alpha = \alpha_{VSS}$ and $\gamma = \gamma_{VSS}/N$, where α_{VSS} and γ_{VSS} should be designed based on the guidelines in [67]. We will demonstrate in the simulation section the choices of a and b do not affect the transient behavior of the mean MRVSS step-size much, but do affect the steady-state mean MRVSS step-size. To show the effectiveness of the choices of the parameters a and b on the steady-state performance of MRVSS algorithm, we make a ratio of the misadjustments of MRVSS algorithm and VSS algorithm as follows

$$\frac{\mathcal{M}}{\mathcal{M}_{VSS}} = \frac{\alpha\gamma(1-b)\mathrm{tr}(\mathbf{R})}{(1-\alpha^2)(1-a)} \frac{1-\alpha_{VSS}^2}{\gamma_{VSS}(3-\alpha_{VSS})\left(\xi_{min}+\xi_{ex}(\infty)\right)\mathrm{tr}(\mathbf{R})}. \tag{3.60}$$

For the setting $\alpha = \alpha_{VSS}$ and $\gamma = \gamma_{VSS}/N$, Eq. (3.60) can be simplified as

$$\frac{\mathcal{M}}{\mathcal{M}_{VSS}} = \frac{\alpha_{VSS}(1-b)}{N(3-\alpha_{VSS})(1-a)\left(\xi_{min}+\xi_{ex}(\infty)\right)}. \tag{3.61}$$

Considering that α_{VSS} is close to one and $\xi_{ex}(\infty) \ll \xi_{min}$, we can approximate (3.61) as

$$\frac{\mathcal{M}}{\mathcal{M}_{VSS}} \approx \frac{(1-b)}{2N\xi_{min}(1-a)}. \tag{3.62}$$

From (3.62), we can see that the MRVSS algorithm can be designed to achieve smaller steady-state misadjustment by choosing $(1-b) \ll (1-a)$ for any level of measurement noise power and filter length N. It is easily observed that for larger measurement noise power (larger ξ_{min}), the improvement of the steady-state misadjustment with MRVSS algorithm is more significant.

For nonstationary enviornments, α should be chosen small for good tracking performance according to (3.47) and γ are given from (3.46). However, in the simulation we found $\gamma^*/3$ rather gives a better tracking performance, this may be due to the assumptions and approximations we have made in the derivation of γ^*. The decaying factors a and b are chosen similarly as made for stationary case.

3.5 Simulation Results

This section presents the applications to system identification and noise cancellation to corroborate the analytic results derived in this chapter and to illustrate the properties of the MRVSS algorithm. In the four examples to be demonstrated, the performances of VSS algorithm [57], NRVSS algorithm [65, 66] and MRVSS algorithm are compared to demonstrate that MRVSS can lead to a better steady-state performance for the same initial transient behavior in stationary environments. For nonstationry environments we demonstrate in Example 3 that MRVSS can provide faster tracking capability. The mean MRVSS step-size behavior related to the choices of a and b and the measurement noise power are demonstrated in Example 1. The sensitivities of VSS, NRVSS and MRVSS to noise power are compared in Example 2. Examples 1, 2 and 3 are based on the problem of system identification. The last example presents a comparison between NRVSS and MRVSS algorithms when applied to noise cancellation.

To begin with, let us describe the parameter settings. We set the parameters according to the suggestions in [66], where $\alpha_{VSS} = 0.9997$; $\gamma_{VSS} = 2 \times 10^{-5}$; $\alpha_{NRVSS} = \alpha_{VSS}$; $\gamma_{NRVSS} = N\gamma_{VSS}/2$; $\alpha = \alpha_{VSS}$; $\gamma = \gamma_{VSS}/N$; $a = 0.9$; $b = 1 - 10^{-5}$. For all the algorithms, we used $\mu(0) = \mu_{max} = 0.01$ and $\mu_{min} = 0$. It can be easily checked that the mean stability condition (3.20) is satisfied with such parameter setting. The results were obtained by averaging 500 ensemble trials in Monte carlo simulations. The SNR is calculated by $\text{SNR} = 10\log_{10} E[x_n^2]/E[(e_n^o)^2]$.

3.5.1 Application to System Identification

The experimental setup of the system identification task is given in Figure 2.1, the unknown plant and the LMS adaptive filters have the same number of ten taps ($N = 10$), where the plant impulse response is designed using a Hanning window with unit norm ($\mathbf{w}_0^{oT}\mathbf{w}_0^o = 1$). Unless stated otherwise, in all examples the input signal x_n is with unit power and the additive measurement noise e_n^o is zero-mean, white Gaussian and statistically independent of x_n.

Table 3.1: Steady-state misadjustments for VSS, NRVSS and MRVSS algorithms under different SNR conditions and comparisons between theoretical models and Monte Carlo simulations for MRVSS algorithm (all results are in dB).

SNR	Analytical values	Simulated values		
	MRVSS [Eq.(3.46)]	VSS	NRVSS	MRVSS
20	-54.78	-24.56	-32.82	-53.01
10	-54.78	-14.45	-23.77	-52.23
0	-54.78	-12.64	-15.27	-52.20
-10	-54.78	-12.64	-12.72	-52.18

Example 1: White Gaussian Input and Stationary Environment

The plant impulse response is time invariant ($\mathbf{w}_n^o = \mathbf{w}_{n-1}^o$) and the input is white Gaussian. We use SNR = 60, 20, 10, 0 and -10 dB. Table 3.1 presents the steady-state misadjustments for VSS, NRVSS and MRVSS for SNR ranging from 20 to -10 dB. The theoretical values are also obtained based on Eq. (3.49), which are compared with the simulation results. The main causes of mismatch between the analytical and simulation results seem to be the following: (a) the assumption of (3.27); (b) the neglected terms involving γ^2 or $E[\mu_\infty^2]$ in (3.31) and (3.44), respectively; and (c) the independent assumption. Figures 3.1 and 3.2 show the excess mean-square error (EMSE) learning curves and mean step-size behaviors, respectively, of VSS, NRVSS and MRVSS algorithms up to iteration $40,000$ at SNR = 10 dB. All algorithms present approximately the same initial transient behavior but MRVSS achieves lower steady-state EMSE. The predicted and simulation mean step-size behaviors agree well with each other. Figure 3.3 shows the evolution of the mean MRVSS step-size for different settings of the decaying factors a and b. It can be seen that the transient behavior of MRVSS algorithm is not sensitive to the choices of a and b, but the steady-state EMSE or misadjustment is affected by the choice of a and b. The smaller a or larger b gives the lower steady-state EMSE. However,

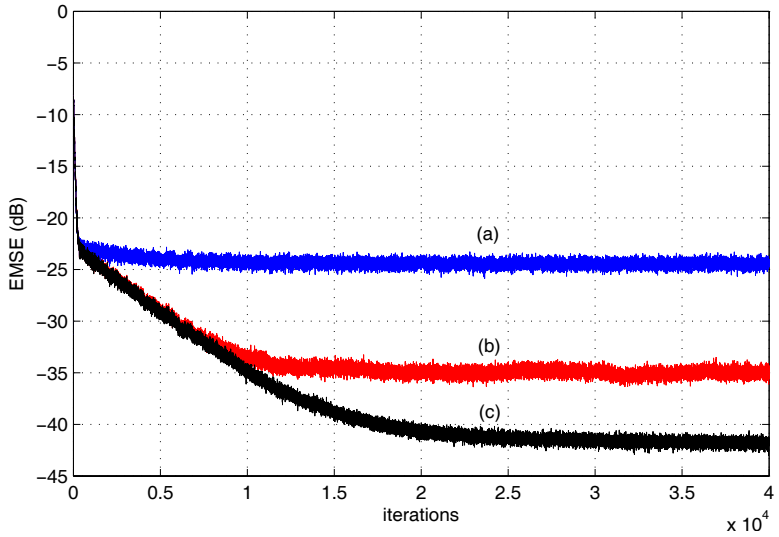

Figure 3.1: Comparisons of excess mean-square error (EMSE) between (a) VSS, (b) NRVSS and (c) MRVSS simulations for Example 1: the white Gaussian input and stationary plant with SNR = 10 dB.

if small b or large a is used, poor steady-state EMSE may be produced, which is consistent with our analytic result. Figure 3.4 shows the evolution of the MRVSS step-size for SNR = 20, 10, 0 and -10 dB. It can be seen that the mean MRVSS step-size has low sensitivity to the SNR changes, so does the steady-state misadjustment of MRVSS algorithm as demonstrated in Example 2.

Example 2: Autoregressive Input and Stationary Environment

The plant impulse response is time invariant, and both the white Gaussian input and a correlated first-order autoregressive input ($x_n = a_1 x_{n-1} + u_n$) with $a_1 = 0.9$ were tested. The measurement noise has an abrupt power variation at iteration $50,000$ such that SNR changes from 60 dB to 20 dB. Figures 3.5 and 3.7 show the EMSE behaviors of VSS, NRVSS and MRVSS algorithms, and Figures 3.6 and 3.8 show the evolution of the mean step-sizes of VSS, NRVSS and MRVSS algorithms. It can be seen that all algorithms perform equally under high SNR conditions, whereas under low SNR conditions, the MRVSS algorithm shows a better performance in steady-state conditions.

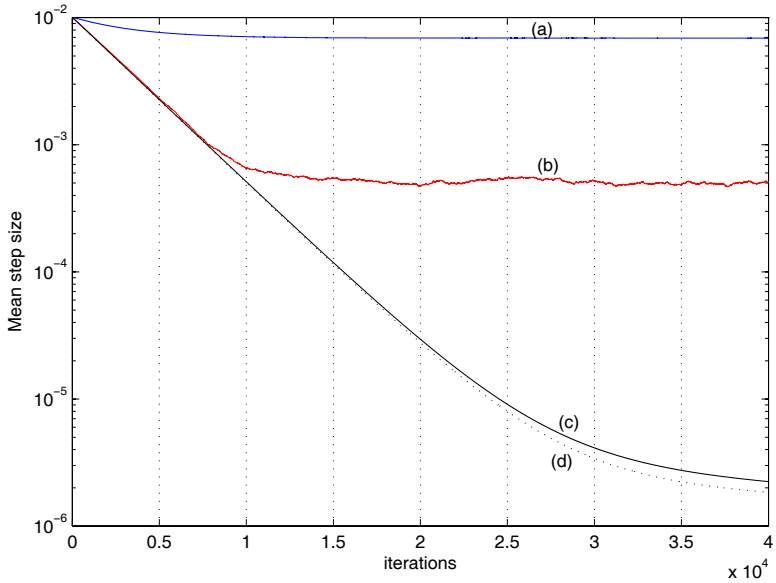

Figure 3.2: Comparisons of mean behavior of the step-size (μ_n) between (a) VSS, (b) NRVSS, (c) MRVSS simulations, and (d) MRVSS model for Example 1: the white Gaussian input and stationary plant with SNR = 10 dB.

Example 3: Autoregressive Input and Nonstationary Environment

The plant impulse response is time variant modeled as a first-order random walk process as in Eq. (3.13) with $\sigma_r^2 = 10^{-10}$, and the input is a correlated first-order autoregressive process ($x_n = a_1 x_{n-1} + u_n$) with $a_1 = 0.7$ [66]. We tested at SNR = 20 dB. The parameters of MRVSS algorithm were reset as $\alpha = 0.5$ and $\gamma = 0.2$ according to Eqs. (3.47) and (3.46). All the other parameters were unchanged. Figures 3.9 and 3.10 show the tracking behaviors and the evolution of mean step-sizes of VSS, NRVSS and MRVSS algorithms. It can be seen that the MRVSS algorithm achieves a faster tracking behavior and a slightly lower misadjustment. We need to point out that the accumulative nature of MRVSS may lower its recovery ability for abrupt changes of the plant impulse response when small γ is usesd. This is because the MRVSS has the memory of past power of output errors. In such cases, the suggestion in [61] could be used. That is, the step-size should switch automatically to μ_{max} at the abrupt change, where the system is monitored and the abrupt changes are detected employing the power of output error with suitable threshold (chosen according to the application at hand). The power of output error is recursively

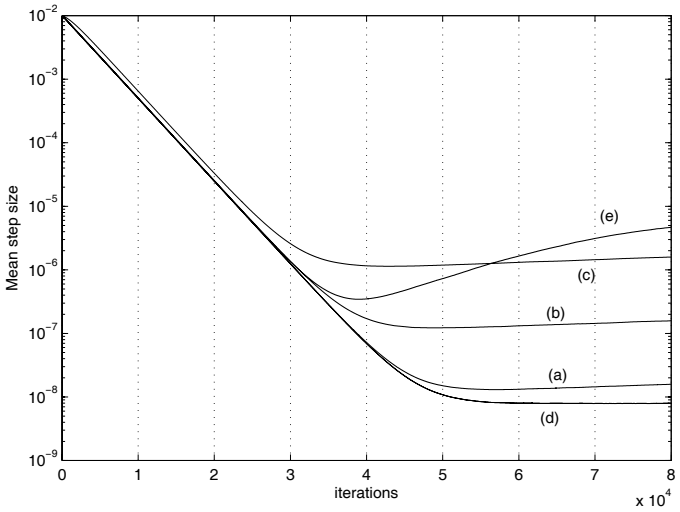

Figure 3.3: Comparisons of mean behavior of the MRVSS step-size (μ_n) for Example 1: the white Gaussian input and stationary plant with (a) $a = 0.9, b = 1 - 10^{-5}$, (b) $a = 0.99, b = 1 - 10^{-5}$, (c) $a = 0.999, b = 1 - 10^{-5}$, (d) $a = 0.9, b = 1 - 10^{-6}$ and (e) $a = 0.9, b = 1 - 10^{-4}$.

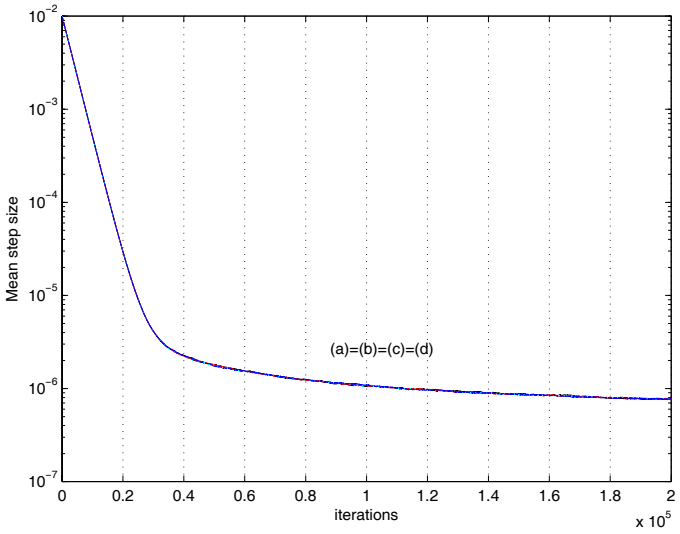

Figure 3.4: Comparisons of mean behavior of the MRVSS step-size (μ_n) for Example 1: the white Gaussian input and stationary plant with SNR: (a) -10dB, (b) 0 dB, (c) 10 dB and (d) 20 dB.

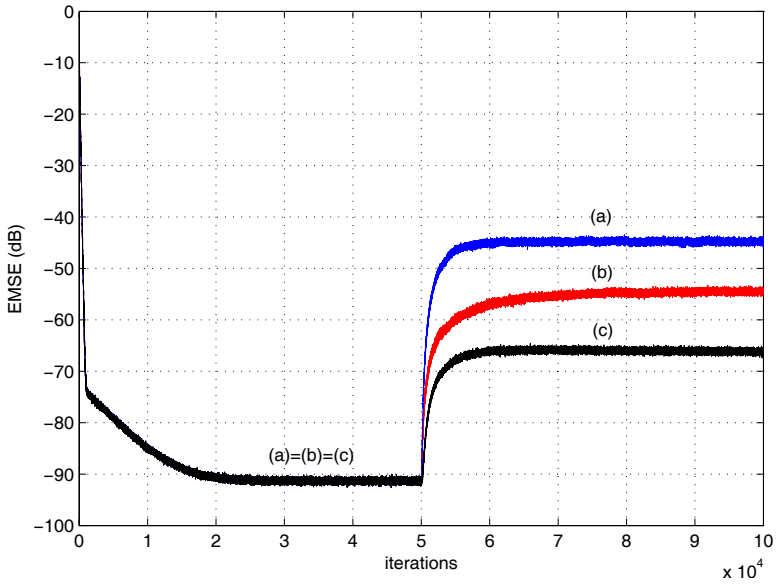

Figure 3.5: Comparisons of excess mean-square error (EMSE) between (a) VSS, (b) NRVSS and (c) MRVSS simulations for Example 2: the white Gaussian input and stationary plant with SNR change from 60 dB to 20 dB.

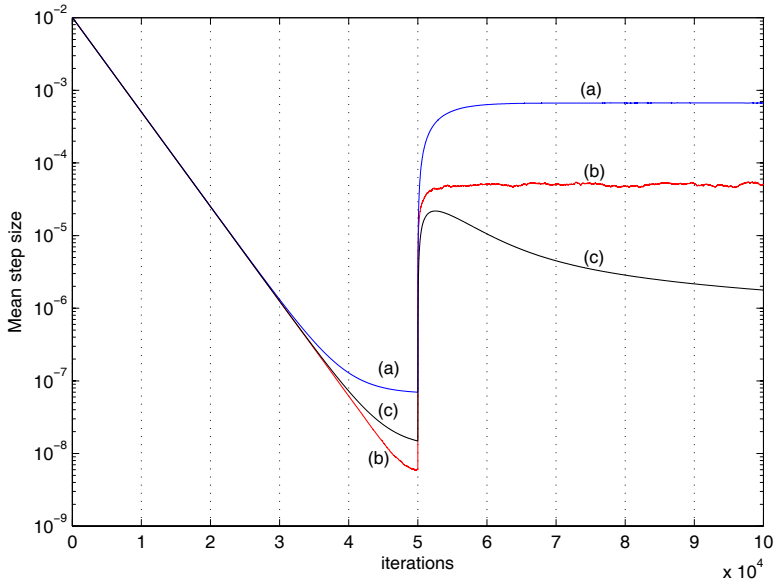

Figure 3.6: Comparisons of mean behavior of the step-size (μ_n) between (a) VSS, (b) NRVSS and (c) MRVSS simulations for Example 2: the white Gaussian input and stationary plant with SNR change from 60 dB to 20 dB.

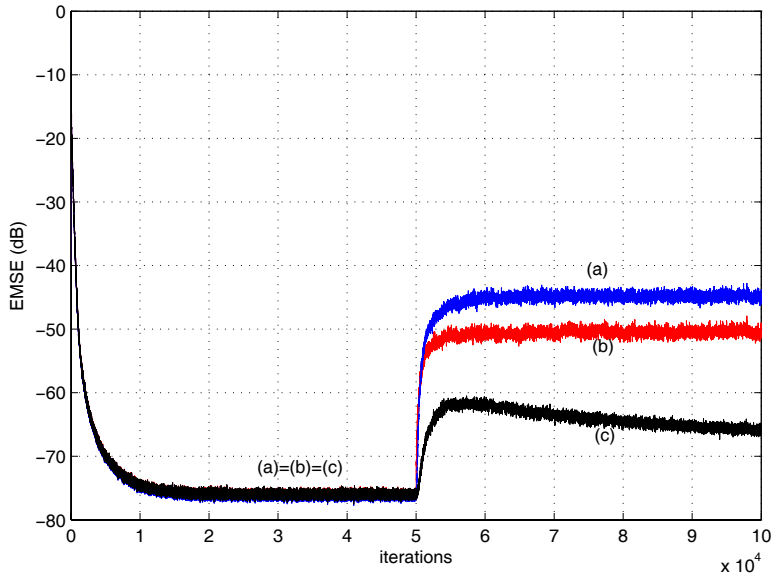

Figure 3.7: Comparisons of excess mean-square error (EMSE) between (a) VSS, (b) NRVSS and (c) MRVSS simulations for Example 2: the correlated first-order autoregressive input and stationary plant with SNR change from 60 dB to 20 dB.

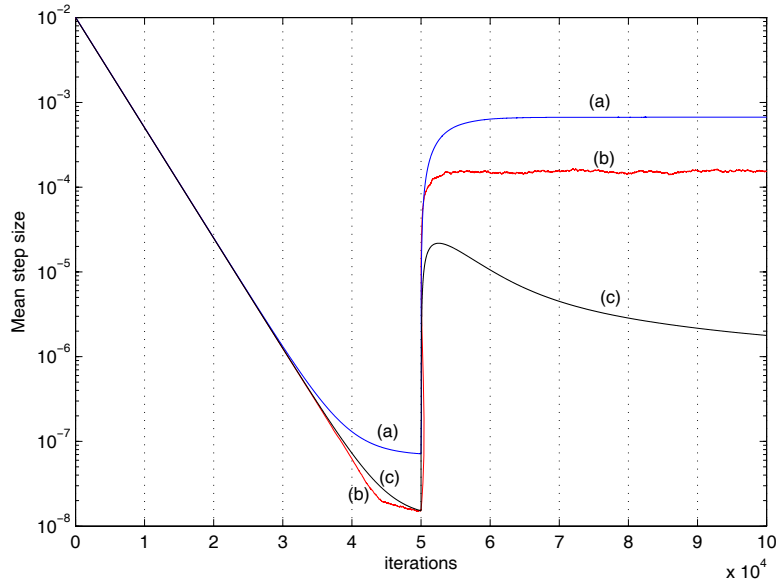

Figure 3.8: Comparisons of mean behavior of the step-size (μ_n) between (a) VSS, (b) NRVSS and (c) MRVSS simulations for Example 2: the correlated first-order autoregressive input and stationary plant with SNR change from 60 dB to 20 dB.

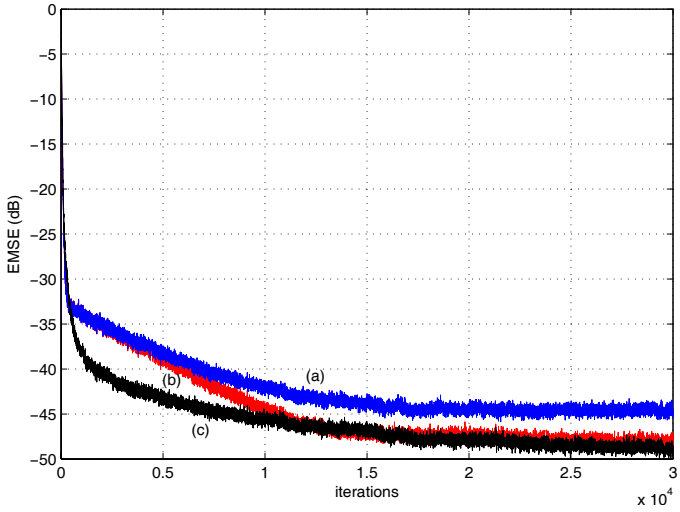

Figure 3.9: Comparisons of excess mean-square error (EMSE) between (a) VSS, (b) NRVSS and (c) MRVSS simulations for Example 3: the correlated first-order autoregressive input and nonstationary plant with SNR= 20 dB.

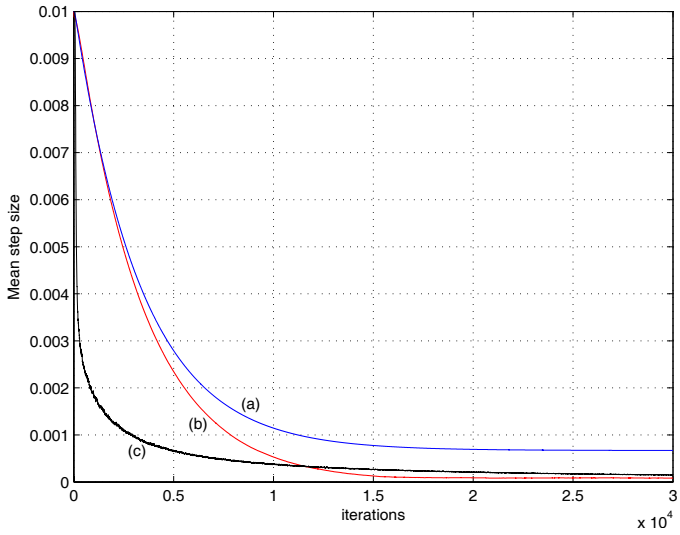

Figure 3.10: Comparisons of mean behavior of the step-size (μ_n) between (a) VSS, (b) NRVSS and (c) MRVSS simulations for Example 3: the correlated first-order autoregressive input and nonstationary plant with SNR= 20 dB.

updated as $E_n = \beta E_{n-1} + (1-\beta)e_n^2$. The additional cost is two extra multiplications and one extra addition. For MRVSS algorithm, the values of A_n and B_n are re-initialized consequently. Although such scheme is applicable in the applications, NRVSS and VSS are more recommended in such situations for their better recovery abilities.

3.5.2 Application to Noise Cancellation

In this last example, we study the performance of MRVSS algorithm for the application to noise cancellation where the output error e_n is expected to converge to the desired noiseless signal, which is correlated in time. In this example, the task is to remove an additive pure sinusoidal noise generated by ($u_n = 0.8\sin(\omega n + 0.5\pi)$ with $\omega = 0.1\pi$ from a square wave generated by ($s_n = 2 \times ((\mathrm{mod}(n, 1000) < 1000/2) - 0.5)$) where $\mathrm{mod}(n, 1000)$ computes the modulus of n over 1000. The summation of s_n and u_n serves as the reference response of the adaptive filters. The input to the adaptive filters is also a sinusoidal signal ($x_n = \sqrt{2}\sin(\omega n)$) with $\omega = 0.1\pi$, which is in unit power. The SNR is therefore 0 dB. The parameter settings were the same as used in Examples 1 and 2. The length of the adaptive filters is ten. Since the VSS algorithm diverged in the simulation, only the performance of NRVSS algorithm and MRVSS algorithm are compared. Figure 3.11 shows the reference response and the processed results by the NRVSS algorithm and MRVSS algorithm. It can be seen that both algorithms are capable of suppressing the interference. Figures 3.12 and 3.13 shows the instantaneous step-size behavior of both algorithms. The step-size values in steady state are approximately equal while MRVSS achieves slightly faster initial adaptation velocity. Next we consider another type of additive noise generated by ($u_n = \sum_{i=0}^{9} h_i v(n-i)$, where the transfer function h is randomly generated as $h = randn(10, 1)$ and v_n is a zero-mean white Gaussian noise with unit variance. The inputs of all the adaptive filters were set to equal to v_n. The desired signal remains the same as above. The results after noise cancellation are shown in Figure 3.14. It can be seen that the MRVSS algorithm provides a better result for removing this type of Gaussian noise.

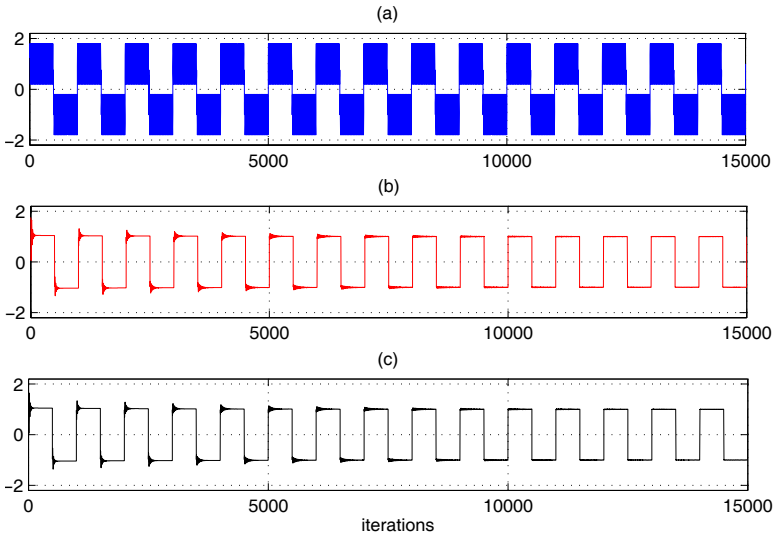

Figure 3.11: Comparisons between NRVSS and MRVSS algorithms applied to noise cancellation application. (a) the desired square wave plus a pure sinusoidal noise; (b) NRVSS processed square wave; and (c) MRVSS processed square wave.

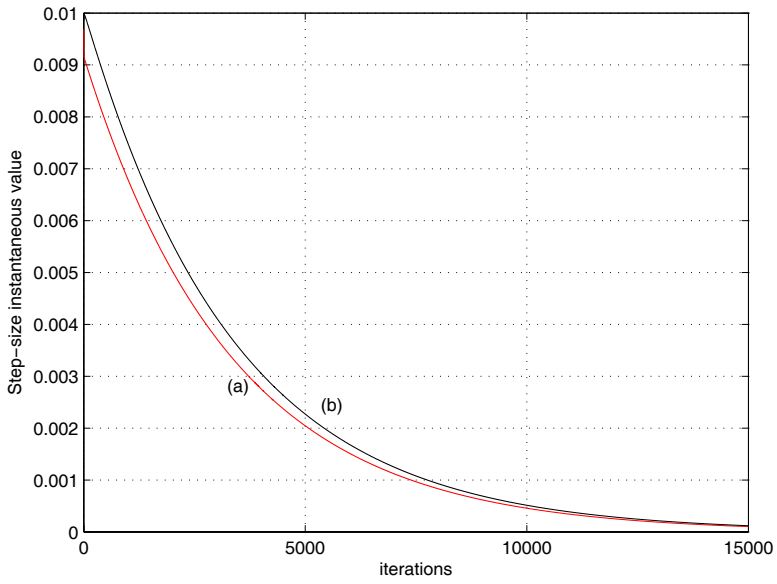

Figure 3.12: Instantaneous values of NRVSS (a) and MRVSS (b) step-sizes for the application of noise cancellation.

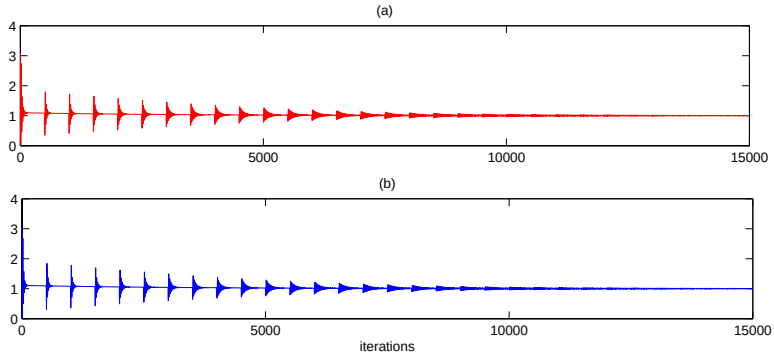

Figure 3.13: MSE values of NRVSS (a) and MRVSS (b) for the application of noise cancellation.

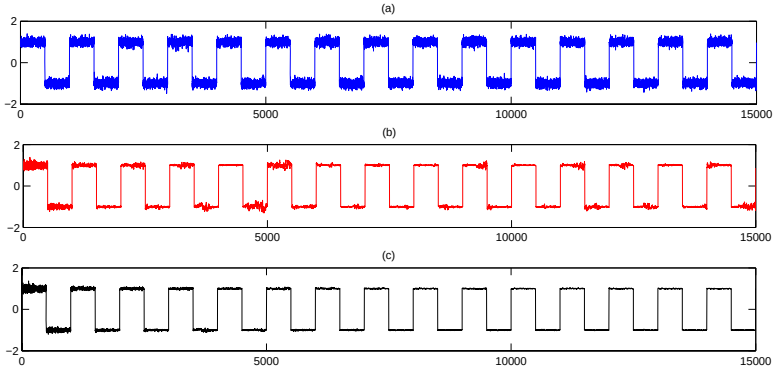

Figure 3.14: Comparisons between NRVSS and MRVSS algorithms applied to noise cancellation application. (a) the desired square wave plus an additive Gaussian noise; (b) NRVSS processed square wave; and (c) MRVSS processed square wave.

3.6 Concluding Remarks

In this chapter, we presented a new variable step-size LMS algorithm with a quotient form. We gave two reasons for the proposal of a quotient form for the adjustment of variable step-size. That is, it can ensure the smooth decrease of the step-size and the determination of steady-state step-size dependent only on the constant parameters. We also gave analysis on the convergence in mean and mean-square, steady-state performance and mean step-size behavior of the resulting MRVSS algorithm. Analysis and computer simulation show that the MRVSS algorithm is less sensitive to the power of the measurement noise, and provides better tracking capability in nonstationary environment when compared to VSS and NRVSS. The additive computational cost is negligible and real-time implementation of MRVSS is realizable. We also gave some design guidelines for the setting of two new added decaying factors. In the next chapter, we move to the variable step-size approach for the NLMS algorithm.

Chapter 4

Variable Step-Size NLMS Algorithms

4.1 Motivation of New Variable Step-size NLMS Algorithms

From the discussion in Chapters 2 and 3, we know that the LMS algorithm is one of the simplest algorithms for adaptive filtering and widely used in many applications. The quotient form of the variable step-size LMS algorithm improves the performance in terms of convergence speed, steady-state fluctuation and tracking capability. Another concern about the LMS algorithm is that it is sensitive to the scaling of its input when choosing a maximum step-size to guarantee stability [1, 39, 54].

The NLMS algorithm, however, solves this problem by normalizing with the power of the input. For the NLMS algorithm, the step-size also governs the convergence speed and the steady-state excess mean-square error. As we discussed in Chapter 3, many variable step-size approaches for the LMS algorithm have been proposed [57–61]. The main criteria is to measure the adaptive filter state based on the output error. For instance, Kwong and Johnston [57] used the square of the prediction error. Aboulnasr and Mayyas [58] used the square of the time-averaged estimate of the output error autocorrelation

at adjacent time. Pazaitis and Constantinides [59] used the time-averaged estimates of the squared and the fourth-order output error. In [61], Haweel adopted the sign of the instantaneous error. Although the variable step-size approaches designed for the LMS algorithm can be applied to the NLMS algorithm, experiments show that they are not as effective as the variable step-size approaches directly designed based on the NLMS algorithm [9, 10, 62]. In these variable step-size approaches for the NLMS algorithm, the derivative of the error functions with respective to the step-size was often considered. For instance, in [60], Ang and Farhang computed the derivative of the gradient power with respect to the step-size. Sulyman and Zerguine [9] used the derivative of the squared instantaneous error with respect to the step-size. In [10, 62], the authors obtained the optimum step-size by minimizing the mean-square deviation (MSD) and presented the estimates. All these variable step-size approaches were claimed to work effectively for the NLMS algorithm. However, to our best knowledge, the combined properties of the additive noise and the input excitation have not been considered in estimating the optimum step-size of the NLMS algorithm.

In this chapter, we make several considerations on the additive noise and the input excitation to estimate the optimum step-size that minimizes the MSD of the NLMS algorithm, and this leads to a new class of variable step-size schemes for the NLMS algorithm. We will show that the developed algorithms with these new schemes have simpler or equivalent forms but achieve better performance compared to the representative proposal of VS-NLMS in [10] and other approaches. We finally demonstrate the performance of the new class of variable step-size NLMS algorithms in the application of system identification in several situations of the input excitation, the additive noise and the channel.

4.2 Design of New Variable Step-Size NLMS Algorithms

Consider a set of observable complex data d_n that fit into the multiple regression model

$$d_n = \mathbf{x}_n^H \mathbf{w}^o + e_n^o \tag{4.1}$$

where $\mathbf{w}^o$ is an unknown parameter vector that is to be estimated, e_n^o accounts for additive noise from system impairments with variance σ_v^2, $\mathbf{x}_n$ denotes input vector with dimension N, and $\mathbf{x}_n^H$ denotes the Hermitian transpose of $\mathbf{x}_n$. We use $\mathbf{w}_n$ to denote a recursive estimate of the unknown parameter $\mathbf{w}^o$ at iteration n.

4.2.1 Optimal Variable Step-Size

The adaptation rule of the NLMS algorithm is given by [1]

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu_n e_n^* \frac{\mathbf{x}_n}{\|\mathbf{x}_n\|^2} \tag{4.2}$$

where μ_n is the step size, and $e_n = d_n - \mathbf{x}_n^H \mathbf{w}_n$. Compared to the update form of the LMS algorithm, the input vector, $\mathbf{x}_n$, is normalized by the input power, $\|\mathbf{x}_n\|^2$. Eq. (4.2) is a complex form of the NLMS algorithm. The variable step-size schemes for the complex NLMS algorithm will be designed in the following sections. The transition from the complex form to real form is straightforward.

We now derive the step-size that minimizes the MSD of the algorithm. Let the weight-error vector be $\boldsymbol{\epsilon}_n = \mathbf{w}^o - \mathbf{w}_n$. Then the update recursion (4.2) can be rewritten as

$$\boldsymbol{\epsilon}_n = \boldsymbol{\epsilon}_{n-1} - \mu_n \frac{e_n^* \mathbf{x}_n}{\|\mathbf{x}_n\|^2} \tag{4.3}$$

Using (4.3) into the definition of MSD [2, p. 325] yields

$$\begin{aligned} E\|\boldsymbol{\epsilon}_n\|^2 &= E\|\boldsymbol{\epsilon}_{n-1}\|^2 - 2\mu_n \mathrm{Re}\left(E\left[\frac{e_n \mathbf{x}_n^H \boldsymbol{\epsilon}_{n-1}}{\|\mathbf{x}_n\|^2}\right]\right) \\ &\quad + \mu_n^2 E\left[\frac{e_n e_n^*}{\|\mathbf{x}_n\|^2}\right] \\ &= E\|\boldsymbol{\epsilon}_{n-1}\|^2 - \triangle(\mu_n) \end{aligned} \tag{4.4}$$

Maximizing the quantity

$$\triangle(\mu_n) = 2\mu_n \mathrm{Re}\left(E\left[\frac{e_n \mathbf{x}_n^H \boldsymbol{\epsilon}_{n-1}}{\|\mathbf{x}_n\|^2}\right]\right) - \mu_n^2 E\left[\frac{e_n e_n^*}{\|\mathbf{x}_n\|^2}\right] \tag{4.5}$$

with respect to μ_n, yields the optimum step-size

$$\mu_n^o = \frac{\mathrm{Re}\left(E\left[\frac{e_n \mathbf{x}_n^H \boldsymbol{\epsilon}_{n-1}}{\|\mathbf{x}_n\|^2}\right]\right)}{E\left[\frac{e_n e_n^*}{\|\mathbf{x}_n\|^2}\right]} \tag{4.6}$$

Note that obtaining the optimum step-size here follows the same way as described in [10, 62] such that the MSD is minimized at each iteration.

4.2.2 New Variable Step-Size NLMS Algorithms

To estimate the optimum step-size, we consider the additive noise sequence and the input data into several cases. Assuming that the additive noise sequence is zero-mean and independent of the input data, μ_n^o is approximated as

$$\mu_n^o \approx \frac{E\left[\frac{\boldsymbol{\epsilon}_{n-1}^H \mathbf{x}_n \mathbf{x}_n^H \boldsymbol{\epsilon}_{n-1}}{\|\mathbf{x}_n\|^2}\right]}{E\left[\frac{\boldsymbol{\epsilon}_{n-1}^H \mathbf{x}_n \mathbf{x}_n^H \boldsymbol{\epsilon}_{n-1}}{\|\mathbf{x}_n\|^2}\right] + \frac{\sigma_v^2}{E\|\mathbf{x}_n\|^2}} \tag{4.7}$$

Case 1: Supposing that the additive noise sequence is identically and independently distributed, and supposing that the excitation $\|\mathbf{x}_n\|^2$ can be approximated by a constant, the optimal step-size can be simplified as follows:

$$\mu_n^o \approx \frac{p_n}{p_n + \sigma_v^2} \tag{4.8}$$

where $p_n \triangleq E[\varepsilon_n]$, and $\varepsilon_n \triangleq \boldsymbol{\epsilon}_{n-1}^H \mathbf{x}_n \mathbf{x}_n^H \boldsymbol{\epsilon}_{n-1}$. Note that for uncorrelated noise sequence it holds that $E[e_n^o e_{n-1}^o] = 0$. Motivated by this fact, we estimate ε_n by using the time-averaged autocorrelation function of the instantaneous error at adjacent time as follows:

$$\hat{\varepsilon}_n = \alpha \hat{\varepsilon}_{n-1} + (1-\alpha) e_n e_{n-1} \tag{4.9}$$

where α is a smoothing factor chosen as $0 \leq \alpha < 1$.

The estimate scheme in (4.9) has also been used as a criterion in [58] to design the variable step-size for LMS, and resulted in excellent performance. Using (4.9), p_n is estimated by

$$\hat{p}_n = \alpha \hat{p}_{n-1} + (1-\alpha)\,|\hat{\varepsilon}_n| \tag{4.10}$$

Using (4.9) and (4.10), the first proposed variable step-size (VSI) for NLMS becomes

$$\mu_n = \mu_{max} \cdot \frac{\hat{p}_n}{\hat{p}_n + C_1} \tag{4.11}$$

where C_1 is a positive constant. From (4.8) and (4.11), we know that C_1 is related to the noise variance σ_v^2, and this quantity can be approximated in many practical applications. Similarly to the proposed scheme in [10], when $\hat{p}_n$ is large in adaptation stage, μ_n tends to μ_{max} for fast convergence. On the other hand, when $\hat{p}_n$ is small in steady-state, the step-size is small for low misadjustment. Thus the conflicting requirements can be met. To guarantee filter stability, μ_{max} is chosen less than 2.

Case 2: Supposing that the additive noise sequence e_n^o is identically and independently distributed, whereas the excitation $\|\mathbf{x}_n\|^2$ is time varying, the terms $\|\mathbf{x}_n\|^2$ in (4.7) cannot be canceled out. In this case, let $p'_n \triangleq E\left[\frac{\varepsilon_n}{\|\mathbf{x}_n\|^2}\right]$, where ε_n has the same definition as in Case 1. Using (4.9) for estimating ε_n, we can then estimate p'_n by time-averaging as follows:

$$\hat{p}'_n = \alpha \hat{p}'_n + (1-\alpha)\frac{|\hat{\varepsilon}_n|}{\|\mathbf{x}_n\|^2} \tag{4.12}$$

Using (4.7), (4.9), and (4.12), the second proposed variable step-size (VSII) for NLMS becomes

$$\mu_n = \mu_{max} \cdot \frac{\hat{p}'_n}{\hat{p}'_n + C_2} \tag{4.13}$$

where C_2 is a positive constant. From (4.7) and (4.13), we know that C_2 is related to $\sigma_v^2 / E\|\mathbf{x}_n\|^2$, and this quantity is inversely proportional to SNR.

Case 3: Supposing that the additive noise sequence is correlated, and supposing that the excitation $\|\mathbf{x}_n\|^2$ can be approximated by a constant, multiplying the numerator and denominator of (7) by $E\|\mathbf{x}_n\|^2$ twice yields

$$\mu_n^o \approx \frac{E\left[\boldsymbol{\epsilon}_{n-1}^H \mathbf{x}_n \|\mathbf{x}_n\|^2 \mathbf{x}_n^H \boldsymbol{\epsilon}_{n-1}\right]}{E\left[\boldsymbol{\epsilon}_{n-1}^H \mathbf{x}_n \|\mathbf{x}_n\|^2 \mathbf{x}_n^H \boldsymbol{\epsilon}_{n-1}\right] + \sigma_v^2 E\|\mathbf{x}_n\|^2} \tag{4.14}$$

Let $\mathbf{p}_n \triangleq \mathbf{x}_n \mathbf{x}_n^H \boldsymbol{\epsilon}_{n-1}$, which is the transformation of $\boldsymbol{\epsilon}_{n-1}$ by matrix $\mathbf{x}_n \mathbf{x}_n^H$. Since $\|\mathbf{p}_n\|^2 = \boldsymbol{\epsilon}_{n-1}^H \mathbf{x}_n \|\mathbf{x}_n\|^2 \mathbf{x}_n^H \boldsymbol{\epsilon}_{n-1}$, the optimal step-size in (14) becomes

$$\mu_n^o = \frac{E\|\mathbf{p}_n\|^2}{E\|\mathbf{p}_n\|^2 + \sigma_v^2 E\|\mathbf{x}_n\|^2} \tag{4.15}$$

Note that for zero-mean noise, it holds that $E[\mathbf{p}_n] = E[\mathbf{x}_n e_n^*]$. Following the approach in [10], $\mathbf{p}_n$ is estimated by time-averaging as follows:

$$\hat{\mathbf{p}}_n = \alpha \hat{\mathbf{p}}_{n-1} + (1-\alpha) e_n^* \mathbf{x}_n \tag{4.16}$$

Using $\|\hat{\mathbf{p}}_n\|^2$ instead of $E\|\mathbf{p}_n\|^2$ in (4.15), which has been shown a good approximation in [10], the third proposed variable step-size (VSIII) for NLMS becomes

$$\mu_n = \mu_{max} \cdot \frac{\|\hat{\mathbf{p}}_n\|^2}{\|\hat{\mathbf{p}}_n\|^2 + C_3} \tag{4.17}$$

where C_3 is a positive constant. From (4.15) and (4.17), we know that C_3 is related to $\sigma_v^2 E\|\mathbf{x}_n\|^2$, and this quantity can be approximated by the product of the noise variance and the input signal energy.

Case 4: Supposing that the additive noise sequence is correlated, and supposing that the excitation $\|\mathbf{x}_n\|^2$ is time varying, we rewrite (4.7) as

$$\mu_n^o \approx \frac{E\left[\frac{\boldsymbol{\epsilon}_{n-1}^H \mathbf{x}_n \|\mathbf{x}_n\|^2 \mathbf{x}_n^H \boldsymbol{\epsilon}_{n-1}}{\|\mathbf{x}_n\|^4}\right]}{E\left[\frac{\boldsymbol{\epsilon}_{n-1}^H \mathbf{x}_n \|\mathbf{x}_n\|^2 \mathbf{x}_n^H \boldsymbol{\epsilon}_{n-1}}{\|\mathbf{x}_n\|^4}\right] + \frac{\sigma_v^2}{E\|\mathbf{x}_n\|^2}} \tag{4.18}$$

Let $\mathbf{p}_n \triangleq \mathbf{x}_n \mathbf{x}_n^H \boldsymbol{\epsilon}_{n-1}$ again, and $p_n'' \triangleq E\left[\|\mathbf{p}_n\|^2 / \|\mathbf{x}_n\|^4\right]$. Following the same way as we did in Case 3, $\mathbf{p}_n$ is estimated by (4.16), and p_n'' is then estimated by time-averaging as follows:

$$\hat{p}_n'' = \alpha \hat{p}_{n-1}'' + (1-\alpha)\|\hat{\mathbf{p}}_n\|^2 / \|\mathbf{x}_n\|^4 \tag{4.19}$$

Using (4.16), (4.18), and (4.19), the fourth proposed variable step-size (VSIV) for NLMS becomes

$$\mu_n = \mu_{max} \cdot \frac{\hat{p}_n''}{\hat{p}_n'' + C_4} \tag{4.20}$$

where C_4 is a positive constant, and this quantity can be approximated by the way that is used for approximating C_2.

Compared to the proposed variable step-size scheme for NLMS in [10], the class of proposed variable step-size schemes in this chapter minimize the same quantity of the MSD and perform by the similar way, where the norm of the error vector or the weighted error vector is used as criterion to determine the filter state during the adaptation. For the comparisons of computational complexity and stability, the number of multiplications for calculating the step-size is 7 in VSI-NLMS, 8 in VSII-NLMS, $3N$+3 in VSIII-NLMS, whereas $3N$+4 in VS-NLMS and $3N$+7 in VSIV-NLMS. The VSI-NLMS, VSII-NLMS

and VSIII-NLMS have less computational cost. The VSIII-NLMS and VI-NLMS require the same storage space as the VS-NLMS for storing a vector, and the VSI-NLMS and VSII-NLMS need less storage space since there are no vectors for calculating the step-sizes. The VSI-NLMS and VSIII-NLMS are more numerically stable due to the lack of dividing the input energy during calculating the step-size. For all the proposed schemes, μ_{max} should be chosen as less than 2 to guarantee filter stability.

4.3 Simulation Results

4.3.1 Application to System Identification

The performance of the proposed class of variable step-size NLMS algorithms is evaluated by carrying out the computer simulation in the framework of adaptive system modeling problem. We use a Hamming window to generate the unknown system as a finite impulse response (FIR) lowpass plant with cutoff frequency of 0.5. The adaptive filter and the unknown system are assumed to have the same number of taps equal to 10. The estimate of the MSD defined by $E\left\|\mathbf{w}^o - \mathbf{w}_n\right\|^2$ is plotted via the iteration number n by ensemble averaging over 100 independent trials. The standard NLMS, the VSS-NLMS [9], and the VS-NLMS [10] are chosen for comparison purpose. The parameter $\rho = 8E - 4$ is used for the VSS-NLMS [9], and the other parameter settings for the various algorithms in all the tested cases are shown in Table 4.1. For all the cases except Case b, the additive noise is zero-mean white Gaussian sequence with variance of $10^{-2.5}$.

Case a: We use a zero-mean, white Gaussian sequence for the input signal. The plot of MSD for the various algorithms is shown in Figure 4.1. From Figure 4.1, we can see that all the VS NLMS algorithms except VSS-NLMS [9] can achieve the convergence rate that the standard NLMS algorithm achieves at the step-size equal to one. Compared to VS-NLMS [10], the proposed VSI-NLMS and VSII-NLMS achieve the same misadjustment but have slightly slower convergent periods, and the proposed VSIII-NLMS and VSIV-NLMS show slightly faster convergence rate.

Case b: In this example, we also consider the white Gaussian input but a correlated additive noise sequence which is generated by $v_n e_n^o = 0.9 e_{n-1}^o + \omega_n$, where ω_n is a zero-mean white Gaussian noise sequence with variance of $10^{-2.5}$. Figure 4.2 shows the MSD

Table 4.1: Parameter Settings for the Various VS NLMS Algorithms in Different Cases

Algorithm	Set	case a	case b	case c	case d	case e
NLMS	μ	1.0	1.0	1.0	1.0	1.0
VSS-NLMS [9]	μ_0	0.001	0.001	0.1	0.001	1.0
VS-NLMS [10]	C	5E−5	8E−4	5E−4	0.002	1E−5
VSI-NLMS	C_1	0.01	0.02	0.04	1.0	0.02
VSII-NLMS	C_2	5E−4	0.006	0.004	0.1	5E−4
VSIII-NLMS	C_3	0.03	0.005	0.03	0.15	0.001
VSIV-NLMS	C_4	5E−4	0.003	5E−4	0.001	1E−5

for the various algorithms. The convergence rate of the proposed VSI-NLMS and VSII-NLMS become slower compared to the other two proposed schemes as well as VS-NLMS [10], but is much faster than that of VSS-NLMS [9] for comparable misadjustment.

Case c: Here we study the various algorithms in tracking a constantly varying channel which is generated by a random walk model: $h_n = h_{n-1} + \delta_n$, where δ_n is a white Gaussian sequence with zero-mean and variance of 10^{-4}. The experimental result is plotted in Figure 4.3. It is seen that all the algorithms have identical convergence rate and equivalent tracking capabilities.

Case d: In this example, the input signals are obtained by filtering a zero-mean, white Gaussian sequence through a first-order system $G(z) = 1/1 - 0.9z^{-1}$. The plot of the MSD is given in Figure 4.4. Observe that the proposed VSI-NLMS and VSII-NLMS that are estimated using the uncorrelation property of the additive noise outperform the other schemes, and all the new proposed schemes show better performance than the other two schemes.

Case e: In this last example, we consider the input signals which are obtained by passing a zero-mean, white Gaussian sequence through a third-order system $G(z) =$

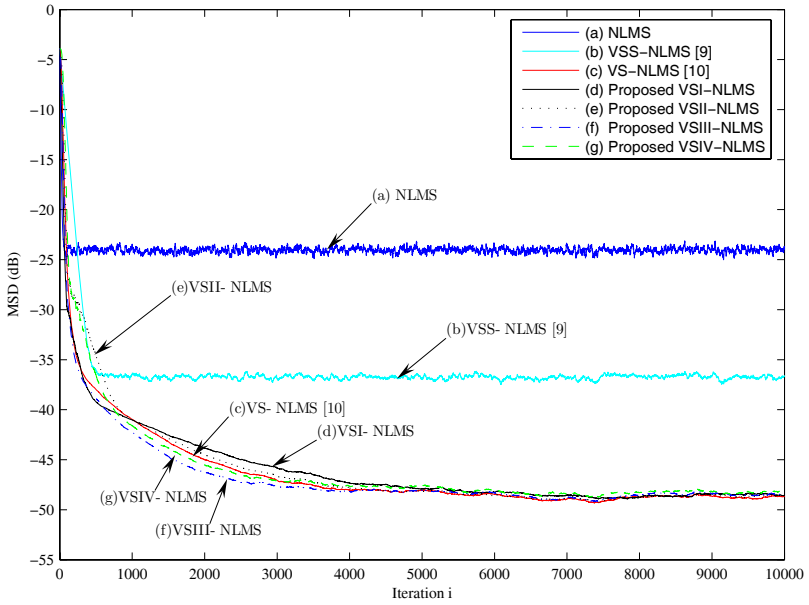

Figure 4.1: Plot of the MSD for the various algorithms with white input and uncorrelated noise.

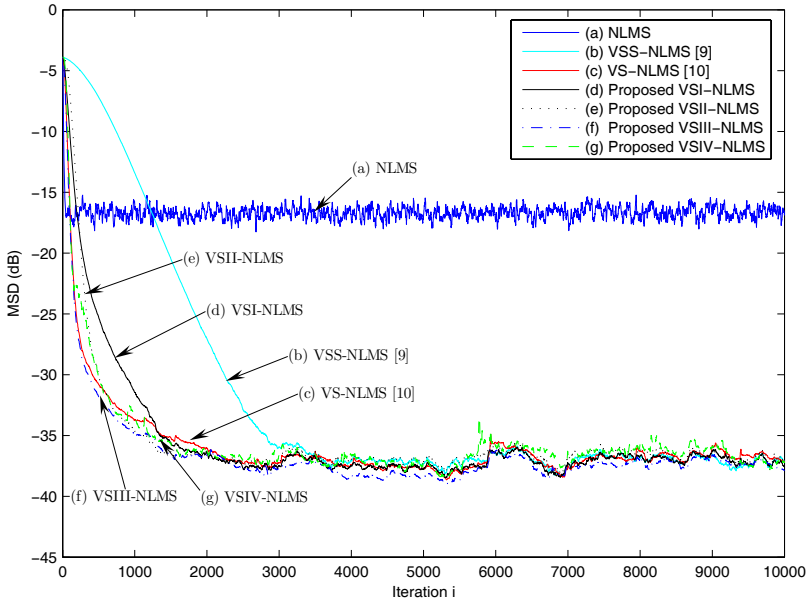

Figure 4.2: Plot of the MSD for the various algorithms with white input and correlated noise.

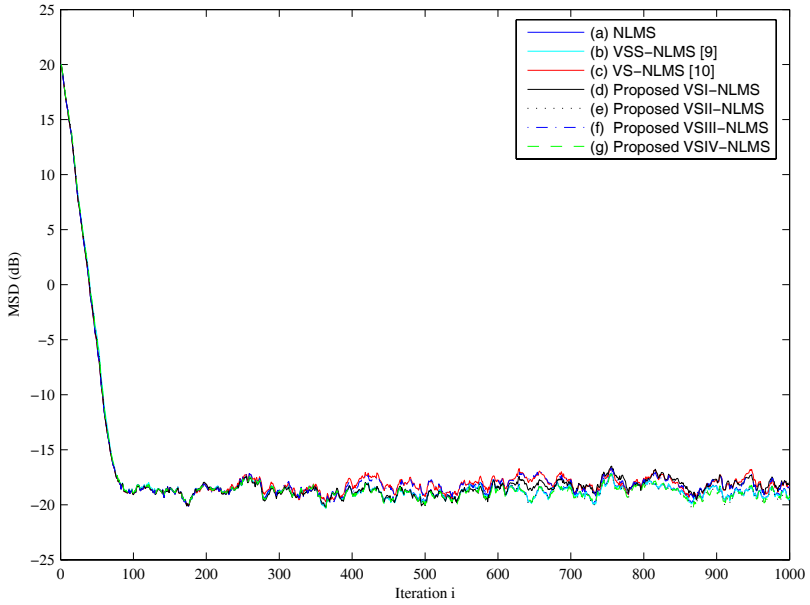

Figure 4.3: Plot of the MSD for the various algorithms with white input and uncorrelated noise for constantly varying channel.

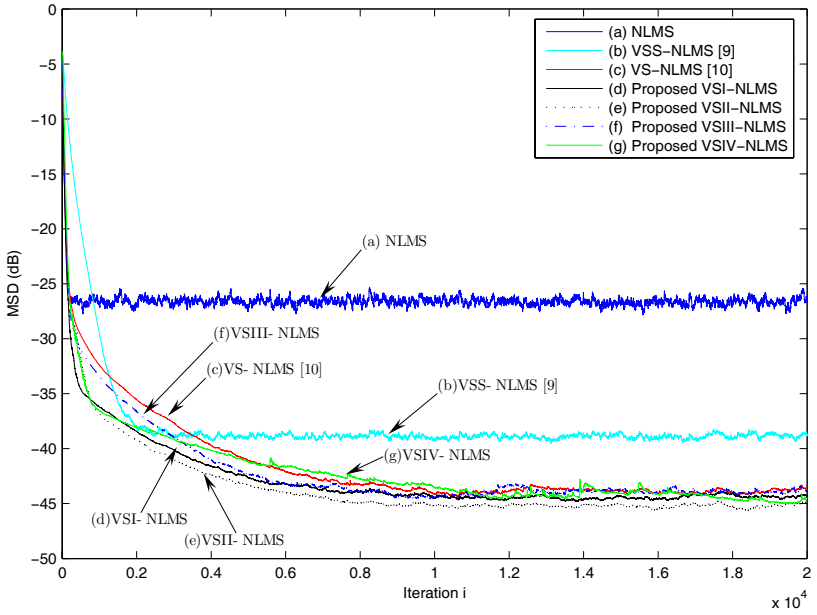

Figure 4.4: Plot of the MSD for the various algorithms with nonwhite Gaussian input AR(1) and uncorrelated noise.

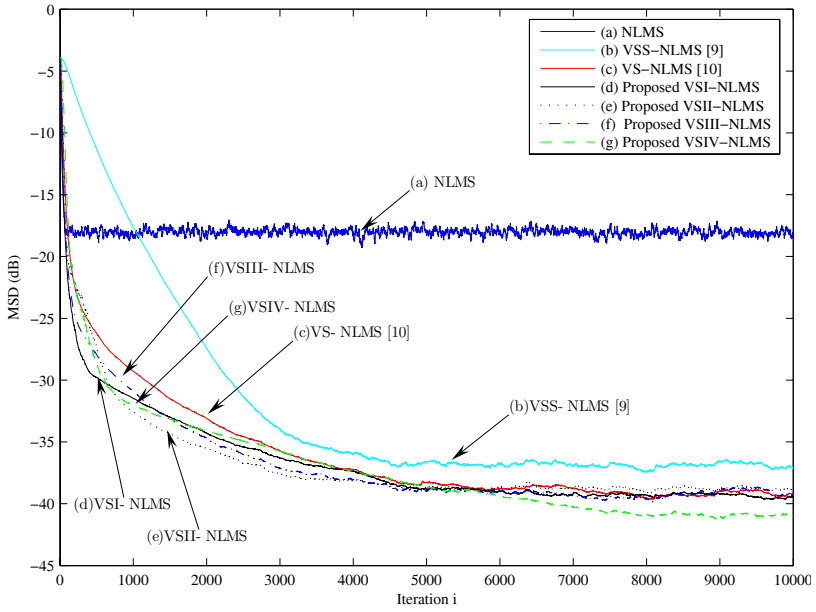

Figure 4.5: Plot of the MSD for the various algorithms with nonwhite Gaussian input AR(3) and uncorrelated noise.

$0.44/1-1.5z^{-1}+z^{-2}-0.25z^{-3}$. As shown in Figure 4.5, the convergence properties of all the algorithms are similar to those demonstrated in Case d, where the proposed schemes outperform VS-NLMS [10] and VSS-NLMS [9].

Based on the above simulation results, we summarize that all the compared schemes achieve the same tracking capability. Among all the schemes the proposed VSI-NLMS and VSII-NLMS perform better in the situations of the correlated input of AR model and uncorrelated noise with reduced complexity, and the proposed VSIII-NLMS and VSIV-NLMS show equivalent or better performance compared to VS-NLMS [10], and much better performance than VSS-NLMS [9] and NLMS. Following the comparisons in [10], the proposed schemes will also outperform the schemes proposed in [57–59, 62]. These results show that the estimates of the optimum step-size using the uncorrelation properties of the additive noise and the constant input excitation perform satisfactorily. In the real applications, criterion is required to pick one of the proposed variable step-size NLMS algorithm for overall good performance. Based on the design and simulation results, we make some conclusions as follows. For the stationary, time-invariant input signal, if the additive noise is identically and independently distributed, uncorrelated with

the input, the VSI-NLMS algorithm can be selected for its simplicity and reasonably good performance. If the additive noise is correlated with itself, the VSIII-NLMS algorithm can be selected for better performance. For the nonstationary, time-varying input signal, if the additive noise is identically and independently distributed, uncorrelated with the input, the VSII-NLMS algorithm can be selected. And if the additive noise is correlated with itself, the VSIV-NLMS algorithm turns out to be the choice for good performance but with relatively high computational cost.

4.4 Concluding Remarks

In this chapter, we have presented a new class of variable step-size NLMS algorithms. The properties of the additive noise and the input excitation are considered to simplify the estimates of the optimum step-size that minimizes the MSD of NLMS. The resulted schemes give simple forms and improved filter performance. In the next chapter, we discuss the LMS algorithms in transform domain that improve the convergence by pre-processing the inputs for reduced input eigenvalue spread.

Chapter 5

Transform-Domain LMS Algorithms[1]

5.1 Introduction

In the previous chapters we noted that despite the simplicity and robustness of the LMS and NLMS adaptive algorithms, their convergence behaviors depend on the eigenvalue spread of the correlation matrix, $\mathbf{R}$, of the input process: the larger the input eigenvalue spread, the slower the convergence of the algorithms [1, 34, 68]. The variable step-size approaches we discussed in the previous chapters are not sufficient to address this problem. The stability conditions of the LMS algorithm prevent the maximum step-size being large for large eigenvalue spread of the input autocorrelation matrix. Therefore, the convergence mode corresponding to the smallest eigenvalue may require a very long convergence time.

It was shown that the eigenvalues of the correlation matrix, $\mathbf{R}$, are directly related to the power spectral density of the input process. Hence, the convergence behavior of the LMS algorithm can be alternatively explained to be frequency dependent in the sense

[1]The results of this chapter were first published in IEEE Trans. on Signal Processing, vol. 57, no. 1, pp. 119–130, January 2009.

that for an adaptive filter the weight convergence at a given frequency bin depends on the relative value of the power spectral density of the underlying input signal at that frequency bin. A large value of the input power spectral density at that frequency will result fast convergence. On the other hand, the LMS algorithm converges very slowly over those frequency bins in which the adaptive filter is poorly excited. This concept has already been clarified with the experiments in the previous chapters. A white Gaussian input process makes the LMS algorithm converge fast whereas a highly correlated first-order autoregressive process slows down the convergence of the LMS algorithm.

The solution may be intuitively considered for solving the above mentioned problem of slow convergence of the LMS algorithm in time domain. That is, a set of bandpass filters are employed to partition the adaptive filter input into a few subbands and use a normalization process to equalize the energy content in each of the subbands. The preprocessed subband signals can then be used for the input of the LMS adaptive filter. This concept has been already adopted into the design of the LMS algorithms in transform domain by using an fixed, data-independent orthogonal transform.

In this chapter, we study this family of adaptive algorithms which operate in a transform domain. Two categories of the transform-domain adaptive algorithms have been developed in the literature. One category is the block implementation of the LMS algorithm in the frequency domain [35, 69–71], which takes advantages of the computational efficiency of fast Fourier transform and inverse Fourier transform during the computation of circular convolution. The convergence of this class of algorithms can be greatly improved by choosing a learning rate that is inversely proportional to the energy of the corresponding input. Another category is the non-block implementation of the LMS algorithm first introduced by Narayan in [72] and subsequently developed and analyzed in [73–81]. This class of algorithms improve the convergence by preprocessing the inputs with a fixed data-independent orthogonal transform and power normalization, and are referred to as transform-domain LMS algorithms. The orthogonal transforms already used in this class of algorithms include the discrete Fourier transform (DFT), discrete cosine transform (DCT), discrete sine transform (DST), discrete Hartely transform (DHT), discrete Wavelet transform (DWT), Walsh-Hadamard transform (WHT) etc. The resulting algorithms are generally called DFT-LMS, DCT-LMS, etc. Since they can process the input data in real time and have low computational complexity comparable to the standard

LMS adaptive algorithm with partial update techniques[74, 75, 82] and the LMS spectrum analyzers [83–86], this class of transform-domain LMS adaptive algorithms become more and more popular, and are of our interest in this thesis.

In this chapter, we study the stability and convergence performance of the algorithms DFT-LMS, DCT-LMS, DHT-LMS and DST-LMS for autoregressive (AR) input process of second order. We first derive a simple form for the mean-square stability of the algorithms and formulate the steady-state performance of the algorithms. Next, we show that for first-order AR process with root $\rho \in (-1, 1)$, the eigenvalue spread after DST and power normalization tends to $1/(1-\rho^2)$. The result shows a better convergence performance of DST-LMS than that of DCT-LMS for highpass first-order AR process. We then extend the study to the second-order AR process and to all the four algorithms. Our analysis shows that in general DCT-LMS and DST-LMS provide better convergence than DFT-LMS and DHT-LMS. DCT-LMS provides the best convergence performance with lowpass second-order AR process, and DST-LMS provides the best convergence performance with bandpass and highpass second-order AR processes. DFT-LMS and DHT-LMS always show same performance with such process. The challenge of extending the methodology to higher order AR processes is also discussed. Finally, we verify the theoretical results with the computer simulations.

5.2 Transform-Domain LMS Algorithms in Analytical Forms

5.2.1 Derivation of Analytical Forms

Considering the tap-delayed inputs $x_n, x_{n-1}, \cdots, x_{n-N+1}$, the LMS algorithm directly uses the tap-delayed inputs to adapt the filer coefficients, whereas in the transform-domain LMS algorithms transformations are first performed to transform the tap-delayed inputs $x_n(i)$ into $u_n(i)$ using an $N \times N$ discrete orthogonal transform matrix, and then the transformed signals are normalized with the square root of their estimated power.

The whitened input signal denoted by $v_n(i)$ is then applied to the LMS algorithm for updating filter coefficients.

The block diagram of the transform-domain LMS algorithm is depicted in Figure 5.1. It is seen that the inputs of the LMS filter are decorrelated by a fixed, data-independent orthogonal transformation matrix. To optimize the convergence speed of each individual weight, power normalization is then applied to the learning rate of the LMS algorithm. Various fixed, data-independent transformations have been considered in the literature as mentioned in the section of introduction. In this thesis, we will investigate the transformation matrices of DFT/DCT/DHT/DST. The other transformation matrices like DWT and WHT are lack of closed forms and therefore will not be covered. For DCT, there are more than one transformation forms. The DCT-II was chosen because of its superior performance. Specifically, the four $N \times N$ transformation matrices to be studied are defined as follows:

$$\mathbf{T}_N(i,l) \triangleq \begin{cases} \mathbf{F}_N(i,l) = \sqrt{\frac{1}{N}} e^{j(\frac{2\pi il}{N})} & \text{DFT} \\ \mathbf{H}_N(i,l) = \sqrt{\frac{1}{N}} \left(\cos\left(\frac{2il\pi}{N}\right) + \sin\left(\frac{2il\pi}{N}\right)\right) & \text{DHT} \\ \mathbf{C}_N(i,l) = \sqrt{\frac{2}{N}} \kappa_i \cos\left(\frac{i(l+1/2)\pi}{N}\right) & \text{DCT} \\ \mathbf{S}_N(i,l) = \sqrt{\frac{2}{N+1}} \sin\left(\frac{il\pi}{N+1}\right) & \text{DST} \end{cases} \tag{5.1}$$

where $\mathbf{T}_N$ denotes a transformation matrix which is selected to be a unitary matrix, and $\mathbf{T}_N(i,l)$ is the element in the ith row and lth column with $i, l = 0, 1, \cdots, N-1$. The parameter $\kappa_i = 1/\sqrt{2}$ for $i = 0$ and 1 otherwise.

The transformation process is to apply the transformation matrix $\mathbf{T}_N$ to the input vector $\mathbf{x}_n$ with $\mathbf{x}_n = [x_n, x_{n-1}, \cdots, x_{n-N+1}]$. It results in the transformed vector $\mathbf{u}_n$ given by $\mathbf{u}_n = \mathbf{T}_n \mathbf{x}_n$. Let the weight vector in the transform domain be $\mathbf{w}_n$ which is a transformed version of the weight vector in the time domain. After applying power-normalization to the transformed vector, the updating equation of the transform-domain LMS algorithms is written as (see e.g., [72], [87] and [40]):

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu \mathbf{P}_n^{-1} e_n \mathbf{u}_n^*, \tag{5.2}$$

where the output error e_n is given by

$$e_n = d_n - \mathbf{w}_n^H \mathbf{u}_n, \tag{5.3}$$

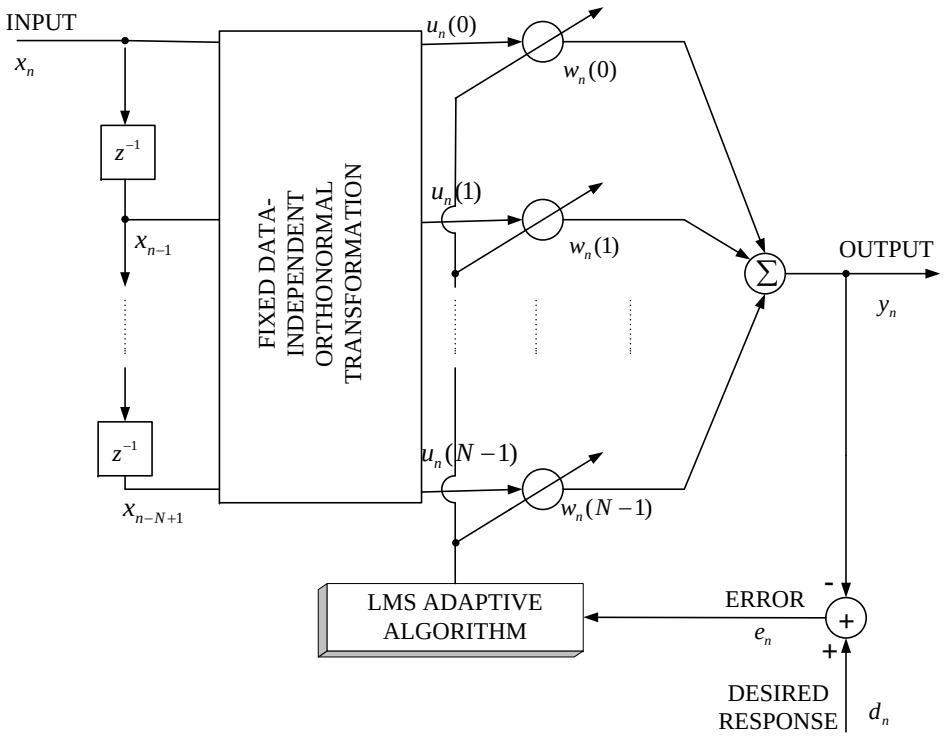

Figure 5.1: Block diagram of transform-domain LMS algorithm.

where d_n denotes the desired response, $\mathbf{w}_n^H \mathbf{u}_n$ gives the filter output of y_n, the superscript H denotes the Hermitian transposition, $\mathbf{u}_n^*$ denotes the complex conjugate of $\mathbf{u}_n$, μ is the step-size, and $\mathbf{P}_n$ is a diagonal matrix of the estimated input powers defined as $\mathbf{P}_n = \text{diag}[\sigma_n^2(i), i = 0, 1, \cdots, N-1]$. Here, $\sigma_n^2(i)$ is the estimated power of the ith input $u_n(i)$, which can be obtained with a low-pass algorithm

$$\sigma_n^2(i) = \beta \sigma_{n-1}^2(i) + (1-\beta)\left|u_n(i)\right|^2, \tag{5.4}$$

with β a constant smoothing factor.

It can be checked that the orthogonal transforms has so-called band-partitioning property from a frequency domain view [38]. This equates the orthognalization property of the transforms in the time domain. This, in turn, implies that the resultant autocorrelation matrix of the transformed vector $\mathbf{u}_n$ is more likely to be an orthogonal matrix.

The power normalization is applied to equalize the eigenvalues of the autocorrelation matrix, thereby, improving the convergence behavior of the LMS algorithm in the transform domain.

From the implementation point of view of the equation (5.2), it can be considered that the power normalization is applied on the fixed step-size. This, in turn, provides multiple learning rates for the weight adaptation. However, for the convergence performance and stability analysis of the algorithm this consideration will increase the analysis difficulty. In order to make the analysis tractable, the power normalization has been equivalently performed on the transformed vector $\mathbf{u}_n$, which resulted in a simplified form equivalent to that of the LMS algorithm as seen in [88] and [38]. Since this form is highly related to our analysis results, we now give a brief derivation of it.

Premultiplying both sides of (5.2) by $\mathbf{P}_n^{1/2}$, we yield

$$\mathbf{P}_n^{1/2}\mathbf{w}_{n+1} = \mathbf{P}_n^{1/2}\mathbf{w}_n + \mu\mathbf{P}_n^{-1/2}e_n\mathbf{u}_n^*, \tag{5.5}$$

As long as the input process is not too non-stationary, it can be assumed that the estimated input power $\sigma_{n+1}^2(i)$ is close enough to $\sigma_n^2(i)$ for $i = 0, 1, \cdots, N$, which leads to the simplification of the equation (5.5) into

$$\tilde{\mathbf{w}}_{n+1} = \tilde{\mathbf{w}}_n + \mu e_n\mathbf{v}_n^*, \tag{5.6}$$

where $\tilde{\mathbf{w}}_n \triangleq \mathbf{P}_n^{1/2}\mathbf{w}_n$, $\mathbf{v}_n \triangleq \mathbf{P}_n^{-1/2}\mathbf{u}_n$, and the output error e_n is reformed as

$$\begin{aligned} e_n &= d_n - \mathbf{w}_n^H\mathbf{u}_n \\ &= d_n - \mathbf{w}_n^H\mathbf{P}_n^{1/2}\mathbf{P}_n^{-1/2}\mathbf{u}_n \\ &= d_n - \tilde{\mathbf{w}}_n^H\mathbf{v}_n. \end{aligned} \tag{5.7}$$

The weight update equations (5.6) and (5.7) suggest that the transform-domain LMS algorithm has an equivalent form of a conventional LMS algorithm with the normalized input vector. The significance of this result is that the analytical results of the conventional LMS algorithm can immediately be applied to evaluate the solution of the transform-domain LMS algorithms. In the mean square error sense, the weight vector $\tilde{\mathbf{w}}_n$ in (5.6) converges to $\mathbf{P}^{1/2}\mathbf{T}_N\mathbf{w}_o$, where $\mathbf{P}$ is defined as $\mathbf{P} \triangleq E[\mathbf{P}_n]$, and $\mathbf{w}_o$ is known

as the Wiener solution that provides the best mapping from the input vector $\mathbf{x}_n$ to the output y_n. Applying inverse transformation and power normalization $\mathbf{T}_N^{-1}\mathbf{P}^{-1/2}$ to the solution of (5.6), the Wiener solution in the time domain is then obtained. This indicates that the two forms of (5.2) and (5.6) provide equivalent solutions in stationary and mildly nonstationary environment.

The form of (5.6) is thus referred to as an analytical form of the transform-domain LMS algorithm. It is noted that the analytical forms (5.6) and (5.7) can only be used for analyzing the transform-domain LMS algorithms, it is not advised to implement these forms in practice for adapting an LMS algorithm. This is because the variances of the power estimates can never be completely reduced to zero. Therefor, the normalized signals are likely to be noisy [89].

It is, however, advantageous for us to perform some explanation and mathematical analysis of these algorithms. From the analytical forms of the transform-domain LMS algorithms, we can see that two more stages are introduced for the transformation and power normalization compared to the LMS algorithm. These two stages ensure that the time domain signals are orthogonally transformed into the frequency domain, in which the power of the input signals is decomposed and redistributed to the different frequency bins. For each frequency bin, the input power is estimated and normalized such that all the frequency bins carry equal values of power. As a result, the correlation nature of the input signals x_n is diminished and the convergence can be improved. However, the fixed orthogonal DFT/DHT/DCT/DST transforms are not perfect decorrelators and they produce some power leakage from each frequency bin to the others. In the next section, we brief the intuitive justifications of the transformation and power normalization of the transform-domain LMS algorithms.

5.2.2 Filtering and Geometrical View of Transform-Domain LMS Algorithms

The filter theory is applied here to explain intuitively the mechanism of transform-domain LMS algorithm. In the $N \times N$ transformation matrix $\mathbf{T}_N$, each row of the matrix can

be characterized by an impulse response $h_i(l) = \mathbf{T}_N(i,l)$ and the corresponding transfer function is given by

$$H_i(\omega) = \sum_{l=0}^{N-1} h_i(l) e^{j\omega l}. \tag{5.8}$$

For the DFT and DCT, $H_i(\omega)$ can be easily obtained and given as

$$H_i(\omega) = \begin{cases} \sqrt{\dfrac{1}{N}} \dfrac{1 - e^{-j\omega n}}{1 - e^{-j\omega} e^{j\frac{2\pi i}{N}}} & \text{DFT} \\ \sqrt{\dfrac{2}{N}} \kappa_i \cos\left(\dfrac{i\pi}{2N}\right) \dfrac{(1 - e^{-j\omega})(1 - (-1)^i e^{-j\omega N})}{1 - 2\cos(\dfrac{i\pi}{N}) e^{-j\omega} + e^{-2j\omega}} & \text{DCT} \end{cases} \tag{5.9}$$

From the transfer function of DFT and DCT as well as their magnitude plots for numerical values of N, it can be observed that the DFT and DCT can be seen as a bank of bandpass filters with different central frequencies, main lobes and side lobes, and different leakage properties [88]. Due to the presence of side lobes the unwanted spectral leakage is introduced. Depending on the characteristics of the input signals, the side lobes of the filterbank will contribute varying amount of spectral leakage. For lowpass signals, the DCT filterbank is expected to perform better than the DFT-based one because the DCT filterbank has significantly smaller low frequency side lobes.

The transform-domain LMS algorithm can also be illustrated geometrically. Figure (5.2) shows three slices through different MSE performance surfaces. It illustrates that a unitary transformation of the inputs rotates this hyperellipsoid and brings it into approximate alignment with the coordinate axes. It is seen that the alignment is imperfect due to the leakage in the transforms. The power normalization stage forces the ellipsoid to be round-shaped, which means that it does modify the eigenvalue spread of the transformed autocorrelation matrix. The better the alignment of the hyperellipsoid with the coordinate axes, the more efficient the power normalization will be.

Although the decorrelation natures of these orthogonal transforms can be generally explained based on the corresponding transfer functions, theoretical analysis is necessary to mathematically show their orthogonalizing capabilities since general guidelines for selection of the transformation is highly desired. In this chapter, we will deal with the stability and convergence performance of the transform-domain LMS algorithm with AR input process, specifically, for the second-order AR process.

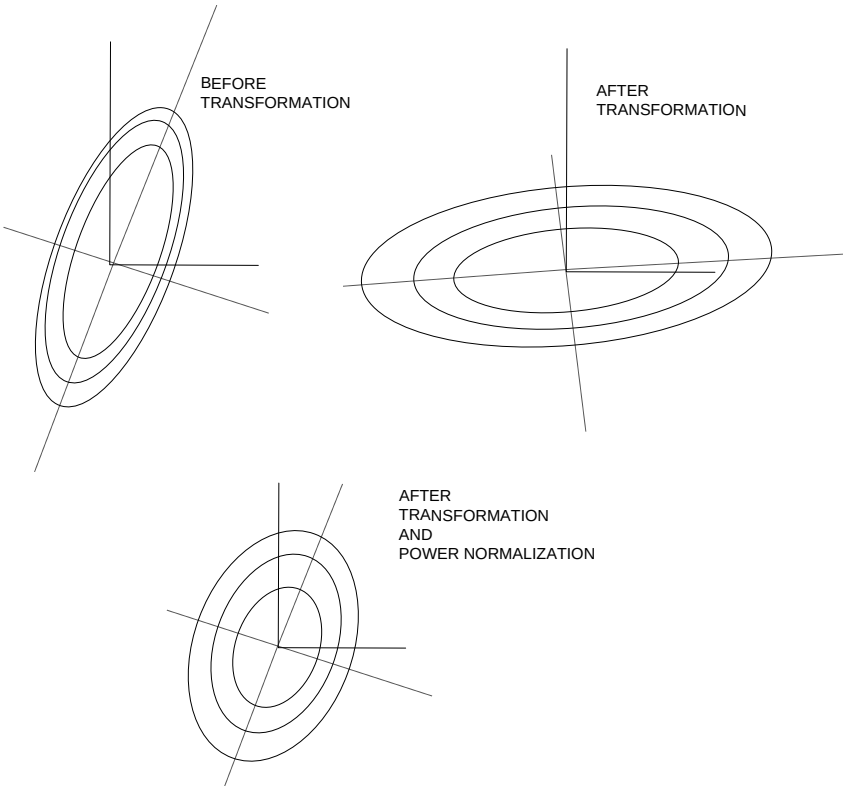

Figure 5.2: Block diagram of transform-domain LMS algorithm.

5.3 Analysis of Stability and Steady-State Performance

For the wide-sense stationary input process, such as the AR process, the form of (5.6) can be deduced for making analysis on the stability and steady-state performance. Based on the simplified form of (5.6), the stability condition for the LMS algorithm presented in [40] equivalently guarantees mean-square convergence of the transform-domain LMS algorithms. The condition for mean-square convergence of the transform-domain LMS algorithms has also been studied in [90]. As a consequence, it can be shown that the forms of (5.2) and (5.6) are mean-square stabilized with the following sufficient condition

$$0 < \mu < \frac{2}{3\mathrm{Tr}(\mathbf{V}_N)}. \tag{5.10}$$

where $\mathrm{Tr}(\cdot)$ denotes the trace of matrix and the autocorrelation matrix $\mathbf{V}_N$ is defined as

$$\mathbf{V}_N = E[\mathbf{v}_n \mathbf{v}_n^H] = (\mathrm{diag}\,\mathbf{U}_N)^{-1/2}\mathbf{U}_N(\mathrm{diag}\,\mathbf{U}_N)^{-1/2} \tag{5.11}$$

with $\mathbf{U}_N = E[\mathbf{u}_n \mathbf{u}_n^H] = \mathbf{T}_N \mathbf{R}_N \mathbf{T}_N^H$ and $\mathbf{R}_N = E[\mathbf{x}_n \mathbf{x}_n^H]$. The trace of $\mathbf{V}_N$ can be obtained as follows

$$\begin{aligned}\mathrm{Tr}(\mathbf{V}_N) &= \mathrm{Tr}((\mathrm{diag}\,\mathbf{U}_N)^{-1/2}\mathbf{U}_N(\mathrm{diag}\,\mathbf{U}_N)^{-1/2}) \\ &= \mathrm{Tr}((\mathrm{diag}\,\mathbf{U}_N)^{-1/2}(\mathrm{diag}\,\mathbf{U}_N)^{-1/2}\mathbf{U}_N) \\ &= N,\end{aligned} \tag{5.12}$$

where N is the length of algorithm. The sufficient condition of (5.10) is therefore led to the following simple form:

$$0 < \mu < \frac{2}{3N}. \tag{5.13}$$

It is observed that the stability condition on the step-size is only related to the algorithm length, and is therefore independent of the input data, which is quite similar to the normalized LMS algorithm [1].

To measure the steady-state performance of an adaptive algorithm, the misadjustment defined as $\mathcal{M} = \frac{\xi_\infty - \xi_{min}}{\xi_{min}}$ is to be computed. For very small step-size, the misadjustment of the LMS algorithm was derived in [1] with Butterweck's iterative procedure [45] as discussed in Chapter 2. Since the form of (5.6) is equivalent to that of the LMS algorithm, the result for the LMS algorithm can be therefore used for the transform-domain LMS algorithms such that the misadjustment is given by

$$\mathcal{M} \approx \frac{\mu}{2}\mathrm{Tr}(\mathbf{V}_N) = \frac{\mu N}{2}. \tag{5.14}$$

It is seen that the misadjustment is only related to the algorithm length and independent of the input data. The justification for the theoretical results in this section is given in the later evaluation section.

5.4 Convergence Analysis of Transform-Domain LMS Algorithms with AR Inputs

It is known that the various modes of convergence of the transform-domain LMS algorithms are determined by the eigenvalues of $\mathbf{V}_N$ given in (5.11), and the eigenvalue

spread of $\mathbf{V}_N$ governs the convergence performance. Therefore, obtaining the eigenvalue spread of $\mathbf{V}_N$ can help the study of the convergence properties of the transform-domain LMS algorithms.

In the following sections, we will first express the general autocorrelation matrix of AR process and briefly review the analytical results reported in the literature about the asymptotic eigenvalue distributions of $\mathbf{V}_N$ for DFT-LMS, DCT-LMS or DHT-LMS and first-order Markov process. We will then establish the similar results for DST-LMS with first-order AR process. Next, we will extend the derivation to establish results for the various algorithms with more complicated second-order AR process. The extension of the methodology to higher order AR process are also discussed. Finally, we discuss the convergence performance of the various algorithms for both first-order and second-order AR processes based on the obtained asymptotic eigenvalue spreads.

5.4.1 Generalized Autocorrelation Matrix of AR Process

The time-domain description of the AR process is governed by the difference equation given by [1]

$$x_n + a_1 x_{n-1} + a_2 x_{n-2} + \cdots + a_M x_{n-M} = \omega_n, \tag{5.15}$$

where $a_1, a_2, \cdots, a_M$ are coefficients of real or complex valued. Here, we are concerned with the real valued case. The value of M gives the order of the process, and ω_n is usually drawn from a white-noise process of zero mean and variance σ_ω^2. Without loss of generality, the noise variance σ_ω^2 is chosen to make the variance of x_n equal unity. The AR process can thus be considered as white noise passing through an all-pole algorithm with the transfer function

$$\mathcal{H}(z) = \frac{1}{(1-\rho_1 z^{-1})(1-\rho_2 z^{-2})\cdots(1-\rho_M z^{-M})}, \tag{5.16}$$

where the poles of the algorithm $\rho_1, \rho_2, \cdots, \rho_M$ are defined by the roots of the characteristic equation

$$1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_M z^{-M} = 0, \tag{5.17}$$

and they are required to lie inside the unit circle in the z-plane for stability of the model. If the characteristic equation (5.17) is solvable, the autocorrelation function $r(m)$ of the AR process can be generally expressed as [1]

$$r(m) = E[x_n x_{n-m}^*] = \sum_{k=1}^{M} c_k \rho_k^m, \tag{5.18}$$

where $c_1, c_2, \cdots, c_M$ are constants and $|\rho_k| < 1$ for all k. Using (5.18), we may express the $N \times N$ autocorrelation matrix, $\mathbf{R}_N$, of a general Mth order AR process x_n as follows

$$\mathbf{R}_N = \sum_{k=1}^{M} c_k \mathbf{R}_N^k, \tag{5.19}$$

with

$$\mathbf{R}_N^k \triangleq \begin{pmatrix} 1 & \rho_k & \rho_k^2 \cdots & \rho_k^{N-1} \\ \rho_k & 1 & \rho_k^2 \cdots & \rho_k^{N-2} \\ \rho_k^2 & \rho_k & 1 & \\ \vdots & \vdots & \ddots & \vdots \\ \rho_k^{N-1} & \rho_k^{N-2} & \cdots & 1 \end{pmatrix}. \tag{5.20}$$

In the next sections, we will mathematically analyze performance of these algorithms with first-order and second-order AR input processes, by deriving the asymptotic eigenvalue distributions of the input autocorrelation matrix after the orthogonal transformations and power normalization. It is noted that the autocorrelation matrices after the DFT/DHT/DCT/DST transformations and power normalization are no longer Toeplitz, and the asymptotic theory of Toeplitz matrix [91] for computing the eigenvalue spread of the input autocorrelation matrix cannot be applied. The analysis presented in [88] provides a fundamental method for the analysis. Before we start our analysis, we give the definition and theorem introduced in [88], which are to be used during the analysis.

Definition 1: Let $p_N(\lambda)$ and $q_N(\lambda)$ be the eigenvalue distributions of the matrices $\mathbf{A}_N$ and $\mathbf{B}_N$ belonging to the sequences $\mathcal{A}$ and $\mathcal{B}$. The two sequences are said to be asymptotically equivalent in the strong sense iff

$$\lim_{N\to\infty} p_N(\lambda) = \lim_{N\to\infty} q_N(\lambda). \tag{5.21}$$

In this case, it is denoted as $\mathbf{A}_N \approx \mathbf{B}_N$.

Theorem 1: Let $\phi_N = \mathbf{A}_N - \mathbf{B}_N$.

$$\text{If } \lim_{N\to\infty} rank(\phi_N) = 0, \text{ then } \mathbf{A}_N \approx \mathbf{B}_N. \tag{5.22}$$

Theorem 1 provides that if the rank of ϕ_N tends to zero as N tends to infinity, all the eigenvalues of ϕ_N converge to zero. In this case, we can infer that the two matrices $\mathbf{A}_N$ and $\mathbf{B}_N$ have the same asymptotic eigenvalue distributions. Such a matrix ϕ_N is referred to as a rank zero *perturbation*.

5.4.2 Previous Results on First-Order AR Inputs

It can easily be checked that the first-order AR process and the first-order Markov process are equivalent when M is equal to one and the constant c_1 is made to be one. Therefore, they have the same forms of autocorrelation matrix $\mathbf{R}_N$. The results presented in [88] and [49] for first-order Markov process are thus also valid for the first-order AR process. These results can be generally described as follows. The author of [88] showed that for first-order Markov process with correlation parameter of $\rho \in [0, 1]$, the eigenvalues of the transformed process v_n in DFT-LMS all converge to one, except two of them, one converges to $1/(1+\rho)$ and another one converges to $1/(1-\rho)$, and the eigenvalues in DCT-LMS also all converges to one except that two of them converge to $1/(1+\rho)$. The eigenvalue spread achieved by DFT-LMS tends to $(1+\rho)/(1-\rho)$, and to $1+\rho$ by DCT-LMS. In [49], the author showed that the eigenvalue spread of $\mathbf{V}_N$ achieved by DHT-LMS tends to $(1+|\rho|)/(1-|\rho|)$ for $\rho \in (-1, 1)$ ($|\cdot|$ stands for the absolute value), which is the same as DFT-LMS for $\rho \in [0, 1)$. It can be checked that for $\rho \in (-1, 0]$ the eigenvalue spreads achieved by DFT-LMS and DCT-LMS become to $(1-\rho)/(1+\rho)$ and $1/(1+\rho)$, respectively. The negative ρ does not deteriorates convergence performance of DFT-LMS and DHT-LMS, but deteriorates convergence performance of DCT-LMS. The results will be the same for first-order AR process with the root $\rho \in (-1, 1)$.

5.4.3 DST-LMS with First-Order AR Inputs

The autocorrelation matrix of v_n with DST-LMS is expressed as

$$\mathbf{V}_N = (\operatorname{diag} \mathbf{U}_N)^{-1/2} \mathbf{U}_N (\operatorname{diag} \mathbf{U}_N)^{-1/2}, \tag{5.23}$$

with $\mathbf{U}_N = \mathbf{S}_N \mathbf{R}_N \mathbf{S}_N^H$. Solving for the asymptotic eigenvalues λ of $\mathbf{V}_N$ in (5.23) is to solve for λ from the equation:

$$\lim_{N\to\infty} \det(\mathbf{V}_N - \lambda \mathbf{I}_N) = 0, \tag{5.24}$$

where $\mathbf{I}_N$ standing for the identity matrix of size N.

Eq. (5.24) can be simplified by matrix theory. Substituting $\mathbf{U}_N$ and (5.23) into (5.24) obtains

$$\lim_{N\to\infty} \det\left((\operatorname{diag}\mathbf{U}_N)^{-\frac{1}{2}} \mathbf{S}_N \mathbf{R}_N \mathbf{S}_N^H (\operatorname{diag}\mathbf{U}_N)^{-\frac{1}{2}} - \lambda \mathbf{I}\right) = 0. \tag{5.25}$$

Multiplying the terms $\mathbf{S}_N^H (\operatorname{diag}\mathbf{U}_N)^{1/2}$ to the left side and $(\operatorname{diag}\mathbf{B}_N)^{1/2}\mathbf{S}_N$ to the right side of (5.25) and using the unitary property of matrix $\mathbf{F}_N$, it leads (5.25) to the following generalized eigenvalue problem [92]:

$$\lim_{N\to\infty} \det(\mathbf{R}_N - \lambda \mathbf{D}_N) = 0, \tag{5.26}$$

where $\mathbf{D}_N$ is given by

$$\mathbf{D}_N = \mathbf{S}_N^H \operatorname{diag}(\mathbf{S}_N \mathbf{R}_N \mathbf{S}_N^H) \mathbf{S}_N. \tag{5.27}$$

It was proved in [88], the inverse problem of (5.27) is advantageous to be solved, which is

$$\lim_{N\to\infty} \det(\mathbf{R}_N^{-1}\mathbf{D}_N - \lambda^{-1}\mathbf{I}_N) = 0. \tag{5.28}$$

The advantage of solving (5.28) comes from the well organized structure of the matrix product $\mathbf{R}_N^{-1}\mathbf{D}_N$, which can be observed in Figures 6 and 8 of [88]. The problem (5.28) can be made simple to solve by finding a simple matrix of $\tilde{\mathbf{X}}_N$ such that $\tilde{\mathbf{X}}_N \approx \mathbf{X}_N = \mathbf{R}_N^{-1}\mathbf{D}_N$. That is, the matrices $\tilde{\mathbf{X}}_N$ and $\mathbf{X}_N$ are asymptotically equivalent in strong sense when N approaches to infinity as defined by the *Definition 1.* After proceeding computations and plotting of $\mathbf{X}_N$ for different values of N, we found the following candidate for $\tilde{\mathbf{X}}_N$:

$$\tilde{\mathbf{X}}_N = \begin{pmatrix} 1-\rho^2 & -\rho^3 & -\rho^4 & \cdots & -\rho^N & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & & & \\ \vdots & \vdots & & \ddots & \vdots & \\ 0 & & \cdots & 0 & 1 & 0 \\ 0 & -\rho^N & \cdots & -\rho^4 & -\rho^3 & 1-\rho^2 \end{pmatrix}. \tag{5.29}$$

Here, we use ρ for the root of $1+a_1z^{-1}=0$. The proof for the validation of $\tilde{\mathbf{X}}_N \approx \mathbf{X}_N = \mathbf{R}_N^{-1}\mathbf{D}_N$ can be established by equivalently proving $\mathbf{S}_N\mathbf{R}_N\tilde{\mathbf{X}}_N\mathbf{S}_N^H \approx \operatorname{diag}\mathbf{U}_N$.

Using the expressions of $\mathbf{R}_N$, the autocorrelation matrix of first-order AR process, and the matrix $\tilde{\mathbf{X}}_N$ in (5.29), the elements of $\mathbf{R}_N\tilde{\mathbf{X}}_N$ are given by

$$\mathbf{R}_N\tilde{\mathbf{X}}_N(k,l) = \sum_{i}^{N-1}\mathbf{R}_N(k,i)\tilde{\mathbf{X}}_N(i,l) =$$

$$\begin{cases} \rho^k - \rho^{k+2} & l=0 \\ \rho^{|k-l|} - \rho^{k+l+2} - \rho^{2N-k-l} & 1\le l\le N-2 \\ \rho^{N-k-1} - \rho^{N-k+1} & l=N-1. \end{cases} \tag{5.30}$$

Let the matrix $\tilde{\mathbf{Y}}_N$ be defined as

$$\tilde{\mathbf{Y}}_N(k,l) = \rho^{|k-l|} - \rho^{k+l+2} - \rho^{2N-k-l}. \tag{5.31}$$

It turns out that $\tilde{\mathbf{Y}}_N \approx \mathbf{R}_N\tilde{\mathbf{X}}_N$ since they differ only by $\mathcal{O}(\rho^N)$. The original problem is thus equivalent to proving $\mathbf{S}_N\tilde{\mathbf{Y}}_N\mathbf{S}_N^H \approx \operatorname{diag}\mathbf{U}_N$. Following the similar procedures as made in Section B.2. of Appendix B in [88], the matrix $\tilde{\mathbf{Y}}_N$ can be decomposed as

$$\tilde{\mathbf{Y}}_N = \mathbf{R}_N - \rho^2\mathbf{G}_N^{\rho} - \rho^{2N}\mathbf{G}_N^{1/\rho}, \tag{5.32}$$

where the matrix $\mathbf{G}_N$ is similarly defined as

$$\mathbf{G}_N^{\rho}(k,l) \triangleq \rho^{k+l} \tag{5.33}$$

and the matrix $\mathbf{G}_N^{1/\rho}$ is obtained by replacing ρ by $1/\rho$ in (5.33). Therefore, the matrix $\mathbf{A}_N$ with definition of $\mathbf{A}_N \triangleq \mathbf{S}_N\tilde{\mathbf{Y}}_N\mathbf{S}_N^H$ can be written as the sum of three matrices ($\mathbf{L}_N, \mathbf{M}_N$, and$\mathbf{U}_N$), where $\mathbf{L}_N = -\rho^2\mathbf{S}_N\mathbf{G}_N^{\rho}\mathbf{S}_N^H$, $\mathbf{M}_N = -\rho^{2N}\mathbf{S}_N\mathbf{G}_N^{1/\rho}\mathbf{S}_N^H$, and $\mathbf{U}_N = \mathbf{S}_N\mathbf{R}_N\mathbf{S}_N^H$. The elements of $\mathbf{L}_N, \mathbf{M}_N$, and $\mathbf{U}_N$ can be computed using the similar method as presented in Appendix B of [88]. The details are therefore not shown. The results are given by

$$\mathbf{L}_N(p,q) = -\frac{\frac{\rho^4}{2N+2}\sin\left(\frac{p\pi}{N+1}\right)\sin\left(\frac{q\pi}{N+1}\right)}{\left(1-2\rho\cos\left(\frac{p\pi}{N+1}\right)+\rho^2\right)\left(1-2\rho\cos\left(\frac{q\pi}{N+1}\right)+\rho^2\right)} + \mathcal{O}(\rho^N), \tag{5.34}$$

$$\mathbf{M}_N(p,q) = \frac{(-1)^{p+q}\frac{\rho^2}{2N+2}\sin\left(\frac{p\pi}{N+1}\right)\sin\left(\frac{q\pi}{N+1}\right)}{\left(1-2\rho\cos\left(\frac{p\pi}{N+1}\right)+\rho^2\right)\left(1-2\rho\cos\left(\frac{q\pi}{N+1}\right)+\rho^2\right)} + \mathcal{O}(\rho^N), \tag{5.35}$$

and

$$\mathbf{U}_N(p,q) = \begin{cases} -\frac{1}{2}\mathbf{M}_N(p,q) + \mathcal{O}(\rho^N) & k+l \text{ even}, \ k \neq l \\ \frac{1-\rho^2}{1-2\rho\cos(p\pi/N)+\rho^2} & k = l \\ 0 & k+l \text{ odd}. \end{cases} \tag{5.36}$$

The diagonal elements of $\mathbf{L}_N$ and $\mathbf{M}_N$ decrease as $1/(2N+2)$ as N increases. This guarantees the asymptotic equivalence $\operatorname{diag}(\mathbf{L}_N + \mathbf{M}_N) \approx 0$, and thus, $\operatorname{diag}\mathbf{A}_N \approx \operatorname{diag}\mathbf{U}_N$. The nondiagonal elements of $\mathbf{A}_N$ canceled out asymptotically. We conclude that $\mathbf{A}_N \approx \operatorname{diag}\mathbf{U}_N$, which is the desired result.

It is noteworthy that there are many choices of $\tilde{\mathbf{X}}_N \approx \mathbf{X}_N$. It is obvious that the optimal choice of $\tilde{\mathbf{X}}_N$ in this case is the exact matrix of $\mathbf{R}_N^{-1}\mathbf{D}_N$. Unfortunately, this matrix is hard to be computed and it may not have a uniform result for different N. Since we are dealing with the problem (5.28) in infinite sense, any choice of the matrix $\tilde{\mathbf{X}}_N \approx \mathbf{X}_N$ provides the same solution according to *Definition 1*. We choose the simple form (5.29) as solving for the eigenvalues of $\tilde{\mathbf{X}}_N$ is easy. The results are obtained that the $(N-2)$ "central" eigenvalues of $\tilde{\mathbf{X}}$ are equal to 1 and the two remaining eigenvalues are both equal to $1-\rho^2$. Therefore, We have the $(N-2)$ eigenvalues of $\mathbf{V}_N$ converge to one and the remaining two of them converge to $1/(1-\rho^2)$. We thus conclude that the eigenvalue spread of the autocorrelation matrix $\mathbf{V}_N$ with DST-LMS is such that

$$\lim_{N\to\infty}\begin{pmatrix}\text{Eigenvalue spread after DST}\\ \text{power normalization}\end{pmatrix} = \frac{1}{1-\rho^2}. \tag{5.37}$$

To give an idea of the orders of magnitude, Figure 5.3 shows the eigenvalue spread of $\mathbf{V}_N$ in DST-LMS for increasing N and for different values of ρ. It is seen that the values are less than 6 for $|\rho| \leq 0.9$ as N increases to relatively large. For not very large N, the asymptotic level can then be reached for ρ not very close to unity, this is due to the fact that the asymptotic equivalences used in the derivation are related to the decay of ρ^N. It can also be observed that the eigenvalue spreads are symmetric for positive and negative ρ.

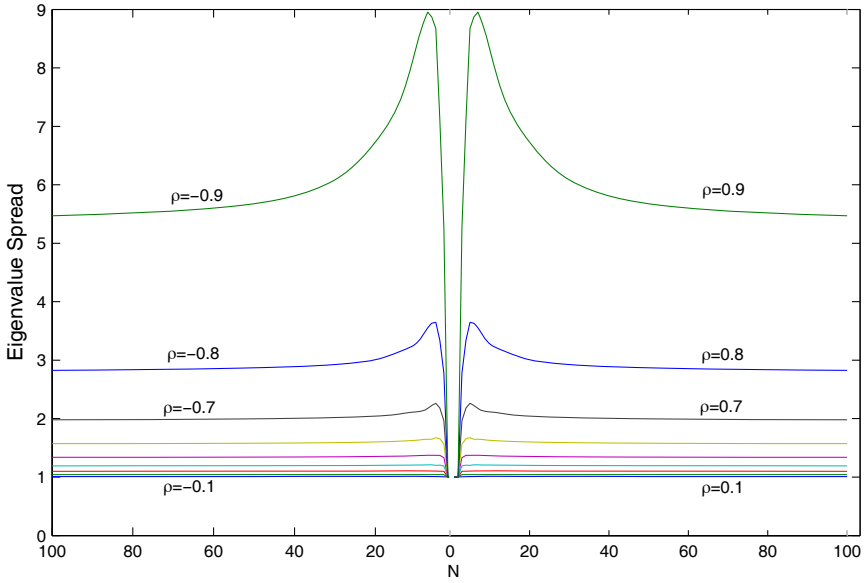

Figure 5.3: Eigenvalue spread of $\mathbf{V}_N$ for first-order AR process in DST-LMS.

5.4.4 Transform-Domain LMS algorithms with Second-Order AR Inputs

When M is equal to two, the $N \times N$ autocorrelation matrix of $\mathbf{R}_N$ is given by

$$\mathbf{R}_N = c_1 \mathbf{R}_N^1 + c_2 \mathbf{R}_N^2, \tag{5.38}$$

with the constants

$$c_1 = \frac{\rho_1(1-\rho_2^2)}{(\rho_1-\rho_2)(1+\rho_1\rho_2)}, c_2 = -\frac{\rho_2(1-\rho_1^2)}{(\rho_1-\rho_2)(1+\rho_1\rho_2)}, \tag{5.39}$$

where ρ_1 and ρ_2 are the two roots of the characteristic equation: $1 + a_1 z^{-1} + a_2 z^{-2} = 0$. These two roots can be real valued or complex-conjugate pair, and bounded by $|\rho_1|, |\rho_2| < 1$ for stability of the AR model.

Since matrices $\mathbf{R}_N^1$ and $\mathbf{R}_N^2$ are Toeplitz, the matrix $\mathbf{R}_N$ is also Toeplitz. Using the asymptotic theory of Toeplitz matrix [91] yields the power spectrum of x_n as follows

$$\begin{aligned} P(\omega) &= \sum_{l=-\infty}^{+\infty} r(l) e^{-j\omega l} \\ &= \frac{c_1(1-\rho_1^2)}{1-2\rho_1\cos\omega+\rho_1^2} + \frac{c_2(1-\rho_2^2)}{1-2\rho_2\cos\omega+\rho_2^2}, \end{aligned} \tag{5.40}$$

where $r(l)$ denotes the autocorrelation function defined as

$$r(l) \triangleq E\left[x_n x_{n-l}\right] = \tilde{\rho}^l, \tag{5.41}$$

with $r(l) = r(-l)$ for real valued signal x_n. Considering the general case $\rho_1 > \rho_2 \geq 0$, the maximum and minimum of power spectrums are obtained with $\cos\omega$ equal to 1 and -1 respectively. The asymptotic eigenvalue spread of $\mathbf{R}_N$ is thus given by

$$\begin{aligned}\lim_{N\to\infty}\left(\frac{\lambda_{max}}{\lambda_{min}}\right) &= \frac{P_{max}(\omega)}{P_{min}(\omega)} \\ &= \frac{\rho_1/(1-\rho_1)^2 - \rho_2/(1-\rho_2)^2}{\rho_1/(1+\rho_1)^2 - \rho_2/(1+\rho_2)^2}.\end{aligned} \tag{5.42}$$

Note that the result (5.42) gives the upper bound of the eigenvalue spread for finite value of N. It can easily be checked that this eigenvalue spread can be extremely large when both ρ_1 and ρ_2 are close to 1 . Therefore, when the LMS algorithm is used to update filter coefficients with such inputs, the convergence rate can be considerably slow.

In the following part, we determine the eigenvalues of the preprocessed input autocorrelation matrix of $\mathbf{V}_N$, which is the transformed and power normalized version of $\mathbf{R}_N$ in (5.38). The similar procedures as the derivation for DST-LMS with first-order AR process are followed.

For expressing simplicity, we define some notations as follows: $\tilde{\rho}^l \triangleq c_1\rho_1^l + c_2\rho_2^l$ for $l = 0, 1, \cdots, N-1$, $c \triangleq -\rho_1\rho_2$ and $s_n \triangleq \sum_{i=0}^{n} \rho_1^i \rho_2^{n-i}$ for $n = 1, 2, \cdots, N-1$.

From the derivations of (5.28) and (5.38), the matrix $\mathbf{D}_N$ is generally expressed as

$$\mathbf{D}_N = c_1\mathbf{D}_N^1 + c_2\mathbf{D}_N^2 \tag{5.43}$$

with

$$\mathbf{D}_N^i \triangleq \mathbf{T}_N^H \,\mathrm{diag}(\mathbf{T}_N\mathbf{R}_N^i\mathbf{T}_N^H)\mathbf{T}_N, \quad i = 1, 2. \tag{5.44}$$

It can be seen that the matrices $\mathbf{D}_N^1$ and $\mathbf{D}_N^2$ have the similar expressions as (5.27) but differ in the transformation matrix. If the same transformation matrix is used, the matrices $\mathbf{D}_N^1$ and $\mathbf{D}_N^2$ can easily be obtained from the results for first-order case. In the following parts, we will derive the eigenvalue distributions of $\mathbf{V}_N$ for all the algorithms separately.

5.4.5 DFT-LMS with Second-Order AR Inputs

The autocorrelation matrix of the signal $v_n(i)$ after performing DFT transformation and power normalization to the input signal x_n has the same form as given in (5.23) with $\mathbf{U}_N = \mathbf{F}_N\mathbf{R}_N\mathbf{F}_N^H$. Similarly, we need to solve for the asymptotic eigenvalues λ of the autocorrelation matrix $\mathbf{V}_N$ from

$$\lim_{N\to\infty} \det(\mathbf{V}_N - \lambda\mathbf{I}) = 0, \tag{5.45}$$

It leads to the following generalized eigenvalue problem [92]:

$$\lim_{N\to\infty} \det(\mathbf{R}_N - \lambda\mathbf{D}_N) = 0, \tag{5.46}$$

where $\mathbf{D}_N$ is defined as

$$\mathbf{D}_N = \mathbf{F}_N^H(\operatorname{diag}\mathbf{U}_N)\mathbf{F}_N. \tag{5.47}$$

To derive the matrix $\mathbf{D}_N$, we compute the diagonal elements of $\mathbf{U}_N$ first as follows:

$$\begin{aligned}\mathbf{U}_N(m,m) &= \sum_{p,q=0}^{N-1} \mathbf{F}_N(m,p)\mathbf{R}_N(p,q)\mathbf{F}_N^H(q,m) \\ &= \frac{1}{N}\sum_{p,q=0}^{N-1} \mathbf{R}_N(p,q)e^{j(m(p-q)2\pi/N)}.\end{aligned} \tag{5.48}$$

The elements of $\mathbf{D}_N$ are then given by:

$$\begin{aligned}\mathbf{D}_N(k,l) &= \sum_{m=0}^{N-1} \mathbf{F}_N^H(k,m)\mathbf{U}_N(m,m)\mathbf{F}_N(m,l) \\ &= \frac{1}{N}\sum_{m=0}^{N-1} \mathbf{U}_N(m,m)e^{j(m(l-k)2\pi/N)}.\end{aligned} \tag{5.49}$$

For the justification of the circulant nature of matrix $\mathbf{D}_N$, it is seen that the elements $\mathbf{D}_N(k+1,l+1)$, $\mathbf{D}_N(k,0)$ and $\mathbf{D}_N(0,N-k)$ are easily obtained from (5.49). The Toeplitz and circulant matrix $\mathbf{D}_N$ are verified by the equalities $\mathbf{D}_N(k,l) = \mathbf{D}_N(k+1,l+1)$ for $k,l = 0,1,\cdots,N-2$ and $\mathbf{D}_N(k,0) = \mathbf{D}_N(0,N-k)$ according to the matrix theory [91].

As a result, only the first row of $\mathbf{D}_N$ needs to be evaluated:

$$\begin{aligned}
\mathbf{D}_N(0,l) &= \sum_{m=0}^{N-1} \mathbf{F}_N^H(0,m)\mathbf{B}_N(m,m)\mathbf{F}_N(m,l) \\
&= \frac{1}{N^2}\sum_{p,q=0}^{N-1} \tilde{\rho}^{|p-q|} \sum_{m=0}^{N-1} e^{j(m(l+p-q)2\pi/N)} \\
&= \frac{1}{N}\sum_{p,q=0}^{N-1} \tilde{\rho}^{|p-q|}\delta(l+p-q)_{\mathrm{mod}N} \\
&= \frac{1}{N}\sum_{p=0}^{N-1-l} \tilde{\rho}^{l} + \frac{1}{N}\sum_{p=N-l}^{N-1} \tilde{\rho}^{N-l} \\
&= \tilde{\rho}^l - \frac{l}{N}\tilde{\rho}^l + \frac{l}{N}\tilde{\rho}^{N-l},
\end{aligned} \tag{5.50}$$

where $l = 0, 1, \cdots, N-1$. We replace the matrix $\mathbf{D}_N$ with a simpler and Toeplitz matrix $\tilde{\mathbf{D}}_N$ with the first row defined by

$$\begin{aligned}
\tilde{\mathbf{D}}_N(0,l) &= \mathbf{R}_N(0,l) + \frac{l}{N}\mathbf{R}_N(0,N-L) \\
&= \tilde{\rho}^l + \frac{l}{N}\tilde{\rho}^{N-l}.
\end{aligned} \tag{5.51}$$

The asymptotic equivalence $\tilde{\mathbf{D}}_N \approx \mathbf{D}_N$ can be validated as follows. By Theorem 1, showing $\tilde{\mathbf{D}}_N \approx \mathbf{D}_N$ is equivalent to showing the difference matrix $\phi_N = \mathbf{D}_N - \tilde{\mathbf{D}}_N$ is rank zero *perturbation*. Since $\mathbf{D}_N$ and $\tilde{\mathbf{D}}_N$ are Toeplitz, ϕ_N is also Toeplitz with the first row given by

$$\phi_N(0,l) = \mathbf{D}_N(0,l) - \tilde{\mathbf{D}}_N(0,l) = -\frac{l}{N}\tilde{\rho}^l. \tag{5.52}$$

Using the asymptotic theory of Toeplitz matrix [91], the power spectrum of the signal having the autocorrelation matrix ϕ_N (5.52) as N approaches to infinity is equal to

$$\begin{aligned}
P(\omega) &= \lim_{N\to\infty} \sum_{l=0}^{N-1} -\frac{l}{N}\tilde{\rho}^l(e^{-j\omega l} + e^{j\omega l}) \\
&= \lim_{N\to\infty}\left[\mathcal{R}\left(\frac{-2c_1\rho_1^N e^{j\omega N}}{\rho_1 e^{j\omega}-1}\right) + \mathcal{R}\left(\frac{2c_2\rho_2^N e^{j\omega N}}{\rho_2 e^{j\omega}-1}\right)\right] \\
&\quad -2c_1 \lim_{N\to\infty}\mathcal{R}\left(\frac{\rho_1 e^{j\omega}(\rho_1^N e^{j\omega N}-1)}{N(\rho_1 e^{j\omega}-1)^2}\right) \\
&\quad +2c_2 \lim_{N\to\infty}\mathcal{R}\left(\frac{\rho_2 e^{j\omega}(\rho_2^N e^{j\omega N}-1)}{N(\rho_2 e^{j\omega}-1)^2}\right) \\
&= 0.
\end{aligned} \tag{5.53}$$

Since the power spectrum $P(\omega)$ is equal to zero for all frequencies $\omega \in [0, 2\pi]$, ϕ_N is a rank zero *perturbation*. By virtue of Theorem 1, $\mathbf{D}_N \approx \tilde{\mathbf{D}}_N$.

It is seen that the asymptotical equivalence of $\mathbf{D}_N$ and $\tilde{\mathbf{D}}_N$ is verified based on the fact that the difference terms $c_1 l/N\rho_1^l$ and $c_2 l/N\rho_2^l$ converge to zero when N goes to infinity independent of l. It turns out that solving (5.46) for λ is equivalent to solving the following equation

$$\lim_{N\to\infty} \det(\mathbf{R}_N - \lambda\tilde{\mathbf{D}}_N) = 0. \tag{5.54}$$

Since $\tilde{\mathbf{D}}_N$ is not singular, multiplying both sides of (5.54) by $\det(\tilde{\mathbf{D}}_N^{-1})$ and using matrix theory, we have

$$\lim_{N\to\infty} \det(\tilde{\mathbf{D}}_N^{-1}\mathbf{R}_N - \lambda\mathbf{I}) = 0. \tag{5.55}$$

We noted that solving the inverse of problem (5.55) provides algebraic simplicity. Therefore, we have

$$\lim_{N\to\infty} \det(\mathbf{R}_N^{-1}\tilde{\mathbf{D}}_N - \lambda^{-1}\mathbf{I}) = 0. \tag{5.56}$$

Next, we compute the simple matrix of $\tilde{\mathbf{X}}_N$ such that $\tilde{\mathbf{X}}_N \approx \mathbf{X}_N = \mathbf{R}_N^{-1}\tilde{\mathbf{D}}_N$. To have an idea of how the elements of the matrix $\mathbf{X}_N$ are distributed, we plotted in Figure 5.4 the matrix of $\mathbf{X}_N$ for the choice of $a_1 = -1.1, a_2 = 0.3$ and N equal to 50. It can be seen that most elements of $\mathbf{X}_{50}$ are close to zero except the elements along the main diagonal and side rows. In fact, these small elements are going to converge to zero as N approaches to infinity. From this observation and computations of $\mathbf{R}_N^{-1}\tilde{\mathbf{D}}_N$ for small N, it turns out that $\mathbf{X}_N = \mathbf{R}_N^{-1}\tilde{\mathbf{D}}_N$ can be replaced by $\tilde{\mathbf{X}}_N \approx \mathbf{X}_N$ which has a simple form:

$$\tilde{\mathbf{X}}_N = \begin{pmatrix} 1 & s_{N-1} & s_{N-2} & \cdots & s_3 & s_2 & s_1 \\ 0 & 1 & cs_{N-3} & cs_{N-4} & \cdots & cs_1 & c \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & & \ddots & & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 & 0 & 0 \\ c & cs_1 & cs_2 & \cdots & cs_{N-3} & 1 & 0 \\ s_1 & s_2 & s_3 & \cdots & s_{N-2} & s_{N-1} & 1 \end{pmatrix}. \tag{5.57}$$

The proof for $\tilde{\mathbf{X}}_N \approx \mathbf{X}_N \triangleq \mathbf{R}_N^{-1}\tilde{\mathbf{D}}_N$ is given as follows. From the matrix theory and Theorem 1, showing that $\tilde{\mathbf{X}}_N \approx \mathbf{R}_N^{-1}\tilde{\mathbf{D}}_N$ is equivalent to showing that $\mathbf{R}_N\tilde{\mathbf{X}}_N \approx \tilde{\mathbf{D}}_N$ or to showing that $\phi_N = \mathbf{R}_N\tilde{\mathbf{X}}_N - \tilde{\mathbf{D}}_N$ is asymptotically a rank zero matrix. In the following

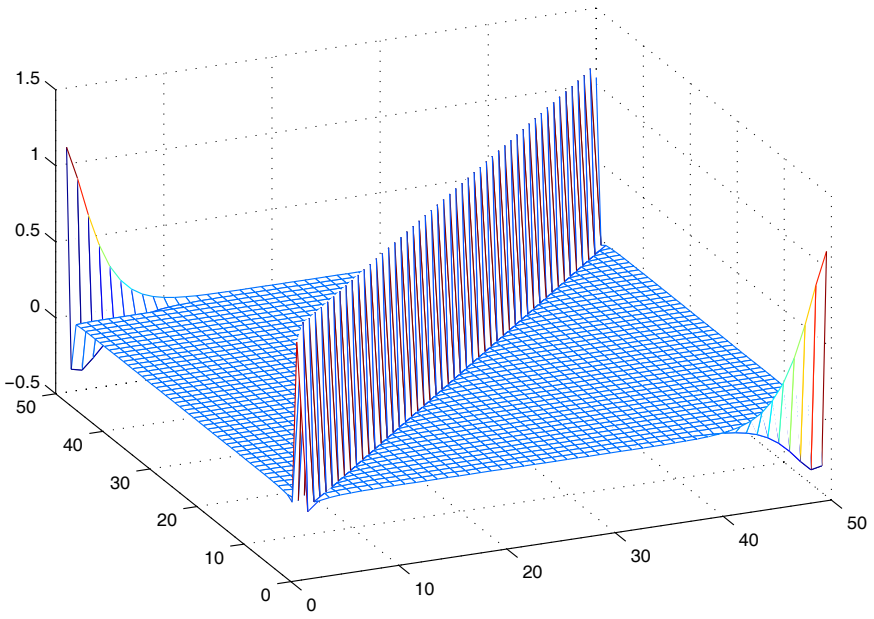

Figure 5.4: 3-D plot of the matrix $\mathbf{X}_N = \mathbf{R}_N^{-1}\mathbf{D}_N$ in DFT-LMS with AR process of $a_1 = -1.1, a_2 = 0.3$.

section, we evaluate the elements of $\tilde{\mathbf{D}}_N$ and $\mathbf{R}_N\tilde{\mathbf{X}}_N$ first, and then show their difference ϕ_N is rank zero *perturbation*.

Using the circulant nature of $\tilde{\mathbf{D}}_N$, the elements of $\tilde{\mathbf{D}}_N$ can be expressed as

$$\tilde{\mathbf{D}}_N(k,l) = \begin{cases} \tilde{\rho}^{l-k} + \frac{l-k}{N}\tilde{\rho}^{N+k-l} & k \le l \\ \tilde{\rho}^{N+l-k} + \frac{N+l-k}{N}\tilde{\rho}^{k-l} & k > l. \end{cases} \tag{5.58}$$

The elements of $\mathbf{R}_N\tilde{\mathbf{X}}_N$ are derived as follows.

$\mathbf{R}_N\tilde{\mathbf{X}}_N(k,l) = \sum_{m=0}^{N-1} \mathbf{R}_N(k,m)\tilde{\mathbf{X}}_N(m,l)$

case $l = 0$:

$$\begin{aligned} &= \sum_{m=0}^{N-1} \tilde{\rho}^{|k-m|}\tilde{\mathbf{X}}_N(m,0) \\ &= \tilde{\rho}^k + c\tilde{\rho}^{|N-2-k|} + s_1\tilde{\rho}^{N-1-k}, \end{aligned} \tag{5.59}$$

case $l = 1$:

$$\begin{aligned}
&= \sum_{m=0}^{N-1} \tilde{\rho}^{|k-m|} \tilde{\mathbf{X}}_N(m,1) \\
&= s_{N-1}\tilde{\rho}^k + \tilde{\rho}^{|k-1|} + c s_1 \tilde{\rho}^{|N-2-k|} \\
&\quad + (s_1^2 + c)\tilde{\rho}^{N-1-k} \\
&= s_{N-1}\tilde{\rho}^k + \tilde{\rho}^{|k-1|} + \tilde{\rho}^{N+1-k},
\end{aligned} \tag{5.60}$$

case $2 \leq l \leq N-3$:

$$\begin{aligned}
&= \sum_{m=0}^{N-1} \tilde{\rho}^{|k-m|} \tilde{\mathbf{X}}_N(m,l) \\
&= \tilde{\rho}^k s_{N-l} + c\tilde{\rho}^{|k-1|} s_{N-l-1} + \tilde{\rho}^{|k-l|} \\
&\quad + c\tilde{\rho}^{|N-2-k|} s_l + \tilde{\rho}^{N-1-k} s_{l+1} \\
&= \tilde{\rho}^{N+k-l} + \tilde{\rho}^{|k-l|} + \tilde{\rho}^{N-k+l},
\end{aligned} \tag{5.61}$$

case $l = N-2$:

$$\begin{aligned}
&= \sum_{m=0}^{N-1} \tilde{\rho}^{|k-m|} \tilde{\mathbf{X}}_N(m,N-2) \\
&= (s_1^2 + c)\tilde{\rho}^k + c s_1 \tilde{\rho}^{|k-1|} + \tilde{\rho}^{|N-2-k|} \\
&\quad + s_{N-1}\tilde{\rho}^{N-k-1} \\
&= \tilde{\rho}^{k+2} + \tilde{\rho}^{|N-2-k|} + \tilde{\rho}^{N-k-1} s_{N-1},
\end{aligned} \tag{5.62}$$

case $l = N-1$:

$$\begin{aligned}
&= \sum_{m=0}^{N-1} \tilde{\rho}^{|k-m|} \tilde{\mathbf{X}}_N(m,N-1) \\
&= s_1\tilde{\rho}^k + c\tilde{\rho}^{|k-1|} + \tilde{\rho}^{N-k-1} \\
&= \tilde{\rho}^{k+1} + \tilde{\rho}^{N-k-1}.
\end{aligned} \tag{5.63}$$

Summarizing all the cases, the elements of $\mathbf{R}_N \tilde{\mathbf{X}}_N$ are given by

$$\mathbf{R}_N \tilde{\mathbf{X}}_N(k,l) = \begin{cases}
\tilde{\rho}^k + c\tilde{\rho}^{|N-2-k|} + s_1\tilde{\rho}^{N-1-k} & l = 0 \\
s_{N-1}\tilde{\rho}^k + \tilde{\rho}^{|k-1|} + \tilde{\rho}^{N+1-k} & l = 1 \\
\tilde{\rho}^{N+k-l} + \tilde{\rho}^{|k-l|} + \tilde{\rho}^{N-k+l} & 2 \leq l \leq N-3 \\
\tilde{\rho}^{k+2} + \tilde{\rho}^{|N-2-k|} + \tilde{\rho}^{N-k-1} s_{N-1} & l = N-2 \\
\tilde{\rho}^{k+1} + \tilde{\rho}^{N-k-1} & l = N-1.
\end{cases} \tag{5.64}$$

Subtracting (5.64) from (5.58), we obtain the matrix of ϕ_N as

$$\phi_N(k,l) = \begin{cases} \tilde{\rho}^N & l=0, k=0 \\ \frac{k}{N}\tilde{\rho}^k & l=0, k>0 \\ \tilde{\rho}^k s_{N-1} + \tilde{\rho}^{N+1-k} - \frac{1-k}{N}\tilde{\rho}^{N-1+k} & l=1, k\leq 1 \\ \tilde{\rho}^k s_{N-1} + \frac{k-1}{N}\tilde{\rho}^{k-1} & l=1, k>1 \\ \frac{N+k-l}{N}\tilde{\rho}^{N+k-l} + \tilde{\rho}^{N-k+l} & 2\leq l\leq N-3, k\leq l \\ \frac{k-l}{N}\tilde{\rho}^{k-l} + \tilde{\rho}^{N+k-l} & 2\leq l\leq N-3, k>l \\ \tilde{\rho}^{N-k-1}s_{N-1} + \frac{k+2}{N}\tilde{\rho}^{k+2} & l=N-2, k\leq N-2 \\ s_{N-1} + \tilde{\rho}^{N+1} - \tilde{\rho}^{N-1} + \frac{1}{N}\tilde{\rho} & l=N-2, k=N-1 \\ \frac{k+1}{N}\tilde{\rho}^{k+1} & l=N-1. \end{cases} \tag{5.65}$$

It is easy to check that all the terms s_{N-1}, $\frac{l}{N}\tilde{\rho}^l$, $\tilde{\rho}^{N-1}$ and $\tilde{\rho}^{N+1}$ converge to zero for very large N and $l \geq 0$ due to the fact $|\rho_1| < 1$ and $|\rho_2| < 1$. Therefore, all the elements of ϕ_N converge to zero. We conclude that the matrix ϕ_N is asymptotically rank zero *perturbation*. By virtue of Theorem 1, we have $\tilde{\mathbf{X}}_N \approx \mathbf{R}_N^{-1}\tilde{\mathbf{D}}_N$.

Compared to the matrix $\tilde{\mathbf{X}}_N$ provided in [88] for DFT-LMS with Markov-1 signals, the matrix (5.57) is more complicated with two more non-diagonal rows. However, it is still very simple for solving for its eigenvalues since the $(N-4)\times(N-4)$ central submatrix is identity and the elements of the first two rows and the last two rows are just inverted. To solve for the eigenvalues λ, we rewrite (5.56) as

$$\lim_{N\to\infty} \det(\tilde{\mathbf{X}}_N - \tilde{\lambda}\mathbf{I}) = 0. \tag{5.66}$$

From the structure of matrix $\tilde{\mathbf{X}}_N$ in (5.57), we have that the $N-4$ eigenvalues of $\tilde{\mathbf{X}}_N$ are given by the eigenvalues of the $(N-4)\times(N-4)$ central submatrix of $\tilde{\mathbf{X}}_N$, which are equal to 1. The other four eigenvalues of $\tilde{\mathbf{X}}_N$ computed from (5.66) are given by

$$\begin{aligned} \tilde{\lambda}_{1,2} &= 1 + 0.5s_1(1+c) \pm \tau, \\ \tilde{\lambda}_{3,4} &= 1 - 0.5s_1|(1+c) \pm \tau, \end{aligned} \tag{5.67}$$

with

$$\tau = 0.5\sqrt{s_1^2(1+c^2) + 2c(\rho_1^2+\rho_2^2)}. \tag{5.68}$$

From the whole derivation, which is based on the strong asymptotic equivalence, we know that the eigenvalues of $\mathbf{V}_N$ asymptotically converge to the inverse eigenvalues of $\tilde{\mathbf{X}}_N$ as N tends to infinity. By virtue of Theorem 1, we have the asymptotic eigenvalues of $\mathbf{V}_N$

with four of them equal to the inverse of $\tilde{\lambda}_{1,2}$ and $\tilde{\lambda}_{3,4}$, and the rest equal to 1. Considering the case $\rho_1, \rho_2 \in [0, 1[$, the maximum and minimum eigenvalues of $\mathbf{V}_N$ are obvious. Finally, we conclude that the asymptotic eigenvalue spread of the autocorrelation matrix $\mathbf{V}_N$ obtained after DFT and power normalization is given by

$$\lim_{N\to\infty}\left(\frac{\lambda_{max}}{\lambda_{min}}\right) = \frac{1+0.5|s_1|(1+c)+\tau}{1-0.5|s_1|(1+c)-\tau}, \tag{5.69}$$

with τ given by (5.68).

Remark 1: It is easy to check for the special case of Markov-1 signals, that is, the correlation factor ρ_2 is set to zero, that the result of (5.69) becomes $(1+\rho_1)/(1-\rho_1)$, which is the same form as provided in [88].

Remark 2: From (5.69) we can easily notice that the DFT is not a perfect decorrelator. The input eigenvalue spread of DFT-LMS can still be very high to limit its uses for strongly correlated input signals, which can also be seen from the 3-D plot of matrix $\tilde{\mathbf{X}}_N$ in Figure 5.4. There are non-zero dominant elements determined by the correlation factors appearing in the off-diagonal, which introduces large eigenvalue spread when ρ_1 and ρ_2 close to 1.

5.4.6 DHT-LMS with Second-Order AR Inputs

It has been theoretically proved in [89] that DHT-LMS achieves the same asymptotic eigenvalue spread as DFT-LMS for Markov-1 input signals. We extend this result to second-order AR input process in this section. That is, we will show that second-order AR inputs, DHT-LMS also achieves the same asymptotic eigenvalue spread as DFT-LMS. Therefore, the asymptotic eigenvalue spread of DHT-LMS can actually be obtained from that of DFT-LMS.

Proceeding the same way as we did for DFT-LMS, the problem of solving for the asymptotic eigenvalue spread of DHT-LMS is led to solve the following generalized eigenvalue problem

$$\lim_{N\to\infty}\det(\mathbf{R}_N - \lambda\hat{\mathbf{D}}_N) = 0, \tag{5.70}$$

where $\hat{\mathbf{D}}_N$ is given by

$$\hat{\mathbf{D}}_N = \mathbf{H}_N^H \operatorname{diag}\mathbf{U}_N\mathbf{H}_N, \tag{5.71}$$

where $\mathbf{U}_N = \mathbf{H}_N \mathbf{R}_N \mathbf{H}_N^H$. For real matrix $\mathbf{H}_N$, the Hermitian transpose performs matrix transpose only. Since the generalized eigenvalue problems (5.46) and (5.70) have the same forms, it is expected to show that the asymptotic equivalence $\hat{\mathbf{D}}_N \approx \mathbf{D}_N$ with $\mathbf{D}_N$ given in (5.47). The proof for $\hat{\mathbf{D}}_N \approx \mathbf{D}_N$ is provided as follows.

Let us consider the DHT analysis matrix $\mathbf{H}_N$ expressed with the DFT analysis matrix $\mathbf{F}_N$ using Euler's identity as

$$\mathbf{H}_N = \frac{1}{2}(\mathbf{F}_N + \mathbf{F}_N^H + j\mathbf{F}_N^H - j\mathbf{F}_N). \tag{5.72}$$

Substituting $\mathbf{H}_N$ into (5.71) and through some manipulations as made in [49], we simplify the matrix $\hat{\mathbf{D}}_N$ to

$$\begin{aligned}\hat{\mathbf{D}}_N = &\underbrace{\mathbf{F}_N^H \operatorname{diag}(\mathbf{F}_N \mathbf{R}_N \mathbf{F}_N^H)\mathbf{F}_N}_{\mathbf{E}_1} \\ &+ \underbrace{\frac{1}{2}\mathbf{F}_N^H \operatorname{diag}(\mathbf{F}_N \mathbf{R}_N \mathbf{F}_N)\mathbf{F}_N^H}_{\mathbf{E}_2} \\ &- \underbrace{\frac{1}{2}\mathbf{F}_N \operatorname{diag}(\mathbf{F}_N \mathbf{R}_N \mathbf{F}_N^H)\mathbf{F}_N}_{\mathbf{E}_3}.\end{aligned} \tag{5.73}$$

From (5.73), we can see that the part $\mathbf{E}_1$ of $\hat{\mathbf{D}}_N$ has the same form as $\mathbf{D}_N$ (5.47) of DFT-LMS. We wish to prove the difference of $\mathbf{E}_2$ and $\mathbf{E}_3$ is asymptotically rank zero *perturbation*. Let us evaluate the parts $\mathbf{E}_2$ and $\mathbf{E}_3$ separately.

It is easy to check that the matrix $\mathbf{E}_2$ is a bisymmetric Hankel matrix [93]. Therefore, we only need to compute its first row:

$$\begin{aligned}\mathbf{E}_2(0,l) &= \frac{1}{2}\sum_{m,p,q=0}^{N-1} \mathbf{F}_N^H(0,m)\mathbf{F}_N(m,p)\mathbf{R}_N(p,q)\mathbf{F}_N(q,m)\mathbf{F}_N^H(m,l) \\ &= \frac{1}{2N^2}\sum_{p,q=0}^{N-1} \tilde{\rho}^{|p-q|} \sum_{m=0}^{N-1} e^{\frac{2\pi m(p+q-l)}{N}} \\ &= \frac{1}{2N}\sum_{p,q=0}^{N-1} \tilde{\rho}^{|p-q|}\delta(p+q-l)_{\mathrm{mod}N} \\ &= \frac{1}{2N}\sum_{p=0}^{l} \tilde{\rho}^{|l-2p|} + \frac{1}{2N}\sum_{p=l+1}^{N-1} \tilde{\rho}^{|l+N-2p|}.\end{aligned} \tag{5.74}$$

Similarly, we find that $\mathbf{E}_3$ is a symmetric Hankel matrix. Both its first row and last column are to be evaluated with the same way as we did for $\mathbf{E}_2(0,l)$, which are given by

$$\mathbf{E}_3(0,l) = \frac{1}{2N}\sum_{p=0}^{N-l}\tilde{\rho}^{|N-l-2p|} + \frac{1}{2N}\sum_{p=N-l+1}^{N-1}\tilde{\rho}^{|2N-l-2p|}, \tag{5.75}$$

$$\mathbf{E}_3(k,N-1) = \frac{1}{2N}\sum_{p=0}^{N-k+1}\tilde{\rho}^{|N+1-k-2p|} + \frac{1}{2N}\sum_{p=N-k+2}^{N-1}\tilde{\rho}^{|2N+1-k-2p|}. \tag{5.76}$$

Let $\bar{\mathbf{E}} \triangleq \mathbf{E}_2 - \mathbf{E}_3$. Following the symmetric Hankel properties of $\mathbf{E}_2$ and $\mathbf{E}_3$, $\bar{\mathbf{E}}$ is also be symmetric Hankel, which means to evaluate the elements of $\bar{\mathbf{E}}$, we only need to compute its first row and last column. After some algebra, we obtain the results as follows [49]:

$$\bar{\mathbf{E}}(0,l) = \frac{1}{N}\tilde{\rho}^{l} - \frac{1}{N}\tilde{\rho}^{N-l}, \tag{5.77}$$

$$\bar{\mathbf{E}}(k,N-1) = \begin{cases} \frac{1}{N}\tilde{\rho}^{N-1} - \frac{1}{N}\tilde{\rho} & k=0 \\ \frac{1}{N}\tilde{\rho}^{k-1} - \frac{1}{N}\tilde{\rho}^{N-k+1} & k \neq 0. \end{cases} \tag{5.78}$$

Next, we would like to show the power spectrum of the signals with the autocorrelation matrix $\bar{\mathbf{E}}$ is zero as N goes to infinity. Proceeding the same way as made in [49] to multiply $\bar{\mathbf{E}}$ by the counter-identity matrix[2] $\mathbf{J}$ results in the Toeplitz matrix $\bar{\mathbf{E}}^J \triangleq \bar{\mathbf{E}}\mathbf{J}$. Since this procedure does not change the eigenvalues of $\bar{\mathbf{E}}$, we can apply the asymptotic theory of Toeplitz matrix [91] on $\bar{\mathbf{E}}^J$to compute its power spectrum as

$$\begin{aligned} P_{\bar{\mathbf{E}}^J}(\omega) &= \sum_{l=-(N-1)}^{0}\bar{\mathbf{E}}(0,N-1+l)e^{-j\omega l} + \sum_{k=1}^{N-1}\bar{\mathbf{E}}(k,N-1)e^{-j\omega k} \\ &= \frac{1}{N}\sum_{l=-(N-1)}^{0}\left(\tilde{\rho}^{N-1+l} - \tilde{\rho}^{1-l}\right)e^{-j\omega l} + \frac{1}{N}\sum_{k=1}^{N-1}\left(\tilde{\rho}^{k-1} - \tilde{\rho}^{N-k+1}\right)e^{-j\omega k} \\ &= \frac{c_1}{N}\left(\frac{e^{j\omega(N-1)}+e^{-j\omega}}{1-\rho_1 e^{-j\omega}} + \frac{e^{-j\omega}+\rho_1 e^{-j\omega N}}{1-\rho_1^{-1}e^{-j\omega}}\right) \\ &\quad -\frac{c_2}{N}\left(\frac{e^{j\omega(N-1)}+e^{-j\omega}}{1-\rho_2 e^{-j\omega}} + \frac{e^{-j\omega}+\rho_2 e^{-j\omega N}}{1-\rho_2^{-1}e^{-j\omega}}\right) + \mathcal{O}(\rho_1^N) - \mathcal{O}(\rho_2^N). \end{aligned} \tag{5.79}$$

[2]The counter-identity matrix is defined as the matrix with ones across its cross diagonal and zeros elsewhere.

From (5.79), we see that $\lim_{N\to\infty} P_{\bar{\mathbf{E}}^J}(\omega) = 0$ for all frequencies $\omega \in [0, 2\pi]$. We conclude that $\bar{\mathbf{E}}^J$ is rank zero *perturbation*, which implies that $\bar{\mathbf{E}}$ is rank zero *perturbation*. Therefore, we have the asymptotic equivalence that $\hat{\mathbf{D}}_N = \mathbf{E}_1 + \mathbf{E}_2 - \mathbf{E}_3 \approx \mathbf{E}_1 = \mathbf{D}_N$, which is the desired result.

It is implied that DHT-LMS and DFT-LMS actually achieve the same asymptotic eigenvalue spreads of the preprocessed input autocorrelation matrix. Therefore, the eigenvalues and eigenvalue spread of the preprocessed second-order AR input autocorrelation matrix of DHT-LMS are obtained directly from those of DFT-LMS as N tends to infinity. That is, four of them converge to the inverse of $\tilde{\lambda}_{1,2}$ and $\tilde{\lambda}_{3,4}$, and the rest $N-4$ eigenvalues converge to 1. The asymptotic eigenvalue spread of DHT-LMS is the same expression as given in (5.69) for DFT-LMS.

Remark 3: Compared to the complex DFT-LMS, DHT-LMS processes with real values. Therefore, it is computationally efficient to process real input signals with DHT-LMS. However, DHT-LMS still has weak orthogonalization capability, which is the same as DFT-LMS for highly correlated second-order AR input signals as seen from (5.69).

5.4.7 DCT-LMS with Second-Order AR Inputs

To find the eigenvalue distribution of the autocorrelation matrix after DCT and power normalization for DCT-LMS, again we solve the similar problem as DFT-LMS with matrix $\mathbf{V}_N$ given by

$$\mathbf{V}_N = (\operatorname{diag} \mathbf{U}_N)^{-1/2} \mathbf{U}_N (\operatorname{diag} \mathbf{U}_N)^{-1/2}, \tag{5.80}$$

where $\mathbf{U}_N = \mathbf{C}_N \mathbf{R}_N \mathbf{C}_N^H$. Following the derivation for DFT-LMS, we are considering the inverse problem shown below

$$\lim_{N\to\infty} \det(\mathbf{R}_N^{-1}\mathbf{D}_N - \lambda^{-1}\mathbf{I}) = 0, \tag{5.81}$$

with $\mathbf{D}_N = \mathbf{C}_N^H \mathbf{R}_N \mathbf{C}_N$. For the case of DCT-LMS, the matrix $\mathbf{D}_N$ is known hard to be computed. Fortunately, the simple matrix $\tilde{\mathbf{X}}_N \approx \mathbf{R}_N^{-1}\mathbf{D}_N$ for the first-order Markov input can easily be "guessed" as shown in [88], then the matrix $\mathbf{Y}_N = \mathbf{R}_N \tilde{\mathbf{X}}_N$ asymptotic equivalent to $\mathbf{D}_N$ can be obtained. For the second-order case, we have that $\mathbf{Y}_N^i =$

$\mathbf{R}_N^i \tilde{\mathbf{X}}_N \approx \mathbf{D}_N$ for $i = 1, 2$. This implies that $\mathbf{D}_N \approx \tilde{\mathbf{D}}_N = c_1 \mathbf{Y}_N^1 + c_2 \mathbf{Y}_N^2$. Applying the result of $\mathbf{Y}_N$ as derived in the Appendix B of [88], we therefore obtain that

$$\tilde{\mathbf{D}}_N(k,l) = \begin{cases} \tilde{\rho}^k + \tilde{\rho}^{k+1} & l = 0 \\ \tilde{\rho}^{k+l+1} + \tilde{\rho}^{2N-k-l-1} + \tilde{\rho}^{|k-l|} & 1 \le l \le N-2 \\ \tilde{\rho}^{N-k-1} + \tilde{\rho}^{N-k} & l = N-1. \end{cases} \tag{5.82}$$

It turns out that we have another simple form $\tilde{\mathbf{X}}_N$, which is asymptotically equivalent to $\mathbf{R}_N^{-1}\mathbf{D}_N$ in strong sense, which is defined as

$$\tilde{\mathbf{X}}_N = \begin{pmatrix} 1+s_1 & s_2 & s_3 & \cdots & s_{N-2} & s_{N-1} & 0 \\ c & 1+cs_1 & cs_2 & \cdots & cs_{N-3} & cs_{N-2} & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & & \ddots & & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 & 0 & 0 \\ 0 & cs_{N-2} & cs_{N-3} & \cdots & cs_2 & 1+cs_1 & c \\ 0 & s_{N-1} & s_{N-2} & \cdots & s_3 & s_2 & 1+s_1 \end{pmatrix}. \tag{5.83}$$

The proof for $\tilde{\mathbf{X}}_N \approx \mathbf{X}_N \triangleq \mathbf{R}_N^{-1}\mathbf{D}_N$ is provided as follows.

It is known that showing $\tilde{\mathbf{X}}_N \approx \mathbf{R}_N^{-1}\mathbf{D}_N$ is equivalent to showing that $\mathbf{R}_N\tilde{\mathbf{X}}_N \approx \mathbf{D}_N$ or $\mathbf{C}_N\mathbf{R}_N\tilde{\mathbf{X}}_N\mathbf{C}_N^H \approx \operatorname{diag}\mathbf{U}_N$. Since the DCT transform matrix $\mathbf{C}_N$ is not with the exponential property of $\mathbf{F}_N$, $\mathbf{D}_N$ is not Teoplitz and circulant anymore. So solving for $\mathbf{D}_N$ becomes a difficult task. For the mathematic simplicity, we will show the asymptotic equivalence $\mathbf{C}_N\mathbf{R}_N\tilde{\mathbf{X}}_N\mathbf{C}_N^H \approx \operatorname{diag}\mathbf{U}_N$.

The elements of $\mathbf{R}_N\tilde{\mathbf{X}}_N$ are computed as

$$\mathbf{R}_N\tilde{\mathbf{X}}_N(k,l) = \sum_{i=0}^{N-1} \mathbf{R}_N(k,i)\tilde{\mathbf{X}}_N(i,l) =$$

$$\begin{cases} \mathbf{R}_N(k,0)\tilde{\mathbf{X}}_N(0,0) + \mathbf{R}_N(k,1)\tilde{\mathbf{X}}_N(1,0) & l = 0 \\ \mathbf{R}_N(k,0)\tilde{\mathbf{X}}_N(0,l) + \mathbf{R}_N(k,1)\tilde{\mathbf{X}}_N(1,l) & \\ \quad +\mathbf{R}_N(k,N-2)\tilde{\mathbf{X}}_N(N-2,l) & \\ \quad +\mathbf{R}_N(k,N-1)\tilde{\mathbf{X}}_N(N-1,l) & 1 \le l \le N-2 \\ \mathbf{R}_N(k,N-2)\tilde{\mathbf{X}}_N(N-2,N-1) & \\ \quad +\mathbf{R}_N(k,N-1)\tilde{\mathbf{X}}_N(N-1,N-1) & l = N-1. \end{cases} \tag{5.84}$$

Introducing $\mathbf{R}_N$ and $\tilde{\mathbf{X}}_N$ (5.83) into (5.84), we have

$$\mathbf{R}_N\tilde{\mathbf{X}}_N(k,l) = \begin{cases} \tilde{\rho}^k + \tilde{\rho}^{k+1} & l = 0 \\ \tilde{\rho}^{k+l+1} + \tilde{\rho}^{2N-k-l-1} + \tilde{\rho}^{|k-l|} & 1 \le l \le N-2 \\ \tilde{\rho}^{N-k-1} + \tilde{\rho}^{N-k} & l = N-1. \end{cases} \tag{5.85}$$

For simplicity, we define another simple matrix $\mathbf{Z}_N$ as

$$\mathbf{Z}_N(k,l) = \tilde{\rho}^{k+l+1} + \tilde{\rho}^{2N-k-l-1} + \tilde{\rho}^{|k-l|}. \tag{5.86}$$

It is easy to check that $\mathbf{Z}_N$ and $\mathbf{R}_N\tilde{\mathbf{X}}_N$ are asymptotically equivalent in strong sense since they differ only by $\mathcal{O}(\tilde{\rho}^N)$.

Now it is sufficient to show $\mathbf{C}_N\mathbf{Z}_N\mathbf{C}_N^H \approx \operatorname{diag}\mathbf{U}_N$. We decompose $\mathbf{Z}_N$ as

$$\mathbf{Z}_N = \mathbf{Y}_N^1 - \mathbf{Y}_N^2, \tag{5.87}$$

with the matrices given as $\mathbf{Y}_N^1 = c_1(\rho_1\mathbf{G}_N^{\rho_1} + \rho_1^{2N-1}\mathbf{G}_N^{1/\rho_1} + \mathbf{R}_N^1)$ and $\mathbf{Y}_N^2 = c_2(\rho_2\mathbf{G}_N^{\rho_2} + \rho_2^{2N-1}\mathbf{G}_N^{1/\rho_2} + \mathbf{R}_N^2)$, where the matrices $\mathbf{G}_N^{\rho_1}$ and $\mathbf{G}_N^{\rho_2}$ are defined as $\mathbf{G}_N^{\rho_1}(k,l) \triangleq \rho_1^{k+l}$, and $\mathbf{G}_N^{\rho_2}(k,l) \triangleq \rho_2^{k+l}$. The matrices $\mathbf{G}_N^{1/\rho_1}$ and $\mathbf{G}_N^{1/\rho_2}$ are obtained by replacing ρ_1 with $1/\rho_1$ in $\mathbf{G}_N^{\rho_1}$, and ρ_2 with $1/\rho_2$ in $\mathbf{G}_N^{\rho_2}$, respectively. Let the matrix $\mathbf{A}_N^1$ and $\mathbf{A}_N^2$ be defined as $\mathbf{A}_N^1 \triangleq \mathbf{C}_N\mathbf{Y}_N^1\mathbf{C}_N^H$ and $\mathbf{A}_N^2 \triangleq \mathbf{C}_N\mathbf{Y}_N^2\mathbf{C}_N^H$. Then $\mathbf{A}_N^1$ and $\mathbf{A}_N^2$ can be written as the sum of three matrices ($\mathbf{L}_N^1$, $\mathbf{M}_N^1$ and $\mathbf{U}_N^1$) and ($\mathbf{L}_N^2$, $\mathbf{M}_N^2$ and $\mathbf{U}_N^2$), respectively, where

$$\mathbf{L}_N^1 = c_1\rho_1\mathbf{C}_N\mathbf{G}_N^{\rho_1}\mathbf{C}_N^H \tag{5.88}$$

$$\mathbf{M}_N^1 = c_1\rho_1^{2N-1}\mathbf{C}_N\mathbf{G}_N^{1/\rho_1}\mathbf{C}_N^H \tag{5.89}$$

$$\mathbf{U}_N^1 = c_1\mathbf{C}_N\mathbf{R}_N^1\mathbf{C}_N^H \tag{5.90}$$

and

$$\mathbf{L}_N^2 = c_2\rho_2\mathbf{C}_N\mathbf{G}_N^{\rho_2}\mathbf{C}_N^H \tag{5.91}$$

$$\mathbf{M}_N^2 = c_2\rho_2^{2N-1}\mathbf{C}_N\mathbf{G}_N^{1/\rho_2}\mathbf{C}_N^H \tag{5.92}$$

$$\mathbf{U}_N^2 = c_2\mathbf{C}_N\mathbf{R}_N^2\mathbf{C}_N^H. \tag{5.93}$$

We wish to prove that $\mathbf{A}_N^1 - \mathbf{A}_N^2 \approx \operatorname{diag}\mathbf{U}_N$, which is equivalent to proving equivalences $\mathbf{A}_N^1 \approx \operatorname{diag}\mathbf{U}_N^1$ and $\mathbf{A}_N^2 \approx \operatorname{diag}\mathbf{U}_N^2$ with the fact that $\operatorname{diag}\mathbf{U}_N = \operatorname{diag}\mathbf{U}_N^1 - \operatorname{diag}\mathbf{U}_N^2$. Here we have formulated two similar problems which can be solved with the analytical method provided in [88]. That is, for the asymptotic equivalence $\mathbf{A}_N^1 \approx \operatorname{diag}\mathbf{U}_N^1$, the analytical expressions for the elements of each of the three matrices $\mathbf{L}_N^1$, $\mathbf{M}_N^1$ and $\mathbf{U}_N^1$ are derived to show that $\operatorname{diag}\mathbf{A}_N^1 \approx \operatorname{diag}\mathbf{U}_N^1$ first, and then to show that the nondiagonal terms of $(\mathbf{A}_N^1 - \operatorname{diag}\mathbf{U}_N^1)$ are $\mathcal{O}(\rho_1^N)$. These two results warrant the asymptotic equivalence $\mathbf{A}_N^1 \approx \operatorname{diag}\mathbf{U}_N^1$.

Let us first evaluate the elements of the matrix $\mathbf{L}_N^1$.

$$\begin{aligned}\mathbf{L}_N^1(k,l) &= c_1\rho_1 \sum_{m,n=0}^{N-1} \mathbf{C}_N(k,m)\mathbf{G}_N^{\rho_1}(m,n)\mathbf{C}_N(l,n) \\ &= c_1\rho_1\kappa_k\kappa_l \frac{2}{N} \sum_{m,n=0}^{N-1} \cos\frac{k(m+\frac{1}{2})\pi}{N} \cos\frac{l(n+\frac{1}{2})\pi}{N} \rho_1^{m+n} \\ &= c_1\rho_1\kappa_k\kappa_l \frac{2}{N} \left(\sum_{m=0}^{N-1} \cos\frac{k(m+\frac{1}{2})\pi}{N}\rho_1^m\right)\left(\sum_{n=0}^{N-1} \cos\frac{l(n+\frac{1}{2})\pi}{N}\rho_1^n\right). \end{aligned} \tag{5.94}$$

Following the deriving procedures presented in [88] to break the cosines into exponentials, the problem reduces to evaluate the auxiliary quantity

$$\alpha^L(k) \triangleq \sum_{m=0}^{N-1} e^{jkm\pi/N}\rho_1^m = \frac{1}{1-\rho_1 e^{jk\pi/N}} + \frac{-\rho^N e^{jk\pi}}{1-\rho_1 e^{jk\pi/N}}. \tag{5.95}$$

The expression for $\mathbf{L}_N^1(k,l)$ is then obtained as

$$\mathbf{L}_N^1(k,l) = \frac{\frac{2}{N}c_1\rho_1(1-\rho_1^2)\kappa_k\kappa_l \cos\left(\frac{k\pi}{2N}\right)\cos\left(\frac{l\pi}{2N}\right)}{\left(1-2\rho_1\cos\left(\frac{k\pi}{N}\right)+\rho_1^2\right)\left(1-2\rho_1\cos\left(\frac{l\pi}{N}\right)+\rho_1^2\right)} + \mathcal{O}(\rho_1^N). \tag{5.96}$$

Similarly, we evaluate the matrix $\mathbf{M}_N^1$.

$$\begin{aligned}\mathbf{M}_N^1(k,l) &= c_1\rho_1^{2N-1} \sum_{p,q=0}^{N-1} \mathbf{C}_N(k,p)\mathbf{G}_N^{\rho_1}(p,q)\mathbf{C}_N(l,q) \\ &= c_1\rho_1\kappa_k\kappa_l \frac{2}{N} \left(\sum_{p=0}^{N-1} \cos\frac{k(p+\frac{1}{2})\pi}{N}\rho_1^{N-1-p}\right) \\ &\quad \times \left(\sum_{q=0}^{N-1} \cos\frac{l(q+\frac{1}{2})\pi}{N}\rho_1^{N-1-q}\right). \end{aligned} \tag{5.97}$$

Let $m = N-1-p$ and $n = N-1-q$, (5.97) becomes

$$\begin{aligned}\mathbf{M}_N^1(k,l) &= c_1\rho_1\kappa_k\kappa_l \frac{2}{N} \left(\sum_{m=0}^{N-1} \cos\left(\frac{k(m+\frac{1}{2})\pi}{N}+k\pi\right)\rho_1^m\right) \\ &\quad \times \left(\sum_{n=0}^{N-1} \cos\left(\frac{l(n+\frac{1}{2})\pi}{N}+l\pi\right)\rho_1^n\right) \\ &= (-1)^{k+l}c_1\rho_1\kappa_k\kappa_l \frac{2}{N} \left(\sum_{m=0}^{N-1} \cos\frac{k(m+\frac{1}{2})\pi}{N}\rho_1^m\right) \\ &\quad \times \left(\sum_{n=0}^{N-1} \cos\frac{l(n+\frac{1}{2})\pi}{N}\rho_1^n\right). \end{aligned} \tag{5.98}$$

We find that $\mathbf{M}_N^1(k,l) = (-1)^{k+l}\mathbf{L}_N^1(k,l)$ by comparing (5.94) and (5.98).

Finally, we evaluate the elements of the matrix $\mathbf{U}_N^1$ as

$$\begin{aligned}\mathbf{U}_N^1(k,l) &= c_1 \sum_{m,n=0}^{N-1} \mathbf{C}_N(k,m)\mathbf{R}_N^1(m,n)\mathbf{C}_N(l,n) \\ &= c_1\kappa_k\kappa_l \frac{2}{N} \sum_{m,n=0}^{N-1} \cos\frac{k(m+\frac{1}{2})\pi}{N} \cos\frac{l(n+\frac{1}{2})\pi}{N} \rho_1^{|m-n|}. \end{aligned} \tag{5.99}$$

Proceeding the same way as we did for $\mathbf{L}_N^1$, we get

$$\mathbf{U}_N^1(k,l) = \begin{cases} -2\mathbf{L}_N^1(k,l) + \mathcal{O}(\rho_1^N) & k+l \text{ even}, k \neq l \\ \dfrac{c_1(1-\rho_1^2)\kappa_k\kappa_l}{1-2\rho_1\cos(k\pi/N)+\rho_1^2} & k = l \\ 0 & k+l \text{ odd}. \end{cases} \tag{5.100}$$

Adding the matrices $\mathbf{L}_N^1$, $\mathbf{M}_N^1$ and $\mathbf{U}_N^1$ together, the matrix $\mathbf{A}_N^1$ is obtained as

$$\mathbf{A}_N^1(k,l) = \begin{cases} \mathcal{O}(\rho_1^N) & k+l \text{ even}, k \neq l \\ \dfrac{c_1(1-\rho_1^2)\kappa_k^2}{1-2\rho_1\cos(k\pi/N)+\rho_1^2} + 2\mathbf{L}_N^1(k,k) & k = l \\ 0 & k+l \text{ odd}. \end{cases} \tag{5.101}$$

Comparing (5.100) with (5.101), it is seen that the diagonal elements of $\mathbf{A}_N^1$ and $\mathbf{U}_N^1$ differ by $2\mathbf{L}_N^1(k,k)$, which decrease with the step $1/N$ as N increases. This guarantees the asymptotic equivalence $\operatorname{diag}\mathbf{A}_N^1 \approx \operatorname{diag}\mathbf{U}_N^1$. For nondiagonal elements of $\mathbf{A}_N^1$, which are either zero or in $\mathcal{O}(\tilde{\rho}_1^N)$. This guarantees the matrix $\mathbf{A}_N^1$ asymptotically becomes diagonal. As a result, we have that $\mathbf{A}_N^1 \approx \operatorname{diag}\mathbf{U}_N^1$.

Proceeding the same way as we did for the asymptotic equivalence $\mathbf{A}_N^1 \approx \operatorname{diag}\mathbf{U}_N^1$, it can be proved that $\mathbf{A}_N^2 \approx \operatorname{diag}\mathbf{U}_N^2$. Therefore, we have $\mathbf{A}_N^1 - \mathbf{A}_N^2 \approx \operatorname{diag}\mathbf{U}_N$, which implies $\mathbf{C}_N\mathbf{Z}_N\mathbf{C}_N^H \approx \operatorname{diag}\mathbf{U}_N$. Finally, we have $\tilde{\mathbf{X}}_N \approx \mathbf{R}_N^{-1}\mathbf{D}_N$.

Now we solve for the eigenvalues from the equivalent equation

$$\lim_{N\to\infty} \det(\tilde{\mathbf{X}}_N - \tilde{\lambda}\mathbf{I}) = 0, \tag{5.102}$$

where $\tilde{\lambda}$ denotes the inverse value of λ. The $N-4$ eigenvalues of $\tilde{\mathbf{X}}_N$ that are given by the eigenvalues of the $(N-4)\times(N-4)$ central identity submatrix of $\tilde{\mathbf{X}}_N$ are equal to 1. The rest four of them computed from (5.102) are given by

$$\begin{aligned}\tilde{\lambda}_{1,2} &= 1 + 0.5\left[s_1(1+c) + cs_{N-2}\right] \pm \tau_1, \\ \tilde{\lambda}_{3,4} &= 1 + 0.5\left[s_1(1+c) - cs_{N-2}\right] \pm \tau_2, \end{aligned} \tag{5.103}$$

where τ_1 and τ_2 are given by

$$\tau_1 = \sqrt{a_1 - b_1},\ \ \tau_2 = \sqrt{a_2 - b_2} \tag{5.104}$$

with

$$\begin{aligned}
a_1 &= \left(2 + s_1(1+c) - s_{N-2}\right)^2, \\
b_1 &= 4\left[\left(1 + s_1(1+c) - c^2\right) - c\left((1+s_1)s_{N-2} - s_{N-1}\right)\right], \\
a_2 &= \left(2 + s_1(1+c) + s_{N-2}\right)^2, \\
b_2 &= 4\left[\left(1 + s_1(1+c) - c^2\right) + c\left((1+s_1)s_{N-2} - s_{N-1}\right)\right].
\end{aligned}$$

For large N, the terms s_{N-1} and s_{N-2} converge to zero. Therefore, we can approximate (5.103) as

$$\begin{aligned}
\tilde{\lambda}_{1,2} &= 1 + 0.5s_1(1+c) \pm \tau, \\
\tilde{\lambda}_{3,4} &= 1 + 0.5s_1(1+c) \pm \tau,
\end{aligned} \tag{5.105}$$

with τ given by (5.68). We conclude that the asymptotic eigenvalue spread of the auto-correlation matrix $\mathbf{S}_N$ after DCT and power normalization is such that

$$\lim_{N\to\infty}\left(\frac{\lambda_{max}}{\lambda_{min}}\right) = \frac{1 + 0.5s_1(1+c) + \tau}{1 + 0.5s_1(1+c) - \tau}. \tag{5.106}$$

Remark 4: Noticing from (5.105) and (5.67) that the asymptotic eigenvalues for DCT-LMS are subset of the asymptotic eigenvalues for DFT-LMS. It is also the case for Markov-1 input signals seen from [88]. This may trace back to the transform formula as the DCT transform matrix can be decomposed into two exponential forms similar to DFT.

Remark 5: From (5.106) we can easily notice that the DCT is a better decorrelator compared to DFT for the second-order AR process inputs. To illustrate it, in Figure 5.5 the 3-D plot of matrix $\tilde{\mathbf{X}}_N$ in DCT-LMS is plotted. It is observed that the non-zero elements are distributed around the main diagonal, which introduces smaller eigenvalue spreads when compared to DFT-LMS as shown Figure 5.4.

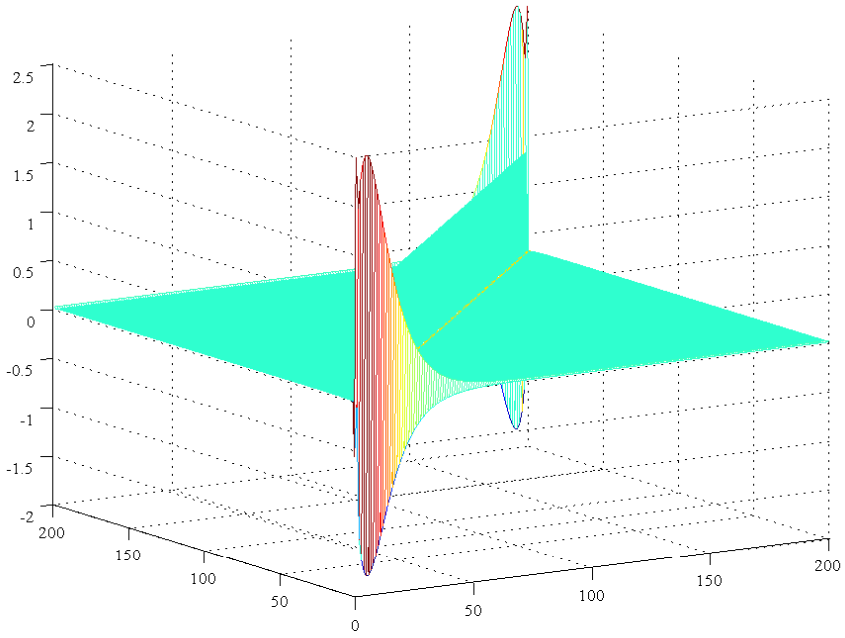

Figure 5.5: 3-D plot of the matrix $\tilde{\mathbf{X}}_N$ for DCT-LMS.

5.4.8 DST-LMS with Second-Order AR Inputs

Similarly, for DST-LMS we can obtain an asymptotic equivalent matrix of $\mathbf{D}_N$ from the derivation for first-order case, which is

$$\tilde{\mathbf{D}}_N(k,l) = \begin{cases} \tilde{\rho}^k - \tilde{\rho}^{k+2} & l = 0 \\ \tilde{\rho}^{|k-l|} - \tilde{\rho}^{k+l+2} + \tilde{\rho}^{2N-k-l} & 1 \le l \le N-2 \\ \tilde{\rho}^{N-k-1} - \tilde{\rho}^{N-k+1} & l = N-1. \end{cases} \tag{5.107}$$

Again, proceeding the similar computations and plotting of $\mathbf{R}_N^{-1}\tilde{\mathbf{D}}_N$ for different values of N, we found the following matrix

$$\tilde{\mathbf{X}}_N = \begin{pmatrix} 1-s_2 & -s_3 & -s_4 & \cdots & -s_{N-1} & -s_N & 0 \\ -cs_1 & 1-cs_2 & -cs_3 & \cdots & -cs_{N-2} & -cs_{N-1} & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & & \ddots & & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 & 0 & 0 \\ 0 & -cs_{N-1} & -cs_{N-2} & \cdots & -cs_3 & 1-cs_2 & -cs_1 \\ 0 & -s_N & -s_{N-1} & \cdots & -s_4 & -s_3 & 1-s_2 \end{pmatrix}. \tag{5.108}$$

The proof of $\tilde{\mathbf{X}}_N \approx \mathbf{R}_N^{-1}\mathbf{D}_N$ follows the same procedures as we did for DCT-LMS. Similarly, we need to show that $\mathbf{S}_N\mathbf{R}_N\tilde{\mathbf{X}}_N\mathbf{S}_N^H \approx \operatorname{diag}\mathbf{U}_N$ where $\mathbf{U}_N = \mathbf{S}_N\mathbf{R}_N\mathbf{S}_N^H$.

Using the expressions for $\mathbf{R}_N$ and $\tilde{\mathbf{X}}_N$ in (5.108), the elements of $\mathbf{R}_N\tilde{\mathbf{X}}_N$ are obtained as

$$\mathbf{R}_N\tilde{\mathbf{X}}_N(k,l) = \begin{cases} \tilde{\rho}^k - \tilde{\rho}^{k+2} & l = 0 \\ \tilde{\rho}^{|k-l|} - \tilde{\rho}^{k+l+2} + \tilde{\rho}^{2N-k-l} & 1 \le l \le N-2 \\ \tilde{\rho}^{N-k-1} - \tilde{\rho}^{N-k+1} & l = N-1. \end{cases} \tag{5.109}$$

It turns out that the following matrix $\mathbf{Z}_N$ is asymptotic equivalent to $\mathbf{R}_N\tilde{\mathbf{X}}_N$:

$$\mathbf{Z}_N(k,l) = \tilde{\rho}^{|k-l|} - \tilde{\rho}^{k+l+2} + \tilde{\rho}^{2N-k-l}. \tag{5.110}$$

Then it is sufficient to show $\mathbf{S}_N\mathbf{Z}_N\mathbf{S}_N^H \approx \operatorname{diag}\mathbf{U}_N$. By decomposing $\mathbf{Z}_N$ as $\mathbf{Z}_N = c_1\mathbf{Y}_N^1 + c_2\mathbf{Y}_N^2$ with $\mathbf{Y}_N^1$ and $\mathbf{Y}_N^2$ are given as $\mathbf{Y}_N^1 = \mathbf{R}_N^1 - \rho_1^2\mathbf{G}_N^{\rho_1} - \rho_1^{2N}\mathbf{G}_N^{1/\rho_1}$ and $\mathbf{Y}_N^2 = \mathbf{R}_N^2 - \rho_2^2\mathbf{G}_N^{\rho_2} - \rho_2^{2N}\mathbf{G}_N^{1/\rho_2}$, where the matrices $\mathbf{G}_N^{\rho_1}$, $\mathbf{G}_N^{\rho_2}$, $\mathbf{G}_N^{1/\rho_1}$ and $\mathbf{G}_N^{1/\rho_2}$ are defined the same way as we made for DCT-LMS, we have formulated two similar problems which can be solved by the same way as we did for DST-LMS with first-order AR process. Following the same procedures, it can be proved that $\mathbf{S}_N\mathbf{Y}_N^1\mathbf{S}_N^H \approx \operatorname{diag}\mathbf{U}_N^1$ and $\mathbf{S}_N\mathbf{Y}_N^2\mathbf{S}_N^H \approx \operatorname{diag}\mathbf{U}_N^2$, where $\mathbf{U}_N^1 = \mathbf{S}_N\mathbf{R}_N^1\mathbf{S}_N^H$ and $\mathbf{U}_N^2 = \mathbf{S}_N\mathbf{R}_N^2\mathbf{S}_N^H$. Noting that $\operatorname{diag}\mathbf{U}_N = c_1\operatorname{diag}\mathbf{U}_N^1 + c_2\operatorname{diag}\mathbf{U}_N^2$, we therefore conclude that $\mathbf{S}_N\mathbf{Z}_N\mathbf{S}_N^H \approx \operatorname{diag}\mathbf{U}_N$, which is the desired result.

The eigenvalues of $\mathbf{V}_N$ in DST-LMS are then obtained from computation of the eigenvalues of $\tilde{\mathbf{X}}_N^{DST}$: the $N-4$ eigenvalues of $\mathbf{V}_N$ converge to one, and for the remaining four eigenvalues, two of them converge to

$$\lambda_{1,2} = 1/\left(1 - 0.5s_2(1+c) + \tau'\right) \tag{5.111}$$

and another two converge to

$$\lambda_{3,4} = 1/\left(1 - 0.5s_2(1+c) - \tau'\right) \tag{5.112}$$

with τ' is given by

$$\tau' = 0.5\sqrt{s_2^2(1-c)^2 + 4cs_1s_3}. \tag{5.113}$$

Now, we can conclude that the eigenvalue spread of $\mathbf{V}_N$ in DST-LMS is such that

$$\lim_{N\to\infty}\left(\frac{\lambda_{max}}{\lambda_{min}}\right) = \frac{1 - 0.5s_2(1+c) + \tau'}{1 - 0.5s_2(1+c) - \tau'}. \tag{5.114}$$

5.4.9 Towards Eigenvalue Distributions with Higher Order AR Inputs

In this section, we will discuss that the methodology of finding the asymptotic eigenvalues of the autocorrelation matrix of $\mathbf{V}_N$ discussed for second-order AR process is extendable to higher order AR process where M is larger than two. The procedures will be quite similar, however, the computations will be increased.

The matrix $\mathbf{D}_N$ of the inverse problem as shown in (5.28) with the general AR process can be expressed as

$$\mathbf{D}_N = \sum_{k=1}^{M} c_k \mathbf{T}_N^H (\mathrm{diag}(\mathbf{T}_N \mathbf{R}_N^k \mathbf{T}_N^H) \mathbf{T}_N). \tag{5.115}$$

Similarly, we define that $\mathbf{D}_N^k \triangleq \mathbf{T}_N^H (\mathrm{diag}(\mathbf{T}_N \mathbf{R}_N^k \mathbf{T}_N^H) \mathbf{T}_N)$. From the derivation of second-order case, we know the form of $\mathbf{D}_N^k$ can be directly obtained from the results of $\mathbf{D}_N$ or its asymptotic equivalent matrix for the first-order case. The only difference is on the parameter ρ_k. The linear combination of $\mathbf{D}_N^k$ and c_k gives the matrix $\mathbf{D}_N$ for any Mth order AR process. Since it can easily be proved that $\mathbf{D}_N^{DHTk} \approx \mathbf{D}_N^{DFTk}$ as we discussed for second-order case, therefore, the asymptotic equivalence $\mathbf{D}_N^{DHT} \approx \mathbf{D}_N^{DFT}$ always be established for any Mth order AR process. This implies that DFT-LMS and DHT-LMS have the same convergence performance for the algorithms of long length and AR process. To determine the eigenvalue spread, the next thing is to compute and plot the matrix product $\mathbf{X}_N = \mathbf{R}_N^{-1} \mathbf{D}_N$ when the various transformation matrices are used. The difficulties are due to the computation of $\mathbf{R}_N^{-1}$ and seeking the simple matrices such that $\tilde{\mathbf{X}}_N \approx \mathbf{X}_N$. Since the results of higher-order cases cover the results of low-order cases, the simple matrices $\tilde{\mathbf{X}}_N$ for higher-order cases should be more complicated. It will be a challenge problem of computing the simple matrix $\tilde{\mathbf{X}}_N$ as well as its eigenvalue distribution.

5.4.10 Comparisons on Convergence Performance

By using the results that are obtained in the previous sections, we now are ready to compare the convergence performance of the various transform-domain LMS algorithms

for both first-order and second-order AR processes. The results can also be interpreted as the comparisons of orthogonalizing capabilities of the various fixed data-independent transforms for second-order AR process.

For the first-order AR process with the root $\rho \in (-1, 1)$, the convergence relations are easily compared based on the asymptotic eigenvalue spreads obtained for the first-order AR process. It can be summarized as DCT-LMS>DST-LMS>DFT-LMS=DHT-LMS>LMS for $(\sqrt{5}-1)/2 < \rho < 1$, and DST-LMS≥DCT-LMS>DFT-LMS=DHT-LMS>LMS for $-1 < \rho \leq (\sqrt{5}-1)/2$, where ρ is the correlation parameter of the process. We can see that when ρ is equal to zero, the AR process is white noise, and all the adaptive algorithms will therefore show equivalent convergence performance.

For the second-order case with real or complex roots of $|\rho_1|, |\rho_2| < 1$, the convergence relation for DFT-LMS, DCT-LMS and DHT-LMS can be directly obtained from the expressions of asymptotic eigenvalue spreads, which is DCT-LMS>DFT-LMS=DHT-LMS. However, the comparisons of the convergence relations between DFT-LMS, DCT-LMS and DST-LMS are not straightforward from the expressions. Using the asymptotic eigenvalue spreads of $\mathbf{V}_N$ in terms of the parameters ρ_1 and ρ_2 obtained in (5.69), (5.106) and (5.114), we can plot the asymptotic eigenvalue spreads of $\mathbf{V}_N$ versus ρ_1 and ρ_2 in DFT-LMS, DCT-LMS and DST-LMS respectively. Notice that the second-order AR process is given with $a_1 = -(\rho_1 + \rho_2)$ and $a_2 = \rho_1\rho_2$. Figures. 5.6, 5.7 and 5.8 show one of the plots for DFT-LMS, DCT-LMS and DST-LMS, where the values of ρ_1 and ρ_2 were distributed from 0 to 0.9, which indicates that the AR process is the lowpass case. It can be observed that when ρ_1 and ρ_2 are with small values, the values of (5.69), (5.106) and (5.114) are also small. That means all the algorithms show good convergence performance. However, when both ρ_1 and ρ_2 approach to unity, the eigenvalue spread of DCT-LMS have the least increases, and the values of DST-LMS and DFT-LMS increase significantly. By further comparing their subtractions between the eigenvalue spreads of (5.69), (5.106) and (5.114), we found that DCT-LMS and DST-LMS always achieve the smaller asymptotic eigenvalue spreads than DFT-LMS, and in general DCT-LMS achieves the smaller asymptotic eigenvalue spreads than DST-LMS. This implies that DCT-LMS gives the best convergence performance with such input process. The similar comparisons for all the other cases of ρ_1 and ρ_2 were made and the results are summarized

Table 5.1: Convergence Relations of DCT-LMS and DST-LMS for Second-Order AR Input Process

Region of Roots	Convergence Relations
$\rho_1, \rho_2 \in (0, 1)$	DST-LMS<DCT-LMS
$\rho_1 \in (0, 1), \rho_2 \in (-1, 0)$	DST-LMS>DCT-LMS
$\rho_1, \rho_2 \in (-1, 0)$	DST-LMS>DCT-LMS
$\rho_1, \rho_2 = a \pm bj, a \in (0, 1)$	DST-LMS<DCT-LMS
$\rho_1, \rho_2 = a \pm bj, a \in (-1, 0)$	DST-LMS>DCT-LMS

in Table 5.1. Since in general both DCT-LMS and DST-LMS always outperform DFT-LMS, DFT-LMS was not shown in Table 5.1.

The derivations for the asymptotic eigenvalue distribution show that instead of solving the original problem with infinite size, we actually manipulate a finite and exact eigenvalue problem. Note that the asymptotic eigenvalue spreads depend on the decay of the terms $\tilde{\rho}^N$, s_N or $\frac{l}{N}\tilde{\rho}^l$ in the derivations based on the strong asymptotic equivalence. As the size N increases, these terms are decreasing, and the simulated eigenvalue spreads of $\mathbf{V}_N$ should be approaching to the asymptotic values for large N. To see this, we listed in Table 5.2 that the simulated values of the eigenvalue spreads for several values of N for the process with the roots of $\rho_1 = 0.8$ and $\rho_2 = 0.7$. It is observed that the simulated values tend to the asymptotic values as N increases. Therefore, the derived asymptotic results of (5.69), (5.106) and (5.114) can provide good approximations for the finite-length algorithms.

For comparison, the eigenvalue spread in the standard LMS algorithm tends to 2.6×10^3 for $\rho_1 = 0.8$ and $\rho_2 = 0.7$. Therefore, the improvements of the convergence performance with the transform-domain LMS algorithms can be significant.

From the above comparisons, it can be concluded that none of them will always perform the best. For the first and second-order AR processes, the DCT-LMS and DST-LMS generally provide better convergence than DFT-LMS and DHT-LMS. Specifically,

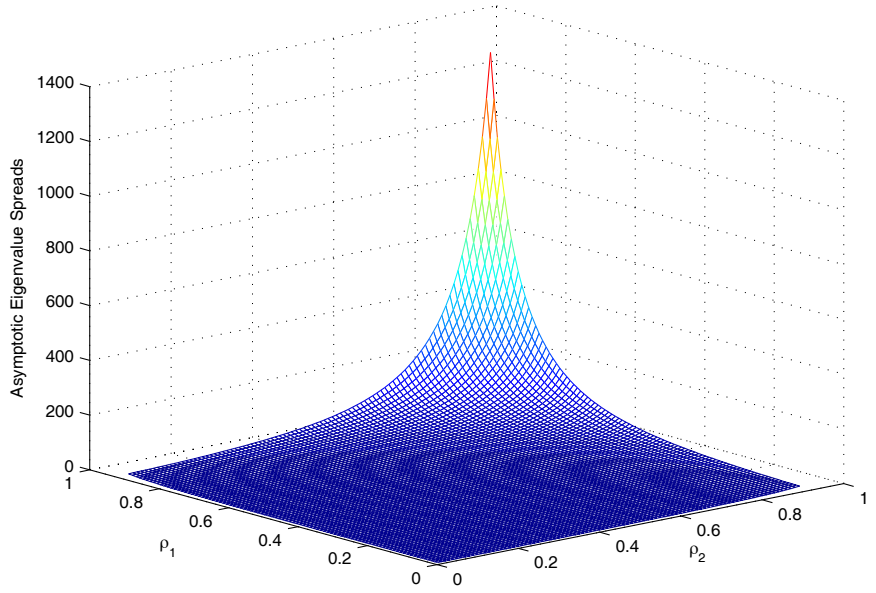

Figure 5.6: Asymptotic eigenvalue spread versus ρ_1 and ρ_2 in DFT-LMS for second-order lowpass AR process.

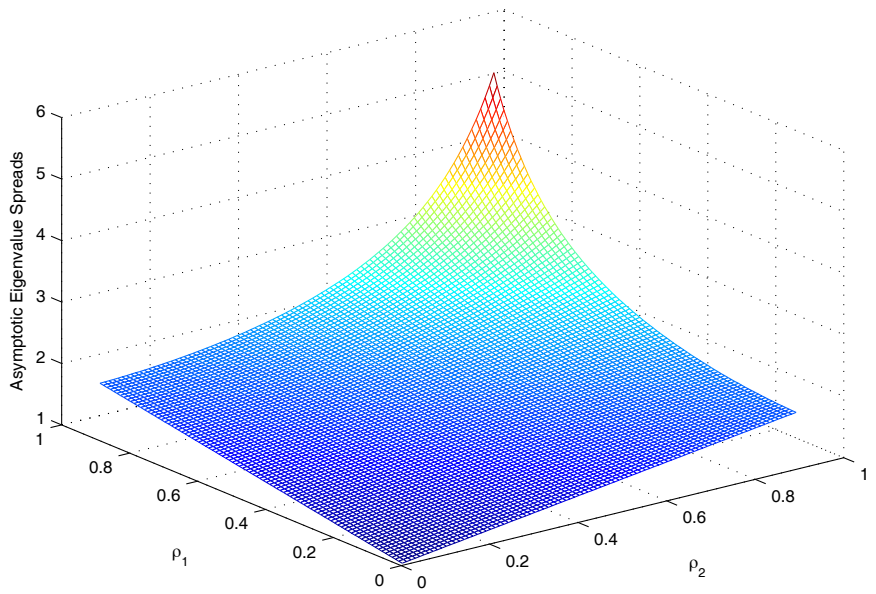

Figure 5.7: Asymptotic eigenvalue spread versus ρ_1 and ρ_2 in DCT-LMS for second-order lowpass AR process.

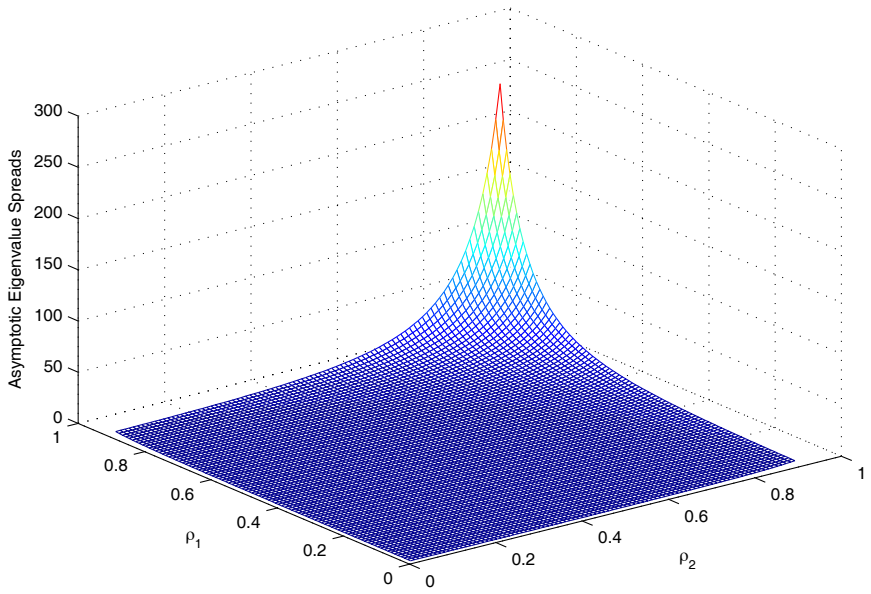

Figure 5.8: Asymptotic eigenvalue spread versus ρ_1 and ρ_2 in DST-LMS for second-order lowpass AR process.

Table 5.2: Eigenvalue Spreads of DFT-LMS, DHT-LMS, DCT-LMS and DST-LMS for Second-Order AR Input Process with $\rho_1 = 0.8, \rho_2 = 0.7$ for Increasing Size N

Algorithms	100	200	500	1000	$N \to \infty$
DFT-LMS	60.72	74.60	87.31	92.74	99.0
DHT-LMS	62.51	67.53	83.02	90.21	99.0
DCT-LMS	2.74	2.82	2.87	2.89	2.91
DST-LMS	16.21	16.33	16.89	17.16	17.49

the DCT-LMS performs the best for the case of lowpass inputs, whereas DST-LMS performs the best for the cases of bandpass and highpass inputs. These results can also be interpreted as comparing orthogonalizing capabilities of the various fixed data-independent transforms. Since the power spectral density of the underlying input process can be roughly estimated, the purpose of selecting an appropriate transform shall be

achieved.

5.5 Simulation Results

5.5.1 Evaluation with System Identification

In this section, we examine the analytical results obtained in previous sections through modeling the problem of system identification with the transform-domain LMS algorithms as well as the LMS adaptive algorithm for comparison.

The unknown plant to be modeled is a 21^{st} order finite impulse response (FIR) lowpass plant with cutoff frequency of 0.5, designed using a Hamming window. The input signal is applied to both the unknown plant and the adaptive algorithms that have the same number of weight taps. A white Gaussian noise is added to the system such that signal-to-noise ratio is 30dB. For all the simulations, the form of (5.2) was used. We used $\beta = 0.95$ for the estimate of the input power. The initial weight vector of LMS was set on a slow mode, and the initial weight vectors of the other algorithms are obtained by transforming from the initial weight vector of LMS by the corresponding transformation matrices. All learning curves have been averaged over 200 ensembles.

Four inputs from the AR process are selected for the simulations and generated as follows. The first input is $x_n = -0.9x_{n-1} + \omega_n$ with $\rho = -0.9$, the second input is $x_n = 1.5x_{n-1} - 0.56x_{n-2} + \omega_n$ with $\rho_1 = 0.8, \rho_2 = 0.7$, and the third input is $x_n = -1.6x_{n-1} - 0.89x_{n-2} + \omega_n$ with $\rho_1 = -0.8 + 0.5j, \rho_2 = -0.8 - 0.5j$, where ω_n is white noise with the variance that makes the power of x_n equal unity. The last input is $x_n = -1.6x_{n-1} - 0.89x_{n-2} + \omega_n$ with $\rho_1 = -0.8 + j0.5, \rho_2 = -0.8 - j0.5$, where ω_n is colored and generated by $\omega_n = 0.5\omega_{n-1} + \phi_n$ and ϕ_n is white noise with the variance that makes the power of x_n equal unity.

According to the mean-square stability condition of (5.13), the step-size μ for all the algorithms should be less than 0.031. We chose relatively small step-size for good steady-state performance in the simulations. Figure 5.9 shows the learning curves of the various adaptive algorithms for the first input with a common step-size $\mu = 0.002$. Figures 5.10–5.12 show the learning curves for input two, three and four respectively, where the

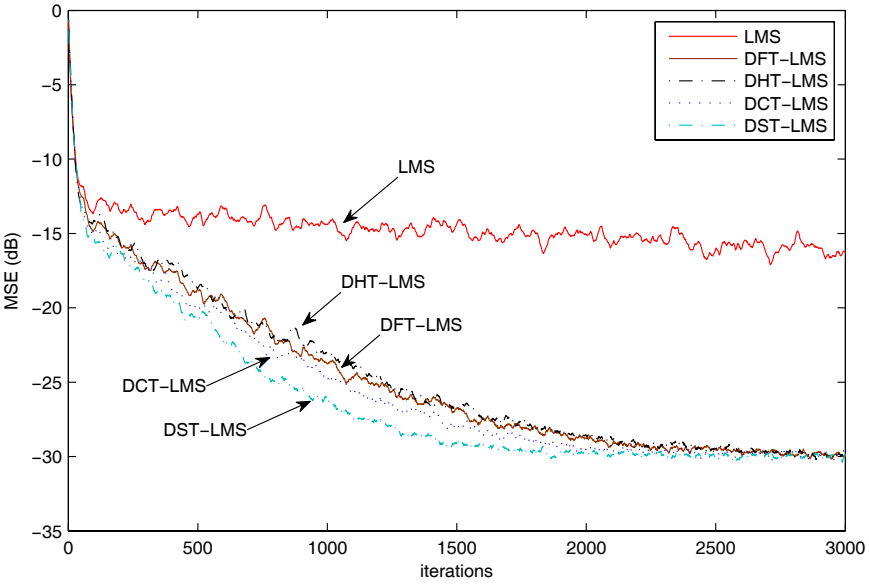

Figure 5.9: Learning curves of the various filters for input one.

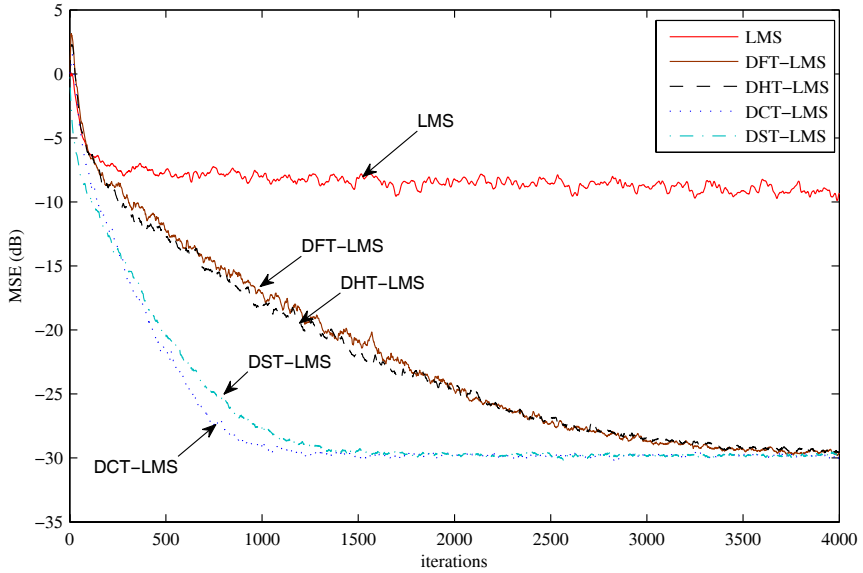

Figure 5.10: Learning curves of the various filters for input two.

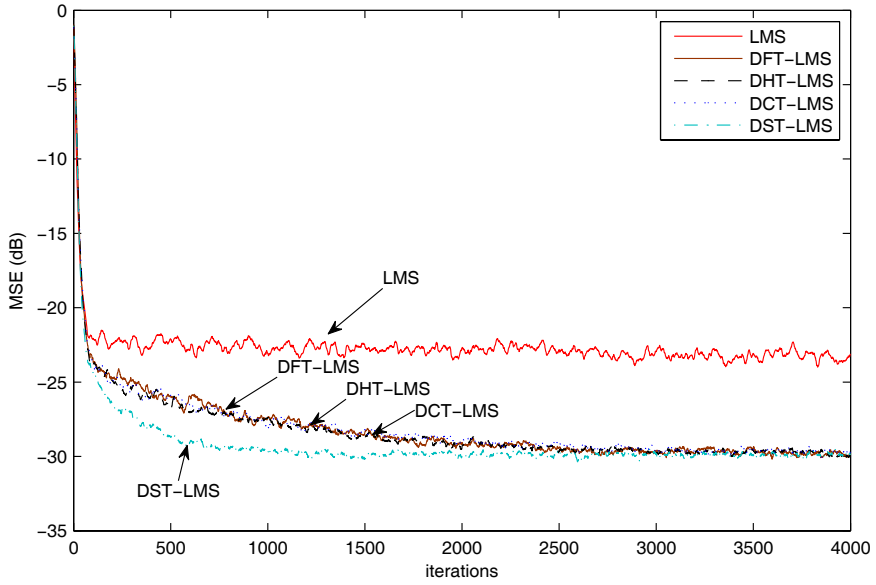

Figure 5.11: Learning curves of the various filters for input three.

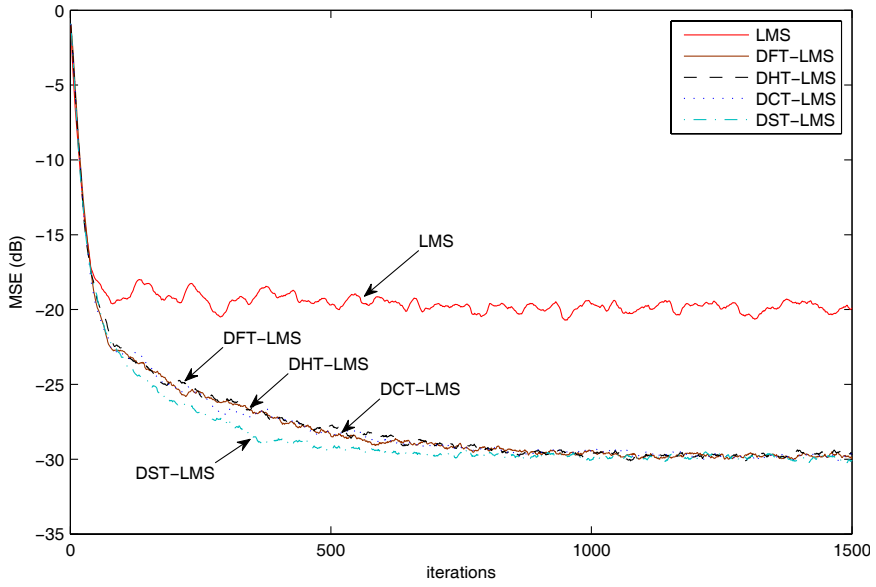

Figure 5.12: Learning curves of the various filters for input four.

Table 5.3: Comparison of Theoretical Misadjustments and Simulated Misadjustments

Inputs	One	Two	Three	Four
Theoretical	0.021	0.053	0.053	0.053
DFT-LMS	0.026	0.055	0.062	0.053
DHT-LMS	0.025	0.057	0.054	0.055
DCT-LMS	0.035	0.067	0.071	0.063
DST-LMS	0.024	0.055	0.059	0.055

common step-size was chosen as $\mu = 0.005$. From Figures 5.9–5.12, it can be observed that DCT-LMS and DST-LMS show better convergence performance than DFT-LMS and DHT-LMS, and LMS shows the worst convergence performance. Moreover, DCT-LMS gives better convergence performance with lowpass inputs as shown in Figure 5.10, whereas DST-LMS shows better convergence performance with bandpass and highpass inputs as shown in Figures 5.9, 5.11 and 5.12. DFT-LMS and DHT-LMS always show similar convergence performance. Table 5.3 shows that the theoretical misadjustments computed from (5.14) agree well with the simulated values. Therefore, the analytical results are supported by the computer simulations.

5.5.2 Evaluation with Other Input Processes

In this section, we take some other examples of third-order AR process, second-order moving-average (MA) process and autoregressive-moving-average (ARMA) with order (1,1). Let the three input processes be AR(3): $x_n = 1.8x_{n-1} - 1.05x_{n-2} + 0.2x_{n-3} + \omega_n$, MA(2): $x_n = \omega_n + 0.5\omega_{n-1} - 0.7\omega_{n-2}$ and ARMA(1,1): $x_n = -0.8x_{n-1} - 0.3\omega_{n-1} + \omega_n$ which are selected as lowpass, bandpass and highpass inputs respectively. Again, ω_n is white noise with the variance that makes the power of x_n equal unity. For the finite length filter of order $N = 100$, we compared the eigenvalue spreads of the input

Table 5.4: The Eigenvalue Spreads of the Input Autocorrelation Matrices of the Selected Input Processes with the Various Adaptive Filters.

Inputs	AR(3)	MA(2)	ARMA(1,1)
LMS	6.42×10^3	73.47	461.83
DFT-LMS	145.3	6.93	18.94
DHT-LMS	153.83	6.88	18.02
DCT-LMS	3.42	2.78	13.37
DST-LMS	26.92	1.55	3.74

autocorrelation matrices processed by LMS, DFT-LMS, DHT-LMS, DCT-LMS and DST-LMS, respectively, in Table 5.4. We observed that DCT-LMS performs the best for the lowpass third-order AR input process, whereas DST-LMS performs the best for the bandpass second-order MA input process and the highpass ARMA input process with order (1,1). In addition, DFT-LMS and DHT-LMS performs equivalently. These results may partially confirm our analytical results for the other classes of input process.

5.6 Concluding Remarks

In this chapter, we formulated the mean-square stability and steady-state performance of the transform-domain LMS algorithms, analyzed and compared their convergence performance based on the eigenvalue spreads of the preprocessed input autocorrelation matrices of second-order AR process. Specifically, we derived that the eigenvalue spread of the autocorrelation matrix of the first-order AR process after DST and power normalization tends to $1/(1-\rho^2)$, and for the second-order AR process, the eigenvalue spread achieved in DFT-LMS, DCT-LMS or DST-LMS tends to (5.69), (5.106) or (5.114), respectively. The comparisons of eigenvalue spreads and computer simulations show that in general DCT-LMS and DST-LMS give better convergence performance than DFT-LMS

and DHT-LMS, DCT-LMS works the best for the lowpass inputs, whereas DST-LMS works the best for the bandpass or highpass inputs. All the transform-domain LMS algorithms show better convergence performance than the plain LMS algorithm for the AR inputs with the condition that the input signals are correlated. In the next chapter, we will see another type of adaptive algorithms for fast convergence which are based on conjugate gradient method.

Chapter 6

Conjugate Gradient Algorithms[1]

6.1 Introduction

From our previous discussions, we know that the LMS algorithm is very popular for its simplicity and robustness. The incorporation of variable step-sizes and orthogonal transformations has successfully enhanced the performance of the LMS algorithm. However, for the adaptive filtering problem where the training data is not sufficient for the convergence of the LMS algorithms, the RLS-type algorithms based on deterministic minimization criterion should be more attractive. This is because the RLS-type algorithms are able to converge quickly and have a looser requirement for the amount of training data. However, the RLS algorithm usually have poor numerical stability and high computation cost. To be noted that more research on the fast implementation of the RLS-type algorithms have been carried out. Although the so-called fast RLS algorithms have computational cost as low as the LMS algorithm, they are likely to suffer the numerical problem.

In this chapter, we turn our focus to another type of adaptive algorithm called the conjugate gradient (CG) algorithm [94]. The CG algorithm was derived based on the CG method [95] and has been successfully applied for the adaptive filtering problem. To

[1]The results of this chapter were first published in Signal Processing, vol. 89, no. 5, pp. 894–900, May 2009.

Table 6.1: Basic CG Algorithm

Initialize $\mathbf{w}_0 = 0, \mathbf{g}_0 = \mathbf{p}_0 - \mathbf{R}\mathbf{w}_0, \mathbf{u}_1 = \mathbf{g}_0$

For $n = 1, 2, \cdots$,until convergence

$$\alpha_n = \frac{\|\mathbf{g}_{n-1}\|^2}{\mathbf{u}_n^T \mathbf{R} \mathbf{u}_n}$$

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \alpha_n \mathbf{u}_n$$

$$\mathbf{g}_n = \mathbf{g}_{n-1} - \alpha_n \mathbf{R} \mathbf{u}_n$$

$$\beta_n = \frac{\|\mathbf{g}_n\|^2}{\|\mathbf{g}_{n-1}\|^2}$$

$$\mathbf{u}_{n+1} = \mathbf{g}_n + \beta_n \mathbf{u}_n$$

End

make the description of the CG algorithm clear, next we briefly introduce the derivation and adaptation equations of the basic CG algorithm.

6.1.1 Basic CG Algorithm

Consider a minimization problem for the following quadratic performance function:

$$\xi(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T \mathbf{R} \mathbf{w} - \mathbf{p}^T \mathbf{w}, \tag{6.1}$$

where $\mathbf{R}$ is $N \times N$ square matrix and positive definite, $\mathbf{p}$ and $\mathbf{w}$ are vectors with dimension N. Solving for the vector $\mathbf{w}^o$ that minimizes the quadratic performance function (6.1) is equivalent to solving the linear equation $\mathbf{R}\mathbf{w}^o = \mathbf{p}$. It has been shown that the CG algorithm can recursively and efficiently solve this linear equation [95, 96] using the weight update equation listed in Table 6.1. The vector $\mathbf{w}_n$ is the estimate of $\mathbf{w}^o$ at index n, and the step size α_n is chosen to minimize the performance function $\xi(\mathbf{w}_{n-1} + \alpha_n \mathbf{u}_n)$ along the update direction, and β_n is chosen such that the current direction vector $\mathbf{u}_n$ is linearly independent and $\mathbf{R}$-conjugate to all the previous direction vectors $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_{n-1}$. The vector $\mathbf{w}_n$ finally converges to the optimal solution $\mathbf{w}^o$ in finite number of iterations.

It has been shown that the stability and convergence performance of the CG algorithm are highly determined by the parameters α_n and β_n, as well as the data windowing schemes for autocorrelation matrix $\mathbf{R}$ and cross-correlation vector $\mathbf{p}$. Recently, many modifications to the CG algorithm have been proposed [94, 97–101]. These modified CG algorithms are successfully applied into the adaptive equalizer, system identification, linear prediction and so on. These modified CG algorithms have convergence properties superior to those of the LMS algorithms and computational cost less than that of the classic RLS algorithm. Moreover, the instability problems existing in the RLS algorithm are not likely to occur in the CG algorithms.

In the next section, we provide a study about the design of the parameters α_n and β_n based on the RLS normal equation.

6.2 Design of New Parameters for the CG Algorithm

We provide an alternative design for the parameters α_n and β_n in this section. Consider the following RLS normal equation in sample-by-sample update case:

$$\mathbf{R}_{n-1}\mathbf{w}_n = \mathbf{p}_{n-1}, \tag{6.2}$$

where $\mathbf{w}_n$ is the estimate of the solution which minimizes the performance function (6.1) when the estimates of the autocorrelation $\mathbf{R}_{n-1}$ and cross-correlation $\mathbf{p}_{n-1}$ are available. We have known that the solution of this normal equation for the $n-1$ data inputs is the least square estimate. For the CG algorithm, the step size α_n can be easily derived from the normal equation (6.2). Substituting the weight update equation given in Table 6.1 into (6.2), after rearrangement we have

$$\mathbf{p}_{n-1} = \mathbf{R}_{n-1}(\mathbf{w}_{n-1} + \alpha_n \mathbf{u}_n). \tag{6.3}$$

Using the gradient vector $\mathbf{g}_{n-1} = \mathbf{p}_{n-1} - \mathbf{R}_{n-1}\mathbf{w}_{n-1}$ for (6.3), we yield:

$$\mathbf{u}_n = (\alpha_n \mathbf{R}_{n-1})^{-1}\mathbf{g}_{n-1}. \tag{6.4}$$

Equation (6.4) shows that the optimal vector $\mathbf{u}_n$ is actually the transformed gradient vector $\mathbf{g}_{n-1}$ by the matrix $(\alpha_n \mathbf{R}_{n-1})^{-1}$. Premultiplying (6.4) by $\alpha_n \mathbf{u}_n^T \mathbf{R}_{n-1}$ and through simple manipulations, the step size is obtained

$$\alpha_n = \frac{\mathbf{u}_n^T \mathbf{g}_{n-1}}{\mathbf{u}_n^T \mathbf{R}_{n-1} \mathbf{u}_n}. \tag{6.5}$$

Notice that the result of the step size (6.5) has the same form as given in [99]. Therefore, it minimizes the following quadratic performance function along the searching direction $\mathbf{u}_n$

$$\xi\big(\mathbf{w}_n\big) = \frac{1}{2}\mathbf{w}_n^T \mathbf{R}_{n-1} \mathbf{w}_n - \mathbf{p}_{n-1}^T \mathbf{w}_n. \tag{6.6}$$

It can be shown that with the knowledge of the direction $\mathbf{u}_n$ (6.4), we can reach the least square solution in one step. However, it may not be possible to get this direction without the matrix inverse. What the CG algorithm does is to divide this optimal direction into several orthogonal directions. When sample-by-sample update is performed, the orthogonality property is lost between the update directions. In that case, the direction vector needs to be reset to the true gradient periodically for the convergence of the algorithm, leading to slow convergence. A non-reset method called Polak-Ribiere method proposed in [96]. We now present an alternative non-reset method by considering the update direction $\mathbf{u}_{n+1}$ that minimizes the norm $\|\mathbf{u}_{n+1} - (\alpha_n \mathbf{R}_{n-1})^{-1} \mathbf{g}_n\|$. We design the parameter β which is given by

$$\beta_n = \underset{\beta_n \in \mathcal{R}}{\operatorname{argmin}} \left\| \mathbf{u}_{n+1} - (\alpha_n \mathbf{R}_{n-1})^{-1} \mathbf{g}_n \right\| \tag{6.7}$$

where the direction $\mathbf{u}_{n+1}$ is computed as:

$$\mathbf{u}_{n+1} = \mathbf{g}_n + \beta \mathbf{u}_n. \tag{6.8}$$

To solve for β_n, substituting (6.8) into (6.7) and setting the derivative of the norm with respect to β_n to zero, we obtain

$$\beta_n = \frac{{\mathbf{u}_n}^T \mathbf{g}_n - \alpha_n {\mathbf{g}_n}^T \mathbf{R}_n \mathbf{u}_n}{\alpha_n {\mathbf{u}_n}^T \mathbf{R}_n \mathbf{u}_n} \tag{6.9}$$

The parameter (6.9) is considered as a non-reset method since the best approximation of the true direction is always ensured.

6.3 The Data Windowing Schemes for the CG Algorithms

6.3.1 The Existing Data Windowing Schemes

When implementing the adaptive CG algorithm for adaptive filtering problem with the input vector of dimension N denoted by $\mathbf{x}_n$ and the desired output d_n, the autocorrelation matrix $\mathbf{R} \triangleq E[\mathbf{x}_n \mathbf{x}_n^T]$ and the cross-correlation vector $\mathbf{p} \triangleq E[d_n \mathbf{x}_n]$ are usually not available. The two data windowing schemes that have been used to estimate $\mathbf{R}$ and $\mathbf{p}$ are shown as follows [99].

a) Finite Sliding Data Windowing Scheme:

$$\mathbf{R}_n = \frac{1}{M} \sum_{j=n-M+1}^{n} \mathbf{x}_j \mathbf{x}_j^T \tag{6.10}$$

$$\mathbf{p}_n = \frac{1}{M} \sum_{j=n-M+1}^{n} d_j \mathbf{x}_j \tag{6.11}$$

b) Exponentially Decaying Data Windowing Scheme:

$$\mathbf{R}_n = \lambda \mathbf{R}_{n-1} + \mathbf{x}_n \mathbf{x}_n^T \tag{6.12}$$

$$\mathbf{p}_n = \lambda \mathbf{p}_{n-1} + d_n \mathbf{x}_n \tag{6.13}$$

where λ is the forgetting factor.

Notice that in the first approach the time ensemble average over a window of finite length M is evaluated. The accuracy of the estimated $\mathbf{R}_n$ and $\mathbf{p}_n$ is highly related to the data window size M. When small window size of $M < N$ is used, the estimates of $\mathbf{R}$ and $\mathbf{p}$ have a high variance and non-full rank. Since the modified CG algorithms with this approach tend to converge to the form $\mathbf{R}_n^{-1}\mathbf{p}_n$ for each data update [99], the misadjustment is usually very high. Moreover, since the estimate of correlation matrix is not full rank, the direction vector that is used to update the filter coefficients cannot be projected to all N directions, thus the modified CG algorithms show slow convergence. On the other hand, though large window size can be used to improve the overall performance, the computational cost will increase dramatically. In the second approach, the

exponential form provides a long memory of the estimates, and variances of the estimated $\mathbf{R}_n$ and $\mathbf{p}_n$ eventually converge to zero. Therefore, the misadjustment that is comparable to that of the RLS algorithm can be achieved in the modified CG algorithms as seen in [99] and [100]. The issue may be pointed out that at the initial stages ($n < N$), the estimated correlation matrix is not full rank since the updating matrix $\mathbf{x}_n\mathbf{x}_n^T$ is always with rank one. The exponential decay due to the forgetting factor in the scheme introduces slow initial convergence of the estimated correlation matrix, which causes slow starting convergence of the modified CG algorithms that are implemented using this approach.

In the previous modified CG algorithms, these two data windowing schemes of estimating the correlation matrix and the cross-correlation vector have been applied. It was shown that the performance of the modified adaptive CG algorithms based on the iterative CG method for adaptive filtering is highly related to the ways of computing the correlation matrix and the cross-correlation vector. The current approaches of implementing the CG algorithm using the data windows of exponential form or sliding form result in either loss of convergence or increase in misadjustment. For instance, the analysis and simulations presented in [94, 99] show that the modified CG algorithms, which are implemented using the finite sliding data windowing scheme and run several iterations per data update, have the convergence behavior and misadjustment that are dependent on the length of the data window. When small window size is used, the convergence rate becomes slow and the misadjustment increases. On the other hand, the modified CG algorithms, which are implemented using the exponentially decaying data windowing scheme and run one iteration per coefficient and data update, have the convergence that is dependent on the input eigenvalue spread. When the input with large eigenvalue spread is applied, the convergence rate is considerably slow.

6.3.2 A New Generalized Data Windowing Scheme

To improve the convergence performance as well as the misadjustment of the modified CG algorithms, in the this section, we present a new generalized data windowing scheme which combines the features of the finite sliding data windowing scheme and the exponentially decaying data windowing scheme given as follows:

c) Generalized Data Windowing Scheme:

$$\mathbf{R}_n = \lambda \mathbf{R}_{n-1} + \frac{1}{M} \sum_{j=0}^{M-1} \mathbf{x}_{n-j} \mathbf{x}_{n-j}^T \tag{6.14}$$

$$\mathbf{p}_n = \lambda \mathbf{p}_{n-1} + \frac{1}{M} \sum_{j=0}^{M-1} d_{n-j} \mathbf{x}_{n-j} \tag{6.15}$$

where λ is again the forgetting factor and M is the window size. As can be seen that the generalized data windowing scheme in (6.14) and (6.15) combines the exponentially decaying form and finite sliding form. In general, the exponential form accumulates all the past data for the estimation, thereby decreasing variance of the estimates. Compared to the finite sliding data windowing scheme, this new scheme improves the misadjustment. On the other hand, the finite sliding form is used for the update, resulting in the rank of the updating matrix equal to N when $M \geq N$ and $n \geq N$. As a result, this scheme changes the weighting of the updating form so that the decay due to the forgetting factor is not exponentially decreasing any more. Though at the initial stages, the estimated correlation matrix is also not full rank, the initial convergence of the estimation increases due to $\mathbf{x}_{n-j}\mathbf{x}_{n-j}^T, j = 1, 2, \cdots, M-1$ are presents in $\lambda \mathbf{R}_{n-1}$. This improves the initial convergence of the modified CG algorithms.

Compared to the existing data window schemes, the new scheme has increased computational cost. However, it can be reduced to a reasonable level by reformulating $\mathbf{R}_n$ and $\mathbf{p}_n$ as follows.

$$\tilde{\mathbf{R}}_n = \tilde{\mathbf{R}}_{n-1} + \frac{1}{M} \left(\mathbf{x}_n {\mathbf{x}_n}^T - \mathbf{x}_{n-M} {\mathbf{x}_{n-M}}^T \right) \tag{6.16}$$

$$\tilde{\mathbf{p}}_n = \tilde{\mathbf{p}}_{n-1} + \frac{1}{M} \left(d_n \mathbf{x}_n - d_{n-M} \mathbf{x}_{n-M} \right) \tag{6.17}$$

$$\mathbf{R}_n = \lambda \mathbf{R}_{n-1} + \tilde{\mathbf{R}}_n \tag{6.18}$$

$$\mathbf{p}_n = \lambda \mathbf{p}_{n-1} + \tilde{\mathbf{p}}_n \tag{6.19}$$

Here, we assume $\mathbf{x}_{n-M} = \mathbf{0}, d_{n-M} = 0$ for $n < M$. Compared to the scheme in (6.10) and (6.11), the new scheme introduces the extra computation for $\lambda \mathbf{R}_{n-1}$, and compared to the scheme in (6.12) and (6.13), the extra computation for $\mathbf{x}_{n-M}\mathbf{x}_{n-M}^T$ is introduced. The computational complexity in terms of number of multiplications and additions is listed

Table 6.2: A Comparison of the Estimated Computational Cost per Iteration for the Data Windowing Schemes

Schemes	Multiplications ($\times$)	Additions ($+$)
a)	$\frac{1}{2}N^2 + 2MN + \frac{3}{2}N$	$\frac{1}{2}(M-1)N^2 + \frac{3}{2}(M-1)N$
b)	$\frac{1}{2}N^2 + \frac{7}{2}N$	$\frac{1}{2}N^2 + \frac{3}{2}N$
c)Eqs. (6.14)-(6.15)	$N^2 + 2MN + 3N$	$\frac{1}{2}MN^2 + \frac{3}{2}MN + N$
c)Eqs. (6.16)-(6.19)	$N^2 + 7N$	$\frac{3}{2}N^2 + \frac{9}{2}N$

in Table 6.2. If the value of $\mathbf{x}_{n-M}\mathbf{x}_{n-M}^T$ is previously stored, this extra computation load can be avoided. Note that the recursive implementations of a sliding window shown in (6.16) and (6.17) can lead to considerable numerical problems when implemented in finite precision hardware, where finite-precision errors are linearly accumulating in the values of $\tilde{\mathbf{R}}_n$ and $\tilde{\mathbf{p}}_n$ over time. If left unchecked, such errors can make (6.16) and (6.17) useless. For this reason, we recommend the technique of a periodically-time-varying system designed in [102, 103] for approximating the sliding data window, which results in the following recursive implementation for $\tilde{\mathbf{R}}_n$:

$$\tilde{\mathbf{R}}_n = \begin{cases} \gamma[\tilde{\mathbf{R}}_{n-1} - \frac{1}{M}\mathbf{x}_{n-M}\mathbf{x}_{n-M}^T] + \frac{1}{M}\mathbf{x}_n\mathbf{x}_n^T \text{ if } n \text{mod} M = 0 \\ \tilde{\mathbf{R}}_{n-1} - \frac{\gamma}{M}\mathbf{x}_{n-M}\mathbf{x}_{n-M}^T + \frac{1}{M}\mathbf{x}_n\mathbf{x}_n^T \quad \text{otherwise,} \end{cases} \tag{6.20}$$

where $0 \ll \gamma < 1$. The cross-correlation vector $\tilde{\mathbf{p}}_n$ has a similar update. It is noteworthy that the technique used in (6.20) produces an output signal that is mathematically-equivalent to $\tilde{\mathbf{R}}_n$ in (6.16) at every Mth time instant and it stabilizes the marginal instability of this system [102, 103].

Using (6.16)–(6.20) and the weight update equation in Table 6.1, the residual vector can be computed as

$$\begin{aligned} \mathbf{g}_n &= \mathbf{p}_n - \mathbf{R}_n\mathbf{w}_n \\ &= \lambda\mathbf{g}_{n-1} - \alpha_n\mathbf{R}_n\mathbf{u}_n + \tilde{\mathbf{p}}_n - \tilde{\mathbf{R}}_n\mathbf{w}_{n-1}. \end{aligned} \tag{6.21}$$

6.4 New Accelerated CG Algorithms for Adaptive Filtering

Taking into account the generalized data windowing scheme with simplified computation, the pseudocode of the first modified CG algorithm is shown in Table 6.3. We reference it as GDW-CGI in this thesis. Note that we follow the same parameters of α_n and β_n as that are used in [99] except that the *ad hoc* parameter η of [99] is replaced by the forgetting factor λ here since it is proved to be an optimal choice in [100]. The convergence rate of GDW-CGI weakly depends on the eigenvalue spread of $\mathbf{R}$ and the dependence can be reduced by increasing the reusing data size M. The block processing allows the CG algorithms to converge faster than the CG algorithms with sample-by-sample processing. That is because the CG algorithm with block processing can run several iterations per data update. However, the block processing CG algorithms described in [94] and [99] has the output MSE dependent on the data window size. Usually a large length of data window must be used to achieve a misadjustment comparable to the RLS algorithm. The general data windowing scheme allows the CG algorithm to achieve as low misadjustment as the RLS algorithm independent of the data window size. In addition, the fast convergence is also ensured with independence of the eigenvalue spread $\mathbf{R}$. Following the same approach used in [94] and [99], the second modified CG algorithm referenced as GDW-CGII shown in Table 6.4 has the same form as CG2 in [99] except that the finite sliding data windowing scheme in CG2 is replaced with the generalized data windowing scheme. Note that the parameter β simplified in the block processing CG2 algorithm from the fact $\mathbf{u}_n^T\mathbf{g}_n = 0$ has the same form as designed by the control Liapunov function (CLF) method [100]. It is observed when the termination condition is set to $min(n, N, M)$, the GDW-CGII algorithm results in better performance. We believe that the generalized data windowing scheme can also be extended to other type CG algorithms like the constrained CG algorithm [97] and the sample-based CG algorithm [101].

Table 6.3: Modified CG Algorithm Using Generalized Data Windowing Scheme (GDW-CGI)

Initialize $\mathbf{w}_0 = 0, \mathbf{g}_0 = \mathbf{p}_0 - \mathbf{R}_0\mathbf{w}_0, \mathbf{u}_0 = \mathbf{g}_0$

For $k = 1, 2, \cdots$, until convergence

$$\tilde{\mathbf{R}}_n = \begin{cases} \gamma[\tilde{\mathbf{R}}_{n-1} - \frac{1}{M}\mathbf{x}_{n-M}\mathbf{x}_{n-M}^T] + \frac{1}{M}\mathbf{x}_n\mathbf{x}_n^T & \text{if } k\text{mod}M = 0, \\ \tilde{\mathbf{R}}_{n-1} - \frac{\gamma}{M}\mathbf{x}_{n-M}\mathbf{x}_{n-M}^T + \frac{1}{M}\mathbf{x}_n\mathbf{x}_n^T & \text{otherwise} \end{cases}$$

$$\tilde{\mathbf{p}}_n = \begin{cases} \gamma[\tilde{\mathbf{p}}_{n-1} - \frac{1}{M}d_{n-M}\mathbf{x}_{n-M}^T] + \frac{1}{M}d_n\mathbf{x}_n^T & \text{if } k\text{mod}M = 0, \\ \tilde{\mathbf{p}}_{n-1} - \frac{\gamma}{M}d_{n-M}\mathbf{x}_{n-M}^T + \frac{1}{M}d_n\mathbf{x}_n^T & \text{otherwise} \end{cases}$$

$$\mathbf{R}_n = \lambda\mathbf{R}_{n-1} + \tilde{\mathbf{R}}_n$$

$$\alpha_n = \lambda\frac{\mathbf{u}_n^T\mathbf{g}_{n-1}}{\mathbf{u}_n^T\mathbf{R}_n\mathbf{u}_n}$$

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \alpha_n\mathbf{u}_n$$

$$\mathbf{g}_n = \lambda\mathbf{g}_{n-1} - \alpha_n\mathbf{R}_n\mathbf{u}_n + \tilde{\mathbf{p}}_n - \tilde{\mathbf{R}}_n\mathbf{w}_{n-1}$$

$$\beta_n = \frac{(\mathbf{g}_n - \mathbf{g}_{n-1})^T\mathbf{g}_n}{\mathbf{g}_{n-1}^T\mathbf{g}_{n-1}}$$

$$\mathbf{u}_{n+1} = \mathbf{g}_n + \beta_n\mathbf{u}_n$$

End

6.4.1 Convergence Behavior of the GDW-CGI Algorithm in Norm-2 Sense

In this section, we show the convergence behavior of the GDW-CGI algorithm in norm-2 sense. Let us consider the instantaneous square deviation of the algorithm defined below.

$$\mathcal{D}_n = (\mathbf{w}^o - \mathbf{w}_n)^T(\mathbf{w}^o - \mathbf{w}_n) = \|\mathbf{w}^o - \mathbf{w}_n\|^2, \tag{6.22}$$

where $\mathbf{w}^o$ is considered as the vector of system coefficients and $\mathbf{w}_n$ is the nth estimate of $\mathbf{w}^o$. The difference is computed as

$$\Delta\mathcal{D}_n = \mathcal{D}_n - \mathcal{D}_{n-1} = \|\mathbf{w}^o - \mathbf{w}_n\|^2 - \|\mathbf{w}^o - \mathbf{w}_{n-1}\|^2. \tag{6.23}$$

Table 6.4: Modified CG Algorithm Using Generalized Data Windowing Scheme (GDW-CGII)

Initialize $\mathbf{w}_0 = 0$

For $n = 1, 2, \cdots$, until convergence

$$\tilde{\mathbf{R}}_n = \begin{cases} \gamma[\tilde{\mathbf{R}}_{n-1} - \frac{1}{M}\mathbf{x}_{n-M}\mathbf{x}_{n-M}^T] + \frac{1}{M}\mathbf{x}_n\mathbf{x}_n^T & \text{if } k\text{mod}M = 0, \\ \tilde{\mathbf{R}}_{n-1} - \frac{\gamma}{M}\mathbf{x}_{n-M}\mathbf{x}_{n-M}^T + \frac{1}{M}\mathbf{x}_n\mathbf{x}_n^T & \text{otherwise} \end{cases}$$

$$\tilde{\mathbf{p}}_n = \begin{cases} \gamma[\tilde{\mathbf{p}}_{n-1} - \frac{1}{M}d_{n-M}\mathbf{x}_{n-M}^T] + \frac{1}{M}d_n\mathbf{x}_n^T & \text{if } k\text{mod}M = 0, \\ \tilde{\mathbf{p}}_{n-1} - \frac{\gamma}{M}d_{n-M}\mathbf{x}_{n-M}^T + \frac{1}{M}d_n\mathbf{x}_n^T & \text{otherwise} \end{cases}$$

$\mathbf{R}_n = \lambda\mathbf{R}_{n-1} + \tilde{\mathbf{R}}_n$

$\mathbf{p}_n = \lambda\mathbf{p}_{n-1} + \tilde{\mathbf{p}}_n$

$\mathbf{g}(0) = \mathbf{p}_n - \mathbf{R}_n\mathbf{w}_{n-1}, \mathbf{u}(1) = \mathbf{g}(0)$

For $k = 1$ to $min(n, N, M)$

$\alpha_k = \dfrac{\mathbf{u}_k^T\mathbf{g}_{k-1}}{\mathbf{u}_k^T\mathbf{R}_n\mathbf{u}_k}$

$\mathbf{w}_k = \mathbf{w}_{k-1} + \alpha_k\mathbf{u}_k$

$\mathbf{g}_k = \mathbf{g}_{k-1} - \alpha_k\mathbf{R}_n\mathbf{u}_k$

$\beta_k = -\dfrac{\mathbf{g}_k^T\mathbf{R}_n\mathbf{u}_k}{\mathbf{u}_k^T\mathbf{R}_n\mathbf{u}_k}$

$\mathbf{u}_{k+1} = \mathbf{g}_k + \beta_k\mathbf{u}_k$

End

$\mathbf{w}_n = \mathbf{w}_k$

End

Substituting the weight update equation given in Table 6.1 into (6.23), the difference is rewritten as

$$\Delta\mathcal{D}_n = -2\alpha_n \left(\mathbf{w}^o - \mathbf{w}_{n-1}\right)^T \mathbf{u}_n + \alpha_n^2 \left\|\mathbf{u}_n\right\|^2. \tag{6.24}$$

Assuming the desired response d_n fits the multiple linear regression model

$$d_n = \mathbf{x}_n^T \mathbf{w}^o + e_n^o \tag{6.25}$$

where e_n^o accounts for white measurement noise with zero mean. Applying the general data windowing scheme (6.14) and (6.15) into the model (6.25) and noticing that the accumulating of $e_n^o \mathbf{x}_n$ converges to zero, we have

$$\mathbf{p}_n = \mathbf{R}_n \mathbf{w}^o. \tag{6.26}$$

Rewriting α_n using (6.26) yields

$$\alpha_n = \lambda \frac{\mathbf{u}_n^T \mathbf{R}_{n-1}(\mathbf{w}^o - \mathbf{w}_{n-1})}{\mathbf{u}_n{}^T \mathbf{R}_n \mathbf{u}_n}. \tag{6.27}$$

If we assume the input process is stationary, the α_n in (6.27) can be approximated as

$$\alpha_n \approx \lambda \frac{\mathbf{u}_n^T(\mathbf{w}^o - \mathbf{w}_{n-1})}{\|\mathbf{u}_n\|^2}. \tag{6.28}$$

Substituting (6.28) into (6.24), we obtain

$$\Delta \mathcal{D}_n = -\frac{\lambda(2-\lambda)\left(\mathbf{u}_n^T(\mathbf{w}^o - \mathbf{w}_{n-1})\right)^2}{\|\mathbf{u}_n\|^2} < 0. \tag{6.29}$$

From (6.23) and (6.29), we have

$$\begin{aligned} \lim_{n\to\infty} \mathcal{D}_n &= \lim_{n\to\infty} \mathcal{D}_{n-1} + \lim_{n\to\infty} \Delta \mathcal{D}_n \\ &= \mathcal{D}_0 + \lim_{n\to\infty} \sum_{i=1}^{n} \Delta \mathcal{D}_i \geq 0. \end{aligned} \tag{6.30}$$

Eq. (6.30) implies that

$$\lim_{n\to\infty} \sum_{i=1}^{n} \Delta \mathcal{D}_i \geq -\mathcal{D}_0. \tag{6.31}$$

Eq. (6.31) indicates that the term, $\Delta \mathcal{D}_n$, is summable for infinite n. Therefore, we can conclude that $\lim_{n\to\infty} \Delta \mathcal{D}_n = 0$, which is $\lim_{n\to\infty} \left(\mathbf{u}_n^T(\mathbf{w}^o - \mathbf{w}_{n-1})\right) = 0$. Since the vectors $\mathbf{u}_n$ and $\mathbf{w}^o - \mathbf{w}_{n-1}$ are not orthogonal, it can be concluded that the weight vector estimate $\mathbf{w}_n$ tends to the optimal weight vector $\mathbf{w}^o$ as n tends to infinity. The GDW-CGI algorithm, in theory, produces zero misadjustment, which is equivalent to the RLS algorithm [1]. Note that the result is also valid for the other CG algorithms with sample by sample implementation.

6.5 Simulation Results

6.5.1 Application to System Identification

The performance of the proposed algorithms is evaluated by carrying out the computer simulation in the framework of adaptive system modeling problem. We use a Hamming window to generate the unknown system as a finite impulse response (FIR) lowpass plant with cutoff frequency of 0.5. The filter length of N with a low order of 20 and a high order of 200 are tested. The adaptive filter and the unknown system are assumed to have the same number of taps. Both stationary and nonstationary input signals are used. To be specified, the stationary input signal is obtained by filtering a white, zero-mean, Gaussian random sequence through a first-order system $G(z) = 1/(1 - \rho z^{-1})$, where ρ is known as the single pole of the filter, and the nonstationary input signal is generated by the linear time-varying system $x_n = \mathbf{h}_n^T \mathbf{r}_n$, with $\mathbf{h}_0 = [0.1, 0.2, 0.3, 0.4, 0.5]$, the random walk model $\mathbf{h}_n = \mathbf{h}_{n-1} + \mathbf{v}_n$, and $\mathbf{r}_n$ and $\mathbf{v}_n$ are Gaussian noise vectors with zero mean and correlation matrices of $\sigma_r^2\mathbf{I}$ ($\sigma_r^2 = 1$) and $\sigma_v^2\mathbf{I}$ ($\sigma_r^2 = 10^{-4}$) ($\mathbf{I}$ stands for an identity matrix), respectively.

In the simulations, we choose a high SNR of 100 dB in order to show the performance for a wide range considering that the SNR has little effect on the performance of the CG algorithms. The estimate of mean-square deviation (MSD) defined by $E\left\|\mathbf{w}^o - \mathbf{w}_n\right\|^2$ is plotted via the iteration number $k = 1...n$ by ensemble averaging over 100 independent trials.

Figures 6.1–6.4 show the MSD for the CG-CLF algorithm [100], the Chang-Willson algorithm [99], the GDW-CGI algorithm, and the RLS algorithm, where the scheme of data windowing is with an exponential form. Figures 6.5–6.7 show the MSD for the standard CG algorithm [94] that uses the finite sliding data windowing scheme, the GDW-CGII algorithm, and the RLS algorithm. For all the algorithms, the parameters are chosen as $\lambda = \lambda_f = \eta = 0.99$ and $\gamma = 0.995$. In Figures 6.1 and 6.2, the window size is selected as $M = 10, 20, 30$ for the GDW-CGI algorithm to show how the convergence behavior is related to the window size. A filter length $N = 20$ and the input signals generated with the single poles $\rho = 0, 0.9$ are used to provide the input eigenvalue spreads equal to 1 and 212 respectively. To show the effects of the choice of λ on the GDW-CGI

algorithm, we also tested the case of $\lambda = 0.9$. In Figure 6.3, we test the tracking properties of the various algorithms for tacking the unknown time-varying system generated by a random walk model $\mathbf{w}_n^o = \mathbf{w}_{n-1}^o + \boldsymbol{\omega}_n$ with Gaussian noise vector $\boldsymbol{\omega}_n$ of zero mean and correlation matrix of $\sigma_\omega^2\mathbf{I}$ ($\sigma_\omega^2 = 10^{-10}$). We use $\rho = 0$ and $M = N = 20$. In Figure 6.4, we test the algorithms for high order case with $M = N = 200$ and $\rho = 0$. From Figures 6.1–6.4, we have made the following observations.

- The proposed GDW-CGI algorithm outperforms the other two CG algorithms by around N iterations, and has the convergence and tracking property comparable to the RLS algorithm for less correlated input signals provided that the window size M is close to N.

- The convergence rate of the proposed GDW-CGI algorithm is not very sensitive to the choice of λ, and improves as the window size M increases. However, the improvement becomes insignificant when $M > N$.

- The MSD of the proposed GDW-CGI algorithm decreases from the starting point, whereas the other compared algorithms start to decrease after several iterations depending on length of the filter.

- All the compared algorithms achieve the same misadjustment and equivalent tracking performance.

In Figures 6.5 and 6.6, we use $N = 20$ and $M = n_w = 10, 20$, where n_w represents the window size of the finite sliding data windowing scheme. Figure 6.5 shows the results for the stationary input signal of $\rho = 0$ and Figure 6.6 shows the results for the nonstationary input signal. As expected that the proposed GDW-CGII algorithm converges faster and has lower misadjustment compared to the standard CG algorithm [94] when $M = n_w = 10$ is used, this is because the estimated $\mathbf{R}_n$ using the finite sliding data windowing scheme has high variance and non-full rank. When we use $M = N = 20$, where $\mathbf{R}_n$ has full rank for both algorithms, it is observed that both algorithms converge in about N iterations. However, a higher misadjustment is produced by the standard CG algorithm. Moreover, when $M = N$ is chosen, the computational cost will be considerably high. Compared to the RLS algorithm, the proposed GDW-CGII algorithm converges faster. All the

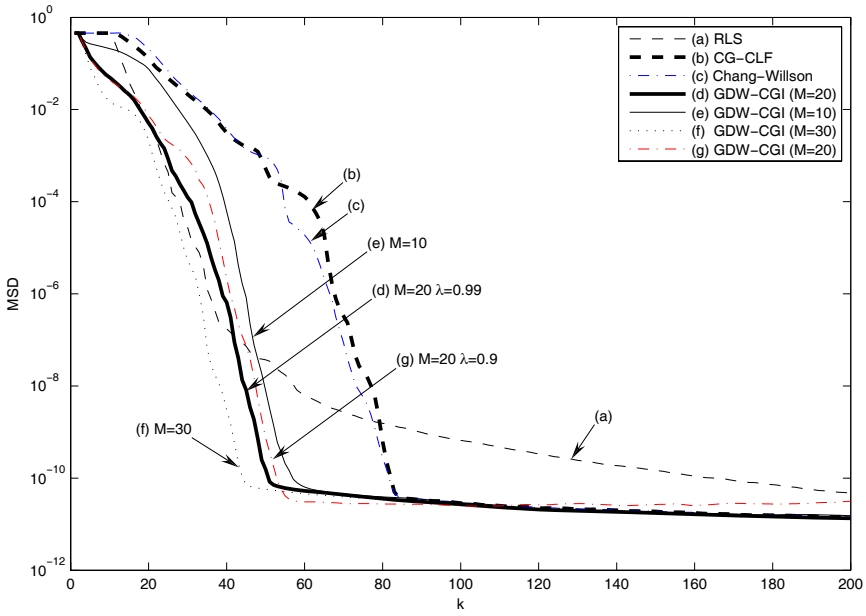

Figure 6.1: Plot of the MSD for RLS, CG-CLF, Chang-Willson, and GDW-CGI ($\lambda = 0.9, 0.99, \lambda_f = \eta = 0.99, \gamma = 0.995$, Input: $\rho = 0$, SNR=100dB, $N = 20$).

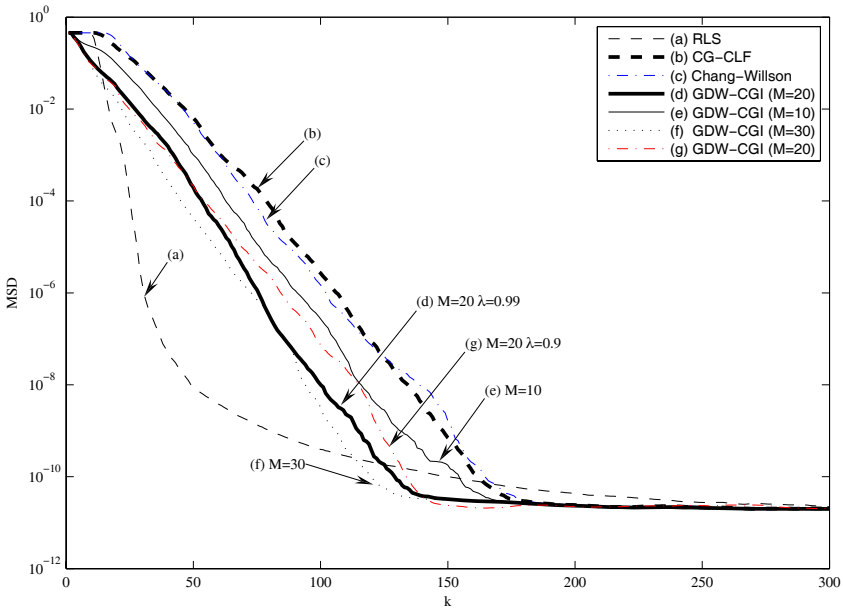

Figure 6.2: Plot of the MSD for RLS, CG-CLF, Chang-Willson, and GDW-CGI ($\lambda = 0.9, 0.99, \lambda_f = \eta = 0.99, \gamma = 0.995$, Input: $\rho = 0.9$, SNR=100dB, $N = 20$).

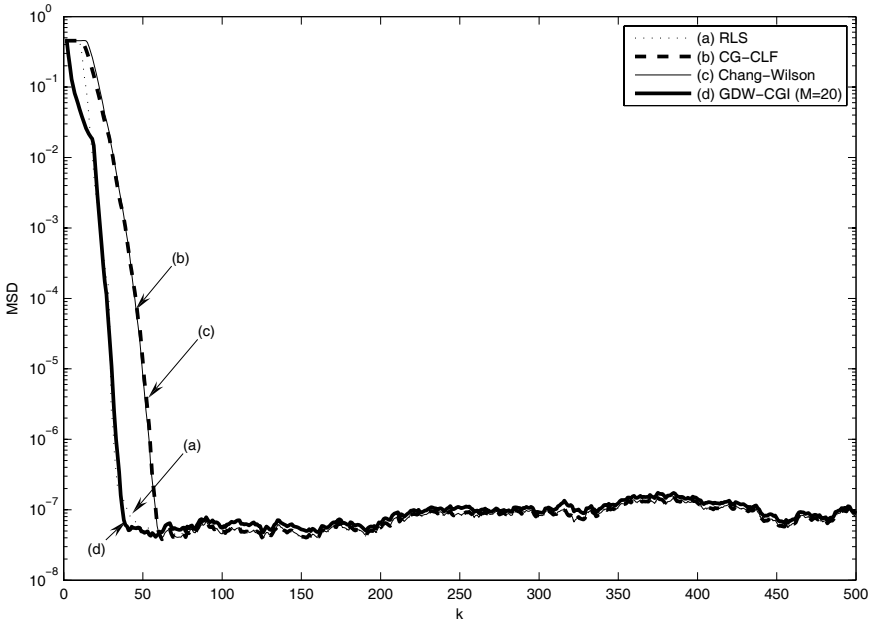

Figure 6.3: Plot of the MSD for RLS, CG-CLF, Chang-Willson, and GDW-CGI in time-varying system ($\lambda = \lambda_f = \eta = 0.99$, $\gamma = 0.995$, Input: $\rho = 0$, SNR=100dB, $N = 20$).

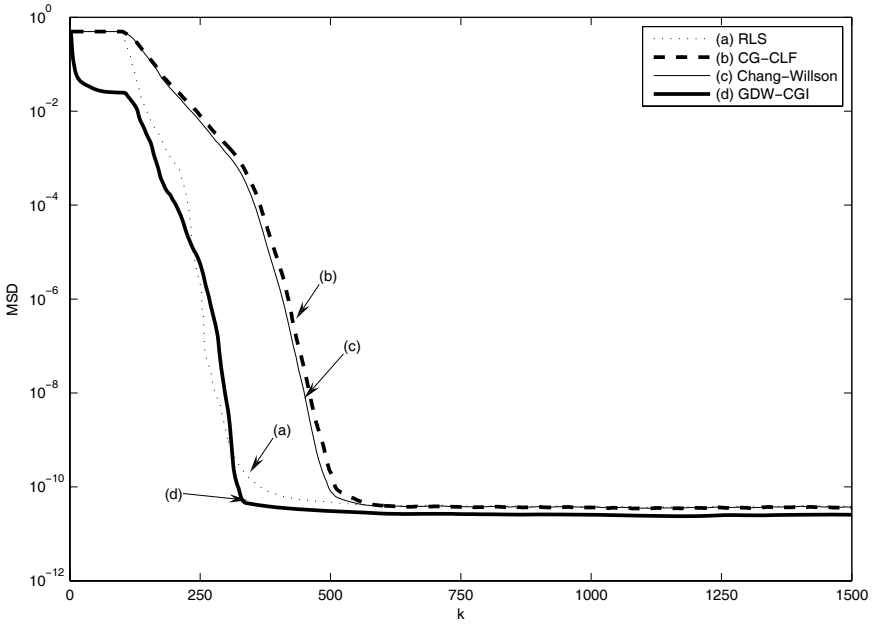

Figure 6.4: Plot of the MSD for RLS, CG-CLF, Chang-Willson, and GDW-CGI ($\lambda = \lambda_f = \eta = 0.99$, $\gamma = 0.995$, Input: $\rho = 0$, SNR=100dB, $N = 200$).

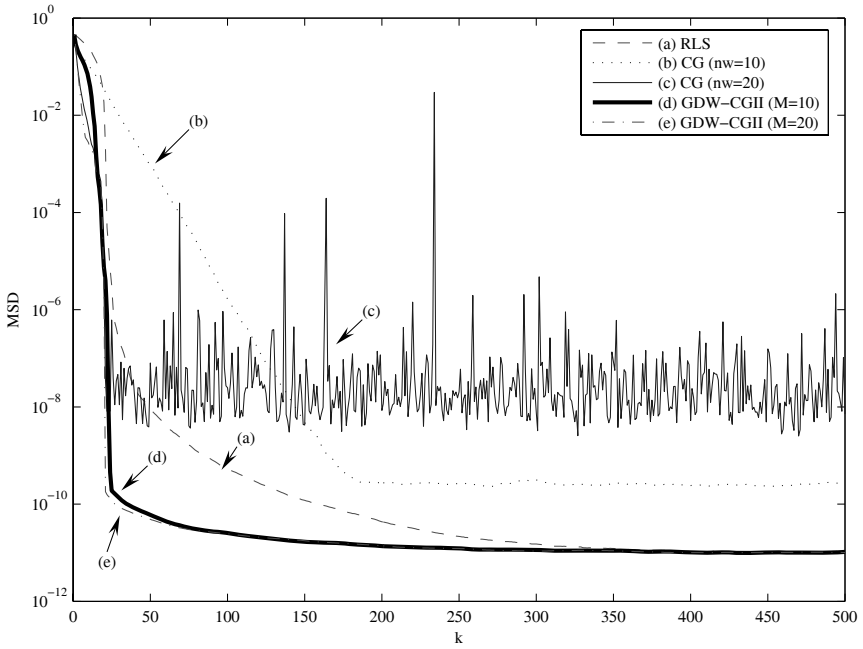

Figure 6.5: Plot of the MSD for RLS, CG, and GDW-CGII ($\lambda = 0.99, \gamma = 0.995$, Input: $\rho = 0$, SNR=100dB, $N = 20$).

compared algorithms have the convergence property independent of the input eigenvalue spread. In Figure 6.7, we show the MSD of the GDW-CGII algorithm as window size varies with $\rho = 0.9$. Observe that the GDW-CGII algorithm has a slow convergence for $M = 1$. For such case, the proposed algorithm is identical to the *steepest descent* method [100]. The convergence has rapid improvements as the window size increases. From the results, we consider that the window size $M = N/2$ is a satisfactory choice in terms of computation load and filter performance.

6.5.2 Application to Adaptive Channel Equalization

The experiment setup of the adaptive channel equalization is described as follows. The input signal applied to the channel is a random Bernoulli sequence r_n=$\pm$1, which has zero-mean and unit variance. The channel is described as [1]

$$h_n = \begin{cases} 0.5\left(1 + \cos\dfrac{2\pi(k-1)}{W}\right), k = 1, 2, 3 \\ 0, \qquad\qquad\qquad\qquad \text{otherwise} \end{cases} \tag{6.32}$$

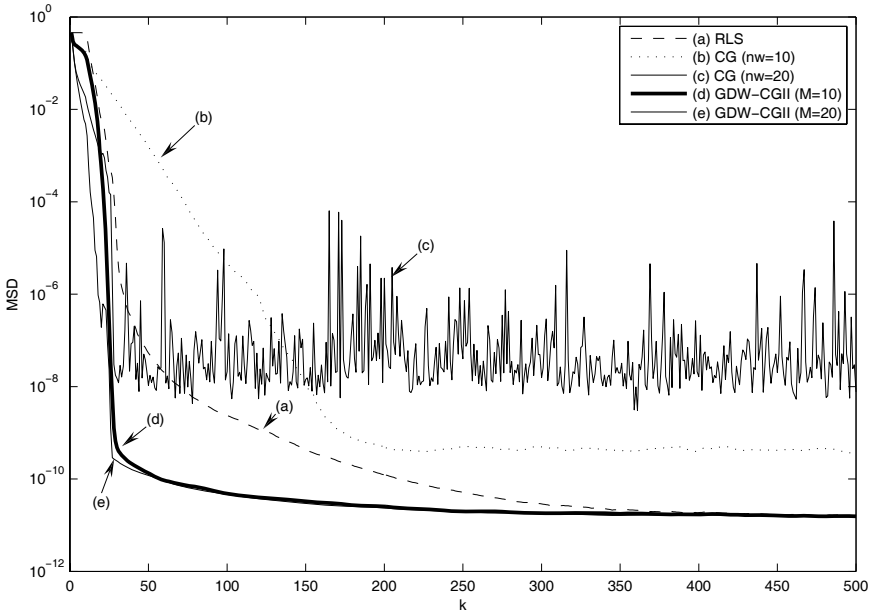

Figure 6.6: Plot of the MSD for RLS, CG, and GDW-CGII with the nonstationary input signal ($\lambda = 0.99, \gamma = 0.995, \sigma = 0.995$, SNR=100dB, $N = 20$).

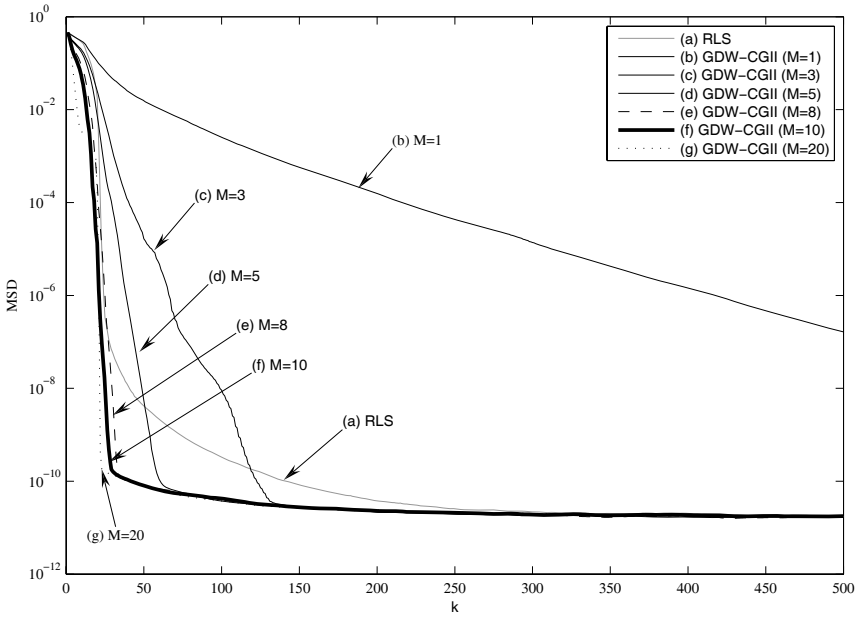

Figure 6.7: Plot of the MSD for RLS and GDW-CGII for different window size ($\lambda = 0.99, \gamma = 0.995$, Input: $\rho = 0.9$, SNR=100dB, $N = 20$).

where the parameter W controls the distortion of the channel. The output signal x_n which can be modeled as $x_n = h_n * r_n + v_n$ served as the tap inputs of the equalizer, where $*$ is convolution symbol and v_n is a corruption white Gaussian noise with zero mean and variance σ_v^2 equal to 0.001. It is known that changing the parameter W in the simulation will affect eigenvalue spread of the correlation matrix of the tap inputs x_n in the equalizer.

In our simulation, the equalizer with 11 taps was employed. The simulations were performed with the parameter W set to 2.9 and 3.5, which implied that the eigenvalue spread equal to 6.0782 and 46.8216, respectively. The parameters were chosen as λ=η=0.99. Figures 6.8 and 6.9 shows the ensemble average MSE behaviors of the proposed GDW-CGI, the CG-CLF algorithm, the Chang-Willson method and the RLS method. The number of reusing data was chosen such that M=5, 11, 15 for the proposed algorithm. In this case, it can be observed that the proposed algorithm has a better transient behavior than the RLS method, and significantly improved over the CG-CLF algorithm and the Chang-Willson method for M=11, 15. For the case of M=5, which produces less computation cost, the proposed algorithm provides a less increase in the convergence, but still better than the CG-CLF and Chang-Willson algorithms. All the compared CG algorithms give MSE comparable with the RLS algorithm in steady state with 1.74×10^{-3} for W=2.9 and 1.4×10^{-2} for W=3.5. From the simulation, we can make a conclusion that when the number of reusing data M increases, the convergence rate of the proposed algorithm converges faster. So it provides the user with flexibility in the choice of the number of reusing data by considering convergence and the added computations. It is observed that choosing M in the range $N/2 \geq M \geq N$ provides reasonable convergence with affordable computational complexity.

6.6 Concluding Remarks

In this chapter, we have presented and analyzed a new approach to the implementation of the CG algorithm for adaptive filtering based on a generalized data windowing scheme which combines the features of the finite sliding window scheme and the exponentially decaying data window scheme. The modified adaptive CG algorithms show improved

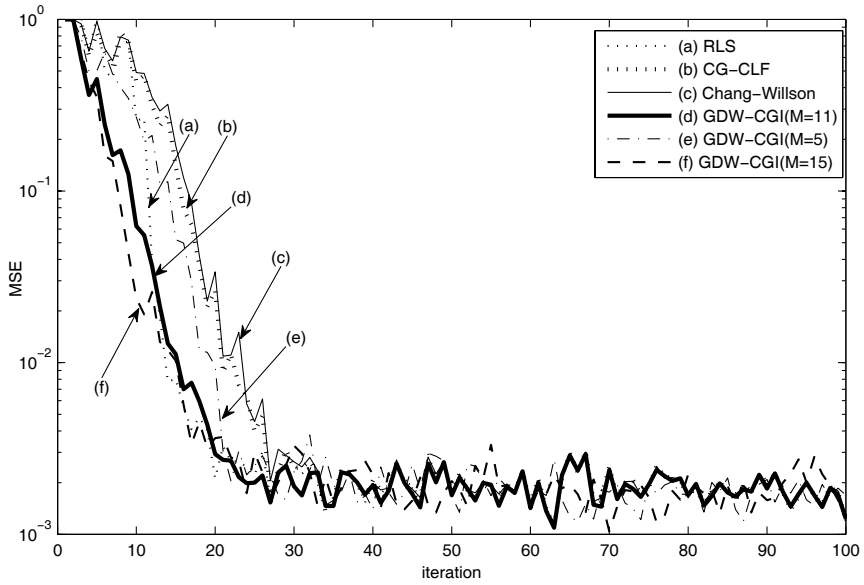

Figure 6.8: Ensemble-average MSE behaviors of various algorithms for adaptive transversal equalizer with W=2.9.

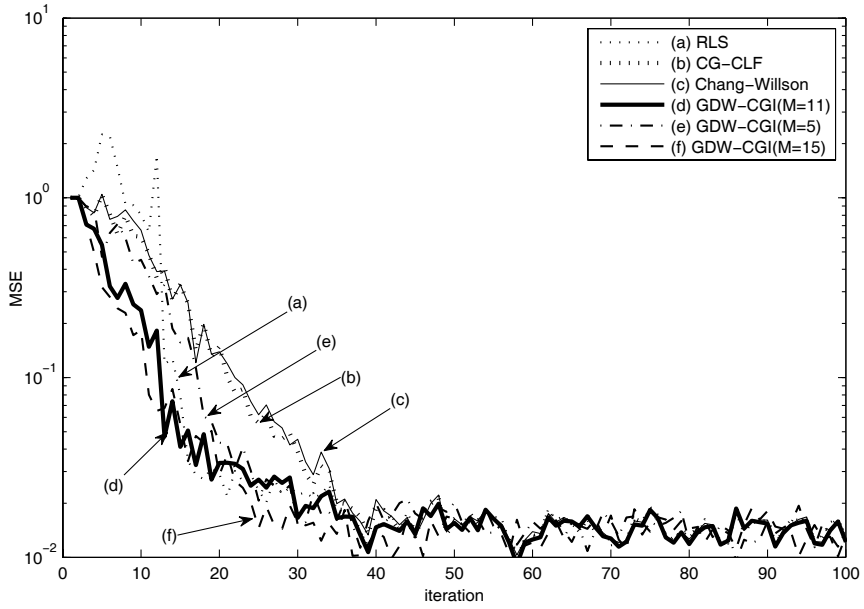

Figure 6.9: Ensemble-average MSE behaviors of various algorithms for adaptive transversal equalizer with W=3.5.

filter performance of accelerated convergence rate and low misadjustment. Besides the application of adaptive system modeling, we also tested the proposed scheme in adaptive channel equalization and prediction, and obtained similar improved performance.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

The problem of optimizing the mapping from input data into some target output has been given a lot of attentions in the literature. Numerous adaptive algorithms have been created based on some criterion functions like the minimum mean-square error and the least-square error. Among them, the LMS algorithm and the RLS algorithm are very popular. However, the LMS algorithm has the problems of slow convergence and the tradeoff between the convergence and steady-state precision. The RLS algorithm has the problems of high computational complexity and numerical instability. These issues have brought in much research effort. To improve the algorithms in terms of computational complexity, numerical robustness, fast convergence and low estimation error, there are still much more work remaining to be done. For instance, the existing variable step-size approaches for the LMS algorithm suffer from either high computational complexity or the power of the measurement noise, the properties of the input and measurement noise in the variable step-size NLMS algorithms are not yet fully explored, little work has been done on mathematically proving why the LMS algorithm with orthogonal transforms and power normalization improve convergence performance for time-correlated inputs such as autoregressive process, and the existing approaches of implementing the CG algorithm using the data windows of exponential form or sliding form result in either loss of convergence or increase in misadjustment.

In this thesis, we first focus on the variable step-size approaches for the LMS algorithm. Since the fixed step-size LMS algorithm can not handle well the tradeoff between the convergence and steady-state mean-square error, a time-varying step-size has been adopted such that large values of the step-size are set at the initial stages for fast convergence and small values of the step-size at the steady-state for lowering the solution fluctuation. It was found that the existing variable step-size adjustment schemes based on the output error or its high statistics introduces either high computational complexity or poor robustness to the power of measurement noise. The work in [67] demonstrated that the VSS algorithm developed by Kwong and Johnston in [57] is probably the best low-complexity variable step-size LMS algorithm available in the literature if well designed, except for its limited robustness to the measurement noise power. The NRVSS algorithm proposed in [65, 66] improved the robustness, but relies on the knowledge of reference signal power. We proposed to use a variable step-size with a quotient form of filtered versions of the error statistics. The filtered estimates of the error are based on exponential windows, applying different decaying factors for the estimations in the numerator and denominator. We demonstrated that the new MRVSS algorithm reduces the sensitivity to the power of measurement noise, improves the steady-state performance for comparable transient behavior and introduces negligible increase in the computational cost. We studied the mean and mean-square convergence, the steady-state performance and the mean step-size behavior of the new algorithm under a slow time-varying system model. These studies could be served as design guidelines for the implementation of the new algorithm in practical applications.

Because the LMS algorithm is sensitive to the scaling of its input when choosing a maximum step-size to guarantee stability, the NLMS algorithm (a normalized version of the LMS algorithm) gives much convenience for setting of the step-size. Although the variable step-size adjustment schemes developed for the LMS algorithms can also be adopted into the NLMS algorithms, the work in [10, 62] shows that the variable step-size adjustment schemes developed from the NLMS algorithm perform more efficiently in terms of convergence and steady-state mean-square error. We followed the work of [10, 62] to derive the optimal step-size which minimizes the mean-square deviation, and we then estimated the optimal step-size under several considerations for the additive

noise and input excitation. A new class of variable step-size NLMS algorithms were proposed. We demonstrated that these new algorithms have simple forms and improved filter performance.

Next, we turned our focus to the transform-domain LMS algorithms. Instead of using a scalar time-varying step-size in the LMS algorithm, the transform-domain LMS algorithms use a fixed, data-independent orthogonal transform like DFT, DCT, DWT, etc to decorrelate the input by merging the power into the main diagonal of input autocorrelation matrix and then apply power normalization to redistribute the power spectral density in order to reduce the eigenvalue spread. In the implementation of the transform-domain LMS algorithms, the power normalization operates on the step-size of the algorithm, which results in a multiple variable step-size approach. The overall structure combining the transformation and power normalized LMS filtering is then referred to as a two-layer linear adaptive structure [104]. In this work, we examined the effects of the transformation and power normalization on the eigenvalues of the input autocorrelation matrix of the transform-domain LMS algorithms up to second-order AR process. The theorem of strong asymptotic equivalence allows us to perform the computations for eigenvalue distributions of the preprocessed input autocorrelation matrix in infinite dimension. We demonstrated that for both first-order and second-order AR input processes, DCT-LMS and DST-LMS outperform DFT-LMS and DHT-LMS. DCT-LMS shows the best convergence performance for lowpass inputs, whereas DST-LMS shows the best convergence performance for all other inputs. We also showed that DFT-LMS and DHT-LMS achieve the same convergence performance. Computer simulation showed similar results for low order MA and ARMA input processes. Experiments show that all these fixed data-independent transforms are not perfect decorrelators. Due to the power leakage in the transform, the preprocessed input autocorrelation matrix is not a diagonal matrix. This limits their performance to catch up the RLS algorithm which uses the inverse matrix of the estimated input autocorrelation matrix. However, the RLS algorithm is computationally intensive and numerically unstable. Although several modifications on the RLS algorithm have been suggested. These fast RLS algorithms, however, still have a tendency to become numerically unstable.

The technique of conjugate gradient optimization algorithm, which uses the conjugate direction method are suitable for efficient implementation without suffering from any

known instability problems. However, the existing sliding data windowing scheme or exponentially decaying data windowing scheme for the CG algorithm suffers from either increase in misadjustment or loss of convergence. In this work, we proposed a generalized data windowing scheme for the CG algorithm which combines the features of both the existing two data windowing schemes. We demonstrated that the modified CG algorithms accelerate the convergence performance and achieve the misadjustment comparable to the RLS algorithm.

Adaptive filters as well as the techniques of adaptation algorithms have been extensively studied and widely used in the literature. By the nature of self-designing, adaptive filters can adjust themselves to different environments. For a specific application, the choice of one adaptive filter or algorithm over another could be determined by one or more of the following factors:

- *Rate of convergence.* This quantity describes the transient behavior of an adaptive algorithm. The adaptive algorithms with fast rate of convergence usually require less training data and number of iterations to converge to the optimum Wiener solution.

- *Misadjustment.* This quantity describes steady-state behavior of an adaptive algorithm. This is a quantitative measure of the amount by which the ensemble averaged final value of the mean-squared error exceeds the minimum mean-squared error produced by the optimal Wiener filter.

- *Computational Requirements.* This is an important parameter from a practical point of view. The parameters of interest include the number of operations required for one complete iteration of the algorithm and the amount of memory needed to store the required data and also the program. These quantities influence the price of the computer needed to implement the adaptive filter.

- *Numerical Robustness.* The implementation of adaptive filtering algorithms on a digital computer, which inevitably operates using finite word-lengths, results in quantization errors. These errors sometimes can cause numerical instability of the adaptive algorithm. An adaptive filtering algorithm is said to be numerically robust when its digital implementation using finite-word-length operations is stable.

Another practical measure of performance is the number of computations needed for the adaptive algorithm to reach steady state. This measure combines the rate of convergence and computational requirements by taking the product of the number iterations needed for the algorithm to converge close enough to the optimum solution and the number of computations needed per iteration. The other factors could also include the tracking capability and structure of information flow in the algorithm. These factors relating to the overall performance of the adaptive algorithms can lead us to the most proper choice.

7.2 Future Work

To conclude, we would like to list a number of areas that are related to the work in this thesis and we believe that further investigation might prove useful and interesting.

- Although the criterion of mean squared error has been widely used in adaptive filtering problems, in many other signal processing fields, such as image signal processing and speech processing, mean squared error has been found to be a poor indication of signal fidelity. Selection of cost function is likely to be application-dependent. For example, for image processing, the desired error measure should probably be based on human perception. For automatic medical image analysis and biometrics, perhaps the final rate of correct pattern recognition is the ultimate goal to optimize [105]. It will be potentially beneficial for applications in these new fields to discover the opportunities and challenges of adaptive filter design when the cost function is different.

- The design of adaptive filters discussed in the thesis is under the assumption that no priori knowledge about the input statistics is available. However, in the real world, people often have some statistics priori knowledge about the signals they are working on. It is usually beneficial to use statistical models that are derived from such priori knowledge, for example, under a Bayesian framework for signal processing and analysis. Further investigations may be made for how to incorporate such priori knowledge into the design of adaptive filters.

- The effectiveness and advantages of the transform-domain LMS algorithms over its countparts in time domain are already investigated extensively in [86, 90] and in Chapter 5 of this thesis. Specifically, the effectiveness and advantages of the LMS algorithm with transforms of DFT, DHT, DCT and DST are shown to be dependent on the power spectral density of the second-order autoregressive input process. In addition, they are often related to the statistical nature of the input signal. For example, Fourier-types of transforms can improve energy compaction of certain types of signals (for example, natural image signals), but may not have the same effect on other types of signals (for example, signals that have a flat spectrum, e.g., spike trains in neuronal signals). From such point of view, it will be beneficial to discuss further how the transform-domain adaptive filter adapts to the nature of the input signals? Why transform-domain method is better than original domain method in terms of such adaption? How is it related the power normalization process? What is the difference between these orthogonal transforms in terms of capturing the nature of the input signals? Generating the eigenvalue results obtained in Chapter 5 to higher order inputs of AR, MA and ARMA processes as well as to the DWT, WHT and wavelet-based transforms is also interesting for future work.

- It is noted that the transforms used for the LMS adaptive filter in the literature are orthogonal. However, no theoretical analysis shows the importance of such orthogonality of the transforms to the LMS algorithm. Is it possible to apply non-orthogonal transforms into the LMS algorithm? Note that sacrificing orthogonality can often lead to gains in other aspects, for example, sparseness, translation-invariance, etc. Some tight frames are also energy preserving (in mean-squared sense), but have better decorrelation power than the orthogonal transforms. It will be interesting to explore any performance improvement of using this kind of transforms for adaptive filters. Further theoretical analysis is also desirable.

- Although the time-domain LMS algorithm is capable of tracking the statistic changes of the input process, the statistic changes should be very slow due to the tracking ability of the LMS algorithm. On the other hand, the algorithms developed based

on Kalman filtering theory have faster tracking capabilities due to their design in nature. They are, however, much more complicated for the usage in some real applications such as acoustic echo cancellation. Does the transform-domain LMS algorithms improve the tracking performance? Further investigations are desirable.

- This thesis studied the applications of the proposed adaptive filters in system identification, noise cancellation and channel equalization. Potential applications could also include autoregressive spectral analysis, adaptive line enhancement, speech coding, acoustic echo cancellation, adaptive beamforming, time delay estimation, etc. Exploring the usage of the proposed adaptive filters in any of the potential applications will be beneficial.

List of Author's Other Publications during Ph.D. Candidature

[1] S. Khoo, Z. Man, and S. Zhao, "Comments on 'Adaptive multiple-surface sliding control for non-autonomous systems with mismatched uncertainties' ", Automatica, vol. 44, no. 11, pp. 2995–2998, November 2008.

[2] S. Khoo, Z. Man, and S. Zhao, "Sliding Mode Control for A Class of Complex Nonlinear Systems with T-S Fuzzy models," accepted for publication in WSEAS Transactions on Systems, 2008.

[3] S. Khoo, Z. Man, and S. Zhao, "Sliding Mode Control for T-S Fuzzy Systems," International Journal of Systems and Control, vol. 1, pp. 22–30, 2007.

[4] Suiyang Khoo, Lihua Xie, Zhihong Man and Shengkui Zhao,"Observer-based robust finite-time cooperative consensus control for multi-agent networks", Proceedings of the 4th IEEE Conference on Industrial Electronics and Applications, Xi'an of China, pp. 1883–1888, May 2009.

[5] Khoo Suiyang, Man Zhihong and Zhao Shengkui, "Adaptive Fast Finite Time Control of A Class of Nonlinear Uncertain Systems", Proceedings of IEEE 3rd International Conference on Industrial Electronics and Applications, Singapore, pp. 2167–2172, June 2008.

[6] Khoo Suiyang, Man Zhihong and Zhao Shengkui, "Terminal sliding mode control for MIMO T-S fuzzy systems", Proceedings of the 6th International Conference on

Information, Communications, Signal Processing, Singapore, pp. 1–5, December 2007.

[7] Khoo Suiyang, Man Zhihong and Zhao Shengkui, "Adaptive Fuzzy Multi-surface Sliding Mode Control for A Class of Nonlinear Systems", Proceedings of the 2nd IEEE Conference on Industrial Electronics and Applications, Harbin of China, pp. 2340–2345, May 2007.

[8] Khoo Suiyang, Man Zhihong and Zhao Shengkui, "Sliding Mode Control of Fuzzy Dynamic Systems", Proceedings of the 9th International Conference on Control, Automation, Robotics and Vision, Singapore, pp. 1–6, December 2006.

References

[1] S. Haykin, *Adaptive Filter Theory.* 4th ed. Englewood Cliffs, NJ: Prentice Hall, 2002.

[2] A. A. Beex and J. R. Zeidler, "Interference suppression with minimal signal distortion," in *Proceedings of the Intl Conf. on Acoustics, Speech and Signal Processing*, vol. 6, (Hong Kong), pp. 225–228, Apr. 2003.

[3] S. M. Kuo and D. R. Morgan, "Active noise control: A tutorial review," *Proc. of the IEEE*, vol. 87, pp. 943–973, June 1999.

[4] B. Widrow, J. R. G. Jr., J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, E. D. Jr., and R. C. Goodlin, "Adaptive noise cancelling: Principles and applications," *Proc. of the IEEE*, vol. 63, pp. 1692–1716, Dec. 1975.

[5] S. Y. Choi and T. W. Lee, "A negentropy minimization approach to adaptive equalization for digital communication systems," *IEEE Trans. on Neural Networks*, vol. 15, pp. 928–936, July 2004.

[6] R. D. Gitlin, E. Y. Ho, and J. E. Mazo, "Passband equalization of differentially phasemodulated data signals," *Bell Syst. Technical Journal*, vol. 52, pp. 219–238, Feb. 1973.

[7] J. G. Proakis, *Digital Communications*, ch. 11. 4th ed. New York: McGraw-Hill, 2001.

[8] S. Qureshi, "Adaptive equalization (data transmission)," *IEEE Communication Magazine*, vol. 20, pp. 9–16, Mar. 1982.

[9] A. I. Sulyman and A. Zerguine, "Convergence and steady-state analysis of a variable step-size NLMS algorithm," *Signal Processing*, vol. 83, pp. 1255–1273, June 2003.

[10] H. C. Shin, A. H. Sayed, and W. J. Song, "Variable step-size NLMS and affine projection algorithms," *IEEE Signal Processing Letters*, vol. 11, pp. 132–135, Feb. 2004.

[11] B. Farhang-Boroujeny, "On statistical efficiency of the LMS algorithm in system modeling," *IEEE Trans. on Signal Processing*, vol. 41, pp. 1947–1951, May 1993.

[12] M. Mboup, M. Bonnet, and N. Bershad, "LMS coupled adaptive prediction and system identification: A statistical model and transient mean analysis," *IEEE Trans. on Signal Processing*, vol. 42, pp. 2607–2615, Oct. 1994.

[13] S. Theodoridis, "Adaptive filtering algorithms," in *Proceedings of the 18th IEEE Conf. on Instrumentation and Measurement Technology*, (Budapest, Hungary), pp. 1497–1501, May 2001.

[14] B. Widrow, *"Adaptive filters," in Aspects of Network and System Theory.* R. Kalman and N. DeClaris, Eds. New York: Holt, Rinehart, and Winston, 1971.

[15] V. Stewart, C. F. N. Cowan, and S. Sezer, "Adaptive echo cancellation for packet-based networks," *Lecture Notes for Computer Science*, vol. 3124, pp. 516–525, 2004.

[16] D. L. Duttweiler, "A twelve-channel digital echo canceler," *IEEE Trans. on Communications*, vol. 26, pp. 647–653, May 1978.

[17] K. C. Ho, "Performance of multiple LMS adaptive filters in tandem," *IEEE Trans. on Signal Processing*, vol. 49, pp. 2762–2773, Nov. 2001.

[18] V. G. Koll and S. B. Weinstein, "Simultaneous two-way data transmission over a two-wire circuit," *IEEE Trans. on Communications*, vol. 21, pp. 143–147, Feb. 1973.

[19] S. Affes, S. Gazor, and Y. Grenier, "An algorithm for multisource beamforming and multitarget tracking," *IEEE Trans. on Signal Processing*, vol. 44, pp. 1512–1522, June 1996.

[20] L. C. Godara, "Improved lms algorithm for adaptive beamforming," *IEEE Trans. on Antennas and Propagation*, vol. 38, pp. 1631–1635, Oct. 1990.

[21] C. C. Ko, "A simple, fast adaptive algorithm for broad-band null steering arrays," *IEEE Trans. on Antennas and Propagation*, vol. 39, pp. 122–125, Jan. 1991.

[22] A. Massa, M. Donelli, F. G. B. D. Natale, S. Caorsi, and A. Lommi, "Planar antenna array control with genetic algorithms and adaptive array theory," *IEEE Trans. on Antennas and Propagation*, vol. 52, pp. 2919–2924, Nov. 2004.

[23] B. Widrow, P. Mantey, L. Griffiths, and B. Goode, "Adaptive antenna systems," *Proc. of the IEEE*, vol. 55, pp. 2143–2159, Dec. 1967.

[24] N. J. Bershad and O. M. Macchi, "Adaptive recovery of a chirped sinusoid in noise, Pt. II: Performance of the LMS algorithm," *IEEE Trans. on Signal Processing*, vol. 39, pp. 595–602, Mar. 1991.

[25] R. L. Campbell, N. H. Younan, and J. Gu, "Performance analysis of the adaptive line enhancer with multiple sinusoids in noisy environment," *Signal Processing*, vol. 82, pp. 93–101, Jan. 2002.

[26] M. Ghogho, M. Ibnkahla, and N. J. Bershad, "Analytic behavior of the LMS adaptive line enhancer for sinusoids corrupted by multiplicative and additive noise," *IEEE Trans. on Signal Processing*, vol. 46, pp. 2386–2393, Sept. 1998.

[27] J. T. Rickard, J. R. Zeidler, M. J. Dentino, and M. Shensa, "A performance analysis of adaptive line enhancer-augmented spectral detectors," *IEEE Trans. on Circuits and Systems*, vol. 28, pp. 534–541, June 1981.

[28] J. R. Treichler, "Transient and convergent behavior of the adaptive line enhancer," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 27, pp. 53–62, Feb. 1979.

[29] J. R. Zeidler, E. H. Satorius, D. M. Chabries, and H. T. Wexler, "Adaptive enhancement of multiple sinusoids in uncorrelated noise," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 26, pp. 53–62, June 1978.

[30] G. L. Plett, "Adaptive inverse control of unmodeled stable SISO and MIMO linear systems," *Intl J. on Adaptive, Control and Signal Processing*, vol. 16, pp. 243–272, May 2002.

[31] G. L. Plett, "Adaptive inverse control of linear and nonlinear systems using dynamic neural networks," *IEEE Trans. on Neural Networks*, vol. 14, pp. 360–376, Mar. 2003.

[32] M. Shafiq, "Adaptive inverse control: internal model control structure," *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 87, no. 8, pp. 2164–2167, 2004.

[33] B. Widrow and E. Walach, *Adaptive Inverse Control*. Upper Saddle River, NJ: Prentice-Hall, 1996.

[34] B. Widrow and M. E. Hoff,Jr., "Adaptive switching circuits," tech. rep., Stanford Electron. Labs., Stanford, CA, June 1960.

[35] M. Dentino, J. McCool, and B. Widrow, "Adaptive filtering in the frequency domain," *Proc. of the IEEE*, vol. 66, pp. 1658–1659, Dec. 1978.

[36] G. A. Clark, S. K. Mitra, and S. R. Parker, "Block implementation of adaptive digital filters," *IEEE Trans. on Circuits and Systems*, vol. 28, pp. 584–592, June 1981.

[37] R. M. Gray and L. D. Davisson, *Random Process: A Mathematical Approach for Engineers*. Prentice-Hall, Englewood Cliffs, NJ, 1986.

[38] B. Farhang-Boroujeny, *Adaptive Filter: Theory and Applications*. Chichester, UK: John Wiley & Sons, 1998.

[39] B. Widrow and S. D. Steam, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

[40] A. Feuer and E. Weinstein, "Convergence analysis of LMS filters with uncorrelated Gaussian data," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 33, pp. 222–230, Feb. 1985.

[41] R. R. Bitmead, "Convergence in distribution of LMS-type adaptive parameter studies," *IEEE Trans. on Automatic Control*, vol. 28, pp. 524–535, 1983.

[42] R. R. Bitmead, "Convergence properties of LMS adaptive estimators with unbounded dependent inputs," *IEEE Trans. on Automatic Control*, vol. 29, pp. 477–479, 1984.

[43] L. Guo, L. Ljung, and G. J. Wang, "Necessary and sufficient conditions for stability of LMS," *IEEE Trans. on Automatic Control*, vol. 42, pp. 761–770, 1997.

[44] V. Solo, "The stability of LMS," *IEEE Trans. on Signal Processing*, vol. 45, pp. 3017–3026, 1997.

[45] H. J. Butterweck, "A wave theory of long adaptive filters," *IEEE Trans. on Circuits and Systems, Part I, Fundamentals, Theory and Applications*, vol. 48, pp. 739–747, June 2001.

[46] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Process*. 4th ed. New York: McGraw-Hill, 2002.

[47] J. M. Cioffi, "Limited-precision effects in adaptive filtering," *IEEE Trans. on Circuits and Systems*, vol. 34, pp. 821–833, July 1987.

[48] B. Hassibi, A. H. Sayed, and T. Kailath, "H^∞ optimality of the LMS algorithm," *IEEE Trans. on Signal Processing*, vol. 44, pp. 267–280, Feb. 1996.

[49] M. Kamenetsky, *Accelerating the convergence of the LMS adaptive algorithm*. PhD thesis, Standford University, June 2005.

[50] B. Widrow and E. Walach, "On the statistical efficiency of the LMS algorithm with nonstationary inputs," *IEEE Trans. on Information Theory*, vol. IT-30, pp. 211–221, Mar. 1984.

[51] T. Kailath, *Linear Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1980.

[52] D. T. L. Lee, M. Morf, and B. Friedlander, "Recursive least-squares ladder estimation algorithms," *IEEE Trans. on Circuits and Systems*, vol. 28, pp. 467–481, June 1981.

[53] F. Ling, D. Manolakis, and J. G. Proakis, "Numerically robust least-squares lattice-ladder algorithm with direct updating of the reflection coefficients," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 34, pp. 837–845, Aug. 1986.

[54] A. H. Sayed, *Fundamentals of Adaptive Filtering.* New York: Wiley, 2003.

[55] R. Harris, D. Chabries, and F. Bishop, "A variable step (VS) adaptive filter algorithm," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 34, pp. 309–316, Apr. 1986.

[56] V. J. Mathews and Z. Xie, "A stochastic gradient adaptive filter with gradient adaptive step size," *IEEE Trans. on Signal Processing*, vol. 41, pp. 2075–2087, June 1993.

[57] R. Kwong and E. W. Johnston, "A variable step size LMS algorithm," *IEEE Trans. on Signal Processing*, vol. 40, pp. 1633–1642, July 1992.

[58] T. Aboulnasr and K. Mayyas, "A robust variable step-size LMS-type algorithm: Analysis and simulations," *IEEE Trans. on Signal Processing*, vol. 45, pp. 631–639, Mar. 1997.

[59] D. I. Pazaitis and A. G. Constantinides, "A novel kurtosis driven variable step-size adaptive algorithm," *IEEE Trans. on Signal Processing*, vol. 47, pp. 864–872, Mar. 1999.

[60] W. P. Ang and B. Farhang-Boroujeny, "A new class of gradient adaptive step-size LMS algorithms," *IEEE Trans. on Signal Processing*, vol. 49, pp. 805–810, Apr. 2001.

[61] T. I. Haweel, "A simple variable step size LMS adaptive algorithm," *Int. J. Circuit Theory*, vol. 32, pp. 523–536, Nov. 2004.

[62] A. Mader, H. Puder, and G. U. Schmidt, "Step-size control for acoustics echo cancellation filters-An overview," *Signal Processing*, vol. 80, pp. 1697–1719, 2000.

[63] S. Koike, "A class of adaptive step-size control algorithms for adaptive filters," *IEEE Trans. on Signal Processing*, vol. 50, pp. 1315–1326, June 2002.

[64] J. Okello and et al., "A new modified variable step-size for the LMS algorithm," *Proceedings of the Intl Sym. on Circuits and Systems*, pp. 170–173, 1998.

[65] M. H. Costa and J. C. M. Bermudez, "A robust variable step-size algorithm for lms adaptive filters," *Proceedings of the Intl Conf. on Acoustics, Speech and Signal Processing*, pp. 93–96, 2006.

[66] M. H. Costa and J. C. M. Bermudez, "A noise resilient variable step-size LMS algorithm," *Signal Processing*, vol. 88, pp. 733–748, Mar. 2008.

[67] C. G. Lopes and J. C. M. Bermudez, "Evaluation and design of variable step-size adaptive algorithms," *Proceedings of the Intl Conf. on Acoustics, Speech and Signal Processing*, pp. 3845–3848, 2001.

[68] B. Widrow and J. M. McCool, "Stationary and nonstationary leaming characteristics of the LMS filter," *Proc. of the IEEE*, vol. 64, pp. 1151–1162, Aug. 1976.

[69] R. R. Bitmead and B. D. O. Anderson, "Adaptive frequency sampling filters," *IEEE Trans. on Circuits and Systems*, vol. 28, pp. 524–534, June 1981.

[70] F. A. Reed and P. L. Feintuch, "A comparison of LMS adaptive cancelleres implemented in the fequency domain and the time domain," *IEEE Trans. on Circuits and Systems*, vol. 28, pp. 610–615, June 1981.

[71] A. O. Ogunfunmi and A. M. Peterson, "On the implementation of the frequency-domain LMS adaptive filter," *IEEE Trans. on Circuits and Systems, Part II*, vol. 39, pp. 318–322, May 1992.

[72] S. S. Narayan, A. M. Peterson, and M. J. Narashima, "Transform domain LMS algorithm," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 31, pp. 609–615, June 1983.

[73] D. F. Marshal, W. K. Jenkins, and J. J. Murphy, "The use of orthogonal transforms for improving performance of adaptive filters," *IEEE Trans. on Circuits Systems*, vol. 36, pp. 474–484, Apr. 1989.

[74] K. Mayyas and T. Aboulnasr, "Reduced-complexity transform-domain adaptive algorithm with selective coefficient update," *IEEE Trans. on Circuits and Systems, Part II*, vol. 51, pp. 136–142, Mar. 2004.

[75] S. Attallah and S. W. Liaw, "Analysis of DCTLMS algorithm with a selective coefficient updating," *IEEE Trans. on Circuits and Systems, Part II*, vol. 48, pp. 628–632, June 2001.

[76] R. C. Bilcu, P. Kuosmanen, and K. Egiazarian, "A transform domain LMS adaptive filter with variable step-size," *IEEE Signal Processing Letters*, vol. 9, pp. 51–53, Feb. 2002.

[77] S. Hosur and A. H. Tewfik, "Wavelet transform domain LMS algorithm," *Proceedings of the IEEE Intl Conf. on Acoustics, Speech and Signal Processing*, vol. 3, pp. 508–510, Apr. 1993.

[78] M. Doroslovacki and H. Fan, "Wavelet-based adaptive filtering," *Proceedings of the IEEE Intl Conf. on Acoustics, Speech and Signal Processing*, vol. 3, pp. 488–491, Apr. 1993.

[79] N. Erdol and F. Basbug, "Wavelet transform based adaptive filters: analysis and new results," *IEEE Trans. on Signal Processing*, vol. 44, pp. 2163–2171, Sept. 1996.

[80] S. Hosur and A. H. Tewfik, "Wavelet transform domain adaptive FIR filtering," *IEEE Trans. on Signal Processing*, vol. 45, pp. 617–630, Mar. 1997.

[81] K. Mayyas, "New transform-domain adaptive algorithms for acoustics echo cancellation," *Digital Signal Processing*, vol. 13, pp. 415–432, July 2003.

[82] K. Doğançay, "Complexity considerations for transform-domain adaptive filters," *Signal Processing*, vol. 83, pp. 1177–1192, June 2002.

[83] B. Widrow, P. Baudrenghien, M. Vetterli, and P. F. Titchener, "Fundamental relations between the LMS algorithm and the DFT," *IEEE Trans. on Circuits and Systems*, vol. 34, pp. 814–819, July 1987.

[84] J. C. Liu and T. P. Lin, "LMS-based DHT analyzer," *Electronic Letters*, vol. 24, pp. 483–485, Apr. 1988.

[85] S. S. Wang, "LMS algorithm and discrete orthogonal transforms," *IEEE Trans. on Circuits and Systems*, vol. 38, pp. 949–951, Aug. 1991.

[86] F. Beaufays and B. Widrow, "On the advantages of the LMS spectrum analyzer over non-adaptive implementations of the sliding-DFT," *IEEE Trans. on Circuits and Systems*, vol. 42, pp. 218–220, Apr. 1995.

[87] D. I. Kim and P. D. Wilde, "Performance analysis of the DCT-LMS adaptive filtering algorithm," *Signal Processing*, vol. 80, pp. 1629–1654, Aug. 2000.

[88] F. Beaufays, "Transform-domain adaptive filters: An analytical approach," *IEEE Trans. on Signal Processing*, vol. 43, pp. 422–431, Feb. 1995.

[89] F. Beaufays, *Two-layer linear structures for fast adaptive filtering.* PhD thesis, Standford University, June 1995.

[90] C. L. Jae and K. U. Chong, "Performance of transform-domain LMS adaptive digital filters," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 34, pp. 499–510, June 1986.

[91] R. M. Gray, "Toeplitz and circulant matrices: a review," tech. rep., Aug. 2002.

[92] G. H. Golub and C. F. V. Loan, *Matrix Computations.* 2nd ed, Baltimore: Johns Hopkins University Press, 1989.

[93] R. M. Reid, "Some eigenvalue properties of persymmetric matrices," *SIAM Rev.*, vol. 39, pp. 313–316, June 1997.

[94] G. K. Boray and M. D. Srinath, "Conjugate gradient techniques for adaptive filtering," *IEEE Trans. Circuits and Systems, Part I, Fundamentals, Theory and Applications*, vol. 39, pp. 1–10, Jan. 1992.

[95] Y. Saad, *Iterative Methods for Sparse Linear Systems.* Boston, MA: PWS-Kent, 1996.

[96] D. G. Luenberger, *Linear and Nonlinear Programming.* 2nd ed. Reading, MA: Addison-Wesley, 1984.

[97] J. A. Apolinário, M. L. R. de Campos, and C. P. B. O., "The constrained conjugate gradient algorithm," *IEEE Signal Processing Letters*, vol. 7, pp. 351–354, Apr. 2000.

[98] T. Bose and M. Q. Chen, "Conjugate gradient method in adaptive bilinear filtering," *IEEE Trans. on Signal Processing*, vol. 43, pp. 1503–1508, Jan. 1995.

[99] P. S. Chang and A. N. Willson, "Analysis of conjugate gradient algorithms for adaptive filtering," *IEEE Trans. on Signal Processing*, vol. 48, pp. 409–418, Feb. 2000.

[100] O. Diene and A. Bhaya, "Adaptive filtering algorithms designed using control Liapunov functions," *IEEE Signal Processing Letters*, vol. 13, pp. 224–227, Apr. 2006.

[101] K. S. Joo, T. Bose, and G. F. Xu, "Image restoration using a conjugate based algorithm," *J. Circuits, Systems and Signal Processing*, vol. 16, no. 2, pp. 197–206, 1997.

[102] S. C. Douglas and J. K. Soh, "A numerically-stable sliding-window estimator and its application to adaptive filters," *Proceedings of the 31st Asilomar Conf. on Signals, Systems and Computers*, Nov. 1997.

[103] J. K. Soh and S. C. Douglas, "A sliding-window-covariance fast transversal filter employing periodic leakage," *Proceedings of the 8th IEEE Digital Signal Processing Workshop*, Aug. 1998.

[104] F. Beaufays and B. Widrow, "Two-layer linear structures for fast adaptive filtering," *Proceedings of the World Congress on Neural Networks*, vol. 3, pp. 87–93, June 1994.

[105] W. Zhou and A. C. Bovik, "Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures," *IEEE Signal Proc. Mag.*, vol. 26, pp. 98–117, Jan. 2009.