Artificial Neural Network Based Channel Equalization

A thesis submitted in fulfilment of the requirements for the degree of Master of Technology (Research)

in

Electronics & Communication Engineering

Under the guidance of

Prof. S. K. Patra

By

Devi Rain Guha

Department of Electronics and Communication Engineering
National Institute of Technology, Rourkela, INDIA

Department of Electronics & Communication Engineering
NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA
ORISSA, INDIA – 769 008

CERTIFICATE

This is to certify that the thesis titled "***Artificial Neural Network Based Channel Equalization***", submitted to the National Institute of Technology, Rourkela by **Devi Rani Guha**, Roll No. **60609004** for the award of the degree of ***Master of Technology (Research)*** in Electronics & Communication Engineering, is a bonafide record of research work carried out by her under my supervision and guidance.

The candidate has fulfilled all the prescribed requirements.

The thesis is based on candidate's own work and has not been submitted elsewhere for a degree / diploma.

In my opinion, the thesis is of required standard for the award of a Master of Technology (Research) degree in Electronics & Communication Engineering.

To the best of my knowledge, she bears a good moral character and decent behaviour.

Dr. S. K. Patra
(Professor)
Department of ECE
NATIONAL INSTITUTE OF TECHNOLOGY
Rourkela-769008 (INDIA)
Email: skpatra@nitrkl.ac.in

ACKNOWLEDGEMENT

I take the opportunity to express my reverence to my supervisor, Prof. S. K. Patra, for his guidance, inspiration and innovative technical discussions during the course of this work. He is not only a great teacher with deep vision but also a very kind person. His trust and support inspired me for taking right decisions and I am glad to work with him.

I express my respect to all master scrutiny committee members and my teachers Prof. J. K. Satapathy, Prof. K. K. Mahapatra, Prof. G. S. Rath, Prof. G. Pand, Prof. S. Meher and Prof. Susmita Das, for their contribution in my studies and research work. They have been great sources of inspiration to me and I thank them from the bottom of my heart.

I would like to thank all the faculty members and staffs of the Department of Electronics and Communication Engineering, N.I.T. Rourkela for their inspiration, cooperation and provided me all official and laboratory facilities in various ways for the completion of this thesis.

I would also like to thank all my friends for their cooperation and encouragement for the completion of this thesis.

My indebted respect and thanks to my loving parents (Sri. Gopal Chandra Guha and Smt. Bela Rani Guha) and elder sisters (Ujjala didi and Karabi didi) for their love, sacrifice, inspiration, suggestions and support. They are my first teachers after I came to this world and have set great examples for me about how to live, study and work. Also, my special thanks to little friends Dev, Puja, Surjo and chotku as they are the key to my steps towards success.

Last but not the least; I take this opportunity to express my regards and obligation to my late grand-father and mother, for their blessings.

Devi Rani Guha

ABSTRACT

The field of digital data communications has experienced an explosive growth in the last three decade with the growth of internet technologies, high speed and efficient data transmission over communication channel has gained significant importance. The rate of data transmissions over a communication system is limited due to the effects of linear and nonlinear distortion. Linear distortions occure in from of inter-symbol interference (ISI), co-channel interference (CCI) and adjacent channel interference (ACI) in the presence of additive white Gaussian noise. Nonlinear distortions are caused due to the subsystems like amplifiers, modulator and demodulator along with nature of the medium. Some times burst noise occurs in communication system. Different equalization techniques are used to mitigate these effects.

Adaptive channel equalizers are used in digital communication systems. The equalizer located at the receiver removes the effects of ISI, CCI, burst noise interference and attempts to recover the transmitted symbols. It has been seen that linear equalizers show poor performance, where as nonlinear equalizer provide superior performance.

Artificial neural network based multi layer perceptron (MLP) based equalizers have been used for equalization in the last two decade. The equalizer is a feed-forward network consists of one or more hidden nodes between its input and output layers and is trained by popular error based back propagation (BP) algorithm. However this algorithm suffers from slow convergence rate, depending on the size of network. It has been seen that an optimal equalizer based on maximum a-posterior probability (MAP) criterion can be implemented using Radial basis function (RBF) network. In a RBF equalizer, centres are fixed using K-mean clustering and weights are trained using LMS algorithm. RBF equalizer can mitigate ISI interference effectively providing minimum BER plot. But when the input order is increased the number of centre of the network increases and makes the network more complicated. A RBF network, to mitigate the effects of CCI is very complex with large number of centres.

To overcome computational complexity issues, a single neuron based chebyshev neural network (ChNN) and functional link ANN (FLANN) have been proposed. These neural networks are single layer network in which the original input pattern is expanded to a higher dimensional space using nonlinear functions and have capability to provide arbitrarily complex decision regions.

More recently, a rank based statistics approach known as Wilcoxon learning method has been proposed for signal processing application. The Wilcoxon learning algorithm has been applied to neural networks like Wilcoxon Multilayer Perceptron Neural Network (WMLPNN), Wilcoxon Generalized Radial Basis Function Network (WGRBF). The Wilcoxon approach provides promising methodology for many machine learning problems. This motivated us to introduce these networks in the field of channel equalization application. In this thesis we have used WMLPNN and WGRBF network to mitigate ISI, CCI and burst noise interference. It is observed that the equalizers trained with Wilcoxon learning algorithm offers improved performance in terms of convergence characteristic and bit error rate performance in comparison to gradient based training for MLP and RBF. Extensive simulation studies have been carried out to validate the proposed technique. The performance of Wilcoxon networks is better then linear equalizers trained with LMS and RLS algorithm and RBF equalizer in the case of burst noise and CCI mitigations.

ACRONYMS & ABBREVIATIONS

ADI	Adjacent channel Interference
ANN	Artificial Neural Network
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BFO	Bacterial Foraging Optimization
BNI	Burst Noise Interference
BP	Back Propagation
CCI	Co channel Interference
ChNN	Chebyshev Neural Network
DCR	Digital Cellular Radio
DFE	Decision Feedback Equalizer
DSP	Digital Signal Processing
FIR	Finite Impulse Response
FLANN	Functional Link Artificial Neural Network
GA	Genetic Algorithm
GD	Gradient Descent
IIR	Infinite Impulse Response
ISI	Inter Symbol Interference
LAN	Local Area Network
LMS	Least Mean Square
MAP	Maximum a-posteriori Probability
MMSE	Minimum Mean Square Error
MLP	Multi Layer Perceptron
MLSE	Maximum Likelihood Sequence Estimator
MSE	Mean Square Error
PSO	Particle Swarm Optimization
PDF	Probability Density Function
RBF	Radial Basis Function

RLS	Recursive Least Square
SNR	Signal to Noise Ratio
SI	System Identification
SV	Support Vector
TE	Transversal Equalizer
WNN	Wilcoxon neural network
WMLPN	Wilcoxon Multi Layer Perceptron Network
WGRBFN	Wilcoxon Generalized Radial Basis Function Network

LIST OF FIGURES

Chapter 2

Chapter 3

LIST OF TABLES

CONTENTS

Chapter 1

Introduction

Chapter 1

Introduction

The advent of high speed global communication ranks as one of the important developments of human civilization from the second half of twentieth century to till date. This was only feasible with the introduction of digital communication systems. Today there is a need for high speed and efficient data transmission over the communication channels. It is a challenging task for the engineers and scientists to provide a reliable communication service by utilizing the available resources effectively in-spite many factors that distort the signal. The main objective of the digital communication system is to transmit symbols with minimum errors. The high speed digital communication requires large bandwidth, which is not possible due to limited resources available.

This chapter is organised as follows. Following this introduction, section 1.1 describes the theme of the thesis. Section 1.2 describes the motivation of the work. Sections 1.3 provide a brief literature survey on equalisation in general and nonlinear equalisers in particular. At the end, section 1.4 presents the thesis layout.

1.1. Theme of the thesis

Digital communication systems are designed to transmit high speed data over communication channels. During this process the transmitted data is distorted, due to the effects of linear and nonlinear distortions. Linear distortion includes inter-symbol interference (ISI), co-channel interference (CCI) in the presence of additive noise [1, 2]. The non-ideal frequency response characteristic of the channel causes ISI, where as CCI occurs in cellular radio and dual-polarized microwave radio,for efficient utilization of the allocated channels bandwidth by reusing the frequencies in different cells.

Burst noise [3] is a high intensity noise which occures for short duration of time with fixed burst length means a series of finite-duration Gaussian noise pulses. Nonlinear distortions are caused due to the subsystem like amplifiers, modulator and demodulator along with nature of the medium. Compensating all these channel distortion calls for channel equalization techniques at the receiver side which aids reconstruct the transmitted symbols correctly.

Adaptive channel equalizers have played an important role in digital communication systems. Generally equalizer works like an inversed filter which is placed at the front end of the receiver. Its transfer function is inverse to the transfer function of the associated channel [4], is able to reduce the error causes between the desired and estimated signal. This is achieved through a process of training. During this period the transmitter transmits a fixed data sequence and the receiver has a copy of the same.

The main aim of the thesis is to develop and investigate novel artificial neural network equalizer [2], which can be trained with linear, nonlinear or evolutionary algorithms, so as to minimize the error caused in the desired signal.

In this thesis we consider linear gradient based algorithms like least-mean-square (LMS), recursive-least-square (RLS) to train the weights of the adaptive equalizer [1] and by iterative process minimize the mean square error. Generally these linear equalizers show poor performance than nonlinear equalizers. To overcome this problem artificial neural network equalizers are used. Artificial neural network (ANN) is a powerful tool in solving complex applications such as function approximation, pattern classification, nonlinear system identification and adaptive channel equalization [1, 5]. An ANN based multi layer perceptron (MLP) equalizer [6, 7] is a feed-forward network, consists of one or more layer of neural nodes with in input and output layers and is trained using popular error based back propagation (BP) algorithm. But it has a drawback of slow convergence. Another standard neural network structure that has been seen to provide optimal equalizer based on maximum a-posterior probability (MAP) criterion is based on radial basis function (RBF) network[8, 9]. The RBF network is a three layer standard simple structure. It provides optimal bit error rate performance similar to optimized Bayesian equalizer [10]. But one drawback in the RBF network is that if equalizer order increases, the number of centre of the network also increases and it makes the RBF network more complex.

Different methods have been proposed [11] to train ANN based equalizers. A new learning algorithm named Wilcoxon learning algorithm has been proposed recently. Wilcoxon learning is a rank based statistics approach used in linear and nonlinear learning regression problems and is usually robust against outliers. In this method, weights and parameters of the network are updated using simple rules based on gradient descent principle. This Wilcoxon learning algorithm can be used on different neural networks. These networks include Wilcoxon Neural Network (WNN), Wilcoxon Multilayer Perceptron Neural Network (WMLPNN), Wilcoxon Fuzzy Neural Network (WFNN), and Kernel-based Wilcoxon Regressor (KWR). The Wilcoxon approach provides a promising methodology for many machine learning problems. This has motivated us to use this technique for Channel Equalization. This has been not used before for channel equalization.

To overcome the problem of computational complexity a single neural network based nonlinear artificial neural network (ANN) equalizer named as Chebyshev neural network (ChNN) [12], functional link ANN (FLANN) [13, 14] is used. These neural networks are single layer network in which the original input pattern is expanded to a higher dimensional space using nonlinear functions and they have capacity to form an arbitrarily complex decision region by generating nonlinear decision boundaries. This enhanced space is then used for the channel equalization process. The advantage of ChNN and FLANN is that they provide superior performance in terms of convergence characteristic, computational complexity and bit error rate over a wide range of channel conditions. But ChNN have advantages over FLANN, that Chebyshev polynomials are computationally more efficient than FLANN trigonometric polynomials.

Evolutionary algorithms [15] have also been used to minimize the distortion of the communication system. Genetic Algorithm and Particle Swarm Optimization [16, 17] based approach are popular method to achieve adaptive channel equalization. Recently optimization techniques have been used to train the adaptive equalizer, named as Bacteria Foraging Optimization (BFO) technique [18]. The equalizers provide improved performance than linear equalizer and MLP equalizer in terms of convergence characteristic and bit error rate, but it has a drawback that computational complexity is more as compared to linear and nonlinear equalizers.

1.2. Motivation for work

The digital communication techniques can be attributed to the invention of the automatic linear adaptive equaliser in the late 1960’s [19]. From this modest start, adaptive equalisers have gone through many stages of development and refinement in the last 5 decade. Early equalisers were based on linear adaptive filter algorithms [20] with or without a decision feedback. Alternatively Maximum Likelihood Sequence Estimator (MLSE) [21] was implemented using the Viterbi [22] algorithm. Both forms of the equalisers provided two extremities in-terms of performance achieved and the computational cost involved. The linear adaptive equalisers are simple in structure and easy to train but they suffer from poor performance in severe conditions. On the other hand, the infinite memory MLSE provide good performance but at the cost of large computational complexity.

In mobile radio channels always changes and multipath causes time dispersion of the digital information is known as inter-symbol-interference, it makes too difficult to detect the actual information at the receiver. Mitigate this problem using adaptive linear equalizer but it needs large training data sequences for the equalizer and also shows poor performance.

Compensate the linear equalizers problems by using equalizers based on Maximum a-posterior probability (MAP) principle these were also called Bayesian equalizers [9]. These Bayesian equalizers techniques used like Artificial Neural Networks (ANN) [7], radial basis function (RBF) [8], recurrent network [23], Kalman filters, Fuzzy systems [24, 25] etc for nonlinear signal processing. RBF equalizer provides optimal bit error rate performance similar to optimized Bayesian equalizer. But one drawback in the RBF network is that if equalizer order increased, the centre of the network is also increased and its make the network complex and increases the conversation period.

To overcome this computational complexity problem, an efficient nonlinear artificial neural network equalizer structure for channel equalization is used named Chebyshev Neural Network (ChNN) [12], and Functional link ANN (FLANN) [13, 14] (descried as section 1.1) These novel single layer neural network provide superior performance in terms of computational complexity and bit error rate over a wide range of channel conditions. This motivated us to apply this ANN structures in the field of channel equalization to mitigate the ISI, CCI and burst noise interference in communication channels.

Evolutionary algorithms have been used to minimize the distortion of the communication system. The evolutionary principles have led scientists in the field of “Foraging Theory” to hypothesize that it is appropriate to model the activity of foraging as an optimization process like Bacterial Foraging Optimizations (BFO) [18], Ant-Colony Optimizations (ACO) [26] and Particle Swarm Optimization (PSO) [16, 17]. This optimization technique encourages us to use this algorithm in the channel equalization processes and compared its performance with ANN structure performance.

More recently, a rank based statistics approach known as Wilcoxon learning method [11] has been proposed for signals processing application to mitigate the linear and nonlinear learning problems. This Wilcoxon learning algorithm can be used on different neural networks. This motivated us to introduce this learning strategy in the field of Channel Equalization.

1.3. Background Literature Survey

Nyquist laid the foundation for digital communication over band limited analogue channels in 1928, with the enunciation of telegraph transmission theory. The research in channel equalisation started much later in 1960's and was centred on the basic theory and structure of zero forcing equalisers. The LMS algorithm by Widrow and Hoff in 1960 [19] paved the way for the development of adaptive filters used for equalisation. But it was Lucky [5] who used this algorithm in 1965 to design adaptive channel equalisers. With the popularisation of adaptive linear filters in the field of equalisation their limitations were also soon revealed. It was seen that the linear equaliser, in-spite of best training, could not provide acceptable performance for highly dispersive channels. This led to the investigation of other equalisation techniques beginning with the Maximum Likelihood Sequence Estimator (MLSE) equaliser [21] and its Viterbi implementation [22] in 1970's. In this field in 1970's and 1980's were the developments of fast convergence and/or computational efficient algorithms like the recursive least square (RLS) algorithm, Kalman filters. 1980's saw the beginning of development in the field of ANN [1]. The multi layer perceptron (MLP) based symbol-by-symbol equalisers was developed in 1990[33]. This brought new forms of equalisers that were computationally more efficient than MLSE and could provide superior performance compared to the conventional equalisers with adaptive filters. But it has a drawback of slow convergence rate, depending upon the number of nodes and layers. Another new implementation were done in symbol-by-symbol equalizers using the maximum a-posterior probability (MAP) principle these were also called

Bayesian equalizers [24]. These Bayesian equalizers have been approximated using nonlinear signal processing techniques like radial basis function (RBF) [8], recurrent network [23], Kalman filters [10], Fuzzy systems [24-25] etc.

During 1989 to 1995 some efficient nonlinear artificial neural network equalizer structure for channel equalization were proposed, those include Chebyshev Neural Network [12], Functional link ANN [13-14]. These neural networks are single layer network in which the original input pattern is expanded to a higher dimensional space using nonlinear functions thus providing an arbitrarily complex decision region by generating nonlinear decision boundaries. This enhanced space is then used for the channel equalization process. Both the networks provide good performance and comparatively low computational cost.

Evolutionary algorithms are also used to provide improved equalizer performance. In 2002 Kevin M. Passino described the Optimization Foraging Theory in article “Biomimicry of Bacterial Foraging” [18]. BFO technique consider the genes of those animals have successful foraging strategies since they are more likely to enjoy reproductive success and after many generations, poor foraging strategies are either eliminated or shaped into good one (redesigned). Such evolutionary principles have led scientists in the field of “Foraging Theory” to hypothesize that it is appropriate to model the activity of foraging as an optimization process. This optimization process used to develop adaptive controllers and cooperative control strategies for autonomous vehicles, also in the field of digital communication system like channel equalization and identification.

More recently in 2008, a rank based statistics approach known as Wilcoxon learning method [11] has been proposed for signals processing application to mitigate the linear and nonlinear learning problems. As per Jer-Guang Hesieh, Yih-Lon-Lin and Jyh-Horng Jeng the Wilcoxon learning algorithm has been applied to neural networks like Wilcoxon Multilayer Perceptron Neural Network (WMLPNN), Wilcoxon Generalized Radial Basis Function Network (WGRBF). The Wilcoxon approach provides promising methodology for many machine learning problems. We approach this method for digital communication system like channel equalization and identification.

1.4. Thesis Layout

Following the chapter on **Introduction,** The rest of the thesis is organised as follows

Chapter 2 provides the fundamental concepts of channel equalisation and discusses linear and nonlinear interferences like ISI, CCI and burst noise interference in a DCS. This chapter analyses the channel characteristics that bring out the need for an equaliser in a communication system. Subsequently an equaliser classification is presented which puts in context the work undertaken in this thesis. This chapter also describes the need of adaptive filter in channel equalization processes and also explains the gradient based adaptive algorithms used in channel equalizer for parameter updating.

Chapter 3 provides the introduction of soft computing techniques. This chapter describes neural network and its advantage in communication. This chapter also describes the artificial neural network equalizer like MLP, RBF, FLANN, ChNN, WMLPN and WGRBFN.

Chapter 4 This chapter represents evolutionary algorithm "bacterial foraging optimization" technique with some simulation results.

Chapter 5 This chapter represents all the simulation results and discussion. These equalizers have been simulated for different channel distortion conditions which include ISI, CCI and Burst Noise interference. The ANN equalizers like MLP, RBF, FLANN, ChNN, WMLP, and WGRBF have been simulated for performance evaluation. The performances of these equalizers have been compared with linear equalizers trained with LMS and RLS algorithm. BFO based training for linear equalizer has been simulated. BER has been used as the performance criteria for evaluating equalizers

Finally **Chapter 6** summarises the work undertaken in this thesis and points to possible directions for future research.

Chapter 2

Channel Equalization Techniques an Overview

Chapter 2

Channel Equalization Techniques an Overview

This chapter represent the development of artificial neural network based adaptive channel equalisers for a variety of channel impairments and brings out the need of an adaptive equaliser in a digital communication system (to mitigate the linear, nonlinear destruction like as Inter-symbol Interference, Co-channel Interference, Burst noise interference) and describes the classification of adaptive equalisers.

This chapter is organised as follows. Following this introduction, section 2.1 discusses the digital communication system in general. Section 2.2 discusses the propagation channel model in a digital communication system. Section 2.3 discusses the general concept of interferences ISI, CCI, ACI channel and burst noise interference. Section 2.4 discusses gradient based adaptive algorithms. Section 2.5 discusses the different types of channel models need for equalization. Section 2.6 discusses need of channel equalizer in digital communication system; subsequently describe the classification of adaptive equalisers. Section 2.7 discusses the optimal Bayesian symbol by symbol equaliser for ISI channels. Finally, section 2.8 provides the concluding remarks.

2.1 Digital Communication System

The general block diagram of a digital communication system is presented in Figure 2.1. In digital communication system, some of the blocks are not shown in the Figure 2.1. The data source constitutes the signal generation system that generates the information to be transmitted. The work of the encoder in the transmitter encode is to The information bits before transmission so as to provide redundancy in the system. This in turn helps in error correction at the receiver end. Some of the typical coding schemes used are convolutional codes, block codes and grey codes. The encoder does not form an essential part of the communication system but is being increasingly used. The digital data transmission requires very large bandwidth. The efficient use of the available bandwidth is achieved through the transmitter filter, also called the modulating filter. The modulator on the other hand places the signal over a high frequency carrier for efficient transmission. Some of the typical modulation schemes used in digital communication systems are amplitude shift keying (ASK), frequency shift keying (FSK), pulse amplitude modulation (PAM) and phase shift keying (PSK) modulation.

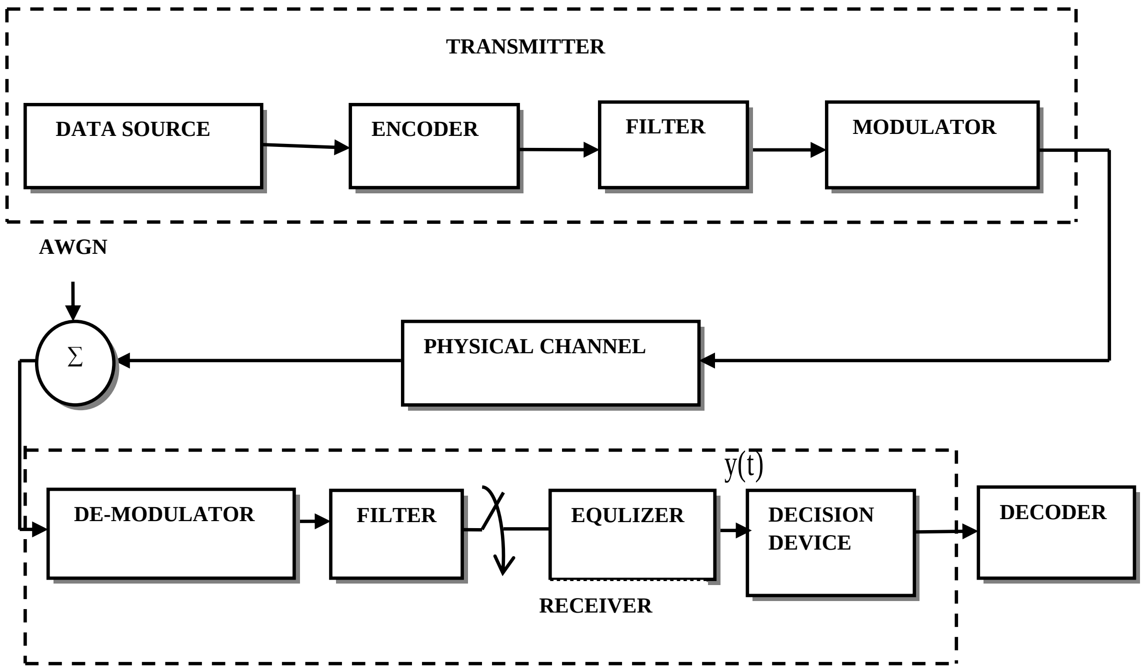


Figure.2.1 Block diagram of a digital communication system

The channel is the medium through which information propagates from the transmitter to the receiver. At the receiver the signal is first demodulated to recover the baseband transmitted signal. This demodulated signal is processed by the receiver filter, also called receiver demodulating filter, which should be ideally matched to the transmitter filter and channel. The equaliser in the receiver removes the distortion introduced due to the channel impairments. The decision device provides the estimate of the encoded transmitted signal. The decoder reverses the work of the encoder and removes the encoding effect revealing the transmitted information symbols.

2.2 Propagation Channel

This section discusses the channel impairments that mitigate the performance of a digital communication system (DCS). The DCS considered here is shown in Figure 2.1. The transmission of digital pulses over an analogue channel would require infinite bandwidth.

An ideal physical propagation channel should behave like an ideal low pass filter represented by its frequency response, ideal low pass filter represented by its frequency response,

$$H_c(f) = |H_c(f)| \exp(j\theta f) \tag{2.1}$$

Where $H_c(f)$ represents the Fourier transform (FT) of the channel and Θ is the phase response of the channel.

The amplitude response of the channel $|H_c(f)|$ can be defined as,

$$|H_c(f)| = \begin{cases} k_1 & |f| \le \omega_c \\ 0 & |f| > \omega_c \end{cases} \tag{2.2}$$

Where k_1 is a constant and ω_c is the upper cutoff frequency. The channel group delay characteristic is given by

$$\tau(f) = -\frac{1}{2\pi}\frac{\partial\theta(f)}{\partial f} = k_2 \tag{2.3}$$

Where k_2 is an arbitrary constant. The conditions described in (2.2) and (2.3) constitute fixed amplitude and linear phase characteristics of a channel. This channel can provide distortion free transmission of analogue signal band limited to at least ω_c. Transmission of the infinite bandwidth digital signal over a band limited channel of ω_c will obviously cause distortion. This demands for the infinite bandwidth digital signal is band limited to at least ω_c, to guarantee distortion free transmission. This work is done with the aid of transmitter and receiver filters shown in Figure.2.2. The combined frequency response of the physical channel, transmitter filter and the receiver filter can be represented as,

$$H(f) = H_T(f)H_c(f)H_R(f) \tag{2.4}$$

Where H_T (f), $H_c(f)$,$H_R(f)$ represents the FT of the transmitter, channel and receiver respectively. When the receiver filter is matched to the combined response of the propagation channel and the transmitter filter, the system provides optimum signal to noise ratio (SNR) at the sampling instant. As channel impulse response is not known beforehand, the receiver filter impulse response $h_R(t)$ is generally matched to the transmitter filter impulse response $h_T(t)$. This condition can be represented as

$$H_R(f) = H_T^*(f) \tag{2.5}$$

$$h_R(t) = h_T^*(-t) \tag{2.6}$$

Where, $H_T^*(f)$ and $h_T^*(t)$ are complex conjugates of $H_T(f)$ and $h_T(t)$ respectively. It is desired to select $H(f)$ so as to minimise the distortion at the output of the receiver filter at sampling instants. For the ideal channel presented in (2.1), the design of transmitter and receiver filters is the raised cosine filter and is given by,

$$H_{TR}(f)=\begin{cases} T & 0\le f\le \dfrac{1-\beta}{2T} \\ \dfrac{T}{2}\left\{1+\cos\left[\dfrac{\pi T}{\beta}\left(|f|-\dfrac{1-\beta}{2T}\right)\right]\right\} & \dfrac{1-\beta}{2T}\le |f|\le \dfrac{1+\beta}{2T} \\ 0 & |f|>\dfrac{1+\beta}{2T} \end{cases} \tag{2.7}$$

$$H_{TR}(f)=H_T(f)H_R(f) \tag{2.8}$$

Where, T is the source symbol period and $\beta, 0\le\beta\le 1$, is the excess bandwidth and H_{TR} is the FT of the combined response of transmitter and receiver filter. The plot of this combined filter response is presented in Figure 2.2. Figure 2.2(a) and Figure 2.2(b) represents the impulse response and frequency response of the combined filter respectively.

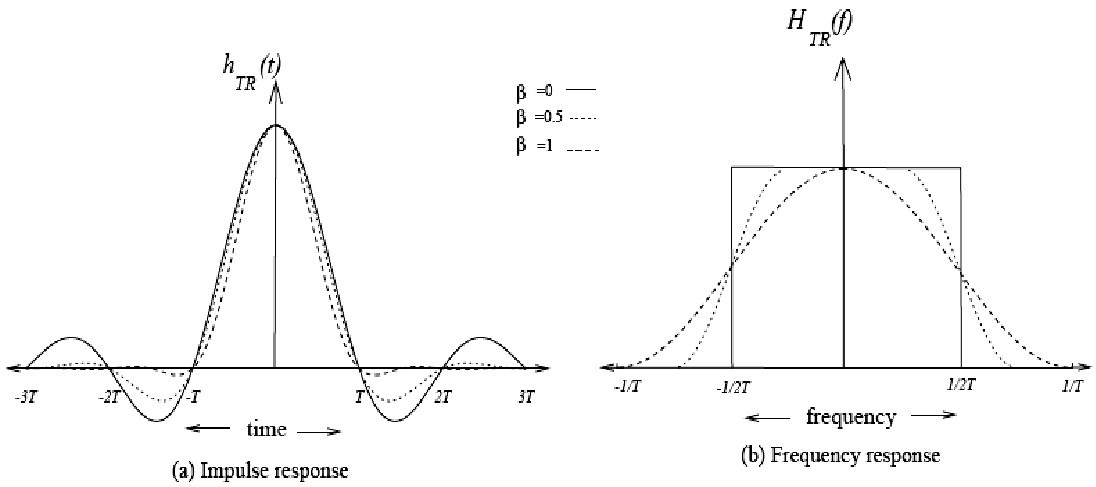


Figure 2.2. Raised cosine pulse and its spectrum

From the Figures 2.2(a) and 2.2(b), it can be observed that any value of β can provide distortion free transmission if the receiver output is sampled at the correct time. A sampling timing error causes ISI, which reduces with an increase in β. The special case of β =0 provides a pulse satisfying the condition

$$h_{TR}(t) = \frac{\sin\left(\frac{\pi t}{2}\right)}{\left(\frac{\pi t}{2}\right)} \tag{2.9}$$

Under this condition the channel can provide highest signalling rate[3], $T = 1/2\omega_c$. At the other extreme, $\beta = 1$ provides a signalling rate equal to reciprocal of the bandwidth $T = 1/\omega_c$. In this process, selection of β provides a compromise between quality and signaling speed.

It has been assumed that the physical channel is an ideal low pass filter (2.1). However, in reality all physical channels deviate from this behaviour. This introduces ISI even though the receiver is sampled at the correct time. The presence of this ISI requires an equaliser to provide proper detection.

In general all types of DCS's are affected by ISI. Communication systems are also affected by other forms of distortion. Multiple access techniques give rise to CCI and adjacent channel interference (ACI) in addition to ISI. The presence of amplifiers in the transmitter and the receiver front end causes nonlinear distortion. Fibre optic communication systems are also affected by nonlinear distortion [3]. On the other hand the mobile radio channels are affected by multi-path fading due to relative motion between the transmitter and receiver [4].

In the following subsections these channel impairments are discussed and the channel models are presented. These models are used in the later chapters for evaluating equalisation algorithms that have been presented in this thesis. The discussions in these subsections are limited only to the channel effects that have been analysed in this thesis.

2.3. Interference

Today's communication systems transmit high speed data over the communication channels. During this process the transmitted data is corrupted due to the effect of linear and nonlinear distortions.

Linear distortion includes inter-symbol interference (ISI), co-channel interference (CCI), and adjacent channel interference (ACI) in the presence of additive white Gaussian noise (AWGN).

The nonlinear distortion occurs in the system by impulse noise, modulation, demodulation, amplification process, cross-talk in the communication pipelines and depended on the nature of the channel. The following sections briefly describe the linear and nonlinear interferences.

2.3.1 Inter Symbol Interference (ISI)

Inter-symbol interference (ISI) arises when the data transmitted through the channel is dispersive, in which each received pulse is affected somewhat by adjacent pulses and due to which interference occurs in the transmitted signals.

In Figure. 2.3. Shown the block diagram of baseband binary data transmission system, cascade of the transmitter filter $h_T(t)$, the channel $h_C(t)$ and the receiver $h_R(t)$ matched filter and the T spaced sampler.

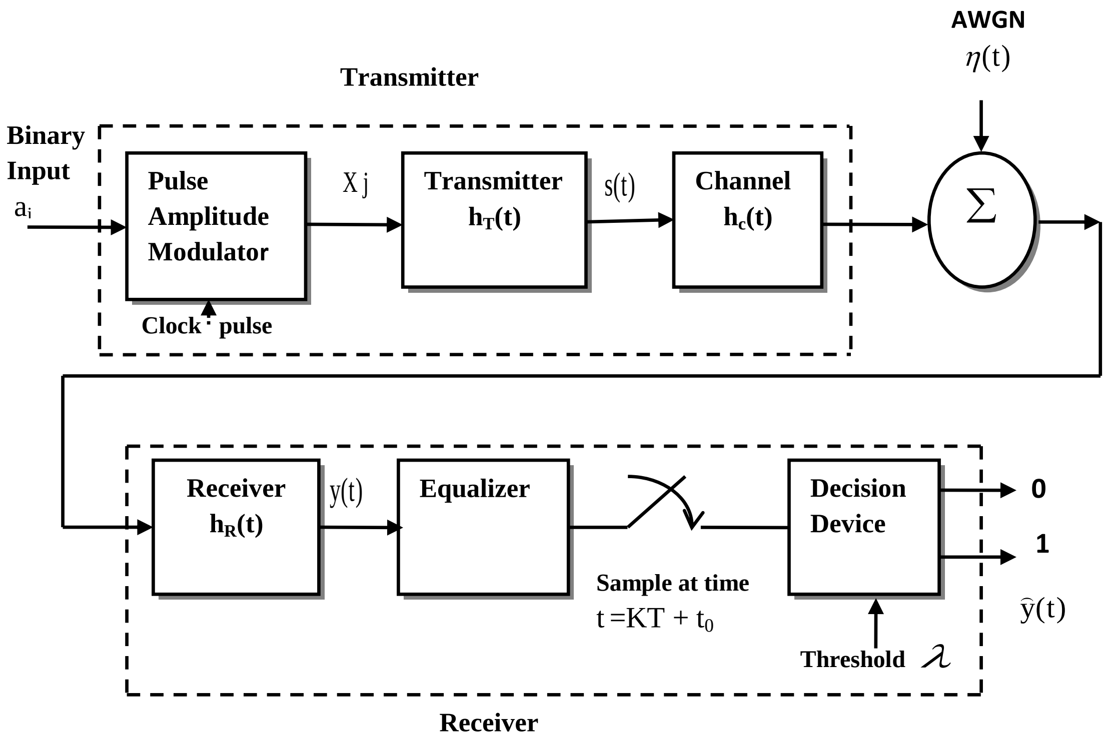


Figure. 2.3 Baseband binary data transmission system

Here, the incoming binary pulse sequence consists of symbols 1 and 0, each of duration T. The pulse amplitude modulation modifies this binary sequence into a new sequence of short pulses (approximating a unit impulse), whose amplitude x_j is represented in the polar from

$$x_j = \begin{cases} +1 & \text{if} \quad \text{symbol} \quad a_j \quad \text{is} \quad 1 \\ -1 & \text{if} \quad \text{symbol} \quad a_j \quad \text{is} \quad 0 \end{cases} \tag{2.10}$$

The sequence of short pulses so produced is applied to a transmit filter of impulse response $h_T(t)$, producing the transmitted signal

$$s(t) = \sum_j x_j h_T(t - jT) \tag{2.11}$$

In addition, the channel adds random noise to the signal at the receiver input. The channel observed output y(t) is given by the sum of the noise free channel output $\hat{y}(t)$, which in turn is formed by the convolution of the transmitted sequence s(t) with the channel taps h_C, $0 \le n-1$ and adaptive white Gaussian noise $\eta(t)$.

The received filter output y(t) is written as

$$y(t) = \mu \sum_j x_j h_C(t - jT) + \eta(t) \tag{2.12}$$

Where μ is a scaling factor use to account of amplitude changes incurred in the course of signal transmission through the system, and $h_C(t - jT)$ represent the effect of transmission delay. To simplify the exposition, we have put this delay equal to zero in equation (2.12) without loss of generality.

Generally the receive filter output y(t) is sampled at time t = iT ,where i is a integer values and $-\infty \le t \le \infty$.

$$y(i) = \mu x_i \quad + \quad \mu \sum_{t=-\infty}^{\infty} x_j h_C[(i - j)T] \quad + \quad \eta(i) \tag{2.13}$$

In equation (2.13), the first term μx_i represents the contribution of the i^{th} transmitted bit. The second term represents the residual effect of all other transmitted bits on the decoding of the i^{th} bit, this residual effect due to the occurrence of pulses before and after the sampling instant i^{th} is called inter-symbol interference (ISI). The last term $\eta(i)$ represents the noise sample at time t .

In the absence of both ISI and noise, we observe from equation (2.13) that

$$y(i) = \mu x_i \tag{2.14}$$

Which shows that, under these ideal conditions, the i^{th} transmitted bit is decoded correctly. The unavoidable presence of ISI and noise in the system, however, introduces errors in the decision device at the receiver output. Therefore, in the design of the transmit and receive filters, the objective is to minimize the effects of noise and ISI and thereby deliver the digital data to their destination with the smallest error rate possible.

The ISI is zero if and only if $h(t-jT)=0,\ j \neq 0$; that is, if the channel impulse response has zero crossings at T-spaced intervals. In channel impulse response. When the impulse response has such uniformly spaced zero crossings, it is said to satisfy Nyquist's first criterion. In frequency-domain terms, this condition is equivalent to

$$H'(f) = \sum_{n_a} H(f - \frac{n}{T}) = \text{Constant} \qquad |f| \leq \frac{1}{2T} \qquad (2.15)$$

H(f) is the channel frequency response and $H'(f)$ is the "folded" (aliased or overlapped) channel spectral response after symbol-rate sampling. The band $|f| \leq 1/2T$ is commonly referred to as the Nyquist or minimum bandwidth. When H(f)=0 for $|f| > 1/T$ (the channel has no response beyond twice the Nyquist bandwidth), the folder response $H'(f)$ has the simple from

$$H'(f) = H(f) + H\left(f - \frac{1}{T}\right) \qquad 0 \leq f \leq 1/T \qquad (2.16)$$

Figure 2.4 (a) and (d) shows the amplitude response of two linear-phase low-pass filters: one an ideal filter with Nyquist bandwidth and the other with odd (or vestigial) symmetry around $1/2T$ hertz. As illustrated in figure 2.4 (b) and (e), the folded frequency response of each filter satisfies Nyquist's first criterion.

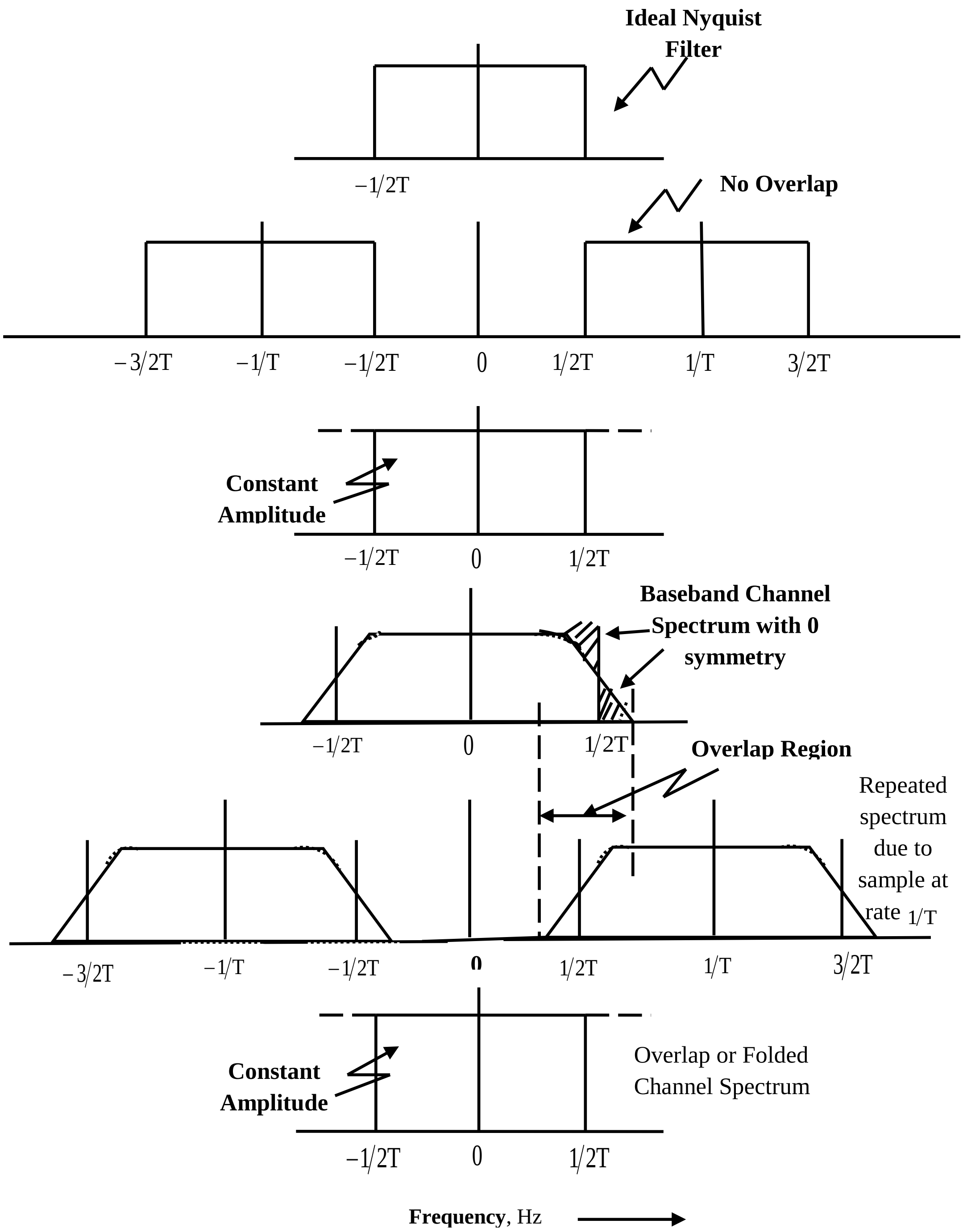


Figure. 2.4(a)-(f) Linear phase filters which satisfy Nyquist's first criterion

In practice, the effect of IS1 can be seen from a trace of the received signal on an oscilloscope with its time base synchronized to the symbol rate.

2.3.2. Co-channel Interference and Adjacent Channel Interference

Co-channel Interference (CCI) and Adjacent Channel Interference (ACI) occur in communication systems due to multiple access techniques using space, frequency or time. CCI occurs in cellular radio and dual-polarized microwave radio, for efficient utilization of the allocated channels frequencies by reusing the frequencies in different cells.

Figure.2.5 shows a digital communication system model where s(t) is the transmitted symbol sequence, $\eta(t)$ is additive white Gaussian noise, $y(t)$ is a received signal sequence sampled at the rate of the symbol interval T_s, $\hat{y}(t-d)$ is an estimate of the transmitted sequence s(t) and d denotes the delay associated with estimation. The received signal is additionally corrupted by n co-channel interference sources. The receiver has a copy of the training signal transmitted by the transmitter.

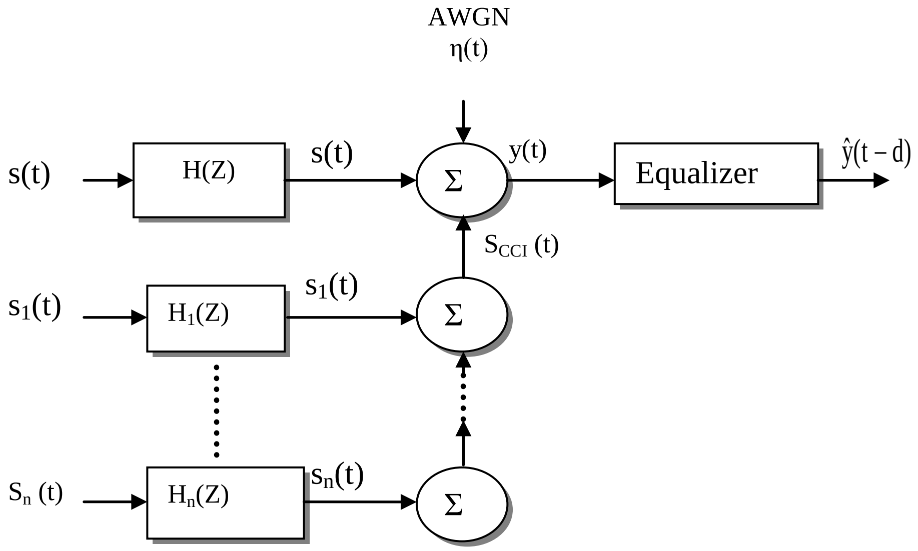


Figure. 2.5. Communication system model with Co-channel interference

The received signal sequence is defined by the following equation.

$$y(t) = s(t) + s_{CCI}(t) + \eta(t) \tag{2.17}$$

Where s(t) is the output of the desired channel, $Scc_I(t)$ is the co-channel interference component. The desired signal s(t) and co-channel signal $Scc_I(t)$ are represent as

$$s(t) = \sum_{i=0}^{n} h(i)s(t-i) \tag{2.18}$$

$$s_{CCI}(t) = \sum_{j=1}^{k} \sum_{i=0}^{n_{hj}-1} h_j(i)s_j(t-i) \tag{2.19}$$

Where s(t) and $s_j(t)$ are the desired and co-channel data symbols respectively, h(i) and $h_j(i)$ are the impulse responses of the desired channel and the j^{th} co-channel, having n and n_{hj} taps respectively. Furthermore, the desired and co-channel data symbols and noise samples are assumed to be mutually uncorrelated. Without loss of generality the transmitted sequences can be assumed to be bipolar (± 1). The signal-to-noise ratio (SNR) and the signal-to-interference ratio (SIR) are defined as

$$\mathrm{SNR} = \frac{\sigma_s^2}{\sigma_e^2} \qquad \mathrm{SIR} = \frac{\sigma_s^2}{\sigma_{CCI}^2} \tag{2.20}$$

Where $\sigma_e^2, \sigma_s^2, \text{and} \sigma_{CCI}^2$, are the noise variance, the signal power and the co-channel signal power respectively.

In digital communication system adjacent channel interference is causes due to inter carrier spacing between different cells in time division multiple access (TDMA)[13] and inter carrier spacing among carriers in the same cell in FDMA[12,14,15] systems. The frequency spectrum of the signals that carry the desired signal, the Co-channel Interference and Adjacent Channel Interference signals is presented in Figure 2.6.

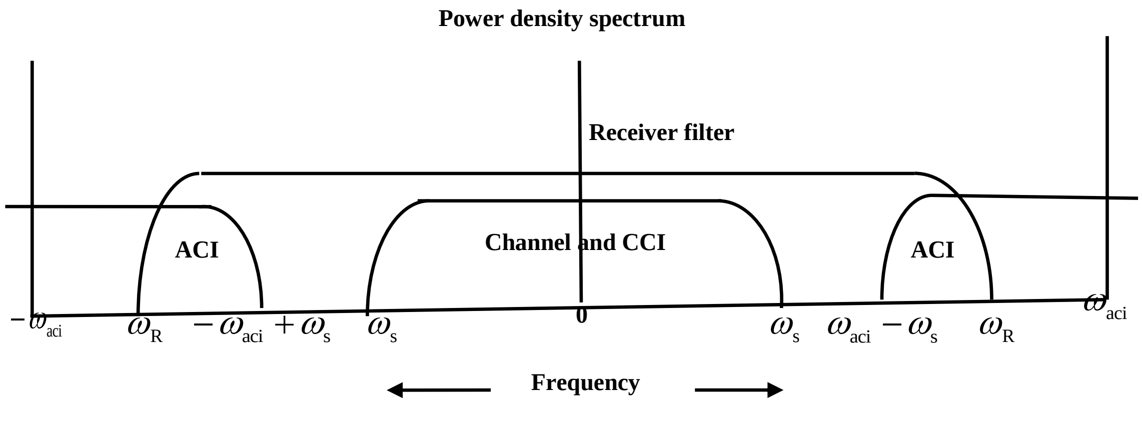


Figure.2.6 Spectrum of desired signal, CCI and ACI in DCS

Here the signal of interest occupies a double sided bandwidth of ω_S The Co-channel Interference signal also occupies the same frequency band. The Adjacent Channel Interference signal centre frequency is spaced at ω_{aci} with respect to the desired carrier. The receiver filter rejects signal beyond ω_R. The guard band provided in the system is $\omega_{aci_s} - 2\omega_S$. From the Figure 2.6 it can be seen that a portion of the signal spectrum in the neighbouring carrier with respect to the signal of interest is received by the receiver filter and this signal is the main cause of ACI.

2.3.3. Burst Noise Interference

Burst noise is a high intensity noise which occurs for short duration of time with fixed burst length means a series of finite-duration Gaussian noise pulses. As shown in Figure. 2.7 The block diagram of burst noise model. The receiver input is s(t) + n_b(t) where s(t) is the binary signal component and n_d(t) is the noise Component [17]. The noise is given by

$$n_d(t) = \eta(t) + n_b(t) \tag{2.21}$$

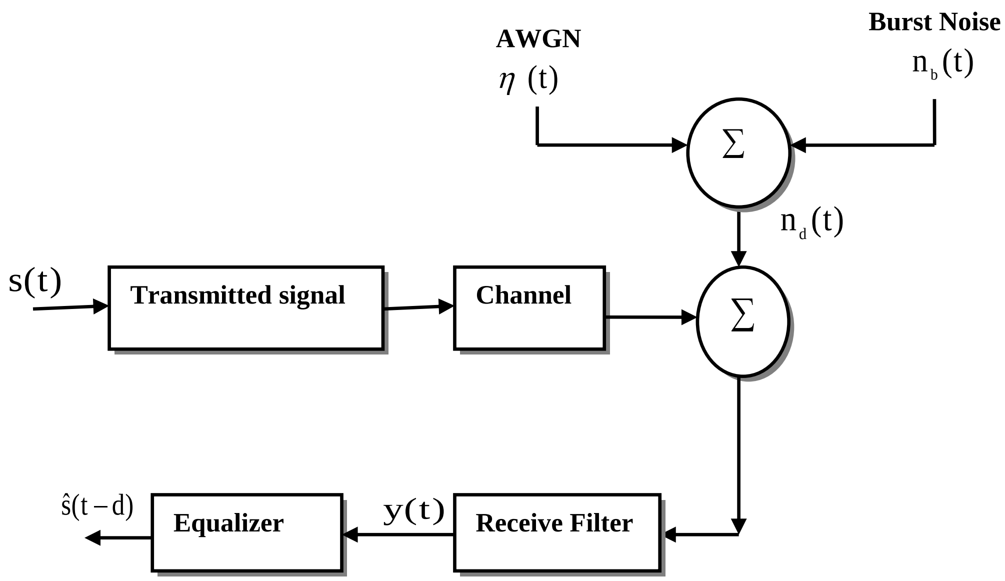


Figure. 2.7 Block diagram of Burst noise model

Where $\eta(t)$ is the background Gaussian-noise component and n_b(t) is the burst-noise component. The combination of the background Gaussian noise and burst noise is referred to as bursty noise.

The burst-noise component of the channel noise, let $\hat{s}(t)$ denote a sample function from a delta-correlated Gaussian stochastic process with zero mean and double-sided power spectral density (PSD) $N_b/2$ and let $\{t_i\}$ denote a set of Poisson points with average rate v. The burst noise component is expressed as

$$n_b(t) = \hat{s}(t)\sum_{i=-\infty}^{\infty}\Pi\left(\frac{t-t_i-T/2}{T}\right) \tag{2.22}$$

Where $\Pi(t/T)$ is defined to be a unit-amplitude pulse of width T centred at t = 0. When two pulses overlap, the stochastic process is doubled in amplitude in the overlapping interval. In eq. (2.22), T is the time duration of each Gaussian-noise burst and t_i is the time at which the burst begins. The double-sided PSD for burst noise is

$$s_b(f) = vd(N_b/2), \qquad -\infty < f < \infty \tag{2.23}$$

and is easily derived via the autocorrelation function and the Wiener-Khinchine theorem. Since the PSD for burst noise is constant, the process is white. The background Gaussian-noise component $\eta(t)$, is assumed to be zero-mean and delta correlated with double-sided PSD.

$$s(f) = N/2, \qquad -\infty < f < \infty \tag{2.24}$$

If $\eta(t)$ and $\hat{s}(t)$ is uncorrelated then descriptions for Gaussian noise and burst noise, it is clear that burst noise is characterized by Gaussian noise which contains bursts of larger variance Gaussian noise. Only four parameters are required to completely describe bursty noise; the mean burst rate v, the burst duration T**,** the single-sided PSD for the Gaussian noise $\eta(t)$, and the single-sided PSD for the burst noise N_b**.** Since $\eta(t)$ and $\hat{s}(t)$ are uncorrelated, the double-sided PSD for bursty noise is

$$s(f) + s_b(f) = (N + N_b vd)/2 \equiv N_I/2 \tag{2.25}$$

The fraction of bursty noise is defined as

$$\psi = \frac{s_b(f)}{s(f) + s_b(f)} \tag{2.26}$$

This parameter is useful because it allows the limiting cases of bursty noise to be considered. For example, $S_b(f) = 0$ yields $\psi = 0$, which corresponds to a Gaussian-noise channel, and s(f) = 0 yields $\psi = 1$, which corresponds to a burst-noise channel. It is an easy matter to determine N_I and ψ from v, d, N_g and N_b. N and N. Together, v, d, N_I, and ψ are the only parameters required to completely describe bursty noise and will be used for all subsequent developments. Since burst locations are given by a Poisson distribution, bursty noise is a stationary stochastic process.

2.4 The Adaptive Filter

Adaptive filters used for wide range of applications like Direct Modelling (System Identification), Inverse Modelling, Channel Equalization, etc. Channel equalization was one of the first applications of adaptive filters and is described in the pioneering work of Lucky [19]. Today, it remains as one of the most popular uses of an adaptive filter.

2.4.1 Gradient Based Adaptive Algorithm

An adaptive algorithm is a procedure for adjusting the parameters of an adaptive filter to minimize a cost function chosen for the task at hand. In this section, we describe the general form of many adaptive FIR filtering algorithms and present a simple derivation of the LMS adaptive algorithm. In our discussion, we only consider an adaptive FIR filter structure in Figure.2.8. Such systems are currently more popular than adaptive IIR filters because

(1) The input-output stability of the FIR filter structure is guaranteed for any set of fixed coefficients, and

(2) The algorithms for adjusting the coefficients of FIR filters are simpler in general than those for adjusting the coefficients of IIR filters.

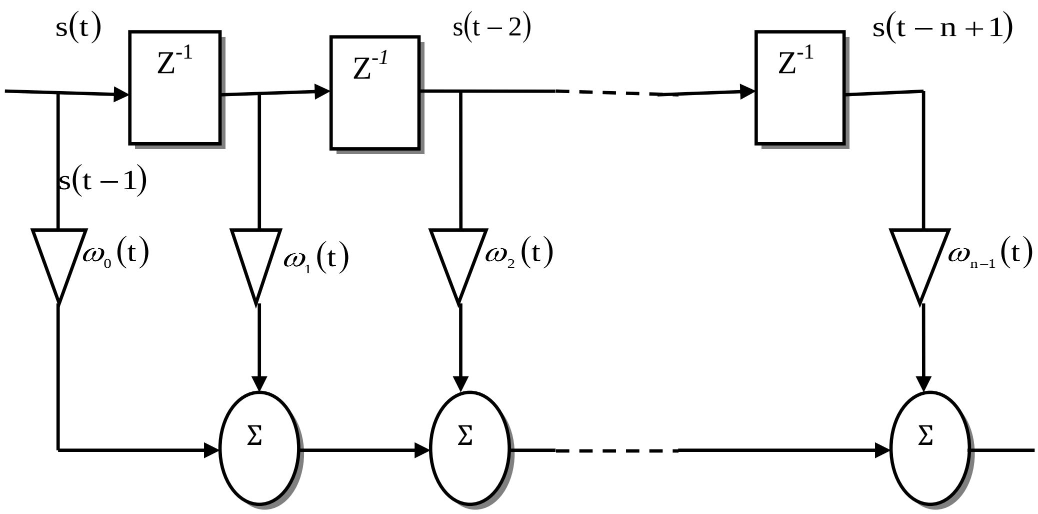


Figure. 2.8 Structure of an FIR filter

Figure.2.8 shows the structure of a direct-form FIR filter, also known as a tapped- delay-line or transversal filter, where z^{-1} denotes the unit delay element and each $\omega_i(t)$ is a multiplicative gain within the system. In this case, the parameters in $\omega(t)$ correspond to the impulse response values of the filter at time n. We can write the output signal y(t) as

$$y(t) = \sum_{i=0}^{n-1} \omega_i(t) \; s(t-i) = W^T(t) \; S(t) \tag{2.27}$$

Where,

$S(t) = [s(t), s(t-1) \cdots s(t-n+1)]^T$ denotes the input signal vector and T denotes vector transpose.

$\omega_i(t) = [\omega_0(t), \omega_1(t) \cdots, \omega_{n-1}(t)]^T$ is $\{\omega_i(t)\}, \quad 0 \le i \le n-1$ are the n parameters of the system at time t. The general form of an adaptive FIR filtering algorithm is

$$W(t+1) = W(t) + \mu(t)\, G\,(e(t)\, s(t)\, \psi(t)) \tag{2.28}$$

where G(•) is a particular vector-valued nonlinear function, $\mu(t)$ is a step size parameter, $e(t)$ and s(t) are the error signal and input signal vector, respectively, and $\psi(t)$ is a vector of states that store pertinent information about the characteristics of the input and error signals. In the simplest algorithms, $\psi(t)$ is not used.

The form of G(•) in (2.28) depends on the cost function chosen for the given adaptive filtering task. The Mean-Squared Error (MSE) cost function can be define as

$$J_{MSE}(t) = \frac{1}{2} \int_{-\infty}^{\infty} e^2(t)\, p_t(e(t))\; d\,e(t) \tag{2.29}$$

$$= \frac{1}{2} E\{e^2(t)\} \tag{2.30}$$

Where, $p_t\,(e\,(t))$ represents the probability density function of the error at time t and E{•} is the expectation integral on the right-hand side of (2.30).

In adaptive FIR filters the coefficient of W(t) are updated to minimize $J_{MSE}(t)$. The formulation of this problem for continuous-time signals and the resulting solution was first derived by Wiener [27]. Hence, this optimum coefficient vector $W_{MSE}(t)$ is often called the *Wiener solution* to the adaptive filtering problem. $W_{MSE}(t)$ can be found from the solution to the system of equations

$$\frac{\partial J_{MSE}(t)}{\partial w_i(t)} = 0, \qquad 0 \le i \le L-1 \tag{2.31}$$

Taking derivatives of $J_{MSE}(t)$ in (3.3) we obtain

$$\frac{\partial J_{MSE}(t)}{\partial w_i(t)} = E\left\{e(t)\frac{\partial e(t)}{\partial w_i(t)}\right\} \tag{2.32}$$

$$= -E\{e(t)s(t-i)\} \tag{2.33}$$

To expand the last result by defining the matrix $R_{SS}(t)$ (autocorrelation matrix) and vector $P_{dS}(t)$ (cross correlation matrix). Thus, so long as the matrix $R_{SS}(t)$ is invertible, the optimum Wiener solution vector for this problem is

$$W_{MSE}(t) = R_{SS}^{-1}(t)P_{dS}(t) \tag{2.34}$$

Another method of *steepest descent* is an optimization procedure for minimizing the cost function J(t) with respect to a set of adjustable parameters W(t). This procedure adjusts each parameter of the system according to relationship

$$w_i(t+1) = w_i(t) - \mu(t)\frac{\partial J(t)}{\partial w_i(t)} \tag{2.35}$$

As per this, the i^{th} parameter of the system is updated according to the derivative of the cost function with respect to the i^{th} parameter. These weights vector can be represented as

$$W(t+1) = W(t) - \mu(t)\frac{\partial J(t)}{\partial W(t)} \tag{2.36}$$

Where, $\partial J(t)/\partial W(t)$ is a vector of derivatives $\partial J(t)/\partial w_i(t)$.

The iterative solution to this can be represented as

$$W(t+1) = W(t) + \mu(t)(P_{ds}(t) - R_{ss}(t)W(t)) \tag{2.37}$$

It can be seen that the steepest descent procedure depends on the statistical quantities E{d(t)s(t-i)} and E{s(t-i)s(t-j)} contained in $P_{dS}(t)$ and $R_{SS}(t)$, respectively.

2.4.2 Least Means Square Algorithm

The cost function J(t) chosen for the steepest descent algorithm of eq.(2.34) determines the coefficient solution obtained by using adaptive filter. If the MSE cost function in (2.33) is chosen, the resulting algorithm depends on the statistics of s(t) and d(t) because of the expectation operation that defines this cost function. One such cost function is the least-squares cost function given by

$$J_{LS}(t) = \sum_{i=0}^{t} \alpha(i)(d(i) - W^T(t)S(i))^2 \tag{2.38}$$

The weight update equation for LMS can be represented as

$$W(t+1) = W(t) + \mu e(t)S(t) \tag{2.39}$$

Where μ is learning factor, equation (2.39) requires only multiplications and additions to implement. In fact, the number and type of operations needed for the LMS algorithm is nearly the same as that of the FIR filter structure with fixed coefficient values and hence LMS has become very popular.

In effect, the iterative nature of the LMS coefficient updates is a form of time-averaging that smoothes the errors in the instantaneous gradient calculations to obtain a more reasonable estimate of the true gradient.

2.4.3 Recursive Least Squares Algorithm

The recursive least squares (RLS) algorithm is another algorithm for determining the coefficients of an adaptive filter. In contrast to the LMS algorithm, the RLS algorithm uses information from all past input samples (and not only from the current tap-input samples) to estimate the (inverse of the) autocorrelation matrix of the input vector. To decrease the influence of input samples from the far past, a weighting factor for the influence of each sample is used. This cost function can be represented as

$$J[t] = \sum_{i=1}^{t} \rho^{t-i} | e[i, t] |^2 \tag{2.40}$$

Where, the error signal $e_i[i,t]$ is computed for all times $1 \leq i \leq t$ using the current filter coefficients $w[t] : e[i,t] = d[i] - w^T[t]s[i]$, where s[$i$] and w^T represents input signal and transpose of the channel coefficient vector respectively.

Analogous to the derivation of the LMS algorithm we find the gradient of the cost function with respect to the current weights can be represented as nomenclature

$$\Delta_h J[t] = \sum_{i=1}^{t} \rho^{t-i} \left(-2E(d[i]s[i]) + 2E\left(s[i]s^T[i]\right)w[t]\right) \tag{2.41}$$

Where, s^T represents the transpose of the input signal vector. If search for the minimum of the cost function by setting its gradient to zero $\Delta_h J[t] = 0$.

Finally, the weights update equation is

$$w[t] = w[t-1] + \mu[t]\left(d\ [t] - s^T[t]w[t-1]\right) \tag{2.42}$$

The equations to solve in the RLS algorithm at each time step (2.42). The RLS algorithm is computationally more complex than the LMS algorithm. The RLS algorithm typically shows a faster convergence compared to the LMS algorithm.

Example 2.1. In this example for channel equalization we used the LMS and RLS algorithm. For simulation used the structure of the equalizer is a single input linear adaptive neural network equalizer, parameters details given in the table below. The BER plot is plotted for different delay of 0, 1, 2, and 3 respectively.

FIR Filter	**Channel (Non-minimum phase channel)**	**Training Samples**	**Testing BER Samples**	**SNR**
5- Tap	$H_1(Z)= 0.5 + Z^{-1}$	**1000**	**100000**	**30dB**

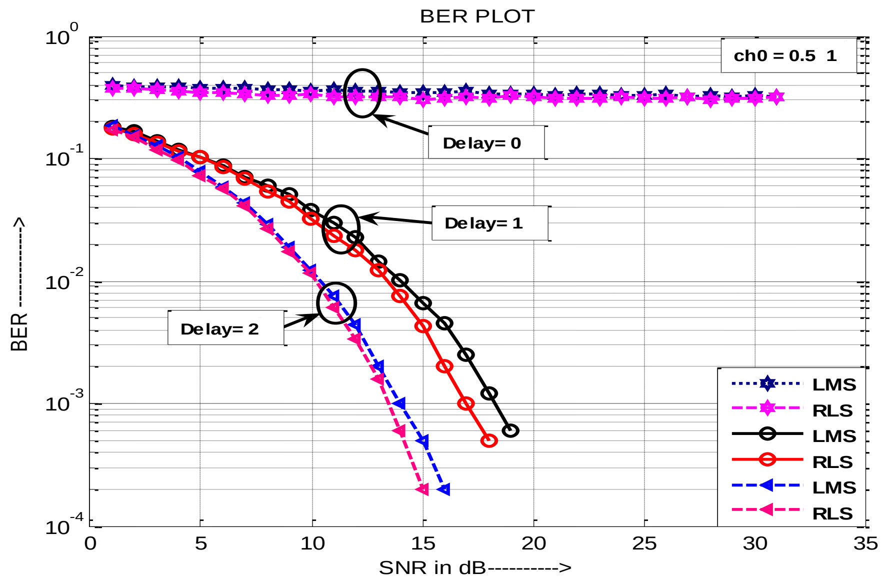


Figure. 2.9 BER performance of LMS and RLS based equalizer for ch_0

From the BER performance it is seen that RLS perform better then LMS based equalizer.

2.5 Channels Models

When all the root of the model z-transform lie within the unit circle, the channel is termed minimum phase [21] the inverse of a minimum phase channel is convergent, illustrated by Equation (2.43)

$$
\begin{aligned}
H(z) &= 1.0 + 0.5z^{-1} \\
\frac{1}{H(z)} &= \frac{1}{1.0 + 0.5z^{-1}} \\
&= \sum_{i=0}^{\infty}\left(-\frac{1}{2}\right)^{i} z^{-i} \\
&= 1 - 0.5z^{-1} + 0.25z^{-2} - 0.125z^{-3} + \cdots
\end{aligned} \tag{2.43}
$$

Where as the inverse of non-minimum phase channels are not convergent, given as

$$
\begin{aligned}
H(z) &= 0.5 + 1.0z^{-1} \\
\frac{1}{H(z)} &= \frac{z}{1.0 + 0.5z} \\
&= z.\left[\sum_{i=0}^{\infty}\left(-\frac{1}{2}\right)^{i} z^{-i}\right] \\
&= z.\left[1 - 0.5z + 0.25z^{2} - 0.125z^{3}\right]
\end{aligned} \tag{2.44}
$$

Since equalizers are designed to invert the channel distortion process they will in effect model the channel inverse. The minimum phase channel has a linear inverse model therefore a linear equalization solution exists. However, limiting the inverse model to *m*-dimensions will approximate the solution and it has been shown that non-linear solutions can provide a superior inverse model in the same dimension.

A linear inverse of a non-minimum phase channel does not exist without incorporating time delays. A time delay creates a convergent series for a non-minimum phase model, where longer delays are necessary to provide a reasonable equalizer. Equation (2.45) describes a non-minimum phase channel with a single delay inverse and a four sample delay inverse. The latter of these is the more suitable form for a linear filter.

$$
\begin{aligned}
H(z) \quad &= 0.5 + 1.0z^{-1} \\
z^{-1}\frac{1}{H(z)} &= \frac{1}{1 + 0.5z} = 1 - 0.5z + 0.25z^{2} - 0.125z^{3} + \cdots (non\ causal) \\
z^{-4}\frac{1}{H(z)} &= z^{-3} - 0.5z^{-2} + 0.25z^{-1} - 0.125z + \cdots (truncated\ and\ causal)
\end{aligned} \tag{2.45}
$$

The three-tap maximum phase channel $\mathrm{H(z)} = 0.26 + 0.93\mathrm{z}^{-1} + 0.26\mathrm{z}^{-2}$ and $\mathrm{H(z)} = 0.3482 + 0.8704\,\mathrm{z}^{-1} + 0.3482\,\mathrm{z}^{-2}$ is also used throughout this thesis for simulation purposes. A channel delay'd' is included to assist in the classification, so that the desired output becomes $\hat{\mathrm{s}}(\mathrm{t} - \mathrm{d})$.

2.6 Need of Channel Equalizer

Digital communication systems transmitted high speed and efficient data over the communication channels. During this process the transmitted data is distorted, due to the effect of linear and nonlinear distortions. Linear distortion includes inter-symbol interference (ISI), co-channel interference (CCI) in the presence of additive white Gaussian noise (AWGN). Nonlinear distortion are caused due to the subsystem like amplifiers, modulator, demodulator, etc, Compensating all these channel distortion calls for channel equalization techniques at the receiver side, to reconstruct the transmitted symbols correctly. For which generally an adaptive equalization technique is used.

2.6.1 Adaptive Equalisation

It is very difficult for estimating both the channel order and the distribution of energy among the taps and even it is very difficult to predict the effect of the environment on these taps. so it necessary that the equalization process must be adaptive, means the equaliser need to be adapted very frequently with the changing environment. This includes two phases [16]. Firstly the equaliser needs to be trained with some known samples in the presence of some desired response (Supervised Learning). After training the weights and various parameters associated with the equaliser structure is frozen to function as a detector. These two processes are frequently implemented to keep the equaliser adaptive. We call “the Equaliser is frozen” if we keep the adaptable parameters of the equaliser constant. A typical digital communication system with adaptive equalizer is shown in Figure.2.9.

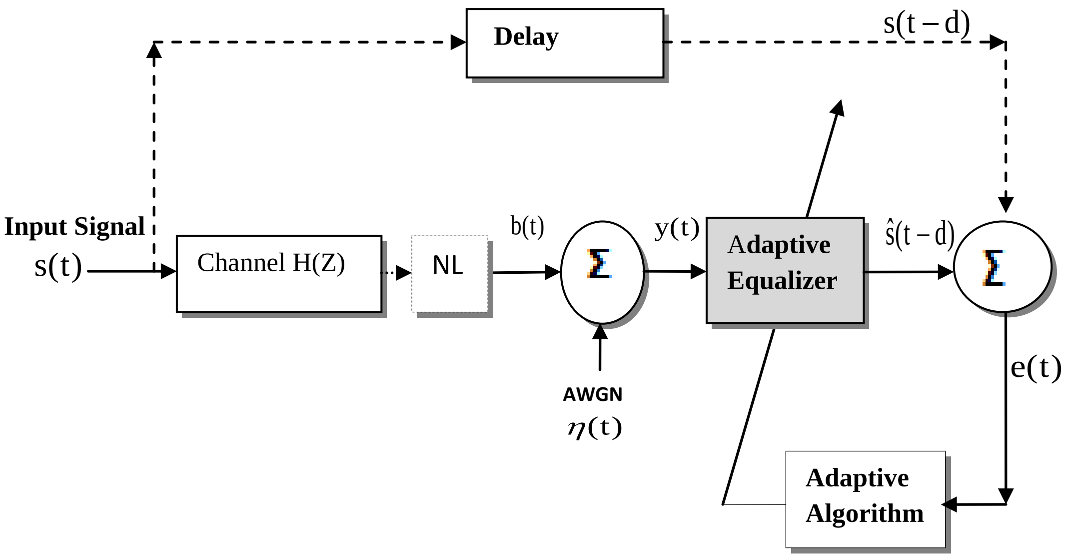


Figure.2.10 Block diagram of a digital transmission system with equalizer.

The transmitted symbols are given as s(t) for discrete time instant. They are then passed into the channel model which may be linear or nonlinear. A finite impulse response (FIR) model is widely used to model a linear channel whose uncorrupted output at time instant t may be written as

$$y(t) = \sum_{i=0}^{n_a-1} h_i s(t-i) + \eta(t) \tag{2.46}$$

Where, h_i are the channel tap values and t is the length of the FIR channel. The "NL" block represents the nonlinear distortion of the symbols in the channel and its output may be expressed as

$$b(t) = \psi\left[(s(t), s(t-1), \ldots. s(t-r+1)) \ldots (h_1, h_2, h_3, \ldots. h_{n_a-1})\right] \tag{2.47}$$

Where, $\psi(.)$ is some nonlinear function generated by the "NL" block. The channel output y(t) is corrupted with additive white Gaussian noise (AWGN) $\eta(t)$. This corrupted signal is compared with the delay versions of input signal and finds the error $e(t)$. This error is used to update the adaptable parameters of the equaliser using some adaptive algorithm. These steps constitute the training process of the equalisation. After the completion of training, the equaliser output is compared with some threshold and decision is made regarding the symbol received.

2.6.2 Need for nonlinear equalisers

The main reason why nonlinear equalisers are preferred over their linear counterpart is that the linear equalizers do not perform well on channels which have deep spectral nulls in the pass-band. In an attempt to compensate for the distortion, the linear equaliser places too much gain in the vicinity of the spectral nulls, thereby enhancing the noise present in these frequencies.

Linear equalizers view equalisation as inverse problem while non-linear equalisers view equalisation as a pattern classification problem where equalizer classifies the input signal vector into discrete classes based on transmitted data.

Example 2.2. Consider the following example of the channel states for the two channels,

$$H_1(z) = 1 + 0.5z^{-1}$$

$$H_2(z) = 0.34 + 0.87z^{-1} + 0.34z^{-2}$$

For these two channels, channel $H_1(z)$ is a minimum phase channel and hence classification is not a big problem in this channel. Problem starts when equalising the non-minimum phase channels [44]. The channel state diagram for the channel $H_1(z)$ is shown in figure. 2.11. The channel state diagram for channel $H_2(z)$ is shown in figure. 2.12. For these two channels, the channel state diagram is plotted for delay zero and at a SNR 20dB.

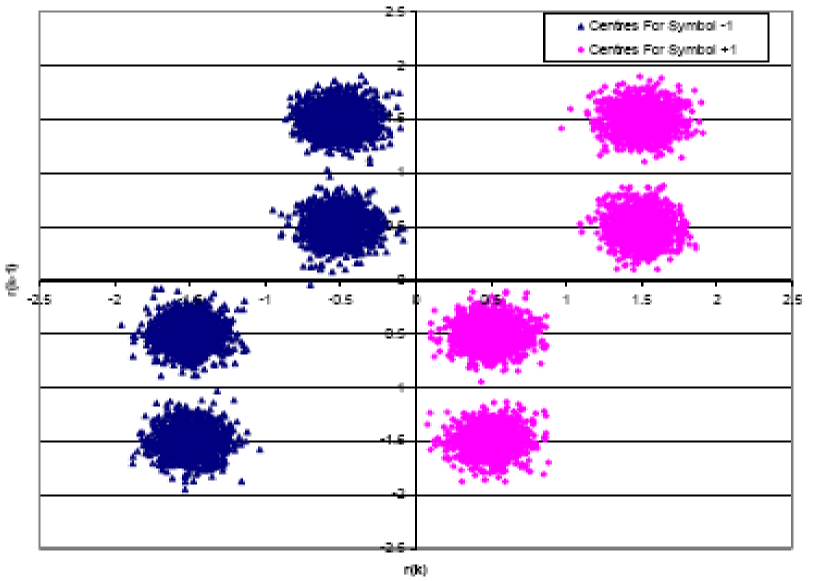


Figure. 2.11. Channel State diagram for channel $H_1(z)$

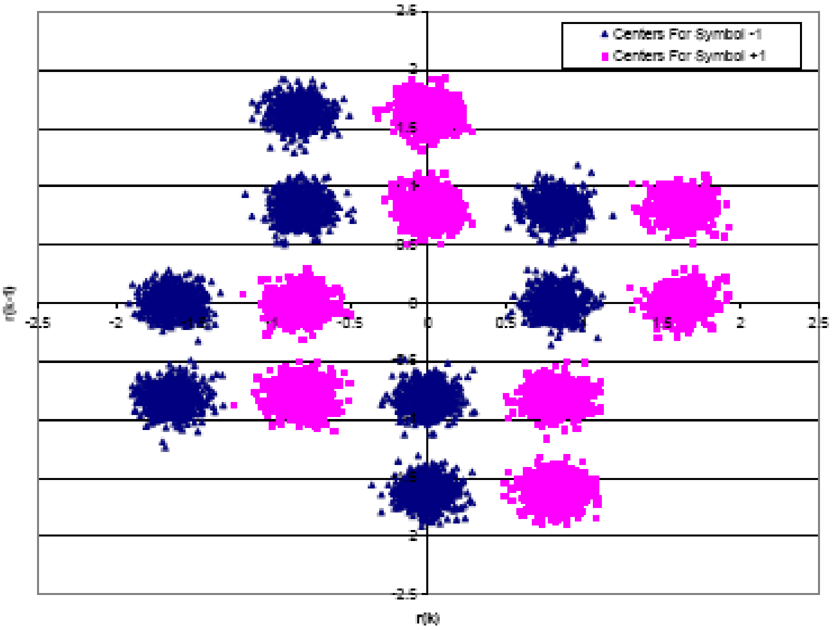


Figure. 2.12. Channel State diagram for channel $H_2(z)$

Channel $H_2(z)$ is a family of mixed phase channel. For this channel, a simple linear decision boundary cannot classify the symbols easily. It needs a nonlinear decision boundary or even a hyper-plane in multi-dimensional channel space. Such a decision boundary cannot be achieved using a linear filter.

2.6.3 Adaptive Equalizer classification

This section provides adaptive equalizer classification and specifies the domain of the investigation undertaken in this thesis. The general equalizer classification is presented in Figure.2.12. In general the family of adaptive equalizers can be classified as supervised equalizers and unsupervised equalizers.

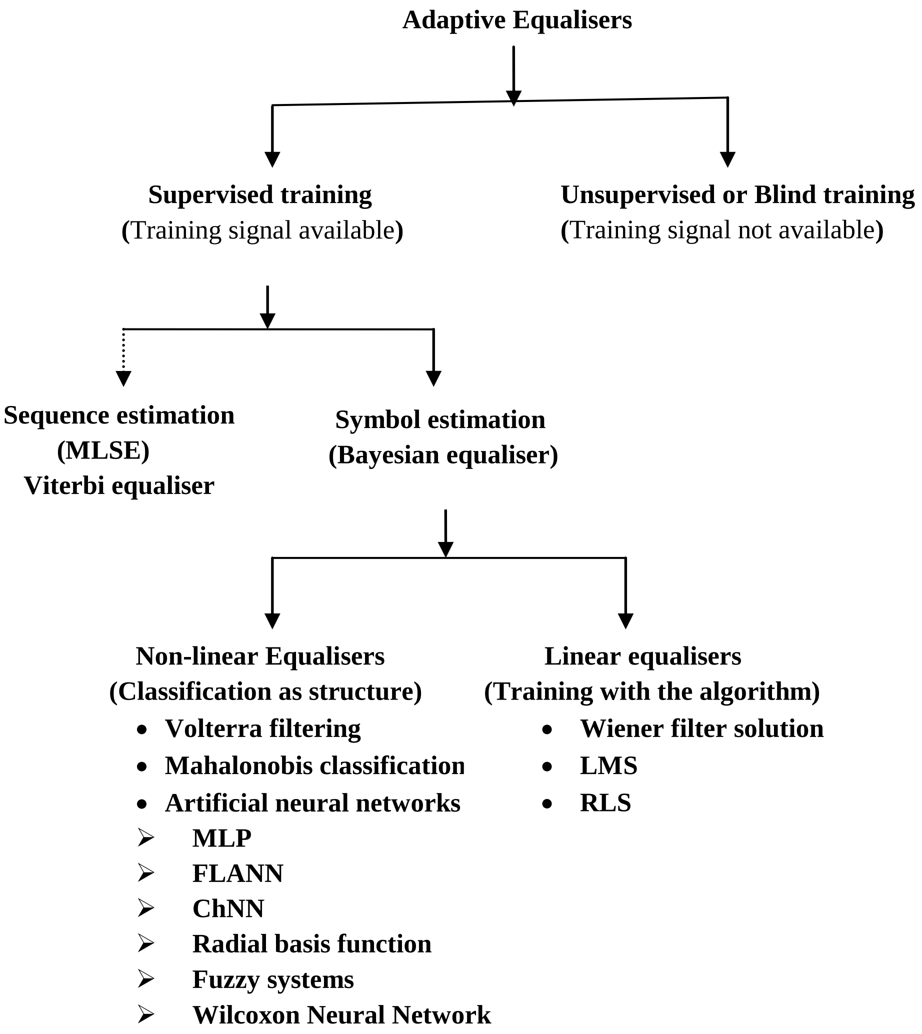


Figure. 2.13 Classification of Adaptive Equalizer

2.7 Optimal symbol-by-symbol equaliser: Bayesian equaliser

The optimal symbol-by-symbol equaliser is termed as Bayesian equaliser. To derive the equaliser decision function the discrete time model of the baseband communication system is presented in Figure 2.13.

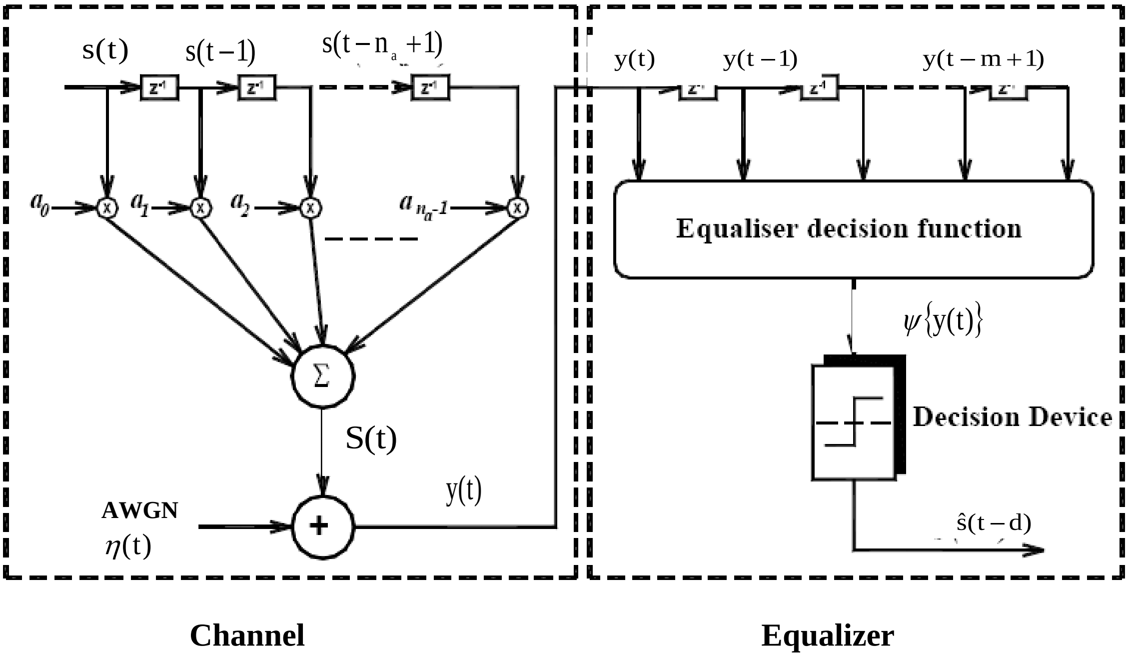


Figure 2.14. Discrete time model of a digital communication system

The equaliser uses an input vector $y(t) \in R^m$, the *m* dimensional space where the term *m* is the feed forward order of the equaliser. The equaliser provides a decision function $\psi\{y(t)\}$ based on the input vector which is passed through a decision device to provide the estimate of transmitted signal $\hat{s}(t-d)$, where d is a delay associated with equaliser decision. The communication system is assumed to be a two level PAM system, where the transmitted sequence $s(t)$ is drawn from an independent identically distributed (i.i.d) sequence comprising of {±1} symbols. Where, a_i are the channel tap values and i is the length of the FIR channel. The noise source is Additive White Gaussian Noise (AWGN) characterised by zero mean and a variance of σ_N^2 .

The received signal y(t) at sampling instant t can be represented as,

$$y(t) = S(t) + \eta(t) = \sum_{i=0}^{n_a-1} a_i s(t-i) + \eta(t) \tag{2.48}$$

The equaliser performance is described by the probability of misclassification w.r.t. Signal to Noise Ratio (SNR). The SNR is defined as,

$$SNR = \frac{\varepsilon[y(t)^2]}{\varepsilon[N(t)^2]} = \frac{\sigma_s^2 \sum_{i=0}^{n_a-1} a_i^2}{\sigma_N^2} \tag{2.49}$$

where, $\mathcal{E}$ is the Expectation operator, σ_s^2 represents the transmitted signal power and $\sum_{i=0}^{n_a-1} a_i^2$ is the channel power. With the assumption that the signal is drawn from an i.i.d. sequence of {±1}, the signal power becomes $\sigma_s^2 = 1$. Hence, the SNR can be represented as,

$$\text{SNR} = 10 \log_{10} (1 / \sigma_N^2) \text{ dB} \tag{2.50}$$

The equaliser uses the received signal vector $y(t) = [y(t), y(t-1), \ldots \ldots, y(t-m+1)]^T \in R^m$ to estimate the delayed transmitted symbol $\hat{s}(t-d)$. The decision device at the equaliser output uses a sgn(x) function Hence, the estimate of the transmitted signal given by the equaliser is

$$\hat{s}(t-d) = \mathbf{sgn}(\psi\{y(t)\}) = \begin{cases} +1 & \text{if} \quad \psi\{y(t)\} \geq 0 \\ -1 & \text{if} \quad \psi\{y(t)\} < 0 \end{cases} \tag{2.51}$$

The performance of an equaliser can be evaluated as follows. For bit error rate (BER) calculation if the equaliser is tested with statistically independent random data sequence of 10^7 channel samples then an error value e_i is generated in the following manner.

$$e_i = \begin{cases} +1 & \text{if} \quad \hat{s}(t-d) = s(t-d) \\ -1 & \text{if} \quad \hat{s}(t-d) \neq s(t-d) \end{cases} \tag{2.52}$$

Then the BER is evaluated in decimal logarithm as

$$\mathbf{BER} = \mathbf{log}_{10}\left(\sum_{i=1}^{10^7} \mathbf{e}_i \,/\, \mathbf{10}^7\right) \tag{2.53}$$

The process of equalisation discussed here can be viewed as a classification process in which the equaliser partitions the input space $\mathbf{y(t)} \in R^m$, into two regions corresponding to each of the transmitted sequence +1 /-1 [25, 26,27]. The loci of points which separate these two regions are termed as the decision boundary. If the received signal vector is perturbed sufficiently to cross the decision boundary due to the presence of AWGN, misclassifications result. To minimise the probability of misclassifications for a given received signal vector $y(t)$, the transmitted symbol should be estimated based on $\mathbf{s(t)} \in \{\pm 1\}$ having a maximum a-posteriori probability (MAP) [28,29]. The partition which provides the minimum probability of misclassification is termed as optimal (Bayesian) decision boundary.

2.7.1 Channel States

The concept of channel states is introduced here. The equaliser input vector has been defined as $y(t) = [y(t), y(t-1), \ldots \ldots, y(t-m+1)]^T \in R^m$, the *m* dimensional observation space. The vector $S(t)$ is the noise free received signal vectors $y(t) = [y(t), y(t-1), \ldots \ldots, y(t-m+1)]^T \in R^m$. Each of these possible noise-free received signal vectors constitutes a channel state. The channel states are determined by the transmitted symbol vector

$$s(t) = [s(t), s(t-1), \ldots \ldots, s(t-m-n_a+2)]^T \in R^{m+n_a-1} \quad (2.54)$$

Here $y(t)$ can be represented as $y(t) = H[s(t)]$, a_i are the channel tap values and i is the length of the FIR channel. where matrix $H \in R^{m \times (m+n_a-1)}$ is the channel matrix which can be expressed as

$$H = \begin{bmatrix} a_0 & a_1 & \cdots & a_{n_a-1} & 0 & \cdots & 0 & \cdots & 0 \\ 0 & a_0 & \cdots & a_{n_a-2} & a_{n_a-1} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \ddots & & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & \cdots & \cdots & a_0 & \cdots & a_{n_a-1} \end{bmatrix} \quad (2.55)$$

Since the channel input sequence $s(t)$ has $N_s = 2^{m+n_a-1}$ combinations, the noise-free channel output vector $S(t)$ has N_s states, which are constructed with N_s sequences of $s(t)$, This can be denoted as,

$$s_j(t) = [s\ (t), s_j(t-1), \ldots \ldots, s_j(t-m-n_a+2)]^T, \qquad 1 \le j \le N_s \quad (2.56)$$

The corresponding channel states of y(t), denoted as $c_{j,}$ are given by

$$c_j = S(t) = \psi[s_j(k)], \qquad 1 \le j \le N_s \quad (2.57)$$

The channels state matrix $C_d = \{c_j\}, \ \ 1 \le j \le N_s$, can be partitioned into two subsets according to the values of the transmitted symbol $s(t-d)$, i.e.

$$C_d = C_d^{(+)} \cup C_d^{(-)} \quad (2.58)$$

Where,

$$C_d^{(+)} = \{S(t) | s(t-d) = +1\}$$
$$C_d^{(-)} = \{S(t) | s(t-d) = -1\}$$

Each set $C_d^{(+)}, C_d^{(-)}$ contains $N_s/2$ channel states. Here the channel states $c_j \in C_d^{(+)}$, are termed the positive channel states and $c_j \in C_d^{(-)}$ are termed the negative channel states.

Example:

An example is considered to show the channel states. The channel considered here is represented by its z-transform,

$$H(z) = H_1(z) = 1 + 0.5z^{-1}$$

This channel is a minimum phase channel. The equaliser length considered here is m=2. This equaliser has Ns =8 channel states. The channel states for this equaliser are presented in Table 2.1 and are located at S(t) with its components taken from scalars $[S(t), S(t-1)]^T$.

No.	$\mathbf{c_j}$	$s(t)$	$s(t-1)$	$s(t-2)$	$S(t)$	
					$S(t)$	$S(t-1)$
1	c_1	1	1	1	1.5	1.5
2	c_2	1	1	-1	1.5	0.5
3	c_3	1	-1	1	0.5	-0.5
4	c_4	1	-1	-1	0.5	-1.5
5	c_5	-1	1	1	-0.5	1.5
6	c_6	-1	1	-1	-0.5	0.5
7	c_7	-1	-1	1	-1.5	-0.5
8	c_8	-1	-1	-1	-1.5	-1.5

Table 2.1: Channel states calculation for channel $H(z) = 1 + 0.5z^{-1}$ with m=2.

2.7.1 Symbol-by-symbol Adaptive Nonlinear Equalisers

Some of the popular forms of nonlinear equalisers are introduced in this section. Nonlinear equalisers treat equalisation as a nonlinear pattern classification problem and provide a decision function that partitions the input space R^m to the number of transmitted symbols. This principle is called as Bayesian equalizers [24] principles. These Bayesian equalizers techniques used like Artificial Neural Networks (ANN) [7], radial basis function (RBF) [8], recurrent network [23], Kalman filters, Fuzzy systems [24, 25] etc for nonlinear signal processing.

2.8 Conclusion

In this chapter the adaptive equalizer trained using gradient based algorithms LMS and RLS has been derived and its BER performance is presented. The channel state diagram of minimum and mixed phase channels is simulated and represented. Other forms of nonlinear equalisers using the ANN and fuzzy techniques have also been introduced. The ANN equalisers and evolutionary approach based equalizers introduced here are used in subsequent chapters for demonstrating the equalization performance of the equalizer in linear and nonlinear interference condition of channels.

Chapter 3

Soft Computing Technique for Channel Equalization

Chapter 3

Soft Computing Technique for Channel Equalization

The beginning of 1980 saw the beginning of development in the field of artificial neural network. Artificial neural networks (ANN) are powerful tools to solve a variety of problems in many complex applications like pattern recognition, function approximation, time series prediction, optimization, associative memory, adaptive channel equalization and control. This chapter discusses the different types of ANN, the need of this in field of communication system.

This chapter is organised as follows. Following this introduction, section 3.1 discusses the basic soft computing techniques. Section 3.2 discusses use of neural network in a wireless communication system. Sub-section 3.2.1 discusses the advantage of neural network in communication field. Section 3.3 discusses the basic concept artificial neural network and its advantages indifferent application field. Section 3.4 discusses the multilayer perceptron network. Section 3.5 discusses the signal layer functional like artificial neural network. Section 3.6 discusses the signal layer Chebyshev artificial neural network with advantages. Section 3.7 discusses the generalized radial basis function neural network. Section 3.8 describes Wilcoxon learning techniques. Sub-section 3.8.1 discusses the Wilcoxon Multilayer Perceptron Neural Network. Sub-section 3.9.2 discusses the Wilcoxon Generalized Radial Basic Function Network techniques. Finally, section 3.9 provides the concluding remarks.

3.1 Soft Computing

Soft computing is a consortium of methodologies that works synergistically and provides flexible information processing capabilities for handling real-life ambiguous situations. It has been observed that simplicity and complexity of systems are relative and many conventional mathematical models are challenging and very productive.

Generally speaking, soft computing techniques resemble biological processes more closely than traditional techniques; these are based on formal logical systems, such as sentential logic and predicate logic. Soft computing techniques are intended to complement each other. Components of soft computing includes neural network (NN), fuzzy system (FS), evolutionary computation (EC) including evolutionary algorithm, Hamony search, swarm intelligence, probability including Bayesian network, Chaos theory, rough sets and signal processing tools such as wavelets.

Each soft computing methodology consists of powerful properties and different advantages, like Neural networks are nonparametric, robust to noise and have a good ability to model highly non-linear relationship, Fuzzy sets provide a natural framework for the process in dealing with uncertainty or imprecise data and Wavelet transform provides a tool to analyze media in the fashion of multi-resolution.

Soft computing techniques also have some restrictions that do not allow their individual application in some cases, because when the input data are large the training times of neural networks are excessive and tedious. The theoretical basis of evolutionary algorithm is weak, especially on algorithm convergence. Rough sets are sensitive to noise and have the NP problems on the choice of optimal attribute reduction and optimal rules.

3.2 Neural Network

The concept neural networks started in the late-1800s as an effort to describe how the human mind performed. These ideas started being applied to computational models with Turing's B-type machines and the perceptron.

Today in general form a neural network is a machine that is designed by using electronic components or is simulated in software on a digital computer. To achieve good performance, neural networks employ a massive interconnection of simple computing cells referred to as 'Neurons' or 'processing units', Hence a neural network viewed as an adaptive machine can be defined as .

A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experimental knowledge and making it available for use. It resembles the brain in two respects:

1. Knowledge is acquired by the network from its environment through a learning process.

2. Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.

The procedure used to perform the learning process is called a learning algorithm, the function of which is to modify the synaptic weights of the network in an orderly fashion to attain a desired design objective. Such an approach is the closest to linear adaptive filter theory, which is already well established and successfully applied in many diverse fields (Widrow and Stearns, 1985; Haykin, 1996). McCulloch and Pitts have developed the neural networks for different computing machines.

3.2.1 Advantage of Neural Network

Neural network information learning processing capabilities make it possible to solve complex problems. The use of neural networks offers the following useful properties and capabilities includes nonlinearity, adaptively, massive parallelism, uniformity of analysis and design, learning ability, generalization ability, input-output mapping, fault tolerance, evidential response, contextual Information, VLSI implementability, distributed representation and computation and neurobiological analogy.

The capability of neural networks marked the modelling of nonlinear adaptive systems which could provide high degree of precision, fault tolerance and adaptability compared to other forms of mathematical modelling [43]. So the artificial neural networks are predominantly used for equalization.

3.3 Artificial Neural network

The late 1980's saw the beginning of development in the field of artificial neural network (ANN) [1]. Artificial Neural Network (ANN) have become a powerful tool for many complex applications including functional approximation, nonlinear system identification, motor control, pattern recognition, adaptive channel equalization and optimization. ANN is capable of performing nonlinear mapping between the input and output space due to its large parallel interconnection between different layers and the nonlinear processing characteristics.

An artificial neuron basically consists of a computing element that performs the weighted sum of the input signal and the connecting weight. The weighted sum is added with the bias called threshold and the resultant signal is passed through a nonlinear activation function. Common types of activation functions are sigmoid and hyperbolic tangent. Each neuron is associated with three parameters whose learning can be adjusted. These are the connecting weights, the bias and the slope of the nonlinear function. For the structural point of view a NN may be single layer or it may be multilayer. As mention in chapter 1 (Literature Survey), we know that how the ANN structure are modified continuously to overcome the drawbacks of the slow convergence rate and complexity of the structure.

In this section represent the different ANN structure we have used for simulation work with details mathematical description, advantage and disadvantages, such as MLP, RBF [13], FLANN [36, 37], ChNN [38, 39] and more recently a rank based statistics approach known as Wilcoxon learning method [40] have been proposed for signals processing application.

3.4 Multilayer Perceptron Network

In 1958, Rosenblatt demonstrated some practical applications using the perceptron. The perceptron is a single level connection of McCulloch-Pitts neurons is called as Single-layer feed forward networks. The network is capable of linearly separating the input vectors into pattern of classes by a hyper plane. Similarly many perceptrons can be connected in layers to provide a MLP network, the input signal propagates through the network in a forward direction, on a layer-by-layer basis. This network has been applied successfully to solve diverse problems.

Generally MLP is trained using popular error back-propagation algorithm. The scheme of MLP using four layers is shown in Figure.3.2. s_i represent the inputs s_1, s_2, ….. , s_n to the network, and y_k represents the output of the final layer of the neural network. The connecting weights between the input to the first hidden layer, first to second hidden layer and the second hidden layer to the output layers are represented by W_i, W_{ji}, W_{kj} respectively. The final output layer of the MLP may be expressed as

$$y_k = \psi_k\left[\sum_{k=1}^{P_2} w_{kj}\,\psi_j\left(\sum_{j=1}^{P_1} w_{ji}\,\psi_i\left\{\sum_{i=1}^{n} w_i\, s_i + b_i\right\} + b_j\right) + b_k\right] \tag{3.1}$$

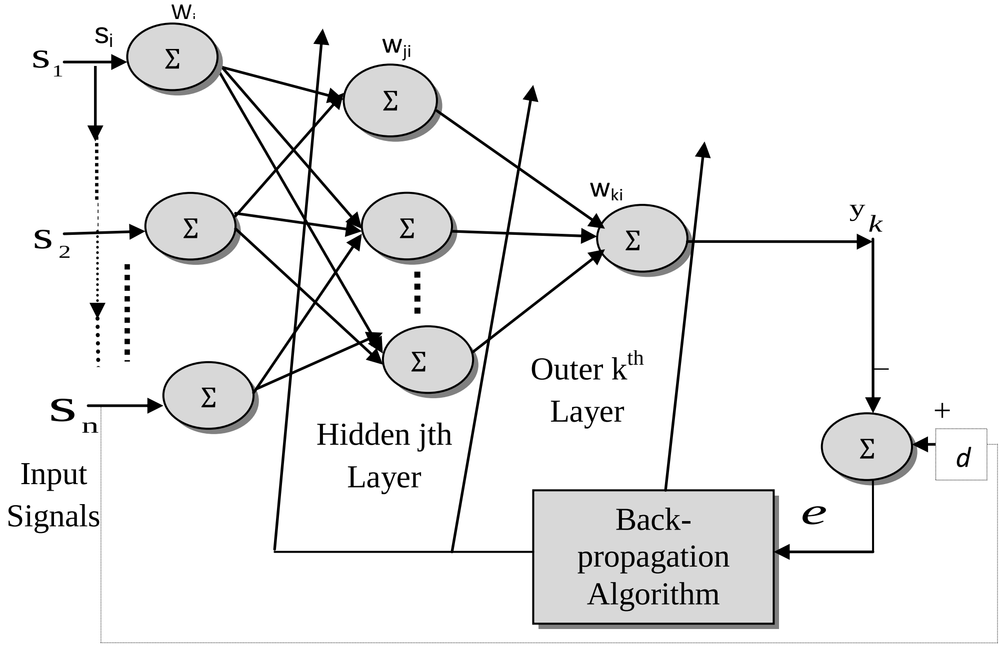


Figure. 3.1. MLP Neural Network using Back-Propagation Algorithm

Where, P_1, P_2 and P_3 are the number of neurons in the layer. b_i, b_j and b_k is the threshold to the neurons of the layer, n is the number of inputs and $\psi(\cdot)$ is the nonlinear activation function respectively. Most popular from of activation functions for signal processing application are sigmoid and the hyperbolic tangent since there are differentiable.

The time index t has been dropped to make the equations simpler. The final output $y_k(t)$ at the output of neuron k, is compared with the desired output $d(t)$ and the resulting error signal $e(t)$ is obtained as

$$e(t) = d(t) - y_k(t) \tag{3.2}$$

The instantaneous value of the total error energy is obtained by summing all error signals over all neurons in the output layer, that is

$$\xi(t) = \frac{1}{2}\sum_{k=1}^{P_3} e^2(t) \tag{3.3}$$

This error signal is used to update the weights and thresholds of the hidden layers as well as the output layer. The updated weights are,

$$w_{kj}(t+1) = w_{kj}(t) + \Delta w_{kj}(t) \tag{3.4}$$

$$w_{ji}(t+1) = w_{ji}(t) + \Delta w_{ji}(t) \tag{3.5}$$

$$w_{i}(t+1) = w_{i}(t) + \Delta w_{i}(t) \tag{3.6}$$

where, $\Delta w_{kj}(t), \Delta w_{ji}(t), \text{and}\, \Delta w_{i}(t)$ are the changes in weights of the second hidden layer-to-output layer, first hidden layer-to-second sub-hidden layer and input layer-to-first hidden layer respectively. That is,

$$\begin{aligned}\Delta w_{kj}(t) &= -2\mu \frac{d\xi(t)}{dw_{kj}(t)} = \mu \mathrm{e}(\mathrm{t}) \frac{dy_k(t)}{dw_{kj}(t)} \\ &= \mu e(t) \psi_k^{'} \left[\sum_{k=1}^{P_2} w_{kj}\, \mathrm{s}_k + b_k \right] \mathrm{s}_k \end{aligned} \tag{3.7}$$

Where, μ is the convergence coefficient ($0 \le \mu \le 1$). Similarly the thresholds of each layer can be updated in a similar manner, i.e.

$$b_k(t+1) = b_k(t) + \Delta b_k(t) \tag{3.8}$$

$$b_j(t+1) = b_j(t) + \Delta b_j(t) \tag{3.9}$$

$$b_i(t+1) = b_i(t) + \Delta b_i(t) \tag{3.10}$$

Where, $\Delta b_k(t), \Delta b_j(t)$ and $\Delta b_i(t)$ are the changes in thresholds of the output, hidden and input layer respectively.

Example. **3.1.** Here we consider the BP algorithm based MLP equalizer for channel equalization application. The BER plot is plotted for different delay of 0, 1, 2 and 3 respectively for below given network

Minimum Phase Channel	Structure of MLP network	No. of Training Samples	No. of Testing Samples	SNR in dB
$H_1(Z) = 1 + 0.5\, Z^{-1}$	3- input nodes 9- hidden nodes 1- output node	1000	100000	30dB

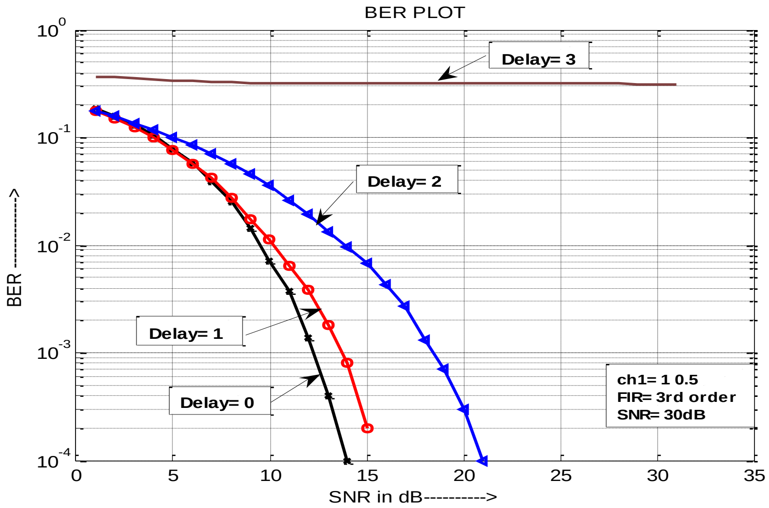


Figure. 3.2 BER Performance of MLP equalizer for Ch_1

3.5 Functional Link Artificial Neural Network

FLANN or Pao-network was originally proposed by Pao [13], which is a novel single layer ANN network in which the original input pattern is expanded to a higher dimensional space using nonlinear functions, which provides arbitrarily complex decision regions by generating nonlinear decision boundaries. The main purpose of enhanced the functional expansion block to used for the channel equalization process.

Each element undergoes nonlinear expansion to form M elements such that the resultant matrix has the dimension of $N{\times}M$. The functional expansion of the element x_k by power series expansion is carried out using the equation given in

$$s_i = \begin{cases} x_k & \text{for } i = 1 \\ x_k^l & \text{for } i = 2,3,4,\ldots,M \end{cases} \tag{3.11}$$

Where, $l = 1,2,\cdots,M$ for trigonometric expansion,

$$s_i = \begin{cases} x_k & \text{for } i = 1 \\ \sin\left(l\pi x_k\right) & \text{for } i = 2,4,\ldots,M \\ \cos\left(l\pi x_k\right) & \text{for } i = 3,5,\ldots,M+1 \end{cases} \tag{3.12}$$

Where, $l = 1,2,\cdots,M/2$. In matrix notation the expanded elements of the input vector E, is denoted by **S** of size $N \times (M+1)$.

The bias input is unity. So, an extra unity value is padded with the **S** matrix and the dimension of the **S** matrix becomes $N \times Q$, where $Q = (M+2)$.

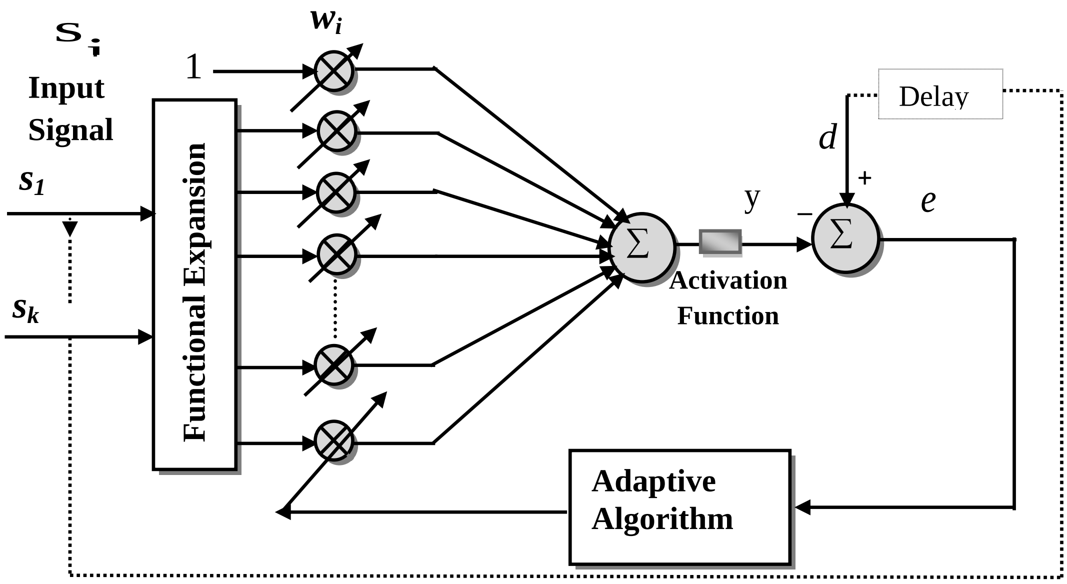


Figure.3.3 Structure of the FLANN model

Let the weight vector is represented as W having Q elements. The output y is given as

$$\mathrm{y} = \sum_{i=1}^{Q} \psi(\mathrm{s_i w_i}) \tag{3.13}$$

In matrix notation the output can be,

$$\mathbf{Y} = \mathbf{S} \cdot \mathbf{W}^{T} \tag{3.14}$$

The time index t has been dropped to make the equations simpler. At t^{th} iteration the error signal $e(t)$ can be computed as

$$\mathrm{e(t) = d(t) - y(t)} \tag{3.15}$$

The weight vector can be updated by least mean square (LMS) algorithm, as

$$\mathrm{W}(t+1) = \mathrm{W}(t) + \mu \mathrm{e}(t)\mathrm{S}(t) \tag{3.16}$$

Where μ denotes the step-size $(0 \le \mu \le 1)$, which controls the convergence speed of the LMS algorithm.

Example 3.2. Here we consider the FLANN equalizer for channel equalization application. For simulation used a structure with details parameter given below, The BER plot is plotted for different delay of 1 respectively.

Mixed-Phase Channel	St**ructure of FLANN** network	No. of Training Samples	No. of Testing Samples	SNR in dB
0.26 + 0.93Z $^{-1}$ **+ 0.26 Z** $^{-2}$	1- input nodes 7- expansional function nodes 1- output node	1000	100000	30dB

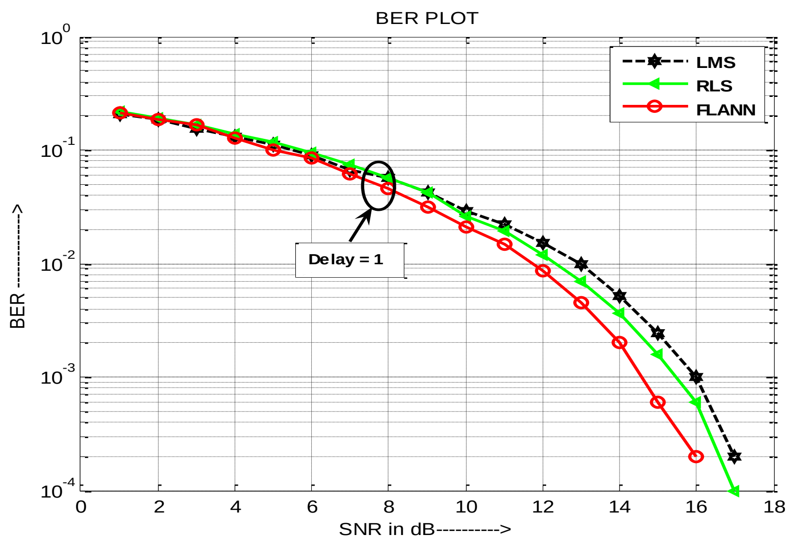


Figure. 3.4 BER Performance of FLANN equalizer compared with LMS, RLS based equalizer for Ch_2.

It is seen from above simulation results that FLANN equalizer perform better than LMS and RLS based equalizer.

3.6 Chebyshev Artificial Neural Network

Chebyshev artificial neural network (ChNN) [12], it is similar to FLANN. The difference being that in a FLANN the input signal is expanded to higher dimension using functional expansion. In Chebyshev the input is expanded using Chebyshev polynomial. The Chebyshev polynomials generated using the recursive formula given as

$$S_{n+1} = 2x\, S_n(x) - S_{n-1}(x) \tag{3.17}$$

The first few Chebyshev polynomials are given as

$$\begin{aligned} S_0(x) &= 1 \\ S_1(x) &= x \\ S_2(x) &= 2x^2 - 1 \\ S_3(x) &= 4x^3 - 3x \end{aligned} \tag{3.18}$$

The weight vector represented as w_i, here i = 0, 1, 2 . . . n. The weighted sum of the components of the enhanced input is passed through a hyperbolic tangent nonlinear function to produce an output $y(t)$. Similarly as FLANN network given in section 3.6 the ChNN weights are updated by LMS algorithm.

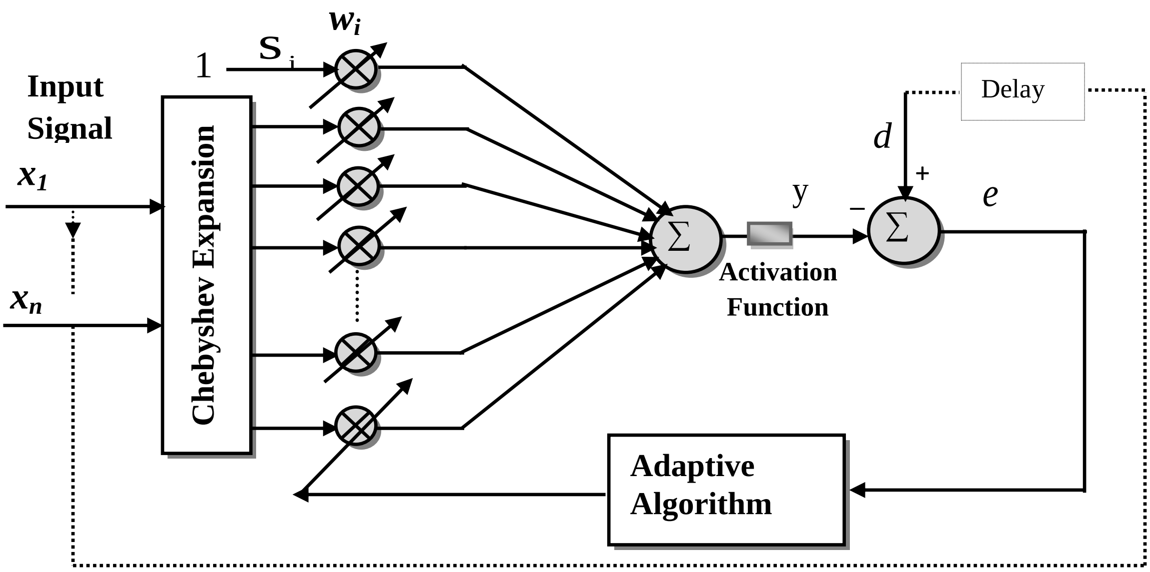


Figure.3.5 Structure of the Chebyshev neural network model

The advantage of ChNN over FLANN is that the Chebyshev polynomials are computationally more efficient than using trigonometric polynomials to expand the input space.

Example 3.4.

Here we consider the ChNN equalizer for channel equalization application. For simulation used the structure is given below. The BER plot is plotted for different delay of 0 and1 respectively.

non-minimum phase channel	**Structure of ChNN** network	No. of Training Samples	No. of Testing Samples	SNR in dB
$0.5 + Z^{-1}$	1- input nodes 5- expansional function nodes 1- output node	1000	100000	30dB

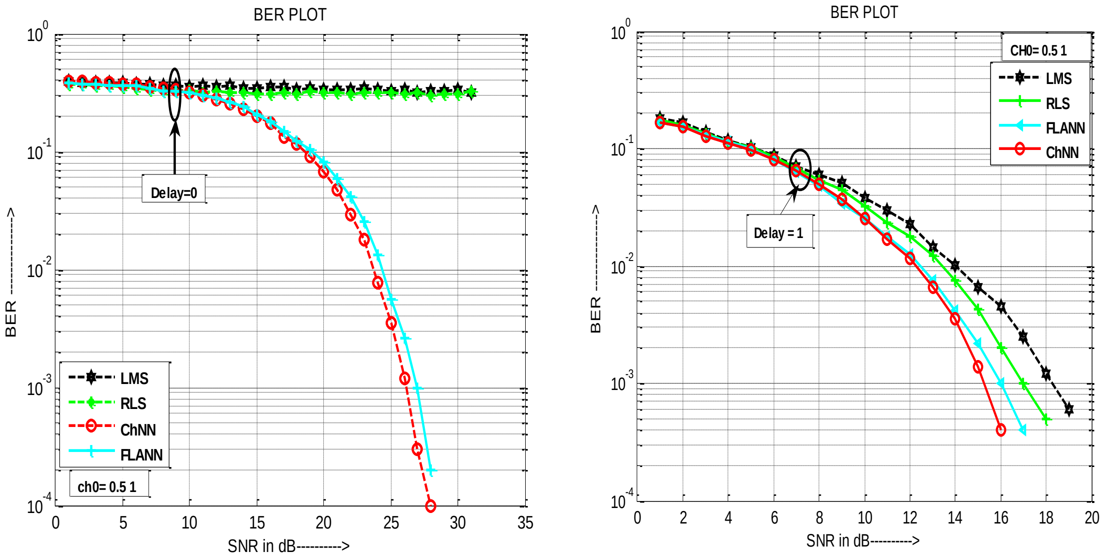


Figure. 3.6 BER Performance of ChNN equalizer compared with FLANN and LMS, RLS based equalizer for ch_0 for delay= 0 and 1.

In terms of results show superior Performance in both linear and nonlinear channel equalizers the BER, MSE floor.

3.7 Radial Basis Function Equalizer

The RBF network was originally developed for interpolation in multidimensional space [6, 7]. The schematic of this RBF network with *m* inputs and a scalar output is presented in Figure.3.8. This network can implement a mapping $F_{rbf} : R^m \rightarrow R$ by the function

$$y = F_{rbf}\{S\} = \sum_{i=1}^{n} w_i \psi\left(\left\| s_i - c_i \right\|\right) \qquad (3.19)$$

Where $S \in R^m$ is the input vector $s_i := [s_1, s_2 \ldots s_n]^T \in R^n$, $\psi(\cdot)$ is the given function from R^+ to R, $w_{i,}\ 1 \le i \le n$ are weights and $c_i \in R^m$ are known as RBF centres. The centres of the RBF networks are updated using k-means clustering algorithm. This RBF structure can be extended for multidimensional output as well. Gaussian kernel is the most popular form of kernel function for equalization application, it can be represented as

$$\psi(y) = \exp\left(\frac{y}{-\sigma_r^2}\right) \tag{3.20}$$

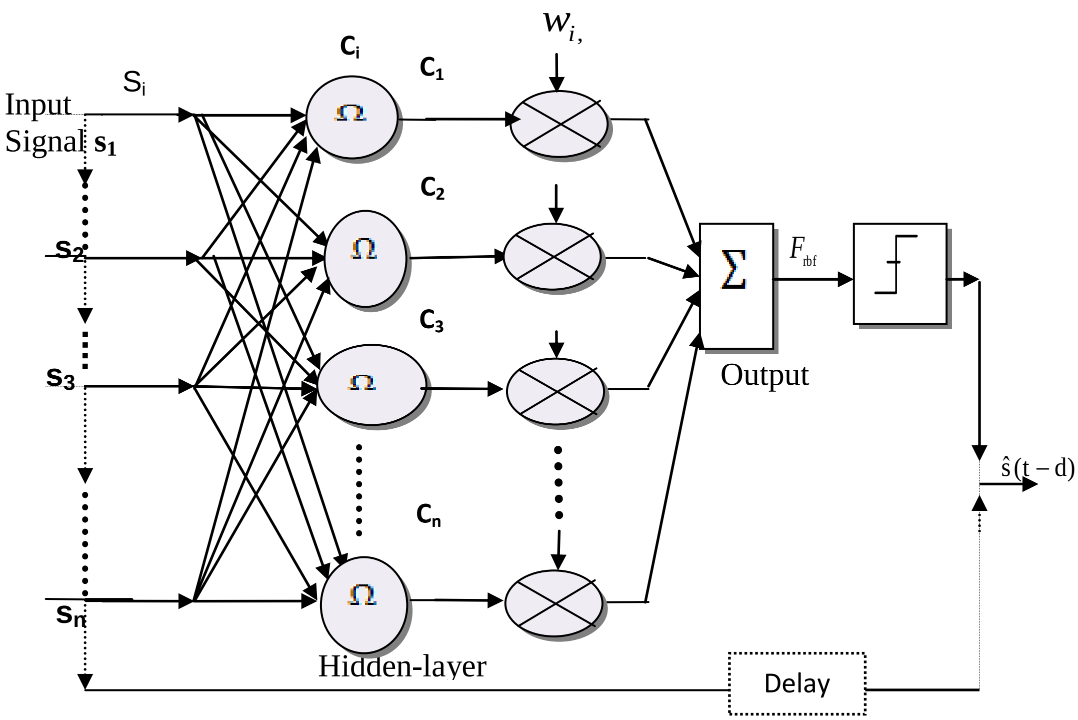


Figure.3.7 Structure of the Radial basis function network equalizer

Here, the parameter σ_r^2 controls the radius of influence of each basis functions and determines how rapidly the function approaches 0 with y. In equalization applications the RBF inputs are presented through a TDL. Training of the RBF networks involves setting the parameters for the centres C_i, spread σ_r and the linear weights ω_i. RBF spread parameter, σ_r^2 is set to channel noise variance σ_η^2 this provides the optimum RBF network as an equaliser. The RBF networks are easy to train since the training of centres, spread parameter and the weights can be done sequentially and the network offers a nonlinear mapping, maintaining its linearity in parameter structure at the output layer.

One of the most popular schemes employed for training the RBF in a supervised manner is to estimate the centres using a clustering algorithm like the k-means clustering and setting

σ_r^2 to an estimate of input noise variance calculated from the centre estimation error. The output layer weights can be trained using popular stochastic gradient LMS algorithm.

The RBF equaliser can provide optimal performance with small training sequences but they suffer from computational complexity. The number of RBF centres required in the equaliser increases exponentially with equaliser order and the channel delay dispersion order. This increases all the computations exponentially.

Example 3.5. Here we consider the RBF equalizer for channel equalization application. For simulation the network details is given below. The BER plot is plotted for different delay of 1 and 2 respectively.

minimum phase channel	**Structure of** network	No. of Training Samples	No. of Testing Samples	SNR in dB
$1+0.5Z^{-1}$	2- input nodes 8- Centres nodes 1- output node	100	100000	30dB

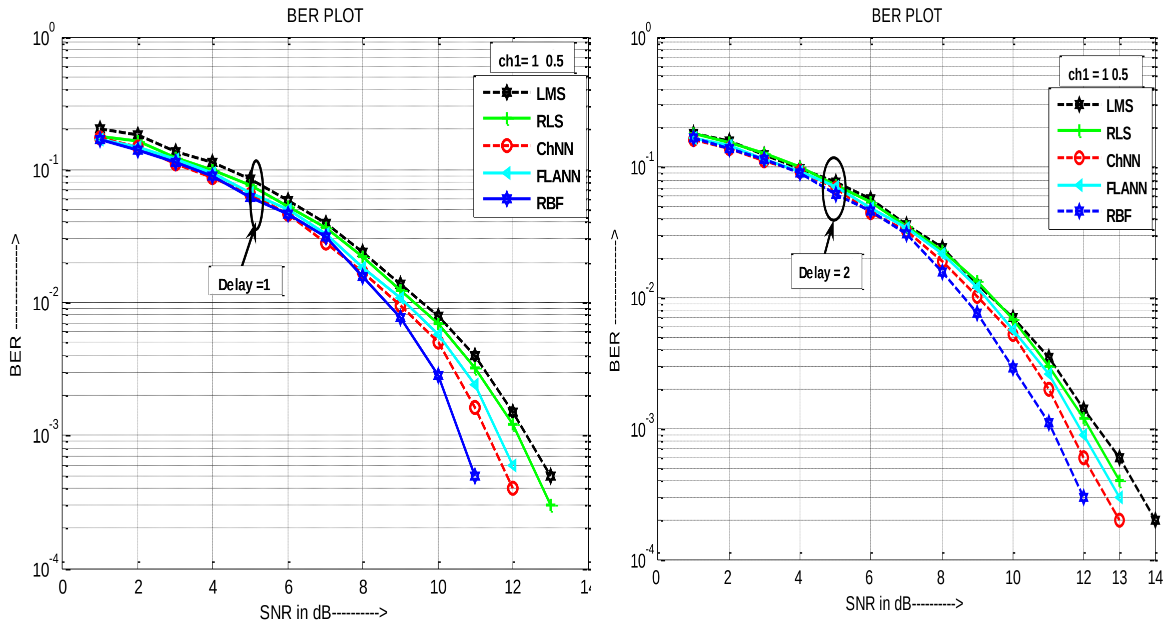


Figure. 3.8 BER Performance RBF equalizer compared ChNN, FLANN, LMS, RLS equalizer for ch_1 for delay=1 and 2.

The Simulation results RBF provided superior Performance from both linear and nonlinear channel equalizers in terms of BER, MSE floor.

3.8 Wilcoxon Learning

Wilcoxon learning [11] is a rank based statistics approach used in linear and nonlinear learning problems. This form of training is robust against outliers. Here the weights and parameters of the network are updated using simple rules based on gradient descent principle. As per Jer-Guang Hesieh, Yih-Lon-Lin and Jyh-Horng Jeng the Wilcoxon provides a promising methodology for many machine learning problems [11]. This motivated us to introduce this learning strategy in the field of Channel Equalization along with ANN.

Here, we investigate two learning machines, namely Wilcoxon Neural Network (WNN), Wilcoxon Generalized Radial Basis Function Neural Network (WGRBFNN). These provide alternative learning machines when faced with general nonlinear problems.

In the Wilcoxon learning machines the Wilcoxon norm of a vector is used as the objective function. To define the Wilcoxon norm of a vector we need a score function $\varphi : [0,1] \rightarrow \mathrm{R}$ i.e. is a function which is not decreasing function is defined as

$$\int_0^1 \varphi^2(\mathrm{u})\,\mathrm{du} < \infty \tag{3.21}$$

The score associated with the score function is defined by

$$\mathrm{a}(\mathrm{i}) = \varphi\left(\frac{\mathrm{i}}{l+1}\right) \qquad \mathrm{i} \in \underline{l} \tag{3.22}$$

Where $\underline{l}$ is a positive integer and as a pseudo-norm function is defined as

$$\begin{aligned} \|\mathrm{v}\|\mathrm{w} &= \sum_{\mathrm{i}=1}^{l} \mathrm{a}(\mathrm{R}(\mathrm{v}_{\mathrm{i}}))\mathrm{v}_{\mathrm{i}} = \sum_{\mathrm{i}=1}^{l} \mathrm{a}(\mathrm{i})\mathrm{v}(\mathrm{i}) \\ v &= [v_1, v_2 \ldots\ldots v_l]^T \in R^l \end{aligned} \tag{3.23}$$

Where, $\mathrm{R}(v_i)$ denotes the rank of v_i among $v_1, \ldots, v_l$, $v_1 \leq \ldots \leq v_l$, are the ordered values of $v_1, \ldots, v_l$, $\mathrm{a}(\mathrm{i}) = \varphi[(\mathrm{i}/(l+1)]$, and $\varphi(\mathrm{u}) = \sqrt{12}\,(\mathrm{u} - 0.5)$. We call $\|v\|\mathrm{w}$ define in equation (3.23) the Wilcoxon norm of the vector v .

3.8.1 Wilcoxon Neural Network

As referring section 3.8, the learning regressor is quite robust against outliers. The Wilcoxon neural network consists of Multilayer Perceptron Neural Network trained with

Wilcoxon learning method, is named as Wilcoxon Multilayer Perceptron Neural Network (WMLPNN) [11]. Consider the neural network as shown in figure.3.10.

For equalization WMLPNN, has one input layer with n+1 nodes, one hidden layer with m+1 nodes, and one output layer with one nodes

$$x = [x_1, x_2 \ldots\ldots x_n]^T \in R^n \quad \mathrm{S} := [\mathrm{s}_1, \mathrm{s}_2 \ldots \mathrm{s}_n, \mathrm{s}_{n+1}]^T = [\mathrm{x}_1, \ldots\ldots \mathrm{x}_n 1]^T \in \mathrm{R}^{n+1} \qquad (3.24)$$

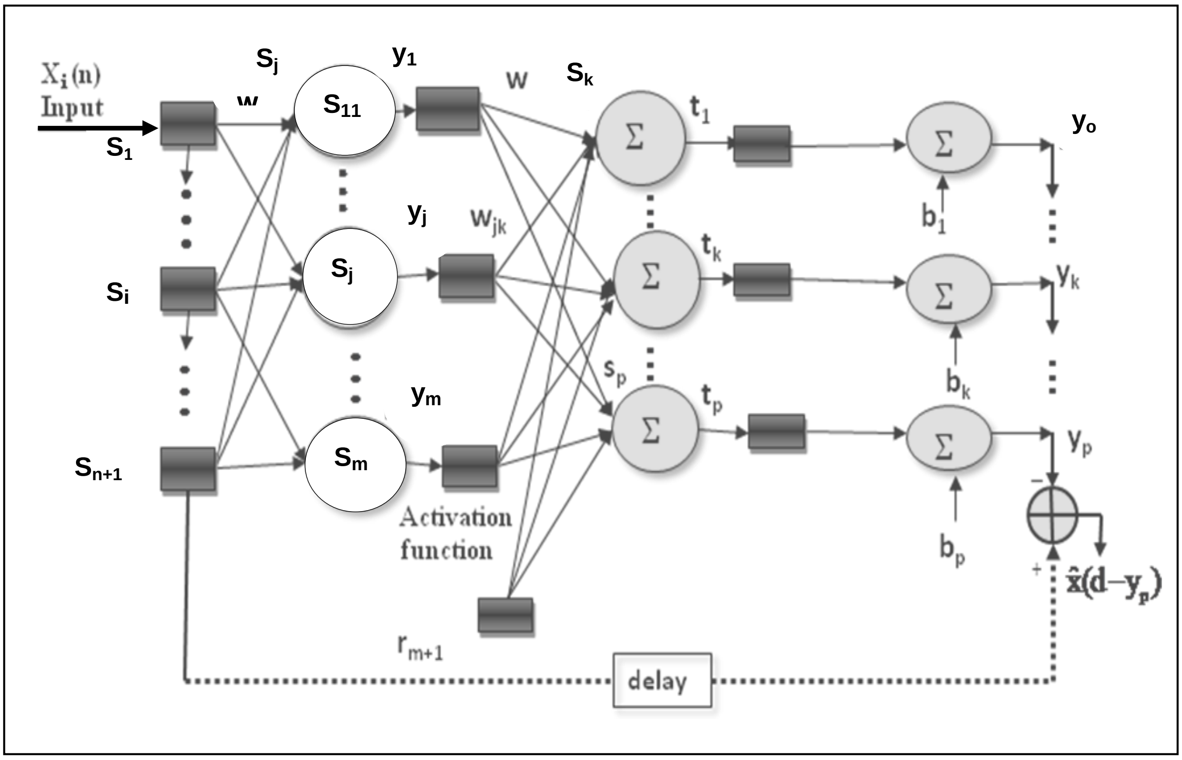


Figure. 3.9 Structure of Wilcoxon MLP neural network

Lets, w_{ji} denote the connection weight from the i^{th} input node to the input of the jth hidden node. Then, the input S_j and the output y_j of the jth hidden node are given by, respectively

$$\mathrm{S}_j = \sum_{i=1}^{n+1} \mathrm{w}_{ji} \mathrm{S}_i \qquad \mathrm{s}_{n+1} := 1 \qquad \mathrm{y}_j = \psi \mathrm{h}_j(\mathrm{S}_j) \qquad \mathrm{j} \in \underline{\mathrm{m}} \qquad (3.25)$$

Where, $\psi \mathrm{h}_j$ is the active function of the jth hidden node. Commonly used activation function is unipolar logistic function

$$\mathrm{y}_j = \psi_{hj}(\mathrm{S}_j) = \frac{1}{1 + \mathrm{e}^{-\mathrm{S}_j}} \qquad (3.26)$$

Let's, w_{kj} denote the connection weight from the output of the jth hidden node to the input of the k^{th} output node. Then, the input S_k and output t_k of the k^{th} output node are given by, respectively

$$\mathrm{S}_{\mathrm{k}} = \sum_{j=1}^{m+1} \mathrm{w}_{\mathrm{kj}} \mathrm{y}_{\mathrm{j}} \qquad \mathrm{r}_{\mathrm{m+1}} := 1$$
$$\mathrm{t}_{\mathrm{k}} = \psi_{\mathrm{ok}}(\mathrm{S}_{\mathrm{k}}) \qquad \mathrm{k} \in \underline{\mathrm{p}} \tag{3.27}$$

Where, ψ_{0k} is the activation function of the k[th] output nodes. Same as MLP equalizer here the output activation functions can be chosen as sigmoidal functions, while for regression problems, the output activation functions can be chosen as linear function with unit slope. The final output of the network is given by

$$\mathrm{y}_{\mathrm{k}} = \mathrm{t}_{\mathrm{k}} + \mathrm{b}_{\mathrm{k}} \qquad \mathrm{k} \in \underline{\mathrm{p}} \tag{3.28}$$

Where, b_k is the bias.

Define all the input parameters used in the input layer are

$$\begin{aligned}
&\mathrm{S} := [\mathrm{s}_1 \ldots \mathrm{s}_m]^T \in \mathrm{R}^m \\
&\mathrm{Y} := [\mathrm{y}_1 \ldots \mathrm{y}_m, \mathrm{y}_{m+1}]^T \in \mathrm{R}^{m+1} \qquad \mathrm{r}_{m+1} = 1 \\
&\mathrm{w}_k := [\mathrm{w}_{k1} \ldots \mathrm{w}_{km}, \mathrm{w}_{k(m+1)}]^T \in \mathrm{R}^{m+1} \\
&\mathrm{W} := [w_{ji}] \in \mathrm{R}^{m \times (n+1)} \\
&\psi_h(\mathrm{S}) := [\psi_{h1}(\mathrm{s}_1), \psi_{h2}(\mathrm{s}_2), \ldots, \psi_{hm}(\mathrm{s}_m) \psi_{h(m+1)}(\mathrm{s})]^T \in \mathrm{R}^{m+1} \\
&\psi_{h(m+1)}(\mathrm{s}) = 1
\end{aligned} \tag{3.29}$$

From equation (3.30)-(3.31), we have

$$\begin{aligned}
&\mathrm{S} = \mathrm{WX} \qquad && \mathrm{Y} = \psi_h(\mathrm{S}) \qquad && \mathrm{s}_k = \mathrm{w}_k^T \mathrm{y}_j \\
&\mathrm{t}_k = \psi_{0k}(\mathrm{s}_k) \qquad && \mathrm{y}_k = \mathrm{t}_k + \mathrm{b}_k \qquad && \mathrm{k} \in \underline{\mathrm{p}}
\end{aligned} \tag{3.30}$$

Let $\mathrm{X} \subseteq \mathrm{R}^n$ and $\mathrm{Y} \subseteq \mathrm{R}^p$, suppose we are given the training set

$$\mathrm{S} := \{(x_q, \mathrm{d}_q)\}_{q=1}^{l} \subseteq \mathrm{X} \times \mathrm{Y} \quad \text{or} \quad \mathrm{S} := \{(\mathrm{x}_q, \mathrm{d}_q)\}_{q=1}^{l} \subseteq \mathrm{R}^{n+1} \times \mathrm{R}^p \tag{3.31}$$

In the following, we will use the subscript q to denote the qth example. For instance, x_{qi} denotes the i[th] component of qth pattern $x_q \in \mathrm{R}^n, \mathrm{q} \in \underline{l}$, and $\mathrm{i} \in \underline{\mathrm{n}}$.

In a WNN, the approach is to choose network weights that minimize the Wilcoxon norm of the total residuals

$$\Psi_{\text{total}} = \sum_{k=1}^{p}\sum_{q=1}^{l} a\left(R\left(p_{qk}\right)\right)p_{qk} = \sum_{k=1}^{p}\sum_{q=1}^{l} a(q)p_{(q)k} \qquad (3.32)$$

$$p_{qk} := d_{qk} - t_{qk} \qquad q \in \underline{l} \qquad k \in \underline{p}$$

The Wilcoxon norm of residuals at the k[th] output node is given by

$$\Psi_k := \|p_k\|w := \sum_{q=1}^{l} a\left(R\left(p_{qk}\right)\right)p_{qk} = \sum_{q=1}^{l} a(q)p_{(q)k} \qquad (3.33)$$

From equation (3.32) and (3.33), we have

$$\Psi_{\text{total}} = \sum_{k=1}^{p} \Psi_k \qquad (3.34)$$

The NN used here is the same as that used in standard ANN, except the bias term at the outputs. The main reason is that the Wilcoxon norm not a usual norm, but a pseudonorm (semi-norm). In particular

$$\| v \| w = 0 \qquad v := [v_1, \ldots.., v_l]^T \in R^l \qquad (3.35)$$

Implies that $v_1 = \ldots = v_l$. This means that, without the bias terms, the resulting predictive function with small Wilcoxon norm of total residuals may deviate from the true function by constant offsets.

Using incremental gradient descent algorithm. In this algorithm Ψ_ks are minimized in sequence represent as

$$\begin{aligned}
\Psi &= \|p_k\|w = \sum_{q=1}^{l} a(q)[d_{(q)k} - t_{(q)k}] \\
&= \sum_{q=1}^{l} a(q)[d_{(q)k} - \psi_{0k}(s_{(q)k})] \\
&= \sum_{q=1}^{l} a(q)[d_{(q)k} - \psi_{0k}(w_k^T y_{(q)j})] \\
&= \sum_{q=1}^{l} a(q)[d_{(q)k} - \psi_{0k}(w_k^T \psi_h(s_{(q)j}))] \\
&= \sum_{q=1}^{l} a(q)[d_{(q)k} - \psi_{0k}(w_k^T \psi_h w_{ji} s_{(q)i}))]
\end{aligned} \qquad (3.36)$$

First, propose an updating rule for the output weights. It is given by

$$w_k \leftarrow w_k - \mu \frac{\partial \Psi_k}{\partial w_k}, \qquad k \in \underline{p} \qquad (3.37)$$

Where, $\mu > 0$ is the learning rate. From equation (3.37), we have

$$\frac{\partial \Psi_k}{\partial w_k} = \sum_{q=1}^{l} a\ (q)(-1)\ \psi'_{ok}\left(S_{(q)k}\right)\ y_{(q)j}$$
$$= -\sum_{q=1}^{l} a\ \left(R_{(p_{qk})}\right)\ \psi'_{ok}\left(S_{qk}\right)\ y_{qj} \tag{3.38}$$

Where $\psi'_{0k}(\cdot)$ denotes the total derivative of $\psi_{ok}(\cdot)$ with respect to its argument and s_{qk} is the kth component of the qth vector s_q. Hence, the updating rule becomes

$$w_{kj} \leftarrow w_{kj} + \mu.\sum_{q=1}^{l} a\ \left(R(p_{qk})\right)\psi'_{ok}\left(s_{qk}\right)y_{qj}, \qquad j \in \underline{m+1} \tag{3.39}$$

Where, y_{qj} is the jth component of the qth vector y_q.

Next, propose an updating rule for the input weights. It is given by

$$w \leftarrow w - \mu \frac{\partial \Psi_k}{\partial w} \tag{3.40}$$

Where,

$$\frac{\partial \Psi_k}{\partial w} = -\sum_{q=1}^{l} a\ (q)\psi'_{ok}\left(s_{(q)k}\right)\begin{bmatrix} w_{k1}\ \psi'_{h1}\left(s_1\right) \\ w_{k2}\ \psi'_{h2}\left(s_2\right) \\ w_{km}\ \psi'_{hm}\left(s_m\right) \end{bmatrix}_{(q)} \times \left[x_1, x_2 \ldots\ldots\ x_n,\ x_{n+1}\right]_{(q)}$$

Hence, the updating rule becomes

$$w_{ji} \leftarrow w_{ji} + \mu.\sum_{q=1}^{l} a\ \left(R\left(p_{qk}\right)\right)\psi'_{ok}\left(s_{qk}\right)\ w_{kj}\ \psi'_{hj}\left(s_{qj}\right)\ s_{qi} \tag{3.41}$$

Where $\psi'_{hj}(\cdot)$ denotes the total derivative of $\psi_{hj}(\cdot)$ with respect to its argument and y_{qj} is the jth component of the qth vector u_q.

The bias term b_k, $k \in \underline{p}$ is given the median of the residuals at the kth output node, i.e,

$$b_k = \underset{1 \le q \le l}{\mathrm{med}}\left\{ d_{qk} - t_{qk} \right\} \tag{3.42}$$

Through extensive simulations study we can observed the performance of WMLPNN equalizer.

Example 3.6. Here we consider the WMLPNN equalizer for channel equalization application. The BER plot is plotted for different delay of 0 and 2 respectively. For simulation the network details is given below.

Minimum Phase Channel	Structure of WMLPNN network	No. of Training Samples	No. of Testing Samples	SNR in dB
$1 + 0.5\,Z^{-1}$	3- input nodes 9- hidden nodes 1- output node	1000	100000	30dB

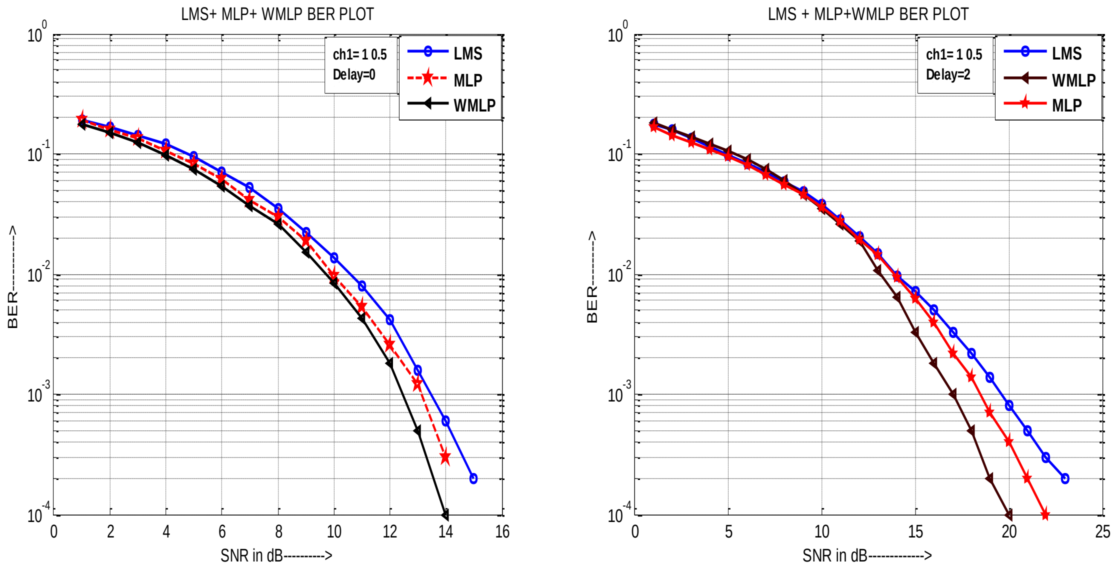


Figure. 3.10 BER Performance equalizer compared MLP and LMS based linear equalizer for ch_1, delay=0 and 2.

From the above simulation results we observed that the WMLPNN equalizer perform well than MLP and linear equalizer.

3.8.2 Wilcoxon Generalized Radial Basis Function Neural Network

Instead of the general GRBFNs, we will consider a commonly used class of approximating functions $g : R^n \rightarrow R^p$ with Gaussian basis functions [41] define as

$$x = [x_1, x_2 \ldots\ldots x_n]^T \in R^n \qquad y = [y_1, y_2, \ldots y_p]^T \in R^p$$
$$s := [s_1, s_2 \ldots s_n]^T = [x_1, \ldots\ldots x_n 1]^T \in R^n \tag{3.43}$$

Then, the predictive function *f* is a nonlinear map given by

$$y_k = F_k(s) = \sum_{j=1}^{m} w_{kj} \exp[-\sum_{i=1}^{n} (s_i - c_{ji})^2 / w_{ji}] + b_k \qquad k \in \underline{p} \qquad (3.44)$$

Where, w_{kj} is the connection weight from the j^{th} hidden node to the k^{th} output, c_{ji} is the center of the j^{th} basis function, w_{ji} is the i^{th} variance of the j^{th} basis function, i.e., $w_{ji} = 2\sigma_{ji}^2 > 0$ and b_k is the bias term.

In this network, there are one input layer with *n* nodes, one hidden layer with *m* node and one output layer with *p* nodes. We have p bias terms at the output nodes.

Define, for $i \in \underline{n}, j \in \underline{m}, k \in \underline{p}$

$$y_j = \sum_{i=1}^{n} (s_i - c_{ji})^2 / w_{ji}, \quad y_j = \exp(-s_j), \quad t_k = \sum_{j=1}^{m} w_{kj} y_j \qquad (3.45)$$

Then, from equation (3.79), we have $y_k = t_k + b_k$

We are given the same training set S as in section (3.10.1). The Wilcoxon norm Ψ_k of residuals at the k^{th} output node is same as defined in section (3.10.1). The incremental gradient descent algorithm requires that $\Psi_k s$ be minimized in sequence. By similar derivations, the weights updating rules are given by

$$w_{kj} \leftarrow w_{kj} - \mu \frac{\partial \Psi_k}{\partial w_{kj}}$$

$$= w_{kj} + \mu . \sum_{q=1}^{l} a\,(R\,(\rho_{qk}))\, y_{qj}, \qquad j \in \underline{m} \qquad (3.46)$$

$$c_{ji} \leftarrow c_{ji} - \mu \frac{\partial \Psi_k}{\partial c_{ji}}$$

$$= c_{ji} + \mu . w_{kj} . \sum_{q=1}^{l} a\,(R(\rho_{qk}))\, y_{qj} \frac{2(s_i - c_{ji})^2}{w_{ji}^2} \qquad (3.47)$$

$$w_{ji} \leftarrow w_{ji} - \mu \frac{\partial \Psi_k}{\partial w_{ji}}$$

$$= w_{ji} + \mu . w_{kj} . \sum_{q=1}^{l} a\,(R(\rho_{qk}))\, y_{qj} \frac{(s_{qi} - c_{ji})^2}{w_{ji}^2} \qquad (3.48)$$

Where $\mu > 0$ is the learning rate.

The bias term b_k, $k \in \underline{p}$ is given the median of the residuals at the k^{th} output node, i.e,

$$b_k = \underset{1 \le q \le l}{\text{med}} \{ d_{qk} - t_{qk} \} \qquad (3.49)$$

Through extensive simulations study we can observed the performance of WRBFNN equalizer.

Example 3.7. Here we consider the WGRBF equalizer for channel equalization application structure details given below. The BER plot is plotted for different delay of 0 and 1 respectively.

minimum phase channel	Structure of network	No. of Training Samples	No. of Testing Samples	SNR in dB
$1+0.5Z^{-1}$	2- input nodes 8- Centres nodes 1- output node	100	100000	30dB

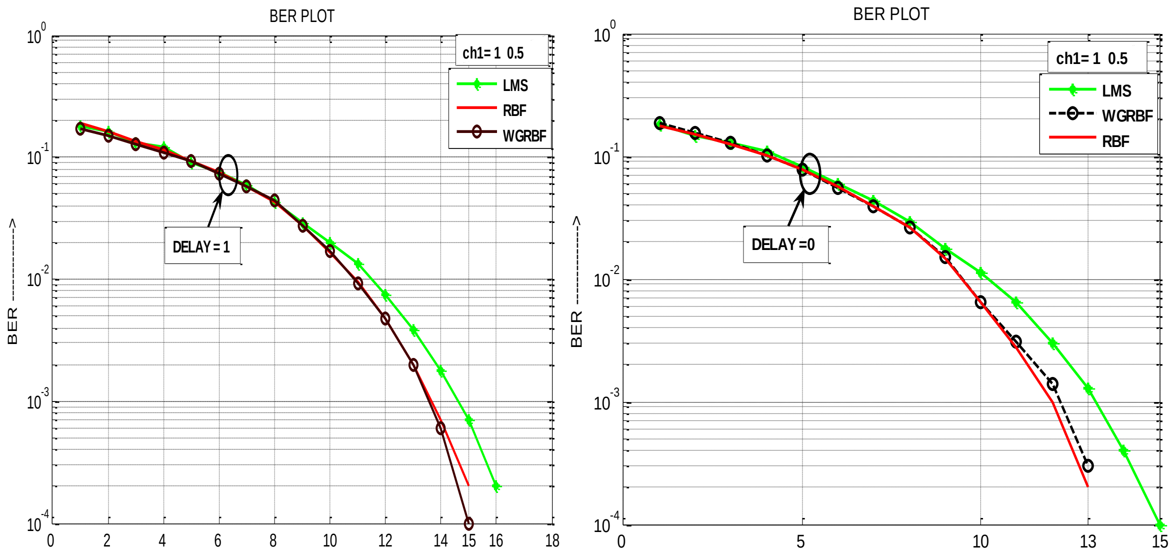


Figure. 3.11 BER Performance WGRBF equalizer compared RBF, LMS based equalizer for ch_1, delay=0 and 1.

From the above simulation results we observed that the WGRBF equalizer perform closer to RBF equalizer.

3.9 Conclusion

This chapter discusses the different form of channel equalization techniques with their structure. MLP network, FLANN, ChNN, RBF and newly approached Wilcoxon learning algorithm for MLP and RBF has been analysed in details. From simulation study we observed that all networks mitigate the digital communication channel destructions with their own pros and cons. Among all these network WGRBF worked same as RBF network providing optimal performance. The performance analysis has been presented in details in chapter 5.

Chapter 4

Evolutionary Approach

Chapter 4

Evolutionary Algorithm Bacterial Foraging Optimization Technique for Channel Equalization

Evolutionary algorithms are stochastic search methods that mimic the metaphor of natural biological evolution [44]. Evolutionary algorithms based approaches are popular method to achieve adaptive channel equalization to minimize the distortion of the communication system. The evolutionary principles have led scientists in the field of "Foraging Theory" to hypothesize that it is appropriate to model the activity of foraging as an optimization process like Bacterial Foraging Optimizations (BFO) [18], Ant-Colony Optimizations (ACO) [26] and Particle Swarm Optimization (PSO) [16, 17]. This optimization technique encourages us to use this algorithm in the channel equalization processes and compared its performance with ANN structure performance.

This chapter is organised as follows. Following this introduction, section 4.1 describes the Evolutionary Algorithms. Section 4.2 describes Different Types of Evolutionary Approaches. Sections 4.3 represent basic principles of Bacterial Foraging Optimization technique with simulation result, at the end section 4.4 presents the Conclusion.

4.1 Evolutionary Algorithms

Evolutionary algorithms are stochastic search methods that mimic the metaphor of natural biological evolution [44]. Evolutionary algorithms operate on a population of potential solutions applying the principle of survival of the fittest to produce better and better approximations to a solution. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and breeding them together using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals that they were created from, just as in natural adaptation.

Evolutionary algorithms model natural processes, such as selection, recombination, mutation, migration, locality and neighbourhood. Figure.4.1 presents the process of working simple evolutionary algorithm. Evolutionary algorithms work on populations of individuals instead of single solutions. In this way the search is performed in a parallel manner.

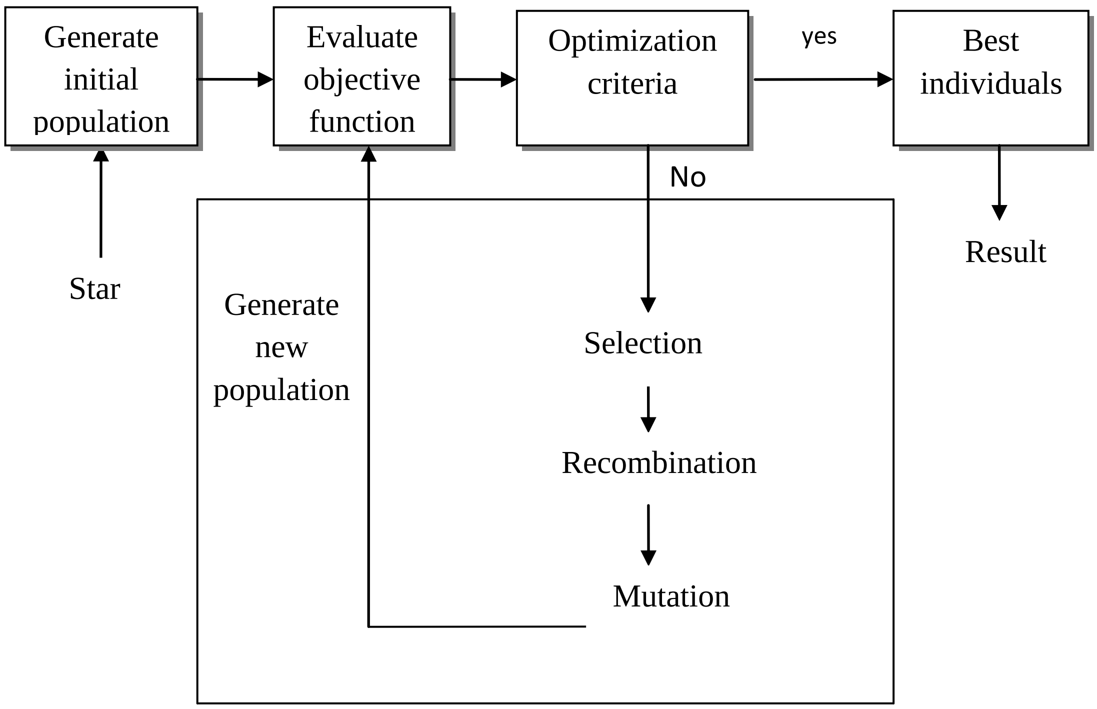


Figure 4.1 Structure of a single population evolutionary algorithm

At the beginning a number of individuals (the population) are randomly initialized. The objective function is then evaluated for these individuals and the initial generation is produced. If the optimization criteria are not met the creation of a new generation starts. Individuals are selected according to their fitness for the production of offspring. Parents are recombined to produce offspring. All offspring will be mutated with a certain probability. The fitness of the offspring is then computed. The offspring are inserted into the population replacing the parents, producing a new generation. This cycle is performed until the optimization criteria are reached.

Such a single population evolutionary algorithm is powerful and performs well on a wide variety of problems. However, better results are obtained by introducing multiple subpopulations. Every subpopulation evolves over a few generations isolated before one or more individuals are exchanged between the subpopulation.

Evolutionary algorithms differ substantially from traditional search and optimization methods. The most significant differences are:

- Evolutionary algorithms search a population of points in parallel, not just a single point.
- Evolutionary algorithms do not require derivative information or other auxiliary knowledge; only the objective function and corresponding fitness levels influence the directions of search.
- Evolutionary algorithms use probabilistic transition rules, not deterministic ones.
- Evolutionary algorithms are generally more straightforward to apply, because no restrictions for the definition of the objective function exist.
- Evolutionary algorithms can provide a number of potential solutions to a given problem. The final choice is left to the user

4.2 Different Types of Evolutionary Approaches

- **Genetic algorithm** - This is the most popular type of EA. One seeks the solution of a problem in the form of strings of numbers (traditionally binary, although the best representations are usually those that reflect something about the problem being solved - these are not normally binary), virtually always applying recombination operators in addition to selection and mutation.
- **Evolutionary programming** - Like genetic programming, only the structure of the program is fixed and its numerical parameters are allowed to evolve.
- **Evolution strategy** - Works with vectors of real numbers as representations of solutions, and typically uses self-adaptive mutation rates.
- **Genetic programming** - Here the solutions are in the form of computer programs, and their fitness is determined by their ability to solve a computational problem.
- **Learning classifier system** - Instead of a using fitness function, rule utility is decided by a reinforcement learning technique.

The differential evolution based on vector differences and is therefore primarily suited for numerical optimization problems.

- **Particle swarm optimization** - Based on the ideas of animal flocking behaviour. Also primarily suited for numerical optimization problems.
- **Ant colony optimization** - Based on the ideas of ant foraging by pheromone communication to form path. Primarily suited for combinatorial optimization problems.
- **Bacterial foraging** - Based on the ideas of bacteria foraging by swimming and tumbling. Primarily suited for combinatorial optimization problems.

4.3 Basic Bacterial Foraging Optimization

Natural selection tends to eliminate animals with poor foraging strategies and favour the propagation of genes of those animals that have successful foraging strategies, since they are more likely to enjoy reproductive success. After many generations, poor foraging strategies are either eliminated or shaped into good ones. This activity of foraging led the researchers to use it as optimization process. The *E. coli* bacteria that are present in our intestines also undergo a foraging strategy. The control system of these bacteria that dictates how foraging should proceed can be subdivided into four sections, namely, chemotaxis, swarming, reproduction, and elimination and dispersal [45].

For initialization, we must choose the parameter for optimization are represent as

P = Dimension of search space

S = Number of bacteria to be used for searching the total region

N_c = Number of chemotactic steps,

N_s = Number of reproduction steps,

N_{re} = Swim length after which tumbling of bacteria will be under taken in a chemotatic loop,

C(i) = Step size,

N_{ed} = Number of elimination and dispersal events to be imposed over the bacteria.

P_{ed} = Probability of elimination and dispersal event will continue,

θ^i = Initial values for i^{th} bacterium position ,

In case of swarming, we will also have to pick the parameters of the cell-to-cell attractant functions; here we will use the parameters given above. Also, initial values for the θ^i, i = 12….S must be chosen. Choosing these to be in areas where an optimum value is likely to exist is a good choice. Alternatively, we may want to simply randomly distribute them across the domain of the optimization problem. The algorithm that models bacterial population chemotaxis, swarming, reproduction, elimination, and dispersal is given here (initially j=k=l=0). For the algorithm, note that updates to the θ^i automatically result in updates to P. Clearly, we could have added a more sophisticated termination test than simply specifying a maximum number of iterations.

- **Chemotaxis Stage**

This process in the control system is achieved through swimming and tumbling via Flagella. Each flagellum is a left-handed helix configured so that as the base of the flagellum (i.e., where it is connected to the cell) rotates counter clockwise, as viewed from the free end of the flagellum looking toward the cell, it produces a force against the bacterium so it pushes the cell. On the other hand, if they rotate clockwise, each flagellum pulls on the cell, and the net effect is that each flagellum operates relatively independently of others, and so the bacterium tumbles about. Therefore, an *E. coli* bacterium can move in two different ways; it can run (swim for a period of time) or it can tumble, and alternate between these two modes of operation in the entire lifetime. To represent a tumble, a unit length random direction $\phi(j)$ is generated, this will be used to define the direction of movement after a tumble. In particular

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + c(i)\phi(j) \tag{4.1}$$

Where θ^i (j+1, k, l) represents the i^{th} bacterium at j^{th} chemotactic k^{th} reproductive and l^{th} elimination and dispersal step. C(i) is the size of the step taken in the random direction specified by the tumble (run length unit). Lets, N_{is} signal samples are passed through the model. The output compared with the desired signal to calculate the error as

$$j = \sum_{k=1}^{N_{is}} e^2(k) = \sum_{k=1}^{N_{is}} \left[y(k) - \hat{y}(k) \right]^2 \tag{4.2}$$

This is the objective function of the BFO. We need to minimize the error square by using this technique in channel equalization.

- **Swarming Stage**

When a group of *E. coli* cells is placed in the center of a semisolid agar with a single nutrient chemo-effecter (sensor), they move out from the center in a traveling ring of cells by moving up the nutrient gradient created by consumption of the nutrient by the group. Moreover, if high levels of succinate are used as the nutrient, then the cells release the attractant aspartate so that they congregate into groups and, hence, move as concentric patterns of groups with high bacterial density. The spatial order results from outward movement of the ring and the local releases of the attractant; the cells provide an attraction signal to each other so they swarm together. The mathematical representation for swarming can be represented by

$$\begin{aligned} J_{cc}(\theta, P(j,k,l)) &= \sum_{i=1}^{S} J_{cc}^{i}(\theta, \theta^{i}(j,k,l)) \\ &= \sum_{i=1}^{S} [-d_{attract} \exp(-w_{attract} \sum_{m=1}^{p} (\theta_m - \theta_m^i)^2)] \\ &\quad + \sum_{i=1}^{S} [h_{repellent} \exp(-w_{repellent} \sum_{m=1}^{p} (\theta_m - \theta_m^i)^2)] \end{aligned} \tag{4.3}$$

Where, $J_{cc}(\theta, P(j,k,l))$ is the cost function value to be added to the actual cost function to be minimized to present a time varying cost function, S is the total number of bacteria, P is the number of parameters to be optimized which are present in each bacterium, and $d_{attarct}, w_{attract}, h_{repellent}, w_{repellent}$ are different coefficients that are to be chosen properly. Let $J_{last} = J(i,j,k,l)$ to save this value since we may find a better cost via a run.

For $i = 1,2,3.....S$. The tumbling/ swimming decision is taken. A random vector $\Delta(i)$, with each element $\Delta_m(i)$ generated random number. Its generates random number in the range of [-1, 1]. Check whether $\theta^i(j+1,k,l) < \theta^i(j,k,l)$ then change the current position of the bacteria to a new position using the equation

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + c(i) \times \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{4.5}$$

Unit counter < N_s, else tumble. If j < N_c then go to the chemotaxis loop otherwise go to reproduction step and population Sorted in ascending order of cost function.

- **Reproduction Stage**

For the given k and l, and for each i =1, 2…..S.

let $J_{health}^{i} = \sum_{j=1}^{N_c+1} J(i, j, k, l)$ be the health of bacterium i (a measure of how many nutrients it got over its lifetime and how successful it was at avoiding noxious substances). Sort bacteria and chemotactic parameters $C(i)$ in order of ascending cost J_{health} (higher cost means lower health). The S_r bacteria with the highest J_{health} values die, i.e., the

$s_r = \frac{s}{2}$ least healthy bacteria die and the other healthier bacteria each split into two bacteria, which are placed in the same location. This makes the population of bacteria constant.

- **Elimination and Dispersal Stage**

It is possible that in the local environment, the lives of a population of bacteria changes either gradually (e.g., via consumption of nutrients) or suddenly due to some other influence. Events can occur such that all the bacteria in a region are killed or a group is dispersed into a new part of the environment. They have the effect of possibly destroying the chemotactic progress, but they also have the effect of assisting in chemotaxis, since dispersal may place bacteria near good food sources. From a broad perspective, elimination and dispersal are parts of the population-level long-distance motile behaviour. This section is based on the work in [45].

Elimination-dispersal: For i = 1, 2… S .With probability P_{ed} , eliminate and disperse each bacterium (this keeps the number of bacteria in the population constant). To do this, if you eliminate a bacterium, simply disperse one to a random location on the optimization domain.

If $l < N_{ed}$ then go to $l = l+1$ elimination and dispersal loop, otherwise end.

Through extensive simulations study we can observe the performance of adaptive equalizer trained using BFO algorithm.

Example 4.1 Here we consider the BFO equalizer for channel equalization application. For simulation the network details is given below. The BER plot is plotted for different delay of 0, 1 respectively.

Mixed phase channel	**Structure of BFO equalizer**	No. of Training Samples	No. of Testing Samples	SNR in dB
$0.30 + 0.90Z^{-1} + 0.30 Z^{-2}$	No. of Bacteria – 20 No. of chemotactic steps – 10 No. of reproduction steps – 3 Dimension of search space – 3 Swim length after tumbling – 20 No. of elimination and dispersal – 5 Probability of elimination & dispersal- 0.01 Step size – 0.01	100	100000	30dB

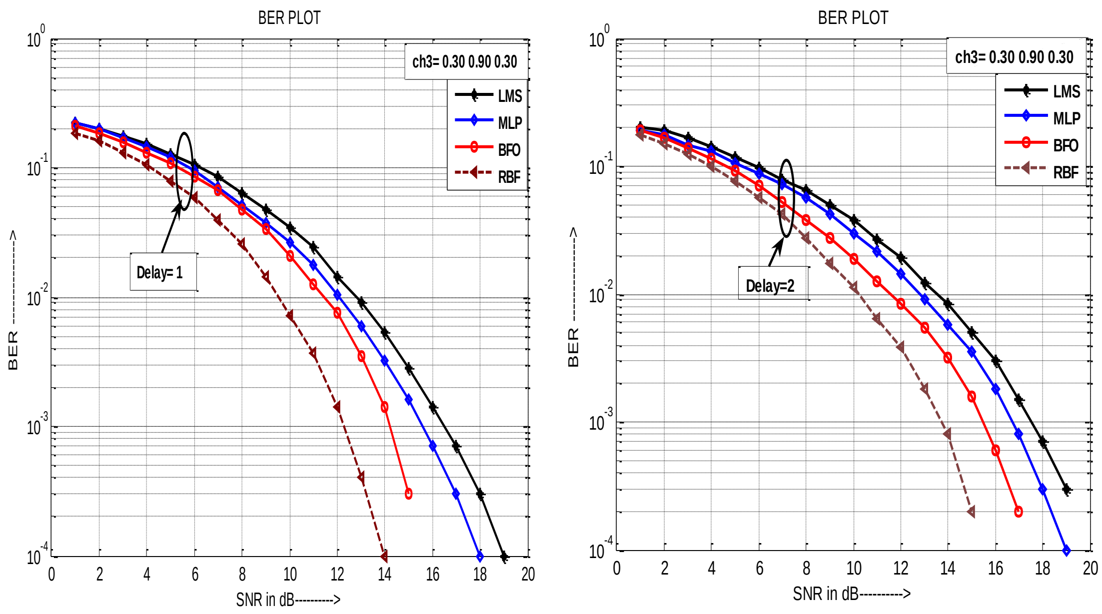


Figure. 4.2 BER Performance BFO trained linear equalizer compared with RBF, MLP and LMS equalizer for ch_3, delay= 1 and 2.

From the above simulation results we observed that the BFO equalizer perform better than linear equalizer and MLPNN, but lower performance as compared to RBF equalizer.

4.4 Conclusion

This chapter introduces the concept of the ‘Evolutionary Algorithms’ with more emphasis on ‘Bacterial Foraging’. The basic ‘Bacterial Foraging Algorithm’ is explained. Some simulation results have also been presented to validate the efficacy of the proposed algorithms. The performance analysis has been presented in details in chapter 5.

Chapter 5

Results & Discussion

Chapter 5

Results & Discussion

This chapter demonstrates the performance of linear and nonlinear equalizer, with the extensive simulation. The performance parameters that have been used in the simulations include convergence characteristic, bit error rate performance, structure and computational complexcity. These have been investigated for a wide variety of channel conditions. A wireless communication system is affected by inter-symbol interference, co-channel interference in the presence of additive white Gaussian noise, many times the signal is also affected by the burst noise. Burst noise can be modelled as a series of finite-duration Gaussian noise pulses of fixed duration and Poisson occurrence times. Adaptive equalization techniques have also been used to mitigate these effects and results presented here. Performance of ANN, RBF, FLANN, ChNN, WMLP and WGRBF equalizer have been analysed for equalization in a variety of channel conditions. There performances have been compared with linear equalizers trained with LMS and RLS algorithms. Additionally simulation study has also been done for BFO based training with linear equalizer structure.
The transmitted signal s(t) in all tests were generated randomly from an independent identically distributed (i. i. d) sequence. The equalizers were trained with 1000 samples of training data. The convergence characteristics of equalizers were observed through MSE plot with respect to iteration. The BER performance provides the actual performance of the equalizer. This was computed using 100,000 samples for each signal to noise ratio condition.

For simulation studies five different distortion conditions have been considered. The distortion conditions as under

1. Channel with ISI in presence of AWGN.
2. Channel with ISI in presence of Burst noise and AWGN.
3. Channel corrupted with ISI, nonlinearity of ISI and AWGN.
4. Channel corrupted with ISI, CCI in presence of AWGN.

5. Channel corrupted with ISI, CCI in presence of Burst noise and AWGN.

Following this section, in section 5.1 discusses the channels models used for simulation studies. Section 5.2 discusses the simulation results of all ANN and linear based equalizer for channels were distorted by ISI. Section 5.3 discusses the simulation results of all ANN and linear based equalizer for channels were distorted by nonlinearity and ISI. Section 5.4 discusses the simulation results of all ANN and linear based equalizer for channels were distorted by ISI and burst noise interference. Section 5.5 discusses the simulation results of all ANN equalizer for channels were distorted by ISI and CCI. Section 5.6 discusses the simulation results of all ANN equalizer for channels were distorted by ISI, CCI and Burst noise interference, subsequently in all section subsection consists of comparison study among all equalizer we consider for simulation. In the last Section 5.7 present the conclusion and remark. All the programs are written in Matlab ver. 7.1.

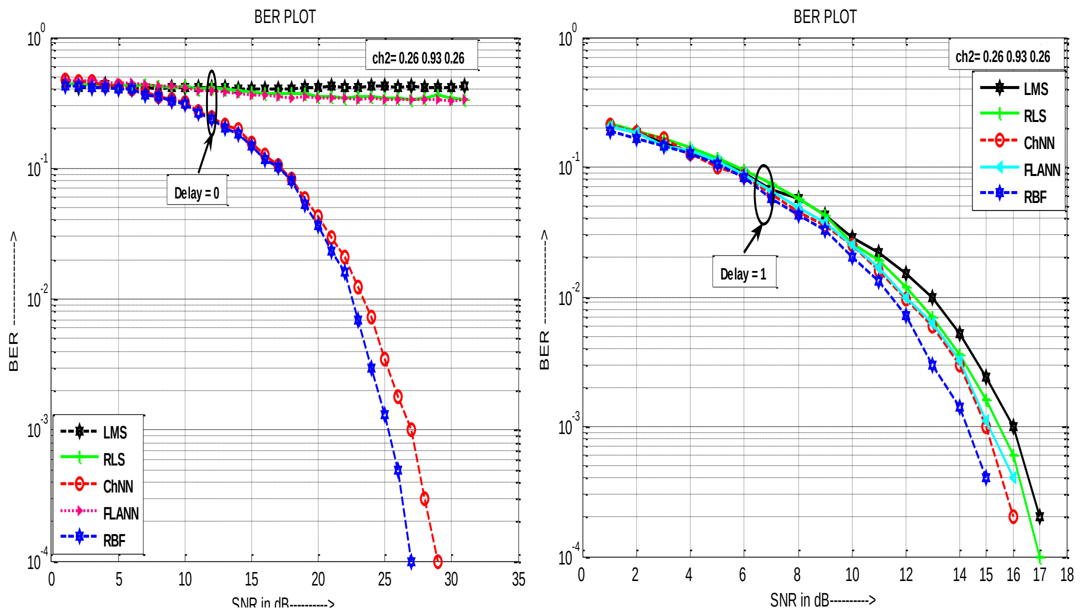
5.1. Performance analysis of equalizers for ISI channels

Here the performances of the different equalizers in terms of convergence rate and bit error rate have been analysed. The channels tested are presented at Annexure-1. The equalizers were trained with 1000 samples of training data. Transmitted uniformly distributed bipolar random numbers {-1, +1}. The training samples were passed through the channel and AWGN was added to the output of the channel. For mathematical convenience, the received signal power was normalised to unity. Thus the received signal to noise ratio (SNR) is simply the reciprocal of the noise variance at the input of the equaliser. For bit error rate (BER) performance calculation 100,000 samples were consider for each signal to noise ratio. The BER plot were analysed for different equalizers. The experimental simulation results are presented below.

5.1.1 Performance analysis of ChNN and FLANN equalizer

Example 5. 1.

For this, we present the performance result for a mixed phase channel, whose transfer function is $H_2(Z) = 0.26 + 0.93Z^{-1} + 0.26Z^{-2}$. ChNN equalizer consists of single input, four different Chebyshev polynomial functions in functional expansion block. The FLANN equalizer consists of single input, Seven different trigonometric function including power series function in functional expansion block. There were compared FIR filter with presented 5-tap for LMS and RLS equalizer. The BER plot is plotted for different decision delay 0 and 1 respectively and is presented if figure 5.1.



(Delay= 0) (Delay= 1)

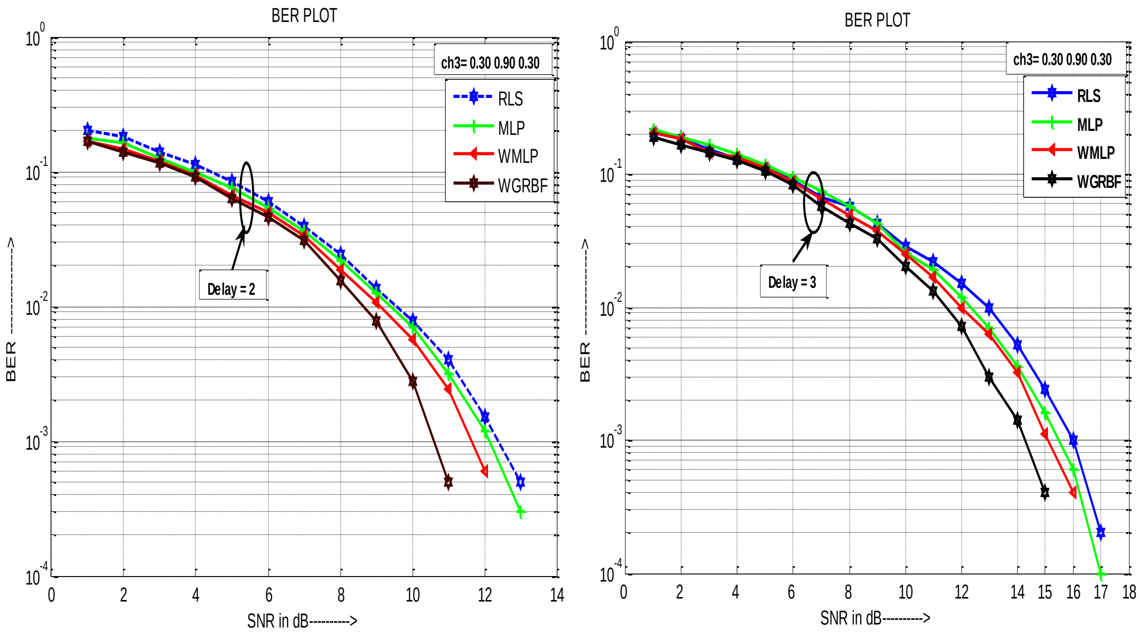
Figure. 5.1 BER performance of ChNN, FLANN compared with RBF and LMS, RLS based linear equalizer for ch_2.

The performance of FLANN and ChNN equalizers were also compared with RBF equalizer of 2nd order, which provides MAP decision performance. From above simulation we observed that ChNN equalizer provides better performance than FLANN, LMS and RLS based equalizer in terms of bit error rate over a wide range of channel conditions. But RBF provides superior performance.

5.1.2 Performance analysis of WMLPNN and MLP equalizer

Example 5. 2.

For this, we present the performance result for a mixed phase channel, whose transfer function is $H_3(Z) = 0.30 + 0.90Z^{-1} + 0.30\,Z^{-2}$. Both the equalizer structure consists of 3 input, 30 hidden nodes and 1 output nodes. But RBF and RLS equalizer consists of 3-tap FIR filter. The MSE and BER plot is plotted for different delay 2 and 3 respectively.



(Delay= 2) (Delay= 3)

Figure.5.2. BER performance of MLPN & WMLPNN equalizer compared with RBF and RLS based linear equalizer for $ch_{3.}$

From above simulation we observed that the proposed WMLPNN equalizer provides better performance than as MLP and RLS equalizer. This equalizer even outperforms MLP equalizer. The WGRBF equalizer outperforms both form of MLP equalizer.

5.1.1 Performance analysis of WGRBF and RBF equalizer

Example 5.3.

Here we analyse the performance of the channel is a mixed phase channel with transfer function is $H_2(Z) = 0.26 + 0.93Z^{-1} + 0.26\ Z^{-2}$.

Both the equalizer consists of structure is 2 input, 16 centres, 1 output. The input is provided through a TDL. The BFO and LMS equalizer consists of 3-tap FIR filter. The MSE and BER plot for the equalizer is plotted for different delay 0 and 1 respectively.

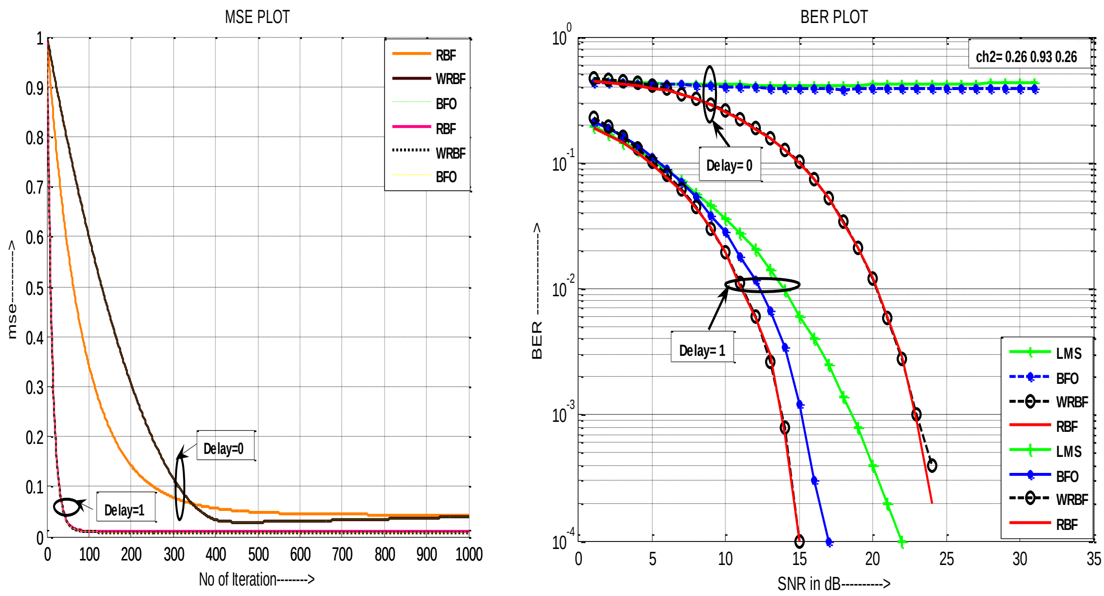


Figure.5.3. MSE & BER performance of RBFN & WGRBFN equalizer compared with BFO and LMS trained linear equalizer for ch_2, Delay= 0 and1.

This channel is a mixed mode channel, where the zeros are present inside and outside the unit circle. The analysis of the decision boundary of this channel provided by an optimal equalizer with different decision delay has been analysis in [24], hence form this it is seen that at decision delay zero the decision boundary is highly nonlinear and is nearly linearly at delay equal to one, for this reason the MSE curve provides better performation for delay one compared to delay zero. Similar performance is obtained in BER performance as well. Since linear equalizer provide a linear decision boundary the LMS and RLS based equalizer completely failed at delay zero.

The WGRBF proposed here provides MSE performance same as RBF and performs better than BFO trained linear equalizer. Similarly its BER performance is close to RBF equalizer and superior to linear structures including BFO trained linear equalizer.

5.2. Performance analysis of equalizers for channels with ISI and Burst noise interference

In the next study the performance of equalizers discussed were evaluated for a burst noise channel. The parameters taken for the simulation were same as those taken for channels with ISI. The burst noise added to the main channel is a heigh intensity noise which occuring for short duration of time. The duration of noise is fixed burst length means a series of finite-duration Gaussian noise pulses. The burst noise was added to only with 5% of the samples. The noise was added to 5 consecutive samples in every 100 samples. The location of these 5 consecutive samples was considered randomly.

5.2.1 Performance analysis of WGRBF and RBF equalizer

Example 5.4.

For a burst noise channel a mixed phase channel was used the transfer function is $H_2(Z) = 0.26 + 0.93Z^{-1} + 0.26\ Z^{-2}$.Both the equalizer consists of structure is 2 input, 16 centres, 1 output. The performance was compared with LMS equalizer consisting of 3-tap FIR filter. The convergence MSE and BER plot for the equalizer is plotted for delay 0 and 1 respectively

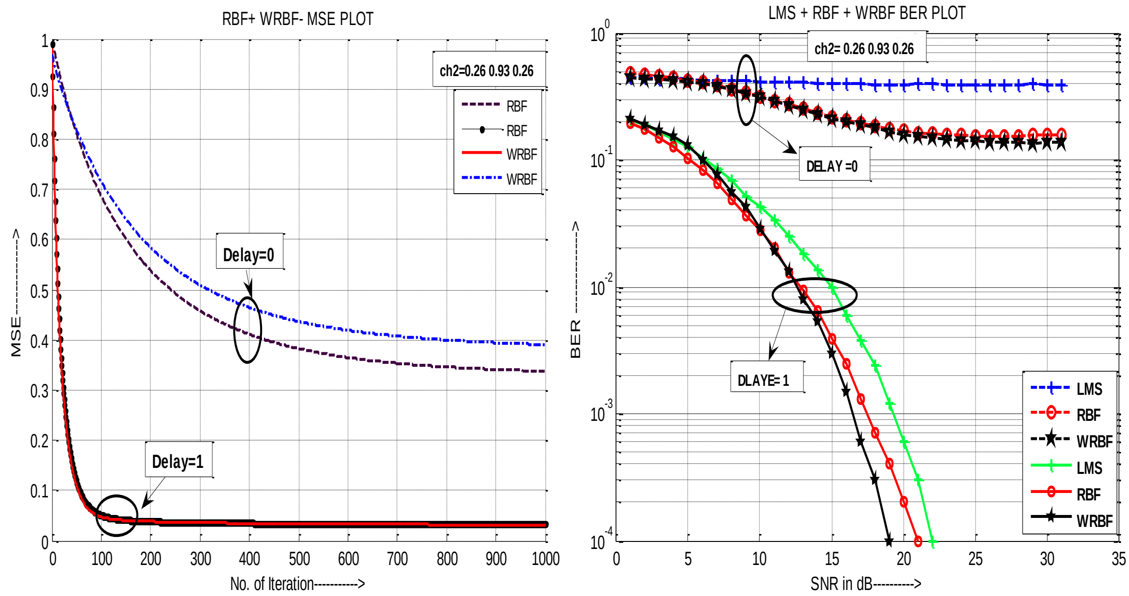


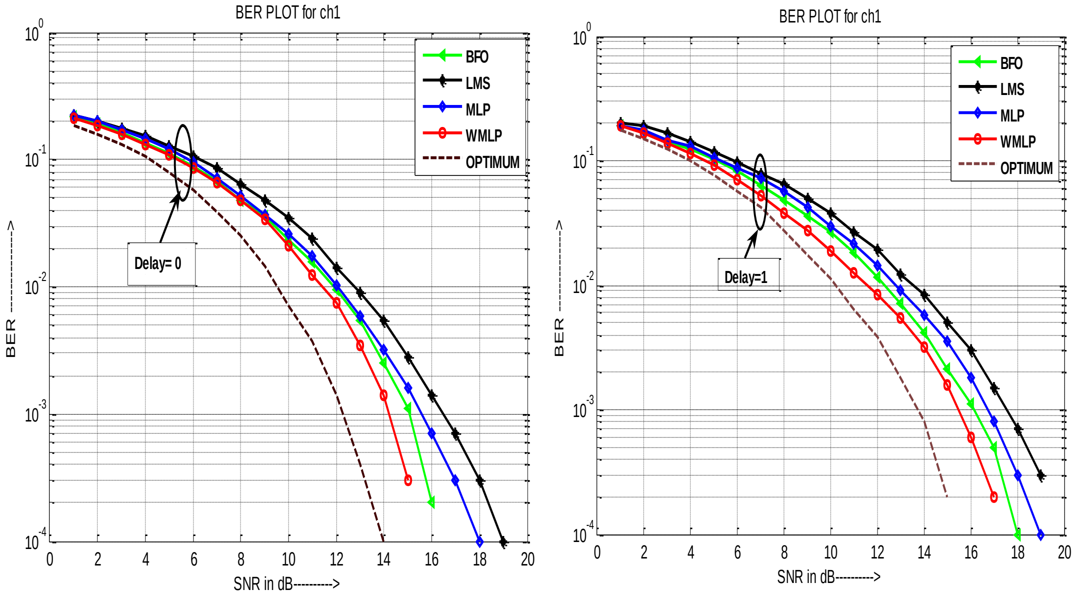
Figure.5.4. MSE & BER performance of RBFN & WGRBFN equalizer compared with LMS trained linear equalizer for ch_2, Delay= 0 and 1.

From above simulation we observed that the WGRBF proposed here provides MSE performance same as RBF and performs better than LMS trained linear equalizer. For BER performance at delay zero all equalizer fail considerable, with one delay WGRBF outperform RBF equalizer.

5.2.2 Performance analysis of WMLPNN and MLP equalizer

Example 5.5.

For this performance we used is a minimum phase channel whose transfer function is $H_1(Z)= 1 + 0.5\, Z^{-1}$. Both the equalizer consists of structure is 3 input nodes, 9 hidden nodes, 1 output node. The performances have been compared with 2-tap RBF equalizer. BFO and LMS equalizer consists of 3-tap FIR filter. The BER plot is plotted for different delay 0 and 1 respectively.



(Delay= 0) (Delay= 1)

Figure.5.5. BER performance of WMLPNN & MLP equalizer compared with RBF and BFO, LMS trained linear equalizer for ch_1

From above simulation we observed that the proposed WMLPNN equalizer provides better performance than as MLP equalizer and BFO and RLS trained linear equalizer. This equalizer even outperforms MLP equalizer. The RBF equalizer outperforms both form of MLP equalizer. This provides MAP decision performance.

5.3. Performance analysis of equalizers for channels with ISI and Nonlinearity

The parameters taken for the simulation were same as those taken for channels with ISI discussed in section 5.1.

5.3.3 Performance analysis of ChNN and FLANN equalizer

Example 5.6.

For this performance we consider the mixed phase channel whose transfer function is $H_3(Z) = 0.30 + 0.90Z^{-1} + 0.30\,Z^{-2}$, and the nonlinearity $b(t) = s(t) + 0.2\,s^2(t) - 0.1s^3(t) + 0.5\cos(\pi s(t))$. ChNN equalizer consists of single input, four different Chebyshev polynomial functions in functional expansion block. The FLANN equalizer consists of single input, Seven different trigonometric function including power series function in functional expansion block. The FIR filter had 5-tap filter. LMS and RLS equalizer consists of 5-tap FIR filter and trained with input 1000 samples. The BER plot for these equalizers is plotted for delay 2 is shown in figure 5.6.

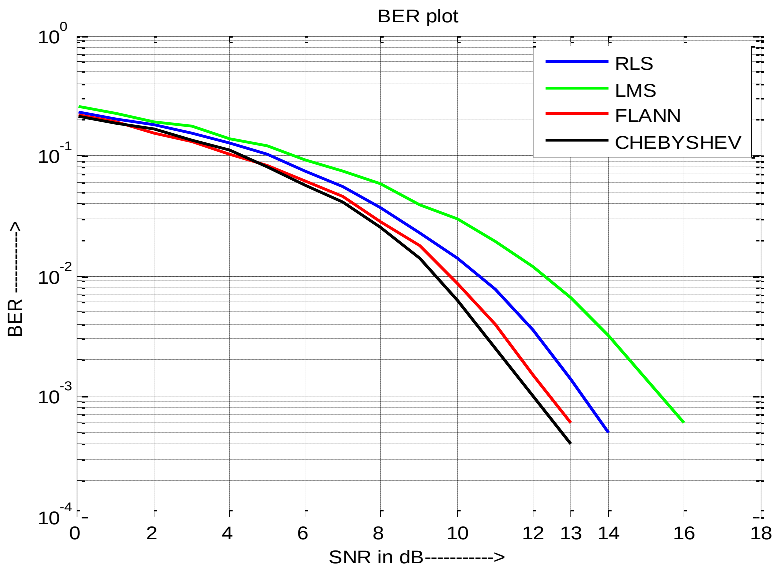


Figure. 5.6 BER performance of ChNN, FLANN compared with RBF and LMS, RLS based linear equalizer for ch_3, delay= 2.

From above simulation we observed that ChNN equalizer provides better performance than FLANN, LMS and RLS based equalizer. RLS algorithm structure is nonlinear and it has convergence speed is faster than LMS algorithm, and it's perform better than LMS trained linear equalizer.

5.4. Performance analysis of equalizers to combat CCI in ISI environment

In this study the channel was corrupted with ISI and Co-channel interference. Consider the SIR was 10, 13, 15, 16 to 30 dB, difference in the presence of adaptive white Gaussian noise (AWGN). The parameters taken for the simulation are same as those taken for channels with ISI discussed in section 5.2. Generally ch0, ch1 are considered as main channel and ch2 has been considered as co-channel.

5.4.1 Performance analysis of WGRBF and RBF equalizer

Example 5. 7.

For this simulation the desired channel is $Ch_1=1+0.5Z^{-1}$ and CCI is $ch_2=0.26+0.93Z^{-1}+0.26Z^{-2}$, and SIR=15dB was consider , both the equalizer consist of as 2-input, 8-center and 1-output. For analyses of how much BER plot performance degraded in equalization if one channel corrupted with CCI and without CCI interference. For which consider optimum equalizer is the one where CCI has not been consider. The BER plot is plotted for different delay 0 and 1 respectively as shown in figure 5.7.

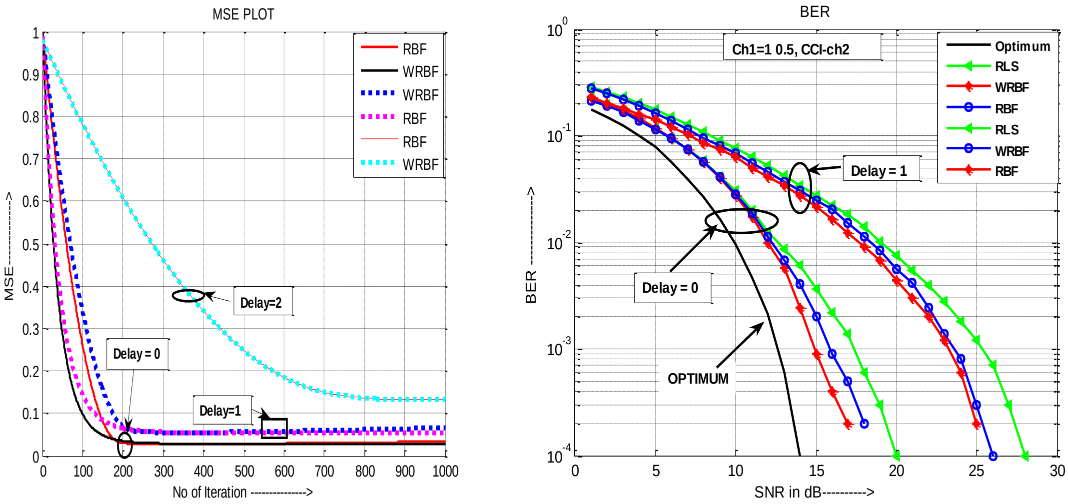


Figure.5.7. MSE & BER performance of RBFN & WGRBFN equalizer compared with LMS based equalizer and optimum equalizer, Delay= 0 & 1.

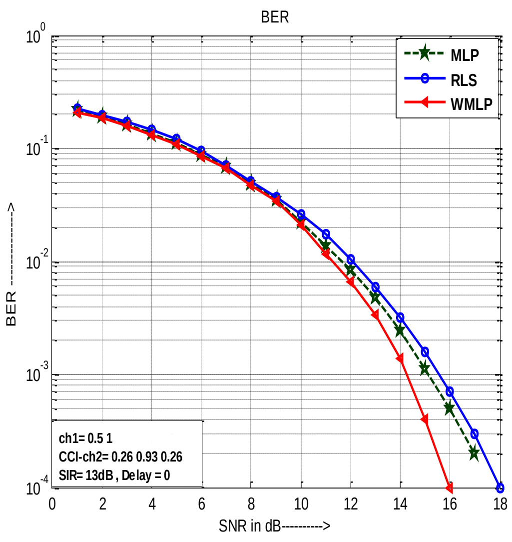
From above simulation we observed that the WGRBF and RBF equalizer treated CCI as noise. WGRBF performs better than RBF equalizer and LMS trained equalizer.

Computational complexity of example 5.7 RBF & WGRBF equalizer is given below

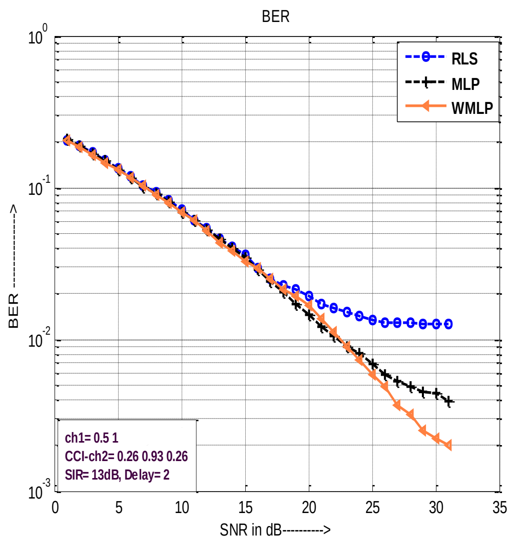
Operation	RBF 2-8-1	WRBF 2-8-1
Addition	33	33
Trigonometric	-	-
Multiplication	64	72
Exponcial	8	8
Tanh(.)	-	-

5.4.2 Performance analysis of WMLPNN and MLP equalizer

Example 5. 8. For analyseing the performance of WMLPNN and MLP equalizer consider the desired channel $ch_0=0.5+Z^{-1}$ and CCI is $ch_2=0.26+0.93Z^{-1}+0.26Z^{-2}$. And SIR=13dB were consider. Both the equalizer consist of as 3-input nodes, 9-hidden nodes and 1-output. The BER plot is plotted for different delay 0 and 2 respectively.



(Delay= 0)



(Delay=2)

Figure.5.8. BER performance of WMLPNN & MLP equalizer compared with RLS based equalizer. Delay= 0, 2.

From above simulation we observed that the proposed WMLPNN equalizer provides better performance than as MLP and RLS trained linear equalizer. Similar performance was observed for other channel and co-channel combination also.

5.4.3 Performance analysis of ChNN and FLANN equalizer

Example 5. 9.

For this simulation the desired channel is $Ch_1=1+0.5Z^{-1}$ and CCI is $ch_2=0.26+0.93Z^{-1}+0.26Z^{-2}$, and consider SIR=13dB , the RBF equalizer consist of as 2-input, 8-center and 1-output. ChNN equalizer consists of single input, five different Chebyshev polynomial functions in functional expansion block. The FLANN equalizer consists of single input, Seven different trigonometric function including power series function in functional expansion block. The input is provided through a TDL. The BER plot is plotted for different delay 1 respectively. LMS and RLS equalizer consists of 3-tap FIR filter and trained with input 1000 samples. The BER plot is plotted for different delay 1 respectively.

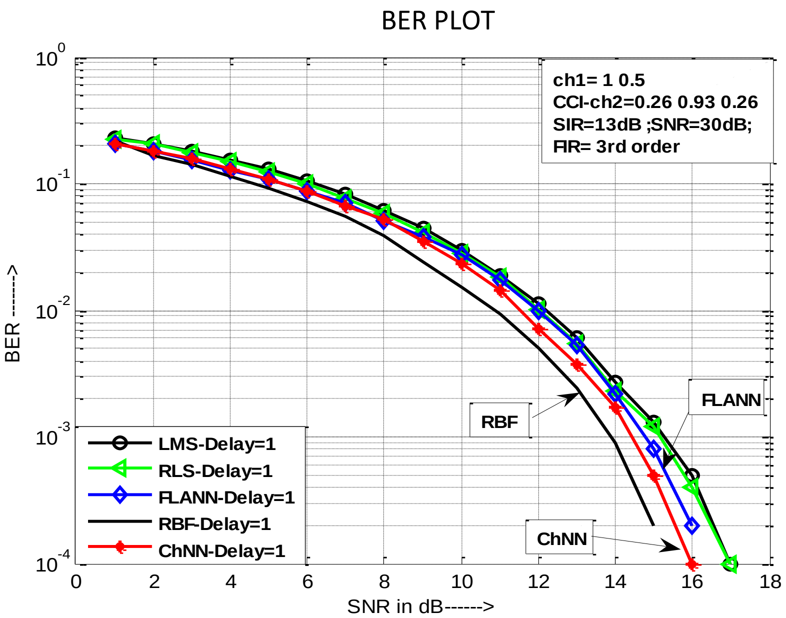


Figure. 5.9 BER performance of ChNN, FLANN compared with RBF and LMS, RLS based equalizer, delay= 1.

From above simulation we observed that ChNN equalizer provides better performance than FLANN, LMS and RLS based equalizer. But RBF provides superior performance than all other equalizer. Similar performance was observed for other channel and co-channel combination also.

Computational complexity of example 5.9 RBF, FLANN and ChNN equalizer given

Operation	RBF 2-8-1	FLANN 1-7-1	ChNN 1-5-1
Addition	33	21	18
Trigonometric	-	6	-
Multiplication	64	35	22
Exponcial	8	-	-
Tanh(.)	-	1	1

Example 5. 10.

For this simulation the desired channel is $ch_2 = 0.26+0.93Z^{-1}+0.26Z^{-2}$ and CCI is $Ch_1=1+0.5Z^{-1}$, Consider were SIR=15dB and both the equalizer consist of as 2-input, 16-center and 1-output. The parameters taken for the simulation are same as those discussed in example 5.10. The BER plot is plotted for different delay 0 respectively.

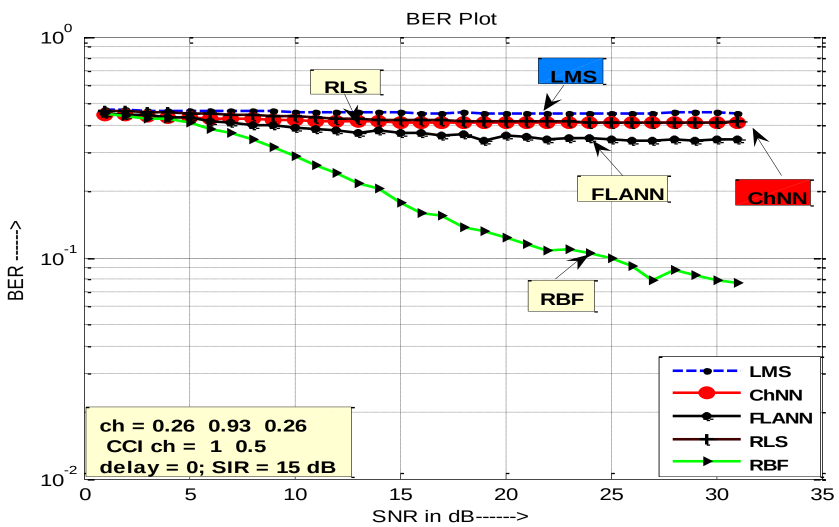


Figure. 5.10 BER performance of ChNN, FLANN compared with RBF and LMS, RLS based equalizer, delay= 0

From above simulation we observed that all equalizer in delay zero fail to recognise the pattern due to CCI, but RBF, ChNN equalizer recognise the pattern in ISI. Generally at high SNR condition the performance difference can be considerable, at a BER 10^{-4} RBF provides 1dB performance superior over ChNN and ChNN provides 0.5dB performance over a FLANN. Similar performance was observed for other channel combination.

5.5. Performance analysis of equalizers for channels with ISI, CCI and Burst noise interference

In this study the channel was corrupted with ISI, CCI and burst noise interference. Consider the SIR was 10, 13, 15, 16 to 30 dB difference in the presence of adaptive white Gaussian noise (AWGN). Also the burst noise added to the main channel is a heigh intensity noise which occuring for short duration of time. The duration of noise is fixed burst length means a series of finite-duration Gaussian noise pulses. The burst noise was added to only with 5% of the samples with SNR 5dB to 10dB. The noise was added to 5 consecutive samples in every 100 samples. The location of these 5 consecutive samples was considered randomly.

5.5.1 Performance analysis of WGRBF and RBF equalizer

Example 5. 11.

For analyseing the performance of WGRBF and RBF equalizer consider the desired channel IS $Ch_1=1+0.5Z^{-1}$ and CCI-$ch_2 = 0.26+0.93Z^{-1}+0.26Z^{-2}$ with SIR=15dB, and Burst noise was added to desired channel with SNR of 5dB for 5% of the samples. Both the equalizer consists of as 2-input, 8-center and 1-output. The BER plot is plotted for different delay 0, 1 and 2 respectively.

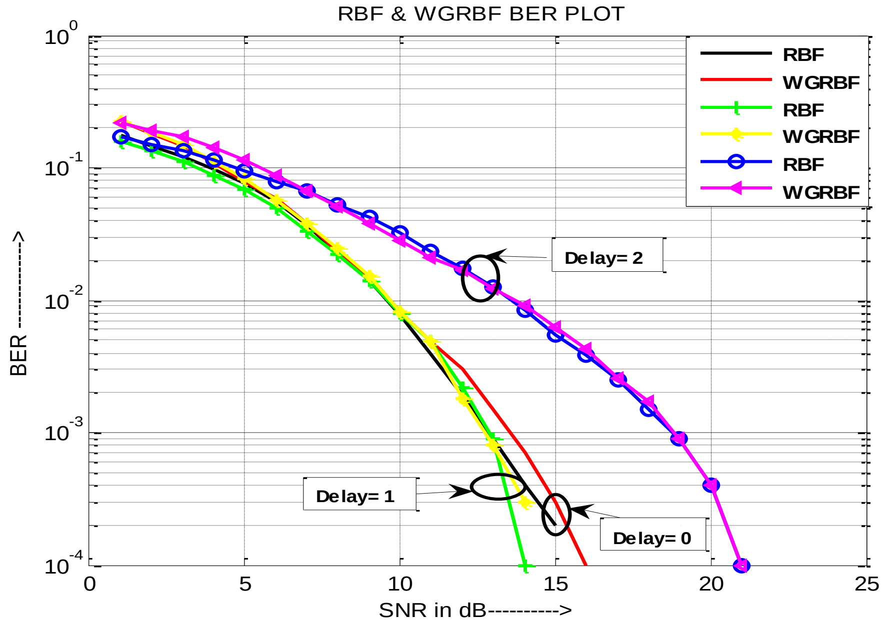


Figure.5.11. BER performance of RBFN & WGRBFN equalizer
Delay= 0, 1 and 2.

From above simulation we observed that the WGRBF proposed here provides performance closer to RBF equalizer. Similar performance was observed for other channel and co-channel combined with burst noise also.

5.5.2 Performance analysis of WMLPNN and MLP equalizer

Example 5. 12.

For analyseing the performance of WMLPNN and MLP equalizer consider the desired channel is $Ch_1=0.5+ Z^{-1}$ and CCI-ch_2 $=0.26+0.93Z^{-1}+0.26Z^{-2}$ with SIR=13dB and Burst noise was added to desired channel with SNR of 10dB for 5% of the samples. Both the equalizer consist of as 3-input nodes, 30-hidden nodes and 1-output. The BER plot is plotted for different delay 1 and 2 respectively.

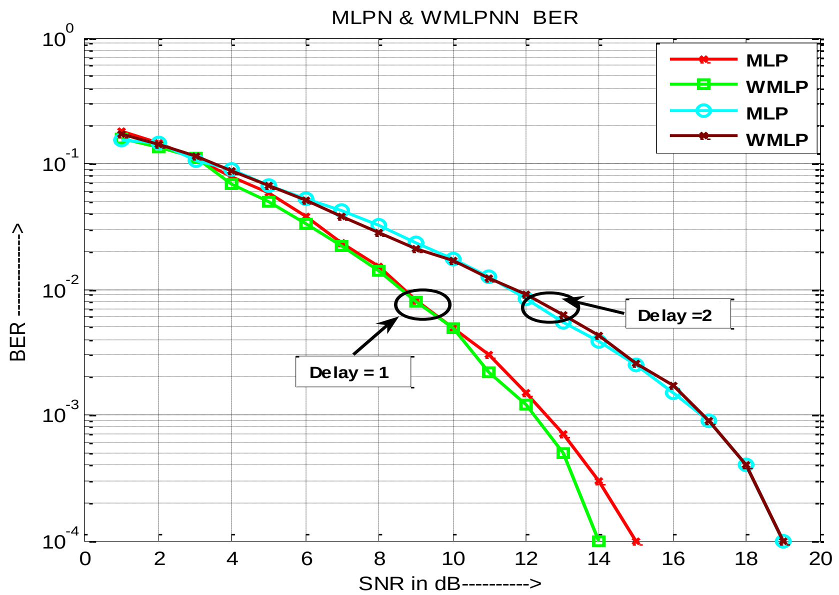


Figure.5.12. BER performance of MLPN & WMLPNN equalizer Delay= 1and 2.

From above simulation we observed that the WMLPNN equalizer proposed here provides better performance than MLP equalizer in delay 1. Similar performance was observed for other channel combination.

5.6. Conclusion

This chapter analyses in details the performance of different types of linear and nonlinear ANN based equalizer like MLP, RBF, FLANN, ChNN and linear adaptive equalizer trained using LMS, RLS or BFO algorithm for channel equalization in digital communication system. Their performance was compared with proposed wilcoxon neural network equalizer. Through extensive simulation study we observed that the proposed Wilcoxon learning algorithm trained neural network equalizer and WGRBF equalizer performed similar as RBF equalizer in ISI and high intensity burst noise interference condition, but its perform better in CCI environment than RBF equalizer. Also WMLPNN equalizer performs better than MLP, BFO and linear equalizer in all ISI, CCI and burst noise environment. RBF equalizer provides MAP decision performance.

Chapter 6

Conclusion

Chapter 6

Conclusion

The main aim of the thesis is to develop novel artificial neural network equalizer (trained with linear, nonlinear and evolutionary algorithms) to mitigate the linear and nonlinear distortion like ISI, CCI and burst noise interferences occurs in the communication channel and can provide minimum mean square error and bit-error-rate plot for wide variety of channel condition.

The research carried out for this thesis primarily discusses the different types of linear and nonlinear equalizers. Performance of ANN based equalizer using MLP, RBF, FLANN, ChNN and linear adaptive equalizers (trained with LMS, RLS or BFO algorithm) are compared with the proposed Wilcoxon neural network equalizer. The proposed neural network equalizer provided work out performances in CCI and burst noise environment than RBF equalizer. This chapter summarises the work reported in this thesis, specifying the limitations of the study and provides some pointers to future development.

Following this introduction section 6.1 discusses the main contribution of this thesis. Section 6.2 provides the limitations and section 6.3 presents few pointers towards future work.

6.1 Contributions of thesis

- The first chapter of the thesis introduced to digital communication system, literature survey and its applications. It also provides a brief overview of the theme of the thesis. The second chapter discussed the algorithms used to train the equalizer and need of adaptive equalizer.
- In this chapter2 analyses the linear equalizer performance. The equalizer trained using LMS and RLS algorithm. We observed that RLS provides faster convergence rate than LMS equalizer.

- In the chapter3, 4 and 5 analyse the performance of the ANN based equalizer like MLP, RBF, FLANN, ChNN in extensively noisy channel condition, like the transmitted signals corrupted by ISI, CCI and burst noise interference. Using different form of channel equalization techniques to mitigate the effects of the interference in communication system. Through extensive simulation study we observed that MLP equalizer is a feed-forward network trained using BP algorithm, it performed better than the linear equalizer, but it has a drawback of slow convergence rate, depending upon the number of nodes and layers. Optimal equalizer based on maximum a-posterior probability (MAP) criterion can be implemented using Radial basis function (RBF) network. RBF equalizer mitigation all the ISI, CCI and BN interference and provide minimum BER plot. But it has one draw back that if input is increased the number of centres of the network increases and makes the network more complicated.

More recently a rank based statistics approach known as Wilcoxon learning method has been proposed for signals processing application to mitigate the linear and nonlinear learning problems. We proposed this network for channel equalizer in communication system to mitigate the all the ISI, CCI and BN interference. In this thesis we used WMLP and WGRBF network. It is seen that the performance of equalizers Viz. Wilcoxon MLPNN and WGRBFNN provide better performance than MLP, BFO, FLANN and linear equalizer in all the ISI, CCI and BN interference environment, but not better than RBF equalizer. Where as WGRBF equalizer perform similar to RBF equalizer in ISI, Burst noise environment, but it perform nearly better than RBF equalizer in CCI environment (observed from extensive simulation study which is presented as figure.5.2, 5.3, 5.4, 5.5, 5.7, 5.8, 5.11, 5.12).

As we know that RBF equalizer provides MAP decision performance (i.e optimized performance), the proposed WGRBF equalizer also provides optimal performance and WMLP equalizer provides better performance as compared to other equalizer we considered. So both proposed equalizer in channel equalization case is showing superior performance.

The Evolutionary algorithm discussed in this thesis has been applied for training a transversal equalizer. Normally a transversal equalizer provides linear decision boundary. Optimization of the weights using BFO algorithm can provide best possible weights.

6.2 Limitations of the work

- All the simulation conducted for BPSK signals. The performance of the equalizer proposed for other forms of modulation like QPSK, MARY-PSK, QAPSK and other modulation forms has been considered.
- The Wilcoxon learning algorithm has for burst noise limited to 5% of the samples. This model deviates from the burst noise model. A details analysis will provide better in depth to the problem. Also the equalization is basically an iterative process of minimization of mean square error, so these equalizers take more training time.
- These equalization techniques for use in recent applications like 2G, 3G communication techniques will be helpful in understanding the problem.

6.3 Scope for further research

Addition of all three interferences like ISI, CCI and Burst noise interference can be computationally more complex. Under these circumstances fuzzy equalisers could provide major performance advantages. The study of fuzzy equalisers for mobile communication systems like GSM systems could provide alternative equalisation strategies.

Recently it has been observed that fractionally spaced equalisers can provide additional benefit in interference mitigation in the form of CCI and ISI. One of the possible directions for research is investigating fractionally spaced fuzzy equalisers for interference limited communication system applications. Also recently used OFDM technique also used to minimize the interferences in the communication system.

Annexure

Channels models used for Simulation studies

The channels used for evaluation of equalization technique were presented in Table 1 and Table 2.

Table. 1. Linear channels simulated

Sl. No.	channel	Channel Type
ch_0	$0.5 + Z^{-1}$	Non-minimum
ch_1	$1 + 0.5Z^{-1}$	Minimum
ch_2	$0.26+0.93Z^{-1}+0.26Z^{-2}$	Mixed
ch_3	$0.30+0.90Z^{-1}+0.30Z^{-2}$	Mixed
ch_4	$0.34+0.87Z^{-1}+0.34Z^{-2}$	Mixed

Table .2. Non-linearity in Channels

SL. No	Non- Linearity
NL=0	$b(t) = s(t)$
NL=1	$b(t) = \tanh(s(t))$
NL=3	$b(t) = s(t) + 0.2\, s^2(t) - 0.1\, s^3(t)$
NL=4	$b(t)= s(t)+ 0.2\, s^2(t) - 0.1s^3(t)+0.5\cos(\pi s(t))$

Bibliography

[1]. S. U. H. Qureshi, “Adaptive Equalization,” IEEE, vol. 73, pp. 1349–1387, September 1985

[2]. S. Haykin, *Neural Networks - A Comprehensive Foundation*. New York: Macmillan, 2006.

[3]. William J. Ebel, and William H. Tranter,” The Performance of Reed-Solomon Codes on a Bursty-Noise Channel”, *IEEE Transactions on communications*, Vol. 43, no. 2/3/4, February/March/April 1995.

[4]. S. Haykin, Adaptive Filter Theory. Englewood *Cliff, NJ, USA*: Prentice Hall, 1991.

[5]. R.W. Lucky, “Techniques for adaptive equalization of digital communication systems," *Bell System Tech*. J., pp. 255-286, Feb 1966.

[6]. Gibson, G.J., S.Siu and C.F.N.Cowan,”Application of multilayer perceptrons as adaptive channel equalizer,” IEEE Int. Conf. Acoust., Speech, Signal Processing, Glasgow, Scotland, pp.1183-1186, 1989

[7]. M. Meyer and G. Pfeiffer, "Multilayer perceptron based decision feedback equalizers for channels with intersymbol interference,”. *IEE, Pt* IVol 140, No 6, , pp 420-424 Dec 1993.

[8]. B. Mulgrew, “Applying Radial Basis Functions,” IEEE Signal Processing Magazine, vol. 13, pp. 50–65, March 1996.

[9]. S. Chen, S. McLaughlin, and B. Mulgrew, “Complex-valued Radial Basis Function Network, Part I: Network Architecture and Learning Algorithms,” Signal Processing (Eurasip), vol. 35, pp. 19–31, January 1994.

[10]. S. Chen, B. Mulgrew and S. Mc Laughlin, “Adaptive Bayesian Equaliser with Decision Feedback”, *IEEE Transactions on Signal Processing*, vol.41, pp. 2918-2927, September 1993.

[11]. Jer-Guang Hsieh, Yih-Lon Lin , and Jyh-Hong Jeng, “ Preliminary Stdy on Wilcoxon Learning Machines”, *IEEE Transactions on Neural Networks* , VOL.19 , No. 2,Febaruary-2008.

[12]. T. T. Lee and J. T. Teng, "The Chebyshev-polynomials-based unified model neural network for function approximation," *IEEE Trans. Systems Man and Cybernetics*, Part B, vol. 28, pp. 925-935, Dec. 1998.

[13]. Pao Y. H., Adaptive Pattern Recognition and Neural Network, Reading, MA, Addison Wesley, Chapter 8, pp.197-222, December 1989.

[14]. J. C. Patra and R. N. Pal, "A functional link artificial neural network for adaptive channel equalization", *EURASIP Signal Processing Jnl*., Vol. 43, No. 2, 1995.

[15]. Fogel D., “What is evolutionary computing”, *IEEE spectrum magazine*, Feb. 2000, pp.26-32

[16]. J.Kennedy and R.C.Eberhat,” Partical Swarm Optimizer “, IEEE Int. Conf. On Neural Networks,IV, 1942-1948. Piscataway,NJ:IEEE Service Center, 1995.

[17]. Y.Shi , R.Eberhart, “A Modified Partical Swarm Optimizer”, Proc. IEEE Int. Conf. on Evolutionary Computation, pp. 69-73, 1998.

[18]. Kevin M. Passino,”Biomimicary of Bacterial Foraging for Distributed Optimization and Control”, IEEE Control System Magazine, pp.52-67, June-2002.

[19]. B. Widrow and M. E. Hoff(Jr), “Adaptive Switching Circuits,” *in IRE WESCON Conv.*, vol. 4, pp. 94–104, August 1960.

[20]. B. Friedlander, Ed., Lattice filters for adaptive processing, ser. 829-867, vol. 70(8). *Proceeding of the IEEE*, August 1982.

[21]. F. R. Magee Jr and J. G. Proakis, “Adaptive Maximum-Likelihood Sequence Estimation for Digital Signaling in the Presence of Intersymbol Interference,” *IEEE Transactions on Information Theory,* vol. IT-19, pp. 120–124, January 1973.

[22]. G. D. Forney, “The Viterbi Algorithm,” *IEEE*, vol. 61, pp. 268–278, March 1973.

[23]. J. Cid-Sueiro, A. Artes-Rodriguez, and A. R. Figueiras-Vidal, “Recurrent Radial Basis Function Network for Optimal Symbol-by-Symbol Equalisation,” *Signal Processing (Eurasip)*, vol. 40, pp. 53–63, October 1994.

[24]. S. K. Patra and B. Mulgrew, "Fuzzy Implementation of Bayesian Equalizer in Presence of intersymbol and Co-channel Interference," IEEE Transactions on Communication system, 1998.

[25]. S. K. Patra and B. Mulgrew, "Efficient Architecture for Bayesian Equalization using Fuzzy Filters," IEEE Transaction Circuits and Systems-II: Analog and Digital Signal Processing, vol. 45, number. 7, pp. 812–820, July 1998.

[26]. Dorigo, M. and Gambardalla, L.M., Ant Colony System: A Cooperative Learning Application to Traveling Salesman Problem, IEEE Transactions on Evolutionary Computation, Vol.1, No.1, and April.1997.

[27]. D.Godard," hannel Equalization Using Kalman filter for Fast Data Transmission", IBM Journal Res. Department, Vol. 18, pp. 267-273, May 1974.

[28]. J. Ido, M. Okada, and S. Komaki, "New Neural Network Based Nonlinear and Multipath Distortion Equalizer for FTTA Systems," IEICE Transactions: Communication, vol. E80-B, pp. 1138–1144, August 1997.

[29]. R. Steel(Ed), Mobile Radio Communication. Pentec Press, London, 1992.

[30]. R. W. Lucky, J. Salz and E. J. Weldon, Jr., Principles of Data Communication. New York: McCraw-Hill, 1968.

[31]. J. G. Proakis and J, H. Miller, "An adaptive receiver for digital signalling through channels with intersymbol interference," IEEE Trans. Inform. Theory, vol. IT-15, pp. 484-497, July 1969.

[32]. K. Feher, Digital Communications: Satellite/ Earth Station Engineering. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[33]. C. L. Fenderson, J. W. Parker, P. D. Quigley, S. R. Shepard, and C. A. Siller, Jr., "Adaptive transversal equalization of multipath propagation for 16-QAM, 90-Mb/s digital radio," AT& T Bell Lab. Tech. Journal., vol. 63, pp. 1447-1463, Oct. 1984.

[34]. CClIT Recommendation V.27 bis, "4800/2400 bits per second modem with automatic equalizer standardized for use on leased telephone-type circuits," Int. Telegraph and Telephone Consultative Committee, Geneva, Switzerland,1980.

[35]. S. Chen, G. J. Gibson, C. F. N. Cowan, and P. M. Grant, "Reconstruction of Binary Signals using an Adaptive Radial Basis Function Equalizer," Signal Processing (Eurasip), vol. 22, pp. 77–93, January 1991.

[36]. B. Mulgrew, "Nonlinear Signal Processing for Adaptive Equalisation and Multi-user Detection," in Proceedings of the European Signal Processing Conference, EUSIPCO, (Island of Rhodes, GREECE), pp. 537–544, 8-11 September 1998.

[37]. J. Cid-Sueiro and A. R. Figueiras-Vidal, "Channel Equalization with Neural Networks," in Digital Signal Processing in Telecommuniocations - European Project COST#229 Technical Contributions (A. R. Figueiras-Vidal, ed.), pp. 257–312, London, U.K.: Springer-Verlag, 1996.

[38]. S. Chen, S. McLaughlin, and B. Mulgrew, "Complex-valued Radial Basis Function Network, Part II: Application to Digital Communication Channel Equalization," Signal Processing (Eurasip), vol. 36, pp. 175–188, March 1994.

[39]. S. B. Widrow, Adaptive signal processing. New Jersey: Prentice-Hall Signal processing series, 1985.

[40]. R.W. Lucky, "Techniques for adaptive equalization of digital communication systems," Bell Sys. Tech. J., pp. 255{286, Feb 1966.

[41]. E.A. Robinson and T. Durrani, Geophysical Signal Processing. Englewood Cliffs, NJ: Prentice-Hall, 1986.

[42]. O.Macchi, Adaptive processing, the least mean squares approach with ap-plications in transmission. West Sussex. England: John Wiley and Sons, 1995.

[43]. Proakis. J. G. Digital Communications. New York: McGraw-Hill, 2004.

[44]. G.J. Gibson, S. Siu and C.F.N. Cowan, "The Application of Nonlinear Structures to the Reconstruction of Binary Signals", *IEEE Transactions on Signal Processing*, vol. 39, no. 8, pp. 1877-1884, August 1991.

[45]. S. Chen, B. Mulgrew and P.M. Grant, "A Clustering Technique for Digital Communication Channel Equalisation Using Radial Basis Function Networks", *IEEE Transactions on Neural Networks*, vol.4, pp. 570-579,July 1993.

[46]. R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons, 1973.

[47]. K. Abend and B.D. Fritchman, "Statistical Detection for Communication Channels with Intersymbol Interference", *Proceedings of the IEEE*, vol.58, pp. 779-785, May 1970.

[48]. S. Haykin, *Digital Communications*, John Wiley and Sons, 2006.

[49]. R. Parishi, E. D. D. Claudio, G. Orlandi, and B. D. Rao, "Fast Adaptive Digital Equalization by Reccurent Neural Networks," *IEEE Transactions on Signal Processing*, vol. 45, pp. 2731–2739, November 1997.

[50]. M.J. Bradley and P. Mars, "Application of Recurrent Neural Networks to Communication Channel Equalisation", *International Conference on Acoustics, Speech and Signal Processing, ICASSP-95*, vol. 5, pp. 3399-3402, 9-12 May 1995.

[51]. J.D. Ortiz-Fuentes and M.L. Forcada, "A Comparison between Recurrent Neural Network Architectures for Digital Equalisation", *IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP-97, vol. 4, pp. 3281-3284, 21-24 April 1997.

[52]. M. T. O zeden, A. H. Kayran, and E. Panayirci, "Adaptive Volterra Channel Equalisation with Lattice Orthogonalisation," *IEE Proceedings - Communication*, vol. 145, pp. 109–115, April 1998.

[53]. J. Patra, R. Pal, B.N.Chatterji, and G. Panda, "Identification of nonlin-ear dynamic systems using functional link artificial neural networks," *IEEETransactions on Systems,Man and cybernetics*, vol. 29, no. 2, pp. 254-262,April 1999.

[54]. J. C. Patra and A. C. Kot, "Nonlinear dynamic system identification using Chebyshev functional link artificial neural networks" *IEEE Tran. SMC*, vol.32, pp. 505-51 1, Aug 2002.

[55]. R. V. Hogg, J. W. McKean and A. T. Craig, Introduction to Mathematical Statistic, 6th edition, Englewood Cliffs, NJ: Prentice-Hall, 2005.

[56]. S. Wan and L. E. Banta, "Parameter incremental learning algorithm for neural networks", IEEE transaction, Neural Network, vol.17, no. 6, pp. 1424-1438, November 2006.

[57]. S. Mishra "A Hybrid Least Square-Fuzzy Bacterial Foraging Strategy For Harmonic Estimation*", IEEE Transactions On Evolutionary Computation*, Vol. 9, No. 1, February 2005, pp 61-73.

[58]. B. W. Lee and B. J. Sheu, "Parallel Hardware Annealing for Optimal Solutions on Electronic Neural Networks," *IEEE Transactions on Neural Networks*, vol. 4, pp. 588–598, July 1993.

[59]. R.J. Williams and D. Zipser, "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks", *Neural Computation*, vol. 1,pp. 270-280, 1989.

[60]. Lippmann. R. P, "An Introduction to Computing with Neural Nets", IEEE ASSP Magazine, pp.4-22, April 1987.

Dissemination of the Research Work

Journal

1. Devi Rani Guha, Sarat Kumar patra, ”Minimum Bit Error Rate Channel Equalizer using Artificial Neural Network”, *International Journal of Computational Intelligence and Healthcare Informatics*, Vol.2,No.1,pp.107-111,April-2009.

International conference

1. Devi Rani Guha, Sarat Kumar Patra,”Channel Equalization for ISI channels using Wilcoxon Generalized RBF Network”, *IEEE (978-1-4244-4837-1), ICIIS-09* University of Peradeniya, Srilanka, December 2009.
2. Devi Rani Guha, Sarat Kumar Patra,”ISI & Burst Noise Interference Minimization Using Wilcoxon Generalized Radial Basis Function Equalizer”, *IEEE (978-1-4244-5615-4)*, *ICMENS-09*, Dubai, UAE, December 2009.
3. Devi Rani Guha, Sarat Kumar Patra,”Novel Approach to Cochannel Interference Mitigation Using Wilcoxon Generalized Radial Basis Function Network*”, IEEE (978-1-4244-4858-6), INDICON-09,* Gujarat, pp. 194-197, December 2009.
4. Devi Rani Guha, Sarat Kumar Patra,”Linear & Nonlinear Channel Equalization Using Chebyshev Artificial Neural Network”, *ACM (978-1-60558-351-8), ICAC3’09,* Mumbai, pp. 553-558, January 2009.
5. Devi Rani Guha, Sarat Kumar Patra,” Novel Approach to ISI Minimization Using Wilcoxon Multilayer Perceptron Neural Network”, *IJcICT, IIMT& UniMAP,* Malaysia,BBSR, January 2010.
6. Devi Rani Guha, Sarat Kumar Patra,” Application of Wilcoxon Generalized Multilayer Perceptron Neural Network for Channel Equalization”, *ICRAES’10, KSR College of Engineering*, Tiruchengode, January 2010.

7. Devi Rani Guha, Sarat Kumar Patra," Co-Channel Interference Minimization Using Wilcoxon Multilayer Perceptron Neural Network", *IEEE conference, ITC'10*, Cochin, Kerala, March 2010.

8. Devi Rani Guha, Sarat Kumar Patra," Linear Adaptive Channel Equalization using Functional Link Artificial Neural Network", *International Conference on Communication, computers and Instrumentation(ICCCI-09)*, VESIT, Mumbai, pp.224-228, January 2009.

9. Devi Rani Guha, Sarat Kumar Patra, "Channel Equalization using Nonlinear Artificial Neural Network", *International Conference on Communication, computers and Networking (ICCCN-08), IEEE sponsored*, Karur, pp.21-26, December 2008.

Dedicated to my parents....