Neural Networks-based Decision Feedback Equalization using Lattice Structure

by

Ahmar Shafi

A Thesis Presented to the

FACULTY OF THE COLLEGE 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

May, 1999

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA

UMI®
800-521-0600

Neural Networks-based Decision Feedback Equalization using Lattice Structure

BY

Ahmar Shafi

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

May 1999

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS

DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by

AHMAR SHAFI

under the direction of his Thesis Advisor 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. Maamar Bettayeb (Chairman)

Dr. Azzedine Zerguine (Co – Chairman)

Dr. Maan Kousa (Member)

Dr. Saud Al – Semari (Member)

Dr. Hussain Al – Duwaish (Member)

Dr. Samir Al – Baiyat
Department Chairman

Dr. Abdullah M. Al – Shehri
Dean of Graduate Studies

Date 16/8/1999

Acknowledgments

In the very first place, I must thank to Allah the Almighty and the Most Merciful for His blessings throughout my life in general and in the course of this thesis in particular. May His peace and blessings be upon Prophet Muhammad and his family.

I acknowledge the support and facilities provided by King Fahd University of Petroleum and Minerals.

I would like to pay a high tribute to both of my thesis advisors, Dr. Maamar Bettayeb and Dr. Azzedine Zerguine, for their invaluable guidance and helpful ideas throughout the thesis work. Their appreciation and words of encouragement gave a new life to my efforts in hard times. They helped me a lot and I am really grateful to them. I would also like to thank my other three committee members, who also happen to be my teachers in different courses. In fact, I started working on lattice-based equalizers, the core of my thesis, when I was taking the course of Data Communications with Dr. Maan Kousa. The simulation strategies, which I have used in my thesis work, I learnt them mostly from Dr. Saud Al-Semari when I took two courses with him. I learnt a lot about artificial neural networks when I was doing a research project with Dr. Hussain Al-Duwaish on neural networks-based system identification. I would also like to thank these three teachers of mine for their kind evaluation of my thesis work and appreciate their constructive comments about the same.

I would also like to thank Uvais Qidwai for his invaluable assistance throughout my stay at KFUPM. I am bound to thank to my friends Rais-ul-Haque Adnan, Nayyer

Grami and Abdul Karim Bala for their kind help in various aspects and situations. I also wish to thank my friends Shafat Abrar and Zainulabadin for their constructive criticism of my work. I am thankful to Zeesahn Jilani, Nomaan Khan and Rafay Hassan for providing a wonderful company. It would be ungrateful if I do not acknowledge my friends Fareed, Ajmal, Nadeem, Faisal, Kashif and Junaid for their kind support in the form of allowing me to use the best available computing resources in EE-graduate computer lab.

Last but not least are the most important people in my life, my caring parents and grandmother, whom I can not thank enough for all their support, patience, and prayers. May Allah reward all of the above persons in His best way.

Contents

List of Figures

Abstract

Name: Ahmar Shafi
Title: Neural Networks-based DFE with Lattice structure
Major Field: Electrical Engineering
Date of Degree: May 1999

The heavily distorting channels limit the use of linear equalizers and the use of the nonlinear equalizers becomes admissible. Most nonlinear equalizers, e.g. Voltera type, are usually too complicated to meet real-time processing demands. Neural networks-based equalizers, especially the Multi-Layer Perceptron (MLP) based equalizers, are computationally efficient alternative to currently used nonlinear filter realizations. The drawback of the MLP-based equalizers is their slow rate of convergence, which limit their use in practical systems.

In this thesis, we have studied and analyzed one way to increase the convergence rate of the MLP-based equalizer by employing the well-known fast-converging lattice filter processor module at the input of the MLP equalizer. Lattice algorithms are known for their fast convergence and insensitivity to the eigen-value spread of the channel correlation matrix.

The performance of the new pre-orthogonalized neural-network based adaptive nonlinear filters is evaluated by applying them as adaptive equalizers for linear, nonlinear and time-variant data communication channels in both additive White and Colored Gaussian noise environments for PAM signaling. The powerful RLS algorithm together with the lattice filter lead to the improvement in the performance of the MLP-based equalizers in terms of convergence rate, Mean-Squared Error (MSE), and bit error rate (BER).

The analysis of the proposed scheme is done by means of computer simulations and the results are presented in the form of scatter diagrams, eye diagrams, learning curves and bit error rate (BER) curves for different equalizer configurations and channel conditions.

King Fahd University of Petroleum & Minerals

Dhahran 31261, Saudi Arabia

الاسم: أنور شفيع

العنوان: DFE المعتمد على الشبكات العصبية مع بنية شبيكية

التخصص: الهندسة الكهربائية

التاريخ: مايو ١٩٩٩م

إن القنوات الشديد التشويه للمعلومات تحد من استخدام المعدلات الخطية فيصبح استخدام المعدلات الغير خطية معقولا. معظم المعدلات الغير خطية معقدة جدا, مثلا معدل فولتيرا, مما يجعلها غير مناسبة للاستخدام الحقيقي. إن المعدلات المعتمدة على الشبكات العصبية وخصوصا الملاحظ المتعدد الطبقات, فعالة حسابيا مما يجعلها بديل للمعدلات الغير خطية. ولكن من عيوب هذه المعدلات المعتمد على الملاحظ المتعدد الطبقات أنها بطيئة التقارب مما يحد من استخداماتها العملية.

في هذه الرسالة, قمنا بدراسة وتحليل طريقة لزيادة سرعة تقارب معدلات الملاحظ المتعدد الطبقات باستخدام مرحلة متبورة وسريعة التقارب تدعى معالج مرشح شبيكي في مدخل معدل الملاحظ المتعدد الطبقات. الخوارزميات الشبيكية معروفة بسرعة التقارب وعدم حساسيتها بانتشار القيم الهامة لمصفوفة ارتباط القناة.

قيم أداء المرشحات الجديدة المتكيفة الغير خطية المعتمد على الشبكات العصبية المتعامدة مبدئيا باستخدامها في المعدلات المتكيفة لقنوات إرسال المعلومات الخطية والغير خطية والمتغيرة زمنيا في وجود تشويش جاوسي ابيض وملون بالنسبة لإشارات PAM. إن قوة خوارزمية RLS بالإضافة إلى المرشح الشبيكي أدى إلى تحسن كبير في أداء المعدلات المعتمدة على الملاحظ المتعدد الطبقات وهذا التحسن يظهر في سرعة التقارب و انخفاض متوسط مربع الخطأ وكذلك انخفاض معدل الخطأ في الخانة.

ولقد جرى دراسة وتحليل هذه الطريقة المقترحة باستخدام محاكاة الكمبيوتر وعرضت النتائج في أشكال رسوم بيانية للانتشار و رسوم بيانية للعين ومنحنيات التعليم ومنحنيات معدل الخطأ في الخانة لمعدلات في حالات مختلفة وقنوات مختلفة.

درجة الماجستير في العلوم

جامعة الملك فهد للبترول والمعادن

الظهران, المملكة العربية السعودية

مايو ١٩٩٩م

Chapter 1

INTRODUCTION

1.1 Equalization

In efficient digital communication systems, the effect of each symbol transmitted over time dispersive channel extends beyond the time interval used to represent that symbol. The distortion caused by the resulting overlap is called Intersymbol Interference (ISI). This distortion is one of the major obstacles to reliable high-speed data transmission over low-background-noise channels of limited bandwidth [32], [43]. All pulse-modulation systems, including frequency-shift keying (FSK), phase shift keying (PSK), and quadrature amplitude modulation (QAM) give rise to intersymbol interference.

This has been a trend for the past three decades to design and develop data communication systems that make efficient use of the available channel bandwidth. The design objective has been to achieve highest possible transmission rate at a specified probability of error. A primary limitation in attempts to achieve a high transmission rate is the time dispersion suffered by the signal at the receiving end, which is transmitted through a band-limited channel. In data transmission, the time dispersion

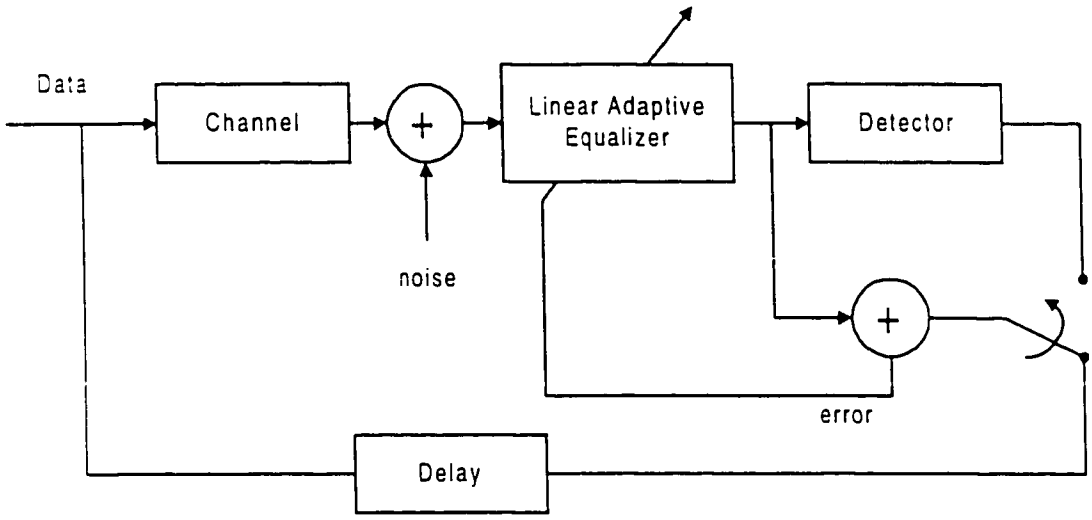


Figure 1.1: The block diagram of over all communication system with channel equalizer

imparted on the transmitted signal results in an overlap in time between successive symbols known as (ISI). The term "Equalizer" has been used to describe filters used to compensate for such distortions in the amplitude and delay characteristics of the channel.

Equalization techniques may be divided into two general categories: Linear and Non-linear equalizers. A linear equalizer is usually implemented as a transversal (FIR) filter with adjustable tap coefficients. The adjustment of the tap coefficients is carried out adaptively during the transmission of information based on the minimization of a cost function. The error signal needed to adjust the tap coefficients, is formed by subtracting the decision at the output of the detection device from the equalizer output [Fig. 1.1].

However, at and during start-up, a short training sequence is used for the calculation of this error signal.

Nonlinear equalizers [43], [42], [11], differ from linear equalizers in applications where the channel distortion is too severe for a linear equalizer to handle. In particular, a linear equalizer does not perform well on channels with deep spectral nulls in their amplitude characteristics or with nonlinear distortion. In an attempt to compensate for the channel distortion, the linear equalizer places a large gain in the vicinity of the spectral null and, as a consequence, significantly enhances the additive noise present in the received signal.

Several nonlinear equalization techniques have been developed over the past three decades [42] and [41]. The computational complexities of some of these methods, however, grows exponentially with ISI length (i.e. the memory length of the channel), and the degree of nonlinearity. Consequently, they are impractical for channels in which the intersymbol interference spans ten or more symbols, or the nonlinearity is of degree higher than three [40].

1.2 Artificial Neural Networks

Artificial neural networks (ANNWs) are mathematical models of biological neural networks. The fundamental idea of ANNWs is to organize many simple identical processing elements into layers to perform more sophisticated tasks. Because of their capability for efficient modeling of arbitrary nonlinearities, there has been recent interest in employing ANNWs in adaptive equalization for data communication

channels [7], [13], and [51]. In this case, the linear adaptive filter is replaced by a neural network.

Artificial neural networks use non-linear computational elements to solve a complex problem. The primitive neural network is composed of a single neuron, which is also called single layer perceptron. Earlier work on single layer perceptron was limited owing to the fact that only linear decision boundaries could be formed in signal space. However, later developments showed that multiple layers could be used to form much more complex (non-linear) decision boundaries [58], [12] . If the number of layers in the perceptron is increased to three, a considerably more complex decision regions having highly non-linear boundaries can be formed.

A multilayer perceptron consists of several hidden layers of neurons that are capable of performing complex, non-linear mappings between the input and output layers. The hidden layers provide the capability to use nonlinear sigmoid function to create intricately curved partitions of space. The three-layer perceptron (two hidden layers, and an output layer) is sufficient for the non-linear Decision Feedback Equalizer (DFE) structure because a three- layer perceptron can generate arbitrarily complex, non-linear decision regions [54].

1.3 Lattice Filters

It is widely recognized that when the input samples entering the equalizer are highly correlated, convergence to the optimum filter coefficients is slow when using the Least Mean Square (LMS) algorithm [35], [46]. This is due to the fact that the convergence

speed of the LMS algorithm depends upon the number of coefficients used and the eigen-value spread of the channel autocorrelation matrix [14].

Fast convergence results when successive corrections to the tap coefficients of the equalizer are adaptively decorrelated. A class of adaptive algorithms was proposed by Satorius and Alexander for channel equalization [46] for the channels that exhibit high correlation in the input data samples. These algorithms are called Adaptive Lattice (AL) algorithms, which require a number of operations per update that is proportional to the length of the filter and they generate a set of orthogonal set of signal components that can be used as inputs to the equalizer. The generation of these components is done through a Gram-Schmidt type of orthogonalization procedure.

One useful property related to the adaptive lattice filter concerns the rate of convergence of its reflection coefficients to their optimum values, which are independent of the eigen value spread of the autocorrelation matrix of the filter input. Another advantage of the lattice structure is, due to the orthogonality property, its ability to increase or decrease the number of stages without affecting the parameters of the previous stages [46].

1.4 Channel Distortions

According to Shannon's theory [47] , the upper limit of transmission rate over a band-limited channel with additive white Gaussian noise (AWGN) is the channel capacity which decreases monotonically with decrease in signal to noise ratio (SNR). But, in real world applications there are other channel distortions affecting data transmission besides AWGN and include linear and nonlinear distortions. The equalization

techniques described in this thesis attempt to reduce the effect of linear and nonlinear distortions.

1.4.1 Linear Channel Distortion

For decades, engineers have developed a band-limited linear filter model for most communication channels. The ideal characteristics of a distortion free band-limited channel is that it has a constant amplitude response and a linear phase response within its bandwidth and is zero elsewhere, i.e.,

$$C(f) = \begin{cases} |C(f)|\, e^{j\theta(f)} & if\ |f| < W \\ 0 & otherwise \end{cases} \tag{1.1}$$

where W is the bandwidth,

$$|C(f)| = const. \tag{1.2}$$

is the amplitude response and

$$\tau(f) = -\frac{1}{2\pi}\frac{d\theta(f)}{df} = const. \tag{1.3}$$

is the envelope delay characteristic.

When either $|C(f)|$ or $\tau(f)$ changes with frequency, the channel distorts the transmitted signal in amplitude and/or delay. Such amplitude and delay distortions are usually referred to as linear channel distortions.

1.4.2 Nonlinear Channel Distortion

Though the linear assumption for communication systems has been widely adopted and used, in many cases nonlinear distortions significantly contribute to the system degradation. This can be clarified through the following real examples.

In telephone channels, the nonlinearity arises principally from an inaccurate companding (compressing-expanding) process in telephone transmission. It is conjectured that the error rate performance of data transmission systems operating at greater than 4800 bps is almost entirely determined by nonlinear distortions [40].

Another example of nonlinear channels is found in satellite communications. The satellite amplifier is usually driven near to the saturation point and exhibits highly nonlinear characteristics [52], [49].

Also, in the magnetic recording process, the use of magneto-resistive (MR) read causes the "positive-negative peak asymmetry", which acts as a nonlinear distortion in the magnetic storage channel [40].

1.5 Problem Statement

Fig. 1.2 depicts a DFE based on multilayer perceptron (MLP) [12], [51] . It is found that, the MLP-based DFE converges to a lower value of steady state mean squared error (SSMSE) as compared to the conventional gradient DFE. Thus, the MLP-based DFE provides better BER performance in heavily distorting channels. [51]. On the other hand, since the Back Propagation (BP) algorithm is no more than a generalized LMS algorithm, it suffers from the same problems as the linear LMS algorithm does and, in particular, exhibits relatively slow convergence when affected by channels with spectral nulls in their frequency response. Also, taking into account the complex nonlinear relationship between the inputs and the outputs of the MLP, the application of the BP algorithm can lead to a local minimum and never reach the global optimal point [21].

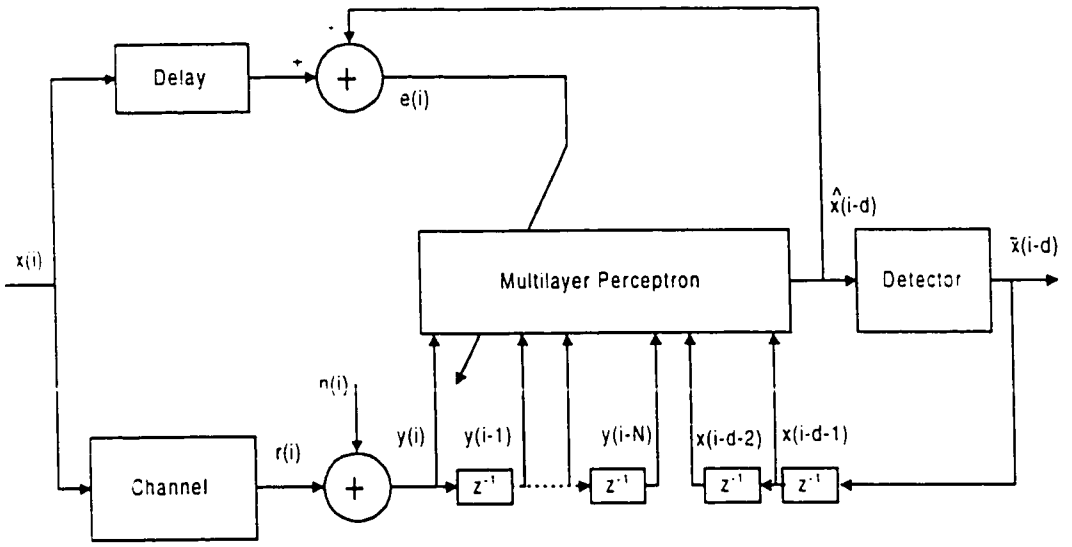


Figure 1.2: The block diagram of Multi-Layer Perceptron (MLP) Equalizer

Because of the above-mentioned problems, the past decade has been an area of intensive research to develop better training algorithms for the MLP. So far several fast algorithms have been proposed, but these algorithms achieve fast convergence at the cost of very high computational complexity.

1.6 Proposed Work

A powerful means of improving the performance of the MLP-based DFE, independently of the eigen-value spread of the autocorrelation matrix of the input signal, is through the use of a pre-orthogonalization scheme. This can be achieved using a lattice filter module at the input to the MLP-based DFE. In this case, the effect of the eigen-value spread will be reduced substantially. Ultimately, improvements in

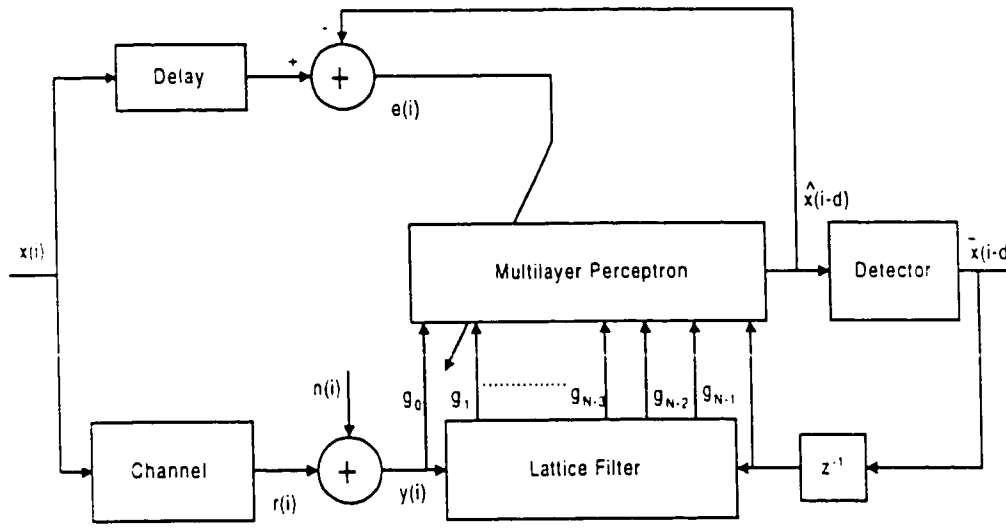


Figure 1.3: The block diagram of Multi-Layer Perceptron (MLP) Equalizer with Lattice filter pre-processor at the input of MLP

convergence rate, bit error rate (BER) and steady state mean-squared error (MSE) values will be achieved.

Since orthogonalization of the input signal is known to dramatically improve the convergence rate of the LMS-based algorithm [46], [34], [35], [27], the same idea is applied to the MLP-based structures.

Lattice filters are well known for their orthogonalization properties and fast rate of convergence. In this thesis, we study and analyze the effect of pre-orthogonalizing the input data to the equalizer, as shown in Fig. 1.3, by employing the famous lattice filter module at the input of the equalizer.

Our proposed work, therefore will be based on the above argument to develop such equalizer and to evaluate its performance in both the additive white Gaussian noise

(AWGN) and additive colored Gaussian noise (ACGN) environments for the following performance indices:

- Convergence rate
- Steady state mean squared error (SSMSE)
- Bit error rate (BER)

Computer simulations are used to evaluate the improvements in the performance of MLP over the conventional MLP-DFE.

1.7 Thesis Organization

This thesis is organized as follows. Chapter 1 gives an introduction to the problem and the proposed work. Chapter 2 deals with the description of adaptive equalization. It also highlights the main points of the Lattice filters and discusses their properties. Chapter 3 elaborates on the artificial neural networks (ANNW's), mainly the description of Multilayer Perceptron and Back Propagation Algorithm is addressed in this chapter. Then the use of ANNW's as a tool for adaptive channel equalization is presented.

Chapter 4 is devoted to the performance evaluation of the proposed scheme in static channels having different eigen-value spreads. The system is analyzed in Additive White Gaussian Noise and Additive Colored Gaussian Noise environments. Results are given in the form of learning curves and bit error rate curves for the quantitative analysis of the system. For the qualitative measure of the system's performance, the eye diagrams and constellation diagrams are also given for different channel conditions and at different stages of the over-all system.

Chapter 5 presents the simulation results for nonlinear channels having different degrees of nonlinearities as well as for the time varying channels. Simulation results show that significant performance improvement is achieved by decorrelating the MLP equalizer input data for the case of nonlinear channels and the use of the RLS algorithm for lattice coefficients update result in a better tracking of the time varying characteristics of the time varying channels.

The conclusion, some suggestions and recommendations for further work are given in chapter 6.

Chapter 2

ADAPTIVE CHANNEL EQUALIZATION

2.1 Introduction

A signal transmitted through a communication channel is usually corrupted by sources like Gaussian thermal noise and additive and/or multiplicative noise. In digital communications, the distortion caused by the multipath resulting in time delay and phase transformation of the transmitted signal causing (ISI), is also very severe. To mitigate this effect, channel equalization is used. For both time-varying and static channels. the equalization must be adaptive. Equalization dates back to the use of loading coils to improve the characteristics of twisted-pair telephone cables for voice transmission. In its broad sense, the term "equalizer" applies to any signal processing device designed to deal with (ISI). In radio channels, ISI is due to multipath propagation. which may be viewed as transmission through a group of channels with

different relative amplitudes and delays [42]. One special requirement of radio channel equalizers is that they be able to track the time-varying fading characteristics [43].

In a digital communication system, the nonuniform transmission characteristics of the channel causes intersymbol interference. The distortion of data pulses by the channel results in these pulses being smeared out in time so as to overlap other transmission pulses. This intersymbol interference is one of the direct chief degrading factors in the present systems and becomes the determining factor in the design of high rate systems.

In this section, we will present a model that characterizes the ISI. The digital modulation methods to which this treatment applies are PAM, PSK, and QAM. The transmitted signal for these three types of modulation may be expressed as

$$s(t) = v_c(t)\cos 2\pi f_c t - v_s(t)\sin 2\pi f_c t$$

$$= Re[v(t)\, e^{j2\pi f_c t}] \qquad (2.1.1)$$

where $v(t) = v_c(t) + jv_s(t)$ is called the equivalent low-pass signal, f_c is the carrier frequency and Re denotes the real part of the quantity on the brackets.

In general, the equivalent low pass signal is expressed as

$$v(t) = \sum_{n=0}^{\infty} I_n\, g(t-nT) \qquad (2.1.2)$$

where g is the basic pulse shape that is selected to control the spectral characteristics of the transmitted signal, I_n the sequence of transmitted information symbols selected from a signal constellation consisting of M points, and T is the signal interval.

The signal s(t) is transmitted over a bandpass channel that can be characterized by an equivalent low pass frequency response C(f). Consequently, the equivalent low pass received signal can be represented as

$$r(t) = \sum_{n=0}^{\infty} I_n\, h(t-nT) + \omega(t) \qquad (2.1.3)$$

where h(t) = v(t) * c(t), and c(t) is the impulse response of the equivalent low pass channel, the * denotes convolution, and ω(t) represents the additive noise in the channel.

To characterize the ISI, suppose that the received signal is passed through a receiving filter at the rate 1/T samples/sec. In general, the optimum filter at the receiver is matched to the received signal pulse h(t), and its output is given by

$$y(t) = \sum_{n=0}^{\infty} I_n\, x(t-nT) + v(t) \qquad (2.1.4)$$

where x(t) is the signal pulse response of the receiving filter and v(t) is the response of the receiving filter to the noise w(t).

Now, if y(t) is sampled at times t=kT, k=0,1,2,....., we have

$$y(kT) = y_k = \sum_{n=0}^{\infty} I_n\, x(kT - nT) + v(kT) \qquad (2.1.5)$$

$$= \sum_{n=0}^{\infty} I_n\, x_{k-n} + v_k \quad k = 0,1,2,\ldots$$

$$= I_k + \sum_{\substack{n=0 \\ n\neq k}}^{\infty} I_n\, x_{k-n} + v_k$$

The term I_k represents the desired information symbol at the kth sampling instant, and the term $\sum_{\substack{n=0 \\ n\neq k}}^{\infty} I_n\, x_{k-n}$ represents the ISI, and v_k is the additive noise at the kth sampling instant.

2.2 Linear Equalizers

The most common type of channel equalizer used in practice to reduce ISI is a linear transversal filter with adjustable coefficients [Fig. 2.1]. Adaptive equalizers update their parameters on a periodic basis during the transmission of data and, thus, they are capable of tracking a slowly time-varying channel response. The corrupted data is input to the delay line of the equalizer. The delayed data is then convolved with the equalizer coefficients to get the estimate of the transmitted symbol. The detector quantizes the estimate to the nearest symbol in the signal space. Depending upon the mode of operation, the error signal is generated by subtracting the estimated symbol from either the locally generated true symbol for training mode or from the detector's output for the decision directed mode. This error signal with suitable weighting is then used to update the coefficients of the equalizer.

Linear equalization was shown to be effective in practical applications involving linear distortion [42]. There are situations, however, where such linear transversal FIR equalizer is incapable of performing well. For instance, if the amplitude characteristics of the channel has nulls, the amplitude characteristics of the equalizer becomes very large in the vicinity of the nulls. Consequently, it will enhance the additive noise. Also, linear equalizer will not perform well for nonlinear channel distortions.

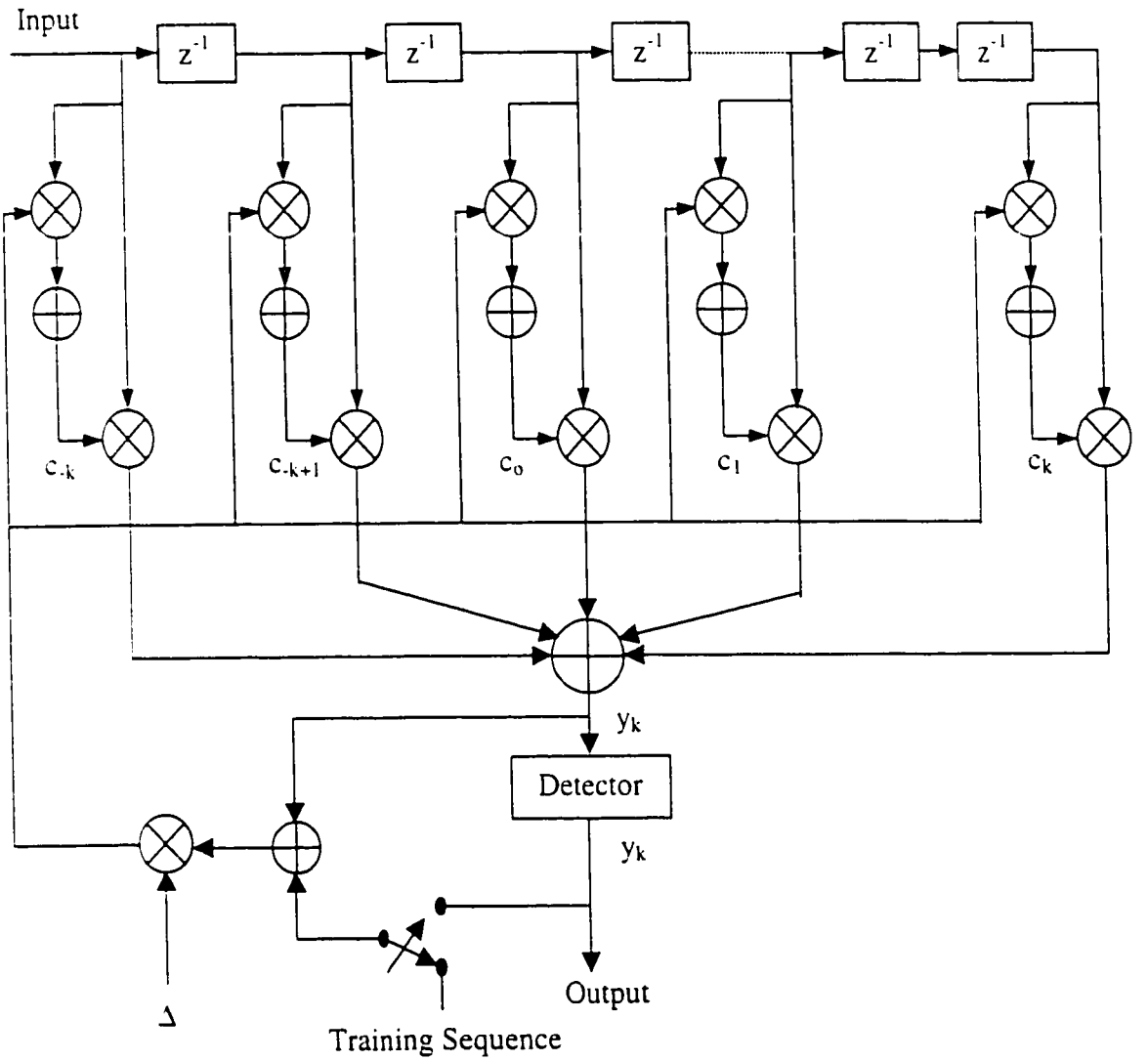


Fig. 2.1 Linear Adaptive Equalizer based on MSE

2.3 Nonlinear Equalizers

For decades, people have been aware of nonlinear phenomenon in communications systems. However, due to the lack of convenient mathematical tools, not until late 1970s did people begin to pay more attention to the nonlinear channel modeling problem and begin to work on techniques for combating such nonlinear distortions. There are several nonlinear equalizers, such as the decision feedback equalizer (DFE), the probabilistic symbol-by-symbol equalizer and the maximum-likelihood sequence estimation [42]. Only, the DFE is briefed here and this is due to its application in this work.

2.3.1 Decision Feedback Equalizers

Linear equalizers are effective only on channels where ISI is not severe and they perform poorly on channels having spectral nulls [42]. A Decision Feedback Equalizer (DFE) is a non-linear equalizer that is widely used in situations where ISI is large. It has been proved theoretically and experimentally that the DFE performs significantly better than a linear equalizer of equivalent complexity [11].

The basic idea of the DFE is that if the values of the symbols already detected are assumed to be correct, then the ISI contributed by these symbols can be canceled exactly by subtracting past symbol values with appropriate weighting from the equalizer output [42]. The feedback part has a role of reshaping the received signal so that the ISI at the output of the feedforward filter is causal, and the current pulse height is as high as possible with respect to the residual trailing ISI, which is then subtracted in the feedback part without noise enhancement.

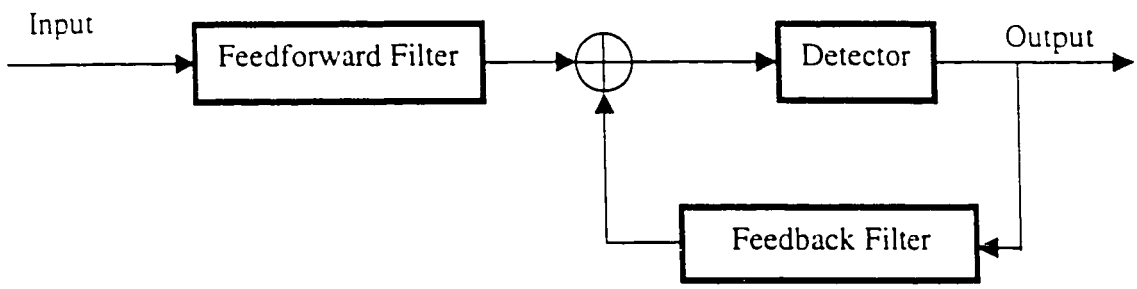


Fig. 2.2 Block diagram of decision feedback equalizer

The DFE consists of two filters [43], as depicted in Fig. 2.2. The first filter is called a *feedforward filter*. Where the input is the received ISI corrupted signal from the channel. The second filter is a *feedback filter*, where its input is the previously detected symbols. The output of the feedback filter is subtracted from the output of the feedforward filter to form the input to the detector as shown in Fig. 2.2. Thus the input to the detector is

$$z_m = \sum_{n=0}^{N1} c_n x_{m-n} + \sum_{n=1}^{N2} b_n I_{m-n} \qquad (2.3.1)$$

where $\{c_n\}$ and $\{b_n\}$ are the adjustable coefficients of the feedforward and feedback filters, respectively. I_{m-n}, $n=1,2,....N_2$, are the previously detected symbols, N_1+1 is the length of the feedforward filter, and N_2 is the length of the feedback filter. Based on the input z_m, the detector decides which of the possible transmitted symbols is closest in distance to the transmitted signal. The non-linear characteristics of the detector that provides the input to the feedback filter, make the DFE a nonlinear equalizer.

The tap coefficients of the feedforward and feedback filters are selected to minimize some desired performance measure, usually the Mean Squared Error (MSE), and a Stochastic Gradient Algorithm (SGA) is commonly used to implement an adaptive DFE [43].

The eigenvalue spread of the autocorrelation matrix of an equalizer's input data vector plays an important role in the tracking ability and performance of the equalizer. The convergence rate of the LMS algorithm mainly depends upon this spread. If the channel has nulls in its frequency response i.e. (the eigen-value spread is very large), the convergence of the linear equalizer will be very slow. It can also be shown that this is also true for the LMS DFE [51], [12].

2.4 Equalizer Updating Algorithms

2.4.1 The Mean-Squared-Error (MSE) Criterion and Least Mean Squared (LMS) Algorithm

2.4.1.1 The Mean Squared Error Criterion

In the MSE criterion, the tap coefficients {Ci}of the equalizer are adjusted to minimize the mean square value of the error

$$e_k = d_k - y_k \tag{2.4.6}$$

where d_k is the desired signal and y_k is the output signal of the equalizer defined by

$$y_k = \sum C_i \, r_{k-i}$$

$$= R_k^T C \qquad (2.4.7)$$

where R_k is the input to the equalizer, and

$$R_k = [r_0\ r_1 \ldots r_k]^T$$

In most practical instances, the adaptive process is oriented towards minimizing the mean squared value, or average power of the error signal. The performance index for the MSE criterion is defined as

$$J = E[e_k^2] \qquad (2.4.8a)$$

Or equivalently

$$J = E[(d_k - y_k)^2] \qquad (2.4.8b)$$

When the MSE is expressed in terms of the tap coefficients of the equalizer [42], the equation becomes

$$J = J_{min} + (C - C_{op})^T A\ (C - C_{op}) \qquad (2.4.9)$$

Where,

$$A = E[R^T_k R_k]$$

2.4.1.2 The LMS Algorithm

In the LMS algorithm, the tap coefficients of the equalizer are adjusted so as to minimize the mean-square-error (MSE) which is the sum of the squares of all the ISI terms plus the noise power at the output of the equalizer [56], [57]. Therefore, the equalizer based on LMS algorithm maximizes the signal-to-distortion ratio at the output within the constraints of the equalizer length and delay.

The solution for C_{op} involves inverting the input correlation matrix **A**. Alternatively, an iterative procedure may be used to determine C_{op}. The simplest iterative procedure is the method of steepest descent, in which one begins by choosing an initial tap-

coefficient vector, which corresponds to some point on the performance surface in the N-dimensional space of coefficients. The initial gradient vector G_0 is then computed at this point of the performance surface , and each tap coefficient is changed in the direction opposite to its corresponding gradient component. Then, succeeding values are obtained according to the relation

$$C_{k+1} = C_k - \Delta G_k , \qquad k = 0,1,2,\ldots \qquad (2.4.17)$$

Where $\mathbf{C}_k$ represents the set of the coefficients at the kth iteration , and Δ (step size) is a positive number chosen small enough to ensure convergence of the iterative procedure. If the minimum MSE is reached for some $k = k_0$, then $\mathbf{G}_k$ is equal to zero so that further change occurs in the tap coefficients. In general, J_{min} cannot be attained for a finite value of k_0 with the steepest descent method, however, it can be approached as desired for some finite value of k_0.

Given that the gradient vector $\mathbf{G}_k$ depends on both the autocorrelation matrix **A** and the vector **B** of cross correlation, this makes the steepest decent difficult for determining the optimum tap coefficients. Instead, estimates of the gradient vector may be expressed in the form

$$C_{k+1} = C_k - \Delta \hat{G}_k \qquad (2.4.18)$$

Where $\hat{G}_k$ denotes an estimate of the gradient vector $\mathbf{G}_k$, and is defined as

$$\hat{G}_k = -e_k R_k \qquad (2.4.19)$$

since $E\{ \hat{G}_k \} = \mathbf{G}_k$, the estimate $\hat{G}_k$ is an unbiased estimate of the true gradient vector. The LMS for recursively adjusting the tap coefficients of the equalizer is then expressed as

$$C_{k+1} = C_k + \Delta e_k R_k \qquad (2.4.20)$$

2.4.2 Least Squares Criterion and Recursive Least Squares (RLS) Algorithm

2.4.2.1 Least Squares Criterion

The convergence rate of gradient-based LMS algorithm is very slow, especially when the eigen-values of the input covariance matrix have a very large spread. In order to achieve faster convergence, complex algorithms, which involve additional parameters, are used. Faster converging algorithms are based on "least squares" approach, as opposed to the statistical approach used in LMS algorithm. That is, rapid convergence relies on error measures expressed in terms of a time average of the actual received signal instead of the statistical average. This leads to the family of powerful, complex, adaptive signal processing techniques known as recursive least squares (RLS), which significantly improves the convergence of adaptive equalizers.

The gradient algorithm has only a single adjustable parameter i.e. the step size, for controlling the convergence rate. Consequently, the slow convergence is due to this fundamental limitation [42]. In order to achieve faster convergence, the RLS algorithm uses N parameters one for each of the N eigen-values of the correlation matrix.

The least squares error based on the time average is defined as [20], [42]

$$J(n) = \sum_{i=1}^{n} \lambda^{n-i} \, e^{*}(i,n) \, e(i,n) \qquad (2.4.27)$$

Where λ is the weighting factor close to 1, but smaller than 1, $e^{*}(i,n)$ is the complex conjugate of $e(i,n)$ given by

$$e(i,n) = x(i) - y^T_N(i)w_N(n), \qquad 0 \le i \le n \tag{2.4.28}$$

and

$$y_N(i) = [y(i), y(i-1), \ldots, y(i-N+1)]^T \tag{2.4.29}$$

where $\mathbf{y}_N(i)$ is the data input vector at time i, and $\mathbf{w}_N(n)$ is the new tap gain vector at time n.

2.4.2.2 Recursive Least Squares (RLS) Algorithm

The RLS solution requires finding the tap gain vector of the equalizer $\mathbf{w}_N(n)$ such that the cumulative squared error J(n) is minimized. It uses all the previous data to test the new tap gains. The parameter λ is a data weighting factor that weighs recent data more heavily in the computations, so that J(n) tends to forget the old data in a nonstationary environment. The interested reader is referred to [20] and [42] for further detail of the algorithm.

2.5 Adaptive Lattice Algorithms

The problem of equalizing a channel whose channel correlation matrix has a large eigen-value spread is well known. Adaptive gradient algorithms are among the simplest to implement, but the rate of convergence (ROC) of these algorithms depends on the ratio of maximum to minimum eigenvalues of the channel correlation matrix [15]. It is widely recognized that when the input samples entering the equalizer are highly correlated, convergence to the optimum filter coefficients is slow when using the Least Mean Square (LMS) algorithm. Alternative algorithms, which orthogonalize the above matrix, were proposed. In particular, Godard [16] through the application of Kalman

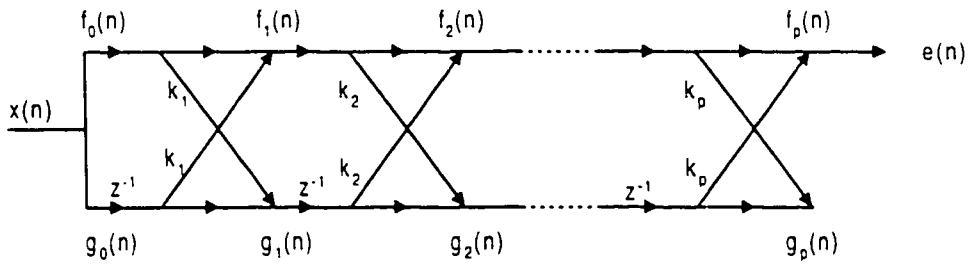


Figure 2.3: The basic two multiplier all-zero lattice

filter theory derived an adaptive self-orthogonalizing algorithm, which had extremely fast convergence properties. The Godard algorithm involves estimating the inverse of the channel correlation matrix through an iterative matrix equation. Even though the Godard's algorithm converges rapidly, the number of operations per update for this algorithm depends upon the number of equalizer taps that create implementation difficulties for large length equalizers. Gitlin and Magee [14] proposed an adaptive self-orthogonalizing algorithm, which provides a compromise between computational complexity and speed of convergence. This algorithm consisted of approximating the inverse of channel correlation matrix by a Toeplitz matrix and involved only one matrix multiplication.

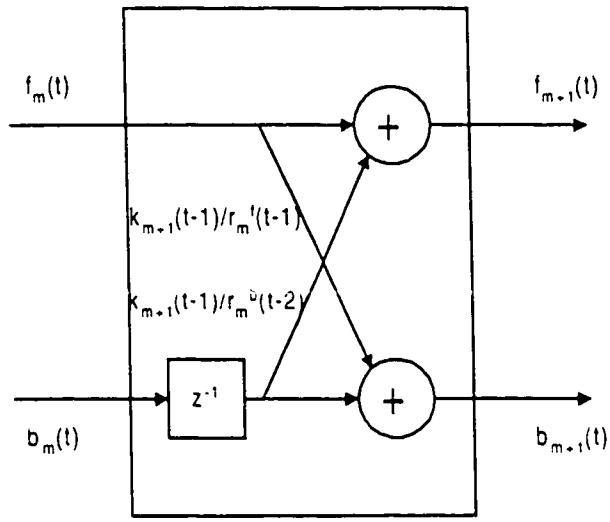
Later, a class of adaptive algorithms that provides self-orthogonalizing capabilities and requires only a number of operation per update were presented [17], [18] and [34], and [35]. These algorithms were called Adaptive Lattice (AL) Algorithms and they generated a set of orthogonal signal components, which can be used as equalizer gain

controls. The generation of these components is done through a Gram Schmidt type of orthogonalization by the use of the lattice filter module. The single all-zero realization of the lattice filter is shown in Fig. 2.3. These AL algorithms were actually proposed for use in areas such as speech processing, noise cancellation and parameter estimation. Then Makhoul in [35] realized the potential application of these algorithms to adaptive linear prediction and adaptive equalization. Satorius in [46] implemented the algorithms for adaptive equalization and his findings showed the strength of these algorithms to decrease the time of convergence or, in other words, to increase the rate of convergence of a linear equalizer. Moreover, the equalizer using AL algorithms were shown to be almost insensitive to the eigen value ratio of the channel.

Although linear lattice equalizers have been successfully simulated and their advantages are obvious, relatively little work has been done on the lattice Decision Feedback Equalizer (DFE) [48], [27], [28], [30]. The main difficulty in implementing lattice DFE is that the conventional multichannel lattice algorithms have been restricted to have the same number of stages in each channel [48]. For DFE this means that the number of feedforward and feedback stages has to be the same. This is undesirable in practice and may cause numerical instability. An attempt to use a LS lattice algorithm for decision feedback equalization was previously given in the paper by M. Shensa [48]. This algorithm holds for a DFE having the same number of feedforward and feedback stages, but it fails to be the optimum least squares decision feedback equalizer, when the number of feedforward and feedback stages is different.

Ling and Proakis [27] have derived the generalized LS multi-channel AL algorithms, implemented and investigated on both time-invariant and time variant channels. Figure

(a)



(b)

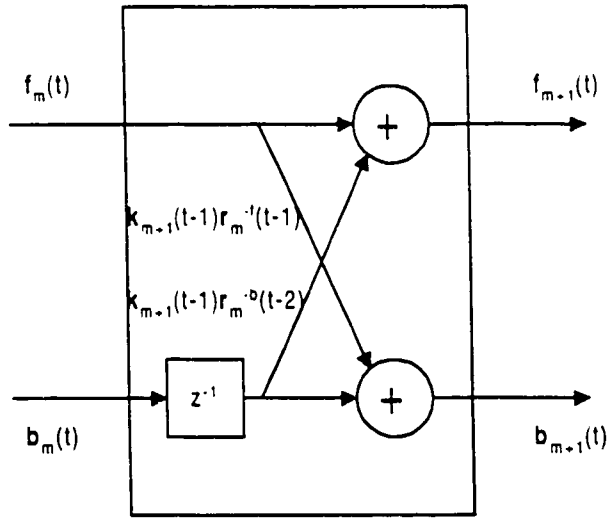


Figure 2.4: (a) Single channel lattice stage

(b) Two channel lattice stage

2.4 shows the single and two channel stages of the proposed lattice filter. Results showed that the LS AL DFE has a much better performance than the LMS DFE. These algorithms are used in this thesis work to investigate the effect of pre-orthogonalization of received symbols in the equalizer. Appendix A gives the LS lattice algorithms for the decision feedback equalizers.

The performance of these lattice algorithms on a fading multi-path HF channel was investigated by Linde in 1988 [61]. The performance of the DFE based on LS lattice was tested for digital cellular radio channels by Anand et al [1] and by Haung et al [64]. A fractionally spaced equalizer using lattice structure was developed and analyzed by Bettayeb and Zerguine in 1995 [2].

Chapter 3

ARTIFICIAL NEURAL NETWORKS

3.1 Introduction

The invention of digital computers has enabled us to do many things that were too complicated or even impossible for human beings to accomplish in a lifetime. However, conventional computers can help us only to some extent, no matter how powerful and sophisticated they are. For example, the computer can manipulate huge numbers at blinding speed, yet it cannot understand unconstrained speaker-independent noise speech. The human brain, on the other hand, can understand heavily slurred speech, yet may be incapable of calculating the square root of a prime number without the aid of pencil and paper or a machine. In other words, conventional computers are good at those problems which have well defined rules and require high numerical accuracy or manipulation of symbols, but lack the ability for the problems which have fewer obvious rules, deal with noisy data and do not require perfect numerical accuracy.

It has been a desire of scientist and engineers for decades to develop intelligent machines, which can imitate the human brain. Artificial neural networks (ANNW) is one kind of technology developed to meet this challenge. It has been observed that the difference in the information processing methods between the conventional computers and the brain are that computers are controlled by a complex control processing unit, rely on algorithm based programs that operate serially and store information at addressed locations in memory. On the other hand, brain relies on highly distributed and connected neurons cells that operate in parallel and seem to store information in variable strength connections called synapses. In a broad sense, ANNW implementations consists of three elements [50]:

- An organized topology (geometry) of interconnected processing elements, the variable strength of the connections is modeled with a set of variable weights.
- A method of encoding (learning) information.
- A method of recalling information.

Artificial Neural Network models have been studied for many years in the hope of achieving human-like performance in the field of signal processing. These models are composed of many nonlinear computational elements operating in parallel and arranged in patterns reminiscent of biological neural networks [21]. Computational elements or nodes are connected via weights that are typically adapted during use to improve performance [31]. There has been recent resurgence in the field of artificial neural networks caused by new network topologies and algorithms, analog VLSI implementation techniques, and the belief that massive parallelism is essential for high

performance signal processing. The fundamental idea of ANNW is to organize many simple identical processing elements into layers to perform more sophisticated tasks.

3.1.1 Properties of Neural Networks:

One of the most important features of the ANNW is that it is adaptive i.e. it infers solutions from the data presented to it, often capturing quite subtle relationships. It is capable of generalization and can capture the most important features of the training data. Therefore, they are, in a way, fault tolerant. Moreover, they have general nonlinear capabilities. In the real world, systems are often nonlinear. Unlike linear problems that they have many common properties, nonlinear problems are rarely generally analyzable, and in turn, it is not easy to develop a general technique for all nonlinear problems. ANNW can deal with broad nonlinear problems with little change of structure.

The other properties of ANNW include [39],[38]:

- Massive parallelism
- High computation rates
- Great capability for non-linear problems
- Continuous adaptation
- Inherent fault tolerance and
- Ease for VLSI implementation.

All these properties make neural networks attractive for various applications.

3.2 Neural Networks-based Equalization

Different artificial neural network architectures like Multi-Layer Perceptron (MLP), Radial Basis Functions (RBF), and recurrent neural nets (RNN) have been proposed in the literature for channel equalization. Among these, the most commonly and most widely used is the multilayer feedforward network (MNN) more commonly known as multilayer perceptron (MLP) and these are the NNW structures which are used throughout this thesis work.

3.2.1 Properties of MLP-based Equalizers

The reason behind this popularity of the MLP-based equalizers include its

- simplicity of computations,
- finite parameterization,
- stability and
- smaller structure size for a particular problem as compared to other structures.

This has also been shown in the literature that a three-layer MLP can generate arbitrarily complex, intricately curved nonlinear decision boundaries [21] [40].

3.2.2 Basic Building Block

The basic element of the multilayer perceptron is the neuron as shown in Fig. 3.1. Each neuron in the layer has primary local connections and is characterized by a set of real weights $[w_{1j}, w_{2j}, \ldots w_{Nj}]$ applied to the previous layer to which it is connected and a real

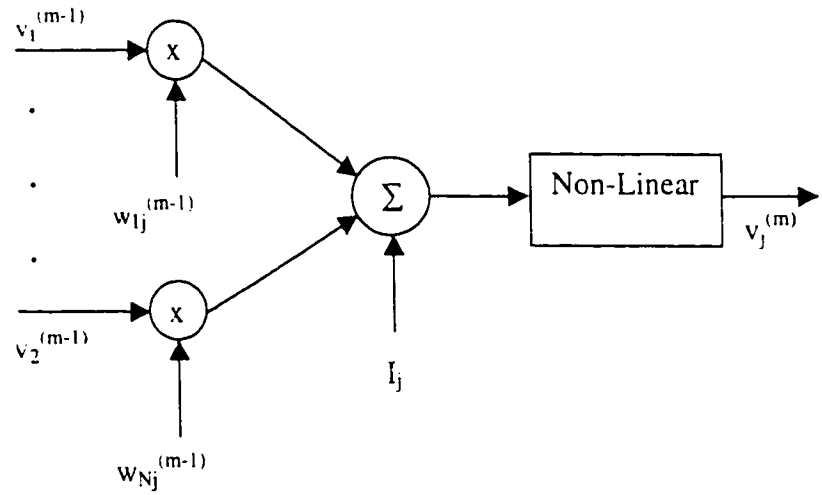


Figure 3.1: j^{th} Neuron in the mth layer

threshold level I_j. The jth neuron in the mth layer accepts inputs $v^{(m-1)}$ from the $(m-1)^{th}$ layer and gives an output $v_j^{(m)}$ given by

$$v_j^{(m)} = f_j\left(\sum_{i=1}^{N} w_{ij}^{(m)} v_i^{(m-1)} + I_j^{(m)} \right) \qquad (3.2.1)$$

The output value $v_j^{(m)}$ serves as the input to the $(m+1)^{th}$ layer to which the neuron is connected. Here, $I_j^{(m)}$ is the threshold level for the jth neuron in the mth layer of the perceptron. The non-linearity function f (.) commonly used in the perceptron is of the sigmoid type [21]:

$$f(x) = (1 - e^{-x})/(1 + e^{-x}) \qquad (3.2.2)$$

The neurons store the knowledge or information in the weights $\{w_{ij}\}$. The weights and thresholds levels are updated during training [51].

3.2.3 Multilayer Architecture

Multilayer feed-forward networks (Fig. 3.2) consists of a set of sensory units (source nodes) that constitute the input layer, one or more hidden layers of computational nodes, and an output layer of computation nodes. The input signal propagates on a layer by layer basis. A multidimensional input is fed to the nodes in the first layer (input layer). The outputs of the first-layer nodes then become the inputs to the nodes in the second layer (first hidden layer) and so on. The outputs of the network are therefore the outputs of the nodes in the final layer (output layer). Thus weighted connections exist between from a node to every node in the succeeding layer, but no connections exist between nodes in the same layer. These neural networks are commonly known as Multilayer Perceptrons. MLPs have been applied successfully to solve some difficult and diverse problems by training them in a supervised manner with a highly popular algorithm known as error back propagation algorithm which is based on error correction learning rule [21].

3.2.4 Back Propagation Algorithm:

Although, they were shown to be powerful in function mapping and pattern classification, the MLPs did not receive much attention in applications until the reintroduction of the back propagation (BP) algorithm by Rumelhart *et al* [45] which

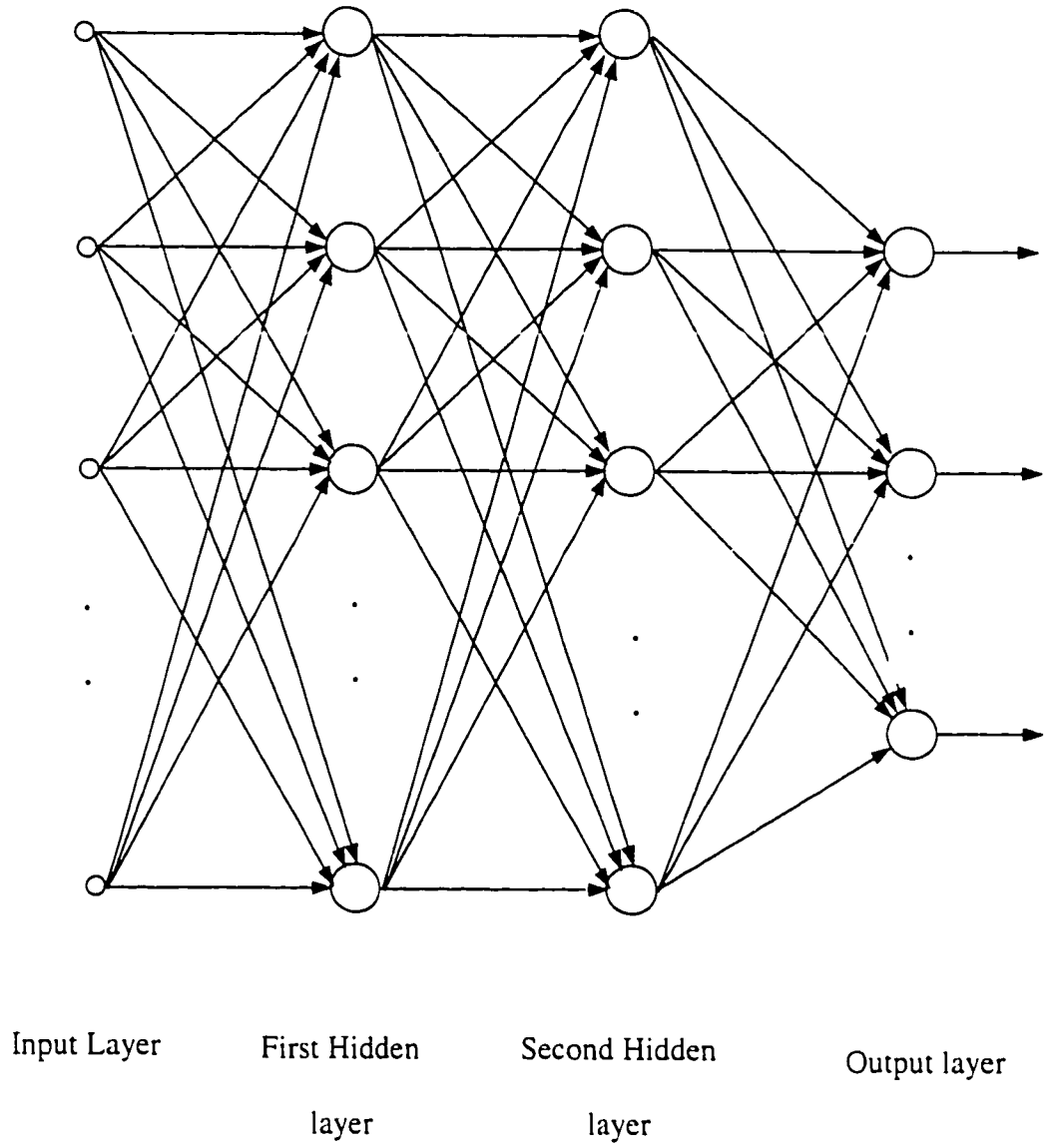


Figure 3.2: Architectural graph of a multilayer perceptron with two hidden layers

was first introduced by Werbos early in 1974 [55]. The BP algorithm is capable of training the MLP adaptively by back propagating the error information. The reason that the back propagation algorithm has become so popular and is still dominantly used in applications is that it is computationally straight forward and is simple to program. On the other hand, since it is no more than a generalized LMS algorithm, it suffers from the same problems as the linear LMS algorithm and, in particular, exhibits relatively slow convergence. Also, taking into account the complex nonlinear relationship between the inputs and the outputs of the MLP, the application of the BP algorithm can lead to a local minimum and never reach the global optimal point.

The error back propagation process consists of two passes through the different layers of the network: a forward pass and a backward pass. In the forward pass, an activity pattern (an input vector) is applied to the sensory nodes of the network, and its effect propagates through the network, layer by layer. Finally, a set of outputs is produced as the actual response of the network. During the forward pass, the synaptic weights of the network are all fixed.

During the backward pass, the synaptic weights are all adjusted in accordance with the error-correction rule. Specifically, the actual response of the network is subtracted from a desired response to produce an error signal. This error signal is then propagated backward through the network, against the direction of the synaptic

connections, Hence the name "error back propagation". The synaptic weights are adjusted so as to make the actual response of the network move closer to the desired response.

3.2.5 Pictorial Representation of Back Propagation Algorithm

Figure 3.2 presents the architectural layout of a multilayer perceptron. The corresponding architecture for back propagation learning, incorporating both the forward and backward phases of the computations involved in the learning process is presented in Fig. 3.3. The multilayer network shown in the top part of this figure accounts for the forward phase.

The notations used in this part of the figure are as follows:

$w^{(l)}$ = Synaptic weight vector of a neuron in layer l

$I^{(l)}$ = Threshold of a neuron in layer l

$u^{(l)}$ = Vector of net internal activity levels of neurons in layer l

$v^{(l)}$ = Vector of function signal of neuron in layer l

f (.) = The nonlinear function defined in equation (3.2.2)

The layer index l extends from the input layer (l = 0) to the output layer (l = L).; in Fig. 3.3, we have L = 3; we refer to L as the depth of the network.

The lower part of the network accounts for the backward phase, which is referred to as sensitivity network for computing the local gradients in the back-propagation algorithm.

The notations used in this part of the figure are:

$\delta^{(l)}$ = vector of local gradients of neurons in layer l

e = error vector represented by $e_1, e_2, \ldots, e_q$ as elements

3.3 Perceptron-based DFE

A three-layer perceptron-based decision feedback equalizer structure, as shown in Fig. 3.4, consists of a feedforward filter and a feed back filter. The input to the feedforward

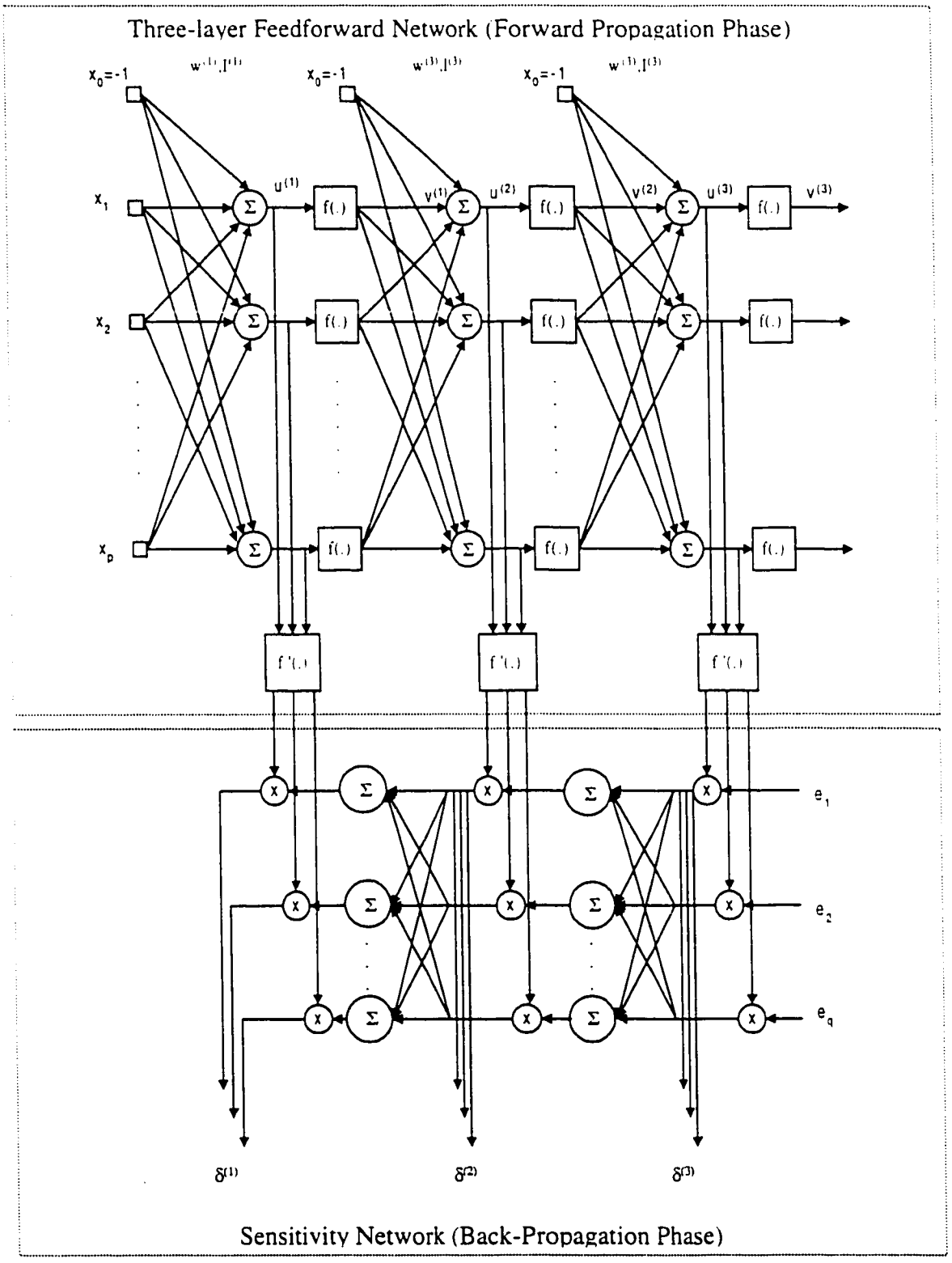


Figure 3.3: Architectural graph of three-layer feedforward network and associated sensitivity network (back-propagating error signals)

filter is the sequence of noisy received signal samples $\{y_n\}$. The input to the feedback filter is the output symbol decision sequence from a nonlinear symbol detector $\{\tilde{x}_{n-d}\}$. Most applications of MLP's use mappings that have binary or bipolar input and output vectors (i.e. each vector component is essentially either 1 or 0 / -1), which are sufficient for pattern classification use. Bunpei *et al* [2] have proven that the three-layered perceptron with an infinite number of computing units can represent an arbitrary function, if its connections are appropriately fixed.

Thus, the three-layer perceptron (two hidden layers and one output layer) is sufficient for the nonlinear DFE structure, because a three-layer perceptron can generate arbitrary complex, nonlinear decision regions [54]

At time n, the input received signal vector and the decision signal vector are given respectively by

$$Y(n)^T = [y_n, y_{n-1}, \ldots\ldots, y_{n-N+1}] \qquad (3.3.1)$$

$$X(n)^T = [\tilde{x}_{n-d-1}, \tilde{x}_{n-d-2}, \ldots, \tilde{x}_{n-d-l}] \qquad (3.3.2)$$

These are in the feed forward filter and feedback filter of the decision feedback equalizer, respectively, where, d is the delay parameter. The decision $\tilde{x}_{n-d}$ is formed by quantizing the estimate $\hat{x}_{n-d}$ in the output layer to the nearest information symbol.

The signals at the input layer of the decision feedback equalizer can be represented by a (N+l) x 1 vector as

$$V^{(0)} = [y_n, y_{n-1}, \ldots\ldots, y_{n-N+1}; \tilde{x}_{n-d-1}, \tilde{x}_{n-d-2}, \ldots, \tilde{x}_{n-d-l}]^T \qquad (3.3.3)$$

The N_1 x 1 vector in the output of the hidden layer 1 is

$$V^{(1)} = [v_1^{(1)}, v_2^{(1)}, \ldots, v_j^{(1)}, \ldots\ldots, v_{N1}^{(1)}]^T \qquad (3.3.4)$$

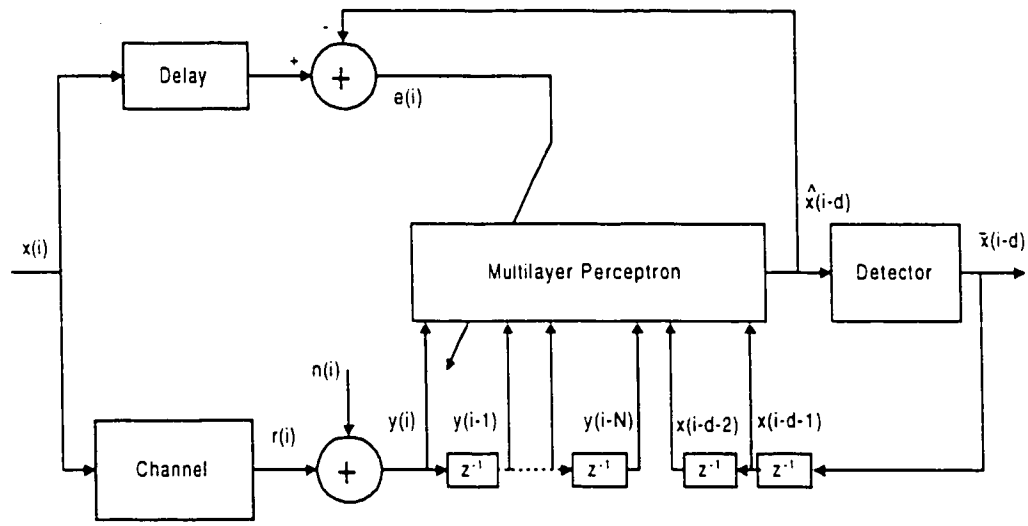


Figure 3.4: Block diagram of MLP-based decision feedback equalizer

where

$$v_j^{(1)} = f_j\left(\sum_{i=0}^{N-1} w_{ij}^{(1)} y_{n-i} + \sum_{p=1}^{l} w_{pj}^{b(1)} \tilde{x}_{n-d-p} + I_j^{(1)}\right) \quad j = 1,2,\ldots,N1 \qquad (3.3.5)$$

where b denotes the feedback tap weight. The N_2 x 1 vector in the output of the hidden layer 2 is

$$V^{(2)} = [v_1^{(2)}, v_2^{(2)}, \ldots, v_j^{(2)}, \ldots\ldots, v_{N1}^{(2)}]^T \qquad (3.3.6)$$

where

$$v_k^{(2)} = f_k\left(\sum_{j=1}^{N1} w_{jk}^{(2)} v_j^{(1)} + I_k^{(2)}\right) \quad k = 1,2,\ldots,N_2 \qquad (3.3.7)$$

The final output is

$$v_0^{(3)} = \tilde{x}_{n-d} = f_0\left(\sum_{k=1}^{N2} w_{k0}^{(3)} v_k^{(2)} + I_0^{(3)}\right) \qquad (3.3.8)$$

where $\tilde{x}_{n-d}$ is the estimated symbol at time n. Substituting equations (3.3.5) and (3.3.7) into eq. (3.3.8), yields

$$\hat{u}_{n-d} = f_0\left(\sum_{k=1}^{N2} w_{k0}^{(3)} f_k\left(\sum_{j=1}^{N1} w_{jk}^{(2)} f_j\left(\sum_{i=0}^{N-1} w_{ij}^{(1)} y_{n-i} + \sum_{p=1}^{l} w_{pj}^{b(1)} \tilde{x}_{n-d-p} + I_j^{(1)}\right) + I_k^{(2)}\right) + I_0^{(3)}\right) \tag{3.3.9}$$

The w's (weights) and I's (threshold levels) in eqn. (3.3.9) are values specified by the training algorithm so that after training is finished, the equalizer will self adapt to the changes in the channel characteristics occurring during transmission (decision direct mode). The nonlinear detector can be modeled by a threshold function g(x) and is defined as

$$g(\hat{x}_{n-d}) = \tilde{x}_{n-d} = \begin{cases} 1 \text{ if } \hat{x}_{n-d} \geq 1 \\ -1 \text{ otherwise} \end{cases} \tag{3.3.10}$$

3.3.1 Eliminating Intersymbol Interference: Decision Feedback Signal

The output $v_j^{(1)}$ of the jth neuron in layer one can be expressed in terms of $\{g_p\}$, the feedback tap weights $\{w_{pj}^{b(1)}\}$ and the transmitted signal $\{x_n\}$; $x_n \in (1,-1)$. Note that g_p is the convolution of the channel impulse response $\{h_p\}$ and the weights $\{w_{ij}^{(1)}\}$. Thus

$$v_j^{(1)} = f_j\left(\sum_p x_{n-p}\, g_p + \sum_{p=1}^{l} w_{pj}^{b(1)}\, \tilde{x}_{n-p} + \eta_n + I_j\right) \tag{3.3.11}$$

where η_n is the additive noise.

The above equation can be written as

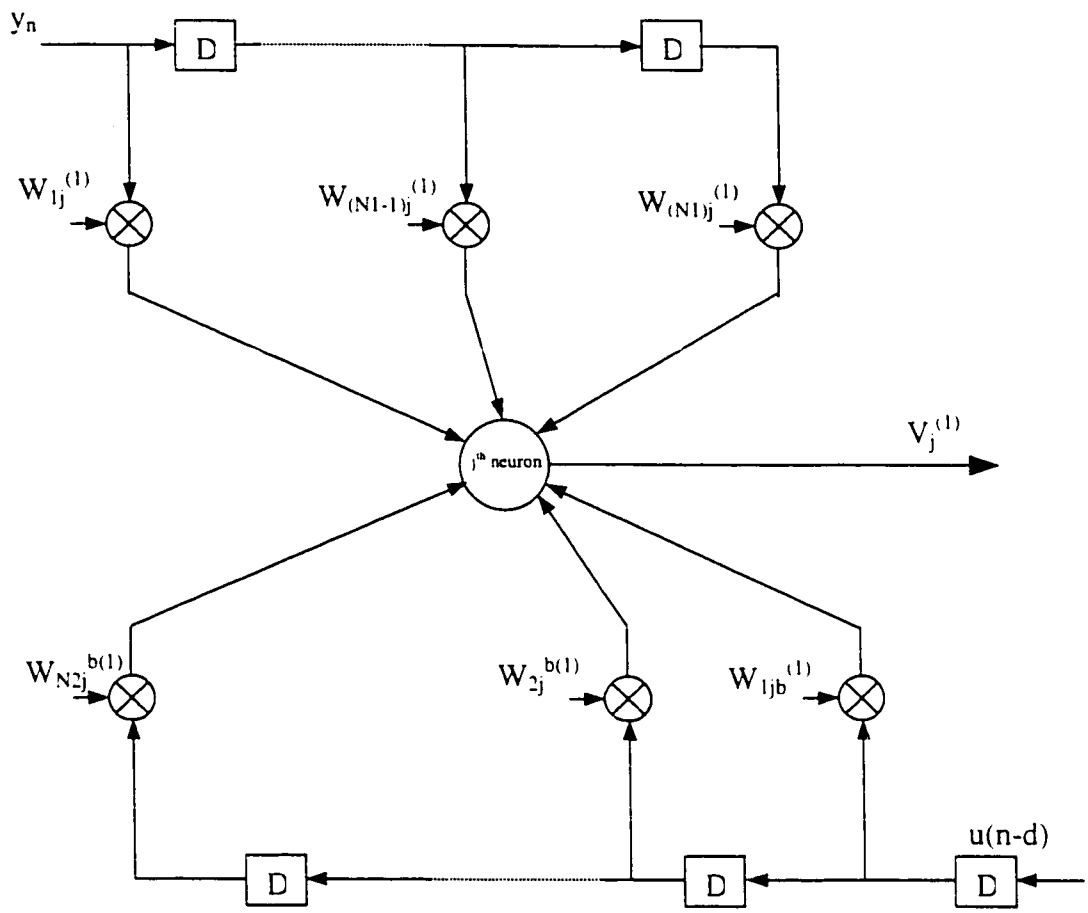


Figure 3.5: j^{th} neuron with feedforward and feedback signals in 1st hidden layer

$$v_j^{(1)} = f_j\Big(x_n g_0 + \sum_{p=1}^{l} (g_p + w_{pj}^{b(1)})\,\tilde{x}_{n-p} + \sum_{p=1}^{l} (x_{n-p} - \tilde{x}_{n-p})\,g_p + \sum_{\substack{p<0 \\ p>l}} x_{n-p}\, g_p + \eta_n + I_j \Big) \tag{3.3.12}$$

If we select

$$g_p = -\,w_{pj}^{b(1)}, \qquad p = 1,2,\ldots,l \tag{3.3.13}$$

and the probability of error is very small, we may assume that the last symbols have been received correctly, i.e.

$$\tilde{x}_{n-p} = x_{n-p}, \qquad p = 1,2, \ldots, l \tag{3.3.14}$$

then the equation can be simplified as

$$v_j^{(l)} = f_j(x_n g_0 + \sum_{\substack{p<0 \\ p>l}} x_{n-p} g_p + \eta_n + I_j) \tag{3.3.15}$$

All ISI from past symbols ($1 \leq p \leq l$) is eliminated without altering the useful signal term $u_n g_0$ or enhancing the noise component η_n. The term $\sum_{\substack{p<0 \\ p>l}} x_{n-p}\, g_p$ in equation (3.3.15) represents the residual ISI and will be reduced as the signal is passed forward. If an incorrect decision is made by the detector, e.g. $\tilde{x}_{n-p} = -x_{n-p}$, the decision errors tend to propagate because they result in residual intersymbol interference and a reduced margin against noise at future decisions.

3.3.2 The Learning Phase

The increments used in updating the weights, Δw_{ij} and threshold levels ΔI_j of the mth layer can be accomplished by the following rules.

$$\Delta w_{ij}^{(m)}(n+1) = \eta \delta_j^{(m)} v_j^{(m-1)} + \alpha \Delta w_{ij}^{(m)}(n) \tag{3.3.16}$$

and

$$\Delta I_j^{(m)}(n+1) = \beta \delta_j^{(m)} \tag{3.3.17}$$

where η is the learning gain, α is the momentum parameter, β is the threshold level adaptation gain, and layer $m \in [1,2,\ldots,M]$

$$\delta_j^{(M)} = (x_j - v_j^{(M)}).(1-v_j^{2(M)}) / 2 \tag{3.3.18}$$

and recursively back propagating the error signal to lower layers

$$\delta_j^{(m)} = (1 - v_j^{2(m)}) \sum_l \delta_l^{(m+1)} w_{jl}^{(m+1)} / 2 \qquad (3.3.19)$$

where m ∈ [1,2,..., M-1], l is the overall neurons in the layer above the neuron j and x_j is the desired output.

To allow rapid learning, a momentum term, $\Delta w_{ij}^{(m)}(n)$, scaled by α is used to filter out high frequency variation of the weight vector. As a result the convergence rate is much faster and the weight changes are smoothed.

3.3.3 A Short Glimpse on the Past Work on ANNW's as a tool for Channel Equalization

The first MLP applications to the channel equalization were reported around 1990, when Gibson, Siu and Cowan *et al* [13] Siu *et al* [51] effectively utilized MLP neural networks as adaptive equalizers for several linear and nonlinear channel models. They demonstrated that neural based equalizer trained by back propagation algorithms showed superior performance over conventional decision feedback equalizer because of its capability to form complex decision regions with nonlinear boundaries. Also, computer simulations showed that the MLP equalizer works better than the linear equalizer for nonlinear channel distortions. Soon after that, Chen et al also analyzed the MLP structures performance for the equalization of nonlinear channels.

In 1991, Gibson [12] et al and Colin et al [9] addressed the application of nonlinear structures (like multi-layer perceptron) to the problem of equalization and achieved very encouraging results. The application of MLP structures for adaptive channel

equalization of linear and nonlinear channels using PAM and QAM signals was first investigated by Peng et al [39], [38], [40]. Al-Mashouq with Reed used multilayer structures for equalization combined with the decoding for severe intersymbol interference channels in 1994 [25].

Recently, Rong and Wang in [6] have used multilayer NNWs for adaptive decision feedback equalization of digital satellite channels using M-ary PSK. Results showed to offer a superior performance as a channel equalizer to the LMS based DFE.

Other structures of Neural Networks like radial basis functions and Recurrent Neural Networks have also been used by many researchers as effective channel equalizers. The application of radial basis functions networks to digital communication channel equalization was first investigated by Sheng Chen *et al* [8]. Later on, radial basis functions were used for identification of nonlinear systems [59], [60]. Recently, P. C. Kumar has used minimal radial basis function neural networks for nonlinear channel equalization [26]. The use of Recurrent Neural Network (RNN) was first proposed by G. Kechriotis *et al* [24]. RNNs have feedback property, which makes them attractive for the equalization of nonlinear channels with deep spectral nulls. Recently, R. Parisi *et al* in [37] and S. Ong *et al* in [36] have investigated the application of recurrent NNWs as a decision feedback channel equalizers demonstrating the faster rate of convergence and small size of such equalizers.

Chapter 4

SYSTEM PERFORMANCE IN STATIC CHANNELS

4.1 Performance Criterion

There are various ways to analyze the performance of a system. Specifically for the equalizer, the main parameters are convergence rate, steady-state mean squared error (SSMSE), bit error rate (BER), and opening of the eye in the eye diagram. These parameters are evaluated for different systems and channels. A brief description and introduction to different analyses performed is given bellow:

4.1.1 Visual Indicators of Performance

In digital communications in general, and in equalizer's performance analysis in particular, the eye diagrams and scatter (constellation) diagrams are widely used as qualitative (visual) indicators of the health of the system. Although, these are not performance "estimators" in the same sense as the learning characteristics and Bit Error

Rate (BER) curves, they are performance "indicators" of the equalizer's performance [23].

4.1.1.1 Eye Diagrams

An eye diagram capsules the degree of distortion due to the inter-symbol interference (ISI) by displaying the detected waveform in a convenient way. The effect of the ISI is to cause the eye to close, thereby reducing the margins for additive noise to cause errors. The eye diagram can be obtained at various points in a system to obtain a sense of the degree of distortion at different stages of the system. The quantity of interest, the "eye opening", is defined as the distance from the decision threshold to the closest trace at the sampling instant. The wider the eye, the better is the performance.

4.1.1.2 Scatter Diagrams

The term "scatter diagram" is often used to describe a plot of the values of a set of samples of a random variable against a common independent variable. In the absence of ISI and noise for BPSK, the points in the scatter diagram would result in two distinct points in the signal space corresponding to +1 and -1. Intersymbol interference and noise result in a deviation of the received samples from the desired signal [42]. Originally, each symbol is an ideal point, but as the effects of noise and distortion come into the scene, that point is represented by a small area around that point containing all the sampled values. The larger the intersymbol interference and noise, the larger is the scattering of the received signal samples relative to the transmitted signal points.

4.1.2 Learning Curves

The learning curves for the equalizer give us the quantitative measure of the convergence rate (or convergence time) and the steady-state mean squared error (SSMSE). The time, in terms of the symbol times, an equalizer takes to converge to its steady-state error is referred to as the convergence time of the equalizer. A fast rate of convergence allows the algorithm to adapt rapidly to a stationary or time-variant environment of unknown characteristics. This parameter is of utmost importance for the performance analysis of an equalizer and this is one of the main parameters on which the use of a particular equalizer is decided for some particular system condition.

4.1.3 Bit error rate (BER) performance

For a digital communication system, the relevant measure of performance is always one related to the system's error-producing behavior. That behavior may be characterized in different ways. The fractional number of errors in a transmitted sequence is certainly informative of the basic reliability of a system. Generally, this fractional number of errors is referred to as error probability, bit error ratio, bit error probability, bit error rate. etc. For brevity, the abbreviation used for all these terms is Bit Error Rate (BER). Many simulation-based approaches [23] are used for estimating the BER but the most commonly used approach is the Monte Carlo simulation.

4.2 Performance Analysis of the Proposed Scheme

Three time-invariant channel models were used in the simulation. The following channel models are taken from different papers and books. Channel 1 is used in [51] and [12] for the performance analysis of the MLP-based equalizers, channel 2 is used by Ling and Proakis [28], [30], and channel 3 is used in [42]. The discrete time channel models are described by the following transfer functions:

$H1(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$

$H2(z) = 0.408 + 0.816z^{-1} + 0.408z^{-2}$

$H3(z) = 0.227 + 0.460z^{-1} + 0.688z^{-2} + 0.460z^{-3} + 0.227z^{-4}$

The impulse response and amplitude spectrums for these channels are shown in Fig. 4.1. These channels are normalized so that the power in the channel (squared sum of the finger heights) is equal to unity. The ratio of maximum to minimum squared values of the channel frequency response is equal to 25 for channel 1 and 81 for channel 2. Since channel 3 has deep nulls in its spectrum, this ratio may be considered to be infinite. The impulse response and the spectral characteristics of channel 3 clearly show that it has the worst response.

The digital message applied to the channel is uniformly distributed bipolar random numbers (-1,1). The channel noise is taken to be Additive White Gaussian Noise (AWGN) with an SNR of 20 dB, 15 dB and 10 dB. In the rest of the thesis, the term SNR is used as a substitute for the average bit energy to noise power ratio i.e. (Eb / No).

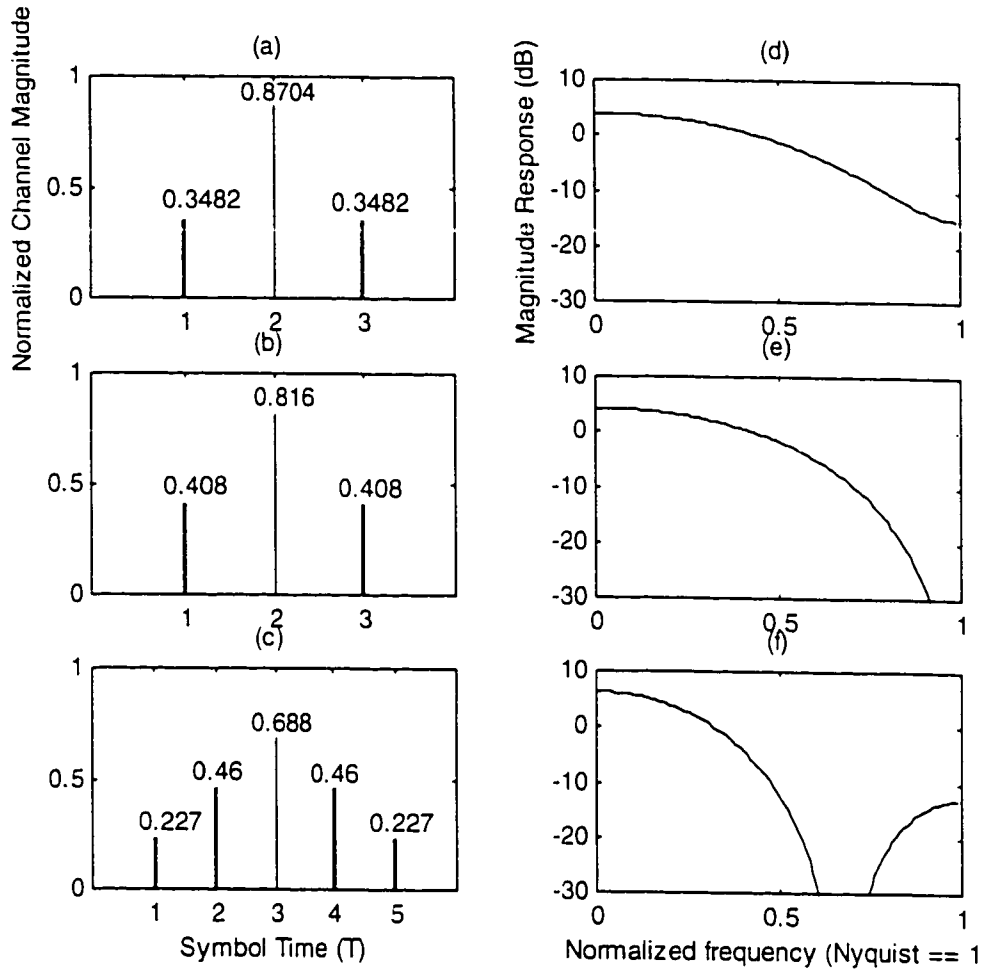


Figure 4.1: (a), (b), (c) Impulse response of channel 1, channel 2, channel 3

(d), (e), (f) Frequency response of channel 1, channel 2, channel 3

The DFE's used have either four feedforward stages and one feedback stage, denoted as (4,1) DFE, or have nine feedforward and two feedback stages, denoted as (9,2) DFE. The (9,2) configuration is chosen for the channels that are not equalizable well by (4,1) configuration. The number of input points for the MLP is 5 or 11. These are represented as follows:

(5,0) = 5 feedforward taps and no feedback stage (Linear equalizer with 5 taps)

(11,0) = 11 feedforward taps and no feedback stage (Linear equalizer with 11 taps)

(4,1) = 4 feedforward taps and 1 feedback stage (DFE with 5 taps)

(9,2) = 9 feedforward taps and 2 feedback stage (DFE with 11 taps)

MLP has one input layer, two hidden layers and one output layer. The number of neurons in different layers are given as:

(9,3,1) → 9 neurons in hidden layer 1, 3 neurons in hidden layer 2, 1 neuron in output layer,

(3,2,1) → 3 neurons in hidden layer 1, 3 neurons in hidden layer 2, 1 neuron in output layer.

The learning gain parameter (η), momentum parameter (α), and threshold level adaptation gain (β) are 0.07, 0.3 and 0.05 respectively. The weighting factor w for lattice algorithm is 0.99 and the step size for the gradient transversal equalizer is the one that gives the fastest convergence rate.

4.2.1 Scatter Diagrams

First the effect of the channels on the data in the form of scatter (constellation) diagram is analyzed. The scatter diagram is plotted in terms of the present channel-output

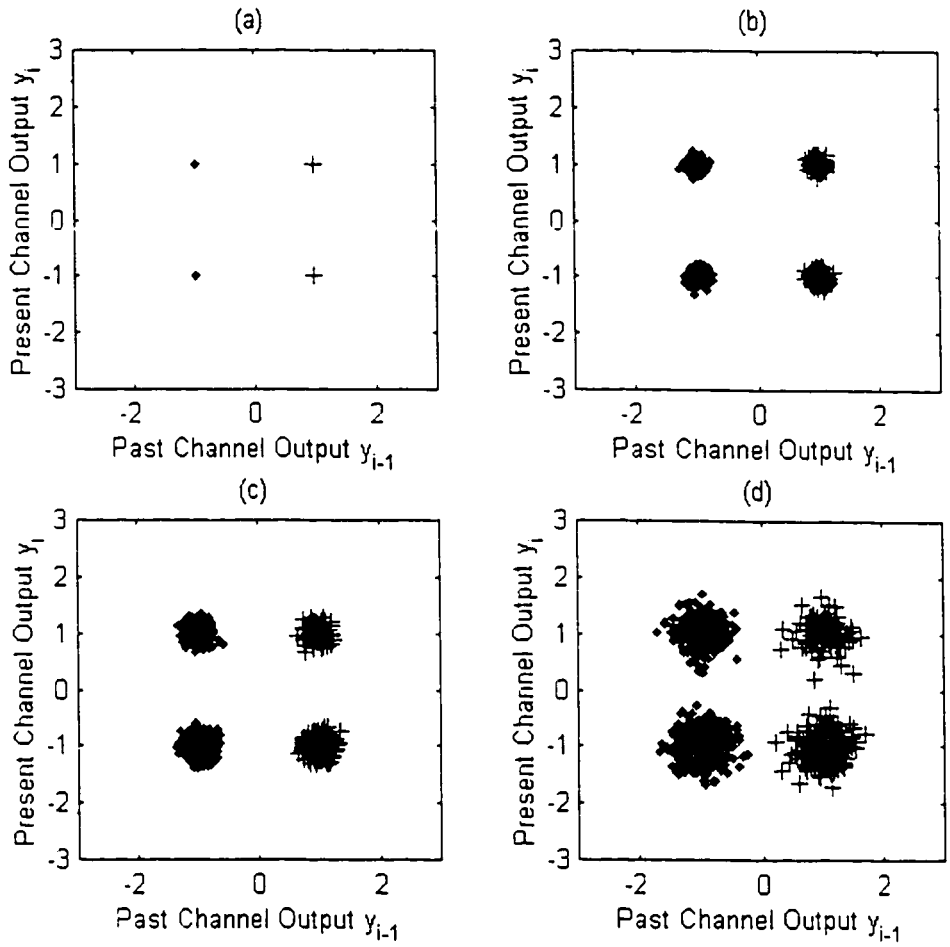


Figure 4.2: Scatter diagrams for Zero ISI case in AWGN environment.

(a) SNR = Infinity,

(b) SNR = 20 dB,

(c) SNR = 15 dB,

(d) SNR = 10dB

(+) REPRESENTS +1 (.) REPRESENTS -1

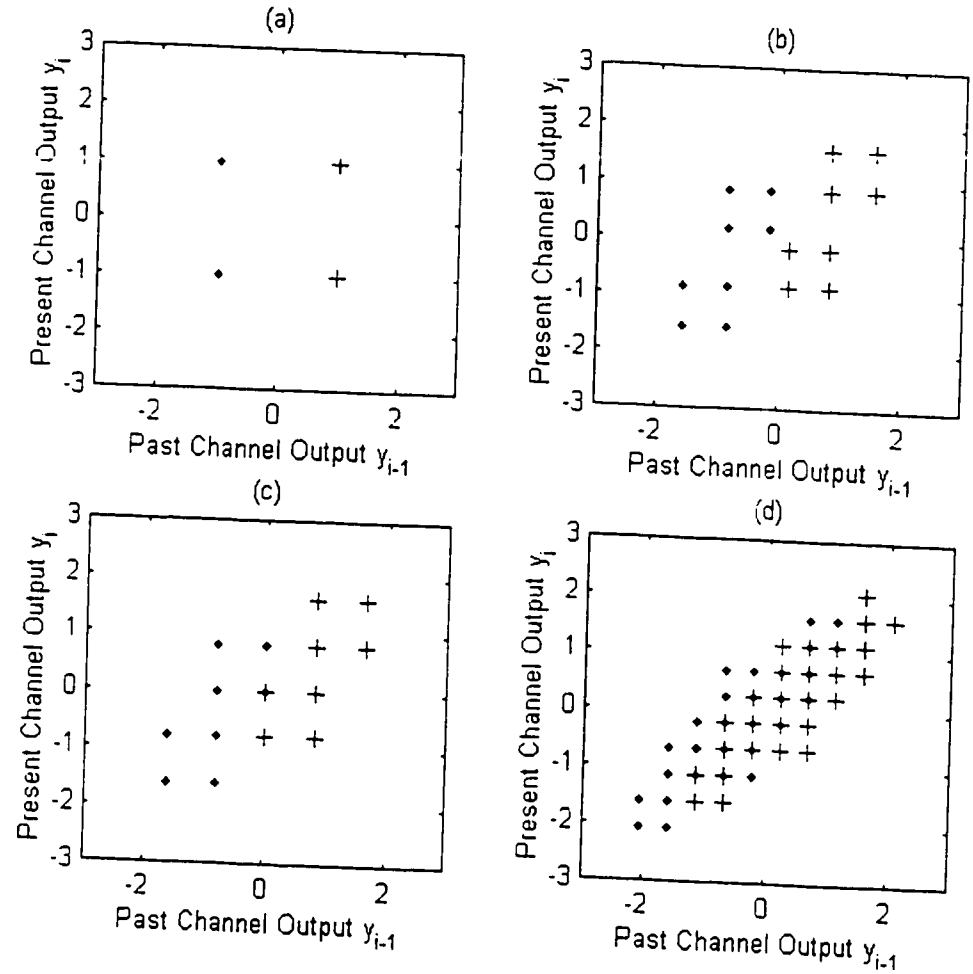


Figure 4.3: Scatter diagrams for Zero noise case in AWGN environment

(a) Zero ISI,

(b) ISI by channel 1,

(c) ISI by channel 2,

(d) ISI by channel 3

(+) Represents +1 (.) Represents -1

symbol as a function of the past channel-output symbol. Fig 4.2 gives the scatter diagram for the case of no ISI. Part (b), (c), and (d) show the case for SNR values of 20 dB, 15 dB, and 10 dB, respectively. The symbols [+1,-1] are clearly seen to be distinguishable and the decision boundary is easy to be formed due to the linear nature of the boundary. In this situation each symbol is represented by a single point in the signal space [Fig 4.2 (a)]. But when the effect of noise comes into action, each symbol is dispersed in the signal space and is represented by the cloud of points.

Fig. 4.3 gives the scatter diagrams for the case of ISI introduced by the three channels. Fig. 4.3 (a) is the ideal case with no ISI and no noise. Fig. 4.3 (b), (c), and (d) show the constellation for channels 1, 2 and 3, respectively, with SNR = Infinity.

As apparent from these figures, each point in the signal space is now represented by more than one point. For channel 1, the symbols are still separable by a nonlinear decision boundary, but for channels 2 and 3, several symbols overlap in the constellation and it is comparatively difficult to draw even nonlinear decision boundaries.

Fig. 4.4, 4.5 and 4.6 shows the scatter diagram for the equalizer output after 100, 1000, and 10000 iterations for channel 1. For these three figures, part (a) gives the unequalized data, parts (b), (c), and (d) show the data equalized by the LMS-DFE, MLP-DFE and MLP-DFE with lattice filter, respectively. All the data is taken at the SNR value of 10 dB. All the DFE's are (4,1) and the MLP is (9,3,1).

As seen, the DFE tries to remove the ISI from the received data, but the noise content is still high causing symbol errors. The MLP-DFE not only removes the ISI, but also reduces the area of the noise cloud by restricting the equalized symbols within a small

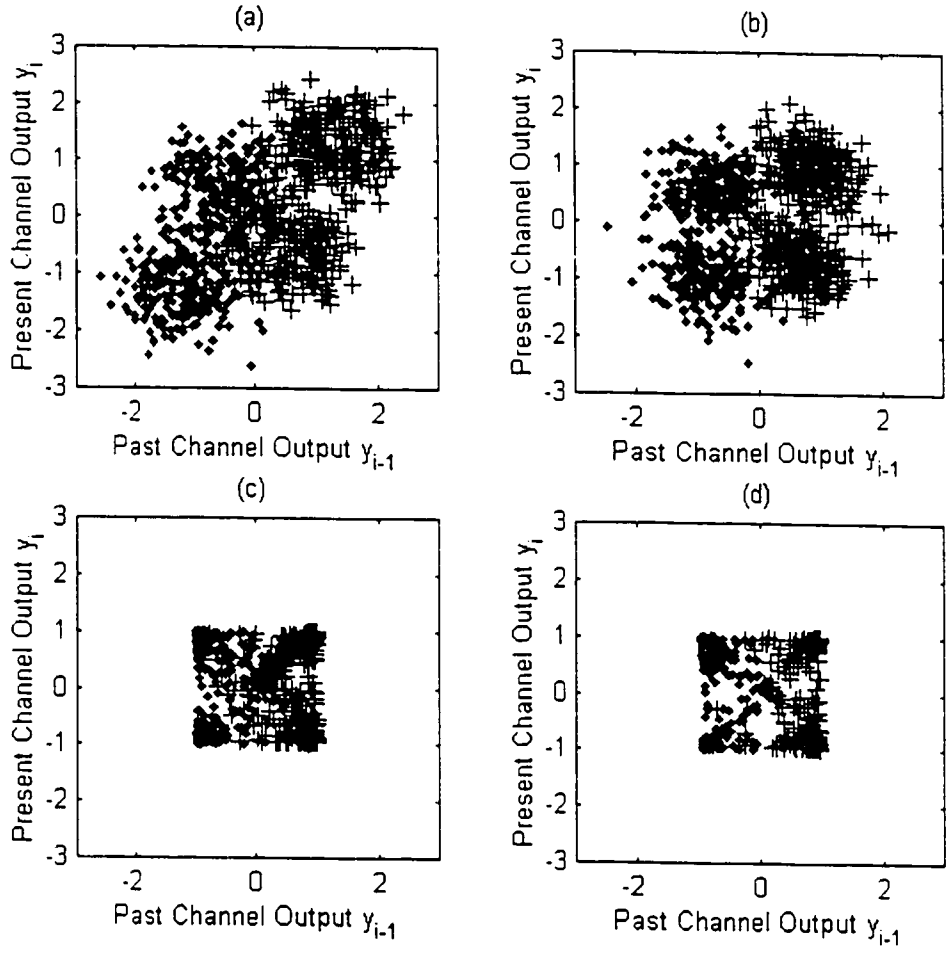


Figure 4.4: Scatter diagrams for channel 1 at SNR=10 dB in AWGN environment, after 100 iterations

(a) Un-equalized data,

(b) Equalized data by (4,1)DFE

(c) Equalized data by (4,1)MLP-DFE

(d) Equalized data by Lattice based (4,1)MLP-DFE

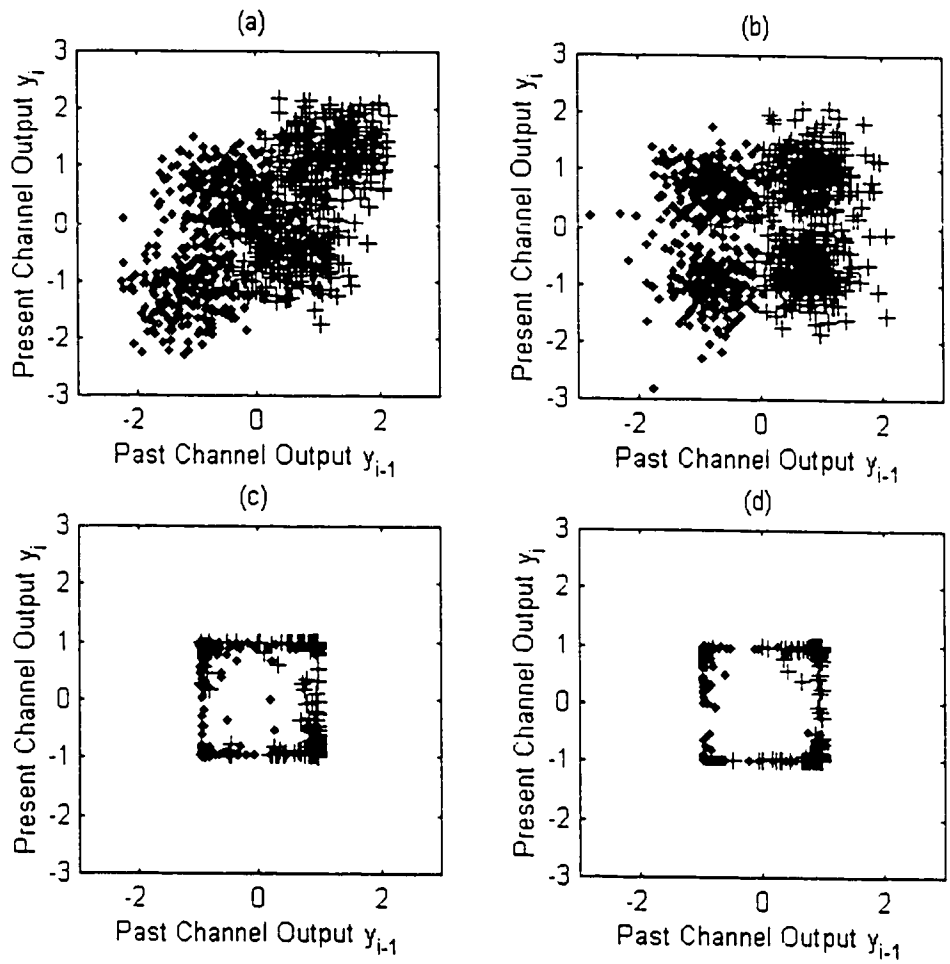


Figure 4.5: Scatter diagrams for channel 1 at SNR=10 dB in AWGN environment, after 1000 iterations

(a) Un-equalized data,

(b) Equalized data by (4,1)DFE

(c) Equalized data by (4,1)MLP-DFE

(d) Equalized data by Lattice based (4,1)MLP-DFE

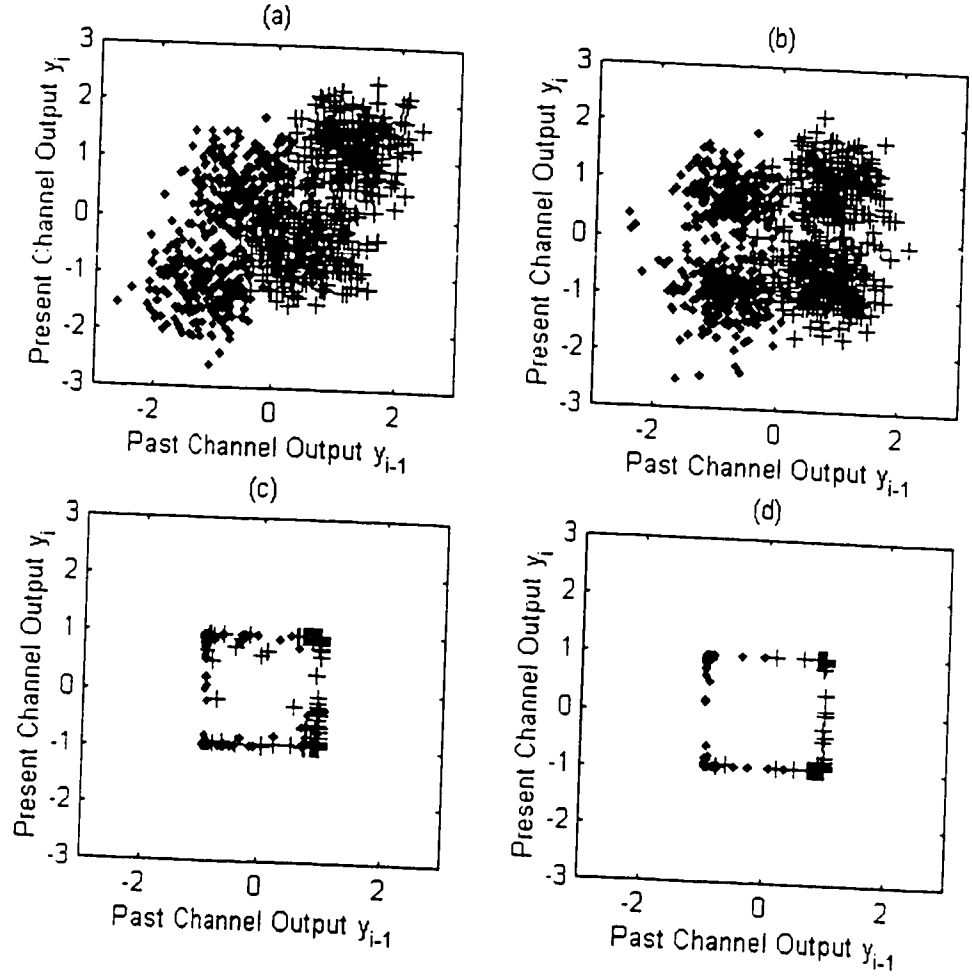


Figure 4.6: Scatter diagrams for channel 1 at SNR=10 dB in AWGN environment, after 10000 iterations

(a) Un-equalized data,

(b) Equalized data by (4,1)DFE

(c) Equalized data by (4,1)MLP-DFE

(d) Equalized data by Lattice based (4,1)MLP-DFE

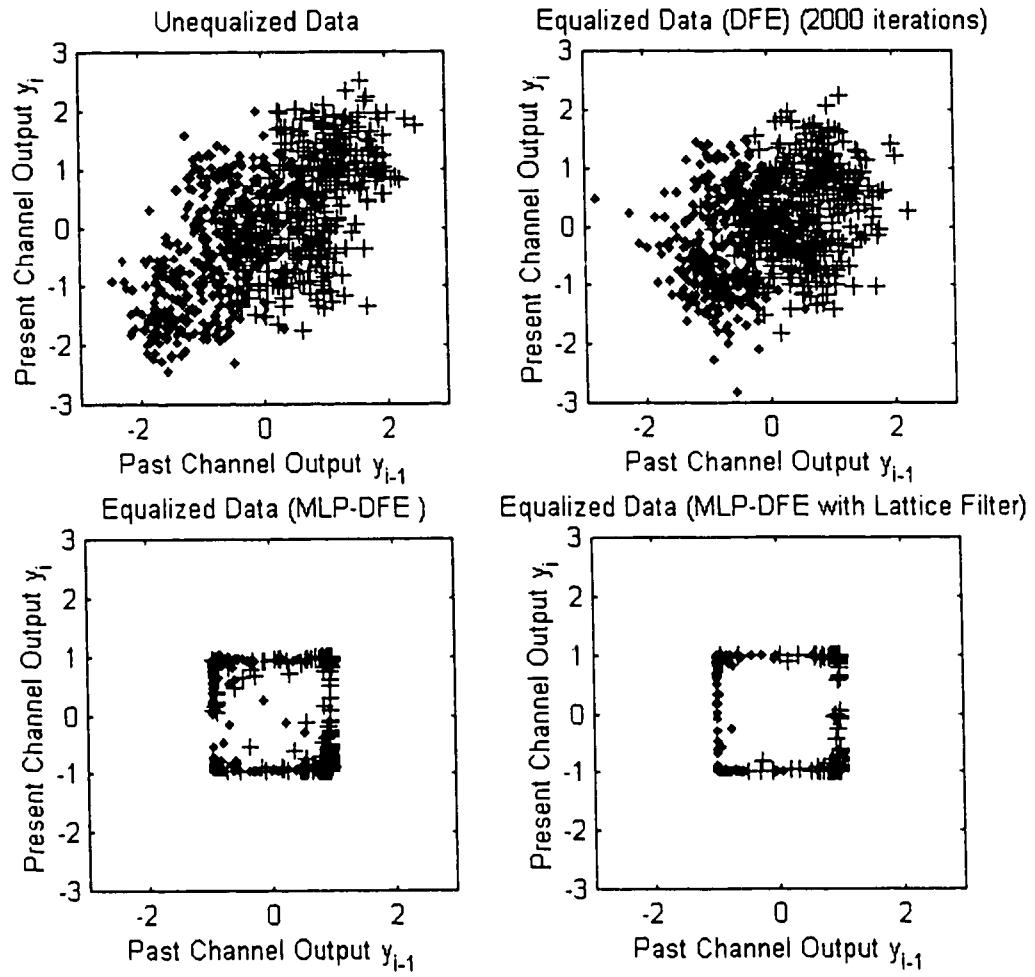


Figure 4.7: Scatter diagrams for channel 2 at SNR=10 dB, in AWGN environment, after 10000 iterations

(a) Un-equalized data,

(b) Equalized data by (4,1)DFE

(c) Equalized data by (4,1)MLP-DFE

(d) Equalized data by Lattice based (4,1)MLP-DFE

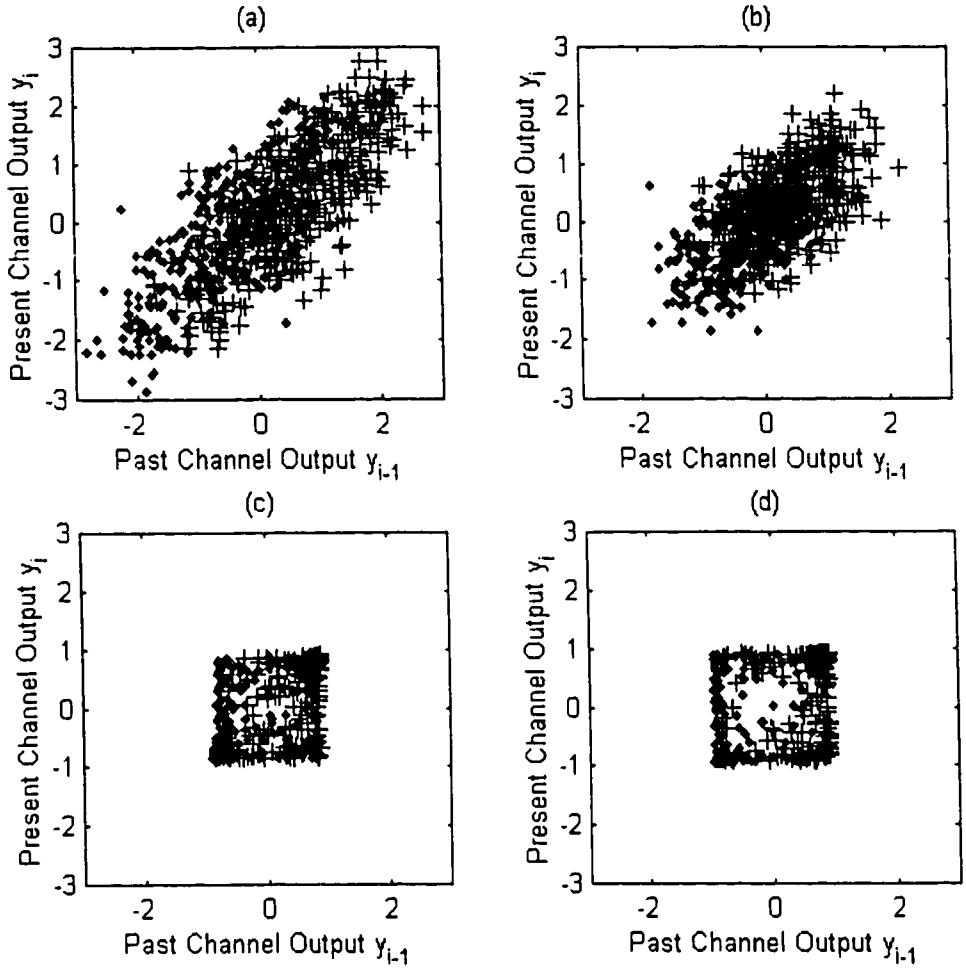


Figure 4.8: Scatter diagrams for channel 3 at SNR=10 dB in AWGN environment, after 10000 iterations

(a) Un-equalized data,

(b) Equalized data by (4,1)DFE,

(c) Equalized data by (4,1)MLP-DFE,

(d) Equalized data by Lattice based (4,1)MLP-DFE

area. All the symbols lie within the square of length two. This is due to the nature of the nonlinearity function used in the neural network. When the preorthogonalization of the input data is employed by the lattice filter, the situation becomes even better and the symbols are seen to be more converged to their original positions.

By increasing the number of iterations to 1000 and 10000, the performance of the DFE remains almost the same, while the performance of the MLP-DFE and Lattice-based MLP-DFE improve.

Figures 4.7 and 4.8 give the same figures for channel 2 and 3 after 10000 iterations. Due the severity of the ISI for these channels, the DFE is incapable of performing well because the symbols now can not be separated by any linear hyper-plane. This is where the strength of neural equalizer comes into action when it creates an intricately curved nonlinear decision boundary to separate the symbols. But since for channel 3, the overlapping of the symbols makes it difficult even for the MLP-equalizer to perform well. Decorrelating the input data improves the performance of the MLP-DFE .

4.2.2 Eye Diagrams

Figures 4.9, 4.10, and 4.11 give the eye diagrams for channel 1, 2 and 3, respectively. Again the DFE used are (4,1) and MLP are (9,3,1), and the data are plotted after 2000 iteration at the SNR of 15 dB. In these figures, the part (a) is the corresponding channel output or equivalently the equalizer input data. Parts (b), (c) and (d) represent the eye diagrams of the equalized data by LMS-DFE, MLP-DFE, and lattice MLP-DFE, respectively.

As can be seen from the figures, the ISI has caused the eye to close to such an extent that it reduces the margin for the additive noise to cause errors. When conventional

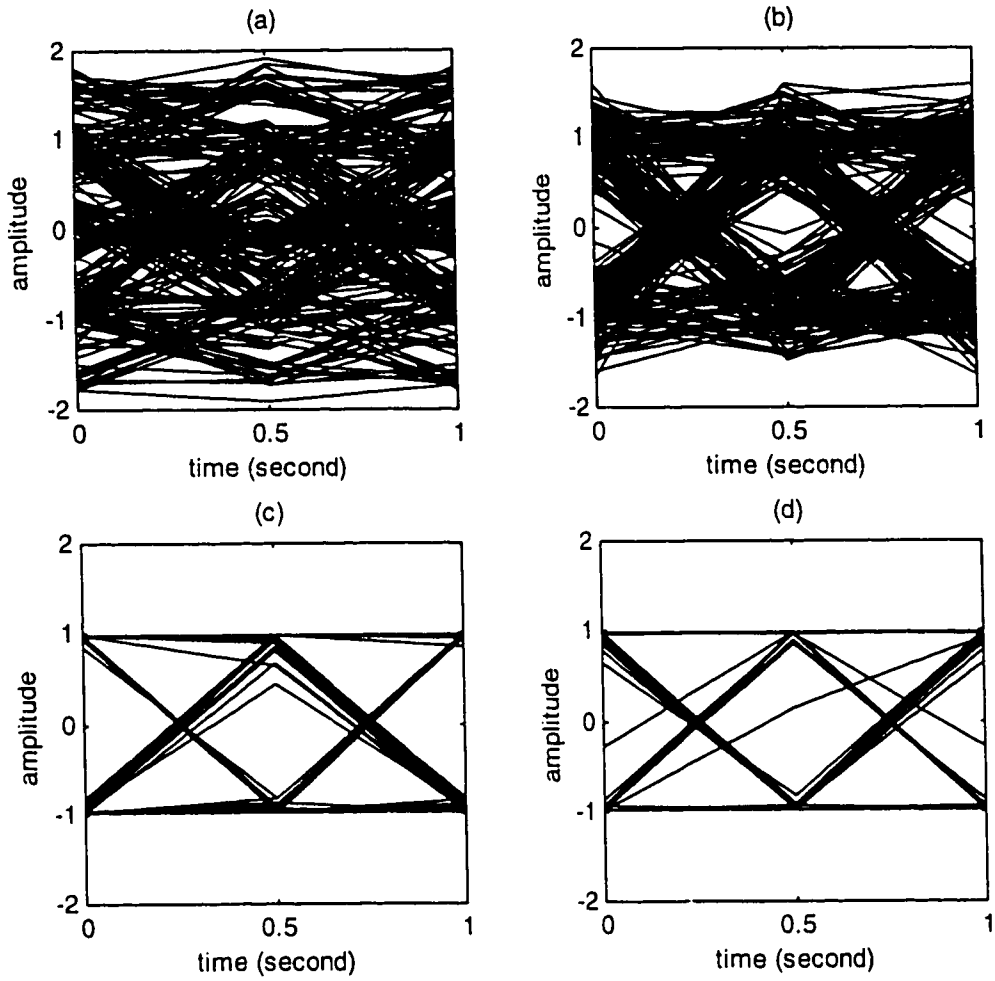


Figure 4.9: Eye diagrams for channel 1 in AWGN environment

(a) Un-equalized data,

(b) Equalized data by (4,1)DFE

(c) Equalized data by (4,1)MLP-DFE

(d) Equalized data by Lattice based (4,1)MLP-DFE

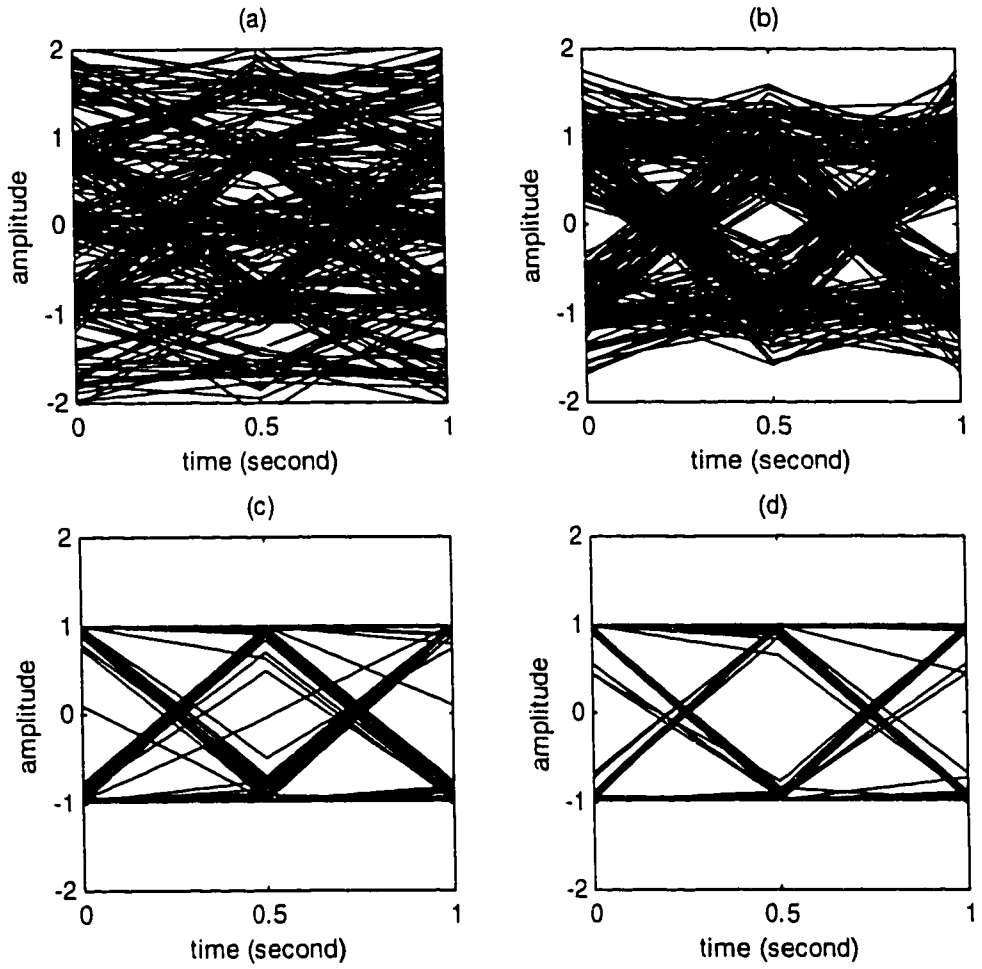


Figure 4.10: Eye diagrams for channel 2 in AWGN environment

(a) Un-equalized data,

(b) Equalized data by (4,1)DFE

(c) Equalized data by (4,1)MLP-DFE

(d) Equalized data by Lattice based (4,1)MLP-DFE

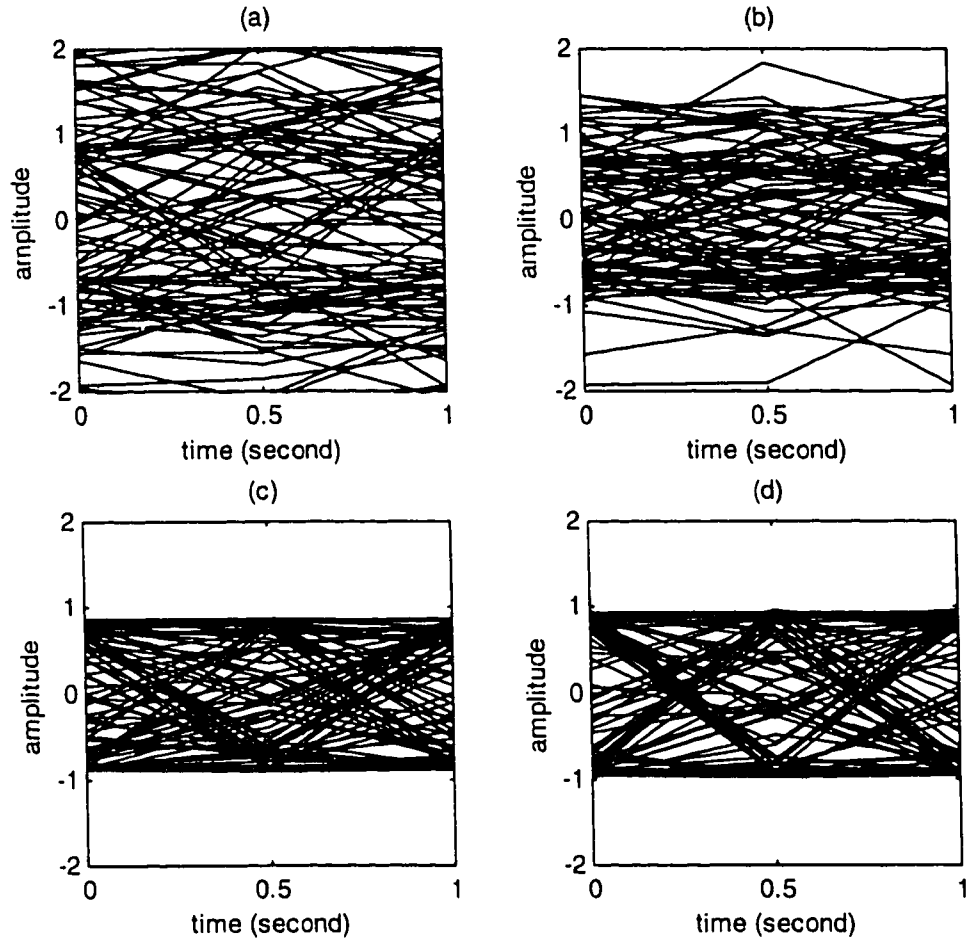


Figure 4.11: Eye diagrams for channel 3 in AWGN environment

(a) Un-equalized data,

(b) Equalized data by (4,1)DFE

(c) Equalized data by (4,1)MLP-DFE

(d) Equalized data by Lattice based (4,1)MLP-DFE

DFE is used to equalize the channel output, it opens the eye to some extent. But the MLP-DFE causes the eye wide-open, thus increasing the noise margin resulting in a better performance. The difference in performance by pre-orthogonalization is evident from the comparison of part (c) and (d) of these figures.

4.2.3 Learning Curves

Simulations were conducted to evaluate the initial convergence rate of the different DFE structures. The performance was determined by taking an average of 600 individual learning curves. Each run had a different random sequence and random starting weights for the perceptron structure.

Figure 4.12 gives the learning curves for channel 1. Part (a) gives the curves for linear equalizers represented as (5,0) DFE's. It is found that for channel 1, the LMS linear equalizer converges in about 500 iterations to a MSE value of about −10 dB. The linear MLP requires at least 1600 iterations to converge to a MSE value of less than -23 dB. This is a result of nonlinear nature of transfer function of the equalizer.

When decorrelation of the received data symbols is employed and applied the resulting data to the MLP with SNR = 20 dB, there is an improvement both in the convergence time and the steady state mean squared error (MSE). For the later case, the MLP-DFE converges in about 1400 iterations to a MSE value of about -26 dB.

The learning curves for (4,1) DFE's are given in part (b). As noted, the DFE's for all the cases, perform better than linear equalizers both in terms of the convergence rate and the steady state MSE. The LMS-DFE converges in about 250 iterations to a value of -14 dB. The MLP-DFE with and without pre-orthogonalization converge relatively slower but to a MSE value which is far lower than that of the LMS-DFE. Without lattice filter,

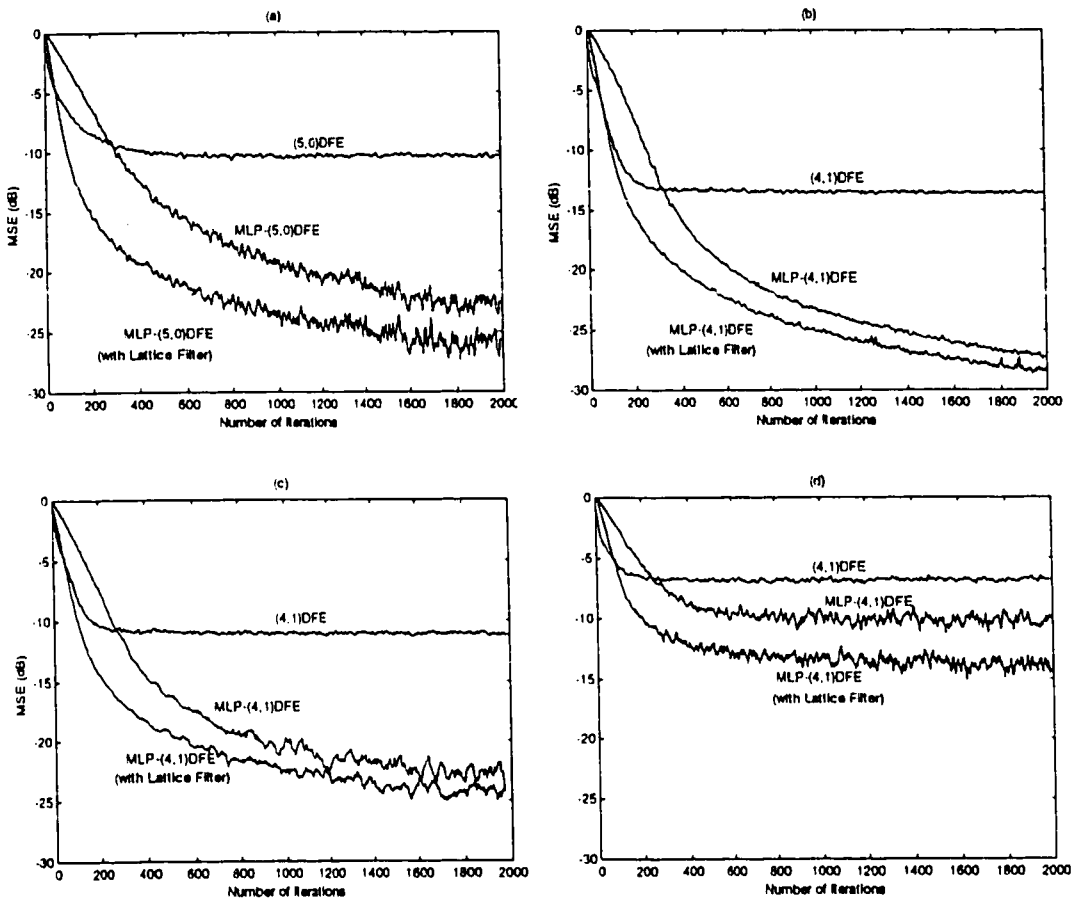


Figure 4.12: Learning curves for channel 1 in AWGN environment

(a) (5,0) Linear equalizers (20 dB),

(b) (4,1) DFE's (20 dB)

(c) (4,1) DFE's (15 dB),

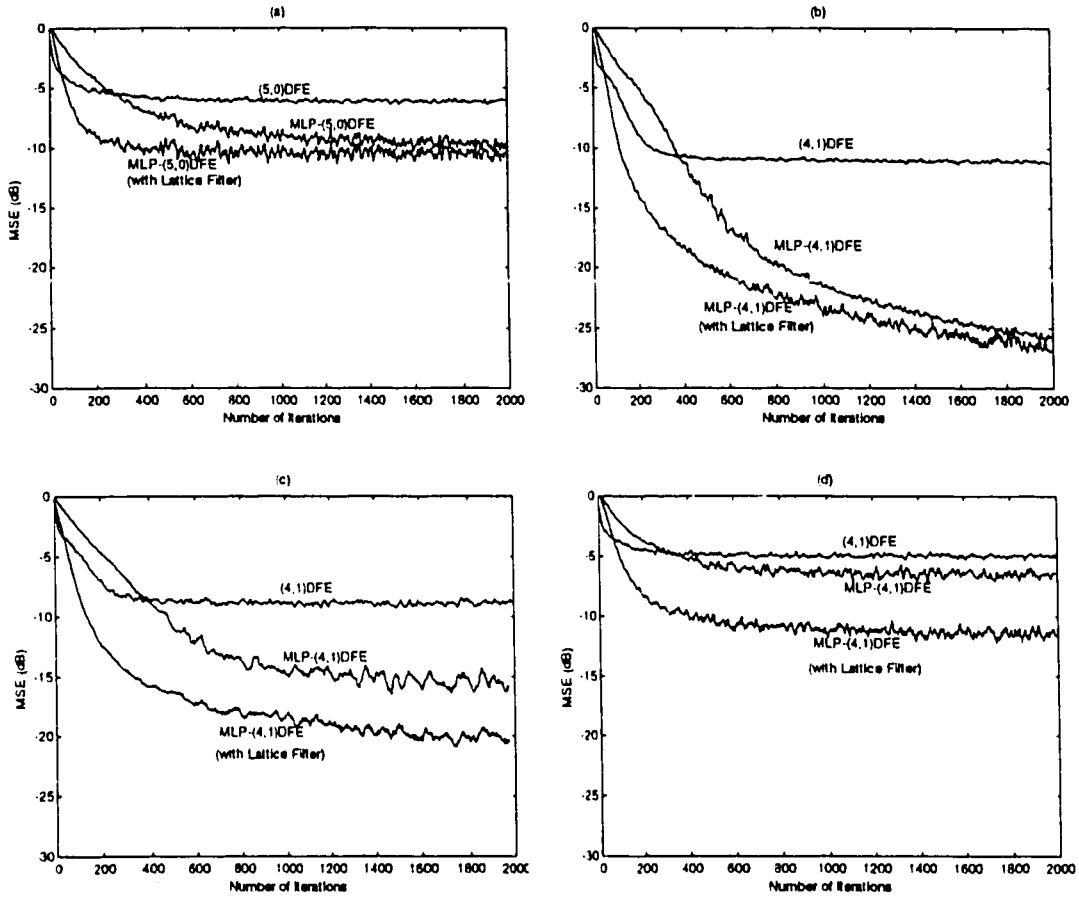


Figure 4.13: Learning curves for channel 2 in AWGN environment

(a) (5,0) Linear equalizers (20 dB),

(b) (4,1) DFE's (20 dB)

(c) (4,1) DFE's (15 dB),

(d) (4,1) DFE's (10 dB)

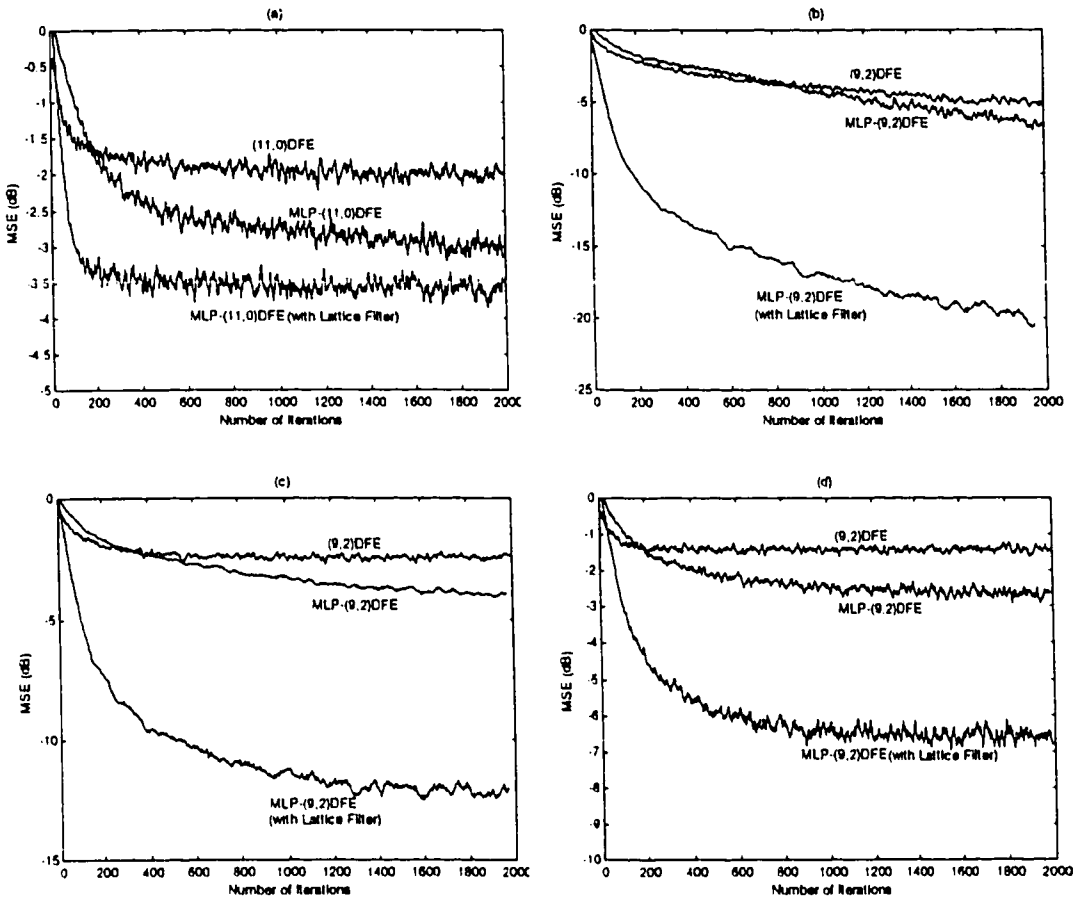


Figure 4.14: Learning curves for channel 3 in AWGN environment

(a) (11,0) Linear equalizers (20 dB),

(b) (9,2) DFE's (20 dB)

(c) (9,2) DFE's (15 dB),

the equalizer converges to around -27 dB while the decorrelation of the data decreases the MSE to value of about -28 dB. The difference is more pronounced as the SNR goes low, as shown in the learning curves for SNR values of 15 and 10 in part (b) and (c), respectively.

From all these figures, it is apparent that the lattice based DFE is outperforming the LMS-DFE and MLP-DFE in both the convergence rate and mean squared error (MSE). Our purpose here is to compare the effect of the presence of lattice module at the input of the MLP-DFE. It means that by applying the lattice filter preprocessor at the input of the MLP-DFE improves the convergence rate along with the decrease in the MSE with the identical parameters used with the actual MLP-DFE.

The same trend persists for channel 2 [Fig. 4.13], which has a very high eigen-value spread. But the difference in convergence time and the MSE between the simple MLP-DFE and the MLP-DFE using Lattice algorithm is now more pronounced. This is due to the eigen-value-ratio insensitivity of the lattice algorithm. For the (4,1) DFE case, there is a distinct MLP-DFE loss in comparison to the channel 1 in the convergence rate but the lattice-based version is not much affected by the eigen-value disparity of the channel. The difference in the two parameters (convergence rate and SSMSE) becomes more pronounced for the SNR value of -15 dB and -10 dB. The corresponding learning curves for channel 3 are given in Fig. 4.14, reflecting the same tendencies as given above.

4.2.3.1 Comparison of (9,3,1) and (3,2,1) MLP Structures

An experiment was conducted for channel 1 to observe the changes in the performance of MLP equalizer by reducing the size of the MLP from (9,3,1) to (3,2,1) i.e. by decreasing the number of neurons by 7. The learning curves for the MLP-DFE with and without pre-orthogonalization are given in Fig. 4.15 and the SNR is set to 10 dB. From the figures, it is observed that the decrease in the network size increases the convergence time of the neural equalizer but the steady state MSE remains unaffected. This is because it takes a long time for the neurons of the small-sized MLP to converge to the optimum weights but once they converge, the mean squared error is the same as that of the comparatively large-sized network. But this trend holds for some particular size of the network, reducing the size even further affects the MSE also because the neurons are no longer able to converge to any optimum value. As noted for the Fig. 4.15, the decrease in the size of the MLP for MLP-DFE causes an increase in the convergence time by around 175 iterations, while the use of lattice filter to decorrelate the input data gives a decrease of only 100 iterations in the convergence time.

Now for channel 2 again (Fig. 4.16), the MSE is the same for the two sizes of the neural network. The (9,3,1) MLP-DFE converges in 800 iterations and (3,2,1) MLP-DFE converges in 1000 iterations but the lattice-based (9,3,1) and (3,2,1) MLP-DFE's converge in around 600 and 800 iterations, respectively. Similarly, Fig. 4.17 gives the corresponding comparison results for channel 3.

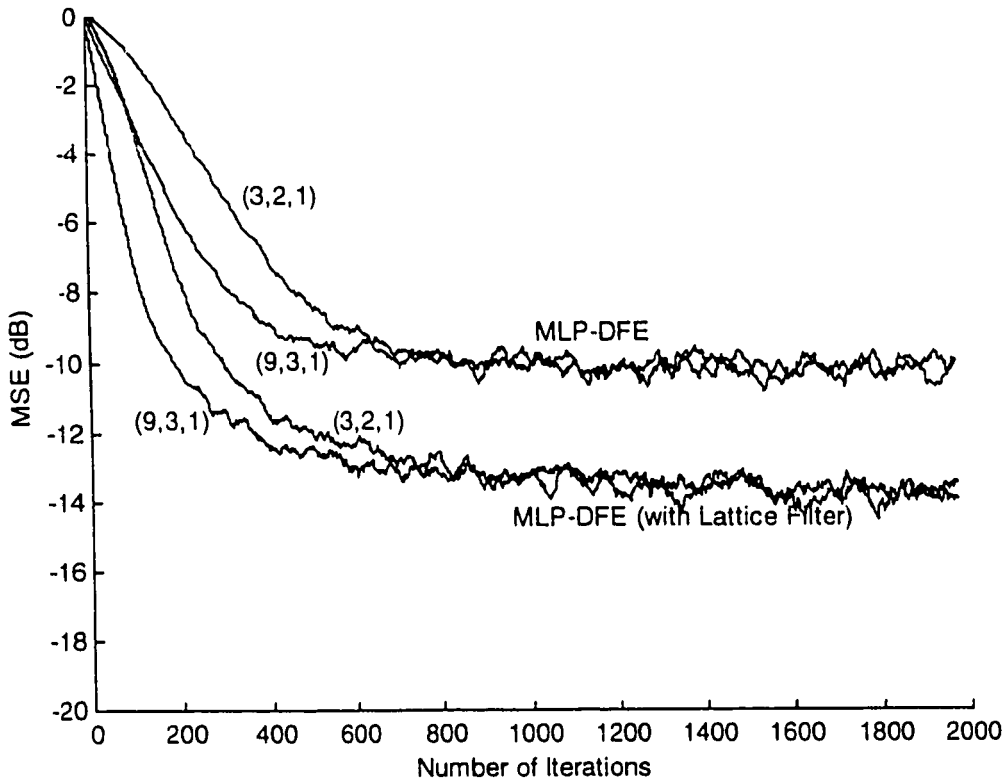


Figure 4.15: Comparison of learning characteristics for MLP with (9,3,1) and (3,2,1) architectures for channel 1 in AWGN environment

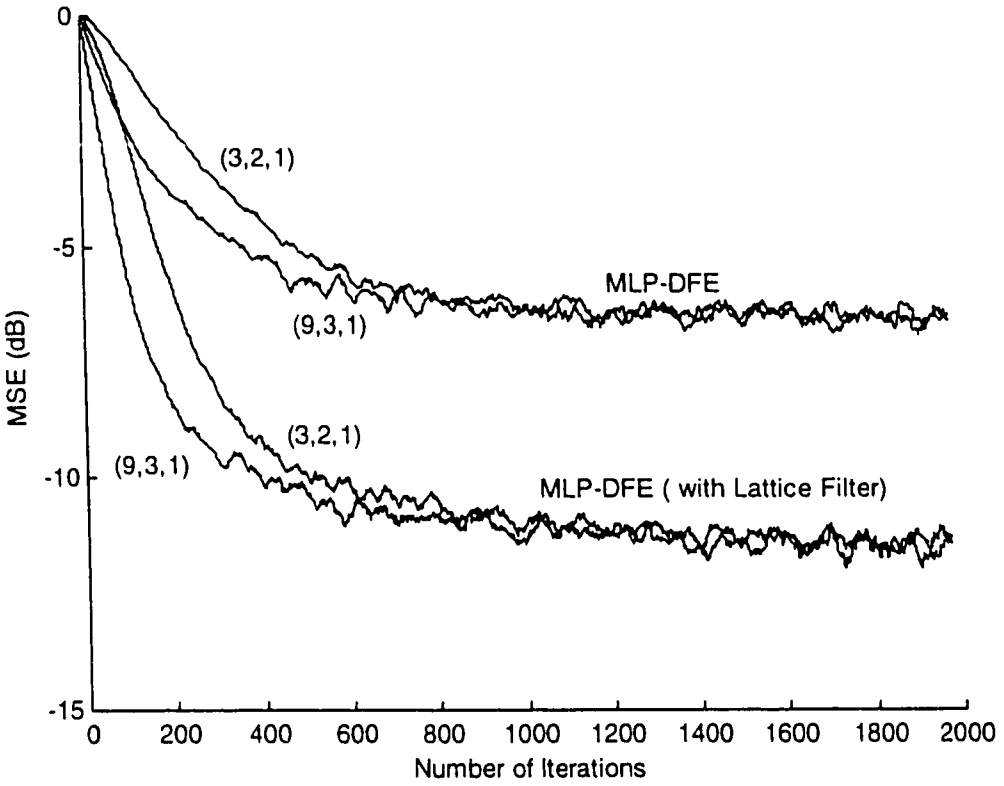


Figure 4.16: Comparison of learning characteristics for MLP with (9,3,1) and (3,2,1) architectures for channel 2 in AWGN environment

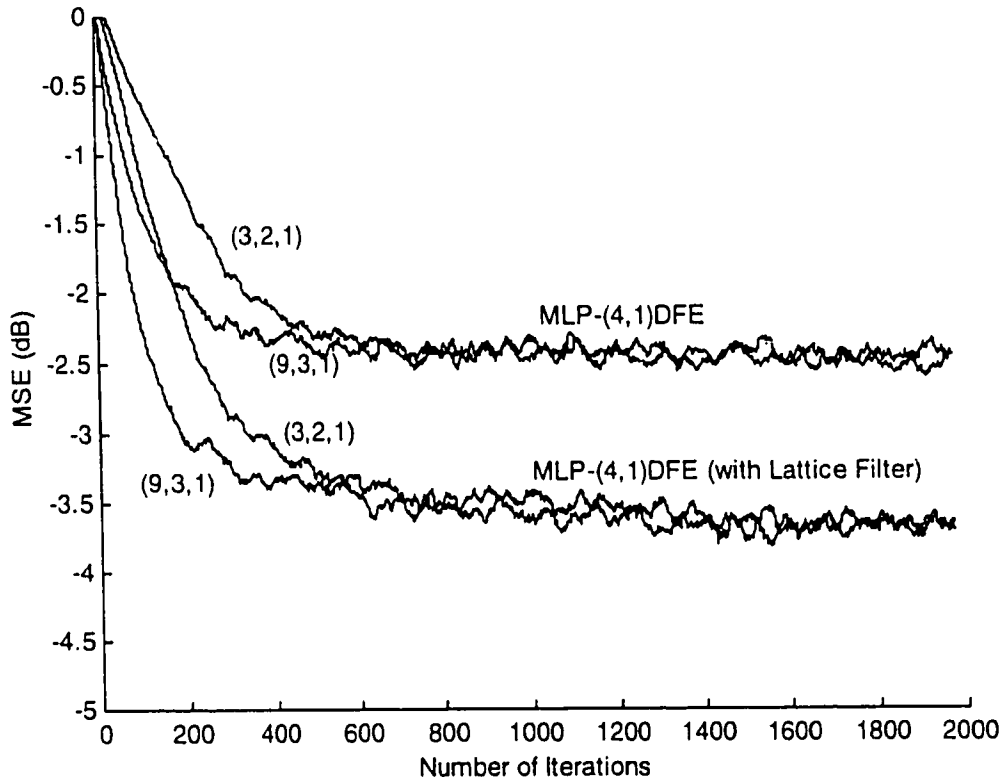


Figure 4.17: Comparison of learning characteristics for MLP with (9,3,1) and (3,2,1) architectures for channel 3 in AWGN environment

4.2.4 Bit Error Rate Performance

Figures 4.18, 4.19 and 4.20 illustrate the BER curves for simple MLP-DFE and Lattice-based MLP-DFE for channel 1 and 2, respectively. The BER is monitored after the training phase.

For reference, the BER performance of linear equalizers is also presented. As noted, the performance of linear DFE is the worst of all. The linear MLP with and without pre-orthogonalization perform almost the same, but it may be observed that the equalizer with lattice structure outperforms the simple MLP-DFE for all SNR values for both of the channels. We get a gain of about 2 dB at BER of 10^{-3} by decorrelating the equalizer-input data for the MLP-DFE. The MLP-DFE structure performs better in comparison with the LMS-DFE structure, when the level of additive noise is high, but deteriorates as the signal to noise ratio improves. This is because of the fact that if the MLP-DFE receives very few samples of signal which are close to the optimal decision boundary, rendering it incapable of forming optimal decision boundary as it does in high noise situation. For reference, the BER performance of the optimal MLSE equalizer is also given for channel 1 and 2.

Again, the difference is more distinct for the case of channel 2 as evident from the Fig. 4.19. The performance of linear equalizers is poor and the LMS-DFE is again inferior to the MLP-DFE. The gain we get by the decorrelation of input data over the MLP-DFE is 2 dB at BER = 10^{-3}.

The case of channel 3 is the worst of all due to the spectral nulls in the frequency response of this channel. The (4,1) DFE, whether, LMS or MLP structure does not perform satisfactorily. Because of the length of the impulse response of this channel, it

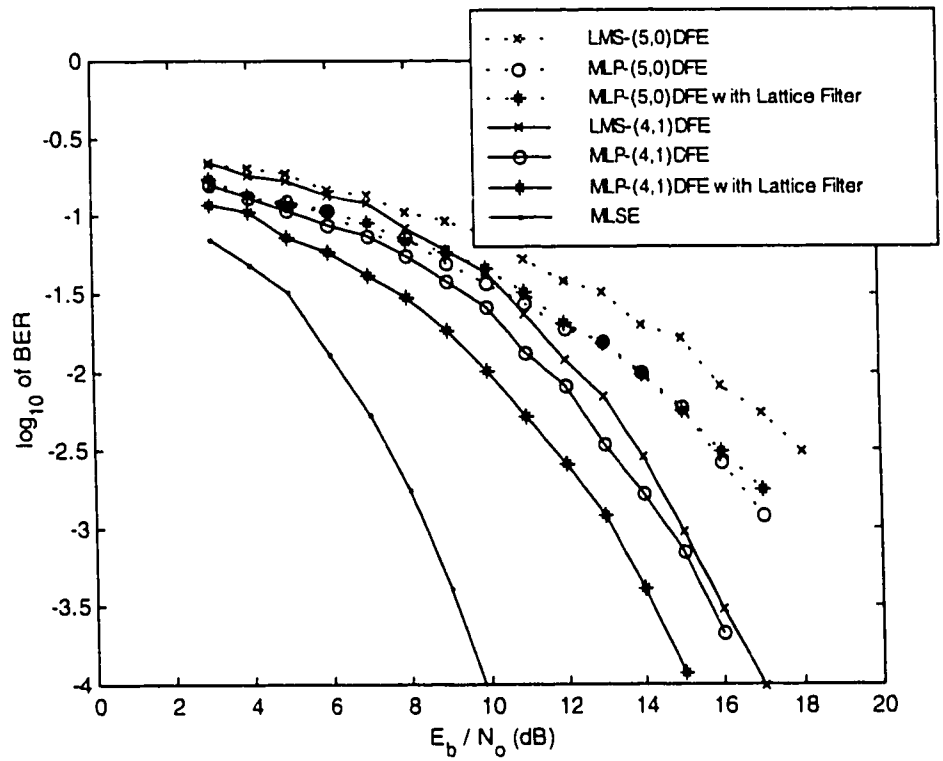


Figure 4.18: BER curves for channel 1 in AWGN environment

Dashed line = Linear Equalizer

Solid line = DFE

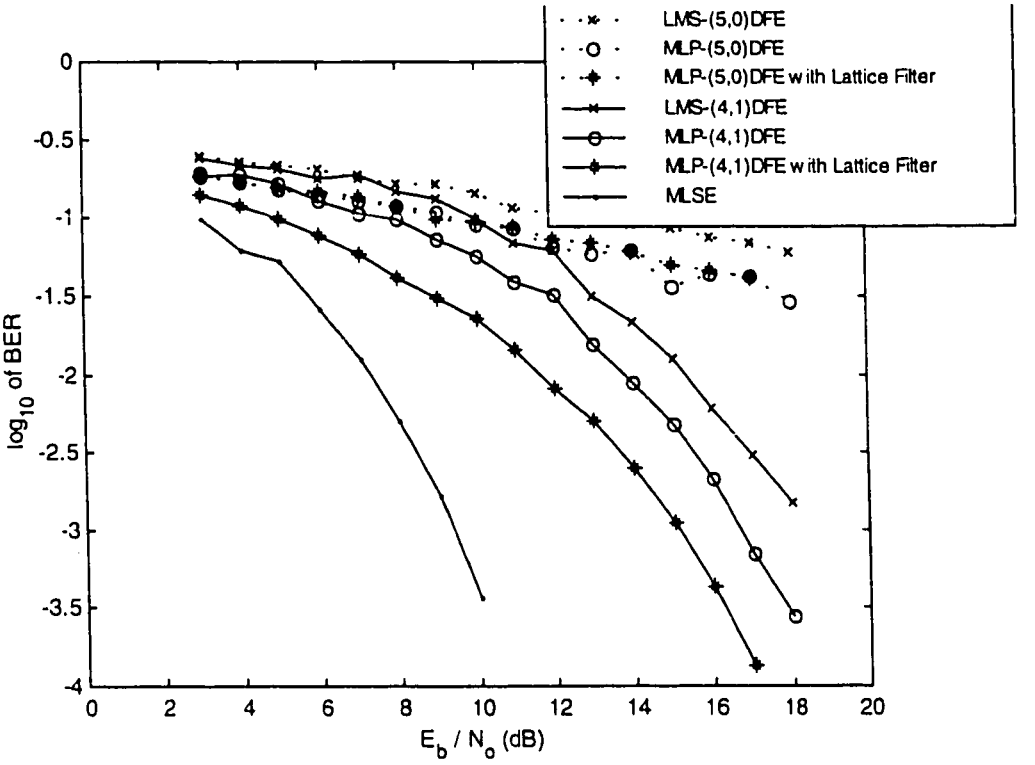


Figure 4.19: BER curves for channel 2 in AWGN environment

Dashed line = Linear Equalizer

Solid line = DFE

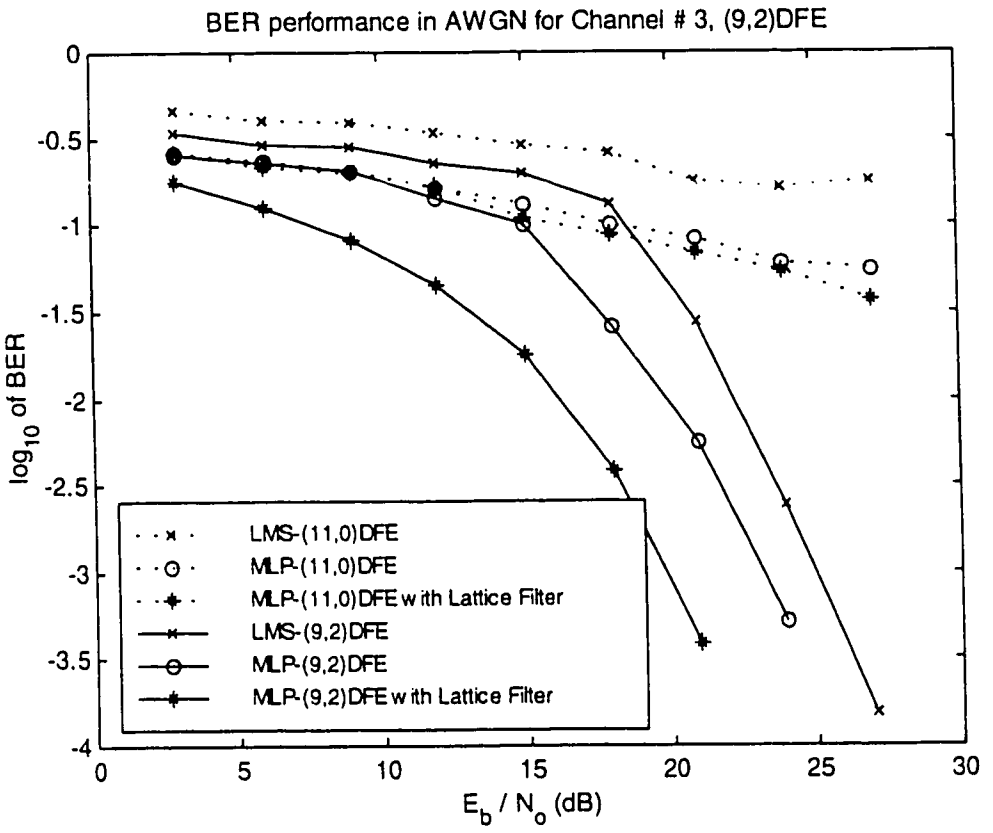


Figure 4.20: BER curves for channel 3 in AWGN environment

Dashed line = Linear Equalizer

Solid line = DFE

is more reasonable to use a bigger structure of the equalizer to combat the ISI. Thus, we used (11,0) linear equalizers and (9,2) DFE's instead. The BER curves for this case are given in Fig. 4.20. Again, the performance of linear equalizers is also given for reference. As noted from these results, the lattice-based linear MLP is performing a little better than the simple linear MLP equalizer as opposed to the case of channel 1 and 2 where there was almost no difference between the two configurations. The MLP-DFE with Lattice structure outperforms the conventional MLP-DFE by 3.5 dB at BER = 10^{-3}.

4.2.5 Effect of Colored Noise

While in many cases the input noise can be considered white, there are some cases in which it is colored and cannot be considered as white. For example, in radar applications, when a target is present, the input to the adaptive prediction filter consists of a signal plus a colored noise component. When there is no target, the input consists of colored noise due to radar clutter. Similarly, in applications involving the equalization of data signals transmitted through modems, the noise component could be either white or colored and could include additive and multiplicative components.

While the effect of white noise is to simply increase the eigenvalues of the input correlation matrix by the variance of the noise, the effect of colored noise is more complicated. Here, the eigen-values are altered by different quantities which depend on the noise correlation matrix and the eigenvectors of the signal correlation matrix [52].

The two parameters of special importance for comparison of the adaptive equalizer's behavior in both cases are the rate of convergence and the steady state mean squared

error. It is proven in the literature [52], that the presence of colored noise instead of white noise can either increase or decrease the convergence rate and the steady state mean squared error of the adaptive equalizer. The increase or decrease in these parameters depends upon the characteristics and nature of the colored noise.

We used the White Gaussian noise to generate the correlated noise by passing it through an IIR filter of specified taps correlation coefficients. The filter model used is auto-regressive IIR filter

$$H(z) = 1/(1 - p*z^{-1})$$

Where p is the tap value or the correlation coefficient of the filter.

The results are presented for the correlation coefficient of 0.7 as done in [65]. The variance of the colored noise is adjusted to have the same value as of AWGN by normalizing the noise power to unity and then multiplying it by the appropriate power value.

4.2.5.1 Learning Characteristics

The learning characteristics curves for the three equalizers in additive colored Gaussian noise (ACGN) and AWGN environments at SNR of 10 dB are given in Fig. 4.21, 4.22 and 4.23 for channel 1, 2, and 3, respectively.

From Fig. 4.21, we see that the performance of all the equalizers in Additive Colored Gaussian Noise (ACGN) is better in terms of steady state error (MSE). But the convergence rate has decreased to some extent. For the case of AWGN, the MLP-DFE converges to a steady state value of −11 dB in about 800 iterations and the Lattice-based MLP-DFE does so to −14 dB in about 700 iterations. On the other hand, the introduction of colored noise of the same variance causes the MLP-DFE to converge to

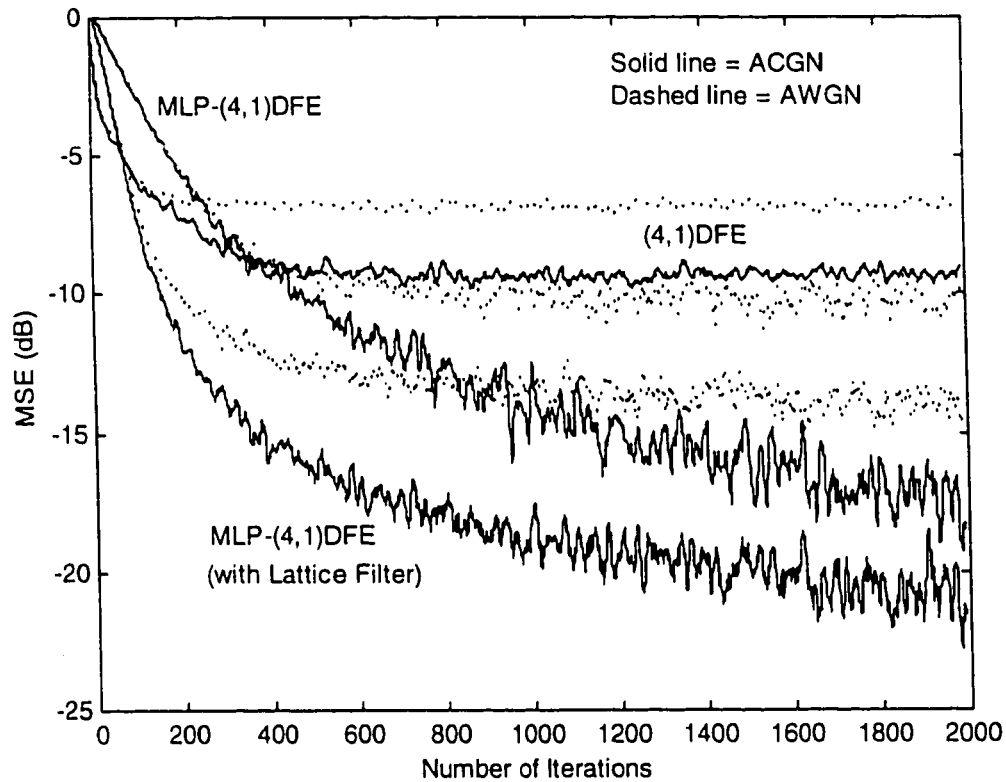


Figure 4.21: Learning curves for channel 1 in AWGN and ACGN environments at SNR = 10 dB

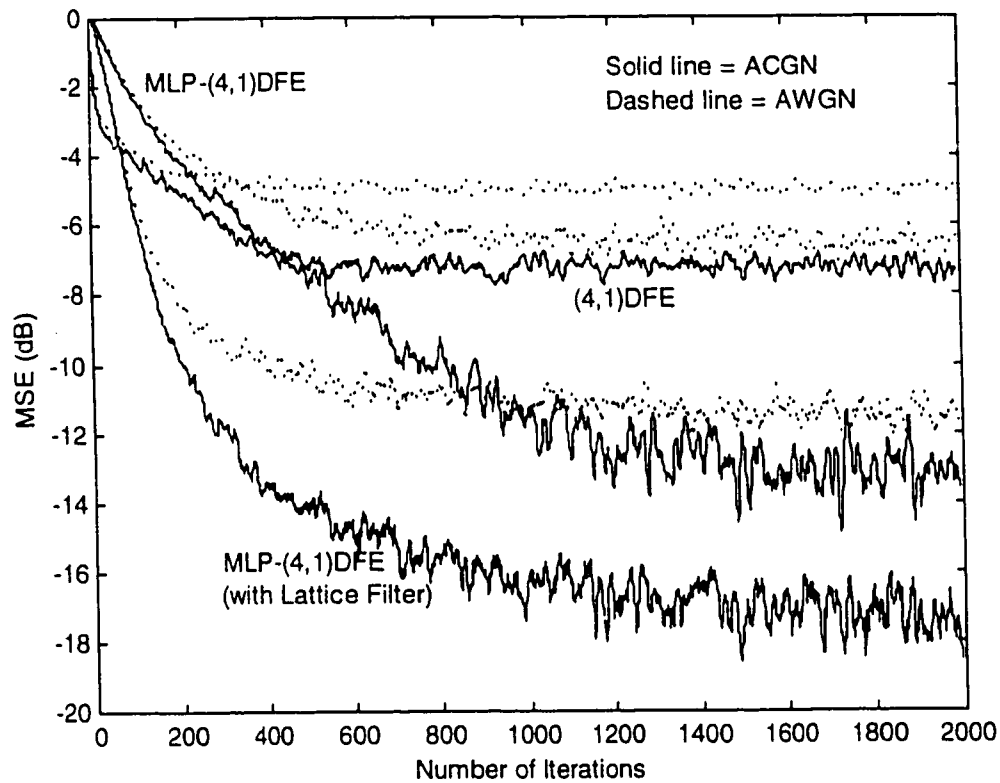


Figure 4.22: Learning curves for channel 2 in AWGN and ACGN environments at SNR = 10 dB

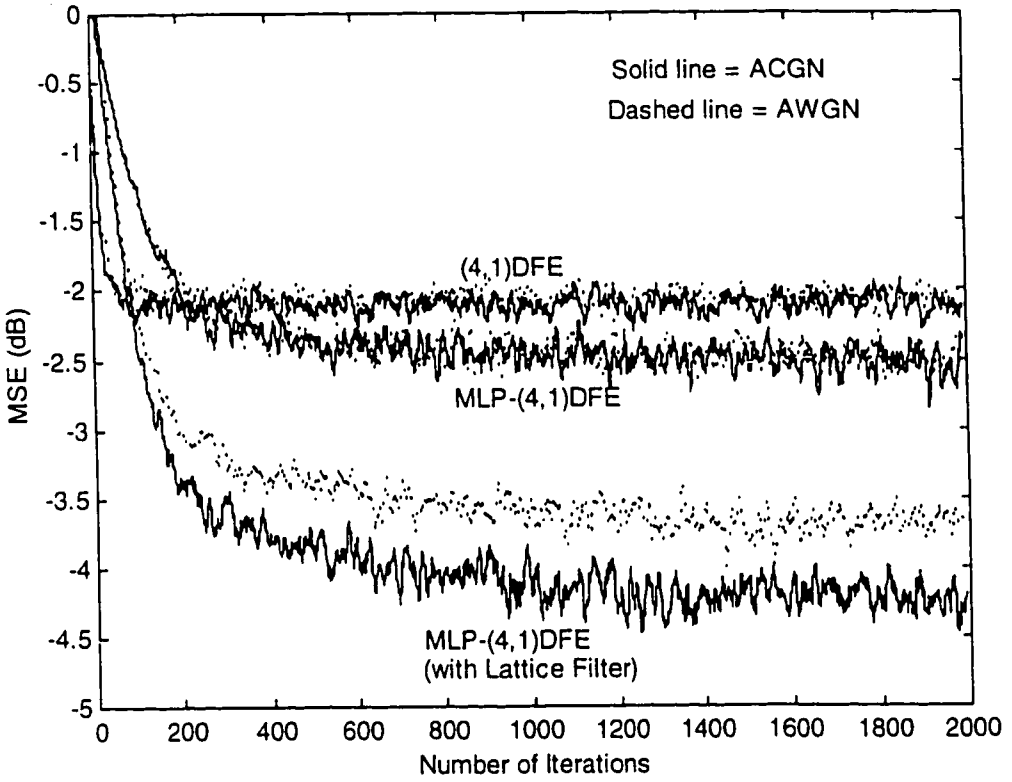


Figure 4.23: Learning curves for channel 3 in AWGN and ACGN environments at SNR = 10 dB

−17.5 dB but it takes 1700 iterations for it to converge. Thus, a gain of 6.5 dB at the expense of 900 iterations. The Lattice based MLP-DFE converges to MSE value of -21 dB in about 1500 iterations. Thus, a gain of 7 dB at the loss of 800 iterations.

Thus, the change in the performance is larger for the case of MLP-DFE with pre-orthogonalization as compared to the conventional MLP-DFE. This is due to the decorrelation property of the lattice structure. The difference becomes more pronounced as the SNR goes low because at high SNR value the effect of noise gets smaller. But the point to be noted in these curves is that the difference in the curves for conventional MLP-DFE and the Lattice MLP-DFE for both noise cases is larger for low SNR values. The reason being as stated above.

For channel 2 and channel 3, the corresponding learning curves in ACGN environment are given in Fig. 4.22. and Fig. 4.23, respectively. Again the difference is apparent from the learning curves at SNR = 10 dB, reflecting the same behavior as that of channel 1. Especially, for channel 3, the difference is far more pronounced as compared to the rest two channels.

4.2.5.2 Bit Error Rate (BER) Curves

The BER performance of different equalizer configurations in the three channels for the case of additive colored Gaussian noise are given in Fig. 4.24, 4.25, and 4.26 .

As seen from Fig. 4.24, the performance of the equalizers is improved in Additive Colored Gaussian Noise environment. The improvement is more pronounced in the low SNR regions where the noise power is high. In AWGN environment, the MLP-DFE with and without decorrelation achieve BER of 10^{-3} at SNR = 13 dB and 14.5 dB,

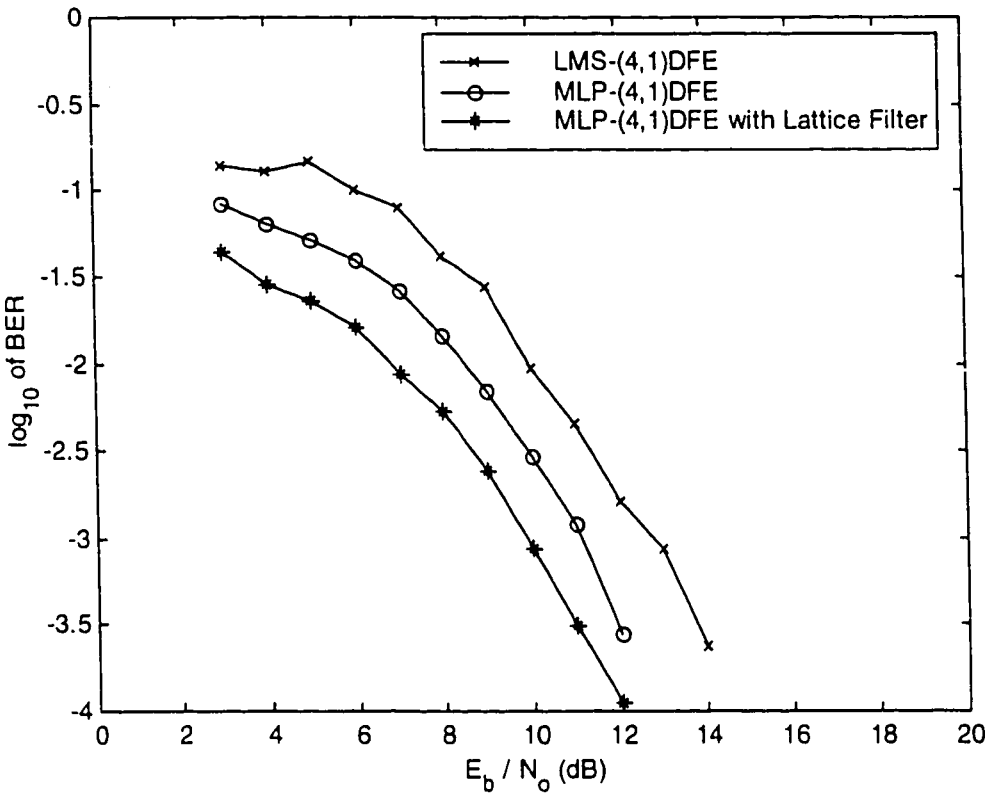


Figure 4.24: BER curves for channel 1 in ACGN environment

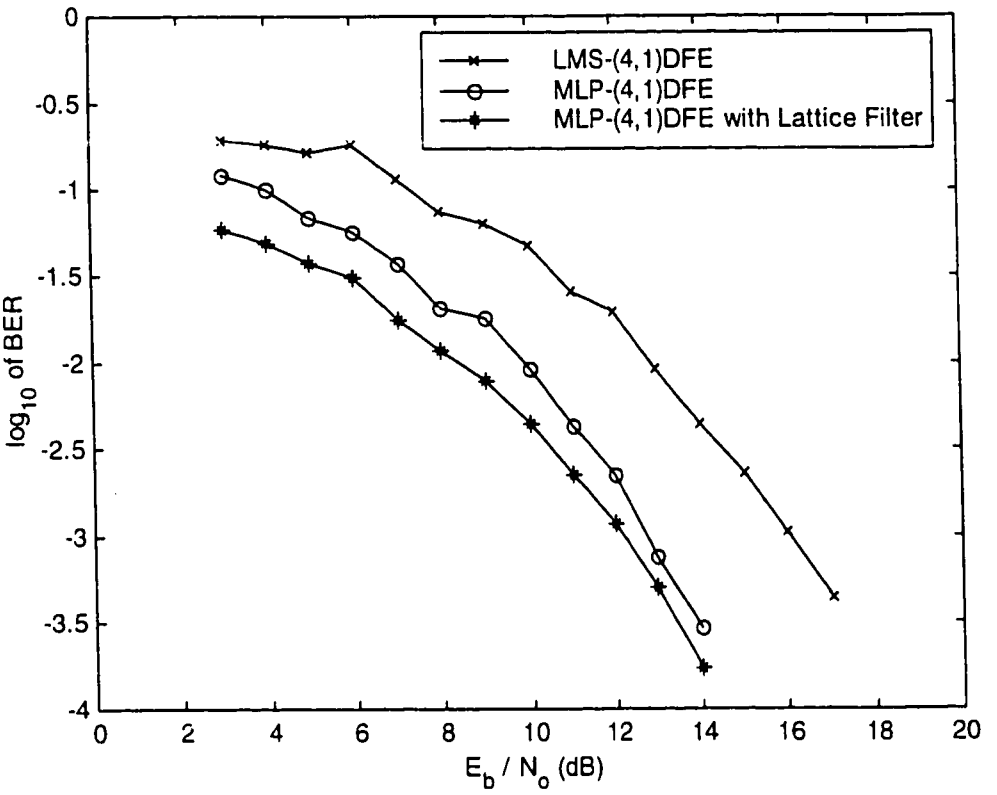


Figure 4.25: BER curves for channel 2 in ACGN environment

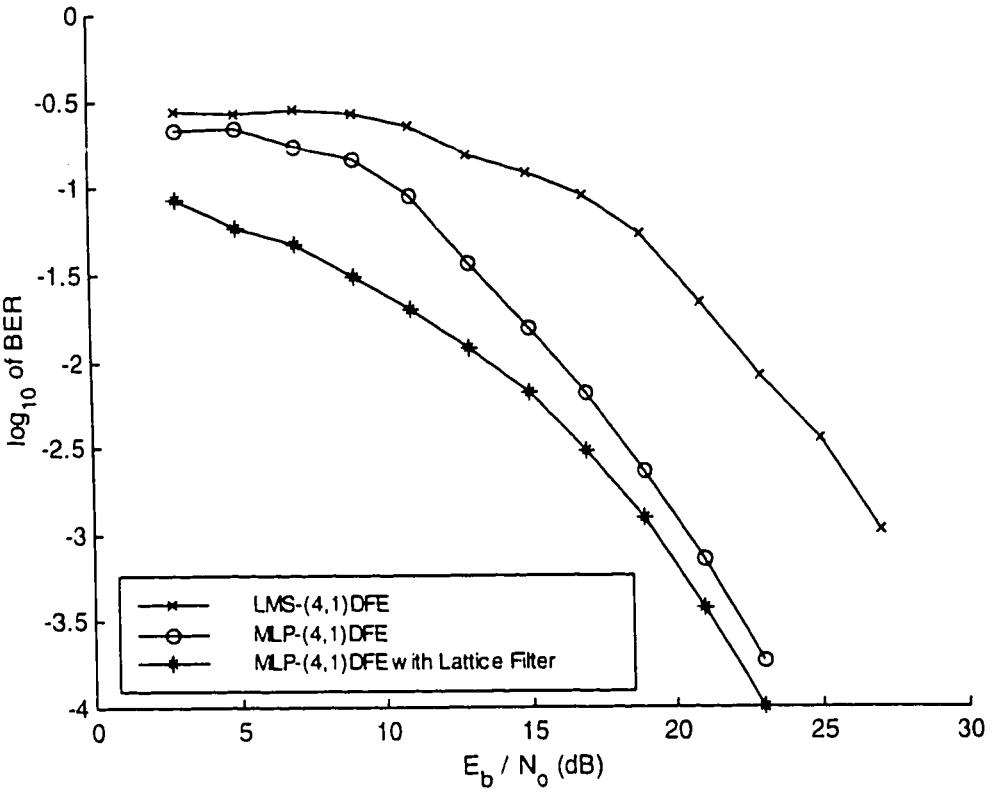


Figure 4.26: BER curves for channel 3 in ACGN environment

respectively. On the other hand, for channel 1 in ACGN environment, the same BER is achieved by the two configurations at SNR = 10 dB and 11.5 dB.

The same trend persists for the channel 2 also (Fig. 4.25). The MLP-DFE attains the BER of 10^{-3} at SNR of 13 dB, while Lattice based MLP-DFE acheives the same BER at 12.5 dB in AWGN case, and the corresponding SNR values for the same BER in ACGN case are17 and 15 dB, respectively. Here we get a difference of 2.5 dB for lattice-based MLP-DFE and 4 dB for simple MLP-DFE at BER of 10^{-3}.

Due to the severe nature of channel 3 (Fig. 4.26), the effect of the colored noise is not prominent as compared to the other two channels. But still the same trend is present. The MLP-DFE and lattice-based MLP-DFE achieve BER of 10^{-3} at SNR values of 21 and 19.5 dB in ACGN case, as opposed to the corresponding SNR values of 23 and 20 dB in AWGN case. Here we get the difference of around 2 dB for MLP-DFE while the difference for lattice based DFE in two noise cases is only 0.5 dB.

From the above, it is observed that all the equalizer configurations are performing better in ACGN environment as compared to the AWGN environment. The difference is more for the case of MLP-DFE than the lattice based DFE. This fact should not surprise the reader because the lattice based-DFE being a pre-orthogonalizing filter, decorrelates the noise samples, thus deteriorating the performance of the equalizer.

4.3 Computational Complexity of the Algorithms

Following is given the computational complexity of different algorithms used in this thesis, i.e. the MLP-based DFE, Lattice-based DFE and gradient-based DFE. Different notations used in these equations are defined as follows:

N1 = # of feedforward taps of the equalizer

N2 = # of feedback taps of the equalizer

N = N1 + N2 = # of inputs to the MLP

L1 = # of neuron in the first hidden layer of MLP

L2 = # of neuron in the second hidden layer of MLP

L3 = # of neuron in the output layer of MLP = 1

LMS Algorithm

Total # of Operations = 2*N +1

# of Divisions = 0

MLP DFE

Total # of Operations = N + 11*L1 + 17*L2 + 5*N*L1 + 6*L1*L2 + 9

# of Divisions = 2*L1 + 2*L2 + 2

Lattice-based MLP-DFE

Total # of Operations = 19*N1 + 40*N2 + 11*L1 + 17*L2 + 5*N*L1 + 6*L1*L2 - 30

# of Divisions = 2*N1 + 2*L1 + 2*L2 + 2

Here, we have taken the # of hidden neurons in the output layer equal to 1 for the equalizer applications. From the above, we see that the LMS algorithm is simplest of all and by using the Lattice filter at the input of MLP-DFE, we are putting a load of (18*N1 + 39*N2 – 39) computations. Thus, there is a trade-off between the performance and complexity of the system.

Chapter 5

SYSTEM PERFORMANCE IN NONLINEAR AND TIME VARYING CHANNELS

In this chapter, we investigate the effect of pre-orthogonalization for some nonlinear and time varying channels.

5.1 Non-linear System Models

A non-linear dynamic single-input single-output channel is a mapping of an input sequence $\{x_i\}$ to an output sequence $\{y_i\}$,

$$y_i = \theta(x_i, x_{i-1}, \ldots, x_{i-N}) \qquad (5.1.1)$$

Where $\theta(.)$ is an arbitrary (nonlinear) multivariable function, i denotes the time instant, and N is the memory length of the system, which need not be finite. Though,

the implementation of a system with infinite memory requires in general infinite complexity, by introducing a state space model

$$S_{i+1} = f(s_i, x_i) \qquad (5.1.2a)$$

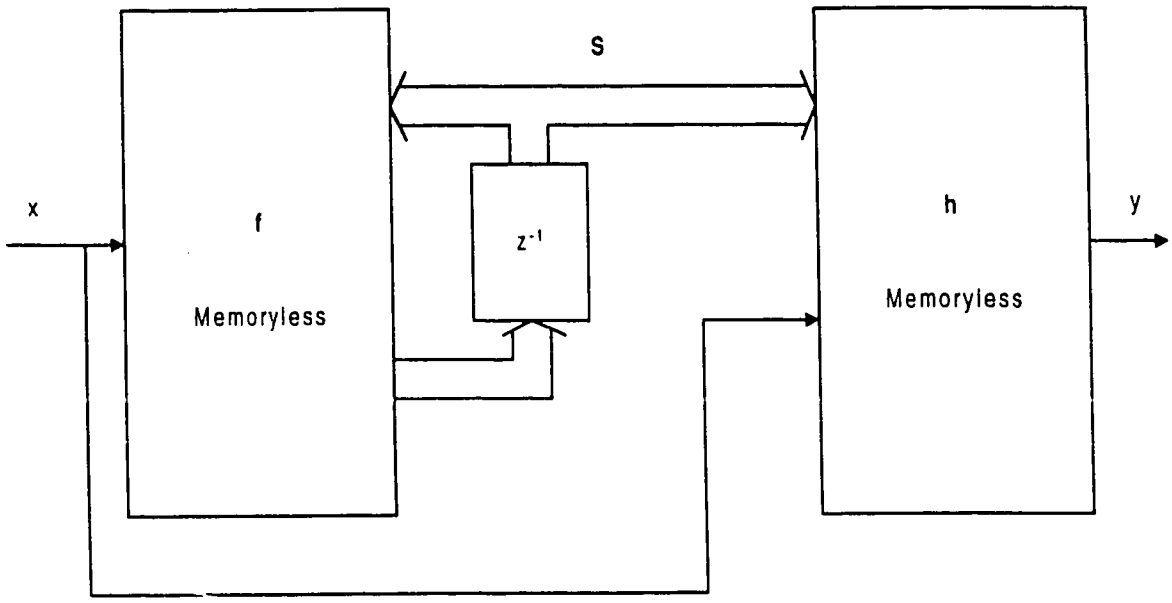


Figure 5.1: A nonlinear system model

$$y_i = h(s_i, x_i) \qquad (5.1.2b)$$

where, $s_i = (s_i^{(1)}, s_i^{(2)}, \ldots, s_i^{(M)})$ is the state vector, a finite hardware implementation is made possible. The generic block diagram of such a nonlinear dynamic system is shown in Fig. 5.1

It consists of two memoryless blocks f and h with memory elements in between. The input-to-state map f has M+1 inputs (x_i and the M elements of s_i), M outputs (the M elements of s_i), while the state-to-output map h has M+1 inputs but only one output. Consequently, the cost of implementing f is much higher than that of implementing h. This has motivated several attempts (the best known by Wiener [53]) to approximate

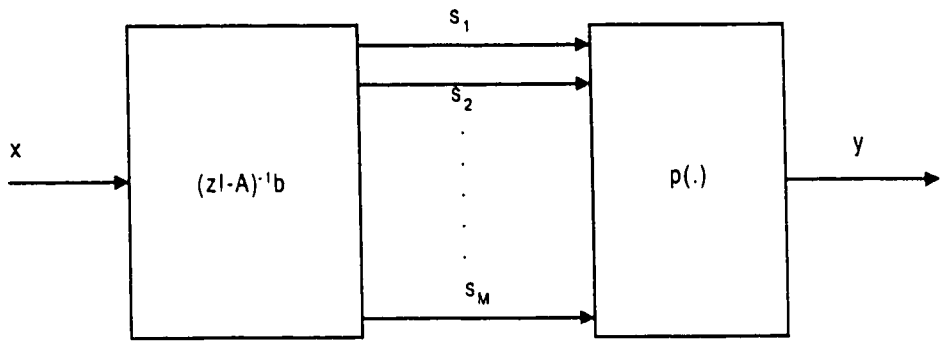


Figure 5.2: An approximation consisting of linear dynamic system with polynomial nonlinearity

nonlinear systems by models of the form Equation of (5.1.2), but with linear input-to-state mapping f. In the following, we discuss this approximation.

5.1.1 Linear Dynamic Polynomial Nonlinearity (LDPN) Approximation

Fig 5.2 shows the block diagram of one kind of approximation which consists of single-input multi-output (SIMO) linear time invariant (LTI) operator followed by a multi-input single-output (MISO) memoryless polynomial nonlinearity. The LTI part can be realized as a finite dimensional linear system

$$s_{i+1} = As_i + bx_i \qquad |eig(A)| < 1 \qquad (5.1.3)$$

which replaces the nonlinear mapping f of Equation (5.1.2) is constrained to be a multi-variable polynomial of degree L, i.e.

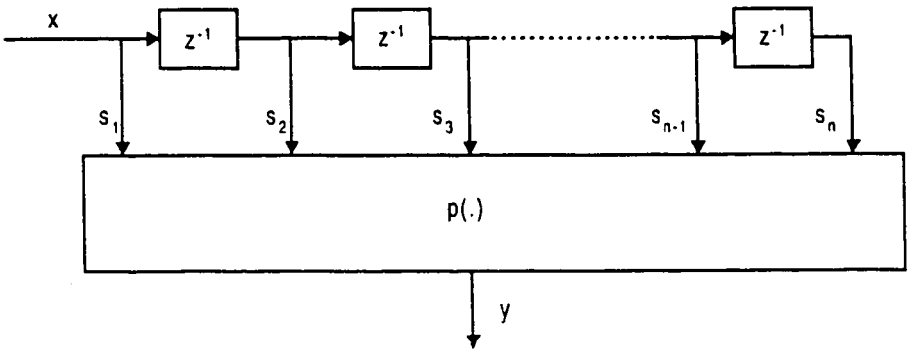


Figure 5.3: A simple approximation structure for discrete time nonlinear system

$$y_i = P(s_i) \qquad (5.1.4)$$

Boyd [4] has shown that such a structure can be used as an approximation, and by letting L and M grow to infinity, one obtains better and better approximation of an arbitrary nonlinear systems with fading memory. It is also demonstrated that an even simpler configuration in Fig. 5.3 is sufficient to approximate discrete time systems. It corresponds to the popular finite Voltera model [4]

$$y_i = h^{(0)} + \sum_{m1=0}^{N} h_{m1}^{(1)} x_{i-m1} + \sum_{m1=0}^{N} \sum_{m2=0}^{N} h^{(2)}_{m1m2}\, x_{i-m1}\, x_{i-m2} + \ldots + \sum_{m1=0}^{N} \ldots \sum_{m2=0}^{N} h^{(L)}_{m1m2\ldots mL}$$

$$x_{i-m1}\, x_{i-m2} \ldots x_{i-mL} + \ldots\ldots\ldots\ldots \qquad (5.1.5)$$

Where, $h^{(l)}$ $(0 \le l \le L)$ is called the discrete-time Voltera Kernel associated with an input data product of order l, and N is the length of the system's memory. For a detailed discussion of the Voltera and Wiener theory the reader may refer to [63].

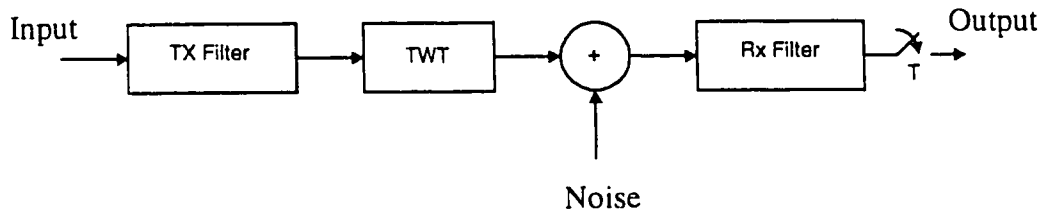


Figure 5.4: A satellite channel model

5.1.2 Non-linear Channels

In satellite communications, the travelling wave tube (TWT) amplifier in satellite usually operates in the saturated region because of the several signal attenuation and the limited power sources of the satellite. Such operation results in severely nonlinear channels. The nonlinear effect of the channel not only spreads the signal spectrum, but also introduces nonlinear amplitude and phase distortions to the channel in-band signal. Fig. 5.4 shows the nonlinear satellite channel model. Besides satellite links, this model has also been applied to voice-band telephone channels.

The first general simple Voltera model with order three for the voice-band channel was proposed by Falconer [10]. A more simplified version of Falconer's model was used by Biglieri et al [3] as shown in Fig. 5.5, where a_2 and a_3 are scalars, which control the severeness of nonlinearity. Therefore, Fig 5.5 results in a highly simplified version of the Voltera model shown in Fig. 5.3. Hence, a significant reduction in the number of parameters of the model is obtained. In this model, only one filter is used and by doing this, we can easily control the severeness of nonlinearity with only two scalars a_2 and a_3.

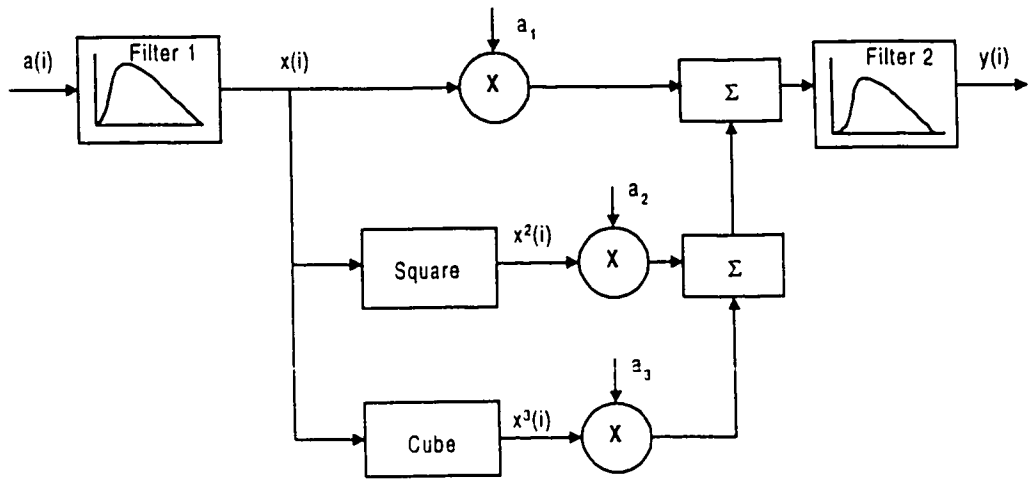


Figure 5.5: A simplified nonlinear channel model

Going back to the discussion of nonlinear system modeling above, we recognize this configuration as a highly simplified version of the Voltera model of Fig. 5.3. This simplification results in a significant reduction in the number of parameters of the model.

5.2 Performance Analysis

5.2.1 Channel Models

Two nonlinear channels with different degrees of nonlinearity are used. The first channel, denoted as non-linear channel I, uses the linear channel 1 as the FIR filter in the nonlinear channel. Whereas, for the second nonlinear channel, denoted as non-linear channel II, the linear channel 2 is used as the FIR filter. The system is tested for two

levels of nonlinearity, first one of mild nature and the other one of higher degree. The two degrees of nonlinearities are controlled by setting the first, second and third-order coefficients, a_1, a_2, a_3 to 1, 0.1 and 0.05 as given in [40], respectively, and in the second case, the three coefficients are set equal to 1, thus increasing the nonlinearity content. The equalizers used have either (11,0) or (9,2) configuration for linear and DFE equalizers, respectively. The MLP used has (9,3,1) structure and the parameters of the network are:

Learning gain parameter (η) = 0.07

Momentum parameter (α) = 0.3

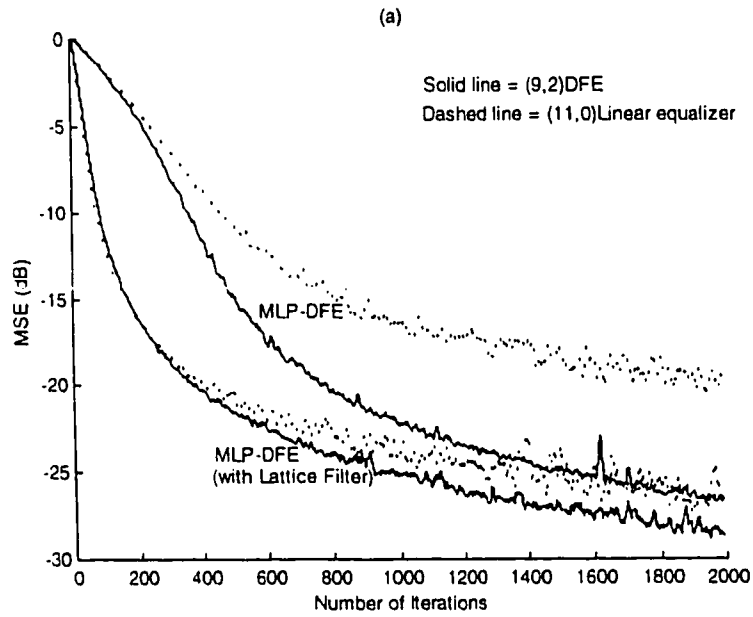
Threshold level adaptation gain (β) = 0.05

The weighting factor w for lattice algorithm is again 0.99 and the step size for the gradient transversal equalizer is the one that gives the fastest convergence.

5.2.2 Learning Characteristics

The learning curves for different equalizer configurations for the two nonlinear equalizers at SNR = 20 dB are given in Fig. 5.6 and 5.7 and the additive noise used is white Gaussian noise.

As noted from Fig. 5.6(a), the learning performance in AWGN environment of channel I with nonlinearity (a_1 = 1, a_2 = 0.1, a_3 = 0.05), there is little deterioration in the convergence time as compared to that of channel 1 in chapter 4. The steady state mean squared error (SSMSE) for both of the cases is 1 dB greater than that of the corresponding linear channel 1.



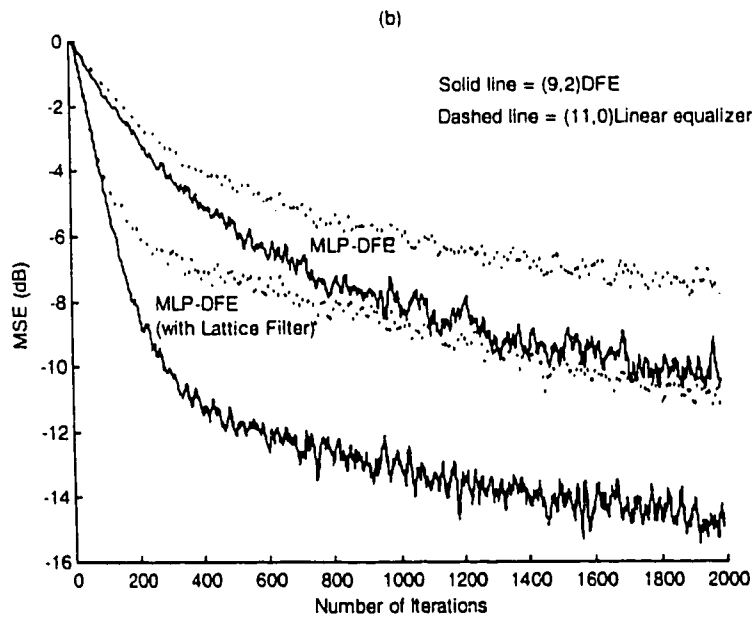
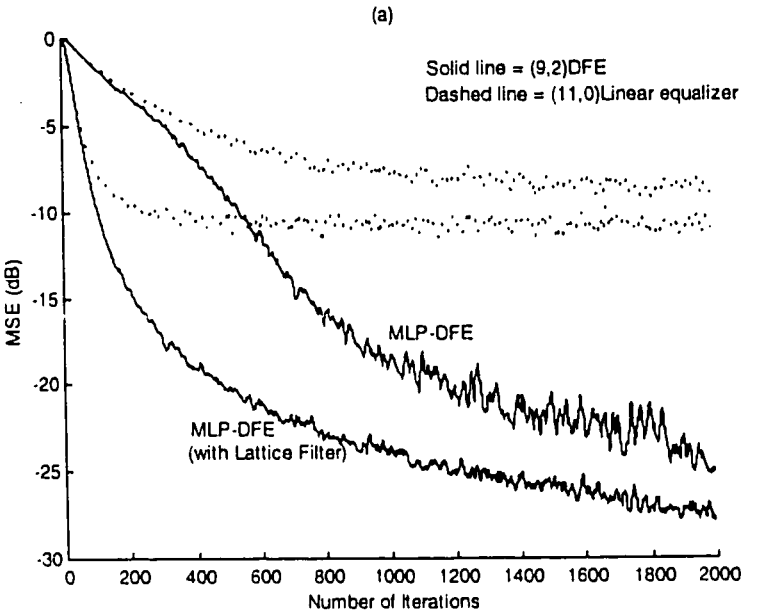


Figure 5.6: Learning Curves for nonlinear channel I in AWGN environment
(a) ($a_1 = 1$, $a_2 = 0.1$, $a_3 = 0.05$)

(b) ($a_1 = a_2 = a_3 = 1$)



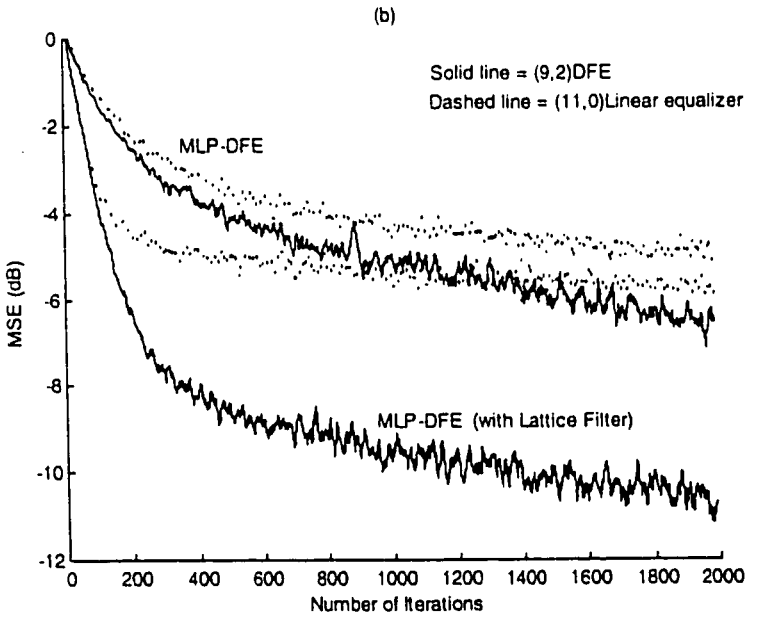


Figure 5.7: Learning Curves for nonlinear channel II in AWGN environment
(a) ($a_1 = 1$, $a_2 = 0.1$, $a_3 = 0.05$)

(b) ($a_1 = a_2 = a_3 = 1$)

If the nonlinearity content of the channel is increased by taking all the nonlinearity coefficients as equal to 1, the performance is expected to deteriorate. This is proved by the learning curves for the MLP-DFE and the Lattice-based MLP-DFE (Fig. 5.6(b)). Here, the difference in the convergence time and the MSE has increased between the two configurations, showing the strength of the Lattice filter for the equalization of nonlinear channels. The MLP-DFE converges in about 1700 iterations to a MSE value of -10 dB, while the decorrelation of the input data causes the MSE to drop to around –15 dB which is attained in about 1500 iterations, thus a gain of 4 dB in MSE and 200 iterations.

When the linear channel 2 is used in place of the FIR filter in the model of nonlinear channel, the performance is expected to degrade due to the increase in eigen-value spread. Fig. 5.7 shows the learning curves for this channel. Part (a) corresponds to the nonlinearity of (a_1 = 1, a_2 = 0.1, a_3 = 0.05) and part (b) corresponds to the nonlinearity of (a_1=a_2= a_3=1).

There is a clear degradation in the convergence time for the MLP-DFE and it could not converge even after 2000 iterations. On the other hand, the MLP-DFE with pre-orthogonalization converges to some extent to a MSE value of –27 dB which is 2.5 dB lower than the MSE attained by the MLP-DFE after 2000 iterations. The performance improvement increases for the case of the highly nonlinear channel, where we find a MSE difference of 4 dB after 2000 iterations.

5.2.2.1 Comparison of (9,3,1) and (3,2,1) structures

Similar to the linear channels, we investigated the comparative loss due to the reduction in size of the MLP structure. The trend remains the same for the case of both of the nonlinear channels. The learning curves for the case of nonlinearity with coefficients (a_1=1, a_2=0.1, a_3=0.05) for the two nonlinear channels at SNR = 10 dB, are shown in Fig. 5.8 and 5.9. As seen from the results, the MSE is unaffected by the decrease in the size, but there is an increase in the convergence time of the respective equalizer configurations. The loss in the convergence rate is found to be less for the case of pre-orthogonalization as compared to the conventional MLP-DFE. The MLP-DFE suffers a loss of about 200 iterations, while the lattice-based counterpart has a loss of about 100 iterations.

5.2.3 Bit Error Rate Performance

Bit error rates for different equalizer configurations are given in Fig. 5.10 - 5.13. As expected, the gradient DFE does not perform at all, while the linear MLP equalizer is still performing to some extent, suggesting that the MLP based equalizer is a powerful tool for the nonlinear channels. This is because the MLP equalizer whether in linear configuration or in the form of decision feedback form has the capability to draw nonlinear decision regions. This capability is not present in the gradient DFE equalizers so it does not work well in this situation. This has also been shown in [40] for larger delay lines to equalize the nonlinear channels, which verifies that the linear equalizer is inherently incapable of nonlinear distortions regardless of equalizer length.

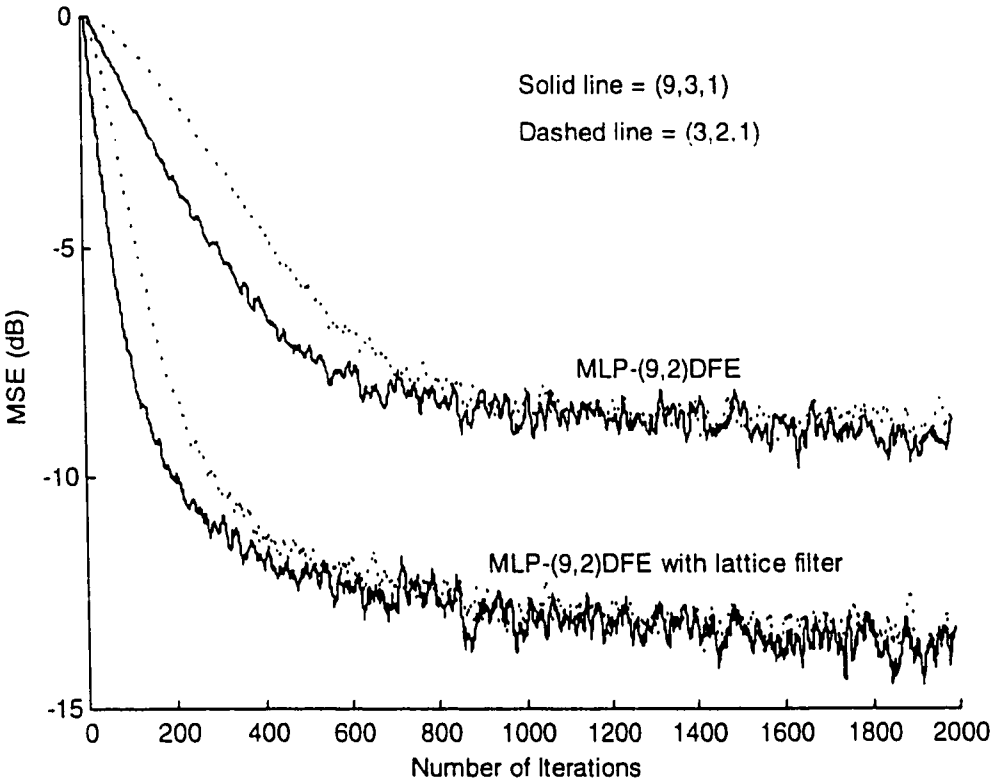


Figure 5.8: Comparison of (9,3,1) and (3,2,1) MLP structure sizes for channel I ($a_1 = 1$, $a_2 = 0.1$, $a_3 = 0.05$) in AWGN environment.

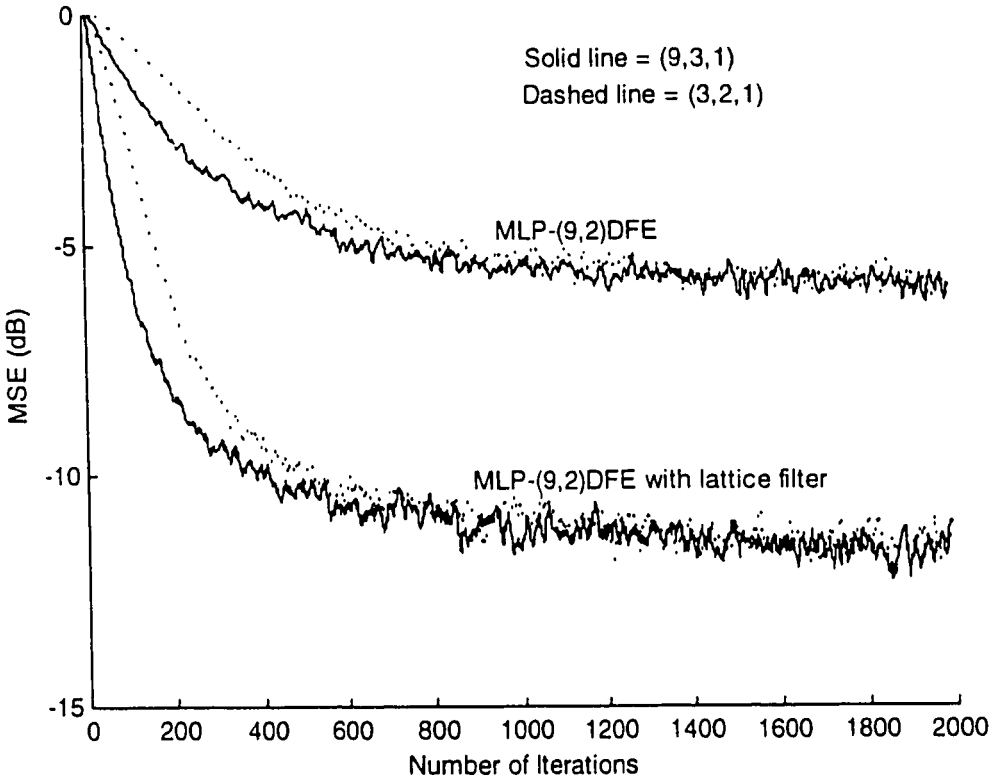


Figure 5.9: Comparison of (9,3,1) and (3,2,1) MLP structure sizes for channel II ($a_1 = 1$, $a_2 = 0.1$, $a_3 = 0.05$) in AWGN environment.

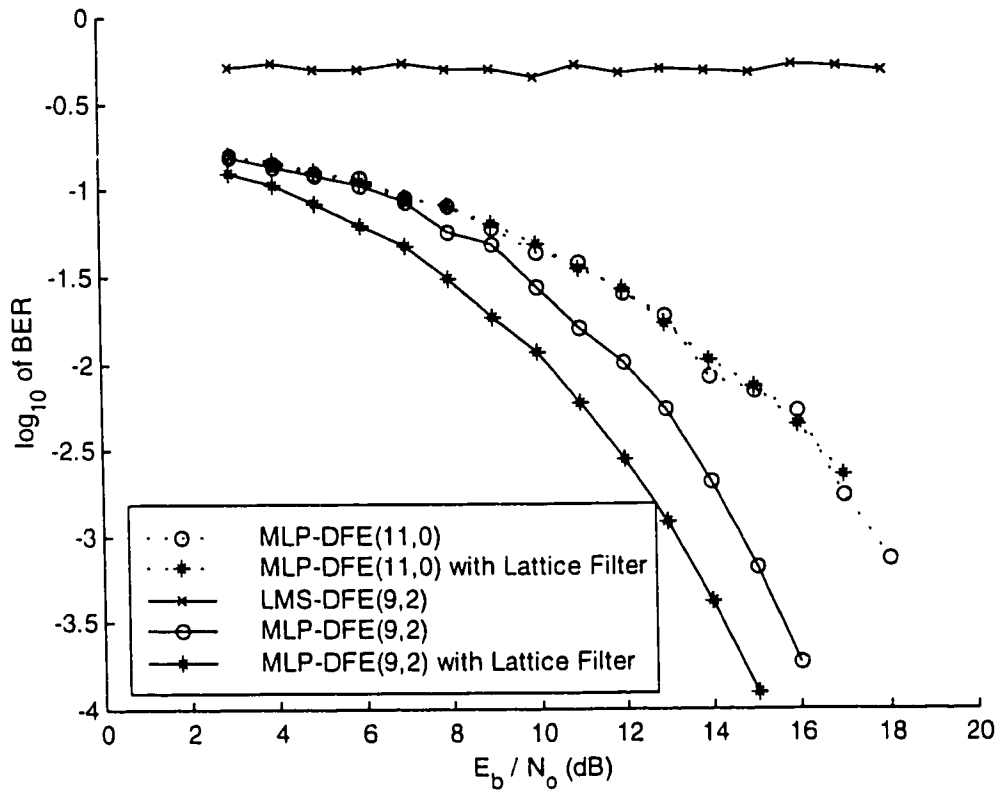


Figure 5.10: BER performance in Channel I ($a_1 = 1$, $a_2 = 0.1$, $a_3 = 0.05$) in AWGN environment.

Solid line represents Feedback Equalizer

Dashed line represents Linear Equalizer

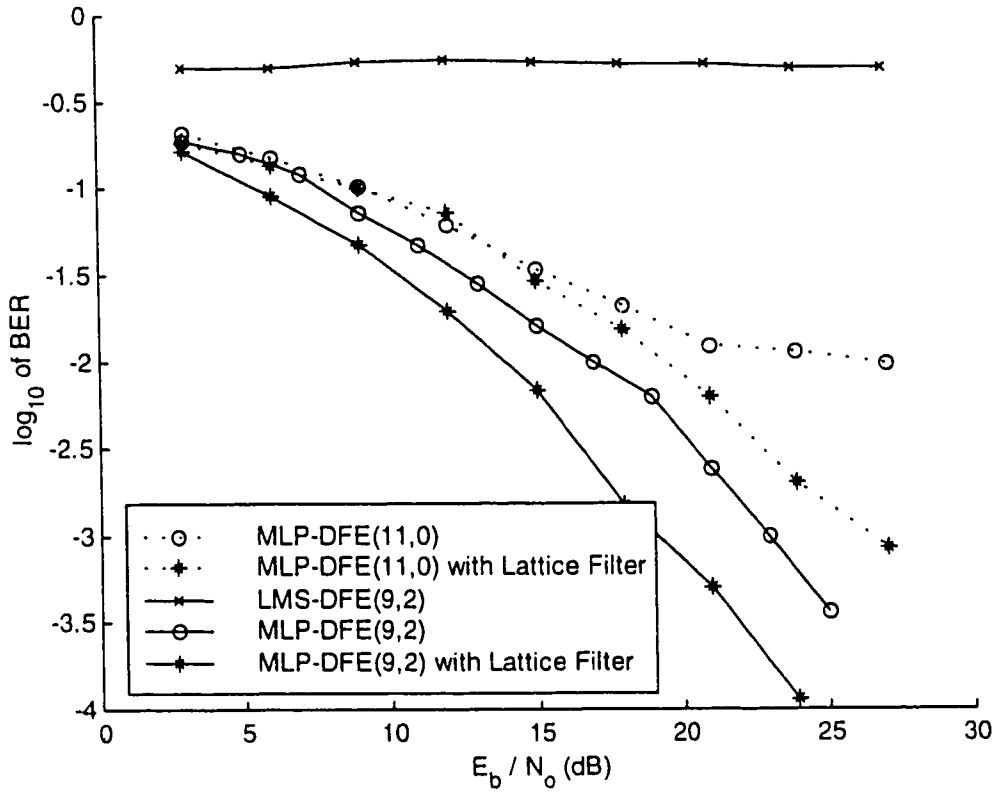


Figure 5.11: BER performance in Channel I ($a_1 = a_2 = a_3 = 1$)

in AWGN environment.

Solid line represents Feedback Equalizer

Dashed line represents Linear Equalizer

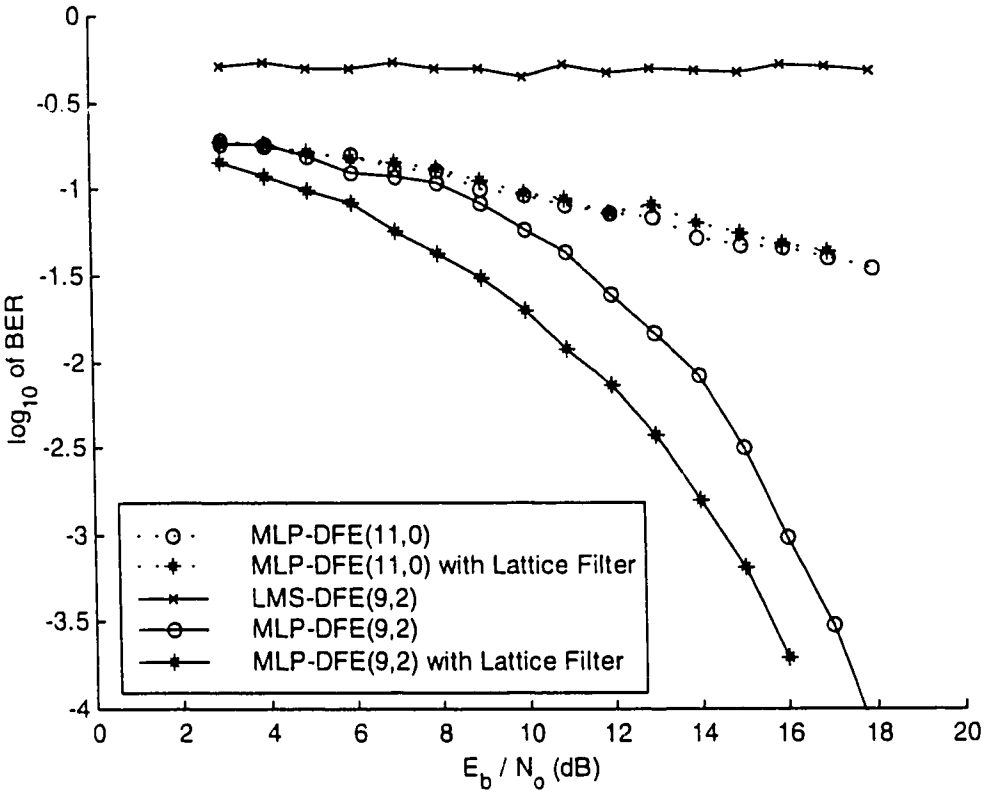


Figure 5.12: BER performance in Channel II ($a_1 = 1$, $a_2 = 0.1$, $a_3 = 0.05$) in AWGN environment.

Solid line represents Feedback Equalizer

Dashed line represents Linear Equalizer

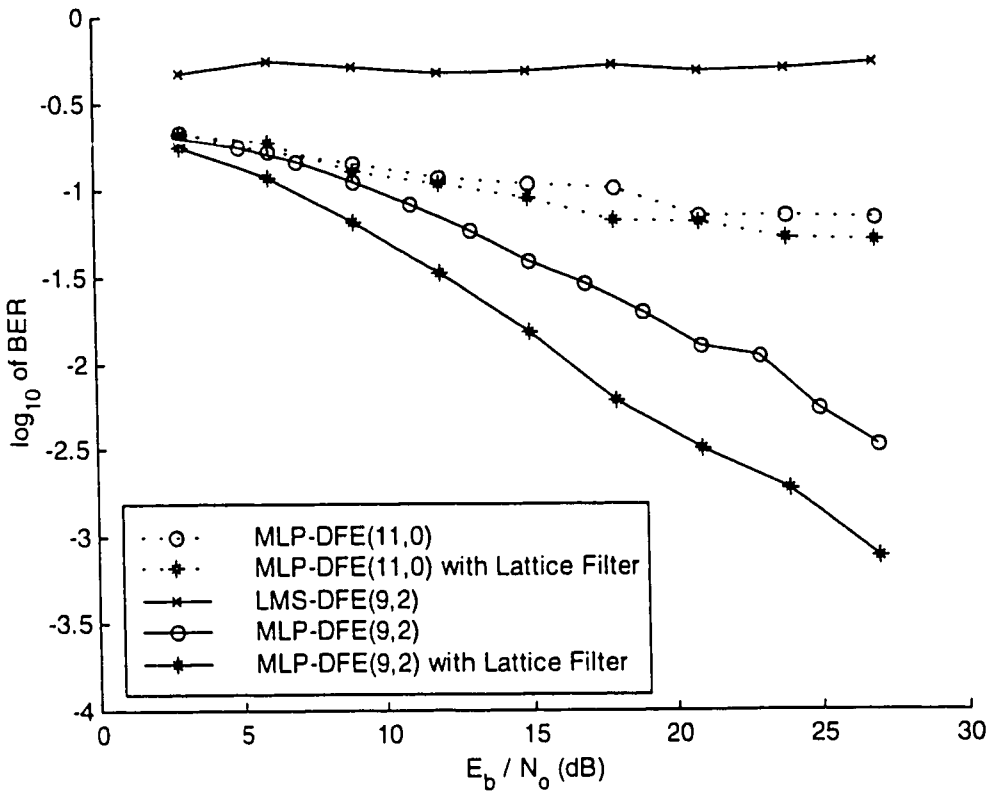


Figure 5.13: BER performance in Channel II ($a_1 = a_2 = a_3 = 1$) in AWGN environment.

Solid line represents Feedback Equalizer

Dashed line represents Linear Equalizer

Fig 5.10 shows that the linear MLP either with or without pre-orthogonalization has the same BER performance in the nonlinear channels, suggesting that the linear MLP is itself sufficient for the equalization of the nonlinear channels and the strength of the lattice filter does not appear because what can be done at most by the linear equalizer is being done by the MLP. But when the MLP-DFE is used the effect of decorrelation comes in the scene and we get around a gain of 2 dB at BER of 10^{-3}.

The same trend continues for the most severe nonlinearity of ($a_1=a_2=a_3=1$), except that the performance has now deteriorated due to the increase in the degree of nonlinearity. The difference in the performance of the two configurations has also increased to 3 dB at BER of 10^{-3}. The Linear version of the two configurations perform almost the same for the first type of nonlinearity but the lattice-based MLP starts outperforming the linear MLP as the SNR increases beyond 20 dB. Up to the SNR of 15 dB both equalizers perform the same but as the SNR increases, the MLP equalizer saturates at a BER value of 10^{-2}. but the curve of lattice MLP continues to decrease and is only 4 dB inferior to the MLP-DFE at BER of 10^{-3}.

The BER curves for the channel 2 with the two different degrees of non-linearities are given in Fig. 5.12 and 5.13.

5.2.4 Effect of Colored Noise

Due to the severe nature of channel, the effect of the linear and nonlinear distortion is high. So that the equalizers in colored noise perform not much different from white noise environment. This fact is demonstrated by the learning and BER curves of the equalizer configurations in both of the channels for the two different degree nonlinearities. The learning curves are given for SNR = 20 dB and the equalizers use

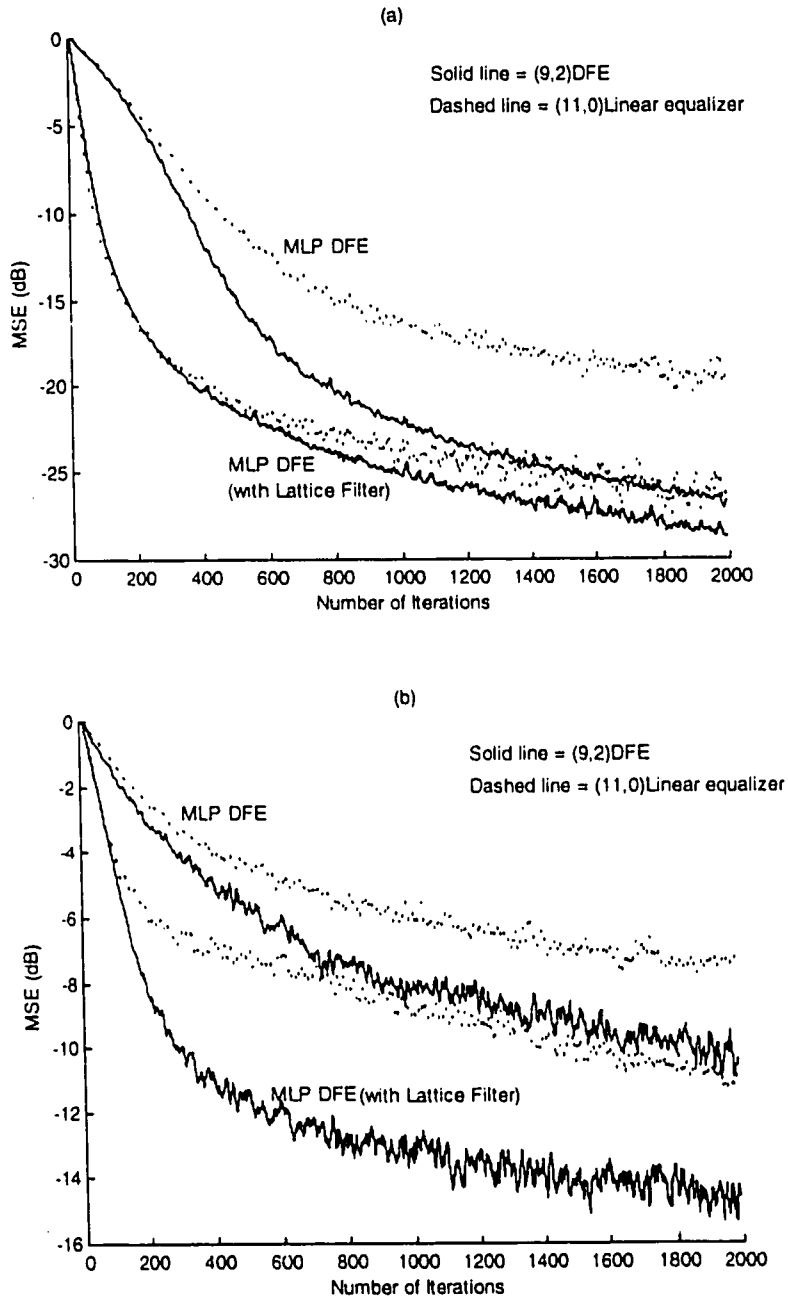


Figure 5.14: Learning Curves for nonlinear channel I (ACGN)
(a) ($a_1 = 1$, $a_2 = 0.1$, $a_3 = 0.05$)

(b) ($a_1 = a_2 = a_3 = 1$)

Solid line represents Feedback Equalizer

Dashed line represents Linear Equalizer

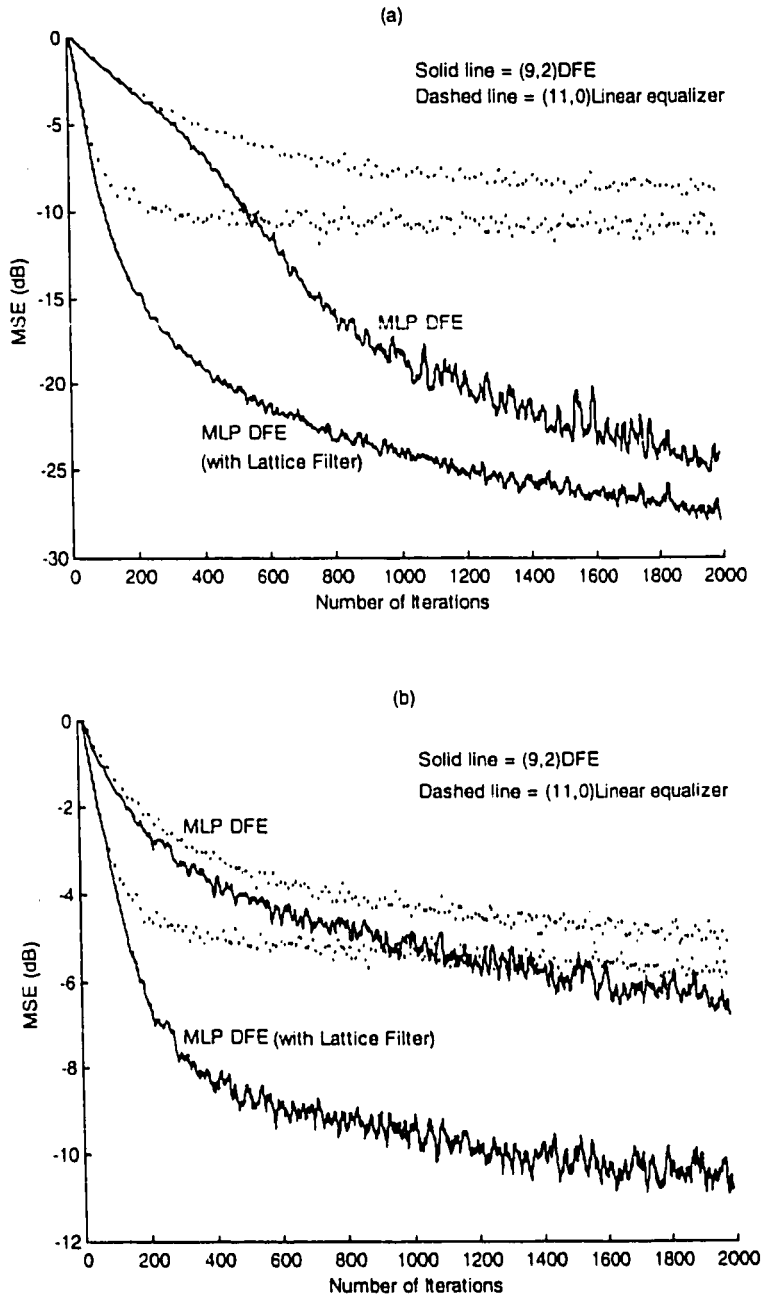


Figure 5.15: Learning Curves for nonlinear channel II (ACGN)
(a) ($a_1 = 1$, $a_2 = 0.1$, $a_3 = 0.05$)

(b) ($a_1 = a_2 = a_3 = 1$)

Solid line represents Feedback Equalizer

Dashed line represents Linear Equalizer

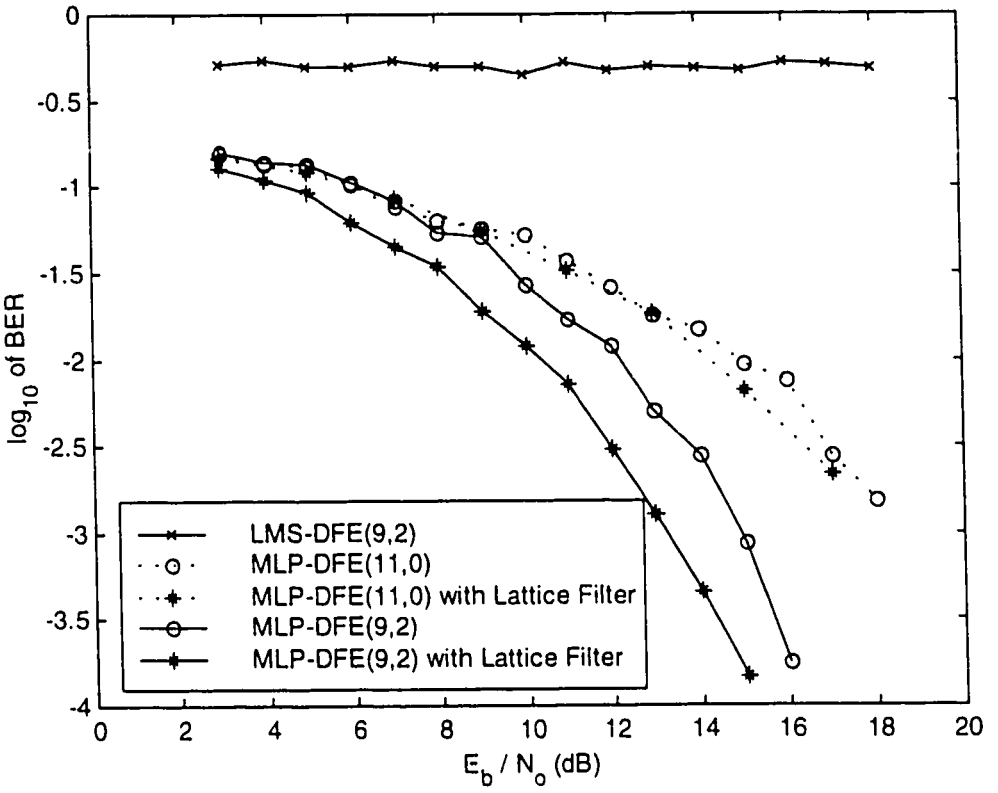


Figure 5.16: BER performance in Channel I ($a_1 = 1$, $a_2 = 0.1$, $a_3 = 0.05$) in ACGN environment.

Solid line represents Feedback Equalizer

Dashed line represents Linear Equalizer

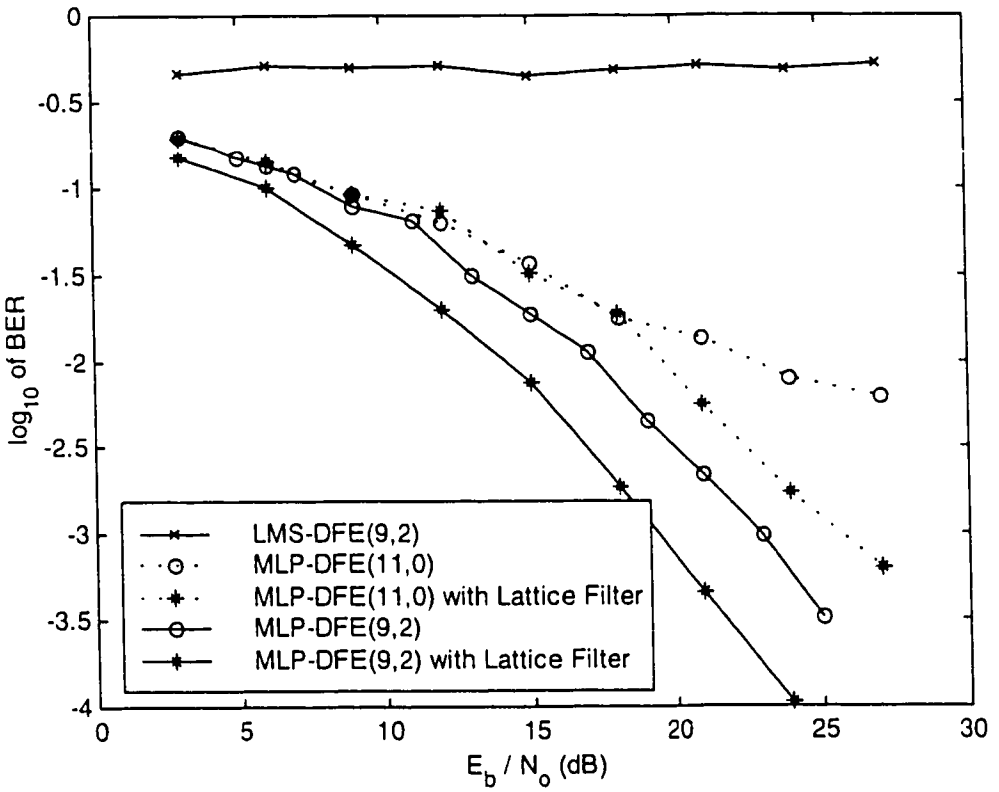


Figure 5.17: BER performance in Channel I ($a_1 = a_2 = a_3 = 1$)

in ACGN environment.

Solid line represents Feedback Equalizer

Dashed line represents Linear Equalizer

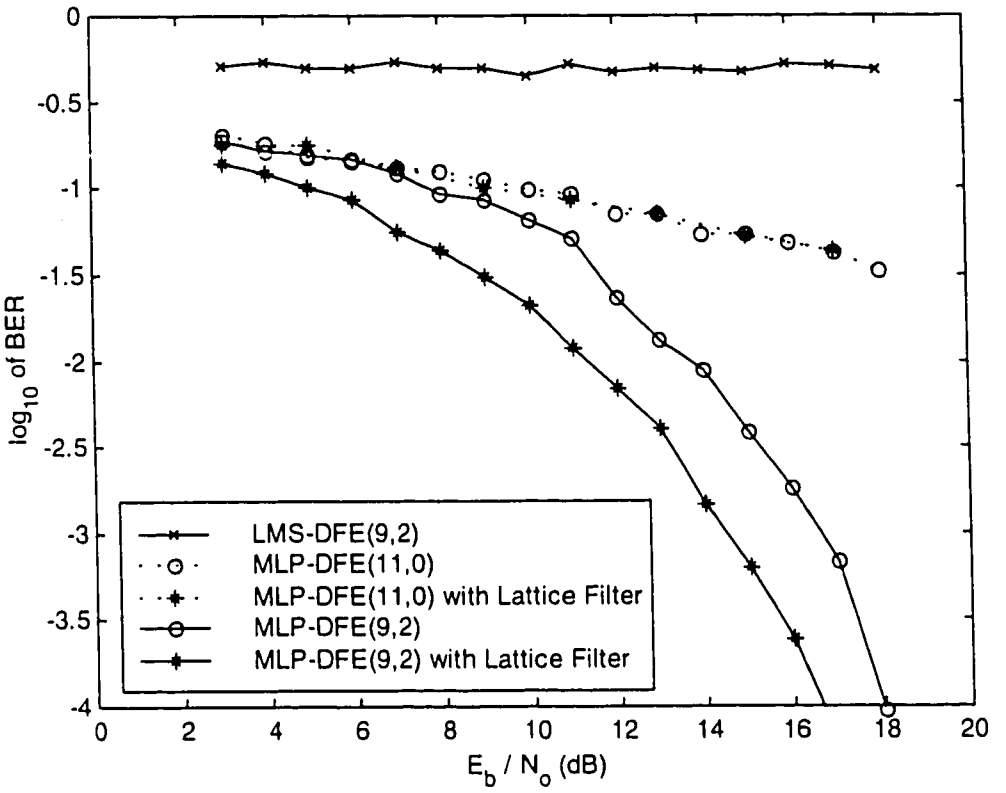


Figure 5.18: BER performance in Channel II ($a_1 = 1$, $a_2 = 0.1$, $a_3 = 0.05$) in ACGN environment.

Solid line represents Feedback Equalizer

Dashed line represents Linear Equalizer

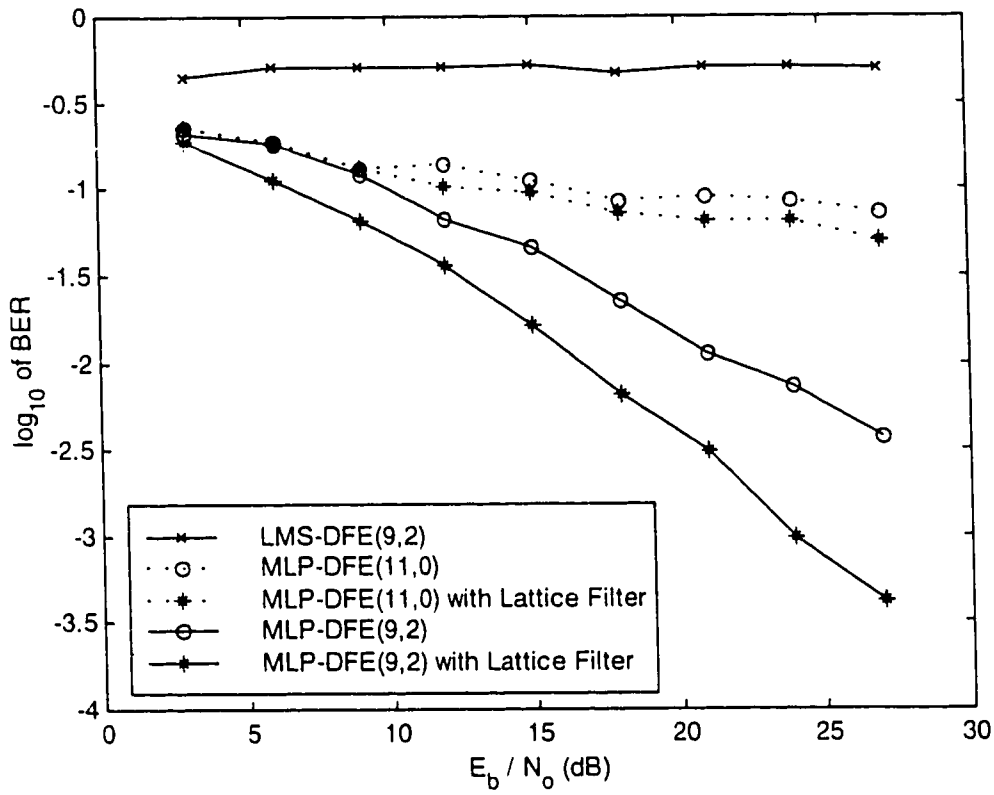


Figure 5.19: BER performance in Channel II ($a_1 = a_2 = a_3 = 1$) in ACGN environment.

Solid line represents Feedback Equalizer

Dashed line represents Linear Equalizer

the same configurations as that in AWGN case. Both the convergence time and the MSE in this case are almost the same as that in AWGN environment.

Similarly, the BER curves verify the results by giving the same error rate performance. Thus, for nonlinear channels the effect of distortion is dominant as compared to the additive noise.

5.3 Time Varying Channel

The time-variant fading channel was used to evaluate the capability of the equalizer to track the changes in the time varying dispersive channel. The discrete time channel model for time-variant fading channel is described by the following transfer function

$H(z) = a_0(t) + a_1(t)z^{-1} + a_2(t)z^{-2}$

Where $a_0(t)$, $a_1(t)$, and $a_2(t)$ are the time varying coefficients of the channel impulse response and are generated on the digital computer by passing white Gaussian noise through a low-pass filter of a specified bandwidth [29].

If we assume that we have a nominal 3 kHz HF channel, the signaling rate is 2400 symbols/s, and the low-pass filter is a two-pole butterworth filter. Then the 3-dB bandwidth of the LPF can be used as a parameter to control the rate of variation of the channel impulse response.

The curves of the first 30000 tap values changing with time for different LPF bandwidths are depicted in Fig. 5.20. Due to the varying nature of the channel, the learning curves are not used as a performance criterion rather, the bit error rate performance is the only way to analyze the system performance.

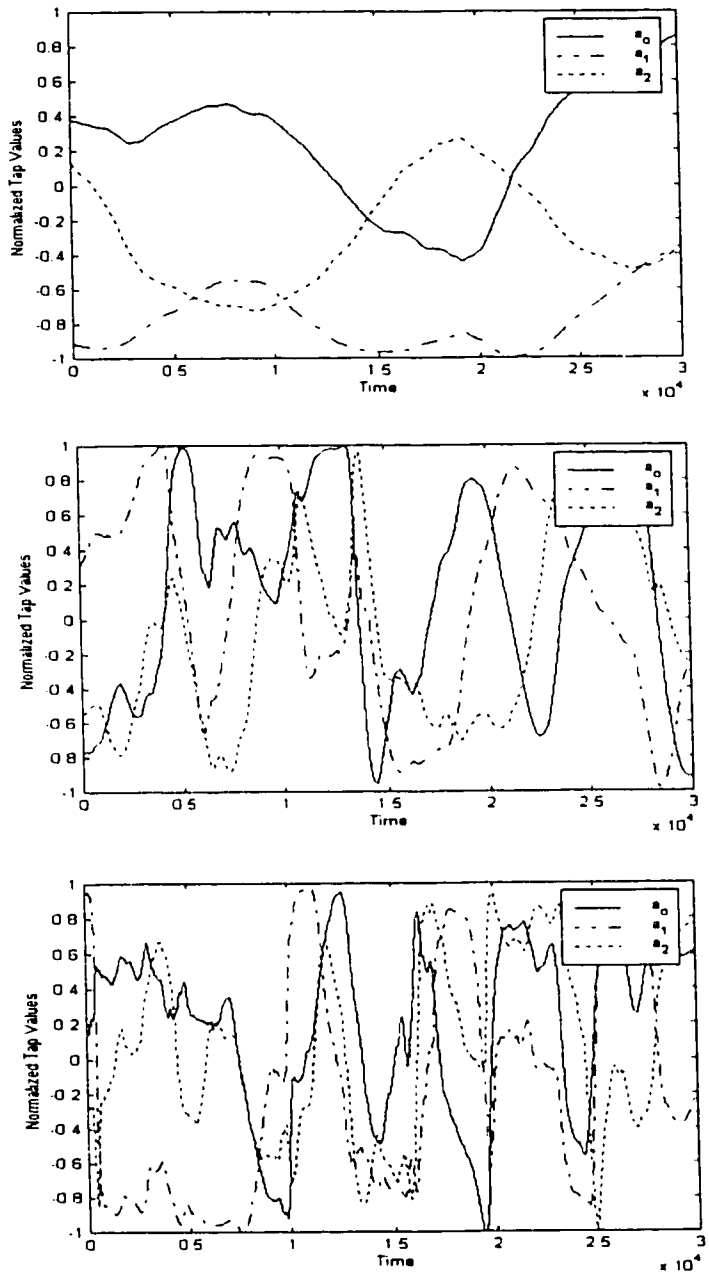


Figure 5.20: Tap values of time-varying channel

(a) (LPF BW = 0.1)

(b) (LPF BW = 0.5)

(c) (LPF BW = 1.0)

5.4 Performance Analysis

Fig. 5.21, 5.22, and 5.23 show the BER performance of the equalizers for the low pass filter bandwidths of 0.1, 0.5 and 1.0, respectively. As evident from the results, the performance of lattice-based MLP is far better than the conventional MLP equalizer, either in linear or feedback form.

As noted from the results, comparatively better performance of the gradient equalizers than their MLP counter parts. For the linear and nonlinear static channels, we found that the MLP equalizers give better performance over the gradient equalizers, but in time-varying channels, the neural equalizer is inferior to the gradient equalizer. As we saw in chapter 4, the convergence time of the MLP equalizer is greater than that of the gradient equalizer. The Lattice-based equalizer performs better due to the fast convergence characteristics of the lattice filter coefficients and the tracking characteristics of the RLS algorithm.

From Fig. 5.21, the lattice-based MLP-DFE clearly performs the best of all configurations by achieving a BER value of 10^{-4} at SNR = 14.5 dB. The BER of MLP-DFE decreases monotonically up to the SNR of 15 dB and after that saturates at around 3×10^{-3}. The linear version of MLP equalizer performs almost the same and saturates to the same level as that of MLP-DFE. The gradient-based DFE is better in performance in both linear and feedback form than the MLP equalizer up to the SNR of 20 dB but saturates to around the same BER value as that of MLP equalizers. The linear Lattice-based MLP is not only better than the other linear equalizers but also gives the comparative results to the other two DFE's configurations.

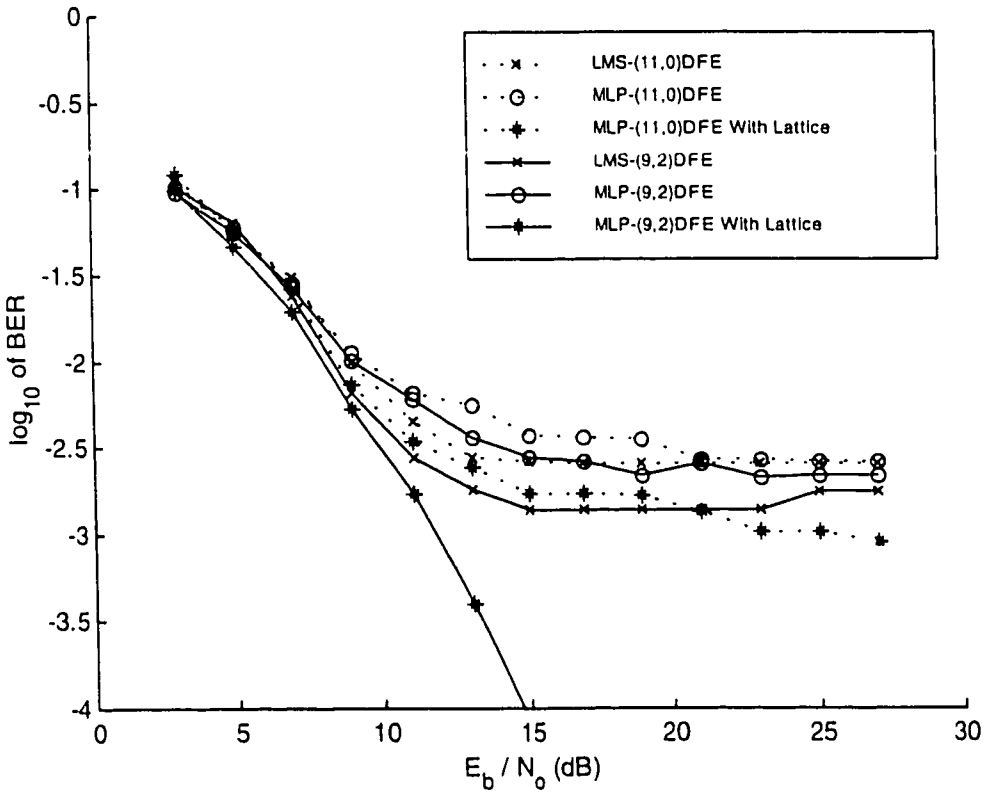


Figure 5.21: BER performance for time-variant channel 5

in AWGN environment (LPF BW = 0.1)

Solid line represents Feedback Equalizer

Dashed line represents Linear Equalizer

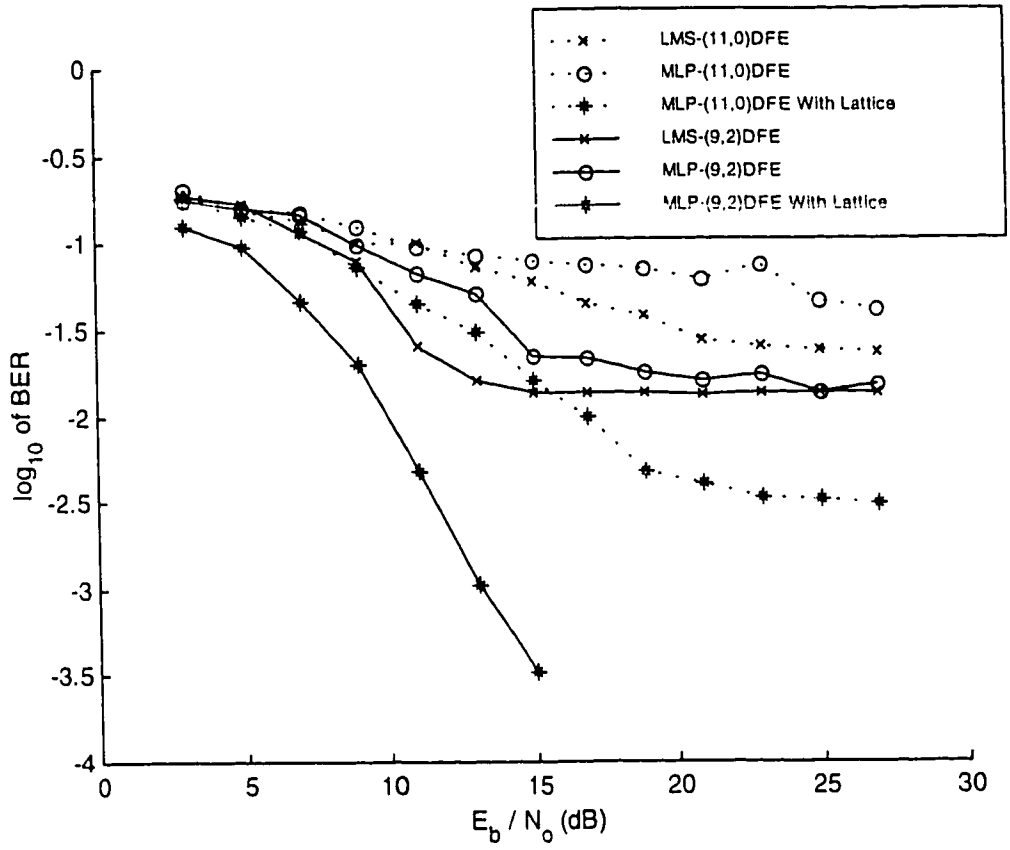


Figure 5.22: BER performance for time-variant channel 5

in AWGN environment (LPF BW = 0.5)

Solid line represents Feedback Equalizer

Dashed line represents Linear Equalizer

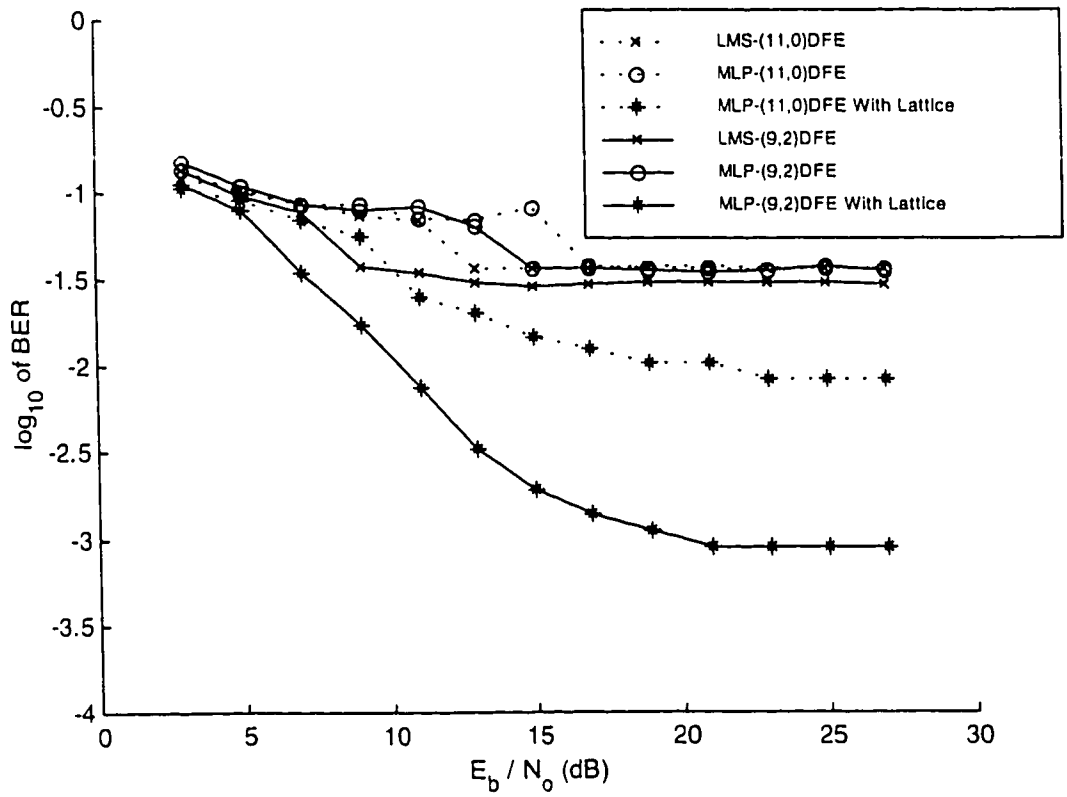


Figure 5.23: BER performance for time-variant channel 5

in AWGN environment (LPF BW = 1.0)

Solid line represents Feedback Equalizer

Dashed line represents Linear Equalizer

When, the rate of variation of the channel impulse response is increased by increasing the bandwidth of the low pass filter, there is a deterioration in the performance of all configurations. The difference between the lattice-based MLP and other two DFE configurations has increased. The Lattice MLP still has the shape of waterfall even up to the BER of $3*10^{-4}$, but the gradient DFE and MLP-DFE saturates after the SNR of 15 dB to the same value of BER = 10^{-2}. The rate of decrease of BER for gradient DFE is higher from 5 to 15 dB than the MLP-DFE but both saturates finally to the same value of BER. The linear lattice-DFE again surpasses the performance of both MLP and gradient DFE.

Fig. 5.23 gives the BER curves for the case of LPF bandwidth of 1.0. As seen from Fig. 5.20, the rate of change of this channel response is the faster than the previous two. Even the lattice-MLP-DFE saturates at BER of 10^{-3} after SNR of about 15 dB. There is a difference of approximately one order of magnitude between the linear lattice MLP and Lattice-MLP-DFE after the saturation. The linear and feedback configurations of both the MLP and gradient based equalizers saturates at almost the same BER value of about $3*10^{-2}$.

The performance curves were also attained for the colored noise environment and it was found that the results in both noise environments are nearly the same. Thus, from this we gather that in time varying channels, the effect of time variations is dominant on the effect of the nature of the noise.

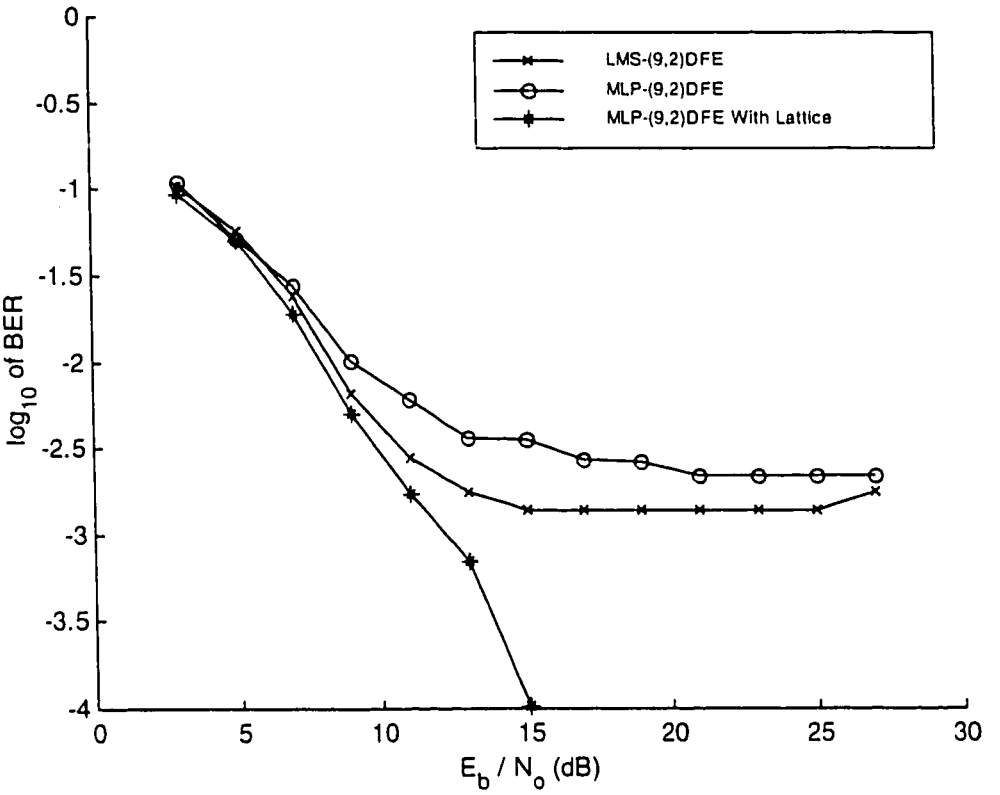


Figure 5.24: BER performance for time-variant channel 5 in ACGN environment (LPF BW = 0.1)

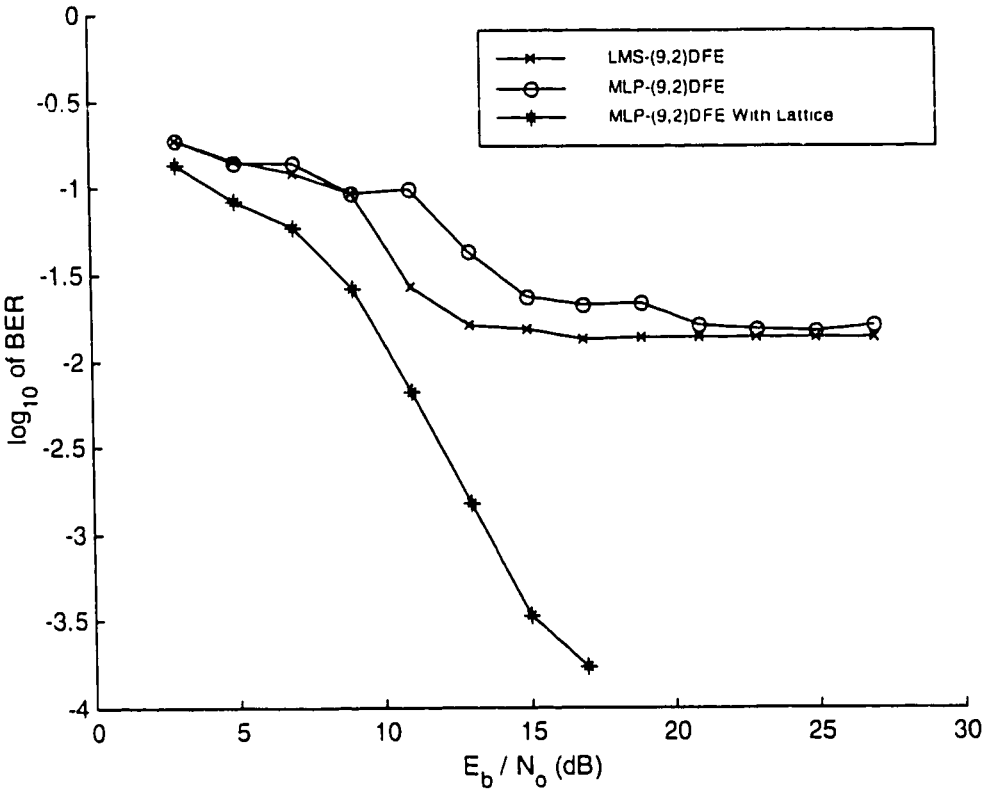


Figure 5.25: BER performance for time-variant channel 5 in ACGN environment (LPF BW = 0.5)

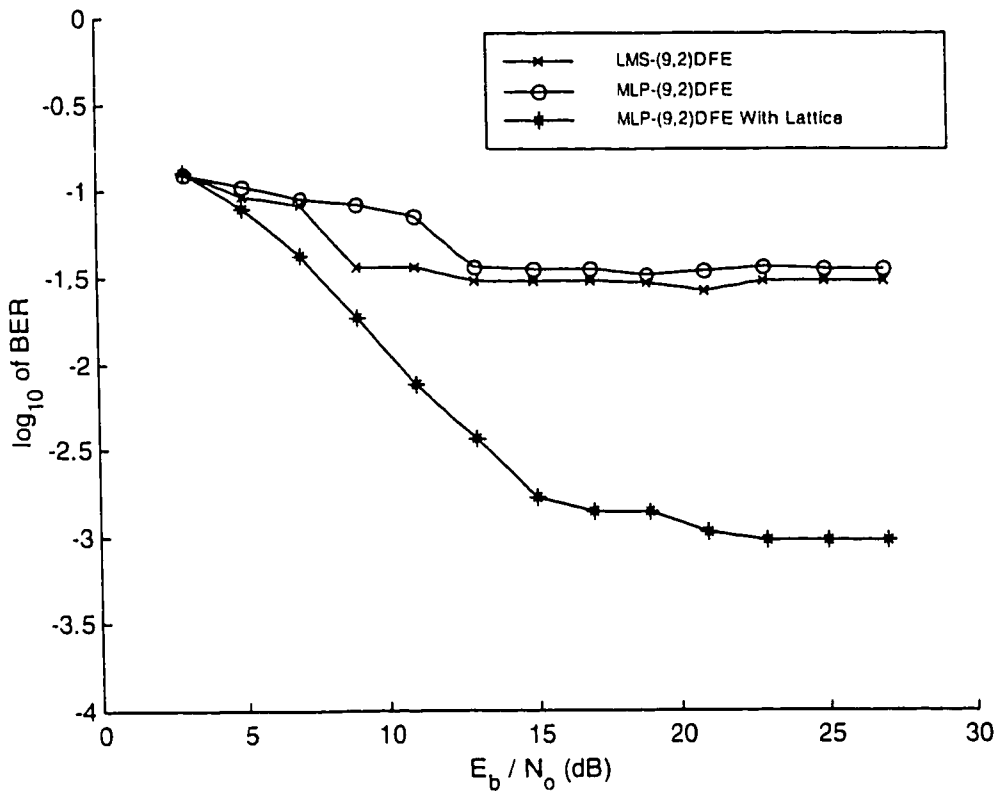


Figure 5.26: BER performance for time-variant channel 5

in ACGN environment (LPF BW = 1.0)

Chapter 6

CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK

6.1 Discussion

In this thesis we have studied an approach of pre-orthogonalization of the received data samples which are input to the MLP-based decision feedback equalizer. The scheme was found to give better results for all channel conditions as compared to the conventional MLP equalizers. But this improvement is at the cost of extra computational complexity. So there is trade-off between complexity and the performance.

Different approaches were adopted to evaluate the performance of the systems in various equalizer configurations and channel conditions. The comparative performance of the equalizers was shown in the form of constellation (Scatter) and eye diagrams to have a qualitative perception of the systems. The MLP-DFE with and without pre-orthogonalization was shown to perform better than LMS-DFE in terms of the convergence of the scattered points to their mean positions in scatter diagrams and the

opening of the eye in the eye diagrams. It was shown that the decorrelation of the equalizer input data help increase the noise margins for the MLP-DFE.

Learning curves and BER performance curves were also used to justify the qualitative results by giving the statistical results of the different equalizer schemes. Linear equalizers were also simulated for the sake of reference. The LMS-DFE was used as the benchmark to evaluate the performance of the equalizers in terms of learning rate and BER.

The computer simulations have illustrated that pre-orthogonalizing the received samples not only improves the convergence rate of the MLP-based DFE but it also decreases the Mean-Squared Error (MSE) causing a better BER performance. The improvement is specially more pronounced in heavily distorting channels having large eigen-value spread of the channel correlation matrix. The decorrelation of the equalizer input data also makes the MLP-DFE insensitive to the eigen-value spread of the channel correlation matrix. The proposed scheme was also found to be more robust, in terms of the convergence rate, to the size of the perceptron structure as shown by the comparison of two structure sizes of the MLP.

When the system was evaluated for correlated Gaussian noise and compared with white Gaussian noise, it was found that for the particular process by which we generated the colored noise causes a decrease in both of the convergence rate and the convergence time of the MLP. The less MSE values give a better BER performance in ACGN as compared to the AWGN environment.

For nonlinear channels, the LMS linear equalizers and LMS-DFE simply fail to perform. Whereas, MLP linear equalizer, though not performing very well, performs

better than LMS-DFE and the MLP-DFE performs even better than that. The MLP linear equalizer with pre-orthogonalization has almost comparable performance in terms of BER as compared to MLP linear equalizer because, the MLP-based equalizer is itself a powerful tool to equalize the nonlinear channel distortion. When the decorrelation is applied to MLP-DFE, it outperforms the performance of simple MLP-DFE. The performance in ACGN for nonlinear channels was found to be nearly the same as that in AWGN environment.

Time varying channels with different rates of variation were used to investigate the ability of the equalizer to track time variations the channel response. The Lattice-based MLP-DFE outperformed both the LMS-DFE and MLP-DFE in terms of BER performance. The LMS-DFE was found to be performing better than the MLP-DFE, suggesting that the MLP-DFE is not a good choice for the equalization of the time varying channels. The performance in ACGN environment was found to be indifferent as compared to the AWGN environment for the case of time varying channels.

6.2 Conclusions

The results of our study can be summarized briefly as follows:

- The use of Lattice Filter for pre-orthogonalization of the MLP equalizer input data results in improvements in terms of the Convergence Rate, MSE, and BER.
- The convergence rate of the Lattice-based MLP-DFE is insensitive of the Eigen-value Ratio of the channel correlation matrix.

- The decorrelation of the equalizer input data also makes the MLP-DFE less sensitive to the size of the MLP.
- The proposed Lattice-based MLP has better tracking capabilities as compared to the conventional MLP gradient-based DFE.
- For linear channels, the performance is better in ACGN environment as compared to the AWGN case.
- For static nonlinear and time varying channels, the performance in both AWGN and ACGN environments is almost the same.

6.3 Suggestions for Further Work

Some suggestions to extend the work are given bellow;

1. We have used one type of the neural equalizer namely the Multi-Layer Perceptron for all of our simulations. Other neural-based equalizers, e.g. Radial Basis Function (RBF) and Recurrent Neural Networks (RNN), also exist. The proposed scheme can also be tested for those configurations in order to generalize the conclusions to all the neural network based equalizers.
2. The scheme should also be tested for the case of frequency selective fading environment.
3. All the equalizers in our simulations are T-spaced equalizers. The performance of the proposed scheme can be analyzed for fractionally spaced DFE as well.
4. In our simulations, the bipolar data is used. The system is to be analyzed for some large-constellation modulation schemes like QAM .

APPENDIX A

A.1 The Least Squares Lattice DFE algorithm

The least squares lattice DFE algorithm [27] is given bellow. In this algorithm, bold face characters represents matrices or vectors. For the derivation of this algorithm, the interested reader is referred to [27], [28].

A.1.1 Initialization

$b_o(t) = f_o(t) = y(t)$

$r_o^f(t) = r_o^b(t) = w\, r_o^f(t-1) + |y(t)|^2$

$e_o(t) = u(t),\ g_o(t) = 0,\ \alpha_o(t) = 1$

$r_m^f(0) = r_m^b(0) = \delta,\ k_m(0) = 0,\ k_m^g(0) = 0 \qquad (m = N_1 - N_2, \ldots, N_1)$

$k_M^b(0) = 0 \qquad (M = N_1 - N_2)$

$\mathbf{r}_m^f(0) = \mathbf{r}_m^b(0) = \delta\, \mathbf{I},\ \mathbf{k}_m(0) = \mathbf{0},\ \mathbf{k}_m^g(0) = \mathbf{0} \quad (m = N_1 - N_2, \ldots, N_1)$

A.1.2 Scalar Stages { ($0 < m \leq N_1 - N_2$) unless otherwise specified }

$f_m(t) = f_{m-1}(t) - k_m(t-1)\, b_{m-1}(t-1) / r_{m-1}^b(t-2)$

$b_m(t) = b_{m-1}(t-1) - k_m(t-1)\, f_{m-1}(t) / r_{m-1}^f(t-1)$

$k_m(t) = w\, k_m(t-1) + \alpha_{m-1}(t-1)\, f_{m-1}(t)\, b_{m-1}(t-1)$

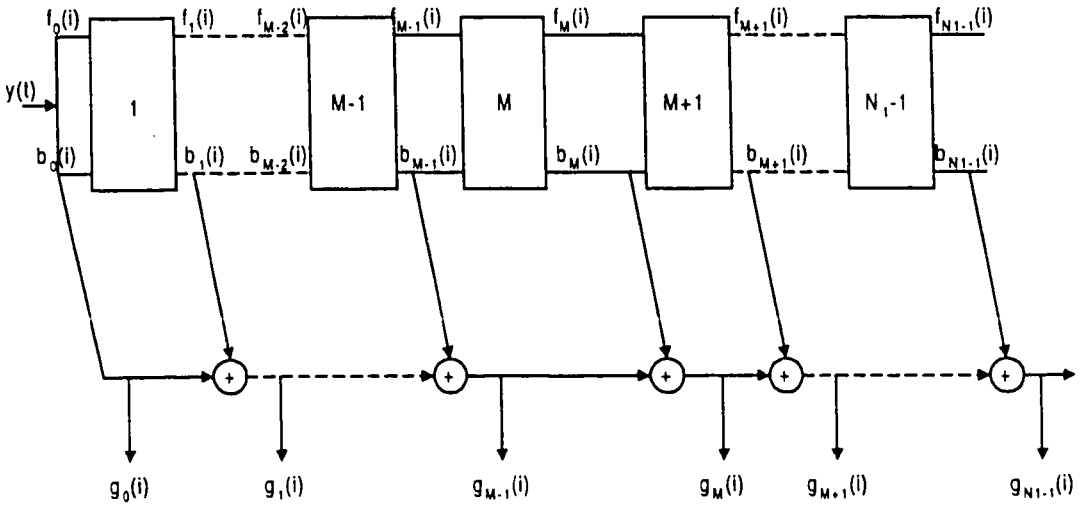


Fig. A.1: Structure of Feedback lattice Filter

$r_m^f(t) = r_{m-1}^f(t) - | k_m(t)|^2 / r_{m-1}^b(t-1)$ $\quad (m < N_1 - N_2)$

$r_m^b(t) = r_{m-1}^b(t-1) - | k_m(t)|^2 / r_{m-1}^f(t-1)$ $\quad (m < N_1 - N_2)$

$g_m(t) = g_{m-1}(t) + k_m^g(t-1) + b_{m-1}(t) / r_{m-1}^b(t-1)$

$\alpha_m(t) = \alpha_{m-1}(t) - |b_{m-1}(t)\ \alpha_{m-1}(t)|^2 / r_{m-1}^b(t)$

$e_m(t) = u(t) - g_m(t)$ $\quad (0 \le m \le N_1 - N_2)$

$k_m^g(t) = w\ k_m^g(t-1) + \alpha_{m-1}(t)\ e_{m-1(t)}\ b_{m-1}(t)$

A.1.3 Transitional Stage { $M = N_1 - N_2$ }

$b_M^*(t) = e_{M-1}(t-1) - k_M^b(t-1)\ f_{M-1}^f(t) / r_{M-1}^f(t-1)$

$k_M^b(T) = w\ k_M^b(t-1) + \alpha_{M-1}(t-1)\ f_{M-1}(t-1)\ e_{M-1}(t-1)$

$f_M(t) = [\ f_M(t)\ \ e_M(t-1)\ ]^T$

$\mathbf{b}_M(t) = [\ b_M(t)\ b_M^*(t)\ ]^T$

$\mathbf{r}_M^f(t) = w\ \mathbf{r}_M^f(t-1) + \alpha_M(t-1)\ \mathbf{f}_M^2(t)$

$\mathbf{r}_M^b(t) = w\ \mathbf{r}_M^b(t-1) + \alpha_M(t)\ \mathbf{b}_M^2(t)$

A.1.4 Two Dimensional Stages {($N_1 - N_2 < m < N_1$) unless otherwise specified}

$\mathbf{f}_m(t) = \mathbf{f}_{m-1}(t) - \mathbf{k}_m(t-1)\ \mathbf{r}_{m-1}^{\cdot b}(t-2)\ \mathbf{b}_{m-1}(t-1)$

$\mathbf{b}_m(t) = \mathbf{b}_{m-1}(t-1) - \mathbf{k}_m(t-1)\ \mathbf{r}_{m-1}^{\cdot f}(t-1)\ \mathbf{f}_{m-1}(t)$

$\mathbf{k}_m(t) = w\ \mathbf{k}_m(t-1) + \alpha_{m-1}(t-1)\ \mathbf{b}_{m-1}(t-1)\ \mathbf{f}_{m-1}(t)$

$\mathbf{r}_m^f(t) = w\ \mathbf{r}_{m-1}^f(t-1) - \alpha_m(t-1)\ \mathbf{f}_m^2(t)$

$\mathbf{r}_m^b(t) = w\ \mathbf{r}_{m-1}^b(t-1) - \alpha_m(t)\ \mathbf{b}_m^2(t)$

$g_m(t) = g_{m-1}(t) + \mathbf{k}_m^g(t-1) + \mathbf{r}_{m-1}^{\cdot b}(t-1)\ \mathbf{b}_{m-1}(t)$ $\qquad (\ m \le N_1\ )$

$\alpha_m(t) = \alpha_{m-1}(t) - \alpha_{m-1}(t)^2\ \mathbf{b}_{m-1(t)}\ \mathbf{r}_{m-1}^{\cdot b}(t)\ \mathbf{b}_{m-1(t)}$

$e_m(t) = u(t) - g_m(t)$

$\mathbf{k}_m^g(t) = w\ \mathbf{k}_m^g(t-1) + \alpha_{m-1}(t)\ \mathbf{b}_{m-1}(t)\ e_{m-1(t)}$ $\qquad (\ m \le N_1\ )$

All the variables in the above algorithm can take complex values as well, so the above algorithm can also be used for M-ary PSK and QAM signals.

References

[1] Anand, N. S. C., Anderson J. B., "An adaptive decision feedback equalizer for digital cellular radio", in *IEEE ICASSP* 1990, pp. 662-667.

[2] Bunpei, B. and Miyake, S., "Capabilities of three-layered perceptrons", IEEE IJCNN, San Diego, Vol. .1., 1988

[3] Bettayeb, M., Zerguine A., "Fractionally-spaced equalization using the lattice structure", *Computers Elect. Engg.* Vol. 21 No. 1, 1995, pp. 47-55.

[4] Biglieri, E., Gersho, A., Gitlin, R. D., and Lim, T. L., "Adaptive cancellation of nonlinear intersymbol interference for voice-band data transmission", *IEEE Journal on Selected Areas in Communications*, vol. SAC-2, No. 5, Sept. 1984, pp. 765-777.

[5] Boud, S. and Chua, L. O., "Fading memory and the problem of approximating nonlinear operators with Voltera Series", *IEEE Trans. Circuits and Systems*, vol. CAS-32, No. 11, Nov. 1985.

[6] Chang, P. R., Bor-Chin Wang, "Adaptive decision feedback equalization for digital satellite channels using multilayer neural networks", *IEEE Journal on selected areas in communications*, vol. 13, No. 2, Feb. 1995.

[7] Chen, S., Gibson, G. J., Cowan, C. F. N., Grant, P. M., "Adaptive equalization of finite nonlinear channels using multilayer perceptrons", *Signal Processing*, vol. 20, 1990, pp. 107-119.

[8] Chen, S. Mulgrew, B., Grant, P. M.,, "A clustering technique for digital communication channel equalization using radial basis function networks", *IEEE Transactions on Neural Networks*, vol. 4, No. 4, July 1993.

[9] Colin, F. N., Cowan, C. F. N., "Communications equalization using non-linear adaptive networks", *IEEE ICASSP*, 1991, pp. 199-202.

[10] Falconer D. D., "Adaptive equalization of channel nonlinearities in QAM data transmission systems", *Bell Sys. Tech. J.*, vol. 57, No. 7, Sept. 1978, pp. 2589-2611,

[11] George, D. A., R. R. Bowen, and J.R. Storey, "An adaptive decision feedback equalizer", *IEEE Transactions on Communications*, vol. COM-19, No. 3, June 1971, pp. 281-293.

[12] Gibson, G. J., Siu, S., and Cowan, C.F.N., "Multilayer perceptron structures applied to adaptive equalizers for data communications", in *IEEE proceedings ICASSP* Glasgow, Scotland, May 1989. pp. 1183-1186.

[13] Gibson, G. J., Siu, S., and Cowan, C.F.N., "The application of nonlinear structures to the reconstruction of binary signals", *IEEE Transactions on Signal Processing*, Vol. 39, No. 8, Aug., 1991.

[14] Gitlin, R. D., Mazo, J. E., and Taylor, M. G., "On the design of gradient algorithms for digitally implemented adjustment filters", *IEEE Trans. Circuit Theory, CT-20,* pp. 125-136, 1973.

[15] Gitlin, R. D., Magee, F. R., "Self-orthogonalizing adaptive equalization algorithms", *IEEE Transaction on Communications*, vol. COM 25 No. 7, July 1977.

[16] Godard, D., "Channel equalization using a Kalman filter for fast data transmission", *IBM Journal of Research & Development*, May 1974, pp. 267-273.

[17] Griffths, L. J., "A continuously adaptive filter implemented as a Lattice structure", in *Proc. IEEE Int. Conference on Acoustics Speech and Signal Processing*, Hartford , CT, May 1977, pp. 683-686.

[18] Griffths, L. J., "An adaptive Lattice structure for noise canceling applications", in *Proc. IEEE Int. Conference on Acoustics Speech and Signal Processing*, Tulsa, OK, April 1978, pp. 87-90.

[19] Haykin, S., "Introduction to Adaptive Filters", New York, Macmillan Pub. Co. 1984

[20] Haykin S., "Adaptive Filter Theory", Prentice-Hall, Englewood Cliffs, New Jersey, 1986.

[21] Haykin S., "Neural Networks: A Comprehensive Foundation", Macmillan College Publishing Company, 1994.

[22] Itakura, F. and Saito, S., "Digital filtering techniques for speech analysis and synthesis", in *Proc. 7th Int. Conf. Acoust.*, Budapest, 1971, paper 25-C1, pp. 261-264.

[23] Jeruchim M. C., et al, "Simulation of Communication Systems", Plenum Press, New York and London, 1992.

[24] Kechriotis, G., Zervas, E., and Manolakos, E. S., "Using recurrent neural networks for adaptive communication channel equalization", *IEEE Transactions on Neural Networks*, vol. 5, No. 2, March 1994.

[25] Khalid A. Al-Mashouq, Reed, I. S. , "The use of neural nets to combine equalization with decoding for severe intersymbol interference channels", *IEEE Transactions on Neural Networks*, vol. 5, No. 6, Nov. 1994, pp. 982-988.

[26] Kumar P. C., P. Saratchandran, and N. Sundararajan, "Nonlinear channel equalization using minimal radial basis function neural networks", in *IEEE ICASSP*, Seattle, Washington, USA, vol. 6, May 12-15,1998, pp. 3373-3376,.

[27] Ling, F. and Proakis, J. G., "A generalized least square lattice algorithm and its application to decision feedback equalization", in *Proc. ICASSP*, Paris, France, May 1982, pp. 1764-1772.

[28] Ling, F. and Proakis, J. G., "Lattice decision feedback equalizers and their application to fading dispersive channels", in *Proc. Int. Conf. Comm.*, Boston, MA, June 1983, pp. C8.2.1-C8.2.5.

[29] Ling, F., Proakis, J. G., "Nonlinear learning characteristic of least squares adaptive algorithms", in *IEEE ICASSP*, San Diego, CA March 1984, pp. 7.3.1-7.3.4

[30] Ling, F., Proakis, J. G., "Adaptive lattice Decision feedback equalizers: Their performance and application to time variant multipath channels", *IEEE Trans. Comm.* vol. COM-33, No. 4, April 1985, pp. 348-356.

[31] Lippmann, R. P., "An Introduction to Computing with Neural Nets", *IEEE ASSP Magazine*, April, 1987.

[32] Lucky, R. W., "Automatic equalization for digital communication", *B.S.T.J.*, 44, No. 4, part 1, April 1965, pp. 547-588.

[33] Makhoul, J., "Linear prediction: A tutorial review", in *Proc. IEEE*, vol. 63, April 1975, pp. 561-580.

[34] Makhoul, J., "Stable and efficient lattice methods for linear prediction", *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-25 No.5, Oct. 1977, pp. 423-428.

[35] Makhoul, J., "A class of all-zero lattice digital filters: properties and applications", *IEEE Transactions on Acoustics, Speech, and Signal processing*, vol. ASSP-26, No. 4, August 1978, pp. 304-314.

[36] Ong, S., You, C., Choi, S., Hong, D., "A decision feedback recurrent neural equalizer as an infinite impulse response filter", *IEEE Transactions on Signal Processing*, Vol. 45, No. 11, Nov. 1997.

[37] Parisi, R., Elio D. Di Claudio, G. Orlandi, B. D. Rao, "Fast adaptive digital equalization by recurrent neural networks", *IEEE Transactions on Signal Processing*, Vol. 45, No. 11, Nov. 1997.

[38] Peng, M., Nikias, C. L. and Proakis, J. G., "Adaptive equalization with neural networks: New multi-layer perceptron structures and their evaluation", in *IEEE ICASSP*, 1992, pp. II-301-II-304.

[39] Peng M., Nikias, C. L. and Proakis, J. G., "Adaptive equalization for PAM and QAM signals with Neural Networks", in *IEEE ICASSP*, 1991, pp. 496-500.

[40] Peng M., "Neural Networks Applications in linear and nonlinear channel equalization", Ph.D. dissertation, Northeastern University Boston Massachusetts, June 1994.

[41] Proakis, J. G., "Advances in Equalization for Intersymbol Interference", in Advances in Communication Systems, vol. 4, A. J. Viterbi (Ed.), Academic Press, New York, 1975.

[42] Proakis, J. G., "Digital Communications", McGraw-Hill, 1989.

[43] Qureshi, S.U.H., "Adaptive Equalization", in *Proc. IEEE*, vol. 73, No. 9, 1985, pp. 1349-1387.

[44] Rosenblatt, F., "Principles of Neurodynamics", Spartan Books, Washington, 1962.

[45] Rumelhart, D. E., Hinton, G. E., Williams, R. J., "Learning internal representations by error propagation", *Parallel Distributed Processing*, vol. 1, chap. 8, D.E. Rumelhart and J.L. McClelland, (Ed), Cambridge, MA, MIT Press, 1986.

[46] Satorius and S. T. Alexander, "Channel equalization using adaptive lattice algorithms", *IEEE Trans. Comm.*, vol. COM 27, June 1979, pp. 899-905.

[47] Shannon, C. E., "A mathematical theory of communication channels", *Bell Sys. Tech. J.*, vol. 27, July, pp. 379-423,

[48] Shensa, M. J., "A least square lattice decision feedback equalizer", in *Proc. IEEE Int. Conf. Comm.*, Seattle, WA, June 1980.

[49] Shirish A., "Improvements in detectors based upon colored noise", IEEE Transactions on Magnetics, vol. 34, No. 1, January 1998, pp. 94-97.

[50] Simpson, P. K., "Artificial Neural Systems", Pergamon Press, Inc. 1990.

[51] Siu, S., Gibson, G. J., Cowan, C. F. N., "Decision feedback equalization using neural network structures and performance comparison with standard architecture", in *IEE Proceedings*, vol. 137, No. 4, August 1990, pp. 221-225.

[52] Umari, M. H., "Effects of colored noise on the performance of LMS adaptive equalizers and predictors", *in Proceedings IEEE symposium on Circuit Systems*, Vol. 5, 1992. pp. 2565-2568.

[53] Wiener N., "Nonlinear Problems in Random Theory", John Wiley & Sons, New York, 1958.

[54] Wieland, A. Leighton, R., "Geometric analysis of neural network capabilities", *1st International conference on Neural Networks*, June 1987, pp. 385-393.

[55] Werbos, P., "Beyond regression: New tools for prediction and analysis in the behavioral sciences", Ph.D. Dissertation, Harvard U., Committee on Applied Mathematics. Nov. 1974.

[56] Widrow, B., Hoff, M. E., Jr., "Adaptive switching circuits", in *IRE Wescon Conference*, Rec., 1960, part 4, pp. 96-104

[57] Widrow, B., Hoff, M. E., Jr., "Stationary and non stationary characteristics of LMS adaptive filter", in *Proc. IEEE*, Aug. 1976, vol. 64, pp. 1156-1162.

[58] Widrow B., Winter, R. G., Baxter, R. A., "Layered neural nets pattern recognition", *IEEE Transaction ASSP*, July 1988, 36, pp 1096-1099

[59] Yingwei, L., Saratchandran, P., and Sundararajan, N., "Adaptive nonlinear system identification using minimal radial basis function neural networks", in *IEEE ICASSP*, vol. 6, 1996, pp. 3521-3524.

[60] Yingwei, L., Saratchandran, P., and Sundararajan, N., "Identification of time varying nonlinear systems using minimal radial basis function neural networks", in *IEE Proc. Control Theory Appl.*, vol. 144, No. 2, March 1997.

[61] Linde, L. P, "The performance of an adaptive least squares lattice decision feedback equalizer on a fading multipath HF channel", *in Proceedings IEEE,* 1988, pp. 86-92.

[62] Ozgunel, S., Kayran A.N., Panayirci, E., "Non-linear channel equalization using multichannel adaptive lattice algorithms", *IEEE International Symposium on Circuits & Systems*, June 1991, pp. 2826-22829.

[63] Schetzen, M., "The Voltera and Wiener Theories of Nonlinear Systems", John Wiley & Sons, 1980.

[64] Huang, W., Rappaport, T. S., "A comparative study of two adaptive equalizers for mobile radio", in Proceedings IEEE, 1991, pp. 765769.

[65] Eweda, E., "Analysis and design of a signed regressor LMS algorithm for stationary and nonstationary adaptive filtering with correlated Gaussian data", IEEE Transactions on Circuits and Systems, vol. 37, No. 11, November 1990.