PERFORMANCE COMPARISON OF ADAPTIVE DECISION FEEDBACK EQUALIZER AND BLIND DECISION FEEDBACK EQUALIZER

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

SİNAN ŞENOL

IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

JANUARY 2004

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Mübeccel Demirekler
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Buyurman Baykal
Supervisor

Examining Committee Members

Assoc. Prof Dr. Engin Tuncer

Assoc. Prof. Dr. Buyurman Baykal

Assist. Prof. Dr. Tolga Çiloğlu

Assist. Prof. Dr. Arzu Tuncay Koç

Hacer Sunay

ABSTRACT

PERFORMANCE COMPARISON OF ADAPTIVE DECISION FEEDBACK EQUALIZER AND BLIND DECISION FEEDBACK EQUALIZER

ŞENOL, Sinan

MSc., Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. Buyurman BAYKAL

January 2004, 87 Pages

The Decision Feedback Equalizer (DFE) is a known method of channel equalization which has performance superiority over linear equalizer. The best performance of DFE is obtained, commonly, with training period which is used for initial acquisiton of channel or recovering changes in the channel. The training period requires a training sequence which reduces the bit transmission rate or is not possible to send in most of the situations. So, it is desirable to skip the training period. The Unsupervised (Blind) DFE (UDFE) is such a DFE scheme which has no training period. The UDFE has two modes of operation. In one mode, the UDFE uses Constant Modulus Algorithm (CMA) to perform channel acquisition, blindly. The other mode is the same as classical decision-directed DFE. This thesis compares the performances of the classical trained DFE method and the UDFE.

The performance comparison is done in some channel environments with the problem of timing error present in the received data bearing signal. The computer aided simulations are done for two stationary channels, a time-varying channel and a frequency selective Rayleigh fading channel to test the performance of the relevant equalizers. The test results are evaluted according to mean square error (MSE), bit-error rate (BER), residual intersymbol interference (RISI) performances and equalizer output diagrams. The test results show that the UDFE has an equal or, sometimes, better performance compared to the trained DFE methods. The two modes of UDFE enable it to solve the absence of training sequence.

Keywords: adaptive equalization, blind equalization, decision feedback equalization, training sequence

ÖZ

UYARLANIR KARAR GERİ BESLEMELİ DENKLEŞTİRİCİ VE GÖZÜ KAPALI UYARLANIR KARAR GERİ BESLEMELİ DENKLEŞTİRİCİNİN PERFORMANS KARŞILAŞTIRMASI

ŞENOL, Sinan
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Buyurman BAYKAL

Ocak 2004, 87 Sayfa

Karar Geri Beslemeli Denkleştirici (KGBD), doğrusal denkleştiriciye performans üstünlüğü olan, bilinen bir kanal denkleştirme metodudur. KGBD'nin en iyi performansı, genellikle, kanalın ilk elde edilmesi ya da kanalda değişiklikleri tespit etme amaçlı olan eğitme süreci ile elde edilir. Eğitme süreci, bit iletim miktarını düşüren ya da çoğu zaman gönderilmesi mümkün olmayan eğitici diziye gereksinim duyar. Bu yüzden, eğitme sürecini atlamak arzu uyandırıcıdır. Gözü Kapalı Uyarlanır KGBD (GKKGBD), eğitme süreci olmayan, bu tip bir KGBD metoddur. GKKGBD'nin iki çalışma modu vardır. Bir modunda, GKKGBD, gözü kapalı bir şekilde kanalı elde etmek için sabit genlikli algoritma (CMA) kullanır. Diğer modu ise klasik karar yöneltmeli KGBD ile aynıdır. Bu tez, klasik eğitilen KGBD metodu ile GKKGBD'nin performanslarını karşılaştırmaktadır. Performans karşılaştırması, bazı kanal ortamlarında, veri taşıyan sinyalde mevcut bulunan zamanlama hatası problemi ile yapılmaktadır.Bilgisayar destekli simülasyonlar, ilgili denkleştiricilerin performansını test etmek için iki durağan kanal, bir zamanla değişen kanal ve frekans seçmeli Rayleigh solan kanalda yapılmıştır. Test sonuçları, ortalama karesel hata (MSE), bit hata oranı (BER), artık semboller arası girişim (RISI) performansları ve denkleştirici çıktı diyagramlarına göre değerlendirilmiştir. Test sonuçları, eğitilen KGBD metodlarına göre karşılaştırıldığında, GKKGBD'nin eşit, veya, bazen, daha iyi performansa sahip olduğunu göstermektedir. GKKGBD'nin iki modu, ona eğitici dizi yokluğu problemini çözmesini sağlamaktadır.

Anahtar Kelimeler: uyarlanır denkleştirme, gözü kapalı denkleştirme, karar geri beslemeli denkleştirme, eğitici dizi

To My Family

ACKNOWLEDGEMENTS

I would like to thank Assoc. Prof. Dr. Buyurman Baykal for his helpful supervision and enlightening support throughout the development and improvement of this thesis. His support has valuable significance to reach the end of thesis.

I thank ASELSAN Inc. for the continuous support for the completion of this thesis.

I give thanks to my colleagues in Crypto Systems Group for their patience and encouragements that continued for all the period of the completion of this thesis.

I am grateful to my family for their all time lasting support.

TABLE OF CONTENTS

ABSTRACT III
ÖZ V
ACKNOWLEDGEMENTS VIII
TABLE OF CONTENTS IX
LIST OF TABLES XI
LIST OF FIGURES XII
LIST OF ABBREVIATIONS XV
CHAPTER
1. INTRODUCTION 1
 1.1. Overview of Digital Communication 1
 1.2. Overview of Equalizers 6
 1.3. Outline of The Thesis 9
2. TRAINED APPROACH 11
 2.1. Timing Recovery 12
  2.1.1. Gardner's Synchronizer 14
  2.1.2. Non-Data-Aided Maximum Likelihood Based Timing Recovery 16
  2.1.3. Interpolation 19
 2.2. Frame Synchronization 20
 2.3. Equalizers 21
  2.3.1. Criteria 23
   2.3.1.1. Zero-Forcing Criterion 23
   2.3.1.2. MSE Criterion 24
  2.3.2. Linear Equalizer 25
  2.3.3. Decision Feedback Equalizer 26
   2.3.3.1. Zero Forcing DFE 27

2.3.3.2. MSE DFE 30
2.4. Adaptive Equalization 31
2.4.1. LMS Algorithm 34
2.4.2. Adaptive Decision Feedback Equalizer 35
3. BLIND EQUALIZATION 39
3.1. Constant Modulus Algorithm 40
3.2. Unsupervised Decision Feedback Equalizer 42
3.2.1. Starting Mode 44
3.2.2. Tracking Mode 47
4. SIMULATIONS 50
4.1. Timing Recovery Results 54
4.2. Equalizer Simulations 59
4.2.1. MSE Results 60
4.2.2. BER Results 66
4.2.3. RISI Results 76
5. CONCLUSIONS 82
REFERENCES 85

LIST OF TABLES

TABLE

4.1.1. MSET Values for the Channel Cases 1 & 4 58

LIST OF FIGURES

FIGURES

1.1.1. The Basic Digital Communication Model 2

1.1.2. Digital Communication Model.. 4

2.1.1. General Digital Timing Recovery System for an Ideal Channel 13

2.1.1.1. The Block Diagram of the Gardner’s Synchronizer................................. 14

2.1.2.1. The Block Diagram of NDA-MLBA 17

2.3.2.1. The Block Diagram of the Receiver with Linear Equalizer...................... 26

2.3.3.1. The Block Diagram of Decision Feedback Equalizer............................... 27

2.3.3.1.1. The ZF DFE Block Diagram.. 28

2.4.1. The General Block Diagram of the Adaptive Equalizer 32

2.4.1.1. The Block Diagram of LMS Algorithm.. 34

2.4.2.1. The Block Diagram of the Adaptive DFE.. 36

3.2.1.1. The Block Diagram of the Starting Mode.. 45

3.2.2.1. The Block Diagram of the Tracking Mode (Classical DFE Mode) 48

4.1. The General Model of the Simulation System 50

4.2. The Zeros of Channels. a) Case 1. b) Case 2. 52

4.1.1. The Timing Recovery Algorithm Output for Channel Case 1 (SNR=16 dB).
a) Gardner’s Algorithm. b) Oerder& Meyr Algorithm. 56

4.1.2. The Timing Recovery Algorithm Output for Channel Case 1 (SNR=8 dB).
a) Gardner’s Algorithm. b) Oerder& Meyr Algorithm. 57

4.1.3. The Timing Recovery Algorithm Output for Channel Case 4 (SNR=16 dB,
Velocity=30 km/h). a) Gardner’s Algorithm. b) Oerder& Meyr Algorithm.. 57

4.1.4. The Timing Recovery Algorithm Output for Channel Case 4 (SNR=8 dB,
Velocity=30 km/h). a) Gardner’s Algorithm. b) Oerder& Meyr Algorithm.. 58

4.2.1.1. The MSE Diagram for UDFE for Channel Case 1 (SNR=20 dB). ... 61
4.2.1.2. The MSE Diagram for Trained DFE for Channel Case 1 (SNR=20 dB). ... 62
4.2.1.3. The MSE Diagram for DD DFE for Channel Case 1 (SNR=20 dB). 62
4.2.1.4. The MSE Diagram for TRDD DFE for Channel Case 1 (SNR=20 dB). ... 63
4.2.1.5. The MSE Diagram for UDFE for Channel Case 1 (SNR=10 dB). ... 64
4.2.1.6. The MSE Diagram for Trained DFE for Channel Case 1 (SNR=10 dB). ... 64
4.2.1.7. The MSE Diagram for DD DFE for Channel Case 1 (SNR=10 dB). 65
4.2.1.8. The MSE Diagram for TRDD DFE for Channel Case 1 (SNR=10 dB). ... 65
4.2.2.1. The BER Diagram for Channel Case 1. (x : UDFE, o : Trained DFE, + : DD DFE, *: TRDD DFE) ... 67
4.2.2.2. The BER Diagram for Channel Case 2. (x : UDFE, o : Trained DFE, + : DD DFE, *: TRDD DFE) ... 68
4.2.2.3. The BER Diagram for Channel Case 3. (x : UDFE, o : Trained DFE, *: TRDD DFE) ... 69
4.2.2.4. The BER Diagram for Channel Case 4 (Velocity=10 km/h). (x : UDFE, o : Trained DFE, + : DD DFE, *: TRDD DFE) ... 69
4.2.2.5. The BER Diagram for Channel Case 4 (Velocity=30 km/h). (x : UDFE, o : Trained DFE, + : DD DFE, *: TRDD DFE) ... 70
4.2.2.6. The Equalizer Output for Channel Case 4 (SNR=14 dB, Velocity=30 km/h). a) UDFE. b) Trained DFE. c) DD DFE. d) TRDD DFE. ... 73
4.2.2.7. The Equalizer Output for Channel Case 4 (SNR=14 dB, Velocity=90 km/h). a) UDFE. b) Trained DFE. c) DD DFE. d) TRDD DFE. ... 74
4.2.2.8. The Equalizer Output for Channel Case 4 (SNR=∞ dB, Velocity=30 km/h). a) UDFE. b) Trained DFE. c) DD DFE. d) TRDD DFE. ... 75
4.2.3.1. The Equivalent Channels for Channel Case 1 (SNR=18 dB). a) DD DFE. b) Trained DFE. c) UDFE. d) TRDD DFE ... 78

4.2.3.2. The RISI Diagram for Channel Case 1. (x : UDFE, o : Trained DFE, + : DD DFE, *: TRDD DFE).. 79
4.2.3.3. The RISI Diagram for Channel Case 2. (x : UDFE, o : Trained DFE, + : DD DFE, *: TRDD DFE).. 79

LIST OF ABBREVIATIONS

A/D	Analog to Digital
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
CMA	Constant Modulus Algorithm
DD	Decision Directed
DCE	Data Circuit-Terminating Equipment
DFE	Decision Feedback Equalizer
DTE	Data Terminal Equipment
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
ISI	Intersymbol Interference
LMS	Least Mean Square
MLSE	Maximum Likelihood Sequence Estimation
MMSE	Minimum Mean Square Error
MSE	Mean Square Error
MSET	Mean Square Error Timing
PSD	Power Spectral Density
RISI	Residual InterSymbol Interference

RLS	Recursive Least Squares
SNR	Signal to Noise Ratio
UDFE	Unsupervised (Blind) Decision Feedback Equalizer
ZF	Zero Forcing

CHAPTER 1

INTRODUCTION

1.1. Overview of Digital Communication

As time passes, the digital communication is becoming more dominant over analog communication. The main factors for this situation are the increasing performance and decreasing cost of digital communication equipment.

Communication involves transfer of some information like voice, image or data from a source to a destination with almost in its original form. Digital communication can be modeled as a system that is given in Figure 1.1.1. [1]. Digital communication performs the job of communication by, firstly, converting the information to a form which is composed of a sequence of binary digits. Then, the sequence is encoded to appropriate symbols and modulated digitally for transfering the information into suitable signal waveforms that can be carried in the channels dedicated for communication. The signal passes the channel and arrives at the receiver. In the receiver of the digital communication signal performs the inverse operation. The received signal is demodulated and the obtained symbols are converted into a binary sequence. The binary sequence is used to form the original information by a source decoder.

The information coded to a binary sequence passes from the phase of modulation with a waveform in the digital modulator [1]. If every bit is modulated with one of two waveforms, then it is called binary modulation. Also, the n-bit sequence coded can be transmitted with one of the 2^n waveforms that are possible for n-bit sequences. If we call M the total number of waveforms which are used to

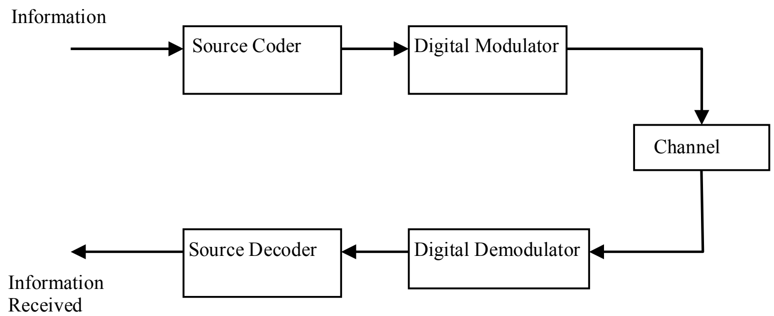


Figure 1.1.1. The Basic Digital Communication Model

send the distinct n-bit sequences, then the digital modulation is called M-ary. The method for M-ary communication is to encode the n-bit sequence into M symbols and then multiply them with a pulse that has a special shape. Whether the modulation is binary or M-ary, the resulting transmitted signal has the waveform

$$y(t) = \sum_{n=0}^{\infty} I_n p(t - nT) \tag{1.1}$$

where I_n is one of the M symbols, p(t) is the pulse shape and T is the sampling interval. The shape of the pulse must obey Nyquist First Criterion for zero Intersymbol Interference (ISI) in order to recover the symbols correctly without interference from the adjacent symbols. Intersymbol Interference (ISI) is the contribution of the adjacent symbol values to the center symbol value because of the nonzero values of the pulse at the adjacent symbol values.The Nyquist First Criterion for zero ISI requires that

$$p(kT) = p_k = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases} \tag{1.2}$$

If this condition is met, the adjacent symbol values (since p_k=0 for k≠0) do not contribute to the center symbol value. This brings the fact that the symbol value is sent to the channel without any error.

The popular pulse shape that satisfies this condition is the raised-cosine pulse [1]. The raised-cosine frequency response has the desired properties and it is given below,

$$P(f) = \begin{cases} T & 0 \le |f| \le \dfrac{1-\beta}{2T} \\ \dfrac{T}{2}\left\{1+\cos\left[\dfrac{\pi T}{\beta}\left(|f| - \dfrac{1-\beta}{2T}\right)\right]\right\} & \dfrac{1-\beta}{2T} \le |f| \le \dfrac{1+\beta}{2T} \\ 0 & |f| > \dfrac{1+\beta}{2T} \end{cases} \tag{1.3}$$

where β is the roll-off factor and its values are between 0 and 1, inclusive.

The impulse response of the raised-cosine pulse is given below,

$$p(t) = \frac{\sin(\pi t/T)}{\pi t/T} \frac{\cos(\pi\beta t/T)}{1-4\beta^2 t^2/T^2} \tag{1.4}$$

In the raised-cosine pulse, the first term, which is equivalent to sinc pulse, ensures the zero crossings required for the Nyquist Criterion.

In general, this pulse is the overall pulse that is obtained in the receiver. In other words, this pulse is the result of the convolution of the transmit filter impulse response and receive filter impulse response. In order to obtain this result, both the transmit pulse and the receive filter must have the impulse response of root-raised cosine pulse [9], [21]. Root-raised cosine pulse is given below,

$$p_{root}(t) = \frac{4\beta}{\pi\sqrt{T}} \frac{\cos\left((1+\beta)\pi t/T\right) + T\sin\left((1-\beta)\pi t/T\right)/(4\beta t)}{1-(4\beta t/T)^2} \tag{1.5}$$

After the bit sequence to be transmitted is modulated, it is given to the channel [1]. Channel is the medium that carries the waveform between the transmitter and receiver. There can be different types of channels. Some channel examples are copper wires, fiber optic channels or air. All of the channel types have different characteristics. They have different channel bandwidth, different phase distortion, signal attenuation, or multipath etc. characteristics. These characteristics distort the transmitted signal at some measure. Also, the channel has another distortion which is additive noise. Additive noise happens in the transmission medium internally and distorts the signal to be transmitted. With these characteristics, the signal that will be transmitted from the source to the destination can be modeled as in Figure 1.1.2.

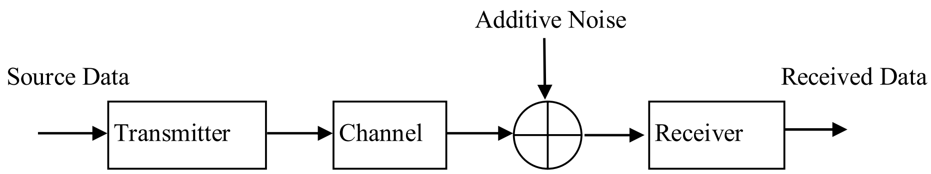


Figure 1.1.2. Digital Communication Model

The signal that is given to the channel may use different frequency bands in order to arrive its destination. But, the analysis of this transmission mechanism is equivalent to its analysis in complex baseband representations. So, throughout the thesis, the analysis of the signals will be carried on the complex baseband representations of the signals.

All channels can be modeled as a filter that has an impulse response which has finite length or infinite length. Channels like air generally exhibit a channel impulse response which is finite. Also, the channel impulse response may be time-invariant or time-varying. The signal that is sent from the transmitter, passed the channel and arrived at the receiver can be formulated as below [1],

$$r_c(t) = c(t; t_1) * y(t) + n(t) = s(t) + n(t) \tag{1.6}$$

where * stands for convolution and $c(t;t_1)$ is the channel that varies with time, t, according to the impulse response varies with time, t_1. n(t) is the complex baseband noise and y(t) is the transmitter output.

As in the upper model, the channel is modeled as additive noise channel. This noise distorts the signal $c(t;t_1)*y(t)$ additively. There are some sources for occurence of noise. These can be internal electronic components which cause thermal noise. Another source is the additive interference in the channel. A suitable model for the characteristics of the noise is Gaussian process. Since this modeling is suitable for analysis, the channel which has this type of noise is called

Additive Gaussian Noise Channel. Also, another property for the channel noise is that power spectrum of the noise is white. With this addition, the noise is called Additive White Gaussian Noise (AWGN). This model has a wide applicability in communications area, so it is the widely accepted model.

When the AWGN channel output signal has arrived at the receiver, the receiver's mission is to recover the transmitted symbols as close as possible to their initial form in the receiver. To do this job, receiver tries to remove the redundancies like carrier frequency modulation, channel distortion, etc. But while doing this job, some distortions inherent in the digital communication behavior occur. One of them is carrier offset which occurs because of carrier frequency mismatches between the transmitter and receiver. The other is the symbol timing offset which occurs because of the wrong sampling of the received signal between a T-second period. When these distortions are presented in the baseband equivalent representations of the received signal, the carrier offset is represented as phase error and symbol timing offset is represented as timing error. The received signal which is the output of the AWGN channel has the following form with phase error and timing error,

$$r_c(t) = s(t;\tau)e^{j\varphi} + n(t) \tag{1.7}$$

where τ is the timing error and ϕ is the phase error. While correcting these distortions, symbol timing can be corrected without knowledge of the carrier phase. Because of this, the timing error is first corrected and after that carrier phase can be estimated.

In, generally, all digital communication receivers, a common receiver behaviour arised from distortions occurs. This behaviour is composed of three parts:

1. **Timing Recovery:** Timing recovery is done to correct timing error.
2. **Phase Recovery:** Phase Recovery is done to correct the carrier phase.
3. **Channel Equalization:** Channel may have significant distortion on the signal. So removing the channel's distortion has a vital importantance.

There is a big literature about algorithms for timing and phase recovery. These employ minimum mean square error (MMSE) or other criteria. In order to reach a satisfactory result, some timing recovery algorithms employ the characteristics of the pulse shape like Gardner's Synchronizer [10] or Mueller and Müller Timing Recovery [11]. Some algorithms are based on the maximum likelihood criterion. In this study, timing recovery algorithms are used to solve the timing errors introduced into the signals produced for simulations. Phase recovery algorithms are not treated in this thesis.

Since the ideal aim of the receiver is to recover the original symbols without error, the distortions caused by the channel, noise and other sources are tried to be minimized. In this thesis, the performance of channel equalizers in the case of AWGN and timing error is considered.

1.2. Overview of Equalizers

The equalizer in its basic form is the filter or generally, a system of filters, that aims to remove the undesirable effects of the transmission system including channel from the signal bearing data that are to be transmitted to the destination [1], [2]. In digital communications system, the frequently faced problem is the Intersymbol Interference (ISI). ISI occurs because of the channel which has an amplitude and phase dispersion. This dispersion causes the signal to interfere with another parts of the signal. This effect causes to ISI. The pulses to carry the data are designed to minimize the ISI effect. The Nyquist criterion that is required for the pulse shape is given below as told before,

$$p(kT) = p_k = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases} \tag{1.8}$$

where p(t) is the pulse shaping function. But the effect of channel distorts this. So, in the receiver, this problem is solved with the design of equalizers. The equalizer generally models the effect of inverse operation of the transmission system. But, while doing this, an undesirable result may occur. This result happens at the points where the equalizer amplifies the signal to remove ISI. This amplification causes the amplification of the noise as well. So, equalizer design and structure gain importance in order to remove ISI while minimizing the noise.

The equalizer can be modeled as a system which has a transfer function. This transfer function will invert the bad effect of transmission system which introduces ISI and noise. Also, some equalizers correct the timing and phase errors to some extent. The simplest equalizer is the linear equalizer which is, generally, implemented with a finite impulse response (FIR) filter. The reason for this filter is its low-complexity and cheap production. But, since its performance is not enough for higher expectations, generally, the more sophisticated equalizer schemes were searched. These searches resulted in a wide variety of equalizer types.

In the design of equalizers there exist different types of design criteria [2]. The most frequently encountered two criteria with their efficiency are told in the sequel. Some equalizers are designed to minimize mean square error (MSE) at the slicer input with the constraint of zero ISI. These are called Zero-Forcing (ZF) equalizers. Some equalizers are designed to minimize the MSE at the input of the slicer by reducing the signal slightly at the slicer input. This reduction of signal results in reduction in MSE, so overall MSE is smaller than that of the ZF equalizer. These equalizers are called MSE equalizers. The MSE equalizer is generally preferred against ZF equalizer because of less noise enhancement.

The linear equalizer is cheap in implementation but its noise performance is not very good. So, in the literatures, some equalizer types which introduce nonlinearity are searched. The most popular of these nonlinear equalizers is the decision feedback equalizer (DFE). The DFE is first proposed by Austin in [19]. This equalizer results in less MSE against linear equalizer, but it has the disadvantage of error propagation in its feedback loop.

As it is told before, most of the time, the channel's and, consequently, the transmission system's transfer functions are not known. Also, the channel's impulse response may vary with time and fade. The result of this is that the equalizer can not be designed a priori, frequently. So, mostly preferred scheme is to exploit adaptive equalizers. Adaptive equalizers use adaptive algorithms to converge to the true coefficients and have the benefit of tracking the changes in the channel impulse response. But, to achieve this, it adds additional complexity to the receiver structure.

Also, the adaptation algorithm plays a significant role for the performance of the equalizer. The most popular algorithm from the aspect of performance and complexity is the Least Mean Squares (LMS) algorithm. It has a good performance and low complexity. It is globally convergent if the desired values are given correctly. The handicap of LMS algorithm for equalizer if the desired symbols are not correct, it does not converge. So, the equalizer using LMS algorithm requires a priori known symbols in case the decisions of the equalizer are wrong. The better algorithm is the Recursive Least Squares (RLS) algorithm which has better convergence characteristics than the LMS algorithm. But, it has higher computational complexity than LMS algorithm. The general RLS algorithm's complexity grows with N^2 where N is the number of equalizer coefficients. There are also RLS algorithms that have computational complexities that grow linearly with the number of equalizer coefficients. These algorithms are called fast RLS algorithms [1].

To get a satisfactory result from the adaptive equalizers, the equalizers are adapted with a a priori known symbol sequence, especially, at the start of the communication. This training period enables the equalizer to reach a point near to the optimum level, but most of the time it is costly. The reason for the costly situation is that the training sequences are not present or not possible to send most of the time. But, there are also situations, where training sequences can be applied.

When the training sequence is not present, the equalizer has a hard job. The usual adaptive equalizers need the initial knowledge of the channel which may have distorted the signal in an unrecoverable way. When this knowledge is not present, it causes the equalizer not to converge. The solution to this problem is the use of blind equalizers. Blind equalizers use different adaptive algorithms that exploit higher order statistical characteristics or cyclostationary statistics of the received signal. The former technique contains the Bussgang algorithms [3] which use higher-order statistics of the signal in an implicit way. For the blind equalizers, the most popular algorithm with its performance is the Constant Modulus Algorithm (CMA) [1], [3]. For blind equalization, CMA has a wide acceptance.

1.3. Outline of The Thesis

The objective of this thesis is the evaluation of the performance of a successful blind equalizer scheme versus the trained DFE which incorporates the periodic retransmission of the training sequence. The simulation environment of the equalizers includes severe channels with AWGN and timing error introduced to the received signals. In this situation of the simulation environment, the performances of the equalizers are compared and results are evaluated in terms of Equalizer Output Diagrams, Mean Square Error (MSE), Bit Error Rate (BER) performance and Residual ISI (RISI) performance.

Chapter 2 treats the subject of trained approach. Firstly, the timing recovery subject is treated and the timing recovery algorithms used are briefly discussed. Secondly, the frame synchronization scheme is introduced with its basic form. Finally, the equalizer scheme, DFE with the trained approach is discussed.

Chapter 3 discusses about the blind approach. The basic blind equalization is introduced, and the popular algorithm constant modulus algorithm (CMA) is treated. The blind equalizer scheme which uses CMA in its adaptation process is discussed. The scheme has two configuration models one of which can be selected according to the situation. This blind scheme will be named shortly Unsupervised DFE (UDFE).

Chapter 4 gives information about the simulation environment and discusses the results of the simulations. The simulation platform and the simulation procedure are told. Then the results of the simulations that are done for different channel configurations are given and discussed by the aid of equalizer output diagrams which are similar to eye diagrams [12] in some manner, MSE diagrams, BER diagrams and RISI diagrams.

Finaly, Chapter 5 gives the conclusion of the thesis.

CHAPTER 2

TRAINED APPROACH

The classical way of adaptive equalizing is the equalizers which use training sequences to adapt their filter coefficients. Because of the infrastructure of the underlying channel or other reasons, sending training sequnces is not always possible. But, there are also situations, where training sequences can be applied.

The method for training sequence approach has a very common way. In this method, the training sequence between the transmitter and receiver is sent before the actual data transmission starts. In this way, the equalizer coefficients are updated to the values closest to the optimal ones. But, sometimes the resending of the training sequence is required because of a change in the channel. Some systems can resend the sequence, some can not.

In this chapter, the equalization process with training sequences is treated. Before the equalization, the timing recovery and phase recovery must be done. After timing and phase recovery, the signal received is used to adapt the filter coefficients during the training sequence transmission. In this thesis, phase recovery is not treated, but timing recovery is considered. Also, the approach different from the simple approach, which sends the training sequence once at the beginning, is considered. In this approach, the transmitted symbols are put in frames and the training sequence is used as a header symbol sequence in front of each frame. So, each frame consists of a training sequence which enables the equalizer to adapt continuously for each training sequence. In order to have this scheme work, the frames must be recognized, that is, the start of the frames must be found. Finding the first symbol of the frame is called frame synchronization.

2.1. Timing Recovery

The timing recovery procedure aims to realize symbol synchronization. The symbol synchronization means to correctly sample the received signal at the correct timing frequency, 1/T and at the correct timing phase, τ which is between 0 and T. Our received signal was

$$r(t) = s(t;\tau) + n(t) \tag{2.1}$$

if we assume there is no carrier offset. If we call this correct time instant t_c, the desired result will be:

$$t_c = kT + \tau \tag{2.2}$$

where k represents the symbol number. The procedure has two parts [2]. The first part is to lock on the symbol rate (1/T) that is the frequency that symbols are sampled. The second part is to find the peak point of the pulse - that is the timing phase (τ) - used to transmit symbols in order to maximize SNR and correctly recover the transmitted symbols. The timing recovery performance is evaluated with the correctness of these two values. The frequency deviation is always present and introduces timing jitter. The timing jitter may be corrected with a suitable timing recovery circuit. The timing phase performance is affected by the shape of the pulse to transmit the symbols. The high excess bandwidth in the design of the pulse shape results in more open eye diagram of the signal, so better performance for the deviations of the timing phase. The general digital timing recovery system is given in Figure 2.1.1 [1].

The receive filter is a FIR filter that has the impulse response that matches to the impulse response of the transmitted pulse shape, $p_{root}(t)$ in order to maximize Signal-To-Noise Ratio (SNR). So, the impulse response of the receive filter, $p_R(t)$ is:

$$p_R(t) = p_{root}(-t) \tag{2.3}$$

where $p_{root}(t)$ is the transmitted pulse shape. Generally, the transmitted pulse and

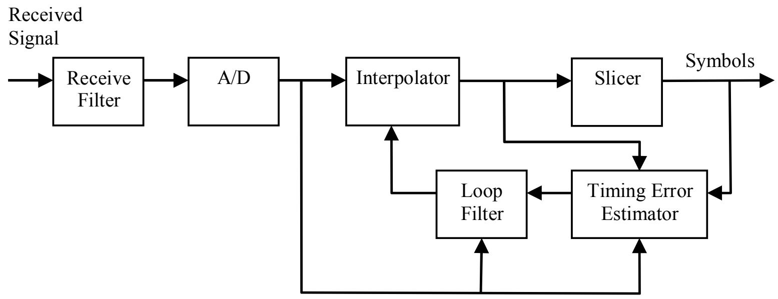


Figure 2.1.1. General Digital Timing Recovery System for an Ideal Channel

receive filter impulse response have the root-raised cosine impulse response. So, the overall pulse shape that results from the convolution of $p_{root}(t)$ and $p_R(t)$ is the raised-cosine pulse, p(t):

$$p(t) = p_{root}(t) * p_R(t) = \int_{-\infty}^{\infty} p_{root}(\tau) p_R(t-\tau) d\tau \tag{2.4}$$

where * stands for convolution. The receive filter output is given to the A/D converter to obtain the signal samples. These samples are given to the interpolator to obtain the intermediate values of the samples. Then, these values are given to the timing error estimator to obtain the error signal to adjust the new timing error. There are various algorithms that calculate the error signal. The error signal is used by the loop filter to obtain the new timing error. The loop filter is, generally, a second order filter that corrects the frequency error and the phase error. Then the new control signal is used by the interpolator to calculate the new intermediate samples of the A/D output samples.

In this section, the timing recovery algorithms will be discussed. The first one is Gardner's Synchronizer and the second one is ML-based timing recovery algorithm. The two algorithms were used in simulations. The reason to use these algorithms is that they do not require the correct decisions, so they can be used before equalization process. Finally, the linear interpolation subject will be considered.

2.1.1. Gardner's Synchronizer

The Gardner's Synchronizer [10] is proposed for Binary Phase-Shift Keying (BPSK) and Quadrature Phase-Shift Keying (QPSK) modulation schemes. The timing recovery method assumes a band-limited pulse is transmitted.

The property of the method is its requirement of only two samples per symbol for the estimation of the timing recovery. Moreover, one of the samples is used for symbol detection. There are some algorithms which use one sample per symbol like Mueller and Müller [11] but they require the symbol decision or the correct symbol for the algorithm. The block diagram of the algorithm is given in Figure 2.1.1.1.

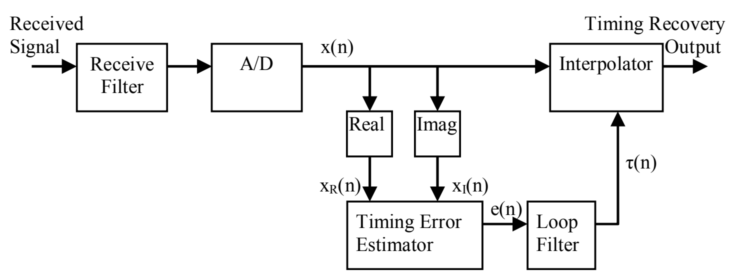


Figure 2.1.1.1. The Block Diagram of the Gardner's Synchronizer

As in the block diagram, the timing error estimator calculates the error signal as needed according to the following equation:

$$e(k) = x_R(k-1/2)[x_R(k) - x_R(k-1)] + x_I(k-1/2)[x_I(k) - x_I(k-1)] \tag{2.5}$$

and, the loop filter can be a simple update equation given below,

$$\tau(k+1) = \tau(k) + \alpha e(k) \tag{2.6}$$

where α is a suitable step size and $\tau(0)=0$.

The algorithm does not depend on carrier phase so timing recovery can be done before phase recovery. The characteristics of the algorithm are told in [10]. When the BPSK modulation is used for the symbols, the imaginary part of the symbols will be zero. But, until carrier recovery is completed, the imaginary part of the received samples will contribute some information to the timing recovery. So, until carrier recovery is completed, the algorithm must consider the imaginary values. After carrier recovery is completed, the imaginary values will only carry noise information. Because of this, after carrier recovery, only real parts must be taken into account. Since, in baseband transmission, only real parts will be present, for this transmission, only real values must be considered as well. For QPSK tranmission, since both real and imaginary parts are used in modulation, both of them must be taken into account.

The algorithm has the best performance for pulses with 40 to 100 percent excess bandwidths, because while the bandwidth gets narrower, the noise affects badly the timing recovery performance. So, for these kinds of pulses, another algorithm which considers the nonlinearity which occurs because of these pulses must be used for narrow bandwidths.

The work of the algorithm can be summarized. The sampler normally takes samples at the symbol instants. For the algorithm, an extra sample must be taken between the two strobe locations. The algorithm is designed to work at the transition instants. That is, for a BPSK modulation, when a transition occurs from a -1 symbol to +1 symbol, or from a +1 symbol to -1 symbol, the algorithm will give timing error. When there is no timing error and a transition occurs, the middle sample will give a zero value. This means a zero error. When the middle sample during a transition, has a value other than zero, it will be proportional to the error. But, since which transition occured is not known (1 to -1, or -1 to 1), the error term is not enough to detect the timing error value. So, to detect the direction that the algorithm will update its timing error value, the difference of the strobe samples is used. As it is told, if a transition is not present since the strobe samples will be the same, the difference will be zero, so middle sample will be multiplied with zero, which results in an unchanged timing error value. When the strobe samples are different, we will obtain a direction value for the middle sample, so the update of the algorithm will result in a change in the timing error.

In order to eliminate fluctuations of the strobe samples because of noise, the sign of the strobe samples may be used. This operation will result in a better noise performance. If, also, the equalization is done before timing recovery, the sign operation will result in a decision-directed operation for timing recovery algorithm. But this kind of operation degrades the acquisition behavior of the receiver, while improving the tracking behavior.

The algorithm introduces self-noise. This self noise originates from the fact that the zero-crossings of the pulses which have less than 100% excess bandwidth do not lie at the mid-point of the strobe instants. This results in a self noise introducing of the algorithm. But, in the average, the mean of the noise is zero.

The algorithm works by considering the three samples of the received signal. This results in that timing recovery is calculated within this three sample period. For slow changes of the timing error, this will have no effect. But, for faster changes of the timing error, the timing error at the first sample and at the last sample may be different.

At a time later than the algorithm it was proved that, under certain conditions (enough observation, excess bandwidth<100%, only significant slope values of pulses at $\pm T/2$, at almost no ISI) the algorithm yields maximum likelihood estimate of the timing error.

2.1.2. Non-Data-Aided Maximum Likelihood Based Timing Recovery

There are timing recovery algorithms that depend on symbol decisions, like MMSE timing recovery, and there are algorithms which do not depend on decisions. Non-Data-Aided Maximum Likelihood Based Timing Recovery (NDA-MLBA) is an algorithm given in [8] to form an algorithm which is derived from maximum likelihood estimation. In general decision directed algorithms have a better performance in tracking the timing error than the quadratic algorithms, that is, the algorithms which have second order cost function. But, also sometimes decisions are not available or reliable. Even if decisions are available, the algorithm may have to be less complex or fast convergent. For these situations, non-feedback (feedforward) algorithms may be preferable.

The algorithm is formed depending on the maximum likelihood oriented arguments, then by using some approximations, it is simplified into another form. The algorithm works on the sample rates that are multiple of the symbol rate (T/2, T/3, T/4,…), but T_s=T/2 is enough for the algorithm to work [8].

The algorithm uses the Fourier series characteristics of the transmitted pulse with timing error by only considering a few Fourier series coefficients of the overall pulse with timing error in case $p_{root}(t)$ is bandlimited to ±1/T. The final block diagram of the algorithm is given in Figure 2.1.2.1.

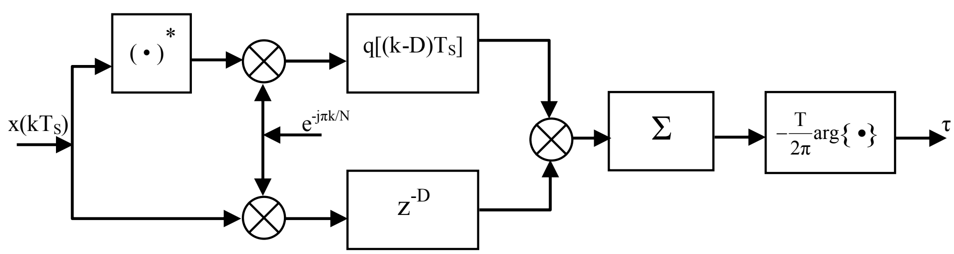


Figure 2.1.2.1. The Block Diagram of NDA-MLBA

In the block diagram, the filter, q(t) has the impulse response,

$$q(t) = \frac{\alpha}{\pi}\frac{\cos(\pi\alpha t/T)}{1-(2\alpha t/T)^2} \tag{2.7}$$

where α is the roll-off factor of the transmitted pulse. The samples, $x(kT_S)$ are obtained by sampling the outputs of an antialiasing filter which has an ideal brick-wall transfer function. The antialiasing filter will have sufficient bandwidth in order not to distort signal components. The resulting equations of the algorithm are given below:

$$\hat{\tau} = -\frac{T}{2\pi}\arg\left\{\sum_{k=D}^{N(L_0+C)-1} x[(k-D)T_S]e^{-j\pi(k-D)/N} z[(k-D)T_S]\right\}, \tag{2.8}$$

$$z(kT_S) = [x^*(kT_S)e^{-j\pi k/N}] * q(kT_S) \tag{2.9}$$

where N is the samples per symbol period, L_0 is the observation interval in terms of T, D is the delay value, T_S is the sampling period. C is an integer and is the semi-duration of q(t) in symbol period, that is,

$$q(t) \approx 0 \quad \text{for } |t| > CT \tag{2.10}$$

The outputs of the filter are given to the two parallel branches. T_S is the sampling period. $x(kT_S)$ is first given to the first branch and it is complex-conjugated, and multiplied with $e^{-j\pi k/N}$. The result is filtered with q(t). In the other branch, the result is multiplied with $e^{-j\pi k/N}$ and delayed D symbol periods. The outputs of the branches are multiplied and the summation is calculated with addition of the previous result. Finally, the argument of the total sum is calculated and scaled with the factor $-T/2\pi$.

But, although this algorithm has a good performance, it needs an antialiasing filter. Another algorithm which is proposed by Oerder and Meyr in [18] uses another equation. It uses a filter matched to the pulse shape instead of antialiasing filter. Also the branch with the filter, q(t) and the delay block are discarded. But, it needs an oversampling of 4 times the symbol period. The resulting equation is

$$\hat{\tau} = -\frac{T}{2\pi}\arg\left\{\sum_{k=0}^{NL_0-1} |x(kT_S)|^2 e^{-j2\pi k/N}\right\} \tag{2.11}$$

where again L_0 is the observation period in terms of symbol period, and N is the samples per symbol, and this time $x(kT_S)$ is the samples which are outputs of the filter matched to the pulse shape.

2.1.3. Interpolation

Interpolation is the reconstruction of a waveform from its samples [8]. Interpolator is used for non-synchronous sampling, that is, when synchronous sampling is not used. The initial requirement of the interpolation is the sampling period's value. If we call the bandwidth of the received signal that is the output of the receive filter B_R, then the sampling period must satisfy:

$$\frac{1}{T_S} \geq 2B_R \tag{2.12}$$

So, the signal samples will be enough to reconstruct the intended waveform. With this condition satisfied, the interpolation equation can be given as below:

$$x(t) = \sum_{i=-\infty}^{\infty} h_I(t - iT_S)x(iT_S), \tag{2.13}$$

$$h_I(t) = \frac{\sin(\pi\, t/T_S)}{\pi t/T_S} \tag{2.14}$$

This interpolation scheme is called ideal interpolation [8]. But this interpolation scheme is impossible to implement because of the infinite summation. There are some other interpolation methods which consider a few terms [8]. By replacing the $h_I(t)$ with some piecewise polynomial functions of durations which may be 2 or 4 times the T_S, good interpolators are formed. The general methods of interpolation will not be discussed here. In interpolation, some variables are defined. When the timing error is τ, L_k is the index of the sample at time t_l that is the time of the sample just before the desired waveform point at time t_k with this timing error. L_k is calculated as below:

$$L_k = \mathrm{int}(\frac{t_k}{T_S}) \tag{2.15}$$

L_k is called basepoint index. Also, the fraction, μ_k which is the position of the time t_k according to the sample time, t_l, is calculated as below:

$$\mu_k = \frac{t_k}{T_S} - L_k \tag{2.16}$$

μ_k is called fractional interval. The interpolation equation updated according to the piecewise linear approximation is given below:

$$x(t_k) = \sum_{i=-V_1}^{V_2} a_i(\mu_k) y[(L_k - i)T_S] \tag{2.17}$$

where $a_i(\mu_k)$ are the new interpolation filter coefficients which depend only on the fractional interval, μ_k and V_1 and V_2 are two integers. The simplest interpolation method is linear interpolator. For linear interpolation, the interpolation filter coefficients are given below:

$$\begin{aligned} a_{-1}(\mu) &= \mu \\ a_0(\mu) &= 1 - \mu \end{aligned} \tag{2.18}$$

where V_1=-1, V_2=0.

For parabolic interpolator V_1=-2 and V_2=1 and the filter coefficients are

$$\begin{aligned} a_{-2}(\mu) &= \alpha\mu^2 - \alpha\mu \\ a_{-1}(\mu) &= -\alpha\mu^2 + (1+\alpha)\mu \\ a_0(\mu) &= -\alpha\mu^2 - (1-\alpha)\mu + 1 \\ a_1(\mu) &= \alpha\mu^2 - \alpha\mu \end{aligned} \tag{2.19}$$

where α is a parameter for accuracy of the interpolation and is generally 0.5. Generally, linear interpolator has enough performance. In this thesis, linear interpolator is used.

2.2. Frame Synchronization

Frame synchronization is done to find the starting position of a frame. The usual way of frame synchronization is to use a sync word that is appended usually at the start of the frame [13]. Then, the distance metric is minimized between the sync word and the received signal. The minimum point of the metric shows us the start of the frame. Since the structure of the adaptive DFE simulated is based on continuously trained DFE with a training sequence for each frame, the frame synchronization is a requirement for the receiver model. In [13] and [15], the standard approach of frame synchronization and a new approach depending on the decoder-assisted frame synchronization scheme for convolutionally encoded signals are represented. In this thesis, the frame synchronizer is assumed ideal.

The standard approach depends on the calculation of a distance metric between the sync word and the received signal. If it is assumed that the sync word which is used as training sequence has length N such that it is always a constant BPSK sequence and

$$d = (d_0 , d_1 , \ldots\ldots, d_{N-1}) \tag{2.20}$$

is the sync word symbol sequence.

The sequence d is generally designed to minimize false frame detection. The general method for the design of this sequence is to use a criterion that will determine the circular correlation of the sequence according to some restriction. The usual restriction to the circular correlation of the sequence is to permit two correlation values. C_{max} is the maximum at the zero-shifted index of the sequence and C_{min} is the minimum at the other shifts of the circular correlation. This sequence coding is called Pseudo Random (PR) sequence. There are also other sequence design types which allow three or four circular correlation values. The equations describing the PR sequence are given below:

$$C_{max} = \frac{1}{N}\sum_{i=0}^{N-1} d_i d_i \tag{2.21}$$

$$C_{min} = \frac{1}{N}\left(\sum_{i=m}^{N-1} d_i d_{i-m} + \sum_{i=0}^{m-1} d_i d_{N+i-m}\right) \quad m \in [1, N-1] \tag{2.22}$$

The PR sequence in the thesis is used to adapt the trained DFE. The PR sequence is used as preamble, that is, in front of data symbols.

2.3. Equalizers

The need for equalizers arise from the fact that the channel has amplitude and phase dispersion which results in the interference of the transmitted signals with one another [1]. The design of the transmitters and receivers depends on the assumption of the channel transfer function is known. But, in most of the digital communications applications, the channel transfer function is not known at enough level to incorporate filters to remove the channel effect at the transmitters and receivers. For example, in circuit switching communications, the channel transfer function is usually constant, but, it changes for every different path from the transmitter to the receiver. But, there are also nonstationary channels like wireless communications. These channels' transfer functions vary with time, so that it is not possible to use an optimum filter for these types of channels. The Intersymbol Interference (ISI) caused by the channels results in a very high increase in Bit Error Rates (BER) .In order to solve the problem of ISI caused by the channels the equalizers are designed.

The optimum equalization method according to the criterion of minimum probability of error is maximum likelihood sequence detection (MLSE) [1]. But it has high complexity. The other method is to use the linear combinations of the received signal samples to remove or reduce the ISI. This equalizer is called linear equalizer and it has limited performance. Also, there is an equalizer type which uses the past decisions of the demodulator to remove or reduce the ISI. This equalizer is called Decision Feedback Equalizer (DFE). Also, Fractionally Spaced Equalizer is a type of equalizer which works at over-baud-rate, that is, at a rate higher than the symbol rate. There is also blind equalization scheme which is told in the next chapter.

Generally, the received signal $r_c(t)$ can be given as below

$$y(t) = \sum_{k=0}^{\infty} I_k p(t - kT) \tag{2.23}$$

$$r_c(t) = c(t; t_1) * y(t) + n(t) = s(t) + n(t) \tag{2.24}$$

where I_n are the transmitted symbols, p(t) is the pulse shape impulse response, $c(t;t_1)$ is the channel impulse response and n(t) is the AWGN, and * stands for convolution. An equivalent discrete-time model can be given for the upper continuous-time model:

$$r_{c,k} = c(k, k_1) * y_k + n_k = s_k + n_k \tag{2.25}$$

where all arguments are the kth and k_1st samples of the relevant arguments. The variance of the noise sequence is given with the following equation:

$$E(n_k^* n_m) = N_0 \delta_{km} \tag{2.26}$$

where N_0 is the variance of the noise and δ is the delta function. Generally, the received continuous-time signal is filtered with a continuous-time receive filter which is matched to the transmitted pulse shape. The output of this receive filter in discrete-time can be given as below:

$$r_k = x_k + z_k \tag{2.27}$$

where x_k is the signal component that has the channel effect and z_k is the filtered noise sequence. The equalizer aims to recover the original symbols from the received signal. To achieve this, there are some equalizer design criteria. The maximum likelihood criterion is used generally with Viterbi Algorithm. The other criteria are discussed below.

2.3.1. Criteria

2.3.1.1. Zero-Forcing Criterion

The zero-forcing criterion has the main purpose of equalizing the output of the receive filter to the tranmitted symbol; after that the criterion requires the minimization of the noise at the input of the decision device [2]. In the view of equalization, this means to remove all of the ISI. The criterion forms from two parts. The first part is to minimize the error at the decision device. The second part is the constraint of equalizing the equalizer transfer function to the inverse of the channel transfer function. The error function that will be used as the cost function will have the following equation:

$$\varepsilon^2 = E(|Q_k - I_k|^2) \tag{2.28}$$

where ε^2 is the error variance, Q_k is the decision device input and I_k is the correct transmitted symbol. The constraint that will be used for the design of the equalizer will be:

$$C(z)F(z) = 1 \tag{2.29}$$

where C(z) is the channel transfer function and F(z) is the equalizer transfer function. The variance of the error with this design criterion will be:

$$\varepsilon^2 = 2N_0 \sum_{k=-\infty}^{\infty} |f_k|^2 \tag{2.30}$$

so, minimizing the error with the given constraint will result in the following transfer function of the equalizer:

$$F(z) = C^{-1}(z) \tag{2.31}$$

The solution of the zero-forcing criterion is the match filter solution to the problem, as it is seen from the equation. So, the solution assumes a match filter solution and assumes the ISI interference is solved completely. But, most of the time, ISI removal causes noise enhancement in some situations [2]. This will result in larger MSE occurence, so the result obtained here can be a lower bound for zero-forcing design.

2.3.1.2. MSE Criterion

The zero forcing criterion minimizes MSE by the constraint of zero ISI. Since the channel inverse is completely implemented, the only term contributes to error is the noise. MSE criterion improves this approach by reducing the signal variance at the decision device input, so the noise will decrease, too [2]. With this approach, the overall MSE will be formed from the noise and some ISI. With this scheme, the error of MSE criterion will be less than that of the zero-forcing criterion.

The general gain that is achieved with MSE criterion over zero-forcing criterion is:

$$\text{gain} = \frac{\varepsilon^2(\text{MSE})}{\varepsilon^2(\text{ZF})} = \frac{\sum_k |c_k|^2}{\sum_k |c_k|^2 + \frac{2N_0}{\sigma_S^2}} \tag{2.32}$$

where c_k are the channel impulse response coefficients, and σ_S^2 is the signal variance. It is seen from the equation that since the second term in the denominator is larger than zero, the gain will be less than 1 which results in that the MSE criterion overall error is always less than zero-forcing criterion's overall error [2].

As in the case of the zero-forcing criterion, the result obtained here doesn't care about the noise enhancement results from the equalization, so the MSE solution told here with the gain equation is a lower bound for the MSE criterion's overall error.

2.3.2. Linear Equalizer

At the output of the receive filter and symbol rate sampler, the equivalent channel transfer function will not be constant. So, when the output is applied to the decision device, it will carry ISI with noise. If the channel effect isn't removed, there will be errors at the output of the decision device. The simplest suboptimum solution to this problem is the use of linear equalizer (LE) [1]. This approach generally incorporates a transversal filter. The filter has computational complexity which depends linearly on the channel's equivalent transfer function, that is, the dispersion length of the channel. But, the linear equalizer has a side-effect. The aim of the linear equalizer is to compensate for the frequency dispersion of the channel. So, at some frequencies where channel has introduced loss, the equalizer must amplify the signal to recover the original signal. This results in the amplification of the noise, too [2]. So, when the signal has reached to the decision device, it carries more noise than previous.

The general equation that expresses the equalizer output that will be given to the decision device is given below:

$$Q_k = \sum_{i=-K}^{K} f_i r_{k-i} \tag{2.33}$$

where Q_k is the decision device input, r_k are the sampler outputs, and the f_k are the linear equalizer coefficients. The block diagram of the system with the linear equalizer is given in Figure 2.3.2.1.

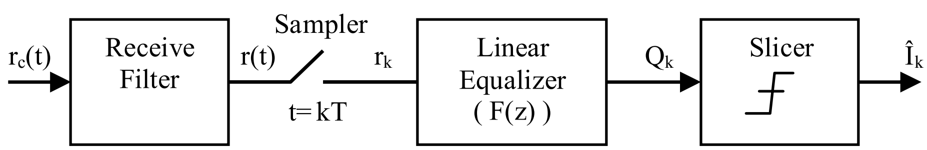


Figure 2.3.2.1. The Block Diagram of the Receiver with Linear Equalizer

2.3.3. Decision Feedback Equalizer

The performance of the linear equalizer is limited in the case of severe ISI. The Decision Feedback Equalizer (DFE) is the improved equalizer according to the linear equalizer by the introduced nonlinearity [2]. The DFE has a noise reduction according to the linear equalizer which suffers from noise enhancement. The drawback of DFE is the error propagation in its feedback loop.

DFE is composed of two filters: The feedforward filter and feedback filter. Both are implemented to work at, in general, symbol rate. Sometimes, the feedforward filter can be fractionally spaced. The feedforward filter is a transversal filter which gets the output of the receive filter and sampler as input. So, it is some way like the linear equalizer. The feedback filter gets the past decisions of the transmitted symbols at the output of the decision device as input. Since the feedback filter uses the past decisions for its output, it is strictly causal. The block diagram of the DFE is given in Figure 2.3.3.1. [2].

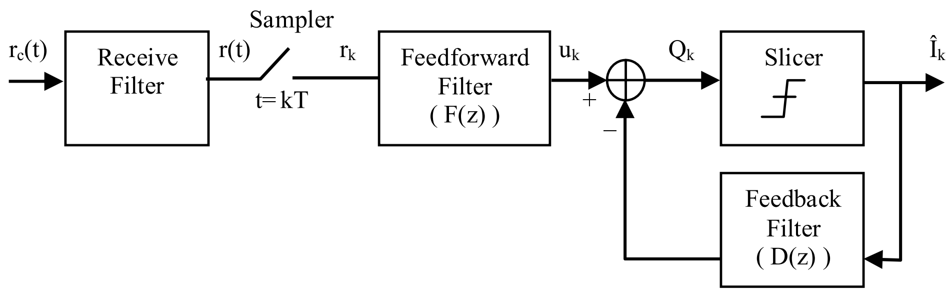


Figure 2.3.3.1. The Block Diagram of Decision Feedback Equalizer

The feedforward filter removes some of the ISI from the received signal, but leaves some of the postcursor ISI on the signal (generally, all postcursor ISI). The feedback filter estimates the residual ISI from the past decisions and subtracts it from the feedforward filter output. The DFE solution is better than linear equalizer with a low-complexity solution.

The low noise enhancement of the DFE arises from the fact that, by assuming no decision errors, the decision device removes all the noise present in the signal [2]. So, the inputs of the feedback filter have no noise, so the outputs of the feedback filter have no noise. Also, the feedforward filter has the less complex problem of removing only precursor coefficients. This results in a better performance for the feedforward filter according to the linear equalizer. But, the assumption of correct symbol decisions at the output of the decision device may not work in practical cases, so error propagation occurs and the performance of the equalizer is degraded.

2.3.3.1. Zero Forcing DFE

The zero forcing DFE is designed according to the zero forcing criterion. Also, there is MSE DFE. The zero forcing DFE design assumes no decision errors occured at the decision device output. The MSE DFE has less noise enhancement than zero forcing DFE like linear equalizer.

In ZF-DFE the feedforward filter is designed first [2]. It removes all ISI except postcursor ISI. Then the feedback filter is implemented to remove the remaining postcursor ISI. The ZF-DFE design is implemented with the same aim as the linear equalizer: the inverse transfer function of the equivalent channel. The ZF DFE block diagram is given in Figure 2.3.3.1.1.. With the inverse of the channel $S_C^{-1}(z)$ is implemented. The DFE feedforward filter comprises the filter that is the inverse of the channel and the second filter (1+D(z)) in Figure 2.3.3.1.1.. The second filter's aim is to reduce the noise to the minimum level. It is assumed that the feedback filter has no erroneous decisions. So it introduces no noise, since it uses the correct symbols with the decision device that removes noise. Also, with the design like in Figure 2.3.3.1.1., the effect of the second filter of the feedforward filter on the signal is removed with the feedback filter. That is, first, the channel ISI is completely removed with $S_C^{-1}(z)$, then some ISI, postcursor ISI, is introduced with (1+D(z)), and finally, the ISI is removed with feedback filter. So, from the point of view of signal carrying symbols, nothing changes. But, with this scheme, feedback filter introduces no noise, and by suitable design of the (1+D(z)), the noise that will be input to the decision device can be minimized. This is possible if the channel has ISI, since, then, the output of the channel inverse filter will have colored noise, and it can be whitened with a suitable whitening filter which will be linear prediction filter [17]. The result is the whitened noise sequence that is input to the decision device.

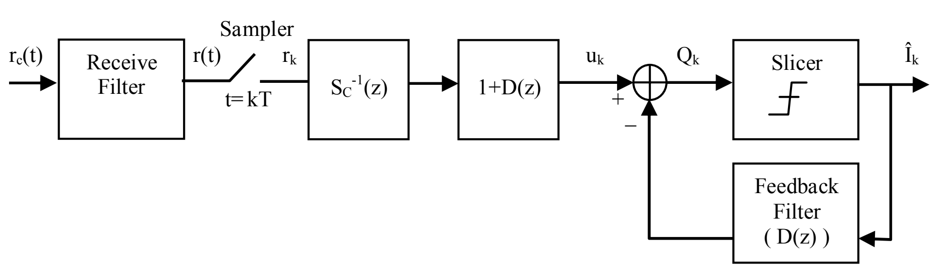


Figure 2.3.3.1.1. The ZF DFE Block Diagram

The structure in Figure 2.3.3.1.1. aims to clearify the DFE scheme. In fact, the general DFE structure is obtained by combining the inverse channel filter and linear prediction error filter which form the feedforward filter [2]. The overall pulse shape that is at the output of the feedforward filter will contain postcursor coefficients. These coefficients are used to implement the coefficients of the feedback filter.

The optimum ZF DFE will have the following error variance at the output of the equalizer:

$$\varepsilon^2 = 2N_0 \exp\left(\frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} \ln(S_C^{-1}(e^{j\omega T}))d\omega \right) \tag{2.34}$$

where N_0 is the noise variance. This result includes the geometric mean of the channel inverse Fourier transform. With this result, the error variance of the ZF DFE is less than the error variance of the ZF linear equalizer. But the ZF DFE error variance has a lower bound of the exact match filter error variance. The feedforward filter will have the following transfer function:

$$C(z) = S_C^{-1}(z)(1 + D(z)) \tag{2.35}$$

The spectral factorization of the white noise at the output of the feedforward filter will include the term $(1+D^*(1/z^*))$, so the feedforward filter transfer function can be shown as below:

$$C(z) = A \frac{1}{1 + D^*(1/z^*)} \tag{2.36}$$

where A is a constant. So, since (1+D(z)) is strictly causal and has all zeros inside the unit circle, C(z) will have all poles outside the unit circle, so it will be anticausal. This shows that the feedforward filter in ZF DFE will be anticausal which is logical in the sense that the filter removes precursor coefficients of the input pulse shape.

2.3.3.2. MSE DFE

The MSE DFE is designed according to the same cost function with ZF DFE, but this time, the constraint of exact inverse of channel is removed. The result is the same as other equalizer types: The MSE criterion results in less MSE than ZF criterion. The equalizer output of the DFE will be:

$$Q_k = \sum_{j=-L_1}^{0} f_j r_{k-j} - \sum_{j=1}^{L_2} d_j \hat{I}_{k-j} \tag{2.37}$$

where L_1+1 is the length of the feedforward filter and L_2 is the feedback filter length. Also, for simplicity, the filters are assumed finite length which is the practical case. The cost function of the MSE criterion is the same as the zero forcing criterion's cost function [1]:

$$J = \varepsilon^2 = E(|I_k - Q_k|^2) \tag{2.38}$$

As in the ZF DFE, the feedforward filter is designed first and then the feeback filter coefficients are obtained from the overall pulse shape at the output of the feedforward filter. The equations will be:

$$\sum_{j=-L_1}^{0} \rho_{ij} f_j = c_{-i}^{*} \quad i = -L_1, \ldots, -1, 0 \tag{2.39}$$

where,

$$\rho_{ij} = \sum_{m=0}^{-i} c_m^{*} c_{m+i-j} + 2N_0 \delta_{ij} \quad i, j = -L_1, \ldots, -1, 0 \tag{2.40}$$

where the input symbols are assumed independent and identically distributed (i.i.d.).

After the feedforward filter is designed according to the upper equations, the feedback filter coefficients are extracted from them [1]:

$$d_k = \sum_{j=-L_1}^{0} f_j c_{k-j}, \quad k = 1, 2, \ldots, L_2 \tag{2.41}$$

With the calculation of the upper coefficients of the feedback filter, the ISI is completely removed if the feedback filter has number of coefficients $L_2 \geq L$ where $L+1$ is the length of the equivalent channel impulse response.

The MSE obtained with this optimum design of the equalizer, is given below [2]:

$$\varepsilon^2 = 2N_0 \exp\left(\frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} \ln\left((S_C(e^{j\omega T}) + \frac{2N_0}{\sigma_I^2})^{-1} \right) d\omega \right) \tag{2.42}$$

This error value will be less than the minimum MSE of ZF DFE. Also, the above value will be less than the minimum MSE of MSE Linear Equalizer. Also, the MSE DFE has the lower bound of match filter MSE for the minimum MSE value.

The DFE has a superior performance according to the linear equalizer [1]. Also, the assumption of correct decisions fed back to the feedback filter holds. The addition of the feedback filter improves the performance. At the same number of equalizer taps, DFE has better performance than linear equalizer. Due to the residual interference in the output of severe channels, the performance of the DFE is degraded. The incorrect decisions fed back to the feedback filter degrade the equalizer performance. For unknown channels, the optimum match filter at the front end can not be designed correctly. In this situation, timing errors may occur at the DFE input. By using fractionally spaced filter for the feedforward filter, the timing error sensitivity may be reduced.

2.4. Adaptive Equalization

The previous section tells about the equalizers in the case of some assumptions. One of the assumptions is the known channel impulse response. In most cases, the channel impulse response can not be known. The channel can be a radio channel which can be fading, or a dial-up modem line which can have severe distortions on the signal. Also, the filters of the equalizer can not have infinite coefficients, since it is not realizable. The optimum filters must be approximated with enough finite-length filters. Since the channel impulse response can not be known a priori and also can be time-varying, the way to design the equalizers is to make them adaptive, that is, according to a performance index, let them to adapt themselves to reach the filter impulse responses that remove the bad effects of noise and ISI at the best way. To equalize the channel in this way is named as adaptive equalization [1].

The general block diagram of the adaptive equalizer is given in Figure 2.4.1. [2].

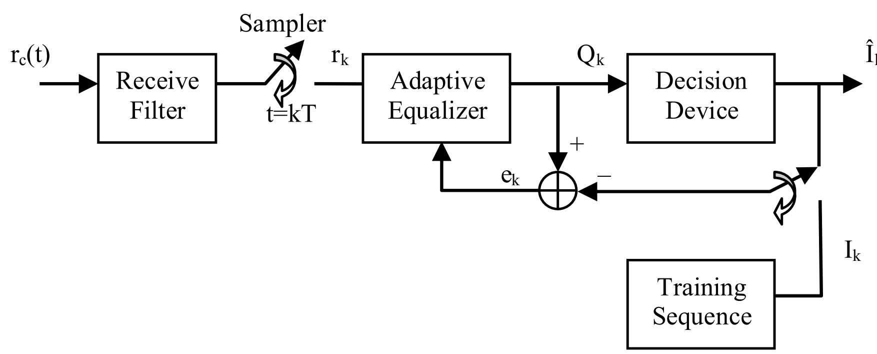


Figure 2.4.1. The General Block Diagram of the Adaptive Equalizer

In the above diagram, the receive filter can not be a match filter because of the unknown channel. But, it is generally low pass filter matched to the transmitted pulse shape to remove out of band noise. After the sampler, comes the equalizer. Equalizer is implemented with FIR filters [2]. The aim is to adapt the coefficients of the equalizer according to a performance index to minimize the ISI and noise to enable the decision device to decide on the correct symbols. The adaptation process is carried out with the error signal between the equalizer output and the correct symbol or symbol decision. The error signal is used by the adaptation algorithm to decide on how to change the filter coefficients.

The error signal is calculated between the decision device input which is aimed to be as close as possible to the decisions and the decision device output or the correct symbol. In order to use the decision device output to calculate the error signal, the decision device output must be correct to prevent false adaptation. In steady state, this is the case. Since, in steady state of the adaptation, the channel inverse is almost implemented, the decision device will decide on correct symbols. In this situation, if the equalizer output and the decisions are very close to each other, then the error term will be near to zero, so adaptation will not change the filter coefficients. If there is an error term away from zero, then the error term will indicate the direction and magnitude of the adaptation, to make the equalizer output close to the correct symbols. The decision-directed operation can protect the steady state, and can track the time variation in the channel until the decisions are not satisfactory. So, decision-directed scheme can track slow variations in the channel [2].

In the initial operation of the equalizer, and in the fast changing channel environments, the decision directed operation can not be enough to remove the ISI and minimize noise, because of the adaptation algorithm requires correct decisions [2]. In this situation, the alternative method indicated in Figure 2.4.1., the training sequence usage is exploited. The training sequence is a known sequence of symbols, which can be a pseudorandom sequence. In this operation, the transmitter sends the signal bearing the training sequence, when the reciever gets the signal, the receiver does not use the decisions to adapt the equalizer coefficients, instead, it uses the training sequence which is known in the receiver, to calculate the error signal:

$$e_k = Q_k - I_k \tag{2.43}$$

where I_k is the training sequence symbols. After some time after which the acquisition period is completed, the equalizer can turn to decision-directed mode and the normal transmission begins. In the situations where the sudden changes occur in the channel, retransmission of the training sequence may be needed, so since in severe fading channels, the channel transfer function may change continuously, retransmission of the training sequence may reduce the data rate and also sometimes, the training sequence may be impossible to resend.

The most used algorithm to adapt the filter coefficients is the Least Mean Squares (LMS) algorithm which is satisfactory in most situations. From the view of performance, the Recursive Least Squares (RLS) Algorithm, which is the exact recursive implementation of the Least Squares Problem, has better performance according to LMS algorithm, but it has higher computational complexity than LMS algorithm. The general RLS algorithm's complexity grows with N^2 where N is the number of equalizer coefficients. There are also RLS algorithms that have computational complexities that grow linearly with the number of equalizer coefficients. These algorithms are called fast RLS algorithms [1].

2.4.1. LMS Algorithm

The LMS algorithm is a linear adaptive filtering algorithm that belongs to the family of the stochastic gradient algorithms [3]. The stochastic gradient algorithms differ from the steepest descent algorithms in that the gradient is not calculated deterministically. The LMS algorithm has two parts. In the first part, the output of a transversal filter is computed according to the tap inputs and the error term is generated according to the difference between the filter output and the desired response. In the second part, the adjustment of the tap weights is done according to the error term. The block diagram of the LMS algorithm is given in Figure 2.4.1.1.

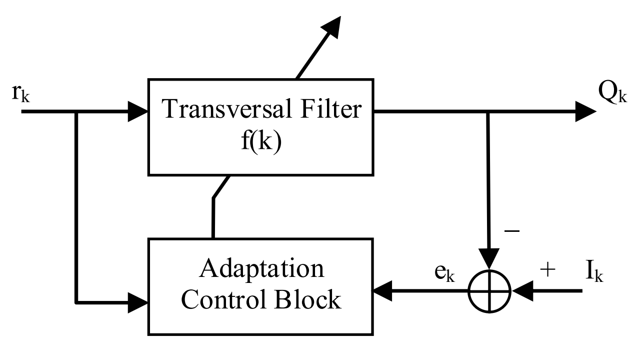


Figure 2.4.1.1. The Block Diagram of LMS Algorithm

The algorithm forms a feedback loop by the error term fed back. The filter produces an output and the difference between the output and the desired term is obtained. This difference is the estimation error term. The estimation error is given to the Adaptation Control Block. Adaptation Control Block multiplies the estimation error with the input taps' complex conjugate and a step size α. The results of the corresponding taps are added to the corresponding filter taps. So, the new filter is obtained [1]:

$$e(k) = I(k) - Q(k) = I(k) - f^T(k)\hat{r}(k) \tag{2.44}$$

$$f(k+1) = f(k) + \alpha\hat{r}^*(k)e(k) \tag{2.45}$$

$$f_n(k+1) = f_n(k) + \alpha r^*(k-n)e(k) \quad 0 \le n \le M-1 \tag{2.46}$$

where f(k) is the filter vector at time k, and $\hat{r}^*(k)$ is the complex conjugate of the input vector at time k, α is the step size parameter, e(k) is the estimation error, I(k) is the desired response at time k. In equation (2.46), $f_n(k)$ is the n^{th} tap of the filter at time k, and $r^*(k\text{-}n)$ is the complex conjugate of the input at time k-n, and other parameters are the same as first equation. Equations (2.45) and (2.46) are equivalent.

The small step size will result in less excess error but in slow convergence rate. The large step size will result in high excess error but high convergence rate.

2.4.2. Adaptive Decision Feedback Equalizer

The first job to implement an adaptive equalizer is to decide on which kind of filters will be used for the equalizer. The filters that will be used in adaptive filters must have constrained complexity, so the filters depending on the adaptive algorithm may be implementable. There are DFE structures, in theory, which may incorporate Finite Impulse Response (FIR) filters or Infinite Impulse Response (IIR) filters in both feedforward and feedback filters. Related information can be found in "DFE Tutorial"[1]. The general approach for the filters is to use a transversal filter which has finite taps. The MSE criterion is a good alternative for the adaptation of the filters, so LMS algorithm may be used to implement the adaptive equalizer. Section 2.4.1. gave an adaptation mechanism which is suitable for a linear equalizer, since it only adapts a transversal filter which has constrained complexity. The adaptive DFE is implemented with the addition of the feedback filter that takes the decisions or the training sequence as inputs. The adaptation mechanism will not change.

The block diagram of the adaptive DFE is given in Figure 2.4.2.1. [2].

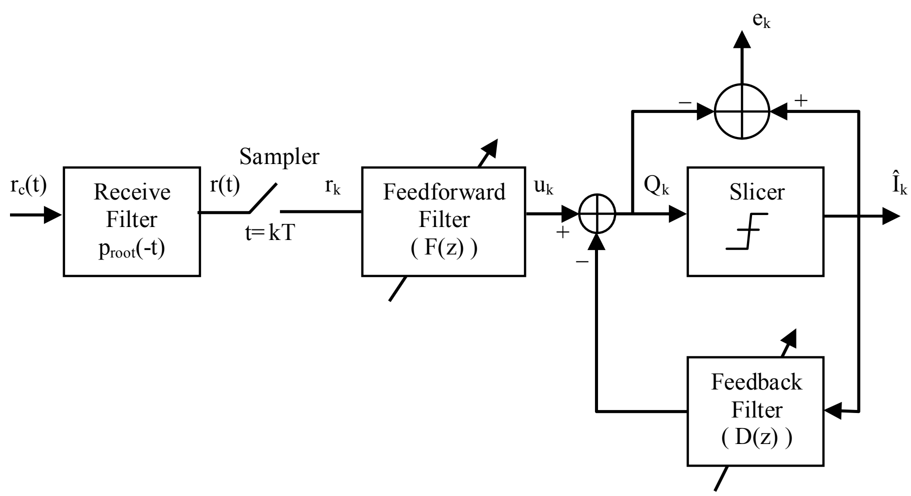


Figure 2.4.2.1. The Block Diagram of the Adaptive DFE

The input signal to the receive filter, $r_c(t)$ is the channel output which is embedded in additive noise. According to this structure, it is assumed that the demodulation is done before equalization. Also, it is assumed that the equalizer works at symbol rate. The extension of this equalizer to linear equalizer case can be obtained by choosing D(z)=0. The adaptation of the equalizer is done according to the error signal, e_k:

---

[1] DFE Tutorial was written by R. A. Casas, P. B. Schniter, Jaiganesh Balakrishnan, Jr. C. R. Johnson and C.U. Berg. DFE Tutorial is available at the web site: http://www.eleceng.ohio-state.edu/~schniter/postscript/dfetutorial.pdf Last visit: 29th January 2004.

$$e_k = I_k - Q_k \tag{2.47}$$

where I_k can be either $\hat{I}_k$ or the training symbol. $\hat{I}_k$ is used in decision directed mode and training symbol is used in training period. There can be decision errors because of the period of initial acquisition of the filters or the false equalizer impulse responses. The errors in the decisions will not be effective until the error rate passes 10% [2]. The feedforward filter will have the following z-transform [16]:

$$F(z) = \sum_{j=-L_1}^{0} f_j z^{-j} \tag{2.48}$$

and the feedback filter will have the following z-transform [16]:

$$D(z) = \sum_{j=1}^{L_2} d_j z^{-j} \tag{2.49}$$

where L_1 and L_2 can be different.

Since F(z) is anticausal, it is not possible to implement it with a transversal filter, so to make it causal, usually a delay of L_1 is introduced before F(z) [2]. The feedback filter is strictly causal.

The stochastic gradient algorithm LMS can be used to adapt the coefficients of the adaptive DFE. The decision device input, Q_k is:

$$Q_k = \sum_{j=-L_1}^{0} f_j r_{k-j} - \sum_{j=1}^{L_2} d_j I_{k-j} \tag{2.50}$$

where I_k will be substituted with decisions in decision-directed mode and training symbols in training mode. The first part in the equation will cancel precursor coefficients with the anti-causal coefficients and the second part in the equation will cancel postcursor coefficients with the strictly causal coefficients. The difference of the equalizer from the linear equalizer is the symbols used instead of received signal samples. With this change, the noise performance is improved [2].

According to the LMS algorithm, the cross-correlation vector is estimated by the idea of stochastic gradient. As known, the steepest descent algorithm calculates the exact cross-correlation vector. This substitution to the steepest descent algorithm will introduce gradient noise to the output of the LMS algorithm used in adaptive DFE as in the transversal filter case. The LMS algorithm calculates the equalizer output as below [2]:

$$V_{k+1} = V_k + \alpha e_k R_k^* \tag{2.51}$$

where V_k is the L_1+L_2+1-by-1 filter vector coefficients made up of

$$V_k = [f_{-L_1}, ..., f_0, d_1, ..., d_{L_2}]^T \tag{2.52}$$

and α is a suitable step size, e_k is the error term calculated as:

$$e_k = I_k - Q_k \tag{2.53}$$

and R_k^* is the complex conjugate of the L_1+L_2+1-by-1 input vector, R_k, coefficients made up of

$$R_k = [r_{k+L_1}, ..., r_k, I_{k-1}, ..., I_{k-L_2}]^T \tag{2.54}$$

The LMS algorithm for the DFE will have the upper configuration. In the R_k vector and the error term, the elements I_k are substituted with the decisions in decision directed mode and are substituted with the training symbols in training period. The performance of the LMS algorithm for the DFE is the same with the transversal filter case. The error propagation in DFE results in a situation which is mathematically hardly tractable, but, as mentioned before, less than 10% [2] error rate in decision directed mode will not affect the DFE.

CHAPTER 3

BLIND EQUALIZATION

The derivation of the equalizers in the previous sections involves the zero forcing and mean square error criteria. The equalizers with these criteria need transmission of a training sequence to obtain the initial adjustment of the equalizers or, sometimes, retransmission of the training sequence to adapt to the channel variations. But, in some situations, the need for skipping the training period, and adaptation with only the received signal is desirable. These equalization schemes are called blind or unsupervised equalization.

The first blind equalization algorithm is introduced by Sato [3]. The blind equalization algorithms can be classified into three groups. The first algorithm for equalizer adaptation is based on the steepest descent. The second group employs the second and higher order statistics of the signal to obtain the blind equalizer. The final one and more complex one is the employment of the maximum likelihood criterion.

The performance of the blind equalization depends on the characteristics of the input signal to the channel and the characteristics of the channel [3]. Generally, the input signal is made up of independent and identically distributed (i.i.d.) symbols and the probability distribution of the input signal is known. Generally, the channel is not known. If the channel is minimum phase, its transfer function will have all zeros inside the unit circle. So, the equalizer will have a tranfer function which is stable. Since the input signal is i.i.d., it will be the innovation of the channel output [3], and so the equalizer will be a whitening filter, so it is easy to solve the blind equalization problem. But, most of the time, the channel may have a transfer function which may not be minimum phase. This means that the channel has zeros outside the unit circle, since it must be stable. So, the equalizer may not be stable which results in a harder problem to solve. The examples to these kinds of channels are telephone channel and fading radio channel.

Adaptive equalizers need a training period, but this is sometimes impossible. For example in multipoint data networks, there is a master-slave situation and so the receivers of the DTEs in the network (slaves) need to be trained [3]. But since severe channel variations change the signals, the data and polling messages of the DCEs (master) are not recognized by the DTEs, or the initial synchronization of the network is missed by the DTEs, the training period will not be realized. Also the throughput of the network is increased because of the training sequence. This makes the blind equalization desirable.

Also in wireless communications, the repeated transmission of the training sequence decreases effective data rate. The multipath fading of the channel makes the data transmission impossible where the blind equalization is needed.

The blind equalization employs the statistics of the transmitted sequence to obtain the blind equalizer solution. The most popular algorithms for digital communications are the Bussgang family of blind equalization and more specifically, the Constant Modulus Algorithm (CMA). In the sequel, the information about CMA will be given, and a blind DFE solution which is simulated with the adaptive DFE and introduced by the Labat et al. [4] will be presented.

3.1. Constant Modulus Algorithm

Constant Modulus Algorithm (CMA) is a special case of Godard algorithm [14] and belongs to the Bussgang family of algorithms, which uses a memoryless nonlinearity function in the output of the equalizer in order to obtain the desired response [3]. Godard algorithm is a steepest descent algorithm and for the cases that no training period is present, it is employed [14]. Godard algorithm is proposed by Godard and the algorithm is designed for the joint operation of the equalization and carrier recovery in two dimensional communication systems. When this job is designed with LMS algorithm in the presence of desired symbols, the adaptive equations obtained for the two separate jobs are coupled, and in the absence of desired symbols, the equations governing the adaptation will not converge. Godard's approach is to separate the job of equalizing and carrier recovery in baseband. So, a cost function for the equalization process which does not depend on the carrier phase is obtained:

$$J(k) = E[(|Q_k|^2 - R_p)^2] \tag{3.1}$$

where Q_k is the filter output and R_p is a positive real constant. This cost function's optimization results in filter coefficients which equalize only symbol amplitude.This cost function does not depend on carrier phase. This cost function is differentiated and the expectation in the derivative is dropped to obtain an LMS type algorithm. The resulting update equation for the equalizer coefficients is as below [3]:

$$f_i(k+1) = f_i(k) + \alpha r^*(k-i) Q_k |Q_k|^{p-2} (R_p - |Q_k|^p) \quad -L_1 \le i \le L_1 \tag{3.2}$$

where α is a suitable step size, $f_i(k)$ is the i^{th} tap of the filter at time k, r(k-i) is the input at time (k-i), p is a positive integer, Q_k is the filter output and R_p is:

$$R_p = \frac{E(|I_k|^{2p})}{E(|I_k|^p)} \tag{3.3}$$

The algorithm introduces a penalty to the deviations of the equalizer output and the penalty is a constant. The algorithm must stop adaptation when perfect equalization is achieved, so the constant R_p's value results in the gradient of the cost function to be equal to zero, when $Q_n=I_n$.

So, the error term for the algorithm is:

$$e(k) = Q_k |Q_k|^{p-2} (R_p - |Q_k|^p) \tag{3.4}$$

With this error term, the algorithm uses the LMS algorithm to update the coefficients.Since the algorithm does not need carrier recovery, the algorithm has low convergence rate. The benefit is the separation of equalization and carrier recovery. Carrier recovery can be done with a decision directed LMS algorithm.

In the case which p=2, the algorithm gets a simple view [3]:

$$f_i(k+1) = f_i(k) + \alpha r^*(k-i) Q_k (R_2 - |Q_k|^2) \quad -L_1 \le i \le L_1 \tag{3.5}$$

and the phase update equation can be a decision directed LMS algorithm:

$$\varphi(k+1) = \varphi(k) + \beta \mathrm{Im}(\hat{I}_k Q_k^* e^{j\varphi(k)}) \tag{3.6}$$

where $\varphi(k)$ is the phase error, $\hat{I}_k$ is the decision at time k and β is a suitable step size. The value of R_p this time is as below:

$$R_2 = \frac{E(|I_k|^4)}{E(|I_k|^2)} \tag{3.7}$$

For the case of p=2, the algorithm is called constant modulus algorithm (CMA).

In order to have the equalizer to prevent sign ambiguity, all taps of the equalizer are initialized to zero except the center tap which is set to a nonzero value with the desired sign.

Among the Bussgang algorithms, the Godard algorithm is the most successful blind equalization algorithm [3]. Since the algorithm's cost function does not require carrier recovery, it is better than the other Bussgang algorithms when phase error is considered. Also, with respect to MSE, the Godard algorithm has better performance than other Bussgang algorithms. Also, in the simulations by Godard, it is shown that the Godard algorithm is convergent and opens the eye diagram successfully.

3.2. Unsupervised Decision Feedback Equalizer

The unsupervised adaptive DFE (UDFE), which is proposed in [4] by Labat, Macchi and Laot, has a novel structure which is a combination of two modes: starting and tracking. The predecessors of the method in [4] are firstly, [6] and then [7]. Later, the method is developed for the multiple input case by Labat and Laot in [5]. The method in [4] is also tested by Kurban in [20] with other blind DFE algorithms.

The equalizer's difference from the classical DFE is the good results in the environment of severe quickly time-varying channels. The superiority of this blind equalizer over classical DFE is the two modes of the equalizer which are interchanged according to a performance criterion.

The effectiveness of this blind scheme is the proposed blind DFE structure [4]. In a conventional DFE, there is a recursive equalizer which suffers from error propagation. Besides, the transversal equalizers are weak for severe channels. This blind equalizer tries to solve these difficulties by breaking the problem into small parts. The partitioning includes four devices: A recursive filter (R), a gain controller (GC), a phase rotator (PR), and a purely transversal filter (T) which has different implementations in literature like all-pass transversal/recursive filter. This choice of transversal filter is derived from the linear minimum MSE equalizer solution by the authors. Also the structure of the equalizer is made adaptive with the criterion of the estimated MSE. When the estimated MSE is higher than a threshold the equalizer stays in its starting mode, and when the estimated MSE is less than a threshold, the equalizer changes its adaptation mechanisms and the order of the above four devices, and acts as a conventional decision directed DFE. The property of this scheme is its reversibility into both modes.

In the starting mode the recursive filter is at the front of the transversal filter, and has the mission of whitening its input [4]. In this mode, the transversal filter removes the remaining ISI with the CMA. In the tracking mode, the equalizer behaves like a conventional decision directed LMS DFE. The performance of the UDFE is equivalent to trained adaptive DFE as shown in [4], in spite of no training sequence.

The authors derive the equalizer's structure from the derivation of linear minimum MSE equalizer. According to the derivation, the equalizer can be divided into two main parts: a recursive stable whitening filter and an anticausal transversal filter. The transversal filter is truncated in its positive powers to obtain a causal filter. And finally, the transversal filter is divided into three parts: GC, PR, and T.

In [4], it is emphasized that the order of the four devices is important in starting mode, that is, in acquisition mode. This fact arises from the mutual dependence of the four devices in their adaptation. So, in starting mode, the order is the GC, R, T, and PR. This order is from the fact that GC and R are dependent, and in order to obtain a suitable R for the tracking mode and an easy adaptation, its output must be normalized to the power of transmitted symbols. So, GC does this job and GC must be in front of R. Since R is a whitening filter, it will make easy the job of T and PR, if it will be placed front. So GC and R are placed in front. T is not affected from the phase error, since it is implemented with CMA and T is next. Finally, to correct the phase error that T can not, PR is placed at last. Because of the adaptation algorithm of this mode, this mode results in a zero forcing equalizer.

When the starting mode operation opens the eye, and the estimated MSE is less than a suitable threshold, the mode of the equalizer is changed to tracking. In the tracking mode, the equalizer behaves like a classical DFE in decision directed mode. In this mode, the order of the devices is GC, T, PR and R. In this mode the equalizer, uses decision directed LMS algorithm for the update of the filters. Since in the previous mode, the equalizer was near to a zero forcing solution, the T is affected from the adaptation criterion, and it will try to approach a MSE solution. But, the R is the whitening filter that is needed for DFE, so it will not change with the mode change.

The UDFE is improved and applied to multiple input case in [5].

3.2.1. Starting Mode

The starting mode's block diagram is given in Figure 3.2.1.1. [4].

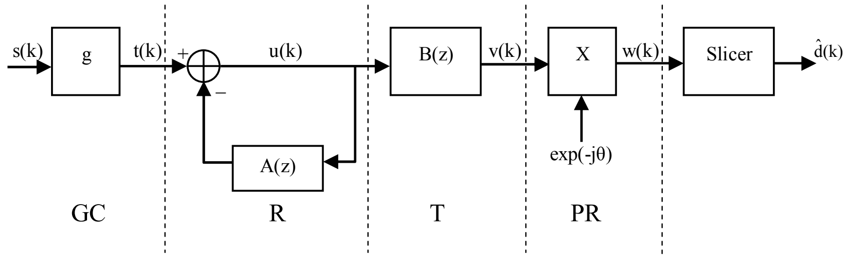


Figure 3.2.1.1. The Block Diagram of the Starting Mode

The starting mode is governed by the following equations:

$$t(k) = gs(k) \tag{3.8}$$

$$u(k) = t(k) - \hat{t}(k) \tag{3.9}$$

where,

$$\hat{t}(k) = \sum_{i=1}^{N} a_i u(k-i) = A^T U_N(k-1) \tag{3.10}$$

$$A = [a_1, ..., a_N]^T \tag{3.11}$$

$$U_N(k-1) = [u(k-1), ..., u(k-N)]^T \tag{3.12}$$

$$v(k) = \sum_{i=0}^{L} b_i u(k-i) = B^T U_{L+1}(k) \tag{3.13}$$

where,

$$B = [b_0, b_1, ..., b_L]^T \tag{3.14}$$

$$U_{L+1}(k) = [u(k), u(k-1), ..., u(k-L)]^T \tag{3.15}$$

$$w(k) = v(k)\exp(-j\theta) \tag{3.16}$$

Finally, the slicer decides on the decisions d(k-δ) with its input w(k).

In the starting mode, the equalizer's four devices have to be adapted according to some criteria. The GC is adapted according to the following equation:

$$E(|u(k)|^2) = \sigma_d^2 \tag{3.17}$$

The filter R will be the whitener of its input, and since according to the upper criterion, its output has constant variance, it will be unique. R will remove ISI by removing the correlation in its input, t(k) and will have the following criterion:

$$I(A) = E(|u(k)|^2) \tag{3.18}$$

which will be minimized by R. T will use one of the blind equalization algorithms. T is using Godard algorithm and minimizes the following cost function:

$$J_G(B) = E\{[|v(k)|^p - R_p]^2\} \tag{3.19}$$

where

$$R_p = \frac{E(|d(k)|^{2p})}{E(|d(k)|^p)} \tag{3.20}$$

Because of the constraint of the adaptation algorithm, the algorithm will converge to a zero forcing solution. The PR uses the following criterion to adapt the phase error:

$$K(\theta) = E\{|v(k)e^{-j\theta(k-1)} - \hat{d}(k)|^2\} \tag{3.21}$$

where the criterion minimizes decision directed MSE. The decision directed approach will approve the being last device of PR, because the other devices may be affected with this criterion.

Finally, the adaptation equations derived from these criteria are given below:

$$G(k) = G(k-1) + \mu_G[1 - |u(k)|^2] \tag{3.22}$$

$$g(k) = \sqrt{|G(k)|} \tag{3.23}$$

where G(0)=1 and μ_G is a suitable step size.

$$t(k) = g(k-1)s(k) \tag{3.24}$$

The stochastic gradient algorithm is used for the adaptation of R:

$$A(k) = A(k-1) + \mu_A u(k) U_N^*(k-1) \tag{3.25}$$

$$u(k) = t(k) - A(k-1)^T U_N(k-1) \tag{3.26}$$

where $A(0) = [0,0,....,0]^T$ and μ_A is a suitable step size.

The stochastic gradient algorithm is used to minimize the blind criterion for T and CMA is used for the adaptation of the T:

$$B(k) = B(k-1) + \mu_B v(k)(R_2 - |v(k)|^2)U^*_{L+1}(k) \tag{3.27}$$

$$v(k) = B^T(k-1)U_{L+1}(k) \tag{3.28}$$

where $B(0) = [0,0,....,0,1,0,....0]^T$ and μ_B is a suitable step size.

Finally, the PR is adapted according to decision directed MSE criterion and with following equations:

$$w(k) = v(k)\exp(-j\theta(k-1)) \tag{3.29}$$

$$\theta(k) = \theta(k-1) + \mu_\theta\left(\varepsilon(k) + \beta\sum_{i=1}^{k}\varepsilon(i)\right) \tag{3.30}$$

$$\varepsilon(k) = \mathrm{Im}\{w(k)[\hat{d}(k) - w(k)]^*\} \tag{3.31}$$

where $\theta(0)=0$, μ_θ is a suitable step size and β is a positive parameter.

3.2.2. Tracking Mode

In starting mode, the blind equalizer reaches a zero forcing solution which has a structure of recursive linear equalizer. When the estimated MSE of the equalizer decreases below the threshold value (open eye), the equalizer changes to tracking mode. In this mode the order of the devices is: GC, T, PR and R. R gets the decisions as input now. This equalizer mode has the structure of a classical decision directed DFE with GC and PR devices added. In this mode, the T will converge to MSE solution instead of zero forcing, so the previous coefficients of the T will change at a small level. R has the same solution form as in the starting mode in this mode. Because of the criterion changed to MSE, the equalizer will result in less MSE.

The tracking mode’s block diagram is given in Figure 3.2.2.1. [4].

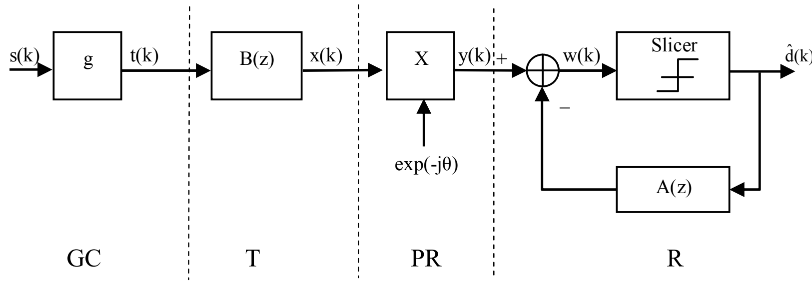


Figure 3.2.2.1. The Block Diagram of the Tracking Mode (Classical DFE Mode)

The performance index, estimated MSE is calculated according to the following equation:

$$M_{DD}(k) = \lambda M_{DD}(k-1) + (1-\lambda)\left|\hat{d}(k) - w(k)\right|^2 \tag{3.32}$$

where λ is the forgetting factor and λ =0.99. This estimated error calculation is decision directed. The criterion to change the mode is that when eye is closed, change to starting mode and when eye is open change to tracking mode:

$M_{DD}(k_0) \geq M_0$: starting mode for $k>k_0$ (3.33)

$M_{DD}(k_0) \leq M_0$: tracking mode for $k>k_0$ (3.34)

where M_0 will be chosen according to the modulation scheme. For 4-QAM case, M_0=0.25.

The adaptation of the devices will be carried out according to the decision directed MSE criterion. The equations of this mode are given below:

$$t(k) = gs(k) \tag{3.35}$$

$$w(k) = y(k) - A^T(k-1)\hat{D}(k) \tag{3.36}$$

$$y(k) = [B^T(k-1)T(k)]\exp(-j\theta(k-1)) \tag{3.37}$$

$$T(k) = [t(k),...., t(k-L)]^T \tag{3.38}$$

$$\hat{D}(k) = [\hat{d}(k-1), \hat{d}(k-2),...., \hat{d}(k-N)]^T \tag{3.39}$$

and the adaptation equations will be:

$$B(k) = B(k-1) + \mu_B[\hat{d}(k) - w(k)]\exp(j\theta(k-1))T^*(k) \tag{3.40}$$

$$A(k) = A(k-1) - \mu_A[\hat{d}(k) - w(k)]\hat{D}^*(k) \tag{3.41}$$

$$\theta(k) = \theta(k-1) + \mu_\theta\left(\varepsilon(k) + \beta\sum_{i=1}^{k}\varepsilon(i)\right) \tag{3.42}$$

$$\varepsilon(k) = \mathrm{Im}\{y(k)[\hat{d}(k) - w(k)]^*\} \tag{3.43}$$

where GC is held constant in this mode. In equation (3.41), the minus sign in the filter update equation originates from the fact that the equalizer output is obtained by the subtraction of the filter R's output.

The estimated error is calculated and tested at each iteration. When the mode is changed to starting mode, the initial values for the devices are:

1) $G(k_0 - 1) = g^2$.

2) R is started with $A(k_0\text{-}1)$ and $U_N(k_0\text{-}1)=0$.

3) T is started with $B(k_0\text{-}1)$ and $U_{L+1}(k_0\text{-}1)=0$.

4) PR is started with $\theta(k_0\text{-}1)$.

CHAPTER 4

SIMULATIONS

The decision feedback equalizers which are given in the past sections have been simulated and a variety of results have been obtained. The simulation platform is the C programming language. In the simulation platform, the communication features are modeled in their general characteristics. In the simulations, the general structure is modeled in baseband complex digital communication. This modeling is suitable for both low-pass representation of both baseband and bandpass systems. The general model of the simulation system is given in Figure 4.1.

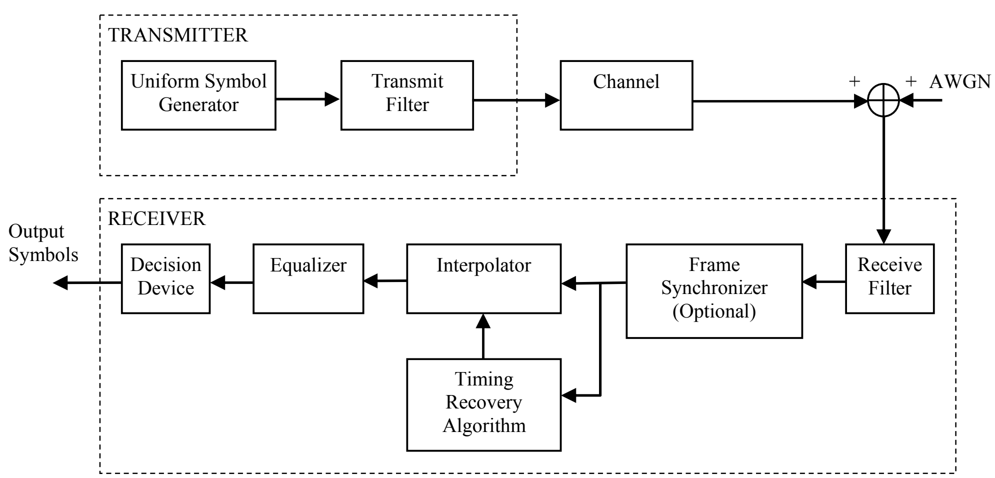


Figure 4.1. The General Model of the Simulation System

In the model, the transmitter is formed from the symbol generator and the transmit filter. The symbol generator produces the discrete sequence of random

BPSK symbols. The BPSK modulation is selected, since BPSK modulation is one of the accepted modulation schemes in the literature. The generated symbols are applied to the transmit filter. The transmit filter is a root-raised-cosine filter and forms the pulse shape of root-raised-cosine pulse shape. The transmit filter produces 8 samples per symbol. So the simulation system works with 8 samples per symbol. Using 8 samples per symbol is preferred to work with algorithms which operate at periods higher than baud rate and to prevent aliasing effects in the receiver.

The rolloff factor for the pulse is 0.5. The rolloff factor used in simulations is always 0.5. Since it is good to have a rolloff factor different than 0 and the most frequently used rolloff factor in the literature is 0.5, in the simulations, the rolloff factor has been equated to 0.5. The length of the filter's impulse response is truncated to the values above the 2% of the maximum value of the pulse.

The produced symbols and filter outputs are processed as blocks. The general working principle of the simulation system is processing per block. All the data signal produced are placed in a block which has a header with length 63 symbols and the data symbols are placed in the remaining part of the block which has 200 symbols length. Each block preserves the state of the previous block. The header of length 63 symbols is a pseudorandom sequence and used as the training sequence for the conventional DFEs. The circular correlation value of the sequence is calculated as follows:

$$\rho(i) = \sum_{k=0}^{62} w_k w_{(k+i) \bmod 63} \qquad 0 \le i \le 62 \tag{4.1}$$

where w_k is the pseudorandom BPSK symbols. $\rho(0)=63$ and $\rho(i)=-1$ for other i.

The blocks produced are applied to the channel filter. The gain control of the Unsupervised (Blind) DFE (UDFE) does the adjustment of the gain at the input of the equalizer to prevent fluctuations in the signal power. The conventional DFEs do not have gain control device, so in order to ease their job, the channel filter coefficients are normalized to unity, always. The simulations are done for 4 channel cases which are symbol interval spaced. In the first two cases, the channel filters are stationary. But, the channel z-transforms have zeros which are very near to the unit circle. So, the channels are severe for the equalizers to recover the symbols. Both channels have 5 taps. The complex impulse responses of the first two cases are given below:

C_1=[2-0.4i   1.5+1.8i   1   1.2-1.3i   0.8+1.6i]   (4.2)

C_2=[0.8264 -0.1653   0.8512   0.1636   0.81]   (4.3)

The zeros of these two stationary channels are given in Figure 4.2.. As seen from the figures both channels are severe.

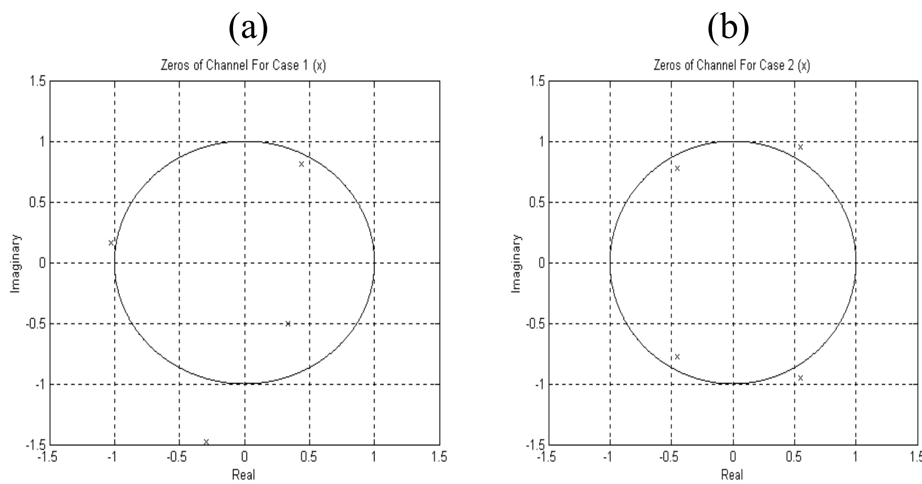


Figure 4.2. The Zeros of Channels. a) Case 1. b) Case 2.

The third channel case is a channel which has a z-transform that has a single zero and after some time another zero which is mobile appears. The second zero is mobile and is half time inside unit circle and half time outside unit circle. This channel is very severe. The channel's second zero corresponds to the classical differential Doppler effect. This effect occurs for the two paths of the channel case 3. When one of the two paths is Doppler corrected, the other path causes to happen a time-varying channel because of the Doppler correction on the first path. The channel's zeros are given below:

$$z_1 = 1.1 \tag{4.4}$$

$$z_2 = \exp(i2\pi/3) + 0.1\exp(i2\pi 10^{-4}(k-2250)) \tag{4.5}$$

The second zero appears after 2250 symbols. The first three channels are the same as the simulated three channels in [4].

The last channel case is the fading channel. The fading channel is Rayleigh frequency selective fading channel. This channel has been formed by a special channel impulse response producing algorithm. In order to implement this channel, 3 taps of fading channel is implemented. The channel taps are suitable to GSM standards for frequency and time intervals (frequency=950 MHz, time interval=271 kbits/s). The simulations for this channel type are done at three velocities of the receiver (v_1=10 km/h, v_2= 30 km/h, v_3=90 km/h).

To the output of the channel, the additive white Gaussian noise (AWGN) is added. With this noise, the signal to noise ratio at the receiver input is calculated according to the following equation:

$$\mathrm{SNR} = \frac{E_b}{\sigma_n^2} \tag{4.6}$$

where E_b is bit energy and with unit variance symbols, E_b is equal to the pulse energy. σ_n^2 is the variance of the noise. SNR is calculated at the channel output (receiver input) and so E_b is updated according to the channel's effect on the bit energy.

Also, in the transmit filter, the timing error is introduced to the transmitted signal, so in the simulations, a timing error is present.

The receive filter is a root-raised-cosine filter with 0.5 rolloff. The filter operates at 8 samples per symbol. So the output of the receive filter has a raised-cosine pulse shape.

The output of the receive filter is given to the frame synchronization block. The frame synchronization block correlates the pseudorandom sequence and the received signal, and tries to find the starting point of the training sequence (header sequence) for each block. In simulations, the frame synchronization is assumed ideal, that is, it is assumed that the starting point of the header is found always correctly. The block is designed to work at 8 samples per symbol.

The interpolator and timing recovery block correct the timing error and produce the new samples with the interpolator. The timing recovery algorithms used are ML-based timing recovery algorithms and Gardner's algorithm. The timing error estimated with these algorithms is given to interpolator which uses linear interpolation, and the new samples are produced.

The equalizers are given the samples which are timing error corrected, and interpolated. The equalizers tested are adaptive DFE and Unsupervised (Blind) DFE. The adaptive DFEs are tested in three ways: in Decision-Directed (DD) mode, Trained Mode and Trained-and-Decision-Directed (TRDD) Mode. In DD DFE, the DFE is trained in the first 6 blocks with the training (header) sequence (6*63=378 Symbols) and in the other blocks it operates in decision directed mode. The Trained DFE has no decision directed mode, it always uses the training (header) sequence to update its filter parameters. So, Trained DFE is continuously trained with the training sequence. The TRDD DFE has both training period and decision directed mode. The TRDD DFE is trained during the training (header) sequence and it is in decision directed mode during the data symbols period. So, TRDD DFE is always adapted with either training sequence or data symbols. The blind DFE is the UDFE that is described in chapter 3. It has no training period, and so it is completely blind.

Finally, the outputs of the equalizer are given to the decision device which is simply a slicer that decides on -1 or +1.

In the sequel, the simulation results are presented. Firstly, the simulation results that evaluate the timing recovery algorithms are presented. And, next, the performances of the adaptive DFE and blind DFE are compared according to some performance criteria.

4.1. Timing Recovery Results

The timing recovery requires frame synchronization and timing recovery algorithm. The frame synchronization in the simulations is provided with ideal frame synchronizer. So, the simulations have no frame errors. In the simulations, mainly two timing recovery algorithms have been incorporated. The first one is Oerder & Meyr algorithm [18] and the second one is Gardner's algorithm [10]. As an ML-based algorithm, the algorithm that is presented in section 2.1.3. is considered as suitable. But, since it requires a different front-end receive filter, instead of it, Oerder & Meyr algorithm has been used.

The algorithms have been tested in two SNR values. These are 16 dB and 8 dB. The two algorithms are applied to two channel cases: case1 and case 4. After 400 symbols, the steady state occurs and the MSE of the timing error (MSET) of the algorithms is calculated as follows:

$$\mathrm{MSET} = \frac{1}{n}\sum_{k=0}^{n-1}\left|\tau(k) - \tau_i\right|^2 \tag{4.7}$$

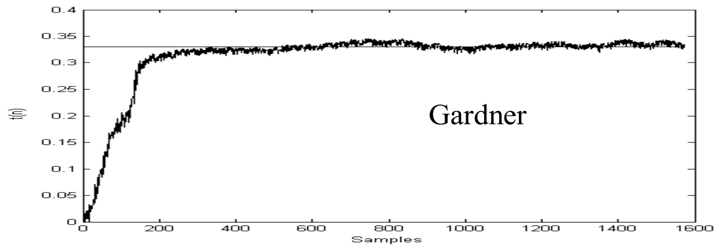
where τ_i is the timing error which is held constant in the simulations and it is 0.33. n is the total number of symbols for which the MSET value is calculated. In the simulations, $n \approx 4500$ symbols. The timing recovery system is not the principal block in the simulations. The main job is to test the equalizer performance in the presence of timing error. So, to test the equalizer performance in the presence of timing error, the timing error is equated to 0.33, which is a random value between 0 and 1.

According to Figures 4.1.1. - 4.1.4., the timing error is estimated very near to the ideal. Gardner's algorithm has some fluctuations on the estimated timing error. Oerder & Meyr algorithm has less fluctuation from the ideal.

The Oerder & Meyr algorithm is fast covergent in channel case1 (about 100 symbols). The algorithm's deviations increase as SNR decreases. In channel case 4, the algorithm's convergence speed decreases to about 400 symbols. And at low SNR in channel case 4, the noise degrades the performance.

The Gardner's algorithm has nearly the same convergence speed in both channel cases. The convergence is reached approximately at 200 symbols. In both channel cases at high SNR, the algorithm's deviations from the ideal are less than the deviations at low SNR. In both channel cases, there is high jitter in the estimated timing error. This jitter is not seen in Oerder & Meyr algorithm.

(a)



(b)

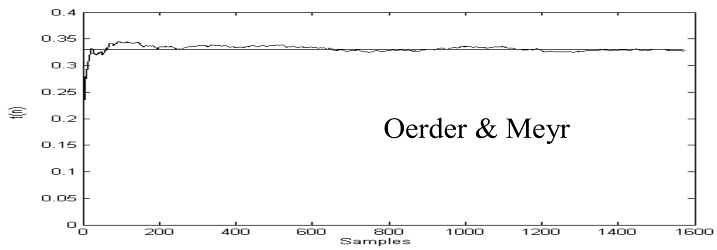


Figure 4.1.1. The Timing Recovery Algorithm Output for Channel Case 1 (SNR=16 dB). a) Gardner's Algorithm. b) Oerder& Meyr Algorithm.

(a)

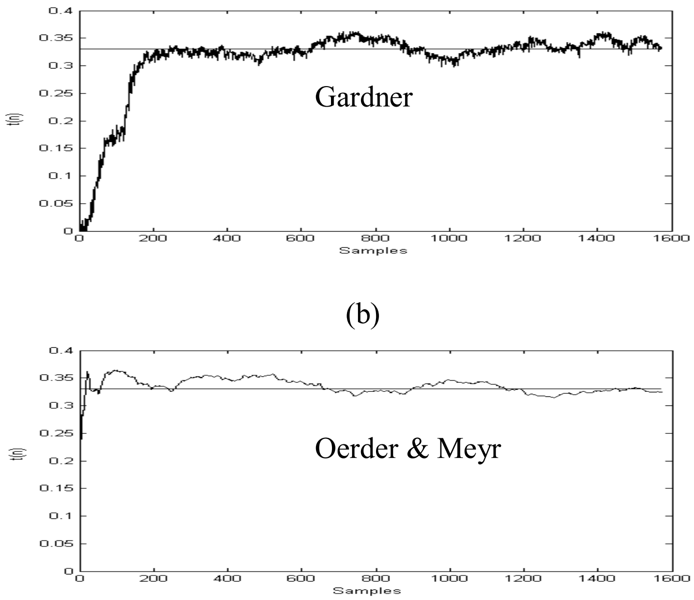


Figure 4.1.2. The Timing Recovery Algorithm Output for Channel Case 1 (SNR=8 dB). a) Gardner's Algorithm. b) Oerder& Meyr Algorithm.

(a)

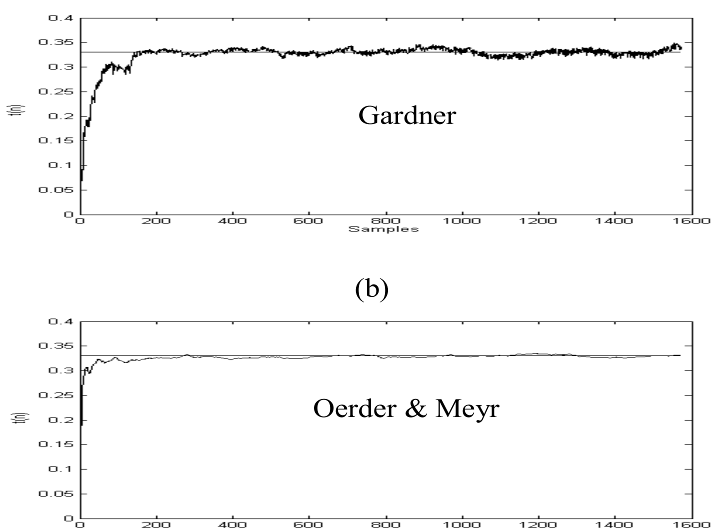
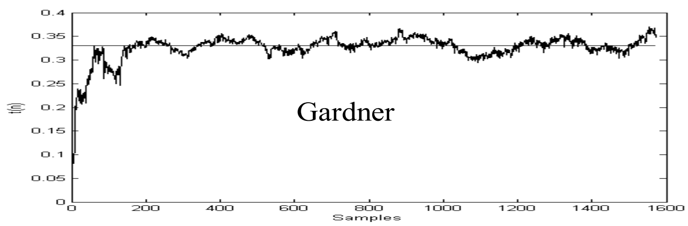


Figure 4.1.3. The Timing Recovery Algorithm Output for Channel Case 4 (SNR=16 dB, Velocity=30 km/h). a) Gardner's Algorithm. b) Oerder& Meyr Algorithm.

(a)



(b)

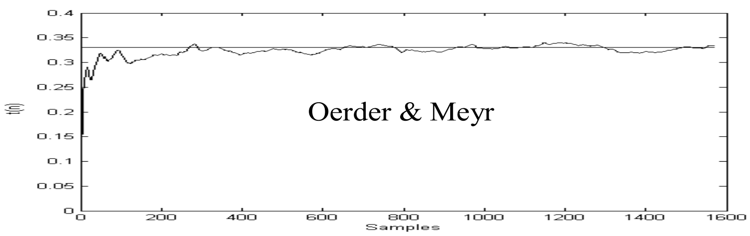


Figure 4.1.4. The Timing Recovery Algorithm Output for Channel Case 4 (SNR=8 dB, Velocity=30 km/h). a) Gardner's Algorithm. b) Oerder& Meyr Algorithm.

The MSET values calculated for the two timing recovery algorithms are given in Table 4.1.1..

Table 4.1.1. MSET Values for the Channel Cases 1 & 4

	Channel Case 1	Channel Case 1	Channel Case 4	Channel Case 4
	SNR=16	SNR=8	SNR=16	SNR=8
Oerder & Meyr	0.0000243	0.0001858	0.0000228	0.000169
Gardner	0.0000826	0.00048	0.0000594	0.000354

According to the table, the Oerder & Meyr algorithm has the lowest MSE values for the timing error. When all the results are compared and the performance measures in the output of the simulations are considered, it is seen that the Oerder & Meyr algorithm has shown better performance than Gardner's algortihm. This evaluation of the timing recovery algorithms is limited to the tests which are done for one trial. Under these conditions, throughout the equalizer simulations, as the timing recovery algorithm, the Oerder & Meyr algorithm has been used.

4.2. Equalizer Simulations

In the simulations, three equalizer types, DD DFE, Trained DFE, TRDD DFE, and the UDFE have been simulated. DD DFE is the classical adaptive DFE that has a 378 symbols length initial training period and after that it is in decision directed mode. Trained DFE is the DFE that is always trained with the header symbol sequence in the blocks. It has no decision directed mode. The TRDD DFE has both training period and decision directed mode. The TRDD DFE is continuously adapted with either the training (header) sequence and or the data symbols. While the training sequence is transmitted, the TRDD DFE is adapted with correct symbols and while the data symbols are transmitted, it is in decision directed mode. The UDFE is the blind DFE algorithm that is described in Section 3.3.. The equalizer performances are compared according to the Mean Square Error (MSE), Bit Error Rate (BER), Residual Intersymbol Interference (RISI) performances and equalizer output diagrams. The MSE tests are done for the first stationary channel case (case 1). The tested channels are two stationary, a time-varying channel, and a fading channel for BER performances as described previously. For RISI measurements, the two stationary channels are simulated. Equalizer outputs are plotted for the channel case 4. The timing error in the equalizer simulations is 0.33. Since the main job is to test the equalizer performance in the presence of timing error, to test the equalizer performance in the presence of timing error, the timing error is equated to 0.33, which is a random value between 0 and 1. The step sizes and parameters for the equalizers are chosen according to the test results of equalizers in the simulations period. The parameters that give the best results in the simulations are chosen as the equalizer parameters. Also, for the UDFE, the suggestions of Labat et al. [4] have been evaluated and affected the parameter choice procedure. The decision directed MSE threshold

(M0) for the UDFE is chosen according to the best results obtained in the simulations. But, for the BPSK modulation scheme, a standard value for the M0 may be chosen between 0.4 and 0.5. This interval depends on the distance of the two symbols (+1 and -1) from each other in the constellation diagram. The offer for the M0 has been extracted from the distance in the constellation diagram.

4.2.1. MSE Results

The MSE Tests have been done for the first stationary channel (case 1). The DFE equalizers DD DFE, Trained DFE, TRDD DFE and UDFE have been tested. All equalizers work at baud rate. The MSE simulations have been done for seeing the convergence speed of the equalizers. The MSE curves have been obtained for 10,000 symbols. The curves show the MSE between the equalizer output and the correct symbol. The curves have been obtained by using 40 different runs and taking the average of them.

In the MSE simulations, the DD DFE, Trained DFE and TRDD DFE used 5 taps for feedforward filter and 5 taps for feedback filter. The UDFE used 10 taps for transversal filter and 5 taps for feedback filter. In the tracking mode, the transversal filter of UDFE has 5 taps of anticausal part, while the classical DFE's have 5 taps of anticausal part. Since the channel has at most 5 taps length, the tap choice is suitable. The step size for DD DFE, Trained DFE and TRDD DFE is 0.02. The parameters used for the UDFE are as follows:

μ_G=0.001

μ_A=0.008 for Starting Mode

μ_B= 0.006 for Starting Mode

μ_A=μ_B= 0.006 for Tracking Mode

μ_θ=0.001

β=0.001

R_2=1

and the forgetting factor, λ for the UDFE is 0.99. The threshold for the estimated decision directed MSE is 0.6. The step sizes are chosen according to the results of the tests. For the UDFE, the suggestions of Labat et al. [4] are considered.

The MSE diagrams are given in Figures 4.2.1.1. - 4.2.1.4. for channel case 1 at a SNR of 20 dB.

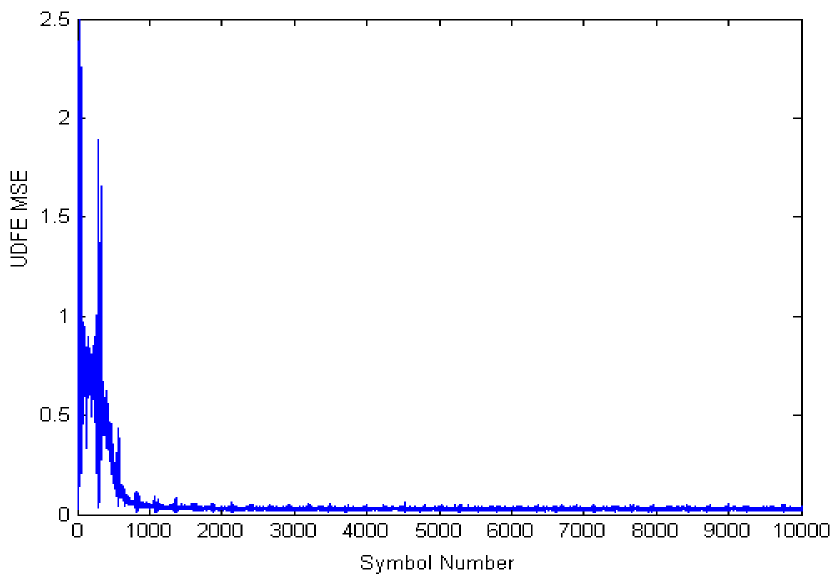


Figure 4.2.1.1. The MSE Diagram for UDFE for Channel Case 1 (SNR=20 dB).

In the Figures 4.2.1.1. - 4.2.1.4., the best equalizer in terms of convergence speed is TRDD DFE. It converges, approximately, after 400 and 450 symbols. The Trained DFE and DD DFE have similar convergence characteristics. They converge about 800 symbols. The UDFE converges after 800 symbols. But it has an initial hardness to reach to the best level because of its starting mode behaviour. But, in general, the UDFE has the same initial period to reach to the steady state as DD DFE and Trained DFE.

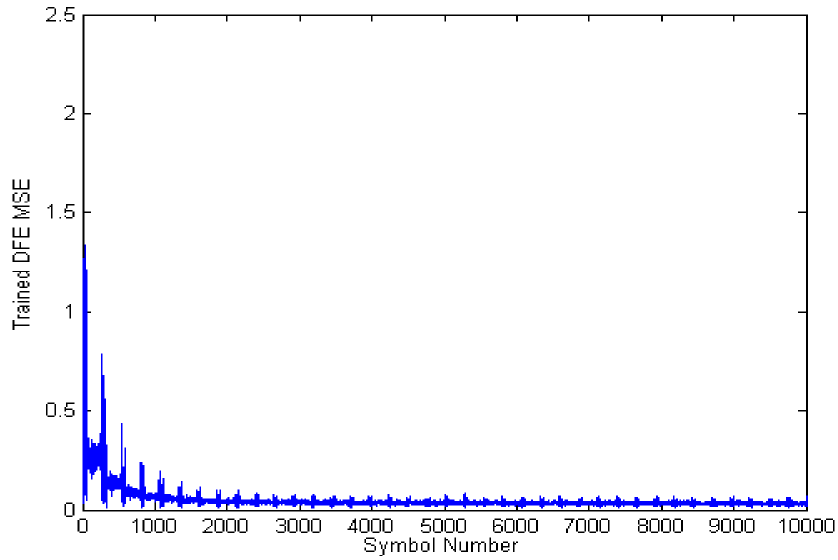


Figure 4.2.1.2. The MSE Diagram for Trained DFE for Channel Case 1 (SNR=20 dB).

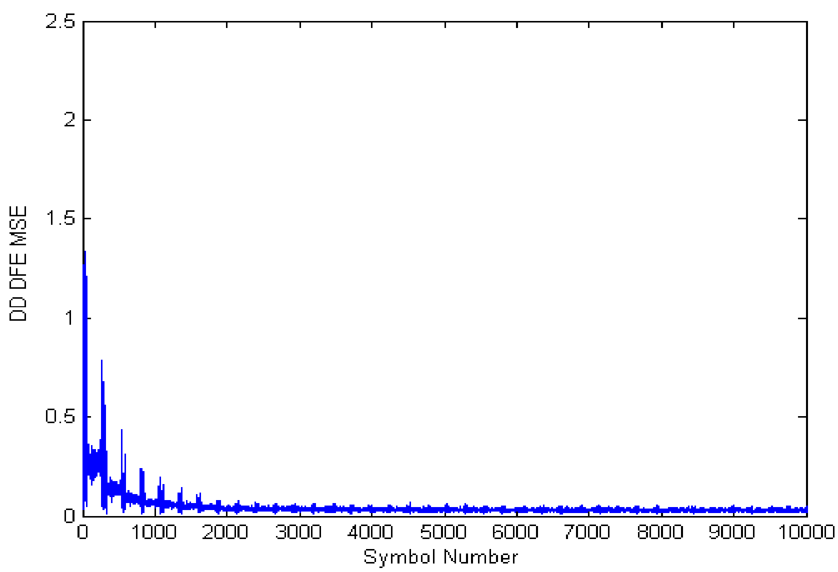


Figure 4.2.1.3. The MSE Diagram for DD DFE for Channel Case 1 (SNR=20 dB).

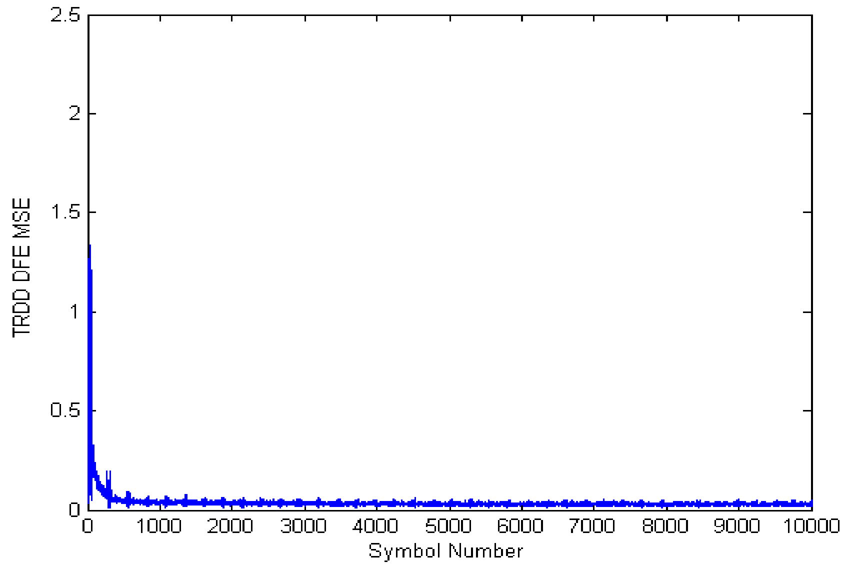


Figure 4.2.1.4. The MSE Diagram for TRDD DFE for Channel Case 1 (SNR=20 dB).

The MSE diagrams are given in Figures 4.2.1.5. - 4.2.1.8. for channel case 1 at a SNR of 10 dB.

In the Figures 4.2.1.5. - 4.2.1.8., the best equalizer in terms of convergence speed is again TRDD DFE. The TRDD DFE reaches steady state after 450 symbols, but, this time, the increasing noise results in higher MSE. The Trained DFE and DD DFE reach the steady state in the same speed as in the previous case. The noise degrades the performance as in TRDD DFE. The UDFE's convergence speed decreases in the high noise case. It reaches steady state after 1000 and 1100 symbols. The increasing noise results in lower performance to pass the starting mode for UDFE.

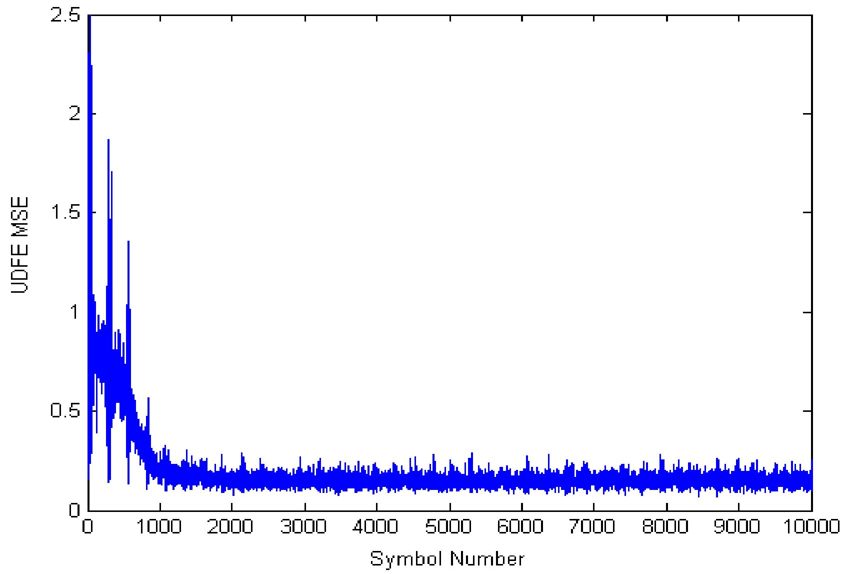


Figure 4.2.1.5. The MSE Diagram for UDFE for Channel Case 1 (SNR=10 dB).

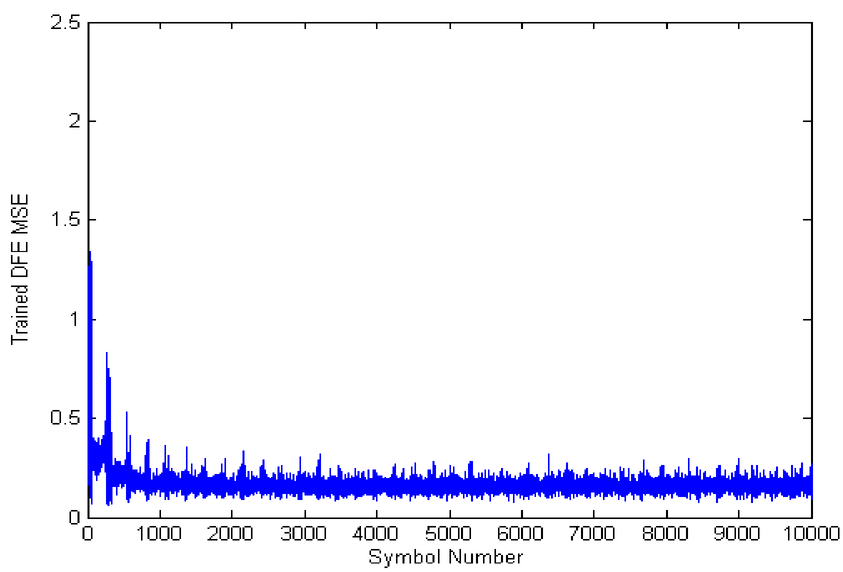


Figure 4.2.1.6. The MSE Diagram for Trained DFE for Channel Case 1 (SNR=10 dB).

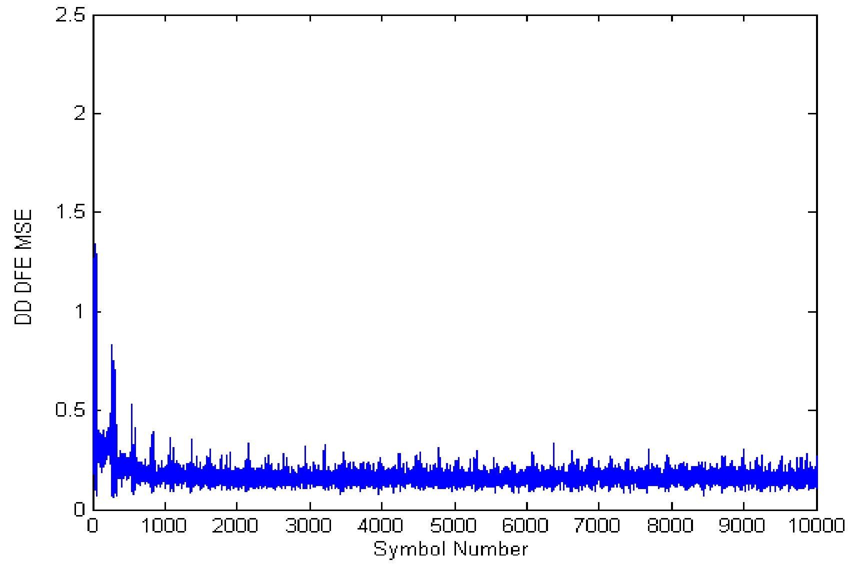


Figure 4.2.1.7. The MSE Diagram for DD DFE for Channel Case 1 (SNR=10 dB).

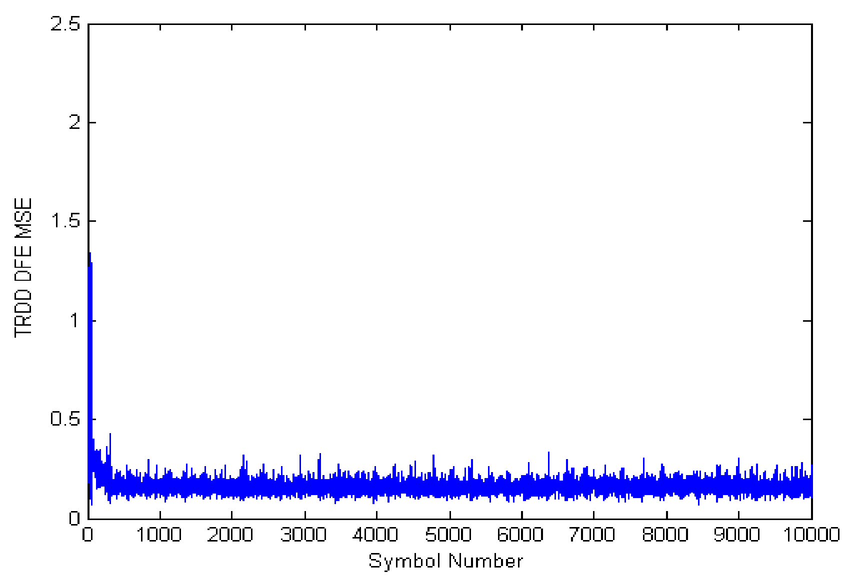


Figure 4.2.1.8. The MSE Diagram for TRDD DFE for Channel Case 1 (SNR=10 dB).

4.2.2. BER Results

The BER tests have been done for the 4 channel cases. The channel cases are two stationary channel cases (case 1 and 2), the time-varying channel (case 3), and the Rayleigh fading channel (case 4). The DFE equalizers DD DFE, Trained DFE, TRDD DFE, and UDFE have been tested. All equalizers work at baud rate. The BER results are obtained for 1,000,000 symbols for the highest SNR case and for 50,000 symbols for the other SNR cases. In channel case 4, at 30 km/h, all simulations are done for 50,000 symbols since BER is high. The BER values are measured with getting the ratio of the total number of erroneous symbols to the total number of symbols. In the BER simulations for channel cases 1,2, and 3, the DD DFE, Trained DFE and TRDD DFE used 5 taps for feedforward filter and 5 taps for feedback filter. The UDFE used 20 taps for transversal filter and 5 taps for feedback filter for the same channel types as in [4]. In the tracking mode, the transversal filter of UDFE has 10 taps of anticausal part, while the classical DFE's has 5 taps of anticausal part. Since the channels have at most 5 taps length, the tap choice is suitable. The step size for DD DFE, Trained DFE and TRDD DFE is between the range {0.001,0.02}. The step size is decreased for low SNR values. The parameters used for the UDFE are as follows:

μ_G=0.001

μ_A=μ_B= (0.002,0.001) for Starting Mode

μ_A=μ_B= (0.006,0.001) for Tracking Mode

μ_θ=0.001

β=0.001

R_2=1

and the forgetting factor, λ for the UDFE is 0.99. The threshold for the estimated decision directed MSE is 0.6 for channel case 1, 0.38 for the channel case 2, and 0.6 for channel case 3.

For channel case 4, the DD DFE, Trained DFE and TRDD DFE have the same number of taps, but UDFE has 10 taps of transversal filter and 5 taps of feedback filter. In the tracking mode, the equivalent channel for UDFE and the classical DFE's has a close number of taps of anticausal and causal parts. Since the channels have 3 taps length, the tap choice is suitable. The step size for DD DFE, Trained DFE and TRDD DFE is between the range {0.01,0.04}. The step size is decreased for low SNR values. The parameters used for the UDFE are as follows:

$\mu_G=0.001$

$\mu_A=\mu_B=(0.001)$ for Starting Mode

$\mu_A=\mu_B=(0.008,0.04)$ for Tracking Mode

$\mu_\theta=0.001$

$\beta=0.001$

$R_2=1$

and the forgetting factor, λ for the UDFE is 0.995. The threshold for channel case 4 is 0.6 for velocity of 10 km/h, 30 km/h and 0.25 for velocity of 90 km/h.

The BER results are given in Figures 4.2.2.1. - 4.2.2.5..

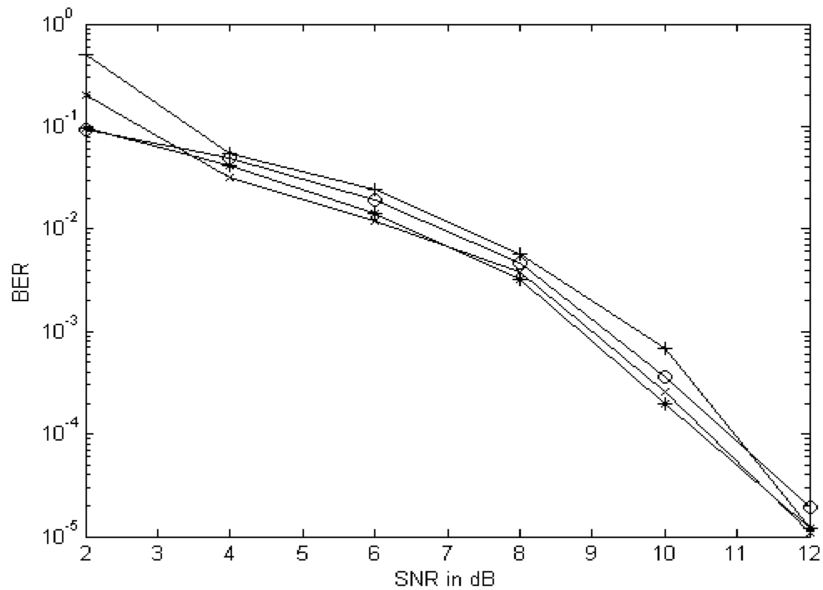


Figure 4.2.2.1. The BER Diagram for Channel Case 1. (x : UDFE, o : Trained DFE, + : DD DFE, *: TRDD DFE)

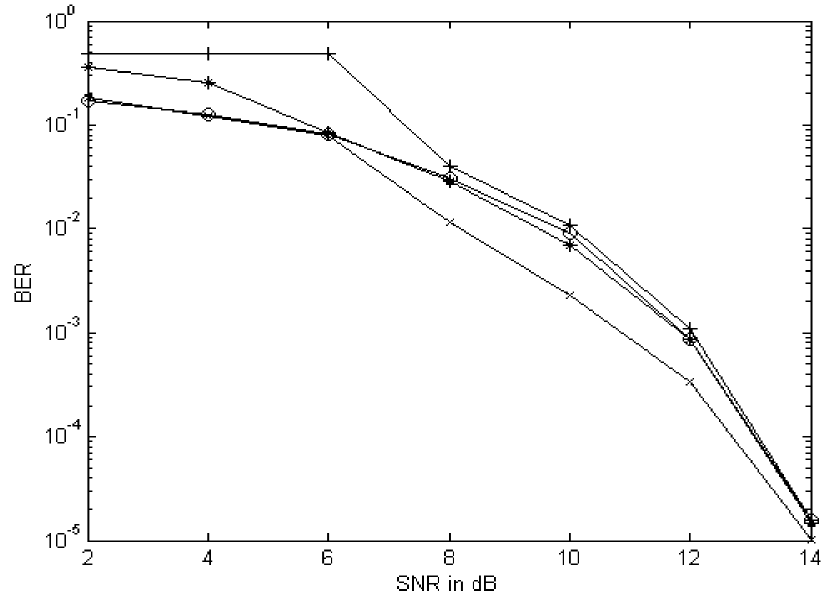


Figure 4.2.2.2. The BER Diagram for Channel Case 2. (x : UDFE, o : Trained DFE, + : DD DFE, *: TRDD DFE)

By examining Figures 4.2.2.1. and 4.2.2.2., for channel cases 1 and 2, the BER performances show that all the equalizers have nearly the same performance. Only at low SNR's, the DD DFE and UDFE can not converge, and Trained DFE and TRDD DFE has high BER. But, in general, the performance is nearly the same for high and moderate SNRs except for the slightly better performance of UDFE in case 2.

By examining Figure 4.2.2.3., for 3rd channel case, the channel has a mobile zero which appears after some time. This change in the channel results in that the DD DFE can not open the eye, and it results in divergence of the equalizer with the error propagation. The UDFE returns to the starting mode and detects the change in the channel, and after opening the eye, it passes to the tracking mode. The Trained DFE, TRDD DFE and UDFE have nearly the same performance, but UDFE has a slightly better performance at high SNR.

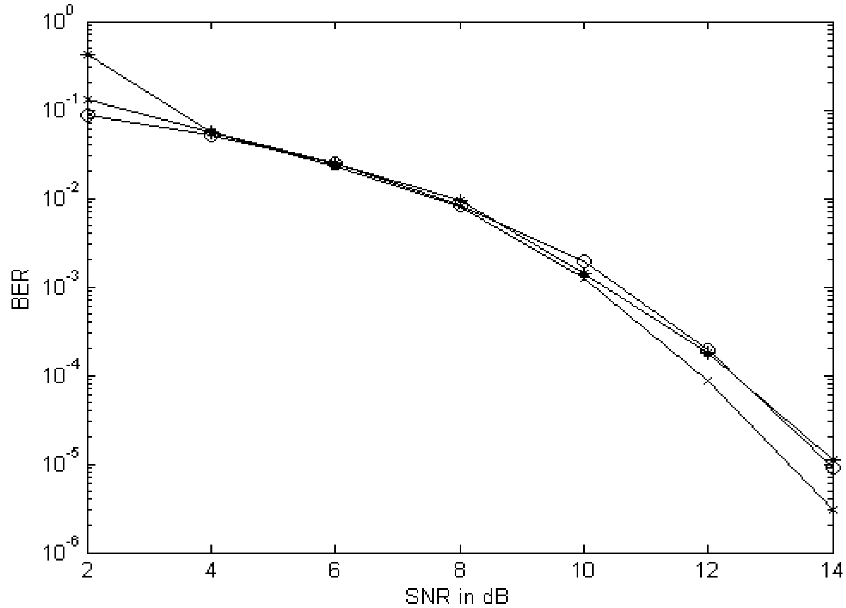


Figure 4.2.2.3. The BER Diagram for Channel Case 3. (x : UDFE, o : Trained DFE, *: TRDD DFE)

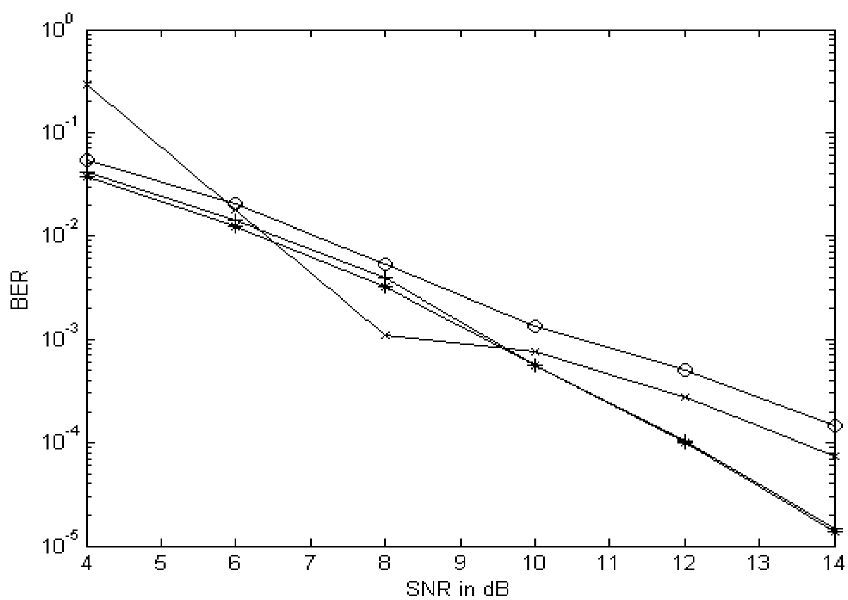


Figure 4.2.2.4. The BER Diagram for Channel Case 4 (Velocity=10 km/h). (x : UDFE, o : Trained DFE, + : DD DFE, *: TRDD DFE)

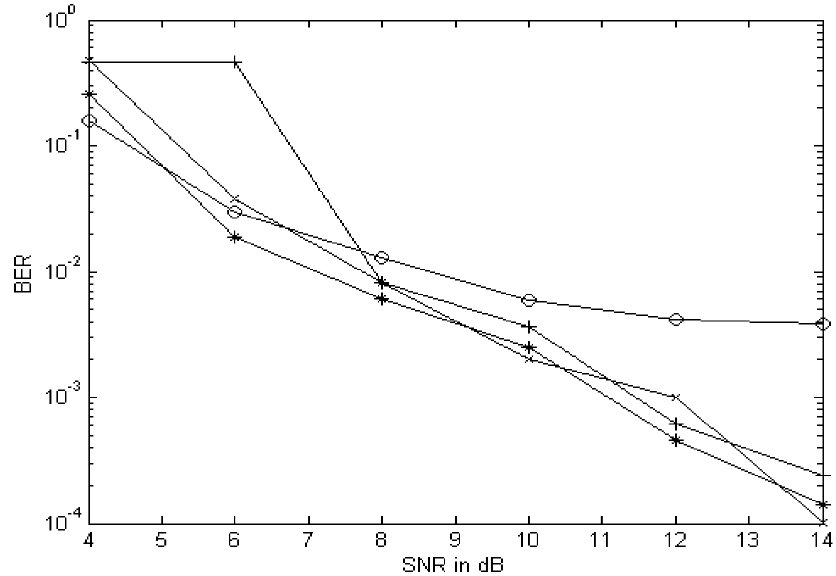


Figure 4.2.2.5. The BER Diagram for Channel Case 4 (Velocity=30 km/h).
(x : UDFE, o : Trained DFE, + : DD DFE, *: TRDD DFE)

For channel case 4, the BER diagrams show that, the DD DFE, TRDD DFE and UDFE have better performance. DD DFE and TRDD DFE are better for 10 km/h and have similar performance, but UDFE is better for 30 km/h at the highest SNR. The UDFE can not pass the starting mode at 4 dB SNR. The DD DFE can not converge at SNR values lower than 8 dB at 30 km/h. The Trained DFE preserves a standard performance with its training period advantage, but BER can not fall below a constant rate since it has no decision directed mode. After 30 km/h, the equalizers' performances are degraded. As it is seen, in BER diagram for 30 km/h, the performances of the DD DFE and UDFE have different characteristics in different SNR's. The TRDD DFE is affected with the increasing speed. The Trained DFE preserves a constant performance, but its BER is increasing with the tracking capability with the training sequence is becoming insufficient. The step size choice of equalizers carries high importance in fading channels, because the step size must be large enough to track the channel variation, and small enough to prevent high BER.

The initial BER of the UDFE is 2 or 3 times of that of the DD DFE, Trained DFE and TRDD DFE at the same SNR values according to the observations of the simulations. In strongly severe cases (low SNR, time-varying channel), the initial BER increases. So, since UDFE has no training period, this results in the fact that the blind algorithm which opens the eye in the starting period has a longer acquisition period for these cases based on the observations. For UDFE, the choice of step size parameters and the configuration decision parameter (M0) has an importance as experienced in the simulations. When the step size parameters are chosen large, it may last shorter to reach the tracking mode, but there is the risk of not passing the starting mode in severe conditions. When the step sizes are chosen smaller, it may last longer to reach tracking mode. The choice of M0 is important, too. During the simulations, it is observed that, when M0 is chosen small, the starting period lasts longer and the advantage of tracking mode to yield lower BER is damaged. But, this choice results in faster detection of eye closing and returning to the starting mode to reopen the eye. When, M0 is chosen large, the advantage of tracking mode appears, and the initial and steady state performances can be better in suitable conditions.

Also, the BER of UDFE in starting mode is higher according to the tracking mode in all channel cases. When the UDFE returns to its starting mode, the total BER increases because some parameters of the UDFE must be initialized to the values given in Section 3.2.2. and this initialization requires some time to reach to the steady state. The blind operation of the UDFE in starting mode is also the cause of the high BER in starting mode.

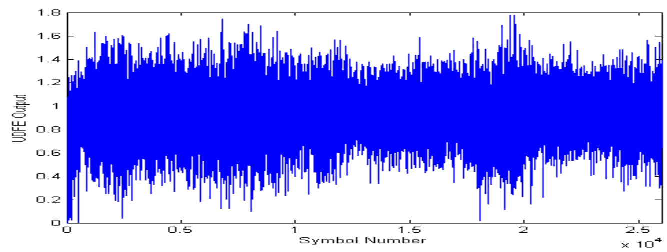
For channel case 4, for the velocities higher than 50 km/h, the equalizers can not track the channel well enough. The DD DFE diverges after 50 km/h. The Trained DFE, TRDD DFE and UDFE yield high BER in these cases. But, UDFE returns to its starting mode when the eye gets closed to reopen the eye.

The equalizer output diagrams have been obtained by taking the absolute value of the real part of the equalizer outputs and drawing the results against symbol numbers. The equalizer outputs for the channel case 4 are plotted in Figures 4.2.2.6. - 4.2.2.8..

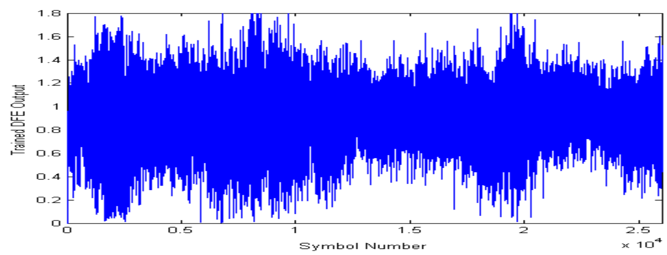
In all the equalizer output diagrams, the fading characteristics of the channel are observable. At some periods of the simulations, the eye gets closed, then, it is opened with the change in the channel impulse response. The vertical lines in the diagrams correspond to these cases. The vertical lines extend away from unity when the equalizer can not track the changes in the channel impulse response well enough. When the equalizers track the channel changes well, the equalizer output diagrams exhibit shorter lines and the ends of the lines are closer to unity.

The equalizer outputs show that at 30 km/h (Figure 4.2.2.6.), the UDFE, DD DFE and TRDD DFE have better performance in opening the eye. The eye is nearly open with some small exceptions. The Trained DFE opens the eye in general, but sometimes the eye gets closed with the absence of decision directed mode. At 90 km/h (Figure 4.2.2.7.), the equalizers have not good performance and eye can not be opened properly. But, the TRDD DFE, UDFE and Trained DFE have successful tries to open the eye, but in general, the high fading characteristics of the channel results in the closing of the eye. The TRDD DFE is the best. The DD DFE is unsuccessful in opening the eye after 2500 symbols, since it has no training period instead of the initial period. But UDFE uses its starting mode to reopen the eye, and after opening the eye, it passes to the tracking mode. When the eye gets closed again, then it returns back to the starting mode. In Figure 4.2.2.8., the equalizer outputs have been obtained for infinite SNR case. In this case, the TRDD DFE, UDFE and DD DFE have similar and better performance according to the Trained DFE. The Trained DFE has some difficulties in tracking the channel since it has no decision directed mode. The infinite SNR case is more smooth according to other cases and the eye is more open. This shows how increasing

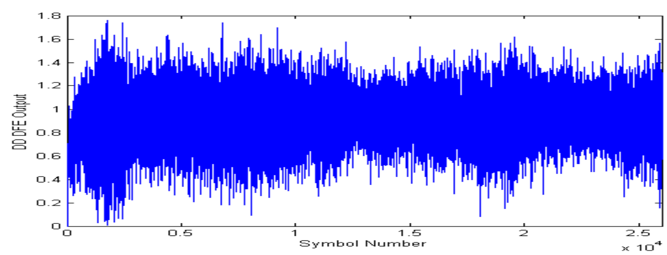
(a)



(b)



(c)



(d)

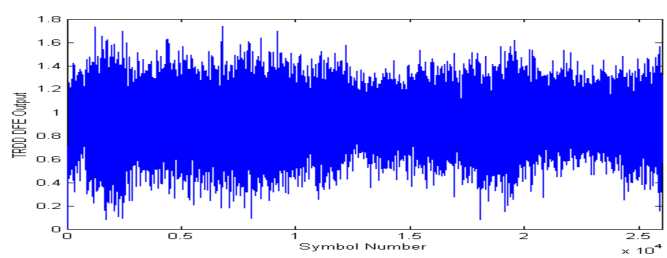
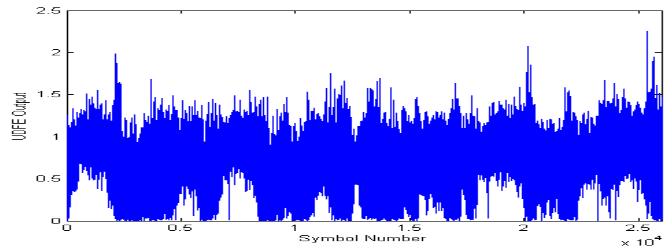
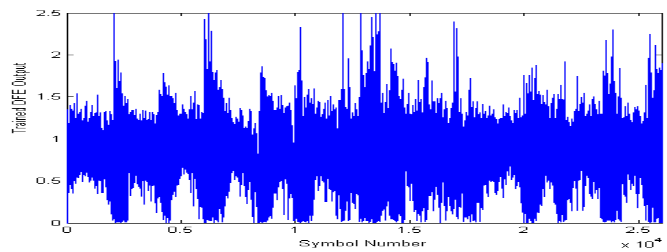


Figure 4.2.2.6. The Equalizer Output for Channel Case 4 (SNR=14 dB, Velocity=30 km/h). a) UDFE. b) Trained DFE. c) DD DFE. d) TRDD DFE.

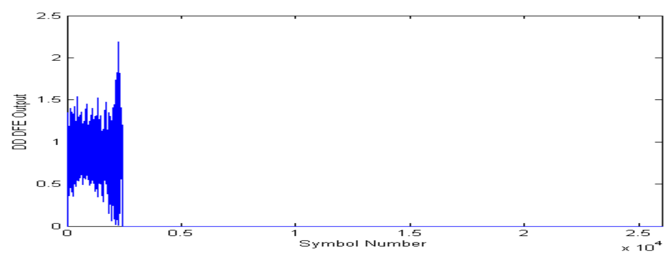
(a)



(b)



(c)



(d)

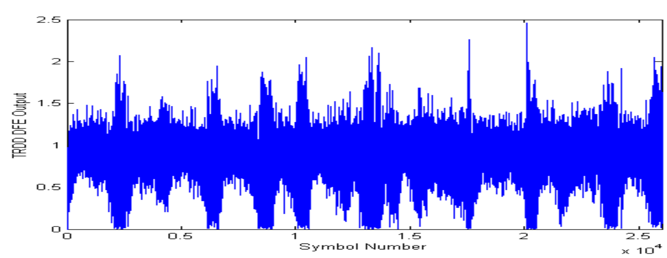
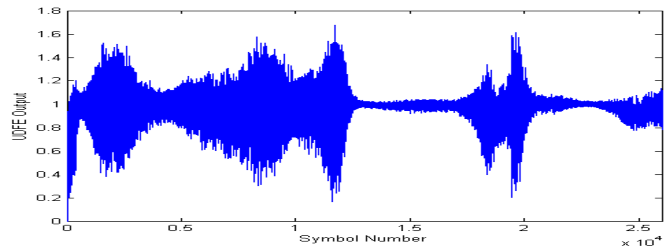


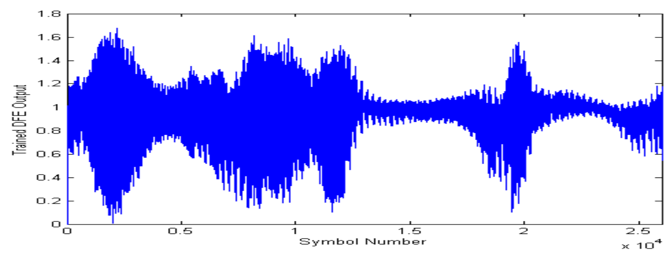
Figure 4.2.2.7. The Equalizer Output for Channel Case 4 (SNR=14 dB, Velocity=90 km/h). a) UDFE. b) Trained DFE. c) DD DFE. d) TRDD DFE.

noise distorts the performance. The best in opening the eye is TRDD DFE when all cases are considered.

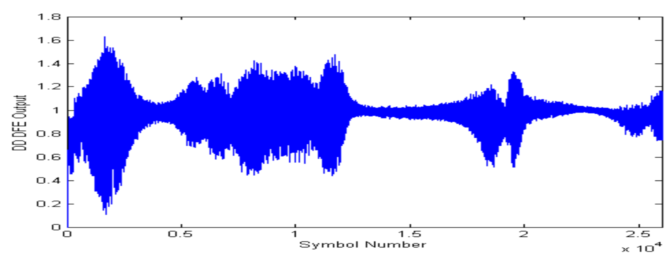
(a)



(b)



(c)



(d)

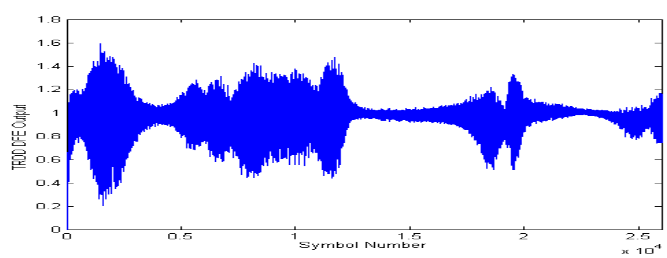


Figure 4.2.2.8. The Equalizer Output for Channel Case 4 (SNR=∞ dB, Velocity=30 km/h). a) UDFE. b) Trained DFE. c) DD DFE. d) TRDD DFE.

4.2.3. RISI Results

The RISI for the simulations are calculated with the channels of case 1 and case 2. In order to calculate RISI, the equivalent channel is found. The adaptive DFE and the UDFE in its tracking mode consist of two main filters: the feedforward filter (transversal filter+GC+PR for UDFE) and feedback filter. At the output of the equalizer, an equivalent pulse shape will appear. The ideal form for this pulse shape is the δ function. In order to form this pulse shape, the feedforward filter will eliminate the precursor ISI (samples of the pulse response at the input of the feedforward filter before the 0^{th} time instance). In order to form the ideal equivalent pulse shape, the postcursor ISI (samples of the pulse response after the 0^{th} time instance) must be canceled. In a DFE, these postcursor ISI values will correspond to the feedback filter's coefficients [2]. So, the equivalent channel will be obtained by subtracting the feedback filter coefficients from the pulse response at the output of the feedforward filter. If we call the taps of the equivalent channel q(k), the equivalent channel will be calculated as follows:

$$q(k) = \sum_{i} c(k-i)f(i) - d(k) \tag{4.8}$$

where c(k) is the channel impulse response, f(k) and d(k) are the feedforward and feedback filters of the classical and blind DFE's, respectively. For the UDFE the feedforward filter is obtained according to the following equation:

$$f(k) = g \times b(k) \times e^{-j\theta} \tag{4.9}$$

where b(k) is the transversal filter of the UDFE and the g and θ are the gain control and phase error components of the UDFE, repectively. In the RISI calculations, the adaptive DD DFE, Trained DFE and TRDD DFE have 5 taps for both of the feedforward and feedback filters. The UDFE has 10 taps for the transversal filter and 5 taps for the feedback filters. In the resultant scheme, the UDFE has 4 taps which are strictly anticausal in the transversal filter as the classical DFE's feedforward filter. The step size for the classical DFE is 0.01. The parameters used for the UDFE are as follows:

μ_G=0.001

μ_A=0.002,0.001

μ_B=0.002,0.001

μ_θ=0.001

β=0.001

R_2=1

and the forgetting factor, λ for the UDFE is 0.99. The threshold for the estimated decision directed MSE is 0.6 for channel case 1 and 0.38 for the channel case 2.

The equivalent channels at SNR of 18 dB for the four simulated DFE's and for channel case 1 are given in Figure 4.2.3.1. The results for the channel case 2 are similar to the channel case 1 results. As seen from the figures, the equivalent channel for all of the DFE's are near to δ(k-5) function, which means simply the removing of all ISI.

The RISI values for the equivalent channels obtained for the channel cases 1 and 2 are calculated according to the following equation:

$$\mathrm{RISI} = \frac{\sum_i |q(i)|^2 - \max_k |q(k)|^2}{\sum_i |q(i)|^2} \tag{4.10}$$

The RISI figures showing the RISI versus SNR for the classical DFE's and UDFE are shown in Figures 4.2.3.2. and 4.2.3.3.. The RISI values are calculated after 15000 symbols.

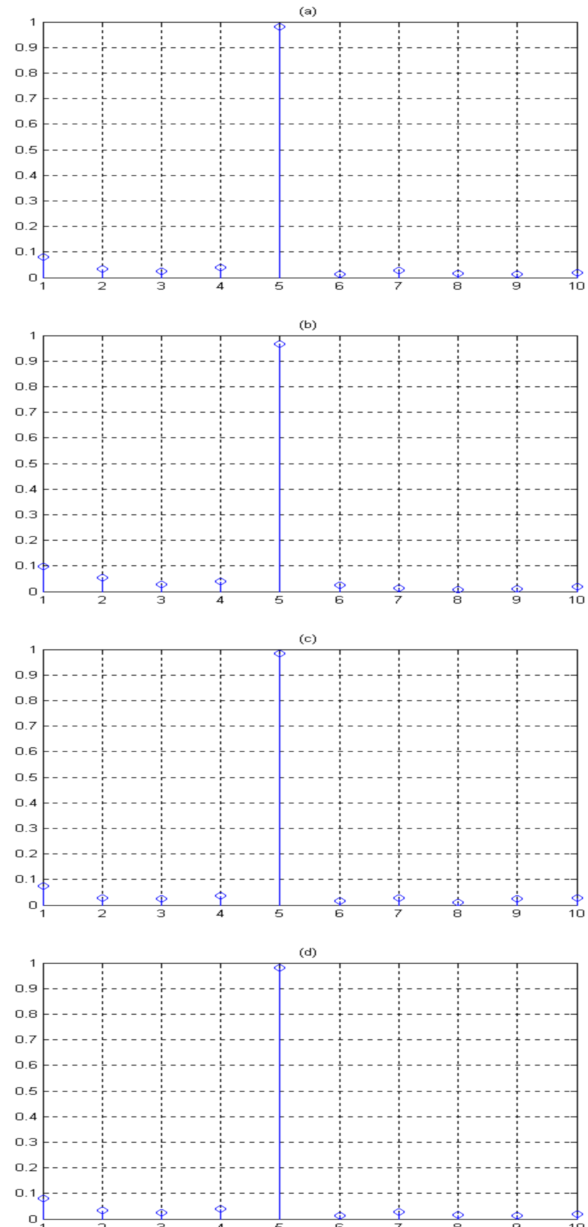


Figure 4.2.3.1. The Equivalent Channels for Channel Case 1 (SNR=18 dB). a) DD DFE. b) Trained DFE. c) UDFE. d) TRDD DFE

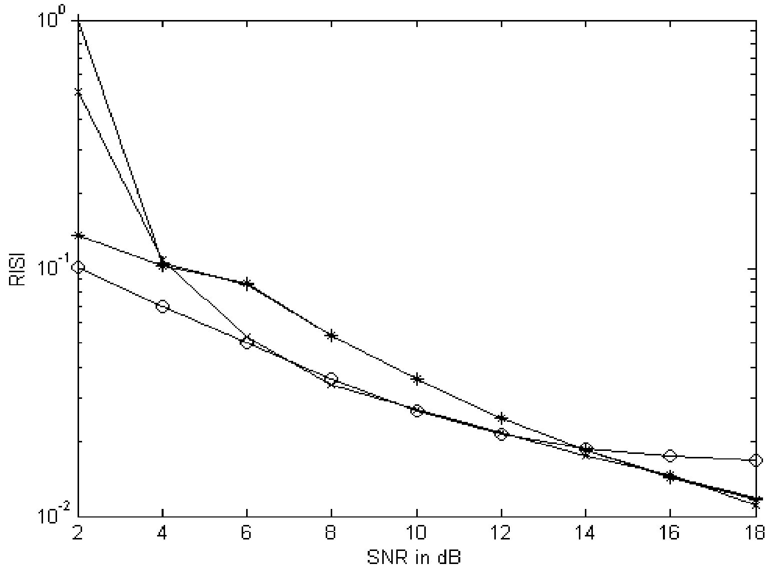


Figure 4.2.3.2. The RISI Diagram for Channel Case 1. (x : UDFE, o : Trained DFE, + : DD DFE, *: TRDD DFE)

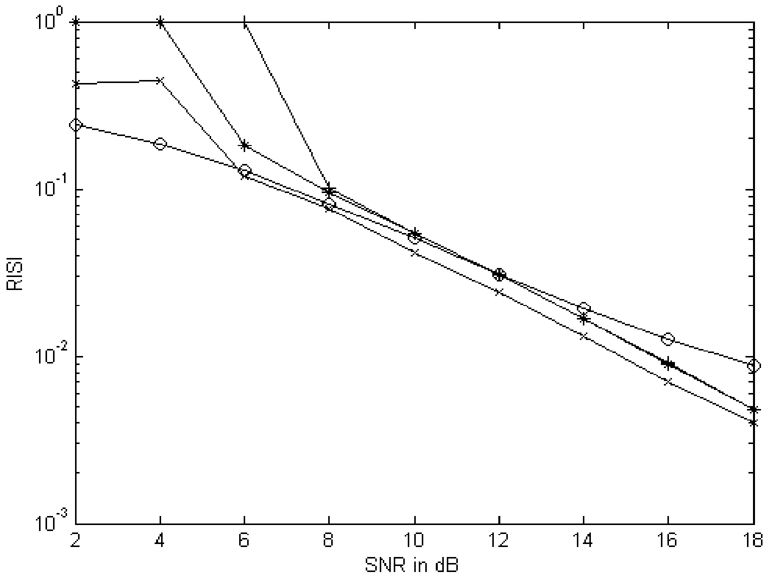


Figure 4.2.3.3. The RISI Diagram for Channel Case 2. (x : UDFE, o : Trained DFE, + : DD DFE, *: TRDD DFE)

In Figures 4.2.3.2. and 4.2.3.3. it is seen that the equalizers have nearly the same performance. For channel case 1, in all cases for high and moderate SNR (SNR>6 dB), the UDFE has the best performance. For low SNR, the UDFE, TRDD DFE and DD DFE have worse performance according to Trained DFE. The UDFE can not pass its starting mode, and it has high RISI for SNR less than 6 dB. The Trained DFE preserves a constant performance with the advantage of its training period. The TRDD DFE has the performance between DD DFE and Trained DFE at low SNR. At high SNR, TRDD DFE achieves the better performance with decision directed mode.

For channel case 2, in the high SNR case, all equalizers have nearly the same performance. In this channel case, it is seen that the noise affects the performance at a higher rate according to channel case 1. The reason of the severity of the channel case 2, results in worse performance at low SNR. UDFE shows better performance for high SNR. The Trained DFE preserves a limited performance, but its performance is increasing with the benefit of the training period to some extent for high SNR. The TRDD DFE and DD DFE have nearly the same performance to UDFE. In the low SNR case for channel 2, the DD DFE can not resist against noise for lower than 8 dB case. For SNR values lower than 6 dB, the TRDD DFE shows bad performance. Also, the UDFE can not pass its starting mode, and it has high RISI for low SNR.The trained DFE does not have as much RISI as the other two because of the training period advantage which protects it to go much away from the optimum.

The RISI diagrams for channel case 1 for DD DFE, TRDD DFE and UDFE have better performance than channel case 2 at low SNR. It shows that the severity of the channel case 2 results in a performance degradation in RISI diagrams according to case 1.

In all cases, the gain control (GC) and phase rotator (PR) used in the UDFE resulted in a good performance of the UDFE with the other devices as seen from the simulation results. The choice of small values for the step sizes (≈0.001) of the

GC and PR results in more stable performance. The phase ambiguity present in the CMA is resolved as seen from the simulations, because the decisions are correct.

CHAPTER 5

CONCLUSIONS

The results of all tests show that, in most of the situations, the UDFE has nearly the same or equal performance to the trained DFEs or sometimes has better performance than others. The general performance of the UDFE is good for time-varying channels which are slowly varying. With some improvements on the structure that will speed up the adaptation, the UDFE is the candidate equalizer structure that is suitable for most of the stationary channel types and slowly time-varying channels to eliminate the training period.

In the thesis, the joint structure of timing recovery and equalization has been modeled and tested. The two parts of the structure are implemented as single blocks of system. The first part is the timing recovery section. The next part is the equalizer block. In order to test the system, the timing error is introduced to the system. Like a packet switching communication, the simulation of the transmission is modeled in frame structure. The frames are formed with independent and identically distributed data and with a header of fixed length known sequence.

The frame synchronization method that is implemented is ideal frame synchronizer. The timing recovery algorithms, Oerder & Meyr and Gardner algorithms are tested in the simulations. Both methods have good results. The algorithms are successful in both stationary and time varying channels. But, Oerder & Meyr algorithm has been preferred in the tests because of the less error it introduced under the limitations of the tests.

The equalizers tested are DD DFE, Trained DFE, TRDD DFE and UDFE which is completely blind in operation. The equalizers have different performances in different performance aspects.

The BER tests have been done for 4 complex channel environments. In stationary channels, the equalizers have nearly the same performance. Although the DD DFE, Trained DFE and TRDD DFE perform initial acquisition of the channel with a training period, the UDFE does this job with its blind algorithm and results in the same steady state error.

In the third channel case, the channel has been changed to a time-varying channel after some period. This change resulted in an unsuccessful condition for the DD DFE which lost its correct state and could not recover the change in the channel, since no training period is possible for it. The Trained DFE, TRDD DFE and UDFE have been able to recover this change in the channel. The Trained DFE and TRDD DFE utilized the training sequence and the UDFE returned to its starting mode to achieve the job. The UDFE later turned to the tracking mode after opening the eye. The resulting BER performances of the three successful methods are equivalent.

In the last channel case, three taps Rayleigh fading channel was the environment. The DD DFE, TRDD DFE and UDFE have better performances in low velocities, but in high velocities after 50 km/h, the DD DFE was not successful. After 50 km/h, the UDFE utilized its starting mode and tried to reopen the eye. At 90 km/h, the DD DFE has been completely lost, but UDFE and Trained DFE have nearly the same performances in opening the eye structure. In opening the eye in fading channels, the TRDD DFE has the best performance.

In the RISI calculations, two severe stationary channels same as in the BER tests, have been tested. In both situations and at high and moderate SNR values, the UDFE and other equalizers have nearly the same performance. The equivalent channel resulted from the equalization shows that the UDFE is enough for these channel environments.

When the MSE results are considered, the UDFE has a relatively longer initial acquisition period according to the trained DFEs at low SNRs. But, in high SNR, this period decreases, and the performance of the UDFE comes closer to the performance of the trained DFEs with the point of view of initial acquisiton. Also, the choice of the parameters of the UDFE must be handled carefully. For example, the parameter choice may result in a smooth mode change of UDFE, especially, from starting mode to tracking mode. The correct choice results in a high performance improvement. Lastly, the starting mode can not be passed at the lowest SNR values, which degrades the performance. But, this problem is present in the trained DFEs, too. The advantage of the trained DFEs is their relatively less parameters in the adaptation. But, this has a cost of requiring the training period in severe conditions like initial acquisiton of the channel, sudden channel changes, and time-varying environments.

REFERENCES

[1] John G. Proakis, *Digital Communications*, McGraw-Hill Companies, Inc., International Edition, 2001.

[2] Edward A. Lee, David G. Messerschmitt, *Digital Communication*, Kluwer Academic Publishers, Boston, 1988.

[3] Simon Haykin, *Adaptive Filter Theory*, Prentice-Hall, Inc., 1996.

[4] Joel Labat, Odile Macchi, Christophe Laot, "Adaptive Decision Feedback Equalization: Can You Skip the Training Period?", IEEE Transactions on Communications, vol. 46, no. 7, pp. 921-930, July 1998.

[5] Joel Labat, Christophe Laot, "Blind Adaptive Multiple-Input Decision Feedback Equalizer with a Self-Optimized Configuration", IEEE Transactions on Communications, vol. 49, no. 4, April 2001.

[6] J. Labat, C. Laot, O. Macchi, "Dispositif dégalization adaptatif pour systémes de communications numériques" French Patent 9510832, September 15, 1995.

[7] J. Labat, O. Macchi, C. Laot, N. Lesquin, "Is training of adaptive equalizers still useful?", Proc. Globecom, vol. 2, London, U.K., pp. 968-972, 1996.

[8] Umberto Mengali, Aldo N. D'Andrea, "Synchronization Techniques for Digital Receivers", Plenum Press, 1997.

[9] I. Glover, P. Grant, *Digital Communications*, Prentice Hall Europe, 1998.

[10] Floyd M. Gardner, "A BPSK/QPSK Timing-Error Detector for Sampled Receivers", IEEE Transactions on Communications, vol. COM-34, no. 5, pp. 423-429, May 1986.

[11] Kurt H. Mueller, Markus Müller, "Timing Recovery in Digital Synchronous Data Receivers", IEEE Transactions on Communications, vol. COM-24, no. 5, pp. 516-531, May 1976.

[12] Simon Haykin, *Communication Systems*, John Wiley & Sons, Inc., 1994.

[13] M. Mostofa K. Howlader, Brian D. Woerner, "Frame Synchronization of Convolutionally Coded Systems for Packet Transmission", IEEE Communications Letters, vol. 5, no. 7, pp. 307-309., July 2001.

[14] D. N. Godard, "Self-Recovering Equalization and Carrier Tracking in a Two-Dimensional Data Communication System", IEEE Transactions on Communications, COM-28, pp. 1867-1875, 1980.

[15] M. Mostofa K. Howlader, Brian D. Woerner, "Decoder-Assisted Frame Synchronization for Packet Transmission", IEEE Journal on Selected Areas in Communications, vol. 19, no. 12, pp. 2331-2345, December 2001.

[16] Alan V. Oppenheim, Ronald W. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, Inc., 1999.

[17] Charles W. Therrien, *Discrete Random Signal and Statistical Signal Processing*, Prentice-Hall, Inc., 1992.

[18] M.Oerder and H.Meyr, "Digital Filter and Square Timing Recovery", IEEE Transactions on Communications, COM-36, pp. 605-611, May 1988.

[19] M. E. Austin, "Decision Feedback Equalization for Digital Communication Over Dispersive Channels", M.I.T. Research Laboratory of Electronics Technical Report 461, August 1967.

[20] Esra Kurban, "Blind Adaptive Decision Feedback Equalization", Master Thesis, Middle East Technical University, September 2002.

[21] R. McEliece, R. Palanki, "Intersymbol Interference in Pulse-Amplitude Modulation Signaling Systems Satisfying a Spectral Mask Contraint", IPN Progress Report, 42-150, August 2002.