THE LEAKY LEAST MEAN MIXED NORM ALGORITHM

BY

MOHAMMED ABDUL NASAR

A Thesis Presented to the

DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

ELECTRICAL ENGINEERING

JUNE, 2010

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **MOHAMMED ABDUL NASAR** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN ELECTRICAL ENGINEERING**.

Thesis Committee

Dr. Azeddine Zerguine (Adviser)

Dr. Tareq Y. Al Naffouri (Member)

Dr. Samir Abdul-Jauwad (Member)

Dr. Samir Abdul-Jauwad
Department Chairman

Dr. Salam Zummo
Dean of Graduate Studies

15/12/10

Date

Dedicated to my loving Mother & Father

ACKNOWLEDGMENTS

In the name of Allah, the Most Beneficent Most Merciful

All praise is due to Allah, the source of all knowledge and strength. I acknowledge His infinite mercy and grace in making this work a success. And may His peace and blessings be upon his final messenger Muhammad, a guidance and inspiration to our lives. The successful completion of this work was made possible by a number of major contributions from different persons and organizations alike, to whom I wish to express my due gratitude.

I am grateful to King Fahd University of Petroleum & Minerals for giving me the opportunity to carry out this task under the guidance of a scholarly faculty and a proper research environment. My sincere gratitude and thanks goes to my thesis advisor Dr. Azeddine Zerguine for his meticulous attention, his guidance, and his patience with me. I am thankful to him for the multitudes of favors I have taken from him, and for all the things he has taught me. I would also like to extend my appreciation to my thesis committee members Dr. Samir Abdul-Jauwad and Dr. Tareq Y. Al-Naffouri for their helpful insights and advice.

Acknowledgment is due to Mr. Ubaid and Mr. Moinuddin for long hours of discussions on my thesis, and for their countless favors ever since I entered

KFUPM.

I am grateful to all my friends I have made during my time here. I will never forget the time we spent together. We certainly enjoyed doing our thesis during the same time, sharing long hours of frustration and hardwork together, spending long nights keeping awake for each other just so that thesis simulations would seem less difficult.

My heartfelt gratitude goes to my parents and my family. I would never have been able to pursue this task without their cooperation and understanding. I am eternally indebted to my father and mother for all their prayers, concern, love and understanding.

TABLE OF CONTENTS

LIST OF FIGURES

THESIS ABSTRACT

NAME:	Mohammed Abdul Nasar
TITLE OF STUDY:	The Leaky Least Mean Mixed Norm Algorithm
MAJOR FIELD:	Electrical Engineering
DATE OF DEGREE:	MAY 2010

The least-mean square (LMS)and the least-mean fourth (LMF)are the two important adaptive schemes. They have several advantages and disadvantages, these are combined together in one which is named as the least-mean mixed norm (LMMN) in order to utilize the benefits of both the algorithms in which mixing parameter is fixed. The aim of this thesis is to derive the Leaky Least Mean Mixed Norm algorithm and asses its performance using the energy conservation concept. The performance evaluation includes the steady state, tracking and the transient analysis of the proposed algorithmand then a new weighted sum of LMS and LMF has been proposed in which the mixing parameter is time varying and has the ability to adapt the variations in the environment. A number of simulations will be carried out to experimentally verify the theoretical findings.

خلاصة الرسالة

الاسم الكامل: محمد عبد الناصر.

عنوان الرسالة: خوارزمية (Leaky Least Mean Mixed Norm).

التخصص: الهندسة الكهربائية.

تاريخ الشهادة: يونيو 2010

(Least Mean Square) و (Least Mean Fourth) عبارة عن نوعين مهميين من النظام المتكيف (adaptive scheme). كلا النظامين لهما ميزات كثيرة ومساوئ أيضا وعند دمج النظامين في نظام واحد (Least Mean Mixed Norm) سيستفاد من الايجابيات لكلا النظامين ودمجهما في النظام المقترح .إن هدف هذه الرسالة هو اشتقاق هذه خوارزمية (Leaky Least Mean Mixed Norm) ومن ثم تقييم أدائها باستخدام مبدأ حفظ الطاقة. تقييم الأداء لهذه الخوارزمية المشتقة سيشمل الحالة المستقرة (steady state) والتتبع والحالة العابرة (transient state) لهذه الخوارزمية المقترحة ومن ثم سيكون هناك مجموع للنظامين (LMS) و (LMF) ليكون لدينا نظام (LMMN) يستطيع التكيف مع التغير في المحيط المجاور. هناك أيضا عدد من المحاكاة لمحاولة إثبات ما توصل إليه نظريا.

Nomenclature

Notations

$\mathbf{u}_n$	:	Input regression vector at time n
$\mathbf{R}$	:	Regressor covariance matrix
$\mathbf{w}_\mathrm{o}$	:	Unknown system weight vector
$\mathbf{w}_n$	:	Adaptive filter weight vector
v_n	:	Noise at time n
σ_u^2	:	Variance of input regressors
σ_v^2	:	Variance of noise
ξ_n^k	:	k^{th} order moment of noise
M	:	Filter length
d_n	:	Output of unknown system at time n
e_n	:	Output error for adaptive system at time n
J	:	Cost function
μ	:	Step size of adaptive algorithm
α	:	Leakage Factor
$\mathbf{w}_k$	:	Weight error vector at time k
e_{an}	:	a priori output estimation error at time n
$e_{an}^{\mathbf{A}}$	:	weighted a priori output estimation error at time n for weighting matrix $\mathbf{A}$
$\mathbf{T}$	:	Companion Form Matrix of $\mathbf{R}$
$\mathbf{I}$	:	Identity matrix

Abbreviations

LMS	:	Least Mean Square
LMF	:	Least Mean Fourth
LMMN	:	Least Mean Mixed Norm
LLMMN	:	Leaky Least Mean Mixed Norm
VLLMMN	:	Variable Leaky Least Mean Mixed Norm
MSE	:	Mean Squared Error
EMSE	:	Excess Mean Squared Error
MSD	:	Mean Square Deviation

CHAPTER 1

INTRODUCTION

Adaptive systems play a vital role in the development of modern communications. Whenever there is a requirement of processing a signal that results from an unknown statistics, then adaptive filter proves out to be a best solution to the problem. Because of their ability to achieve high efficiency and high reliability they are widely used in variety of applications such as equalization [1], noise cancelation, linear prediction [2] and in system identification[3], [4], just to name a few.

The theory, benefits and applications of adaptive filters have been widely described in literature (see [3], [4] and references therein). We will go into more detail into the aforementioned applications of adaptive filtering in the next section.

The most important motivation for the development of adaptive filter theory has been the tracking of changes in parameters of the environment in which the filter is being used. Of course, with changes in the environment, the parameters of the filters being used will also change to keep the behavior of the overall system of the filter and the environment to continue to be agreeable to our purposes.

As an example, consider the use of adaptive filters in wireless communication systems. An inherent property of wireless communication channels is their time-varying behavior which is shown by their changing amplitude and phase response characteristics. In order to combat the Inter Symbol Interference (ISI) occurring due to the multipath property of these channels, the inverse filter of the channel to remove the ISI, requires the capability to change its parameters in accordance with changes in the wireless channels so that the behavior of the overall system of the channels and inverse filter, i.e., minimum ISI, is maintained. In communication literature, such an inverse filter is known as an "equalizer" and equalizers which have the property of adapting themselves to the channel are known as "adaptive equalizers" [5].

An adaptive filter is characterized by the adaptive algorithm that is implemented therein. These adaptive filter algorithms can be classified in a number of ways. For example, we can classify them according to batch-processing algorithms which process a collection of data inputs at the same time or online algorithms which process the input data as it arrives i.e in real time. They can also be categorized according to supervised and unsupervised adaptive filters where the former use a training sequence to adjust its parameters in the beginning and then switch to decision directed mode at the steady state to track variations in the environment whereas the latter do not use a training sequence at all and instead use the statistical properties of the signals.

The common property that all these algorithms share is the use of a cost function which describes the deviation of the actual behavior of the filter from the behavior that is needed. The algorithm then processes the signals with the aim of reducing this deviation, or equivalently, minimizing the cost function.

The two most widely used algorithms for adaptive filters are the Least Mean Squares(LMS) and the Recursive Least Squares(RLS) algorihtms ([4], [3]). There are many other algorithms derived from the LMS algorithm such as the sign LMS, Leaky LMS algorithms to name a few. It belongs to the gradient type algorithmic schemes inheriting low computational complexity and slow convergence but minimizing the mean square value of the error in a stochastic approximation sense. Another approach to improve the performance of the LMS is by employing a time varying step size in the standard LMS [6]. Thus large step size is used when the algorithm is far away from the optimal solution speeding up the convergence rate and small step size is used to improve the overall performance. To serve this purpose several criteria have been used, such as squared instantaneous error [6], sign changes of successive samples of the gradient [7]. Whereas a new variable step size LMS algorithm was proposed in [8], in which step size is adjusted according to the squared of the time averaged estimate of the error. Thus from the discussion above it can inferred that LMS is well established in adaptive filtering while the LMF has gained attention recently ([9],[14]).

From this point onwards, we will only consider the supervised adaptive filtering category.

1.1 Adaptive Filters

The Wiener filter provides the optimum solution in the mean square sense. However an adaptive filter provides an elegant solution if the filter is required to operate in the non-stationary environment or in an environment whose statistics are not known to us. Thus adaptive filters are defined as the filters whose characteristics can be changed or modified in accordance to our objectives and which can accomodate themselves to the changes in the environment without any user intervention.

There are different ways of classifying the adaptive filters. When our subject of interest is input-output mapping adaptive filters can be classified as linear and non-linear groups. Where linear adaptive filters compute an estimate of the desired response by using a linear combination of the available set of the observations applied to the input. On the contrary, when the mapping between the input-output is required to be non-linear we make use of an non-linear adaptive filters.

The adaptive filters generally rely on a recursive algorithms for its operation, which makes it possible for the filter to track the changes in the environment where complete knowledge of the relevant signal is not available. To begin with the algorithm starts with some predetermined set of initial conditions that are completely ignorant about the environment. In a stationary environment after successive iterations the algorithm tries to converge to an optimum solution.

1.1.1 System Model for Adaptive Filters

Before proceeding to give a general overview of the prominent online adaptive algorithms, it is instructive that we formulate the problem that is solved using the theory of the adaptive filters. Considering the case of an adaptive system identification as shown in Fig. 1.1. The output d_n is given by

$$d_n = \mathbf{u}_n \mathbf{c} + v_n, \tag{1.1}$$

where

$$\mathbf{c} = [c_1, c_2, ..., c_N]^T \tag{1.2}$$

is the vector of the unknown system parameters and

$$\mathbf{u}_n = [u_1, u_2, ..., u_N] \tag{1.3}$$

is the input data vector at time n, v_n is the plant noise, N is the number of plant parameters and $[.]^T$ is the transpose operation. The inputs $u_1, u_2..., u_N$ may be successive samples of some signal, such as in the case of adaptive echo cancelation and adaptive line enhancement. They may also be the instantaneous outputs of M parallel sensors, such as in the case of adaptive beamforming. The identification of the plant is performed by an adaptive FIR filter whose weight vector $\mathbf{w}_n$, assumed of dimension M, is adapted on the basis of error e_n given by

$$e_n = d_n - \mathbf{u}_n \mathbf{w}_n. \tag{1.4}$$

It is important to note at this point that regardless of whether the problem to be solved using adaptive filters is a system identification problem, a channel estimation problem or an inverse system estimation problem etc., the same adaptive filter algorithm can be used. The only difference between the different problems is the definition of e_n. For example e_n defined above for the plant identification problem is the difference between the known output of the unknown system and the output of the FIR adaptive filter whereas for the inverse system estimation problem, e_n is defined as the difference between the output of the inverse system and the known input d_n at time n to the system whose inverse system is to be estimated.

It is this error e_n which is used as the independent variable in the objective function for adaptive filtering. But since e_n is a function of the weight vector $\mathbf{w}_n$, the objective function can, therefore, be formulated as function of this weight vector and minimization of the cost function will give us the optimal weight vector in the sense of the objective function used. This important observation will be useful when we review some of the more important and prominent applications of adaptive filtering in the coming section.

1.2 Applications of Adaptive Filters

Adaptive filtering has a wide variety of applications in different fields. Although these applications are quite different in nature but they have one basic common feature: an input signal and a desired response to compute the error which in

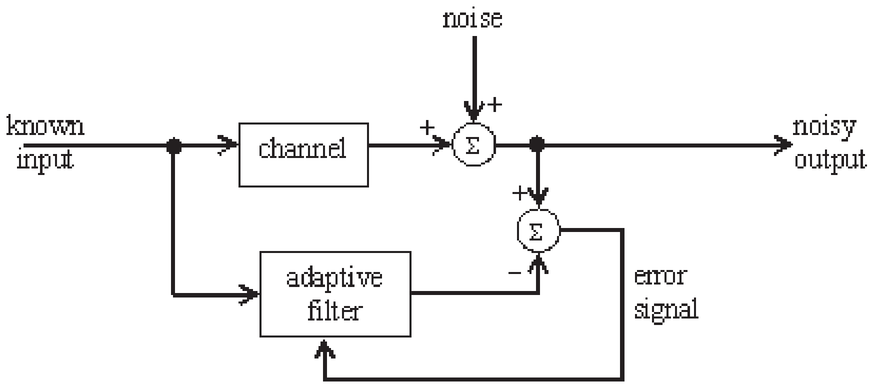

Figure 1.1: Adaptive filter.

turn is used to control a set of adjustable filter coefficients. However adaptive filters can be classified into four categories based on the way the desired response is extracted as :

- Inverse modeling or equalization
- Noise or echo cancelation
- System Identification
- Prediction

These applications are detailed next.

1.2.1 Inverse modeling or equalization

Adaptive filters that make use of an inverse model that best fits the unknown system come under this part of application. Thus at convergence inverse of the transfer function of the unknown system is approximated by the adaptive filter as shown in Fig. 1.2. In order to ensure that the input to the adaptive filter

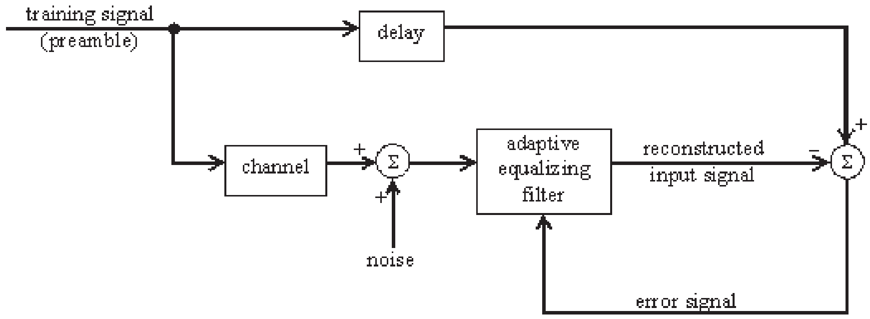

Figure 1.2: Inverse modeling problem.

has a minimum phase and suitable for equalization we introduce a delay into the desired response path. The primary use of the inverse modeling is to reduce the Inter Symbol Interference(ISI) in digital receivers which is achieved through the use of channel equalization for digital communication.

1.2.2 Noise or echo cancelation

In this type of application adaptive filter is used to cancel the unknown interference in a primary signal. This type of application is generally used in adaptive noise cancelation or echo cancelation. As we know echo cancelation is one of the important task to be performed by an adaptive algorithm in wireless communication systems. Thus LMMN algorithm is found to be useful in an long echo cancelers with two sections(the near end and the far end) separated by a bulk delay to reduce the number of coefficients [9]. The system model consists of an input signal and an unknown system, the same input is fed to the system and an adaptive filter. In the proposed plan it consists of applying the LMS algorithm in the near end and the LMF algorithm at far end section of the echo canceler which

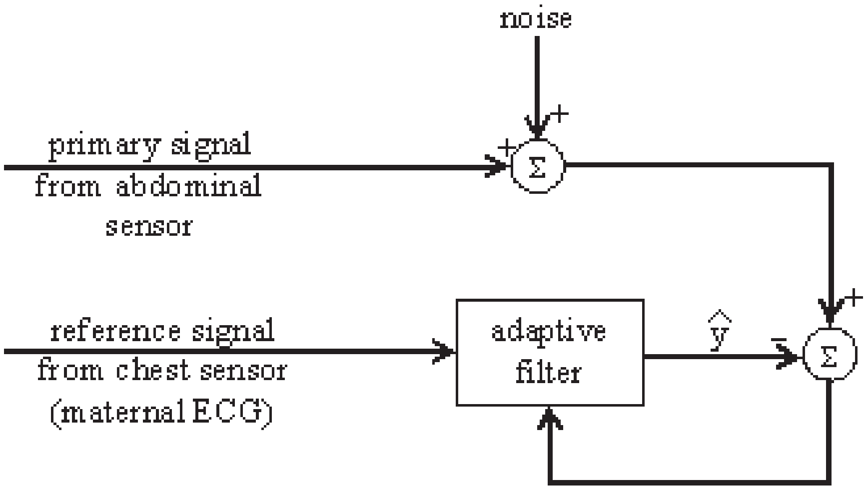

Figure 1.3: Noise Cancelation Problem.

is called as the LMMN algorithm. Thus using the LMMN algorithm we lead to a lower minimum mean square error which in turn results in less misadjustment and a faster convergence compared to other standard algorithms[3] [4].

1.2.3 System Identification

This part of application comes into picture when we want to model a certain system whose parameters are unknown to us and may be time-varying. In such a case we feed the same known input to both the system and the adaptive filter. The responses of the adaptive filter and the system are then compared and the difference between them i.e error is then used to adjust the parameters of the filter. As the number of iterations increase, the parameters of the adaptive filter approach those of the system.

1.2.4 Prediction

In this application the adaptive filter is used to provide the best prediction of the present values of the input signal from its previous values. The configuration shown in the figure is used for this purpose, where the desired signal d_n is the instantaneous value and the input to the adaptive filter is the delayed version of the same signal.

1.3 Adaptive Filter Theory

We will now briefly describe the fundamental ideas that are most widely used in the design of adaptive algorithms. First we will describe what is meant by steepest descent and Newton's methods and then we will talk about stochastic gradient methods in the context of adaptive filtering. Finally, we will list some of the prominent stochastic gradient algorithms that have been developed.

1.3.1 Steepest Descent Method

The steepest descent method [3], [4] is a popular method used in unconstrained optimization. The basic idea of the steepest descent method is to use a scalar cost function of a variable, be it scalar-valued, vector-valued or matrix-valued, and iteratively find the optimum value of this independent variable such that the cost function is minimum at that optimal value.

This can be put in mathematical form as, considering a cost function $J(\mathbf{w})$ which is continuously differentiable function of some unknown weight vector $\mathbf{w}$.

Thus we want to find an optimal solution $\mathbf{w}_0$ that satisfies the following condition:

$$J(\mathbf{w}_0) \leq J(\mathbf{w}). \tag{1.5}$$

In the steepest descent method, we start with an initial guess for $\mathbf{w}_0$ and denote it by $\mathbf{w}(0)$ and generate a sequence of weight vectors $\mathbf{w}_1, \mathbf{w}_2, ...$, such that the cost function $J(\mathbf{w})$ comes closer to a local minimum at each iteration that is

$$J(\mathbf{w}_{n+1}) < J(\mathbf{w}_n). \tag{1.6}$$

Before proceeding further, it is necessary that the reason for stating that the cost function reaches it local minimum value be understood. The reason is that the function may not be a convex function in which case the only local minimum is the global minimum. This brings forth a drawback of the steepest descent method, that is, this method does not distinguish between local and global minima and hence, depending on the choice of the initial guess, the cost function could converge to different values.

We can represent the steepest descent recursive equation in more explicit form as

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu\mathbf{p} \tag{1.7}$$

where $\mathbf{w}_{n+1}$ is the updated weight vector at time $n+1$, $\mathbf{w}_n$ is the current weight vector, μ is the step size,n is the time index and $\mathbf{p}$ is the update direction vector. It is shown in [4] that a proper choice for $\mathbf{p}$ such that $\mathbf{w}$ converges to the proper value is given by

$$\mathbf{p} = -\mathbf{B}[\nabla_{\mathbf{w}} J(\mathbf{w}_n)] \tag{1.8}$$

where $\mathbf{B}$ is any positive-definite matrix and

$$\nabla_{\mathbf{w}} J(\mathbf{w}) = \frac{dJ}{d\mathbf{w}} \mid \mathbf{w} = \mathbf{w}_n. \tag{1.9}$$

This value for $\mathbf{p}$ has an interesting interpretation. The direction of $\mathbf{p}$ at a point is opposite to the direction in which the cost function is increasing, which incidently, is the direction of the gradient vector of the function at that point.Therefore, we move along the surface of the cost function in towards its minimum (1.8). When the cost function value reaches it local minimum, relative to the initial guess, the gradient will become zero and the weight vector converges to a finite value.

Another important aspect of the steepest descent method is the selection of a proper step size μ. A value too small will lead to slow convergence whereas a value too large might make the method unstable. It can be shown that the range of values μ can take while keeping the algorithm stable are between 0 and $\frac{2}{\lambda_{max}}$ where λ_{max} is the largest eigenvalue of the correlation matrix $\mathbf{R}_u$ of the input vector $\mathbf{u}$ given as

$$\mathbf{R}_{\mathbf{u}} = E[\mathbf{u}_n \mathbf{u}_n^T]. \tag{1.10}$$

The main advantage of the steepest descent method is its simplicity. However, the convergence rate may be too slow in the case of steepest descent method. This is due to the fact that this method is based on the first order approximation of the error-performance surface around the current point in that it only uses the first-order derivatives i.e. the gradient, in its update equation.

A faster rate of convergence can be achieved by using a second-order approximation of the error-performance surface around the current point, which translates to assigning to $\mathbf{B}$ the value of inverse of the Hessian matrix $\mathbf{H}$ of $\mathbf{R_u}$. This method is known as Newton's method.

Stochastic Gradient Methods

There are two types of objective functions used in adaptive filtering- stochastic and deterministic. Objective functions which are given in terms of statistics of the input signals are known as stochastic random variables whereas functions which act on the actual values of the signals are known as deterministic.

When using the steepest descent method to optimize stochastic cost functions, the gradient and the Hessian matrices of the stochastic cost function with respect to the weight vector are also stochastic in nature. However, in practice, we do not have information about the stochastic properties of the signal and only have the instantaneous values. For this reason, when using the steepest descent method in this case, we try to approximate the gradient and/or the Hessian Matrix using functions. The resulting algorithms are known as Stochastic gradient algorithms.

Because we are using approximations to the true gradient and/or Hessian matrix, there will be a difference in the successive values of weight vector obtained using the steepest descent method and the corresponding stochastic gradient method. This difference is termed as gradient noise. The more accurate the approximation functions, the closer the performance of the stochastic gradient algorithm will be to the corresponding steepest descent algorithm and smaller will be the gradient noise [4].

A stochastic gradient algorithm based on the steepest descent method to minimize the mean square error criterion is the Least Mean Square (LMS) algorithm and the stochastic gradient algorithm for the least mean fourth criterion is the Least Mean Fourth (LMF) [15] algorithm, the combination of which gives us the Least Mean Mixed Norm algorithm(LMMN) which will be the subject of interest in this thesis.

1.4 Least Mean Algorithms

This type of algorithms minimize the statistical average of the error i.e $E[f(e_n)]$ which is a convex function of the filter coefficients $\mathbf{w}_n$. Thus $\mathbf{w}_n$ can be adapted using steepest descent algorithm as :

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \mu \nabla E[f(e_n)] \tag{1.11}$$

where $\nabla E[f(e_n)]$ represents the gradient of $E[f(e_n)]$ with respect to $\mathbf{w}_n$. Thus different least mean algorithms can be obtained with suitable choice of function some of them are:

1.4.1 Least Mean Square Algorithm

If $f(e_n) = e_n^2$ the least mean square algorithm is obtained which is one of the most important algorithm in adaptive filtering. Filter coefficients are updated according to the following equation.

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu e_n \mathbf{u}_n \tag{1.12}$$

where the error signal e_n is defined as $e_n = d_n - y_n$, $\mathbf{u}_n$ is the tap input vector, and $\mathbf{w}_n$ represents the tap weights of the adaptive filter. The parameter μ is a positive constant called step size which is used to control size of the correction applied to the tap weights.

The LMS is very simple to implement and yet capable of achieving satisfactory performance under right conditions. The major limitation in working with LMS algorithm is its slow convergence and its sensitivity to variations in the input signal correlation matrix.

In a non-stationary environment, the orientation of the error performance varies continuously with time. In this case the LMS algorithm has a added task of continuously tracking the bottom of the error surface [4].

1.4.2 Least Mean Fourth Algorithm

In this algorithm we minimize the fourth power of the error. It is a general case of family of the steepest descent algorithms [4] and its weight update equation is given by

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu e_n^3 \mathbf{u}_n \tag{1.13}$$

The LMF algorithm has a faster convergence as compared to the LMS algorithm but has higher steady state error. The complexity of the LMF algorithm is more compared to the LMS algorithm because of the higher power of the e_n involved in the adaption of the weights.

1.4.3 Least Mean Mixed Norm Algorithm

As we know that LMS algorithm can achieve low steady state error than the LMF algorithm but the convergence speed is slower. So keeping the fact in view that convex addition of two convex function is also a convex function, a class of mixed norm was developed [10]. The cost function of LMMN is linear mixture of $J_2(n) = E[e_n^2]$ and $J_4(n) = E[e_n^4]$ which is given as:

$$J_n = \frac{\delta}{2} J_2(n) + \frac{(1-\delta)}{4} J_4(n) \tag{1.14}$$

where $\delta \epsilon [0, 1]$ controls the mixture. The gradient vector which defines the search direction is

$$\nabla J_n = -E\{e_n\{\delta + (1-\delta)e_n^2\}\mathbf{x}(n)\} \tag{1.15}$$

than we define a stochastic gradient algorithm based on an instantaneous estimate of ∇J_n. Finally the update equation of the LMMN algorithm is

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu e_n\{\delta + (1-\delta)e_n^2\}\mathbf{x}(n) \tag{1.16}$$

where μ is the step size. The adaptation algorithm reduces to LMS and LMF algorithm respectively for $\delta = 1$ and $\delta = 0$.

1.5 The Weight drift problem

To begin with discussing the weight drift problem in the basic algorithm i.e. LMS algorithm, we will start with the LMS recursion which is given by

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu e_n \mathbf{u}_n \tag{1.17}$$

$$e_n = d_n - \mathbf{w}_n^T \mathbf{u}_n \tag{1.18}$$

where d_n can be taken from the basic model as

$$d_n = \mathbf{u}_n^T \mathbf{c} + v_n, \tag{1.19}$$

where $\mathbf{c}$ is a weight vector and v_n is the noise.

The weight drift problem can be understood by the following example. Assume, that at iteration $\mathbf{n}$, the input vector x_n is orthogonal to the weight error vector $\mathbf{w}\tilde{_{n+1}} = \mathbf{c} - \mathbf{w}_{n+1}$. It then follows that $v_n = d_n - \mathbf{u}_n^T \mathbf{w}_n$. Consequently, the weight error vector satisfies the update equation

$$\tilde{\mathbf{w}}_n = \tilde{\mathbf{w}}_n + \mu \mathbf{u}_n v_n \tag{1.20}$$

Taking norms of the above equation we get

$$\| \tilde{\mathbf{w}_{n+1}} \|^2 = \| \tilde{\mathbf{w}}_n \|^2 + \mu^2 \| \mathbf{u}_n \|^2 v_n^2 \tag{1.21}$$

solving this equation at N samples we get

$$\| \tilde{\mathbf{w}}_N \|^2 = \| \tilde{\mathbf{w}}_0 \|^2 + \sum_{n=1}^{N} \mu^2 \| \mathbf{u}_n \|^2 v_n^2 \tag{1.22}$$

Thus it can be conveyed from this relation that as $\| \tilde{\mathbf{w}}_N \|^2 \rightarrow \infty$ as $N \rightarrow \infty$, if $\mu \mathbf{u}_n v_n$ is not a finite energy sequence. This problem does not occur with the Leaky LMS algorithm [11], the recursive equation of which is given by :

$$\mathbf{w}_{n+1} = (1 - \mu\alpha)\mathbf{w}_n + \mu \mathbf{u}_n e_n \tag{1.23}$$

where α is the leakage parameter. The term leakage stems from the fact that, unlike the conventional LMS, where the weights remain stationary in case of stalling, in Leaky LMS, the weights "leak out" in case stalling occurs i.e. the input sequence becomes zeros. To see how Leaky LMS mitigates the drift problem in LMS algorithm, using the same example and by the same steps of computation, we get

$$\| \tilde{\mathbf{w}_{n+1}} \|^2 = (1 - \mu\alpha)^2 \| \tilde{\mathbf{w}_n} \|^2 + \mu^2 \| \mathbf{u}_n \|^2 v_n^2 \tag{1.24}$$

so that $\| \tilde{\mathbf{w}}_n \|^2$ remains bounded for $0 < \mu\alpha < 1$. However, the Leaky LMS does add bias to the solution and $\| \tilde{\mathbf{w}_n} \|$ does not reach 0 except for the case $\alpha = 0$ which is the case for LMS [22].

1.6 Motivation for Leaky LMMN

The description and the use of the basic adaptive filtering algorithm i.e. the LMS algorithm has been described in the previous section. The LMMN algorithm just like the LMS algorithm suffers from the weight drift problem. Taking this fact into consideration we shall employ the leakage technique to the LMMN algorithm and refer it as Leaky-LMMN, the basic cost function of which is given as

$$J(\mathbf{w}) = \alpha||\mathbf{w}||^2 + \{\delta E[e_n^2] + (1 - \delta)E[e_n^4]\}. \tag{1.25}$$

For the algorithm defined in the above equation we will make use of a fixed mixing or a weighted factor that is predetermined. But instead in this work we make use of an self-adapting time variable weighting factor "α_n" [13]. This factor is then updated every iteration so it is large when we are away from the optimum and decreases as we approach the optimum.

1.7 Thesis Objectives

The aim of this thesis is to derive the Leaky-LMMN algorithm taking Leaky-LMS as the counterpart, than establishing the condition for convergence and compare the performance of the proposed algorithm in terms of the steady state, transient and the tracking analysis and then we compare the performance of the Leaky LMMN and the traditional LMMN with time varying mixing parameter.

The objectives of the thesis can be outlined as :

1. To examine the convergence properties of the proposed algorithm (Leaky LMMN) and to derive sufficient and necessary condition for the convergence in the mean and to find the weight error vector recursion in the mean square sense.

2. To analyze the steady-state performance of the proposed algorithm and to derive the expression for excess mean square error at the steady-state.

3. To present the simulation scenario in support of the analytical analysis.

4. To derive tracking analysis of Leaky-LMMN, to show how capable is newly proposed algorithm in tracking changes in the environment.

5. To derive transient analysis of the proposed algorithm.

6. To compare the convergence speed of the Variable Leaky LMMN algorithm with the traditional LMMN algorithm.

7. To compare Variable weight mixed norm algorithm with the Fixed Mixed norm algorithm.

CHAPTER 2

PROPOSED LEAKY LEAST MEAN MIXED NORM ADAPTIVE ALGORITHM

2.1 Introduction

In this chapter we will make use of the cost function in the development of the proposed algorithm which will lead to the derivations of the steepest descent algorithm and the stochastic gradient algorithms. Then making use of this stochastic gradient update equation we will formulate the proposed leaky least mean mixed norm algorithm. In the process we will first derive the fundamental weighted energy relation making appropriate assumptions when required, than this weighted energy relation will be used in the steady state, transient and the tracking analysis of the Leaky Least Mean Mixed Norm algorithm.

2.2 The Leaky Least Mean Mixed Norm Algorithm

The Least Mean Squares(LMS) algorithm is one of the most widely used adaptive scheme. It has several desirable features and at the same time some limitations. As such several LMS variants have been produced that trade off some of the LMS features and enhances its performance. In here we deal with the class of least mean square algorithms that employ an error nonlinearity $f(e_n)$ instead of the error term in LMS adaptation ([17], [18], [19], [20]). Like example Least Mean Fourth algorithm [15] and the Least Mean Mixed Norm Algorithm [21]. The error nonlinearity used in the mixed norm algorithm is given by

$$f(e_n) = \alpha * e_n + (1 - \alpha) * (e_n)^3 \tag{2.1}$$

with α as the mixing parameter [13], [21]. This algorithm is found to provide a better performance in both Gaussian and Non-Gaussian environments than either LMS or the LMF.

Following the discussion above we will derive the Leaky Least Mean Mixed Norm algorithm. The assupmtions that are useful for the analysis can be stated as :

A1 There exists a vector $\mathbf{c}$ such that $d_n = \mathbf{u}_n \mathbf{c} + v_n$.

A2 The noise sequence $\{v_n\}$ is i.i.d. with zero odd order moments and variance

$\sigma_v^2 = E[v_n]^2$.

A3 The sequence v_n is independent of $\mathbf{u}_j$,$\mathbf{w}_k$ for all j,k.

A4 The regressor covariance matrix is $\mathbf{R}_u = E[\mathbf{u}_n^T \mathbf{u}_n] > 0$.

We will make use of a system identification model given by (1.1) in developing the proposed algorithm. The desired response and the cost function that will be used as then basis in the development of the proposed algorithm are given as:

$$\begin{aligned} d(n) &= \sum_{i=0}^{M-1} \mathbf{u}_i \mathbf{c}_i + v(n) \\ &= \mathbf{u_n c} + v(n) \end{aligned} \tag{2.2}$$

The stochastic cost function which is used as a basis for the proposed algorithm is given as

$$\mathbf{J}(\mathbf{w}) = \delta E[e_n^2] + (1-\delta)E[e_n^4] + \alpha \parallel \mathbf{w}_n^2 \parallel \tag{2.3}$$

where α is the leakage factor. Now we get the direction vector $\mathbf{p}$ from equation (1.8) as:

$$P = -e_n[\delta + (1-\delta)e_n^2]](-\mathbf{u}^T) - 2\alpha\mathbf{w}_n \tag{2.4}$$

substituting this in (1.7), the resulting steepest descent update equation minimizes as:

$$
\begin{aligned}
\mathbf{w}_{n+1} &= \mathbf{w}_n + \mu\mathbf{P} \\
&= \mathbf{w}_n + \mu[\mathbf{u}^T e_n[\delta + (1-\delta) \parallel e_n \parallel^2] - 2\alpha\mathbf{w}_n]
\end{aligned}
\tag{2.5}
$$

$$
\mathbf{w}_{n+1} = (1 - 2\mu\alpha)\mathbf{w}_n + \mu\mathbf{u}_n^T e_n[\delta + (1-\delta) \parallel e_n \parallel^2] \tag{2.6}
$$

where we have used the equation (1.18). Now to minimize the equation (2.3) we just remove the expectation operator and the factor 2 gets absorbed into $'\mu'$ and the resulting equation is given as:

$$
\mathbf{w}_{n+1} = (1 - \mu\alpha)\mathbf{w}_n + \mu\mathbf{u}_n^T e_n[\delta + (1-\delta) \parallel e_n \parallel^2] \tag{2.7}
$$

The resulting equation given by (2.7) helps in preventing the weight drift problem caused either by the inadequacy due to excitation in the input sequence or due to the finite precision effects.

2.3 Fundamental Energy Conservation Relation

To study the performance behaviour of the Leaky Least Mean Mixed Norm algorithm we make use of the basic fundamental energy conservation relation [4], [23]-[25], proves out to be a useful framework in the analysis of adaptive filters in this thesis. Because of its wide spread application it can used in different adaptive algorithms without resorting to any restrictive assumptions that are generally encountered in the literature review of adaptive filtering algorithms. Some of them being the gaussianity assumption on the noise and the independent assumption to name a few. Even the general nature allows us for the easy comparison between the different algorithms. The weighted squared Euclidean norm of a vector $\mathbf{x}$ is defined as

$$||\mathbf{x}||^2_{\mathbf{A}} = \mathbf{x}^T\mathbf{A}\mathbf{x}, \tag{2.8}$$

where $\mathbf{A}$ is some positive-definite symmetric weighting matrix. The choice $\mathbf{A} = \mathbf{I}$ results in the standard Euclidean norm of x

$$||\mathbf{x}||^2 = \mathbf{x}^T\mathbf{x}. \tag{2.9}$$

To start with the energy conservation relation for the Leaky LMMN, we will begin with the Leaky LMMN update equation which is given by (2.7). Now subtracting both sides of the equation (2.7) from $\mathbf{c}$, we get

$$\mathbf{w}_n = \mathbf{c} - \tilde{\mathbf{w}_n} \tag{2.10}$$

$$\mathbf{c} - \tilde{\mathbf{w}_{n+1}} = \mathbf{c} - (1-\mu\alpha)\mathbf{w}_n - \mu\mathbf{u}_n^T e_n[\delta + (1-\delta) \parallel e_n \parallel^2] \tag{2.11}$$

$$\tilde{\mathbf{w}_{n+1}} = (1-\mu\alpha)\tilde{\mathbf{w}}_n + \mu\alpha\mathbf{c} - \mu\mathbf{u}_n^T e_n[\delta + (1-\delta) \parallel e_n \parallel^2] \tag{2.12}$$

Now taking the Euclidean norm of both sides of equation (2.12) and using some positive definite weighting matrix $\mathbf{A}$, we get the following weighted energy conservation relation:

$$
\begin{aligned}
\| \tilde{\mathbf{w}_{n+1}} \|^2 &= (1-\mu\alpha)^2 \| \tilde{\mathbf{w}}_n \|^2_{\Sigma} + \| \mu\alpha\mathbf{c} \|^2_{\Sigma} + \mu^2 \| \mathbf{u}_n \|^2 \| e_n \|^2 [\delta + (1-\delta) \| e_n \|^2] \\
&\quad + 2\mu\alpha(1-\mu\alpha)\mathbf{c}^T \sum \tilde{\mathbf{w}}_n - 2\mu(1-\mu\alpha)e_n[\delta + (1-\delta) \| e_n \|^2]\mathbf{u}_n \sum \tilde{\mathbf{w}}_n \\
&\quad - 2\mu^2\alpha\mathbf{c}^T \sum e_n[\delta + (1-\delta) \| e_n \|^2]\mathbf{u}_n \qquad (2.13)
\end{aligned}
$$

$$
\begin{aligned}
\| \tilde{\mathbf{w}_{n+1}} \|^2 &= (1-\mu\alpha)^2 \| \tilde{\mathbf{w}}_n \|^2_{\Sigma} + \| \mu\alpha\mathbf{c} \|^2_{\Sigma} + \mu^2 \| \mathbf{u}_n \|^2 \| e_n \|^2 [\delta + (1-\delta) \| e_n \|^2] \\
&\quad - 2\mu(1-\mu\alpha)e_n[\delta + (1-\delta) \| e_n \|^2]e_a^{\Sigma}(n) \\
&\quad + 2\mu\alpha\mathbf{c}^T \sum [(1-\mu\alpha)\mathbf{w}_n - \mu e_n[\delta + (\tilde{1}-\delta) \| e_n \|^2]\mathbf{u}_n] \qquad (2.14)
\end{aligned}
$$

where $e_a^{\mathbf{A}}(n) = \mathbf{u}_n\mathbf{A}\mathbf{w}_n$ is the weighted a-priori estimation error. For $\mathbf{A} = \mathbf{I}$, we have the standard a-priori estimation error $e_a(n)$.Thus weighted energy conservation relation given by equation (2.14) will be used in the coming chapters to study the performance of the Leaky LMMN adaptive algorithm in terms of

Steady State Analysis, which relates to determining the steady state values of $E[||\mathbf{w}_n||^2]$, $E[e_a^2(n)]$ and $E[e_n^2]$.

Stability, which relates to determining the range of values of the step-size over which $E[||\mathbf{w}_n||^2]$ and $E[e_a^2(n)]$ remain bounded.

Transient Analysis, which is concerned with studying the time evolution of $E[||\mathbf{w}_n||^2]$ and $E[e_a^2(n)]$.

2.4 Performance Analysis of the Proposed Leaky LMMN Algorithm

In this chapter, the results of the computer simulations are presented in order to investigate the performance behaviors of the proposed Leaky LMMN algorithm. A number of simulation results are carried out to support the theoretical findings.

In order to start our discussion we will first bring into light that the conventional LMMN algorithm suffers from weight drift problem, thus we will make use of the Leaky LMMN algorithm to overcome this weight drift problem. Then we will proceed our discussion with showing a good comparison between the theoritical findings of the proposed Leaky LMMN algorithm and the simulation results. Then we will compare the convergence speed of the traditional LMMN algorithm with the proposed Leaky LMMN algorithm. Finally we will conclude our simulations with comparison of variable weight Leaky LMMN and the Fixed mixed norm algorithm. These simulations can be divided into the following categories:

1. The Leaky LMMN mitigates the weight drift problem that is encountered in the conventional LMMN algorithm.
2. Comparison of the transient performance of the Leaky LMMN and the simulation results for Gaussian, Uniform and Laplacian noise environments at noise variance of 0.1, 0.01 and 0.001.
3. Comparison of the tracking performance of the Leaky LMMN and the simulation results for Gaussian and Uniform noise environments at noise variance

of 0.1, 0.01 and 0.001.

4. Comparison of the convergence speed of LMMN algorithm and the proposed Leaky LMMN algorithm in achieving the same steady state error.

5. Comparison of the variable Leaky LMMN algorithm with the Fixed Mixed Norm algorithm.

2.4.1 Comparison of LMMN and Leaky LMMN in Weight Drift Environment

In this section, we will present the simulation to show how weight drift problem occurs in the LMMN algorithm and how it can be prevented from happening using the Leaky LMMN. In this simulation, the parameters have been chosen to speed up the weight drift phenomenon as was done in [22]. The true weight error vector is given by $[0.7071\ -0.7071]^T$ while the input regressor vector is randomly assigned values of $\pm[0.5\ -0.5]$ with equal probability so that the input covariance matrix is singular. The output noise and the quantization noise are grouped together and modeled as a Gaussian random vector with mean $[0.49\ -0.49]^T$ whose elements are independent of each other and have a variance of 10^{-3} . The number of quantization bits for the adaptive filter coefficients and the regressor values are set to 10. The step size was taken to be 0.0156 and the product of the step size and the leakage factor was set at 0.002. We make a single run over 10^4 samples and have taken the infinite norms of the updated weight vectors in case of both the LMMN and the Leaky LMMN.

As can be seen from Fig. 2.1, we see that in the case of LMMN, the parameter drift causes the adaptive filter weights to blow up while in the case of the Leaky LMMN, the adaptive filter weights are bounded.

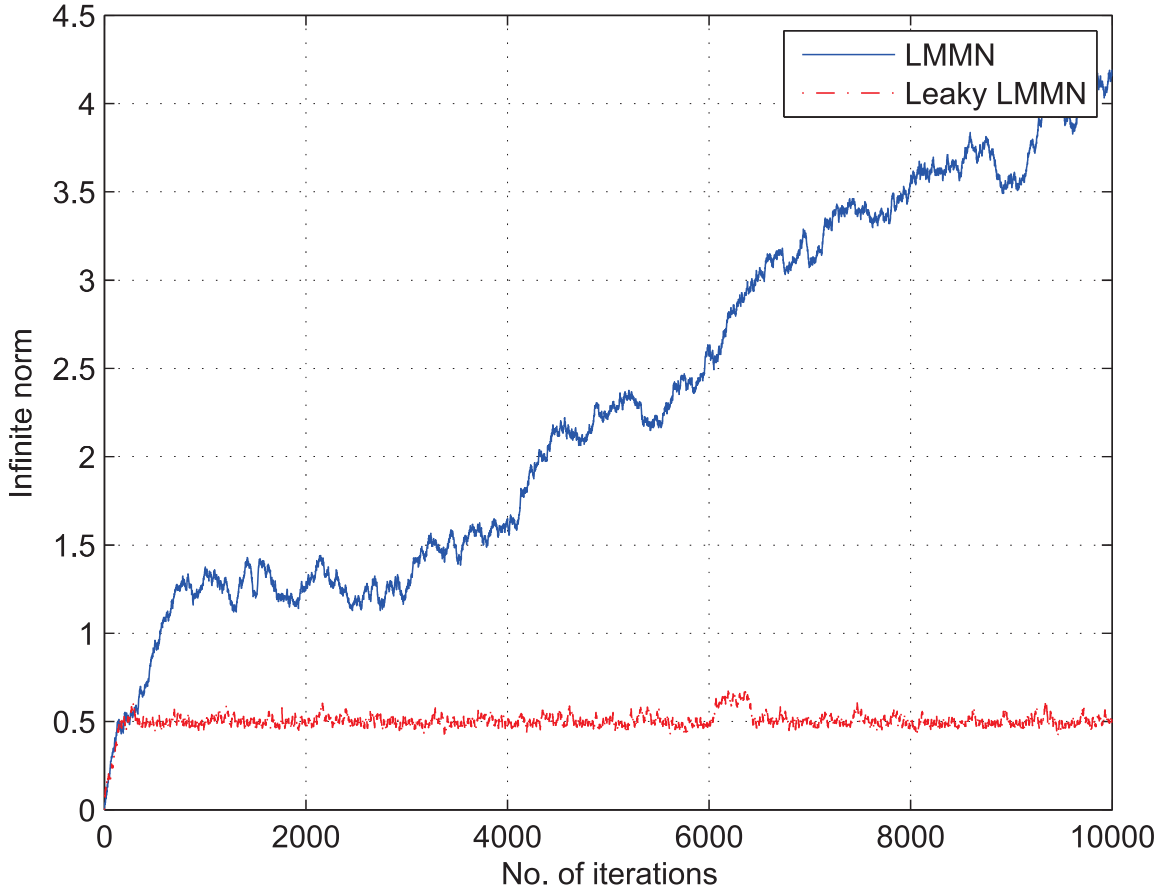

Figure 2.1: Parameter drift situation.

CHAPTER 3

TRANSIENT ANALYSIS OF THE PROPOSED LEAKY LMMN ADAPTIVE ALGORITHM

3.1 Introduction

As we now that the adaptive filters are time invariant and have the inherent ability to track the changes in the environment. Their function can best be analyzed in terms of how fast they adapt changes in the signal statistics this is what termed as the transient performance of the adaptive filter. Thus in our further discussion we will be making use of the fundamental energy conservation relation for our analysis.

3.2 Transient Analysis

The transient analysis carried out in this chapter will basically concentrate on the ranges of the step size for which the $||\tilde{\mathbf{w}_n}||$ and e_n remain bounded in terms of both mean and mean-square sense. Then the analysis will be carried out to find how $E[||\tilde{\mathbf{w}_n}||^2]$ and $E[e_n^2]$ evolve with time.

3.2.1 Mean Convergence Behavior

We will start with the 2.12 Leaky LMMN weight error vector recursion and taking the expectation of the same, we get

$$E[\tilde{\mathbf{w}_{n+1}}] = (1-\mu\alpha)E[\tilde{\mathbf{w}_n}] + \mu\alpha\mathbf{c} - \mu E[\mathbf{u}_n^T e_n[\delta + (1-\delta) \parallel e_n \parallel^2]] \tag{3.1}$$

To solve for $E[\mathbf{u}_n^T e_n[\delta + (1-\delta) \parallel e_n \parallel^2]]$, we will use the following assumption:

A6 The regressors $\mathbf{u}_n$ are Gaussian distributed.

To make the analysis more tractable for performance comparisons we make use of the assumption **A6** though it is not practical in communication scenarios [26]. Thus making use of this assumption we can find out the simplified expression as:

$$\begin{aligned} E[\mathbf{u}_n^T e_n[\delta + (1-\delta) \parallel e_n \parallel^2]] &= \delta E[\mathbf{u}_n^T e_n] + (1-\delta)E[\mathbf{u}_n^T e_n^3] \\ &= \delta\mathbf{R}E[\tilde{\mathbf{w}_n}] + 3(1-\delta)(\sigma_v^2+\zeta)\mathbf{R}E[\tilde{\mathbf{w}_n}] \\ &= \{\delta + 3(1-\delta)(\sigma_v^2+\zeta)\}\mathbf{R}E[\tilde{\mathbf{w}_n}] \end{aligned} \tag{3.2}$$

now putting this in equation (3.1) we get final weight error recursion as:

$$E[\tilde{\mathbf{w}_{n+1}}] = [I - \mu[\alpha I + (\delta + 3(1-\delta)(\sigma_v^2 + \zeta)\mathbf{R}]]E[\tilde{\mathbf{w}}_n] + \mu\alpha\mathbf{c} \tag{3.3}$$

where $\zeta = E[e_{an}^2]$.

Now to find the mean convergence condition on the step-size we will make use of the Cramer-Rao bound which is given by $\vartheta \leq \zeta$, this approach is taken from [27], where it is associated with estimating the $\mathbf{u}_n\mathbf{c}$ by $\mathbf{u}_n\mathbf{w}_n$, then from (3.1) we acknowledge that $\tilde{\mathbf{w}}_n$ is convergent in the mean if the eigenvalues of $[I - \mu[\alpha I + (\delta + 3(1-\delta)(\sigma_v^2 + \zeta)\mathbf{R}]]$ lie between -1 and 1. From this we can find out the range of step-size values for which $\tilde{\mathbf{w}}_n$ remains bounded which is given as

$$-1 < 1 - \mu[\alpha + (\delta + 3(1-\delta)(\sigma_v^2 + \zeta)\mathbf{R})] < 1 \tag{3.4}$$

$$-2 < -\mu[\alpha + (\delta + 3(1-\delta)(\sigma_v^2 + \zeta)\mathbf{R})] < 0 \tag{3.5}$$

$$0 < \mu < \frac{2}{\alpha + [\delta + 3(1-\delta)(\sigma_v^2 + \zeta)\mathbf{R}]} \tag{3.6}$$

3.2.2 Constructing the Learning Curves

Now after finding the range of step-sizes for which the weight vector $\tilde{\mathbf{w}}_n$ remains bounded, we will move on to construct the state space model that describes the time evolution of $E[||\mathbf{w}_n||^2]$ and $E[e_{an}^2]$. Taking expectation of (2.14), we get

$$\begin{aligned} E[\| \mathbf{w}_{n+1} \|^2] &= (1-\mu\alpha)^2 E[\| \tilde{\mathbf{w}}_n \|^2] + \| \mu\alpha\mathbf{c} \|^2 + \mu^2 E[\| \mathbf{u}_n \|^2 e_n^2(\delta + (1-\delta)e_n^2)^2] \\ &\quad -2\mu(1-\mu\alpha)E[e_n(\delta + (1-\delta)e_n^2)e\Sigma_{an}] \\ &\quad +2\mu\alpha\mathbf{c}^T\Sigma[(1-\mu\alpha)E[\tilde{\mathbf{w}}_n] - \mu E[e_n(\delta + (1-\delta)e_n^2)\mathbf{u}_n]] \end{aligned} \quad (3.7)$$

Now to proceed further we have to evaluate the terms

$E[\| \mathbf{u}_n \|^2 e_n^2(\delta+(1-\delta)e_n^2)^2]$, $E[e_n(\delta+(1-\delta)e_n^2)e\Sigma_{an}]$ and $E[e_n(\delta+(1-\delta)e_n^2)\mathbf{u}_n]$.

Evaluation of Term $E\left[e_n(\delta + (1-\delta)e_n^2)e_{a\,n}^{\Sigma})\right]$

Making use of the assumption that $e_a(n)$ and $e_a^{\Sigma}(n)$ are jointly gaussian by **A6** and making use of the independent assumption of $v(n)$ by **A2**, then using the Price's theorem [4] we can evaluate $E\left[e_n(\delta + (1-\delta)e_n^2)e\Sigma_{an}\right]$ as

$$E[e_n[\delta + (1-\delta) \| e_n \|^2]e_{a\,n}^{\Sigma}] = \delta E[e_{an}e_a^{\Sigma}(n)] + (1-\delta)E[e_n^3 e_a^{\Sigma}(n)] \quad (3.8)$$

using the Price Theorem

$$E[e_a^{\Sigma}(n)g[e_n]] = Ee_a(n)e_a^{\Sigma}(n)(\frac{Ee_a^*(n)g[e_n]}{E||e_a(n)||^2}) \quad (3.9)$$

here $g[e_n] = e_n$ but $e_n = e_a(n) + v_n$

$$E[e_a(n)e_a^{\Sigma}(n)] = E[e_a(n)e_a^{\Sigma}(n)] * H_G \quad (3.10)$$

where

$$H_G = \frac{Ee_a^*(n)e_a(n)}{E[| \, e_a^{\Sigma}(n)]^2} \tag{3.11}$$

and $E[e^3(n)e_a^{\Sigma}(n)] = E[e_a(n)e_a^{\Sigma}(n)] * H_G$ here $H_G = 3[E[||e_a(n)||^2] + \sigma_v^2]$ further we can write this equation as $H_G = 3[\sigma_v^2 + \zeta]$

Evaluation of Term $E\,[\| \, \mathbf{u}_n \, \|^2 \; e_n^2(\delta + (1-\delta)e_n^2)^2]$

To evaluate this term, we will the following approximation [4]

A7 The adaptive filter is long enough so that $||\mathbf{u}_n||^2_{\mathbf{A}}$ is independent of e_n.

This assumption allows us to write $E\,[\| \, \mathbf{u}_n \, \|^2 \; e_n^2(\delta + (1-\delta)e_n^2)^2]$ as

$$\begin{aligned} E[\| \, \mathbf{u}_n \, \|^2_{\Sigma} \; e_n^2[\delta + (1-\delta)e_n^2]^2] &= \delta^2 E[\| \, \mathbf{u}_n \, \|^2_{\Sigma} \; e_n^2] + (1-\delta)^2 E[\| \, \mathbf{u}_n \, \|^2_{\Sigma} \; e_n^6] \\ &\quad + 2\delta(1-\delta)E[\| \, \mathbf{u}_n \, \|^2_{\Sigma} \; e_n^4] \end{aligned} \tag{3.12}$$

and substituting $e_n = e_a(n) + v_n$ we get

$$\begin{aligned} E[\| \, \mathbf{u}_n \, \|^2_{\Sigma} \; e_n^2] &= tr(\mathbf{R}\Sigma)E[e_a^2(n) + v_n^2] \\ &= tr(\mathbf{R}\Sigma)(\zeta + \sigma_v^2) \end{aligned}$$

$$\begin{aligned} E[\| \, \mathbf{u}_n \, \|^2_{\Sigma} \; e_n^6] &= tr(\mathbf{R}\Sigma)E[e_n^6] \\ &= tr(\mathbf{R}\Sigma) * H_U \end{aligned}$$

here $H_U = 15\zeta^3 + 45\zeta^2\sigma_v^2 + 15\zeta\xi_v^4 + \xi_v^6$

$$E[\| \mathbf{u}_n \|_\Sigma^2 e_n^4] = tr(\mathbf{R}\Sigma) * H_U \tag{3.13}$$

here $H_U = 6\zeta^3 + 18\zeta^2\sigma_v^2 + 9\zeta^2\xi_v^4 + \xi_v^6$ Using (3.18),(3.11) and (3.13) in (3.7) and some algebraic manipulation, we get the following result:

$$\begin{aligned} E[\| \tilde{\mathbf{w}_{n+1}} \|_\Sigma^2] &= (1-\mu\alpha)^2 E[\| \tilde{\mathbf{w}}_n \|_\Sigma^2] + \| \mu\alpha\mathbf{c} \|_\Sigma^2 + \mu^2[\delta^2 H_U^{LMS} + (1-\delta)^2 H_U^{LMF} \\ &\quad +2\delta(1-\delta)E[\| e_n \|^4]] + 2\mu(1-\mu\alpha)E[e_a(n)e_a^\Sigma(n)]H_G \\ &\quad +2\mu\alpha\mathbf{c}^T\Sigma\mathbf{J}E[\tilde{\mathbf{w}}_n] \end{aligned} \tag{3.14}$$

where $H_U = \delta^2 h_U^{LMS} + (1-\delta)^2 h_U^{LMF} + 2\delta(1-\delta)[3\zeta^2 + 4\zeta^2\sigma_v^2 + \xi_v^4]$ and

$H_G = \delta h_G^{LMS} + (1-\delta)h_G^{LMF}$ with ξ_v^4 and ξ_v^6 are the fourth and sixth order moments of v_n.

Evaluation of Term $E\left[e_n(\delta + (1-\delta)e_n^2)\mathbf{u}_n\right]$

Here we make use of **A6** stating that $\mathbf{u}_n$ are Gaussian Regressors we have $E[e_n\mathbf{u}_n] = \mathbf{R}E[\tilde{\mathbf{w}}_n]$ substituting this in the third term we get as :

$$(1-\mu\alpha)E[\tilde{\mathbf{w}}_n] - \mu E[e_n(\delta+(1-\delta)e_n^2)\mathbf{u}_n] = [I - \mu\{\alpha I + [\delta + 3(1-\delta)(\sigma_v^2+\zeta)]\mathbf{R}\}]E[\tilde{\mathbf{w}}_n] \tag{3.15}$$

Finally substituting the above equations in (3.14)

$$\begin{aligned}
E[\| \tilde{\mathbf{w}_{n+1}} \|^2_\Sigma] &= (1-\mu\alpha)^2 E[\| \tilde{\mathbf{w}}_n \|^2_\Sigma] + \| \mu\alpha\mathbf{c} \|^2_\Sigma + \mu^2 tr(\mathbf{R}) H_U \\
&\quad -2\mu(1-\mu\alpha)E[e_a(n)e_a^\Sigma(n)]H_G \\
&\quad +2\mu\alpha\mathbf{c}^T\Sigma\mathbf{J}E[\tilde{\mathbf{w}}_n] \qquad (3.16)
\end{aligned}$$

where

$$\mathbf{J} = [I - \mu\{\alpha I + [\delta + 3(1-\delta)(\sigma_v^2 + \zeta)]\mathbf{R}\}] \qquad (3.17)$$

More is needed in order to evaluate (3.16) since it is hard to evaluate $E\left[e_a(n)e_a^\Sigma(n)\right]$ due to the dependencies among the regressors $\mathbf{u}_n$. Therefore, will make the following assumption [4],[23] making use of the assumption that, the sequence of vectors $\mathbf{u}_n$ are independent and identically distributed. $\mathbf{u}_n$ and $\tilde{\mathbf{w}}_n$ become independent since now $\tilde{\mathbf{w}}_n$ depends only on $\mathbf{u}_{n-1}$. Therefore, we can express $E\left[e_a(n)e_a^\Sigma(n)\right]$ as

$$\begin{aligned}
E[e_a(n)e_a^\Sigma(n)] &= E[\tilde{\mathbf{w}}_n{}^T\mathbf{u}_n^T\mathbf{u}_n\Sigma\tilde{\mathbf{w}}_n] \\
&= E[\tilde{\mathbf{w}}_n^T E[\mathbf{u}_n^T\mathbf{u}_n/\tilde{\mathbf{w}}_n]\Sigma\tilde{\mathbf{w}}_n] \\
&= E[\tilde{\mathbf{w}}_n^T E[\mathbf{u}_n^T\mathbf{u}_n]\Sigma\tilde{\mathbf{w}}_n] \\
&= E[\tilde{\mathbf{w}}_n\mathbf{R}\Sigma\tilde{\mathbf{w}}_n] \\
&= E[\| \tilde{\mathbf{w}}_n \|^2_{\mathbf{R}\Sigma}]. \qquad (3.18)
\end{aligned}$$

From (3.18) we can see that for $\sum = \mathbf{I}$, (3.18) results in

$$\begin{aligned} E[\parallel \tilde{\mathbf{w}_n} \parallel^2_{\mathbf{R}\Sigma}] &= E[e_a(n)e_a(n)] \\ &= \zeta. \end{aligned} \tag{3.19}$$

Now, $E\left[e_a(n)e_a^{\Sigma}(n)\right]$, H_U, and H_G are functions of $\tilde{\mathbf{w}_n}$, so that (3.16) becomes

$$\begin{aligned} E[\parallel \tilde{\mathbf{w}_{n+1}} \parallel^2_{\Sigma}] &= (1-\mu\alpha)^2 E[\parallel \tilde{\mathbf{w}_n} \parallel^2_{\Sigma}] + \parallel \mu\alpha\mathbf{c} \parallel^2_{\Sigma} + \\ &\quad \mu^2 tr(\mathbf{R}) H_U - 2\mu(1-\mu\alpha) E[\parallel \tilde{\mathbf{w}_n} \parallel^2_{\mathbf{R}\Sigma}] H_G \\ &\quad + 2\mu\alpha\mathbf{c}^T \Sigma J E[\tilde{\mathbf{w}_n}] \end{aligned} \tag{3.20}$$

We can now use the above relation to study the transient behavior of the proposed Leaky LMMN adaptive algorithm for both white as well as correlated input data. We will now develop a state-space model for both cases.

3.2.3 Transient Analysis for White Input Data

For white input data i.e $\mathbf{R} = \sigma_u^2\mathbf{I}$, using (3.19), we get

$$\begin{aligned} \zeta &= E[\parallel \tilde{\mathbf{w}_n} \parallel^2_{\mathbf{R}}] \\ &= \sigma_u^2 E[\parallel \tilde{\mathbf{w}_n} \parallel^2] \end{aligned} \tag{3.21}$$

From this, we can see that for white input data,

$$\begin{aligned} H_G &= \delta + 3(1-\delta)(\sigma_v^2 + \zeta) \\ &= \delta + 3(1-\delta)(\sigma_v^2 + \sigma_u^2 E[\| \tilde{\mathbf{w}}_n \|^2]) \end{aligned} \tag{3.22}$$

$$\begin{aligned} H_U &= \delta^2[\zeta + \sigma_v^2] + (1-\delta)^2[15\zeta^3 + 45\sigma_v^2\zeta^2 + 15\xi_v^4\zeta + \xi_v^6] \\ &\quad +2\delta(1-\delta)[3\zeta^2 + 4\sigma_v^2\zeta + \xi_v^4] \\ &= \delta^2[\sigma_u^2 E[\| \tilde{\mathbf{w}}_n \|^2] + \sigma_v^2] + (1-\delta)^2[15[\sigma_u^2 E[\| \tilde{\mathbf{w}}_n \|^2]]^3 \\ &\quad +45\sigma_v^2[\sigma_u^2 E[\| \tilde{\mathbf{w}}_n \|^2]]^2 + 15\xi_v^4\sigma_u^2 E[\| \tilde{\mathbf{w}}_n \|^2] + \xi_v^6] \\ &\quad +2\delta(1-\delta)[3[\sigma_u^2 E[\| \tilde{\mathbf{w}}_n \|^2]]^2 + 4\sigma_v^2\sigma_u^2 E[\| \tilde{\mathbf{w}}_n \|^2] + \xi_v^4] \end{aligned} \tag{3.23}$$

$$tr(\mathbf{R}) = M\sigma_u^2 \tag{3.24}$$

$$\mathbf{J} = 1 - \mu\{\alpha + [\delta + 3(1-\delta)(\sigma_v^2 + \sigma_u^2 E[\| \tilde{\mathbf{w}}_n \|^2])\sigma_u^2]\} \tag{3.25}$$

Using (3.3) and (3.22)-(3.25), we can compactly represent the evolution of the $E[\tilde{\mathbf{w}}_n]$ and $E[||\tilde{\mathbf{w}}_n||^2]$ by the following state space equation:

$$\begin{bmatrix} E[||\mathbf{w}_{n+1}||^2] \\ E[\mathbf{w}_{n+1}] \end{bmatrix} = \begin{bmatrix} f_1 & f_2 \\ 0 & \mathbf{J} \end{bmatrix} \begin{bmatrix} E[||\mathbf{w}_n||^2] \\ E[\mathbf{w}_n] \end{bmatrix} + \mu \begin{bmatrix} \mu\alpha^2||c||^2 + \mu M\sigma_u^2\delta^2\sigma_v^2 + \mu M\sigma_u^2(1-\delta)^2\xi_v^6 + 2\mu M\sigma_u^2\delta(1-\delta)\xi_v^4 \\ \alpha c \end{bmatrix}$$

where

$$\begin{aligned}
f_1 &= (1-\mu\alpha)^2 + \mu^2 M\sigma_u^4\delta^2 + 15(1-\delta)^2\mu^2 M\sigma_u^8 E\left[\| \tilde{\mathbf{w}}_n \|^2\right]^2 \\
&\quad +45(1-\delta)^2 M\sigma_u^6\sigma_v^2 E\left[\| \tilde{\mathbf{w}}_n \|^2\right] + 15(1-\delta)^2\mu^2 M\sigma_u^4\xi_v^4 \\
&\quad +6\delta(1-\delta)\mu^2 M\sigma_u^6 E\left[\| \tilde{\mathbf{w}}_n \|^2\right] + 8\delta(1-\delta)\mu^2 M\sigma_u^4\sigma_v^2 - 2\delta\mu(1-\mu\alpha)\sigma_u^2 \\
&\quad -6(1-\delta)\mu(1-\mu\alpha)\sigma_u^2\sigma_v^2 - 6(1-\delta)\mu(1-\mu\alpha)\sigma_u^4 E\left[\| \tilde{\mathbf{w}}_n \|^2\right] \qquad (3.26)
\end{aligned}$$

and

$$f_2 = 2\mu\alpha\mathbf{J}c^T \qquad (3.27)$$

The time evolution of $E[e_a^2(n)]$ can be found using (3.21) and (3.26). The time evolution of $E[e_n^2]$ is then found by using

$$E[e_n^2] = E[e_a^2(n)] + \sigma_n^2. \qquad (3.28)$$

3.2.4 Transient Analysis for Correlated Data

For uncorrelated data, we see from (3.20) that only unweighted norms of $\mathbf{w}_n$ and $\mathbf{w}_{n+1}$ appear on both sides of the equation. However, when the input data is correlated i.e. $\mathbf{R}$ is a non-diagonal matrix, different weighting matrices will appear on both sides of the equation. To solve this problem, we shall start with

(3.20) and for $\sum = \mathbf{I}$, we get

$$E\left[||\mathbf{w}_{n+1}||^2_{\Sigma}\right] = (1-\mu\alpha)^2 E\left[||\mathbf{w}_n||^2_{\Sigma}\right] + ||\mu\alpha\mathbf{c}||^2_{\Sigma} + \mu^2 tr(\mathbf{R\Sigma})H_U$$
$$-2\mu(1-\mu\alpha)H_G E\left[||\mathbf{w}_n||^2_{\mathbf{R}\Sigma}\right] + 2\mu\alpha\mathbf{c}^T\Sigma\mathbf{J}E\left[\mathbf{w}_n\right] \quad (3.29)$$

It can be seen that a weighted norm of $\mathbf{w}_n$ appears with a weighting matrix . This can be inferred from (3.20) for $\sum = \mathbf{R}$, which leads to

$$E\left[||\mathbf{w}_{n+1}||^2_{\Sigma}\right] = (1-\mu\alpha)^2 E\left[||\mathbf{w}_n||^2\right] + ||\mu\alpha\mathbf{c}||^2_{\mathbf{R}} + \mu^2 tr(\mathbf{R})H_U$$
$$-2\mu(1-\mu\alpha)H_G E\left[||\mathbf{w}_n||^2_{\mathbf{R}}\right] + 2\mu\alpha\mathbf{c}^T JE\left[\mathbf{w}_n\right] \quad (3.30)$$

We see that a weighted norm of $\tilde{\mathbf{w}}_n$ appears again, this time with a weighting matrix $\sum = \mathbf{R}^2$, which can then in turn be inferred from (3.20) for $\sum = \mathbf{R}^3$. Continuing in this fashion, (3.20) for $\sum = \mathbf{R}^{M-1}$ becomes

$$E\left[||\mathbf{w}_{n+1}||^2_{\mathbf{R}}\right] = (1-\mu\alpha)^2 E\left[||\mathbf{w}_n||^2_{\mathbf{R}}\right] + ||\mu\alpha\mathbf{c}||^2_{\mathbf{R}} + \mu^2 tr(\mathbf{R^2})H_U$$
$$-2\mu(1-\mu\alpha)H_G E\left[||\mathbf{w}_n||^2_{\mathbf{R^2}}\right] + 2\mu\alpha\mathbf{c}^T\mathbf{R}JE\left[\mathbf{w}_n\right] \quad (3.31)$$

$$E\left[||\mathbf{w}_{n+1}||^2_{\mathbf{R^{M-1}}}\right] = (1-\mu\alpha)^2 E\left[||\mathbf{w}_n||^2_{\mathbf{R^{M-1}}}\right] + ||\mu\alpha\mathbf{c}||^2_{\mathbf{R^{M-1}}} + \mu^2 tr(\mathbf{R^M})H_U$$
$$-2\mu(1-\mu\alpha)H_G E\left[||\mathbf{w}_n||^2_{\mathbf{R^M}}\right]$$
$$+2\mu\alpha\mathbf{c}^T\mathbf{R^{M-1}}\mathbf{J}E\left[\mathbf{w}_n\right] \quad (3.32)$$

where we see now that a weighted norm of $\mathbf{w}_n$ appears again, this time with a weighting matrix $\sum = \mathbf{R}^M$.

Using the Cayley-Hamilton theorem [4], we can write $\mathbf{R}^M$ as

$$\mathbf{R}^M = -p_{M-1}\mathbf{R}^{M-1} - p_{M-2}\mathbf{R}^{M-2} - \ldots - p_1\mathbf{R} - p_0\mathbf{I}, \tag{3.33}$$

where p_0,p_1,. . .,$p_{M-1}$are the coefficients of the characteristic polynomial of $\mathbf{R}$, given as

$$p(x) = \det(x\mathbf{I} - \mathbf{R}). \tag{3.34}$$

Using (3.33), we have

$$E\left[||\mathbf{w}_n||^2_{\mathbf{R}^M}\right] = -p_{M-1}E\left[||\mathbf{w}_n||^2_{\mathbf{R}^{M-1}}\right] - p_{M-2}E\left[||\mathbf{w}_n||^2_{\mathbf{R}^{M-2}}\right] - \ldots - p_1E\left[||\mathbf{w}_n||^2_{\mathbf{R}}\right] - p_0E\left[||\mathbf{w}_n||^2\right]. \tag{3.35}$$

Ultimately, we can combine (3.3) and (3.29)-(3.32) as

$$\underbrace{\begin{bmatrix} \mathbf{A}_{n+1} \\ E\left[\mathbf{w}_{n+1}\right] \end{bmatrix}}_{\mathbf{W}_{n+1}} = \underbrace{\begin{bmatrix} \mathbf{F}_1 & \mathbf{F}_2 \\ \mathbf{0} & \mathbf{J} \end{bmatrix}}_{\mathbf{F_n}} \underbrace{\begin{bmatrix} \mathbf{A}_n \\ E\left[\mathbf{w}_n\right] \end{bmatrix}}_{\mathbf{W}_n} + \mu \underbrace{\begin{bmatrix} \mathbf{L}_n \\ \alpha c \end{bmatrix}}_{\mathbf{Y}_n} \tag{3.36}$$

with $\mathbf{A}_n, \mathbf{L}_n, \mathbf{F}_2, \mathbf{F}_1$ are given as

$$\mathbf{A}_n = \begin{bmatrix} E\left[||\mathbf{w}_n||^2\right] \\ E\left[||\mathbf{w}_n||^2_{\mathbf{R}}\right] \\ E\left[||\mathbf{w}_n||^2_{\mathbf{R}^2}\right] \\ \vdots \\ E\left[||\mathbf{w}_n||^2_{\mathbf{R}^{M-2}}\right] \\ E\left[||\mathbf{w}_n||^2_{\mathbf{R}^{M-1}}\right] \end{bmatrix} \tag{3.37}$$

$$\mathbf{L}_n = \mu H_U \begin{bmatrix} tr(\mathbf{R}) \\ tr(\mathbf{R}^2) \\ tr(\mathbf{R}^3) \\ \vdots \\ tr(\mathbf{R}^M) \end{bmatrix} + \mu\alpha^2 \begin{bmatrix} ||c||^2 \\ ||c||^2_{\mathbf{R}} \\ ||c||^2_{\mathbf{R}^2} \\ \vdots \\ ||c||^2_{\mathbf{R}^{M-1}} \end{bmatrix} \tag{3.38}$$

$$\mathbf{F}_2 = 2\mu\alpha\mathbf{c}^T J \begin{bmatrix} \mathbf{I} \\ \mathbf{R} \\ \mathbf{R}^2 \\ \vdots \\ \mathbf{R}^{M-1} \end{bmatrix} \tag{3.39}$$

where $H_U = \mu M\delta^2\sigma_v^2 + \mu M(1-\delta)^2\xi_v^6 + 2\mu M\delta(1-\delta)\xi_v^4$ where $\mathbf{J}$ comes from (3.17) and

$$
\mathbf{F}_1 = \begin{bmatrix}
k_1 & -k_2 & 0 & 0 & \cdots & 0 \\
0 & k_1 & -k_2 & 0 & \cdots & 0 \\
0 & 0 & \ddots & \ddots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \ddots & 0 \\
0 & 0 & \cdots & 0 & k_1 & -k_2 \\
k_2 p_0 & k_2 p_1 & \cdots & \cdots & k_2 p_{M-2} & k_1 + k_2 p_{M-1}
\end{bmatrix} \tag{3.40}
$$

where

$$k_1 = (1 - \mu\alpha)^2 \tag{3.41}$$

and

$$k_2 = 2\mu(1 - \mu\alpha)\mathcal{H}_G \tag{3.42}$$

From this, we can see that that the evolution of $E[||\mathbf{w}_n||^2]$ and $E[e_{an}^2]$ can be described by the first and second entries of the state vector $\mathbf{w}_{n+1}$, respectively. The resulting learning curve of the filter is then

$$E[e^2(n)] = E[e_a^2(n)] + \sigma_v^2. \tag{3.43}$$

We can also see that for $\mathbf{R} = \sigma_u^2\mathbf{I}$, (3.36) degenerates to (3.26).

3.2.5 Mean Square Stability

As can be seen from the block triangular structure of $\mathbf{F}_n$ in (3.36), we find that one of the conditions for the mean-square stability of the Leaky LMMN algorithm is that it be mean convergent. The mean convergence condition was found before and shown in (3.6). To find the second condition for the mean-square stability of the Leaky LMMN to hold, we will use the same approach as was done for finding the mean convergence on the step size.

Therefore, let $\vartheta \leq \zeta$ be the Cramer-Rao bound associated with estimating $\mathbf{u}_n\mathbf{c}$ by $\mathbf{u}_n\mathbf{w}_n$; then H_G^* and H_U^* are defined as

$$H_G^* = \delta + 3(1-\delta)\{\sigma_v^2 + \vartheta\} \tag{3.44}$$

$$\begin{aligned} H_U^* \quad = \quad & \delta^2[\vartheta + \sigma_v^2] + (1-\delta)^2[15\vartheta^3 + 45\sigma_v^2\vartheta^2 + 15\xi_v^4\vartheta + \xi_v^4] \\ & +2\delta(1-\delta)[3\vartheta^2 + 4\sigma_v^2\vartheta + \xi_v^4] \end{aligned} \tag{3.45}$$

Using this, let us define $\mathbf{F}_1^*$ and $\mathbf{L}^*$ as follows

$$\mathbf{F}_1^* = \mathbf{F}_1|_{\mathcal{H}_G = \mathcal{H}_G^*}, \tag{3.46}$$

$$\mathbf{L}^* = \mathbf{L}_n|_{\mathcal{H}_U = \mathcal{H}_U^*}. \tag{3.47}$$

$\mathbf{F}_1^*$ can then be written as

$$\mathbf{F}_1^* = \mathbf{I} - \mu\mathbf{G}_1 + \mu^2\mathbf{G}_2, \tag{3.48}$$

where

$$\mathbf{G}_1 = 2(\alpha\mathbf{I} + \mathcal{H}_G^*\mathbf{B}), \tag{3.49}$$

and

$$\mathbf{G}_2 = \alpha(\alpha I + 2\mathcal{H}_G^*\mathbf{B}), \tag{3.50}$$

where in (3.49) and (3.50),

$$\mathbf{B} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 \\ -p_0 & -p_1 & \cdots & \cdots & -p_{M-2} & -p_{M-1} \end{bmatrix} \tag{3.51}$$

From [4], a sufficient condition for $\mathbf{F}_1$ to be stable, and thus constitute the second condition for the mean square stability of the proposed algorithm is that the step size lies in the following range:

$$0 < \mu < \frac{1}{\lambda_{\max}(\mathbf{G}_1^{-1}\mathbf{G}_2)} \tag{3.52}$$

Combining (3.6) and (3.52), we find that the condition for $\mathbf{w}_n$ to converge in both the mean and mean square sense is

$$0 < \mu < \min\left(\frac{2}{\alpha + \mathcal{H}_{\mathcal{G}}^*\lambda_{max}\mathbf{R}}, \frac{1}{\lambda_{max}(G_1^{-1}G_2)}\right) \tag{3.53}$$

To be more explicit, we first note from (3.51) that $\mathbf{B}$ is a companion form matrix of $\mathbf{R}$. Therefore it has the same eigenvalues as $\mathbf{R}$. Let λ_i,λ_i', and λ_i'' be the i^{th}

eigenvalues of $\mathbf{B}$, $\mathbf{G}_1$, and $\mathbf{G}_2$, repectively. Then, from using the matrix eigenvalue properties [28] and (3.49)-(3.51), the relations between them are given as,

$$\begin{aligned} \lambda_i' &= 2(\alpha + \mathcal{H}_{\mathcal{G}}{}^* \lambda_i), \\ \lambda_i'' &= \alpha(\alpha + 2\mathcal{H}_{\mathcal{G}}{}^* \lambda_i). \end{aligned}$$

Furthermore, $\mathbf{G}_1$, $\mathbf{G}_2$ and $\mathbf{B}$ will have the same eigenvectors.

Using this, we find that the i^{th} eigenvalue of $\mathbf{G}_1^{-1}\mathbf{G}_2$ is given by

$$\begin{aligned} \lambda_i^{\mathbf{G}_1^{-1}\mathbf{G}_2} &= \frac{\lambda_i''}{\lambda_i'} \\ &= \frac{\alpha(\alpha + 2\mathcal{H}_{\mathcal{G}}{}^* \lambda_i)}{2(\alpha + \mathcal{H}_{\mathcal{G}}{}^* \lambda_i)} \\ &= \alpha - \frac{\alpha^2}{2(\alpha + \mathcal{H}_{\mathcal{G}}{}^* \lambda_i)}. \end{aligned} \tag{3.54}$$

Furthermore

$$\begin{aligned} \lambda_{\max}^{\mathbf{G}_1^{-1}\mathbf{G}_2} &= \alpha - \frac{\alpha^2}{2(\alpha + \mathcal{H}_{\mathcal{G}}{}^* \lambda_{\max}(\mathbf{R}))} \\ &= \frac{\alpha(\alpha + 2\mathcal{H}_{\mathcal{G}}{}^* \lambda_{\max}(\mathbf{R}))}{2(\alpha + \mathcal{H}_{\mathcal{G}}{}^* \lambda_{\max}(\mathbf{R}))}, \end{aligned} \tag{3.55}$$

and

$$\frac{1}{\lambda_{\max}^{\mathbf{G}_1^{-1}\mathbf{G}_2}} = \frac{2(\alpha + \mathcal{H}_{\mathcal{G}}{}^* \lambda_{\max}(\mathbf{R}))}{\alpha(\alpha + 2\mathcal{H}_{\mathcal{G}}{}^* \lambda_{\max}(\mathbf{R}))} \tag{3.56}$$

Now, by comparing (3.6) and (3.56) and after some algebraic manipulation, we get the following result for the upper bound $\mu_{\max}$ on the step size to ensure mean and mean square stability:

$$\mu_{\max}=\begin{cases} \frac{2}{\alpha+\mathcal{H}_G^*\lambda_{\max}(\mathbf{R})}, & \alpha > \frac{\mathcal{H}_G^*\lambda_{\max}(\mathbf{R})}{4}, \\ \frac{1}{\lambda_{\max}(\mathbf{G}_1^{-1}\mathbf{G}_2)}, & \text{otherwise.} \end{cases} \tag{3.57}$$

3.3 Comparison of the Theoretical and Simulation Results For Transient Analysis

In this section we will compare the theoritical findings relating to the transient analysis of the proposed Leaky LMMN algorithm with that to the simulation results. For the specific purpose we generate a randomly normalized weight vector with number of taps set at 5. Keeping the step size and the leakage factor at 0.01 and 0.001 respectively, and the number of trials at 500 while the number of samples used were set at 10^4.

The transient analysis was carried out for two cases:

1. White input data, where the variance of the regressors was set to unity.

2. Correlated input data, where the eigen value spread of the regressor covariance matrix was set to 5.

Now keeping in view of this two variations in the input data and maintaining all other parameters the same, the simulations were performed for Uniform and Gaussian noise environments with the noise variance value set at 0.1, 0.01 and 0.001. The theoretical curves were generated by using (3.36).

As we can see from the Fig.3.1-3.24, there is a very good match between theory and simulation results.

We can see that the rate of convergence is must more in a given noise environment i.e. type of noise and variance value, for white data as compared to correlated data. The reason for this is that the increase in the eigen spread value of **R** decreases the speed of convergence [4].

We also note that for the same nature of input data i.e. correlated or white, and noise variance, the MSE performance of

the Leaky LMMN is much better in uniform noise than gaussian noise. This is to be expected as the conventional LMMN also performs better in non-gaussian noise scenarios [15],[9].

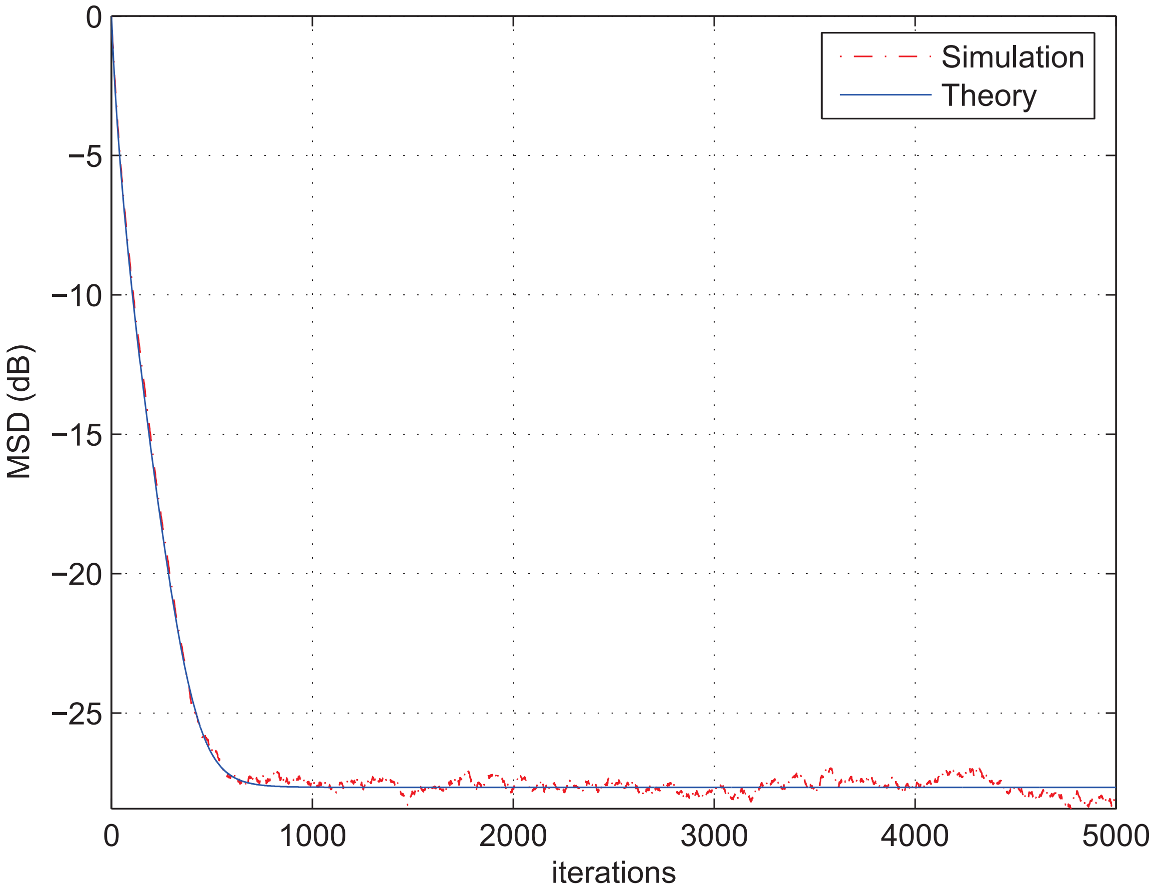

Figure 3.1: Leaky LMMN MSD in Gaussian noise with white data and noise variance 0.1.

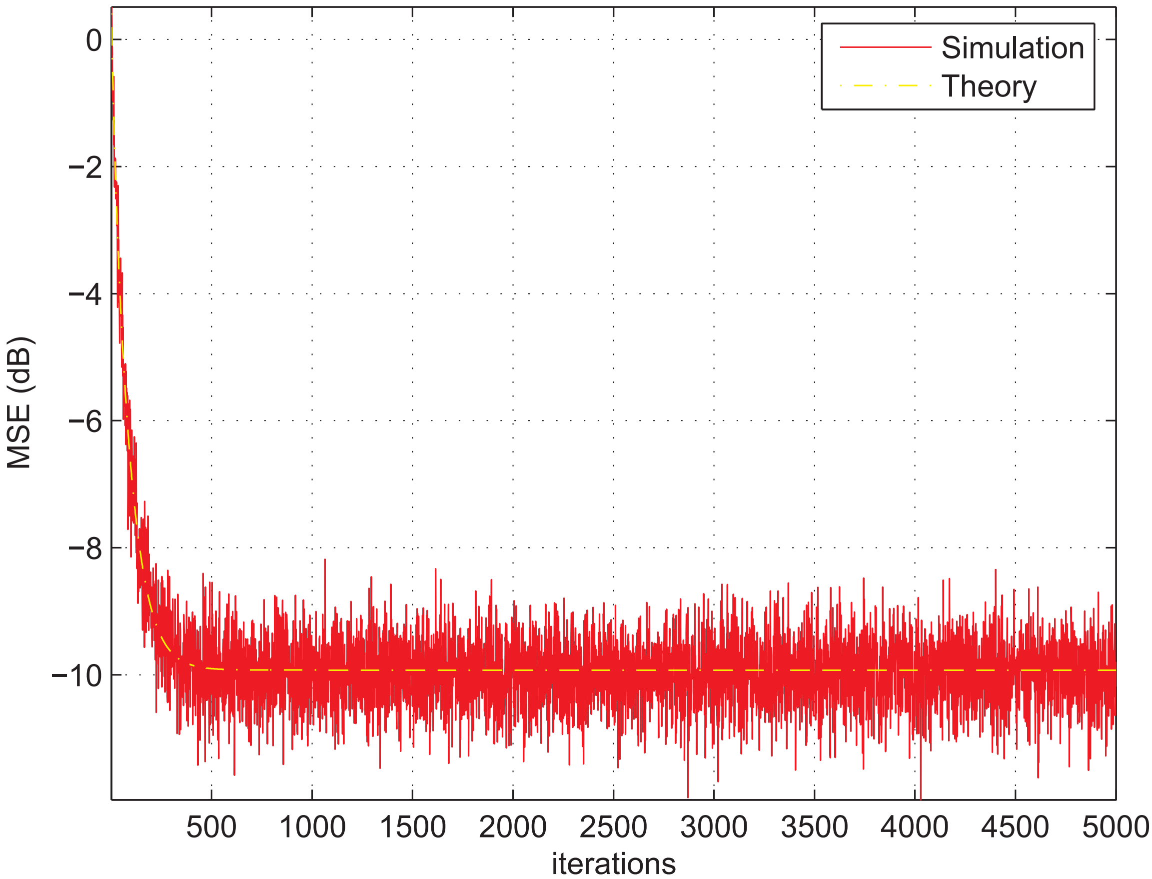

Figure 3.2: Leaky LMMN MSE in Gaussian noise with white data and noise variance 0.1.

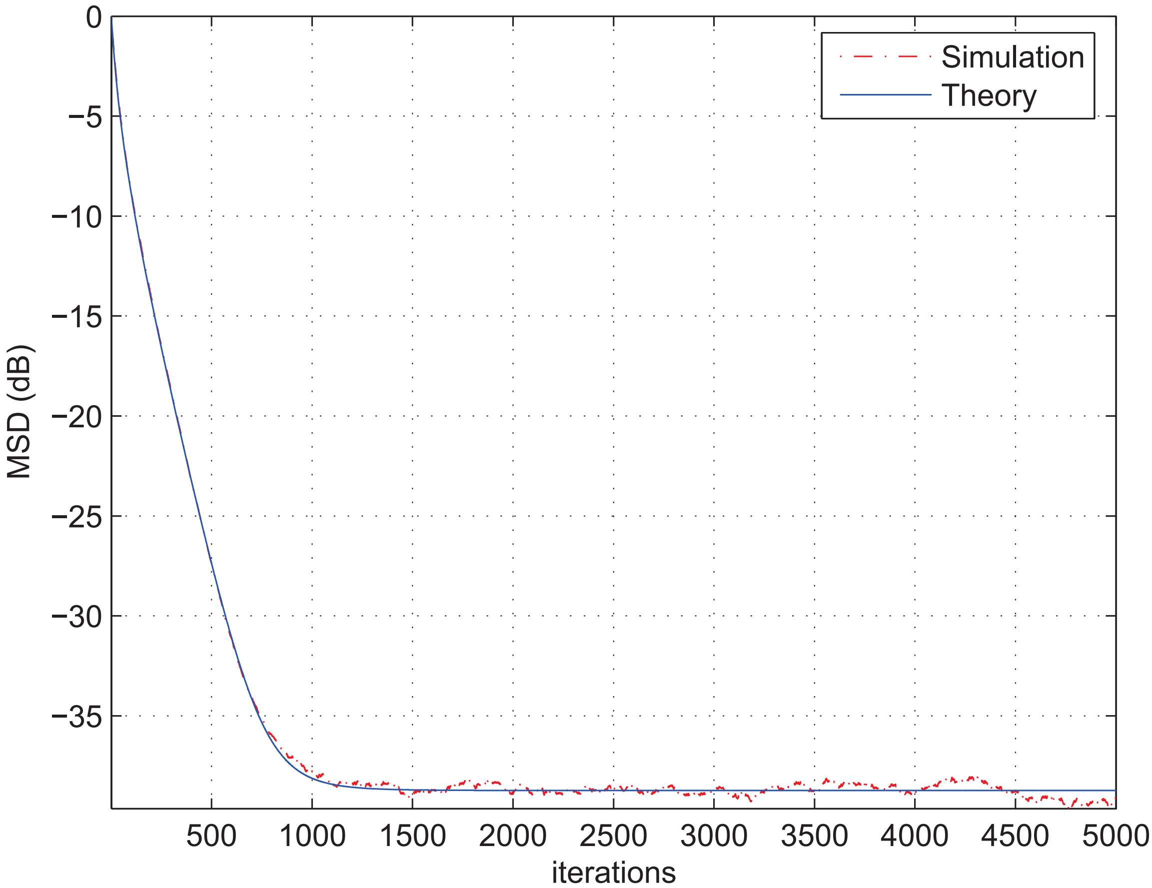

Figure 3.3: Leaky LMMN MSD in Gaussian noise with white data and noise variance 0.2.

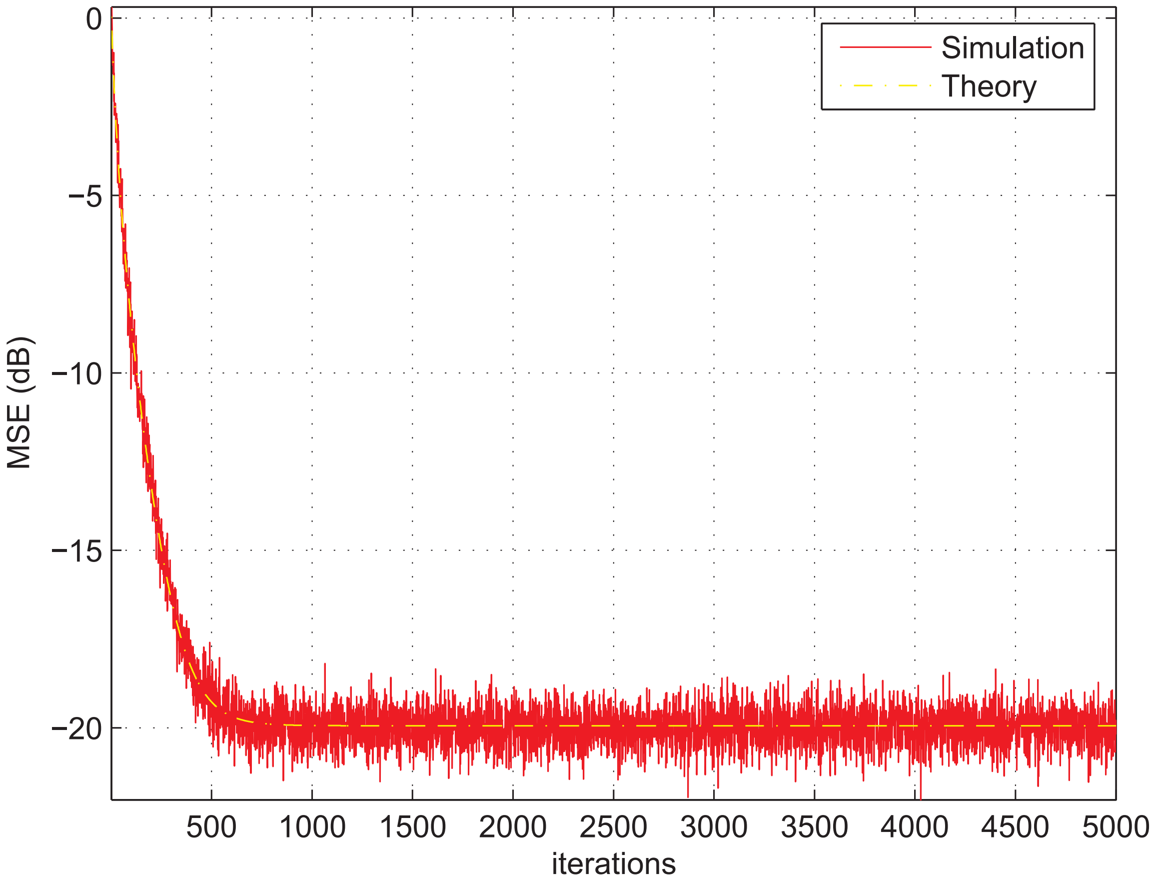

Figure 3.4: Leaky LMMN MSE in Gaussian noise with white data and noise variance 0.2.

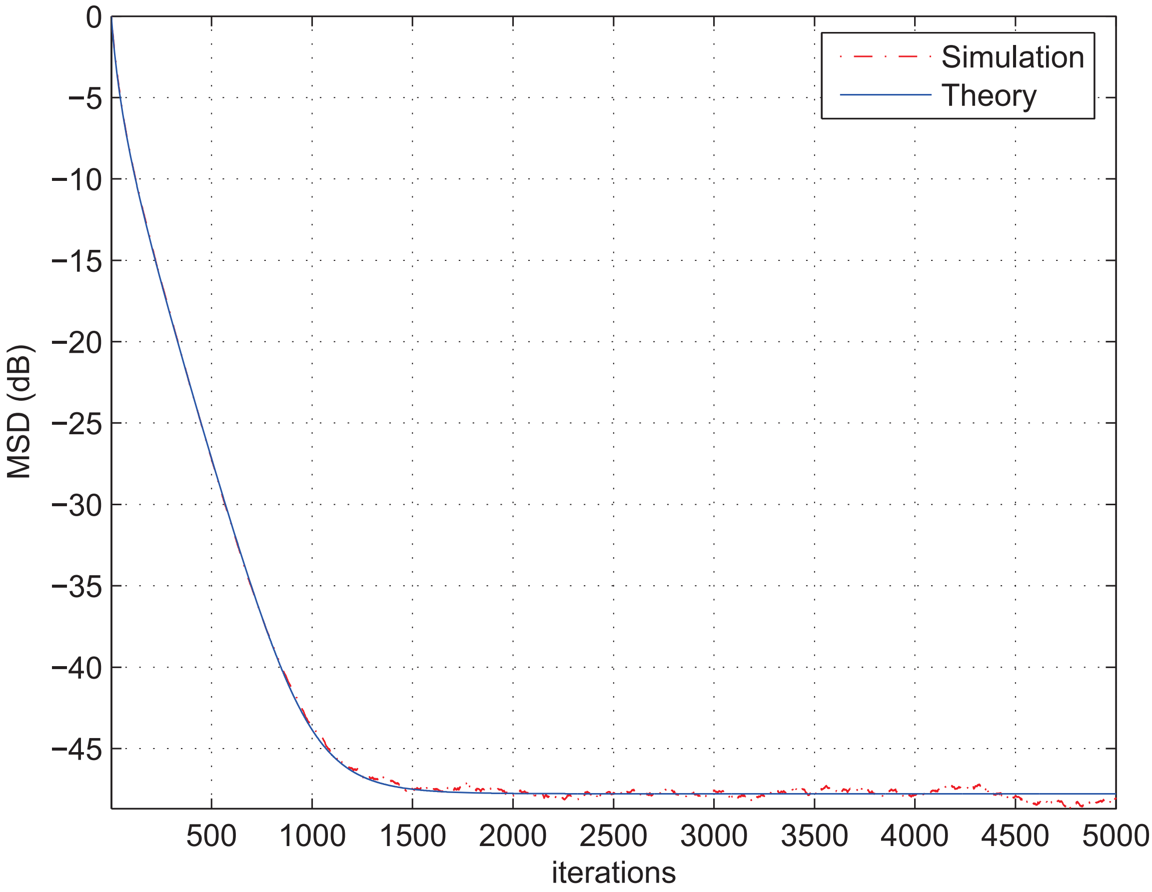

Figure 3.5: Leaky LMMN MSD in Gaussian noise with white data and noise variance 0.3.

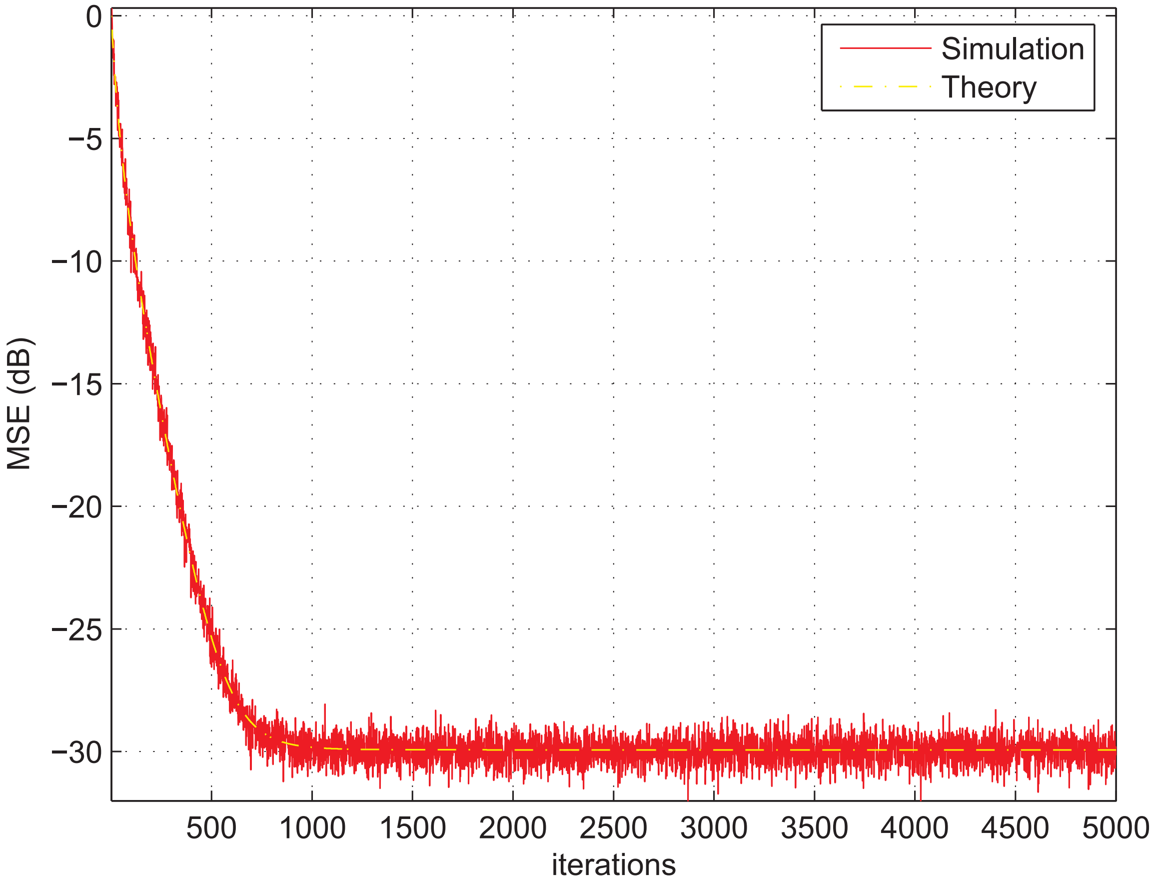

Figure 3.6: Leaky LMMN MSE in Gaussian noise with white data and noise variance 0.3.

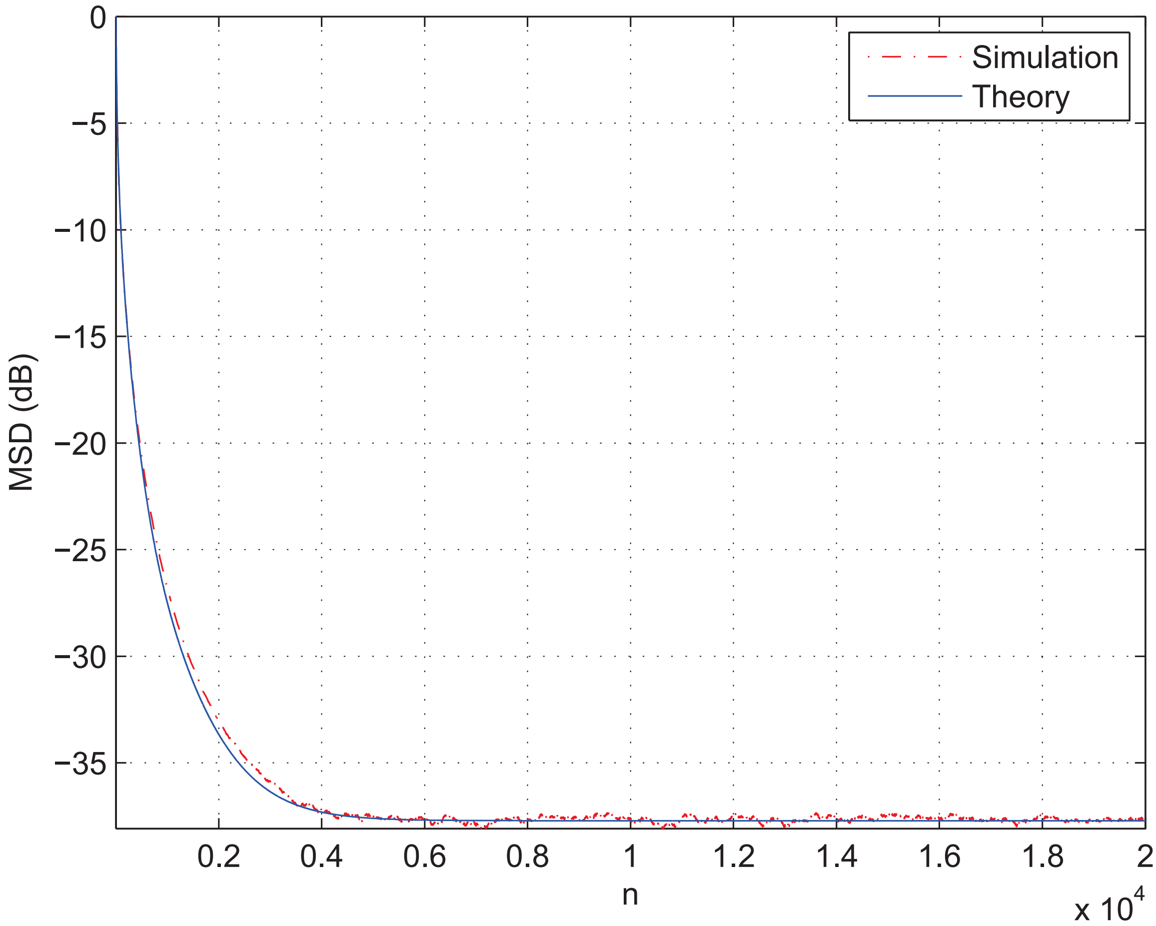

Figure 3.7: Leaky LMMN MSD in Gaussian noise with correlated data and noise variance 0.1.

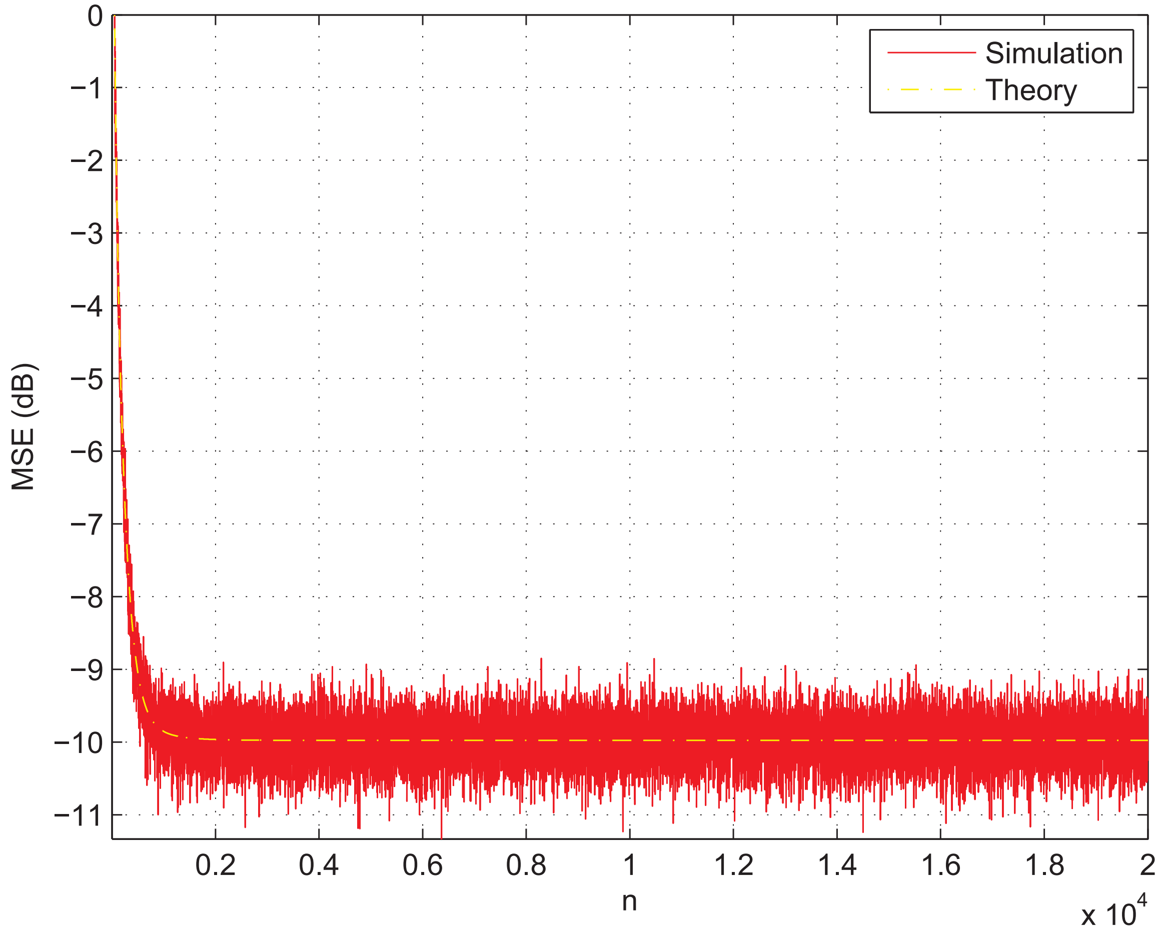

Figure 3.8: Leaky LMMN MSE in Gaussian noise with correlated data and noise variance 0.1.

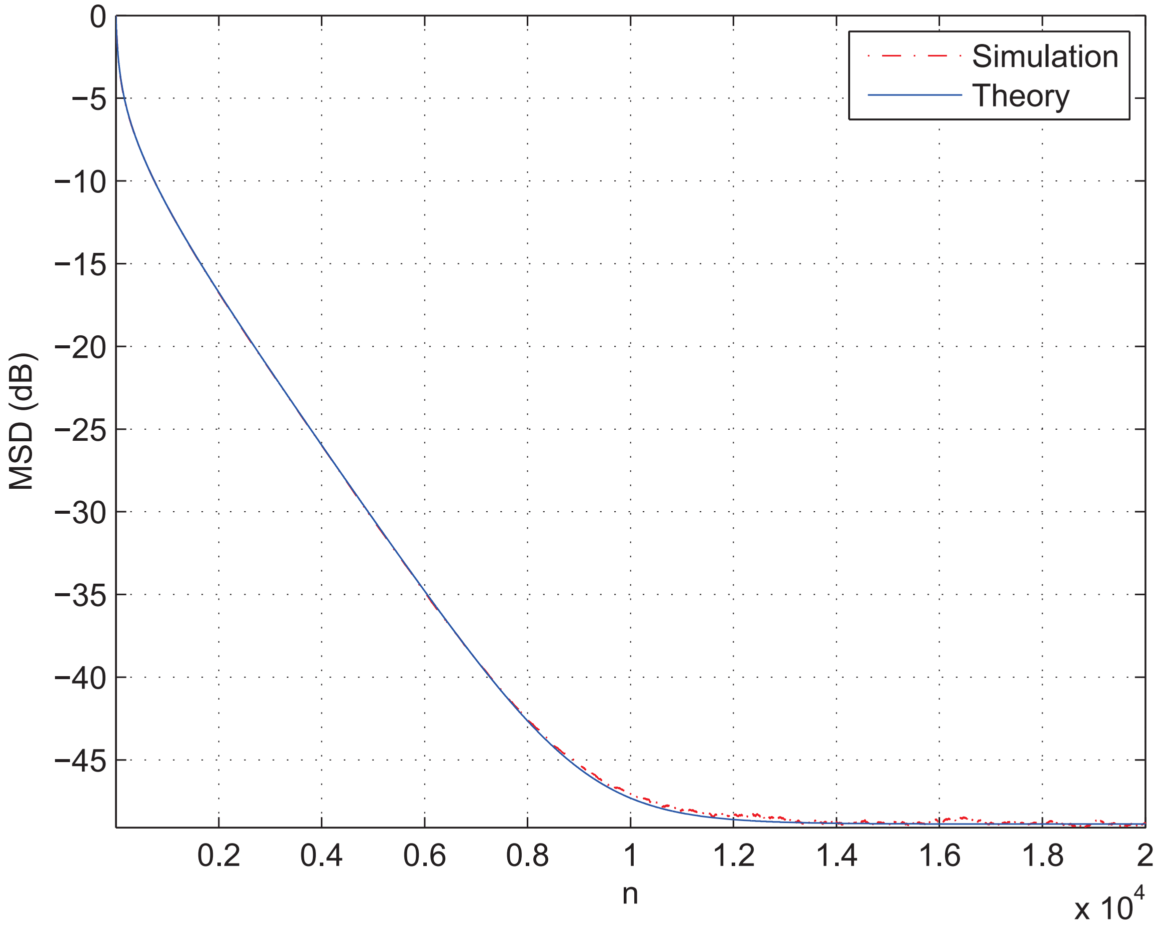

Figure 3.9: Leaky LMMN MSD in Gaussian noise with correlated data and noise variance 0.2.

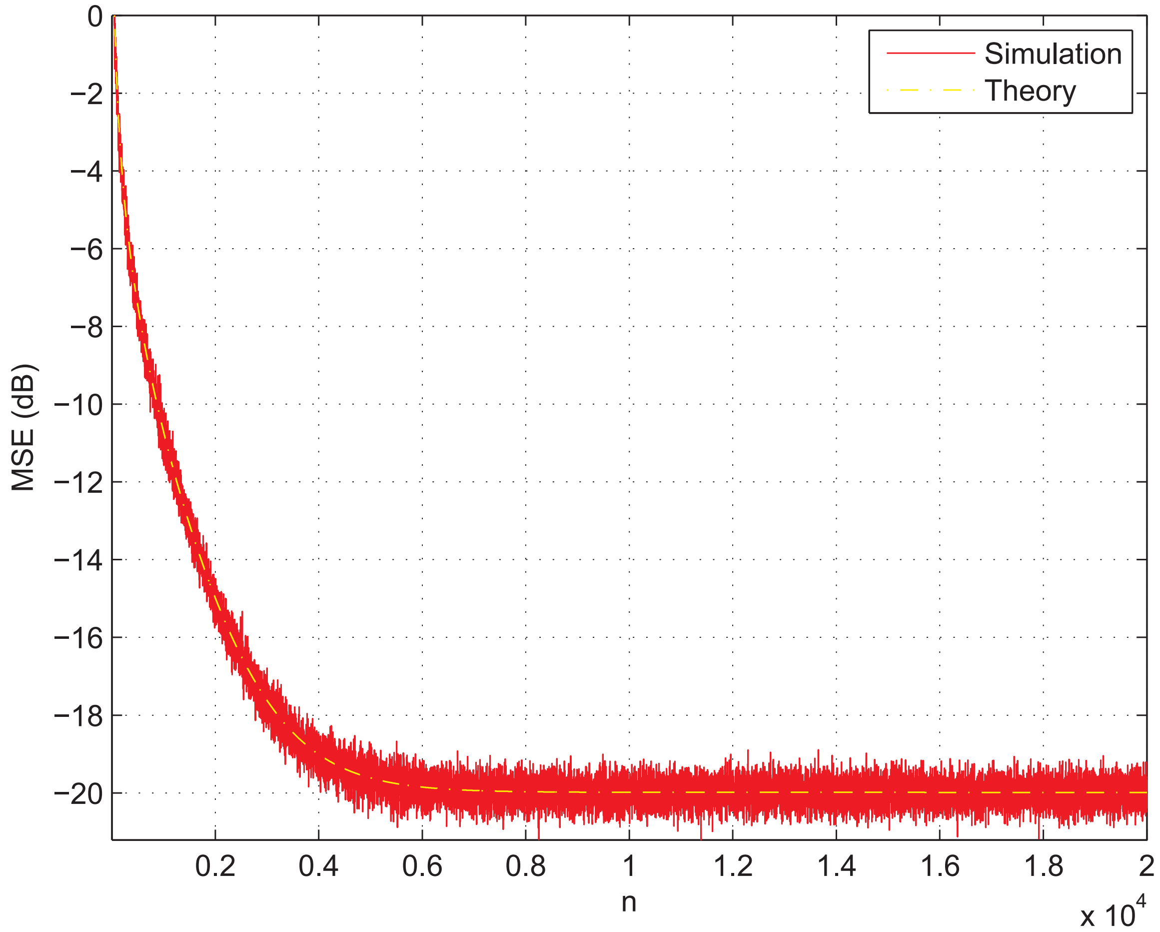

Figure 3.10: Leaky LMMN MSE in Gaussian noise with correlated data and noise variance 0.2.

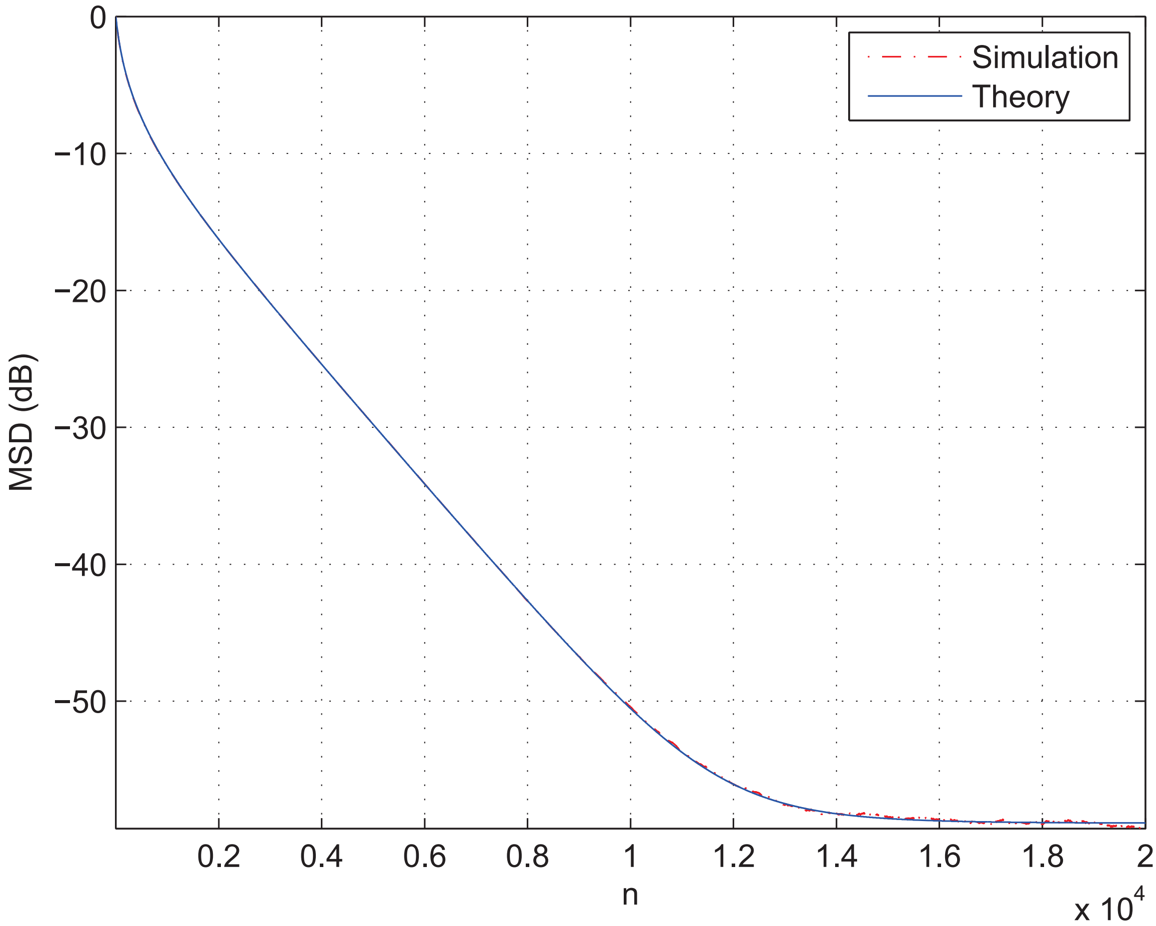

Figure 3.11: Leaky LMMN MSD in Gaussian noise with correlated data and noise variance 0.3.

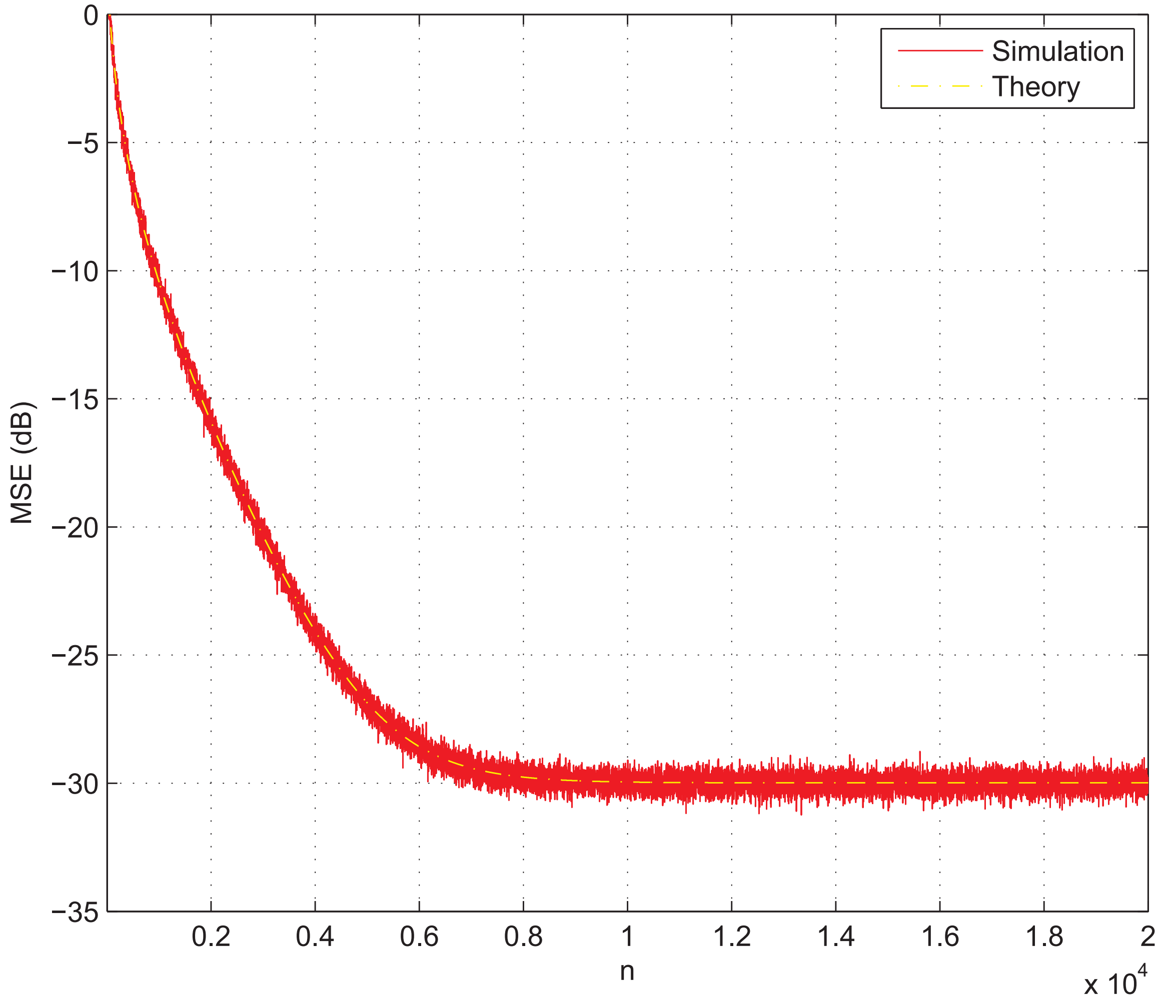

Figure 3.12: Leaky LMMN MSE in Gaussian noise with correlated data and noise variance 0.3.

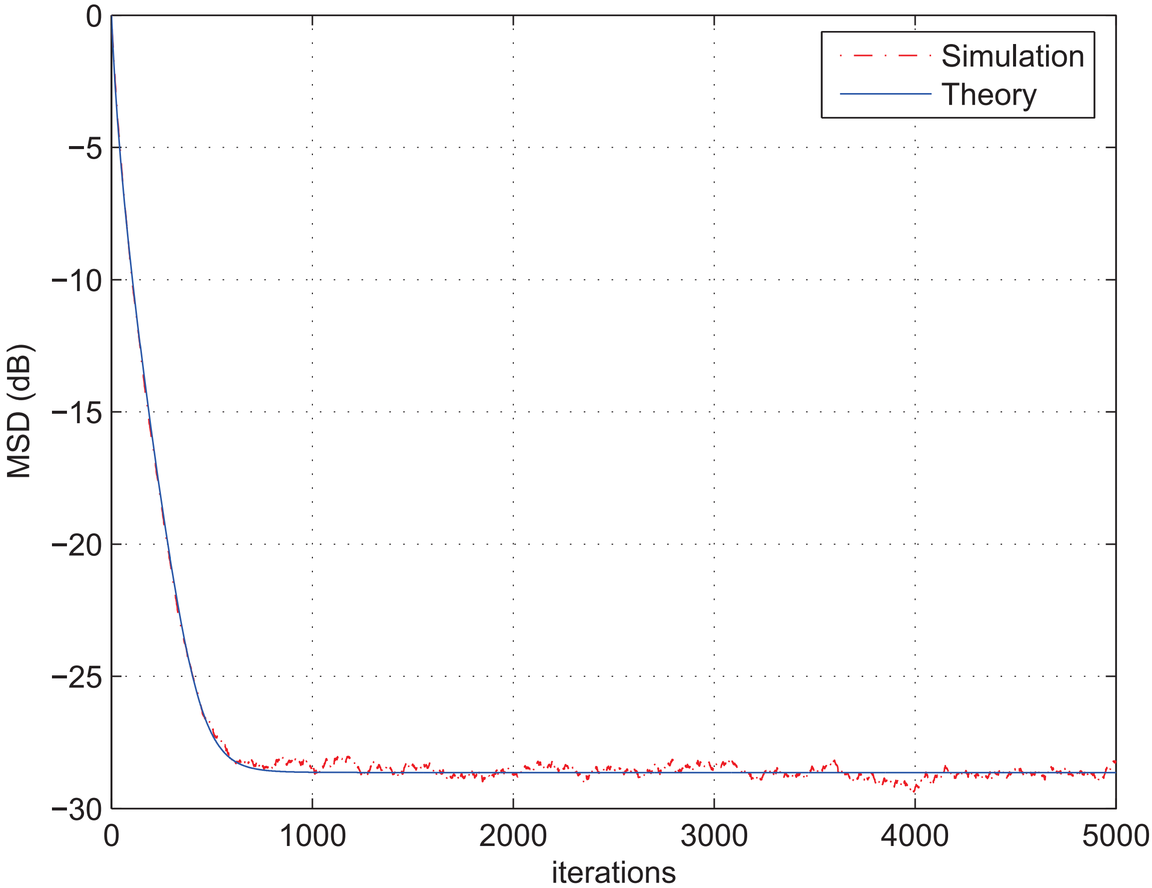

Figure 3.13: Leaky LMMN MSD in Uniform noise with white data and noise variance 0.1.

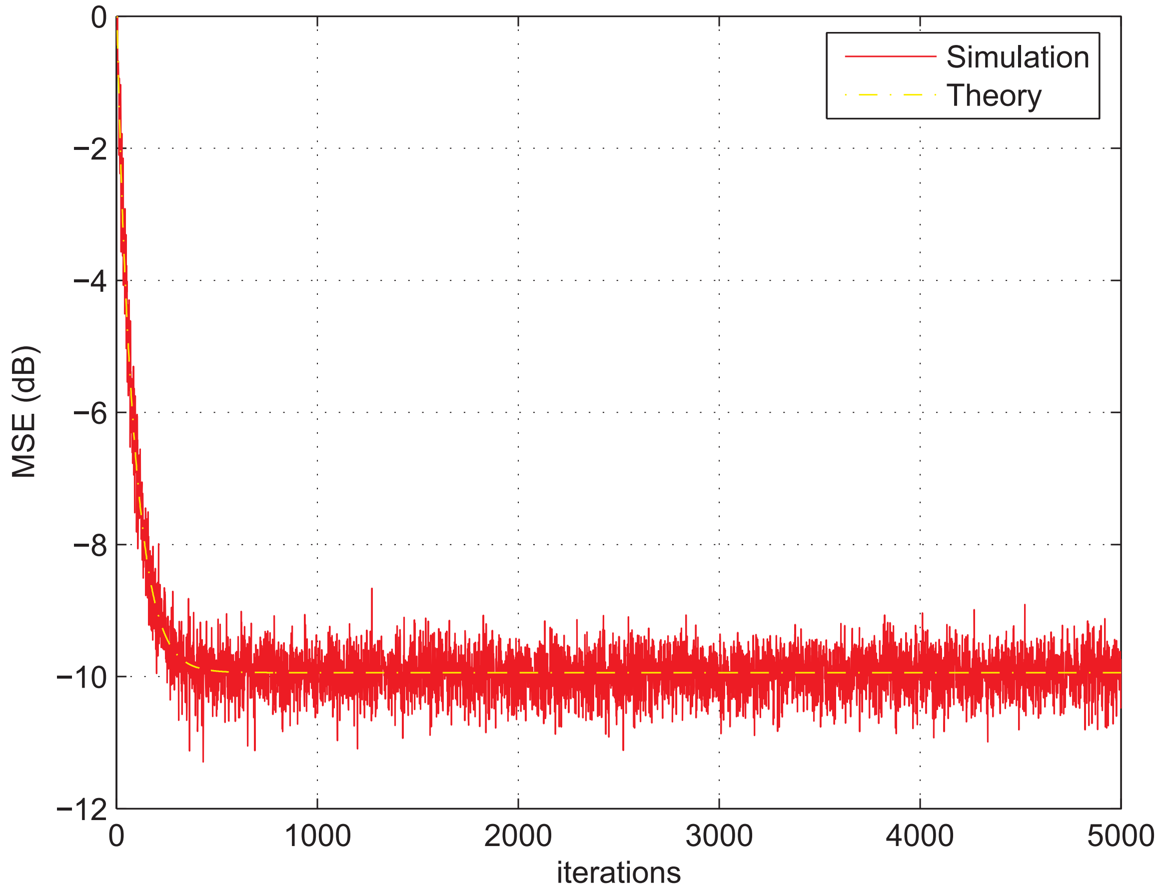

Figure 3.14: Leaky LMMN MSE in Uniform noise with white data and noise variance 0.1.

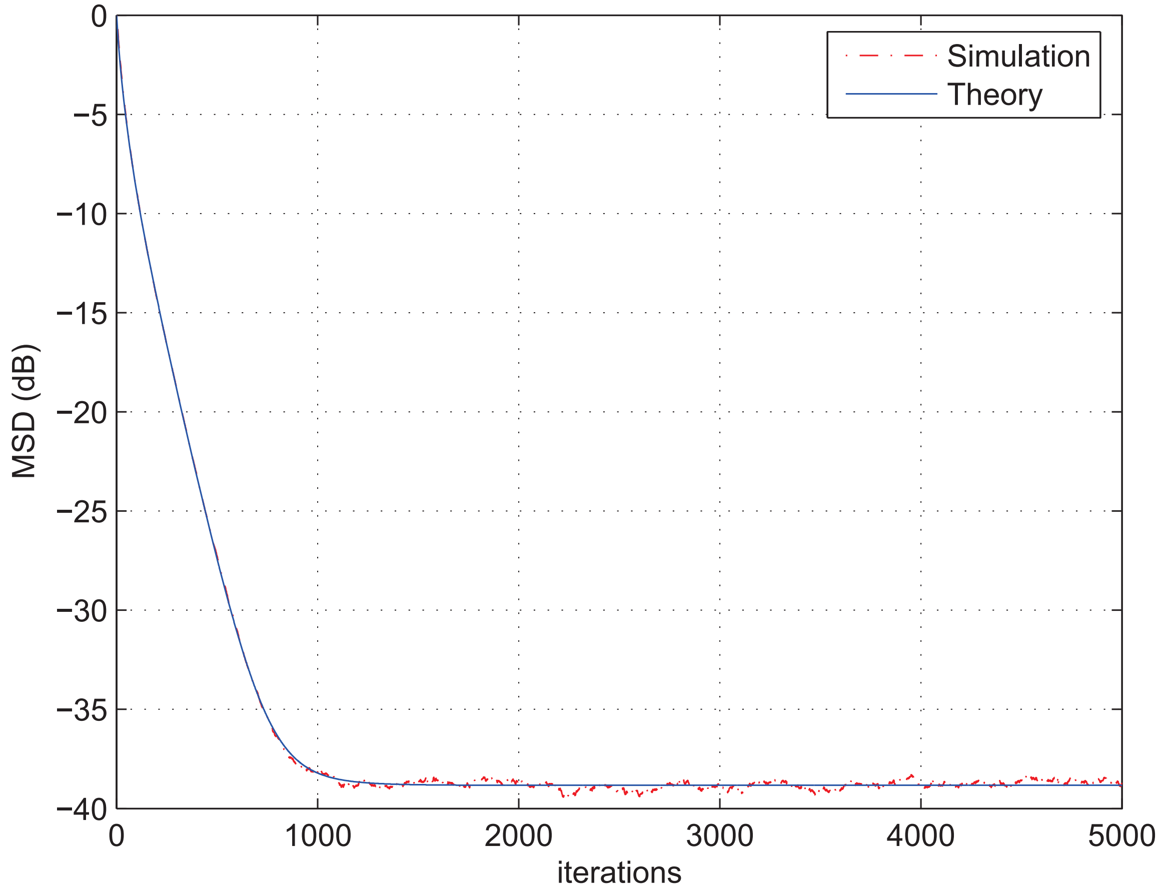

Figure 3.15: Leaky LMMN MSD in Uniform noise with white data and noise variance 0.2.

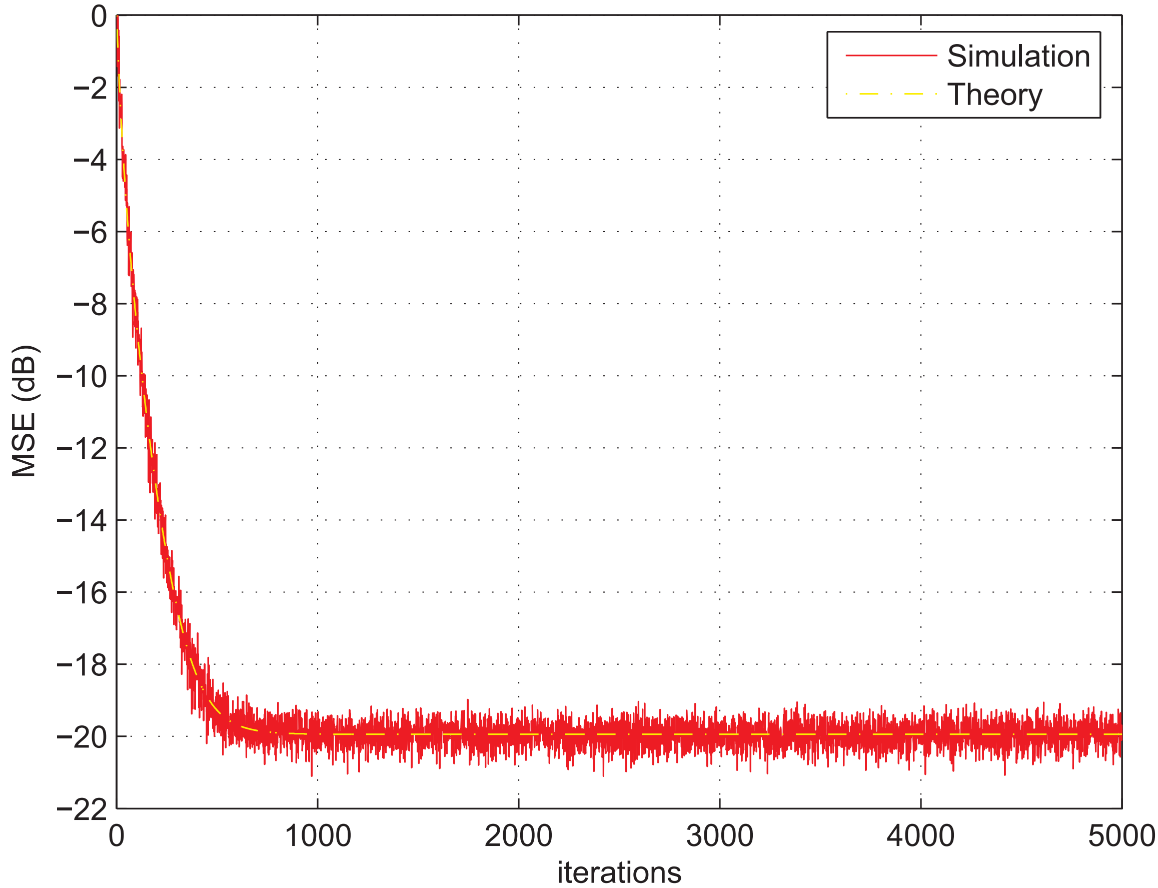

Figure 3.16: Leaky LMMN MSE in Uniform noise with white data and noise variance 0.2.

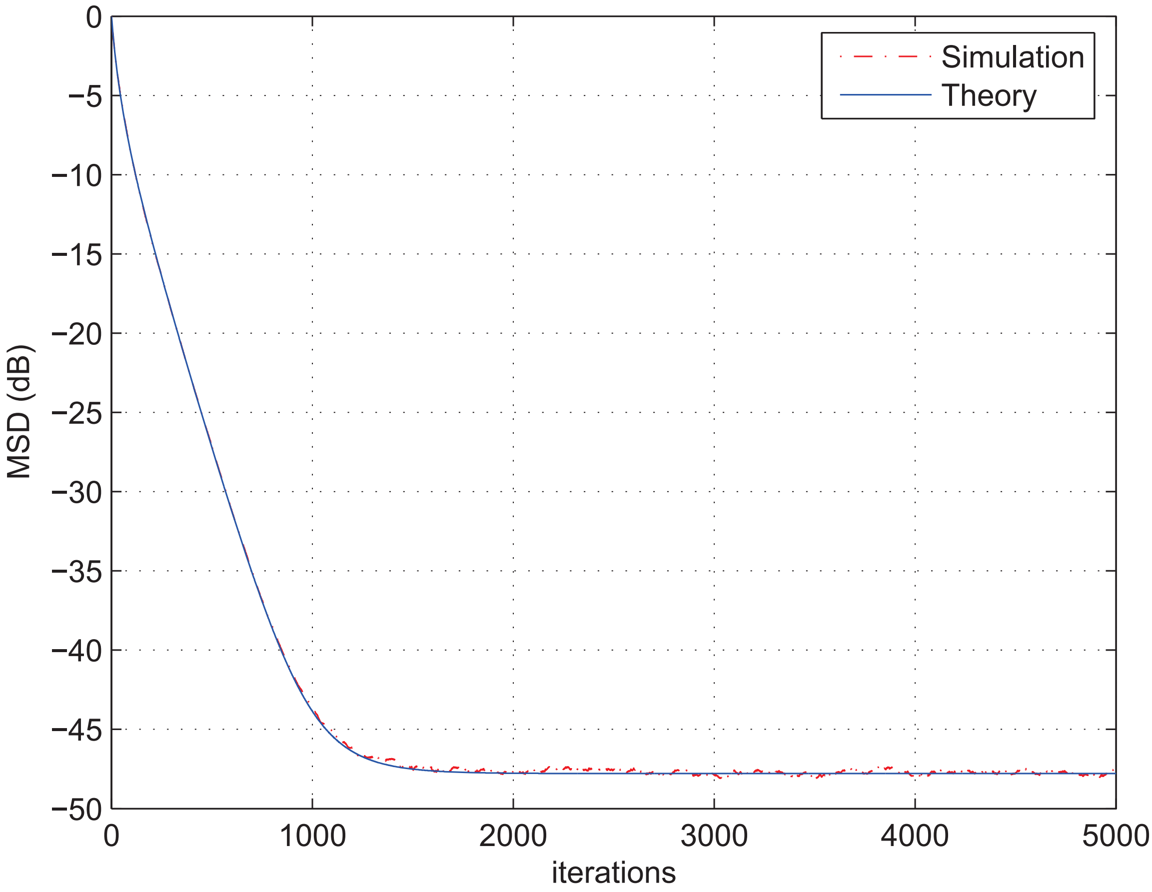

Figure 3.17: Leaky LMMN MSD in Uniform noise with white data and noise variance 0.3.

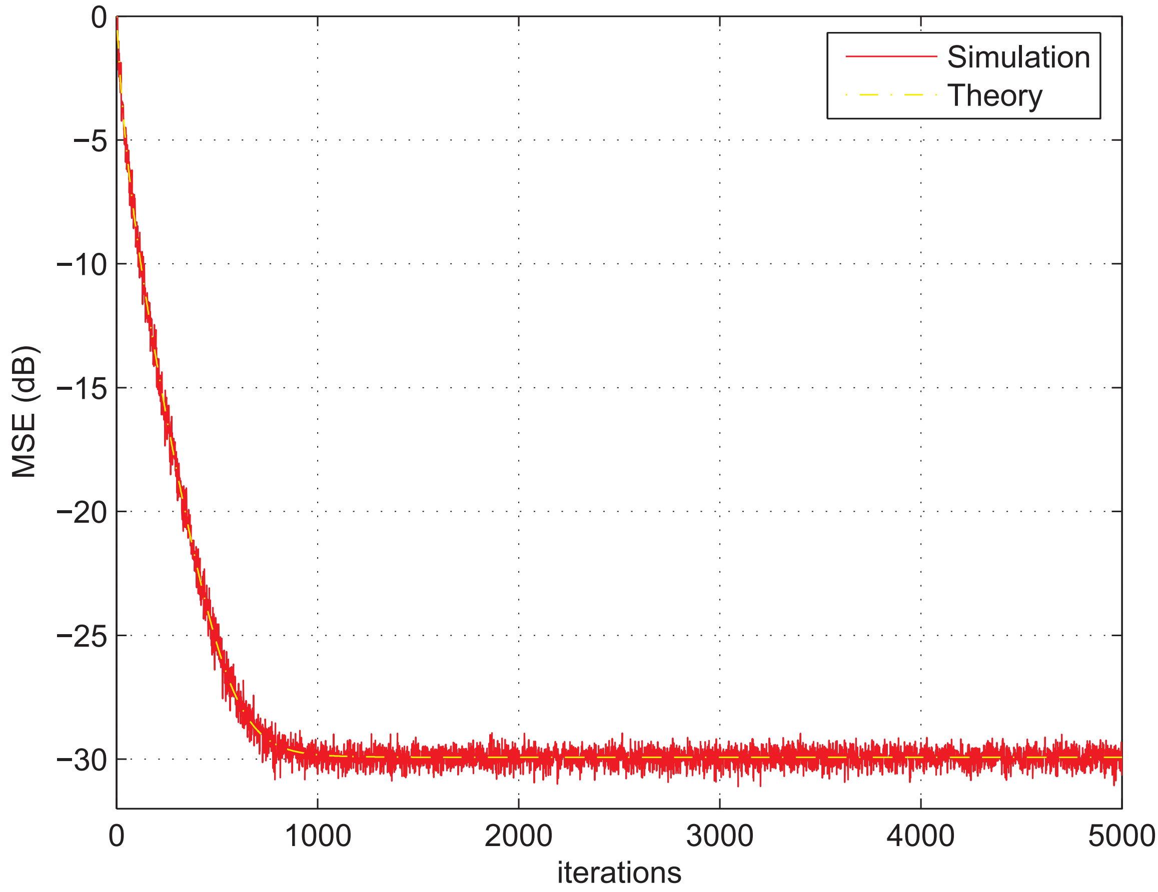

Figure 3.18: Leaky LMMN MSE in Uniform noise with white data and noise variance 0.3.

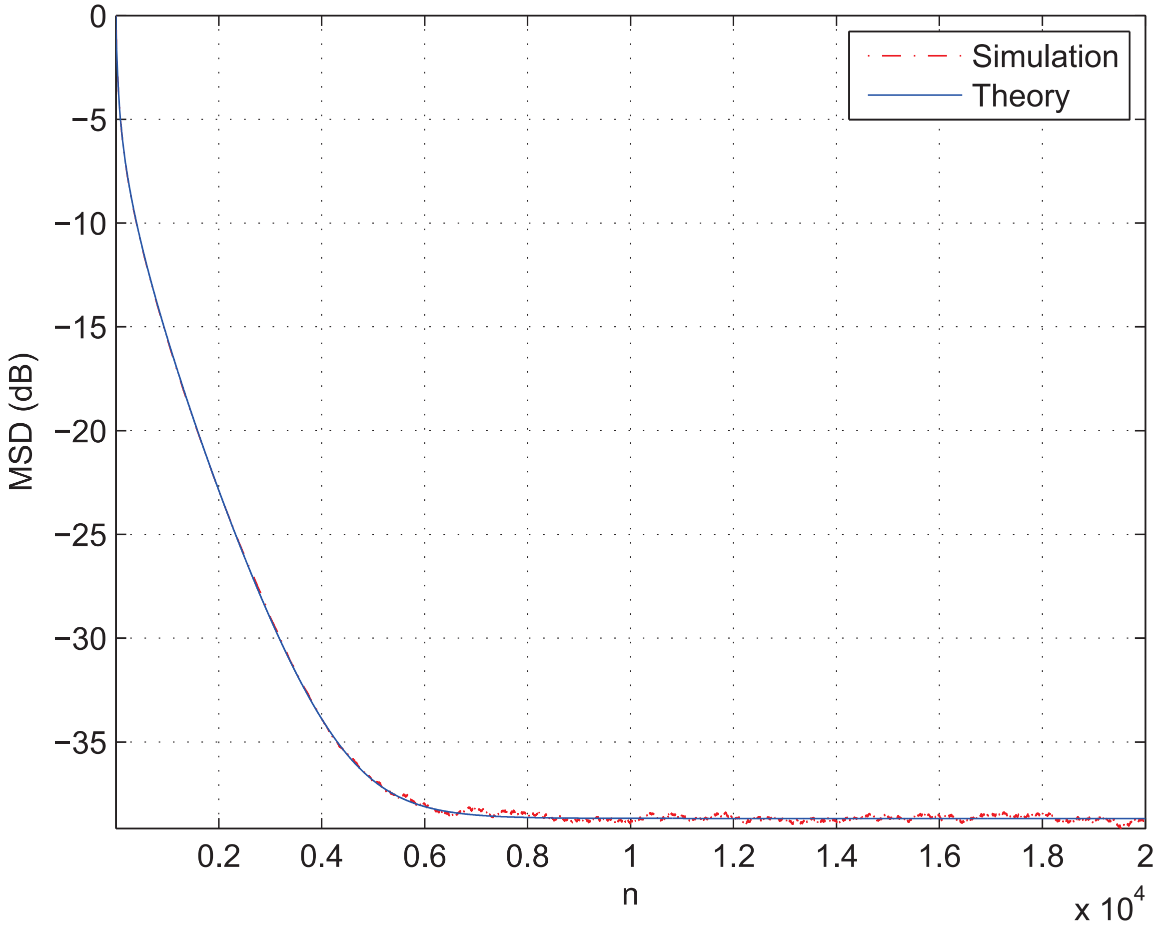

Figure 3.19: Leaky LMMN MSD in Uniform noise with correlated data and noise variance 0.1.

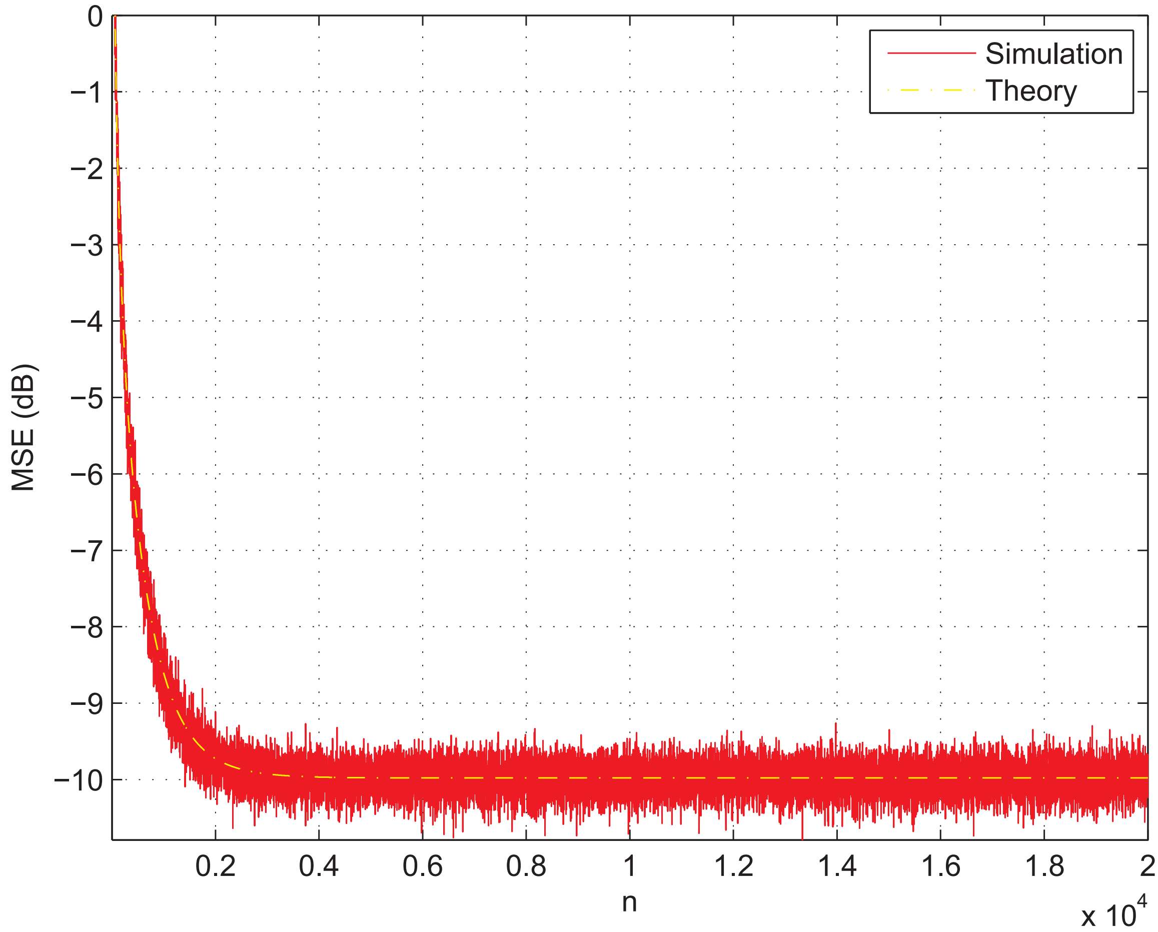

Figure 3.20: Leaky LMMN MSE in Uniform noise with correlated data and noise variance 0.1.

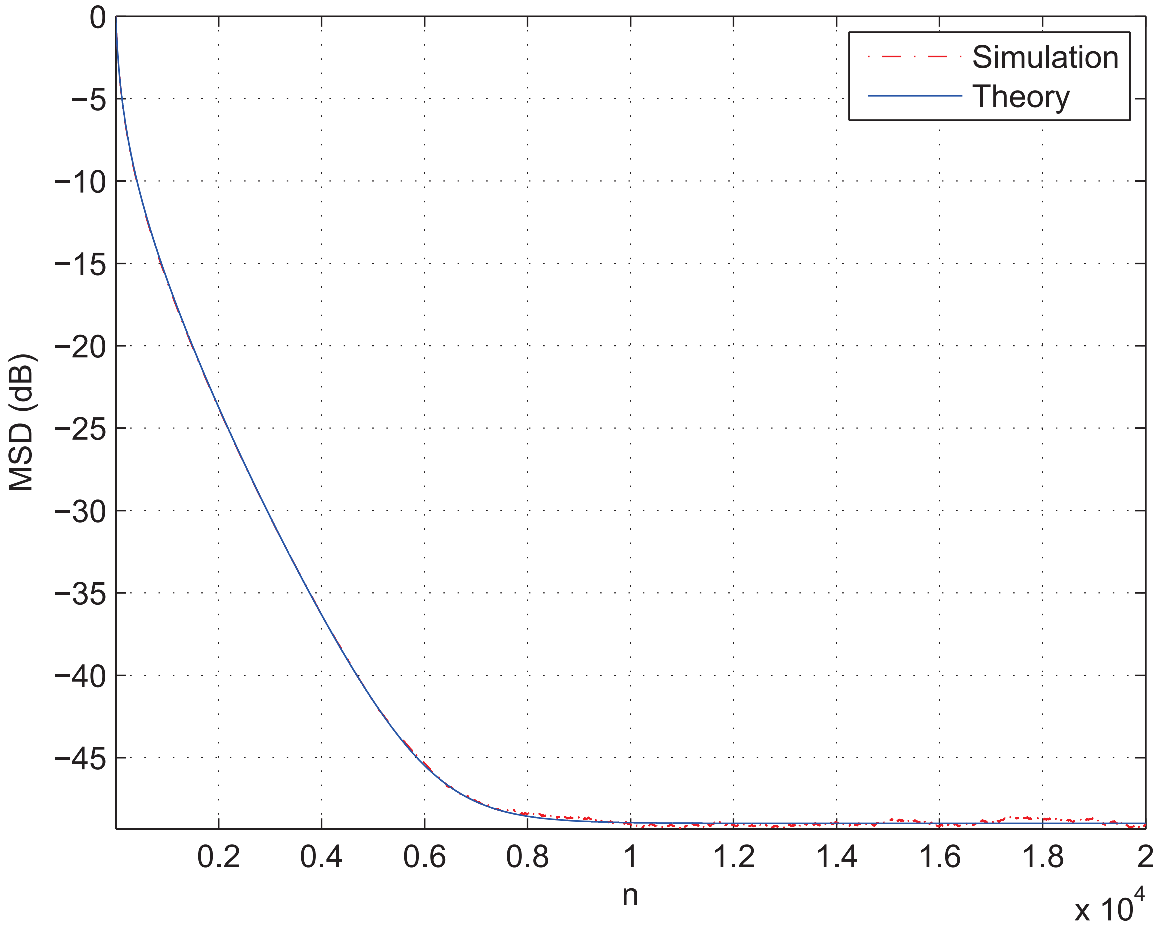

Figure 3.21: Leaky LMMN MSD in Uniform noise with correlated data and noise variance 0.2.

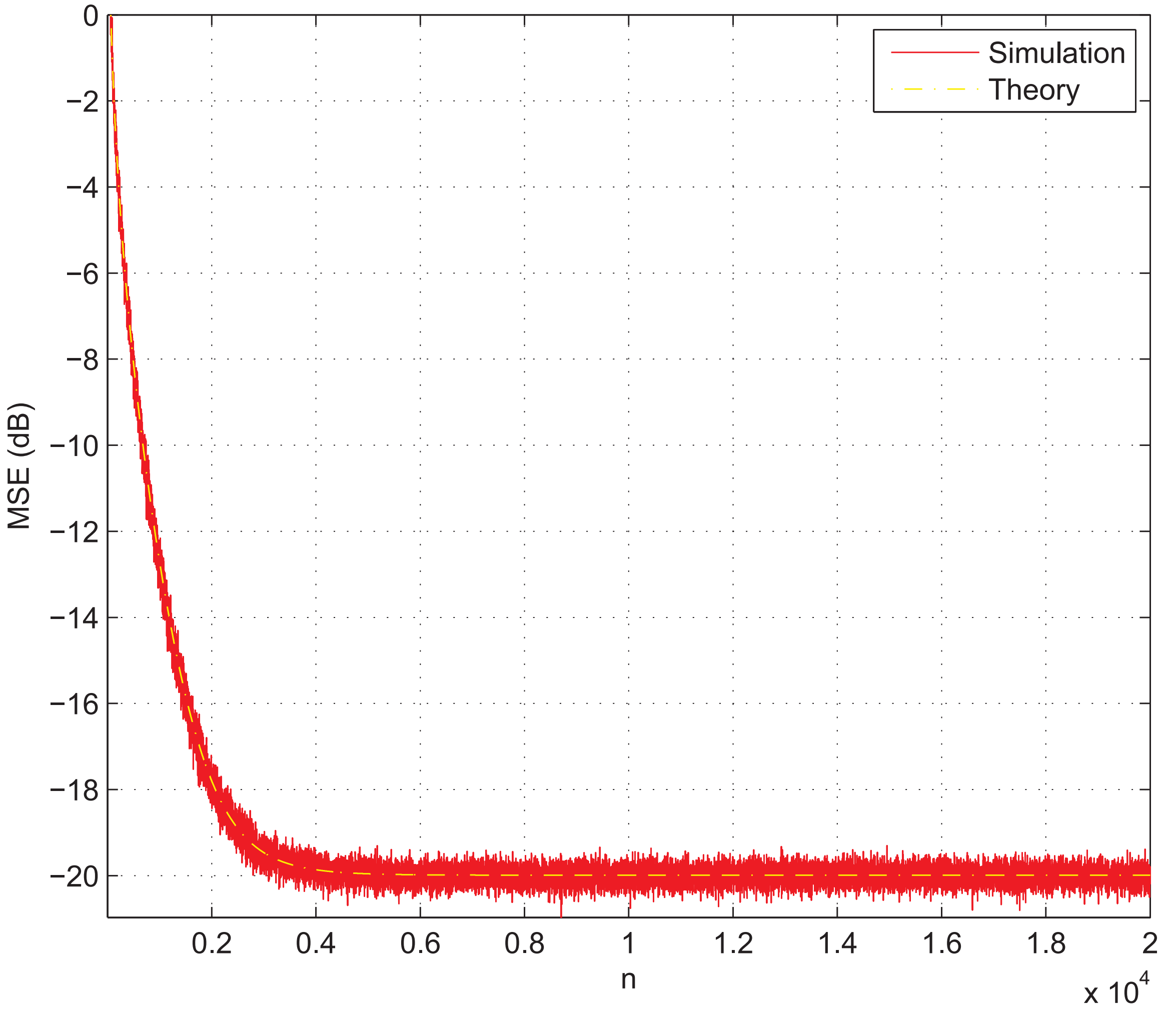

Figure 3.22: Leaky LMMN MSE in Uniform noise with correlated data and noise variance 0.2.

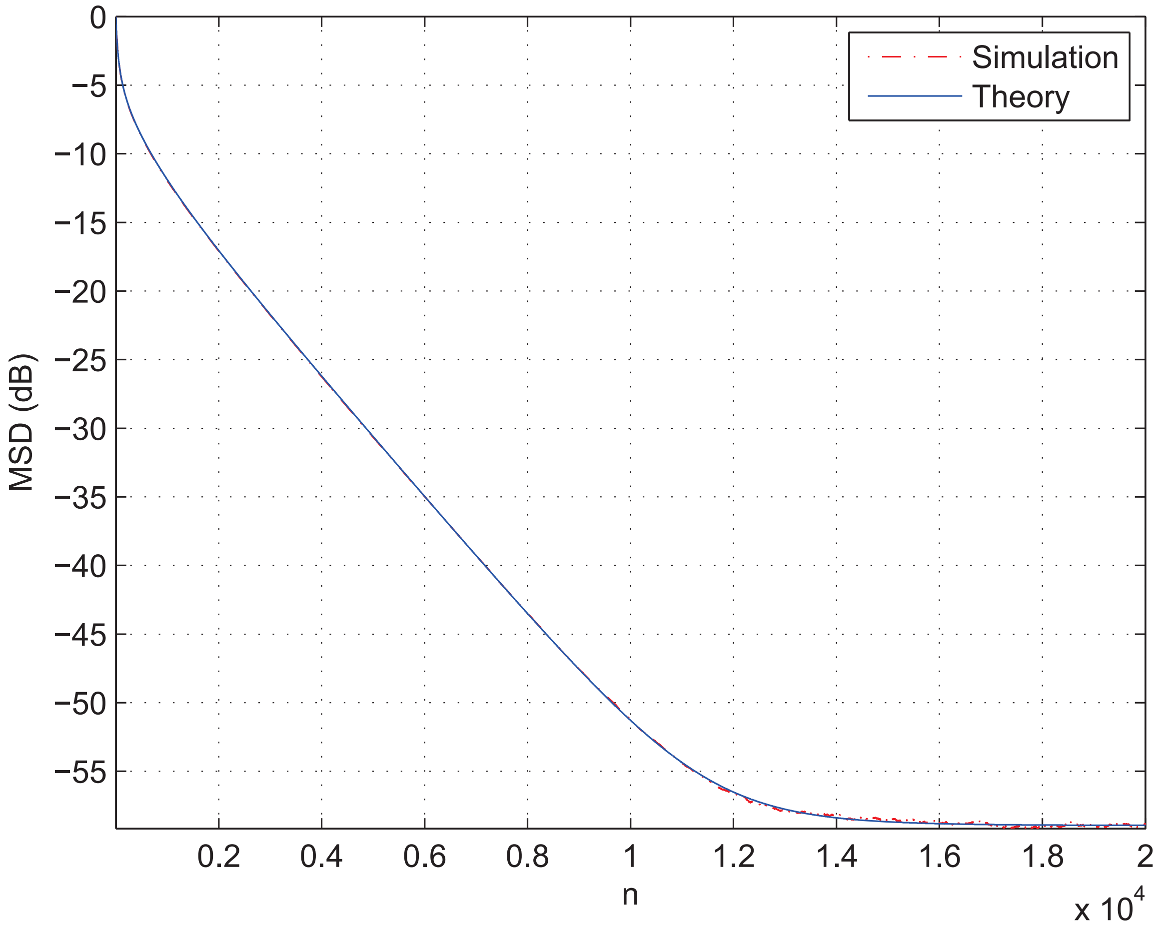

Figure 3.23: Leaky LMMN MSD in Uniform noise with correlated data and noise variance 0.3.

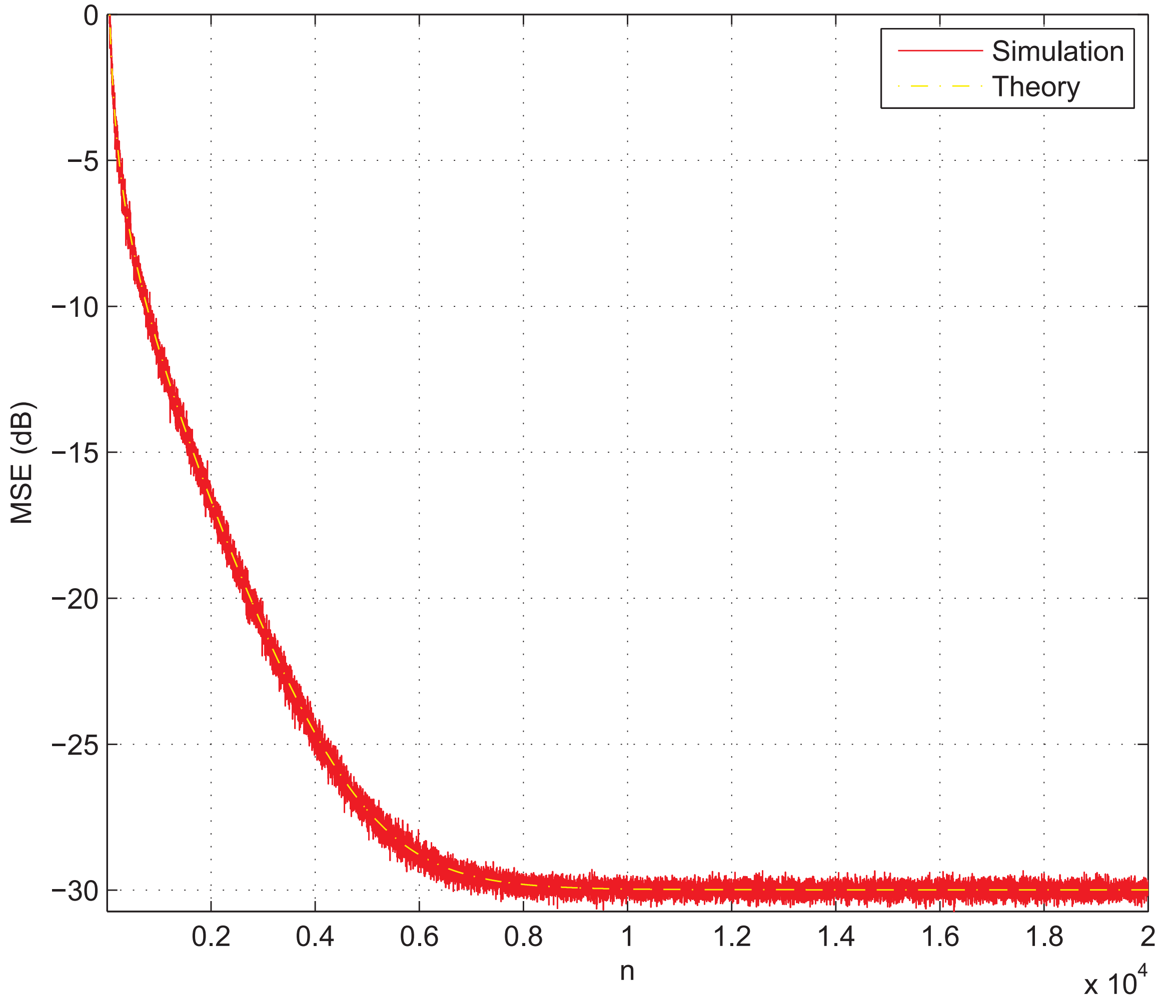

Figure 3.24: Leaky LMMN MSE in Uniform noise with correlated data and noise variance 0.3.

CHAPTER 4

STEADY STATE ANALYSIS OF LEAKY LMMN

In this chapter, the steady state analysis of the proposed Leaky LMMN algorithm is carried out. We will be using the assumptions used in the previous chapter in addition to the following assumption: the regressors $\mathbf{u}_n$ have covariance matrix $\mathbf{R} = \sigma_u^2\mathbf{I}$.

The reason for using this restrictive assumption is to make the analysis more tractable. For the case of correlated regressors, we end up with a single equation with two variables $E[||\mathbf{w}_n||^2]$ and $E[e_a^2(n)]$ which do not have a linear relation between them, thus we end up with an under-determined system. However, for white Gaussian regressors, we have an additional equation that relates $E[||\mathbf{w}_n||^2]$ and $E[e_a^2(n)]$ given by (3.21).

Therefore, we will use (3.21) and (3.26) in our study of the steady-state behavior of the Leaky LMMN. To begin with, we use (3.26) to get

$$\begin{bmatrix} E\left[||\mathbf{w}_{n+1}||^2\right] \\ E\left[\mathbf{w}_{n+1}\right] \end{bmatrix} = \begin{bmatrix} f_1 & f_2 \\ 0 & \mathbf{J} \end{bmatrix} \begin{bmatrix} E\left[||\mathbf{w}_n||^2\right] \\ E\left[\mathbf{w}_n\right] \end{bmatrix} + \mu \begin{bmatrix} \mu\alpha^2||c||^2 + \mu M\sigma_u^2\delta^2\sigma_v^2 + \mu M\sigma_u^2(1-\delta)^2\xi_v^6 + 2\mu M\sigma_u^2\delta(1-\delta)\xi_v^4 \\ \alpha\mathbf{c} \end{bmatrix}$$

$$\begin{aligned} E[\| \tilde{\mathbf{w}_{n+1}} \|^2] &= f_1 E[\| \tilde{\mathbf{w}_n} \|^2] + f_2 E[\tilde{\mathbf{w}}_n] + \mu^2\alpha^2 \| c \|^2 \\ &\quad + \mu^2\sigma_u^2[M\delta^2\sigma_v^2 + M(1-\delta)^2\xi_v^6 + 2M\delta(1-\delta)\xi_v^4] \end{aligned} \tag{4.1}$$

$$E\left[\mathbf{w}_{n+1}\right] = \mathbf{H}E\left[\mathbf{w}_n\right] + \mu\alpha\mathbf{w}_\mathrm{o} \tag{4.2}$$

where H_U, H_G and $\mathbf{J}$ are all defined previously and the terms inside the equations are given by (3.22)-(3.25).

Assuming the step size satisfies the mean and mean square convergence conditions, then at steady state (as $n \to \infty$), we have

$$\lim_{n\to\infty} E\left[||\mathbf{w}_{n+1}||^2\right] = \lim_{n\to\infty} E\left[||\mathbf{w}_n||^2\right] = E\left[||\mathbf{w}_\infty||^2\right] \tag{4.3}$$

$$\lim_{n\to\infty} E\left[\mathbf{w}_{n+1}\right] = \lim_{n\to\infty} E\left[\mathbf{w}_n\right] = E\left[\mathbf{w}_\infty\right] \tag{4.4}$$

Then, taking the limit as $n \to \infty$ on both sides of (4.1)-(4.2), we have

$$\begin{aligned} E\left[||\mathbf{w}_\infty||^2\right] &= f1_\infty E\left[||\mathbf{w}_\infty||^2\right] + f2_\infty E\left[\mathbf{w}_\infty\right] + \mu^2\alpha^2 \parallel \mathbf{c} \parallel^2 \\ &\quad +\mu^2\sigma_u^2[M\delta^2\sigma_v^2 + M(1-\delta)^2\xi_v^6 + 2M\delta(1-\delta)\xi_v^4] \qquad (4.5) \\ E\left[\mathbf{w}_\infty\right] &= \mathbf{J}_\infty E\left[\mathbf{w}_\infty\right] + \mu\alpha\mathbf{c} \qquad (4.6) \end{aligned}$$

where

$$\begin{aligned} f1_\infty &= (1-\mu\alpha)^2 + \mu^2 M\sigma_u^4\delta^2 + 15(1-\delta)^2 M\sigma_u^8 E[\parallel \tilde{\mathbf{w}_\infty} \parallel^2]^2 \\ &\quad +45(1-\delta)^2\mu^2 M\sigma_u^6\sigma_v^2 E[\parallel \tilde{\mathbf{w}_\infty} \parallel^2] + 15(1-\delta)^2\mu^2 M\sigma_u^4\xi_v^4 \\ &\quad +6\delta(1-\delta)\mu^2 M\sigma_u^6 E[\parallel \tilde{\mathbf{w}_\infty} \parallel^2] + 8\delta(1-\delta)\mu^2 M\sigma_u^4\sigma_v^2 \\ &\quad -2\delta\mu(1-\mu\alpha)\sigma_u^2 - 6(1-\delta)\mu(1-\mu\alpha)\sigma_u^2\sigma_v^2 \\ &\quad -6(1-\delta)\mu(1-\mu\alpha)\sigma_u^4 E[\parallel \tilde{\mathbf{w}_\infty} \parallel^2] \qquad (4.7) \end{aligned}$$

$$f2_\infty = 2\mu\alpha\mathbf{J}_\infty\mathbf{c}^T \qquad (4.8)$$

$$\mathbf{J}_\infty = 1 - \mu\{\alpha + [\delta + 3(1-\delta)(\sigma_v^2 + \sigma_u^2 E[\parallel \tilde{\mathbf{w}_\infty} \parallel^2])]\sigma_u^2\} \qquad (4.9)$$

From (4.6) and using (4.9), we get

$$E\left[\mathbf{w}_\infty\right] = \mathbf{J}_\infty E\left[\mathbf{w}_\infty\right] + \mu\alpha\mathbf{c} \qquad (4.10)$$

$$E\left[\mathbf{w}_\infty\right] = [1 - \mu\{\alpha + [\delta + 3(1-\delta)(\sigma_v^2 + \sigma_u^2 E[\parallel \tilde{\mathbf{w}_\infty} \parallel^2])]\sigma_u^2\}]E[\tilde{\mathbf{w}}_\infty] + \mu\alpha\mathbf{c} \qquad (4.11)$$

$$E[\tilde{\mathbf{w}}_\infty] = \frac{\alpha \mathbf{c}}{\alpha + [\delta + 3(1-\delta)(\sigma_v^2 + \sigma_u^2 E[\| \tilde{\mathbf{w}_\infty} \|^2])]\sigma_u^2} \tag{4.12}$$

Let

$$\mathbf{C} = \alpha + [\delta + 3(1-\delta)(\sigma_v^2 + \sigma_u^2 E[\| \tilde{\mathbf{w}_\infty} \|^2])]\sigma_u^2 \tag{4.13}$$

Then using (4.12) in (4.5), we get

$$\begin{aligned} E[\| \tilde{\mathbf{w}_\infty} \|^2] &= f1_\infty E[\| \tilde{\mathbf{w}_\infty} \|^2] + \frac{2\mu\alpha^2(1-\mathbf{C}) \| \mathbf{c} \|^2}{\mathbf{C}} + \mu^2\alpha^2 \| \mathbf{c} \|^2 \\ &\quad + \mu^2 M \sigma_u^2[\delta^2\sigma_v^2 + (1-\delta)^2\xi_v^6 + 2\delta(1-\delta)\xi_v^4] \end{aligned} \tag{4.14}$$

Multiplying both sides of (4.14) by C, we get

$$\begin{aligned} \mathbf{C}E[\| \tilde{\mathbf{w}_\infty} \|^2] &= \mathbf{C}f1_\infty E[\| \tilde{\mathbf{w}_\infty} \|^2] + 2\mu\alpha^2(1-\mathbf{C}) \| \mathbf{c} \|^2 + \mathbf{C}\mu^2\alpha^2 \| \mathbf{c} \|^2 \\ &\quad + \mu^2 M \mathbf{C}\sigma_u^2[\delta^2\sigma_v^2 + (1-\delta)^2\xi_v^6 + 2\delta(1-\delta)\xi_v^4] \end{aligned} \tag{4.15}$$

Opening this expression and grouping together coefficients of different powers of $E\left[||\mathbf{w}_\infty||^2\right]$ together, then after some algebra, we get the following quartic polynomial in $E\left[||\mathbf{w}_\infty||^2\right]$:

$$\sum_{j=0}^{4} \beta_j \left(E\left[||\mathbf{w}_\infty||^2\right]\right)^j = 0, \tag{4.16}$$

where

$$
\begin{aligned}
\beta_0 &= 2\mu\alpha^2 \parallel \mathbf{c} \parallel^2 - 2\mu\alpha^3 \parallel \mathbf{c} \parallel^2 - 2\mu\alpha^2 \parallel \mathbf{c} \parallel^2 \delta\sigma_u^2 \\
&\quad - 6(1-\delta)\mu\alpha^2 \parallel \mathbf{c} \parallel^2 \sigma_u^2\sigma_v^2 + \mu^2\alpha^3 \parallel \mathbf{c} \parallel^2 + \mu^2\alpha^2 \parallel \mathbf{c} \parallel^2 \delta\sigma_u^2 \\
&\quad + 3(1-\delta)\mu^2\alpha^2 \parallel \mathbf{c} \parallel^2 \sigma_u^2\sigma_v^2 + \delta^2\mu^2 M\sigma_u^2\sigma_v^2\alpha \\
&\quad + \delta^3\mu^2 M\sigma_u^4\sigma_v^2 + 3(1-\delta)\delta^2\mu^2 M\sigma_u^4\xi_v^4 \\
&\quad + (1-\delta)^2\mu^2 M\alpha\sigma_u^2\xi_v^6 + \delta(1-\delta)^2\mu^2 M\sigma_u^4\xi_v^6 \\
&\quad + 3(1-\delta)^3\mu^2 M\sigma_u^4\xi_v^8 + 2\delta(1-\delta)\mu^2 M\alpha\sigma_u^2\xi_v^4 \\
&\quad + 2\delta^2(1-\delta)\mu^2 M\sigma_u^4\xi_v^4 + 6\delta(1-\delta)^2\mu^2 M\sigma_u^4\xi_v^6 \qquad (4.17)
\end{aligned}
$$

$$
\begin{aligned}
\beta_1 &= \alpha(1-\mu\alpha)^2 + \delta(1-\mu\alpha)^2\sigma_u^2 + 3(1-\delta)(1-\mu\alpha)^2\sigma_u^2\sigma_v^2 \\
&\quad + \alpha\delta^2\mu^2 M\sigma_u^4 + \delta^3\mu^2 M\sigma_u^6 + 3(1-\delta)\delta^2\mu^2 M\sigma_u^6\sigma_v^2 \\
&\quad + 15\alpha(1-\delta)^2\mu^2 M\sigma_u^4\xi_v^4 + 15\delta(1-\delta)^2\mu^2 M\sigma_u^6\xi_v^4 \\
&\quad + 45(1-\delta)^3\mu^2 M\sigma_u^6\xi_v^6 + 8\alpha\delta(1-\delta)\mu^2 M\sigma_u^4\sigma_v^2 \\
&\quad + 8\delta^2(1-\delta)\mu^2 M\sigma_u^6\sigma_v^2 + 24\delta(1-\delta)^2\mu^2 M\sigma_u^6\sigma_v^2 \\
&\quad - 2\alpha\delta\mu(1-\mu\alpha)M\sigma_u^2 - 2\delta^2\mu(1-\mu\alpha)M\sigma_u^4 - 6\delta(1-\delta)\mu(1-\mu\alpha)M\sigma_u^4\sigma_v^2 \\
&\quad - 6\alpha(1-\delta)\mu(1-\mu\alpha)\sigma_u^2\sigma_v^2 - 6\delta(1-\delta)\mu(1-\mu\alpha)\sigma_u^4\sigma_v^2 \\
&\quad - 18(1-\delta)^2\mu(1-\mu\alpha)\sigma_u^4\xi_v^4 - 6\alpha^2(1-\delta)\mu \parallel \mathbf{c} \parallel^2 \sigma_u^4 \\
&\quad + 3\alpha^2(1-\delta)\mu^2 \parallel \mathbf{c} \parallel^2 \sigma_u^4 + 3\delta^2(1-\delta)\mu^2 M\sigma_u^6\sigma_v^2 \\
&\quad + 3(1-\delta)^3\mu^2 M\sigma_u^6 + 6\delta(1-\delta)^2\mu^2 M\sigma_u^6\xi_v^4 \qquad (4.18)
\end{aligned}
$$

$$
\begin{aligned}
\beta_2 &= 3(1-\delta)(1-\mu\alpha)^2\sigma_u^4 + 3\delta^2(1-\delta)\mu^2 M\sigma_u^8 \\
&+45\alpha(1-\delta)^2\mu^2 M\sigma_u^6\sigma_v^2 + 45\delta(1-\delta)^2\mu^2 M\sigma_u^8\sigma_v^2 \\
&+135(1-\delta)^3\mu^2 M\sigma_u^8\xi_v^4 + 45(1-\delta)^3\mu^2 M\sigma_u^8\xi_v^4 \\
&+6\alpha\delta(1-\delta)\mu^2 M\sigma_u^6 + 6\delta^2(1-\delta)\mu^2 M\sigma_u^8 \\
&+18\delta(1-\delta)^2\mu^2 M\sigma_u^8\sigma_v^2 + 24\delta(1-\delta)^2\mu^2 M\sigma_u^8\sigma_v^2 \\
&-6\delta(1-\delta)\mu(1-\mu\alpha)\sigma_u^6 - 18(1-\delta)^2\mu(1-\mu\alpha)\sigma_u^6\sigma_v^2 \\
&-6\alpha(1-\delta)\mu(1-\mu\alpha)\sigma_u^4 - 6\delta(1-\delta)\mu(1-\mu\alpha)\sigma_u^6 \\
&-18(1-\delta)^2\mu(1-\mu\alpha)\sigma_u^6\sigma_v^2
\end{aligned}
\tag{4.19}
$$

$$
\begin{aligned}
\beta_3 &= 15\alpha(1-\delta)^2\mu^2 M\sigma_u^8 + 15\delta(1-\delta)^2\mu^2 M\sigma_u^{10} \\
&+45(1-\delta)^3\mu^2 M\sigma_u^{10}\sigma_v^2 + 135(1-\delta)^3\mu^2 M\sigma_u^{10}\sigma_v^2 \\
&+18\delta(1-\delta)^2\mu^2 M\sigma_u^{10} - 18(1-\delta)^2\mu(1-\mu\alpha)\sigma_u^8
\end{aligned}
\tag{4.20}
$$

$$
\beta_4 = 45(1-\delta)^3\mu^2 M\sigma_u^{12} \tag{4.21}
$$

Since $E\left[||\mathbf{w}_\infty||^2\right]$ will be very small, we can assume $\left(E\left[||\mathbf{w}_\infty||^2\right]\right)^4$ to be negligible and the problem of finding $E\left[||\mathbf{w}_\infty||^2\right]$ is now solved by finding the roots of the following polynomial equation:

$$
\sum_{j=0}^{3}\chi_j(E\left[||\mathbf{w}_\infty||^2\right])^j = 0, \tag{4.22}
$$

where

$$\chi_j = \frac{\beta_j}{\beta_4}. \tag{4.23}$$

Equation (4.22) has three roots [28]. From simulations, we found that the smallest positive square root of the polynomial gives $E\left[||\mathbf{w}_\infty||^2\right]$.

4.1 Comparison of the convergence speed of the LMMN algorithm and the proposed algorithm in achieving the same steady-state error with white input sequence

In this section the LMMN algorithm and the proposed time-varying Leaky LMMN algorithm are compared in terms of the convergence time when the input is white. In figure 4.1, 4.3, 4.5 and 4.7 it is clear that the proposed algorithm has achieved the same steady state error in lesser number of iterations as compared to the LMMN algorithm. Now considering the Figure 4.1 it is shown that in 20 dB SNR and uniform environment the proposed algorithm achieved the same steady-state error in 4000 iterations earlier than the LMMN algorithm, while considering the case of 30 dB SNR in Uniform environment the proposed algorithm converged almost 7000 iterations earlier than the LMMN algorithm as shown in Figure 4.3. Thus it can inferred from the above discussion that in uniform environment with 30 dB SNR the proposed algorithm performs better than in the case of 20 dB SNR.

In Figure 4.5 and Figure 4.7 the proposed algorithm is compared to the LMMN algorithm in the gaussian enviromment. In Figure 4.5 it is shown that both the algorithms have almost the same convergence time in the case of 20 dB SNR, while from Figure 4.7 it can be inferred that there is a difference of 1000 iterations between the proposed and the LMMN algorithm at 30 dB SNR. Thus it can be concluded that the proposed algorithm performs better at 30 dB SNR in gaussian environment.

The behavior of time varying mixing parameter is also plotted for the respective cases with the corresponding SNR. The curves for the mixing parameter in the case of gaussian environment have slower convergence speed as compared to that of uniform environment as shown in Figure 4.6 and Figure 4.8. Thus the mixing parameter has the better convergence ability in the case of uniform environment.

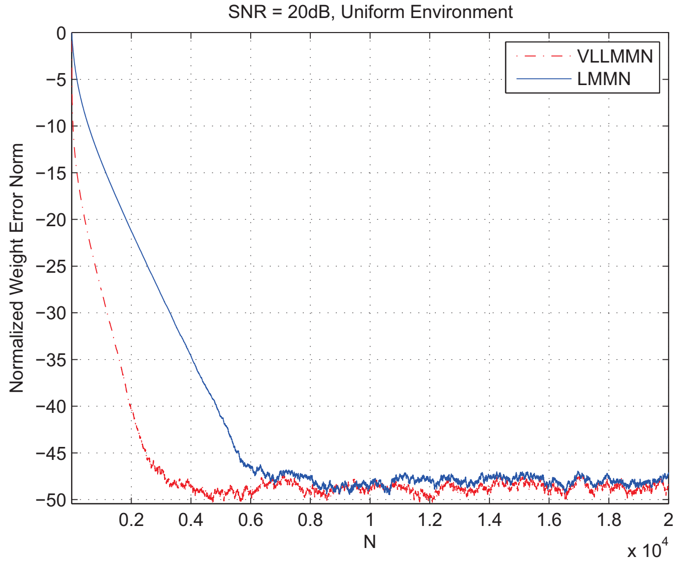

Figure 4.1: Comparison of the convergence speed of the LMMN and proposed algorithm with SNR = 20 dB in uniform environment.

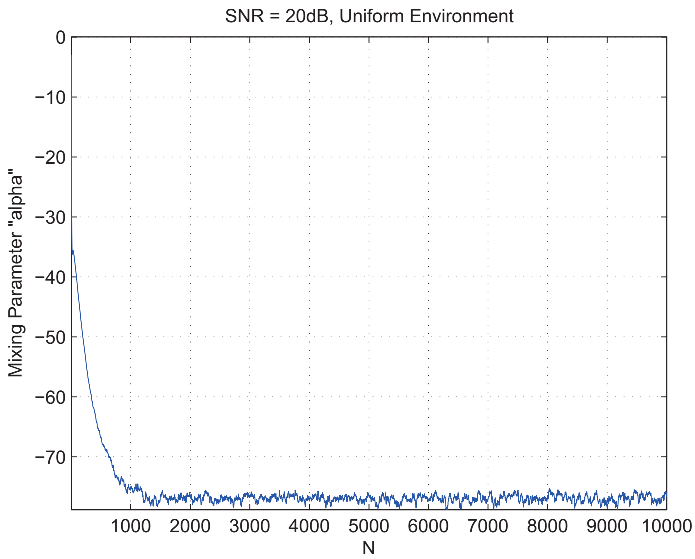

Figure 4.2: Behavior of time varying mixing parameter for the respective case.

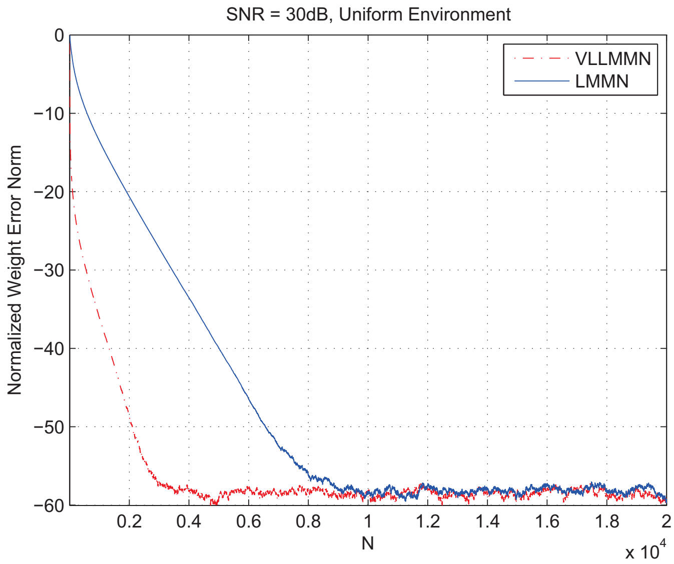

Figure 4.3: Comparison of the convergence speed of the LMMN and proposed algorithm with SNR = 30 dB in uniform environment.

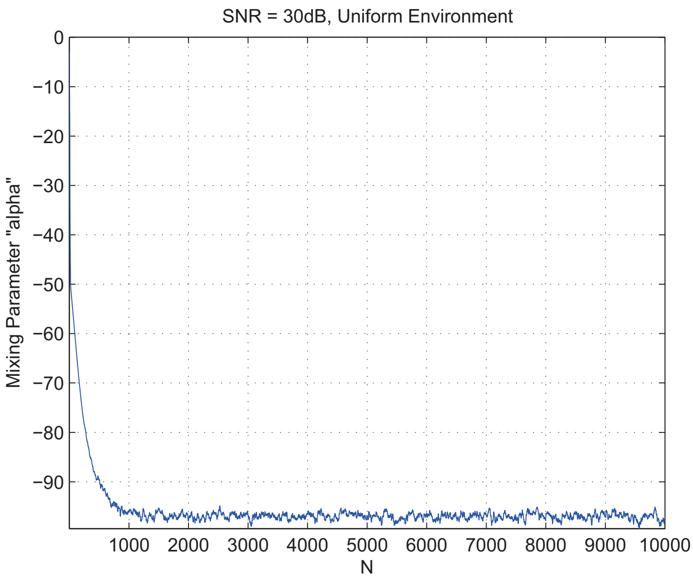

Figure 4.4: Behavior of time varying mixing parameter for the respective case.

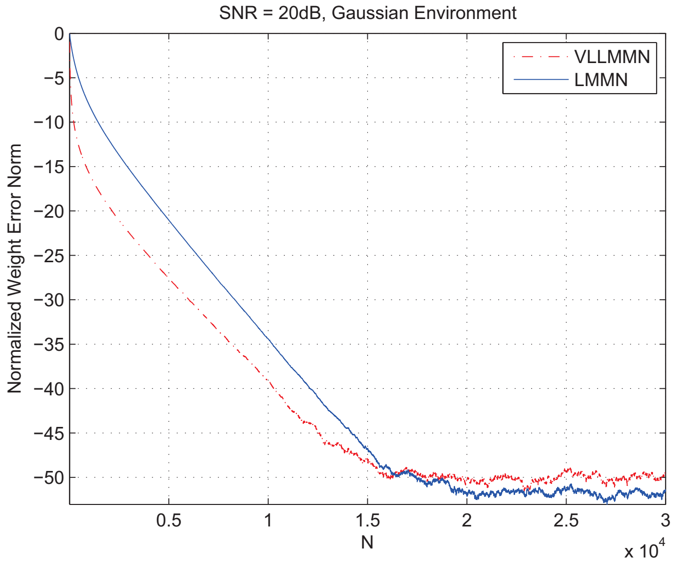

Figure 4.5: Comparison of the convergence speed of the LMMN and proposed algorithm with SNR = 20 dB in gaussian environment.

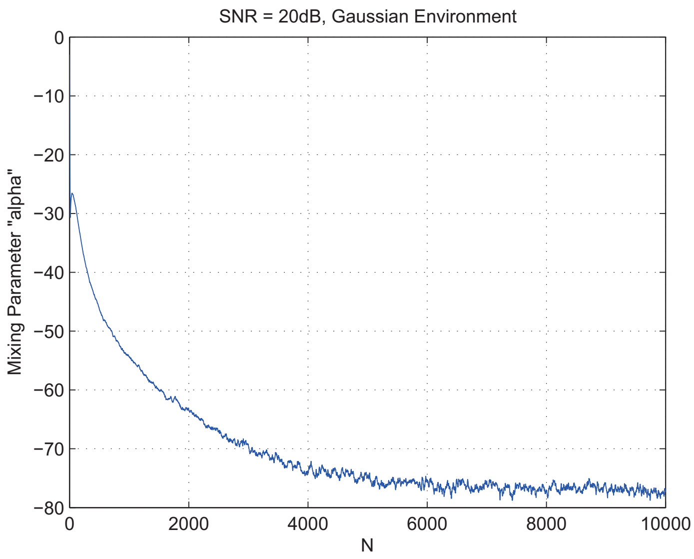

Figure 4.6: Behavior of time varying mixing parameter for the respective case.

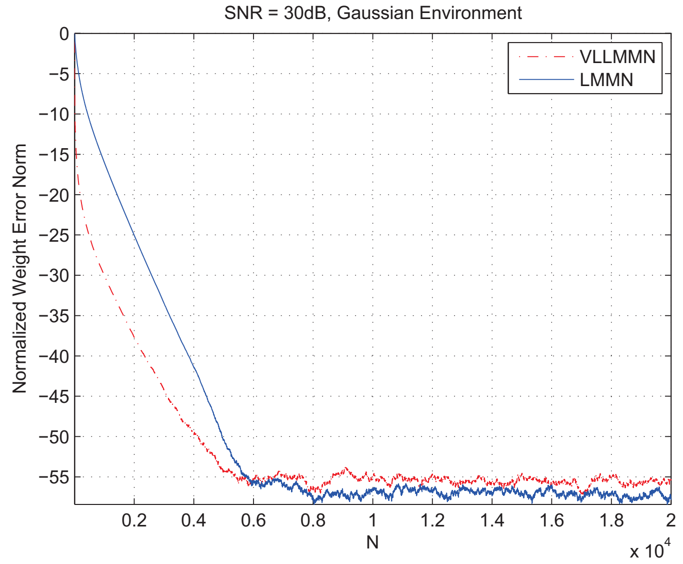

Figure 4.7: Comparison of the convergence speed of the LMMN and proposed algorithm with SNR = 30 dB in gaussian environment.

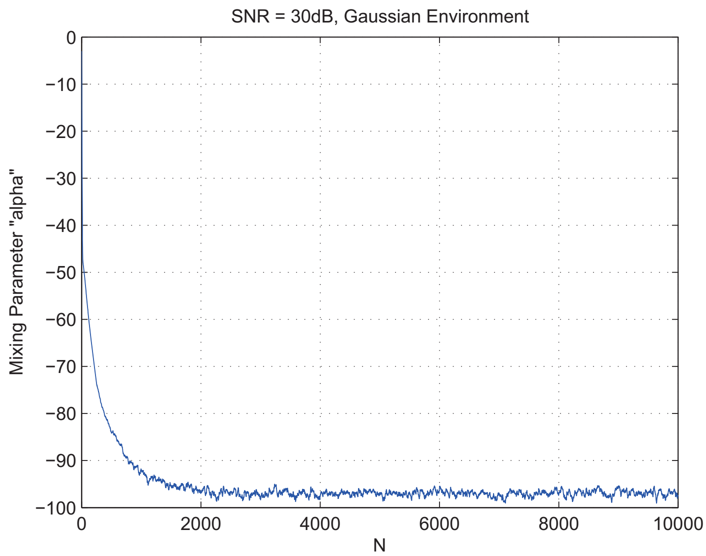

Figure 4.8: Behavior of time varying mixing parameter for the respective case.

CHAPTER 5

TRACKING ANALYSIS OF LEAKY LMMN

The aim of tracking analysis of an adaptive filter is to provide a quantitative measure of how well the adaptive algorithm is able to track variations in the signal statistics. In this chapter, the tracking analysis of the proposed algorithm is carried out. Both the random walk model and the Rayleigh fading model (single path and multipath) to model the time varying channels and the analysis is carried out in the same way as was done for the steady state analysis.

5.1 Random Walk Model

The first order random-walk model for a channel is given as

$$\mathbf{c}_{n+1} = \mathbf{c}_n + q_n \tag{5.1}$$

where $\mathbf{c}_n$ is the time-varying wide-sense stationary unknown system that is to be tracked and $\mathbf{q}_n$ is assumed to be a zero-mean stationary random vector process with a positive-definite covariance matrix $\mathbf{Q}$. It is also statistically independent of all other parameters of the adaptive filter. The noisy measurement that arises from the random walk model is given by

$$d_n = \mathbf{u}_n \mathbf{c}_n + v_n \tag{5.2}$$

It can be seen from the assumptions used for $\mathbf{q}_n$ and (5.1) that

$$\begin{aligned} E[\mathbf{c}_{n+1}] &= E[\mathbf{c}_n] \\ &= \mathbf{c} \end{aligned} \tag{5.3}$$

Now it was observed in [4] that the covariance matrix of $\mathbf{c}_{n+1}$ i.e $\mathbf{C}_{n+1}$ is given by

$$\begin{aligned}
\mathbf{C}_{n+1} &= E[(\mathbf{c}_{n+1} - \mathbf{c})(\mathbf{c}_{n+1} - \mathbf{c})^T] \\
&= E[(\mathbf{c}_n + q_n - \mathbf{c})(\mathbf{c}_n + q_n - \mathbf{c})^T] \\
&= E[(\mathbf{c}_n - \mathbf{c})(\mathbf{c}_n - \mathbf{c})^T] + E[q_n q_n^T] \\
&= \mathbf{C}_{n+1} + \mathbf{Q} \qquad (5.4)
\end{aligned}$$

We see that a positive-definite matrix is added to the covariance matrix of the the unknown system vector at each iteration and thus grows unbounded. A more practical model that can be used is by replacing (5.1) by

$$\mathbf{c}_{n+1} - \mathbf{c} = \varrho(\mathbf{c}_n - \mathbf{c}) + q_n \qquad (5.5)$$

for some scalar $|\varrho| < 1$. In this case, the covariance matrix of $\mathbf{c}_{n+1}$ would tend to a finite steady-state value given by

$$\lim_{n\to\infty} \mathbf{C}_{n+1} = \frac{\mathbf{Q}}{1 - |\varrho|^2} \qquad (5.6)$$

However, the tracking analysis of this model is more demanding. As mentioned in [4], it was found that in the literature it is a convention to assume the value of ϱ to be sufficiently close to 1 to warrant the use of model (5.1) which simplifies our analysis greatly.For this reason , we have used the model (5.1) for tracking analysis of the Leaky LMMN.

5.2 Tracking Analysis of Leaky LMMN for Random Walk Model

To begin with, we shall rewrite the Leaky LMMN update equation, taking the non-stationarity of the channel into account, we get the following recursion:

$$\mathbf{w}_{n+1} = (1-\mu\alpha)\mathbf{w}_n + \mu\mathbf{u}_n^T e_n\{\delta + (1-\delta)||e_n||^2\}. \tag{5.7}$$

Let $\tilde{\mathbf{w}}_j = \mathbf{c}_j - \mathbf{w}_j$, then

$$\begin{aligned}
\mathbf{c}_{n+1} - \mathbf{w}_{n+1} &= \mathbf{c}_{n+1} - [(1-\mu\alpha)\mathbf{w}_n + \mu\mathbf{u}_n^T e_n\{\delta + (1-\delta)||e_n||^2\}] \\
&= \mathbf{c}_{n+1} - (1-\mu\alpha)\mathbf{w}_n - \mu\mathbf{u}_n^T e_n\{\delta + (1-\delta)||e_n||^2\} \\
&= \mathbf{c}_{n+1} - \mathbf{w}_n + \mu\alpha\mathbf{w}_n - \mu\mathbf{u}_n^T e_n\{\delta + (1-\delta)||e_n||^2\} \\
&= \mathbf{c}_{n+1} - \mathbf{w}_n + \mu\alpha(\mathbf{c}_n - \tilde{\mathbf{w}}_n) - \mu\mathbf{u}_n^T e_n\{\delta + (1-\delta)||e_n||^2\} \\
&= \mathbf{c}_n + q_n - \mathbf{w}_n + \mu\alpha(\mathbf{c}_n - \tilde{\mathbf{w}}_n) - \mu\mathbf{u}_n^T e_n\{\delta + (1-\delta)||e_n||^2\} \\
&= (\mathbf{c}_n - \mathbf{w}_n) + q_n + \mu\alpha\mathbf{c}_n - \mu\alpha\tilde{\mathbf{w}}_n - \mu\mathbf{u}_n^T e_n\{\delta + (1-\delta)||e_n||^2\} \\
&= \tilde{\mathbf{w}}_{\mathbf{n}} + q_n + \mu\alpha\mathbf{c}_n - \mu\alpha\tilde{\mathbf{w}}_n - \mu\mathbf{u}_n^T e_n\{\delta + (1-\delta)||e_n||^2\} \qquad (5.8)
\end{aligned}$$

$$\tilde{\mathbf{w}}_{\mathbf{n+1}} = (1-\mu\alpha)\tilde{\mathbf{w}}_{\mathbf{n}} + \mu\alpha\mathbf{c}_n - \mu\mathbf{u}_n^T e_n\{\delta + (1-\delta)||e_n||^2\} + q_n. \tag{5.9}$$

Taking the weighted norms of both sides of (5.8), with $\mathbf{A}$ being the symmetric weighting matrix, and using **A1-A6** along with the assumptions on the statistics of $\mathbf{q}_n$, we get

$$\begin{aligned} E[||\tilde{\mathbf{w}}_{\mathbf{n+1}}||^2_{\sum}] &= (1-\mu\alpha)^2 E[||\tilde{\mathbf{w}}_{\mathbf{n}}||^2_{\sum}] + ||\mu\alpha\mathbf{c}||^2_{\sum} + \mu^2 tr(\mathbf{R}\sum)H_U \\ &\quad -2\mu(1-\mu\alpha)H_G E[||\tilde{\mathbf{w}}_{\mathbf{n}}||^2_{\mathbf{R}\sum}] + 2\mu\alpha\mathbf{c}^T\sum \mathbf{J}E[\tilde{\mathbf{w}}_{\mathbf{n}}] \\ &\quad + tr(\mathbf{Q}\sum) \end{aligned} \tag{5.10}$$

We see that the only difference between (3.20) and (5.10) is the addition term $tr\,(\mathbf{QA})$. Using this fact, we can approach the problem of tracking analysis of the Leaky LMMN in the same way as was done for the steady state analysis for white gaussian data.

Furthermore, after applying the same steps and assumptions done for transient analysis of stationary environment to non-stationary environment expressed by the random walk model, we get the following state space equation representing the evolution of $E[||\mathbf{w}_n||^2]$ and $E[e_a^2(n)]$ in a random walk model:

$$\underbrace{\begin{bmatrix} \mathbf{A}_{k+1} \\ E\left[\mathbf{v}_{k+1}\right] \end{bmatrix}}_{\mathbf{W}_{k+1}} = \underbrace{\begin{bmatrix} \mathbf{F}_1 & \mathbf{F}_2 \\ \mathbf{0} & \mathbf{H} \end{bmatrix}}_{\mathbf{F_k}} \underbrace{\begin{bmatrix} \mathbf{A}_{k} \\ E\left[\mathbf{v}_{k}\right] \end{bmatrix}}_{\mathbf{W}_{k}} + \underbrace{\begin{bmatrix} \mathbf{M}_{k} \\ \mu\alpha\mathbf{c} \end{bmatrix}}_{\mathbf{Y}_{k}} \tag{5.11}$$

where the only difference between (3.36) and (5.11) is the term $\mathbf{M}_k$ given by

$$\mathbf{M}_k = \mu^2 \mathcal{Z}_k \begin{bmatrix} tr(\mathbf{R}) \\ tr(\mathbf{R}^2) \\ tr(\mathbf{R}^3) \\ \vdots \\ tr(\mathbf{R}^M) \end{bmatrix} + \mu^2 \alpha^2 \begin{bmatrix} ||\mathbf{c}||^2 \\ ||\mathbf{c}||^2_{\mathbf{R}} \\ ||\mathbf{c}||^2_{\mathbf{R}^2} \\ \vdots \\ ||\mathbf{c}||^2_{\mathbf{R}^{M-1}} \end{bmatrix} + \begin{bmatrix} tr(\mathbf{Q}) \\ tr(\mathbf{QR}) \\ tr(\mathbf{QR}^2) \\ \vdots \\ tr(\mathbf{QR}^{M-1}) \end{bmatrix} \tag{5.12}$$

5.3 Comparison of the Theoretical and Simulation Results For Tracking Analysis

In this part of the simulations we will consider a non-stationary environment and observe the behavior of the proposed Leaky LMMN algorithm. For this purpose we will make use of random walk model with the step size, leakage factor and the noise variance set at 0.01, 0.001 and 0.001 respectively. As was done in the transient analysis of the proposed Leaky LMMN algorithm, even here we make use of a randomly generated weight vector with the number of taps set at 5. The number of samples were set at 10^4 and the number of trials at 800. During the simulations of the tracking analysis the noise variances of the weight vector elements were set at 10^{-5},10^{-6} and 10^{-7} and carried out for both the cases of Uniform and Gaussian. Theoretical results were generated using (5.11).

We see from the Fig. 5.1-5.12 that the theoretical and the simulation results match.

Moreover, as expected, it is observed that as the variance of the true weight vector decreases from 10^{-5} to 10^{-7}, the MSE performance of the Leaky LMMN improves.

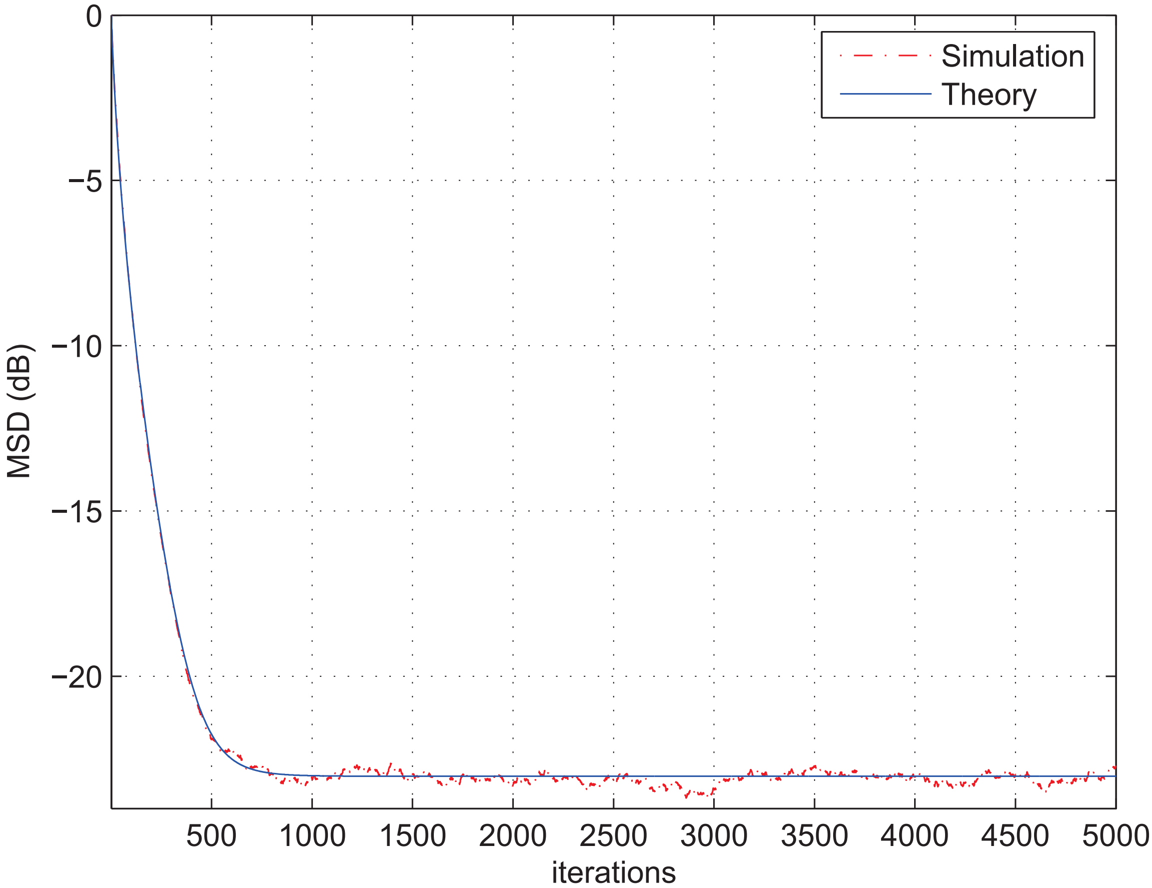

Figure 5.1: Tracking MSD of leaky LMMN in Gaussian noise with weight variance 10^{-5}.

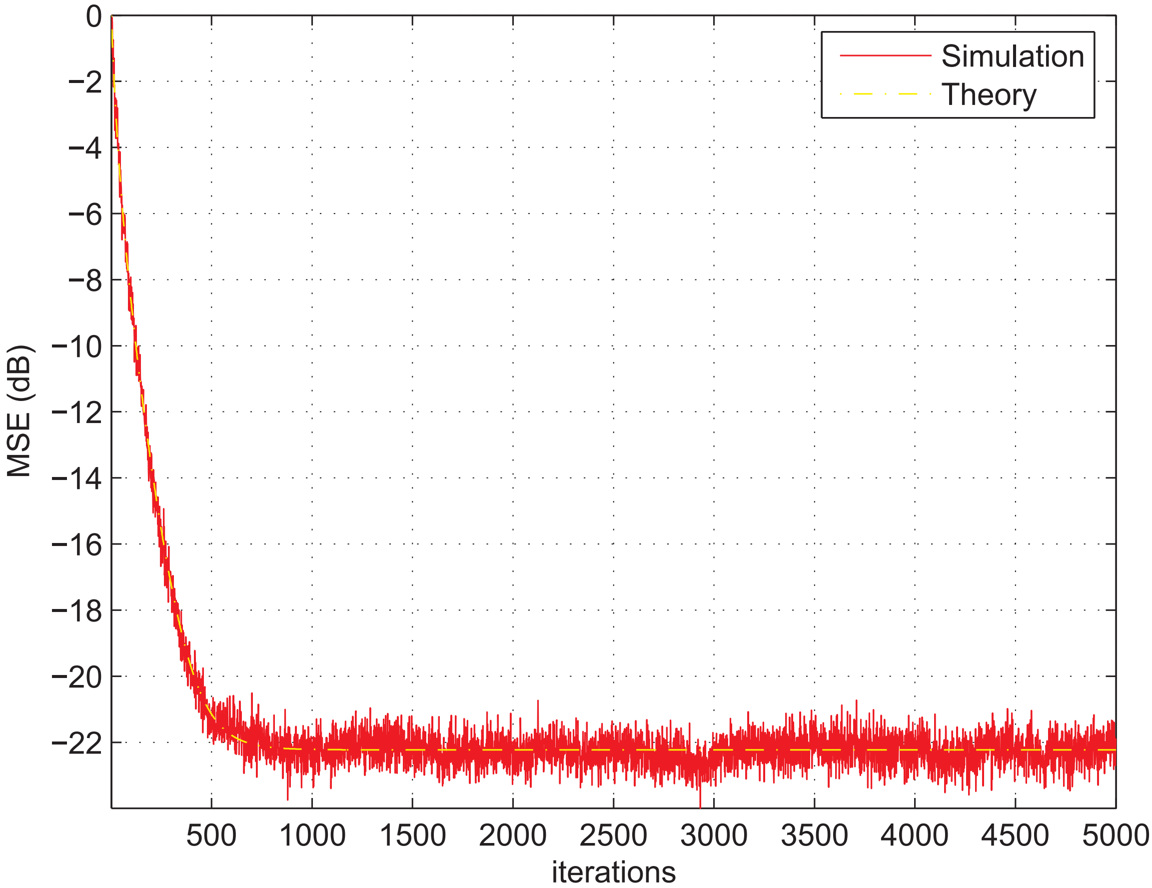

Figure 5.2: Tracking MSE of leaky LMMN in Gaussian noise with weight variance 10^{-5}.

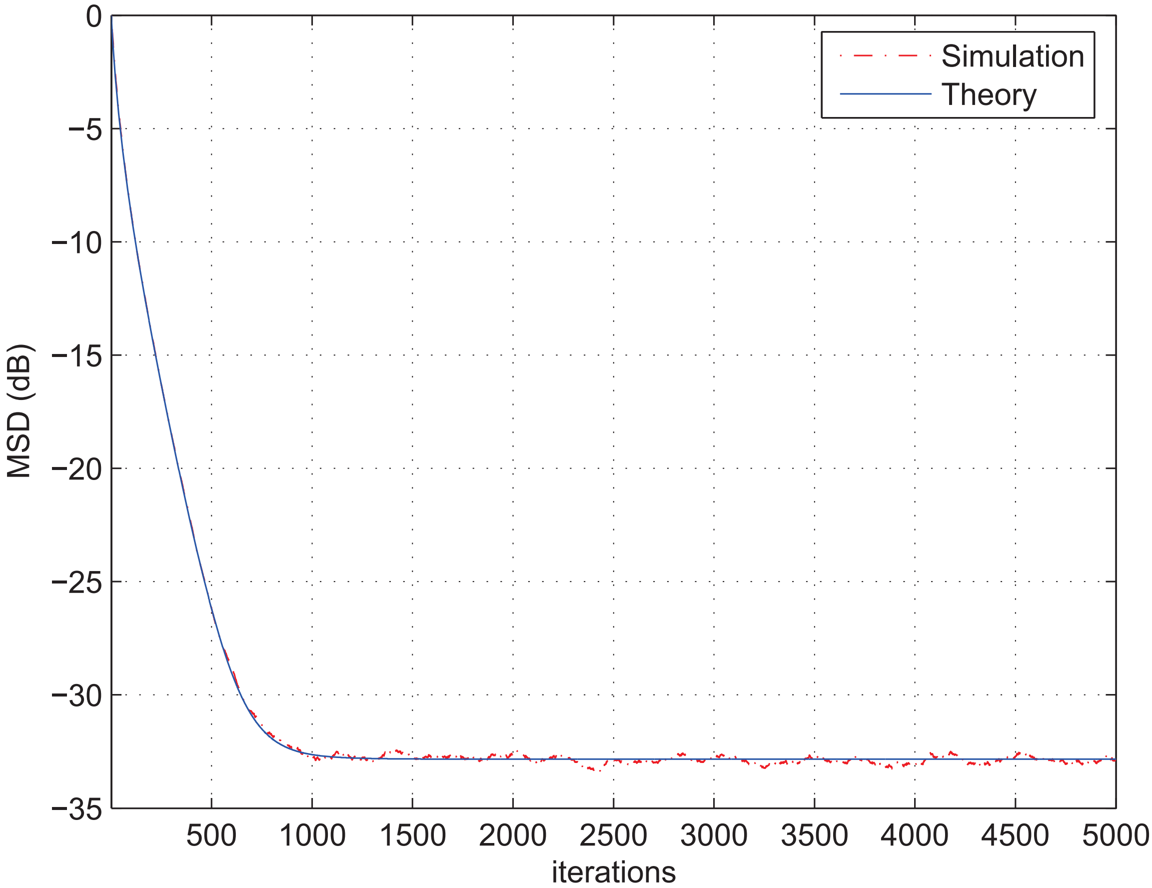

Figure 5.3: Tracking MSD of leaky LMMN in Gaussian noise with weight variance 10^{-6}.

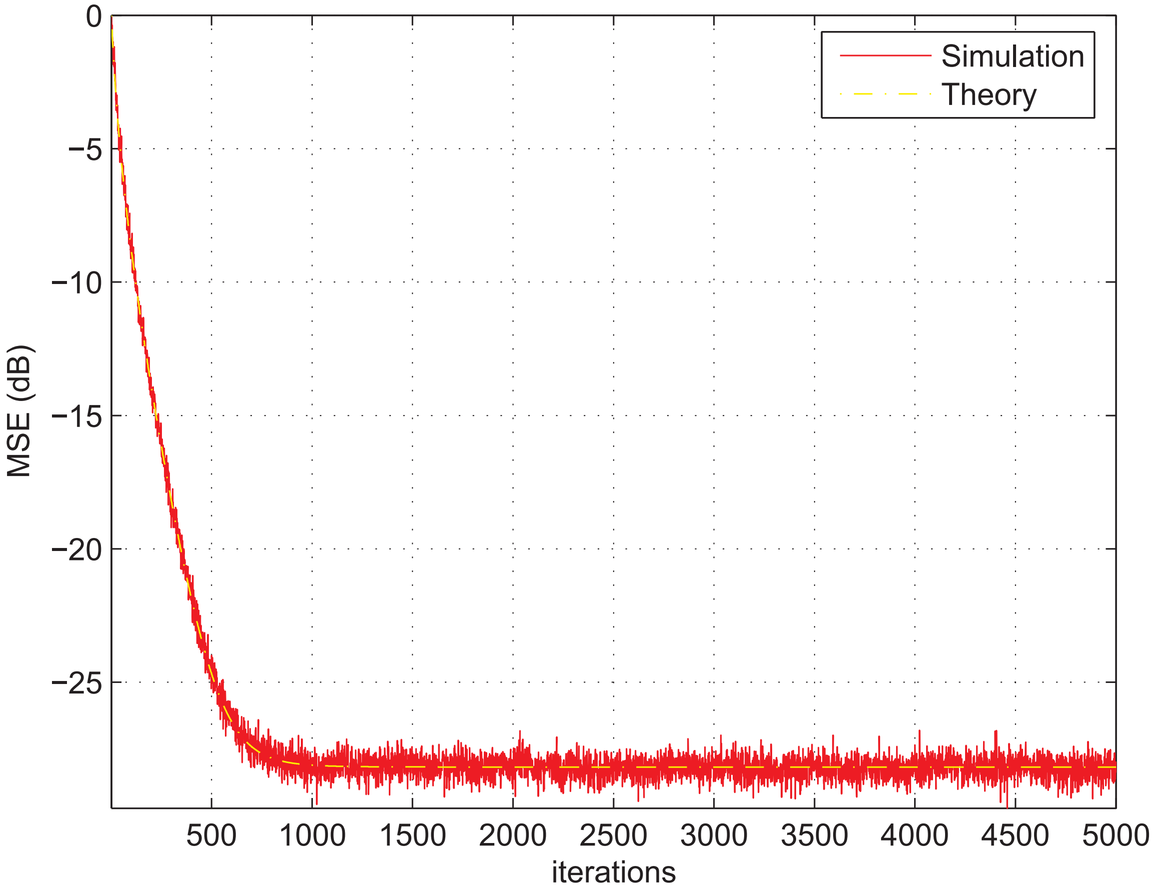

Figure 5.4: Tracking MSE of leaky LMMN in Gaussian noise with weight variance 10^{-6}.

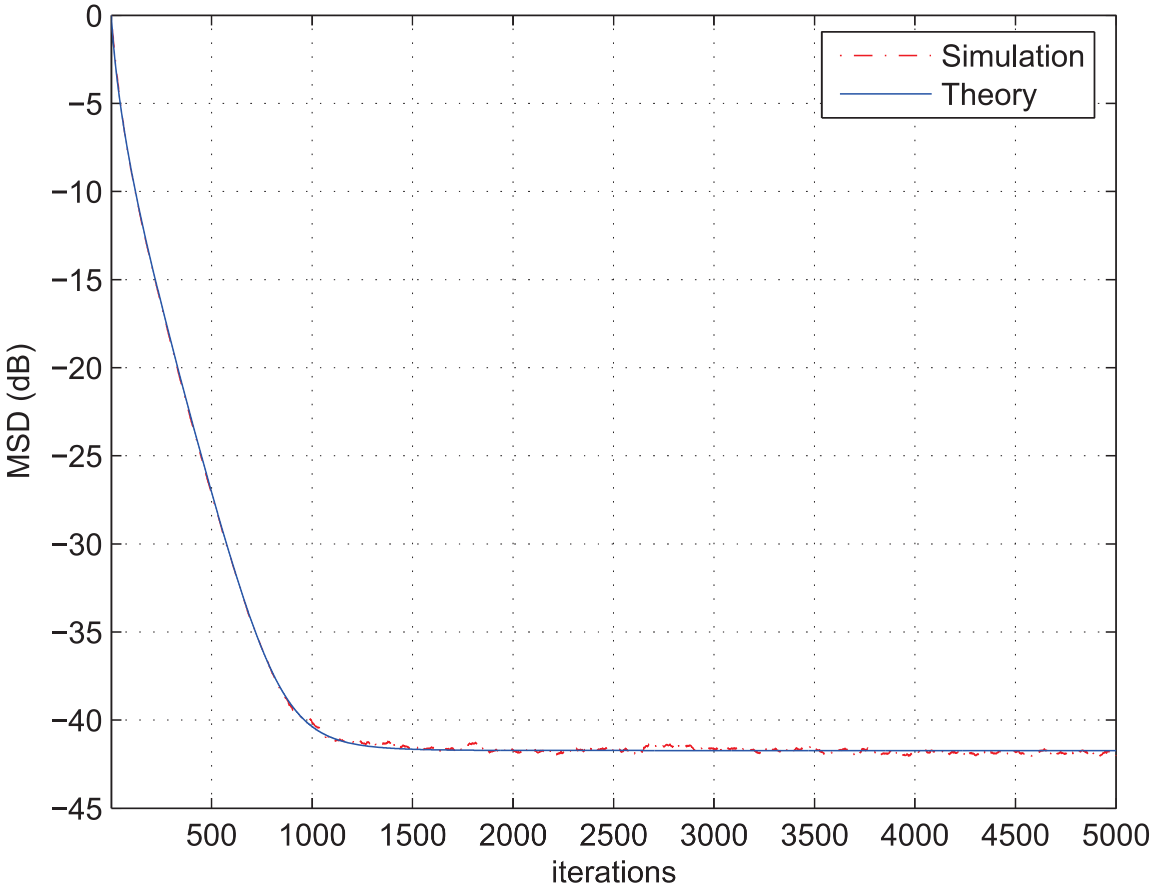

Figure 5.5: Tracking MSD of leaky LMMN in Gaussian noise with weight variance 10^{-7}.

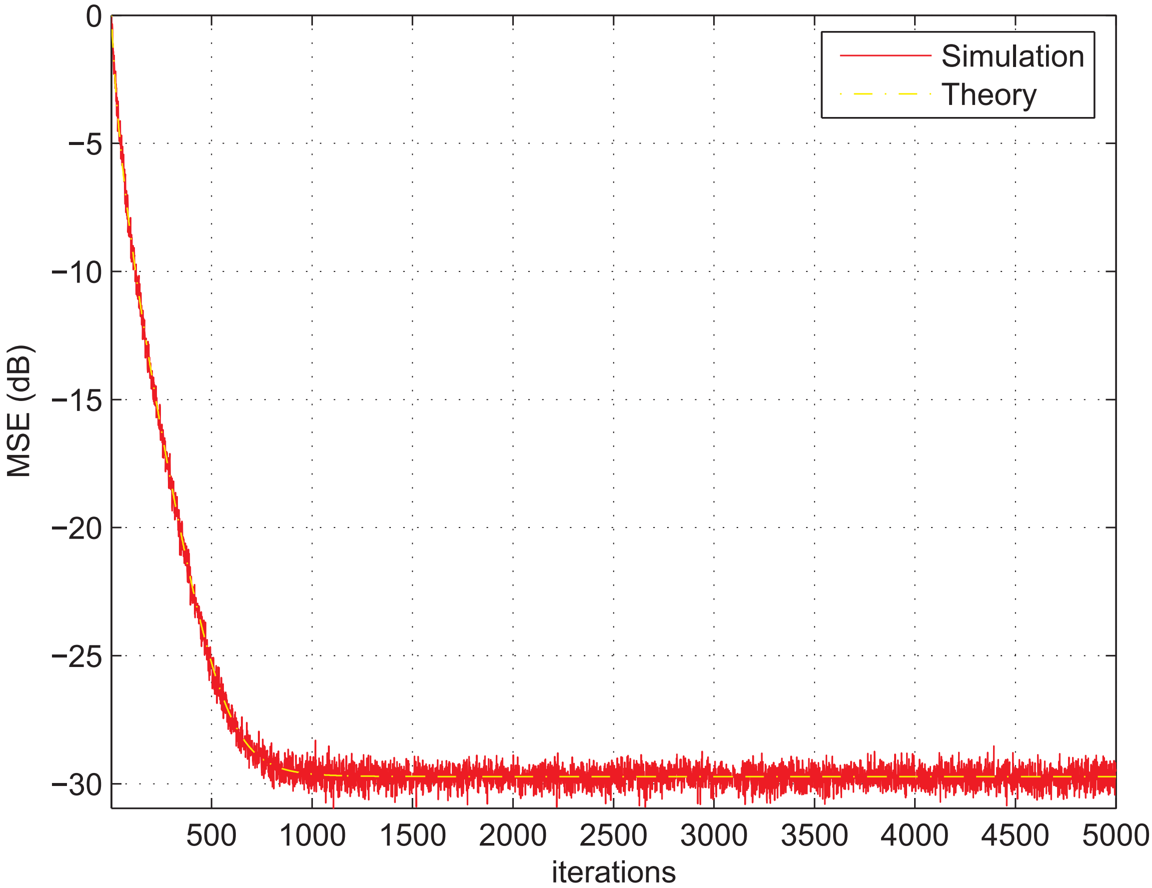

Figure 5.6: Tracking MSE of leaky LMMN in Gaussian noise with weight variance 10^{-7}.

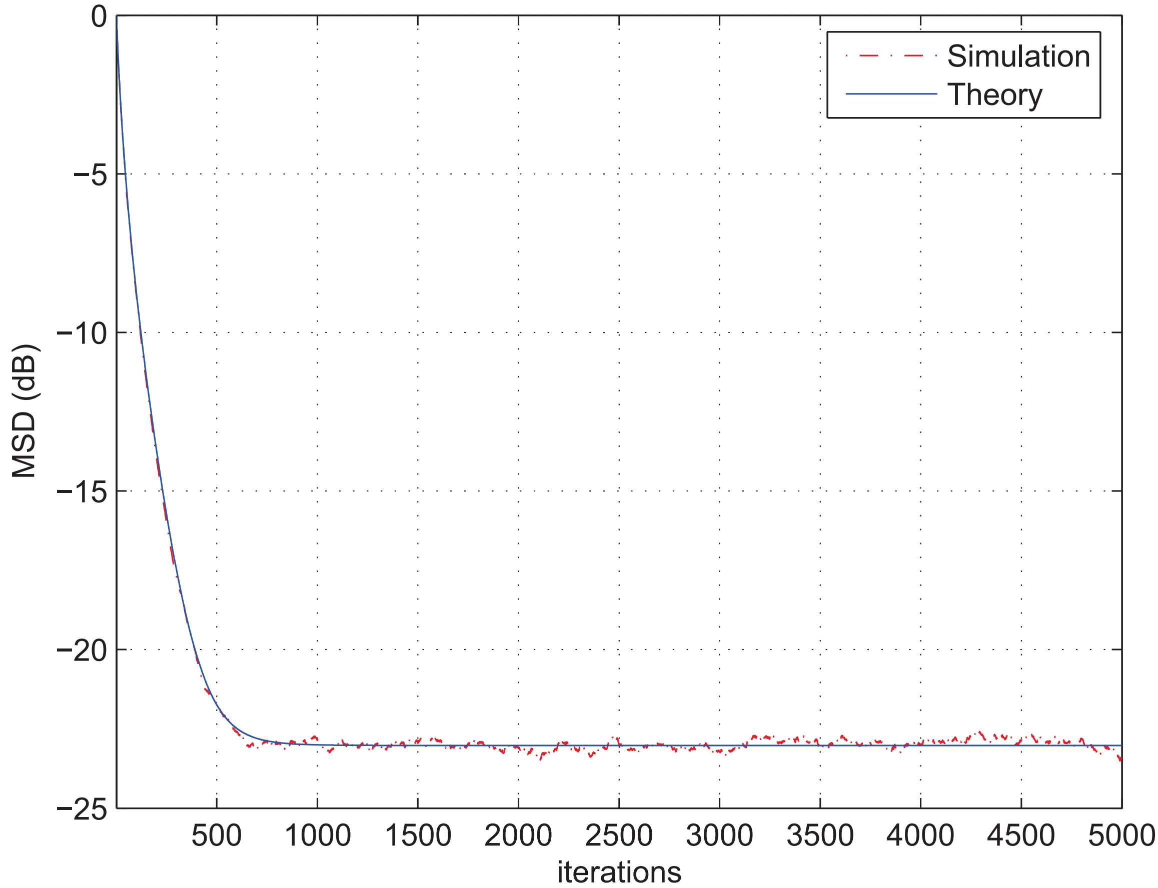

Figure 5.7: Tracking MSD of leaky LMMN in Uniform noise with weight variance 10^{-5}.

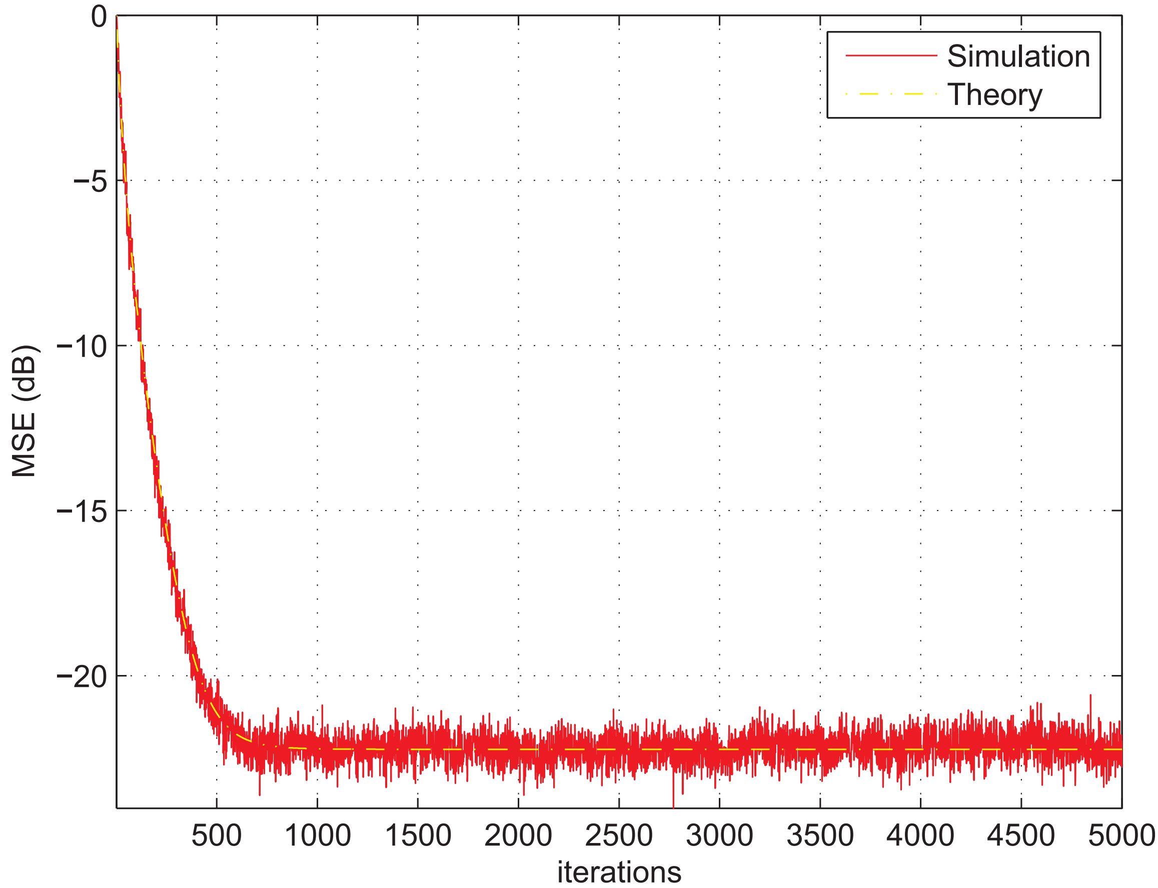

Figure 5.8: Tracking MSE of leaky LMMN in Uniform noise with weight variance 10^{-5}.

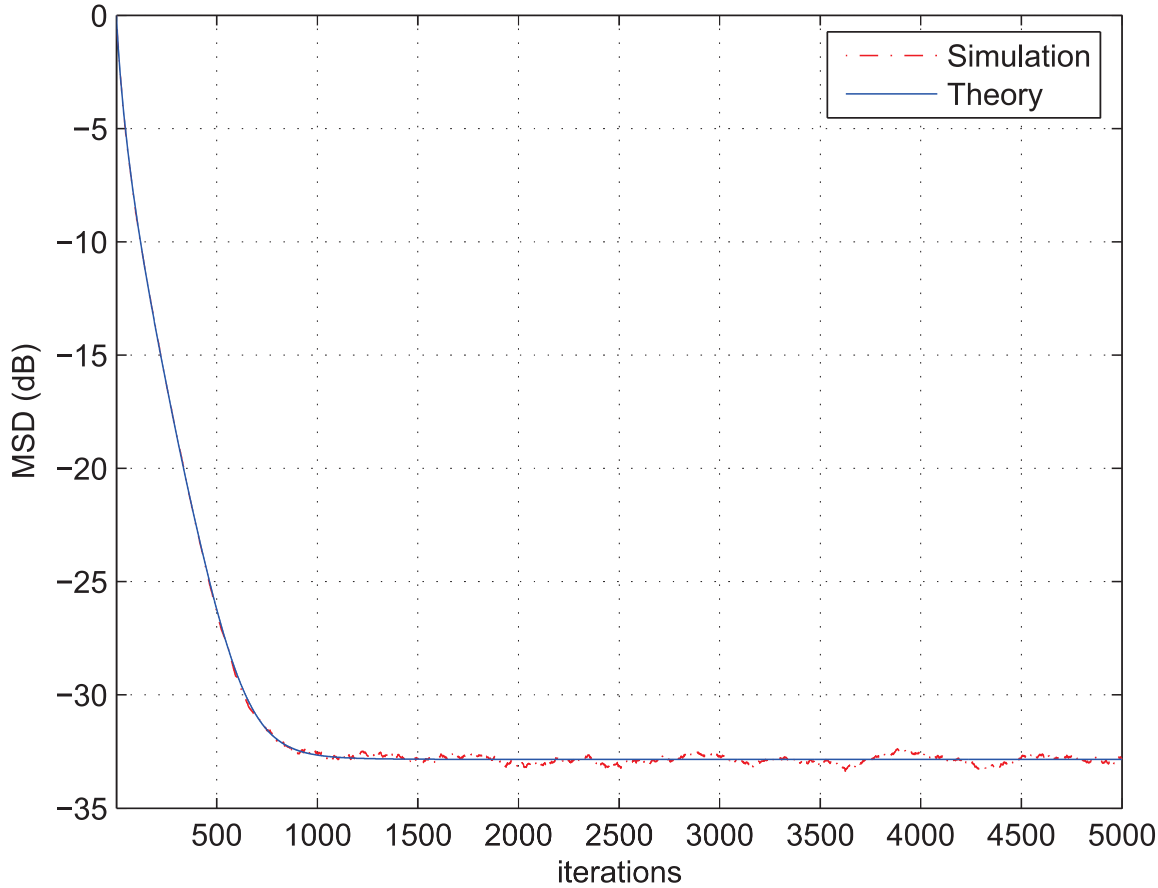

Figure 5.9: Tracking MSD of leaky LMMN in Uniform noise with weight variance 10^{-6}.

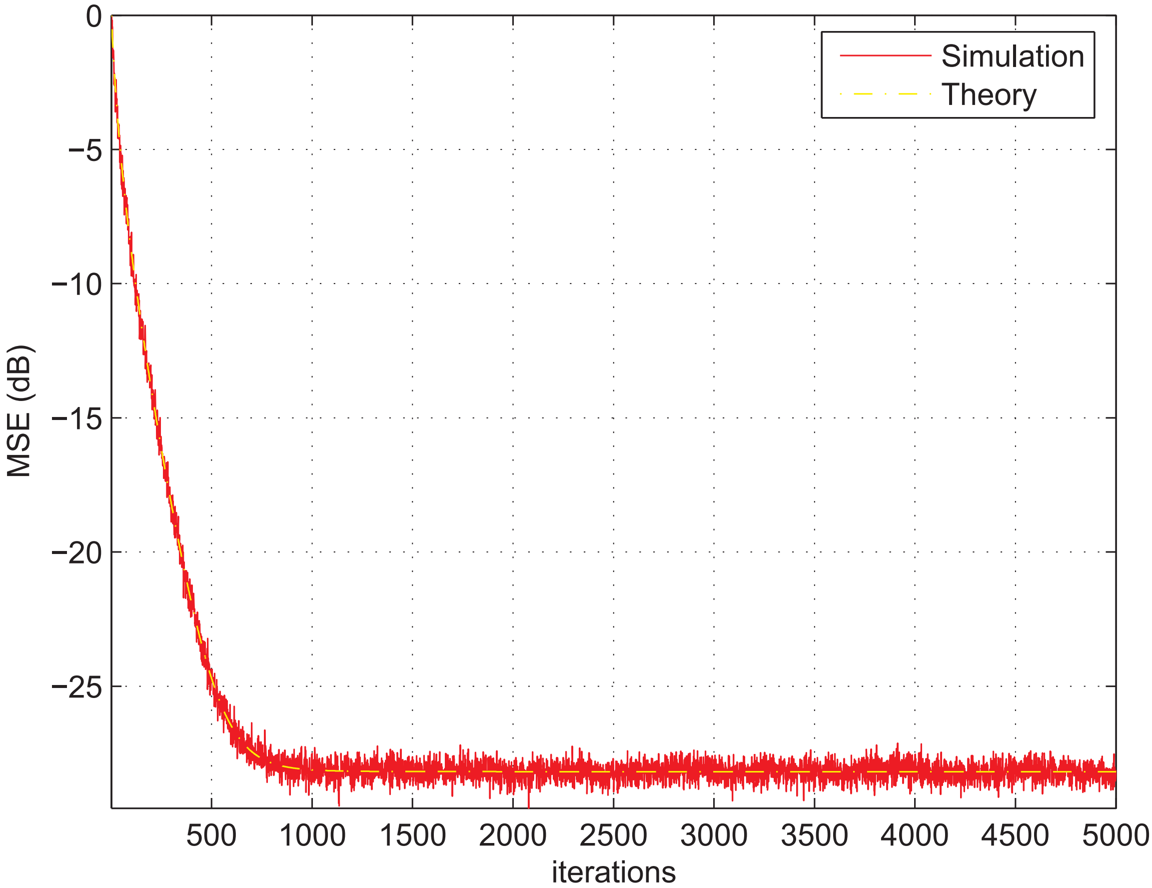

Figure 5.10: Tracking MSE of leaky LMMN in Uniform noise with weight variance 10^{-6}.

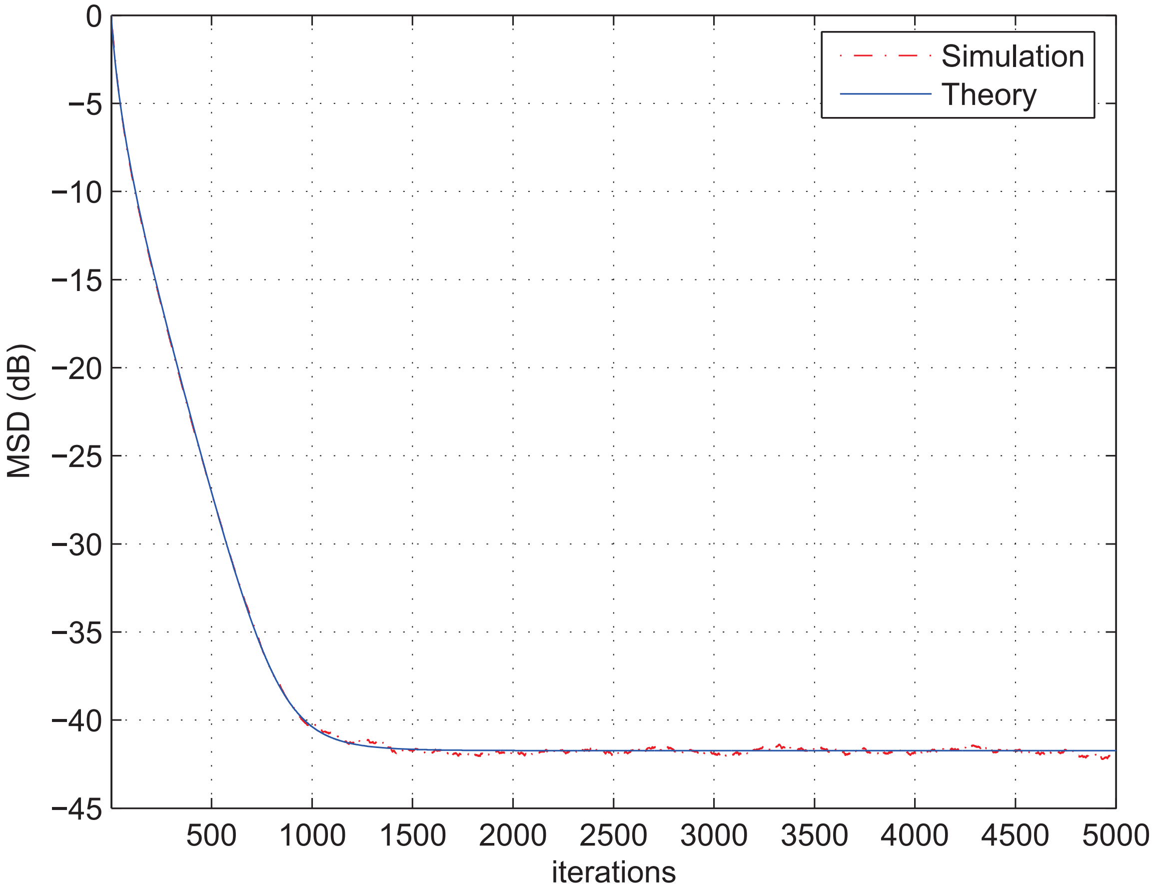

Figure 5.11: Tracking MSD of leaky LMMN in Uniform noise with weight variance 10^{-7}.

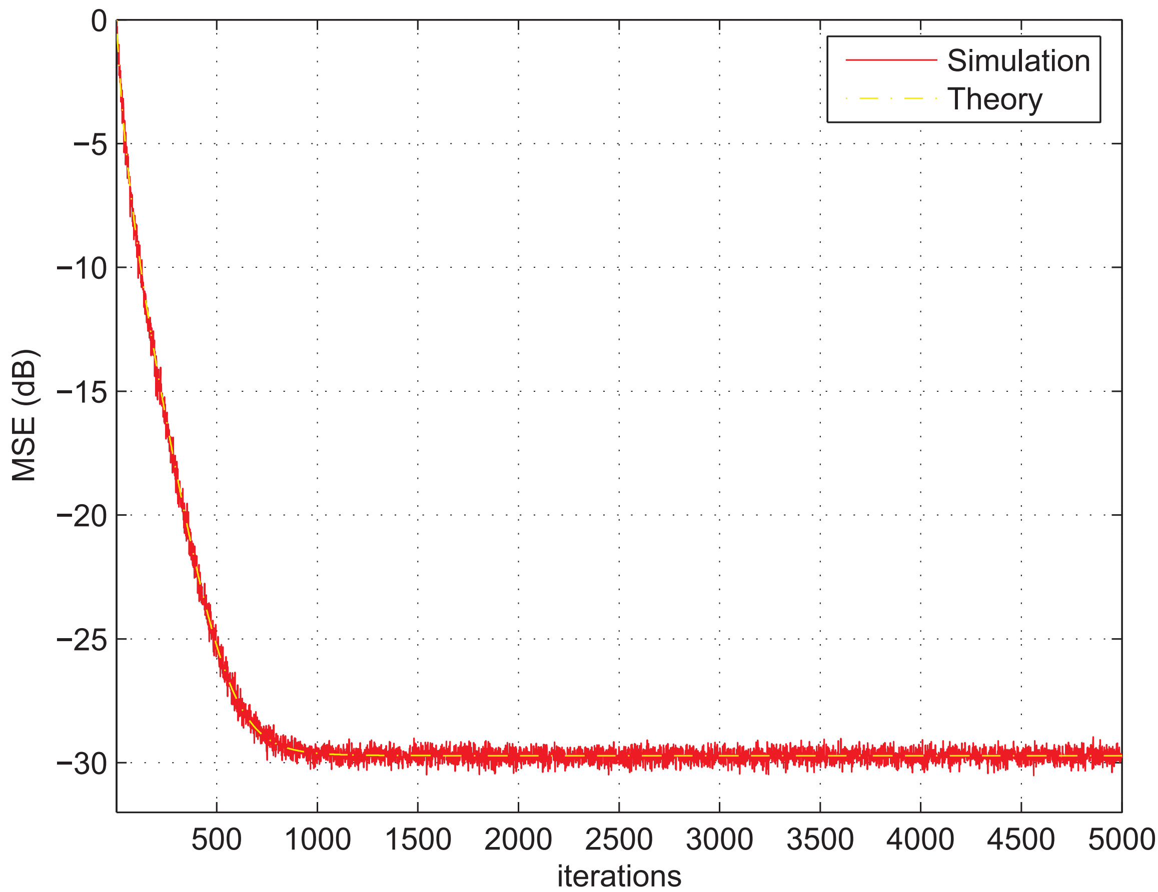

Figure 5.12: Tracking MSE of leaky LMMN in Uniform noise with weight variance 10^{-7}.

5.4 Comparison of the Variable Weight Mixed Norm algorithm with the Fixed Mixed Norm algorithm

As we are aware of the fact that the LMF algorithm has a faster convergence compared to the LMS whereas the later has desirable characteristics in the neighborhood of the optimum. Recently the utilization of a weighted sum of the two performance measures was proposed to combine the advantages of both in the mixed-norm algorithm [10]. The mixed norm LMS-LMF algorithm is dened by the following cost function:

$$\mathbf{J}_n = \alpha E[e_n^2] + (1-\alpha)E[e_n^4] \tag{5.13}$$

where the error is defined as

$$e_n = d_n - \mathbf{u}_n^T \mathbf{w}_n \tag{5.14}$$

The algorithm defined in Equation 5.15 has a fixed mixing parameter that is predetermined by the designer and hence will be unable to track variations in the environment. To overcome this difficulty, a time variation in the weight parameters is proposed [13] and its cost function is defined as:

$$\mathbf{J}_n = \alpha E[e_n^2] + (1-\alpha)E[e_n^4] \tag{5.15}$$

where α_n is the time varying mixing parameter which changes in accordance to the square of the time averaged estimate of the auto correlation of the e_n and e_{n-1} i.e.it is updated as follows [8]:

$$\alpha_{n+1} = \delta\alpha_n + \gamma p_n^2 \tag{5.16}$$

$$p_n = \beta p_{n-1} + (1-\beta)e_n e_{n-1} \tag{5.17}$$

where δ, β and γ are constants.The parameters δ and β, confined to the interval $[0, 1]$, are exponential weighting parameters that govern the quality of estimation and $\gamma > 0$.

Following the above discussion in this section the time-varying Leaky LMMN algorithm is compared to the Fixed Mixed Norm(FMN) LMS and LMF algorithms in both Uniform and Gaussian environments with a signal to noise ratio set at $10dB$ and $20dB$ respectively. The performance measure considered is the normalized weight error norm $10*log10||\mathbf{w}_n - \mathbf{w}_o||^2/||\mathbf{w}_o||^2$ and the results are obtained by averaging over 100 samples. In this we compare the time varying algorithm to the fixed mixed-norm algorithm for different values of mixing parameter α(constant).

In this the fixed mixed norm algorithm is considered with $\alpha = 0.8$ and $\alpha = 0.2$. The FMN algorithm with $\alpha = 0.8$ behaves almost similarly to the LMS algorithm whereas the FMN algorithm with $\alpha = 0.2$ gives a close relation to the LMF algorithm. It can be inferred from the Figures(5.13 - 5.16) that the proposed algorithm results in superior performance over the two versions of the FMN algorithm.This is a result of the fact that the mixing parameter for the proposed algorithm is time varying, which accomodates itself according to changes in the environment.

When we compare the proposed algorithm with the FMN algorithm with $\alpha = 0.8$, we observe that the later has the same convergence rate as that of the proposed algorithm but results in larger excess steady state MSE. On the other hand, when $\alpha = 0.2$ is considered for the FMN algorithm, this results in the same excess steady state MSE as the proposed algorithm, where the later has a faster speed of convergence.

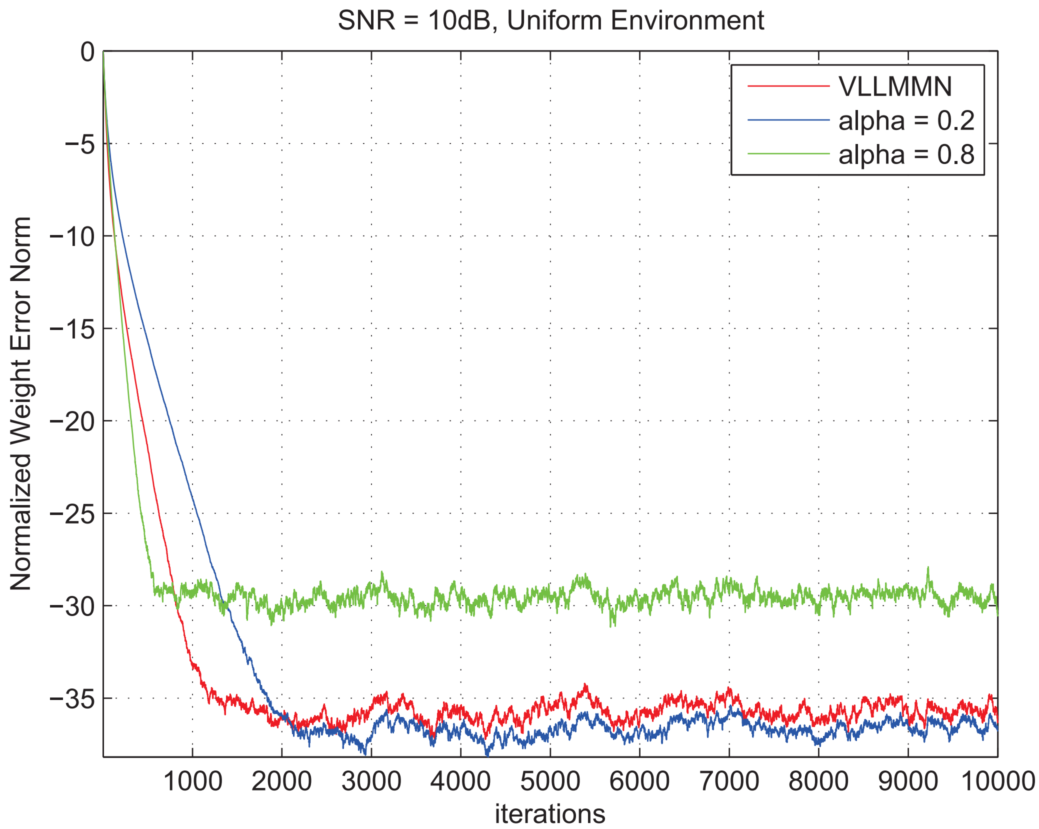

Figure 5.13: Behavior of the proposed algorithm and the FMN algorithm in Uniform noise and noise variance 0.1.

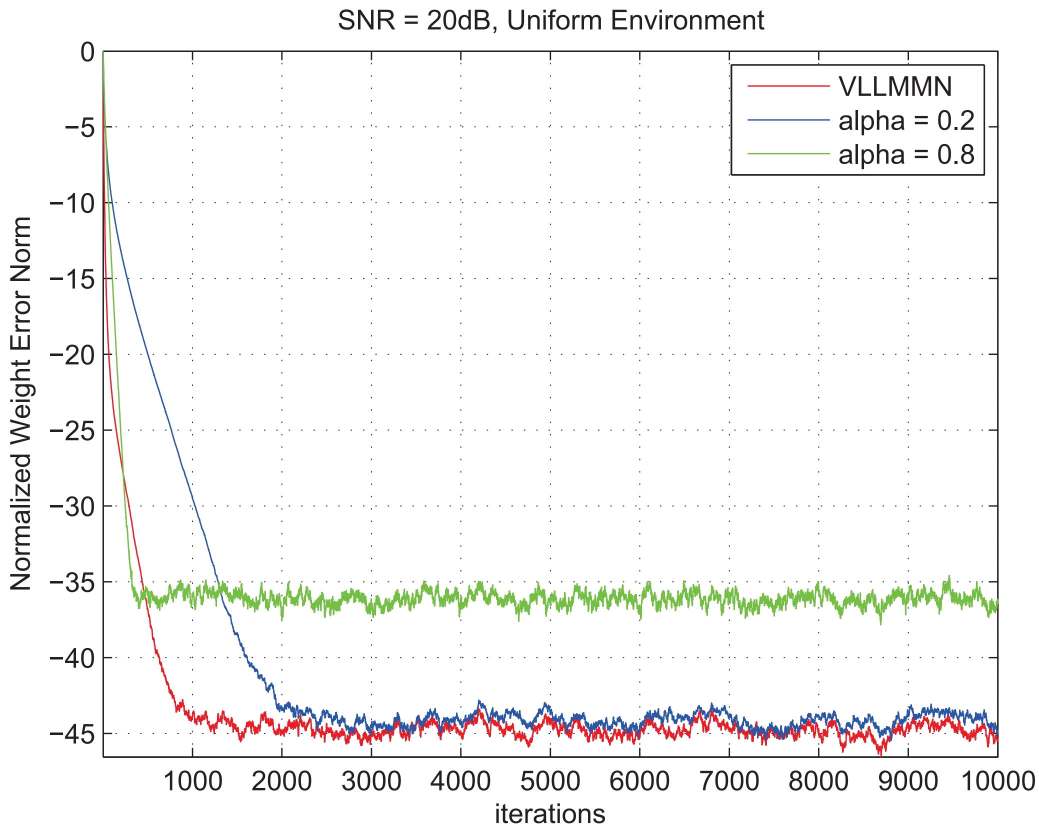

Figure 5.14: Behavior of the proposed algorithm and the FMN algorithm in Uniform noise and noise variance 0.2.

Figure 5.15: Behavior of the proposed algorithm and the FMN algorithm in Gaussian noise and noise variance 0.1.

Figure 5.16: Behavior of the proposed algorithm and the FMN algorithm in Gaussian noise and noise variance 0.2.

CHAPTER 6

THESIS CONTRIBUTIONS AND RECOMMENDATIONS FOR FUTURE WORK

6.1 Thesis Contributions

This work successfully presented the Leaky LMMN algorithm. This algorithm was analyzed in terms of its convergence properties, steady-state and tracking performances and transient behavior. The performance of the proposed algorithm has been supported by presenting the simulation scenarios. the major contributions of this thesis work are as follows:

1. A new LMMN variant with a leakage factor which mitigates weight drift.

2. The convergence analysis of the proposed algorithm derived in terms of the mean and mean square sense and as well as a model for estimating the time

evolution of the mean square error and the mean square deviation for the algorithm.

3. The steady state analysis of the algorithm carried as the limiting case of the transient behavior of the algorithm.

4. Tracking ability of the algorithm analyzed and the model for the time evolution of the algorithm in a non-stationary environment derived.

5. The analytical results compared with the experimental results which support the analysis.

6. The time varying Leaky LMMN algorithm is compared to the traditional LMMN in terms of the convergence speed in achieving the same steady state error.

7. Finally, the time varying Leaky LMMN algorithm is compared to the Fixed Mixed Norm algorithm with variable mixing parameter.

6.2 Recommendations for Future Work

There are a few suggestions regarding future work. In this thesis, a constant leakage factor was used which caused bias in the mean square error. However, by using the various techniques used for removing the bias in the case of Leaky LMS, we can find even better variants of the Leaky LMF that mitigate the weight drift problem without causing a bias. Furthermore, these variants are expected to perform better than their LMF counterparts in terms of steady state misadjustment. The work can even be extended to the subspace model which provides a leakage only in the unexcited modes thus introducing bias while retaining the low computational complexity.

REFERENCES

[1] S. U. H Qureshi. Adaptive equalization. *IEEE Proceedings*, 53:1349-1387, 1985.

[2] L. Makhoul. Linear prediction a tutorial review. *IEEE Proceedings*, 63(4):561-580, 1975.

[3] S. Haykin, Adaptive Filter Theory. Upper Saddle River, NJ:Prentice-Hall, 1996.

[4] A.H. Sayed, Fundamentals of Adaptive Filtering. New York, NY:Wiley Interscience,2003.

[5] J.G. Proakis, Digital Communications, McGraw-Hill, 2001.

[6] R. H. Kwong and E. W. Johnston. A Variable Stepsize LMS algorithm. *IEEE transactions on signal processing*, 35:122-127, 1988.

[7] R. W. Harris, D. M. Chabries, and F. A. Bishop. A Variable Stepsize(VS) algorithm. *IEEE transactions on Acoustic, Speech and Signal Processing*, 34:499-510, 1986.

[8] T. Aboulnasr and K. Mayyas. A Robust Variable Step-Size LMS Type Algorithm: Analysis and Simulation. *IEEE transactions on signal processing*, 45:631-639, 1997.

[9] A. Zerguine, C. F. N. Cowan, and M. Bettayeb. "Adaptive echo cancellation using least mean mixed-norm". IEEE Transactions signal processing., SP-45(5):1340-1343, 1997.

[10] J. A. Chambers, O. Tanrikulu and A. G. Constantinides. "Least mean mixed norm adaptive filtering". Electronic Letters, 30(19):1574-1575, 1994.

[11] J. Cioffo, "Limited Precision Effects in Adaptive Filtering", IEEE Trans. on Signals & Systems, CAS-34, no.7, pp. 821-833, July 1987.

[12] V.H. Nascimento and A.H. Sayed, "Unbiased and Stable Leakage-Based Adaptive Filters", IEEE Trans. Signal Processing, vol. 47, No.12, pp. 3261-3276, December 1999.

[13] T. Aboulnasr and A. Zerguine. Variable Weight Mixed-Norm LMS-LMF Adaptive Algorithm. *IEEE conference on signals, systems and computers*, 1:791-794, 1999.

[14] A. Zerguine, M. Bettayeb, and C. F. N. Cowan. Hybrid LMS-LMF scheme for echo cancellation. *IEEE Proceedings-VIS Image Signal Processing*, 146(4):173-180, 1999.

[15] E. Wallach and B. Widrow, "The Least Mean Fourth (LMF) Adaptive Algorithm and its Family," IEEE Trans. Information Theory, vol. IT-30, no. 2, March 1984.

[16] A. Zerguine, "Convergence Behaviour of the Normalized Least Mean Fourth Algorithm," Proceedings of the 34th Annual Asilomar Conf. on Signals, Systems, and Computers, pp. 275-278, Oct.29-Nov.1, 2000.

[17] N. J. Bershad. On the optimum data nonlinearity in LMS adaptation. *IEEE transactions on signal processing*, 34(1):69-76, 1986.

[18] S. C. Douglas and T. H. Y. Meng. Stochastic gradient adaptation under general error criteria. *IEEE transactions on signal processing*, 42(6):1352-1365, 1994.

[19] T. Y. Al-Naffouri, A. Zerguine, M. Bettayeb. A unifying view of error nonlinearities in LMS adaptation. *Proceedings of the 1998 IEEE ICASSP*, 3:1697-1700,1998.

[20] T. Y. Al-Naffouri, A. Zerguine, M. Bettayeb. Convergence properties of mixed-norm algorithms under general error criteria. *IEEE ISCAS '99*, 3:211-214, 1999.

[21] O. Tanrikulu and J. A. Chambers. Convergence and steady-state properties of the least mean mixed-norm(lmmn) adaptive filtering. *IEEE proceedings-VIS Image Signal Processing.*, 143(3):137-142, 1996.

[22] V.H. Nascimento and A.H. Sayed, "Unbiased and Stable Leakage-Based Adaptive Filters", IEEE Trans. Signal Processing, vol. 47, No.12, pp. 3261-3276, December 1999.

[23] T. Y. Al-Naffouri and A. H. Sayed, "Transient Analysis of Adaptive FIlters with Error Non-Linearities", IEEE Trans. Signal Processing, Vol.51, No.3, pp. 653-663, March 2003.

[24] T. Y. Al-Naffouri and A. H. Sayed, "Transient Analysis of Data-Normalized Adaptive FIlters", IEEE Trans. Signal Processing, Vol.51, No.3, pp. 639-652 March 2003.

[25] T. Y. Al-Naffouri and A. H. Sayed, "Mean Square Analysis of Normalized Leaky Adaptive FIlters", Proceedings of ICASSP, Salt Lake City, Utah, May 2001.

[26] P. I. Hubscher and J. C. Bermudez, " An improved Statistical Analysis of the Least Mean Fourth Algorithm", IEEE Trans. Signal Proc., Vol. 51, No.3, pp. 664-671, March 2003.

[27] A. Zerguine et al, " Convergence and Tracking Analysis of a Variable Normalized LMF (XE-NLMF) Algorithm", Signal Processing, Vol. 89, No. 5, pp. 778-790, May 2009.

[28] A.D. Poularikas, The Handbook of Formulas and Tables for Signal Processing. Boca Raton, FL:CRC Press,1999.

Vitae

- Mohammed Abdul Nasar.
- Nationality : Indian
- Born in Mahabubnagar, India on August 1st, 1985.
- Received Bachelor of Science in Electronics and Communications Engineering from SSJ Engineering College, Mahabubnagar, India in May 2006.
- Joined King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia as a Research Assistant in September 2007.
- Address : Hail Wazarat, Riyadh, KSA.
- Email: *nasar.mohammed@gmail.com*.
- Mobile No : (00966)-506475485