MIMO Channel Equalization and Symbol Detection using Multilayer Neural Network

Athar Waseem
A.H.M. Sadath Hossain

School of Engineering, Dept of Electrical Engineering
Blekinge Institute of Technology
371 79 Karlskrona
Sweden
This thesis is submitted to the School of Electrical Engineering at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science Programme in Electrical Engineering with emphasis on Radio Communication. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:

Author (1):
Athar Waseem
Address: c/o Arbab Alam Rontgenvagen-1 lgh 2115, 14152 Huddinge, Stockholm, Sweden.
E-mail: thr.wsm@gmail.com

Author (2)
A.H.M Sadath Hossain
Address: Minervavagen 1114, LGH 20, 371 41 Karlskrona, Sweden.
E-mail: eee_hossain@hotmail.com

Advisor: Maria Erman
School of Engineering, Dept of Electrical Engineering
Blekinge Institute of Technology

Examiner: Sven Johansson
School of Engineering, Dept of Electrical Engineering
Blekinge Institute of Technology

Internet : www.bth.se/com
ABSTRACT

In recent years Multiple Input Multiple Output (MIMO) systems have been employed in wireless communication systems to reach the goals of high data rate. A MIMO use multiple antennas at both transmitting and receiving ends. These antennas communicate with each other on the same frequency band and help in linearly increasing the channel capacity. Due to the multi paths wireless channels face the problem of channel fading which cause Inter Symbol Interference (ISI). Each channel path has an independent path delay, independent path loss or path gain and phase shift, cause deformations in a signal and due to this deformation the receiver can detect a wrong or a distorted signal. To remove this fading effect of channel from received signal many Neural Network (NN) based channel equalizers have been proposed in literature.

Due to high level non-linearity, NN can be efficient to decode transmitted symbols that are affected by fading channels. The task of channel equalization can also be considered as a classification job. In the data (received symbol sequences) spaces NN can easily make decision regions. Specifically, NN has the universal approximation capability and form decision regions with arbitrarily shaped boundaries. This property supports the NN to be introduced and perform the task of channel equalization and symbol detection.

This research project presents the implementation of NN to be use as a channel equalizer for Rayleigh fading channels causing ISI in MIMO systems. Channel equalization has been done using NN as a classification problem. The equalizer is implemented over MIMO system of different forms using Quadrature Amplitude Modulation scheme (4QAM & 16QAM) signals. Levenberg-Marquardt (LM), One Step Secant (OSS), Gradient Descent (GD), Resilient backpropagation (Rprop) and Conjugate Gradient (CG) algorithms are used for the training of NN. The Weights calculated during the training process provides the equalization matrix as an estimate of Channel. The output of the NN provides the estimate of transmitted signals. The equalizer is assessed in terms of Symbol Error Rate (SER) and equalizer efficiency.

Keywords: MIMO, ISI, NN, SER, Channel Equalizer, QAM.
ACKNOWLEDGEMENT

This thesis work has been done with the great support of our supervisor Maria Erman, who guided us in the right direction, motivated, encouraged and challenged us throughout the work. Thanks to her!

Most importantly special thanks to our family for their support and love all the time.

We are also thankful to our friends who helped us in many matters throughout the work.

Finally, we say thanks to almighty Allah who always bless us and without His blessing this journey would not have been possible for us.

Athar Waseem
A.H.M. Sadath Hossain
CONTENTS

ABSTRACT .............................................................................................................................. II
ACKNOWLEDGEMENT ......................................................................................................... IV
CONTENTS .............................................................................................................................. V
LIST OF FIGURES ................................................................................................................. VII
LIST OF ABBREVIATIONS .................................................................................................... IX
CHAPTER 1 HISTORICAL PERSPECTIVE AND LITERATURE OVERVIEW
   1.1 Introduction .................................................................................................................. 1
   1.2 Problem Statement ......................................................................................................... 1
   1.3 Goals/Objectives ........................................................................................................... 2
   1.4 Methodology ................................................................................................................ 2
CHAPTER 2 WIRELESS CHANNEL MODELS AND DIVERSITY TECHNIQUES
   2.1 Introduction .................................................................................................................. 3
   2.2 Large Scale Fading Or Attenuation .............................................................................. 4
   2.3 Small Scale Fading ....................................................................................................... 5
   2.4 Frequency Dispersion Parameters .............................................................................. 9
   2.5 Wireless Channel Models ........................................................................................... 12
   2.6 Inter-Symbol-Interference Cancellation and Diversity .............................................. 15
CHAPTER 3 CHANNEL EQUALIZATION AND ADAPTIVE ALGORITHMS
   3.1 Introduction .................................................................................................................. 21
   3.2 Channel Equalization ................................................................................................... 21
   3.3 Deconvolution Of A Prior Known Sequence ................................................................ 24
   3.4 Adaptive Algorithm for Equalization of A Prior-Unknown Channel ...................... 25
   3.5 Blind Equalization Algorithms .................................................................................. 26
   3.6 Bussgang Algorithms ................................................................................................. 30
   3.7 Blind Channel Equalization Using Diversified Algorithms ..................................... 36
   3.7.1 Recursive Least Squares Adaptive Algorithms .................................................... 36
CHAPTER 4 NEURAL NETWORKS
   4.1 Introduction .................................................................................................................. 41
   4.2 Fundamental Theory of Neural Networks ................................................................... 43
   4.3 Network Architectures and Algorithms ...................................................................... 45
   4.3.1 The Backpropagation Algorithm ......................................................................... 47
   4.3.2 Resilient Backpropagation .................................................................................... 50
   4.3.3 Conjugate Gradient Algorithms .......................................................................... 50
   4.3.4 Quasi-Newton Algorithms ................................................................................... 51
   4.3.5 Levenberg-Marquardt .......................................................................................... 52
LIST OF FIGURES

Figure 2.1 Multipath channel in wireless communications ........................................ 3
Figure 2.2 Two-ray geometry .................................................................................. 6
Figure 2.3 Channel model ....................................................................................... 6
Figure 2.4 Channel Impulse Response (CIR) of an ideal channel ............................... 7
Figure 2.5 Geometry of Doppler shift ...................................................................... 9
Figure 2.6 Example of computing time-variation of channel ................................. 11
Figure 2.7 Simplest wireless channel model ............................................................. 12
Figure 2.8 Basic methods of diversity combination ................................................. 17
Figure 2.9 Hybrid diversity combination method ................................................... 19
Figure 2.10 Basic configuration OF MIMO system ................................................. 19
Figure 3.1 Continuous time channel method ............................................................ 22
Figure 3.2 Discrete-time model of a wireless channel ............................................ 22
Figure 3.3 General concept of supervised equalization system .............................. 26
Figure 3.4 Bussgang Theorem .............................................................................. 27
Figure 3.5 Basic linear equalization system ............................................................. 27
Figure 3.6 Basic equalization system ..................................................................... 30
Figure 3.7 Basic Decision Feedback Equalizer diagram .......................................... 33
Figure 3.8 Blind equalizer system with diversified algorithms ............................... 36
Figure 3.9 General Recursive Least Squares algorithm .......................................... 37
Figure 4.1 Basic perceptron architecture ............................................................... 42
Figure 4.2 Architecture of the perceptron with general nonlinear activation function 44
Figure 4.3 A multilayer perceptron example of two layer model with N neurons in the
input and M neurons in the output layer ............................................................... 46
Figure 4.4 Decision boundaries for single and two layer network .......................... 46
Figure 4.5 A Two layer model .............................................................................. 47
Figure 5.1 NxN MIMO system with NN based channel estimator & compensator. 54
Figure 5.2 2x2 MIMO Channel ............................................................................ 55
Figure 5.3 Block of ‘NN based channel effect estimator & compensator’ for 2x2 MIMO
.............................................................................................................................. 56
Figure 5.4 Functioning in training mode of neural networks .................................... 57
Figure 5.5 Functioning in operation mode of neural networks ................................. 57
Figure 5.6 Architecture of NN (Same for both receivers) ....................................... 58
Figure 5.7 Flow of simulations ............................................................................. 61
Figure 5.8 4 QAM Complex Symbol Decision Space ............................................ 62
Figure 5.9 16 QAM Complex Symbol Decision Space ......................................... 62
Figure 5.10 3x3 MIMO Channel ........................................................................... 63
Figure 5.11 Block of ‘NN based channel effect estimator & compensator’ for 3x3 MIMO
.............................................................................................................................. 64
Figure 5.12 Architecture of NN for 3x3 MIMO channel (same for all receivers).... 65
Figure 6.1 SER v/s SNR (dB) plot for 2x2 MIMO (4QAM, Training length=16)..... 68
Figure 6.2 SER v/s SNR (dB) plot for 2x2 MIMO (4QAM, Training length=32).... 69
Figure 6.3 SER v/s SNR (dB) plot for 2x2 MIMO (4QAM, Training length=48)... 69
Figure 6.4 SER v/s SNR (dB) plot for 2x2 MIMO (16QAM, Training length=16). 70
Figure 6.5  SER v/s SNR (dB) plot for 2x2 MIMO (16QAM, Training length=32).

Figure 6.6  SER v/s SNR (dB) plot for 2x2 MIMO (16QAM, Training length=48).

Figure 6.7  SER v/s SNR (dB) plot for 3x3 MIMO (4QAM, Training length=16).

Figure 6.8  SER v/s SNR (dB) plot for 3x3 MIMO (4QAM, Training length=32).

Figure 6.9  SER v/s SNR (dB) plot for 3x3 MIMO (4QAM, Training length=48).

Figure 6.10  SER v/s SNR (dB) plot for 3x3 MIMO (16QAM, Training length=16).

Figure 6.11  SER v/s SNR (dB) plot for 3x3 MIMO (16QAM, Training length=32).

Figure 6.12  SER v/s SNR (dB) plot for 3x3 MIMO (16QAM, Training length=48).
**LIST OF Abbreviations**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWGN</td>
<td>Additive White Gaussian Noise</td>
</tr>
<tr>
<td>BPA</td>
<td>Back-propagation Algorithm</td>
</tr>
<tr>
<td>CDMA</td>
<td>Code Division Multiple Access</td>
</tr>
<tr>
<td>CIR</td>
<td>Channel Impulse Response</td>
</tr>
<tr>
<td>CMA</td>
<td>Constant-Modulus Algorithms</td>
</tr>
<tr>
<td>CG</td>
<td>Conjugate Gradient</td>
</tr>
<tr>
<td>DFT</td>
<td>Discrete Fourier Transform</td>
</tr>
<tr>
<td>EM</td>
<td>Electromagnetic</td>
</tr>
<tr>
<td>EW</td>
<td>Exponentially Weighted version</td>
</tr>
<tr>
<td>FDD</td>
<td>Frequency Division Duplex</td>
</tr>
<tr>
<td>FSE</td>
<td>Fractionally-Spaced Equalization</td>
</tr>
<tr>
<td>FIR</td>
<td>Finite Impulse Response</td>
</tr>
<tr>
<td>FDMA</td>
<td>Frequency Division Multiple Access</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning Systems</td>
</tr>
<tr>
<td>GD</td>
<td>Gradient Descent</td>
</tr>
<tr>
<td>HOS</td>
<td>Higher-Order Statistics</td>
</tr>
<tr>
<td>HC</td>
<td>Hybrid Combination</td>
</tr>
<tr>
<td>IDFT</td>
<td>Inverse Discrete Fourier Transform</td>
</tr>
<tr>
<td>IIR</td>
<td>Infinite Impulse Response</td>
</tr>
<tr>
<td>ISI</td>
<td>Inter Symbol Interference</td>
</tr>
<tr>
<td>LS</td>
<td>Least Squares</td>
</tr>
<tr>
<td>LTE</td>
<td>Long Term Evolution</td>
</tr>
<tr>
<td>LOS</td>
<td>Line Of Sight</td>
</tr>
<tr>
<td>LM</td>
<td>Levenber-Marquardt</td>
</tr>
<tr>
<td>LMS</td>
<td>Least Mean Square</td>
</tr>
<tr>
<td>MRC</td>
<td>Maximum Ratio Combination</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>MAP</td>
<td>Maximum A-Posteriori</td>
</tr>
<tr>
<td>MBR</td>
<td>Maximum Bit Rate</td>
</tr>
<tr>
<td>MED</td>
<td>Minimum Entropy Deconvolution</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Square Error</td>
</tr>
<tr>
<td>MIMO</td>
<td>Multiple Input Multiple Output</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>MISO</td>
<td>Multiple Input Single Output</td>
</tr>
<tr>
<td>SIMO</td>
<td>Single Input Multiple Output</td>
</tr>
<tr>
<td>NN</td>
<td>Neural Network</td>
</tr>
<tr>
<td>OFDM</td>
<td>Orthogonal Frequency Division Multiplexing</td>
</tr>
<tr>
<td>OSS</td>
<td>One Step Secant</td>
</tr>
<tr>
<td>OFDMA</td>
<td>Orthogonal Frequency Division Multiple Access</td>
</tr>
<tr>
<td>PAM</td>
<td>Pulse Amplitude Modulation</td>
</tr>
<tr>
<td>PCS</td>
<td>Personal Communication Systems</td>
</tr>
<tr>
<td>PCM</td>
<td>Pulse Code Modulation</td>
</tr>
<tr>
<td>PDCP</td>
<td>Packet Data Control Protocol</td>
</tr>
<tr>
<td>PDN-GW</td>
<td>Packet Data Network Gateway</td>
</tr>
<tr>
<td>RLS</td>
<td>Recursive Least Squares</td>
</tr>
<tr>
<td>Rprop</td>
<td>Resilient Backpropagation</td>
</tr>
<tr>
<td>SER</td>
<td>Symbol Error Rate</td>
</tr>
<tr>
<td>SC</td>
<td>Selective Combination</td>
</tr>
<tr>
<td>QAM</td>
<td>Quadrature Amplitude Modulation</td>
</tr>
<tr>
<td>QS</td>
<td>Quantized State</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>WSS</td>
<td>Wide Sense Stationary</td>
</tr>
</tbody>
</table>
Chapter 1

HISTORICAL PERSPECTIVE AND LITERATURE OVERVIEW

1.1 Introduction

Modern world has been transformed to information-requiring systems; include voice, video, and data with high speed and reliability that could not be predictable even a decade ago. The portability of communicators has additional challenges. To achieve highly reliable and fast communication systems unaffected by the troubles caused due to multipath fading wireless channels is one of the new challenges [1]. While, very few technologies have been commenced and employed over the last decade, Multiple Input Multiple Output (MIMO) is one of them and has got excellent reputation [2] [3]. MIMO communication system has recognized itself as a technology to accomplish the goals of high data rate.

Our goal is the elimination of one of the hurdles that is Inter-Symbol Interference (ISI) in the path of achieving the targets of high data rate and reliable wireless communication systems, whose strength crafts the channel noise insignificant [4].

1.2 Problem Statement

In recent years MIMO systems have been employed in wireless communication systems to reach the goals of high data rate [2][3]. A MIMO use multiple antennas at both transmitting and receiving ends. These antennas communicate with each other on the same frequency band. These multiple antennas help in linearly increasing the channel capacity. Due to the multi paths present in wireless channels, they face the problem of channel fading which cause ISI [4]. Each channel path has an independent path delay, independent path loss or path gain and phase shift, cause deformations in a signal and due to this deformation the receiver can detect a wrong or a distorted signal. The receivers always require the knowledge of Channel Impulse Response (CIR) in order to eliminate these channel effects from received signals [5]. Normally a separate channel estimator is required to obtain the knowledge of CIR. Channel estimators utilizes the known sequences of bits which are transmitted by each transmitter during each transmission burst. These unique known sequences of bits are perfectly known at each receiver.

MIMO systems also face the problem of ISI up to some extent but the channel equalization (to remove all the signal-paths and channel changes) has got more complexity in MIMO technology which is actual challenge. In past many Neural Network (NN) based channel equalizers have been proposed [5] [6] [7]. We also intended to analyze NN based channel equalization.
1.3 Goals/Objectives

The approach of our project is to propose and examine several methodologies, algorithms and configurations to fight against the problem of ISI. Channel equalization is usually done by two approaches that are supervised/training and unsupervised/blind modes. We have analyzed the application of a modified neural network which require comparatively short training period and follow the supervised/training mode for the appropriate channel equalization [1].

Due to high level non-linearity, NN can be efficient to decode transmitted symbols that are affected by fading channels. The task of channel equalization can also be considered as a classification job. In the data (received symbol sequences) spaces NN can easily make decision regions. Specifically, NN has the universal approximation capability and form decision regions with arbitrarily shaped boundaries. This property supports the NN to be introduced and perform the task of channel equalization and symbol detection [8]. Our role is to asses NN based channel equalizer in terms of Symbol Error Rate (SER) and equalizer efficiency for Rayleigh fading channels causing ISI in MIMO systems. This research attempts to determine the effectiveness of several NN-training algorithms by comparing their results. The algorithms include: Levenberg-Marquardt (LM) [9][10], One Step Secant (OSS) [11], Gradient Descent (GD) [12], Resilient backpropagation (Rprop) [13] and Conjugate Gradient (CG) [9][86]. Moreover we intend to ascertain the performance and flexibility of the proposed system by the implementation of the equalizer over MIMO system of different forms using Quadrature Amplitude Modulation scheme (4QAM & 16QAM) signals and also by varying the length of training symbols over a reasonable range. Subsequently all the simulations will be performed in MATLAB to obtain the results for evaluation and comparison.

1.4 Methodology

In order to accomplish the highest success rate in our project, a correct and good methodology is required.

- Develop an analytical overall system model, simulation models for MIMO communication channels and corresponding architectures of neural networks.
- Use the models and architectures to construct MIMO channel equalizer for simulations.
- Execute simulations in Matlab to verify and validate results.
2.1 Introduction

Until the 1970 wireless communication was used for terrestrial links, satellites and broadcasting but in the last three decades it has increased its scope and has gone through many changes [1]. Personal Communication Systems (PCS), Cellular communication systems and wireless networking have been introduced and currently these technologies are dominating in the modern world of wireless communication. The general Additive White Gaussian Noise (AWGN) model is not enough for the modern applications to represent the channel. The presence of Line of sight (LOS) between the transmitters and receivers is never sure in this channel and the presence of multi paths in wireless channels is another important characteristic shown in Figure 2.1[4].

![Figure 2.1 Multipath channel in wireless communications](image)

The basic phenomenon of electromagnetic (EM) wave propagation such as reflection includes more paths between the transmitter and receiver along with original paths.

The common phenomena of electromagnetic wave propagation are [4]:
1. **Reflection:** When EM waves are hit on the objects comes on their way they are reflected by those objects if the wavelength of EM is much smaller than the physical size of the objects.

2. **Diffraction:** Occurs when EM waves hit the obstacles with sharp edges and irregular surfaces that is propagation path encounters some sharp changes.

3. **Scattering:** When the cluster of smaller objects like water vapors having size smaller than the wavelength of EM wave are hit then the copies of EM waves occurs and propagate in several directions.

There are few more phenomena such as refraction and absorption also take place in wireless channels.

The signal power is another important parameter in wireless channels. There are two different cases of power reduction effects.

1. Large-scale effect describes the signal power typically with respect to long wave propagation distances and outcomes in the mean path loss of the signals [4].

2. Small-scale effect is concerned with the relatively quick changes in the signals amplitudes and their powers. It describes the signals power variations with respect to short distances and time intervals round the mean power of the signals [4].

### 2.2 Large Scale Fading or Attenuation

Generally the distance between the transmitters and receivers logarithmically decreases the average power of the received signals. Thus the attenuation occurred due to the distance is called **large scale effect or path loss**. The environment and the medium of propagation also result in some loss of the signal strength.

The average of received signal power at a specific distance is measured by moving the mobile unit (antenna) in a circle. The radius of the circle is kept constant (distance from the transmitter).

The path loss \( L(d) \) (in dB) is the difference between average received signal power \( P(d) \) (in dBm) and transmitted power \( P_t \) at particular distance \( d \).

\[
P(d) = P_t - L(d) , \quad d > d_o \tag{2.1}
\]

The average of the path loss \( \bar{L}(d) \) (in dB), with respect to a referenced distance \( d_o \) at which the path loss is measured and is known, is given by [4]:

\[
\bar{L}(d) = \bar{L}(d_o) + 10n \log_{10} \left( \frac{d}{d_o} \right) \tag{2.2}
\]

\( \bar{L}(d_o) \) is the known path loss at referenced distance \( d_o \), \( n \) is the path loss exponent depends on
the antenna height, frequency and propagation environment. The value of \( n \) is 2 in LOS links and higher than 2 for multi path channels in urban areas [4]. The model given in (2.2) is called the log-distance path loss model. The measure path loss \( L(d) \) is Gaussian random variable. Due to effects like shadowing it is considerably different from the average value and is given by [4]

\[
L(d) = L(d_o) + 10n \log_{10} \left( \frac{d}{d_o} \right) + X_\sigma
\] (2.3)

\( X_\sigma \) is the zero-mean Gaussian random variable (expressed in dB) and \( \sigma \) is the standard deviation (also expressed in dB). This type of path loss is called as log-normal shadowing.

The various values of path loss have been measured at different distances and are gathered in a graph of the path loss (in dB) against the distance (in dB that is \( 10 \log_{10} d \)). The line fitting approximations (e.g. Leas-Squares) can be used to approximate the constant \( n \) [17].

The likelihood of the coverage with in a wireless cellular network is given by the probability \([P_r(d) > \gamma]\) where the distance \( d \) is equal to the radius of the cell.

The coverage area percentage (the region within a cell having acceptable level of power) is given by [4] [16]:

\[
U(\gamma) = \frac{1}{\pi d^2} \int_0^{2\pi} \int_0^d P[P_r(d) > \gamma] r dr d\theta
\] (2.4)

This percentage can also be computed in terms of the erf (error function) as follows [14] [15]:

\[
U(\gamma) = \frac{1}{2} \left( 1 - \text{erf}(a) + \exp \left( 1 - \frac{2ab}{b^2} \right) \left[ 1 - \text{erf} \left( 1 - \frac{ab}{b^2} \right) \right] \right)
\] (2.5)

Where \( a = \gamma - \frac{P_r(d)}{\sigma \sqrt{2}} \), \( b = \frac{10 \log_{10} e}{\sigma \sqrt{2}} \) and \( \text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \)

When the \( a = 0 \) (signal level is \( P_r(d) = \gamma \)), then

\[
U(\gamma) = \frac{1}{2} \left[ 1 + \exp \left( \frac{1}{b^2} \right) \left( 1 - \text{erf} \left( \frac{1}{b} \right) \right) \right]
\] (2.6)

### 2.3 Small Scale Fading

The mobile unit receives several copies of the transmitted signal at different times which are slightly apart. This happens due to multi paths present in wireless channel (that is multi path propagation). These multiple copies are called multipath waves, induce interference and cause significant distortion (fading) known as Inter Symbol Interference (ISI). The wireless signals experience fast changes in amplitudes over a very small time period shown in Figure 2.2 The
waves travel through several paths, therefore cover different distances, and then get sum up at receiver side (1 antenna or antenna array) to cause ISI of such level that by comparison the outcomes of large scale path loss is totally ignored [4] [16].

There are multiple ways to model the radio channels statistically in order to characterize the stochastic behavior of multipath channel fading. The simplest method contains a time varying and linear fading Channel Impulse Response (CIR) represented by \( h(t, \tau) \) [18] [19].

\[
\begin{align*}
S(t) & \rightarrow h(t, \tau) \quad \rightarrow r(t)
\end{align*}
\]

**Figure 2.3 Channel model**

**A. Time Dispersion Parameters**

The perfect channel always has a linear phase response and a constant gain over a desired bandwidth (frequency range). To protect the signal spectral features the transmitted signal frequency spectrum should be less than that frequency range. Figure 2.4 shows an ideal channel \( h(t, \tau) = g_0 \delta(t - \tau) \) where \( g_0 \) is constant.
Figure 2.4 Channel Impulse Response (CIR) of an ideal channel

An ideal channel impulse response involves only one received signal at a time delay of $\tau$, and which do not generate ISI even the gain changes with respect to time as the changing CIR of $h(t, \tau) = g(t)\delta(t - \tau)$ (i.e. $\delta(t - \tau) = \begin{cases} 1 & t = \tau \\ 0 & t \neq \tau \end{cases}$) where $g(t)$ comparatively changes slowly and it may be a complex valued time function.

Let us assume that a multipath channel has N different paths, at kth the power and delay are represented by $P_k$ and $\tau_k$ respectively, then the mean excess delay (weighted average delay) is given by [4]:

$$\bar{\tau} = \frac{\sum_{k=1}^{N} g_k^2 \tau_k}{\sum_{k=1}^{N} g_k^2} \quad (2.7)$$

Its second statistical moment can be computed by [4]:

$$\bar{\tau}^2 = \frac{\sum_{k=1}^{N} g_k^4 \tau_k^2}{\sum_{k=1}^{N} g_k^2} \quad (2.8)$$

The rms value of delay known as Channel delay spread is given by [4]:

$$\sigma_\tau = \sqrt{\bar{\tau}^2 - (\bar{\tau})^2} \quad (2.9)$$

In practical cases the CIR changes very slowly with the time, will be explained later. The channel having time dependent impulse response $h(t, \tau)$ also has time dependent frequency response $H(\omega, \tau)$ [1]:

$$H(\omega, T) = \int_{-\infty}^{+\infty} h(\tau, t)e^{-j\omega \tau}d\tau. \quad (2.10)$$

The correlation coefficient or factor of frequency response is required in order to determine the
characteristics of a wireless channel. It is based on the size ($\Delta \omega$ or $2\pi \Delta f$) of change in frequency [1].

\[
P(\Delta \omega) = \frac{E[H^*(\omega,t)H(\omega+\Delta \omega,t)]}{E[H^*(\omega,t)H(\omega,t)]} = \frac{E[H^*(\omega,t)H(\omega+\Delta \omega,t)]}{E[|H(\omega,t)|^2]} \tag{2.11}
\]

\[
P(\Delta \omega) = \frac{\int_{-\infty}^{\infty} |h(\tau,t)|^2 e^{-j \Delta \omega \tau} d\tau}{\int_{-\infty}^{\infty} |h(\tau,t)|^2 d\tau} \tag{2.12}
\]

In frequency domain the complement of the delay spread is the coherence bandwidth denoted by $B_c$, represents the bandwidth (range of frequencies) where the channel gain remains unchanged (flat) with a linear phase and it can be approximated on the basis of specified value of correlation coefficient.

The coherence bandwidth $B_c$ can be approximated by $B_c \approx \frac{1}{\sigma_T}$ when the correlation coefficient is almost zero, $(\Delta \omega) \approx 0, \Delta \omega = 2\pi B_c$. The change of frequency by $B_c$ results in a totally different or statistically independent gain [4].

The coherence bandwidth for the common value of $P(\Delta \omega) \approx 0.5$ (or 50%) is estimated by $B_c \approx \frac{1}{5\sigma_T}$, which indicates that the gains of channel at $\omega$ and $\omega + B_c$ are same.

In the last when taking $P(\Delta \omega) \approx 0.9$ (or 90%) the coherence bandwidth is approximated by $B_c \approx \frac{1}{50\sigma_T}$. Which shows that the gains of channel are exactly the same at $\omega$ and $\omega + B_c$.

The channel is categorized as flat or flat fading when coherence bandwidth $B_c$ is greater than given signal bandwidth $B_s$ ($B_c > B_s$).

The symbol time duration is denoted by $T_s$ and the minimum signal bandwidth is computed by $B_s = \frac{1}{T_s}$. This signal bandwidth is needed to be atleast a bit less than the channel coherence bandwidth in order to for channel to be considered as flat fading. A rule of thumb is specified by [4].

\[
B_s = \frac{1}{T_s} \leq \frac{1}{10\sigma_T} \text{ or equivalently by } \frac{\sigma_T}{T_s} \leq 0.1
\]

There is no channel compensation required for the case of flat fading, the upper bound of symbol rate in the channel is $R_s \leq \frac{0.1}{\sigma_T}$.

If the above condition is not satisfied that is $B_c > B_s$, then the ISI occurs and the receiver
receives distorted signal [4].

In many multipath channels the effects of distortion due to ISI are major and dominate the noise also. These channels are known as dispersive and in engineering terms the process and efforts required for the elimination of ISI is called channel equalization in the field of wireless communication and de convolution in some other fields like geophysics.

In short the channel behavior can be expressed in two ways; first with respect to the bandwidth of signal and second according to the delay spread of channel [4].

- When \( B_s \ll B_c \) and \( T_s \gg \sigma_t \) the channel is known as frequency-not-selective or flat fading.
- When \( B_s > B_c \) and \( T_s < \sigma_t \) the channel is known as frequency-selective or non-flat.

### 2.4 Frequency Dispersion Parameters

The movability of the mobile unit generates another parameter called Doppler shift in frequency [4]. In simple words some change occur in frequency due to the velocity of mobile unit.

The Doppler shift is denoted by \( f_d \) and can be computed as \( f_d = \frac{v \cos \theta}{\lambda} \), where \( v \) is the relative mobile speed, \( \lambda \) is the frequency wavelength, and \( \theta \) is the angle between the mobile unit direction and wave direction. When the mobile unit is going away from the transmitter then the change in frequency is negative while it is positive when approaching towards the transmitter.

![Figure 2.5 Geometry of Doppler shift](image)

It is clear that each multipath will have different Doppler shifts and hold random natures; the angle \( \theta \) can be taken as random and also uniformly distributed in mostly cases.

There exist many copies of transmitted signal at mobile antenna, traveling along different
directions and paths, which are distinguished by their relative angles and speed. Furthermore in particular cases, the surrounding objects can also be moving, producing time varying Doppler shifts over multiple components.

This respective random change in frequency causes spectral extension called Doppler spread. Therefore Doppler spread is described as the range of frequencies where the Doppler shift is not zero. The Doppler spread of particular channel or the maximum Doppler shift is denoted by \( B_d \).

The wireless channel can be characterized with respect to Doppler spread \( B_d \).

- If the Doppler spread is much lesser than the signal bandwidth that is \( B_d \ll B_s \), the type of fading is called slow fading, therefore its effects are negligible. The channel also changes on a very slow rate and be taken as constant over many symbol time durations.

- If the effects of the Doppler spread are much higher and cannot be overlooked that \( B_s < B_d \), the CIR changes speedily with respect to the time duration of one symbol then the channel is known as fast fading.

The channel properties in time domain can be more specified by introducing another parameter called coherence time, which is the time duration during which the CIR is invariant. If the time separation of any two samples of the channel is less than the coherence time then they are highly correlated, the provided definition depends on the time correlation coefficient.

The correlation coefficient as a function of time difference \( \Delta t \) in the time domain is expressed by:

\[
P(\Delta t) = \frac{\mathbb{E}[h(t)h^*(t+\Delta t)]}{\mathbb{E}[|h(t)|^2]} \tag{2.12}
\]

Normally, the coherence time is given by [4].

\[
T_c \approx \frac{1}{B_d} \tag{2.13}
\]

When the time correlation coefficient of equation (2.12) remains 50% or 0.5 above the coherence time is approximated by [4]:

\[
T_c \approx \frac{9}{16\pi B_d} \tag{2.14}
\]

The geometric average of 2.13 and 2.14 is used as a rule of thumb for digital communication [4].
The channel characteristics can also be classified in terms of coherence $T_c$ time, and the symbol time duration $T_s$ as follows:

- If the $T_s < T_c$, then the complete signal or symbol is affected in the same way by the channel, and the channel is called slow fading.
- If the $T_s > T_c$, some parts of a signal or symbol are affected in different ways because the changes in channel are faster than symbol duration. Hence, the channel is known as fast fading.

In conclusion, wireless channels can be categorized in four types. In terms of delay spread, the channel is called flat (not frequency selective) or frequency selective, and in terms of Doppler spread (coherence time) the channel is considered as slow or fast fading.

In modern wireless communication the channel are considered to be selective or slow fading that means that the channel is highly dispersive, though variations in channel are slow with respect to time.

To examine the time variations of typical channels, take the case in figure 2.6 where mobile unit is moving with speed of 50 mph at an angle of 40° in the cardinal East-West direction.

![Figure 2.6 Example of computing time-variation of channel](image)

The path length of the ray illustrated in figure 2.6 is given by $d_t = \sqrt{d_0^2 + (d_1 + d_2)^2}$ meters, its
corresponding delay is $\tau_1 = \frac{d_1}{C}$ seconds. Taken the geometric distances: $d_0 = 600$ m, $d_1 = 100$ m, $d_2 = 350$ m the delay $\tau = 2.5 \mu$s. Assuming a symbol rate of $R_s = 19.2$ ksps that is equivalent of symbol time $T_s \approx 52 \mu$s the time delays after 10,000 symbols (0.52 s) is calculated as $\tau_2 = 2.50224 \mu$s. That is a change of less than 0.1% in the time delay of the travelling ray. This example validates the applied assumption that time variation of the wireless channels are extensively slow as compared to common symbol rates (Here we have used the worst case scenario).

### 2.5 Wireless Channel Models

The variation of wireless channels are analytically modeled to evaluate their effects on transmitted signals that is required for radio resource management, capacity and coverage optimization. The simplest wireless model based on early discussions is shown in Figure 2.7. In this model the signal amplitude with $\sqrt{L(d)}$ factor is adjusted to compensate large-scale path loss.

![Figure 2.7 Simplest wireless channel model.](image)

The CIR part of the model collects the small scale variations, namely $h(t, \tau)$. The amplitude and the phase of received signal depend on the distance $d$ between transmitter and receiver and the receiver’s horizontal plane angle $\phi$ (azimuth) with respect to reference. This is acceptable in all cases even when the transmitter antenna or antenna array is omni directional owed to the fact that the physical channel is not necessarily azimuthally symmetric [1].

When the angular average of the path loss at particular distance is taken into account then the dependence on distance shown in equation (2.3) and the azimuth angle is ignored.

The dependence on the distance $d$ and azimuth angle $\phi$ is supposed to be disguised and precise notation is absent in order to have sharp and simple mathematical expression.
The dependence on time must be taken into account for the CIR which should be given in accurate notation as \( h(d, \varphi, t) \) or \( h_{d,\varphi}(t) \), but for simplicity it is shown as \( h(t) \) in further discussion.

The linear time-varying response is estimated by delta function for the frequency-selective or flat fading respectively. The delta functions contain usually complex valued amplitudes and variable that requires being statistical model.

- The first case we take when no Line of Sight (LOS) path exists between the transmitter and the receiver. There are \( M \) paths exist in the channel. The received signal consists of delayed and weighted signals and an independent random noise process in terms of an unmodulated sinusoidal carrier signal.

\[
r(t) = \sum_{j=1}^{M} \alpha_j \cos (2\pi f_c t + \varphi_j) + n(t)
\]

\[
r(t) = \cos(2\pi f_c t) \sum_{j=1}^{M} \alpha_j \cos(\varphi_j) - \sin(2\pi f_c t) \sum_{j=1}^{M} \alpha_j \sin(\varphi_j) + n(t)
\]

\[
= \cos(2\pi f_c t) \ast I(t) - \sin(2\pi f_c t) \ast Q(t) + n(t)
\]

The parameters \( I(t) \) and \( Q(t) \) are approximated as iid (identical and independently distributed) zero mean Gaussian random variables with variance \( \sigma^2 \), by using the central limit theorem and valid physical environment assumptions.

The envelope \( R(t) = \sqrt{I^2(t) + Q^2(t)} \) is distributed according to Rayleigh pdf (probability distribution function) which is generally independent of time [4] [16].

\[
f_R(r) = \frac{r}{\sigma^2} \exp \left( -\frac{r^2}{2\sigma^2} \right), r \geq 0
\]

Where the phase component is:

\[
\Theta(t) = \tan^{-1} \left( \frac{Q(t)}{I(t)} \right)
\]

The phase variable is normally acknowledged to be uniformly distributed in \([-\pi, \pi]\) pursues from iid assumption. The average power of the received signal can be estimate by \( \mathbb{E}[I^2] + \mathbb{E}[Q^2] = 2\sigma^2 \).

The small-scale variations of the channel are being collected in the CIR, \( h(\tau, t) \) and are modeled with single or many delta functions controlled by a complex-valued path gain. The simplified CIR is given by:
\[ h(\tau, t) \approx \sum_{j=1}^{M} r_j e^{i\theta_j(t)} \delta(t - \tau_j) \quad (2.20) \]

When \( M = 1 \) in flat fading channel, the CIR reduces to single delta function. The path gains are supposed to be slowly varying with time (time-independent) where the phase the phase components \( \theta_j(t) \) varies rapidly with time. Their distributions for \( R \) and \( \Theta \) are given in (2.19) and (2.20) respectively. The received baseband signal is normalized such that \( \sum_{j=1}^{M} r_j^2 \delta(\tau - \tau_j) \) is obtained by normalizing the signal power in Equation (2.21) is valid.

The last term is obtained by normalizing the signal power in Equation (2.21) is valid.

The square of the magnitude of path gains with \( \sum_{j=1}^{M} r_j^2 \delta(\tau - \tau_j) \) defines the power-delay profile of multipath channels.

- In the second case the path gain models include a dominant LOS component along with different multipath components. The path gain \( I(t) \) and \( Q(t) \) have no more zero-mean, however they are Gaussian having equal variance. Assume \( m_1 = \mathbb{E}[I] \) and \( m_2 = \mathbb{E}[Q] \) with equal variance as \( \sigma^2 \). \( s = \sqrt{m_1^2 + m_2^2} \) is a non-centrality parameter, and \( k = \frac{m_1^2 + m_2^2}{2\sigma^2} = \frac{s^2}{2\sigma^2} \) is the Rice factor. The pdf of the envelope \( R = \sqrt{I^2 + Q^2} \) must be Ricean as follows [4][16]:

\[ f_R(r) = \frac{r}{\sigma^2} \exp\left(\frac{-r^2 + s^2}{2\sigma^2}\right) I_0\left(\frac{rs}{\sigma^2}\right), \quad r \geq 0 \quad (2.22) \]

The \( I_0\left(\frac{rs}{\sigma^2}\right) \) part is the modified Bessel function of the 0th order. This function in closed form can be given by [19]:

\[ I_0(x) = \frac{1}{\pi} \int_0^\pi \cosh(x \sin \xi) d \xi = \frac{1}{2\pi} \int_0^{2\pi} \exp(x \sin \xi) d \xi \quad (2.23) \]

It is calculated using series expansions with \( I_0(0) = 1 \) for details see [19]. Apparently, when the LOS vanishes, the non-centrality \( s = 0 \) and the pdf converge to Rayleigh.

- In the last, for urban zones with close buildings when no dominant LOS component exists, then the small-scale multipath channel is more precisely modeled by the Nakagami distribution that models the magnitude of the path gains better than Rayleigh.
Considering $\mathbb{E}[R^2] = \Omega (\Omega = 2\sigma^2)$ and defining $= \Omega^2(\Omega^2 = 2\sigma^2)$, the Nakagami pdf is given by [23]:

$$f_R(r) = \frac{2}{\Gamma(m)} \left(\frac{m}{\Omega}\right)^m r^{2m-1}\exp\left(-\frac{mr^2}{\Omega}\right), \quad r \geq 0, \; m \leq 0.5 \quad (2.24)$$

Where $\Gamma(m) = \int_0^\infty \exp(-x)x^{m-1}dx$ is the Gamma function with $\Gamma(1) = 1$.

The Nakagami distribution is equivalent to the famous Gamma distribution with parameters $\alpha = m$, $\beta = \frac{\Omega}{m}$ and if random variable $R^2$ is substituted with $x$ then the following pdf is given by:

$$f_x(x) = \frac{1}{\Gamma(\alpha)\beta^\alpha} x^{\alpha-1} \exp\left(-\frac{x}{\beta}\right), \quad x \geq 0 \quad (2.25)$$

Moreover, when $m = 1$ then the Nakagami is equivalent to the Rayleigh distribution.

In the above discussion some important issues need to be stressed.

a. As mentioned above, the components of the path gains $I(t)$ and $Q(t)$ are independent and can be dealt separately. In a few cases, the envelope $R = \sqrt{I^2 + Q^2}$ is supposed to be comparatively constant with respect to time (slow variations) and the phase $\theta = \tan^{-1}\left(\frac{Q}{I}\right)$ can also be constant and disregarded as it is tracked by the receiver. These assumptions estimate the channel with an actual impulse response.

b. In modern digital communication, the received signal (baseband signals), after passing through matched filter and the sample and hold circuit are described by a discrete-time impulse response or by a transversal filter of finite length. Such CIR is defined by the sequence $\{h[j]\}_{j=-1}^{L-1}$, and in terms of unit impulses by $\sum_{j=0}^{L-1} h[j]\delta[k-j]$ when the CIR is calculated by a finite-length sequence of size $L$. In general, the components $\{h[k]\}$ can be real-valued or complex.

c. When the channel model is similar to a recursive filter representation, it can still be satisfactorily modeled with a very large transversal or an infinite filter.

### 2.6 Inter-Symbol-Interface Cancellation and Diversity

Sophisticated processes are required in highly dispersive channels in order to reduce or remove effects of ISI. These channels are considered as invariant during the time needed for the
execution of necessary process. Furthermore some ways of adaptation are required to implement, to compensate and adjust the slow variations in the channel,

a. The first algorithm introduced here is one of the approaches implemented to remove ISI in general. These methods are designed to nullify or mitigate the outcome of the channel response. These techniques are known as channel equalization. The process of channel equalization is a very straightforward and simple adaptive system using training based methods. On the other hand blind equalization where no training period required is more complex. Blind (unsupervised) channel equalization has been performed in literature using various approaches to estimate channel characteristics.

b. The Orthogonal Frequency Division Multiplexing (OFDM) Technique has been employed in last decade to counter the problems of ISI. Although this system also required channel equalization up to some extent, since it is very clear in OFDM that channel seems to be different due to narrow frequency band. As the signal is composed of many parts therefore it helps in to minimize frequency nonselective or flat-fading [20][24].

c. The probability of error can be improved in fading channels by transmitting the information through many independent channels. The techniques based in this idea are called diversity techniques, and the copies information can be achieved in time, frequency or space.

Latest digital communication systems that are integrated with diversity techniques are jointly MIMO (Multiple Input Multiple Output) systems. These systems can also be multiple outputs with single input (SIMO) or multiple inputs with single output (MISO) systems. One diversity gain is described by [21]:

$$G_d = - \lim_{\gamma \to \infty} \frac{\log(P_e)}{\log(\gamma)} \quad (2.26)$$

Where \( \gamma \) is the Signal-to-Noise Ratio (SNR) used in log-scale into the equation and \( P_e \) is the probability of bit error or Bit Error Rate (BER).

MIMO systems can produce diversity in time, frequency or space and utilize diversity to achieve highest performance in terms of BER.

The diversity can be achieved by creating and separating fading channels in time, space or frequency.

a. When frequency division is much greater than coherence bandwidth \( (B_c \approx \frac{1}{\sigma_t}) \) then use different carrier frequencies
b. When the separation in time must be greater than coherence time \( T_c \approx \frac{1}{B_d} \) then use different time slots (temporal diversity).

c. When spatial separation of antennas is required to be equal to or more than the half-wavelength of the carrier frequency \( \frac{\lambda_C}{2} \) then use multiple antennas (spatial diversity).

The temporal diversity is achieved only when mobile unit is in motion, otherwise the moving objects around stationary mobile unit can create zero Doppler shift and independent channels in time that is not practicable.

Moreover polarization diversity (vertical or horizontal) can only provide a diversity of 2\(^{nd}\) order therefore it is not seriously accepted. Second important aspect is to utilize existing diversity in efficient way. To see how this can be achieved see [14]. The first case we consider here is SIMO systems and also a short introduction to MIMO systems.

The methods implemented in SIMO systems to achieve high diversity gain are based on the approach the received signals are collected and combine through various independent channels. These methods include:

- Maximum Ratio Combination (MRC)
- Selective Combination (SC)
- Hybrid Combination (using both of the above)

Figure 2.9 illustrates the basic concept of MRC and SC systems.

![Diagram of MRC and SC systems]

**Figure 2.8 Basic methods of diversity combination**

**A. Maximum Ratio Combination**
When there exist R independent channels (paths), all containing same copies of the information, then the decision statistic is calculated as the weighted sum of the signals collected from all the paths [2] [3] [4].

\[ z[k] = \sum_{j=1}^{R} w_j^* r_j[k] + n[k] \]  

(2.27)

where \( w_j^* \) represent the optimum combining weights, \( r_j[k] \) represent received signals through all paths at the \( r \)th sample time, and the overall noise is denoted by \( n[k] \). Equation (2.27) assumes coherent detection, and includes ML (Maximum Likelihood). The output SNR of the MRC system assuming that each path has equal SNR described by the ratio of symbol energy and PSD of noise \( \gamma_j = \frac{E_s}{N_0} \) is given by:

\[ \gamma_{MRC} = \sum_{j=1}^{R} \gamma_j \]  

(2.28)

Equation (2.28) shows that there is \( R \) times possible increase in SNR value. In fact, the SNRs of each path are not equal, and the mean value \( \mathbb{E}[\gamma_j] \) is required to use to obtain:

\[ \gamma_{MRC} = R \mathbb{E}[\gamma_j] \]  

(2.29)

**B. Selective Combination**

In MRC systems many radio RF chains required important hardware where as the SC systems are based on one single radio that chooses the finest received signal considering the SNR of the channels. The choice can be recognize by finding and selecting the most efficient receiving antenna. The diversity gain using SNR of the output signal has been calculated; channel is assumed as Rayleigh fading channel and by applying other standard assumptions Equation (2.29) is derived [3] [4].

The average SNR in SC systems is computed as:

\[ \bar{\gamma}_{SC} = \mathbb{E}[\gamma_j] \sum_{j=1}^{R} \frac{1}{j} \]  

(2.30)

The increase of \( \sum_{j=1}^{R} \frac{1}{j} \) indicates the average of the output SNR that can be achieved for details see [14].

**C. Hybrid Combination**

The mixture of above two combining methods results in HC systems. Figure 2.9 illustrates the block diagram of HC systems [3] [4].
Applying similar assumptions, the average output SNR is given by:

\[
\bar{\gamma}_H = \mathbb{E}[\gamma_i] J \left[ 1 + \sum_{i=1}^{R} \frac{1}{T} \right] \tag{2.31}
\]

**D. Spatial Multiplexing**

MIMO systems utilize multiple transmitters and receivers to achieve maximum diversity and channel capacity. If \( T \) is the number of transmitting antennas and \( R \) is the number of receiving antennas then the system can transmit up to \( \{T, R\} \) symbols per time slot using spatial multiplexing. MIMO systems achieve the highest spatial diversity of TxR and increase the channel capacity by transmitting more symbols per time slot while maintaining the same diversity level [3] [4] [21].

A measure of spatial diversity gain can be computed as [A.R. TO 2.7].

\[
G_{SD} = \lim_{\gamma \to \infty} \frac{K}{\log \gamma} \tag{2.32}
\]
Where in Equation (2.32), \( K \) is the code rate provided in bits/s which is equal to the number of bits per symbol time, and the number of symbols per time slot that are spatially multiplexed.

In conclusion, there are two major types of MIMO systems. The first one \([21]\) is known as open-loop systems in which there is no information about the channel is known to the transmitter and the receiver estimate the channel for the purpose of equalization and decoding. The second type \([21]\) is called closed loop systems in which the receiver after estimating the channel shares the channel information to the transmitter through a feedback channel. The transmitter also utilizes the information to improve the overall performance of the system.
Chapter 3

CHANNEL EQUALIZATION AND ADAPTIVE ALGORITHMS

3.1 Introduction

Inter symbol interference (ISI) created by multipath within time dispersive channels is compensated by equalization. As shown in Chapter 2, if the modulation bandwidth is greater than the coherence bandwidth of the wireless channel, ISI takes place and modulation bandwidth pulses are extended in time to neighboring symbols. An equalizer located in a receiver balances for the delay characteristics and the average range of expected channel amplitude. Because the channel is generally unidentified and time varying, so equalizers must be adaptive [4].

3.2 Channel Equalization

Dispersive channels slowly varying with time can be approximated by transversal or nonrecursive filters. A recursive model can also be approximated by a relatively large transversal filter by organized approximation error. As mentioned before, the ISI occurred due multiple channel paths is more destructive than channel/receiver noise so that attempts are intended at the target of removing or at least minimizing the distortion created by ISI. In digital communication systems, the distortion is the major reason of producing bit or symbol errors.

The Channel Impulse Response (CIR) is assumed to be a real-valued function. As mentioned in Chapter 2, the channel fading models comprise independent real and imaginary components, so they can be equalized independently and justify the real valued assumption.

Recursive channel models have infinite CIRs (therefore the name Infinite Impulse Response, IIR). But their CIR decays in time when they are stable also and the truncation used is technically justified to keep only a finite support of CIR.

The approximations and models made for multipath radio channels are based on the corresponding baseband model of the communication system. Rest of the discussion contains the same baseband equivalent assumptions.
Continuous-time channel and corresponding CIR denoted by $h(t, \tau)$ is shown in Figure 3.1, where $n(t)$ is channel noise. If the CIR compact support duration is denoted by $T_c$ and the continuous-time symbol duration by $T_s$, then the received signal can be obtained by a convolution and additive noise:

$$r(t) = h(t) \otimes s(t) + n(t) = s(t) \otimes n(t)$$

$$r(t) = \int_0^{T_s} h(t-\tau)s(\tau)d\tau + n(t) = \int_0^{T_s} s(t-\tau)h(\tau)d\tau + n(t)$$

where $\otimes$ is the symbol of convolution. $T = T_c + T_s$ is the time duration of the received signal. In digital communication, signals are suitably sampled and the baseband equivalent system based on appropriate assumptions generate signal samples at each sample time $t = kT_s$, where $T_s$ represents the sampling period. The samples are acquired at each symbol time (sometimes called baud rate) that is, the symbol time and the sampling period are equal. Therefore, the CIR can be expressed as a discrete-time impulse response $\mathbf{h} = (h_0, h_1, ..., h_{N_C-1})^T$, $k = 0, 1, ..., N_C - 1$. There are two significant assumptions about CIR, that it can be taken constant relative to equalization time (slow-varying), and causal (the matter of the choice for the time reference). $s[k] = s(kT_s)$ and $r[k] = r(kT_s)$ are the transmitted and received signals expressed in discrete-time by their samples at the $k^{th}$ time step.

Consequently the transfer function of channel is denoted by $H(z)$. In general when the model is recursive, it can be of Infinite Impulse Response (IIR) and when the model is non-recursive, it
can be of Finite Impulse Response (FIR). In any case, the discrete-time channel is approximated by a transversal filter with finite length $N_C$.

The channel inversion, or ideal equalization is estimation of weight vector denoted by $w$ of a transversal filter of finite length $N_E$ so that it approximately realizes the transfer function $W(z) \approx H^{-1}(z)$.

A perfect equalization involves a doubly-infinite equalizer. The ideal transfer function, or an approximation of it, is obtained when the weight vector of the equalizer achieve some optimum value $w^*$. This means that $W(\omega) \approx H^{-1}(\omega)$ when $W(\omega) = W(z)|_{z=\exp(i\omega)}$ and $H(\omega) = H(z)|_{z=\exp(i\omega)}$ represents the frequency responses of the equalizer and the channel respectively.

It is clear that a non-minimum phase transfer function channel cannot be adequately equalized with enough equalizer delay since its inverse is unstable. Even the stable linear transversal equalizers with successful inversion can be possibly far from the desired response. Also the channel transfer functions with zeros close to the unit circle express deep-nulls (close to zero) in frequency response and are difficult to equalize. If the transfer function of the minimum phase channels or equivalently CIR is known in the MS (Minimum Squares) sense, then their Zero-Forcing equalizer can be computed see [27] [28]:

$$W_{MS}(\omega) = \frac{\sigma^2_{W}H^*(\omega)}{\sigma^2_{W}|H(\omega)|^2 + \sigma^2_n}, \quad -\pi \leq \omega \leq \pi \quad (3.3)$$

where $\sigma^2_{S}$ is the zero-mean input data variance that is also assumed to be WSS (Wide Sense Stationary), $\sigma^2_n$ represents the variance of the zero-mean channel noise process and is independent of the input data. There must be a stable channel. The linear equalizer and the channel system functions $G_N(z)$ and $G_S(z)$; for the channel noise and input signal can be given by:

$$G_S(z) = H(z)W(z), \quad G_N(z) = W(z) \quad (3.4)$$

The channel noise samples are assumed to be iid and white. The ideal equalization can be represented by $H(z)W(z) = c_0z^{-d}$ for some positive integer delay $d$. The equivalent of the channel (CIR) and the equalizer together in the time domain is a convolution:

$$c = h \otimes w \quad (3.5)$$

The outcome of the Equation (3.5) has $M = N_C + N_E$ components computed as:

$$c_k = \sum_{j=0}^{N_C} w_{k-j}h_j = \sum_{j=0}^{N_C} h_{k-j}w_j, \quad k = 0,1, \ldots, M - 1 \quad (3.6)$$

So the time-domain equivalent of an ideal equalization is expressed by the desired response $c = c_0 \delta_d$. It shows that the equalizer and the effect of the channel together generates $c_k = 0$ for $k \neq d$ and $c_k = c_0$.
The single non-zero component is also called the *cursor*. The goal of ideal inversion cannot be reached with finite-length equalizer filters, so instead of looking for completely zero components \(c_k\), there are many non-zero but expectantly very small weights other than the major delayed weight or the cursor.

The process of such inversion also has been recognized as seismic deconvolution see [29][27]. Seismic deconvolution include an acoustic waveform known as seismic wavelet is provided at a shot point with the help of special transducers, and then transmitted all the way through the terrain sub-layers. The gathered seismic traces are vigilantly united and recorded in seismograms. Then the seismograms are processed in an offline mode to determine the sub-layer’s formation. The concept of seismic deconvolution is similar to channel equalization and presents a prosperous background of research literature. The deconvolution method is effectively applied in universally used Global Positioning Systems (GPS) [30]. Their work has been pursued by other people in the GPS application described in [31][32].

In seismic application, Minimum Entropy Deconvolution (MED) is the common algorithm. The procedure is dependent on a new vector norm called varimax see [33][34], which was continued by [35] studies on a different norm he named D-norm. In fact, MED methods represent a class of solutions that employ Higher-Order Statistics (HOS) implicitly.

Because MED is executed offline (execute on recorded seismic waveform traces), therefore it is dissimilar to the adaptive and real-time methods. To conclude, alike to Equation (2.21), the general measure of equalization performance is known as residual ISI, here denoted by \(\text{ISI}_R\). [26] has also applied the worst-case residual ISI for the equalization quality measure that is called Peak Distortion.

### 3.3 Deconvolution of A-Priori Known Systems

This is a common problem relating to any system that can be modeled by a linear transversal filter comprising a finite length tap-weight vector. The main point is the knowledge of system impulse response given by \(\mathbf{h}\). We search for the finest (decrease in optimum in the sense of \(\text{ISI}_R\)) equalizer weight vector \(\mathbf{w}^*\) such that:

\[
\mathbf{h} \odot \mathbf{w}^* = \mathbf{e}_d
\]  

(3.7)

The vector \(\mathbf{e}_d\) is the standard basis, that is:

\[
\mathbf{e}_d = (0, 0, \ldots, 0, 1_d, 0, \ldots, 0)^T
\]  

(3.8)

One can anticipate the key component (cursor value) to be one as compared to \(c_k\), since the system impulse response is generally normalized to unit power, that is \(\mathbf{h} = \mathbf{h}_o/\|\mathbf{h}_o\|_2\) where \(\mathbf{h}_o\) is the response before normalization.
A improved formulation of the deconvolution problem can be acquired by the complete expansion of Equation (3.7) as follows (assuming that \( N_C > N_E \)):

\[
\begin{bmatrix}
  h_0 & 0 & 0 & \cdots & 0 \\
  h_1 & h_0 & 0 & \cdots & 0 \\
  h_2 & h_1 & h_0 & \cdots & 0 \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  h_{N_E-1} & h_{N_E-2} & \cdots & h_1 & h_0 \\
  h_{N_E} & h_{N_E-1} & \cdots & h_1 & \cdots & h_0 \\
  0 & 0 & \cdots & h_{N_C-1} & h_{N_C-2} \\
  0 & 0 & 0 & \cdots & 0 & h_{N_C-1}
\end{bmatrix}
\begin{bmatrix}
w_0^* \\
w_1^* \\
\vdots \\
w_{N_E-1}^*
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
0 \\
\vdots \\
0
\end{bmatrix}
\text{(3.9)}
\]

The matrix in Equation (3.9) is denoted by \( \mathcal{H} \). This matrix is also referred as a filtering matrix. The dimensions of filtering matrix are \((N_C + N_E - 1) \times N_E\) that makes Equation (3.7) as an over-determined system of equations. The solution in Least Square (LS) sense is basically given by the standard equation, symbolically:

\[
w^* = (\mathcal{H}^T \mathcal{H})^{-1} \mathcal{H}^T \mathbf{e}_d
\text{(3.10)}
\]

By examining Equation (3.8) for the filtering matrix, it is clear that it has complete rank of order and therefore the matrix \( \mathcal{H}^T \mathcal{H} \) having size \( N_E \times N_E \) is positive-definite and has complete rank, As a result \( (\mathcal{H}^T \mathcal{H})^{-1} \) is possible, and the equation in (3.10) has a solution (In linear algebra It is well known that positive definite matrixes are invertible with positive-definite inverses).

Since the matrix inversion has high computational cost, therefore, for the case of known impulse response it will be completed once, and most probably, in an offline mode. Efficient algorithms exist for numerical computing the solution of normal equations [36]. In many practical cases, the system is not a priori-known; the case of wireless channels means that the CIR is not known.

### 3.4 Adaptive Algorithm for Equalization of A Priori-Unknown Channels

Although information of the channel is required for reliable communication in most modern systems, this knowledge is not a-priori known and has to be obtained generally using some recursive adaptation algorithm. One can seek to determine, at the least, important characteristics of channel by the application of an adaptive identification algorithm and afterward, if necessary, apply one of the solutions for the equalizer filter.

A direct and more efficient approach is to adaptively find out the best possible weight vector of
the equalizer, without prior knowledge of the channel. The algorithms for equalizing unknown channels are alienated into the supervised mode in which a training or pilot sequence is transmitted that is known to receiver. Apparently training period uses portion of the available bandwidth and air time, and it might not be feasible and efficient in multi-user environments. In spite of resources wastage, supervised techniques are uncomplicated and assure success in convergence. [37] [38] presented excellent tutorials and references on adaptive supervised equalization. Figure 3.3 depicts the general concept of supervised algorithm, which demonstrates the adaptive training of linear (transversal) equalizer filter to search for best possible weight or state vector \( \mathbf{w}^* \).

![Figure 3.3 General concept of supervised equalization system](image)

Generally most of the cellular standards integrate various types of training signal. Here the purpose of the training sequence to estimate the CIR, then its inverse is computed in order to reduce the distortion effect caused by ISI. Once the channel equalization is carried out to achieve an adequately low residual channel ISI, the received data replaces the training signal in the adaptation process to make corrections for slow time-variation of the multipath channel. Blind equalization or estimation techniques, are based exclusively on the received signal samples, assuming that the statistical properties of the channel’s input data is known and incorporated in the corresponding computations.

### 3.5 Blind Equalization Algorithms

There exist many families of algorithms applied for identifying and equalizing unsupervised (blind) systems. Here we briefly introduce Bussgang algorithms. These algorithms absolutely use the Bussgang theorem stochastic processes. The theorem says [39]: “If the input to a memoryless possibly nonlinear system \( y = f(x) \) is a zero-mean normal process \( X(t) \), the cross-correlation
of \(X(t)\) with the resulting output \(Y(t) = f(x(t))\) is proportional to the input correlation, namely:

\[
\mathbb{E}[X(t)X(t-\tau)] = K \mathbb{E}[X(t)Y(t-\tau)] \quad (3.11)
\]

where the constant is given by \(K = \mathbb{E}[f^1(x(t))]\).

![Figure 3.4 Bussgang Theorem](image)

The Bussgang algorithms given such name since all of them use a memory-less (non-linear) function as a decision device to correctly estimate the input data of channel. For binary data case, the decision device is just a slicer that is \(u[k] = \text{sgn}(x[k])\) where \(\text{sgn}(.)\) symbolizes signum or sign function (see Figure 3.4). The estimated data is frequently used as a substitute of some a-priori known training sequences.

![Figure 3.5 Basic linear equalization system](image)

Figure 3.5 shows a switch that is available to change the mode of the algorithm from training mode to blind mode, or the other way around. We continue with the assumption that the channel is slowly varying in time and constant for the purpose of the following discussion. The received signal samples can be computed as follows:
\[ r[k] = \sum_{j=0}^{N_C-1} h[j - k] + n[k] \quad (3.12) \]

\[ r[k] = \mathbf{h}^T \mathbf{s}[k] + n[k] \quad (3.13) \]

where \( \mathbf{s}[k] = (s[k], s[k-1], ..., s[k-N_C+1])^T \) represents the the input symbols vector in Equation (3.13), or data to the channel in the size of the CIR. Similarly the output of the equalizer can be computed by the received signals:

\[ g[k] = \sum_{j=0}^{N_C-1} w[j] r[k - j] \quad (3.14) \]

In the short form, it can be:

\[ g[k] = \mathbf{w}^T[k] \mathbf{r}[k] \quad (3.15) \]

In Equation (3.15) \( \mathbf{w}[k] = (w_0[k], w_1[k], ..., w_{N_E-1}[k])^T \), is the weight vector of linear equalizer at \( k^{th} \) step of the adaptation, and \( \mathbf{r}[k] = (r[k], r[k-1], ..., r[k-N_E+1])^T \) is the input signal samples vector with the size of the equalizer. Finally, the binary data estimate is computed by the slicer, that is:

\[ u[k] = \hat{s}[k - d] = \text{sgn}(g[k]) \quad (3.16) \]

Like optimization problem, one needs to describe a cost function. The common Mean Square Error (MSE) objective function, which completely depends on the weight vector of equalizer, is described as follows:

\[ J(\mathbf{w}[k]) = \mathbb{E}\{e^2[k]\} = \mathbb{E}\{(s[k - d] - g[k])^2\} \quad (3.17) \]

When the actual channel input is not available in blind mode, its estimate is used in its place. This mode is known as decision-directed mode.

\[ J(\mathbf{w}[k]) = \mathbb{E}\{\hat{s}[k - d] - g[k]\}^2 \quad (3.18) \]

Some equalizers initializes the method in the training mode and then switches to decision-directed mode after the channel is approximately equalized in order to adaptively track the small variations of the channel.

One important quality of Bussgang class is the relatively simple procedures. They utilize the renowned Least Mean Squares (LMS) method that is one of the the simplest techniques among the recursive adaptation for optimization [40]. In basic gradient method of LMS, the update
equation is very simple as follows:

\[ \mathbf{w}[k + 1] = \mathbf{w}[k] - \mu \nabla J[k] \quad (3.19) \]

where the learning rate or step size is denoted by \( \mu \). The gradient of the cost function is specified by \( \frac{\partial J}{\partial \mathbf{w}[k]} = \nabla J(\mathbf{w}[k]) \), when the reliance on the equalizer weight vector is clearly shown.

\[
\nabla J(\mathbf{w}[k]) = \frac{\partial J}{\partial \mathbf{w}[k]} = \frac{\partial}{\partial \mathbf{w}[k]} \mathbb{E}\{e^2[k]\} = \frac{\partial}{\partial \mathbf{w}[k]} \mathbb{E}\{(u[k] - g[k])^2\} \\
= \mathbb{E}\left\{ \frac{\partial}{\partial \mathbf{w}[k]} (u[k] - g[k])^2 \right\} \quad (3.20)
\]

The derivative of the error can be computed as:

\[
\frac{\partial}{\partial \mathbf{w}[k]} e[k] = \frac{\partial}{\partial \mathbf{w}[k]} (u[k] - g[k]) = \frac{\partial}{\partial \mathbf{w}[k]} u[k] - \frac{\partial}{\partial \mathbf{w}[k]} g[k] \\
= \frac{\partial}{\partial \mathbf{w}[k]} \text{sgn}(g[k]) = 2 \delta[k] \quad (3.21)
\]

\[
\frac{\partial}{\partial \mathbf{w}[k]} g[k] = \frac{\partial}{\partial \mathbf{w}[k]} (\mathbf{w}^T[k] \mathbf{r}[k]) = \mathbf{r}[k] \quad (3.22)
\]

The result of Equation (3.22) is only valid when \( g[k] \) is very close to zero (dissimilar to \( g[k] \) that has to be close to +1 or -1 after the convergence of equalizer) and can be ignored. Consequently, the gradient of the cost function is given by:

\[
\nabla J(\mathbf{w}[k]) = -2 \mathbb{E}\{e[k] \mathbf{r}[k]\} = -2 \mathbb{E}\{(u[k] - g[k])\mathbf{r}[k]\} - 2 \mathbb{E}\{\mathbf{s}[k - d] - g[k]\} \mathbf{r}[k] \quad (3.24)
\]

The reason behind the simplicity of the LMS technique is that it takes the simplest possible estimate and probably not quite exact gradient by simply removing the expectation operator. That is

\[
\mathbb{E}\left\{ \frac{\partial}{\partial \mathbf{w}[k]} e^2[k] \right\} \approx \frac{\partial}{\partial \mathbf{w}[k]} e^2[k] = -2e[k] \mathbf{r}[k] \quad (3.25)
\]

Therefore, the equalizer weight vector update simplifies to the following one (the coefficient 2 in (3.25) is engrossed in the step size \( \mu \))

\[ \mathbf{w}[k + 1] = \mathbf{w}[k] - \mu \nabla J(\mathbf{w}[k]) = \mathbf{w}[k] + \mu e[k] \mathbf{r}[k] \quad (3.26) \]
3.6 Bussgang Algorithms

Figure 3.6 demonstrate the overall system that includes the channel and the linear equalizer for which a transversal filter with finite number of taps is used.

As mentioned before, the channel noise sample \( n[k] \) is comparatively less compared to ISI measure and its effects have to be considered until most of the ISI has been eliminated. Consequently, the channel noise can be ignored: that means \( \bar{r}[k] \approx r[k] \), where \( \bar{r}[k] \) represents the noise-free signal at the receiver. Again, assume that the CIR (Channel Impulse Response) corresponds to the transmitter filter, the receiver filter and the multipath channel together.

\[
\bar{r}[k] = h \ast s[k] = \sum_{i=0}^{N_c-1} h[i] s[k-i] \approx r[k] \quad (3.27)
\]

\[
\bar{g}[k] = w[k] \ast r[k] = \sum_{i=0}^{N_E-1} w_i[k] r[k-i] \approx g[k] \quad (3.28)
\]

In Equations (3.27) and (3.28) the subscript \( k \) points to the vector values at the \( k \)-th step. The subscript \( k \) in the weight vector of equalizer \( w_k \) could be removed for expediency and be considered as disguised when it simplifies the equations. Equation (3.28) can be expanded as

\[
\bar{g}[k] = w[k] \ast h \ast s[k] = \sum_{j=0}^{N_E-1} w[j] s[k-j-i] \quad (3.29)
\]

Let \( w^* \) denote the optimal weight vector of the linear equalizer. After including optimum weight vector, the Equation (3.28) can be written as

\[
\bar{g}[k] = \sum_{i=0}^{N_E-1} w_i[k] r[k-i] = \sum_{i=0}^{N_E-1} w^*_i r[k-i] = \sum_{i=0}^{N_E-1} (w_i[k] - w_i^*) r[k-i] \quad (3.30)
\]

In Equation (3.30), the first term on the right-hand side is the optimum equalized signal that represents the accurate estimate of the appropriately delayed input signal to the channel \( s[k-d] \).

![Figure 3.6 Basic equalization system](image)
\[ \sum_{i=0}^{N_E-1} w_i^r[k - i] = c_0 \delta[k - d] \quad (3.31) \]

Where \( c_0 \) in Equation (3.31) is the cursor value (see Equation (3.5)). The second term in (3.30) is known as convolution error that is the undesired remainder.

\[ I[k] = \sum_{i=0}^{N_E-1} (w_i[k] - w_i^r)r[k - i] = \sum_{i=0}^{N_E-1} e_i[k]r[k - i] \quad (3.32) \]

The convolution error is approximated to be zero-mean, Gaussian, and independent of the input sequence \( s[k] \) (See [41]).

The last part in Figure 3.5 is a zero-memory (memoryless) nonlinear device in most of the implementations, whose function is denoted by \( Q(.\)\). The choice of nonlinear function is the most important factor on the performance of a particular algorithm. The system with binary input data generally uses a slicer Function for the zero memory nonlinearity.

Now turn attention to the Bussgang family of algorithms. [42] and [28] have presented great description of the general Bussgang algorithms as the base for most of the known techniques in linear equalization. Let us consider again the effect of the channel and the equalizer together by their convolution denoted by \( c[k] = \mathbf{h} \odot w[k] \):

\[ c_i[k] = \sum_{i=0}^{N_C-1} h_{i-j}w_{i-j}[k], \quad i = 0,1,2,...,M - 1 \quad (3.33) \]

\[ \tilde{g}[k] = \sum_{j=0}^{M-1} c_j[k]s[k - j] = c_d[k]s[k - d] + I[k], \quad M = N_E + N_C \quad (3.34) \]

In Equation (3.5) and (3.31), the cursor value is denoted after the algorithm has converged by \( c_d[k] = c_0 \) and then the convolution error is as follows:

\[ I[k] = \sum_{j=0}^{M-1} c_j[k]s[k - j] \quad (3.35) \]

Equation (3.35) is simply a reformulation of Equation (3.32) in terms of the convolution samples \( c_j[k] \).

After convergence of the equalizer to the final point, Equation (3.34) should become \( \tilde{g}[k] = c_0 s[k - d] + I[k] \) having the minimum possible value for the convolution error. When the distribution (pdf) of \( I[k] \) is available, the decision device is a Maximum A-Posteriori (MAP) estimator as described in the following equation.

\[ \hat{s}[k - d]_{MAP} = \arg\max_s P_{\tilde{g}[k]|s[k-d]}(\tilde{g}[k]|s) \quad (3.36) \]

When the distribution of \( I[k] \) is not known, particularly the conditional probability in (3.36) is
not available, making the same approximations by assuming that the input sequence \( s[k] \) is zero-mean iid. Thus the \( c_i[k]s[k - j] \) are independent terms in \( I[k] \) (Equation 3.36) and have finite variance so that the application of the central limit theorem entails that the \( I[k] \) distribution is approximately Gaussian and is independent of \( s[k - d] \). It is also assumed that the process \( I[k] \) to be zero-mean and its variance can be obtained with the knowledge of the variance of the input data sequence (namely \( \sigma_s^2 \)) as follows.

\[
\sigma_I^2 = \sigma_s^2 \sum_{j=0}^{M-1} |c_i[k]|^2 \quad (3.37)
\]

By taking into account \( I[k] \) as the output noise of equalizer with the variance specified by (3.37), the optimum memoryless nonlinearity in the sense of MAP will be the minimum variance estimation as in Equation (2.40) [28]:

\[
Q((g[k]) = \hat{s}[k - d]_{\text{MAP}} = \mathbb{E}\{s[k - d]|g[k]\} \quad (3.38)
\]

The SNR ratio at the equalizer output can now be computed easily by:

\[
\frac{\sigma_s^2 \mathbb{E}[|s[k]|^2]}{\sigma_I^2} \quad (3.39)
\]

In Equation (3.39) \( \mathbb{E}[|s[k]|^2] \) is the input sequence power. The information in Equation (3.39) must be available in order to find the nonlinearity function \( Q(\cdot) \) and when this information is not obtainable or difficult to estimate, the Bussgang algorithms are not optimum then [28]. At last when the nonlinearity has been found, one can solve for the equalizer weights iteratively in the Least Squares (LS) sense using the following update equation:

\[
w[k+1] = (\mathbb{E}\{r[k]r^T[k]\})^{-1}\mathbb{E}\{r[k]Q(r^T[k],w[k])\} \quad (3.40)
\]

One of the simplest adaptive methods is the gradient descent algorithm that can be applied to search for the optimal weight vector with the following update equation.

\[
w[k+1] = w[k] + \mu(Q(g[k]) - g[k]r[k]) \quad (3.41)
\]

In the recursive techniques of Equations (3.40) and (3.41) \( r[k] \) is assumed as the noise-free signal at the receiver.

It has been revealed that as the equalizer is converged, the Bussgang condition is satisfied by the output.

\[
\mathbb{E}\{g[k]g[k - \ell]\} = \mathbb{E}\{g[k]Q(g[k - \ell])\}, \forall \ell \quad (3.42)
\]

Im short, the convergence of the Bussgang algorithm is not known mostly. The local
convergence of the Bussgang algorithms has been proven by [43]. The linear equalization techniques that we analyze in this section involve same basic concept apart from the choice of zero-memory nonlinearity device. They use gradient descent and LMS (Least Mean Square) algorithm in the same way for their iterative adaptation process.

A. The Sato Algorithm

The Sato algorithm is conceivably the simplest technique applied for the purpose of blind equalization and was developed early in the rich research history. It is also one of the algorithms classified as Decision-Directed Equalizers (see Figure 3.7.)

![Figure 3.7 Basic Decision Feedback Equalizer diagram](image)

This algorithm and some improved versions of it that are mentioned here are appropriate for PAM (Pulse Amplitude Modulation), including the simple binary case (2-PAM) for which the nonlinearity is simply \( Q(.) = \text{sgn}(.) \). The cost function that was chosen by Sato without any theoretical justification was:

\[
J(w[k]) = \mathbb{E}\{(g[k] - \gamma_1 \text{sgn}(g[k]))^2\} \quad (3.43)
\]

In Equation (3.43), the choice of \( \gamma_1 \) is an important parameter that is given in Equation (3.45). Therefore the gradient descent based update equation, will be given by the following equation:

\[
w[k+1] = w[k] - \mu (g[k] - \gamma_1 \text{sgn}(g[k])) r[k] \quad (3.44)
\]

The coefficient of \( \frac{1}{2} \) of the gradient in the update equation is absorbed in the step size \( \mu \). The Sato choice for \( \gamma_1 \) was instinctively chosen as:

\[
\gamma_1 = \frac{\mathbb{E}[|s[k]|^2]}{\mathbb{E}[|s[k]|]} \quad (3.45)
\]

This choice only needs the information about input data statistics that are assumed to be
available in advance. The comparison between the Sato algorithm and the simple LMS can be as:

\[ \mathbf{w}[k + 1] = \mathbf{w}[k] - \mu (g[k] - \hat{s}[k - d]) \mathbf{r}[k] \]  

where \( \hat{s}[k - d] = \text{sgn}(g[k]) \) determines the delayed input estimate. Thus, the only dissimilarity is that \( \gamma_1 = 1 \) in the LMS algorithm. In Sato algorithm, the success (convergence) has been ascribed by the choice of \( \gamma_1 \). Consequently, the convergence of the Sato algorithm is based on the probability of the event when the sign of the real error \( g[k] - s[k - d] \) and its estimate \( \gamma_1 \text{sgn}(g[k]) \) agree. Sato’s method was proposed for one dimensional multilevel PAM that is the case of binary signaling. The Benveniste-Goursat-Ruget theorem for convergence [42] using the assumption of doubly-infinite filter size be appropriate to Sato’s method. The theorem states that the global convergence of the Sato can be gained for the input distributions of continuous non Gaussian or sub-Gaussian in particular uniformly distributed input sequences. [44] and [45] have proved that the Sato’s convergence to local minima for QAM (Quadratic Amplitude Modulation) input signals instead of the optimum global minimum. Furthermore, [28] has also shown the possibility of converging to any local minima for multi-level signaling.

B. The Godard Algorithm

[46] introduced a family of algorithms for blind equalization in general M-ary QAM systems known as Constant-Modulus Algorithms (CMA). The QAM systems represent two dimensional digital communication systems. The actual idea is to use a cost function that is based only on the magnitude of the equalizer output, which that the cost function is independent of the phase of equalizer output. A cost function of \( p \)-th order (where \( p \) is a positive integer) is defined by:

\[ J(\mathbf{w}[k]) = \frac{1}{p} \mathbb{E} \left\{ (|g[k]|^p - \gamma_p)^2 \right\} \]  

(3.47)

The new Godard parameter \( \gamma_p \) is a real and positive constant.

\[ \gamma_p = \frac{\mathbb{E}|g[k]|^{2p}}{\mathbb{E}|g[k]|^p} \]  

(3.48)

It appears that the constant \( \gamma_p \) has replaced the unknown input sequence \( s[k] \) in the error samples, which holds the knowledge about the input data distribution. In other words, the Godard algorithm is intended to determine the divergence of the equalizer output \( g[k] \) from a constant modulus \( \gamma_p \) and use it for adaptation.

The general \( p \)-th order error signal is given by

\[ e[k] = (|g[k]|^p - \gamma_p)|g[k]|^{p-2}g[k] \]  

(3.49)
The corresponding update equation can be given by

$$\mathbf{w}[k + 1] = \mathbf{w}[k] - \mu e[k] r[k] \quad (3.50)$$

When of $p = 1$, the cost function decreases to one that can be regarded as a modified Sato algorithm.

$$J(\mathbf{w}[k]) = \mathbb{E}[(|g[k]| - \gamma_1)], \quad \gamma_1 = \frac{\mathbb{E}[|g[k]|^2]}{\mathbb{E}[|g[k]|]} \quad (3.51)$$

For the case of $p = 2$, the cost function is CMA (Constant-Modulus Algorithm), which is well-known in the literature. Equation (3.52) shows the modulus parameter and the corresponding cost function for this important case:

$$J(\mathbf{w}[k]) = \mathbb{E}[(|g[k]|^2 - \gamma_2)], \quad \gamma_2 = \frac{\mathbb{E}[|g[k]|^4]}{\mathbb{E}[|g[k]|^2]} \quad (3.52)$$

The CMA algorithm is proposed for systems which contain the symbols with same amplitude (constant) and different phases in the corresponding constellation of the QAM system and are not possible to be used for the cases of multi-level PAM. Finally, we mention that the algorithm in [46] has better performance as compared to the other Bussgang algorithms in the MSE sense. The algorithm has the capability of equalizing a dispersive channel even when the eye (of eye-pattern) is not initially open [42].

The problem of the system identification or blind equalization has been challenged by many researchers, and their results have concluded in many algorithms, each with some benefits. Of these algorithms, several belong to the family of Bussgang algorithms, which have the Bussgang property of zero memory nonlinearity.

BGR (Benveniste-Goursat-Ruget) is an extended version of the Sato algorithm with the target of approaching the start-up adaptation from the local (nearest) of the optimal tap weights (it was stated that the Sato algorithm has guaranteed convergence when the initial weights are close to the global optimum point.)

[47] introduced the Stop-and-Go algorithm that basically implements the similar adaptation as Sato’s and the following technique excluding that the adaptation at each iteration can be passed over (no change to the state vector) based on the assessment of a condition that flags possible estimates which not correct. This, for example, can be done with the help of a comparison between signs of two errors each belonging to a separate algorithm of this family.
3.7 Blind Channel Equalization using Diversified Algorithm

In wireless channel equalization, blind techniques are abundant and diversified; nevertheless, a fast convergent and robust technique with practically efficient computations is still as an open problem. We have briefly discussed only some of these techniques in previous sections. In particular, the algorithms using Higher Order Statistics (HOS) and Fractionally-Spaced Equalization (FSE) were not reviewed. The reason of ignoring them is the fact that various techniques of HOS and FSE have high computational costs and our target systems require less computational cost with multiple algorithms.

3.7.1 Recursive Least Squares Adaptive Algorithms

In this section we briefly introduce a type of algorithms known as Recursive Least Squares (RLS). RLS algorithms include the inverted covariance matrix of the process (wireless channel) in the update equation that is similar to the Newton-like algorithms or Wiener optimum filter solution, in adaptive techniques. Their end performance measures are frequently poor. RLS techniques and the resulting outcomes are well-known and long term. The references [29][48] and [49] provide complete derivation and proofs of the methods. A related application of RLS in digital communications is specified in [50].

Let us consider the problem of adjusting the weight vector iteratively of a transversal filter applied for equalization of a wireless channel. Figure 3.9 shows a general system setup that
includes the generated received signals unknown to wireless communication channel.

An estimate of the auto-covariance matrix of the method generating the input vector \( \mathbf{r}[k] \) at step \( k \) is given by

\[
\mathbf{R}_k = \frac{1}{k} \sum_{i=1}^{k} \mathbf{r}[i] \mathbf{r}[i]^T
\]  

The simplified running estimate of \( \mathbf{R}_{k+1} \) is commonly as follows.

\[
\mathbf{R}_{k+1} = \left( \frac{k}{k+1} \right) \mathbf{R}_k + \left( \frac{1}{k+1} \right) \mathbf{r}[k+1] \mathbf{r}[k+1]^T
\]  

The basic update equation of weight vector in the RLS type of algorithm includes the inverse covariance matrix. This equation for advancing from \( k \)th step to next can be written as:

\[
\mathbf{w}_{k+1} = \mathbf{w}_k - \mu e[k] \mathbf{R}_k^{-1} \mathbf{r}[k]
\]  

where \( \mu \) is the step size usually equal to one in common RLS implementations, and the error is computed as the difference of the filter output and the desired output \( d[k] \).

\[
e[k] = d[k] - \mathbf{w}_k^T \mathbf{r}[k]
\]  

The required inverse of the covariance matrix for the weight vector update equation needs to be updated in each step with the help of running update Equation (3.54).
The famous Woodbury’s matrix inversion identity is generally used to avoid the costly matrix inversion at each step of adaptation to obtain

$$R_{k+1}^{-1} = (R_k + r[k+1]r^T[k+1])^{-1} \quad (3.57)$$

For ease the inverse of the matrix is denoted by $Q_k = R_k^{-1}$ and the last simplified equation is given by

$$Q_{k+1} = Q_k - \frac{Q_k r[k+1]r^T[k+1]Q_k}{1 + r^T[k+1]Q_k r[k+1]} \quad (3.59)$$

The update equation turns into

$$w_{k+1} = w_k + \mu e[k]Q_k r[k] \quad (3.60)$$

It is suggested in [29] to initialize the weight vector to a zero vector $w_0 = 0$ and the inverse covariance matrix to a large diagonal matrix $Q_o = \delta^{-1}I$ where $\delta$ is a small constant.

### A. Exponentially Weighted Recursive Least Squares

A famous version of the RLS algorithm is the Exponentially Weighted version (EWRLS) in which a forgetting factor typically close to one is used to decrease the effects of comparatively old received information. Specifically in the channels that are modeled by limited-size transversal filters the correlation between the ones belonging to many steps before and the recently received signals is trivial if not zero, therefore the applications of forgetting factors in Exponentially Weighted RLS of the range of $\lambda = 0.99$ to $\lambda = 0.999$ have shown good results, particularly in the case of time-varying channels. The EWRLS algorithm equations including the forgetting factor $\lambda$ are same as in regular RLS and are concluded in the following ($Q_k = R_k^{-1}$).

$$R_{k+1} = \lambda R_k + r[k+1]r^T[k+1] \quad (3.61)$$

$$Q_{k+1}^{-1} = \lambda Q_k^{-1} + r[k+1]r^T[k+1] \quad (3.62)$$

$$Q_{k+1} = \frac{1}{\lambda} \left( Q_k - \frac{Q_k r[k+1]r^T[k+1]Q_k}{\lambda + r^T[k+1]Q_k r[k+1]} \right) \quad (3.63)$$

$$w_{k+1} = w_k + \mu e[k]Q_{k+1} r[k] \quad (3.64)$$
B. Quantized-State Recursive Least Squares

[51] proposed four different algorithms that are similar to the RLS algorithms in structure, to reduce the computational effort and accelerate the convergence. These Quantized-State (QS) algorithms have been effectively applied to few problems constantly [52].

The first two QS algorithms depend on a sliced input signal vector \( \hat{r}[k] \), in which each element is found by \( \hat{r}_j[k] = \text{sgn}(r_j[k]) \). As a result each element of the input vector (also called state) belongs to the binary field of \{+1, −1\}, and collectively they are called the quantized state. The variations between these two proposed Quantized-State algorithms can be easily seen in the following summary of the sets of recursive equations.

Quantized-State 1 (QS-1):

\[
Q_{k+1}^{-1} = \lambda Q_k^{-1} + r[k+1]r^T[k+1], \quad Q_k = R_k^{-1}
\]  (3.65)

\[
Q_{k+1} = \frac{1}{\lambda} \left( Q_k - \frac{Q_k \hat{r}[k+1]r^T[k+1]Q_k}{\lambda + r^T[k+1]Q_k r[k+1]} \right)
\]  (3.66)

\[
w_{k+1} = w_k + \mu e[k]Q_{k+1} \hat{r}[k]
\]  (3.67)

Quantized-State 2 (QS-2):

\[
Q_{k+1}^{-1} = \lambda Q_k^{-1} + \hat{r}[k+1]\hat{r}^T[k+1], \quad Q_k = R_k^{-1}
\]  (3.68)

\[
Q_{k+1} = \frac{1}{\lambda} \left( Q_k - \frac{Q_k \hat{r}[k+1]r^T[k+1]Q_k}{\lambda + r^T[k+1]Q_k r[k+1]} \right)
\]  (3.69)

\[
w_{k+1} = w_k + \mu e[k]Q_{k+1} \hat{r}[k]
\]  (3.70)

The error at \( k^{th} \) step is \( e[k] = d[k] - w^T[k]r[k] \) in all the above equations. It must be elucidated that the learning rate \( \mu \) in QS algorithms has different purpose and effect compared to the RLS and LMS algorithms. In the LMS algorithm, the learning rate (or step size) determines the stability (see [40] or [53]).

For stationary processes, it is well-known that the covariance matrix is Toeplitz and symmetric. Though, the inverse of the covariance matrix is symmetric but not completely Toeplitz. [51] Utilized this fact to more accelerate the required computation. This method directed to two more algorithms in [51].

The purpose of this chapter is not to assess the performance of the each algorithm, which have their own well-known advantages illustrated and documented in the corresponding literature.
An excellent example is QS algorithms, from which the QS-2 is applied to support other algorithms. In practice the step size \( \mu \) for QS algorithms should be adjusted and the equalizer taps be increased to attain high quality performance, however QS-2 has been used for support but no such adjustment to corresponding parameters is made.
4.1 Introduction

Neural networks are looked as intense interconnections of simple computational elements known as perceptrons that are simplified models of neurons present in the human brain. This model is based on biological nervous system according to our understanding. Neural networks also have been taken as extremely parallel networks compiled of several computational elements prearranged in parallel to each other and in turn cascaded to other elements in various layers. Every introduction to neural networks comprises of a basic nervous composition to express the similarity to the human brain. Neural nets have huge potential applications in the field of pattern recognition, speech processing and multi-dimensional signal processing. These applications typically search for high parallelism and high computational costs. In recent years the method of neural nets has been implemented for other engineering problems also including wireless channel equalization and adaptive control. Even though there are early traces of neural nets in the literature by [54], the novel efforts by [55] deserve the distinction of being the first formal study of what was called perceptron analysis. In fact these are adaptive linear combiners improved by a nonlinear function block (See Figure 4.1) [56] and later [57] proposed adaptive resonance theory and the structural design that became known as self-organizing neural nets for pattern recognition. [58] developed Hopfield models for the group of recurrent (feedback) networks, which in combination of an algorithm known as back propagation established huge achievement in various applications. The creation of the early neural network systems was in progress at the same time that other adaptive signal processing algorithms, particularly the LMS algorithm, were effectively applied to several scientific and engineering problems [59]. Some unsatisfactory experiences with this new concept (in addition to the fact that attempts to derive learning rules for neural networks with many layers were not successful) led to a challenge in the research of this field for about a decade. [60] proved the inadequacy of perceptron techniques or specifically the single-layer network in cases where linear classifications were not achievable. They recommended the use of Multi-Layer Perceptrons (MLP) that has nonlinear behavior. [61] also presented a treatment of this subject.
Figure 4.1 Basic perceptron architecture

Most of the neural nets have *feed forward* classifier architecture. [62] developed backpropagation learning algorithm which has turn out to be the most implemented algorithm in lots of applications of neural nets including the applications of signal processing. His work remained unfamiliar until [63] rejuvenated the back propagation algorithm in his MIT report. After the work by [63] and some other people, the research on neural networks gained momentum in the research community, some proposed a few modifications to the architecture of neural nets using backpropagation algorithm and provided the choice in using activation function. Particularly, it has been implemented for the problems of system identification and channel equalization (see [64], [65], [66], [67], [68], [69], [70], [71], [72], [74], [75], [76], [77], and [78]).

[78] proposed the use of *sigmoid* function \( \varphi(x) = \frac{1}{1+e^{-ax}} \) in place of the signum function (\( sgn(.) \)) that was also famous as slicer or hard-limiter to adjust the nonlinear activation. In general, the proposal of the combination of back propagation learning algorithm with multilayer architecture has achieved significant results in many applications. We conclude this section by briefly comparing neural nets with linear adaptive filtering.

The major advantage of neural nets is clearly their immense parallel distributed composition that affords the network strong computing power. Another unique advantage is the generalization, through which the neural nets perform learning process. In short, a trained network can produce reasonably accurate and expected outputs for the inputs that have not been involved during the training process. Therefore, neural nets have the capability of resolving complicated problems. In some massive and hard to solve problems many networks are employed, each having a specific job to execute, so that a suitable structure for each network can be selected and configured. [79] has concluded the main features of general neural networks that are briefly given in the following.
• Nonlinearity: nonlinearity is the well-known feature of neural nets. Although a neural net can also be linear in design; however, their dispersed nonlinearity is utilized in applications. The nonlinear channels cannot be equalized effectively by a linear equalizer; hence, the use of neural networks has become an essential option.

• Input-output mapping: supervised learning is the process of adjusting the synaptic connection weights and bias values by providing a set of training inputs or examples and computing the network error using the target output (desired response). If the synaptic weights have reached fixed and stable values with an adequate number of training inputs, then the network has formed an input-output mapping for a particular problem. It is an exceptional feature of neural networks that unlike other channel equalization techniques, no statistical assumptions are required about the possible input set.

• Adaptivity: neural nets have the capacity to adapt their synaptic weights according to changes occur in the inputs or system environment.

4.2 Fundamental Theory of Neural Networks

The credit goes to [80] for proposing the neural networks as they are known today; they developed a simple model known as “neuron”. However, the perceptron in [81] is recognized as the major building block for several architectures of neural networks which are under investigation during the last two decades. The network structure of the perceptron is very simple and similar to a linear combiner except an addition of bias (constant input) and hard-limiter (signum function). Later on, it was replaced by other nonlinear functions such as hyperbolic tangent or sigmoid etc. here denoted by \( \varphi(.) \) in different applications and configurations shown in Figure 4.2.

The procedure for obtaining the output of perceptron \( \mu \) is referred to as activity or application of non-linear function and computed by simple summation. The perceptron is being trained adaptively for the \( k^{th} \) step or epoch shown in following equations

\[
\begin{align*}
v[k] &= \sum_{j=1}^{L} w_j[k]z_j + w_0b \\
u[k] &= \varphi(v[k])
\end{align*}
\]
[81] proposed the adaptive learning algorithm to adjust the free parameters that are weights of synaptic connections $w_j$, $j = 1, 2, \ldots, L$ for a perceptron with $L$ connections. An additional parameter can be employed for the bias input connection that is known as bias value generally constant with $b = -1$ in many designs. The bias is not an essential part of the perceptron in some applications as it doesn’t affect the performance.

This model was originally developed to solve the pattern recognition or classification problems that’s why they are also known as classifiers. They provided significant results when sets of different patterns were linearly separable in pattern recognition, for example by a hyperplane in $L$ dimensional space or by a straight line in two-dimensional space. The well-known “perceptron convergence theorem” was developed by [82], and some other presented this theorem with minor changes [79]. A linear combination of inputs and the bias form the simplest perceptron equation and generate the activity $v = \sum_{j=1}^{L} w_j[k]z_j + w_o b$. This simplest perceptron separates out two classes of input by a hyperplane through following equation.

$$\sum_{j=1}^{L} w_j z_j + w_o b = 0 \quad (4.3)$$

In early perceptron applications the final decision was made by using a hard-limiter $u = sgn(v)$. The synaptic weights $w_j$ of the perceptron were adapted iteratively. The adaptation requires first a method of calculating the output error and then a method to adjust the weights according to the calculated error.

It is the technique developed for these two steps that helps in creating advanced algorithms and network architecture. The single neuron perceptron has a very limited performance capability; it
can separate only two classes that are linearly separable and is not capable of employing nonlinear functions and more complicated functions. The network must be extended to more layers containing similar neurons.

The only option for further assessment of neural network capability and performance was to extend the network by including more layers containing more neurons. These additional layer and neurons bring capability of separating more than two classes and apparently removes the linear separability constraint. Advantages come with the cost of more computation in addition to the complexity of error and correction formulation. Later on sigmoidal function \( \varphi(x) = \frac{1}{1+e^{-ax}} \) and hyperbolic tangent \( \varphi(x) = \frac{1-e^{-ax}}{1+e^{-ax}} \) were used as the activation functions replacing hard limiter or threshold \( \varphi(x) = sgn(x) \).

### 4.3 Network Architectures and Algorithms

In general neural network architectures, the neurons are joined in the structure of layers. Neurons in the same layer share the same set of inputs in many designs. Networks with only one layer are called single-layer feed forward demonstrated in Figure 4.2. Each neuron in the same layer computes an output, together these outputs can be supplied to input of another layer of neurons, and thus constructing a two-layer feed forward network. Similarly adding more layers creates multilayer feed forward networks shown in Figure 4.3. If the final outputs of the last layer are fed back to the input of the network the new configuration is called recurrent [83]. It also validates the ‘feed forward” modifier for non-recurrent networks.

The parameters that differentiate the latest neural networks are the rule of determining the error \( e[k] \) at each step or epoch, the adaptive method of correcting the weights in each layer corresponding to its error and the choice of using activation function \( \varphi(\cdot) \). The learning process is divided into two types based on the calculation of output error, supervised learning when the target or expected output is known to the network, and unsupervised or blind learning when required output is not available to the network. The modern neural network architectures that are applied for the problem of channel equalization are similar to the typical architectures used for other applications in the light of the theorem given by [84] that states “any continuous function in a closed interval can be approximated by back propagation neural network with one hidden layer.”
A multilayer perceptron example of two layer model with $N$ neurons in the input and $M$ neurons in the output layer.

Decision boundaries for single and two layer networks [12].
A one-layer network classifier can only provide linear decision boundaries, given sufficient number of hidden units, two- or higher-layer networks can apply arbitrary decision boundaries described in Figure 4.4. The decision regions need not be curved, nor connected [12].

4.3.1 The Back Propagation Algorithm

The reason for the success and popularity of adaptive supervised training of multilayer neural networks is the development of back propagation algorithm over recurrent networks. The credit goes to [85] for developing back propagation. Their development has dissatisfied the [61] argument of deficiency in multilayer networks. The network operates in two phases. The first phase is forward propagation in which the output error of the current step is computed with initial synaptic weight values for each layer by the input to output of the network layer by-layer calculations. The second phase is called back propagation in which the gradient of error surface with respect to synaptic weights is computed for each layer from the output layer backward towards the input layer of the network and then it is used for updating the weights of each synaptic connection neuron to neuron in each layer. This is also the kernel of what we know as stochastic approximation explained in chapter 3. It can be concluded that these achievements of the back propagation method has revived the study and application of neural networks.

![Figure 4.5 A two layer model](image-url)
A. Training mode

Training is the process in which weights and bias values of neural networks are updated. The estimate of wireless channels can be calculated in terms of the weights and the biased values of NN. The back-propagation algorithm given in [12] used for the training of NN is as below,

**Step 1:** initialize weights (set to small random values)
**Step 2:** while stopping condition is false, do steps 3 - 10
**Step 3:** for each training symbol, do steps 4 – 9

**Feed forward:**

**Step 4:** $i_l$ are the inputs for 1st layer. Broadcast these input signals to all units in the layer next (the Layer#2)
**Step 5:** each hidden unit $j_m$ sums its weighted input signals and applies its activation function (sigmoid) to compute its output signal,

$$J = \text{sig}(B\{j\} + \sum_{l=1}^{N} i_l W\{jl\}) \quad (4.3)$$

Send this signal to all units in the next layer (i.e. output layer)

**Step 6:** Each output unit $k_n$ sums its weighted input signals and applies its activation function (sigmoid) to compute its output signal.

$$K = \text{sig}(B\{k\} + \sum_{n=1}^{N} j_n W\{kn\}) \quad (4.4)$$

**Back propagation of error:**

**Step 7:** Each output unit $k_n$ receives a target pattern corresponding to the input training pattern, computes its error information term.

$$\delta_{k_n} = (\hat{x}_n - k_n) \text{ sig}'(k_n) \quad (4.5)$$

Calculate its weight correction term (used to update $W\{kj\}$ later)

$$\Delta W\{kj\} = \alpha \delta_K K \quad (4.6)$$

Calculate its bias correction term (used to update $B\{k\}$ later)

$$\Delta B\{k\} = \alpha \delta_K \quad (4.7)$$

Send error $\delta_K$ to the units in the previous layer (i.e. Layer#1)
Step 8: each hidden unit $j$ sums its $\delta_K$ inputs (from units in the layer next). Multiply by the derivative of its activation function to calculate its error information term.

$$\delta_j = \delta_{inj} \cdot \text{sig}'(j)$$  \hspace{1cm} (4.8)

Calculate its weight correction term (used to update $W\{ji\}$ later)

$$\Delta w\{ji\} = \alpha \delta_j I$$  \hspace{1cm} (4.9)

Calculate its bias correction term (used to update $B\{j\}$ later)

$$\Delta B\{j\} = \alpha \delta_j$$  \hspace{1cm} (4.10)

Update weights and biases:

Step 9: each output unit $k_n$ updates its bias and weights

$$W\{kj\} = W\{kj\}_{old} + \Delta w\{kj\}$$  \hspace{1cm} (4.11)

$$B\{k\} = B\{k\}_{old} + \Delta B\{k\}$$  \hspace{1cm} (4.12)

Each hidden unit $j_m$ updates its bias and weights

$$W\{ji\} = W\{ji\}_{old} + \Delta w\{ji\}$$  \hspace{1cm} (4.13)

$$B\{j\} = B\{j\}_{old} + \Delta B\{j\}$$  \hspace{1cm} (4.14)

Step 10: test stopping condition

**B. Classification mode**

In the classification mode, the classes of data are recognized by simply passing through the network and making the straight computations.

For the process of training many other algorithms have been used in the literature. The algorithms used for training in this thesis project are Levenberg-Marquardt (LM), One Step Secant (OSS), Gradient Descent (GD), Resilient backpropagation (Rprop) and Conjugate gradient.
4.3.2 Resilient Backpropagation

Multilayer networks usually apply sigmoid transfer functions in the hidden layers. These functions are known as “squashing” functions, they squeeze an infinite input range into a finite output range. Sigmoid functions are described by the fact that their slopes must advance to zero with increase in input. This originates a problem when steepest descent is used for training with sigmoid functions, because the gradient can contain a very small magnitude, therefore it makes small changes in the weights and bias values, even if the weights and bias values are far away from the optimal values.

The reason behind using resilient backpropagation (Rprop) as training algorithm is to remove these harmful effects of the magnitudes of the partial derivatives. The sign of the derivative can only determine the direction for updating the weights, the magnitude of the derivative has no contribution. The range of the weight change is decided by an update value. Whenever the derivative of performance function with respect to weight has the same sign for two consecutive iterations, the update value weight and bias are increased by a factor $\mu^+$ and decreased by a factor $\mu^-$ when the derivative changes sign. The update value doesn’t change if the derivate is zero. The weight change is decreased when the weights are oscillating. If the weights are oscillating, the weight change is reduced. If the weight keeps on changing in the same direction for several iterations, the weight change increases its magnitude. The Rprop algorithm is completely described in [13].

Rprop is usually much faster than the standard steepest descent algorithm. It has an additional advantage that it requires only a modest increase in memory requirements. Unlike the storage requirement of gradient it only requires to store the update value of each weight and bias.

4.3.3 Conjugate Gradient Algorithms

The basic backpropagation algorithm updates the weights in the negative direction of the gradient (steepest descent direction), in which the performance function decreases more quickly. It results that, even if the function is decreasing more quickly along the negative of the gradient, the convergence is not necessarily faster. The conjugate gradient algorithms perform search along the conjugate direction, which results in faster convergence than the steepest descent directions. In this section only one type of conjugate gradient algorithms is presented. For details and applications of conjugate gradient algorithms see page 12-14 of [9].

In the above discussed training algorithms, a learning rate is used to find out the length of the step size (weight update). In conjugate gradient algorithms, the step size is adjusted during each step (iteration). They determine the step size by searching along the conjugate gradient direction that minimizes the performance function along that line. Search functions can be applied interchangeably with various training algorithms.
A. Fletcher-Reeves

Most of the conjugate gradient algorithms start their search in the steepest descent direction (negative of the gradient) on the first iteration.

$$p_0 = -g_0 \quad (4.15)$$

Then they start searching the line in order to find out the optimal distance and move along the present search direction.

$$w_{k+1} = w_k + \alpha_k p_k \quad (4.16)$$

Then they determine the next search direction such that it is conjugate to last search directions. The procedure to determine the new search direction includes the combination of new steepest descent direction and the previous one.

$$p_k = -g_k + \beta_k p_{k-1} \quad (4.17)$$

Some versions of the conjugate gradient algorithm are differentiated by the approach of computing the constant $$\beta_k$$. The procedure for the Fletcher-Reeves update is

$$\beta_k = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}} \quad (4.18)$$

This is the ratio of the norm squared of the current gradient to the norm squared of the previous gradient. See [86] or [9] for a detail discussion on Fletcher-Reeves conjugate gradient algorithm.

The conjugate gradient algorithms are faster than variable learning rate backpropagation, and they are sometimes faster than Rprop also, although the results differ from problem to problem. The conjugate gradient algorithms demand slightly more storage than the other algorithms. Therefore, these algorithms are suitable for networks having large number of weights (see [9]).

4.3.4 Quasi-Newton Algorithms

Newton's method is an alternative method of the conjugate gradient methods in order to achieve fast optimization. The basic concept of Newton's method is

$$w_{k+1} = w_k - A_k^{-1} p_k \quad (4.19)$$

Where $$A_k^{-1}$$ is the Hessian matrix (second derivatives) of the performance index at the present weights and bias values. The convergence Newton's method is often faster than conjugate gradient methods. The major drawback is that, it is expensive and complex to compute the
Hessian matrix for feedforward networks. There are some algorithms which are based on Newton’s method, but do not require calculation of second derivatives. These are called quasi-Newton or secant methods. These algorithms update an approximate Hessian matrix in each iteration. The update is found as a function of the gradient. The most successful quasi-Newton method in published studies is the Broyden, Fletcher, Goldfarb, and Shanno (BFGS) update.

A. BFGS Algorithm

The BFGS algorithm is explained in [4]. In this algorithm more computation and storage are required in each iteration than the conjugate gradient methods, although it normally converges in less iterations. The approximate Hessian need to be stored, whose dimension is \( n \times n \), where \( n \) is the total number of weights and biases of the network. For very large networks, Rprop and the conjugate gradient algorithms are better to use and for smaller networks, BFGS is an efficient training algorithm.

B. One Step Secant Algorithm

As the BFGS algorithm requires more computation and storage in each iteration than the conjugate gradient algorithms. To fulfill the requirements of less computations and smaller storage a secant approximation is required. The one step secant (OSS) method helps in bridging the gap between the quasi-Newton (secant) algorithms and the conjugate gradient algorithms. This algorithm does not need to store the whole Hessian matrix; it assumes that the previous Hessian was an identity matrix at each iteration. This has an additional benefit that the new search direction can be found without computing a matrix inverse.

The one step secant method is explained in [11]. This algorithm has smaller storage and computation requirements per each iteration or epoch than the BFGS algorithm, but it requires a little more computation and storage per epoch than the conjugate gradient algorithms. It can be regarded as a compromise between conjugate gradient algorithms and full quasi-Newton algorithms.

4.3.5 Levenberg-Marquardt

Similar to the quasi-Newton methods, the Levenberg-Marquardt algorithm was also planned to approach pace of second-order training without computing the Hessian matrix. When the performance function has the typical form of a sum of squares, then the Hessian matrix can be approximated as

\[
H = J^T J
\] (4.20)

and the gradient can be computed as
where the Jacobian matrix $\mathbf{J}$ contains first derivatives of the errors of the network with respect to the weights and biases, and $\mathbf{e}$ represent the vector of network errors. The Jacobian matrix can be obtained using a standard backpropagation technique (see [10]) that is much simpler than computing the Hessian matrix.

The Levenberg-Marquardt algorithm utilizes this approximation to the Hessian matrix in the following Newton-like update:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \left(\mathbf{J}^T \mathbf{J} + \mu \mathbf{I}\right)^{-1}\mathbf{J}^T\mathbf{e}$$  \hspace{1cm} (4.22)

When the scalar $\mu$ reaches zero, this is equivalent to Newton's method, by the approximate Hessian matrix. When $\mu$ has a large value, this turns into gradient descent having a small step size. Newton's method is quicker and more precise near an error minimum, so the target is to switch to Newton's method as rapidly as possible. Thus, $\mu$ is reduced after each successful step (decrease in performance function) and is increased when a step increases the performance function. In this manner, the performance function is constantly reduced in each iteration of the algorithm.

The detailed description of the Levenberg-Marquardt algorithm is provided in [9]. The application of Levenberg-Marquardt for neural networks is described in [10] and [9]. This algorithm emerged as the fastest method for training moderate-sized feedforward neural networks (up to several hundred weights see [9]).
Chapter 5

SIMULATIONS OVERVIEW

5.1 Introduction

During last one decade NN have been frequently used for digital equalization of wireless channels, digital receiver proposed in [1] states that NN based receiver provides better results in terms of symbol error rate for the Rayleigh fading channels. Channel estimation technique proposed in [2] claims that training algorithm NG (natural Gradient) can better compensate variations in the case of time varying multipath satellite channel as compared to BPA (Backpropagation Algorithm). In [3] radial basis functions (RBF) based blind channel estimation is performed and claims that phase problems have been equalized more efficiently.

This research proposes artificial neural networks based channel estimation and compensation technique carries out MIMO systems. The synaptic weights and bias values of the NN provide estimate of the channel. Different training (learning) algorithms have been implemented and analyzed for the calculation of those weights and bias values. The research also tries to determine the effectiveness of different algorithms by comparing their results, which can be used for the purpose of training/learning of NN. Moreover to establish the performance and flexibility of the proposed system; NN have been trained by varying the length of training sequences over suitable ranges and their results are observed. Finally results obtained for channel estimation by using several training algorithms of NN have been compared with each other, which along with comments and observations form part of this report.

5.2 System Model

Figure 5.1 NxN MIMO System with NN based channel estimator & compensator.
Figure 5.1 illustrates the block diagram of the proposed system. System includes NxN MIMO communication channel along with NN based channel estimator and compensator at the receiver end. The NN based estimator & compensator consists of N neural networks, each network works independently on the combined received signals to recover and detect signals transmitted from their respective transmitters (Tx1 to TxN). In this research project the simulations are performed for 2x2 and 3x3 MIMO channels.

5.3 Simulation Model 1

![2x2 MIMO Channel Diagram]

Figure 5.2 2x2 MIMO Channel

Figure 5.2 shows the first simulation model that has 2 x 2 MIMO channel (2 transmit and 2 receive antennas), the noise model is Additive White Gaussian Noise (AWGN). SNR in db (of AWGN) is varied over a reasonable range and the results have been observed. X1 and X2 are transmitted signals from transmit antennas Tx1 (user 1) and Tx2 (user 2) respectively where as Y1 and Y2 are received signals at receive antennas Rx1 and Rx2 respectively which are included by, magnitude changes, frequency shifts, independent communication path (channel) phase shifts, time delays, interfering copies of one another (i.e.\( x_1, x_2 \)) and the additive white Gaussian noise (AWGN) [8].

\[
x_1 = A_1 \cdot e^{j\theta_1} \quad (5.1)
\]

\[
x_2 = A_2 \cdot e^{j\theta_2} \quad (5.2)
\]

\[
Y = H \cdot X + AWGN \quad (5.3)
\]

\[
Y_1 = h_{1,1} \cdot x_1 + h_{1,2} \cdot x_2 + AWGN \quad (5.4)
\]

\[
Y_2 = h_{2,1} \cdot x_1 + h_{2,2} \cdot x_2 + AWGN \quad (5.5)
\]

\[
y_1 = (A_1 \cdot a_{11} \cdot e^{j(\theta_1 + \phi_{11})}) + (A_2 \cdot a_{12} \cdot e^{j(\theta_2 + \phi_{12})}) + AWGN \quad (5.6)
\]
\[ y_2 = (A_1 \cdot a_{21} e^{j(\theta_1 + \phi_{21})}) + (A_2 \cdot a_{22} e^{j(\theta_2 + \phi_{22})}) + AWGN \quad (5.7) \]

Where \( A_1 \) and \( A_2 \) represent magnitudes of interfering copies \( x_1 \) and \( x_2 \) respectively, \( H \) is the channel matrix, \( h_{1,1}, h_{1,2}, h_{2,1}, h_{2,2} \) are independent path channel attributes, \( \theta_1 \) and \( \theta_2 \) are phases of \( x_1 \) and \( x_2 \) respectively, \( a_{11}, a_{21}, a_{12}, a_{22} \) are respective magnitude changes by channel and \( \phi_{11}, \phi_{12}, \phi_{21}, \phi_{22} \) are respective phase changes by channel.

Transmitted signals \( x_1 \) & \( x_2 \) consists of two portions (W.R.T time), first portion is known as pilot sequence and 2\textsuperscript{nd} is called data sequence. The symbols/data present in pilot sequence are also known to receiver where as data sequence contains the data to be communicated. The pilot sequence is normally transmitted during transmission burst due to the time varying property of wireless channels. The length of pilot sequence and transmission burst also varied according to the type channel fading (i.e. slow or fast fading) but then the channel is kept as constant during the transmission time of one specific transmission burst [8]. Therefore in this research project the lengths of transmission burst and pilot sequence are also varied within a reasonable range to check the reliability of proposed design.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure5_3.png}
\caption{Block of ‘NN based channel effect estimator & compensator’ for 2x2 MIMO}
\end{figure}

The block of ‘NN based channel effect estimator & compensator’ includes two similar neural networks that operate at the received data and produce the results independently. NN1 is kept to recover signals \( x_1 \) transmitted from Tx1 and NN2 is reserved to estimate signals \( x_2 \) transmitted from Tx2 [8].
Figure 5.4 Functioning in training mode of neural networks

Figure 5.5 Functioning in operation mode of neural networks

Label ‘A’ represents the pilot sequence and label ‘E’ corresponds to data sequence of transmitted signal $x_1$ from Tx1, where label ‘B’ represents the pilot sequence and label ‘F’ corresponds to data sequence of transmitted signal $x_2$ from Tx2. Pilot sequences for received signals $Y_1$ & $Y_2$ are labeled as ‘C’ and ‘D’ correspondingly, where data sequences for $Y_1$ & $Y_2$ are labeled as ‘Y’ and ‘Z’ respectively.

For each transmission burst the receiver utilizes the samples of pilot sequence (known at receiver) and its corresponding samples of received signals to compute channel effects and impairments. The known pilot sequence transmitted during each transmission burst is unique for
each transmitter. C and D are the received pilot sequences against the transmitted pilot sequences A and B. The proposed design estimates the channel in terms of weights and bias values which are obtained after the process of training the neural networks. So the process of estimation has been replaced with the process of training. The sequences A, B, C and D have been utilized to train the neural networks (calculation of weights and bias values) by providing C & D as an input and A or B as respective target output (A for NN1 & B for NN2). The process of compensating the computed effects of channel from received data sequences is performed by directly passing them through the trained neural networks and getting the output of networks [8]. Figure 5.4 explains the process of training.

In order to eliminate channel impairments from received signals (Y1 & Y2), data sequences (Y & Z) are provided at the input of trained neural networks and passed through them. The output of NN1 provide the estimate of data sequence ‘\( \hat{E} \)’ transmitted from Tx1, and the output of NN2 provide the estimate of data sequence ‘\( \hat{F} \)’ transmitted from Tx2. Figure 5.5 shows the operation mode of NN1 and NN1.

### 5.4 Architecture of Neural Networks for 2x2 MIMO System

![Figure 5.6 Architecture of NN (Same for both receivers).](image)

Main benefit of neural networks is; they do not need a prior mathematical model. They use a learning algorithm to tune the synaptic weight and bias values by using concept of trial and error during the process of training. They don’t need sequential processing and calculations, the neurons of NN work continuously and simultaneously. Feed forward networks known as multi
layer perceptron (MLP) have been designed and analyzed for the purpose of estimation and compensation of channels effects. Feed forward networks contain some hidden layers between input and output layers and they do not engage any feedback connections. The NN based estimator and compensator for simulation model 1 consists of two feed forward (MLP) networks NN1 & NN2, the architecture is same for both. During the training mode both networks identifies channel effects and update their weights and bias values. During the operation mode NN1 is used to recovers x1 and NN2 is employed to recovers x2. The neural networks are designed such that they take four inputs at every instant of time (one for each input node). The four inputs are; Imaginary part of Y1, Real part of Y1, Imaginary part of Y2 and Real part of Y2. The input layer distributes inputs to four hidden nodes. Each input neuron is connected to each hidden neuron which results in 16 connections so the weight matrix of 1st layer is 4x4 matrix and include 16 elements. Similarly every hidden node is connected to two output nodes which results in 8 connections so the weight matrix of 2nd layer is 2x4 matrix and have 8 elements [8].

\[
W^1 = \begin{bmatrix}
  w_{1,1,1} & w_{1,1,2} & w_{1,1,3} & w_{1,1,4} \\
  w_{1,2,1} & w_{1,2,2} & w_{1,2,3} & w_{1,2,4} \\
  w_{1,3,1} & w_{1,3,2} & w_{1,3,3} & w_{1,3,4} \\
  w_{1,4,1} & w_{1,4,2} & w_{1,4,3} & w_{1,4,4}
\end{bmatrix}
\] (5.8)

Where \( w_{1,z,x} \) represents the weight value of connection between input I to node z in 1st layer. \( B^1 \) contains the bias values in 1st layer, \( b_{1,z} \) is the bias value of hidden node z in 1st layer.

\[
B^1 = \begin{bmatrix}
  b_{1,1} \\
  b_{1,2} \\
  b_{1,3} \\
  b_{1,4}
\end{bmatrix}
\] (5.9)

\[
W^2 = \begin{bmatrix}
  w_{2,1,1} & w_{2,1,2} & w_{2,1,3} & w_{2,1,4} \\
  w_{2,2,1} & w_{2,2,2} & w_{2,2,3} & w_{2,2,4}
\end{bmatrix}
\] (5.10)

Where \( w_{2,r,z} \) represents the weight value of connection between node z and node r of output in 2nd layer.

\[
B^2 = \begin{bmatrix}
  b_{2,1} \\
  b_{2,2}
\end{bmatrix}
\] (5.11)

Where \( B^2 \) contains the bias values in 2nd layer, \( b_{2,r} \) is the bias value of output node r in 2nd layer. The architecture of networks (same for NN1 & NN2) is shown in Figure 5.6.
A. Training mode

Training is the method through which network weights and bias values are updated. The proposed system calculates the estimate of channel in terms of neural network’s weights and bias values. Five different algorithms are implemented for the purpose of training and observations are made independently for each case. The algorithms implemented for training are LM, CG, Rprop, OSS and GD Backpropagation. Neural networks are given some input values and corresponding target values for the output, and then the training algorithms compute the weights and bias values of the particular network using those inputs and provided target values [12].

B. Operation mode

During the operation mode received data sequences are provided at the input of networks, passed through them and straight calculations are made to compute the estimate of transmitted data sequences. The output of 1st layer can be computed as

$$Z = purlin\{(W[1].I) + B[1]\} \quad (5.12)$$

Where I represents the input to the networks that is the received signal (received data sequences).

$$I = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix} \quad (5.13)$$

\[ I_1 <= \text{Real}(y_1) \quad I_2 <= \text{Imag}(y_1) \quad I_3 <= \text{Real}(y_2) \quad I_4 <= \text{Imag}(y_2) \]

Z is the output of 1st layer.

$$Z = \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \end{bmatrix} \quad (5.14)$$

$$R = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \quad (5.15)$$

$$R = purlin\{(W[2].Z) + B[2]\} \quad (5.16)$$


R is the output of 2nd layer (output layer).
\[ r_1 \Rightarrow Re(\hat{x}_1|\hat{x}_2) \quad (5.18) \]

\[ r_2 \Rightarrow Im(\hat{x}_1|\hat{x}_2) \quad (5.19) \]

\( r_1 \) represents the output obtained at first node of 2nd layer (estimate of the real part of transmitted signal) and \( r_2 \) represents the output obtained at second node of 2nd layer (estimate of the imaginary part of transmitted signal).

**Figure 5.7 Flow of simulations**

Figure 5.7 shows the flow of simulations. 4QAM and 16QAM signals are used for the simulations. The constellation diagrams of 4QAM and 16QAM signals are shown in Figure 5.8 & 5.9 respectively. Then for each signal type all the training algorithms have been implemented and the length of training sequence has been also varied for each case to analyze the performance of each algorithm i.e. training sequence = 16, 32 and 48.
Figure 5.8 4 QAM Complex Symbol Decision Space

Figure 5.9 16 QAM Complex Symbol Decision Space
5.5 Simulation Model 2

Figure 5.10 3x3 MIMO Channel

Figure 5.10 shows the second simulation model that has 3 x 3 MIMO channel (3 transmit and 3 receive antennas), in a similar way followed in simulation model one the noise model is Additive White Gaussian Noise (AWGN). SNR in db (of AWGN) is varied over a reasonable range and the results have been observed.

X1, X2 and X3 are transmitted signals from transmit antennas Tx1 (user 1), Tx2 (user 2) and Tx3 (user 3) respectively where as Y1, Y2 and Y3 are received signals at receive antennas Rx1, Rx2 and Rx2 respectively which are included by, magnitude changes, frequency shifts, independent communication path (channel) phase shifts, time delays, interfering copies of one another (i.e. x1, x2, x3) and the additive white Gaussian noise (AWGN).

\[
x_1 = A_1. e^{j\theta_1} \tag{5.18}
\]

\[
x_2 = A_2. e^{j\theta_2} \tag{5.19}
\]

\[
x_3 = A_3. e^{j\theta_3} \tag{5.20}
\]

\[
Y = H * X + AWGN \tag{5.21}
\]

\[
Y_1 = h_{1,1}.x_1 + h_{1,2}.x_2 + h_{1,3}.x_3 + AWGN \tag{5.22}
\]
\[ Y_2 = h_{2,1}x_1 + h_{2,2}x_2 + h_{2,3}x_3 + AWGN \]  
(5.23)

\[ Y_3 = h_{3,1}x_1 + h_{3,2}x_2 + h_{3,3}x_3 + AWGN \]  
(5.24)

\[
y_1 = (A_1.a_{11}.e^{j(\theta_1+\phi_{11})} + (A_2.a_{21}.e^{j(\theta_2+\phi_{21})} + (A_3.a_{31}.e^{j(\theta_3+\phi_{31})} + AWGN \]  
(5.25)

\[
y_2 = (A_1.a_{12}.e^{j(\theta_1+\phi_{12})} + (A_2.a_{22}.e^{j(\theta_2+\phi_{22})} + (A_3.a_{32}.e^{j(\theta_3+\phi_{32})} + AWGN \]  
(5.26)

\[
y_3 = (A_1.a_{13}.e^{j(\theta_1+\phi_{13})} + (A_2.a_{23}.e^{j(\theta_2+\phi_{23})} + (A_3.a_{33}.e^{j(\theta_3+\phi_{33})} + AWGN \]  
(5.27)

Where \(A_1, A_3\) and \(A_2\) represent magnitudes of interfering copies \(x_1, x_2\) and \(x_3\) respectively, \(H\) is the channel matrix, \(h_{1,1}, h_{1,2}, h_{1,3}, h_{2,1}, h_{2,2}, h_{2,3}, h_{3,1}, h_{3,2}, h_{3,3}\) are independent path channel attributes, \(\theta_1, \theta_3\) and \(\theta_2\) are phases of \(x_1, x_3\) and \(x_2\) respectively, \(a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}, a_{31}, a_{32}, a_{33}\) are respective magnitude changes by channel and \(\phi_{11}, \phi_{12}, \phi_{13}, \phi_{21}, \phi_{22}, \phi_{23}, \phi_{31}, \phi_{32}, \phi_{33}\) are respective phase changes by channel.

Similarly as in case 1 transmitted signals \(x_1, x_2 \& x_3\) consists of two portions (W.R.T time), first portion is known as pilot sequence and second is called data sequence. The symbols/data present in pilot sequence are also known to receiver whereas as data sequence contains the data to be communicated. Therefore in this case the lengths of transmission burst and pilot sequence are also varied within a reasonable range to check the reliability of 2nd case of proposed design.

Figure 5.11 Block of ‘NN based channel effect estimator & compensator’ for 3x3 MIMO Channel
In the same way followed in model 1 the block of ‘NN based channel effect estimator & compensator’ contains three neural networks that operate at the received data and produce the independent results transmitted independently from different antennas. NN1 is set to recover signals \((x1)\) transmitted from Tx1, NN2 is reserved to compute \((x2)\) transmitted from Tx2 and NN3 is dedicated to estimate \((x3)\) transmitted from Tx3.

### 5.6 Architecture of Neural Networks for 3x3 MIMO System

![Architecture of NN for 3x3 MIMO channel](same for all receivers).

Figure 5.12 shows the architecture of single NN. The architecture is the same for all receivers as it in simulation model 1, working independently at each side to recover their respective signals. The process of training and operation is exactly the same as executed in model 1.

NN for the 3x3 MIMO channel has 6 input nodes in the 1st layer which are connected to 6 hidden nodes resulting in total 36 connections so the weight matrix of 1st layer is 6x6 and includes 36 elements. In Output layer (2nd layer) the 6 hidden nodes are connected to two output.
nodes resulting in 12 connections and weight matrix of 2x6 containing 12 elements.

\[
W[1] = \begin{bmatrix}
    w[1]_{1,1} & w[1]_{1,2} & w[1]_{1,3} & w[1]_{1,4} & w[1]_{1,5} & w[1]_{1,6} \\
    w[1]_{2,1} & w[1]_{2,2} & w[1]_{2,3} & w[1]_{2,4} & w[1]_{2,5} & w[1]_{2,6} \\
    w[1]_{3,1} & w[1]_{3,2} & w[1]_{3,3} & w[1]_{3,4} & w[1]_{3,5} & w[1]_{3,6} \\
    w[1]_{4,1} & w[1]_{4,2} & w[1]_{4,3} & w[1]_{4,4} & w[1]_{4,5} & w[1]_{4,6} \\
    w[1]_{5,1} & w[1]_{5,2} & w[1]_{5,3} & w[1]_{5,4} & w[1]_{5,5} & w[1]_{5,6} \\
    w[1]_{6,1} & w[1]_{6,2} & w[1]_{6,3} & w[1]_{6,4} & w[1]_{6,5} & w[1]_{6,6}
\end{bmatrix}
\]  \hspace{1cm} (5.28)

Where \( w[1]_{z,x} \) represents the weight value of connection between input I to node z in 1st layer. \( B[1] \) contains the bias values in 1st layer, \( b[1]_z \) is the bias value of hidden node z in 1st layer.

\[
B[1] = \begin{bmatrix}
    b[1]_1 \\
    b[1]_2 \\
    b[1]_3 \\
    b[1]_4 \\
    b[1]_5 \\
    b[1]_6 
\end{bmatrix}
\]  \hspace{1cm} (5.29)

\[
W[2] = \begin{bmatrix}
    w[2]_{1,1} & w[2]_{1,2} & w[2]_{1,3} & w[2]_{1,4} & w[2]_{1,5} & w[2]_{1,6} \\
    w[2]_{2,1} & w[2]_{2,2} & w[2]_{2,3} & w[2]_{2,4} & w[2]_{2,5} & w[2]_{2,6} 
\end{bmatrix}
\]  \hspace{1cm} (5.30)

Where \( w[2]_{r,z} \) represents the weight value of connection between node z and node r of output in 2nd layer.

\[
B[2] = \begin{bmatrix}
    b[2]_1 \\
    b[2]_2 
\end{bmatrix}
\]  \hspace{1cm} (5.31)

Where \( B[2] \) contains the bias values in 2nd layer, \( b[2]_r \) is the bias value of output node r in 2nd layer. The architecture of networks (same for NN1 & NN2) is shown in Figure 5.18.

The output of 1st layer can be computed as

\[
Z = purlin\{(W[1].I) + B[1]\}
\]  \hspace{1cm} (5.32)

Where I represents the input to the networks that is the received signal (received data sequences).

\[
I = \begin{bmatrix}
    I_1 \\
    I_2 \\
    I_3 \\
    I_4 \\
    I_5 \\
    I_6
\end{bmatrix}
\]  \hspace{1cm} (5.33)
\[ I_1 \leq R(y_1), \ I_2 \leq T(y_1), \ I_3 \leq R(y_2), \ I_4 \leq T(y_2), \ I_5 \leq R(y_3), \ I_6 \leq T(y_3) \]

\( Z \) is the output of 1st layer.

\[
Z = \begin{bmatrix}
    z_1 \\
    z_2 \\
    z_3 \\
    z_4 \\
    z_5 \\
    z_6
\end{bmatrix}
\]

(5.34)

\[
R = \begin{bmatrix}
    r_1 \\
    r_2
\end{bmatrix}
\]

(5.35)

\[
R = purlin\{(W[2], Z) + B[2]\}
\]

(5.36)

\[
R = purlin\left\{(W[2], (purlin\{(W[1], I) + B[1]\}) + B[2]\right\}
\]

(5.37)

\( R \) is the output of 2nd layer (output layer).

\[
r_1 \Rightarrow Re(\hat{x}_1|\hat{x}_2|\hat{x}_3)
\]

(5.38)

\[
r_2 \Rightarrow Im(\hat{x}_1|\hat{x}_2|\hat{x}_3)
\]

(5.39)

\( r_1 \) represents estimate of the real part of transmitted signal, \( r_2 \) represents the estimate of the imaginary part of transmitted signal obtained at their respective output nodes.
Chapter 6

SIMULATION RESULTS

6.1 Introduction

The two simulation models described in Chapter 5 have been implemented in Matlab. The simulation results for both models are given below:

6.2 Results of Simulation Model 1

![Figure 6.1 SER v/s SNR (dB) plot for 2x2 MIMO (4QAM, Training length=16)](image)

Figure 6.1 SER v/s SNR (dB) plot for 2x2 MIMO (4QAM, Training length=16)
Figure 6.2 SER v/s SNR (dB) plot for 2x2 MIMO (4QAM, Training length=32)

Figure 6.3 SER v/s SNR (dB) plot for 2x2 MIMO (4QAM, Training length=48)
Figure 6.4 SER v/s SNR (dB) plot for 2x2 MIMO (16QAM, Training length=16)

Figure 6.5 SER v/s SNR (dB) plot for 2x2 MIMO (16QAM, Training length=32)
Figure 6.6 SER v/s SNR (dB) plot for 2x2 MIMO (16QAM, Training length=48)

The length of training sequence has been varied, SER is calculated for different values of SNR and then independent observations have been made. Curves obtained using One Step Secant (OSS), Levenberg-Marquardt (LM), Gradient Descent (GD), Resilient (Rprop) and Conjugate Gradient (CG) back propagation training algorithms for training sequence length of 16, 32 & 48 are shown in Figures 6.1, 2, 3, 4, 5 & 6.

To demonstrate the performance of proposed NN based designs results are compared with each other and with the design trained with traditional backpropagation algorithm. The length of training sequences for each case are varied over a reasonable range and observations are made. The curves are compared in terms of SER with respect to SNR for training symbols lengths of 16, 32 & 48 respectively. The figures describe the SER performance with existence of AWGN. It is experienced from simulations that the performance of LM and CG is better than GD, Rprop and OSS. The proposed NN based system trained using LM & CG can better track the channel for training symbols length greater/equal than 16 and NN trained using OSS & Rprop gives better results for training symbols of length equal to 48. It has been verified that wireless channels parameters can be estimated in terms of neural network’s synaptic weights and bias values. Further it has been established that on the cost of data rate SER can be reduced and by increasing length of training sequence the performance and reliability of proposed system get better against that of traditional backpropagation algorithm.
6.3 Results of Simulation Model 2

Figure 6.7 SER v/s SNR (dB) plot for 3x3 MIMO (4QAM, Training length=16)

Figure 6.8 SER v/s SNR (dB) plot for 3x3 MIMO (4QAM, Training length=32)
Figure 6.9 SER v/s SNR (dB) plot for 3x3 MIMO (4QAM, Training length=48)

Figure 6.10 SER v/s SNR (dB) plot for 3x3 MIMO (16QAM, Training length=16)
The observations are very much similar to the case of 2x2 MIMO channel. The SER is calculated for various values of SNR and results are observed by varying the length of training symbols.
The curves are shown in figures 6.7, 8, 9, 10, 11 & 12 for training symbols of length equal to 16, 32 and 48. Simulations are performed by using One Step Secant (OSS) & Levenberg-Marquardt (LM), Gradient Descent (GD), Resilient (Rprop) and Conjugate Gradient (CG) back propagation training algorithms and results obtained are compared with each other to illustrate the performance of proposed design. The comparison curves are drawn in terms of SER with respect to various values of SNR. The figures indicate the SER performance of each training algorithm in the presence AWGN. The simulation results shows that LM performs better amongst all for the case of 4QAM and training length of 16 & 32 symbols. The training algorithms CG, Rprop & OSS performs better for the case of 4QAM when training length of symbols is equal to 48. For the case of 16QAM, LM again achieves better results amongst all when the length of training symbols equal to 16 and CG, Rprop & OSS provide better results for training symbols of length 32 &48. The SER performance of GD is worst in all the cases. It is established that channel parameters can be identified by using NN based channel estimator and channel can be estimated in terms of NN synaptic weights and bias values. It is clear from results that SER can be reduced on the cost of data rate. The performance and reliability of proposed design are improved as compared to the system trained with basic backpropagaion algorithm (GD).
Chapter 7

CONCLUSION AND FUTURE WORK

7.1 Summary and Conclusions

This thesis presents a channel equalization technique based on neural networks. The technique provides channel estimation and compensation for MIMO communication systems. The technique implemented over various MIMO communication models and the results have been examined. The neural networks are employed as channel equalizer using four different training algorithms (i.e. OSS, LM, CG & RProp). Through experimentation it has been ascertained that LM can be more effective for training the neural networks and tracking the channel. The thesis includes simulations and results for all of the abovementioned algorithms. The performance is measured in term of SER for various values of SNR. The curves drawn through simulations substantiate abovementioned fact that LM performs better as compared to other algorithms.

The thesis also includes a comparison of proposed system with basic backpropagation algorithm (GD). The length of training sequences has been varied over a reasonable range to examine the results of proposed design. It has been established that SER can be reduced and OSS, LM, CG & RProp based design provides better performance in terms of SER as compared to basic backpropagation algorithm. The LM gives better results when length of training sequence is equal to 16 symbols while OSS, CG, Rprop provide better performance when length of training sequence is equal to 48 symbols. The simulation results conclude that the consistency and performance of proposed system improves with reasonable increase in length of training symbols against that of basic backpropagation algorithm (GD).

7.2 Future Work

The length of training sequences in pilot based channel equalization is a big constraint. The increase in length of training sequence reduces the speed of communication, so less the length of training symbols more will be the data rate. The extension of current work is to minimize the length of training sequence by keeping the same performance of the system. The proposed system also demands high computation power that is again a constraint over the speed of communication systems. This demand can be reduced by pre-calculation of synaptic weights & bias values for various communication environments and also by introducing unsupervised leaning of neural networks.
REFERENCES