ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
FACULTY OF TECHNOLOGY

Design and Realization of Adaptive Channel Equalizer for Shortwave Communication using TMS320C50 DSP Kit

By
Mesfin Ayalew

Advisor
Dr. Ing. habil P. Haferkorn

December 2003

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
FACULTY OF TECHNOLOGY
ELECTRICAL ENGINEERING DEPARTMENT

Design and Realization of Adaptive Channel Equalizer for Shortwave Communication using TMS320C50 DSP Kit

By
Mesfin Ayalew

December 2003

Design and Realization of Adaptive Channel Equalizer for Shortwave Communication using TMS320C50 DSP Kit

By
Mesfin Ayalew

A thesis submitted to School of Graduate Studies of Addis Ababa University in partial fulfillment for the degree of Master of Science in Electrical Engineering.

Addis Ababa University
December 2003

DECLARATION

I, the undersigned, hereby declare that this thesis is my original work carried out under the supervision of Dr. Ing. habil P. Haferkorn, has not been presented as a thesis for a degree in any other university and that all sources used for the thesis are duly acknowledged.

Mesfin Ayalew

ACKNOWLEDGEMENT

I would like to express my deep appreciation to my advisor, Dr.- Ing. habil P. Haferkorn, for his unreserved cooperation, valuable ideas and constructive comments, which make this work a reality. I would also thank him for providing me with the TMS320C50 Digital Signal Processor Starter Kit in which my work is implemented and tested on.

My special thanks and heartfelt appreciation goes to Ato Taye Abdulkadir who shared me all his resources without reservation. His Comments and suggestions during the development of this material were of great help. Had it not been his keen support, encouragement and cooperation, this work may not be a reality. Thanks a lot, Taye.

Once again I want to thank Ato Fekadu Bedada, Manager - AMTS, Ethiopian Airlines, for his invaluable help and cooperation in my study period. I gratefully acknowledge all the supports I got from him. I would like to extend my gratitude to all my friends in Addis Ababa University and Ethiopian Airlines, who have in one way or another contributed for my work. I am also grateful for W/ro. Wude's family for providing me a conducive research environment during my study period.

Finally, I am indebted to my family for their patience, support and encouragement throughout my study.

Mesfin Ayalew

TABLE OF CONTENTS

CHAPTER THREEE

CHAPTER FOUR

CHAPTER FIVE

CHAPTER SIX

CHAPTER SEVEN

LIST OF ABBREVIATIONS

A/D	Analog to Digital
ACI	Adjacent Channel Interference
AIC	Analog Interface Card
AMTOR	Amateur Teleprinting over Radio
ARQ	Automatic Repeat Request
ASK	Amplitude Shift Keying
ATM	Asynchronous Transfer Mode
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CW	Continuous Wave
D/A	Digital to Analog
DFE	Decision Feedback Equalizer
DPF	Deterministic Phase Function
DPP	Delay Power Profile
DS/SS	Direct Sequence Spread Spectrum
DSB	Double Side Band
DSK	Digital Signal Processor Starter Kit
DSP	Digital Signal Processing
FDM	Frequency Division Multiplexing
FFF	Feed Forward Filter
FFT	Fast Fourier Transform
FH	Frequency Hopping
FH/SS	Frequency Hopping Spread Spectrum
FIR	Finite Impulse Response
FS	Fractionally Spaced
FSK	Frequency Shift Keying
GSM	Global System for Mobile
HF	High Frequency

I/O	Input – Output
ISDN	Integrated Services of Digital Network
ISI	Inter-Symbol Interference
ITS	Institute of Telecommunication Science
JPEG	Joint Photographic Experts Group
LMS	Least Mean Square
LOS	Line of Sight
LTE	Linear Transversal Equalizer
MAP	Maximum a Priori Probability
MIPS	Millions of Instructions per Second
ML	Maximum Likelihood
MLSE	Maximum Likelihood Sequence Estimation
MMSE	Minimize Mean Square Error
MPEG	Motion Pictures Experts Group
MSE	Mean Square Error
OFDM	Orthogonal Frequency Division Modulation
PACTOR	Packet Teleprinting over Radio
PAM	Pulse Amplitude Modulation
PDF	Probability Distribution Function
PLL	Phase Locked Loop
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
RTTY	Radio Teletype
SNR	Signal-to-Noise Ratio
SS	Symbol Spaced
SSB	Single Side Band
TCP/IP	Transmission Control Protocol / Internet Protocol
TDL	Tapped Delay Line
TDM	Time Division Multiplexing
WSSUS	Wide Sense Stationary Uncorrelated Scattering
ZF	Zero Forcing

ABSTRACT

The high frequency (HF) band is a cost effective medium for a long distance transmission beyond the horizon, for limited bandwidth of up to 3 kHz. However, the HF channel is simultaneously a time and frequency dispersive environment. Time spread is two fold and results from multiple reflections between earth surface and ionosphere, where the latter is known to have various reflection layers. This multipath behavior results in various received modes at the receiver site corresponding to a time spread of a few milliseconds. Frequency spread is due to motion of electron density distributions in the ionospheric media, and can be as high as 10 Hz. In addition to time and frequency spread the relative movements of ionospheric layers impose a Doppler shift on transmitted signals. The received signal in fading often modeled by a Rayleigh distributed envelope and a uniformly distributed phase.

Frequency selective fading channels produce intersymbol interference (ISI) in the received signal. In these channels an increase in the power does not improve performance, since additional power amplifies the ISI in step with the desired signal. The maximum likelihood sequence estimator (MLSE) is the optimal method of canceling the ISI; however, it is prohibitively complex. Suboptimal but efficient method of compensating for the ISI is to equalize the channel impairments by applying a filter at the receiver. In general radio channel characteristics are subject to variation in time, and this leads to the need for adaptive equalizers.

In this development, a linear frequency selective slowly fading model is used for the channel. A linear adaptive channel equalizer is considered as a solution to improve the performance of digital communication over this short wave channel. The performance of the equalizer is studied under a software simulated environment using MATLAB programming for a linearly modulated, binary phase shift keyed (BPSK), signals. The TMS320C50 DSK (digital signal processor starter kit) assembler and debugger are used to develop the assembly code so as to realize and implement the filter on TMS320C50 digital signal processor. Finally, the designed equalizer is tested for its performance.

CHAPTER ONE

INTRODUCTION

1.1. Overview of Digital Communication over HF Channel

During the last decades there has been a tremendous growth in communications all over the world. Public telephone networks have evolved from mostly accommodating analogue voice services to also becoming media for efficient data communications. The modems of today use sophisticated technology, which is based on the knowledge acquired at the absolute front line of science and they operate at data rates close to what is theoretically possible. When even higher data rates are needed, most of the communication takes place over satellite links or fiber optic cables, where transmission at several Gbit/s is a reality. The efficient digital communication links are the foundation on which integrated services, like ATM-networks (Asynchronous Transfer Mode), are being built. A driving force in this development is multimedia services, which call for highly reliable and flexible transmission rates.

This scenario stands in sharp contrast with existing wireless radio systems, which are based on technology distant from recent scientific achievements. These systems provide only low-speed data communication at error rates that are far from acceptable in a wireless ATM connection. For future networks to appear transparent to the user in the sense that the same level of service will be provided regardless of the medium of transmission, the radio communications must become considerably more efficient. Data must be conveyed much faster at significantly lower error rates, even when the quality of the link is poor.

Propagation of the radio waves over multiple paths between the transmitter and the receiver is commonly encountered in wireless systems. The multiple paths, which result from reflections, are superimposed at the receiving unit. When the arrival times of the different rays are of the same order of magnitude as the duration of the transmitted symbols, successive symbols are

smeared together. This effect is often referred to as ***Intersymbol Interference (ISI)***. For paths, where the time difference is comparable to the period of the carrier frequency, another effect results. Superposition of many waves with different phases here gives a spatial interference pattern, with narrow “holes” of extremely low signal power, so called deep fades. Those deep fades are located at distances comparable to the wavelength of the carrier frequency and the signal power in a

fade can be so low that communication is impossible. In a general scenario both the ISI and the interference pattern are time-varying due to motion of the transmitter/receiver or of the environment. These fluctuations are yet another problem, since they introduce a time varying distortion of the transmitted signal. The problem becomes serious when the variations are rapid compared with the signaling rate, i.e. when so called fast fading occurs. Suboptimal receivers here suffer from high error rates, which cannot even be lowered by increasing transmitter power. This effect is often referred to as an error floor. [10, 14, 19]

Radio frequency transmission between 3 and 30 MHz, which is called ***High Frequency (HF)*** or "***shortwave***" radio communications, is a widely used communication band for long distances since the invention of radio by Marconi and Popov. HF communications are growing at the moment despite appearing and all-around implementation of cellular and satellite communication systems. It's the only way of achieving global communication coverage without using expensive terrestrial and satellite infrastructure. HF communication still remains fail-safe means for communicating in high latitudes, including communication with mobile objects, like for tracking airplanes on their routes.

HF signal propagation is determined by the fact that it's repeatedly reflected from the ionosphere and the earth surface. That's why it is often called this communication method the "skywave communication". This phenomenon allows amateur radio operators to communicate with their colleagues on the other side of the globe with the radiated power no more than several watts. Still, analog communication using SSB (Single Side Band) without coding doesn't allow to provide a reliable and uninterrupted channel. It is determined by significant changes of propagation conditions depending on the geographical location, frequency, time of day, season, solar activity, and other circumstances.

However, besides long-term instability of the propagation conditions it turns out that the signal passed via the HF channel undergoes significant distortions and the dynamic of these distortions can have sufficiently little time constants. It is determined by the multipath nature of the signal propagation and distributed reflection from the ionosphere. As a result of this, there exist fading. The multipath is characterized by the value of delay between replicas of the signal. Typical values significantly depend on the length of the signal path and hesitate from 1 to 10 milliseconds. Fadings are characterized by the so-called 'bandwidth' which in middle latitudes is within the limits of 2 Hz, but in equatorial and high latitudes it can have the value up to 20 Hz. The results of different measurements in high latitudes showed that quite often the fading bandwidth achieves the value up to 73 Hz. [3, 8]

Frequency shifts and frequency-spread distortions imposed on the transmitted signal by ionosphere reflection are determined by temporal changeability of ionic concentration. Despite the fact that in middle latitudes the frequency shift is measured in fractions of Hertz, this phenomenon can cause prolonged periods of weakening the signal, caused by coherent subtraction of signals coming with different delays.

The information on HF propagation has only statistic nature and forms the background for designing terminal equipment (modems and others), or communication systems as a whole. The received signal in fading is often modeled by a Rayleigh (or Ricean) distributed envelope and a uniformly distributed phase, hence the name Rayleigh fading. It arises, not only in long-distance HF radio communication, but also in cellular systems and in underwater acoustic communications.

Changeability of propagation conditions makes specific requirements of applying adaptive mechanisms as on the modem level, and the controlling protocols level, which decreases the demands to the qualification of the operating personnel.

In early days, the HF radio digital communication systems were designed more to circumvent the problems imposed by the fading than to actually solve them. To avoid too rapid a fluctuation

(relative to the signaling rate) of the received signal power, a sufficiently high signaling rate is chosen. By doing so, the received signal can be regarded approximately constant over a data symbol interval. However, the signaling rate cannot be chosen very high, since that would lead to severe problems with ISI. Hence the choice of transmission rate is a compromise, which probably limits the efficiency of the system. [2,3]

Just 10-15 years ago it was widely believed that 75 bps is the only relevantly reliable rate, while the availability for the rates up to 2400 bps remained very low. Rapid development of digital signal processing during the last decades made possible to substantially improve features of HF communication systems. Modern standards for HF communications take into account all the above mentioned circumstances and set a high level of BER (Bit Error Rate) performance for receivers, modems in particular. As a result, the transmission rate up to 9600 bps in a 3 kHz channel has been achieved, at least experimentally. Data rates on benign skywave channels and on ground-wave paths can now reach 64 kbps, though sometimes using wider channel bandwidths. On the other hand, channel availability for the rates lower than 2400 bps is significantly increased.

Improvements are achieved by introducing new modern reception techniques such as channel equalizer, maximum a posteriori probability (MAP) decoding, etc. The cost of improved performance is increased hardware requirements to HF modems. However, novel DSPs, such as TMS320C50x or TMS320C67x from Texas Instruments fit such hardware requirements with low number of external components.

As mentioned above, there are three main effects caused by fading. Those are the deep fades of the signal power, time varying distortion and intersymbol interference. Even though the cause of all is the same, different strategies are required to deal with each of the impairments. To handle the first one, i.e. the deep fades, diversity techniques are needed. The principle is that a certain information bit shall reach the receiver via different paths. The paths can e.g. be different antennae (space diversity), different frequencies (frequency diversity), different times (time diversity) or combinations thereof. When one of the paths is in a deep fade, the other ones may not; giving a higher resistance against the fades than if only a single path was used.

While the solutions to the problem of the deep fades were to be found in code design and diversity techniques, the problems of the ISI and the time varying distortion can be solved by detection methods. Equalization combined with Maximum Likelihood Sequence Estimation (MLSE) is used in practice to compensate the resulting ISI. Commonly used equalizers for such receivers are the trellis-based maximum likelihood sequence estimator (MLSE), linear equalizer, or nonlinear equalizers – like decision feedback equalizer (DFE). The MLSE is the optimal method of canceling channel induced ISI; however, the complexity of this receiver grows exponentially with the length of the channel impulse response, as a consequence of which simpler suboptimal detectors are required. Therefore, equalization using linear estimation methods is preferred due to ease of implementation and less computational requirements. The well-known LMS (Least Mean Square) algorithm has been used extensively for equalization of channel effects. This approach necessitates a sequence of training data, initially, to determine filter coefficients, prior to actual data transmission. Training may also occur periodically, i.e., sequences of known training symbols are regularly inserted into the stream of transmitted symbols in order to aid the receiver in tracking the time-varying channel impulse response. [14, 19]

An optimal receiver would jointly perform channel estimation, equalization, and decoding, which is an extremely complicated problem. However, in recent works an attempt is made to reduce the computational burden of these schemes by separating out equalizer and estimator and carefully perform an iterative equalization, estimation, and decoding over three receiver modules.

1.2. The Thesis

An overall aim of this thesis can be summarized as: *Improving the throughput performance of digital communication over HF radio channel for a given signal-to-noise ratio (SNR) by implementing appropriate digital signal processing algorithms at the receiver, which counteract any distortion incurred by the channel.* To achieve this, a working model for an HF channel has to be developed and a software simulation of it is used to study its performance.

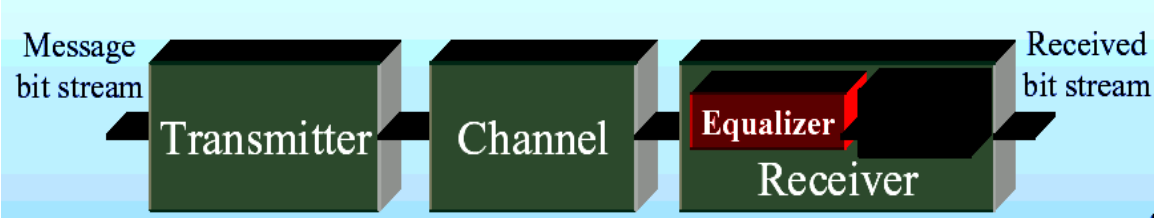


Fig. 1.1. *Model of the communication link*

As discussed above, high frequency (HF) skywave-communication systems exhibit low signal-to-noise (SNR) ratios, and may be subject to slow fading at mid-latitudes and fast fading at high and equatorial latitudes. Furthermore, for modern systems, the radio channel is almost always frequency selective. The performance of skywave HF communication and broadcast systems is dependent on how well the system design is able to compensate for the propagation channel.

Multipath propagation arises because replicas of the transmitted signal arrive at the receiver after reflection from more than one ionospheric layer, and/or after multiple reflections between the ionosphere and the ground. Each signal (or propagation mode) generally arrives with a different time delay, causing either constructive or destructive interference, which, when viewed in the frequency domain, dictates the coherence bandwidth of the channel.

Frequency (Doppler) shifts and frequency-spread distortion can be imposed on the transmitted signal by the temporal variability of the ionosphere, and this defines the coherence time of the channel. At high and mid latitudes, Doppler shifts and spreads of many Hertz are very common, and these are also often associated with spread returns, due to ionospheric irregularities.

The second chapter presents an introductory description of digital communication systems, digital signal representation, and all the required mathematical tools, which are used in this thesis work. Different digital modulation techniques, with more emphasis to binary phase-shift keying (BPSK) - a modulation of great interest for this thesis, are as well discussed.

For this paper, ***a Linear Frequency-Selective, Rayleigh Fading Channel Model,*** the most suitable and practical HF channel model, is considered. To study the effect of multipath,

performance degradation due to channel induced ISI, a three tap channel model is analyzed. A detailed description of HF channel characterization and modeling is treated in chapter 3. It focuses on how to model the two major channel distortions, intersymbol interference due to time dispersion of the transmitted signal as it propagates through multiple paths and Doppler shift (Rayleigh fading due to time-variant nature of the channel), and software simulation of the selected model. In dealing with the multipath nature of the channel that results in ISI, an equivalent discrete-time transversal filter, whose tap gain coefficients obtained from multipath intensity profile of the channel, was developed. The tap length of the filter is determined by multipath spread of the channel. The time-varying nature is interpreted as a dynamic system with uncertainties in its coefficients. These time-varying coefficients are generated by passing Rayleigh distributed noise through a lowpass Butterworth filter. The bandwidth of the Butterworth filter determines the relative bandwidth (fading rate) of the channel.

Chapter 4 treats mitigation techniques employed in modern receivers to combat channel induced distortions and the resulting loss of SNR. The optimal detector for trellis-coded signaling on frequency selective Rayleigh fading channel is described. In most cases it is prohibitively complex; therefore a simplified suboptimal detector, channel equalization, is proposed. Equalizers can operate on received data only or on a combination of received data and previous outputs. As indicated in Fig. 1.1, a linear adaptive linear equalizer is considered as a suboptimal detector in this project work and its detail mathematical derivation is also presented. A main result here is that the simplified detector can be designed to perform close to the optimal one.

Due to its severe impairments, HF channel requires a robust equalizing technique that has to be able to adapt quickly to the changing channel conditions. Robustness, on the other hand, usually means increased computational complexity of the algorithm used for updating the equalizer coefficients. Therefore, a DSP based fully flexible equalizer has to be developed so that it allows real time adjustments of the filter coefficients. For realization of the linear adaptive channel equalizer a Texas Instrument DSP, TMS320C50 DSP Kit, is used. A full description of the kit, hardware architecture and software programming, is discussed in the fifth chapter.

Chapter 6 deals with the whole design and implementation activities considered in this thesis work. As indicated in the very paragraph of this section, the main duty of this thesis work is to improve the error rate performance of digital communication over HF channel by implementing adaptive channel equalization. To test the performance of the equalizer, a linearly modulated, BPSK, uncoded signaling on a frequency selective Rayleigh fading channel, a good model for HF radio channel, is used as input to the receiver. The channel model is software realized using a MATLAB program. The channel corrupted symbols are then fed to an adaptive channel equalizer realized by TMS320C50 DSP Kit and improvement in the performance of the receiver is analyzed for different SNR.

A scientific work like this contains some limitations. Here those are mostly due to idealizations made in order keep the problem at a manageable level. Firstly, no distortion and disturbances except the frequency-selective Rayleigh fading and the additive noise are assumed. Those assumptions are unrealistic, since for example co-channel interference and adjacent channel interferences due to other users in the communication system are not considered. Secondly, perfect knowledge of the first and second order statistics of the fading process is assumed. In a real system, estimation of the channel parameters must be carried out which inevitably leads to estimation errors. Thirdly, the common assumptions of perfect symbol synchronization and perfect knowledge of the carrier frequency in the receiver are made. Fourthly, while implementing the equalizer using the DSP Kit, quantization and round-off errors are not accounted exhaustively.

CHAPTER TWO

DIGITAL COMMUNICATION

Digital communication today is used in a range of products from mobile phone to computer networks. The use of digital communication provides an increased performance compared to previously used analog communication methods. One important factor for the success of both the Internet and the mobile phones are the advances within the process technology. New process generations make it cheaper to implement advanced digital signal processing, which is the enabler for digital communication. Digital signal processing makes it possible to implement communication methods with more complex modulation schemes, adaptive receivers and error correction. These days it is possible to achieve transmission capacities close to the channel capacity theorem stated by Shannon, (2.1). The theorem describes the theoretical capacity limit on a communication channel disturbed by additive white gaussian noise with power spectral density of $N_0/2$, a channel bandwidth of W, and average power level of P. The capacity, C, is then given by

$$C = W \log_2\left(1 + \frac{P}{N_0 W}\right) \ bits/s \qquad 2.1$$

Since the channel capacity is limited, there is a need for techniques that can reduce the required channel capacity for a given service. Three important areas where compression techniques are widely used for better utilization of the channel capacity are transmission of speech, image and video. In a mobile phone system voice data is compressed from 64 kbit/s down to 11.4 kbit/s (GSM-Global System for Mobil, half-rate) keeping an acceptable quality of the speech.

For image and video transmissions different data compression standards, such as the JPEG (Joint Photographic Experts Group) and MPEG (Motion Pictures Experts Group) standards, are widely used. Interesting to note is that even if the available bandwidth keeps increasing compression of image and video signals will be crucial for many years to come. Transmitting standard resolution video with an acceptable quality requires 1.5-2.5 Mbit/s with compression. Transmitting

uncompressed video is not even an option today since this would require data rates above 50 Mbit/s.

2.1. Digital Communication System

A digital communication system can be outlined as shown in Fig. 2.1. The signal is created in a ***digital source*** which for instance can be digital data generated in a computer, digitized speech or digital video. The ***source encoder*** provides a one-to-one mapping from the input signal to a new representation suitable for transmission. The objective is to eliminate or reduce redundancy, i.e., giving the signal a more efficient representation. The ***source decoder*** re-creates the original signal. The ***channel encoder*** and ***decoder*** are used for providing a reliable transmission link by introducing a controlled redundancy that is used for detection and correction of transmission errors. In the ***modulator*** the information is modulated, which gives a signal suitable for transmission using the desired frequency band. It is the way information is mapped onto a signal. The transmitted information is divided into symbols where one symbol has a finite duration. The information content is encoded into the shape of the waveform during the symbol period. Common ways to encode the information is to put it into the amplitude and/or phase of the waveform. A brief description of most commonly used digital modulation techniques is given in section 2.3 of this chapter. The task of the ***detector*** in the receiver end is to detect which signal that was transmitted from the transmitter. Sometimes the detector and the demodulator are collected into one block which in this thesis is referred to as the equalizer.

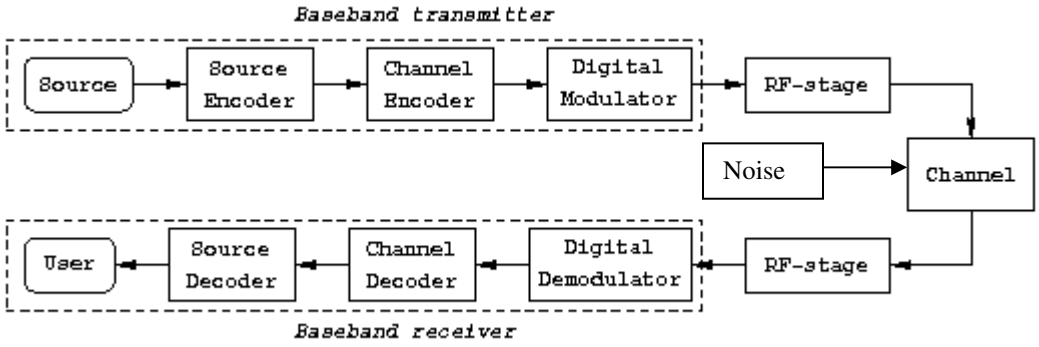


Fig. 2.1. *Digital communication system.*

2.2. Lowpass Representation of Bandpass Signals and Systems [19]

As discussed in the previous section, many digital information-bearing signals are transmitted by some type of carrier modulation. The channel over which the signal is transmitted, in most practical applications, is limited in bandwidth to an interval of frequencies about the carrier. The modulation is performed at the transmitting end of the communication system, indicated in Fig. 2.1, to generate the bandpass signal and the demodulation is performed at the receiving end to recover the digital information involve frequency translations. Without loss of generality and for mathematical convenience to analyze, model and simulate a system, it is desirable to reduce all bandpass signals and channels to equivalent lowpass signals and channels. Thus, this lowpass representation of signals and systems allows to ignore any linear frequency translations encountered in the modulation of a signal for purposes of matching its spectral content to the frequency allocation of a particular channel, and leads to modeling of a digital communication system by its lowpass equivalent. That is, transmission of equivalent lowpass signals through equivalent lowpass channels.

First, let us develop a mathematical representation of a bandpass signal and its lowpass equivalent. Consider a real-valued bandpass signal $s(t)$ at a carrier frequency f_c, an *analytical signal* $s_+(t)$ that contains only positive frequencies can be expressed as

$$s_+(t) = \int_{-\infty}^{\infty} S_+(f) e^{j2\pi ft} df \qquad 2.2.1$$

where $S_+(f) = 2u(f)S(f)$, $S(f)$ - Fourier transform of $s(t)$ and $u(f)$ - Unit step function.

Substituting the expression for $S_+(f)$ into (2.2.1), we obtain

$$s_+(t) = F^{-1}[2u(f)] * F^{-1}[S(f)] \qquad 2.2.2$$

Hence,

$$\begin{aligned} s_+(t) &= \left[\delta(t) + \frac{j}{\pi t}\right] * s(t) \\ &= s(t) + j\frac{1}{\pi t} * s(t) \end{aligned} \qquad 2.2.3$$

where $\left[\delta(t)+\frac{j}{\pi t}\right]=F^{-1}[2u(f)]$ and * denotes convolution. It can clearly be seen that the second term of (2.2.3) is the Hilbert transform of the bandpass signal $s(t)$.

Thus, the equivalent lowpass representation of the bandpass signal can be obtained by performing a frequency translation of $S_+(f)$ as

$$S_l(f)=S_+(f+f_C) \qquad 2.2.4$$

The equivalent time-domain relation, inverse Fourier transforming (2.2.4), is

$$\begin{aligned} s_l(t) &= s_+(t)e^{-j2\pi f_C t} \\ &= [s(t)+j\hat{s}(t)]e^{-j2\pi f_C t} \end{aligned} \qquad 2.2.5$$

where $\hat{s}(t)=\frac{1}{\pi t}*s(t)$ is the Hilbert transform of the bandpass signal. To clearly observe the relationship between the bandpass signal and its lowpass equivalent, (2.2.5) can be rearranged as

$$s(t)+j\hat{s}(t)=s_l(t)e^{j2\pi f_c t} \qquad 2.2,6$$

In general, the signal $s_l(t)$ is a complex-valued, and may be expressed as

$$s_l(t)=x(t)+jy(t) \qquad 2.2.7$$

Substituting (2.2.7) for $s_l(t)$ into (2.2.6) and equating real and imaginary parts, we obtain

$$\begin{aligned} s(t) &= x(t)\cos(2\pi f_c t)+y(t)\sin(2\pi f_c t) \\ \hat{s}(t) &= x(t)\sin(2\pi f_c t)+y(t)\cos(2\pi f_c t) \end{aligned} \qquad 2.2.8$$

The low-frequency signal components $x(t)$ and $y(t)$ can be viewed as amplitude modulations impressed on phase quadrature carrier components $\cos(2\pi f_c t)$ and $\sin(2\pi f_c t)$, respectively. Hence, $x(t)$ and $y(t)$ are called the ***quadrature components*** of the bandpass signal.

Alternatively, (2.2.8) can be have of the form

$$\begin{aligned} s(t) &= \mathrm{Re}\left\{[x(t)+jy(t)]e^{j2\pi f_c t}\right\} \\ &= \mathrm{Re}\left[s_l(t)e^{j2\pi f_c t}\right] \end{aligned} \qquad 2.2.9$$

where Re denotes the real part of the complex-valued quantity in the brackets following. The lowpass signal $s_l(t)$ is usually called the ***complex envelop*** of the real bandpass signal, and is basically the ***equivalent lowpass signal.***

Finally, the bandpass signal representation can as well be expressed as

$$s(t) = a(t)\cos[2\pi f_c t + \theta(t)] \qquad 2.2.10$$

where $a(t) = \sqrt{x^2(t) + y^2(t)}$ is the envelop of $s(t)$, and $\theta(t) = \tan^{-1}\left\{\frac{y(t)}{x(t)}\right\}$ is the phase of $s(t)$. Therefore, (2.2.8), (2.2.9) and (2.2.10) are the equivalent representations of the bandpass signals.

The Fourier transform of the bandpass signal $s(t)$ can also be expressed in terms of the equivalent lowpass signal Fourier transforms. Thus,

$$\begin{aligned} S(f) &= \int_{-\infty}^{\infty} s(t)e^{-j2\pi ft}dt \\ &= \int_{-\infty}^{\infty} \left\{\mathrm{Re}\left[s_l(t)e^{j2\pi f_c t}\right]\right\}e^{-j2\pi ft}dt \\ &= \frac{1}{2}\int_{-\infty}^{\infty}\left[s_l(t)e^{j2\pi f_c t} + s_l^*(t)e^{-j2\pi f_c t}\right]e^{-j2\pi ft}dt \\ &= \frac{1}{2}\left[S_l(f - f_c) + S_l^*(-f - f_c)\right] \end{aligned} \qquad 2.2.11$$

where $S_l(f)$ is the Fourier transform of $s_l(t)$. This defines the basic relationship between the spectrum of the real bandpass signal and the spectrum of the equivalent lowpass signal.

For most practical exercises, it is compulsory to describe the energy contained in the real bandpass signal in terms of its equivalent lowpass signal. To generate this relationship, let us first begin by defining the energy in the signal $s(t)$ as

$$\begin{aligned} E &= \int_{-\infty}^{\infty} s^2(t)dt \\ &= \int_{-\infty}^{\infty} \left\{\mathrm{Re}\left[s_l(t)e^{j2\pi f_c t}\right]\right\}^2 dt \\ &= \int_{-\infty}^{\infty} |s_l(t)|^2 dt + \frac{1}{2}\int_{-\infty}^{\infty} |s_l(t)|^2 \cos\left[4\pi f_c t + 2\theta(t)\right]dt \end{aligned} \qquad 2.2.12$$

For narrow band real value bandpass signal, the real envelop varies slowly relative to the rapid variation exhibited by the cosine function. The energy contributed by the second integral is very

small relative to the first integral in (2.2.12) and, hence, it can be neglected. Thus, for all practical purposes, the energy in the bandpass signal expressed in terms of the equivalent lowpass signal is

$$E = \frac{1}{2}\int_{-\infty}^{\infty} |s_l(t)|^2 dt \qquad 2.2.13$$

Similarly, a bandpass system (channel) described by its impulse response $h(t)$, or by its frequency response $H(f)$ can be represented by its lowpass equivalent system (or channel response) as follows. Suppose that $h(t)$ is real, and define the lowpass equivalent response as

$$H_l(f - f_c) = \begin{cases} H(f) & (f > 0) \\ 0 & (f < 0) \end{cases} \qquad 2.2.14$$

And for real impulse response function $H(f) = H^*(-f)$, we have

$$H(f) = H_l(f - f_c) + H_l^*(-f - f_c) \qquad 2.2.15$$

which resembles (2.2.11) except for the factor $\frac{1}{2}$. Taking the inverse transform of $H(f)$, yields $h(t)$ in the form

$$h(t) = 2\,\mathrm{Re}\left[h_l(t) e^{j2\pi f_c t} \right] \qquad 2.2.16$$

where $h_l(t)$ is the Fourier transform of $H_l(f)$. Generally, the impulse response $h_l(t)$ of the equivalent lowpass filter is complex-valued.

As a narrowband bandpass signals systems can be represented by equivalent lowpass signals and systems, the output of the bandpass system to a bandpass input signal can as well simply be obtained from the equivalent lowpass signal and the equivalent lowpass impulse response of the system by a convolution integral

$$r_l(t) = \int_{-\infty}^{\infty} s_l(\tau) h_l(t - \tau) d\tau \qquad 2.2.20$$

where $r_l(t)$ is the lowpass equivalent of the output bandpass signal of the bandpass system.

2.3. Digital Modulation [14, 19]

In the transmission of digital information over a communication channel, the modulator is the interface device that maps the digital information into analog waveforms that match the characteristics of the channel. The matching is generally performed by taking blocks of $k = \log_2 M$ binary digits at a time from the information sequence $\{a_n\}$ and selecting one of $M = 2^k$ deterministic, finite energy waveforms $\{s_m(t),\ \ m = 1,2,...,M\}$ for transmission over the channel. These waveforms may differ in either amplitude or in phase or in frequency, or some combination of two or more signal parameters. The mapping from the information sequence to the corresponding waveform may be performed under the constraint that a waveform transmitted in any time interval depends on one or more previously transmitted waveforms, ***modulation with memory***, or without any constraint on previously transmitted waveforms, ***memoryless modulation***. The signal dependency, in modulator with memory, is usually introduced for the purpose of shaping the spectrum of the transmitted signal so that it matches the *spectral characteristics of the channel*. Furthermore, the modulation methods can be characterized as either ***linear***, where principle of superposition applies in the mapping of the digital sequence into successive waveforms, or ***nonlinear*** in which superposition does not apply to signals transmitted in the successive time intervals.

For the choice of the modulation scheme in high bit rate data transmission over communication channel there exist three main possibilities with each one again splitting up into different sub-types and implementation approaches. The first main division is into passband and baseband modulation where the passband scheme can be subdivided into single carrier and multicarrier modulation. The main difference between passband and baseband modulation is the fact that in case of baseband modulation the information stream is coded and spectrally shaped, but the position of its spectrum is not changed. In case of passband modulation on the other hand the information is modulated onto a carrier that is not constrained in its frequency, meaning that the transmission spectrum in this case can be transferred to every desired frequency band. Therefore, the main practical difference is the position of the used frequency band. While baseband modulation has a lowpass characteristic, beginning at or near zero, passband modulation has a bandpass characteristic with the used band situated somewhere in the frequency domain. In case of passband modulation one or more carriers may be used for the transmission. In case of multicarrier modulation several carriers of different frequencies are used to subdivide the used

frequency band into a bank of subchannels with each subchannel virtually being an independent passband system. Together these subchannels compose a continuous spectrum very similar to the one of single carrier passband modulation. Here, some of the most commonly used modulation methods that have direct correlation with this thesis work, linear-memoryless modulations, are briefly described.

2.3.1. Pulse Amplitude Modulation (PAM)

In a PAM, which is also called ***amplitude-shift keying (ASK)***, transmission system k bits are transmitted using symbols of $M = 2^k$ discrete amplitude levels as is depicted in Fig. 2.2, showing 4 amplitude levels in order to transmit 2 bit/symbol and the decision limits toward the neighboring states. In Fig. 2.2 also the discrete nature in the time domain can be seen, as only the values of the received signal at the sampling instants (centre of the points in horizontal direction) are of interest.

For the sequence of binary digits at the input of *M-level PAM* arriving at the rate of $R \ \ bits/s$, the signal waveforms may be represented as

$$\begin{aligned} s_m(t) &= \mathrm{Re}\left[A_m g(t) e^{j2\pi f_c t} \right] \\ &= A_m g(t) \cos 2\pi f_c t, \qquad m = 1,2,...,M, \qquad 0 \le t \le T \end{aligned} \tag{2.3.1}$$

where $\{A_m = 1 \le m \le M\}$ denote the set of M possible amplitudes corresponding to $M = 2^k$ possible $k-$bit blocks or symbols.

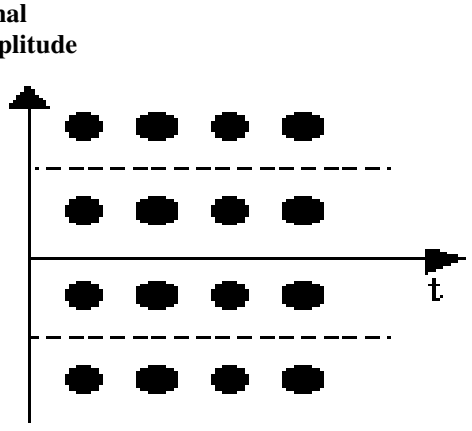


Fig. 2.2. Structure of PAM signaling (4-PAM for transmission of 2 bit/symbol)

The signal amplitudes A_m take the discrete values

$$A_m = (2m - 1 - M)d, \qquad m = 1,2,...,M \tag{2.3.2}$$

where $2d$ is the distance between adjacent signal amplitudes. The waveform $g(t)$ is a real-valued signal pulse whose shape influences the spectrum of the transmitted signal. The symbol rate for the PAM signal is R/k, and the time interval $T_b = 1/R$ is the *bit interval* and the *symbol interval* is $T = k/R = kT_b$. The energy contained in M-PAM signals, E_m, is expressed as

$$\begin{aligned} E_m &= \int_0^T s_m^2(t)dt \\ &= \frac{1}{2}A_m^2 \int_0^T g^2(t)dt \\ &= \frac{1}{2}A_m^2 E_g \end{aligned} \tag{2.3.3}$$

where E_g denotes the energy in the pulse $g(t)$. And it can be noted that the Euclidean distance between any pair of signal points and the minimum value are given as

$$\begin{aligned} d_{mn}^{(e)} &= \sqrt{(s_m - s_n)^2} \\ d_{\min}^{(e)} &= d\sqrt{2E_g} \end{aligned} \tag{2.3.4}$$

The carrier-modulated PAM is a double-sideband (DSB) signal and requires twice the channel bandwidth of the equivalent lowpass signal for transmission. The digital PAM signal is also appropriate for transmission over a channel that does not require carrier modulation, telephone line. In this thesis work, binary PAM $(M = 1)$, which is also called binary phase-shift keyed (BPSK), performance over HF radio channel is considered as it is bandwidth-efficient and commonly used modulation technique for the selected channel. Binary PAM waveforms have the special property that

$$s_1(t) = -s_2(t) \tag{2.3.5}$$

From this relationship, it can be noted that the two signals have the same energy and a cross-correlation coefficient of -1, antipodal signals.

2.3.2. Phase Modulation

In digital phase modulation, usually called phase-shift keying (PSK), k bits of the information sequence are transmitted as $M = 2^k$ distinct phase shifts of the carrier. The M signal waveforms are represented as

$$\begin{aligned} s_m(t) &= \mathrm{Re}\left[g(t) e^{j2\pi(m-1)M} e^{j2\pi f_c t} \right], \qquad m = 1,2,...,M, \qquad 0 \le t \le T \\ &= g(t)\cos\left[2\pi f_c t + \frac{2\pi}{M}(m-1) \right] \\ &= g(t)\cos\frac{2\pi}{M}(m-1)\cos 2\pi f_c t - g(t)\sin\frac{2\pi}{M}(m-1)\sin 2\pi f_c t \end{aligned} \qquad 2.3.6$$

where $g(t)$ the signal pulse shape and $\theta_m = 2\pi(m-1)/M$, $m = 1,2,...,M$ are the M possible phases of the carrier that convey the transmitted information. All the transmitted M-PSK symbols have the same energy and the average energy per symbol is the same as energy of any individual symbols, and expressed as

$$\begin{aligned} E &= \int_0^T s_m^2(t)dt \\ &= \frac{1}{2}\int_0^T g^2(t)dt = \frac{1}{2}E_g \end{aligned} \qquad 2.3.7$$

The signal waveforms may be represented as a linear combination of two orthonormal signal waveforms $f_1(t)$ and $f_2(t)$ as,

$$s_m(t) = s_{m1}(t) f_1(t) + s_{m2}(t) f_2(t) \qquad 2.3.8$$

where

$$\begin{aligned} f_1(t) &= \sqrt{\frac{2}{E_g}} g(t)\cos 2\pi f_c t \\ f_2(t) &= \sqrt{\frac{2}{E_g}} g(t)\sin 2\pi f_c t \end{aligned} \qquad 2.3.9$$

and the two-dimensional vectors $_m = [s_{m1} \quad s_{m2}]$ are given by

$$_m = \left[\sqrt{\frac{E_g}{2}}\cos\frac{2\pi}{M}(m-1) \quad \sqrt{\frac{E_g}{2}}\sin\frac{2\pi}{M}(m-1) \right], \qquad m = 1,2,...,M \qquad 2.3.10$$

Because all symbols have the same amplitude, PSK modulation is less sensitive to nonlinearities in the channel and therefore widely used on power limited radio channels where the amplifiers are driven close to their nonlinear region of operation in order to maximize power efficiency.

2.3.3. Quadrature Amplitude Modulation (QAM)

QAM uses the effect that two signals, using the same carrier frequency, but being modulated onto a sine and a cosine carrier, which do not influence each other in an ideal environment. Basically QAM can be understood as two-dimensional PAM with each of the two carriers (having 90º phase-offset) being used with one-dimensional PAM, and it doubles the bandwidth efficiency of PAM. The total QAM symbol can be understood as a two-dimensional constellation of signal points where the extension in each dimension depends on the number of signaling levels of the one-dimensional PAM. The corresponding signal waveforms may be expressed as

$$\begin{aligned} s_m(t) &= \mathrm{Re}\left[(A_{mc} + jA_{ms})\, g(t) e^{j2\pi f_c t} \right], \qquad m = 1,2,....M, \quad 0 \le t \le T \\ &= A_{mc}\, g(t) \cos 2\pi f_c t - A_{ms}\, g(t) \sin 2\pi f_c t \end{aligned} \qquad 2.3.11$$

where A_{mc} and A_{ms} are the information bearing signal amplitudes of the quadrature carries and $g(t)$ is the signal pulse. From this expression, it is apparent that the QAM signal waveforms may be viewed as combined amplitude and phase modulation.

As in the case of PSK signals, QAM signal waveforms may be represented as a linear combination of orthonormal signal waveforms as $s_m(t)$ described in (2.3.8), and $f_1(t)$, $f_2(t)$ given by (2.3.9), but with different two-dimensional vector space expressed as

$$s_m = \left[A_{mc} \sqrt{\frac{1}{2} E_g} \qquad A_{ms} \sqrt{\frac{1}{2} E_g} \right] \qquad 2.3.12$$

Because the quadrature channels are orthogonal, the modulator can be designed with two signal branches, each configured exactly as a PAM modem, one channel modulating the cosine of the carrier, the other the sine. At the receiver, the two channels are prevented from interfering one another by their orthogonality. The data rate for the QAM is simply the sum of the data rates of the two channels, but the signal bandwidth, which is determined by the pulse shape $g(t)$, is

unchanged from single channel PAM signal. Thus, the bandwidth efficiency of QAM design is twice that of PAM design.

2.3.4. Power Spectra of Linearly Modulated Signals

Analyzing and/or deriving an expression for the power spectrum of a linearly modulated signal, will provide a designer about the bandwidth requirements for the corresponding transmitting channel and the type of modulation which suits the given constraints. For a bandpass signal $s(t)$ with its lowpass equivalent signal $v(t)$, the autocorrelation function of $s(t)$ can be expressed as

$$\phi_{ss} = \mathrm{Re}[\phi_{vv}(\tau)e^{j2\pi f_c\tau}] \qquad 2.3.13$$

where $\phi_{vv}(\tau)$ is the autocorrelation of the equivalent lowpass signal $v(t)$. The Fourier transform of (2.3.13) yield to the desired power density spectrum $\Phi_{ss}(f)$ in the form

$$\Phi_{ss}(f) = \frac{1}{2}\left[\Phi_{vv}(f - f_c) + \Phi_{vv}(-f - f_c)\right] \qquad 2.3.14$$

where $\Phi_{vv}(f)$ is the power density spectrum of $v(t)$.

In a system that employs linear digital modulation, the equivalent lowpass signal $v(t)$ can be represented in general form as

$$v(t) = \sum_{n=-\infty}^{\infty} I_n g(t - nT) \qquad 2.3.15$$

where the transmission rate is $1/T = R/k$ symbols/s and $\{I_n\}$ represents the sequence of symbols that results from mapping $k-bits$ blocks into corresponding signal points from the appropriate signal space diagram. In PAM, the type of modulation considered in this thesis work, the sequence $\{I_n\}$ is real and corresponds to the amplitude values of the transmitted signal. Hence, the autocorrelation function of $v(t)$ is

$$\begin{aligned}\phi_{vv}(t+\tau,t) &= \frac{1}{2}E\left[v^*(t)v(t+\tau)\right] \\ &= \frac{1}{2}\sum_{n=-\infty}^{\infty}\sum_{=-\infty}^{\infty}E\left[I_n^* I_m\right]g^*(t-nT)g(t+\tau-mT)\end{aligned} \qquad 2.3.16$$

Assuming that the sequence of information symbols $\{I_n\}$ is wide-sense stationary with mean μ_i and the autocorrelation function

$$\phi_{ii}(m) = \frac{1}{2} E\left[I_n^* I_{n+m}\right] \qquad 2.3.17$$

Hence (2.3.16) can be expressed as

$$\phi_{vv}(t+\tau, t) = \sum_{m=-\infty}^{\infty} \phi_{ii}(m) \sum_{n=-\infty}^{\infty} g^*(t-nT)\, g(t+\tau-nT-mT) \qquad 2.3.18$$

It can be noted that the second summation is periodic in t variable with period T, and hence $\phi_{vv}(t+\tau, t)$ is also periodic in the t variable with period T. In addition, the mean value of $v(t)$ is periodic with period T. Therefore, $v(t)$ is a stochastic process having periodic mean and autocorrelation function, cyclostationary process. The power spectrum of such a process can be computed by averaging $\phi_{vv}(t+\tau, t)$ over a single period. Thus

$$\overline{\phi}_{vv}(\tau) = \frac{1}{T} \sum_{m=-\infty}^{\infty} \phi_{ii}(m) \phi_{gg}(\tau - mT) \qquad 2.3.19$$

where $\phi_{gg}(\tau) = \int_{-\infty}^{\infty} g^*(t)\, g(t+\tau)\, dt$ is the time-autocorrelation function of $g(t)$.

The average power density spectrum can be obtained by Fourier transforming the relation given by (2.3.19) as

$$\Phi_{vv}(f) = \frac{1}{T} \left|G(f)\right|^* \Phi_{ii}(f) \qquad 2.3.20$$

where $G(f)$ is the Fourier transform of $g(t)$, and $\Phi_{ii}(f)$ denotes the power density spectrum of the information sequence.

The result (2.3.20) depicts that the power density spectrum of a linearly modulated signal depends on the spectral characteristics of the pulse $g(t)$ and the information sequence $\{I_n\}$. That is, the spectral characteristics of $v(t)$ can be controlled by design of the pulse shape and the correlation characteristics of the information sequence. When the information symbols are equally likely and symmetrically positioned, a desirable condition for digital modulation techniques under consideration, the power spectrum density of the transmitted signal will only depend on the spectral characteristics of the signal pulse $g(t)$. Thus, the system designer can

control the spectral characteristics of the digitally modulated signal by proper selection of the characteristics of the information sequence to be transmitted, channel coding.

2.4. Optimum Receiver [19]

So far we have described the most commonly used types of modulation methods that may be used to transmit digital information through a communication channel. At the other end of the communication system, there should be a mechanism by which the transmitted signal has to be recovered. However, the received signal is corrupted by noise of different sources, which degrades the performance of the system. Noise in a communication system may be classified into two main categories, depending on its source. The first, which arises outside the electronic parts of a communication system, mostly includes:

1. ***Natural noise (sky noise)*** - solar and stellar radiation (galactic noise) - lightening discharges and other atmospherics.
2. ***Man-made noise*** - from electrical equipment, e.g. arching contacts in electrical machinery. This noise, which enters the system via the receiving antenna, may be avoided by suitably placing the antenna away from its sources.

The second category of noise, which results from sources within a communication system, is the most common forms of naturally occurring noise. It is also referred to as “circuit noise”. Its main sources are:

1. ***Thermal Noise:*** It occurs due to the random motion of electrons in a conductor by the thermal agitation of atoms. This is perhaps the most fundamental type of noise since it is always present in a conductor without the external application of electrical energy (except at a temperature of absolute zero).
2. ***Shot Noise:*** This noise is encountered in active devices (valves, transistors and diodes) and caused by the discrete nature of electrons and holes flowing in semiconductors and other charge-transfer processes.

Shot noise is in many ways similar to thermal noise. They are both due to random fluctuations of a large number of electrons, have uniform power spectral densities, and furthermore, the mean square current in both cases is directly proportional to the bandwidth of the system. Thermal noise and shot noise both have uniform power spectral densities and Gaussian distributions and, therefore, can be considered as Additive White Gaussian Noise (AWGN) processes.

Therefore, the channel is assumed to corrupt the signal by addition of white gaussian noise, neglecting other signal distortions, like dispersion, fading and Doppler, incurred by the channel. Thus the received signal in the symbol interval $0 \le t \le T$ may be expressed as

$$r(t) = s_m(t) + n(t), \qquad 0 \le t \le T \qquad 2.4.1$$

where $n(t)$ denotes a sample function of the additive white gaussian noise (AWGN) process with power spectral density $\Phi_{nn}(f) = \frac{1}{2}N_0 \ \ W/Hz$. Based on the observation of $r(t)$ over the signal interval, it is desired to design a receiver that is optimum in the sense that it minimizes the probability of making an error. To simplify the design and performance analysis, it is advisable to subdivide the receiver into two parts--the ***signal demodulator*** that converts the received waveform $r(t)$ into an *N*-dimensional vector $\mathbf{r} = [r_1, r_2, ..., r_N]$, where *N* is the dimension of the transmitted signal waveform, and the ***detector*** that decides which of the *M*-possible signal waveforms was transmitted based on the vector $\mathbf{r}$. The signal demodulator can be realized either based on the use of ***signal correlators*** or based on the use of ***matched filters.***

2.4.1. Correlation Demodulator

In this type of demodulator, the received waveform $r(t)$ is expended into a series of linearly weighted orthonormal basis functions $\{f_n(t)\}$, N basis functions that span the transmitted signal space, so that every one of the possible transmitted signals of the set $\{s_m(t), \quad m \le t \le M\}$ can be represented as a weighted linear combination of $\{f_n(t)\}$. The functions $\{f_n(t)\}$ do not span the noise space, but these noise terms which fall outside signal space are irrelevant to the detection of the signal. The demodulator may be realized by letting the received signal $r(t)$ to pass through a parallel bank of N cross-correlators which basically compute the projection of $r(t)$ onto the N basis functions $\{f_n(t)\}$. Thus, we have

$$\begin{aligned} r_k &= \int_0^T r(t) f_k(t) dt = \int_0^T [s_m(t) + n(t)] f_k(t) dt \\ r_k &= s_{mk} + n_k, \qquad k = 1,2,...,N \end{aligned} \tag{2.4.2}$$

where

$$\begin{aligned} s_{mk} &= \int s_m(t) f_k(t) dt, \qquad k = 1,2,...,N \\ n_k &= \int n(t) f_k(t) dt, \qquad k = 1,2,...,N \end{aligned} \tag{2.4.3}$$

The signal is now represented by the vector $\mathbf{s}_m$ with components $s_{mk}, k = 1,2,...,N$, where their values depend on which of the M signals was transmitted. The components $\{n_k\}$ are random variables that arise from the presence of the additive noise. Note that the correlator outputs $(r_1, r_2, ..., r_N)$ are *sufficient statistics* for reaching a decision on which of the M signal was transmitted.

2.4.2. Matched Filter Demodulator

In a matched filter receiver, similar to correlator demodulator, the orthonormal set is chosen such that the modulator waveforms can be expressed as linear combinations of its basis functions $\{f_n(t)\}$. However, instead of using a bank of N correlators to generate the variables $\{r_k\}$, a bank of N linear filters are used. Let us suppose that the impulse responses of the N filters are

$$h_k(t) = f_k(T-t), \qquad 0 \le t \le T \tag{2.4.4}$$

where $\{f_k(t)\}$ are the N basis functions and $h_k(t) = 0$ outside the interval $0 \le t \le T$. The outputs of the filters are

$$\begin{aligned} y_k(t) &= \int_0^t r(\tau) h_k(t-\tau) d\tau \\ &= \int r(\tau) f_k(T-t-\tau) d\tau, \qquad k = 1,2,...,N \end{aligned} \tag{2.4.5}$$

Now, if we sample the output of the filters at $t = T$, we obtain

$$y_k(T) = \int r(\tau) f_k(\tau) d\tau = r_k, \qquad k = 1,2,...,N \tag{2.4.6}$$

Hence, the sampled outputs of the filters at $t = T$ are exactly the set of values $\{r_k\}$ obtained from *N* linear correlators.

The matched filter demodulator has an interesting property that, if a signal $s(t)$ is corrupted by AWGN, the filter (its impulse response matched to $s(t)$) maximizes the signal-to-noise ratio (SNR). The maximum SNR obtained with the matched filter may be derived and has of the form

$$SNR_{\max} = \frac{2}{N_0} \int s^2(t) dt = 2E / N_0 \tag{2.4.7}$$

From the above relation, it can be noted that the output SNR from the matched filter depends on the energy, E, of the waveform $s(t)$ but not on the detailed characteristics of $s(t)$.

However, on the multipath-fast fading channel optimum receiver with matched filter or linear correlators originally developed for AWGN are unable to efficiently handle the rapid (compared with the signaling rate) fluctuations of the received signal power. More sophisticated solutions are needed. It is essential to make the transition from the time continuous received signal to a discrete representation in the receiver, without losing too much accuracy. Both an adequate number of discrete observables and a sufficiently long observation interval are required. When this is fulfilled, error probability curves with steeper slopes and considerably lower error floors than those of matched filter based receivers are obtained.

The Optimum Detector

For a signal transmitted over an AWGN channel, the decision vector $\mathbf{r} = [r_1, r_2, ..., r_N]$ for the detector, which contains all the relevant information in the received signal waveform, can either be produced by a linear correlator or matched filter demodulator. Here, the thing that has to be considered in designing a detector is what optimum decision rule over the observation vector $\mathbf{r}$ should base on. For this development, it is convenient to make two assumptions on the nature of the signal being transmitted and the channel over which the signal is transmitted. One is assuming that there is no memory in signals transmitted in successive signal interval, i.e., memoryless modulation over frequency nonselective (ISI free) channel. And the second assumption is that the transmitted signal has memory, where the modulation is with memory and/or the channel induces ISI.

First let us consider the case where the signal has no memory. Here, we wish to design a signal detector that makes decision on the transmitted signal interval based on the vector $\mathbf{r}$ in each interval such that the probability of correct decision is maximized. There are two decision rules often used in most practical receivers, *maximum a posteriori probability* (MAP) *and maximum likelihood* (ML) *criteria.*

Maximum a posteriori probability (MAP)

The decision is based on selecting the signal corresponding to maximum set of posterior probabilities $\{P(\mathbf{s}_m|\mathbf{r})\}$, the probability of the signal $\mathbf{s}_m$ transmitted given the observation vector $\mathbf{r}$ for all M possible transmitted signals. This criterion maximizes the probability of a correct decision and, hence minimizes the probability of error. Using Bayes' rule, the posterior probabilities may be expressed as

$$P(\mathbf{s}_m|\mathbf{r}) = \frac{p(\mathbf{r}|\mathbf{s}_m)P(\mathbf{s}_m)}{p(\mathbf{r})} \qquad 2.4.8$$

where $p(\mathbf{r}|\mathbf{s}_m)$ is the conditional pdf of the observed vector given $\mathbf{s}_m$, and $P(\mathbf{s}_m)$ is a priori probability of the *mth* signal being transmitted. The denominator of (2.4.8) may be expressed as

$$p(\mathbf{r}) = \sum_{m=1}^{M} p(\mathbf{r}|\mathbf{s}_m)P(\mathbf{s}_m) \qquad 2.4.9$$

From (2.4.8) and (2.4.9), it can be noted that the computation of the posteriori probabilities requires knowledge of the a priori probability $P(\mathbf{s}_m)$ and the conditional pdfs $p(\mathbf{r}|\mathbf{s}_m)$ for $m = 1,2,...,M$.

Maximum Likelihood (ML)

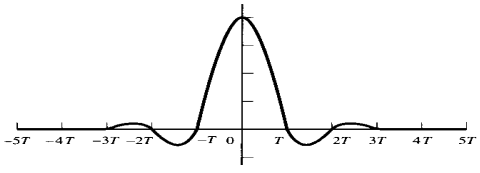
It is a decision criterion based on the maximum $p(\mathbf{r}|\mathbf{s}_m)$ over the *M* possible transmitted signals. A detector based on the MAP criterion and one that is based on ML criterion make the same decisions as long as the a priori probabilities $P(\mathbf{s}_m)$ are equal, i.e., the transmitted signals are equiprobable. For the AWGN channel, the decision rule based on the ML criterion reduces to finding the signal $\mathbf{s}_m$ that is closest in distance to the received signal vector.

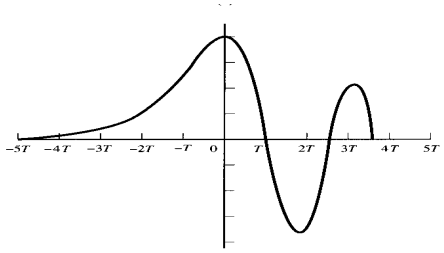
The symbol-by-symbol detector described in the preceding section will be optimum in the sense of minimizing the probability of a symbol error only if the signal has no memory. On the other hand, when the transmitted signal has memory, i.e., the signals transmitted in successive symbol intervals are interdependent due to either modulation with memory or channel induced ISI, the optimum detector is a detector that bases its decision on observation of a sequence of received signals over successive signal intervals. A *maximum likelihood sequence detector* (MLSE) algorithm that searches for the minimum Euclidean distance path through the trellis that characterizes the memory in the transmitted signal (channel induced ISI) is often used in most practical receivers. We may reduce the number of sequences in the trellis search by using *Viterbi algorithm*, a sequential trellis search algorithm for performing ML sequence detection, to eliminate sequences as new data is received from the demodulator.

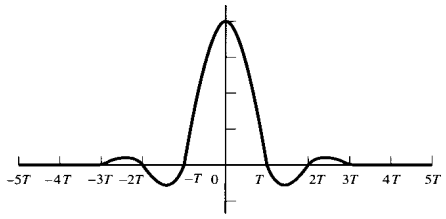
2.5. Band-Limited Channel [19]

The available radio spectrum, HF band in particular, for communication is very limited as compared to the demand, users (broadcasting or two-way link) looking for service. Therefore, there should be a means to simultaneously avail communication link for such users. Frequency-division multiplexing (FDM), time-division multiplexing (TDM) or a combination of the two may be employed to establish the required multiple channels. Thus, this leads a problem of designing a communication system for such a band-limited, limited to some specified bandwidth W Hz. For this condition the channel may be modeled as a linear filter having an equivalent lowpass frequency response $C(f)$ that is zero for $|f| > W$. Moreover, the channel may introduce distortion within the bandwidth of the channel, either due to variable amplitude response of $C(f)$ or nonlinear nature of the channel phase function or both.

As a result of the amplitude and phase distortion caused by the nonideal channel frequency response characteristics $C(f)$, a succession of pulses transmitted through the channel at rates comparable to the bandwidth W are smeared to the point that they are no longer distinguishable as well defined pulses at the receiving terminal. Instead, they overlap and, thus we have channel induced ISI. As illustrated in Fig. 2.3, a band-limited pulse having zeros periodically spaced in time transmitted through a channel modeled as a linear envelope delay characteristics results in a received pulse having zero-crossings that are no longer periodically spaced. If the information is conveyed by pulse amplitude, as in PAM, a case in this thesis work, a sequence of successive pulses would be smeared into one another and the peaks of the pulses will no longer be distinguishable. Thus, results in ISI and may cause erroneous detection, unless a mechanism is devised to compensate the incurred distortion due to nonideal frequency response characteristics of the channel. Here channel equalizers may be used to mitigate the distortion, as discussed in chapter four of this paper.







(c)

Fig. 2.3. *Effects of channel distortion: (a) channel input: (b) channel output: (c) equalizer output*

In addition to linear distortion, signals transmitted through radio channels are subject to other impairments, specifically nonlinear distortion, frequency offset, phase jitters, impulse noise and thermal noise. For mathematical tractability, as the other distortions are usually small and very difficult to correct, the channel model that is adopted in this thesis work is a linear filter with time-varying coefficients that introduces amplitude and delay distortion and adds gaussian noise.

In HF radio channel, time dispersion and, hence, ISI is the result of multiple propagation paths with different path delays. The number of paths and the relative time delays among the paths vary with time, and, for this reason, it is called time-variant multipath channel. The time-variant multipath conditions give rise to a wide variety of frequency response characteristics. Consequently, the channel characteristics can only be described by its statistical parameters. The different statistical functions used to describe the channel characteristics are presented in the third chapter.

For the signal whose lowpass equivalent given by (2.3.15) transmitted over a band-limited channel with a frequency response of $C(f)$, the received signal can be expressed as

$$r_l(t) = \sum_{n=0}^{\infty} I_n h(t - nT) + z(t) \qquad 2.5.1$$

where

$$h(t) = \int_{-\infty}^{\infty} g(\tau)c(t - \tau)d\tau \qquad 2.5.2$$

and $z(t)$ represents the AWGN.

Assuming that the received signal is passed first through a matched filter $H^*(f)$, optimum filter for signal detection, and sampled at a rate $1/T$ samples/s, i.e., at times $t = kT$, $k = 0,1,2,...,$ we obtain

$$y_k = \sum_{n=0}^{\infty} I_n x_{k-n} + v_k, \qquad k = 0,1,... \qquad 2.5.3$$

where $x(t)$ is the pulse representing the response of the receiving matched filter to the input pulse $h(t)$ and $v(t)$ is the response of the receiving filter to the noise.

By setting x_0, arbitrary scale factor, to unity, we obtain

$$y_k = I_k + \sum_{n=1}^{\infty} I_n x_{k-n} + v_k \qquad 2.5.4$$

The term I_k represents the desired information symbol, the second term in (2.5.4) represents the channel induced ISI, and v_k represents additive gaussian noise at the kth sampling instant. For PAM signals, the received signal $y(t)$ can be displayed on the vertical input of an oscilloscope with the horizontal sweep rate set at $1/T$, and results in what is called the ***eye pattern,*** Fig. 2.4. The effect of ISI is, therefore, to cause the eye to close, thereby reducing the margin for additive noise to cause errors.

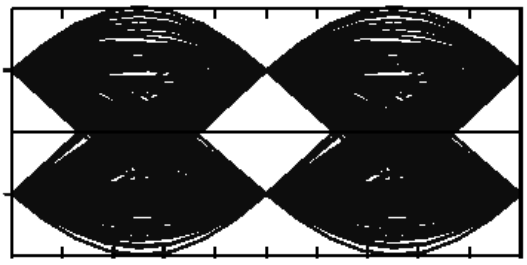


Fig 2.4. *Eye diagram of a binary PAM*

2.6. Discrete-Time Model for a Band-Limited Channel [19]

In dealing with band-limited channels that result in ISI, it is convenient to develop an equivalent discrete-time model for the analog system, as the transmitter sends discrete-time symbols at a rate $1/T$ samples/s and the sampled output of the matched filter at the output is also a discrete-time symbols with samples occurring at a rate $1/T$ per seconds. Consequently, the channel can be modeled by an equivalent discrete-time transversal filter having tap gain coefficients $\{x_k\}$ that spans a time interval of $2LT$ seconds, maximum time-spread of the channel, with the sequence of information symbol $\{I_k\}$ as its input and the discrete-time sequence $\{y_k\}$ given by (2.5.4) output. However, this model has got a difficulty caused by the correlations in the noise sequence at the output of the matched filter. Since it is more convenient to deal with the white noise sequence when simulating the channel and analyzing performance, it is desirable to whiten the noise sequence by further filtering $\{y_k\}$.

Since $x_k = x_{-k}^*$, $X(z)$, the two-sided z-transform of the sampled autocorrelation function $\{x_k\}$ is given as

$$X(z) = \sum_{k=-L}^{L} x_k z^{-k} = F(z)F^*(z^{-1}) \qquad 2.6.1$$

Then an appropriate noise-whitening filter has a z-transform $1/F^*(z^{-1})$. Consequently, passage of the sequence $\{y_k\}$ through the digital filter $1/F^*(z^{-1})$ results in an output sequence $\{u_k\}$

$$u_k = \sum_{k=-L}^{L} f_n I_{k-n} + \eta_k \qquad 2.6.2$$

where $\{\eta_k\}$ is a white gaussian noise and $\{f_k\}$ is a set of tap coefficients of an equivalent discrete-time transversal filter having a transfer function $F(z)$.

2.7. Digital Communication over HF Radio Link

Information networks have arisen from two backgrounds: voice frequency telephony systems and computer networks. The trend is now towards networks that can support both the real-time requirements of voice and video conferencing as well as reliable data transfer. This is being achieved with new network architectures such as Integrated Subscriber Digital Networks (ISDN) and Asynchronous Transfer Mode (ATM) Networks and through Internet technology that can make effective use of voice-band transmission technology for data networking. Mobile computing has arisen from the data transmission capabilities of mobile telephony systems such as GSM and digital-voice satellite systems. These media permit connectivity to the Internet over wide geographical areas at rates up to 9600 bps. It is now common to see laptop computers accessing the e-mail and transferring files and documents over mobile telephones in airport lounges and hotels throughout the world.

HF radio communication has its heritage in long-range voice communications but has been used for point-to-point data transmission for decades. However, HF radio has not been thought of as being suitable for computer networking because of its traditional reliance on manual frequency management and because of the high error rates normally associated with HF transmission.

Amateur packet radio equipments, first developed in 1979, were to progress to running an amateur version of the X.25 packet switched protocol named AX.25. This was extended in 1986 by Phil Karn who implemented TCP/IP on an IBM PC and made it available for amateur radio hobbyists. This initiative enabled radio operators to communicate amongst themselves using the plethora of software applications that rely on the TCP/IP protocols. It also enabled radio interconnection to the global Internet. An example of a manually connected system is ***PACTOR,*** marketed by PAC-Com, that implements a selective repeat Automatic Repeat reQuest (ARQ) scheme operating in a time-division duplex mode. The system not only claims to have performance and throughput improvements over earlier radio telegraphy systems, but also an internal mailbox and functions for file and message handling. ***CLOVER*** is a more ambitious system that employs a four-tone multi-level modulation system. CLOVER uses a low baud rate of 31.25 baud and by using a combination of multiple -level frequency, phase and amplitude

modulation can achieve “in-packet” data rates from 125 to 750 bps over only 500 Hz of HF bandwidth. In addition, CLOVER employs *Reed-Solomon forward error-control coding* and a selective repeat ARQ protocol. A further feature of the system is the inbuilt file transfer protocol that supports 8-bit characters and incorporates data compression. These attributes enable simple binary file transfer. [4, 16, 26]

A third system named ***Globe-Email*** appears to provide a genuine global HF e-mail service that features a high degree of automation. The *Globe-Email* system comprises a network of HF stations around the world, which incorporates a standard HF radio and a low cost Windows-based computer. Network operation is said to be fully automatic; the system adjusts when new stations come on air, or one goes off the air for extended maintenance. From the operator’s perspective the system has the functionality expected in modern e-mail systems. Messages are prepared using an integral text editor to which binary files generated by applications such as word processors or spreadsheets can be attached. The mailer incorporates a control interface to the radio so that when messages are completed, the system automatically connects to a network coast station and forwards the message. Interactive transmission protocols verify connections, verify the veracity of transmission and provide confirmation of delivery. The Globe-Email system also receives messages completely automatically. Globe-Email uses the CLOVER transmission system described earlier but at rates up to 2400 bps - presumably by using several 500 Hz CLOVER waveforms. The marketing information also includes details of the billing rate structure that is based on data actually transferred rather than connect time.

CHAPTER THREE

CHARACTERIZATION AND MODELING OF THE HF COMMUNICATION CHANNEL

The effective design, assessment, and installation of radio network require an accurate characterization of the channel. The channel characteristics vary from one environment to another, and the particular characteristics determine the feasibility of using a proposed communication technique in a given operating environment. Having an accurate channel characterization for each frequency band, including key parameters and detailed mathematical model of the channel, enable the designer or user of the wireless system to predict signal coverage, achievable data rate, and specific performance attributes of alternative signaling and reception schemes.

High frequency (HF) skywave-communication systems exhibit low signal-to-noise (SNR) ratios, and may be subject to slow fading at mid-latitudes and fast fading at high and equatorial latitudes. Furthermore, for modern systems, the radio channel is almost always frequency selective. Until 10 years ago, it was common for the HF user to expect data rates of little more than 75 bit/s and low availabilities. However, with the advent of digital signal processing, data rates have increased significantly to 2400 bit/s, 4800 bit/s, and beyond, even in the standard 3 kHz channel allocation. Data rates on benign skywave channels and on groundwave paths can now reach 64 kbps, albeit sometimes using wider channel bandwidths, and there is an increasing desire to reliably achieve 16 kbps.

The performance of skywave HF communication and broadcast systems is dependent on how well the system design is able to compensate for the propagation channel. For example, an appropriate choice of antenna gain and transmitter power can compensate for free-space losses and for excess attenuation in the *D* region. Of equal importance, however, and especially

pertinent to the design of digital systems, is an appropriate choice of the signaling waveform, since it can, to a greater or lesser extent, compensate for radio-channel distortion. (In this context, the term signaling waveform includes the modulation type, error-correction code, interleaving, and other related issues.) In essence, the waveform aims to compensate for multipath and Doppler effects that can compromise the signal integrity.

Multipath propagation arises because replicas of the transmitted signal arrive at the receiver after reflection from more than one ionospheric layer, and/or after multiple reflections between the ionosphere and the ground. Each signal (or propagation mode) generally arrives with a different time delay, causing either constructive or destructive interference, which, when viewed in the frequency domain, dictates the coherence bandwidth of the channel. Multipath is, of course, a worldwide phenomenon. However, at high and equatorial latitudes, the situation is further complicated because each received "mode" may exhibit time spreading or smearing. Signal dispersion is usually insignificant, unless the bi-static link is operating close to the junction-frequency.

Frequency (Doppler) shifts and frequency-spread distortion can be imposed on the transmitted signal by the temporal variability of the ionosphere, and this defines the coherency time of the channel. At high and equatorial latitudes, Doppler shifts and spreads of many Hertz are very common, and these are also often associated with spread returns, due to ionospheric irregularities.

The complex ionospheric radio-propagation channel is a considerable challenge for the designers of new digital ionospheric-radio systems, and their ultimate success depends critically on a good understanding of the radio-channel multipath and Doppler characteristics. Recent research has focused on the HF channel and, consequently, this thesis work will likewise address this frequency band.

3.1. Characterization of Fading Multipath Channels

When one transmits an extremely short pulse, ideally an impulse, over a time varying multipath HF channel, the received signal might appear as a train of pulses. This is one characterization of multipath medium in which time spread is introduced in the signal transmitted through the channel.

A second characteristic of the channel is due to the time variation in the structure of the medium. As a result of such time variations, the nature of multipath varies with time. That is, if the pulse-sounding experiment is repeated over and over, changes can be observed in the received pulse train, which will include changes in the size of the individual pulses, changes in the relative delay among the pulses, and, quite often, changes in the number of received pulse train. Moreover, the time variations appear to be unpredictable to the user of the channel. Therefore, it is reasonable to characterize the time-variant multipath HF channel *statistically*. [9, 11, 19]

Towards this end, let us examine the effect of the channel on a transmitted signal that is represented in general as

$$s(t) = \mathrm{Re}\left[s_l(t) e^{j2\pi f_c t} \right] \qquad 3.1.1$$

Where $s_l(t)$ is the low-pass equivalent of the band-pass signal, f_c is the carrier frequency of the band-pass signal and "Re" denotes *Real Value.*

It can be assumed that there are multiple propagation paths, and, associated with each path is a propagation delay and an attenuation factor. Both the propagation delays and attenuation factors are time-variant as a result of changes in the structure of the medium. Thus, the received bandpass signal may be expressed in the form

$$x(t) = \sum_n \alpha_n(t) s(t - \tau_n(t)) \qquad 3.1.2$$

where $\alpha_n(t)$ is the attenuation factor for the signal received on the *nth* path, $\tau_n(t)$ is the propagation delay for the *nth* path. Substituting for s(t) from (3.1.1) into (3.1.2) yields the result

$$x(t) = \mathrm{Re}\left\{ \left[\sum_n \alpha_n(t) e^{-j2\pi f_c \tau_n(t)} s_l(t - \tau_n(t)) \right] e^{j2\pi f_c t} \right\} \qquad 3.1.3$$

It is apparent from (3.1.3) that the equivalent lowpass received signal is

$$r_l(t) = \sum_n \alpha_n(t) e^{-j2\pi f_c \tau_n(t)} s_l(t - \tau_n(t)) \qquad 3.1.4$$

Since $r_l(t)$ is the response of an equivalent lowpass channel to the equivalent lowpass signal, $s_l(t)$, it follows that the equivalent lowpass channel is described by the time-variant

$$c(\tau;t) = \sum_n \alpha_n(t) e^{-j2\pi f_c \tau_n(t)} \delta(\tau - \tau_n(t)) \qquad 3.1.5$$

For some channels it is more appropriate to view the received signal as consisting of a continuum of multipath components. In such a case, the received signal $x(t)$ is expressed in the integral form

$$x(t) = \int_{-\infty}^{\infty} \alpha(\tau;t) s(t-\tau) d\tau \qquad 3.1.6$$

where $\alpha(\tau;t)$ denotes the attenuation of the signal component at delay τ and at time instant *t*. Now substituting for $s(t)$ from (3.1.1) into (3.1.6) yields

$$x(t) = \mathrm{Re}\left\{ \left[\int_{-\infty}^{\infty} \alpha(\tau;t) e^{-j2\pi f_c \tau} s_l(t-\tau) d\tau \right] e^{j2\pi f_c t} \right\} \qquad 3.1.7$$

Since the above integral (3.1.7) represents the convolution of $s_l(t)$ with an equivalent lowpass time-variant impulse response $c(\tau;t)$, it follows that

$$c(\tau;t) = \alpha(\tau;t) e^{-j2\pi f_c \tau} \qquad 3.1.8$$

where $c(\tau;t)$ represents the response of the channel at time t due to an impulse applied at time (*t*-τ).

Now let us consider the transmission of unmodulated carrier at frequency f_c. Then $s_1(t) = 1$ for all t, and, hence the received signal for the case of discrete multipath is given as

$$r_l(t) = \sum_n \alpha_n(t) e^{-j2\pi f_c \tau_n(t)}$$

$$r_l(t) = \sum \alpha_n(t) e^{-j\theta_n(t)} \qquad 3.1.9$$

where $\theta_n(t) = 2\pi f_c \tau_n(t)$. Thus, the received signal consists of the sum of a number of time-variant vectors (phasors) having amplitudes $\alpha_n(t)$ and phases $\theta_n(t)$. Note that large dynamic changes in the medium are required for amplitude to change sufficiently to cause significant change in the received signal. On the other hand, the phase will change by 2π radians (*rad)* whenever τ_n changes by $1/f_C$, which is a small number and, hence, the phase can change by 2π rad with relatively small change of the medium. It can also be expected that the delay associated with the different signal path to change at different rate and in unpredictable (random) manner. This implies that the received signal can be modeled as a *random process*. When there are large numbers of paths, the *central limit theorem* can be applied. That is, the received signal can be modeled as a complex valued Gaussian random process. [18, 19]

The multipath propagation model for the channel embodied for the received signal results in signal fading. The fading phenomenon is primarily a result of the time variation in the phases. That is, the randomly time-variant phases at time results in the vector adding *destructively* or *constructively*. Thus, amplitude variations in the received signal, termed ***signal fading***, would happen due to the time-variant nature of the channel.

3.1.1. Channel Correlation Functions [14, 19]

In order to be able to assess the performance capabilities of the HF channel, it needs to have a convenient measure of the time dispersion, or multipath delay spread of the channel. The simplest measure of the multipath delay is the overall span of path delays (i.e., earliest arrival to latest arrival). And also it is necessary to properly measure the apparently random phase and amplitude fluctuations of the received signal caused by the channel.

In order to assess the effectiveness of some signal design and the corresponding performance of a receiving system operating on a given multipath fading channel, it is important to be able to mathematically characterize the behavior of the channel. Because, to the observer or the user, the variations in the received signal are not predictable, but apparently random, the variations are

best described in statistical terms. Most commonly correlation functions and power spectral density functions are utilized to represent the characteristics of HF multipath fading channel.

3.1.1.1. Correlation Properties in Delay Variable

Assuming that the effects of the transmission medium are sufficiently random, and the number of multipath signal components sufficiently large, that it can be invoked the central limit theorem. It can then be assumed that the overall impulse response of the channel is accurately represented by a complex Gaussian process $c(\tau;t)$, given in equation (3.1.8), where inclusion of the variable *t* in the argument indicate that in general the channel impulse response is time-variant.

The autocorrelation of the observed impulse response at two different delays and two different times is given by

$$\phi_c(\tau_1;\tau_2;\Delta t) = E[c^*(\tau_1;t)c(\tau_2;t+\Delta t)] \qquad 3.1.1.1$$

As a way of modeling HF channel multipath fading, it is appropriate to assume it as being wide-sense stationary uncorrelated scattering (WSSUS). This assumption leads to several interesting and useful conclusions. The physical meaning of the assumption, which is valid for most radio transmission channels, is that the signal variations on paths arriving at different delays are uncorrelated and the correlation properties of the channel are stationary. In mathematical terms, the assumption results in simplifying (3.1.1.1) into

$$E[c*(\tau_1;t)c(\tau_2;t+\Delta t)] = \phi_c(\tau_1;\Delta t)\delta(\tau_2-\tau_1) \qquad 3.1.1.2$$

If we let $\Delta t=0$, the resulting autocorrelation function $\phi_c(\tau;0)=\phi_c(\tau)$ is simply the average power output of the channel as a function of time delay τ. For this reason, $\phi_c(\tau)$ is called the ***multipath intensity profile*** or the ***delay power spectrum*** of the channel. In general, $\phi_c(\tau;\Delta t)$ gives the average power output as a function of the time delay τ and the difference Δt is the observation time.

In practice, the delay power spectrum is measured by transmitting very narrow pulse or, equivalently, a wideband signal and cross-correlating the received signal with the delay version of itself. Typically, the measured function $\phi_c(\tau)$ may appear as shown in Fig. 3.1. The range of values of τ over which $\phi_c(\tau)$ is essentially nonzero is called the ***multipath spread*** of the channel and is denoted by $\boldsymbol{T_m}$.

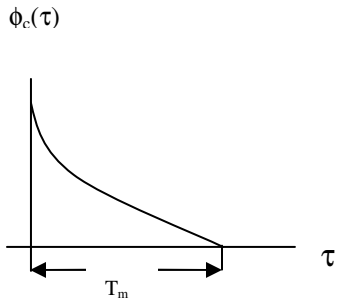


Fig. 3.1. *Multipath Intensity Profile*

3.1.1.2. Multipath Delay Characteristics in the Frequency Domain

Given a channel with impulse response $c(\tau;t)$ and using the WSSUS, the frequency response is defined as the Fourier transform of this function on the argument τ, which is written as

$$C(f;t) = \int_{-\infty}^{\infty} c(\tau,t)e^{-2\pi f\tau}d\tau \qquad 3.1.1.3$$

Now, given the assumption of WSSUS model, the channel impulse response is a wide- sense stationary zero-mean gaussian process in the time variable *t*. Therefore, the frequency response *C(f;t)*, being obtained as a linear operation on $c(\tau;t)$, is also a wide-sense stationary zero-mean gaussian process in *t*. The time domain response shows the arrival of the multipaths, while the frequency response exhibits amplitude variations from one frequency to another. The cause of these variations is the multipath structure of the channel, which causes interference and signal enhancement at certain frequencies but causes destructive interference and deep fades at other frequencies. This channel characteristic is referred to as ***frequency selective multipath channel***.

To characterize these variations statistically, the correlation between values of the frequency response taken at various frequency spacings can be computed. Under the assumption that the channel is wide-sense stationary, the *autocorrelation* in frequency domain is defined as

$$\begin{aligned}\Phi_c(f_1;f_2;\Delta t) &= E[C^*(f_1;t)C(f_2;t+\Delta t)] \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} E[c^*(\tau_1;t)c(\tau_2;t+\Delta t)]e^{j2\pi(f_1\tau_1-f_2\tau_2)}d\tau_1 d\tau_2 \qquad 3.1.1.4 \\ &= \int_{-\infty}^{\infty}\phi_c(\tau_1;\Delta t)e^{-j2\pi\Delta f\tau_1}d\tau_1 = \Phi_c(\Delta f;\Delta t)\end{aligned}$$

where $\Delta f=f_2-f_1$. The new function $\Phi_c(\Delta f;\Delta t)$ is referred to as the spaced-time, spaced-frequency correlation function of the channel.

For a slowly time-varying channel, the value of $\Phi_c(\Delta f;\Delta t)$ calculated with observation times separated by Δt is the same as that found with no time separation and thus we have

$$\Phi_c(\Delta f) = \int_{-\infty}^{\infty}\phi_c(\tau)e^{-j2\pi\Delta f\tau}d\tau \qquad 3.1.1.5$$

which can be determined by transmitting low frequencies Δf apart and determining the correlation between the received signals. The inverse Fourier transform of this function is the delay power spectrum and this relationship is depicted in Fig. 3.2

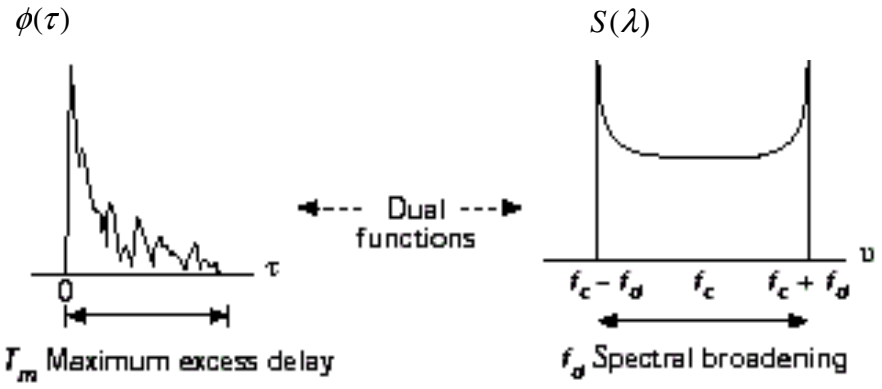


Fig. 3.2. *Relationships among the channel correlation functions and power density functions.*

Since $\Phi_c(\Delta f)$ is an autocorrelation in the frequency domain, it provide us with a measure of the frequency coherence of the channel. As a result of the Fourier transform relationship given in (3.1.1.5), the reciprocal of the multipath spread is a measure of the coherence bandwidth of the channel. That is,

$$(\Delta f)_c \approx \frac{1}{T_m} \qquad 3.1.1.6$$

where $(\Delta f)_c$ denotes the coherence bandwidth. Thus, two sinusoids with frequency separation greater than $(\Delta f)_c$ are affected differently by the channel. When an information-bearing signal is transmitted through the channel, if $(\Delta f)_c$ is small compared to the bandwidth of the transmitted signal, the channel is said to be *frequency-selective*. In this case the channel is severely distorted by the channel. On the other hand, if $(\Delta f)_c$ is large in comparison to the bandwidth of the transmitted signal, the channel is said to be *frequency-nonselective*.

3.1.1.3. Correlation Property in the Time Variable

Signal dispersion and coherence bandwidth, parameters that describe the channel's time-spreading properties in a local area do not offer information about the time-varying nature of the channel caused by ionospheric turbulence. The time variations in the channel are evidenced as a Doppler broadening and, perhaps, in addition as a Doppler shift of a spectral line. In order to

relate the Doppler effects to time variations of the channel, we define the Fourier transform of $\Phi_c(\Delta f;\Delta t)$ with respect to the variable Δt to be the function $S_c(\Delta f;\lambda)$. That is

$$S_c(\Delta f;\lambda) = \int_{-\infty}^{\infty} \Phi_c(\Delta f;\Delta t) e^{j2\pi\lambda\Delta t} d\Delta t \qquad 3.1.1.7$$

With Δf set to zero and $S_c(\Delta f;\lambda) = S_c(\lambda)$, the above relation becomes

$$S_c(\lambda) = \int_{-\infty}^{\infty} \Phi_c(\Delta t) e^{j2\pi\lambda\Delta t} d\Delta t \qquad 3.1.1.8$$

The function $S_c(\lambda)$ is a power spectrum that gives the signal intensity as a function of the Doppler frequency λ. Hence, we call $S_c(\lambda)$ the ***Doppler Power Spectrum*** of the channel. It can clearly be noted that for time-invariant channel there will not be any spectral broadening observed in the transmission of a pure frequency tone.

The range of values of λ over which $S_c(\lambda)$ is essentially nonzero is called the Doppler spread B_d of the channel. Since $S_c(\lambda)$ is related to $\Phi_c(\Delta t)$ by the Fourier transform, the reciprocal of B_d is the measure of the coherence time of the channel. Therefore, a slowly changing channel has a large coherence time or, equivalently a smaller Doppler spread.

3.1.1.4. The Scattering Function

The Fourier transform of $\phi_c(\tau;\Delta t)$ in the Δt variable is called the *scattering function*. That is,

$$S_c(\tau;\lambda) = \int \phi_c(\tau;\Delta t) e^{j2\pi\lambda\Delta t} d\Delta t \qquad 3.1.1.9$$

It represents the rate of variation of the channel at different delays. It provides us with a measure of the average power output of the channel as a function of time delay and the Doppler frequency. To measure the scattering function the received signal in individual taps of a tapped delay line is analyzed in frequency domain.

3.2. HF Channel Model

Models of the signal distortions caused by the ionosphere are important because they facilitate the design and testing of new modems and radio receivers via their incorporation in channel simulators. Given an input signal, $x(t)$, the output signal, $y(t)$, after passage through a radio channel with an impulse response that is described by $c(t;\tau)$, is often represented by

$$y(t) = x(t) * c(t;\tau) + n(t) \qquad 3.2.1$$

where t is the time variable, τ is the delay variable, $n(t)$ is a noise signal, and * indicates convolution. Therefore, if the time-varying channel impulse response is available (either measured or simulated), it may be convolved with the input signal to provide the channel-modulated output signal. This process may be conveniently implemented as a tapped delay line, with time-varying tap gain functions.

There are several probability distributions that can be considered in attempting to model the statistical characteristic of fading channel. When there are a large number of scatterers in the channel that contribute to the signal at the receiver, as in the case of ionospheric propagation, application of central limit theorem leads to a Gaussian process model. If the process is a zero-mean, then the envelop of the channel response at any time instant has a ***Rayleigh Distribution*** and the phase is ***uniformly distributed*** in the interval (0, 2π). [3, 9, 19]

Based on channel sounding measurements and theoretical considerations, statistical channel model can be constructed and implemented as hard- or software simulation. Accurate and efficient computer based channel modeling is important to validate and/or compare different digital scheme under realistic channel condition and to optimize modem design. Since all relevant transmitter and receiver stages can be modeled to operate in the complex-equivalent lowpass domain, link simulation simply consists of inserting parameterizable software channel and noise simulation modules between the modem transmitter and receiver whereby a flexible system simulation is created. In this research work, bandwidth-efficient transmission of linearly modulated signal over linear selective Rayleigh channel model is considered to represent the HF channel.

3.2.1. The Watterson Model [28]

The Watterson model has been the standard representation of the HF channel used in simulators for many years. This model assumes that the channel fading is described by a Rayleigh amplitude distribution, and that the Doppler spread on each propagation mode has a Gaussian power spectrum. The model does not define a shape for delay spread, and most practical implementations assume that each mode exhibits no delay spread.

The Watterson model considers the channel as an ideal tapped delay line, where at each tap, the delayed signal is modulated by a tap gain function, $G_i(t)$. In general, each tap gain function is defined by

$$G_i(t) = \tilde{G}_{ia}(t)e^{j2\pi v_{id}t} + \tilde{G}_{ib}(t)e^{j2\pi v_{ib}t} \qquad 3.2.2$$

The subscripts *a* and *b* indicate the two possible magnetoionic components, the exponentials allow Doppler shifts to be added to the signal, and the tildes (~) indicate that the *G* terms are sampled functions of two independent complex Gaussian ergodic random processes, each with zero mean values, and independent real and imaginary components with equal RMS values. Such complex number sequences exhibit Rayleigh fading. The tap gain functions are also filtered, to produce a Gaussian Doppler spread in the signal's power spectrum. Generally, one tap is used for each propagation mode. In practice, only a single magnetoionic component is simulated, and a limited number of propagation modes (taps) are allowed (*generally four or five*). Furthermore, the facility to include a specular mode is usually provided, which can be used to simulate a line-of-sight or surface wave.

3.2.2. The ITS Model

The Watterson model was long considered too simple for applications pertaining to the high and equatorial ionosphere: for example, it is well known that at high latitudes, the Doppler spectrum is often not Gaussian. To overcome these limitations, a more-complicated ionospheric channel model has been proposed by the Institute of Telecommunication Sciences (ITS) in Boulder, Colorado, USA. This was proposed as a wideband model, but can be used as a narrowband model.

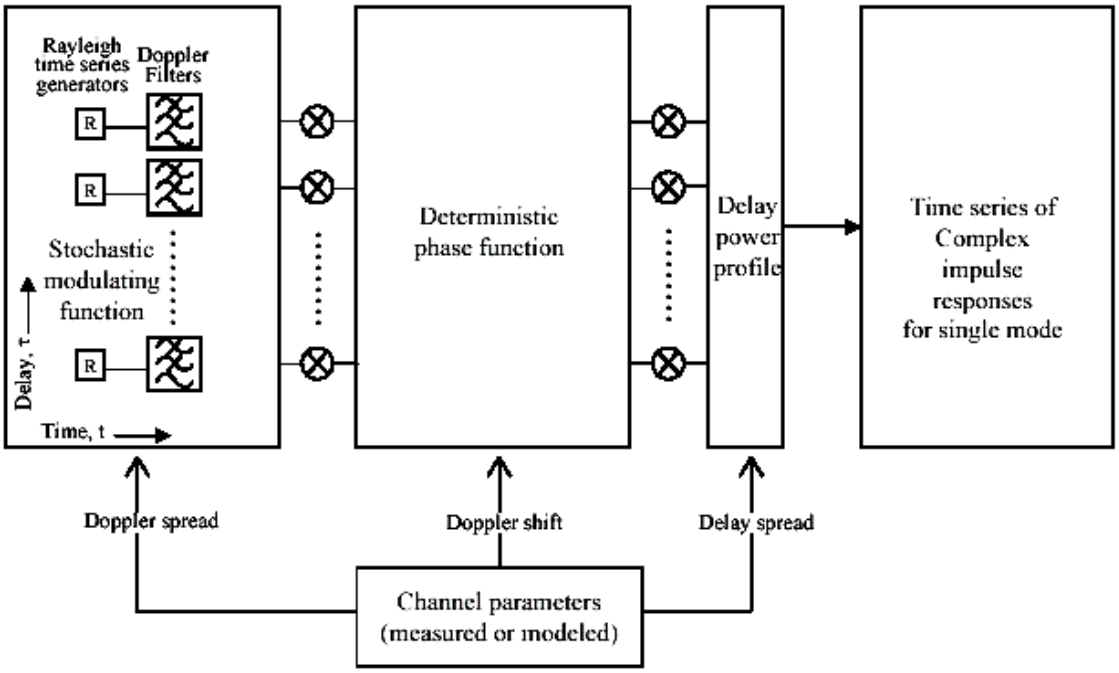


Fig. 3.3. *A block diagram of impulse-response reconstruction*

In the ITS model, the overall channel impulse response is defined as a sum of the impulse responses of each propagation mode, and is a function of time, *t*, and delay, τ (equation 3.2.3). The ITS model represents the impulse response of each mode as the product of three terms: a ***stochastic modulating function,*** $\psi_n(t,\tau)$, which is defined by the Doppler spread and the spectral shape; a ***deterministic phase function,*** $D_n(t,\tau)$, defined by the Doppler shift and the rate of change of the Doppler shift with respect to delay; and the square root of a delay power profile,

$\phi_{cn}(\tau,t)$ defined by the propagation mode's time of flight, delay spread, and maximum power (equation 3.2.2)

$$c(t;\tau) = \sum_n c_n(t;\tau) \qquad 3.2.3$$

and

$$c(t;\tau) = \sum_n \sqrt{\phi_{cn}} D_n(t;\tau)\psi_n(t;\tau) \qquad 3.2.4$$

where the summation n is over the number of modes. The block diagram of the model is shown in Fig 3.3.

3.2.2.1. Stochastic Modulation Function

In order to model the fading of the impulse responses, the stochastic modulating function (SMF), $\psi_n(t,\tau)$, is constructed from an ensemble of time series of random complex numbers. At each delay offset, two independent random-number sequences are constructed, representing the real and imaginary parts of the complex time series. Each is independent, white, and exhibits a Gaussian amplitude distribution. The corresponding complex random-number sequences have an amplitude distribution that conforms to Rayleigh statistics. To constrain the width of the random-number power spectrum – to simulate the required Doppler spread – the random-number sequence is convolved with a filter (Gaussian or Lorentzian), with a power spectrum width corresponding to the Doppler spread (set by the user) of the mode. Independent sequences are produced at each delay offset, thus producing a set of sequences that are uncorrelated in delay but that share the same fading characteristics.

3.2.2.2. Deterministic Phase Function

The Doppler shift of the mode is implemented by multiplying the stochastic modulating function by a deterministic phase function (DPF):

$$D(t;\tau) = e^{i2\pi[f_s + m(\tau-\tau_c)]t} \qquad 3.2.5$$

where t is the time variable, τ is the delay variable, τ_c is the delay of the mode's peak, m is the rate of change of Doppler shift with respect to τ, and f_s is the Doppler shift at $\tau = \tau_c$. This

function allows for the inclusion of slant modes, i.e., modes with a Doppler shift that varies with delay.

3.2.2.3 Delay Power Profile

The delay power profile (DPP) is determined by the shape of the mode in the delay dimension, and is defined by the position (in delay) of its peak, τ_c, and the positions of two points of intersection with a threshold, τ_L and τ_U (Fig. 3.4). The profile is defined by

$$\phi_c(\tau) = Ae^{[\alpha(\ln z + 1 - z)]} \qquad 3.2.6$$

where

$$z = \left\{ (\tau - \tau_L) \Big/ (\tau_c - \tau_L) \right\} > 0$$

and A is the peak power of the mode. The parameters α and τ_L control the width and symmetry of the profile. They depend upon the delay spread and the threshold level, and are calculated by an iterative method.

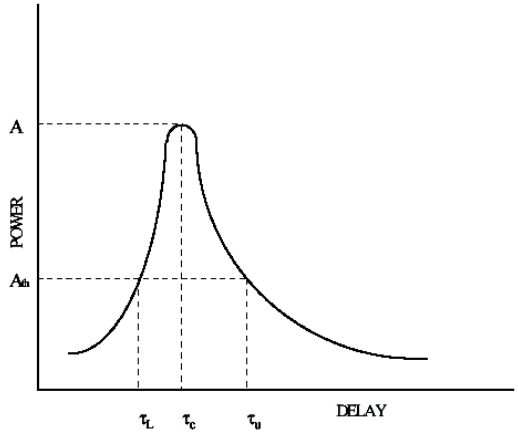


Fig. 3.4. *The delay power profile*

CHAPTER FOUR

HF CHANNEL EQUALIZER

In the study of communication systems the classical (ideal) additive white Gaussian noise (AWGN) channel, with statistically independent Gaussian noise samples corrupting data samples free of intersymbol interference (ISI), is the usual starting point for understanding basic performance relationships. The primary source of performance degradation is thermal noise generated in the receiver. Often, external interference received by the antenna is more significant than thermal noise. This external interference can sometimes be characterized as having a broadband spectrum and is quantified by a parameter called antenna temperature. The thermal noise usually has a flat power spectral density over the signal band and a zero-mean Gaussian voltage probability density function (PDF). When modeling practical systems, the next step is the introduction of band limiting filters. The filter in the transmitter usually serves to satisfy some regulatory requirement on spectral containment. The filter in the receiver often serves the purpose of a classical “matched filter” to the signal bandwidth. Due to the band-limiting and phase-distortion properties of filters, special signal design and equalization techniques may be required to mitigate the filter-induced ISI.

In practical digital communication systems that are designed to transmit at high speed through band-limited channel, the frequency response of the channel is not known with sufficient precision to design optimum filter for modulation and demodulation. For example, in digital communication over HF radio channel, the communication channel exhibit time-variant frequency response characteristics as signal propagation takes place in the atmosphere and near the ground whose nature is highly affected by the climatic condition of the environment. Such channels are also characterized by a number of multipath rays with distinct propagation delay. For such channel, it is not possible to design optimum fixed demodulation filter.

Most radio channels and particularly HF channel, as described in the previous chapter, are more or less time-variant and frequency selective. The frequency selectivity, which arises from

multipath propagation (time spreading of signal) results in channel induced Inter-Symbol Interference (ISI), which, if left uncompensated, causes high error rates. And also the ionospheric turbulence, a continuous motion of the ions in the ionospheric layers that reflect the signals transmitted in the HF frequency band, causes distortion in both signal amplitude and phase, which leads to independent Doppler shift/broadening on each propagation mode. As a result of this time variation in the characteristic of the media, the nature of the multipath varies with time. This *time-variant multipath propagation* results in ***signal fading*** affecting the communication.

4.1. Mitigation Methods [12, 13, 23]

If the channel introduces signal distortion as a result of fading, the system performance can exhibit an irreducible error rate; when larger than the desired error rate, no amount of SNR will help to achieve the desired level of performance. In such cases, the general approach for improving performance is to use some form of mitigation to remove or reduce the distortion. The mitigation method depends on whether the distortion is caused by frequency-selective or fast fading.

In Table 4.1, different mitigation techniques for combating the effects of both signal distortion and loss in SNR are listed. The mitigation approach to be used should follow two basic steps: first, provide distortion mitigation; next, provide diversity.

4.1.1. Mitigation to Combat Frequency-Selective Distortion

A channel is referred to as frequency-selective when the coherence bandwidth of the channel is less than the signal bandwidth. Here, all of the signal's spectral components will be affected by the channel in a different manner. In order to avoid ISI distortion caused by frequency-selective fading, the channel must be made to exhibit flat fading by ensuring that the coherence bandwidth exceeds the signaling rate. Some of the methods used in most practical systems are discussed below.

Channel Equalization [21,27]

Equalization can compensate for the channel-induced ISI that is seen in frequency-selective fading. The process of equalizing the ISI involves some method of gathering the dispersed symbol energy back together into its original time interval. In effect, equalization involves insertion of a filter to make the combination of channel and filter yields a flat response with linear phase. The phase linearity is achieved by making the equalizer filter the complex conjugate of the time reverse of the dispersed pulse. Because in an HF digital radio system the channel response varies with time, the equalizer filter must also change or adapt to the time-varying channel. Such equalizer filters are therefore called ***adaptive equalizers***. An equalizer accomplishes more than distortion mitigation; it also provides diversity. Since distortion mitigation is achieved by gathering the dispersed symbol's energy back into the symbol's original time interval so that it doesn't hamper the detection of other symbols, the equalizer is simultaneously providing each received symbol with energy that would otherwise be lost. There are different types of equalizers, such as linear, decision feedback and maximum likelihood sequence estimation. Design and implementation of linear equalizer, an equalizer selected in which whose performance is to be thoroughly analyzed in this thesis work, is described in detail in section 4.2 of this chapter.

Spread-spectrum technique

Spread-spectrum techniques can be used to mitigate frequency- selective ISI distortion because the hallmark of any spread-spectrum system is its capability to reject interference, and ISI is a type of interference. The spread-spectrum system effectively eliminates the multipath interference by virtue of its code-correlation receiver. Even though channel- induced ISI is typically transparent to DS/SS (Direct-Sequence Spread Spectrum) systems, such systems suffer from the loss in energy contained in all the multipath components not seen by the receiver. The need to gather up this lost energy belonging to the received chip was the motivation for developing the Rake Receiver. The Rake receiver dedicates a separate correlator to each multipath component (finger). It is able to coherently add the energy from each finger by selectively delaying them (the earliest component gets the longest delay) so that they can all be coherently combined.

Frequency-hopping spread spectrum (FH/SS) can as well be used to mitigate the distortion due to frequency-selective fading, provided the hopping rate is at least equal to the symbol rate. Compared to DS/SS, mitigation takes place through a different mechanism. FH receivers avoid multi-path losses by rapid changes in the transmitter frequency band, thus avoiding the interference by changing the receiver band position before the arrival of the multipath signal.

Orthogonal Frequency-Division Multiplexing (OFDM)

Orthogonal frequency-division multiplexing (OFDM) can be used in frequency-selective fading channels to avoid the use of an equalizer by lengthening the symbol duration. The signal band is partitioned into multiple sub-bands, each exhibiting a lower symbol rate than the original band. The sub-bands are then transmitted on multiple orthogonal carriers. The goal is to reduce the symbol rate (signaling rate) on each carrier to be less than the channel's coherence bandwidth.

Pilot Signal

Pilot signal is the name given to a signal intended to facilitate the coherent detection of waveforms. Pilot signals can be implemented in the frequency domain as an in-band tone, or in the time domain as a pilot sequence which can also provide information about the channel state and thus improve performance in fading.

4.1.2. Mitigation to Combat Fast-Fading Distortion

A channel is referred to as fast fading if the symbol rate, $1/T_s$, or the signal bandwidth, W, is less than the fading rate, Doppler spread of the channel. Conversely, a channel is referred to as slow fading if the signaling rate is greater than the fading rate. Doppler spreading, the channel fading rate, therefore, sets a *lower limit* on the signaling rate that can be used without suffering fast fading distortion. For HF communication systems, when teletype or Morse code messages were transmitted at a low data rate, the channels were often fast fading. In order to avoid signal distortion caused by fast fading, the channel must be made to exhibit slow fading by ensuring that the signaling rate exceeds the channel fading rate. For fast fading distortion, the following methods can be used to reduce the error rate.

- Implementing a robust modulation (non-coherent or differentially coherent) that does not require phase tracking, and reduce the detector integration time.

- Increase the symbol rate to be greater than the fading rate, by adding signal redundancy.

- Error-correction coding and interleaving can provide mitigation, because instead of providing more signal energy, a code reduces the required SNR. For a given SNR, with coding present, the error floor will be lowered compared to the uncoded case.

- An interesting filtering technique can provide mitigation in the event of fast-fading distortion and frequency-selective distortion occurring simultaneously. The frequency-selective distortion can be mitigated by the use of an OFDM signal set. Fast fading, however, will typically degrade conventional OFDM because the Doppler spreading corrupts the orthogonality of the OFDM sub-carriers. A polyphase filtering technique is used to provide time-domain shaping and duration extension to reduce the spectral sidelobes of the signal set, and thus help preserve its orthogonality. The process introduces known ISI and adjacent channel interference (ACI), which are then removed by a post-processing equalizer and canceling filter.

4.1.3. Mitigation to Combat Loss in SNR

After implementing some form of mitigation to combat the possible distortion (frequency-selective or fast fading), the next step is to use some form of diversity to improve the error-performance of the receiver. The term “diversity” is used to denote the various methods available for providing the receiver with uncorrelated renditions of the signal. Uncorrelated is the important feature here, since it would not help the receiver to have additional copies of the signal if the copies were all equally poor.

Listed below are some of the ways in which diversity can be implemented:

Time diversity — Transmit the signal on a number of different time slots with time separation of at least the coherence time. Interleaving, often used with error correction coding, is a form of time diversity.

Frequency diversity — Transmit the signal on a number of different carriers with frequency separation of at least coherence bandwidth. Bandwidth expansion is a form of frequency diversity. The signal bandwidth is expanded to be greater than the coherence bandwidth, thus providing the receiver with several independently fading signal replicas.

Spread spectrum is a form of bandwidth expansion that excels at rejecting interfering signals. In the case of direct-sequence spread spectrum (DS/SS), it was shown earlier that multipath components are rejected if they are delayed by more than one chip duration.

Error correction coding coupled with interleaving is probably the most prevalent of the mitigation schemes used to provide improved performance in a fading environment.

To combat distortion	To combat loss in SNR
Frequency selective distortion • Adaptive equalization • Spread spectrum • Orthogonal FDM (OFDM) • Pilot signal	**Flat fading and slow fading** • Some type of diversity to get additional uncorrelated estimate of signal • Error correcting coding
Fast fading distortion • Robust modulation • Signal redundancy to increase signaling rate • Coding and interleaving	**Diversity types** • Time • Frequency • Spatial • Polarization

Table 4.1. *Mitigation techniques*

4.2. Channel Equalizer [5, 14, 19]

As described in chapters two and three, multipath has the harmful effect of causing ISI, but at the same time the signals arriving from different paths are exposed to different fading pattern. The multipath signals can be regarded as a form of diversity, and a smart receiver can use this diversity to improve its performance. As the symbol signaling rate is increased in a multipath channel, the received symbol increasingly flow into one another, (ISI), and this places an upper limit on the rate at which symbols can be transmitted. For data rate beyond the multipath spread of the channel, the channel becomes frequency selective. In such a channel a null may occur in the passband of the channel, which causes performance degradation, and the received waveform is continually changing thereby the matched filtering with the pulse shaping filter does not provide optimum filtering. The traditional method of compensating for ISI is to equalize the channel impairments by applying a filter at the receiver.

4.2.1. Maximum Likelihood Sequence Estimator (MSLE)

The MLSE is the optimum receiver in the presence of ISI. Given the estimate of the channel impulse response, an MLSE receiver uses a trellis diagram with Viterbi algorithm to obtain maximum-likelihood estimates of the transmitted symbol.

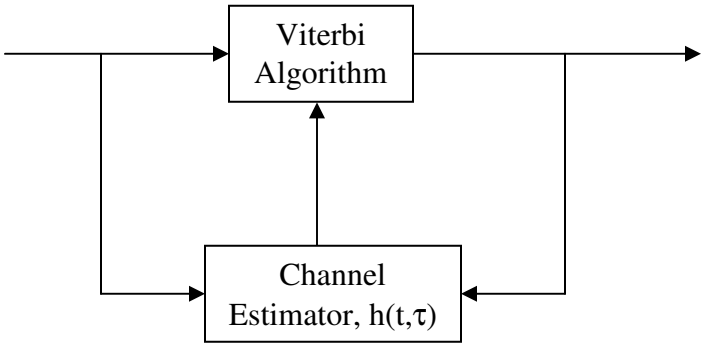


Fig 4.1. *MLSE equalizer block diagram*

In the adaptive MLSE receiver shown in Fig. 4.1, the sampled channel impulse response is measured with the adaptive channel estimator. Given the sampled channel response, $y(n)$, the sequence of the sampled received signal is compared with all possible sequences. The maximum-likelihood procedure is to determine the squared distance between the sequence of the sampled received signal and all possible received sequences and to determine the transmitted symbol sequence with minimal distance. The Viterbi algorithm is used as computationally optimal and efficient solution for this search.

The MLSE is the optimal method of canceling; however, the complexity of this receiver grows exponentially with the length of the channel impulse response. For this reason, an MLSE is an attractive option for channels with short impulse response.

4.2.2. Linear Equalizer

The MLSE for a channel with ISI has a computational complexity that grows *exponentially* with the length of the channel time dispersion. If the size of the symbol alphabet is M and the number of interfering symbols contributing to ISI is L, the Viterbi algorithm computes M^{L+1} matrices for each new received symbol. In most channels of practical interest, such a large computational complexity is prohibitively expensive to implement.

The simplest forms of equalizers, with less computational efficiency but sub-optimal, are linear equalizers or finite impulse response (FIR) equalizers. Generally they are realized in the form of a taped delay line (TDL) as shown in Fig. 4.2. The taped delay line consists of standard delay elements and multipliers that multiply the content of the corresponding delay element with tap-weight, a factor uniquely assigned to a certain number of delay steps. The delay introduced to the signal by the equalizer as a whole depends on the position of the reference tap, the one with the highest tap gain and it corresponds the desired ISI-free sample value.

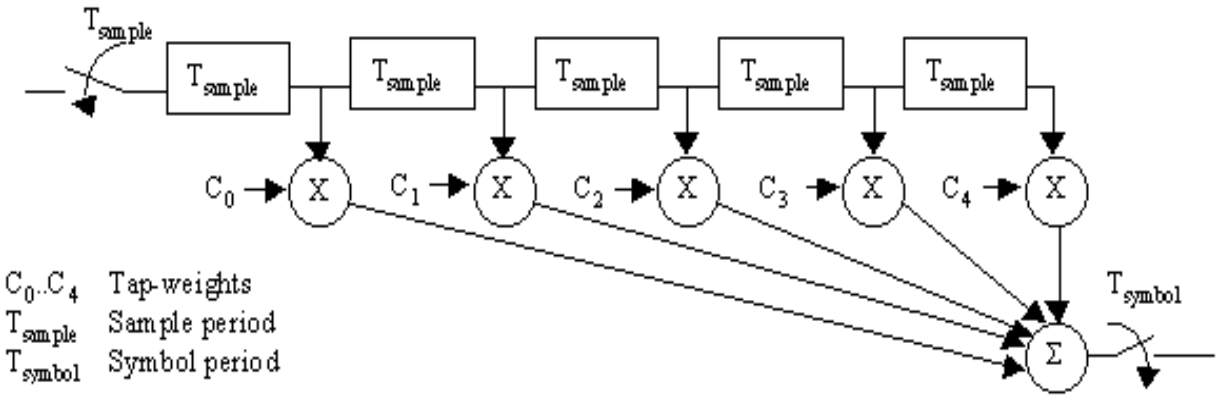


Fig. 4.2. *Five-tap linear transversal filter*

In implementing equalizers, the incoming signal is sampled and quantized at the receivers' front end passed through the equalizer. If the input to the equalizer is sampled in time intervals equal to the symbol period, the system is called a symbol spaced (SS) equalizer while if the sampling period is smaller than the symbol period, it is called a fractionally spaced (FS) equalizer.

The relationship between equalizer input sequence and its output, the estimate of the information sequence, of the *kth* symbol may be expressed as

$$\hat{I}_k = \sum_{j=-K}^{K} c_j v_{k-j} \qquad 4.2.1$$

where $v_k = \sum_{n=0}^{L} f_n I_{k-n} + \eta_k$, with $\{f_n\}$ being a set of channel tap coefficients of the equivalent discrete-time transversal filter and $\{\eta_k\}$ a white gaussian noise sequence added by the channel to the transmitted information sequence $\{I_k\}$. And $\{c_j\}$ are $2K+1$ complex-valued tap coefficients of the equalizer filter.[1]

The estimate $\hat{I}_k$ is quantized to the nearest (in distance) information symbol to form the decision $\tilde{I}$. If $\tilde{I}$ is not identical to the information symbol, an error has been made. Considerable research has been performed on the criterion for optimizing the filter coefficients. Since the most meaningful measure of performance for a digital communication system is the average probability of error, it is desirable to choose the coefficients to minimize this performance index. However, the probability of error is highly nonlinear function of the filter coefficients.

Consequently, this performance index for optimizing the tap-weight coefficients of the equalizer is impractical. Hence, other sub-optimal criteria are used in most practical receiver designs. In this thesis work, two widely used criteria in optimizing equalizer coefficients are considered – ***peak distortion*** and ***mean square error criteria.***

4.2.2.1. Peak Distortion Criterion

The peak distortion is simply defined as the worst-case intersymbol interference at the output of the equalizer. Therefore, the peak distortion criterion is optimizing equalizer coefficients subject to minimization of this performance index.

We observe that the cascade of the discrete-time filter channel model having an impulse response $\{f_n\}$ and an equalizer having an impulse response $\{c_n\}$ can be represented by a single equivalent filter having the impulse response:

$$q_n = \sum_{j=-\infty}^{\infty} c_j f_{n-j} \qquad 4.2.3$$

That is, $\{q_n\}$ is simply the convolution of $\{c_n\}$ and $\{f_n\}$. The equalizer is assumed to have an infinite number of taps. Its output at the kth sampling instant can be expressed in the form

$$\hat{I}_k = q_0 I_k + \sum_{n \neq k} I_n q_{n-k} + \sum_{j=-\infty}^{\infty} c_j \eta_{k-j} \qquad 4.2.4$$

The first term represent a scaled version of the desired symbol, and the second term is undesired intersymbol interference. The peak value of this interference, peak distortion, is

$$D(\bar{c}) = \sum_{\substack{n=-\infty \\ n \neq 0}}^{\infty} |q_n|$$

$$= \sum_{\substack{n=-\infty \\ n \neq 0}}^{\infty} \left| \sum_{j=-\infty}^{\infty} c_j f_{n-j} \right| \qquad 4.2.5$$

Thus, $D(\bar{c})$ is a function of the equalizer tap weights, the vector $\bar{c}$. With equalizer having an infinite number of taps, it is possible to select the tap weights so that $D(\bar{c}) = 0, i.e., q_n = 0$, for all

n except $n = 0$. That is, the intersymbol interference can be completely eliminated. The values of the equalizer tap weights for accomplishing this goal are determined from the condition:

$$q_n = \sum_{j=-\infty}^{\infty} c_j f_{n-j} = \begin{cases} 1, & (n = 0) \\ 0, & (n \neq 0) \end{cases} \qquad 4.2.6$$

By taking the z-transform of the above equation, we obtain

$$Q(z) = C(z)F(z) = 1 \qquad 4.2.7$$

Thus, the discrete transfer function of the equalizer filter is expressed as

$$C(z) = \frac{1}{F(z)} \qquad 4.2.8$$

Note that, complete elimination of the intersymbol interference requires the use of the inverse filter to the linear filter channel model, $F(z)$. As this equalizer filter tries to force the equalized channel impulse response to be zero at all sampling instants, it is said to be ***Zero-Forcing (ZF) Equalizer*** **.**

As stated above, it is possible to completely eliminate all ISI that affect the transmitted signal. However, as realizable equalizers have a finite length, the optimal performance can only be approximated with performance increasing by increasing number of taps. As the time-span in the impulse response has zero crossings at the sampling instant, but does not define what happens between them, already slight deviations in sampling phase or slight sampler phase jitters may lead to considerable loss in performance. Moreover ISI is only minimized by ZF-equalizer if there was already an opening in the eye diagram of the input signal, that is the ISI is not severe enough to close the eye.

Another serious problem of the ZF-equalizer is that it neglects the effect of noise altogether. This effect of noise enhancement increases with increasing attenuation distortion as the higher the difference in loss between the lower and the higher band-edge, the higher is also the difference in equalizer gain. Due to these properties the linear ZF-equalizer can only be used for sub-optimal structure where the main design constraint is the complexity of the equalizer and the computation of the initial tap settings.

4.2.2.2. Mean Square Error (MSE) Criterion

In the MSE criterion, the tap weight coefficients of the equalizer are adjusted to minimize the mean square values of the error at the output of the equalizer. Error in this sense is the difference between the sent symbol and the equalizer output. That is,

$$\varepsilon_k = I_k - \hat{I}_k \qquad 4.2.9$$

where I_k is the information symbol transmitted in the *kth* signaling interval and $\hat{I}_k$ is the estimate of that symbol at the output of the equalizer. Here both ISI and additive noise compose this error; therefore, the signal to distortion pulse noise ratio at the equalizer output is maximized. In this equalizer the total SNR is optimized, but there is no defined treatment of the noise spectrum.

When the information symbols $\{I_k\}$ are complex-valued, the performance index for the MSE criterion, denoted by J, is defined as

$$\begin{aligned} J &= E\left|\varepsilon_k\right|^2 \\ &= E\left|I_k - \hat{I}_k\right|^2 \end{aligned} \qquad 4.2.10$$

From (4.2.10), it can simply be observed that the performance index, J, is a quadratic function of the equalizer coefficients, and by using appropriate algorithm optimum equalizer filter coefficients that minimize J can be determined.

First, we shall derive the tap weight coefficients that minimizes J when the equalizer has an infinite number of taps and this result can easily be projected to finite length equalizer. In this case the estimate $\hat{I}_k$ is expressed as

$$\hat{I}_k = \sum_{j=-\infty}^{\infty} c_j v_{k-j} \qquad 4.2.11$$

Substitution of (4.2.11) into the expression for J given in (4.2.10) and expansion of the result yields a quadratic function of the coefficients. This function can be minimized with respect to the $\{c_j\}$ to yield a set of infinite linear equations for $\{c_j\}$. These sets of equations can as well be

derived by selecting the coefficients $\{c_j\}$ to render the error ε_k orthogonal to the signal sequence $\{v_{k-l}^*\}$ for $-\infty < l < \infty$. Thus, we obtain

$$\sum_{j=-\infty}^{\infty} c_j E(v_{j-k} v_{k-l}^*) = E(I_k v_{k-l}^*), \qquad -\infty < l < \infty \tag{4.2.12}$$

Using the expression for v_k given in chapter 2 (eq. 2,6.2), the above moments can be evaluated and has the following forms

$$\begin{aligned} E(v_{k-j} v_{k-l}^*) &= \sum_{n=0}^{L} f_n^* f_{n+l-j} + N_0 \delta_{lj} \\ &= \begin{cases} x_{l-j} + N_0 \delta_{lj} & (|l-j| \le L) \\ 0 & (otherwise) \end{cases} \end{aligned} \tag{4.2.13}$$

and

$$E(I_k v_{k-l}^*) = \begin{cases} f_{-l}^* & (-L \le l \le 0) \\ 0 & (otherwise) \end{cases} \tag{4.2.14}$$

Now, substituting (4.2.13) and (4.2.14) into (4.2.12) and take the z-transform of both sides of the resulting equation, we obtain

$$C(z)[F(z)F^*(z^{-1}) + N_0] = F^*(z^{-1}) \tag{4.2.15}$$

Therefore, the transfer function of the equalizer based on the MSE criterion is

$$C(z) = \frac{F^*(z^{-1})}{F(z)F^*(z^{-1}) + N_0} \tag{4.2.16}$$

From (4.2.16), it can be noted that the only difference between this expression and the one based on peak distortion is the noise spectral density factor N_0 that appear in (4.2.16). When $N_0 = 0$ the minimization of MSE results in complete elimination of the ISI, and it will become identical to ZF-equalizer. However, when $N_0 \neq 0$, there is both residual ISI and additive noise at the output of the equalizer.

A measure of the residual ISI additive noise is obtained by evaluation the minimum value of J, denoted by $J_{\min}$. Thus, we obtain

$$J_{\min} = E(\varepsilon_k I_k^*)$$
$$= 1 - \sum_{j=-\infty}^{\infty} c_j f_{-j} \qquad 4.2.17$$

As infinite length equalizer is not realizable, let us now analyze the case in which the transversal equalizer spans finite time duration. The output of the equalizer of length $(2K+1)$ in the kth-signaling interval is

$$\hat{I}_k = \sum_{j=-K}^{K} c_j v_{k-j} \qquad 4.2.18$$

The MSE for the equalizer having $(2K+1)$ taps, denoted by $J(K)$, is

$$J(K) = E\left|I_k - \hat{I}_k\right|^2 = E\left|I_k - \sum_{j=-K}^{K} c_j v_{k-j}\right|^2 \qquad 4.2.19$$

Minimization of $J(K)$ with respect to the tap weights, forcing the error to be orthogonal to the signal samples, yields the following simultaneous equations:

$$\sum_{j=-K}^{K} c_j \Gamma_{lj} = \xi_l, \qquad l = -K,..-1,0,1,...K \qquad 4.2.20$$

where

$$\Gamma_{lj} = \begin{cases} x_{l-j} + N_0 \delta_{lj} & (|l-j| \le L) \\ 0 & (otherwise) \end{cases} \qquad 4.2.21$$

and

$$\xi_l = \begin{cases} f_{-l}^* & (-L \le l \le 0) \\ 0 & (otherwise) \end{cases} \qquad 4.2.22$$

The above set of linear equations can be summarized in a matrix form as

$$\overline{\mathbf{\Gamma}}\,\overline{\mathbf{C}} = \overline{\xi} \qquad 4.2.23$$

where $\overline{\mathbf{C}}$ denotes the column vector of $(2K+1)$ tap weight coefficients, $\overline{\mathbf{\Gamma}}$ denotes the $(2K+1)X(2K+1)$ Hermitian covariance matrix with elements Γ_{lj} and $\overline{\xi}$ is a $(2K+1)$-dimensional column vector with elements ξ_l. Then the optimum equalizer coefficients are determined as

$$\overline{\mathbf{C}}_{opt} = \overline{\mathbf{\Gamma}}^{-1}\overline{\xi} \qquad 4.2.24$$

Thus, the solution for optimum equalizer coefficients involves matrix inversion. The optimum tap weight coefficients minimize the performance index $J(K)$, with the result that minimum value of $J(K)$ is

$$J_{\min}(K) = 1 - \overline{\xi}^{t^*} \overline{\Gamma}^{-1} \overline{\xi} \qquad 4.2.25$$

where $\overline{\xi}^t$ represent the transpose of the column vector $\overline{\xi}$.

4.2.3. Decision Feedback Equalizers (DFE)

The linear equalizers of the preceding sections were trying to force the channel impulse response to satisfy the Nyquist criterion and to avoid ISI by this. That approach did not make any use of the knowledge of previously decoded symbols. In different researches, it had been shown that the main part of the ISI-energy is contained in the postcursers. These postcursers arrive at the receiver after the corresponding symbol that caused them. Therefore, the knowledge of the previous symbols and of the channel impulse response allows predicting the curser ISI-contributions of these symbols. In practice, this knowledge is used to compute a replica of the ISI-contributions of previously decoded symbols and to subtract it from the incoming data sample. An equalizer with this kind of operation is called a ***Decision Feedback Equalizer (DFE)***, as the decision made on symbols are fed back into a transversal filter, similar to a linear equalizer that produces the ISI-replica.

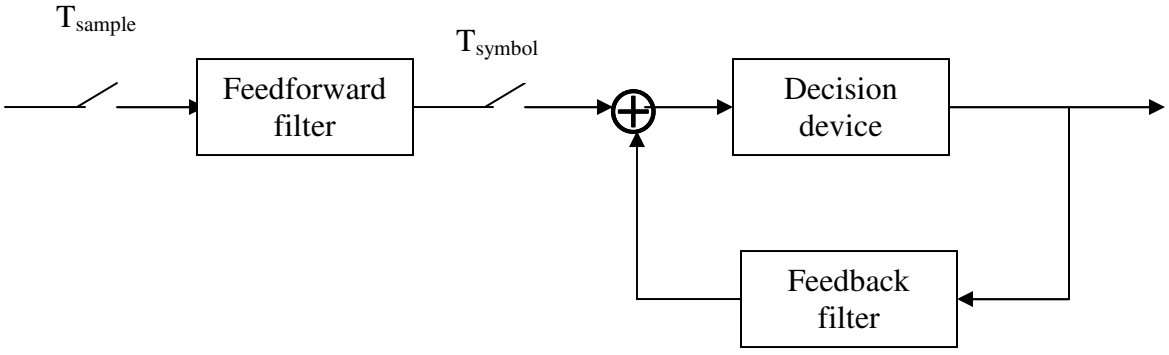


Fig. 4.3. *DFE with fractionally spaced feedforward filter*

However, this kind of operation also has a few disadvantages. The first is the zero delay decisions are needed for the feedback and therefore a direct concatenation with trellis-coded

modulation is somewhat difficult. The second one is that, if a wrong decision is fed back into the feedback filter (FBF) the probability of error rises for the next few symbol periods, as the wrong feedback values travel through the filter. This is the reason why errors in systems, using DFE, normally occur in short burst. However, for the relatively low bit error, the error propagation does not impair performance seriously. But the error propagation may cause problems in tracking slow timing variation and jitter. The FBF is able to treat the postcursers without any noise enhancement, but at the same time the signal energy, that was contained in the postcursers, gets lost, as they are simply cancelled out. This causes the DFE to be somewhat sub-optimal although widely used due to its relatively low complexity and good performance.

As the task of an equalizer is not only the treatment of postcursor-ISI, there is also added the so-called feedforward filter (FFF). The task of this filter, which is built up exactly as the linear equalizers of the previous sections, is to whiten the noise and to minimize the precursor ISI. In its limit of ideal adaptation, the fractionally spaced FFF of a DFE turns out to be the whitened matched filter. For the setting of taps of a DFE the overall performance is taken into account. This means that the optimal FFF is not the same as the optimal linear equalizer, leading therefore to less noise enhancement. Another advantage of this structure is the higher immunity to sampler phase deviations, which makes the performance of such an equalizer better than linear equalizers on most communication channels.

4.2.4. Adaptive Linear Equalizer [6, 17, 19]

In most practical communication systems that employ equalizers, the channel characteristics are unknown a priori and in many cases the channel response is time-variant, a typical characteristic of digital communication over HF radio link. In such a case, the equalizers are designed to be adjustable to the channel response and, for time-variant channels, to be adaptive to time variations in the channel response.

In this section different algorithms used for automatically adjusting the equalizer coefficients to optimize a specified performance index and to adaptively compensate for time variation in the channel will be presented. In the case of a linear equalizer, two different criteria –minimization

of peak distortion and mean-square error at the output of the equalizer- for determining the values of the equalizer coefficients are most often considered for realization of the filter. In this project work as well, MMSE is analyzed and used for the implementation of the equalizer.

4.2.4.1. Zero-Forcing Algorithm

In the peak distortion criterion, the peak distortion, $D(\bar{c})$, given by (4.2.5), is minimized by selecting the equalizer coefficients. In general, there is no simple computational algorithm for performing this optimization, except in the special case where the peak distortion at the input of the equalizer is less than unity, the eye should be open prior to equalization.

For an equalizer of length $2K+1$, the zero forcing solution is achieved by forcing the cross-correlation between the error sequence $\varepsilon_k = I_k - \hat{I}_k$ and the desired information sequence $\{I_k\}$ to be zero for shifts in the range $0 \le n \le K$. That is

$$E(\varepsilon_k I_{k-j}^*) = 0, \qquad j = -K,....,K \tag{4.2.26}$$

When the channel response is unknown, the cross-correlation given above is also unknown. This difficulty can be circumvented by transmitting a known training sequence $\{I_k\}$ to the receiver, which can be used to estimate the cross-correlation by substituting time averages for ensemble average of a training sequence of some predetermined length that equals or exceeds the equalizer length, the equalizer coefficients that satisfy (4.2.26) can be determined.

A recursive algorithm may be used to adjust the equalizer coefficients to their optimal values. The practical recursive algorithm used in many modem design is given as:

$$c_j^{(k+1)} = c_j^{(k)} + \Delta \varepsilon_k I_{k-j}^*, \qquad j = -K,....,K \tag{4.2.27}$$

where $c_j^{(k)}$ is the value of the jth coefficient at time $t = kT$, $\varepsilon_k = I_k - \hat{I}$ is the error signal at time $t = kT$, and Δ is a scale factor that controls the rate of adjustment. The term $\varepsilon_k I_{k-j}^*$ is an estimate of the cross-correlation (ensemble average) $E(\varepsilon_k I_{k-j}^*)$. The averaging operation of the cross-correlation is accomplished by means of first order recursive difference equation algorithm (4.2.27), which represents a simple discrete-time integrator.

Following the training period, after which the equalizer coefficients have converged to their optimal values, the decision at the output of the detector are generally sufficiently reliable so that they can be used to continue the adaptation. This is called a ***decision-directed mode*** of adaptation. Fig 4.4. illustrates the zero-forcing equalizer in the training mode and the adaptive mode of operation.

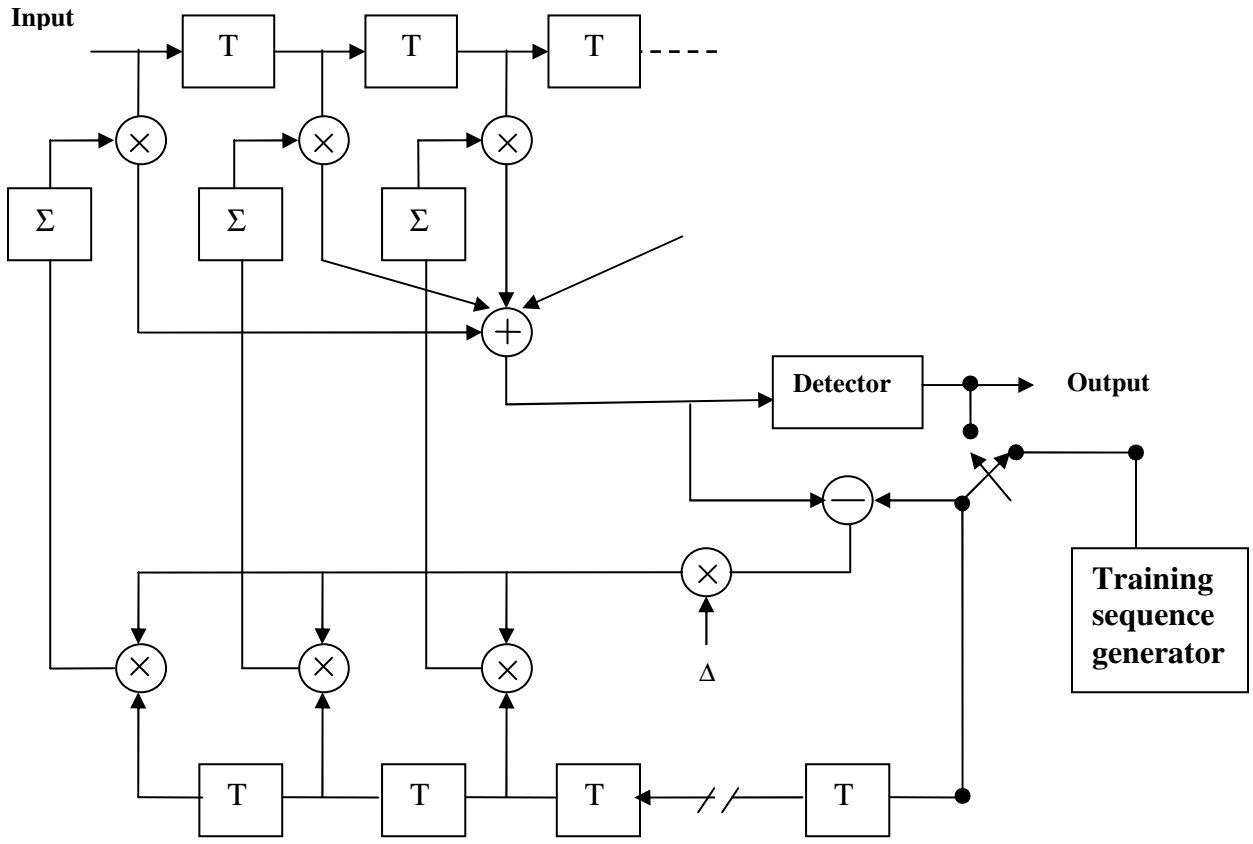


Fig. 4.4. *Adaptive Zero-Forcing Equalizer*

4.2.4.2. LMS Algorithm

As discussed in section (4.2.2.2), by minimization of the MSE performance index, the optimum equalizer coefficients can be determined from the solution of the set of linear equations, expressed in matrix form as

$$\overline{\overline{\Gamma}}\,\overline{C} = \overline{\xi} \qquad 4.2.28$$

where $\overline{\mathbf{\Gamma}}$ is $(2K+1)X(2K+1)$ covariance matrix of the signal samples, $\overline{\mathbf{C}}$ is the column vector of $(2K+1)$ equalizer coefficients, and $\overline{\boldsymbol{\xi}}$ is a $(2K+1)$ –dimensional column vector of channel filter coefficients. The solution for the optimum equalizer coefficients vector can be determined by inverting the covariance matrix, which can efficiently done by use of the *Levinson-Durbin algorithm*.[1, 22]

Alternatively, an iterative procedure that avoids the direct matrix inversion may be used to compute the optimum equalizer coefficients vector. The simplest and common iteration procedure is the method of steepest descent, in which one begins by arbitrarily choosing coefficient vector on the quadratic MSE surface in the $(2K+1)$ –dimensional space of coefficients. And each tap weight is changed in the direction opposite to its corresponding gradient component, which is computed at its respective point. Thus, the succeeding values of the equalizer coefficients vector are obtained according to the relation:

$$\overline{\mathbf{C}}_{k+1} = \overline{\mathbf{C}}_{k} - \Delta\overline{\mathbf{G}}_{k}, \qquad k = 0,1,2,... \tag{4.2.29}$$

where the gradient vector $\overline{\mathbf{G}}_k$ is

$$\overline{\mathbf{G}}_k = \frac{1}{2}\frac{d\mathbf{J}}{d\overline{\mathbf{C}}_k} = \overline{\mathbf{\Gamma}}\,\overline{\mathbf{C}}_k - \overline{\boldsymbol{\xi}} = -E(\varepsilon_k \overline{\mathbf{V}}_k^*) \tag{4.2.30}$$

The vector $\overline{\mathbf{C}}_k$ represent the set of coefficients at the kth iteration, $\varepsilon_k = I_k - \hat{I}_k$ is the error signal at the *kth* iteration, $\overline{\mathbf{V}}_k$ is the vector of received signal samples that makes up the estimate, and Δ is a positive number chosen small enough to ensure convergence of the iterative procedure.

The basic difficulty with the method of steepest descent for determining the optimum tap weight is the lack of knowledge of the gradient vector, which depends on both the covariance matrix and the vector of channel filter coefficients of cross-correlation. In turn, these quantities depend on the equivalent discrete-time channel model filter coefficients and on the covariance of the information sequence and the additive noise, all of which may be unknown at the receiver in general. To overcome this difficulty, estimate of the gradient vector, $\hat{\mathbf{G}}_k$ may be used, and it is expressed as

$$\hat{\mathbf{G}}_k = -\varepsilon_k \overline{\mathbf{V}}_k^* \tag{4.2.31}$$

Several other variations of the LMS algorithms are obtained by averaging or filtering the gradient vectors over several iterations prior to making adjustment of the equalizer coefficients.

In the above discussion, it has been assumed that the receiver has knowledge of the transmitted information sequence in forming the error signal between the desired symbol and its estimate. Such knowledge can be made available during a short training period in which a signal with a known information sequence is transmitted to the receiver for initially adjusting the tap weights. The length of this sequence must be at least as long as the length of the equalizer so that the spectrum of the transmitted signal adequately covers the bandwidth of the channel being equalized. In practice, the training sequence is often selected to be a periodic pseudo-random sequence.

A practical scheme for continuous adjustment of the tap weights may be either a decision-directed mode of operation in which decisions on the information symbols are assumed to be correct and used in place of the information sequence in forming the error signal, or one in which a known pseudo-random-probe sequence is inserted in the information bearing-signal either additively or by interleaving in time and the tap weights adjusted by comparing the received probe symbols with the known transmitted probe symbols.

The convergence properties of the LMS algorithm are governed by the step size parameter Δ. It can simply be verified that the convergence of the recursive algorithm is ensured if Δ satisfies the inequality

$$0 < \Delta < \frac{2}{\lambda_{\max}} \qquad 4.2.32$$

where $\lambda_{\max}$ is the largest eigenvalue of $\overline{\mathbf{\Gamma}}$

The recursive algorithm for adjusting the coefficients of the linear equalizers employs unbiased noisy estimates of the gradient vector. The noise in these estimates causes random fluctuations in the coefficients about their optimal values and, thus, leads to an increase in the MSE at the output of the equalizer. For the degradation in the output SNR of the equalizer due to the excess

MSE (due to noisy gradient estimate) of less than 1dB, the step size parameter should satisfy the following inequality: [29]

$$\Delta < \frac{0.2}{(2K+1)(x_0 + N_0)} \qquad 4.2.33$$

where $x_0 + N_0$ is received signal and noise power.

In digital implementation of the LMS algorithm, which is a case for this project work, the choice of step size parameter becomes more critical. In an attempt to reduce the excess mean square error, it is possible to reduce the step-size parameter to the point where the total mean square error actually increases. This condition occurs when the estimated gradient components of the vector $\varepsilon_k V_k^*$ after multiplication by the small step-size parameter Δ are smaller than one-half of the least significant bit in the fixed-point representation of the equalizer coefficients. In such a case, adaptation ceases. Consequently, it is important for the step size to be large enough to bring the equalizer coefficients in the vicinity of C_{opt} . If it is desired to decrease the step size significantly, it is necessary to increase the precision in the equalizer coefficients. Typically, 16 bits of precision may be used for the coefficients, with about 10-12 of the most significant bits used for arithmetic operations in the equalization of the data. The remaining last significant bits are required to provide the necessary precision for the adaptation process. Thus, the scaled, estimated gradient components $\Delta\varepsilon V_k^*$ usually affect only the least-significant bits in any one iteration. In effect, the added precision also allows for the noise to be averaged out, since many incremental changes in the least significant bits are required before any change occurs in the upper more significant bits used in arithmetic operations for equalizing the data.

CHAPTER FIVE

THE TMS320C50X DIGITAL SIGNAL PROCESSOR

The Texas Instruments TMS320 product line, a 16-bit fixed point and a 32-bit floating-point single chip Digital Signal Processors (DSP), contains a family of digital signal processors designed to support a wide range of high-speed or numeric-intensive DSP applications. The DSPs combine the flexibility of high-speed controller with the numeric capability of an array processor, thereby offering an inexpensive alternative to a multichip bit-slice processor or an expensive commercial array processor.

Because of their computational power, high I/O throughput, and real time programming, the TMS320 processors have been widely adapted in telecommunication, data communication, digital control, and computer applications. In addition, TMS320 processors have efficient hardware intensive DSP-oriented instructions, which make them an ideal choice for easy and flexible realization of signal processing problems.

Architecturally, the TMS320 utilizes a modified Harvard architecture for speed and flexibility. In a strict Harvard architecture, the program and data memories lie in two separate spaces, permitting a full overlap of the instruction fetch and execution. The TMS320 family's modification of the Harvard architecture allows transfer between program and data spaces in which processing power is maximized by maintaining two separate bus structures (program and data).

In this thesis work, the TMS320C50 DSP Starter Kit (DSK), a low-cost, simple standalone application board, is used for implementation of the linear adaptive digital channel equalizer for digital/data communication over HF channel.

5.1. General Description [24, 25]

The ’C5x generation consists of the ’C50, ’C51, ’C52, ’C53, ’C53S, ’C56, ’C57, and ’C57S DSPs, which are fabricated by CMOS integrated-circuit technology. The operational flexibility and speed of the ’C5x are the result of combining an advanced Harvard architecture, a CPU with application-specific hardware logic, on-chip peripherals, on-chip memory, and a highly specialized instruction set. The ’C5x is designed to execute up to 50 million instructions per second (MIPS).

Key features of the ’C5x DSPs are:

- Source-code compatible with ’C1x, ’C2x, and ’C2xx devices
- 35-/50-ns single-cycle fixed-point instruction execution time (28.6/20 MIPS)
- Power consumption control with IDLE1 and IDLE2 instructions for power-down modes
- 224K-word × 16-bit maximum addressable external memory space (64K-word program, 64K-word data, 64K-word I/O, and 32K-word global memory)
- 1056-word × 16-bit dual-access on-chip data RAM
- 9K-word × 16-bit single-access on-chip program/data RAM
- 2K-word × 16-bit single-access on-chip boot ROM
- 32-bit arithmetic logic unit (ALU), 32-bit accumulator (ACC), and 32-bit accumulator buffer (ACCB)
- 16-bit × 16-bit parallel multiplier with a 32-bit product capability
- 0- to 16-bit left and right data barrel-shifters and a 64-bit incremental data shifter
- 16-bit parallel logic unit (PLU)
- Dedicated auxiliary register arithmetic unit (ARAU) for indirect addressing
- Eight auxiliary registers
- 8-level hardware stack
- 4-deep pipelined operation for delayed branch, call, and return instructions
- Eleven shadow registers for storing strategic CPU-controlled registers during an interrupt service routine (ISR)

- Extended hold operation for concurrent external direct memory access (DMA) of external memory or on-chip RAM
- Two indirectly addressed circular buffers for circular addressing
- Single-cycle multiply/accumulate instructions
- Single-instruction repeat and block repeat operations
- Block memory move instructions for better program and data management
- Memory-mapped register load and store instructions
- Conditional branch and call instructions
- Delayed execution of branch and call instructions
- Fast return from interrupt instructions
- Index-addressing mode
- Bit-reversed index-addressing mode for radix-2 Fast Fourier Transforms (FFTs)
- 64K parallel I/O ports (16 I/O ports are memory mapped)
- Sixteen software-programmable wait-state generators for program, data, and I/O memory spaces
- Interval timer with period, control, and counter registers for software stop, start, and reset
- Phase-locked loop (PLL) clock generator with internal oscillator or external clock source
- Full-duplex synchronous serial port interface for direct communication between the ’C5x and another serial device
- Time-division multiplexed (TDM) serial port

5.2. TMS320C5X Architecture [24, 25]

TMS320CX consists of several fundamental modules: a digital signal processing core to perform mathematical operations (Central Processing Unit/CPU), memory to store data and program instructions, and possibly a mixed-signal product to converse between the analog and digital worlds (Peripheral-Interfacing Circuits).

5.2.1. Central Processing Unit (CPU)

The CPU, which is responsible for arithmetic and logic (Boolean) operations, is composed of five major sections, as seen in Fig 5.1.

- Central Arithmetic Logic Section (CALU)
- Multiplication Section
- Input Scaling Section
- Auxiliary Register Arithmetic Unit (ARAU)
- Parallel Logic Unit (PLU)

The central arithmetic logic unit (CALU), implements a wide range of arithmetic and logic functions, most of which execute in a single clock cycle. These functions can be grouped into four categories:

- 16-bit addition
- 16-bit subtraction
- Boolean logic operations
- Bit testing, shifting, and rotating.

The TMS320C5X performs 2's complement arithmetic, using the 32-bit Arithmetic Logic Unit (ALU) – general-purpose arithmetic unit that uses 16-bit word taken from data memory or derived from immediate instructions, or the 32-bit result from the multiplier, and accumulator register. The accumulator stores the output from the ALU and is also the second input to the ALU.

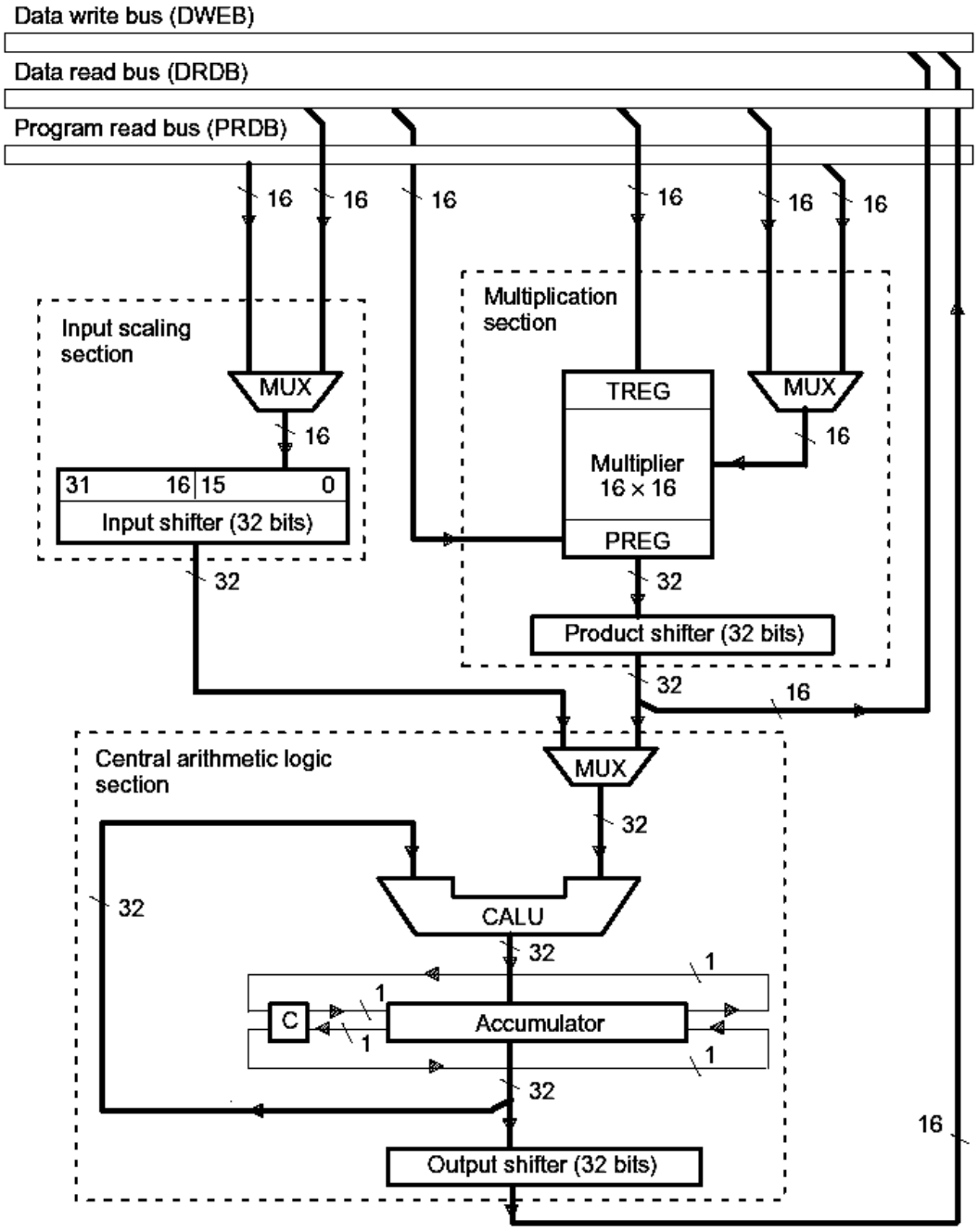


Fig. 5.1 *Block Diagram of the Input Scaling, Central Arithmetic Logic, and Multiplication Sections of the CPU*

The parallel Logic Unit is used to execute logic operations without affecting the content of the accumulator. It provides the bit manipulation ability required of a high-speed controller and simplifies the bit setting, clearing, and testing required with control and status registers.

The CPU also contains the auxiliary register arithmetic unit (ARAU), an arithmetic unit independent of the central arithmetic logic unit (CALU). The main function of the ARAU is to perform arithmetic operations on eight auxiliary registers (AR7 through AR0), which provide flexible and powerful indirect memory addressing, in parallel with operations occurring in the CALU.

The TMS320C5x uses a 16-bit × 16-bit hardware multiplier that can produce a signed or unsigned 32-bit product in a single machine cycle. As shown in Fig. 5.1, the multiplication section consists of:

- The 16-bit temporary register (TREG), which holds one of the multiplicands
- The multiplier, which multiplies the TREG value by a second value from data memory or program memory
- The 32-bit product register (PREG), which receives the result of the multiplication
- The product shifter, which scales the PREG value before passing it to the CALU.

A 32-bit input data-scaling shifter (input shifter) aligns a 16-bit value coming from memory to the 32-bit CALU. This data alignment is necessary for data-scaling arithmetic as well as aligning masks for logical operations. The input shifter operates as part of the data path between program or data space and the CALU and, thus, requires no cycle overhead.

Eight level of hardware stack save the contents of the program counter during interrupt and subroutine calls. On interrupts, the strategic registers (ACC, ACCB, ARCR, INDEX, PMST, PREG, ST0, ST1, TREGs) are pushed onto a one-deep stack and popped upon interrupt return, thus providing a zero-overhead interrupt context switch.

5.2.2. Memory

The total memory address range of the TMS320C5X device is 224K 16-bit words. The TMS320C5X enhanced Harvard architecture supports multiple spaces that can be accessed on three parallel buses. The parallel nature of the architecture allows the device to perform three concurrent operations in any given machine cycle: fetching an instruction, reading an operand, and writing an operand. The three parallel buses are the program read/write bus (PAB), data read bus (DAB1), and data write bus (DAB2). The TMS320C5X memory is organized into four individually selectable spaces: 64K program, 64K local data, 32K global data, and 64K I/O port. Within any of these spaces RAM, ROM, EPROM, EEPROM, or memory-mapped peripherals can reside either on- or off-chip.

The program space contains the instructions to be executed as well as tables used in the execution. The local data space stores data used by the instructions. The global data space can share data with other processors within the system, or can serve as additional data space. The I/O space interfaces to external memory-mapped peripheral and can also serve as extra data storage space.

The TMS320C5X device includes a considerable amount of on-chip memory to aid in system performance and integration. It includes 2K words of boot ROM, 9K words program/data single access RAM (SARAM), and 1056 words of dual access data RAM (DARAM). The boot ROM resides in program space at address 0 and includes a device test and boot code. The SARAM, which requires a full machine cycle to perform a read or a write, can be mapped in program and/or data space and reside at address 0800h in either space. The DARAM that can be read from and written to in the same cycle is configured in three blocks: blocks 0 (B0)-512 words at address 0100h-02FFh in local data memory, or 0FE00h-0FFFh in program space, block 1 (B1)-512 words at address 0300h-0FFFh in local data memory, and block 2 (B2)-32 words at address 060h in local data memory. The memory map of the TMS320C50, for both microprocessor and microcomputer modes, is depicted in fig. 5.2.

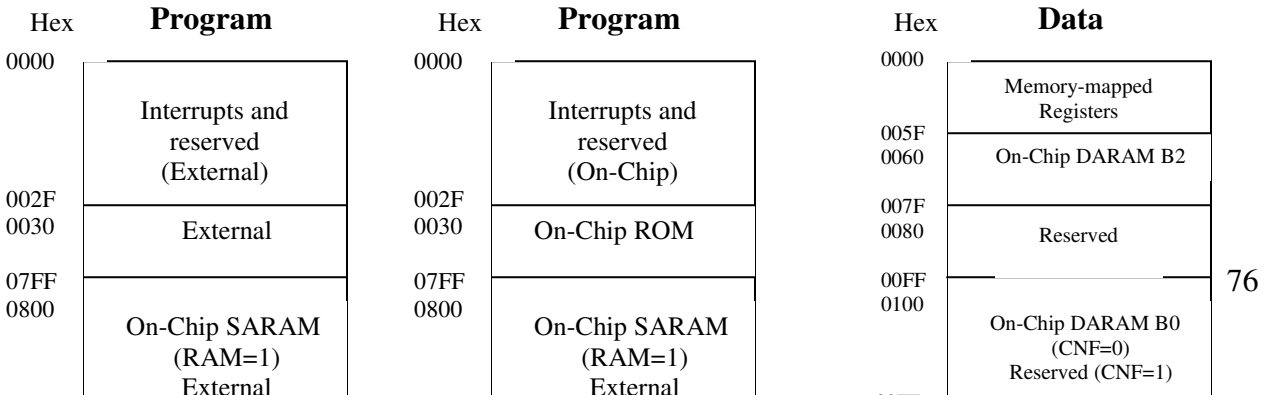


Fig. 5.2 *TMS320C50 Memory Map*

Memory Addressing Mode

The program memory is accessed only by the PAB address bus. The address for this bus is generated by the program counter (PC) when instructions and long immediate operands are fetched. The TMS320C5X devices fetch instructions by putting the PC on the PAB bus and reading the appropriate location in memory. While the read is executing, the PC is incremented for the next fetch. In case of a program address discontinuity (for example, branch, call, return, interrupt, or block repeat), the appropriate address is loaded onto the PC.

The data space address generation is controlled by the decoder of the current instruction. It is read via DAB1 on instructions with only one data memory operand and PAB on instructions with a second data memory operand. An instruction operand can be addressed in eight different ways in this DSP.

a. Direct Addressing Mode

In this mode, the 9-bit data memory page pointer (DP) points to one of 512 pages (1- page= 128 words). The data memory address (dma), specified by the seven LSB of the instruction, points to the desired word within the page. The address on the DAB1 is formed by concatenating the 9-bit DP with the 7-bit dma.

b. Memory-Mapped Addressing Mode

This mode operates similar to the direct addressing mode except that the most significant 9-bits of the address are forced to zero instead of being loaded with the content of the DP. This allows one to directly address the memory-mapped registers of data page zero without the overhead of changing the DP or auxiliary registers (ARs).

c. Indirect Addressing Mode

In indirect addressing, the currently selected 16-bit auxiliary register addresses the data memory through auxiliary register file bus (AFB). While the selected auxiliary register provides the data memory address and the data is being manipulated by the CALU, the contents of the auxiliary register can be manipulated through the ARAU.

d. Short Immediate and Long Immediate Mode

The operand may reside as part of the instruction machine code. In the case of short immediate operand, the operand is contained in the single word instruction, while for long immediate mode the operand immediately follows the opcode in the program sequence.

e. Register Access Mode

In this addressing mode, the operand may come from a CPU register. This type of operand is used in special cases where bit manipulation is performed either by CALU, ARAU, or PLU.

f. Long Immediate addressing Mode

In the long immediate addressing mode, an operand is addressed by the second word of a two word instruction. In this case, the program address/data bus (PAB) is used for the operand fetch. This technique is used when two memory addresses are required for the execution of the instruction.

g. Registered Block Memory Addressing Mode

This mode operates like the long immediate addressing mode with the exception that the address comes from bit block move address register (BMAR). The merit of this technique is that the address of the block of memory to be acted up on can changed during execution of the program.

5.2.3. Peripheral Interface

The peripheral interface circuits connected to the TMS320C50 core CPU are the serial port, TDM (time-division-multiplexed) serial port, timer, software-programmable wait state generators, I/O ports, divide-by-one clock, and XF (external flag) and $\overline{BIO}$ (branch control input). These peripherals are controlled through registers that reside in the memory map. The operation of serial ports and timer is synchronized to the core CPU via interrupts. Setting and clearing bits can enable, disable, initialize, and dynamically reconfigure the peripherals. Data is transferred to and from the peripherals through memory-mapped data registers. When the peripheral is not in use, the internal clocks are shut off from that peripheral, allowing for lower power consumption when the device is in normal run mode or idle mode.

The Full duplex on-chip serial port provides direct communication with serial devices such as codecs, serial A/D (D/A) converters and other serial systems. The serial ports may also be used for interconnection between processors in multiprocessing applications.

5.3. The TMS320C50 DSP Starter Kit (DSK) [24, 25]

The ’C5x DSK is a low-cost, simple, stand-alone application board that helps to experiment with and use ’C5x DSPs for real-time signal processing. The kit has an assembler and debugger to develop, test and refine DSP assembly language.

The DSK assembler is a software interface incorporating the most significant features of an assembler. Unlike many other commercial assemblers it doesn’t go through a linker phase to create an output file. Instead, it uses special directives to assemble a code at an absolute address during the assembly phase. The debugger is a window-oriented interface capable of loading and executing a code with single step, breakpoint and run-time halt capabilities.

Figure 5.3 depicts the basic block diagram of the ’C50. It shows the interconnections, which include the host interface, analog interface, and emulation interface. PC communications are via the RS-232 port on the DSK board. The 32K bytes of PROM contain the kernel program for boot loading. All pins of the ’C50 are connected to the external I/O interfaces. The external I/O interfaces include four 24-pin headers, a 4-pin header, and a 14-pin XDS510 header.

The TLC32040 AIC (Analog Interface Card) on the board provides a single-channel, input/output, voice-quality analog interface with single-chip digital-to-analog (D/A) and analog-to-digital (A/D) conversion of 14 bits dynamic range with variable D/A and A/D sampling rate and filtering. The TLC32040 AIC interfaces to the ’C50 serial port.

DSK analog capabilities are suited to many applications, including audio data processing. Most pre-amplified microphones and speakers can directly be connected to the DSK analog input and output.

The ’C5x DSK application board has no external memory configured on the board. However, the 10K on-chip RAM of the ’C50 provides enough memory for most DSP application programs. The on-chip SARAM is configured as a program and data memory, while the on-chip DARAM (B2) is reserved as a buffer for the status registers.

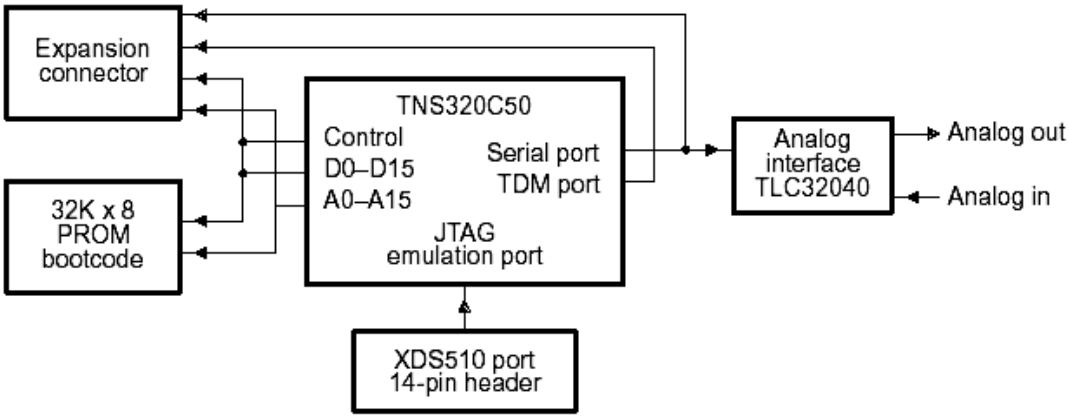


Fig. 5.3 *'C5X DSK Block Diagram*

CHAPTER SIX

DESIGN AND IMPLEMENTATION

Equalization can compensate for the channel-induced ISI that is seen in frequency selective fading. The process of equalizing the ISI involves some method of gathering the dispersed symbol energy back together into its original time interval. In effect, equalization involves insertion of a filter to make the combination of channel and filter yield a flat response with linear phase. The phase linearity is achieved by making the equalizer filter the complex conjugate of the time reverse of the dispersed pulse. Because in HF communication system the channel response varies with time, the equalizer filter must also change or adapt to the time-varying channel, and hence ***adaptive equalizers*.**

For this thesis work, linear transversal equalizer (LTE) architecture is considered, shown in Fig 6.1. That is, the received signal is passed through a tapped delay line with tap spacing of symbol period, SS. The tapped signals are weighted and added to form the equalized output signal.

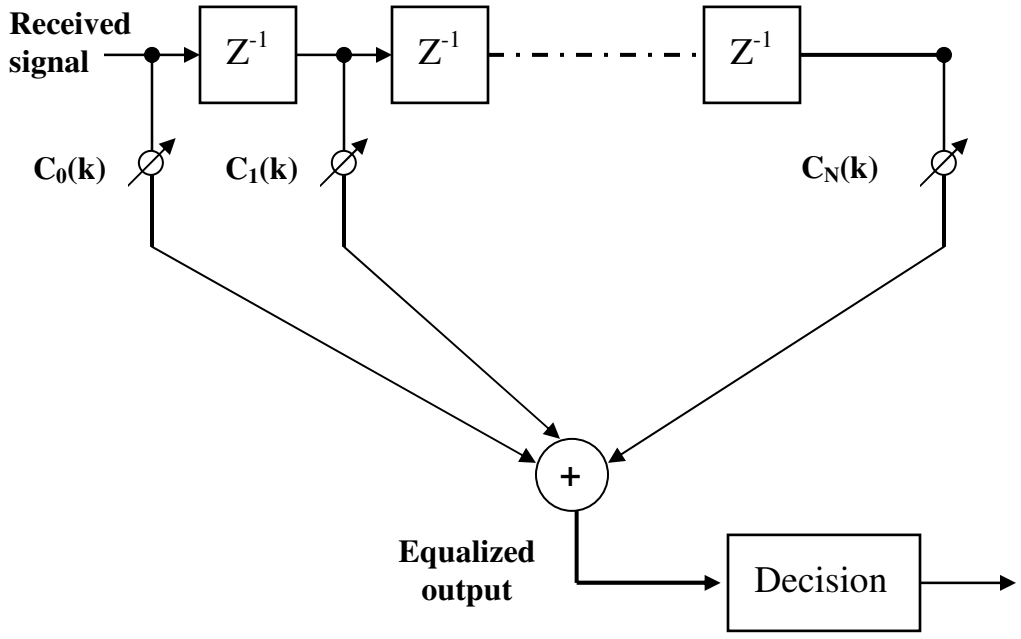


Fig. 6.1. *Tapped-delay-line structure for LTE.*

The optimum tap gains are determined using the MSE criterion, that is, they are adjusted to minimize the mean square of the error signal between equalized signal and the actual transmitted symbol. Furthermore, a least-mean-square (LMS) algorithm is used to adapt the coefficients so as to track the time-variant nature of the channel.

For a linear Binary Phase Shift Keying (BPSK) modulation technique, a type of modulation considered for this thesis work, the optimum tap values could be found from the solution of a set of linear equations. Therefore, the optimum MMSE tap weights obtained using LMS algorithm are given as:

$$\begin{aligned} & c_k(i+1) = c_k(i) + 2\Delta e(i)x(i-k) \\ & where \quad e(i) = I(i) - \tilde{I}(i) \\ & \tilde{I}(i) = \sum_{k=0}^{N-1} c_k x(i-k) \end{aligned} \qquad 6.1$$

where $I(i)$ is transmitted symbol, $\tilde{I}(i)$ is estimate of the transmitted symbol at the output of the equalizer, $x(i)$ is a received symbol, N is length of the equalizing filter and Δ is a scale factor that controls the rate of adjustment.

The delay introduced to a signal by the equalizer depends on the position of the highest tap gain that corresponds to the ISI-free sample value. As indicated in Fig. 6.1., FIR filter is used to realize the equalizer that a certain input symbol only, as many symbol periods as the equalizer taps for SS equalizer, influences its output. Furthermore, practical experiments on HF channel indicate that postcursor-ISI is much longer in duration and high in intensity than precursor-ISI. The length N of the equalizer can be determined from the length of an equivalent FIR filter model of the channel. i.e., the tapped-delay should at least span time spread of the channel.

As described in chapter four, the recursive algorithm for adjusting the coefficients of a linear equalizer employs unbiased noisy estimates of the gradient vector. The noise in these estimates causes random fluctuations in the coefficients about their optimal values and, thus, leads to an increase in the MSE at the output of the equalizer. However, this excess MSE can be controlled by proper selection of step size, Δ. The optimal step size value can, therefore, be evaluated as: [29]

$$\Delta < \frac{2}{(N)(P_r)} \qquad 6.2$$

where N is equalizer tap-delay length and P_r is total received power.

6.1. Implementation on TMS320C50 DSP [15, 24, 25]

This section provides an overview of programming the TMS320C5x DSP Starter Kit (DSK) to realize the selected equalizer. The ’C5x DSK is a low-cost, simple, stand-alone application board that helps to experiment with and use ’C5x DSPs for real-time signal processing. The DSK has a ’C50 onboard to allow full-speed verification of the ’C5x code. It has an assembler and debugger to develop, test and refine DSK assembly language programs. Details of the DSK are presented in chapter five.

The adaptive FIR filter model of the selected digital channel equalizer, described in equation 6.1, is implemented on the TMS320C50 using the resources available for the DSK. The source program consists of source statements that can contain assembler directives, assembly language instruction and comments. The DSK software development flow is depicted in Fig. 6.2.

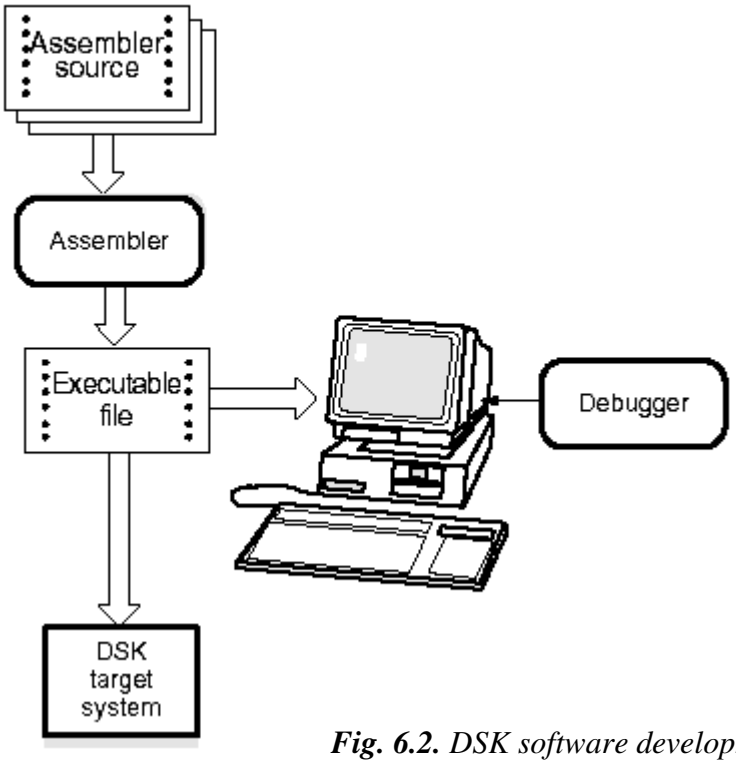


Fig. 6.2. *DSK software development flow*

From equation 6.1, it is clear to note th pted or updated in every symbol interval. Filtering of the incoming signal and computing the error are also be done within the same symbol interval. This is computationally- expensive and time-consuming, which makes real time signal processing difficult unless fast and computationally efficient DSPs with reduced execution time are used. However, TMS320C50 DSP is equipped with a single-cycle multiply/accumulate with data move instructions (like **MPYA, ZALR, RPTB**) and larger on-chip RAM, which make real time channel equalization a success. **MPYA** and **ZALR** are used to reduce the number of instruction in the main adaptation loop. Furthermore, the RPTB (repeat block) instruction allows the block of instructions to be repeated without any penalty for looping.

Quantization errors in the updated coefficients and represented signal values due to the finite word-length effect in the hardware can be minimized if the result is obtained by rounding rather than truncating. For each coefficient in the filter at a given symbol time, the factor $2 * \Delta * e(i)$ is a constant. This factor is therefore computed once and stored in the **T**-register (temporary multiplicand register) for each of the update.

The filter coefficients in the routines used in this thesis work are initially stored in program memory, and then moved to data memory. These coefficients are represented in **Q15** format, i.e., the binary point (represented in two's-complement form) is assumed to follow the most significant bit. This gives a range of ***0.999969*** to ***-1.0*** with increment of ***0.000031.*** The input is also in **Q15** format so that when two **Q15** numbers are multiplied, the result is a number in **Q30** format. When the **Q30** number resides in a 32-bit accumulator, the binary point follows the second most significant digit. The **Q30** number must be adjusted by left-shifting by one while maintaining the most-significant 16 bits of the result to match it with the **Q15** output data format. **SACH** is used to perform this.

The following sample code shows a routine that implements an N-tap FIR filter and an LMS adaptation of its coefficients, MMSE linear equalizer realization. The SARAM of 'C50 DSP is mapped in both program and data space at the same time by setting OVLY and RAM control flags to 1. This feature can be used to advantage by locating the coefficients table in SARAM so that it can be accessed by **MACD** and **MPY** instructions without modifying RAM configuration as **MACD** instruction requires one of its operand to be in program space.

```
              .title 'Linear Adaptive Channel Equalizer (FIR/LTE)'
              .def          ADPFIR
              .def          XN,YN
              .mmregs
*
* This N-tap adaptive FIR filter uses on-chip memory block B0 for
* Coefficients and block B1 for data samples. The newest input
* should be in memory location XN when called. The output will be
* in memory location YN when returned. OVLY=1 and RAM=1 for this subroutine.
*
COEFFP   .set 02000h        ;Program memory address of the coeff. in SARAM
COEFFD   .set 02000h        ;Data memory address of the coeff. in SARAM
ONE      .set 7Ah           ;Constant one.             (DP=0).
DELT_A   .set 7Bh           ;Adaptive constant.        (DP=0).
ERR      .set 7Ch           ;Signal error.             (DP=0).
ERRF     .set 7Dh           ;Error function.           (DP=0).
YN       .set 7Eh           ;Filter output.            (DP=0).
XN       .set 037Fh         ;Newest data sample.
XNM1     .set 0380h         ;Next newest data sample.
XOLD     .set 0380h+(N-1)h  ;Oldest data sample.
*
* Finite impulse response (FIR) filter.
*
ADPFIR   ZPR                ;Clear P register.
         LACC #1,14         ;Load output rounding bit.
         MAR  *,AR3
         LAR  AR3,#XOLD     ;Point to oldest sample.
FIR      RPT  #(N-1)
         MACD COEFFP,*-     ;N-tap FIR filter.
         APAC
```

```
        SACH YN,1          ;Store the filter output.
        NEG                ;Acc=-y(n).
        LAR  AR3,#XN
        ADD  *,15          ;Add the newest input sample.
        SACH ERR,1         ;err(n)=x(n)-y(n).
        DMOV *             ;Include newest sample.
*
* LMS adaptation of Filter Coefficients.
*
        LT   ERR           ;T=err.
        MPY  DELT_A        ;P=delta*err(i).
        PAC                ;errf(i)=delta*err(i).
*
        LACC #(N-1)
        SAMM BRCR          ;N coefficients to update in the loop.
        LAR  AR2,#COEFFD   ;Point to the coefficients.
        LAR  AR3,XOLD      ;Point to the data sample.
        LT   ERRF
        MPY  *-,AR2        ;P=2*delta*err(i)*x(i-N-1).
*
        RPTB LOOP-1        ;For I=0,I<=(N-1),I++.
ADAPT   ZALR *,AR3         ;Load ACCH with ck(i).
        MPYA *-,AR2        ; P=2*delta*err(i)*x(i-K-1).
*                         Acc=ck(i)+2*delta*err(i)*x(i-k).
        SACH *+            ;Store ck(i+1).
*
LOOP    ZALR *,AR3         ;Finally update last coeff. c0(i).
        RETD               ;Delayed return.
        APAC               ;Acc=c0(i)+2*delta*err(i)*x(i).
        SACH *+            ;Save c0(i+1).
```

A full program code for the MMSE linear equalizer, which includes initialization of the processor serial ports and the analog interface circuit, is available in ***Appendix C***. And ***Appendix B*** depicts the flow chart used to develop the program code.

6.2. Simulation and Results

In this thesis work, the performance of a linear MMSE channel equalizer with LMS adaptive algorithm for a typical digital HF channel is studied. As the main concern this report is to study the optimal performance of the selected equalizer, a bandwidth-efficient transmission of linearly modulated signals, Binary Phase Shift Keyed (BPSK) signals, with no error correction and channel coding, over a linear, frequency-selective slowly fading HF channel is considered.

6.2.1. Simulation

In an attempt to implement software 'test bed' most closely matching realistic HF-communication environment, a software model using a MATLAB programming language is created. Fig. 6.3. shows the system configuration block diagram used to test the equalizer

performance. Bit error rate (BER) and eye diagram pattern are used as index of performance. To develop the system configuration the following assumptions are made.

- Transmitter bit rate – 2.4 kbit/second.
- Channel time spread – 3 symbol period, 1.25 ms.
- Slowly fading channel, time-invariant within a symbol period, maximum Doppler Shift – 10 Hz.
- Channel bandwidth – 3 kHz.
- Symbol Spaced (SS) three-tap channel and equalizer model.

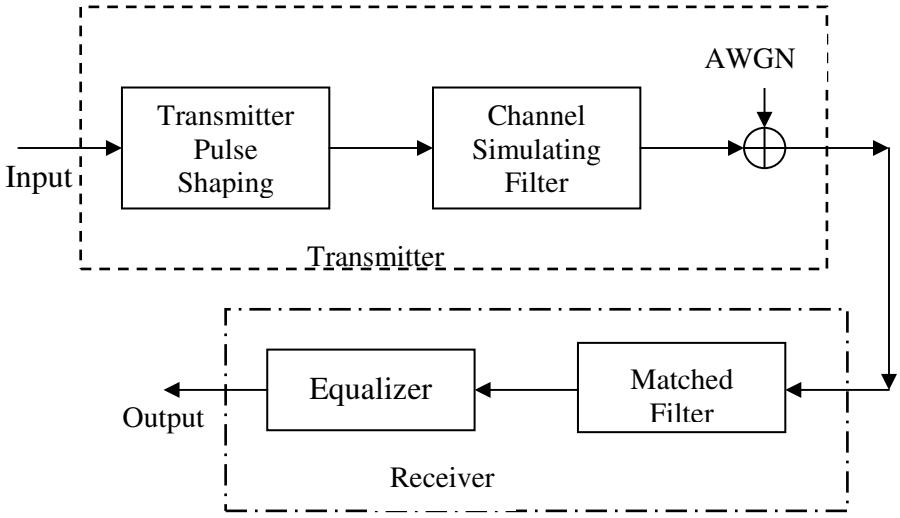


Fig 6.3. *Configuration block diagram for Simulation*

Input Data

To incorporate the natural information bit sequence in practical communication, a software random bit generator is used as a data source for the test setup. The binary digits are then BPSK modulated. That is, binary '0' mapped to 1 and binary '1' to -1.

Transmitter Pulse Shaping Filter

For a bandwidth efficient communication over a band-limited channel, the information sequences are mapped to analog waveforms that match the spectral characteristics of the channel. The mapping is simulated by passing the BPSK modulated information sequence through a rectangular pulse shaping filter. A MATLAB filter toolbox is used for realization of the filter.

Matched Filter

A matched filter, matched to the transmitting rectangular filter, is used to reject the AWGN trapped by the transmitted signal. Here, the AWGN is simulated by generating gaussian distributed numbers with a specified deviation, to represent the given SNR, and added bit by bit to the transmitted signal.

Channel Model Simulation

As described in the third chapter, HF communication is characterized by multipath propagation and fading. Transmitted signals travel over several propagation modes to receiver via multiple reflection form the surface of the earth and ionosphere. Because of different propagation times over different paths, signals arriving at the receiver may be spread in time, causing ISI. In addition, the ionospheric turbulence of the different layers leads to different Doppler shift on each propagation mode.

Therefore, a band-limited HF channel that results in ISI can be modeled by a lowpass equivalent discrete-time model, expressed as a sum of a number of delayed Dirac pulses with time-varying complex path weights. In this thesis, as the case in most practical channel simulations, the delays will be normalized to symbol rate and also the average channel energy is normalized to unity. Empirical data on fading channel, HF in particular, of tap coefficients suggest that they are *Rayleigh distributed*. The transfer function of the discrete-time channel model for a three-tap coefficients, a three symbol period multipath spread channel, is described by:

$$H(z) = c_0(k) + c_1(k)z^{-1} + c_2(k)z^{-2} \qquad 6.3$$

where $c_i(k)$ is the ith path-weight of the fading channel varying with time k.

The Rayleigh fading taps $c_i(k)$ (one tap per symbol interval) can be simulated by feeding White Gaussian Noise (WGN) to a digital filter matched to the respective fading spectrum, i.e. the filter frequency response must be a good approximation to the square root of the (normalized) tap power spectral density, Doppler power spectrum.

For this thesis work, the time-varying coefficients are generated by passing white Gaussian noise through a Butterworth filter. The bandwidth of the Butterworth filter determines the relative bandwidth (fading rate) of the channel. Assuming that we have a nominal 2 kHz HF channel, 2400 symbols/s sampling rate, and a second-order Butterworth filter having a 3 dB bandwidth of 10 Hz, the curves of the channel impulse responses changing with time are depicted in Fig. 6.4.

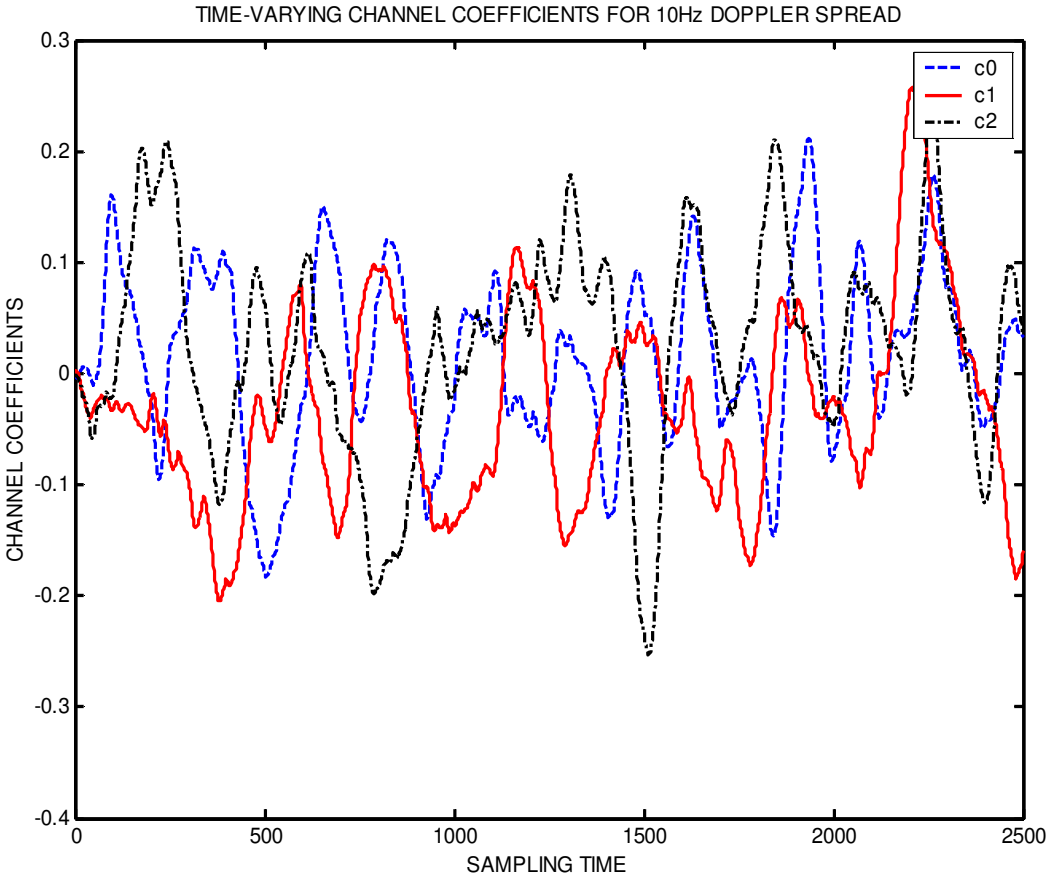


Fig.6.4. *Time varying coefficients*

Equalizer

Inline with the channel tap length and as described in section 6.1, a three-tap linear MMSE equalizer with LMS coefficient adjustment algorithm is considered. To determine optimal parameters, initial tap weights and step size, the selected equalizer is simulated using MATLAB. The software simulation is also used to study the performance of the equalizer for different operating conditions and channel characteristics. To initially adjust the coefficients a training sequence, symbols known to the receiver, is implemented. In all simulations, each data frame

comprises of a training symbols of ***ten percent*** of the transmitted sequence. After the training session, ***decision-oriented mode of operation*** follows to continuously adjust the equalizer coefficients. All simulation results are depicted in the result section. Here the probability of bit error (BER) is considered as a measure of performance.

6.2.2. Results

This section depicts the results of all the simulations and performance of the implemented equalizer for different channel distortions. It includes the performance of the selected equalizer for a purely frequency selective or time dispersive channel having a time spread of three symbol period, and a slowly fading (time-variant) dispersive channel.

The frequency response of a three tap dispersive channel and a corresponding equalizer; and performance of the equalizer are presented in Fig. 6.5 and Fig 6.6, respectively. In the given multipath channel, 60% of the energy is assumed to be transmitted over the first channel, and 25% and 15% over the second and the third paths, respectively, with a data transmission rate of 2.4kbits/s and a 10dB SNR.

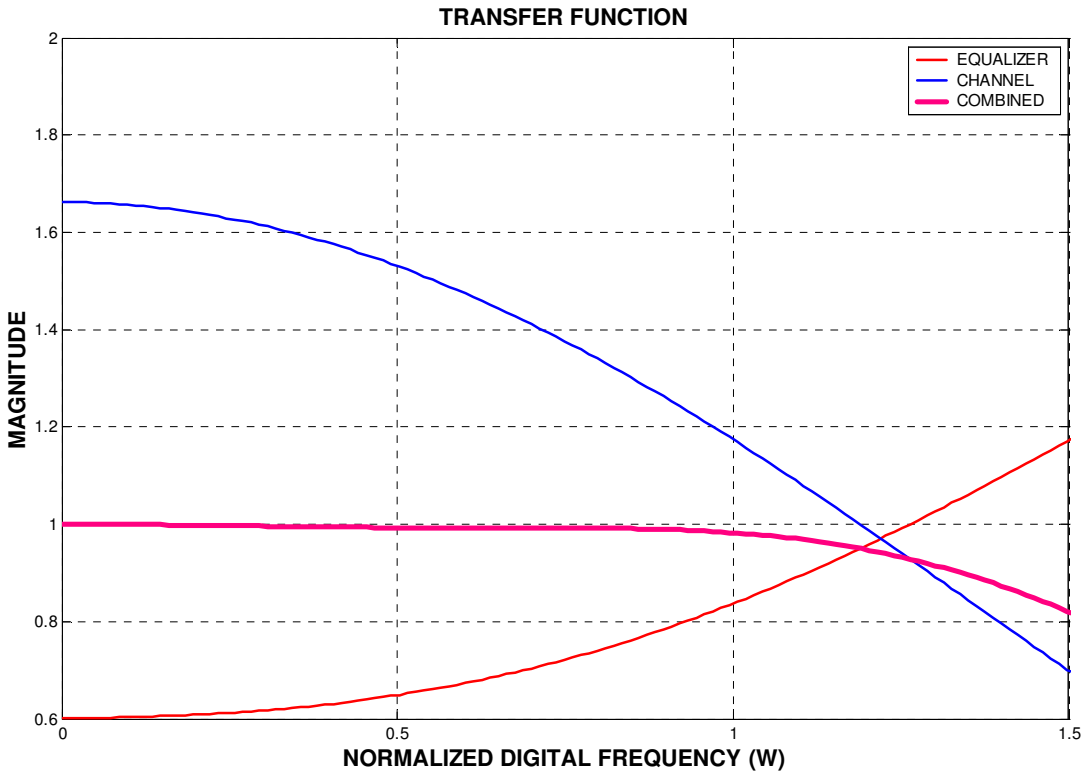


Fig. 6.5. *Channel and Equalizer transfer functions*

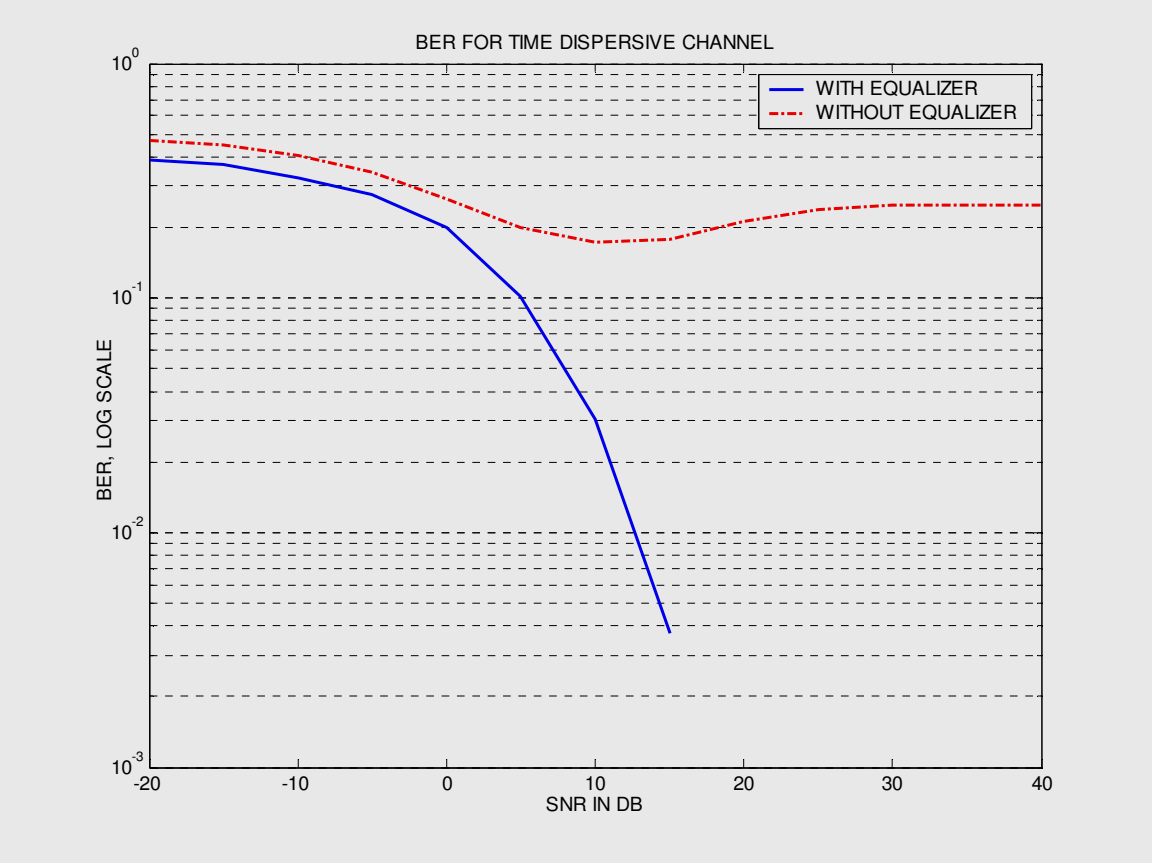


Fig. 6.6. *Equalizer performance*

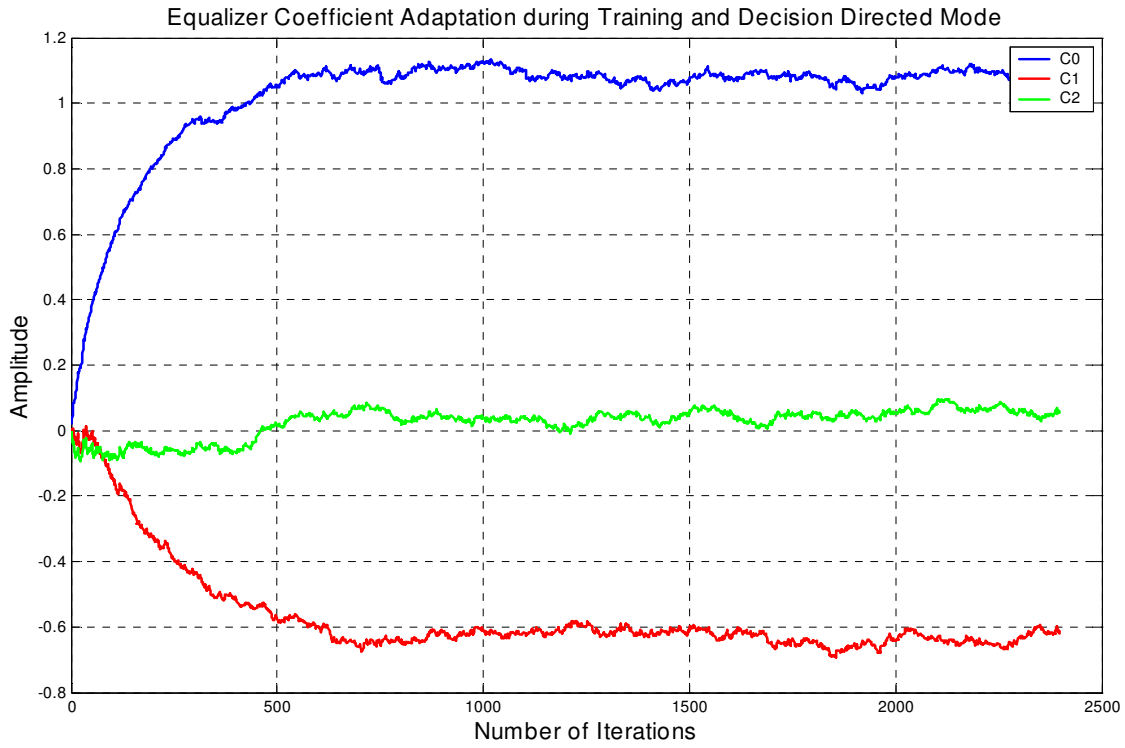
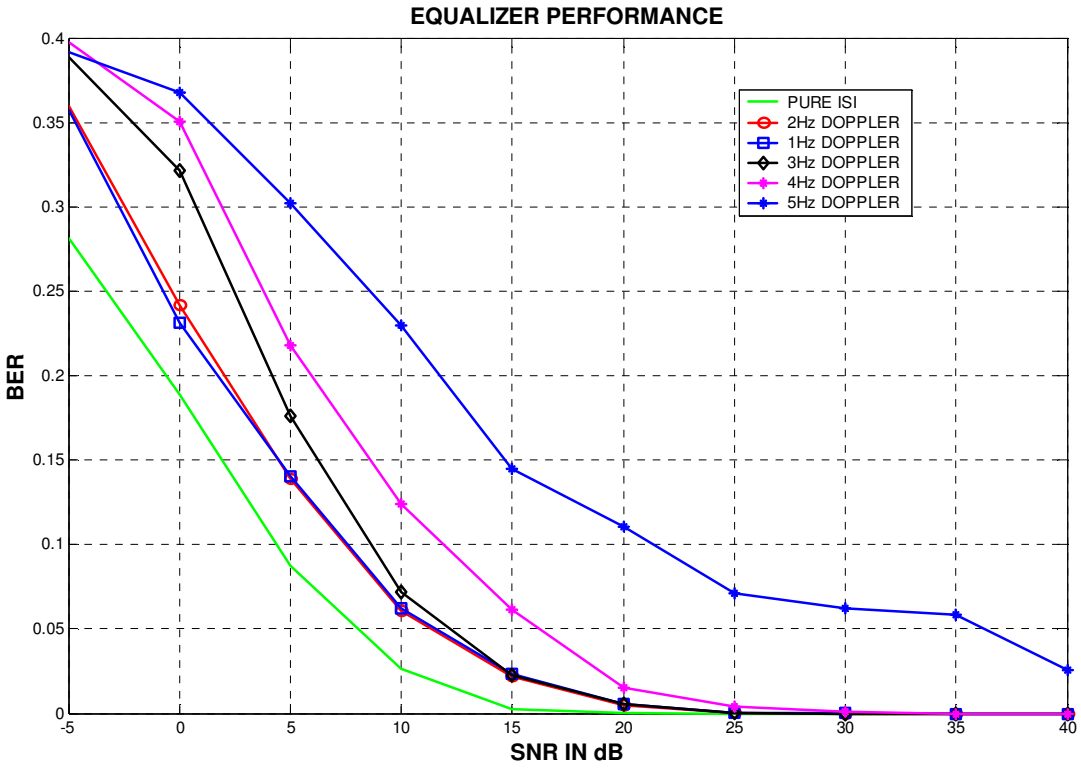
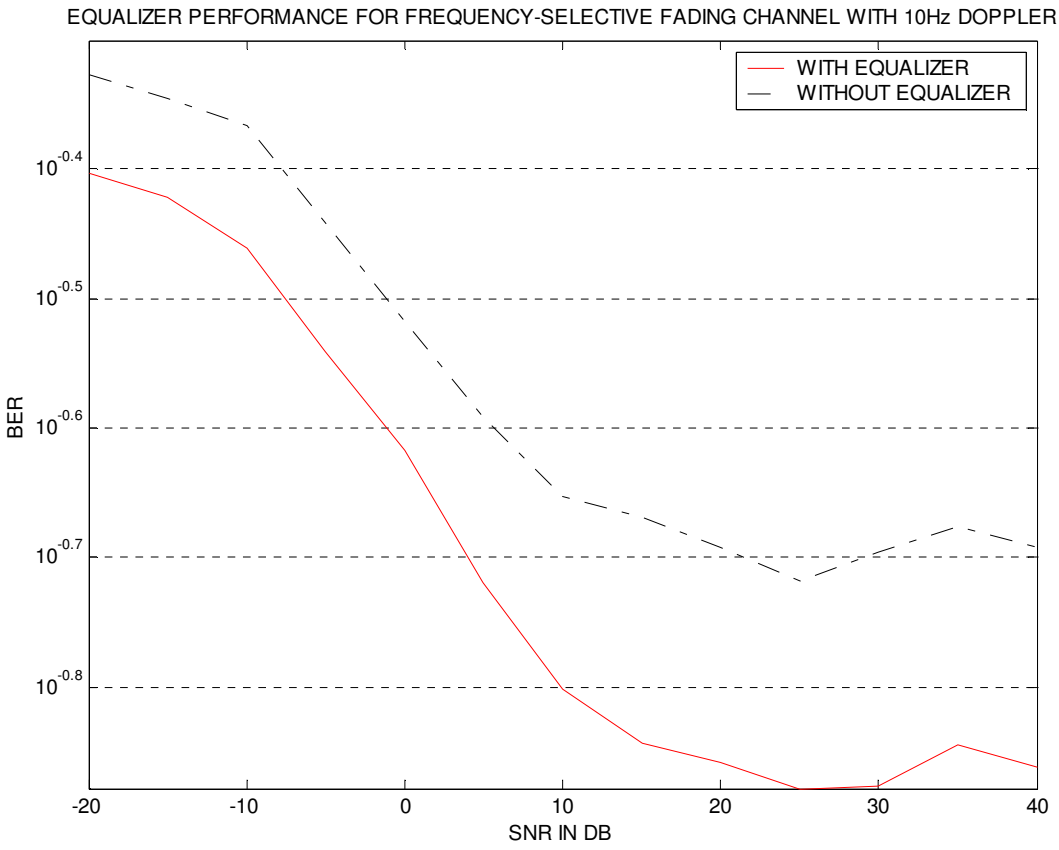


Fig. 6.7. *Equalizer coefficient convergence*

A MATLAB simulation is also made to study the system performance due to Doppler Spread of a time-variant channel. Despite of the bursty nature error pattern for fading channels BER results for different noise and fading processes were found to deviate only 10-20% from their mean settings. Thus the BER curve presented below for a 10Hz Doppler Spread can claim to be fairly reliable as averaging was performed over several deep fades. For slowly fading channel of 10Hz maximum Doppler the BER curves are at least 4dB (BER 10^{-2}) to 6dB (BER 10^{-3}) away from optimal matched filter bound. The curve reflects the sub optimality of the equalizer itself, even for perfectly adjusted coefficients, and serves to compare digital equalizers realized using DSP.


EQUALIZER PERFORMANCE
BER
SNR IN dB
PURE ISI
2Hz DOPPLER
1Hz DOPPLER
3Hz DOPPLER
4Hz DOPPLER
5Hz DOPPLER
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-5
0
5
10
15
20
25
30
35
40

(a)



(b)

Fig. 6.8. *Performance comparison of a system with and without equalizer for a frequency-selective channel.*

The above simulation result ascertains that linear equalizers do not perform well on channels with deep spectral nulls in their amplitude characteristics or with nonlinear distortion, fading channels. Nonlinear equalizers, like decision feedback equalizer (DFE), are, however, superior to linear equalizers in applications where the channel distortion is severe for a linear equalizer to deal with. [7]

CHAPTER SEVEN

CONCLUSIONS AND RECOMMENDATIONS

The problem of how to achieve efficient digital communication over the time-variant frequency selective Rayleigh fading HF channel has been treated. The basic motivation behind this work is underutilization of HF channel, a channel with relatively low cost and wide area coverage, for commercial application, particularly in Ethiopia. This channel has in fact, apart from being used for analog AM shortwave broadcasting, amateur radio and extremely low bit rate data transmissions like CW, RTTY, AMTOR, PACTOR, CLOVER, been largely avoided. The reason for this is probably its traditional reliance on manual frequency management and the high error rates normally associated with HF transmission. Make use of this channel for high throughput and low bit error rate communication systems is practically impossible. That is certainly the case if systems originally designed for AWGN are used. Problems arise since matched filter detectors lead to high error floors. However, implementation of adaptive digital equalizers in receivers gives performance improvements far from those needed. [4, 16]

In that respect, a conclusion of this work is that high throughput transmission with low bit error digital communication over slow fading, dispersive HF channel is indeed possible. However, it is not only possible, but a good performance can also be achieved, provided the properties of the channel are carefully considered in the design process. A thorough treatment of a suboptimal linear adaptive digital equalizer, MMSE with LMS coefficient adaptation algorithm, and its efficient realization using TMS320C50 DSP kit for a linearly modulated signal, BPSK, over a slowly fading, frequency selective Rayleigh fading HF channel model as a starting point, has for that reason been the main thread through this thesis.

7.1. Conclusions

In HF radio system, multiple reflections from the sky and ground leads to a situation where several incoming paths of transmission are superimposed at the receiver. This gives in general rise to both Rayleigh fading and intersymbol interference. Furthermore, the ionospheric turbulence due continuous ion motion makes the intersymbol interference as well as the Rayleigh fading time-varying. Thus, this time-variation implies that not only time- but also frequency - dispersion is introduced to the transmitted signal.

The channel was modeled and simulated under the assumption of perfect knowledge of the statistical properties of the channel, as estimation errors of the noise variance and the shape of the fading spectrum led to very small performance losses. Therefore, a three-tap channel model with a mean tap weights derived form the delay autocorrelation function, delay power spectrum, and their relative variation from the Doppler spread, autocorrelation on absolute time variable, of a typical HF channel was considered for this thesis work, and it is proved to model satisfactorily the real physical channel statistics.

To mitigate ISI problems, many algorithms and structures have been suggested over the last three decades. In this paper, however, performance of a linear adaptive linear equalizer, MMSE with LMS coefficient adaptation algorithm was studied for different channel characteristics, and it is realized on a TMS320C50 DSP kit. As described in the result section of this thesis, chapter six, the MATLAB simulation indicated that a linear MMSE equalizer is capable of combating channel induced ISI to an acceptable performance level that cannot otherwise be attained by increasing the transmitted signal energy. It can also be noted that for a receiver without equalizer the BER will never be below 20% even for high SNR, as additional power amplifies the ISI in step with the desired signal.

However, the selected linear equalizer did not perform well on channels with deep spectral nulls in their amplitude characteristics or with nonlinear distortion. For the three-tap multipath time-variant channel with a maximum Doppler spread of 10 Hz, the channel distortion is found to be severe for the selected linear equalizer to deal with.

In general from the results of this thesis work, we can infer that:

- For a long range HF communication, severe multipath propagation may yield a time spread of up to 6 ms. For a nominal HF bandwidth of 2 kHz and transmission rate of 2.4 kbps it will smear about 16 symbols, large ISI.
- For a purely frequency selective channel, a linear equalizer can reduce the channel induced ISI satisfactorily.
- For a channel with a Doppler spread exceeding 2 Hz, maximum Doppler spread for a typical HF channel at low latitudes, the performance of a linear equalizer is well below the requirements of most application. The equalizer failed to track the channel variation as the channel distortion is no more linear.
- As the HF channel Doppler spread, especially for low latitudes, is well below 2 Hz. Thus, a linear adaptive digital equalizer can serve its purpose for most applications. Linear equalizers are superior to others for their simplicity and less computational complexity.
- A Texas instrument DSPs, such as TMS320C50, can be used to efficiently realize linear equalizers.

7.2. Recommendations

High data rate through HF channel would be extremely beneficial to both the Government and civilian sectors. A data rate of the order of 64 kbps would be compatible with primary T1 standard in U.S. and would allow integration with the current and future telephone, Internet and networking systems. This will allow multimedia (including images) transmission over long distances, over difficult terrain and structures (through LOS, BLOS) over the ocean through surface waves, amongst others. This could be used in the battlefield as well as for the "last mile" problems in wire-line telephony populated areas. Data rates of 64 kbps are comparable to many satellite based communication systems and the HF can also be used as a back up during satellite overload, jamming and during emergencies.

Maximum useful data rate available through a typical HF channel is currently limited to 4800 bps (2400 bps to be more realistic) and several dramatic improvements are needed for achieving 64 kbps. A bandwidth of the order of 3 to 4 kHz is generally assigned for a HF channel and the

bandwidth efficient modulation schemes (*M*-array) cannot be extended too far for a practical HF channel limited by interference, noise, adjacent channel requirements, channel dispersion, multipath and so on. An adaptive system, however, could be developed for much improved throughput over a single channel.

Larger effective bandwidths are typically achieved through spread spectrum and through frequency hopping. Conventional frequency hopping does not provide bandwidth enhancements and multiple frequency slots need to be used for larger effective bandwidths.

In this thesis work, designing and realization of a linear adaptive digital channel equalizer for a typical HF channel is treated, and simulation result indicated that it is not able to deal with nonlinear channel distortions, like large ISI and wide Doppler spread. Nonlinear equalizers are, however, superior to linear equalizers in applications where the channel distortion is severe for a linear equalizer to deal with. A decision feedback equalizer (DFE), for instance, is a nonlinear equalizer widely used in situation where the Inter-Symbol Interference (ISI) is large. Performance analysis, design and implementation of nonlinear equalizers for HF channel can be considered in the future works.

Extension of this work to include more general systems, equalizer performance for coded signals, both error and channel coding, and for nonlinear modulation techniques is an interesting topic for further research.

BIBLIOGRAPHY

1. Andreas Antonio, *Digital Filters Analysis and Design*, McGraw-Hill, Inc., New York, 1990.
2. Clark, A.P. and Hariharan, S., *Adaptive Channel Estimator for an HF Radio Link.* IEEE Trans. Commun., Vol.37, pp. 918-926, Sep.1989.
3. Clark, A.P. and McVerry, F., *Channel Estimation for an HF Radio Link,* IEE Proc., Vol. 128, pp. 33-42, Feb.1981.
4. Edward B.Kalin, *The American Radio Relay League Hand Book CD,* The ARRL, Inc., 1997.
5. Eleftheriou, E. and Falconer, D.D., *Adaptive Equalizer Techniques for HF Channels.,* IEEE J. Sel. Areas Commun., Vol.SAC-5, pp. 238-247, Feb. 1987.
6. Emmanuel C. Ifeachor and Barrie W. Jervis, *Digital Signal Processing: A Practical Approach*, Addison Wesley Longman Ltd., Harlow, England, 1993.
7. Falconer, D.D. et. al., *Comparison of DFE and MLSE Receiver Performance on HF Channel,* IEEE Trans. Commun., Vol. COM-33, No. 5, pp. 845-848, May 1985.
8. Fechtel, S.A. and Meyr, H., *An Investigation of Channel Estimation and Equalization for Moderately Fading HF-Channels, In Proc. IEEE Int. Conf. Commun., ICC'91*, pp. 768-772, Denver, CO, USA, Jun. 1991.
9. Goldberg, B., *300kHz-30MHz MF/HF,* IEEE Trans. Commun. Technol., Vol.COM-14, pp. 767-784, Dec. 1966.
10. Goldberg, B., *300kHz-30MHz MF/HF. IEEE Trans. Commun., Technol.*, Vol. COM-14, pp. 767-784, Dec. 1966.
11. Greenstein, L.J., *A Multipath Fading Channel Model for Terrestrial Digital Radio Systems*, IEEE Trans. Commun., Vol. COM-26, Aug. 1978.
12. Hsu, F.M. et. al., *Adaptive Equalization Techniques for high-speed Transmission on Fading dispersive HF Channels,* In Proc. NTC, pp. 58.1.1-7, Dec.1980.
13. Hsu, F.M. *Square-Root Kalman Filtering for high-speed Transmission on Fading Dispersive HF Channels.* IEEE Trans. Inform. Theory, Vol. IT-28, pp. 753-763, Sep. 1982.

14. Kaveh Pahlavan and Allen H. Levesque, *Wireless Information Networks*, John Wiley & Sons, Inc., New York, 1995.
15. Kun-Shan Lin, *Digital Signal Processing Applications, with the TMS320 Family,* Volume I, Prentice-Hall, Inc., New Jersey, 1987.
16. Lothar Wiesner, *Telegraph and Data Transmission over Short Wave Radio Link,* Hayden & Son Ltd., London, 1979.
17. Oppenheim, Alan V., and R.W., Schafer, *Digital Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1983.
18. Papoulis, A., *Probability, Random Variables, and Stochastic Processes*, McGraw Hill, New York, 1984.
19. Proakis, J .G., *Digital Communications,* McGraw-Hill, NY, 1989.
20. Qureshi, S.U.H., *Adaptive Equalization.* IEEE Proc., Vol. 73, pp. 1349-1387, Sep.1985.
21. S.M. Bozic, *Digital and Kalman Filtering,* Edward Arnold, Plc., London, 1994.
22. Sanjit K. Mitra and James F. Kaisier, *Hand Book for Digital Signal Processing,* John Wiley & Sons, Inc., New York, 1993.
23. Sophocles J. Orpahnidis, *Introduction to Signal Processing,* Prentice-Hall, Inc., New Jersey, 1996.
24. Texas Instruments Incorporated, *TMS320C5X DSP Starter Kit User's Guide,* Customs Printing Company, Missouri, 1996.
25. Texas Instruments Incorporated, *TMS320C5X User's Guide,* Customs Printing Company, Missouri, 1996.
26. Trevor Housley, *Data Communication and Teleprocessing Systems,* Prentice-Hall, Inc., New Jersey, 1979.
27. Tront, R.J. et. al., *Performance of Kalman Decision-Feedback Equalization in HF Radio Modems,* In Proc. NTC, pp. 1617-1621, 1986.
28. Watterson, C.C., Juroshek, W.D., and Bensema, W.D. *Experimental Confirmation of an HF Channel Model*, IEEE Trans. Commun., Vol. COM-18, pp. 792-803, Dec. 1970.
29. Widrow, B., "*Adaptive Filters, I: Fundamentals*," Stanford Electronics Laboratory, Stanford University, Technical Report 6764-6, Dec. 1966.

APPENDIX A

A.1 The TMS320C5X Internet Hardware Summary

Unit	Symbols	Function
Accumulator	ACC (32) ACCH (16) ACCL (16)	A 32-bit accumulator accessible in two halves: ACCH (accumulator high) and ACCL (accumulator low). Used to store the output of the ALU.
Accumulator Buffer	ACCB (32)	A register used to temporarily store the 32-bit contents of the accumulator. This register has a direct path back to the ALU and therefore can be arithmetically or logically acted upon with the ACC.
Arithmetic logic unit	AUL	A 32-bit 2s-complement arithmetic logic unit having two 32-bit input ports and one 32-bitoutput port feeding the accumulator.
Auxiliary Register Arithmetic unit	ARAU	An unsigned 16-bit arithmetic unit used to calculate indirect addresses using the auxiliary, index, and compare registers as inputs.
Auxiliary Register Compare	ARCR (16)	A register used as a limit to compare indirect address against.
Auxiliary Register File	AUXREGS	A register file containing 16-bit auxiliary register (ARO-AR7) used for indirect data address pointers, temporary storage, or integer arithmetic processing through the ARAU.
Auxiliary Register Buffer	ARB (3)	A 3-bit register that holds the previous value contained in the ARP. These bits are stored in ST1.
Auxiliary Register Pointer	ARP (3)	A 3-bit register used as a pointer to the currently selected auxiliary register. These bits are stored in ST0.
Block Move Address Register	BMAR (16)	A 16-bit register that holds an address value for use with block moves or multiply/accumulates.
Block Repeat Active Flag	BRAF (1)	A 1-bit flag indicating that a block repeat is currently active. This bit is normally set when the RPTB instruction executed and cleared when the BRCT register decrements below zero. This bit resides in the PMST register.
Block Repeat Address End Registers	PAER (16)	A 16-bit memory mapped register containing the end address of the segment of codes being repeated.
Block Repeat Address Start Registers	PASR (16)	A 16-bit memory mapped register containing the start address of the segment of codes being repeated.
Block Repeat Counter Register	BRCR (16)	A 16-bits memory-mapped counter register used to limit the number of times the blocks is to repeated .
Bus Request	BR	This signal indicates that a data access is mapped to global memory space as defined by the GREG register.
Carry	C	The signal indicates that a data access is mapped to global memory space as defined by the GREG register .
Central Arithmetic Logic Unit	CALU	The grouping of the ALU, multiplier, accumulator, and scaling shifters.
Circular Buffer Control Register	CBCR (8)	An 8-bit register used to enable/disable the circular buffers and define which auxiliary registers are mapped to the circular buffers.

Unit	Symbols	Function
Circular Buffer End Address	CBER (16) CBER1(16) CBER (16)	Two 16-bit registers indicating circular buffer end addresses. CBSR1 and CBSR2 are associated with circular buffers 1 and 2, respectively.
Circular Buffer Start Address	CBSR (16) CBSR1(16) CBSR (16)	Two 16-bit registers indicating circular buffer start addresses. CBER1 and CBER2 are associated with circular buffers 1 and 2, respectively.
Configure RAM	CNF	This bit indicates whether on-chip dual access RAM blocks are mapped to program or data space. This CNF bit resides in ST1.
Data Bus	DATA	A 16-bit bus used to route data.
Data Memory	DATA MEMORY	This block refers to data memory used with the core and defined in specific device descriptions. It refers to both on-and-off-chip memory blocks in data memory space.
Data Memory Address Bus	DATA ADDRESS	A 16-bit bus that carries the address for data memory access.
Data memory Address Immediate Register	Dam (7)	A 7-bit register containing the immediate relative address with in a 128-word data page.
Data Memory Page Pointer	DP (9)	A 9-bit register containing the address of the current page. Data pages are 128-wordseach, resulting in 512 pages of addressable data memory space (some locations are reserved).
Data RAM map bit	RAM (1)	This bit indicates that if the single-access RAM is mapped into data space.
Dynamic Bit Pointer	TREG2 (4)	A 4-bit register that hold a dynamic pointer for the BITT instruction.
Dynamic Bit Manipulation Register	DBMR (16)	A 16-bit memory-mapped register used as a mask input to the PLU in the absence of a long immediate value.
Dynamic Shift Count	TREG (5)	A 5-bit register that hold a dynamic prescaling shift count for data inputs to the ALU.
External Flag	XF (1)	This bit drives the level of external flag pin and resides in ST1.
Global Memory Allocation Register	GREG (8)	An 8-bit memory-mapped register for specifying the size of global memory space.
Index Register	INDX (16)	This 16-bit memory-mapped register specifies increment sizes greater than 1 for indirect addressing updates. In bit-reversed addressing, the index register define the array size.
Interrupt Flag Register	IFR (16)	A 160bit flag register used to latch the active-low interrupts. The IFR is a memory-mapped register.
Interrupt Mask Bit	INTM (1)	This interrupt mask bit globally masks or enables all interrupts. This bit resides in ST0.
Interrupt Pointer	IPTR (5)	Five bits pointing to the 2k pages where the interrupt vectors currently resides in the system. These bits reside in the PMST register.
Interrupt Mask Register	IMR (16)	A 16-bit memory mapped register to mask interrupts.
Microprocessor/ Microcomputer mode	MP/MC	This bit resides in the PMST register and indicates whether the on-chip ROM is mapped into program memory address space.
Multiplier	MULTIPLIER	A 16 x 16-bit parallel multiplier.
Overflow Flag	OV (1)	This bit resides ST0 and indicates an over flow in arithmetic operation in the ALU.

Unit	Symbols	Function
Overflow Mode	OVM (1)	This bit resides on ST0 and determines whether an overflow in the ALU will wrap around or saturate.
Overlay to Data Space	OVLY (1)	This bit resides in the PMST register and determines whether the on-chip single-access memory will addressable in data address space.
Parallel Logic Unit	PLU	A 16-bit logic unit that executes logic operations from either long immediate operands or the contents DBMR directly upon data location without interfacing with the contents of the CALU registers.
Prefetch Counter	PFC (15-0)	A 16-bit counter used to prefetch program instruction. The PFC contains the address of the instruction currently being prefetch. It is updated when a new prefetch is initiated. The PFC can also address program memory when the block move (BLPD). Multiply-accumulate (MAG/MACD), and table read/write (TBLR/TBLW) instructions are used and can address data memory when the block move (BLDD) instruction is used.
Prescalar Count Register	COUNT (4)	A four-bit register that contains the value for the prescaling operation. When the register contents are used as prescaling data, this register is loaded from the dynamic shift count or from the instruction. In conjunction with BIT and BITT instructions, this register is loaded from the dynamic bit pointer or the instruction word.
Product Register	PREG (32)	A 32-bit product register used to hold the multiplier's product. The high and low words of the PREG can be accessed individually.
Program Bus	PROG DATA	A 16-bit bus used to route instruction (and data for the MAC and MACD instructions).
Program Memory	PROGRAM MEMORY	The block refers to program memory used with the core and defined in specific device descriptions. It refers both on- and off-chip memory blocks accessed in program memory space.
Program Memory Address Bus	PROG ADDRESS	A 16-bit bus that carries the address for program memory access.
Prescaling Shifter	PRESCALER	A 0- to 16-bit left barrel shifter used to prescale data coming into the ALU. Also used to align data for multi-precision operations. This shifter is also used as a 0- to 16-bit right barrel shifter of the ACC.
Postscaling Shifter	POSTSCALER	A 0-to 7-bit left barrel shifter used to postscale data coming out of the CALU.
Product Shifter	P-SCALER	A 0-, 4- or 4-bit left shifter that can remove extra sign bits (gained in the multiply operation) when fixed-point arithmetic is used; or a 6-bit right shifter that can scale the product down to avoid overflow in the accumulation process.
Product Shift Mode	PM (2)	These two bits define the product shift mode. They reside in ST1.
Repeat Counter	RPTC (16)	A 16-bit counter used to control the repeated execution of a single instruction.
Sign Extension Mode	SXM (1)	This bit resides in ST1 and control whether the arithmetic operation will be sign-extended or not. See subsection 3.6.3 for more information.
Stack	STACK	An 8 x 16-bit hardware stack used to store the PC during interrupts and calls. The ACCL and data memory values may also be pushed onto and popped out from the stack.
Status Registers	ST0, ST1, PMST	Three 16-bit status registers that contain status and control bits.
Temporary Multiplicand	TREG0 (16)	A 16-bit register that temporarily holds an operand for the multiplier.

Unit	Symbols	Function
Temporary Register Enable	TRM (1)	This bit defines whether an LT (A,D,P,S) instruction loads all three of the TREGs (0,1,2) to maintain compatibility with the 'C25 or loads just TREG0. This bit resides in the PMST register.
Test/Control Flag	TC (1)	This bit resides in ST1 and stores the results of ALU or PLU test bit operations.

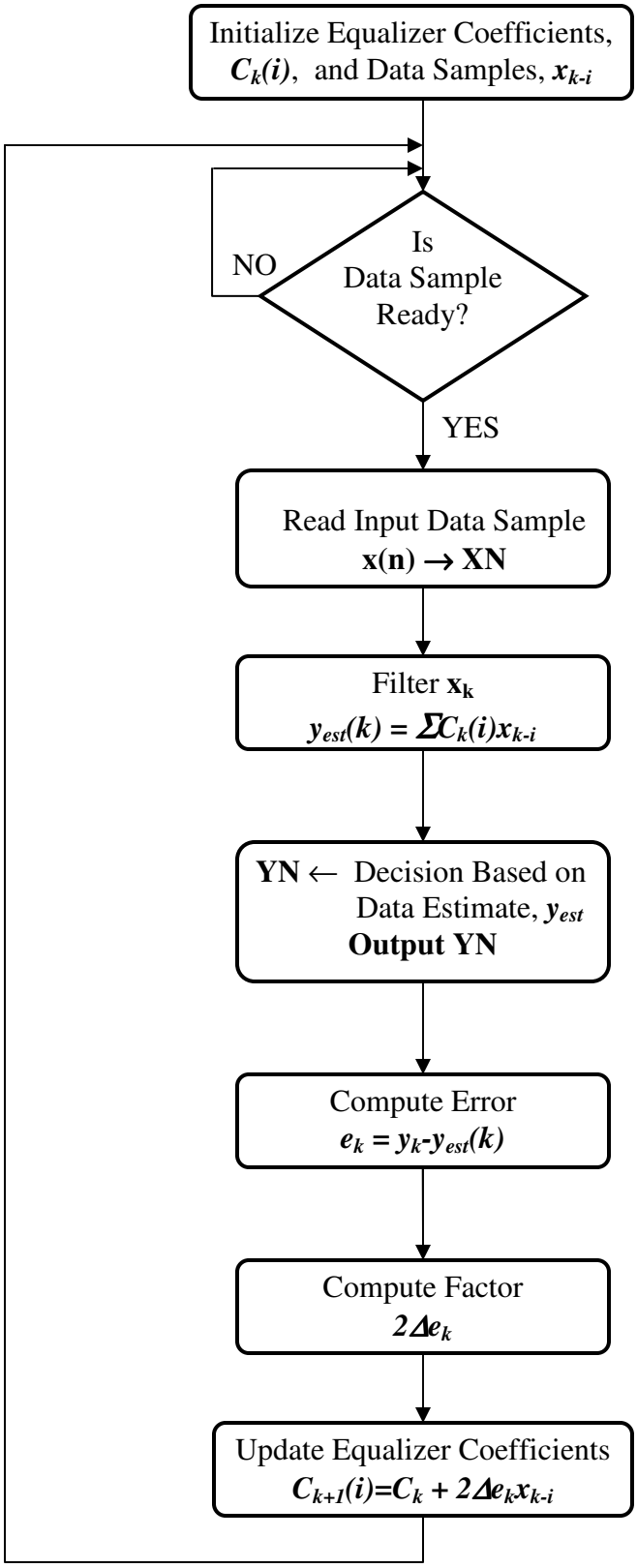
A.2 The TMS320C50 Instruction Set Summary

Accumulator Memory Reference Instruction		
Mnemonics	**Description**	**Words**
ADCB	Add ACCB to ACC with carry	1
ADD	Add to ACC	½
AND	AND with ACC	½
ANDB	AND ACCB with ACC	1
LACB	Load ACC with ACCB	1
LACC	Load ACC with shift	½
LAMM	Load ACC with memory mapped registers	1
NEG	Negate ACC	1
NORM	Normalize contents of ACC	1
OR	OR with ACC	½
SACB	Store ACC in ACCB	1
SACH	Store ACC high with shift	1
SACL	Store ACC low with shift	1
SAMM	Store ACC in memory-mapped registers	1
SFL	Shift ACC left	1
SFR	Shift ACC right	1
XOR	Exclusive-OR with ACC	½
ZAP	Zero ACC and PREG	1
Auxiliary Registers and Data Page Pointer Instructions		
Mnemonics	**Description**	**Words**
CMPR	Compare ARn with ARCR	1
LAAR	Load ARn	½
LDP	Load data page pointer	1
MAR	Modify ARn	1
SAR	Store ARn	1
APL	AND DBMR or constant with data memory value	½
CPL	Compare DBMR or content with data memory value	½
OPL	OR DBMR or constant with data memory value	½
SPLK	Store long immediate to data memory location	2
T Register, P Register and Multiply Instructions		
Mnemonics	**Description**	**Words**
APAC	Add PREG to ACC	1
LTD	Load TREG0, accumulate previous product, and move data	1
MAC	Multiply and accumulate	2
MACD	Multiply and accumulate with data move	2
MADD	Multiply and accumulate with source pointed at by BMAR	1
MPY	Multiply	½
MADS	Multiply and accumulate with source pointed at by BMAR and with data move	1
Branch Instructions		
Mnemonics	**Description**	**Words**
BAND[D]	Branch conditionally	2
CALD[D]	Call subroutine indirect	1
CALL[D]	Call subroutine	2
CC[D]	Call conditionally	2
INTR	Soft interrupt	1

RETC[D]	Return conditionally	1
RETE	Return with context switch and global interrupt	1
TRAP	Software interrupt	1
XC	Execute next instruction(s) conditionally	1
I/O and Data Memory Operations		
Mnemonics	**Description**	**Words**
BLDD	Block move from data memory to data memory	½
BLDP	Block move from data memory to program memory	1
BLPD	Block move from program memory to data memory	½
DMOV	Data move in data memory	1
IN	Input data from port	2
LMMR	Load memory-mapped registers	2
OUT	Output data to port	2
SMMR	Store memory-mapped registers	2
TBLR	Table read	1
TBLW	Table write	1
Control Instructions		
Mnemonics	**Description**	**Words**
BIT	Test bit	1
BITT	Test bit specified by TREG2	1
CLRC	Clear control bit	1
IDLE	Idle until interrupt	1
IDLE2	Idle until interrupt - low power mode	1
LST	Load status register	1
NOP	No operation	1
POP	Pop PC stack to low accumulator	1
POPD	Pop top of stack to data memory	1
PSHD	Push data value to PC stack	1
PUSH	Push low ACC to PC stack	1
RPT	Repeat next instruction	½
RPTB	Repeat block	2
RPTZ	Repeat next instruction and clear ACC and PREG	2
SETC	Set control bit	1
SST	Store status register	1

APPENDIX B

Flow Chart for TMS320C50 Adaptive Channel Equalizer




```
LAMM  DRR
SACL  XN      ; Read data
```

```
LAR AR3, #LSTD_SMPL
RPT  #N
MACD COEFFP, * --
SACH YN,1
```

NB.

- XN – data memory address for the new sample.
- YN – data memory address for the computed output.
- COEFFP - coefficient program memory address.
- LSTD_SMPL – data memory address for last data sample.
- N – Filter Length.

APPENDIX C

Source Code for Implementing Adaptive channel Equalizer on TMS320C50

```
00001 ---- ----      ***************************************************
00002 ---- ----       *                                                 *
00003 ---- ----      * This program implements a three-tap adaptive linear channel      *
00004 ---- ----      * equalizer for a BPSK Modulated signal with a transmission rate  *
00005 ---- ----      * of 2.4 kbps over a  2 kHz frequency selective time-variant       *
00006 ---- ----      * channel with 10 Hz Doppler frequency. LMS algorithm is used     *
00007 ---- ----       * for coefficient updating. The channel SNR is considered to be   *
00008 ---- ----      * 10 dB. The initial filter coefficients are derived from MATLAB  *
00009 ---- ----       * 100 symbols are considered training sequences.                 *
00010 ---- ----       *                                                               *
00011 ---- ----      ***************************************************
00012 ---- ----          .mmregs
00013 ---- ----  *
00014 ---- ----  ;---------------------------------------------------------------
00015 ---- ----  ;
00016 ---- ----
00017 ---- 0f00             .ds     0f00h
00018 0f00 001a  TA       .word    26                      ; Values to initialize the AIC
00019 0f01 001a  RA       .word    26                       ; registers fora sampling frequency
00020 0f02 0011  TB       .word    17                       ; of 2.4 KHz, One sample per symbol
00021 0f03 0011  RB       .word    17                      ; period (center sample).
00022 0f04 0028  AIC_CTR  .word    28h
00023 ---- ----  *
00024 0f05 0000  ERR      .word     0                       ; location for error
00025 0f06 0000  ERRF     .word     0                       ; location for error function
00026 0f07 0000  DELTA    .word    0.0008                   ;adjustment factor
00027 0f08 1eb8  COEFFP             .q15       0.24,-0.065      ;Initial Equalizer Coefficintes
      0f09 f7af
00028 0f0a 047a             .q15       0.035                 ;Initial Equalizer Coefficients
00029 0f0b 0000  BIN_1    .q15       0,1                     ;Signal level for binary one
      0f0c 8000
00030 0f0d f334  BIN_0    .q15       -0.1                    ;Signal level for binary zero
00031 ---- ----             .include "TR_DATA.ASM"          ;A file containing 100 training symbols.
*****************************************
*     OPENING INCLUDE FILE TR_DATA.ASM
*****************************************************************************
00001 ---- ----  ************************************************************
00002 ---- ----  *This file contains the training symbols to initialize                *
00003 ---- ----  *the equalizer coefficients                                           *
00004 ---- ----  ************************************************************
00005
00006 0f0e f334             .q15      -0.1
00007 0f0f 0ccc             .q15       0.1
00008 0f10 0ccc             .q15       0.1
00009 0f11 f334             .q15      -0.1
00010 0f12 0ccc             .q15       0.1
00011 0f13 0ccc             .q15       0.1
00012 0f14 f334             .q15      -0.1
00013 0f15 0ccc             .q15       0.1
```

```
00014 0f16 0ccc             .q15      0.1
00015 0f17 f334             .q15      -0.1
00016 0f18 0ccc             .q15      0.1
00017 0f19 f334             .q15      -0.1
00018 0f1a 0ccc             .q15      0.1
00019 0f1b f334             .q15      -0.1
00020 0f1c 0ccc             .q15      0.1
00021 0f1d 0ccc             .q15      0.1
00022 0f1e 0ccc             .q15      0.1
00023 0f1f f334             .q15      -0.1
00024 0f20 f334             .q15      -0.1
00025 0f21 0ccc             .q15      0.1
00026 0f22 f334             .q15      -0.1
00027 0f23 0ccc             .q15      0.1
00028 0f24 0ccc             .q15      0.1
00029 0f25 f334             .q15      -0.1
00030 0f26 0ccc             .q15      0.1
00031 0f27 0ccc             .q15      0.1
00032 0f28 f334             .q15      -0.1
00033 0f29 f334             .q15      -0.1
00034 0f2a 0ccc             .q15      0.1
                                .
                                .
                                .

00035 ---- ----             .listoff
00107 0f71 f334             .q15      -0.1
00108 0f72 0ccc             .q15      0.1
00109 0f73 f334             .q15      -0.1
00110 0f74 0ccc             .q15      0.1
00111 0f75 0ccc             .q15      0.1
00112 0f76 f334             .q15      -0.1
00113 0f77 f334             .q15      -0.1
00114 0f78 0ccc             .q15      0.1
00115 0f79 f334             .q15      -0.1
00116 0f7a 0ccc             .q15      0.1
>>>>> FINISHED READING ALL FILES
****************************************
*     CLOSING FILE TR_DATA.ASM
****************************************
00032 0f7b 0000  DATA_EST .word    0                          ;Estimated data sample.
00033 0f7c 0000  YN       .word    0                           ;Equalized output data sample.
00034 0f7d 0000  XN       .word    0,0                         ;Initial data samples.
    0f7e 0000
00035 0f7f 0000  XNLAST       .word        0                  ;Initialize the last data sample
00036 ---- ----  *
00037 ---- ----  ***************************************************************
00038 ---- ----  *   Set up the ISR vector                                     *
00039 ---- ----  ***************************************************************
00040 ---- 080a              .ps     080ah
00041 080a 7980  rint:    B       RECEIVE                    ;0A; Serial port receive interrupt RINT.
    080b 0000
00042 080c 7980  xint:    B       TRANSMIT                   ;0C; Serial port transmit interrupt XINT.
    080d 0000
00043 ---- ----  *
00044 ---- ----  *
```

```
00045 ---- ----  ***************************************************************
00046 ---- ----  * TMS32C050 INITIALIZATION                                     *
00047 ---- ----  *                                                             *
00048 ---- ----  ***************************************************************
00049 ---- 0a00          .ps 0a00h
00050 ---- ----         .entry
>>>>> ENTRY POINT SET TO 0a00
00051 0a00 be41  START:  SETC   INTM                ; Disable interrupts
00052 0a01 bc00          LDP    #0                 ; Set data page pointer
00053 0a02 5d07          OPL    #0834h,PMST
    0a03 0834
00054 0a04 bf80          LACC   #0
    0a05 0000
00055 0a06 882a          SAMM   CWSR               ; Set software wait state to 0
00056 0a07 8828          SAMM   PDWSR              ;
00057 0a08 ae04          SPLK   #022h,IMR          ; Using XINT syn TX & RX
    0a09 0022
00058 0a0a 7a80          CALL   AICINIT            ; initialize AIC and enable interrupts
    0a0b 0000
00059 ---- ----  *
00060 ---- ----  ***************************************************************
00061 ---- ----  * This routine enables serial port rx interrupts & configures   *
00062 ---- ----  * TLC32040 for the frame sync. When RINT is triggered, read a   *
00063 ---- ----  * data word from the AIC then equalize the data, filter using   *
00064 ---- ----  * FIR/LTE and send out the equalized symbol.                    *
00065 ---- ----  ***************************************************************
00066 ---- ----  *
00067 0a0c be42          CLRC   OVM                ; OVM = 0
00068 0a0d bf00          SPM    0                  ; PM = 0
00069 0a0e ae04          SPLK   #012h,IMR          ; This turns on receive interrupt only
    0a0f 0012
00070 0a10 be40          CLRC   INTM               ; Enable interrupt mask bit
00071 0a11 b064          LAR    AR0,#100           ; Specifying length of training symbols
00072 0a12 bf09          LAR    AR1,#TRNGSMBL      ; Start address for training symbols
    0a13 0f0e
00073 ---- ----  *
00074 ---- ----  *--------------------------------Main program----------------------------------------------
00075 0a14 7980  WAIT    B      WAIT               ; wait for a receive interrupt
    0a15 0a14
00076 ---- ----  *
00077 ---- ----  *---------------------------- end of main program ------------------------------------------
00078 ---- ----  *
00079 ---- ----  *
00080 ---- ----  *------------- RECIEVER INTERRUPT SERVICE ROUTINE-----------------------
00081 ---- ----  *
00082 ---- ----  RECEIVE:
00083 0a16 bc1e          LDP    #XN
00084 0a17 be40          CLRC   INTM
00085 0a18 0820          LAMM   DRR                ; Load accumulator with word received from AIC
00086 0a19 907d          SACL   XN                 ; Store the value of received word to a variable
00087 ---- ----  *
00088 ---- ----  *--------------------MSE EQUALIZER, FIR FILTERING----------------------
00089 ---- ----  *
00090 0a1a be58  ADPFIR          ZPR
00091 0a1b bf8e                 LACC   #1,14       ; Load output rounding bit
    0a1c 0001
```

```
00092 0a1d 8b8b                 MAR     *,AR3
00093 0a1e 037f                 LAR     AR3,XNLAST   ; Point to oldest sample
00094 0a1f bb02  FIR            RPT     #2
00095 0a20 a390                 MACD  COEFFP,*-      ; 3-tap FIR/LTE equalizer
    0a21 0f08
00096 0a22 be04                 APAC                 ; Accumulate last product
00097 0a23 bc1e           LDP    #DATA_EST
00098 0a24 997b           SACH   DATA_EST,1             ; Store estimated data
00099 ---- ----  *                                      ; Shift once to get rid extra sign bit
00100 0a25 be1e           SACB
00101 0a26 407b           BIT     DATA_EST,0          ; Test sign bit for positive
00102 0a27 e100           BCND    ZERO,TC
    0a28 0000
00103 0a29 bc1e           LDP    #BIN_1
00104 0a2a 100b           LACC   BIN_1
00105 0a2b 7980           B       DECION
    0a2c 0000
00106 0a2d 100d  ZERO     LACC   BIN_0
00107 0a2e 907c  DECION   SACL    YN            ;       Decision made based on data estimate
00108 0a2f be1f           LACB
00109 0a30 be02           NEG                   ; Negate estimated symbol
00110 0a31 8b88           MAR    *,AR0
00111 0a32 7b80           BANZ   TRNG          ; Go to training subroutine
    0a33 0000
00112 0a34 8b8b           MAR    *,AR3
00113 0a35 bf0b           LAR     AR3,#YN
    0a36 0f7c
00114 0a37 2f80           ADD     *,15          ; Compute error
00115 0a38 bc1e           LDP    #ERR
00116 0a39 9905           SACH   ERR,1          ; Store error
00117 0a3a 7980           B       DATAOUT
    0a3b 0000
00118 0a3c 8b88  TRNG     MAR    *,AR0          ; Training session
00119 0a3d 8b99           MAR    *-,AR1
00120 0a3e a980           BLDD   *,#YN
    0a3f 0f7c
00121 0a40 2fa0           ADD     *+,15
00122 0a41 bc1e           LDP    #ERR
00123 0a42 9905           SACH   ERR,1
00124 ---- ----  *
00125 ---- ----  *--------------------SEND EQUALIZED DATA TO THE DAC--------------------------------
00126 ---- ----  *
00127 0a43 697c  DATAOUT       LACL    YN
00128 0a44 6e7c                AND     0fffch       ; Two LSBs must be zero for the AIC
00129 0a45 8821                SAMM  DXR
00130 ---- ----  *
00131 ---- ----  *-----------------------------------------------------------------------------------------------
00132 ---- ----  *THE FOLLOWING ROUTINE COMPUTES COEFFICIENT ADJUSTMENT.
00133 ---- ----  *LMS ALGORITH IS USED TO UPDATE FILTER COEFFICIENTS.
00134 ---- ----  *-----------------------------------------------------------------------------------------------
00135 ---- ----  *
00136 0a46 8b8d           MAR    *,AR5
00137 0a47 bc1e           LDP    #ERR
00138 0a48 7305           LT     ERR            ; T=err
00139 0a49 5407           MPY    DELTA          ; P=delta*err(i)
00140 0a4a be03           PAC                   ; errf(i)=delta*err(i)
```

```
00141 0a4b bf9e              ADD     #1,14                ; Round the result
    0a4c 0001
00142 0a4d 9806              SACH    ERRF                 ; save errf(i)
00143 ---- ----  *
00144 0a4e bf80              LACC    #2                   ; 3 coefficients to update
    0a4f 0002
00145 0a50 8809              SAMM    BRCR                 ; in the loop.
00146 0a51 bf0c              LAR     AR4,#COEFFP          ; Point to the coefficients
    0a52 0f08
00147 0a53 bf0d              LAR     AR5,#XLAST           ; Point the data sample
    0a54 0000
00148 ---- ----  *
00149 0a55 7306              LT      ERRF
00150 0a56 549c              MPY     *-,AR4               ; P=2*delta*err(i)*x(i-2)
00151 ---- ----  *
00152 0a57 bec6              RPTB    LOOP-1
    0a58 0000
00153 0a59 688d  ADAPT       ZALR    *,AR5                ; Load ACCH with ck(i)
00154 0a5a 509c              MPYA    *-,AR4
00155 0a5b 98a0              SACH    *+
00156 0a5c 688d  LOOP        ZALR    *,AR5
00157 0a5d ff00              RETD                         ; Delayed return
00158 0a5e be04              APAC                         ; ACC=c0(i)+2*delta*err(i)*x(i)
00159 0a5f 98a0              SACH    *+                   ; Save c0(i+1)
00160 ---- ----  *
00161 ---- ----  *
00162 ---- ----  *
00163 ---- ----  *----------- TRANSMIT INTERRUPT SERVICE ROUTINE--------------------------
00164 ---- ----  *
00165 ---- ----  TRANSMIT:
00166 0a60 be3a          RETE
00167 ---- ----  *
00168 ---- ----  *
00169 ---- ----  ********************************************************************
00170 ---- ----  *  DESCRIPTION: This routine initializes the TLC320C40 for         *
00171 ---- ----  *               a 2.4Khz sample rate with a gain setting of 1      *
00172 ---- ----  ********************************************************************
00173 ---- ----  *
00174 ---- ----  *
00175 0a61 ae26  AICINIT:          SPLK    #20h,TCR         ; To generate 10 MHz from Tout
    0a62 0020
00176 0a63 ae25                   SPLK    #01h,PRD         ; for AIC master clock
    0a64 0001
00177 0a65 8b88          MAR     *,AR0
00178 0a66 bf80          LACC    #0008h          ; Non continuous mode
    0a67 0008
00179 0a68 9022          SACL    SPC             ; FSX as input
00180 0a69 bf80          LACC    #00c8h          ; 16 bit words
    0a6a 00c8
00181 0a6b 9022          SACL    SPC
00182 0a6c bf80          LACC    #080h           ; Pulse AIC reset by setting it low
    0a6d 0080
00183 0a6e 9821          SACH    DXR
00184 0a6f 9005          SACL    GREG
00185 0a70 bf08          LAR     AR0,#0FFFFh
    0a71 ffff
```

```
00186 0a72 bec4          RPT     #10000          ; and taking it high after 10000 cycles
      0a73 2710
00187 0a74 1088          LACC    *,0,AR0         ; (.5ms at 50ns)
00188 0a75 9805          SACH    GREG
00189 ---- ----          ;------------------------
00190 0a76 bc1e          LDP     #TA             ;
00191 0a77 be47          SETC    SXM             ;
00192 0a78 1900          LACC    TA,9            ; Initialized TA and RA register
00193 0a79 2201          ADD     RA,2            ;
00194 0a7a 7a80          CALL    AIC_2ND         ;
      0a7b 0000
00195 ---- ----          ;------------------------
00196 0a7c bc1e          LDP     #TB
00197 0a7d 1902          LACC    TB,9            ; Initialized TB and RB register
00198 0a7e 2203          ADD     RB,2            ;
00199 0a7f b802          ADD     #02h            ;
00200 0a80 7a80          CALL    AIC_2ND         ;
      0a81 0000
00201 ---- ----          ;------------------------
00202 0a82 bc1e          LDP     #AIC_CTR
00203 0a83 1204          LACC    AIC_CTR,2       ; Initialized control register
00204 0a84 b803          ADD     #03h            ;
00205 0a85 7a80          CALL    AIC_2ND         ;
      0a86 0000
00206 0a87 ef00          RET                     ;
00207 ---- ----
00208 ---- ----  AIC_2ND:
00209 0a88 bc00          LDP     #0
00210 0a89 9821          SACH    DXR             ;
00211 0a8a be40          CLRC    INTM
00212 0a8b be22          IDLE
00213 0a8c bf9f          ADD     #6h,15          ; 0000 0000 0000 0011 XXXX XXXX XXXX XXXX b
      0a8d 0006
00214 0a8e 9821          SACH    DXR             ;
00215 0a8f be22          IDLE
00216 0a90 9021          SACL    DXR             ;
00217 0a91 be22          IDLE
00218 0a92 b900          LACL    #0              ;
00219 0a93 9021          SACL    DXR             ; make sure the word got sent
00220 0a94 be22          IDLE
00221 0a95 be41          SETC    INTM
00222 0a96 ef00          RET                     ;
00223 ---- ----       .end
>>>>> LINE:223  .END ENCOUNTERED
>>>>> FINISHED READING ALL FILES
>>>>> ASSEMBLY COMPLETE: ERRORS:0   WARNINGS:0
```

APPENDIX D

MATLAB Code For System Simulation

This MATLAB 6.1 code simulates a baseband equivalent of a digital communication system used for this thesis work. Bandwidth efficient transmission of linearly modulated signal, BPSK, through a time-varying dispersive HF channel, modeled by a three-tap discrete FIR Filter is considered. The time spread and Doppler frequency spread of the channel are considered to be three symbol period and 10 Hz, respectively. 2.4 kbps transmission rate is assumed.

Performance of the system with and without equalizer is studied using this code. It is also used to derive the initial coefficients of the selected linear adaptive channel equalizer, MMSE, with LMS algorithm for coefficient adaptation.

```
%-------------------------------------------------------------------------------------------------------------
clc
clear all;
fd=10;                                          % Doppler frequency

%-------------------------------------------------------------------------------------------------------------
% Channel filter coefficients are derived using a Butterworth filter to simulate the time variation % of the
tap weights. Energy of the channel model is normalized to unity.
%-------------------------------------------------------------------------------------------------------------
e0=1;                                           % Relative energy of the three paths
e1=30/50;
e2=20/50;
nn=sqrt(e0^2+e1^2+e2^2);
e0=e0/nn; e1=e1/nn; e2=e2/nn;
[b,a]=butter(2,2*fd/2400);


for m=1:13;
    for k=1:10;
Fd=1; Fs=5;
data_sampl=2400;
fc01=randn(1,data_sampl);
fc02=randn(1,data_sampl);
fc11=randn(1,data_sampl);
fc12=randn(1,data_sampl);
fc21=randn(1,data_sampl);
fc22=randn(1,data_sampl);

a01=filter(b,a,fc01);                           % Filtering to simulate 10 Hz Doppler spread.
a02=filter(b,a,fc01);
a11=filter(b,a,fc11);
a12=filter(b,a,fc12);
a21=filter(b,a,fc21);
a22=filter(b,a,fc22);
j=sqrt(-1);
```

```
a0=a01+j*a02;
a1=a11+j*a12;
a2=a21+j*a22;
a0=abs(a0);                                   % Rayleigh faded filter coefficients
a1=abs(a1);
a2=abs(a2);

a0=a0/sqrt(mean(a0.*a0));                     % Normalizing the channel energy to unity
a1=a1/sqrt(mean(a1.*a1));
a2=a2/sqrt(mean(a2.*a2));


a0=e0*[[rcosflt(a0,Fd,Fs,'filter',ones(1,Fs))]' zeros(1,2*Fs/Fd)];
a1=e1*[zeros(1,Fs/Fd) [rcosflt(a1,Fd,Fs,'filter',ones(1,Fs))]' zeros(1,Fs/Fd)];
a2=e2*[zeros(1,2*Fs/Fd) [rcosflt(a2,Fd,Fs,'filter',ones(1,Fs))]'];

%---------------------------------------data-----------------------------------------------------------------
% Random binary digits are generated and BPSK modulated, wave shaped by a rectangular pulse
%-------------------------------------------------------------------------------------------------------------
data=randn(1,data_sampl);
data=(data>=0)-(data<0);
data_Tx=rcosflt(data,Fd,Fs,'filter',ones(1,Fs)/sqrt(Fs));
data_Tx0=[[data_Tx]' zeros(1,2*Fs/Fd)];
data_Tx1=[zeros(1,Fs/Fd) [data_Tx]' zeros(1,Fs/Fd)];
data_Tx2=[zeros(1,2*Fs/Fd) [data_Tx]'];

%-------------------------------------ISI into data-----------------------------------------------------------
% channel induced ISI is injected to the data by passing it through channel filter
%-------------------------------------------------------------------------------------------------------------
data_Txf=(data_Tx0.*a0)+(data_Tx1.*a1)+(data_Tx2.*a2);
data_Txff=data_Txf(1:end-2*Fs/Fd);

%-----------------------------------------AWGN-----------------------------------------------------------------
% System performance is studied for AWGN of -20 dB upto 40 dB, with a step increase of 5 dB.
%-------------------------------------------------------------------------------------------------------------
snr(m)=5*(m-5);
data_Rx = awgn(data_Txff,snr(m),'measured');

%----------------------------------------------Receiver section-----------------------------------------------
data_Rxf=rcosflt(data_Rx,Fd,Fs,'filter/Fs',ones(1,Fs)/sqrt(Fs));    % Bandpass filter
data_Rxff=data_Rx(3:5:end);                                         % Center sampling
r=(data_Rxff>=0)-(data_Rxff<0);
dataf=data(500:end); rf=r(500:end);                                  % Receiver performance
[num,ra]=symerr(dataf,rf);                                          % without equalizer
e_rate_ave1(k)=ra;

%---------------------------------------Initialization---------------------------------------------------------
% Initial equalizer coefficients, training length, and adjustment factor
%-------------------------------------------------------------------------------------------------------------
b01=1.5; b11=-0.65; b21=0.25; delta=0.0008;
trnglen=500;
```

```
%-----------------------------------------Training period-----------------------------------------------
for i=1:trnglen;
   data_out(i)=data(i);
end;
ttc=zeros(1,length(data));
data_est=b01*data_Rxff(1);
   ttc(1)=data_est;
err=data(1)-data_est;
b0=b01+2*delta*err*data_Rxff(1);
data_est=b0*data_Rxff(2)+b11*data_Rxff(1);
      ttc(2)=data_est;
err=data(2)-data_est;
b0=b0+2*delta*err*data_Rxff(2);
b1=b11+2*delta*err*data_Rxff(1);
data_est=b0*data_Rxff(3)+b1*data_Rxff(2)+b21*data_Rxff(1);
       ttc(3)=data_est;
err=data(3)-data_est;
b0=b0+2*delta*err*data_Rxff(3);
b1=b1+2*delta*err*data_Rxff(2);
b2=b21+2*delta*err*data_Rxff(1);

for i=1:trnglen-3;
   data_est=b0*data_Rxff(i+3)+b1*data_Rxff(i+2)+b2*data_Rxff(i+1);
         ttc(i+3)=data_est;
   err=data(i+3)-data_est;
   b0=b0+2*delta*err*data_Rxff(i+3);
   b1=b1+2*delta*err*data_Rxff(i+2);
   b2=b2+2*delta*err*data_Rxff(i+1);
end;
%-------------------------------------Decision directed mode-------------------------------------------
tempo=zeros(1,length(data)-trnglen);
for i=1:length(tempo);
   data_est=b0*data_Rxff(i+trnglen)+b1*data_Rxff(i+trnglen-1)+b2*data_Rxff(i+trnglen-2);
         ttc(i+trnglen)=data_est;
   tempo(i)=(data_est>=0)-(data_est<0);
   err=tempo(i)-data_est;
    b0=b0+2*delta*err*data_Rxff(i+trnglen);
    b1=b1+2*delta*err*data_Rxff(i+trnglen-1);
    b2=b2-2*delta*err*data_Rxff(i+trnglen-2);
end;
%-----------------------------------------Error calculation-----------------------------------------
equalized_data=[data_out tempo];
[NEMB,BERR] = symerr(data,equalized_data);            % Receiver performance
e_rate_ave2(k)=BERR;                                  % with equalizer
end;
e_uneq(m)=mean(e_rate_ave1);
e_eq(m)=mean(e_rate_ave2);
end;
```