

INFORMATION TO USERS

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

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

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

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

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

U·M·I

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Order Number 9234080

Finite-length equalization for multi-carrier transmission systems

Chow, Jacky S., Ph.D.

Stanford University, 1992

Copyright ©1992 by Chow, Jacky S. All rights reserved.

U·M·I

300 N. Zeeb Rd.
Ann Arbor, MI 48106

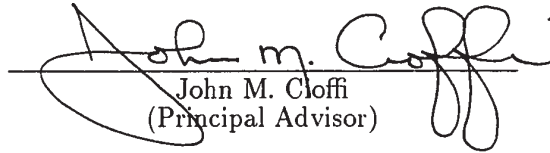
FINITE-LENGTH EQUALIZATION FOR MULTI-CARRIER TRANSMISSION SYSTEMS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

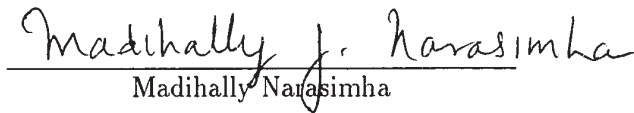
By
Jacky S. Chow
June 1992

© Copyright 1992
by
Jacky S. Chow
All Rights Reserved

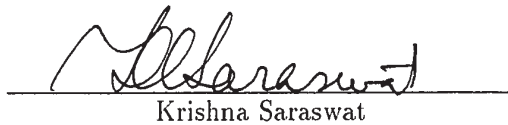
I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.


John M. Cloffi
(Principal Advisor)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.


Madihally Narasimha

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.


Krishna Saraswat

Approved for the University Committee on Graduate Studies:



Abstract

The demand for a higher rate of digital information transfer has led to intensive research in utilizing the full potential of existing communication channels. Baseband and passband modulation techniques have so far been the most commonly used techniques for data transmission. Although theoretically both techniques can achieve the highest possible performance, their performance levels are often well below optimum in practice. Furthermore, the cost of implementing such transceiver systems with near optimum performance can be prohibitive.

The goal of this dissertation is to design a transceiver system with an optimum balance of cost and performance, and is also flexible enough to be applied to a wide range of communication channels. We propose a multi-carrier modulation scheme called discrete multi-tone (DMT) modulation. The basic idea of multi-carrier modulation is to use a set of independent subcarriers in the frequency domain, so that different amounts of information are allocated to different subcarriers according to the channel gain of the subcarriers. Multi-carrier modulation is also known to be optimum theoretically with an infinite number of subcarriers. For channels with severe intersymbol interference, a straightforward implementation of multicarrier systems with near-optimum performance results in excessive cost. The key feature of our DMT system is the inclusion of a finite-length equalizer at the front-end of the receiver, in addition to the cyclic prefix that effectively ensures independence between successive blocks of data. The combination of the finite-length equalizer and the cyclic prefix allows the DMT system to achieve near-optimum performance and at the same time remain computationally and cost effective. To maximize achievable data rate for given cost and computation constraints, we have derived a finite-length

optimization algorithm, based on the minimum mean-square error criterion. The underlying structure of multi-carrier modulation allows the same hardware to apply to a wide range of applications. In particular, by evaluating the feasibility of providing a reliable transmission of full-duplex 1.6 Mbps over copper twisted-pair wires, we found that near theoretical optimum performance can be achieved with reasonable complexity. To address potential problems that may arise in implementing the DMT system in practice, we have developed a real-time prototype of the DMT system using multiple commercially available DSPs. We also wrote Monte-Carlo computer simulations as a cross-check on both our performance evaluation and real-time prototype. Finite-length effects that are not completely characterized by the performance analysis are identified through computer simulations and are accounted for with a more in-depth analysis. Overall, we conclude that our DMT system, with its high performance/cost ratio and great flexibility, is often the best alternative for high-speed digital transmission.

Acknowledgements

I would like to express my deepest appreciation to Professor John Cioffi, who not only serves as my research supervisor, providing valuable and countless resources, insights, and intuitions, but also constantly gives me support, encouragement, and reassurance. I would like to thank Professor Krishna Saraswat, Professor Madihally Narasimha, and Professor Abbas El Gamal for being in my orals and reading committee. I would also like to thank John A.C. Bingham, G David Forney Jr., Russell Hsing, Joseph Lechleider, and Steven Ungar for fruitful discussions.

Numerous fellow graduate students (many graduated before me) at Stanford University have provided me with constant support and companionship. I would like to thank the following in alphabetical order: W. Abbott, N. Al-Dhahir, P. Algoet, J. Aslanis, P. Bednarz, H. Bims, K. Boonyanit, P. Chow, P. Cosman, I. Dobbelaere, G. Dudevoir, K. Fisher, P. Fortier, M. Genossar, M. Goldberg, M. Ho, S. Kasturia, I. Lee, S. Lin, H. Lou, P. Okrah, S. Piche, E. Plummer, A. Ruiz, N. Sands, S. Siu, N. Swenson, J. Tu, P. Voois, S.K. Wilson, R. Ziegler, and T. Zogakis. I would also like to thank J. DeBolt, C. Coe, and M. Parker for their assistance on all the details involved for the past years.

I would like to thank my parents for their support.

I dedicate this dissertation to my wife.

Contents

Abstract	iv
Acknowledgements	vi
1 Introduction	1
1.1 Outline of Dissertation	4
1.2 Contributions	5
2 Background of Multi-Carrier Systems	7
2.1 Basic Digital Communication Systems	7
2.1.1 Baseband (PAM) Systems	7
2.1.2 Passband (QAM) Systems	11
2.2 Basic Structures of Multi-Carrier Systems	15
2.2.1 Discrete Fourier Transform Vectors as Modulating Vectors . .	18
2.2.2 Addition of Cyclic Prefix to the DFT/IDFT Multi-Carrier System	21
2.2.3 Optimality of Multi-Carrier Systems	23
2.3 Summary	24
3 The Discrete Multi-Tone Transceiver	25
3.1 Maximum Likelihood Receivers and Multi-Carrier Systems	25
3.2 Optimality and Equivalence of Infinite-Length Equalizers	28
3.2.1 Minimum Mean-Square Error Equalization	29
3.2.2 Solutions	32
3.2.3 Comparison of MMSE-DFE and ARMA	37

3.3	Design Rule for Finite-Length Equalizers	38
3.3.1	Finite-Length Terminology	38
3.3.2	An Algorithm to Find the Best Finite-Length Equalizer for Multi-Carrier System	40
3.4	The Discrete Multi-Tone (DMT) Transceiver	41
3.4.1	System Description	41
3.4.2	Performance Analysis for the Discrete Multi-Tone System . . .	46
3.4.3	Channel Identification and Initialization Protocol	48
3.4.4	Transmit Power Level	50
3.5	Computational Requirements	51
3.5.1	Transmitter Complexity	52
3.5.2	Receiver Complexity	52
3.6	Summary	55
4	Applications of the DMT System	56
4.1	Subscriber Loop Characterization and Modeling	57
4.1.1	CSA Loop Characteristics	58
4.1.2	Channel Modeling	59
4.1.3	Capacity	61
4.2	DMT System for HDSL	63
4.2.1	The DMT HDSL Transmitter	63
4.2.2	The DMT HDSL Receiver	64
4.3	Derivation of Performance Evaluation	65
4.4	Performance Results for Multi-Carrier Systems	67
4.4.1	Coding	67
4.4.2	Average Input Power	70
4.4.3	Block Length	73
4.4.4	Sampling Rate	73
4.4.5	Noise Margins	76
4.4.6	HDSL Occupancy	78
4.4.7	Impulse Noise	78

4.4.8	Effect of Loop Length	80
4.4.9	Discussion	81
4.5	Summary	82
5	Performance of the Finite-Length DMT System	84
5.1	Finite-Length Effects	84
5.2	Theoretical Analysis of the Finite-Length DMT System	87
5.2.1	Analysis of Finite-Length DMT System	87
5.2.2	Matrix Representation of the Finite-Length DMT System	90
5.3	Monte-Carlo Computer Simulations for the DMT system	97
5.3.1	Channel and Noise Characterization	98
5.3.2	Finite-Length Algorithm for Computing Taps Settings	99
5.3.3	Signal-to-Noise Ratio Measurement	99
5.3.4	Steady-State Performance	100
5.4	Computer Simulations Results	101
5.4.1	Effects of Length of Feedforward Equalizer and Cyclic Prefix	101
5.4.2	Comparisons of Ideal Performance Analysis and Computer Simulation Results	104
5.5	Summary	111
6	A Real-Time DMT Prototype	112
6.1	Specification for the DMT Prototype	112
6.1.1	DMT Hardware Requirement	113
6.1.2	DMT Software Requirement	114
6.2	Hardware for the DMT Prototype	114
6.2.1	Motorola DSP 56001	116
6.2.2	Interface Board (FIF0)	116
6.3	Software for the DMT Prototype	117
6.3.1	Encoder	119
6.3.2	Inverse FFT and FFT	119
6.3.3	Cyclic Prefix	124
6.3.4	Feedforward Equalizer	124

6.3.5	One-Tap LMS Filters and Decoder	124
6.4	Testing of the DMT Prototype	124
6.5	Summary	126
7	Conclusions	127
7.1	Summary of Results	128
7.2	Continuing Research Topics	130
A	Unequal SNR for Real and Imaginary Parts	133
B	Average Energy with Cyclic Prefix	137
C	Listings of Matlab Programs	141
C.1	Matlab program <i>nextfilter.m</i>	141
C.2	Matlab program <i>nextts.m</i>	142
C.3	Matlab program <i>ftot.m</i>	143
C.4	Matlab program <i>noiseauto.m</i>	144
C.5	Matlab program <i>computetaps.m</i>	144
C.6	Matlab program <i>deltap.m</i> and <i>deltan.m</i>	146
C.6.1	Matlab program <i>deltap.m</i>	146
C.6.2	Matlab program <i>deltan.m</i>	147
C.7	Matlab program <i>findsnr.m</i>	148
C.8	Matlab program <i>gen4pam.m</i>	155
C.9	Matlab program <i>bitassign.m</i>	155
C.10	Matlab program <i>datapump.m</i>	156
	Bibliography	164

List of Tables

3.1	Algorithm to find the best finite-length equalizer for DMT system . .	41
3.2	Transmitter complexity at 640 kHz sampling rate	52
3.3	(Old) Receiver complexity with full block LMS	53
3.4	(Old) Receiver complexity with efficient updating block LMS	54
3.5	(Revised) Receiver complexity at 640 kHz	54
4.1	Capacity and flat-input capacity for CSA loops, 800 kHz rate and with 6 dB margin (in Mbps)	62
5.1	Achievable data rate for the DMT system with $M = 11, \nu = 7, N = 512$	102
5.2	Achievable data rate for the DMT system with $M = 7, \nu = 4, N = 512$	102
5.3	Achievable data rate for the DMT system with $M = 3, \nu = 4, N = 512$	103
5.4	Achievable data rate for the DMT system with $M = 3, \nu = 2, N = 512$	103
5.5	Achievable data rate for the DMT system with $M = 3, \nu = 4, N = 128$	103
5.6	Achievable data rate for multi-carrier systems with no feedforward equalizer	104
5.7	Comparisons of ideal performance analysis and computer simulations ($N=128$)	106
5.8	Comparisons of ideal performance analysis and computer simulations ($N=512$)	106

List of Figures

1.1 Attenuation, delay, and distortion	2
2.1 A basic digital communication system	8
2.2 Mapping between bits and input symbols (2PAM)	9
2.3 Mapping between bits and input symbols (4PAM)	9
2.4 Mapping between bits and input symbols (8PAM)	10
2.5 Mapping between bits and input symbols: Gray code (8PAM)	10
2.6 16 QAM Constellation	12
2.7 A basic structure of multi-carrier systems	16
2.8 A basic discrete-time multi-carrier system	17
2.9 Visualization of multi-carrier systems	18
2.10 A basic multi-carrier transmitter	20
2.11 A basic multi-carrier receiver	20
2.12 Cyclic prefix	22
3.1 Input-assisted equalization	27
3.2 Some choices for the input-aided ML equalizer.	34
3.3 The DMT transmitter	42
3.4 The DMT receiver (old)	43
3.5 The DMT receiver (revised)	45
3.6 A Simplified DMT Transceiver System	47
4.1 CSA Channels : length (in feet) / gauge	58
4.2 Impulse response of a CSA loop	61
4.3 Block diagram of a specific DMT HDSL transmitter	63

4.4	Uncoded data rate for CSA channels (6 dB margin included)	68
4.5	Effect of coding and average input power on data rate (6 dB margin included)	69
4.6	Bit distribution of Channel H with 5.5 dB code	71
4.7	Bit distribution of Channel A with 5.5 dB code	72
4.8	Effect of different blocklength on data rate (6 dB margin included) . .	74
4.9	Effect of sampling rate on data rate (6 dB margin included)	75
4.10	Effect of noise margins on data rate	77
4.11	Effect of HDSL occupancy on data rate (6 dB margin included) . . .	79
4.12	Effect of loop length on data rate (6 dB margin included)	80
5.1	Smooth channel transfer function with small block length	85
5.2	Channel with frequency notches	86
5.3	Block diagram to analyse finite-length DMT systems	87
5.4	Achievable data rate for multi-carrier systems with no feedforward equalizer	105
5.5	Transfer function of the 9-kft, 26-gauge wire	107
5.6	Transfer function of the 9-kft, 26-gauge wire (lowest 4 kHz)	108
5.7	SNR for the DMT system with $M = 11$, $\nu = 7$, and $N = 512$	109
5.8	SNR for the multi-carrier system with no feedforward equalizer, $\nu = 7$, and $N = 512$	110
6.1	DMT prototype hardware configuration	115
6.2	DMT transmitter prototype software functions	117
6.3	DMT receiver prototype software functions	118
6.4	DMT prototype test set-up	125

Chapter 1

Introduction

There has been a tremendous increase in demand for reliable, high-speed digital communication as modern technology has transcended the transmission and processing of discrete data. This trend is evident in the popularity of computers, compact discs, facsimile machines, modems, cellular phones, to name a few, and the anticipated arrival of multi-media video and audio communications, teleconferencing, digital audio and video tapes and discs, high-definition television, and so on. Transmission media exist in various forms, such as copper twisted-pair wires, coaxial cables, optical fibers, satellites, and microwave radio channels. All of these media have physical limitations, and the transmitted data typically suffers from degradations such as attenuation, delay, and distortion, as shown in Figure 1.1. Attenuation reduces the magnitude of the transmitted signal, delay results in a time shift of the transmitted data, and distortion of the channel changes the shape of the transmitted signal. When a steady stream of signals is transmitted, the distortion results in interference between neighboring transmitted data. In other words, the current received sample is an attenuated, delayed version of the current transmitted sample plus a corrupted version of the previous and future transmitted samples. This corruption of a received sampled data by neighboring signals is called intersymbol interference (ISI), which is treated as a noise impairment in most simple data transmission systems, and occurs in virtually all physical channels.

Different transmission media have different types of noise. For a representative

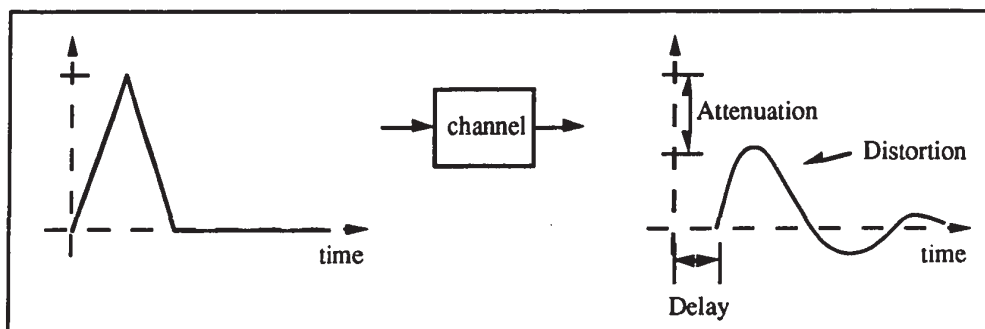


Figure 1.1: Attenuation, delay, and distortion

application in digital communications, we will focus on copper twisted-pair wires. The main impairments include crosstalk noise from neighboring copper wires grouped in the same bundle; thermal, electronic, and quantization noises originating from physical devices that transmit and receive data such as analog-to-digital converter (ADC) and digital-to-analog converter (DAC); and impulse noise caused by lightning, wind, switching transients, power surge, and other natural or man-made sources.

Baseband (pulse amplitude modulation, PAM) and passband (quadrature amplitude modulation, QAM) modulation techniques are the two most commonly used modulation schemes for data communications. Their popularity stems from the fact that they are relatively easy to understand and implement, and the cost associated with such a system can be very low, especially if limited equalization is performed at the receiver. However, these low-cost, straightforward implementations often achieve only a small fraction of the potential capacity of transmission media such as copper twisted-pair wires and coaxial cables. Equalization techniques have thus been investigated to improve the throughput or to reduce the error rate. Equalization is a method for recovering the signal energy in the presence of intersymbol interference. The decision feedback equalization (DFE) method in conjunction with baseband or passband systems is known to be optimum when infinite length equalizers are allowed.

With finite length equalizers, the number of feedforward and feedback taps of a DFE system can be fairly high, especially if the channel of interest has severe intersymbol interference (ISI) and/or severe noise impairments. The resulting transceiver system can be computationally intensive and costly to implement. More importantly, in order to optimize performance for either a baseband or a passband system, the sampling rate needs to be adjusted for different channel characteristics, which either requires installing different hardware for different channels according to their optimum sampling rates, or designing the hardware with a variable sampling rate device. The former system is clearly unacceptable and the latter system can be difficult to design and implement.

The goal of this dissertation is to design a transceiver system that can achieve near optimum performance and can be implemented commercially. This means that the system must have reasonable complexity and delay, so that the cost of implementing a single unit of the transceiver is acceptable. However, a successful commercial product also requires that a fixed hardware be used for a reasonable range of channels given a specific application, and thus the transceiver system must be fairly flexible and adaptive so that excellent performance can be guaranteed over the channels of interest. Of course, it is impossible and extremely inefficient to use a fixed hardware for all possible applications, considering the diversity of transmission media and applications such as the extremely high data-rate-transfer capability of a short fiber-optic line and the extremely low data-rate-transfer capability of deep-space communications over millions of miles apart. However, when modulation method is applied to channels within reasonably defined limits, a fixed hardware design can be crucial for commercial success.

As an illustrative example, consider the high-speed digital subscriber line (HDSL) service that intends to provide bi-directional high-speed data transmission over a set of digital subscriber lines. This set of lines are copper twisted-pair wires connecting central offices and local customer premises, and the line characteristics are similar but different since the distances and the gauges of the various wires are different. A fixed hardware allows the system to be installed to each of these subscriber loops directly. However, if different hardware is required for different loops, then each subscriber

line needed to be tested and examined in order to install the correct version of the hardware. The cost of man-power associated with such testing is simply unacceptable.

We propose a multi-carrier modulation scheme called discrete multi-tone (DMT) modulation transceiver in this dissertation as a solution to the aforementioned problems. The DMT transceiver can achieve near-optimum performance with the flexibility and the adaptivity of the system so that given an application over a set of channels, the same hardware can be used.

1.1 Outline of Dissertation

We briefly define the problems of achieving near capacity data rate for digital communication and propose a transceiver system in this Chapter. In Chapter 2, basic baseband and passband digital communication systems are first described as a background to understand multi-carrier modulation systems. The basic concept of multi-carrier modulation is then described. Two previously proposed multi-carrier systems are examined briefly: the “vector coding” method in [1] and the discrete Fourier transform based system in [2].

Chapter 3 presents the proposed discrete multi-tone (DMT) modulation system. The optimality of this DMT system is proved in Section 3.2 using input-assisted equalization theory for infinite complexity, and a finite-length optimization algorithm is presented in Section 3.3 to maximize achievable data rate with any given complexity constraint. The structure of the DMT system is then described in detail, including the derivation of performance analysis, system initialization, and transmit power level. The complexity of implementing the DMT system is also estimated, based on instruction counts using standard digital signal processors.

Chapter 4 shows the performance of the DMT system applied to a specific application for purposes of illustration. The application on which we focus is the high-speed digital subscriber line (HDSL) service that will provide reliable bi-directional transmission of 1.6 Mbps over one or two copper twisted-pair wires. We also briefly describe the characteristics and impairments of the subscriber loops. The full potential or capacity of these wires are also presented to illustrate how close the DMT system is

to the optimum performance. An idealized performance analysis is used to arrive at the results. Effects of coding, average input power, block length, sampling rate, noise margins, and HDSL occupancy are examined.

Chapter 5 identifies the finite-length effects that have not been taken into account in the performance results in Chapter 4. Two approaches are presented to model the finite-length effects and to derive the performance level more accurately. However, neither approach completely characterizes the finite-length DMT system. A set of Monte-Carlo computer simulation programs are written to account for all effects. The difference between the ideal theoretical performance and the results obtained from computer simulations are compared. We found that in general the performance degradation caused by finite-length effects are minor, and can be absorbed into performance margin used with ideal analysis.

Chapter 6 describes a real-time prototype of the DMT system, implemented with multiple Motorola 56001 DSPs. The prototype serves to verify proof-of-principle. The design of hardware and software are described.

Chapter 7 summarizes the main results of this dissertation. Future research areas are suggested based on the results and observation drawn from this study.

1.2 Contributions

The focus on multi-carrier modulation techniques have been a constant effort in Prof. Cioffi's "coders research group". As such, some of the ideas in this dissertation appear in several other student's work and vice versa. Chapter 2 presents basic communication systems as a background to understand multi-carrier modulation techniques. Previous multi-carrier modulation systems by former "coders" are described. In particular, the system proposed by [2] serves as a backbone of the multi-carrier transceiver proposed in this dissertation. Chapter 3 describes the discrete multi-tone (DMT) modulation system, which is an enhanced version of the multi-carrier system proposed by [2]. A new technique is presented by including a time-domain feedforward equalizer at the receiver front-end, first suggested by fellow "coder" Glen Dudevoir. The analysis showing that the DMT system is optimum with infinite complexity in

Section 3.2 is new. The finite-length optimization algorithm in Section 3.3, the DMT receiver structures in Section 3.4, and the estimation of the complexity of the DMT system in Section 3.5 were developed by the author. The performance analysis is derived by Prof. Cioffi, and has been applied to analyze performance of the DMT system for different applications by the “coders group” in general. The author first observed that the signal-to-noise ratios are unequal between the real and the imaginary part of a complex subchannel, and subsequently proved the phenomenon, as well as the effect of the cyclic prefix on the average input power, in Appendix B. Chapter 4 evaluates the feasibility of applying the DMT system to the high-speed digital subscriber line (HDSL) service. The set of subscriber loops are first studied by Duduvoir and Jim Aslanis. Aslanis also derived the capacity of these loops in his dissertation. The performance results in Section 4.4 are new and resulted from a joint effort by Jerry Tu and the author. Chapter 5 is entirely new and is developed by the author. The finite-length effects are identified, and two approaches are derived to model the finite-length DMT system accurately. To encompass all possible effects, a set of Monte-Carlo computer simulation programs have been written over the years by the author to identify potential problems and project accurate results. Difference between ideal analysis and computer simulation results are compared. Chapter 6 describes a prototype of the DMT system, and is a joint project between Prof. Cioffi, Cory Modlin, Tu, and the author, as well as help from other consultants on hardware fabrication and debugging.

Chapter 2

Background of Multi-Carrier Systems

In this Chapter, we first describe basic digital communication systems as a background to understand multi-carrier systems. Two popular digital transmission schemes are described: baseband (pulse amplitude modulation, PAM) and passband (quadrature amplitude modulation, QAM). Basic idea and structure of multi-carrier systems are then presented. The multi-carrier system we focus in this Chapter is the backbone of our proposed DMT system in Chapter 3.

2.1 Basic Digital Communication Systems

To facilitate the understanding of multi-carrier systems, baseband and passband communication systems are presented. In particular, since a multi-carrier system can be visualized as the combination of a group of QAM channels, the performance of QAM systems will be derived.

2.1.1 Baseband (PAM) Systems

A basic digital communication system is shown in Figure 2.1. $\{x_k\}$ is a sequence of input symbols that are intended to be transferred through a communication channel.

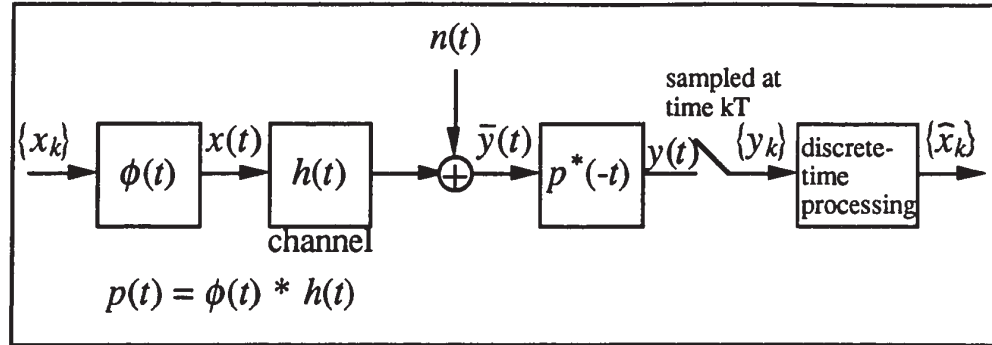


Figure 2.1: A basic digital communication system

The spacing between the symbols to be transmitted is the symbol period, T (in seconds). Depending on the amount of information, the time index k could go from a distant past to a remote future ($-\infty$ to ∞). Each input symbol x_k is originated from some binary data through a one-to-one mapping. The simplest mapping is shown in Figure 2.2 by mapping $0 \rightarrow -d$ and $1 \rightarrow +d$ (or vice versa). For many applications that require transferring a substantial amount of information in a relatively short period of time, several binary data or bits are mapped into a single input symbol. Some typical mappings are shown in Figure 2.3 and Figure 2.4, where we map the binary sequence in increasing symbol levels for illustration purposes. These mappings are often called “coding” since the receiver is required to know the mapping in order to retrieve the original binary sequence (a process called “decoding”). For lossless, noiseless channels, any one-to-one mapping can be decoded perfectly at the receiver. However, all physical channels suffer from varying degrees of imperfection, and the estimated input symbol $\hat{x}_k$ at the output of the receiver does not always correspond to the actual input symbol. When $\hat{x}_k \neq x_k$, a symbol error is made. A good design of digital communication system would try to minimize the occurrence of any symbol error, and would try to limit the estimated input symbol to be as close to the actual input symbol as possible. As an example, we assume Figure 2.4 is used. Suppose $x_k = d$, and a symbol error occurs, then $\hat{x}_k$ is most likely to be $-d$ or $3d$ since these are the closest symbol levels to x_k . If $\hat{x}_k = -d$, then we decode it to 011 instead of

100, and three bit errors result with a single symbol error. To reduce the bit error rate, other mappings can be used. Figure 2.5 shows a mapping called Gray code, where adjacent symbol error always results in only one bit error.

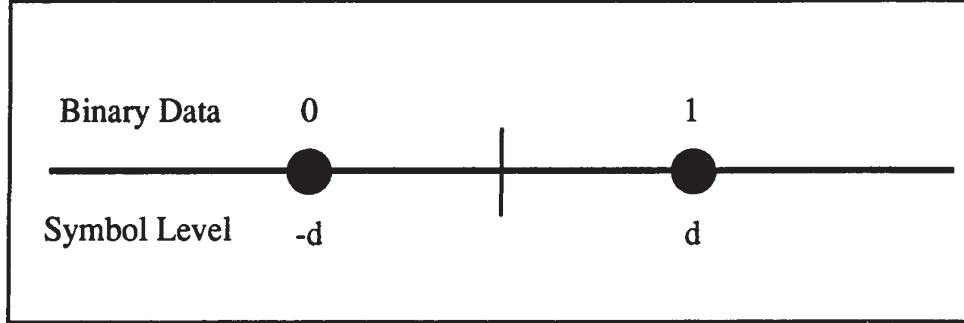


Figure 2.2: Mapping between bits and input symbols (2PAM)

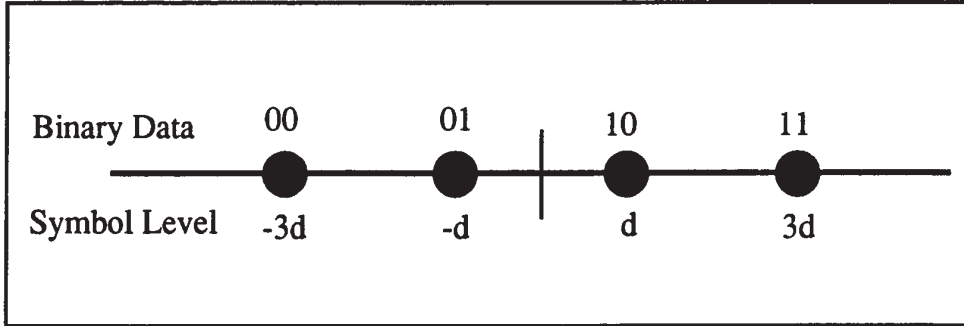


Figure 2.3: Mapping between bits and input symbols (4PAM)

Figures 2.2 to 2.5 belong to a class of signal constellations known as pulse amplitude modulation, PAM, since the input symbol represents the amplitude of the modulating pulse. As shown in Figure 2.1, $\phi(t)$ is the transmit filter used to modulate the input symbols, and $h(t)$ is the impulse response of the communication channel. The PAM transmit filter $\phi(t)$ is normally defined by

$$\phi(t) = \frac{1}{T} \text{sinc}\left(\frac{t}{T}\right), \quad (2.1)$$

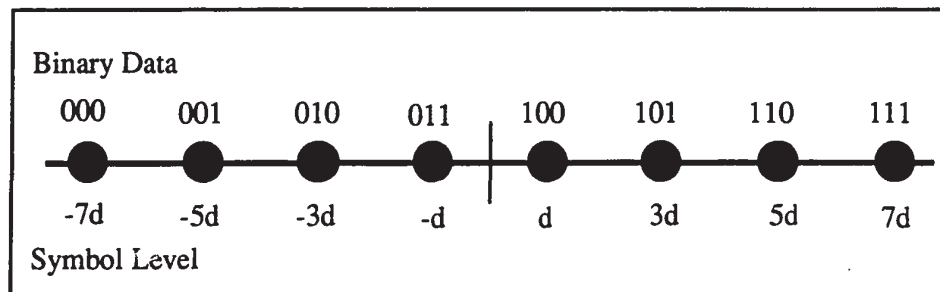


Figure 2.4: Mapping between bits and input symbols (8PAM)

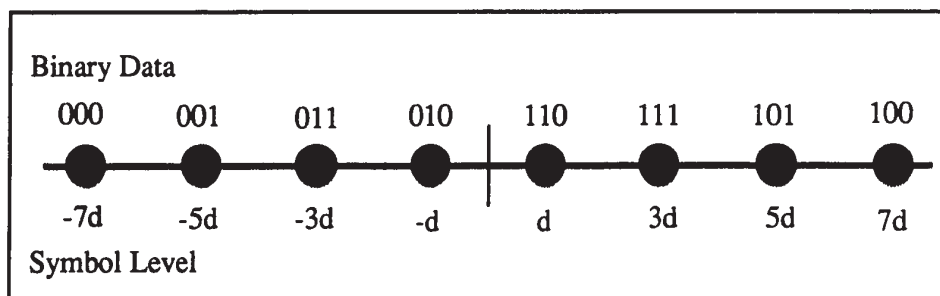


Figure 2.5: Mapping between bits and input symbols: Gray code (8PAM)

where

$$\text{sinc}(x) \triangleq \frac{\sin(\pi x)}{\pi x}. \quad (2.2)$$

We define the channel pulse response, $p(t)$, to be the convolution of the transmit filter and the channel impulse response, $p(t) = \phi(t) * h(t)$. The channel output $\tilde{y}(t)$ is given by

$$\tilde{y}(t) = \sum_{k=-\infty}^{+\infty} x_k p(t - kT) + n(t), \quad (2.3)$$

where $n(t)$ is an additive white Gaussian noise.

Without loss of optimality, the receiver may consist of a matched filter, $p^*(-t)$, followed by a symbol-rate sampler [3]. The block labeled discrete-time processing in Figure 2.1 may simply be a decision circuit for the estimated input symbol, or an elaborate equalizer such as a decision feedback equalizer (DFE) to improve overall performance.

2.1.2 Passband (QAM) Systems

Instead of using a single transmit filter, $\phi(t)$, a passband (or quadrature amplitude modulation, QAM) uses two transmit filters, and are usually

$$\phi_1(t) = \sqrt{\frac{2}{T}} \text{sinc}\left(\frac{t}{T}\right) \sin\left(\frac{k\pi t}{T}\right), \quad k = \text{integer}, \quad (2.4)$$

$$\phi_2(t) = \sqrt{\frac{2}{T}} \text{sinc}\left(\frac{t}{T}\right) \cos\left(\frac{k\pi t}{T}\right), \quad k = \text{integer}, \quad (2.5)$$

called the in-phase and quadrature components.

We can think of a QAM system as a two-dimensional system, and the signal constellation is on a two-dimensional plane. As a background to understand multi-carrier systems, we only focus on square QAM constellations in this chapter. A square QAM constellation contains some power-of-4 points, arranged regularly within a square boundary, in two dimensions. Figure 2.6 shows a 16-point QAM constellation where each axis represents a one-dimension PAM constellation. This figure also shows the 4-QAM constellation contained within the 16-QAM constellation. The smallest

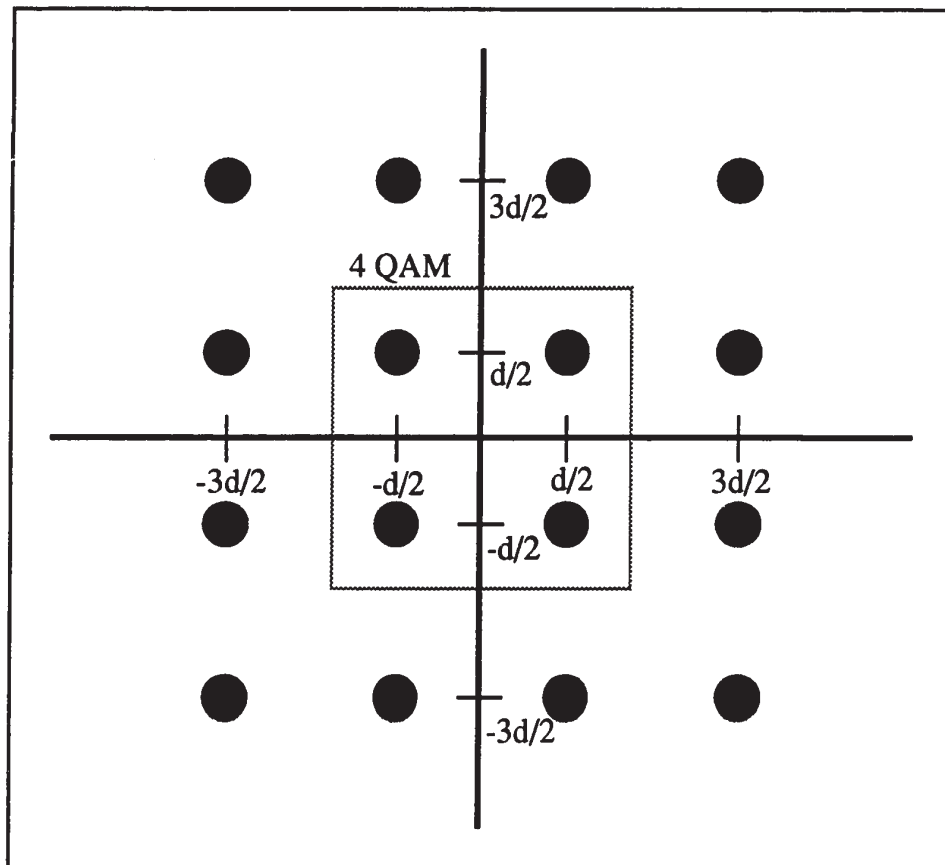


Figure 2.6: 16 QAM Constellation

distance between points is given by d , and the constellation is centered at the origin and has zero mean. In many cases, all points in a QAM constellation are equally likely to be transmitted, and the average input energy is then equal to

$$\mathcal{E} = \frac{(L-1)d^2}{6}, \quad (2.6)$$

where L is the points in the QAM constellation and is a power of 4 for square constellations.

To analyse the performance of a QAM transmission system, we assume that the channel is ISI-free and has a power gain of $|H|^2$. The average input energy of the system is $\mathcal{E}$, and the noise variance per dimension is equal to σ^2 . The channel output signal-to-noise ratio (SNR) is given by

$$SNR = \frac{\mathcal{E}|H|^2}{2\sigma^2}. \quad (2.7)$$

The distance between the points in the input signal constellation is equal to d , so that the minimum distance between the QAM points at the output of the channel, d_{min} , is given by

$$d_{min}^2 = |H|^2 d^2. \quad (2.8)$$

Rewriting equation (2.6) with equation (2.8),

$$L = 1 + \frac{6\mathcal{E}|H|^2}{d_{min}^2}. \quad (2.9)$$

The probability of symbol error, P_e , in QAM systems is closely approximated by [4], [5]:

$$P_e \leq 4Q \left[\frac{d_{min}}{2\sigma} \right], \quad (2.10)$$

where the Q-function is defined by

$$Q(x) \triangleq \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-u^2/2} du. \quad (2.11)$$

Since $Q(-x) = 1 - Q(x)$, we only need to compute values for positive x . However, this integral cannot be evaluated in closed form, and [6] includes a graph to read the values of the Q-function for a certain range of x values. For $x > 3$, $Q(x)$ can be approximated by [6]

$$Q(x) \approx \frac{1}{\sqrt{2\pi}x} e^{-x^2/2}. \quad (2.12)$$

We define a convenient quantity called the SNR gap, Γ , by

$$\Gamma = \frac{d_{min}^2}{12\sigma^2}, \quad (2.13)$$

so that

$$d_{min}^2 = 12\Gamma\sigma^2. \quad (2.14)$$

Substituting equations (2.7) and (2.14) into equation (2.9),

$$L = 1 + \frac{SNR}{\Gamma}, \quad (2.15)$$

The number of bits that can be transmitted, b , is

$$b = \log_2 L = \log_2 \left(1 + \frac{SNR}{\Gamma} \right), \quad (2.16)$$

and the data rate, R , is

$$R = \frac{b}{T}. \quad (2.17)$$

The capacity or the number of bits that can be reliably transmitted through a channel is derived in the famous Shannon's paper [7], and the capacity theorem is given by

$$C = \log_2(1 + SNR) \text{ (Bits/2 Dimensions)} \quad (2.18)$$

The quantity Γ is called the SNR gap since it is a measure of loss with respect to theoretically optimum performance (capacity). As Γ approaches 1 (0 dB), the achievable data rate of the QAM system approaches capacity.

The SNR gap Γ consists of three components: the inverse bit error rate, $[Q^{-1}(P_b)]^2$, the margin γ_m that is the amount of extra performance required to ensure adequate performance, and the coding gain γ_c of any applied code, including shaping gain/loss, nearest-neighbor penalties, and correlated noise adjustments.

$$\Gamma = \frac{\gamma_m [Q^{-1}(P_b)]^2}{3\gamma_c}, \quad (2.19)$$

or in dB (taking $10 \log_{10}$ of the above expression),

$$\Gamma \text{ (dB)} = [Q^{-1}(P_b)]^2 + \gamma_m - \gamma_c - 4.77 \text{ (dB)}. \quad (2.20)$$

In practice, a target bit error rate is set as a performance requirement of a digital communication system, and that determines the inverse bit error rate.

2.2 Basic Structures of Multi-Carrier Systems

Multi-carrier modulation methods use an optimized frequency-division allocation of energy and bits to maximize the reliably achievable data rates that can be transmitted over bandlimited communication channels. These systems are very easily described because they use frequency-division multiplexing to transport a single input data stream on several carriers within the usable frequency band of the channel. The simplicity of multitone methods accounts for their use in some of the earliest data transmission modems, such as the Collin's Kineplex modems of 1957 [8]. While the simplicity and appeal of multitone modulation has existed for decades, it is only very recently that their optimal performance has been proven theoretically (see [9], [2]) and demonstrated in practice with recent products, such as Telebit's Trailblazer modems [10], and NEC's multitone voiceband and groupband modems [11]. These highest-performance products are sold at only a fraction of the cost of comparable performance single-channel or QAM modems.

The basic structure of multi-carrier systems is shown in Figure 2.7. Instead of using a single modulating filter as in Section 2.1, we use a set of N modulating filters called basis functions, and are usually chosen to be orthonormal

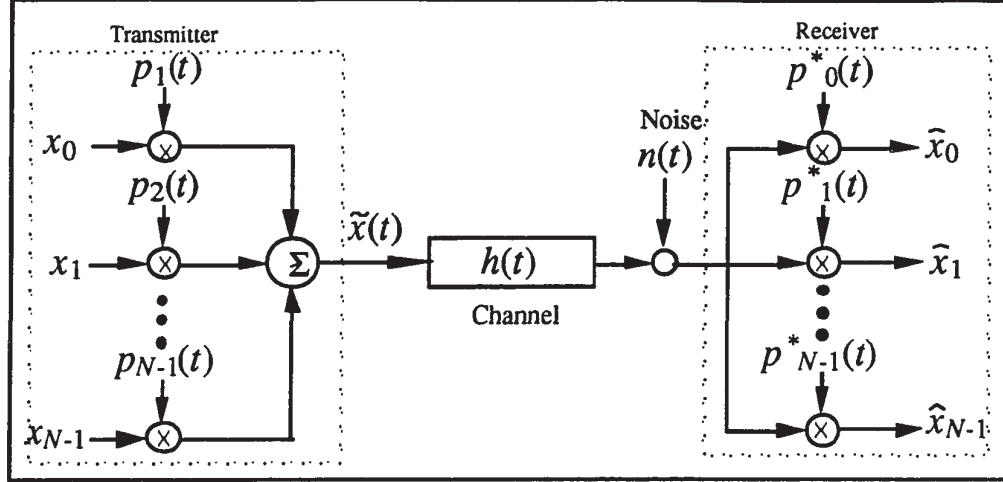


Figure 2.7: A basic structure of multi-carrier systems

$$\int_{-\infty}^{+\infty} p_i(t) p_j^*(t) dt = \delta_{ij} \quad (2.21)$$

$$\triangleq \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}, \quad (2.22)$$

where $*$ denotes complex conjugate.

At the receiver, similar to Section 2.1, a set of matched filters $p_i^*(-t)$, $i = 0, \dots, N-1$ is used. We assume for all following analysis that the system is sampled at a rate that is at least twice as high as any signal frequency, so that sampling theory arguments apply and the discrete-time system, as shown in Figure 2.8, is equivalent to the actual continuous-time system. A block of input bits is grouped into N subsymbols, x_k , $k = 0, \dots, N-1$, that are independently modulated by N -dimensional sampled-sinusoid modulating vectors, $\mathbf{p}_k$, and then summed to form an N -dimensional block of modulated transmit signal samples, $\tilde{x}_k$, $k = 0, \dots, N-1$. As before, the modulating vectors, $\mathbf{p}_k$, are orthonormal

$$\mathbf{p}_i \odot \mathbf{p}_j^* = \delta_{ij}, \quad (2.23)$$

where $\odot$ means the dot product of the two vectors. The corresponding demodulating vectors at the receiver are $\mathbf{p}_k^*$.

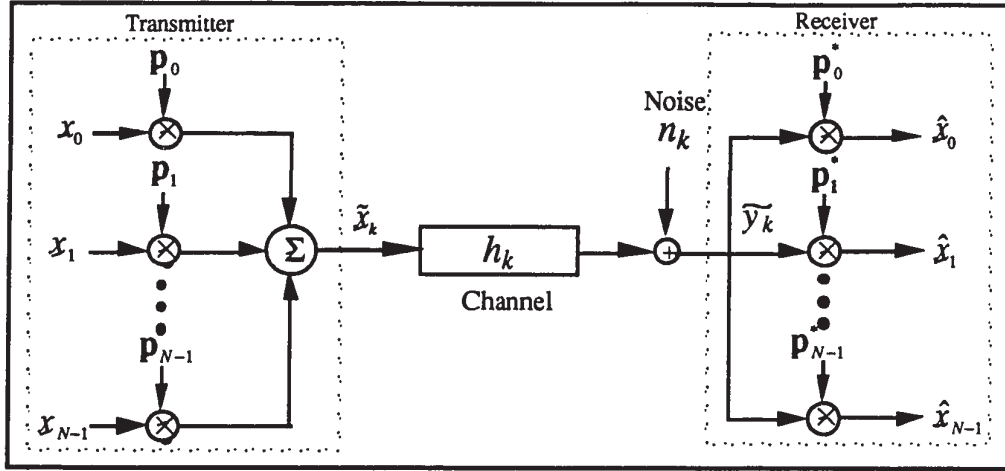


Figure 2.8: A basic discrete-time multi-carrier system

To visualize the basic idea of how multi-carrier systems perform, we consider a hypothetical channel transfer function as shown in Figure 2.9. A multi-carrier modulation system is based on a frequency division multiplexing scheme that maximizes performance by dividing the channel into a set of parallel, independent subchannels in the frequency domain, so that more bits (or information) can be assigned to the better frequency subchannels. The subchannels that have poor response, as shown in Figure 2.9 near D.C. and at high frequencies, will be allocated little or no information. As we will see later, this ability of assigning different amounts of information across different subchannels is precisely the reason why multi-carrier systems can achieve optimum performance.

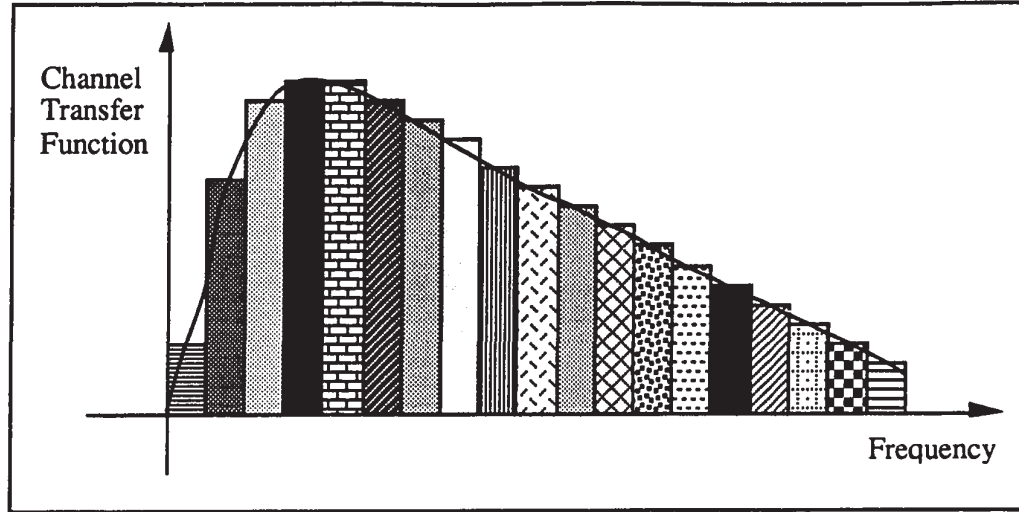


Figure 2.9: Visualization of multi-carrier systems

2.2.1 Discrete Fourier Transform Vectors as Modulating Vectors

Several sets of orthonormal vectors have been proposed as the modulating and demodulating vectors for multi-carrier systems. [12], [2], and [9] suggest using inverse discrete Fourier transform vectors as the modulating vectors and discrete Fourier transform vectors as the demodulating vectors. [1] proposes using eigenvectors of the channel autocorrelation matrix as the modulating/demodulating vectors. Vectors based on the Hartley transform and the Cosine transform, among others, are proposed in [13]. Other techniques can be found, for example, in [10] and [14].

In practice, two important factors affect the choice of the modulating vectors significantly. The first issue is the complexity required to perform the modulation and is likely to be the limiting factor as to how long the block length N can be. The second issue is the adaptivity of the modulating vectors: whether different sets of orthonormal vectors are required for different channel characteristics. Eigenvectors of the channel autocorrelation matrix as modulating vectors have been proved to be optimum [1]. However, eigenvectors are not attractive when viewed in terms of

both the abovementioned factors. The computation required to modulate a block of N symbols is proportional to N^2 , and the eigenvectors are channel dependent. On the other hand, discrete Fourier transform vectors satisfy both practical issues, as explained shortly.

The multi-carrier system on which we focus uses the Inverse Discrete Fourier Transform (IDFT) vectors as the modulating vectors, and so $\mathbf{p}_k$, the k^{th} modulating vector, in Figure 2.8 is given by

$$\mathbf{p}_k = [p_{k,0} \ p_{k,1} \ \dots \ p_{k,N-1}] \quad , \quad (2.24)$$

where

$$p_{k,n} \triangleq \frac{1}{\sqrt{N}} e^{+j \frac{2\pi kn}{N}} \quad k, n = 0, \dots, N-1 \quad , \quad (2.25)$$

and the demodulating vectors, $\mathbf{p}_k^*$, is

$$\mathbf{p}_k^* = [p_{k,0}^* \ p_{k,1}^* \ \dots \ p_{k,N-1}^*] \quad , \quad (2.26)$$

where

$$p_{k,n}^* \triangleq \frac{1}{\sqrt{N}} e^{-j \frac{2\pi kn}{N}} \quad k, n = 0, \dots, N-1 \quad , \quad (2.27)$$

the discrete Fourier transform (DFT) vectors. Note that we have used a normalized version of the N point DFT/IDFT as follows

$$\text{IDFT: } \tilde{x}_m = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_k e^{+j \frac{2\pi km}{N}} \quad m = 0, \dots, N-1 \quad , \quad (2.28)$$

$$\text{DFT: } x_k = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} \tilde{x}_m e^{-j \frac{2\pi km}{N}} \quad k = 0, \dots, N-1 \quad , \quad (2.29)$$

Unlike eigenvectors of the channel autocorrelation matrix, the DFT/IDFT modulating/demodulating vectors are fixed for all channel characteristics. Furthermore, the DFT/IDFT vectors can be implemented efficiently with well-known fast Fourier transform (FFT) algorithms and inverse fast Fourier transform (IFFT) algorithms. For DFT/IDFT vectors implemented with FFT/IFFT algorithms, the number of instructions required to modulate a block of N symbols is proportional to $N \log_2 N$ instead of N^2 , and the difference can be significant when the block length N is large.

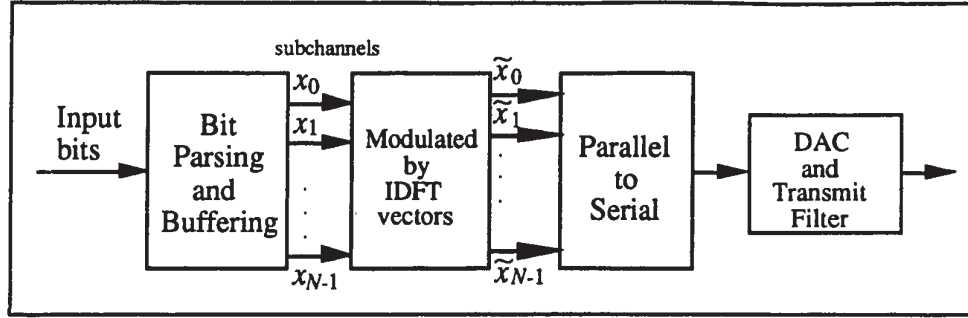


Figure 2.10: A basic multi-carrier transmitter

The block diagram of a basic multi-carrier transmitter is shown in Figure 2.10. The bit parsing block groups the raw binary bits according to an appropriate bit allocation and parses them to the corresponding subchannels. Each block of input symbols, $\{x_k\}$, $k = 0, \dots, N - 1$, is then modulated by the IDFT vectors, which is implemented by taking the inverse FFT of the block. The modulated block of data is then converted to a serial stream of symbols before transmitting to the channel through the digital-to-analog converter (DAC) and the transmit filter.

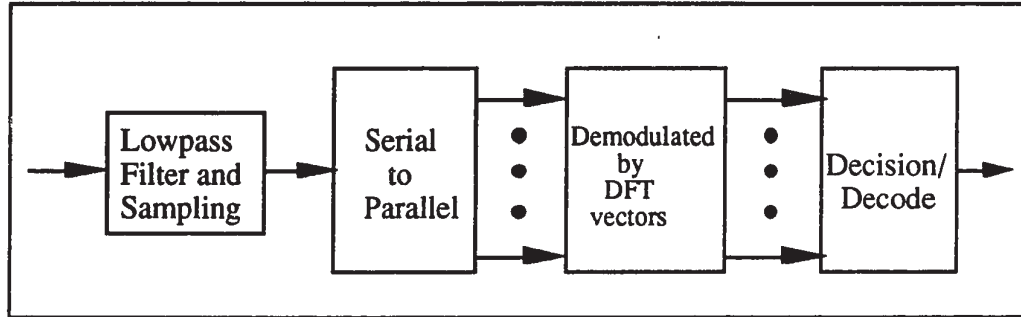


Figure 2.11: A basic multi-carrier receiver

A corresponding basic multi-carrier receiver is shown in Figure 2.11. The received signals first pass through a low-pass filter before sampling. The sampled data are then grouped together and demodulated by the DFT vectors, which is again implemented

by taking the FFT of each block. Finally, the bits are decoded with the appropriate signal constellation for each subchannel.

As in the case between baseband (PAM) and passband (QAM) systems, the output symbols of the transmitter can be either real (baseband) or complex (passband). Similar to QAM systems, if we modulate the multi-carrier system with some carrier frequency, complex output symbols are used. However, for applications where no carrier is used, the output symbols of the transmitter need to be real. To ensure a real-valued transmit signal, $\tilde{x}_k = \Re(\tilde{x}_k)$, it is necessary that we impose the constraint on the subsymbols that ($*$ denotes complex conjugate, $\Re$ denotes real part)

$$x_n = x_{N-n}^* \quad n = 1, \dots, N/2 - 1, \quad (2.30)$$

and

$$x_0 = \Re(x_0), \quad x_{N/2} = \Re(x_{N/2}). \quad (2.31)$$

2.2.2 Addition of Cyclic Prefix to the DFT/IDFT Multi-Carrier System

The basic structure of multi-carrier systems in Section 2.2.1 is optimum if the block length is allowed to be infinite (see Section 2.2.3). However, for practical systems with reasonable complexity and delay, the block length N has to be finite and reasonably small. For channels with severe intersymbol interference (ISI), this basic multi-carrier systems can suffer a substantial performance degradation.

To tackle this problem, [12], [2] suggested adding a cyclic prefix (CP) to the multi-carrier system. A cyclic prefix of length ν consists of the last ν samples of the transmitted block of samples. The CP prefixes the block of transmitted samples $\tilde{x}_0, \dots, \tilde{x}_{N-1}$ by the repeated samples, $\tilde{x}_{N-\nu}, \dots, \tilde{x}_{N-1}$, as shown in Figure 2.12. The length of the cyclic prefix, ν , is the order of a finite-length polynomial that is used to approximate the bandlimited channel characteristic.

To understand how the cyclic prefix works, we first examine one particular property of discrete Fourier transform, namely, multiplying two DFT sequences correspond

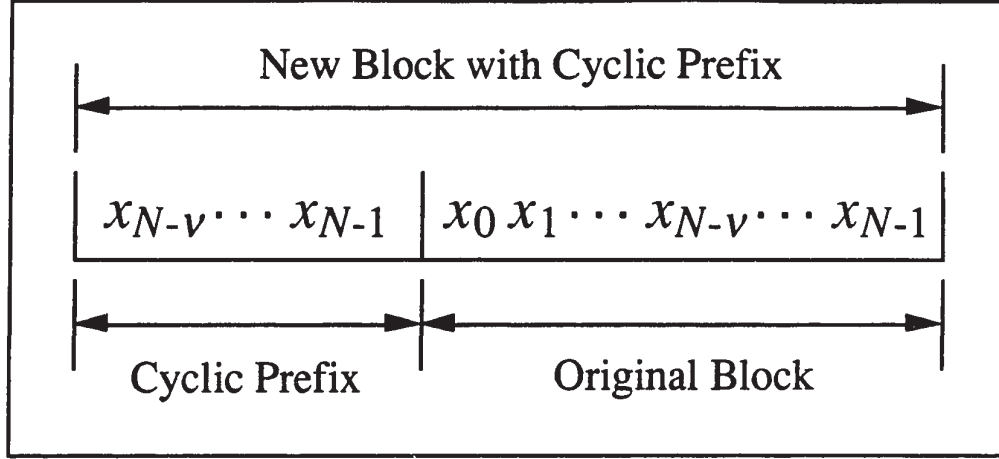


Figure 2.12: Cyclic prefix

to circular convolution of the two time sequences [15]. Denote $\text{DFT}(\{\tilde{x}_m\})$ as the DFT of the time sequence $\{\tilde{x}_m\}$, then

$$\text{DFT}(\{\tilde{x}_m\}) \otimes \text{DFT}(\{h_m\}) = \text{DFT}(\{\tilde{x}_m \bowtie h_m\}), \quad (2.32)$$

where $\tilde{x}_m \bowtie h_m$ means the circular convolution of the two sequences, and $\otimes$ denotes the component by component multiplication. The effect of filtering a sequence, however, corresponds to linear convolution of the sequence and the filter response. The channel output, $\tilde{y}_m$, in Figure 2.8 is a **linear** convolution of the channel input, $\tilde{x}_m$, and the channel response, h_m , plus noise n_m . For the time being, we will neglect the noise n_m so that the effect of cyclic prefix can be shown more directly.

$$\tilde{y}_m = \tilde{x}_m * h_m \quad (2.33)$$

Section 3.8 of [15] described how to use DFT/IDFT to compute a linear convolution of two sequence. The idea of using cyclic prefix is the converse of that method: by adding the cyclic prefix to the block to transmitted samples, the channel output sequence appears to be the circular convolution of the channel input sequence and the channel response

$$\text{With Cyclic Prefix: } \tilde{y}_m = \tilde{x}_m \bowtie h_m, \quad (2.34)$$

and after $\tilde{y}_m$ is demodulated by the DFT vectors,

$$\text{DFT}(\{\tilde{y}_m\}) = \text{DFT}(\{\tilde{x}_m \bowtie h_m\}), \quad (2.35)$$

and from equation (2.32), x_k can be retrieved by

$$x_k = \text{DFT}(\{\tilde{x}_m\}) \quad (2.36)$$

$$= \text{DFT}(\{\tilde{x}_m \bowtie h_m\}) \oslash \text{DFT}(\{h_m\}) \quad (2.37)$$

$$= \text{DFT}(\{\tilde{y}_m\}) \oslash \text{DFT}(\{h_m\}), \quad (2.38)$$

in the absence of noise, where $\oslash$ means the component by component division.

Thus the addition of cyclic prefix ensures independence between successive blocks of data and effectively eliminates ISI within the block when used in conjunction with the DFT/IDFT vectors. However, the addition of the cyclic prefix also causes an overall rate loss of $\nu/(N + \nu)$ since the CP carries no additional information or data. For channels with severe ISI, or, in other words if ν is excessively large, we would have to use a very large block length in order to minimize the effect of the rate loss, and that could result in prohibitive cost and unacceptable latency in practice.

The goal of this dissertation is to design a multi-carrier system that can achieve near-optimum performance and at the same time have reasonable complexity and latency (which means reasonably small block length). The detail of this system is described in Chapter 3.

2.2.3 Optimality of Multi-Carrier Systems

As mentioned earlier, the performance of multi-carrier systems has been proven theoretically to be optimum in [9] and [2]. In addition, Algoet and Cioffi have recently published the optimality of multi-carrier systems based on maximizing mutual information [16], [17]. Tu, in his Ph.D. thesis [18], summarizes Algoet's results with

particular emphasis on showing the optimality of vector coding and DFT-based multi-carrier systems. Instead of duplicating the efforts, the author hereby refers to the above publications to show the optimality of multi-carrier systems in theory.

2.3 Summary

In this Chapter, we include some basic digital communication systems as a background to understand multi-carrier systems. Passband (QAM) systems are analyzed in detail, since the multi-carrier system can be visualized as a set of QAM channels. A basic multi-carrier system is described. The use of discrete Fourier transform (DFT) vectors and inverse DFT vectors as the modulating and demodulating vectors simultaneously satisfy two important practical issues: the modulation can be implemented efficiently (via FFT algorithms) and the modulating vectors are invariant to channel characteristics. A cyclic prefix is added in the case for channels with intersymbol interference (ISI) to ensure independence between successive blocks of data and effectively eliminate ISI within the block when used in conjunction with DFT/IDFT, in the absence of noise. This basic multi-carrier system can achieve the highest possible performance if the block length is allowed to be infinite. For channels with severe ISI, a finite and reasonable complexity DFT-based system as described in this Chapter could suffer substantial performance degradation. Chapter 3 propose a multi-carrier structure that tackles this specific problem.

Chapter 3

The Discrete Multi-Tone Transceiver

We propose a multi-carrier modulation system called discrete multi-tone (DMT) modulation, which is based on the DFT-based multi-carrier system proposed in [2]. In this chapter we first look at the goal of the transceiver design, and then the optimality of the proposed design is shown for infinite complexity. A finite-length algorithm is then derived to maximize achievable data rate with a given type of specified complexity constraint. Finally, we describe the DMT transceiver in detail, including an estimation of the complexity of the system, transmit power level, and system initialization protocol.

3.1 Maximum Likelihood Receivers and Multi-Carrier Systems

Maximum Likelihood (ML) receivers that minimize probability of error in data transmission over channels with intersymbol interference can be realized in two popular forms, multi-carrier demodulators or sequence (Viterbi) detectors. Multi-carrier modulation/demodulation methods have long been known as optimum on the intersymbol interference channel [8] because they directly implement maximum likelihood (ML)

detection for the given transmitted signal (which is a sum of independently modulated subcarrier signals, whence the name “multi-carrier”). Multi-carrier methods require an infinite number of subcarriers to be used to achieve data rates near capacity [19]. Sequence (Viterbi) detectors are comparatively less ancient [3], and can be used to implement maximum-likelihood detection for single-carrier or baseband data transmission systems. Sequence detectors also require an infinite number of states to decode transmitted signals at rates near capacity on a bandlimited (or intersymbol interference) channel. The transmission engineer is then invariably left with the problem of trying to design a transmission system that exhibits performance as close to optimum as possible for given finite cost of implementation. The solution to this problem often leads to approximation of ML performance with a finite-complexity receiver. This chapter focuses on one approach to this problem that approximates optimum performance with a significantly reduced complexity receiver that consists of a finite-complexity ML detector and a corresponding front-end equalizer.

The measure of complexity for an ML detector will be specified by ν , the order of a finite-length polynomial that is used in the design of the finite-complexity ML receiver. That is, both sequence detectors and multi-carrier receivers need specification of some polynomial

$$b(D) = \sum_{m=0}^{\nu} b_m D^m \quad (3.1)$$

that is used to approximate the bandlimited channel characteristic at some sampling rate (and D is the complex delay variable corresponding to that sampling rate) in the discrete-time approximation to the channel. The multi-carrier receiver requires a block length N (basically twice the number of subcarriers) that is significantly larger than ν to implement the receiver without significant rate loss (a factor of $\nu/(N + \nu)$ in multi-carrier implementations). The sequence detector requires M^ν states where M is the number of transmitted symbol levels in a single-carrier or baseband transmission method. In both ML receivers, receiver cost is dominated by elements that depend directly on ν so that a fixed cost of implementation will force a given maximum value of ν that can be tolerated in the design. With ν as the complexity constraint parameter, the function of an equalizer is then to filter the channel so that the combined equalizer-channel characteristic closely approximates a

$b(D)$ that is chosen as the best $b(D)$ for the given channel. This concept is generically illustrated in Figure 3.1. For a given channel length of $\nu + 1$ samples, we need to find the best polynomial $b(D)$ and equalizer so that the ML detector performance is optimized when designed for that $b(D)$. We will suggest that the best such choice of (non-zero) $b(D)$ minimizes the mean-square error between the equalizer output and the filtered channel-input sequence $x(D)b(D)$ in Section 3.2.

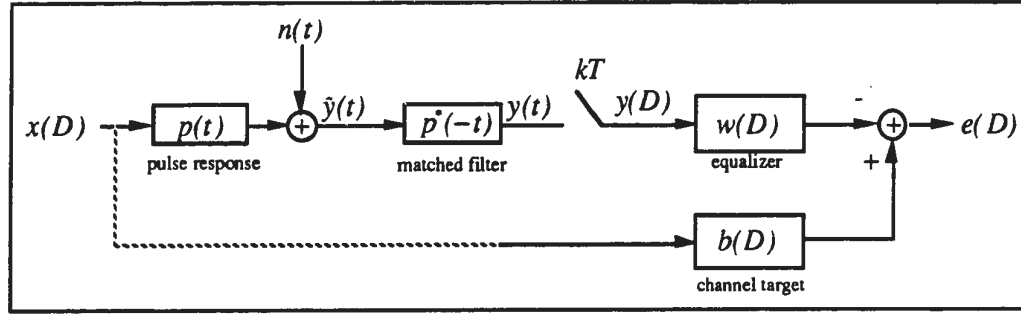


Figure 3.1: Input-assisted equalization

Section 3.2 also introduces notation and discusses a general approach to decision-aided equalization, paralleling some earlier developments of Salz and Muller [20], but continues on that earlier theory by focusing on only those receivers that can be implemented without need for any previous or future decisions. We derive in Section 3.2 a class of equalization structures that all have the same performance, and this class is found to include the MS-WMF that was conjectured in [21] to be a canonical receiver front-end. Surprisingly, this class also includes the so-called Auto-Regressive Moving Average (ARMA)-based filter that was empirically found in [22] to be an excellent front-end for some multi-carrier implementations. This class is also believed to include structures that are the infinite-length limits of some earlier systems derived in [23], [24] for M -ary single-carrier transmission on intersymbol interference channels. Our structures differ from those of [23] and [24] in that the equalization filtering and the ML detection are separable, for instance permitting the use of the same equalizer also with ML multi-carrier demodulator or a sequence detector. This motivates our

investigation of the case for finite ν in Section 3.3.

In Section 3.3, by constraining not only ν , but also the complexity of the equalizer, we find the optimum finite-impulse-response equalizer settings.¹ In this finite-length case, the relative delay characteristics of the equalizer and of $b(D)$ become crucial, especially as ν and the length of the equalizer become increasingly constrained. We introduce a search procedure that determines the best of the relative delay characteristics and quite often produces an equalizer that is not closely related to decision feedback, the latter of which might have been a designer's choice previous to the results of this dissertation.

With the results of Section 3.3, it is possible to analyze specific transmission channels and to ascertain the minimum ν and equalizer settings to approximate infinite-complexity performance closely. We often find the computation reduction is very large compared to the straightforward ML implementation and that the resulting reduced complexity is comparable (or often even less complex) to previous sub-optimal equalization approaches. In Section 5.4, we consider an actual design example that is based on a study of digital subscriber loops. We find that enormous complexity reduction occurs for negligible performance loss with respect to optimum.

3.2 Optimality and Equivalence of Infinite-Length Equalizers

The general baseband complex decision-aided, or in our study **input-aided**, equalization problem appeared in Figure 3.1. The channel output $\tilde{y}(t)$ is modelled as

$$\tilde{y}(t) = \sum_k x_k p(t - kT) + n(t) \quad (3.2)$$

where x_k is the channel input **symbol**, supplied at some **symbol rate** $1/T$, $p(t)$ is the channel **pulse response** (the convolution of the **transmit filter**, $\phi(t)$, and the channel **impulse response** $h(t)$, $p(t) = \phi(t) * h(t)$), and $n(t)$ is additive white Gaussian noise with power spectral density $N_0/2$ per real dimension. The channel

¹We note this deviates from an earlier approach of [23], [24] in which only ν is fixed but not the complexity of the equalizer.

input symbols may not be M -ary symbols in the case of multi-carrier modulation or coded single-carrier modulation, and in our development are only characterized as being independent identically distributed discrete-time samples with zero mean and **symbol energy** $E\{|x_k|^2\} = \mathcal{E}_x$. The channel output signal $\tilde{y}(t)$ is convolved with a matched filter $p^*(-t)$ (where a superscript of $*$ denotes conjugate), sampled at rate $1/T$, and then filtered by a linear equalizer. It is well-known [3] that matched-filter output is information lossless, and further ([25], Chapter 4) that when the equalizer is invertible that its output is also information lossless with respect to the channel input. This means that a maximum-likelihood detector applied to the signal at the output of the equalizer w_k in Figure 3.1 is equivalent in performance to the overall maximum-likelihood detector for the channel.

Heuristically, the function of the linear equalizer is to make the equalizer output appear like the output of the second filter, which is the convolution of the desired channel shaping function b_k and the known channel input x_k . We shall choose minimum mean-square error between the equalizer output and the desired channel shaping as a good measure of equalizer performance. To proceed further, we need to define and develop some mathematical concepts related to the channel.

3.2.1 Minimum Mean-Square Error Equalization

The **error sequence** can be written

$$e_k = b_k * x_k - w_k * y_k \quad (3.3)$$

where $*$ denotes convolution. The **signal** is defined as the sequence $b_k * x_k$, which has signal energy $\mathcal{E}_s = \|b\|^2 \mathcal{E}_x$. The **D-Transform** of a sequence x_k will be defined (see [21] for a more detailed development) by $x(D) \triangleq \sum_k x_k D^k$. We often use the notation $x^*(D^{-1}) = \sum_k x_{-k}^* D^k$ to represent the time-reversed conjugate of the sequence. We can thus write

$$e(D) = b(D)x(D) - w(D)y(D) \quad . \quad (3.4)$$

We minimize the **mean-square error** as

$$\sigma_e^2 = \min_{w_k, b_k} E[|e_k|^2] \quad (3.5)$$

where E denotes statistical expectation. A trivial solution is $w(D) = b(D) = 0$ with $\sigma_e^2 = 0$. To avoid this trivial solution, we constrain the signal power to be positive so that $\mathcal{E}_s > 0$ or, equivalently, $\|b\|^2 = \text{constant} > 0$. A related quantity is the **channel signal-to-noise ratio**

$$\text{SNR}_{\text{MFB}} = \frac{\mathcal{E}_s}{\sigma_e^2} . \quad (3.6)$$

When $\|b\|^2 > 0$, we will later see that SNR_{MFB} is independent of $\|b\|^2$. Maximizing SNR_{MFB} is then equivalent to minimizing σ_e^2 . While our problem formulation is the same as the formulation of decision feedback equalization (see [21]), we differ here in that we do **not** restrict $b(D)$ to be causal or monic.

An **autocorrelation sequence** for any sequence x_k is defined by $r_{xx,k} \triangleq E[x_l x_{l-k}^*]$ with D-transform $R_{xx}(D)$. The **power spectral density** of the sequence is $R_{xx}(\theta) \triangleq R_{xx}(D)|_{D=e^{-j\theta}}$ and the energy of the sequence is $\mathcal{E}_x = r_{xx,0} = \frac{1}{2\pi} \int_{-\pi}^{\pi} R_{xx}(\theta) d\theta$. The **cross-correlation sequence** between two sequences x_k and y_k is then $r_{xy,k} = E[x_l y_{l-k}^*]$ with D-transform $R_{xy}(D)$. It will be convenient to write ²

$$R_{xx}(D) = E[x(D)x^*(D^{-1})] \quad (3.7)$$

and

$$R_{xy}(D) = E[x(D)y^*(D^{-1})] . \quad (3.8)$$

As discussed in [21], any autocorrelation sequence that satisfies $\int_{-\pi}^{\pi} \ln |R_{xx}(\theta)| d\theta < \infty$ is canonically factorizable as $R_{xx}(D) = \tilde{\mathcal{E}}_x \tilde{x}(D) \tilde{x}^*(D^{-*})$ where $x(D)$ is monic ($x_0 = 1$), causal ($x_k = 0 \forall k < 0$), and minimum phase (all poles and zeros of the polynomial are outside the unit circle). The error sequence has autocorrelation transform $R_{ee}(D)$ and $r_{ee,0}$ is equal to the mean-square error.

A well-known principal in MMSE estimation is that the error sequence values e_k should be orthogonal to the inputs of estimation, some of which are the samples y_k . Thus $E[e_k y_l^*] = 0 \forall k, l$, which is compactly written

$$R_{ey}(D) = 0 \quad (3.9)$$

²Actually, we are implying a limiting normalized sum for each coefficient in the resultant polynomial of the form $\lim_{L \rightarrow \infty} \frac{1}{2L+1} \sum_{k=-L}^L$.

$$= E [e(D)y^*(D^{-1})] \quad (3.10)$$

$$= b(D)R_{xy}(D) - w(D)R_{yy}(D) \quad . \quad (3.11)$$

then,

$$w(D) = b(D) \frac{R_{xy}(D)}{R_{yy}(D)} \quad . \quad (3.12)$$

We note that

$$y(D) = R_{pp}(D)x(D) + n(D) \quad , \quad (3.13)$$

where $R_{pp}(D) \triangleq \frac{1}{2\pi} \int_{-\pi}^{\pi} \sum_n |P(\frac{\theta-2\pi n}{T})|^2 d\theta$. Then,

$$R_{xy}(D) = \mathcal{E}_x R_{pp}(D) \quad (3.14)$$

$$R_{yy}(D) = R_{pp}(D)(\mathcal{E}_x R_{pp}(D) + N_0) \quad . \quad (3.15)$$

We define the **channel autocorrelation** as $R_0(D) = \mathcal{E}_x R_{pp}(D) + N_0$, which always has canonical factorization (when $N_0 > 0$),

$$R_0(D) = \mathcal{E}_0 g(D)g^*(D^{-1}) \quad . \quad (3.16)$$

This factorization is called the “key equation” in [21].

We compute the error autocorrelation sequence as

$$R_{ee}(D) = b(D) \left(R_{xx}(D) - \frac{R_{xy}(D)}{R_{yy}(D)} R_{yx}(D) \right) b^*(D^{-1}) \quad , \quad (3.17)$$

where the reader may recognize the autocorrelation function of the MMSE linear equalizer error ([21]) in the center of the above expression. We simplify this expression further using the key equation to obtain

$$R_{ee}(D) = \left(\frac{\mathcal{E}_x}{\mathcal{E}_0} \right) N_0 \cdot \frac{b(D)b^*(D^{-1})}{g(D)g^*(D^{-1})} \quad . \quad (3.18)$$

Defining the (squared) norm of a sequence as

$$\|x\|^2 \triangleq \sum_{k=-\infty}^{\infty} |x_k|^2 \quad , \quad (3.19)$$

the mean square error from (3.18) is

$$\sigma_e^2 = \left(\frac{\mathcal{E}_x}{\mathcal{E}_0} \right) N_0 \left\| \frac{b}{g} \right\|^2 \quad . \quad (3.20)$$

From the Cauchy-Schwarz lemma, we know

$$\|b\|^2 = \left\| \frac{b}{g} \cdot g \right\|^2 < \left\| \frac{b}{g} \right\|^2 \|g\|^2 \quad (3.21)$$

or

$$\left\| \frac{b}{g} \right\|^2 > \frac{\|b\|^2}{\|g\|^2} . \quad (3.22)$$

Thus the MMSE is

$$\sigma_e^2 = \left(\frac{\mathcal{E}_x}{\mathcal{E}_0} \right) N_0 \frac{\|b\|^2}{\|g\|^2} , \quad (3.23)$$

which can be achieved by a multitude of $b(D)$, only one of which is causal and monic.

The MMSE solution to our equalization problem is then achieved by any $b(D)$ that satisfies

$$b(D)b^*(D^{-1}) = \frac{\|b\|^2}{\|g\|^2} g(D)g^*(D^{-1}) . \quad (3.24)$$

From (3.12) to (3.16), one finds $w(D)$ from $b(D)$ as

$$w(D) = \frac{\mathcal{E}_x}{\mathcal{E}_0} \frac{b(D)}{g(D)g^*(D^{-1})} , \quad (3.25)$$

which is always invertible when $N_0 > 0$. The channel signal-to-noise ratio is

$$\text{SNR}_{\text{MFB}} = \frac{\mathcal{E}_0}{N_0} \|g\|^2 , \quad (3.26)$$

independent of $b(D)$ as promised earlier.

Note again that any invertible choice of $w(D)$ would still lead to an output for which ML detection is equivalent to $\tilde{y}(t)$. Forney ([3]) chose a filter that was equivalent to the limit as $N_0 \rightarrow 0$ of the causal choice of $w(D)$ in (3.25), while Ungerboeck ([26]) in his study of generalized ML receivers choose $w(D) = 1$. Both of these choices lead to ML receivers that are also optimum, but neither choice also minimizes mean-square error. We shall see that a choice that also minimizes mean-square error can lead to much lower complexity of implementation for a given level of performance.

3.2.2 Solutions

As mentioned earlier, there are many choices for $b(D)$. We investigate some of those choices, which motivate our finite-length study of Section 3.3.

The MMSE-DFE

The Minimum-Mean-Square Error Decision Feedback Equalizer (MMSE-DFE), see for instance [27], chooses

$$b(D) = g(D) \quad (3.27)$$

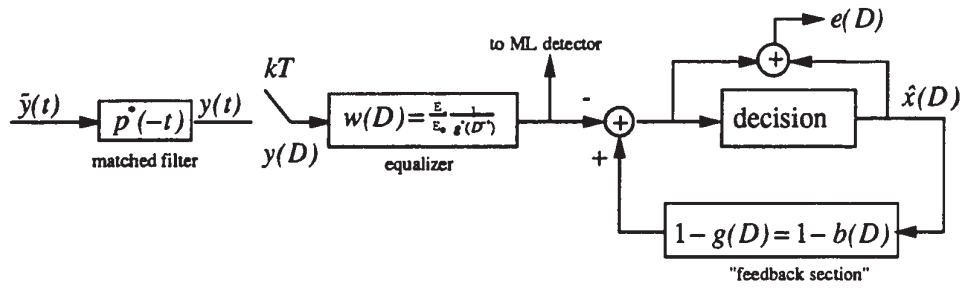
so that $b(D)$ is monic and causal. This particular causal choice for $b(D)$ has the advantage that previous decisions, if assumed to be correct, are all that is needed to implement the input-aided equalization problem – so it is possible to implement both filters at the receiver with symbol-by-symbol decisions as shown in Figure 3.2(a). So-called “Tomlinson-Precoded” approaches to DFE [28] also require that the choice of $b(D)$ be monic and causal.

In ML detection with the DFE, the feedback filter $b(D)$ would not be implemented, but the ML detector would need to be designed with knowledge of the symbol constellation, $b(D)$, and the properties of $e(D)$ in Figure 3.1. The error $e(D)$ is white, but not necessarily Gaussian because of the dependency on $x(D)$. Nevertheless, an ML detector designed for $b(D)$ under the assumption that $e(D)$ is a white Gaussian noise sequence would perform very close to the true ML detector that considered the exact statistics of $e(D)$. This ML detector could be implemented by the Viterbi algorithm if $b(D)$ is of finite degree [3].

For intermediate complexity ML detectors [29], one could design the ML sequence detector for some truncated version of $b(D)$ that kept only the first $\nu + 1$ taps of $b(D)$,

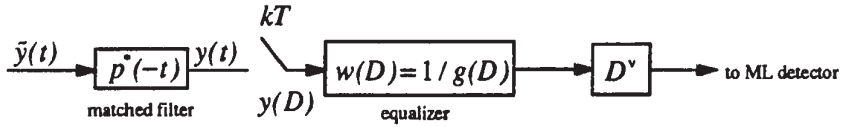
$$\tilde{b}(D) = \sum_{k=0}^{\nu} b_k D^k \quad (3.28)$$

and the feedback structure would be used to cancel previous intersymbol interference that was caused by data symbols more than ν symbol periods in the past as shown in Figure 3.2(b). For such a truncated response, the performance of the ML detector would be as high as possible among such truncated choices with the DFE because of the minimum-phase property of $b(D) = g(D)$. However, when considering finite-complexity, the length of $\tilde{b}(D)$ is only one parameter of interest, and the structure suggested here is often far from the best structure for a given implementation complexity as we shall see in Sections 3.3 and 5.4.



a). Decision Feedback Special Case ($b(D)$ is causal, min phase)

(note that $b(D)$ must be causal and monic even with Tomlinson Precoding)



b). ARMA Special Case ($w(D)$ is causal, min phase)

Figure 3.2: Some choices for the input-aided ML equalizer.

The ARMA Equalizer

The Auto Regressive Moving Average (ARMA) Equalizer corresponds to the choice

$$b(D) = \frac{\mathcal{E}_0}{\mathcal{E}_x} g^*(D^{-1}) \quad . \quad (3.29)$$

In this case, $b(D)$ is noncausal and maximum phase (all roots inside the unit circle). This is another optimum solution and the MMSE level and the SNR_{MFB} are the same as all the other optimum solutions, including the MMSE-DFE. One would not make the choice in (3.29) if symbol-by-symbol decisions are of interest because neither future decisions are available in the receiver nor can a Tomlinson precoder be implemented (properly) with a noncausal $b(D)$.

We call this structure an ARMA equalizer because $w(D) = 1/g(D)$ is monic and causal and because we can write

$$y(D) = \frac{b(D)}{w(D)} x(D) + \frac{e(D)}{w(D)} \quad . \quad (3.30)$$

When the error signal is small (as is usually the case in digital communication), the input/output relation between $x(D)$ and $y(D)$ is known (almost) as an ARMA model.³ The only difference between this model and a true ARMA model is that $b(D)$ and $w(D)$ in our development are not necessarily finite-length polynomials (they can themselves be rational fractions of polynomials or of infinite duration). However, if we assume that in practical situations both $b(D)$ and $w(D)$ can be construed as finite in length, then the relation in (3.30) is a “pole/zero” model for the data transmission channel that is stable because $w(D)$ is now minimum phase. Since $b(D)$ is also considered to be of some finite length ν , then we can write

$$y(D) = D^{-\nu} \cdot \frac{\bar{b}(D)}{w(D)} x(D) + \frac{e(D)}{w(D)} \quad (3.31)$$

where $\bar{b}(D) = \sum_{k=0}^{\nu} g_k^* D^k$ ($\neq b^*(D^{-1})$), a time-reversed $b(D)$ having the same magnitude but different phase. Note $\bar{b}(D)$ is causal and since $w(D)$ is monic and causal, then $\bar{b}(D)/w(D)$ is indeed a pole/zero or ARMA model for the channel characteristic (modulo the linear-phase offset implied by the $D^{-\nu}$ term).

³Actually, this model is more exactly called an “ARMAX” model when the error signal is non-zero, as discussed in [30].

Forney [3] once conjectured that a finite-state ML detector might exist for an infinite-length channel model, but left that possibility for future work. To the best of our knowledge, such a detector had (to this point) not been found. Now, using the ARMA interpretation of the channel as a pole/zero model (corresponding to an infinite-length channel) with a finite number of poles and zeros, we see that equalization of $y(D)$ by $w(D)$ produces

$$w(D)y(D) = D^{-\nu}\bar{b}(D)x(D) + e(D) \quad (3.32)$$

a white-noise (actually, white-error) channel model that was produced by an invertible lossless transformation from $y(D)$. To the degree that $e(D)$ approximates a Gaussian noise, then a Viterbi detector with a finite number of states (M^{N_z}) where $N_z = \nu$ is the number of zeros in the pole/zero channel model will be an optimum detector by the famous reversibility theorem of Wozencraft and Jacobs [25]. Thus, we have the following proposition:

Proposition 3.2.1 (Proposed Solution – Forney’s Conjecture) *The solution to Forney’s once-conjectured finite-complexity maximum-likelihood sequence detector for a channel that can be modeled by a pole/zero polynomial is to use the filter $w(D)$ of the ARMA equalizer on the matched filter output, and then to use a Viterbi sequence detector on the resulting equalizer output where that sequence detector has been designed to have $M^{N_z} = M^\nu$ states where ν is the (finite) number of zeros in the channel model.*

One might continue by noting that when $N_0 \rightarrow 0$, the ARMA model reduces to what could be called a zero-forcing (exact) pole/zero model (similar to the zero-forcing DFE) if $w(D)$ is still invertible. In this case, the feedforward filter (if it exists) would just cancel the channel poles. Of course, since the noise has zero variance, the resulting detector would be trivial. When the noise is non-zero, we suggest the above ARMA equalizer.

3.2.3 Comparison of MMSE-DFE and ARMA

Both of the MMSE-DFE and the ARMA equalizer have the same performance (when ML detection is used). Essentially the causality/noncausality conditions of the feedforward filters and feedback filters have been interchanged. Because $b(D)$ is never used in the receiver (only for our analysis), either structure could be used in a practical situation. A Viterbi detector designed for M -ary transmission would have M^ν states in either case, and would perform the same. A multi-carrier modulation method used with $w(D)$ equalization would also perform the same for either choice of $w(D)$ (or any other optimum choice).

Thus, it would appear that since all structures are equivalent, we might as well continue with the use of the MMSE-DFE, since it is a well-known receiver structure. However, when finite-length filters are used, this can be a very poor choice in attempting to maximize performance for a given complexity of implementation. As a trivial example, consider a channel that is maximum phase, the feedforward filter for a MMSE-DFE, in combination with the matched filter, will try to convert the channel to minimum-phase ([31]) at the output of $w(D)$, leading to a long feedforward section when (as in practice) the matched-filter and feedforward filter are implemented in combination in a fractionally spaced equalizer ([32]). On the other hand, the ARMA equalizer feedforward section (combined with matched filter) is essentially (modulo an anti-alias filter before sampling) one nonzero tap that adjusts gain. The opposite would be true for a minimum-phase channel.

In data transmission, especially on cables or over wireless transmission paths, the channel characteristic is almost guaranteed to be of mixed-phase if any reflections (multi-path, bridge-taps, gauge-changes, slight imbalance of terminating impedances) exist. It is then natural (and correct) to infer that the best input-aided equalization problem is one that chooses $b(D)$ and $w(D)$ to be both mixed-phase also. The problem then becomes for a finite-length feedforward section ($w(D)$) of $M+1$ taps and a finite-length channel model of $\nu+1$ taps ($b(D)$), to find the best $b(D)$ and $w(D)$ for an ML detector designed for $b(D)$, the signal constellation, and additive white Gaussian noise (even if the noise is not quite Gaussian or white). This is the problem addressed in Section 3.3 and the solution is often neither decision feedback nor ARMA filtering.

3.3 Design Rule for Finite-Length Equalizers

Having established the possibility of certain mixed- (or maximum-) phase filterings being best for finite-length $w(D)$ and $b(D)$ under the assumed signal-to-noise measure of performance, we proceed in this section to derive the settings for $w(D)$ and $b(D)$. In so doing, we will absorb matched filtering in a finite-length fractionally spaced feedforward equalizer, as would most likely be implemented in practice (even if the feedforward filter is approximated through symbol-spacing). This requires a development of matrix terminology that we discuss in Subsection 3.3.1. We then proceed in Subsection 3.3.2 to derive the solution for $w(D)$ and $b(D)$ for any given fixed lengths under a variation of relative phasing of the feedforward and feedback sections. We find that the best combination is further determined by searching all possible solutions in the classes of allowed relative delays or phases for the one with the best channel SNR. We then pose this solution as best before illustrating the advantages in Section 5.4.

3.3.1 Finite-Length Terminology

With fractionally spaced equalizer sampling at time instants $t = kT - \frac{iT}{l}, i = 0, \dots, l-1$, equation (3.2) becomes

$$\tilde{y}(kT - \frac{iT}{l}) = \sum_m x_m p(kT - \frac{iT}{l} - mT) + n(kT - \frac{iT}{l}), \quad i = 0, \dots, l-1, \quad (3.33)$$

or equivalently,

$$\underline{\tilde{y}}_k = \sum_m x_m \underline{p}_{k-m} + \underline{n}_k, \quad (3.34)$$

where

$$\underline{\tilde{y}}_k = \begin{bmatrix} \tilde{y}(kT) \\ \tilde{y}(kT - \frac{T}{l}) \\ \vdots \\ \tilde{y}(kT - \frac{l-1}{l}T) \end{bmatrix}, \quad \underline{p}_k = \begin{bmatrix} p(kT) \\ p(kT - \frac{T}{l}) \\ \vdots \\ p(kT - \frac{l-1}{l}T) \end{bmatrix}, \quad \text{and } \underline{n}_k = \begin{bmatrix} n(kT) \\ n(kT - \frac{T}{l}) \\ \vdots \\ n(kT - \frac{l-1}{l}T) \end{bmatrix}. \quad (3.35)$$

We also assume that $p(t)$ extends only over a finite time interval $0 \leq t \leq LT$, thus $\underline{p}_k = 0$ for $k < 0, k > L$. For $M + 1$ successive l -tuples of samples of $\tilde{y}(t)$,

$$\tilde{\mathbf{y}}_k = \mathbf{P}\mathbf{x}_k + \mathbf{n}_k, \quad (3.36)$$

where

$$\tilde{\mathbf{y}}_k = \begin{bmatrix} \tilde{y}_k \\ \tilde{y}_{k-1} \\ \vdots \\ \tilde{y}_{k-M} \end{bmatrix}, \quad \mathbf{x}_k = \begin{bmatrix} x_k \\ x_{k-1} \\ \vdots \\ x_{k-M-L} \end{bmatrix}, \quad \text{and } \mathbf{n}_k = \begin{bmatrix} n_k \\ n_{k-1} \\ \vdots \\ n_{k-M} \end{bmatrix}, \quad (3.37)$$

and

$$\mathbf{P} = \begin{bmatrix} \underline{p}_0 & \underline{p}_1 & \cdots & \underline{p}_L & 0 & \cdots & 0 \\ 0 & \underline{p}_0 & \underline{p}_1 & \cdots & \underline{p}_L & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \underline{p}_0 & \underline{p}_1 & \cdots & \underline{p}_L \end{bmatrix}. \quad (3.38)$$

We set $l = 1$ from now on to simplify notations, noting that the following algorithms are valid for other values of l . We assume that the length of $w(D)$ is $M + 1$, $\mathbf{w} = [w_0 \ w_1 \ w_2 \ \dots \ w_M]$, and the length of $b(D)$ is $\nu + 1$, $\mathbf{b} = [b_0 \ b_1 \ b_2 \ \dots \ b_\nu]$. The matched filter is absorbed into the feedforward equalizer $\mathbf{w}$. The error is

$$e_k = \mathbf{b}\mathbf{x}_{k-\Delta} - \mathbf{w}\tilde{\mathbf{y}}_k, \quad (3.39)$$

where Δ is an integer corresponding to delay (positive) or advance (negative), and

$$\mathbf{x}_{k-\Delta}^T = [x_{k-\Delta} \ x_{k-\Delta-1} \ \dots \ x_{k-\Delta-\nu}]. \quad (3.40)$$

3.3.2 An Algorithm to Find the Best Finite-Length Equalizer for Multi-Carrier System

Once we decide on the values for M and ν , we proceed to the following $\nu + 1$ cases: setting $b_i = 1$ for each i in turn, $0 \leq i \leq \nu$.

For each case, equation (3.39) now becomes

$$e_k = x_{k-\Delta-i} - \mathbf{v} \mathbf{u}_k, \quad (3.41)$$

where

$$\mathbf{u}_k = \begin{bmatrix} \tilde{\mathbf{y}}_k \\ \mathbf{x}_{(k-\Delta)|i} \end{bmatrix}, \quad (3.42)$$

$$\mathbf{x}_{(k-\Delta)|i}^T = [x_{k-\Delta} \ x_{k-\Delta-1} \ \dots \ x_{k-\Delta-i+1} \ x_{k-\Delta-i-1} \ \dots \ x_{k-\Delta-\nu}], \quad (3.43)$$

and

$$\mathbf{v} = [\mathbf{w} \ \mathbf{b}_i], \quad (3.44)$$

where

$$\mathbf{b}_i = [-b_0 \ -b_1 \ \dots \ -b_{i-1} \ -b_{i+1} \ \dots \ -b_\nu]. \quad (3.45)$$

To minimize MSE, set $E[e_k \mathbf{u}_k^*] = 0$, which leads to the following solution:

$$\mathbf{v} = R_{x_{k-\Delta-i} \mathbf{u}} R_{\mathbf{u} \mathbf{u}}^{-1}, \quad (3.46)$$

and

$$\sigma_{MSE,i}^2 = E[e_k e_k^*] \quad (3.47)$$

$$= \mathcal{E}_x - \mathbf{v} R_{\mathbf{u} x_{k-\Delta-i}}^*. \quad (3.48)$$

In each case, Δ is varied to minimize MSE. Note that $b_0 = 1$ case corresponds to the familiar DFE solutions.

After performing the above algorithms for all ν cases, the solution corresponding to the smallest $\sigma_{MSE,i}^2$ is the optimum tap setting for the system.

For multi-carrier system, the effect of finite FFT size as well as the effect of cyclic prefix (signal has cyclic prefix while noise does not) contributes to the overall achievable data rate. While using the MMSE solution provides near optimum performance, other solutions using the above algorithms should be checked and compared also. Table 3.1 summarize the algorithm for finding the best FIR taps for multi-carrier system.

1. Fix desired values for M and ν .
2. Set $i=0$.
3. Calculate the MSE and the taps setting according to equation (3.41) to (3.48), with various values for Δ .
4. Pick 2 to 3 different Δ s from step (3) corresponding to the smallest MSE and save those taps settings.
5. Increment i by 1, repeat step (3) and (4) until i exceeds ν .
6. Check the achievable data rate with each taps setting, and pick the one with the highest throughput.

Table 3.1: Algorithm to find the best finite-length equalizer for DMT system

3.4 The Discrete Multi-Tone (DMT) Transceiver

3.4.1 System Description

We assume from this point on that the channel input samples are real. The block diagram of the DMT transmitter is shown in Figure 3.3. The input data bit stream is parsed and transformed to $N/2$ QAM subsymbols (there are actually $N/2 - 1$ complex QAM subchannels, and a real channel at D.C. and at Nyquist). A coset-encoder can be applied if trellis coding is used [33], [34]. The trellis encoder sequentially

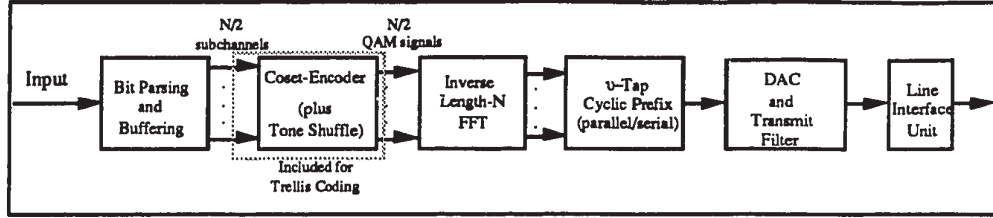


Figure 3.3: The DMT transmitter

processes the frequency-indexed subsymbols to avoid the large latency and memory requirement that would occur with multiple trellis encoders. This process requires that the subchannel indices be shuffled to avoid any minor correlation between the noise on adjacent frequency subchannels. An N -point complex-to-real IFFT is then performed to the output of the encoder. Finally, a cyclic prefix of length ν is placed at the beginning of the corresponding block of modulated transmit samples, and the extended block of $N + \nu$ samples is then applied to the channel serially through the DAC and line interface unit.

The DMT receiver is shown in Figure 3.4 and parallels the structure of Figure 3.1. The error equation, modulo the effects of FFTs and IFFT, is given by

$$e(D) = b(D)x(D) - w(D)y(D), \quad (3.49)$$

which is the same as equation (3.4).

After low-pass filtering (LPF) and sampling, the received signals are processed by a short, fixed-tap FIR feedforward equalizer, $w(D)$. The redundant symbols caused by the cyclic prefix are also removed during this process. The resulting data are then grouped into blocks of N symbols, and a length- N FFT is performed on each block. The output data of the FFT are then multiplied, independently, by a band of complex, one-tap filters. This set of 1-tap filters is used to adjust the gain and phase of each subchannel so that a common decision element can be used. $DFT(b)$ is the N -point FFT of the channel target polynomial, $b(D)$. This block is obtained from the finite-length algorithm. The feedforward equalizer is made adaptive to account

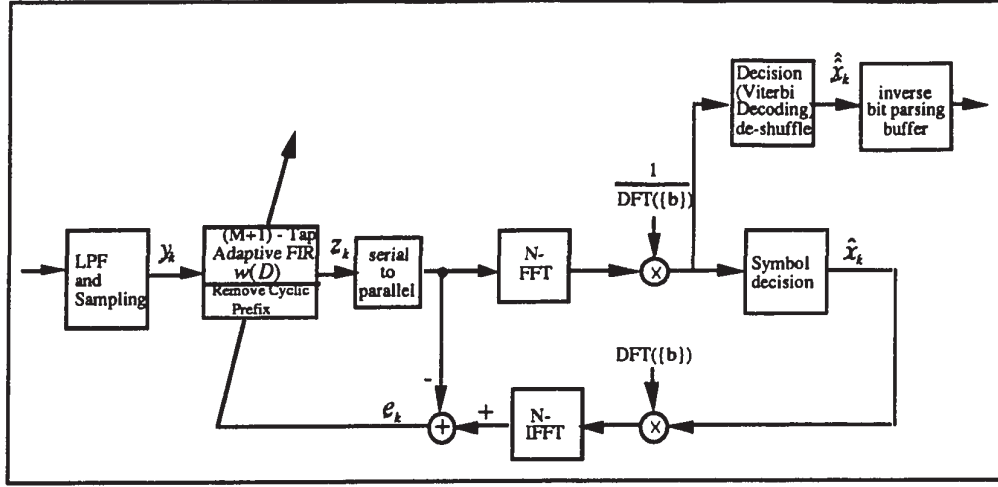


Figure 3.4: The DMT receiver (old)

for any inaccuracy in channel estimation, and small variations of channel and noise characteristics over time caused by, for instance, temperature variation. A Viterbi decoder is required if trellis coding is desired. In that case, the symbol decision, along with the one-tap filters and the N -point IFFT at the lower portion of Figure 3.4, is used only to derive an error signal to update the weights of the adaptive equalizers.

Many adaptive algorithms are available for use. Here, we focus on one particular algorithm that is known as the time-domain block least mean-square (LMS) algorithm, studied extensively in [35]. Assuming we have an $(M+1)$ -tap FIR adaptive filter, $\mathbf{w}$, and the block length that we use is equal to N , (so after adding the cyclic prefix, we have a block of $N + \nu$ samples) then

$$\mathbf{w}_k = \begin{bmatrix} w_{k,0} & w_{k,1} & \dots & w_{k,M} \end{bmatrix}, \quad (3.50)$$

where k is the time index. The input to this FIR equalizer is

$$\mathbf{y}_k = \begin{bmatrix} y_k & y_{k-1} & \dots & y_{k-M} \end{bmatrix}, \quad (3.51)$$

and the output during the k^{th} iteration is

$$z_k = \mathbf{w}_k \mathbf{y}_k^T. \quad (3.52)$$

Note that there is no need to compute the first ν output points for each block of $N + \nu$ received points, since we need to remove the redundant samples before taking the DFT of the equalizer output. The error vector is generated every $N + \nu$ samples of y_k

$$\mathbf{e} = \begin{bmatrix} e_{k+\nu} & e_{k+\nu+1} & \dots & e_{k+\nu+N-1} \end{bmatrix}, \quad (3.53)$$

and the weight updating equation is

$$\mathbf{w}_{k+N+\nu} = \mathbf{w}_k + 2\mu \sum_{i=0}^{N-1} e_{k+\nu+i} \mathbf{y}_{k+\nu+i}. \quad (3.54)$$

The drawback of the DMT receiver in Figure 3.4 is that it requires a fairly substantial amount of computation, as shown in Section 3.5 and [36]. To reduce the complexity of the DMT receiver, the error signal is pushed back to the frequency domain. The new, revised DMT receiver appears in Figure 3.5 and this receiver theoretically achieve the same performance as the previous receiver. The main advantage of this revised DMT receiver is that it has a substantially lower complexity requirement, as shown in Section 3.5. However, the error equation is now given by (modulo the effects of FFT)

$$e(D) = x(D) - \frac{w(D)}{b(D)} y(D), \quad (3.55)$$

which is essentially equation (3.4) divided by $b(D)$. The minimum mean-square error criteria as described in Section 3.2 leads to the same solution.

The feedforward equalizer $w(D)$ is now fixed and the taps setting is determined through the finite-length algorithm. The adaptive portion of the receiver is now shifted to the set of one-tap complex filters.

As before, the last N samples of the $N + \nu$ samples that correspond to the transmit block are extracted from the equalizer output. An N -point FFT is performed on the equalizer output. The FFT outputs, $Z_n, (n = 0, 1, \dots, N - 1)$ are multiplied by N

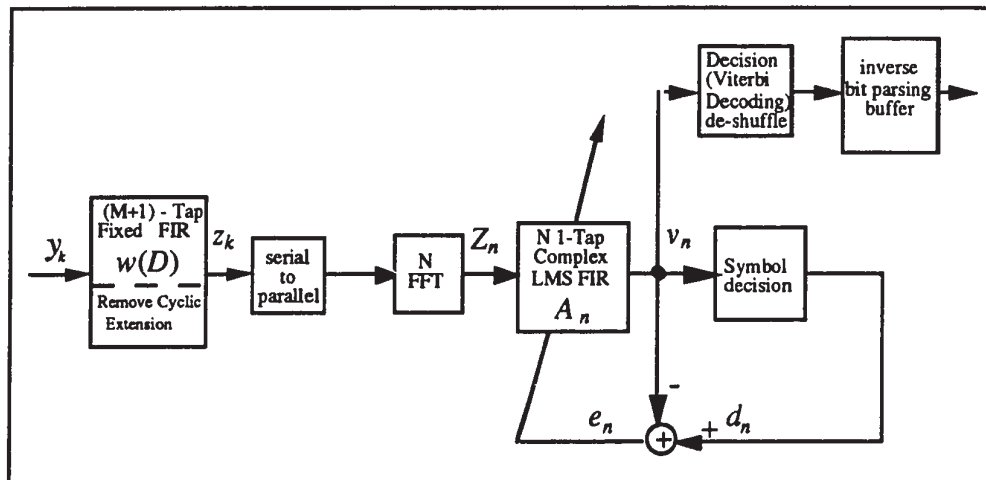


Figure 3.5: The DMT receiver (revised)

complex, 1-tap adaptive filters, A_n , ($n = 0, 1, \dots, N - 1$) so that a common decision device can be used to estimate the subsymbols on each of the subchannels. The initial tap setting for A_n is

$$A_n = B_n^{-1}, \quad n = 0, 1, \dots, N - 1, \quad (3.56)$$

where

$$B_n = \sum_{k=0}^{N-1} b_k e^{-j \frac{2\pi}{N} kn}, \quad n = 0, 1, \dots, N - 1, \quad (3.57)$$

the FFT of $b(D)$. The resulting output data,

$$v_n = A_n Z_n, \quad n = 0, 1, \dots, N - 1, \quad (3.58)$$

can then be decoded, which in the case of an applied trellis code requires a Viterbi decoder as shown. The symbol decisions are used (in the coded case) only to derive an error signal, so that the adaptive updating mechanism can be used to allow for slight channel variation. The corresponding error is

$$e_n = d_n - v_n, \quad n = 0, 1, \dots, N - 1. \quad (3.59)$$

Standard LMS algorithm can be used for updating,

$$A_n \leftarrow A_n + 2\mu_n e_n Z_n^*, \quad n = 0, 1, \dots, N - 1, \quad (3.60)$$

where μ_n is the step size for each subchannel, and can be adjusted independently to optimize the rate of convergence.

Since b_k and z_k are real, $B_n = B_{N-n}^*$ and $Z_n = Z_{N-n}^*$, so that (3.58) - (3.60) are computed only through $n = N/2$.

3.4.2 Performance Analysis for the Discrete Multi-Tone System

The discrete multi-tone (DMT) transceiver system can be visualized as combining a set of QAM channels in parallel. As a result, the derivation of the achievable data rate for the discrete multi-tone (DMT) transceiver system parallels closely to that of QAM systems as described in Section 2.1.2.

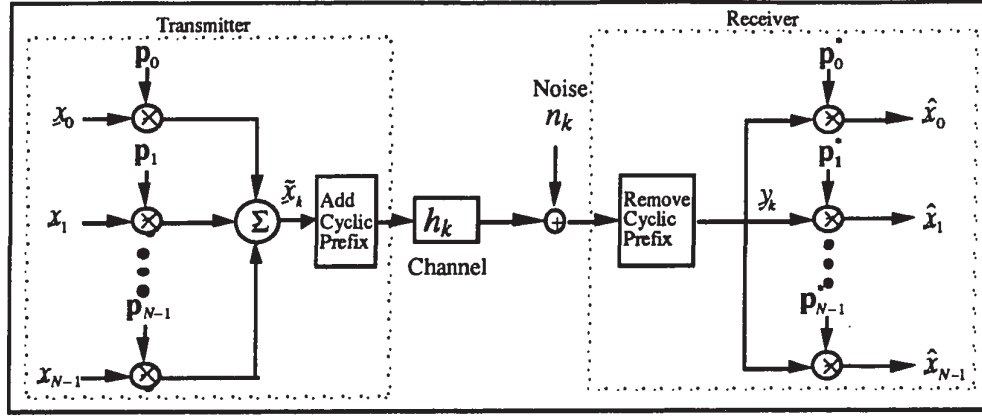


Figure 3.6: A Simplified DMT Transceiver System

Consider the block diagram shown in Figure 3.6, the length of the cyclic prefix is equal to ν . When $\nu + 1$ is the length of the channel impulse response in sample periods, then the successive blocks of transmitted data do not interfere with each other, and one can write (see, [2]) that

$$y_k = |H_k| \tilde{x}_k + n_k, \quad (3.61)$$

where

$$H_k = h_0 + h_1 e^{-j\frac{2\pi}{N}k} + h_2 e^{-j\frac{2\pi}{N}2k} + \dots + h_\nu e^{-j\frac{2\pi}{N}\nu k}. \quad (3.62)$$

For the DMT system, a block length of N corresponds to $N/2 - 1$ QAM subchannels and one real channel at D.C. and at Nyquist frequency. Many communication channels have a notch at D.C. and usually have a poor response near Nyquist, and so these two channels are typically not used. For channels that have reasonable gains at D.C. and/or Nyquist, we could treat them as two PAM channels. To simplify the equations, we assume from now on that D.C. and Nyquist subchannels are not used.

Assume that $\mathcal{E}_n = E|x_n|^2$ is the signal energy of the n^{th} subsymbol and σ_n^2 is the corresponding n^{th} subchannel noise energy per dimension. The signal-to-noise ratio (SNR) on the n^{th} subchannel is denoted by SNR_n and is computed by

$$SNR_n = \frac{\mathcal{E}_n |H_n|^2}{\sigma_n^2} \quad , \quad (3.63)$$

where $|H_n|^2$ is the power gain of the n^{th} subchannel. Parallel to Equation (2.16), the number of bits for the n^{th} subchannel, b_n , is given by

$$b_n = \log_2 \left(1 + \frac{SNR_n}{\Gamma} \right) \quad , \quad (3.64)$$

where Γ is the SNR gap as defined in Section 2.1.2

$$\Gamma \text{ (dB)} = [Q^{-1}(P_b)]^2 + \gamma_m - \gamma_c - 4.77 \text{ (dB)}. \quad (3.65)$$

A trellis code [33] of coding gain γ_c can be applied to each subchannel so that the energy allocated to this channel will sustain a greater number of bits than in the absence of the applied trellis code. If no code is used, $\gamma_c = 0$ dB.

The total data rate, R , (in bits/sec) is given by

$$R = \frac{S}{N + \nu} \sum_{n=1}^{N/2-1} b_n \quad (3.66)$$

$$= \frac{S}{N + \nu} \sum_{n=1}^{N/2-1} \log_2 \left(1 + \frac{SNR_n}{\Gamma} \right) \quad (3.67)$$

$$= \frac{S}{N + \nu} \log_2 \left[\prod_{n=1}^{N/2-1} \left(1 + \frac{SNR_n}{\Gamma} \right) \right] \quad , \quad (3.68)$$

where S is the sampling rate.

3.4.3 Channel Identification and Initialization Protocol

As the DMT transmitter requires knowledge (either directly, or indirectly through knowing the bit allocation, b_n) of the channel, the channel impulse response and crosstalk noise spectrum must be estimated during initialization. The more accurate the initial estimate, the more reliable the transmission with DMT. Accuracy requirements are significantly mitigated by the adaptive filters in the receiver in practice, but are still important. (We note that decision feedback approaches that employ

Tomlinson precoding [37], [38] also need to know the channel at the transmit side and require an even greater accuracy than DMT [39], even when an adaptive feedforward section is used in the DFE receiver.) Rather than pursue channel identification in detail in this dissertation, (see for example [40],[41],[42] for details) we here only briefly summarize the basic steps in the initialization protocol. The complete start-up time (including channel identification, SNR calculation, and bit and energy allocation) is about a few seconds ⁴.

1. After receiving a request for transmission signal, some loopback and test patterns are exchanged, followed by timing and synchronization patterns.
2. A predefined pseudo-random sequence is sent from the transmitter to the receiver for spectral estimation. A channel pulse response is estimated from the received training pattern.
3. Based on the result of the above channel spectral estimation, and after deciding on the number of taps for the feedforward equalizer and the length of the cyclic prefix, the finite-length optimization algorithm is performed to identify the FIR equalizer setting and the initial taps setting for the set of one-tap LMS filters.
4. The SNR for each frequency-indexed subchannel is then computed, and finally the transmit bit and energy allocation across the frequency band is calculated.
5. The bit assignments are sent to the transmitter, in a reliable fashion.
6. For bi-directional transmission, the above steps are repeated for the other direction.
7. After frame synchronization and start commands are exchanged, the DMT transceiver system is ready for actual data transmission.

We have not considered the implementation of echo cancellation in this dissertation. We note that low-computation/high-performance echo cancellation is possible with DMT (see [43]).

⁴A fast start-up algorithm to find the taps for the feedforward equalizer that is suitable for DSP implementation has been proposed by Prof. John Cioffi and the author (patent pending).

3.4.4 Transmit Power Level

As shown in Section 4.4.2, we can improve performance by increasing the average input power. However, due to physical constraints, only a certain range of average input power level is feasible. As a result, maximum average input power or energy is a common constraint that is usually imposed at the input of the channel (or equivalently, at the output of the transmitter).

Transmit Power Adjustment

We have assumed an average energy constraint of $\mathcal{E}_x$ per transmitted subsymbol, before adding the cyclic prefix. The IFFT and FFT do not change the energy of a block of data, so the energy constraint specified at the output of the IFFT in the transmitter is the same as at the input of the IFFT, i.e., the total average energy per block at the input of the IFFT is limited to $N\mathcal{E}_x$. However, the total average energy of a ν -sample cyclic prefix is not necessarily equal to $\nu\mathcal{E}_x$. We will thus briefly investigate the impact of cyclic prefix to the average energy constraint.

The real part and the imaginary part of the noise variance after the FFT at the receiver are in general unequal, for any specific frequency band, due to the correlated, non-i.i.d., crosstalk noise at the input to the FFT (see Appendix A). This inequity of variance is more profound at smaller block length (larger than 4), and the difference between the real and imaginary noise variance gradually diminishes as longer block length is used; as block length approaches infinity, the difference becomes zero. To optimize bit error rate throughout the subchannels, we would then need to assign unequal signal energy on the real and imaginary part of each frequency band. A fairly tight bound for the average energy of a block of $N + \nu$ samples (after cyclic prefix is added) is derived in Appendix B.

However, as the block length increases, the difference between the real and the imaginary noise variance becomes insignificant, and we can simply assign equal energy on the real and imaginary part on each corresponding frequency band without any noticable performance degradation. In this case, as derived in Appendix B, each sample at the output of the IFFT has the same average energy $\mathcal{E}_x$. Consequently, the

ν -sample cyclic prefix does not change the energy constraint.

3.5 Computational Requirements

In this section we describe the computational requirements to implement the DMT transceiver. The computational complexity includes only the functions that are unique to DMT; those functions that are common to both DMT and conventional single-carrier transceivers are not included.

The following are the assumptions in computing the total complexity:

- We assume that for certain applications, the DMT transceiver can be implemented using commercially available DSPs, and so the complexity is measured in terms of Million Instructions Per Second (MIPS).
- The sampling rate is equal to 640 kHz. We choose this number because as section 4.4.4 shows, further increase in sampling rate for HDSL application does not significantly increase the throughput, but it does increase the cost substantially. However, the sampling rate is by no means fixed at 640 kHz for all applications.
- A real addition, a real multiplication, or a real multiplication followed by a real addition (multiply-accumulate) is counted as one instruction; a complex multiplication is counted as four instructions.
- The number of required instructions to compute an N -point real-input FFT (or equivalently N -point real output IFFT) is assumed to be $1.5N \log_2 N$, where N is less than or equal to 128; otherwise, the number of instructions is equal to $2N \log_2 N$. (N is a power of 2.)
- Computations required in synchronization, timing recovery, decision block, and coding/decoding are not included, since these components are common to other schemes such as DFE and thus have no impact on our comparison study.

- The echo canceller, which is required for single-duplex and dual-duplex options, is discussed separately in [43] and has a comparable complexity to the echo canceller used in baseband system. We note that complexity of echo cancellation can be significantly reduced in either case using the methods of [44] or [45].

Thus, we focus only on the major transmit and receive functions that are significantly different from other schemes in estimating computational requirement. Computations for channel estimation, SNR calculation, and bit and energy allocation during the start-up protocol are not included, as they are done “off-line” and have no effect during actual data transmission. We note that an FFT peripheral for the DSP would be of great significance in further lowering the listed computational requirements to follow.

3.5.1 Transmitter Complexity

The major transmit function is the IFFT that is performed once per block. The computational requirement of the transmitter is summarized in Table 3.2 for a sampling rate of 640 kHz. The computation required for adding a cyclic prefix is trivial and is not included.

Complex-to-real IFFT Size (N)	64	128	256	512	1024
Complexity (in MIPS)	5.8	6.7	10.2	11.5	12.8

Table 3.2: Transmitter complexity at 640 kHz sampling rate

3.5.2 Receiver Complexity

In this section, we first examine the complexity of the original DMT receiver as shown in Figure 3.4. The complexity of the revised DMT receiver is then estimated and compared with the old DMT receiver.

For a block of $N + \nu$ received data, as shown in Figure 3.4, $(M + 1)N$ instructions are required to obtain the output z_k , N instructions are needed for computing the

error vector, $(M+1)(N+1)$ instructions are necessary for updating the weights of the adaptive equalizer $w(D)$, and $8N$ instructions are required for multiplying $DFT(b)$ and $1/DFT(b)$. In addition, one N -point real input FFT and one N -point real output IFFT are required. The total number of instructions per block is given by

$$\text{Total Complexity per Block} = (M+1) + (4 \log_2 N + 2M + 11)N. \quad (3.69)$$

Table 3.3 shows the complexity of the receiver with a sampling rate of 640 kHz.

Complexity (in MIPS)	M=4	M=9	M=15	M=31	M=63
N=64	23.7	30.2	37.9	58.6	99.8
N=128	25.6	32.1	39.8	60.3	101.4
N=256	32.7	39.1	46.8	67.3	108.3
N=512	35.2	41.6	49.3	69.8	110.8
N=1024	37.8	44.2	51.9	72.3	113.3

Table 3.3: (Old) Receiver complexity with full block LMS

To reduce the computational requirement for this receiver, a Block LMS with efficient updating is presented [36]. Instead of using all the N error data in the summation to update the weights for each block (see equation (3.54)), we use only $M+1$ out of N for each block. With this new updating scheme, only $M+1$ instructions are needed to obtain the errors, and only $(M+1)^2$ instructions are required for updating. The total number of instructions per block is given by

$$\text{Total Complexity per Block} = (M+1)(M+2) + (4 \log_2 N + M + 9)N. \quad (3.70)$$

The resulting complexity is shown in Table 3.4.

It is easy to show that this algorithm, similar to the regular Block LMS, converges to minimum mean-square error with a slower rate of convergence. Note that there is only a slight increase (or even decreases as for $M = 63$) in complexity when we increase the blocklength, because as we fix M and increase N , the reduction of

Complexity (in MIPS)	M=4	M=9	M=15	M=31	M=63
N=64	20.1	24.1	29.6	47.7	99.2
N=128	21.9	25.5	30.2	44.3	80.3
N=256	28.9	32.3	36.5	48.7	77.0
N=512	31.4	34.7	38.7	50.0	74.3
N=1024	33.9	37.2	41.1	51.9	74.3

Table 3.4: (Old) Receiver complexity with efficient updating block LMS

complexity for updating the weight offsets the increase of complexity for computing a longer FFT/IFFT.

Figure 3.5 shows the revised DMT receiver. For a block of $N + \nu$ received data, $(M + 1)N$ instructions are required for computing the outputs $z_0, \dots, z_{N-1}$, and $2N \log_2 N$ instructions are needed for the N -point real input FFT. Recall that (3.58) - (3.60) are computed only through $n = N/2$, and so only $2N$ instructions are required to compute v_n , (3.58), N instructions for obtaining the error samples, and $3N$ instructions for updating the weights. The computation required per block is

$$\text{Total Complexity per Block} = (2 \log_2 N + M + 7)N. \quad (3.71)$$

The complexity of the receiver is shown in Table 3.5 with these stated assumptions.

Complexity (in MIPS)	M=4	M=9	M=15	M=31	M=63
N=64	12.8	16.0	19.8	30.1	50.6
N=128	13.8	17.0	20.8	31.0	51.5
N=256	17.3	19.2	24.3	34.6	55.0
N=512	18.6	21.8	25.6	35.8	56.3
N=1024	19.8	23.0	26.9	37.1	57.6

Table 3.5: (Revised) Receiver complexity at 640 kHz

As shown in Section 4.4, for many applications, near optimum performance can be achieved with a feedforward equalizer of about 10 taps, and a blocklength on the order of $N = 512$ would be sufficient; so the total computational power under the

stated assumptions for the revised receiver is about 22 MIPS, which is fairly modest by today's standard. We also observe that the old receiver with efficient updating requires about 35 MIPS, and that means over 50% increase in complexity. This is the main reason why we propose using Figure 3.5 as a preferred DMT receiver.

3.6 Summary

In this Chapter we derived a multi-carrier modulation structure based on general theory of input-aided equalization. This multi-carrier modulation system, which we called discrete multi-tone (DMT) transceiver, is shown to be optimum with infinite complexity. Equally important, a finite length optimization algorithm is derived so that near-optimum performance can be achieved with any given complexity constraint.

The structure of the DMT transceiver is described in detail. The main added features as compared to the DFT-based system described in Chapter 2 are the time-domain feedforward equalizer, which effectively shortens the cyclic prefix without affecting performance, and the set of 1-tap LMS filters to account for any channel mis-estimation and variation.

The performance analysis of the DMT system is described, and can be conceived as a set of QAM channels operating on a set of ISI-free passband channels, partitioning the channel of interest. Transmit power level and system initialization are also considered. Finally, we estimated the complexity of the DMT system with a target subscriber line application and with the assumption that the DMT system is implemented in commercially available digital signal processors. We concluded that the total complexity (without coding) is about 35 MIPS for a sampling rate of 640 kHz.

Chapter 4

Applications of the DMT System

With a careful allocation of bits and transmit power to the subchannels, it is well-known [9], [2] that a multi-carrier system is capable of performing at the highest theoretical limits and that no other system can exceed its performance. There are now at least 3 commercial 19.2 kbps voiceband modems, marketed by 3 different companies, that use multi-carrier systems to attain excellent performance. This chapter describes the feasibility of applying the DMT system to HDSL application.

The excellent high-performance/cost trade-off of multi-tone modulation makes it a strong candidate for the transceiver implementation for High-bit-rate Digital Subscriber Lines (HDSL), a 1.6 Mbps digital subscriber service on twisted-pair channels of up to two miles. Chapter 3 presented a multi-tone modulation structure that is of particularly high-performance and low-cost for the HDSL application. We generally find that in this Chapter even on worst-case channels, a 1.6 Mbps (1.536 Mbps user data plus 64 kbps control channel) data rate is achievable at an error rate of 10^{-7} on a single twisted pair (with 6 dB margin). Furthermore, we find that the described multi-tone system exhibits excellent margins (nearly 18 dB) for the so-called dual-duplex option of providing 1.6 Mbps on two coordinated twisted pairs.¹

The intersymbol interference, crosstalk, and attenuation of 24-gauge and/or 26-gauge twisted pair is particularly severe within the two-mile Carrier Serving Area

¹We note that results in this dissertation use channel models with less attenuation than those of previous studies in [36], [39], [46], [47], based on measured line results of several industrial sources.

(CSA) requirements, as we discuss in Section 4.1. Section 4.1 also briefly examines and reviews theoretically and practically achievable maximum data rates for HDSL. Equalization and coding methods have to be carefully applied in the HDSL application to render the high transported data rates reliable. In the presented approach, a particularly effective combination of a small amount of time-domain equalization and of a larger amount of multi-tone modulation is used to transmit the data reliably. The equalizer is used to shorten the effective response length of the data channel, while the multi-tone modulation is used to mitigate the remaining intersymbol interference and the effects of crosstalk, as is described in Section 3.4. The DMT system is fully adaptive, employing both an adaptive algorithm for update of the adaptive filters and also a special channel identification procedure that is used to initialize and update the transmitter bit allocation to frequency-indexed subchannels. This relatively simple procedure is essential for maximizing performance. We also estimated the complexity in terms of number of arithmetic operations in Section 3.5 and illustrated that the DMT transceiver can be implemented with programmable signal processors, although those processors may need to be custom designed for lowest cost.

The performance in achievable data rate of the DMT system is analyzed and characterized in Section 4.4 as a function of input power, coding, channel characteristic, number of tones or carriers (block length), sampling rate, noise margins, and HDSL occupancy.

4.1 Subscriber Loop Characterization and Modeling

In any study of HDSL transceivers, the channel models are critical to specification of performance. We outline in detail the HDSL channel models that we use throughout the study of DMT HDSL transceivers in this Chapter. We note that performance projections based on different channel models can lead to erroneous claims and comparisons. A detailed tutorial on HDSL technologies is presented by Lechleider in [48].

4.1.1 CSA Loop Characteristics

The Carrier Serving Area (CSA) is an identifiable subset of the current subscriber loop population in the United States, and our HDSL study is restricted to loops that satisfy minimal CSA requirements. The CSA consists of mainly 24- and 26-gauge twisted-pair, with lengths on 24-gauge wire to 12 kft and on 26-gauge wire to 9 kft. CSA loops can also contain bridge taps of limited length; a complete specification of the CSA appears in [49]. Impairments in the CSA data-transmission channel include:

- intersymbol interference (ISI),
- crosstalk noise coupled from adjacent loops within the same cable bundle,
- impulse noise (caused by switching transients, lightning and other electrical machinery),
- echo noise resulting from the combination of imperfect hybrids and gauge changes on the line as well as signals reflected from bridged taps,
- electronics noise (such as quantization noise in analog-to-digital converters (ADC) and thermal noise in the analog portion of the transmitter and receiver), and
- inductive noise at 60 Hz and its harmonics caused by nearby power lines.

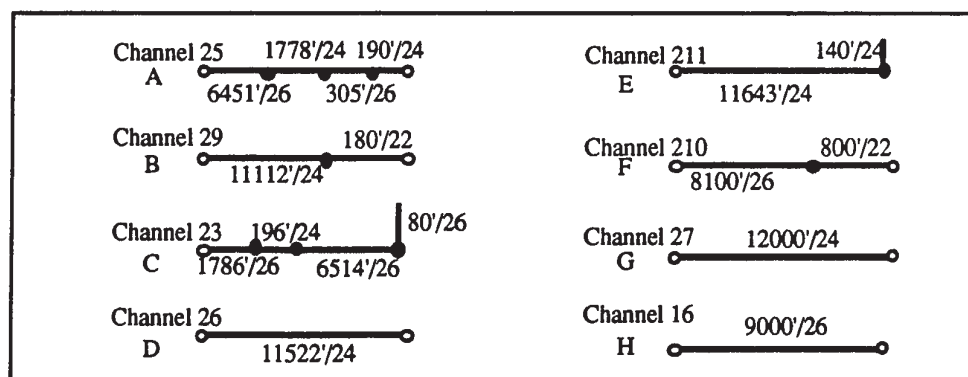


Figure 4.1: CSA Channels : length (in feet) / gauge

We consider a set of representative CSA channels provided by Bell Communications Research [50] that have the respective loop cable arrangements illustrated in Figure 4.1. We use a letter designation (A-H) on these channels for reference. These channels are generally representative of some of the worst, in terms of channel impulse response, that are expected to be found within the CSA limits. In general, longer wire length, smaller wire size, and presence of more and longer bridge taps results in poorer achievable data rates. We also note that on-premises wiring is often impossible to characterize statistically, so that a 6 dB margin is imposed upon the performance of any HDSL transceiver to ensure that 10^{-7} error rate is maintained [51]. In all our loop models in this paper, we include this 6 dB margin simply by increasing loop attenuation for the loops in Figure 4.1 by 6 dB (see also footnote 1 in this Chapter).

4.1.2 Channel Modeling

A line simulation program based on standard transmission line modeling and a standard set of twisted-pair characteristics was used to generate an impulse response for each of the channels in Figure 4.1. The effect of the coupling transformer at each end of the loop termination was included by adding a double-pole double-zero high pass digital filter with a cutoff frequency of 300 Hz. These characteristics were found to be in near exact agreement with those provided by separate such programs at various industrial groups working in the HDSL field, [52], [50]. One such time-domain impulse response is shown in Figure 4.2 for Channel H.

Crosstalk noise coupled from adjacent wire pairs in the cable bundle is the dominant noise impairment in CSA loops. Various characterizations have been proposed [53], [54], and we will use [53], which models crosstalk noise as near-end crosstalk (NEXT) with the following transfer function:

$$|H_{\text{NEXT}}(f)|^2 = K_{\text{NEXT}} f^{\frac{3}{2}}, \quad (4.1)$$

where K_{NEXT} is the coupling coefficient, and f is frequency in Hz; and far-end crosstalk (FEXT):

$$|H_{\text{FEXT}}(f, l)|^2 = K_{\text{FEXT}} f^2 |H(f)|^2 l, \quad (4.2)$$

where K_{FEXT} is the coupling coefficient, $|H(f)|^2$ is the channel power gain, and l is the length of the wire in kft. Since far-end crosstalk noise is attenuated by the channel, the resulting noise power is negligible compared to near-end crosstalk, hence we will only consider NEXT. The coupling coefficient K_{NEXT} for a 50-pair cable was empirically determined to be approximately 10^{-13} , and K_{FEXT} to be approximately 10^{-16} [55].

To model the effects of residual echo after cancellation, we note that a typical CSA loop attenuation is on the order of 24 dB, and a good echo canceller presumably can reduce the echo to a level of 30-40 dB below the received signal. Assuming an average input power of 10 mW (10 dBm), this translates into -44 to -54 dBm of total echo noise power across the two-sided bandwidth of 640 kHz, which corresponds to a noise power spectral density of 6.22×10^{-11} to 6.22×10^{-12} mW/Hz. This residual echo noise is assumed to be Additive, White, Gaussian Noise (AWGN), and we fix this AWGN to have a power spectral density of -110 dBm/Hz for subsequent analysis and simulation. The electronics noise as mentioned earlier is typically on the order of -140 to -170 dBm/Hz [46] and is thus, by comparison to echo noise, negligible. Of course, crosstalk noise is larger than either echo noise or electronics noise.

While impulse noise is more difficult to characterize, some [56] have attempted to model its effect. Impulse noise usually consists of millivolt pulses (say up to 40 mV) that last a few μs to 1000 μs .

Inductive noise can be modeled as a fixed noise level at 60 Hz, but the high-pass filter simulating the transformers eliminates the inductive noise.

Pole-Zero Modeling

To facilitate simulation of CSA loops for performance analysis, we constructed a pole-zero (IIR) model for each channel impulse response at various sampling rates, using standard system identification tools. An impulse response obtained for Channel H is compared to the impulse response generated by a pole-zero model in Figure 4.2. Since the pole-zero model closely approximates the true response, the use of pole-zero models for subsequent studies is justified.

The number of poles and zeros varies with different CSA channels, but they all fall into the same range: 5 to 8 zeros, and 6 to 12 poles. We also found that most of

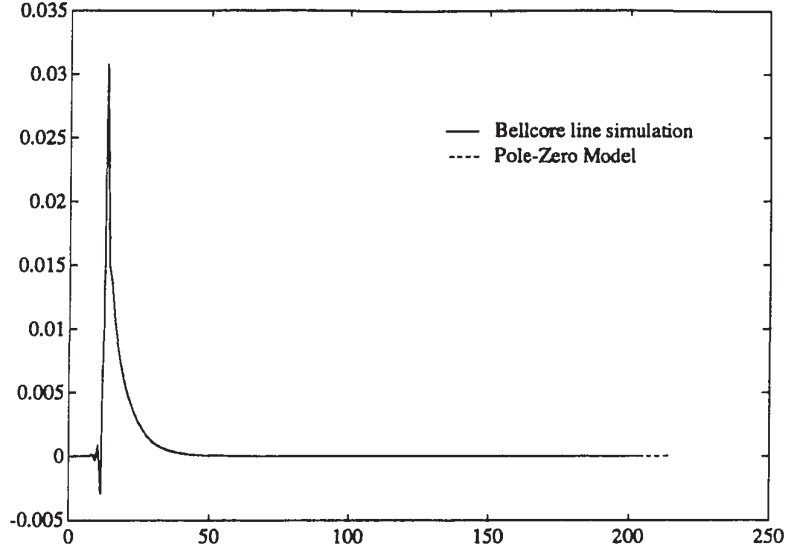


Figure 4.2: Impulse response of a CSA loop

the CSA channels are not minimum phase, i.e., some of the zeros from the pole-zero models lie outside the unit-circle in the z -plane.

4.1.3 Capacity

The capacity of these loops with the aforementioned models is computed in [46], at the sampling rates of 400 and 800 kHz. As explained later in Section 4.4.4 and in [46], a sampling rate of 400 kHz is not sufficient to maximize the HDSL transceiver performance on most CSA channels. Table 4.1 shows the capacity of the set of CSA channels as shown in Figure 4.1, as well as the capacity with a flat input spectrum (called “flat-input capacity” in Table 4.1) at 800 kHz sampling rate. The two columns under “Capacity” list the capacity using the optimum ([46]) transmit power distribution, and represent the best data rates that can be achieved with an 800 kHz sampling rate. “Crosstalk Bound” assumes the crosstalk noise is the only noise impairment (and thus ignores the contribution of AWGN), while “11 dBm Power” includes the

Channel	Capacity with 6 dB Margin		Flat-Input Capacity with 6 dB Margin		
	Crosstalk Bound	11 dBm Power	0 dBm Power	10 dBm Power	17 dBm Power
A	2.42	2.30	1.87	2.29	2.40
B	2.39	2.27	1.85	2.27	2.37
C	2.32	2.20	1.78	2.19	2.30
D	2.31	2.19	1.77	2.19	2.29
E	2.26	2.14	1.72	2.13	2.24
F	2.29	2.17	1.75	2.16	2.27
G	2.19	2.07	1.65	2.06	2.16
H	2.14	2.02	1.60	2.01	2.11

Table 4.1: Capacity and flat-input capacity for CSA loops, 800 kHz rate and with 6 dB margin (in Mbps)

effect of AWGN, with 11 dBm transmit input power. The three columns under “Flat-Input Capacity” do not optimize transmit power distribution; instead, a flat input power spectral density is assumed. The effects of near-end crosstalk, 6 dB margin, and AWGN are included for the calculation, and the maximum achievable data rate based on the flat input power spectrum are shown for 3 different input powers: 1, 10, and 50 mW (0, 10, and 17 dBm, respectively). Notice that with a sufficiently high input power (10 dBm or above), the penalty in achievable data rate without optimizing the transmit power distribution becomes negligible. Thus, we will assume a flat input power spectrum is used, and this will simplify analysis and simulation. Also, this allays any practical concern about loop-to-loop energy-spectrum variation.

We can infer from this table that 1.6 Mbps can be reliably transmitted over most CSA channels. However, an excellent transceiver design is required to accomplish this goal on either a single pair, or on two coordinated pairs with a large performance margin such as 12 dB. Even with only a 6 dB performance margin on two pairs, 800 kbps is 40-50% of capacity and more than a trivial receiver is required. On the other hand, a data rate of 160 kbps on CSA channels (as in the case of basic rate ISDN) is well below fundamental limits on these channels (although it may be a little more difficult on longer (18 kft) channels). Thus, one would not expect that a simple receiver used for basic rate ISDN would necessarily be good for use in HDSL.

4.2 DMT System for HDSL

In practice with HDSL, the dominant noise is crosstalk noise, which is not white, so that $\sigma^2 \rightarrow \sigma_n^2$, a subchannel dependent noise energy. When N is chosen sufficiently large, the noise on each of the subchannels will be approximately white within the occupied frequency band and will be uncorrelated with the noise on any of the other subchannels and the noise. In order to compute the bit allocation outlined by equation (3.64), the channel characteristic must be known at the transmitter, which is achieved as described in Section 3.4.3. The energy allocation is approximated by a flat transmit energy spectrum, so that $\mathcal{E}_x$ is not a function of the frequency index n . Our studies have found that very little is lost in using this flat energy distribution in the HDSL application. The poorest subchannels carry only dummy data and are simply ignored by the receiver. This flat transmit spectrum has the effect of rendering the crosstalk power spectrum from any DMT HDSL transmitter constant in practice.

4.2.1 The DMT HDSL Transmitter

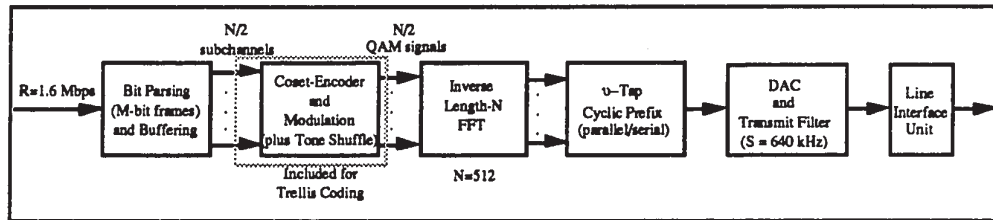


Figure 4.3: Block diagram of a specific DMT HDSL transmitter

A specific embodiment of the HDSL transmitter is illustrated in block diagram form in Figure 4.3. In the specific implementation of the system that we study, ν is always limited (for ease of hardware implementation purpose) to a maximum of 8, and later in Section 4.4.3 and 4.4.4, we fix the sampling rate S to be equal to 640 kHz and $N = 512$.

For a single twisted pair to carry $R = 1.6$ Mbps (the **single-duplex** option), the input data bit stream is parsed into M -bit blocks, where $M = (N + \nu)R/S = 1300$. These bits are then transformed into (a maximum of) $N/2 = 256$ QAM subsymbols

that are then applied to a trellis encoder. The trellis coder sequentially processes the frequency-indexed subsymbols to avoid the large latency and memory requirement that would occur with multiple trellis encoders. This process requires that the subchannel indices be “shuffled” to avoid any minor correlation between the noise on adjacent-frequency subchannels. The output of the trellis encoder is then modulated, as described earlier, through the use of an $N = 512$ inverse FFT. Finally, an 8-sample cyclic prefix is placed in the beginning of the corresponding block of modulated transmit samples, and the extended block of $N + \nu$ samples is then applied to the channel through the digital-to-analog converter (DAC) and line-interface unit. The sampling rate is 640 kHz, which leaves the 256 subchannels effectively separated by 1.25 kHz. This corresponds to a **block symbol period** of $512 + 8 = 520$ samples or $812.5\mu\text{s}$. 8 samples is considerably shorter than most HDSL channel impulse responses at 640 kHz. An unusual and simple equalizer is used in the receiver to contain intersymbol interference to 8 samples, or impulse-response length to 9 samples.

For the so-called **dual-duplex** option with two twisted pairs, the 1300-bit blocks are parsed into the 512 QAM subchannels that exist on both of the channels. There are then two IFFT's, two DACs, and two line-interface units. However, only one trellis code is necessary with this option with respect to the single-duplex option. This improves performance margin but increases cost of the dual-duplex option. If sufficient margin can be obtained using the single pair solution (called “single-duplex”), then it is preferable from a cost standpoint.

4.2.2 The DMT HDSL Receiver

The HDSL receiver is identical to Figure 3.5, with the specific sampling rate of 640 kHz, a block length of $N = 512$; the corresponding HDSL transmitter with 1300-bit frames has a data rate of $R = 1.6\text{Mbps}$.

4.3 Derivation of Performance Evaluation

In our analysis, we include ISI, AWGN, and near-end crosstalk noise as impairments for the channels. We also separately consider impulse noise later in this chapter. As described in Section 4.1.2, the near-end crosstalk has the transfer function $10^{-13} f^{3/2}$, and the AWGN is assumed to be at -110 dBm/Hz. We also note that the maximum achievable data rate is very close to capacity when using a flat transmit power spectrum as long as adequate input power is used, on all the CSA loops of interest (see Table 4.1). Thus, we use a flat power spectrum at the channel input, over the utilized transmit bandwidth. We assume a mix of 25% HDSL service, 25% Basic Access, and 50% voice-grade service in a typical cable bundle. When 25% HDSL occupancy is exceeded, telephone company managers are expected to replace the bundle with fiber. The crosstalk noise power spectral density, PSD_{NEXT} , based on the assumption that all users who subscribe the HDSL service use the same signaling strategy and transmit power, is then equal to

$$\text{PSD}_{\text{NEXT}} = \begin{cases} \text{PSD}_{\text{in}} 10^{-13} f^{3/2}, & f \leq 50 \text{ kHz} \\ \frac{\text{PSD}_{\text{in}}}{4} 10^{-13} f^{3/2}, & f > 50 \text{ kHz} \end{cases} \quad (4.3)$$

where PSD_{in} is the input signal power spectral density, and a 50 kHz cutoff frequency is assumed for services other than HDSL. For the DMT system this occupancy assumption is also justified by the fact that a length 512 DMT system has considerably enhanced immunity to impulsive noise than symbol-by-symbol modulation approaches, as discussed later in this chapter. Thus, an artificially high crosstalk margin or gain is redundant in the DMT system, so we use only 6 dB to cover the case of unknown on-premises wiring. Signal, crosstalk noise, and AWGN are processed by the equalizer, which has a fixed taps setting obtained from the finite length algorithm described in Section 3.3.

The signal-to-noise ratio (SNR) for each QAM subchannel n is obtained, as in Section 3.4.2

$$\text{SNR}_n = \frac{\mathcal{E}_n |H_n|^2}{\sigma_n^2}, \quad (4.4)$$

where σ_n^2 is now the total noise energy, including crosstalk and AWGN, of the n^{th} subchannel.

We could calculate the number of points assigned to each QAM subsystem by the following equation, as was done in [39]:

$$\text{Number of points for subchannel } n \triangleq L_n = \frac{3SNR_n}{[Q^{-1}(P_b)]^2} + 1 . \quad (4.5)$$

Equation (4.4) is valid as long as the noise is additive Gaussian noise [57].

However, equation (2.15) also computes the number of points for each QAM subchannel. In addition, it also includes the effect of coding and margin, and is a more general and preferred equation

$$L_n = 1 + \frac{SNR_n}{\Gamma}, \quad (4.6)$$

where Γ is given by

$$\Gamma \text{ (dB)} = [Q^{-1}(P_b)]^2 + \gamma_m - \gamma_c - 4.77 \text{ (dB)} \quad (4.7)$$

$$= 9.8 + \gamma_m - \gamma_c \text{ (dB)}, \quad (4.8)$$

for a bit error rate of 10^{-7} .

We also implicitly assume the number of nearest neighbor is equal to 1, and coding gains of any applied codes with higher nearest neighbor counts can be correspondingly reduced by an appropriate factor. The number of bits, b_n , assigned to subchannel n is then

$$b_n = \log_2 L_n \quad (4.9)$$

$$= \log_2 \left(1 + \frac{SNR_n}{\Gamma} \right) . \quad (4.10)$$

In general, b_n is not an integer; we can either truncate it to an integer, or (in the coded case) we can round it to the nearest fraction (such as 3.25 bits per 2D symbol) that can be implemented by multi-dimensional trellis codes [58], [39], [33], [34]. The total data rate, R , (in bits/sec) is obtained:

$$R = \frac{S}{N} \sum_n b_n , \quad (4.11)$$

where S is the sampling rate.

Since the length of the cyclic prefix, ν , is a design parameter as described in Section 3.3, we will neglect the rate loss caused by the cyclic prefix to compute the performance results in Section 4.4.

4.4 Performance Results for Multi-Carrier Systems

In this section, we will analyze the performance of the DMT transceiver system. Various parameters that affect the performance are examined: coding, average input power, block length², sampling rate, noise margins, and HDSL occupancy. We then focus on the tradeoff between performance and cost. Other techniques that could achieve similar performance are compared.

The performance of a representative set of CSA loops, as shown in Figure 4.1, is evaluated analytically with different system parameters. Without coding and with a sampling rate of 640 kHz, the achievable data rates as a function of average input power are shown in Figure 4.4. Use of small gauge (26g) wire and the existence of bridge taps severely limit the data rates of this set of “lossy” end of the local loops. Channel H, which consists of 9000 feet of 26-gauge wire, has the poorest performance among the set of channels, and we will focus on this channel for the remainder of this section, since the performance results of Channel H can serve as a “lower bound” for the other channels considered.

4.4.1 Coding

As shown in Figure 4.4, only some of the channels studied achieve the objective 1.6 Mbps data rate with 6 dB margin without a code, using one twisted pair. With two twisted pairs however, the T1 rate can be achieved without code. Since the achievable data rate is approximately doubled when using two twisted pairs instead of a single

²Throughout this dissertation, block length refers to the size of the data block before adding the cyclic prefix, i.e., block length = N .

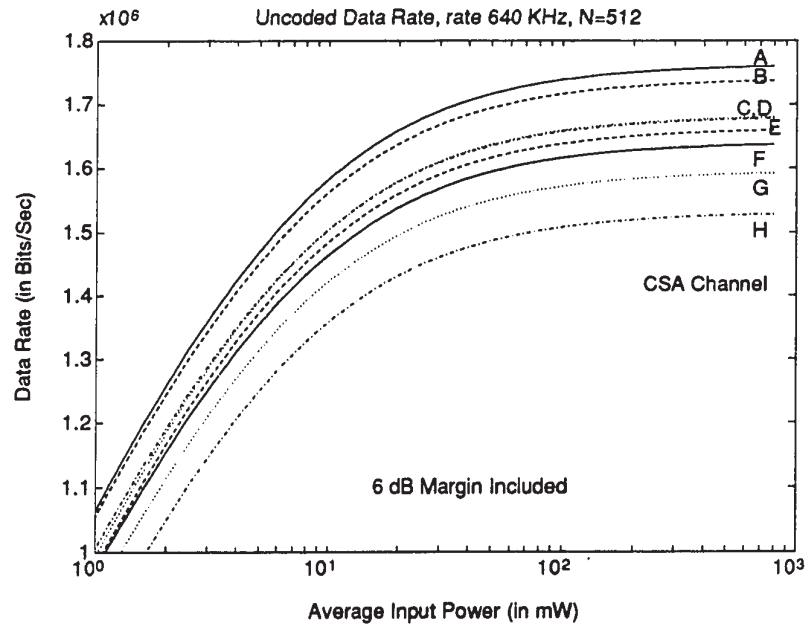


Figure 4.4: Uncoded data rate for CSA channels (6 dB margin included)

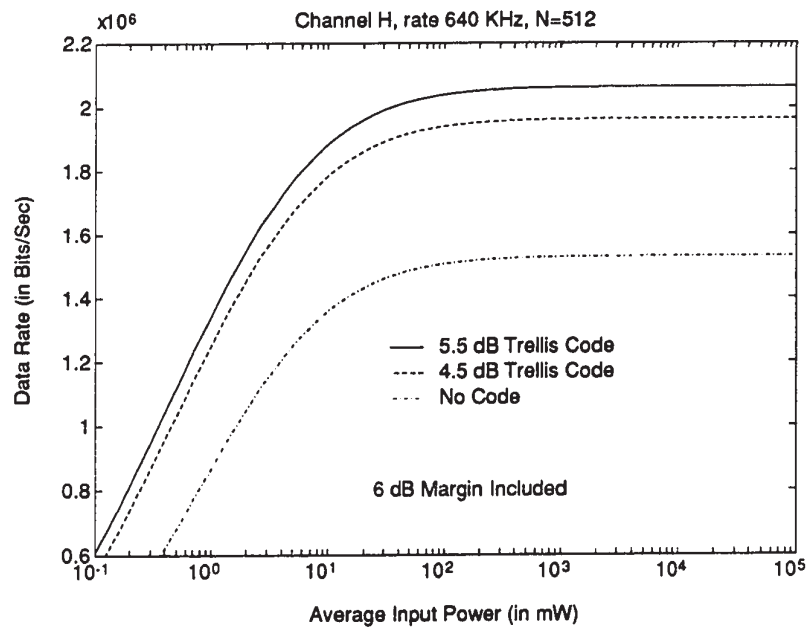


Figure 4.5: Effect of coding and average input power on data rate (6 dB margin included)

wire-pair, we will from here on focus on the single twisted-pair case. However, when considering the dual-duplex twisted pair case, one only needs to look at performance in our plots at 800 kbps instead of 1.6 Mbps. A trellis code can be used to improve the data rate. We consider two codes: a 4-dimensional 16-state 4.5 dB gain trellis code and a more powerful 4-dimensional 5.5 dB gain trellis code, which presumes use of some good shaping code, as for instance described in [59]. We assume the loss due to the increase of the number of nearest neighbors is offset by shaping gains [33], so that an effective gain of 4.5 and 5.5 dB respectively are assumed for the two codes. At lower bits per subchannel, the four-dimensional code has a higher minimum distance when there are no parallel transitions in the trellis describing the code. Thus, while it is readily possible to achieve shaping gain at lower bits per subchannel, the coding gain of the original code is higher and compensates for the lack of shaping gain in the common four-dimensional code. The effect of concatenating a trellis code to the DMT system on Channel H is shown in Figure 4.5 using a block length of 512 and a sampling rate of 640 kHz. With the powerful 5.5 dB trellis code, this 9000 feet 26 gauge channel can achieve the T1 rate with significant margin.

As described earlier, different number of bits are assigned to the subchannels according to the SNR of each subchannel. However, the number of bits assigned to any subchannel cannot be arbitrarily large, as an unrealistically large signal point constellation would have to be used. We therefore propose that a maximum of 10 information bits should be used in any subchannel. Figure 4.6 shows the information bit distribution of Channel H, with a 5.5 dB code. The achievable data rate decreases by less than 1% when we apply the limit of 10 information bits/subchannel. Even for the best channel, (Channel A, see Figure 4.7) the loss in throughput is only slightly above 1%. We thus conclude that the bit assignment can be implemented practically.

4.4.2 Average Input Power

The achievable data rate of Channel H is plotted against average input power in Figure 4.5. We observe while the data rate generally improves with increasing input power, the improvement is significant in the very low input power region and becomes minimal at very high input power. The reason is that we assume all users of the CSA

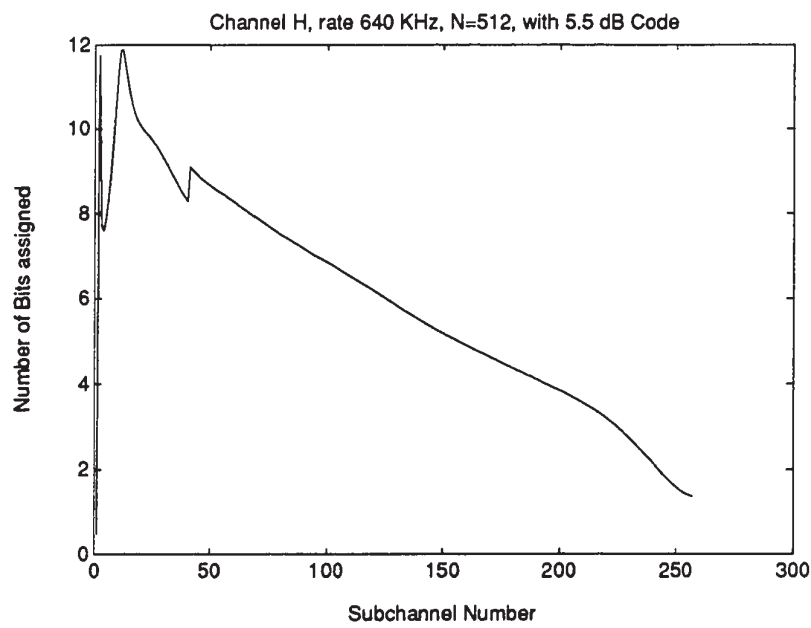


Figure 4.6: Bit distribution of Channel H with 5.5 dB code

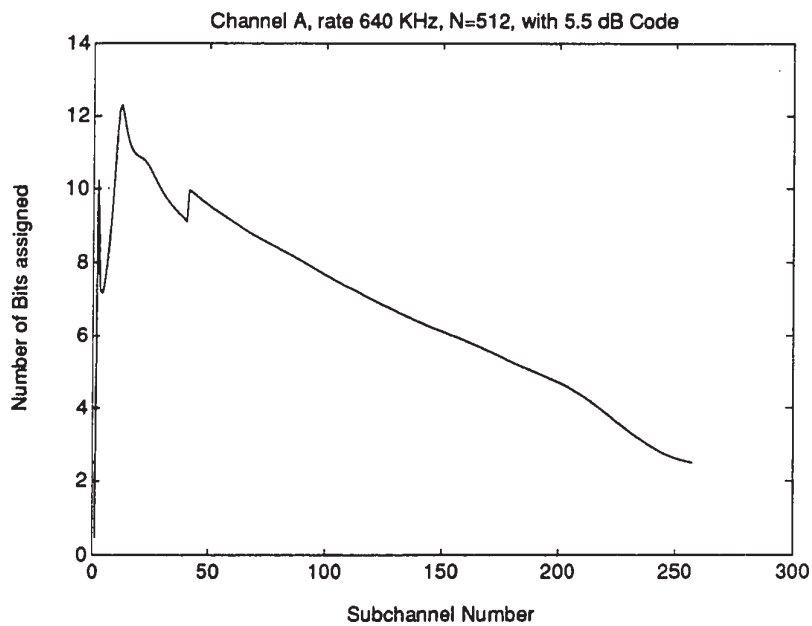


Figure 4.7: Bit distribution of Channel A with 5.5 dB code

loops within the same 50-wire bundle use the same signaling strategy and transmitter power (assuming the same earlier HDSL occupancy in (4.2)). As a result, the crosstalk noise that results from interfering neighbors that employ the same scheme is directly proportional to the input power used. With very low input power, the resulting crosstalk noise is so small that the dominating impairment becomes the AWGN, and within this region an increase of input power directly improves the SNR and thus the data rate improves noticeably. However, as input power increases further, the crosstalk noise becomes comparable and finally dominates the AWGN, and over this crosstalk dominating region, increase in input power causes a corresponding increase in the crosstalk noise. As a result, there is virtually no net gain in achievable data rate in the very high input power region. An average power in the range of 10 to 50 mW is sufficient to transmit high data rates reliably, and physical limitations favor the use of lower input power. We thus decide on using 10 mW as the average input power.

4.4.3 Block Length

We expect a higher data rate as longer block length is used, as shown in Figure 4.8 for Channel H with 10 mW of input power and a 640 kHz sampling rate. However, the higher complexity, increased memory requirements, and inherently longer delay required for long block lengths make it unattractive for practical systems. We conclude that a block length of 512 is sufficient to provide enough frequency resolution, while simultaneously maintaining reasonable cost and delay. We note that even block lengths as low as 128 and 256 (plus cyclic prefix) will also achieve near optimum performance.

4.4.4 Sampling Rate

Sampling rate has a profound impact on the maximum achievable data rate. A low sampling rate forces a large number of bits to be squeezed into a limited bandwidth, and this can significantly lower the performance.

As an example, the total number of bits per block (assume using the same block

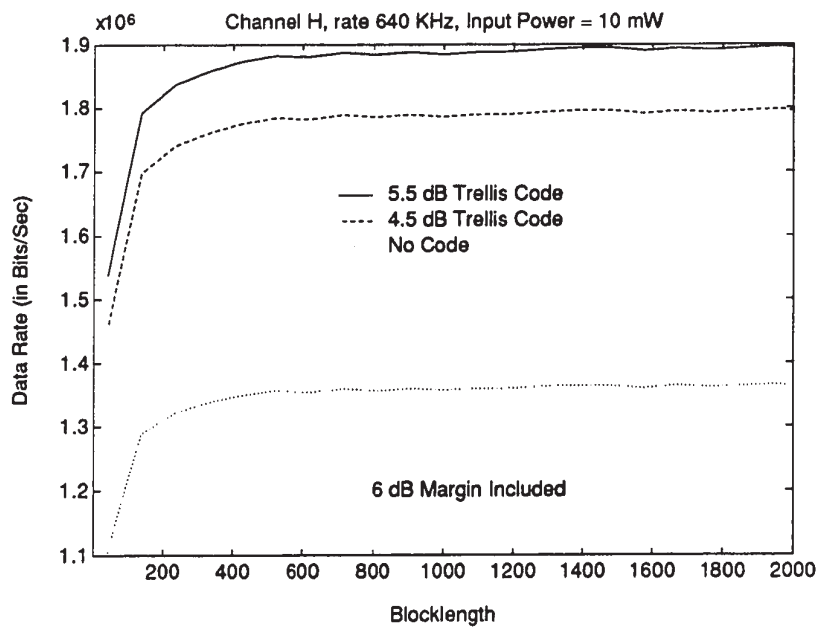


Figure 4.8: Effect of different blocklength on data rate (6 dB margin included)

length) has to double in order to maintain the same data rate if we halve the sampling rate, but to double the total number of bits would require an exponential increase in signal-to-noise ratio (SNR) in order to keep the same bit error rate. Although the crosstalk noise is less with a lower sampling rate because of the $f^{3/2}$ frequency dependency, the improvement in SNR is not sufficient to secure the bit error rates. Since we fix $\text{BER} = 10^{-7}$ as a criterion, this implies that we won't be able to keep the data rate as desired.

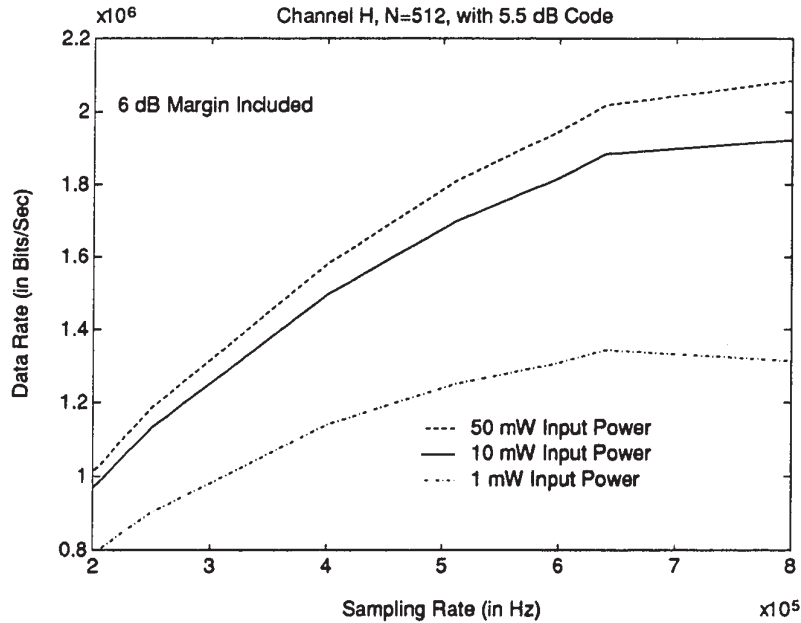


Figure 4.9: Effect of sampling rate on data rate (6 dB margin included)

This phenomenon is clearly depicted in Figure 4.9, where the throughput of Channel H is plotted against different sampling rates with a 5.5 dB trellis code. This figure is generated with a set of pole-zero models derived for Channel H at the following sampling rates: 200, 250, 400, 512, 600, 640, and 800 kHz. We observe that by going from 400 kHz down to 200 kHz, for example, the data rate decreases by more than 30%, a clear indication that the sampling rate was too low. On the other hand, as

sampling rate increases, the crosstalk noise becomes more severe at the higher frequency indices and eventually degrades the SNR to a point that it may no longer be practical to assign any bits to those frequency subchannels. The improvements in terms of achievable data rate as we increase the sampling rate further becomes negligible since the added bandwidth contributes little to the system while the cost of implementing the system keeps on increasing. As a reasonable compromise between performance and cost, we choose 640 kHz as the sampling rate for the DMT system. We note that DMT is not sensitive (at practical transmit power for HDSL) to picking the sampling rate too high, and that 640 kHz is sufficiently high. On the other hand, decision feedback approaches (in particular those based on baseband PAM (2B1Q), [37] or on carrier-less QAM [38]) are sensitive to optimum choice of the sampling rate (see [27]) and that sampling too high or too low is not optimum with these approaches. In practice, voiceband modems that use decision feedback equalizers do vary the sampling rate. However, variation of sampling rate in HDSL is impractical, yielding a significant advantage of DMT with respect to decision-feedback approaches.

4.4.5 Noise Margins

As mentioned in Section 4.1.2, we do not attempt to model all possible impairments that may be presented in the system: impairments such as non-linearity and impulse noise are not included (so far). In addition, the models that are included for the impairments may not always predict the noise accurately. As Bellcore suggested [49], some kind of margin is desirable in order to ensure that the actual performance will not degrade below an acceptable level. We have thus included a 6 dB margin throughout Section 4.3. In this section, we examine the noise margin further by increasing the crosstalk coupling coefficient K_{NEXT} by a factor of 2 for an additional 3 dB noise margin, a factor of 4 for an additional 6 dB noise margin, and a factor of 16 for an additional 12 dB noise margin. The achievable data rate based on this new crosstalk coupling coefficients are calculated and Figure 4.10 shows the results for Channel H with a 5.5 dB code. The throughput drops to about 1.4 Mbps with a 12 dB margin and down to about 1 Mbps with a 18 dB margin, but it is still possible to achieve 1.6 Mbps with a 9 dB margin. Other CSA channels result in

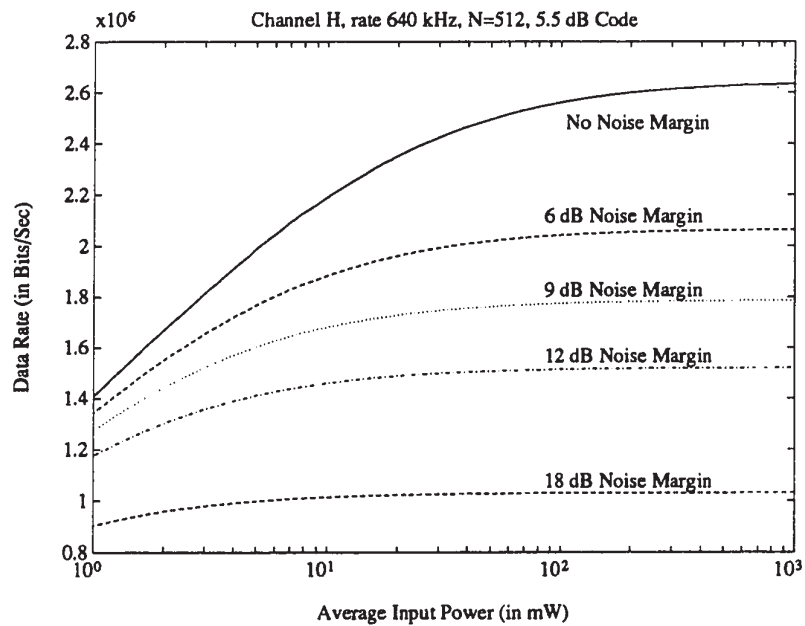


Figure 4.10: Effect of noise margins on data rate

similar performance degradation but at higher data rates. Overall, 99% of the CSA loops have a theoretical margin of approximately 9 dB for a single wire-pair, and over 18 dB margin with two coordinated twisted pair (dual-duplex).

4.4.6 HDSL Occupancy

Throughout the analysis we have assumed the cable bundle has 25% HDSL occupancy, a realistic assumption for the next few years. We now investigate the worst case scenario: 100% HDSL occupancy, where the near-end crosstalk is most severe. We model this effect by modifying Equation (4.2) to

$$\text{PSD}_{\text{NEXT}} = \text{PSD}_{\text{in}} 10^{-13} f^{3/2} . \quad (4.12)$$

The resulting throughput is shown in Figure 4.11 for Channel H. The achievable data rate drops down to about 1.5 Mbps with a 6 dB noise margin. This situation is not expected to occur in practice, as a bundle with high HDSL occupancy would be replaced by fiber for telephone company cost reasons.

4.4.7 Impulse Noise

Discrete multi-tone modulation has an inherent tolerance to impulse noise, especially when the duration of the impulse is less than one block length in sample periods. The immunity to a very short burst (only one sample affected) is easily computed to be

$$\text{Impulse Margin} = 10 * \log_{10}(N) \text{ dB} , \quad (4.13)$$

with respect to one-dimensional modulation (like 2B1Q). The margin is $10 * \log_{10}(N/2)$ with respect to any QAM approach. This means that the variance of the impulse noise sample must exceed the crosstalk noise level nominally present by an amount that makes the total noise sample variance $10 \log_{10}(N)$ larger. For $N = 512$, these margins are 27 dB and 24 dB, respectively, far in excess of the 12 dB normally quoted. In practice, impulse noise can span many sampling periods, especially at the high sampling rates used in HDSL. A long impulse might span as many as 40 sampling periods at 640 kHz. Some particularly bad impulses could exceed one block length.

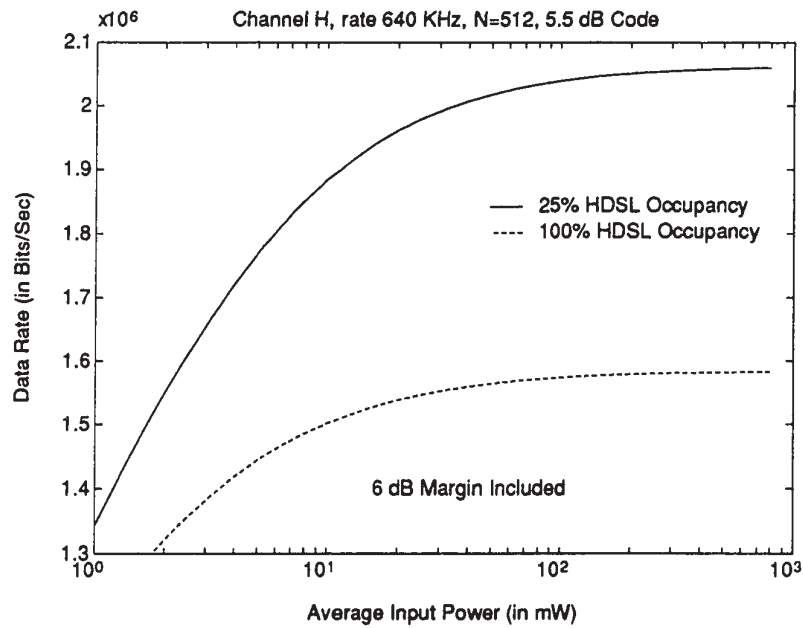


Figure 4.11: Effect of HDSL occupancy on data rate (6 dB margin included)

For an impulse of 40 samples, the above margins would reduce to approximately 13 dB and 10 dB, respectively; nevertheless, these numbers are still excellent with respect to decision-feedback approaches. To the extent that margins quoted for HDSL depend some extent upon impulse noise immunity, those numbers can be reasonably reduced for DMT in HDSL.

Of course, the price paid for the added immunity is the latency, which however at 1 to 3 ms is still well within acceptable limits for latency in the loop transceiver system itself.

4.4.8 Effect of Loop Length

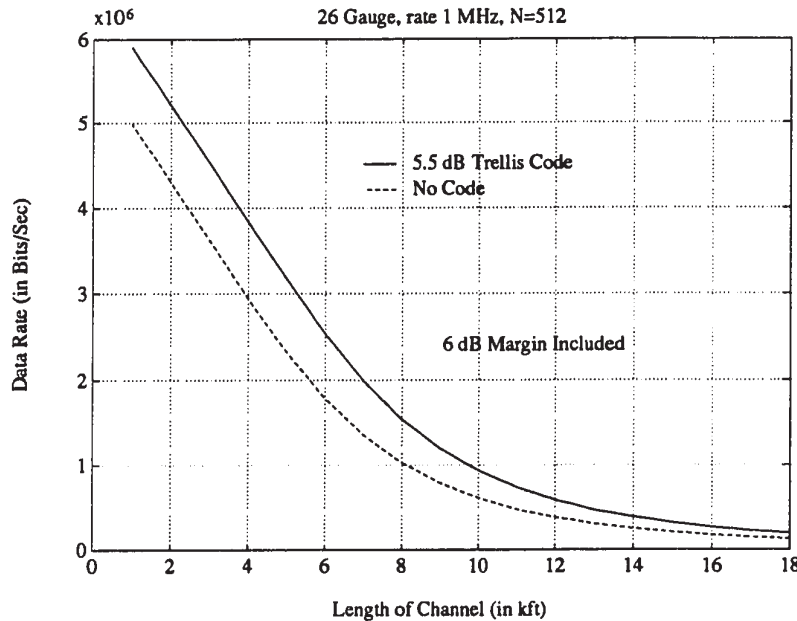


Figure 4.12: Effect of loop length on data rate (6 dB margin included)

To complete the performance results, Figure 4.12 shows the achievable data rate as a function of the length of a 26-gauge twisted-pair wire. The sampling rate is set at

1 MHz with 100% HDSL occupancy. We can see that the data rate decreases as the length of the loop increases, since longer loop suffers from larger attenuation.

4.4.9 Discussion

Our results clearly illustrate fundamental tradeoffs between performance and computational requirement. For instance, larger block lengths and higher sampling rates can always lead to performance improvement. Also, increasing transmit power can always improve performance. Nevertheless, our results indicate that a system with block length $N = 512$ and an $M + 1 = 10$ tap feedforward equalizer at sampling rate 640 kHz with 10 mW of transmit power is sufficient to achieve the desired data rates at the target error rate of 10^{-7} on nearly all CSA channels. Our projected computational requirement on a representative DSP (Motorola 56000) indicate that these numbers lead to a DMT transceiver that is practical for implementation. Further, both the transmitter (near-end) and receiver (far-end) might be implemented on a single programmable DSP (with considerable compute power remaining for echo cancellation) if that DSP were custom designed so that it had an FFT peripheral in addition to the standard multiply-accumulate arithmetic unit. This estimate, however, does not include the implementation of the Viterbi decoder for any applied trellis code. Nevertheless, a programmable Viterbi Signal Processor has been proposed [60] that could on a single VLSI circuit implement any of the codes that were used in this study.

Theoretically, it has been shown [27] [61] that decision-feedback systems can be designed to achieve almost as high performance as DMT systems if the DFE system is allowed infinite complexity. Such systems require knowledge of the channel at the transmitter (as does DMT) in the form of both a Tomlinson precoder, an optimized transmit filter, and an optimized sampling rate. The code used in the DFE system is the same in performance and complexity as the one used in the DMT system. A detailed study in [47] showed that approximately 128 total tap coefficients, feedforward and feedback combined, were needed by a decision feedback equalizer that was optimized and performed close to the same as the specific DMT system analyzed in this paper. An adaptive equalizer with 128 taps requires in excess of 100 Mips of

processing power, which is significantly more than the DMT system's 35 Mips.

There are other practical advantages of DMT with respect to baseband systems (like 2B1Q). One is improved tolerance to impulse noise and the ability to shape the transmit spectrum automatically. For instance, some of the subchannels in the lower 4 kHz might reliably transmit data even on a disfunctioning HDSL channel, such as one that has not had a loading coil removed, thus allowing routine maintenance functions on a channel that would otherwise be useless with baseband transmission. Another advantage is the possibility of a cost-effective programmed transmitter and receiver combination that would permit modification or alteration of function through software change (rather than redesign of a custom VLSI circuit) should unforeseen problems arise during field testing. A third advantage is the cost reduction that occurs when using the service in the single-duplex mode. The transmitter that is used for dual-duplex is already cost-effective but also sufficiently powerful to transmit 1.6 Mbps on a single twisted-pair reliably, albeit with reduced margin. This is because the second line interface and analog-to-digital interfaces need not be duplicated, and also because no (other) hardware changes are necessary (only software) to implement the single duplex. The possibility of the single-duplex service being far more desirable than dual-duplex has already been espoused by some potential HDSL customers [62], as sufficient copper for the two-twisted-pair service may not exist in many areas. A fourth practical advantage is that services such as the Asymmetric Digital Subscriber Line (ADSL) can also be implemented via software change in the implementing hardware [63].

4.5 Summary

In this Chapter we evaluated the performance of the DMT system as applied to high-speed digital subscriber line (HDSL) application. The digital subscriber lines are the local telephone wires, consisting of mostly 24- and 26-gauge copper twisted-pair wires. HDSL service targets a data rate of 1.6 Mbps bi-direction transmission over one or two copper wires. The noise impairments include crosstalk noise, echo, thermal, electronic, and impulse noise. We inspected the channel characteristics and derived models for the noise impairments. The full potential or capacity of these set

of loops with the assumed noise models are evaluated as an indication of how well the DMT system can perform relative to the theoretical maximum. The performance results are shown based on certain ideal conditions. We found that the use of a trellis code, higher input power, longer block length, and higher sampling rate generally improves the performance. We concluded that the target rate of 1.6 Mbps can be easily achieved with an appropriately designed DMT transceiver system.

Chapter 5

Performance of the Finite-Length DMT System

The performance results in Chapter 4 are based on the ideal theoretical performance analysis as described in Section 4.3. Certain ideal conditions are assumed, and thus the projected performance results are slightly optimistic. We begin this Chapter by identifying various finite-length effects. A matrix representation is derived to model the DMT system more accurately. However, this matrix description of the DMT system does not cover all finite-length effects. As a result, computer simulation programs are written to account for the remaining finite-length effects. The difference in performance between ideal theoretical analysis and computer simulation is then compared.

5.1 Finite-Length Effects

In Section 4.3, we ignore the effects of cyclic prefix to arrive at the achievable data rate. The length of the cyclic prefix, in practice, is a design parameter that has a close relationship with hardware specification since it could affect the hardware circuitry for the detection of frame boundary. The impact of cyclic prefix is more profound when the block length is relatively small, since the rate loss caused by the cyclic prefix is $\nu/(N + \nu)$. In this Chapter, we will vary both N and ν to examine the effects of

cyclic prefix.

The performance analysis derived in Section 4.3 also assumes that the DFT/IDFT vectors are ideal, brickwall, bandpass filters. In practice, neighboring subchannels do interfere with each other. Again, the interference is more significant as the block length decreases.

The assumption that the frequency response of each subchannel is approximately flat, and thus is free of ISI and can be characterized by the power gain $|H_n|^2$ is another ideal assumption that is true only if the block length is infinite. This assumption becomes a poor one when either the block length is small, or the frequency response of the channel changes rapidly, as in the case when frequency notches are present. To illustrate these phenomena, we examine the two cases separately.

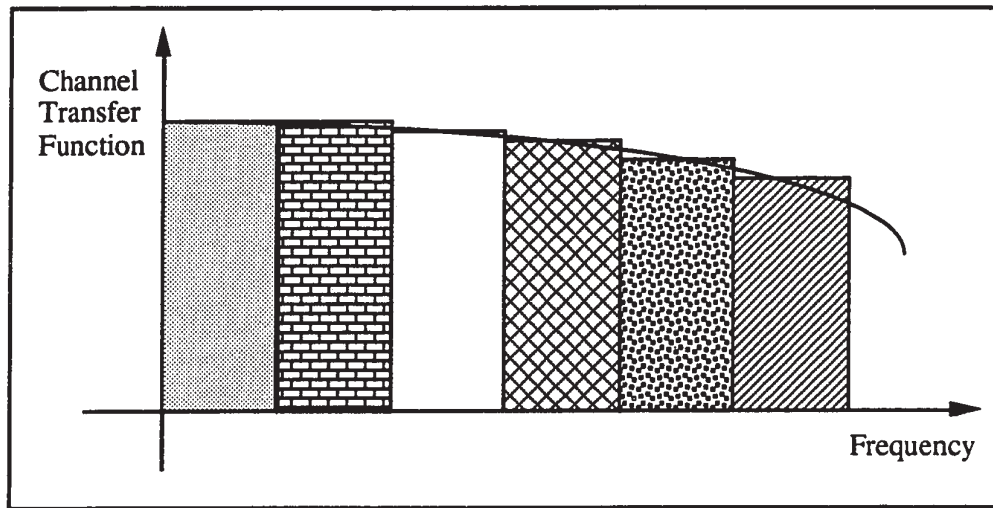


Figure 5.1: Smooth channel transfer function with small block length

Figure 5.1 shows a channel with no frequency notches and has a relatively smooth transfer function. If a DMT system with a small block length is used, as shown in Figure 5.1, then the frequency response of each subchannel is not flat, and thus each subchannel exhibits ISI and indeed interferes with other subchannels. As a result, the assumption that each frequency-indexed subchannel is flat and ISI-free is invalid.

Figure 5.2 shows a channel with a notch at D.C. Even with a DMT system with a large block length, the subchannels close to D.C. still suffer from significant ISI. In this case, the ISI-free assumption is no longer valid for the subchannels near D.C.

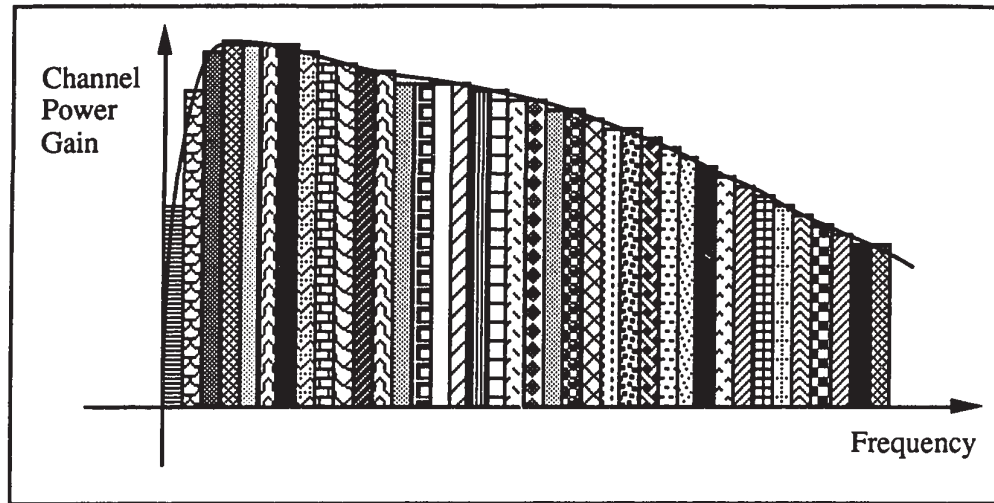


Figure 5.2: Channel with frequency notches

This effect and the data rate loss caused by cyclic prefix are the two most dominant finite-length effects. Section 5.4 will examine the performance degradation caused by these effects.

Other subtle finite-length effects include the fact that the noise is not cyclically extended though the signal is, and that the (revised) DMT receiver results in an error equation as given by equation (3.55), instead of the error equation (3.4) used in Section 3.2. In addition, the derivation in Section 3.2 does not take into account the effects of the finite length FFT/IFFT operations, and the interplay between time/frequency, since the data domain of the DMT system is in the frequency domain.

The decoupled, one-tap complex LMS filters in the frequency domain also contribute to the overall achievable data rate.

5.2 Theoretical Analysis of the Finite-Length DMT System

The finite-length effects described in Section 5.1 invalidate one of the key equations in Section 3.4.2, namely, the SNR for the n^{th} subchannel, equation (3.63),

$$SNR_n = \frac{\mathcal{E}_n |H_n|^2}{\sigma_n^2} . \quad (5.1)$$

In this Section, we assume the length of $w(D)$ and $b(D)$ are finite and fixed, as described in Section 3.3. In addition, the taps setting for $w(D)$ and $b(D)$ are known and obtained through the finite-length optimization algorithm. Performance analysis using noise autocorrelation functions similar to those used in the derivation of Section 3.2 is first presented to compute SNR_n with finite-length effects. A second analysis method, based on matrix representation of the DMT system, is then derived.

5.2.1 Analysis of Finite-Length DMT System

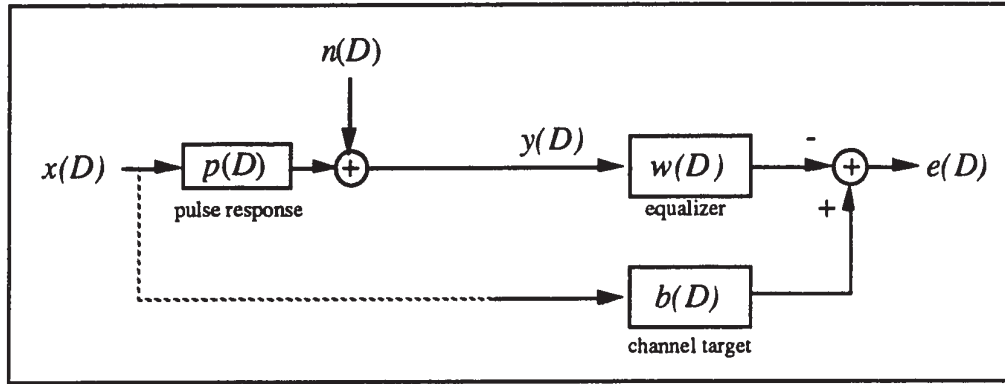


Figure 5.3: Block diagram to analyse finite-length DMT systems

In practice, the matched filter in Figure 3.1 is absorbed into the feedforward equalizer $w(D)$. With appropriate sampling rate (satisfying Nyquist condition), the continuous

channel response can be converted to a discrete channel response without loss of information. The block diagram of the input-assisted equalization in Figure 3.1 is now converted to Figure 5.3, where the quantities $p(D)$, $w(D)$, and $b(D)$ are all finite and known.

The error equation, as before, is

$$e(D) = b(D)x(D) - w(D)y(D) \quad . \quad (5.2)$$

As before, to minimize mean-square error, $E[e_k y_l^*] = 0 \forall k, l$, and be written as

$$E[e(D)y^*(D^{-1})] = 0 \quad (5.3)$$

$$= b(D)R_{xy}(D) - w(D)R_{yy}(D) \quad , \quad (5.4)$$

so

$$w(D) = b(D) \frac{R_{xy}(D)}{R_{yy}(D)} \quad , \quad (5.5)$$

the same as in Section 3.2.

However, since the matched filter is now part of $w(D)$,

$$y(D) = p(D)x(D) + n(D), \quad (5.6)$$

and so

$$R_{yy}(D) = p(D)p^*(D^{-1})\mathcal{E}_x + R_{nn}(D), \quad (5.7)$$

$$R_{yx}(D) = p(D)\mathcal{E}_x, \quad (5.8)$$

and

$$R_{xy}(D) = \mathcal{E}_x p^*(D^{-1}). \quad (5.9)$$

The error autocorrelation sequence is given by

$$\begin{aligned} R_{ee}(D) &= \mathcal{E}_x b(D) b^*(D^{-1}) + w(D) R_{yy}(D) w^*(D^{-1}) - \\ &\quad b(D) R_{xy}(D) w^*(D^{-1}) - w(D) R_{yx}(D) b^*(D^{-1}) \end{aligned} \quad (5.10)$$

$$\begin{aligned} &= \mathcal{E}_x b(D) b^*(D^{-1}) + w(D) R_{yy}(D) w^*(D^{-1}) - \\ &\quad \mathcal{E}_x b(D) p^*(D^{-1}) w^*(D^{-1}) - \mathcal{E}_x w(D) p(D) b^*(D^{-1}), \end{aligned} \quad (5.11)$$

where we can then substitute the known values for $p(D)$, $w(D)$, and $b(D)$. Note that the two parameters, i and Δ , as described in Section 3.3, need to be taken into account to ensure correct relative delay between $b(D)$ and $w(D)$.

As an example, assume $i = 0$, some fixed value of Δ , the number of taps in $w(D)$ is $M + 1$ and the number of taps in $b(D)$ is $\nu + 1$, then

$$w(D) = w_0 + w_1 D + \dots + w_M D^M, \quad (5.12)$$

and

$$b(D) = D^\Delta + b_1 D^{\Delta+1} + \dots + b_\nu D^{\Delta+\nu}. \quad (5.13)$$

The autocorrelation of the filtered-input sequence $u(D) \triangleq b(D)x(D)$, $R_{uu}(D)$, is given by

$$R_{uu}(D) = \mathcal{E}_x b(D) b^*(D^{-1}). \quad (5.14)$$

It is well known that the Fourier transform of an autocorrelation sequence is the power spectral density [15]

$$\Phi_{uu,m} = \sum_{k=0}^{N-1} R_{uu,k} e^{-j \frac{2\pi k m}{N}} \quad k = 0, \dots, N-1, \quad (5.15)$$

where

$$R_{uu}(D) = \dots + R_{uu,-2} D^{-2} + R_{uu,-1} D^{-1} + R_{uu,0} + R_{uu,1} D + R_{uu,2} D^2 + \dots, \quad (5.16)$$

and

$$\Phi_{ee,m} = \sum_{k=0}^{N-1} R_{ee,k} e^{-j\frac{2\pi km}{N}} \quad k = 0, \dots, N-1, \quad (5.17)$$

where

$$R_{ee}(D) = \dots + R_{ee,-2}D^{-2} + R_{ee,-1}D^{-1} + R_{ee,0} + R_{ee,1}D + R_{ee,2}D^2 + \dots \quad (5.18)$$

The SNR of the n^{th} subchannel, SNR_n , is then equal to

$$SNR_n = \frac{\Phi_{uu,n}}{\Phi_{ee,n}}. \quad (5.19)$$

Once the SNRs are determined, the other equations to determine the achievable data rate are the same as in Section 3.4.2. This method provides a fairly accurate way of measuring the achievable data rate for finite-length DMT systems. However, it does not take into account the full effects of the block length N and the cyclic prefix.

5.2.2 Matrix Representation of the Finite-Length DMT System

This approach is similar to the approach described in Section 5.2.1, since we again assume known $p(D)$, $w(D) = w_0 + w_1D + \dots + w_MD^M$, and $b(D) = b_0 + b_1D + \dots + b_\nu D^\nu$. In addition, we take into account the effects of the block length N , and the cyclic prefix, by using a block approach.

To simplify notations, we define $z(D) = w(D)y(D)$, and $u(D) = b(D)x(D)$. The error equation is now

$$e(D) = u(D) - z(D) \quad (5.20)$$

Assume for now that $\Delta = 0$, the case where $\Delta > 0$ will be discussed later. Assume that $h(D)$ is the combined response of the channel, $p(D)$, and the feedforward equalizer, $w(D)$, so

$$h(D) = p(D)w(D) \quad (5.21)$$

$$= h_0 + h_1 D + \dots + h_{LL} D^{LL} \quad (5.22)$$

In general, the finite-length algorithm in Section 3.3 does not force the combined response to be limited to exactly $\nu + 1$ taps, since a minimum mean-square error solution is used rather than a zero-forcing solution. Thus we assume that $h(D)$ could extend beyond several blocks, especially if the block length N is small.

A block of equalizer output samples, $\mathbf{z}$, is given by

$$\mathbf{z} = H\mathbf{x} + W\mathbf{n} + H'\mathbf{x}' + H''\mathbf{x}'' + \dots \quad (5.23)$$

where

$$\mathbf{z} = \begin{bmatrix} z_{N-1} \\ z_{N-2} \\ \vdots \\ z_1 \\ z_0 \end{bmatrix}, \quad (5.24)$$

$$\mathbf{x} = \begin{bmatrix} x_{N-1} \\ x_{N-2} \\ \vdots \\ x_1 \\ x_0 \end{bmatrix}, \quad (5.25)$$

$$\mathbf{n} = \begin{bmatrix} n_k \\ n_{k-1} \\ \vdots \\ n_{k-N-M+1} \end{bmatrix}, \quad (5.26)$$

$$W = \begin{bmatrix} w_0 & w_1 & \dots & w_M & 0 & 0 & \dots & 0 \\ 0 & w_0 & w_1 & \dots & w_M & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 & w_0 & \dots & w_M \end{bmatrix}, \quad (5.27)$$

and

$$H = \begin{bmatrix} h_0 + h_N & h_1 + h_{N+1} & \dots & h_{\nu-1} + h_{N+\nu-1} & h_\nu & h_{\nu+1} & \dots & h_{N-1} \\ h_{N-1} & h_0 + h_N & \dots & h_{\nu-2} + h_{N+\nu-2} & h_{\nu-1} & h_\nu & \dots & h_{N-2} \\ h_{N-2} & h_{N-1} & \dots & h_{\nu-3} + h_{N+\nu-3} & h_{\nu-2} & h_{\nu-1} & \dots & h_{N-3} \\ \vdots & \ddots & \ddots & \vdots & \ddots & \ddots & \ddots & \vdots \\ h_{N-\nu} & h_{N-\nu+1} & \dots & h_{N-1} & h_0 & h_1 & \dots & h_{N-\nu-1} \\ h_{N-\nu-1} & h_{N-\nu} & \dots & h_{N-2} & 0 & h_0 & \dots & h_{N-\nu-2} \\ h_{N-\nu-2} & h_{N-\nu-1} & \dots & h_{N-3} & 0 & 0 & \dots & h_{N-\nu-3} \\ \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\ h_1 & h_2 & \dots & h_\nu & 0 & 0 & \dots & h_0 \end{bmatrix} \quad (5.28)$$

where

$$H = H_0 + H_{CP}, \quad (5.29)$$

and

$$H_0 = \begin{bmatrix} h_0 & h_1 & h_2 & \dots & h_{N-1} \\ 0 & h_0 & h_1 & \dots & h_{N-2} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & h_0 \end{bmatrix}, \quad (5.30)$$

$$H_{CP} = \begin{bmatrix} h_N & h_{N+1} & \dots & h_{N+\nu-1} & 0 & \dots & 0 \\ h_{N-1} & h_N & \dots & h_{N+\nu-2} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ h_1 & h_2 & \dots & h_\nu & 0 & \dots & 0 \end{bmatrix}. \quad (5.31)$$

$\mathbf{x}$ is the current block of input symbols, and since we add the cyclic prefix $x_{N-\nu}, \dots, x_{N-1}$, each block of input symbols spans $N + \nu$ time symbols. The effect of the cyclic prefix symbols is accounted for by including H_{CP} as part of H .

$\mathbf{x}'$ is the next block of input symbols, and $H'\mathbf{x}'$ takes into account the effect of the next block of input symbols to the current block of equalized symbols $\mathbf{z}$. The net effect of previous blocks of input symbols is introducing ISI, and is treated as noise.

$$\mathbf{x}' = \begin{bmatrix} x'_{N-1} \\ x'_{N-2} \\ \vdots \\ x'_1 \\ x'_0 \end{bmatrix}, \quad (5.32)$$

and

$$H' = H'_0 + H'_{CP}, \quad (5.33)$$

where

$$H'_0 = \begin{bmatrix} h_{N+\nu} & h_{N+\nu+1} & \dots & h_{2N+\nu-1} \\ h_{N+\nu-1} & h_{N+\nu} & \dots & h_{2N+\nu-2} \\ \vdots & \ddots & \ddots & \vdots \\ h_{\nu+1} & h_{\nu+2} & \dots & h_{N+\nu} \end{bmatrix}, \quad (5.34)$$

and

$$H'_{CP} = \begin{bmatrix} h_{2N+\nu} & h_{2N+\nu+1} & \dots & h_{2N+2\nu-1} & 0 & \dots & 0 \\ h_{2N+\nu-1} & h_{2N+\nu} & \dots & h_{2N+2\nu-2} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ h_{N+\nu+1} & h_{N+\nu+2} & \dots & h_{N+2\nu} & 0 & \dots & 0 \end{bmatrix}. \quad (5.35)$$

If $LL > N + 2\nu$, then $H'' \neq 0$ and can be similarly derived. The filtered-input sequence is given by

$$\mathbf{u} = B\mathbf{x} \quad (5.36)$$

where

$$\mathbf{u} = \begin{bmatrix} u_{N-1} \\ u_{N-2} \\ \vdots \\ u_1 \\ u_0 \end{bmatrix}, \quad (5.37)$$

and

$$B = \begin{bmatrix} b_0 & b_1 & \dots & b_{\nu-1} & b_\nu & 0 & \dots & 0 \\ 0 & b_0 & b_1 & \dots & b_\nu & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ b_2 & \dots & b_\nu & 0 & \dots & 0 & b_0 & b_1 \\ b_1 & \dots & b_{\nu-1} & b_\nu & 0 & \dots & 0 & b_0 \end{bmatrix}. \quad (5.38)$$

The error autocorrelation matrix is given by

$$\mathbf{R}_{ee} = \mathbf{R}_{uu^T} + \mathbf{R}_{zz^T} - \mathbf{R}_{uz^T} - \mathbf{R}_{zu^T}, \quad (5.39)$$

where T denotes the complex conjugate transpose, and

$$\mathbf{R}_{uu^T} = \mathcal{E}_x \mathbf{B} \mathbf{B}^T, \quad (5.40)$$

$$\mathbf{R}_{zz^T} = \mathcal{E}_x \mathbf{H} \mathbf{H}^T + W \mathbf{R}_{nn^T} W^T + \mathcal{E}_x \mathbf{H}' \mathbf{H}'^T + \mathcal{E}_x \mathbf{H}'' \mathbf{H}''^T + \dots \quad (5.41)$$

where $\mathbf{R}_{nn^T}$ is the noise autocorrelation matrix and is a constant times the identity matrix if the noise is white. We also assume that different blocks of input symbols are not correlated, so that

$$\mathbf{R}_{\mathbf{x}^{(i)} \mathbf{x}^{(j)T}} = 0, \quad i \neq j, \quad (5.42)$$

and that there is no correlation between the noise and the blocks of input symbols

$$\mathbf{R}_{\mathbf{n} \mathbf{x}^{(j)T}} = 0. \quad (5.43)$$

Similarly,

$$\mathbf{R}_{\mathbf{u}\mathbf{z}^T} = \mathcal{E}_x \mathbf{B} \mathbf{H}^T, \quad (5.44)$$

and

$$\mathbf{R}_{\mathbf{z}\mathbf{u}^T} = \mathcal{E}_x \mathbf{H} \mathbf{B}^T. \quad (5.45)$$

Similar to Section 5.2.1, the SNR of the n^{th} subchannel, SNR_n , is then equal to

$$SNR_n = \frac{\Phi_{\mathbf{u}\mathbf{u},n}}{\Phi_{\mathbf{e}\mathbf{e},n}}, \quad (5.46)$$

where $\Phi_{\mathbf{u}\mathbf{u},n}$ is the power spectral density and is obtained by taking the Fourier transform of $\mathbf{R}_{\mathbf{u}\mathbf{u}^T}$.

However, $\mathbf{R}_{\mathbf{e}\mathbf{e}}$ in general is not Toeplitz. In fact, it is not even stationary. A method to estimate the autocorrelation sequence from $\mathbf{R}_{\mathbf{e}\mathbf{e}}$ is to average each time lag with a triangular window. Since multiplying a triangular window in the time domain corresponds to convolving with a sinc^2 function [64], the triangular window guarantees that the estimated power spectral density to be non-negative at all frequencies.

Denote each element in $\mathbf{R}_{\mathbf{e}\mathbf{e}}$ by $R_{i-1,j-1}$

$$\mathbf{R}_{\mathbf{e}\mathbf{e}} = \begin{bmatrix} R_{0,0} & R_{0,1} & \dots & R_{0,N-1} \\ R_{1,0} & R_{1,1} & \dots & R_{1,N-1} \\ \vdots & \ddots & \ddots & \vdots \\ R_{N-1,0} & R_{N-1,1} & \dots & R_{N-1,N-1} \end{bmatrix}, \quad (5.47)$$

and let the estimated autocorrelation sequence be $R_{AVE,k}$, where k is the time lag. The process of averaging with a triangular window is described by

$$R_{AVE,k} = \frac{1}{N} \sum_i R_{i,i+k}, \quad k = 0, \dots, N-1. \quad (5.48)$$

We have assumed so far that $\Delta = 0$. When $\Delta > 0$, instead of using equations (5.21) and (5.22), we define

$$g(D) = p(D)w(D) \quad (5.49)$$

$$= g_0 + g_1 D + \dots + g_{LL} D^{LL} \quad (5.50)$$

$$= g_0 + g_1 D + \dots + g_{\Delta-1} D^{\Delta-1} + h(D) D^{\Delta} \quad (5.51)$$

so

$$h(D) \triangleq h_0 + h_1 D + \dots + h_{LL-\Delta} D^{LL-\Delta} \quad (5.52)$$

and

$$h_i = g_{i+\Delta}, \quad i = 0, \dots, LL - \Delta. \quad (5.53)$$

The non-zero (though small) terms $g_0, \dots, g_{\Delta-1}$ are the pre-cursor ISI, which is an additional noise term introduced by future blocks of input symbols.

The block of equalizer output samples, $\mathbf{z}$, is now given by

$$\mathbf{z} = H\mathbf{x} + W\mathbf{n} + G\mathbf{x}^F + H'\mathbf{x}' + H''\mathbf{x}'' + \dots \quad (5.54)$$

where the extra term $G\mathbf{x}^F$ is the pre-cursor ISI,

$$G = \begin{bmatrix} g_{\Delta-1} & g_{\Delta-2} & \dots & g_0 & 0 & \dots & 0 \\ g_{\Delta-2} & \dots & g_0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ g_0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \quad (5.55)$$

and

$$\mathbf{x}^F = \begin{bmatrix} x_{N-\nu}^F \\ \vdots \\ x_{N-1}^F \\ x_0^F \\ x_1^F \\ \vdots \\ x_{N-\nu-1}^F \end{bmatrix}, \quad (5.56)$$

where $\mathbf{x}^F$ is the future blocks of input symbols.

Equation (5.54) assumes that $\Delta \leq N$. If $\Delta > N$, then precursor ISI from other future blocks need to be included, and can be similarly defined.

The rest of the derivation is identical to the case $\Delta = 0$.

This method takes into account the effects of N and ν , and thus is more accurate than the approach in Section 5.2.1. However, the resulting SNR is only an approximation due to the non-stationary problem. In addition, this method still does not include the effect of the one-tap adaptive filter.

5.3 Monte-Carlo Computer Simulations for the DMT system

To account for all finite-length effects as explained in Section 5.1, the author has written a set of Monte-Carlo computer simulation programs for the DMT system. This set of computer simulation programs is written in a software package called **Matlab**, a high-level pseudo-code language, and has been modified and updated over the past three years. The latest version of the programs are included in Appendix C and will be described in this Section. This set of simulation programs also serves as a bridge between theoretical analysis and the real-time prototype implemented with multiple commercially available digital signal processors (see Chapter 6).

5.3.1 Channel and Noise Characterization

The channel characteristics in the form of a time-domain response or a frequency response (both magnitude and phase information are required) is needed to perform simulations. Sampling rate is an implicit parameter since the simulation is done entirely in discrete time. For a fixed channel, different sampling rates result in different time and frequency responses. Thus, the first parameter we need to choose is the sampling rate. The characteristics of the CSA channels described in Section 4.1 were obtained from Bellcore with 16 MHz sampling rate. We subsequently decimated the response to arrive at an appropriate sampling rate, such as 640 kHz. Since the frequency response can be derived from the time response and vice versa, only one type of response is needed.

Likewise, noises needed to be characterized properly. Additive white Gaussian noise (AWGN) is relatively straightforward to generate in **Matlab**, since it has a one-line random number generator with an option of normal distribution. Other noises such as near-end crosstalk noise (NEXT) are typically defined in terms of transfer functions (see equation (4.1)). In order to obtain crosstalk noise samples, a filter is first designed in **Matlab** based on the frequency characteristics of the noise (see the **Matlab** program *nextfilter.m* in Appendix C.1). AWGN samples then pass through the filter to generate the crosstalk noise samples. This process is shown in the program called *nexts.m* in Appendix C.2. This program assumes that other wires in the same bundle are using the same multi-carrier scheme, and so it uses the output symbols of a multi-carrier transmitter as the input to the crosstalk noise shaping filter. Of course, these symbols at the output of a multi-carrier transmitter needed to be uncorrelated with the output symbols of the DMT transmitter that is being simulated. To ensure this is the case, *nexts.m* uses 4 QAM as the signal constellations for all subchannels, whereas the DMT system to be simulated uses different signal constellations.

5.3.2 Finite-Length Algorithm for Computing Taps Settings

The next step for the simulation programs is to go through the finite-length algorithm, described in Section 3.3, to obtain the taps setting for the feedforward equalizer. The channel pulse response and the noise (combining AWGN and Crosstalk) autocorrelation sequence are the input to the algorithm. If the channel response is given in the frequency domain, then the program *ftot.m* in Appendix C.3 converts the frequency response of the channel to a time response. Program *noiseauto.m* in Appendix C.4 computes the noise autocorrelation sequence based on the frequency characteristics of the noise and the assumed average signal input power as described in Chapter 4.

After deciding on the number of feedforward taps ($M + 1$) and the length of the cyclic prefix (ν), program *computetaps.m* in Appendix C.5 performs the finite-length algorithm by going through all possible i 's, and for each i searching through all possible Δ to find the minimum MSE taps setting for each i (see Section 3.3). This program calls two subroutines *deltap.m* and *deltan.m* in Appendix C.6, which compute the taps setting for positive and negative Δ s, respectively.

5.3.3 Signal-to-Noise Ratio Measurement

Once the taps setting for the feedforward equalizer is obtained, a Monte-Carlo simulation program *findsnr.m* in Appendix C.7 is used to measure the signal-to-noise ratio (SNR) of each subchannel. As explained in Sections 5.1 and 5.4.2, the actual SNRs for some subchannels can be significantly different from the theoretical values obtained from equation (3.63).

Since the purpose of this program is to measure actual SNR and not to measure bit error rate, it assumes that each subchannel uses a 16 QAM constellation. The 16 QAM constellation is decomposed into two 4 PAM constellations, one represents the real component and the other corresponds to the imaginary part. These 4 PAM input symbols (per dimension) are generated in the program *gen4pam.m* in Appendix C.8.

The program *findsnr.m* first obtains the one-dimensional input symbols from *gen4pam.m*, then combines them into 16 QAM constellation points. This is performed for the $N/2 - 1$ QAM subchannels. The program assumes that D.C. and

Nyquist subchannels are not used. Simple changes can be made to include these two subchannels. Hermitian symmetry is then applied to form the N -point block in the frequency domain, and IFFT is then performed to the N complex symbols. Cyclic prefix of length ν is then added to the output block of N real symbols.

The channel is simulated digitally, and as mentioned in Section 4.1, effects such as transformers can be modeled and included in the channel response. The basic **Matlab** function *filter* performs a general infinite impulse response (IIR) filter, so that the channel response can be either in the form of a finite impulse response (zeros only) or a infinite impulse response (pole-zero model). Crosstalk noise samples obtained from *nexts.m*, and AWGN generated within *findsnr.m*, are then added to the channel output symbols. Taking into account of appropriate delay, the channel output symbols with noises are filtered by the feedforward equalizer. The redundant symbols caused by cyclic prefix are also removed. An N -point FFT is then performed to the equalized symbols.

If we run through the simulation with sufficient training symbols, the set of one-tap, complex, LMS filters will converge to their optimum settings. To save time, in *findsnr.m* we directly compute their converged settings after running through certain number of block iterations. This set of converged taps is then used to compute the SNR for each subchannel. Based on the SNR values obtained, the achievable data rate and performance margin can be evaluated as described in Section 3.4.2. The program *bitassign.m* in Appendix C.9 then computes the bit allocation for each subchannel based on the SNRs and/or performance margin. This bit assignment information is required for the final steady-state simulation.

5.3.4 Steady-State Performance

This Section describes the **Matlab** Monte-Carlo simulation program called *datapump.m* Appendix C.10, which simulates the steady-state DMT transceiver system and computes the overall bit error rate.

The bit assignment is obtained from *bitassign.m*, and the equalizer settings from *computetaps.m*, the initial taps setting for the set of one-tap LMS filter from *findsnr.m*. This *datapump.m* program is very similar to the *findsnr.m* program, except that

1. each subchannel uses the actual number of bits assigned.
2. the actual data rate is fixed.
3. a bit decoder is implemented to decode the final output symbols to the actual bits.
4. the overall bit error rate is computed after enough block iterations.

The bit error rate for each individual subchannel is also computed in this program, which tracks the performance of each subchannel.

5.4 Computer Simulations Results

To demonstrate the difference between the ideal performance results and the actual performance, we use the channel response of a 9-kft, 26-gauge copper wire inside the carrier serving area (CSA) loops that were studied extensively in Section 4.4 and in [22]. The structure of the DMT system that we consider is also described in detail in Section 3.4. We fix the sampling rate of the DMT system at 640 kHz. We assume the noise impairments are intersymbol interference (ISI) and additive white Gaussian noise (AWGN) with a power spectral density of -110 dBm/Hz and an average input power of 10 mW. The bit error rate is set at 10^{-7} and a noise margin of 6 dB is assumed throughout this section. No coding is used. In order to account for all finite-length effects, the Monte-Carlo computer simulation program as described in Section 5.3 is used to generate the performance results for this Section.

5.4.1 Effects of Length of Feedforward Equalizer and Cyclic Prefix

We pick several different values for the length of the feedforward equalizer $M + 1$, and the length of the cyclic prefix, ν . The shorter the equalizer length, the less computational power required and thus the lower the cost. The shorter the cyclic prefix, the less significant the data loss caused by the extra samples transmitted. The

results for $M = 11$, $\nu = 7$, and $N = 512$ based on the algorithm outlined in Table 4.1 are shown in Table 5.1. Each column in the table lists the MMSE for the optimum Δ , and the data rate for the multi-carrier system using the taps setting derived from the algorithm. The achievable data rates for the multicarrier system do include the penalty caused by cyclic prefix.

i	0	1	2	3
σ_{MSE}^2 with Optimal Δ ($\times 10^{-3}$)	4.37	2.99	2.78	2.62
Δ	10	8	3	7
Data Rate (in Mbps)	1.791	1.765	1.776	1.741

i	4	5	6	ν
σ_{MSE}^2 with Optimal Δ ($\times 10^{-3}$)	2.19	2.37	2.49	2.92
Δ	7	5	3	3
Data Rate (in Mbps)	1.799	1.784	1.796	1.798

Table 5.1: Achievable data rate for the DMT system with $M = 11$, $\nu = 7$, $N = 512$

The achievable data rates among various values of i do not exhibit significant differences. This phenomenon manifests the good tolerance of the variations of the feedforward equalizer for the DMT system, and also raises the need for checking performance with other equalizer settings in addition to the optimum solution. Table 5.2 to Table 5.4 show similar results for different values of M and ν .

i	0	1	2	3	ν
σ_{MSE}^2 with Optimal Δ ($\times 10^{-3}$)	6.71	4.99	3.44	5.06	5.27
Δ	7	5	7	4	3
Data Rate for Multicarrier System (in Mbps)	1.802	1.792	1.803	1.762	1.761

Table 5.2: Achievable data rate for the DMT system with $M = 7$, $\nu = 4$, $N = 512$

The most important advantage of the finite-length optimization algorithm compared to [22] is the ability to reduce the number of feedforward taps and the length of the cyclic prefix with virtually no performance degradation, as evident between Table 5.1 and Table 5.3. The achievable data rate actually improves when we reduce the length of the equalizer by 8 and the cyclic prefix by 3. This translates to over 5 million

i	0	1	2	3	ν
σ_{MSE}^2 with Optimal Δ ($\times 10^{-3}$)	10.77	7.63	6.57	10.68	15.99
Δ	6	4	3	4	2
Data Rate (in Mbps)	1.83	1.83	1.82	1.73	1.67

Table 5.3: Achievable data rate for the DMT system with $M = 3, \nu = 4, N = 512$

i	0	1	ν
σ_{MSE}^2 with Optimal Δ ($\times 10^{-2}$)	1.56	2.33	2.46
Δ	6	6	3
Data Rate for Multicarrier System (in Mbps)	1.787	1.774	1.730

Table 5.4: Achievable data rate for the DMT system with $M = 3, \nu = 2, N = 512$

instructions per second (MIPS) of savings in terms of computational power. Thus a multi-carrier receiver can be designed that requires less than 15 MIPS. Table 5.5 shows the results when the block length is reduced from 512 to 128. Compared with Table 5.3, the throughput of this DMT system with smaller block length decreases by about 4%, while at the same time the total complexity decreases by more than 10% and the latency is reduced by 75%.

i	0	1	2	3	ν
σ_{MSE}^2 with Optimal Δ ($\times 10^{-3}$)	10.77	7.63	6.57	10.68	15.99
Δ	6	4	3	4	2
Data Rate (in Mbps)	1.76	1.76	1.76	1.61	1.51

Table 5.5: Achievable data rate for the DMT system with $M = 3, \nu = 4, N = 128$

To illustrate the high performance/cost ratio of the finite-length DMT system, we consider the extreme case in which no feedforward equalizer is used in the multi-carrier system, as would be the case for the multi-carrier system described in [2]. Since the channel response of most of the CSA loops are very long, if we ensure that there is absolutely no inter-block interference, the rate loss caused by a long cyclic prefix would result in a substantial performance degradation with a block length of

512 or less. As an alternative, we allow varying amounts of inter-block interference (IBI) by shortening the length of the cyclic prefix.

ν	6	13	19	25	31
Data Rate (in Mbps)	1.362	1.590	1.680	1.702	1.692

Table 5.6: Achievable data rate for multi-carrier systems with no feedforward equalizer

Table 5.6 shows the achievable data rate for some values of ν with a block length of 512, and Figure 5.4 plots the achievable data rate for block lengths of 128 and 512 with no feedforward equalizer. With the optimum size cyclic prefix, the throughput decreases by about 6.2% and the receiver complexity decreases from 14.8 MIPS to 12.2 MIPS. While the above rate loss may not be significant because of the relatively large block length of 512, other systems that have stringent end-to-end delay requirement may restrict the maximum block length to 128, in which case the resulting performance drops by nearly 20%, and would be unacceptable for most applications. Comparing the DMT system with a short feedforward equalizer and a block length of 128 (see Table 5.5) with the multi-carrier system without the feedforward equalizer and a block length of 128 (Figure 5.4), we find that the former system has a better data rate, comparable complexity, and a huge (75%) reductions in latency and memory requirements. Thus we have improved the system performance in terms of data rate, latency, and memory size - without increasing the cost. In order to achieve the same level of throughput as the DMT system and not using the feedforward equalizer, the block length would need to be increased to at least 1024, which would increase the complexity, latency, and memory requirement even further.

5.4.2 Comparisons of Ideal Performance Analysis and Computer Simulation Results

Table 5.7 and Table 5.8 summarize the difference between ideal performance analysis and computer simulation results. These tables also isolate one of the finite-length effects, namely, the rate loss caused by the cyclic prefix. Instead of using equation (4.11), we use equation (3.66) and substitute appropriate values for ν to compare

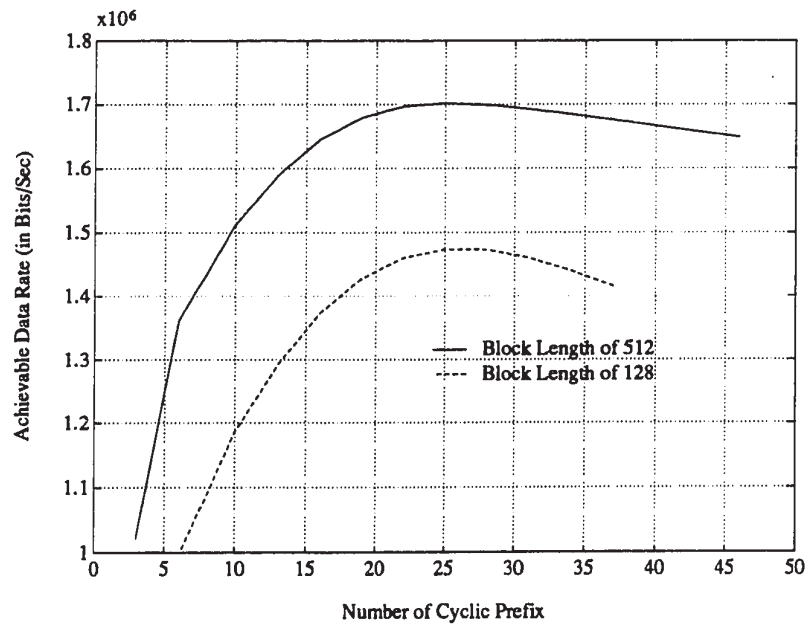


Figure 5.4: Achievable data rate for multi-carrier systems with no feedforward equalizer

with the finite-length DMT system with the same length of cyclic prefix. Table 5.7 compares the difference with a block length of 128. We can see that about 1 dB margin is loss with a cyclic prefix of length 7, even in the ideal analysis case. In order for the rate loss due to cyclic prefix to be negligible (say, for example, less than 1 dB of performance margin), the length of the cyclic prefix should be less than 5% of the block length. Comparing the second and third rows of Table 5.8, which use a block length of 512, to the same rows in Table 5.7, we can see that the effects of cyclic prefix are negligible as the block length increases. Again, as expected, the longer the block length, the better the performance, and this is evident in both ideal analysis and computer simulations.

		Data Rate (in Mbps)	Loss in Margin (in dB)
Ideal Performance Analysis	$\nu = 0$	1.87	0
	$\nu = 4$	1.81	0.5
	$\nu = 7$	1.77	0.9
Computer Simulations	$\nu = 4, M = 3$	1.76	1.1
	$\nu = 7, M = 11$	1.72	1.4

Table 5.7: Comparisons of ideal performance analysis and computer simulations (N=128)

		Data Rate (in Mbps)	Loss in Margin (in dB)
Ideal Performance Analysis	$\nu = 0$	1.90	0
	$\nu = 4$	1.88	0.1
	$\nu = 7$	1.87	0.2
Computer Simulations	$\nu = 4, M = 3$	1.83	0.6
	$\nu = 7, M = 11$	1.81	0.8

Table 5.8: Comparisons of ideal performance analysis and computer simulations (N=512)

From Table 5.7 and Table 5.8, we also observe that after the effect of the rate loss caused by cyclic prefix is removed (comparing the margin between the ideal performance analysis and the computer simulation results with the same length of

cyclic prefix), other finite-length effects reduce the performance margin further by about 0.5 dB.

To identify the source of the performance degradation, we first look at the power gain (transfer function) of this 9-kft, 26-gauge wire (Channel H). The power gain up to the Nyquist frequency (sampling rate = 640 kHz) is plotted in Figure 5.5. Figure 5.6 magnifies the gain between D.C. and 4 kHz.

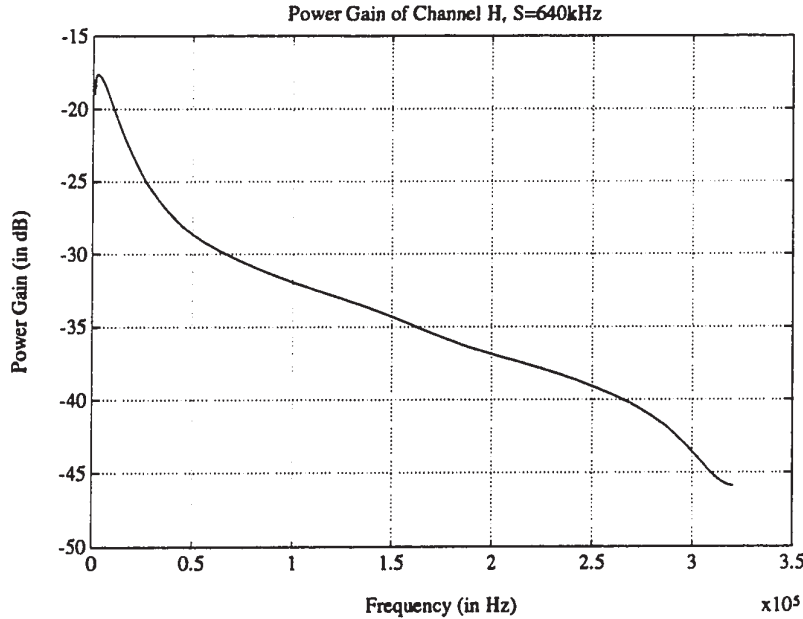


Figure 5.5: Transfer function of the 9-kft, 26-gauge wire

The frequency notch at D.C. is caused by the transformer at each end of the wire, and copper twisted-pair channels usually have transformers on both ends. As Figure 5.5 shows, the transfer function of the channel varies rapidly near D.C., and in fact changes significantly for the lowest 3 kHz range. As a result, the subchannels in the lowest 3 kHz range cannot be modeled as ISI-free subchannels, as described in Section 5.1, and the equation to compute the SNR of the subchannels, equation (3.63), is invalid for these subchannels.

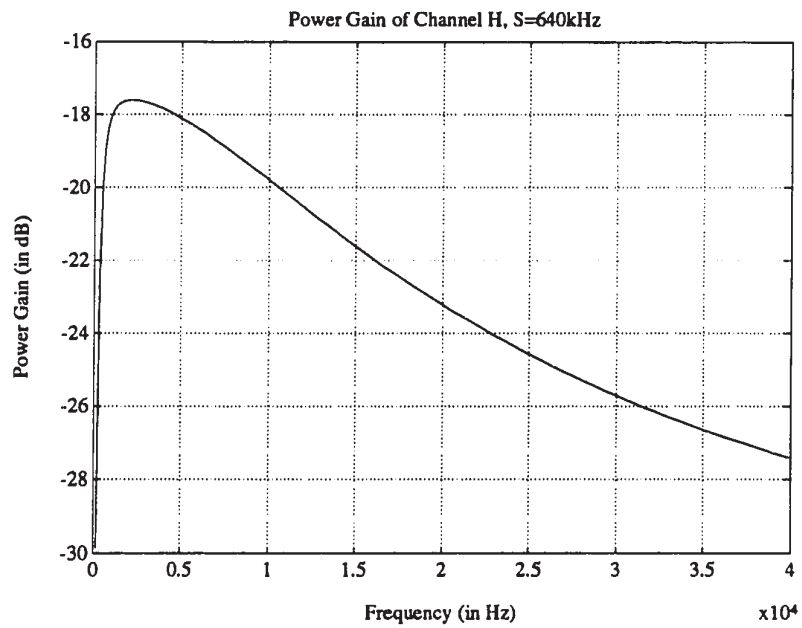


Figure 5.6: Transfer function of the 9-kft, 26-gauge wire (lowest 4 kHz)

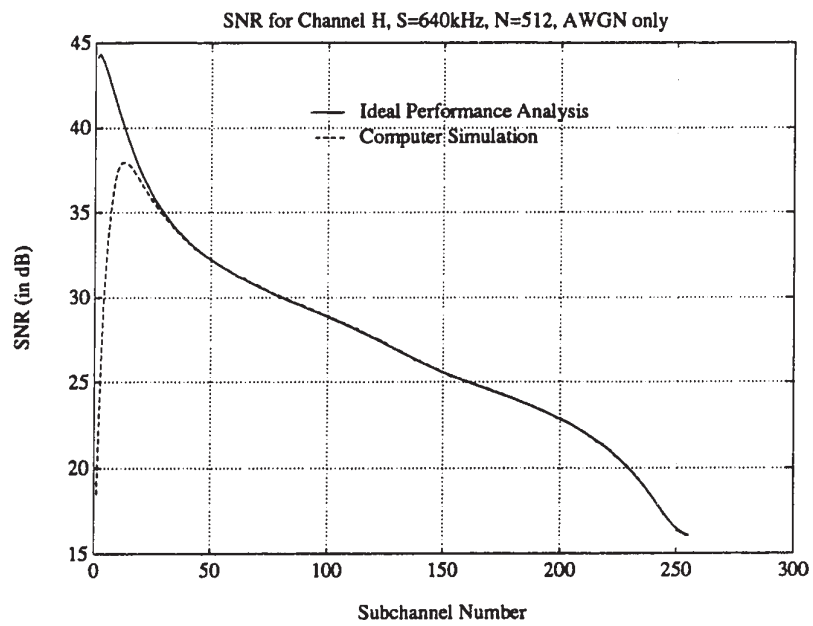


Figure 5.7: SNR for the DMT system with $M = 11$, $\nu = 7$, and $N = 512$

Using the computer simulation programs as described in Section 5.3, the SNRs of the subchannels for the DMT system with $M = 11$, $\nu = 7$, and $N = 512$ are shown in Figure 5.7, along with the ideal SNRs computed with equation (3.63). The subchannels closest to D.C. suffer the most degradation in performance, and the SNR loss for these subchannels accounts for the 0.6 dB loss in margin as shown in Table 5.8.

This phenomenon is even more profound when the multi-carrier system with no feedforward equalizer and with the optimum length of cyclic prefix ($\nu = 25$ with a block length of 512, see Table 5.6) is simulated. Figure 5.8 shows the actual SNRs for this system as well as the ideal SNRs.

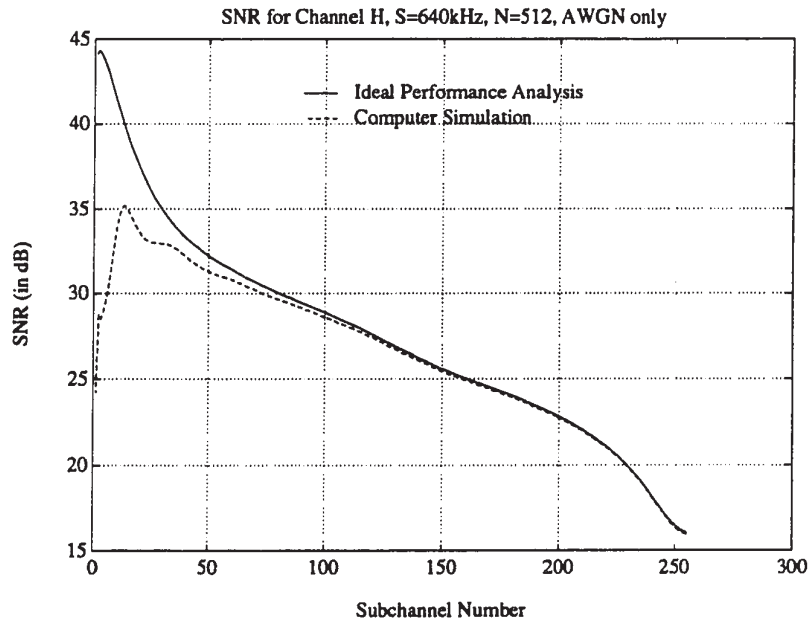


Figure 5.8: SNR for the multi-carrier system with no feedforward equalizer, $\nu = 7$, and $N = 512$

5.5 Summary

In this Chapter we summarized the finite-length effects that Chapter 4 did not consider. We identified the two dominant finite-length effects: the rate loss caused by cyclic prefix and the SNR degradation for the subchannels located at the portions where the channel transfer function varies rapidly. The rate loss caused by the cyclic prefix can be easily included as long as we know the length of the cyclic prefix used in the DMT system. To capture the SNR degradation, Section 5.2 presented two different methods to calculate the SNR based on more accurate models of the DMT system. Each of the two methods has its limitation, and thus the SNR values computed are not exactly the same as the actual SNRs. Nevertheless, both techniques capture most of the finite-length effects and project the SNRs very close to the actual values. A set of computer simulation programs, as described in Section 5.3, is written to encompass all finite-length effects. This set of programs computes the actual SNR as well as measuring the overall bit error rate of the DMT system.

Chapter 6

A Real-Time DMT Prototype

Computer simulation programs provide an accurate and easy way to verify the validity of the theory of multi-carrier systems. To support the simulation results and to substantiate the theory one step further, a real-time prototype serves as an ultimate proof-of-principle and as a final bridge between computer simulation and commercial products. This Chapter describes the design and implementation of the DMT transceiver prototype project at Stanford University. Although the initial target was the digital subscriber line (DSL) application, the DMT prototype is implemented with general purpose digital signal processors (DSP), so that different applications and designs can be verified through software prototyping.

6.1 Specification for the DMT Prototype

The DMT prototype is a combination of electronic hardware and signal processing software. The functionality, availability, and the cost determines the choice of suitable hardware, and the availability of manpower and development tools limit the scope of the software specification. In this Section, we briefly describe the hardware and software requirements.

6.1.1 DMT Hardware Requirement

We decided that general purpose digital signal processors (DSP) should be used as the platform for the DMT system. This allows that most of the signal processing functional blocks as described in Section 3.4 be implemented in software, and allows maximum flexibility to add, delete, and modify functional blocks.

The DSP should also have a high MIPS rating (an X-MIPS DSP means the ability of execute X million instructions per second) since a real-time prototype is targeted. However, the MIPS rating could be misleading since a particular function may require X instructions on one DSP and Y instructions on another DSP due to different architecture and pipelining. As a result, we pinpointed several computational intensive units as identified in Section 3.5 to serve as the benchmark for choosing the DSP. The most computational intensive function was identified as the FFT/IFFT blocks, followed by the time-domain feedforward equalizer and the 1-tap LMS filters.

The Motorola DSP 56000/1 series use a pipelined structure and perform FFT/IFFT more efficient than any other DSPs in the current market. It also performs sums of products extremely efficiently, which is the key computational element to implement the time-domain equalizer.

Another important consideration is the dynamic range of the DSP, or, in other words the number of bits for the data path. A DMT system applied to channels with severe ISI typically results in a dynamic range requirement of 60 to 80 dB, and thus a minimum of 16 bits is required for the data width.

The cost of the wide variety of commercially available DSPs varies significantly. Due to limited available research funds, the cost of the DMT prototype must be within budget.

Fortunately, the cost of the Motorola 56001 DSP is reasonable, and a discount is offered to Universities for research purpose! We thus decided, based on the aforementioned issues, to use Motorola DSP 56001s.

Based on the complexity estimation (Section 3.5), we concluded that multiple Motorola DSP 56001s would be required to implement the DMT system for the HDSL application in real-time. As a result, an interface board was required to provide a communication and data transfer link between DSPs. There is no commercially

available interface board to connect multiple Motorola DSPs, though a board linking two Motorola DSPs is available. A custom design interface board was required and described in Section 6.2.2. Other hardware requirements included analog-to-digital converters (ADC), digital-to-analog converters (DAC), a line-interface hybrid circuit, actual line or hardware line simulator, and additional circuitry for timing, framing, and other control functions.

6.1.2 DMT Software Requirement

Most of the functions of the DMT transceiver are implemented in software. This is necessary since the initial cost and time to design custom functional hardware peripheral for the DMT system was beyond the scope of our research project. The flexibility of software programs also provides an easy way to test and debug the system, and allows quick modifications for different system parameters and applications.

The key software requirements are the optimality and modularity of the codes for each function. Optimized codes allow the functions to be performed within the real-time specification and to reduce the cost and the hardware requirements. The codes for different functions needed to be modular since different functions could run on different DSPs. Another benefit of modularity is that rearranging and modifying individual functions can be done more efficiently, and that testing and debugging of the integrated system can be performed much easier since each individual function is first tested and verified. In addition, this allows specific algorithms to be independently designed and optimized to meet critical time bottlenecks, as in the case of the FFT/IFFT functions. In order to optimize the execution time, most DMT prototype software that is time-critical is to be written directly in DSP assembly code, and the specific assembly code is DSP dependent. Other off-line or non-time-critical functions can be first written in high-level languages and then translated into assembly code.

6.2 Hardware for the DMT Prototype

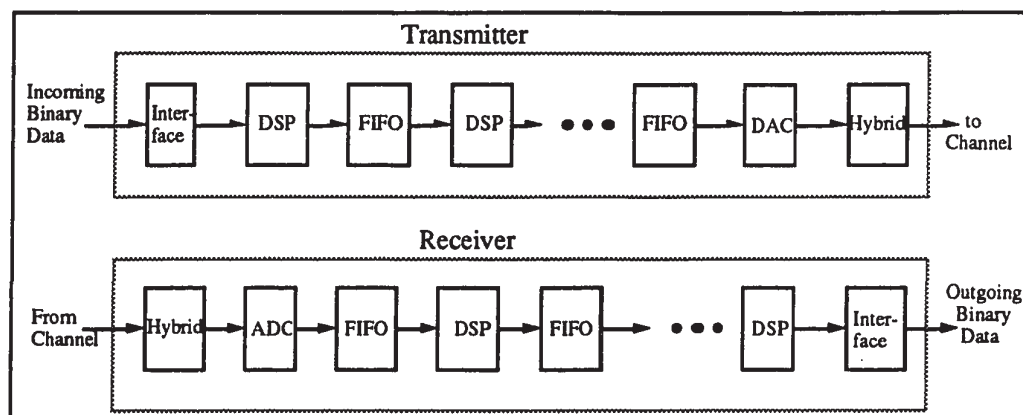


Figure 6.1: DMT prototype hardware configuration

The hardware configuration of the DMT prototype is shown in Figure 6.1. The heart of the prototype is the set of Motorola 56001 DSPs, which implements most of the functional blocks of the DMT system. A custom-designed interface board (FIFO) is used to connect any two successive DSPs so that data can be transferred from one DSP to another. The binary data source and sink for the prototype is a bit error rate counter, and an interface card is also required between the counter and the DSPs. In addition, a digital-to-analog converter and a hybrid (transformer) circuit are required to convert the digital symbols to analog form, perform suitable filtering, and output the waveform to the channel through the hybrid. A corresponding hybrid circuit (with anti-aliasing low-pass filter) and an analog-to-digital converter are needed at the front-end of the DMT receiver.

In this Section we will describe the DSP and the FIFO interface board in more detail. The hybrid circuits were designed by a consultant, and the interface card between the error rate counter and the DSPs was designed by a summer student. Since these two hardware components are not completed as of the date of this writing, we will not show detail description of these two components.

6.2.1 Motorola DSP 56001

The Motorola 56001 DSP is housed in a board called the “application development module” (ADM), along with serial and parallel input and output ports, an 8K off-chip RAM, and a host of other circuitry to provide programming, testing, and debugging features through a computer.

The DSP 56001 is a user-programmable, HCMOS, low-power, general purpose digital signal processor with 24-bit fixed-point arithmetic. It has two independent, expandable data memory spaces, two address generation units (AGUs), and a data arithmetic logic unit (ALU) with two accumulators, and two shifter/limiter circuits. The duality of the architecture allows a convenient way of partitioning the two memory spaces into real and imaginary spaces for complex arithmetic, and into data and coefficient spaces for filtering functions. The two 48-bit accumulators are augmented by two 8-bit accumulator extension registers, providing up to 56-bit of precision for intermediate results. There are four 24-bit general purpose data registers that can be concatenated to two 48-bit registers. In addition, eight 16-bit address registers, each with its own 16-bit offset register and 16-bit modifier register, are divided into two separate files, so that a parallel data access to the two memory spaces can be done in one instruction, along with an ALU operation. Real numbers are represented in fractional notation, which limits the range of real numbers to between -1 and +1. An earlier version of the DSP 56001 uses a 20.48 MHz clock. With two clock cycle per instruction, the DSP 56001 is a 10.24 MIPS processor. A more recent version of the DSP 56001 we obtained uses a 27 MHz clock and provides 13.5 MIPS of computing power.

6.2.2 Interface Board (FIFO)

The interface (FIFO) board allows the data to be transferred from one DSP to another unidirectionally, and also serves as the interface between the DSPs and the DAC/ADC. It is based on the first-in, first-out (FIFO) architecture and uses three memory chips of 2K deep and 9-bit wide, concatenating the three chips provides the

necessary 24-bit wide data path. The design of the FIFO board allows a single address to be used for reading and a single address (different) to be used for writing, and thus no address updating is needed for the DSPs to write to and read from the FIFO board. This FIFO board also has additional circuitries for checking how full or empty the memory chips are, so that the DSP writing to the FIFO board will not cause an overflow in memory space, and the DSP reading from the FIFO board will not read an empty memory space. The FIFO board was designed by Stanford Ph.D. students Cory Modlin and Jerry Tu, and fabricated by Bill Turner of Lintell Engineering.

6.3 Software for the DMT Prototype

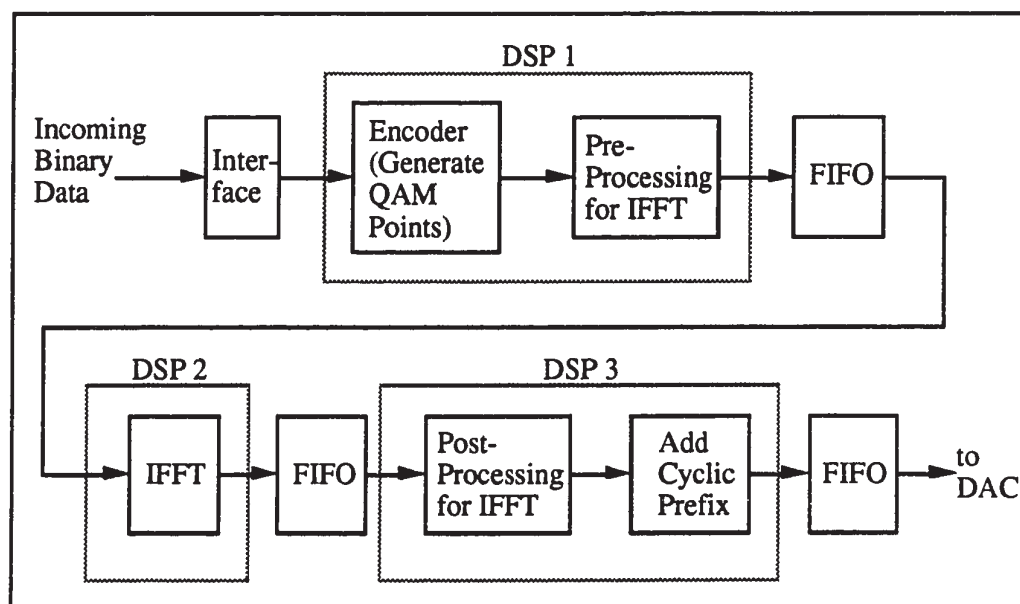


Figure 6.2: DMT transmitter prototype software functions

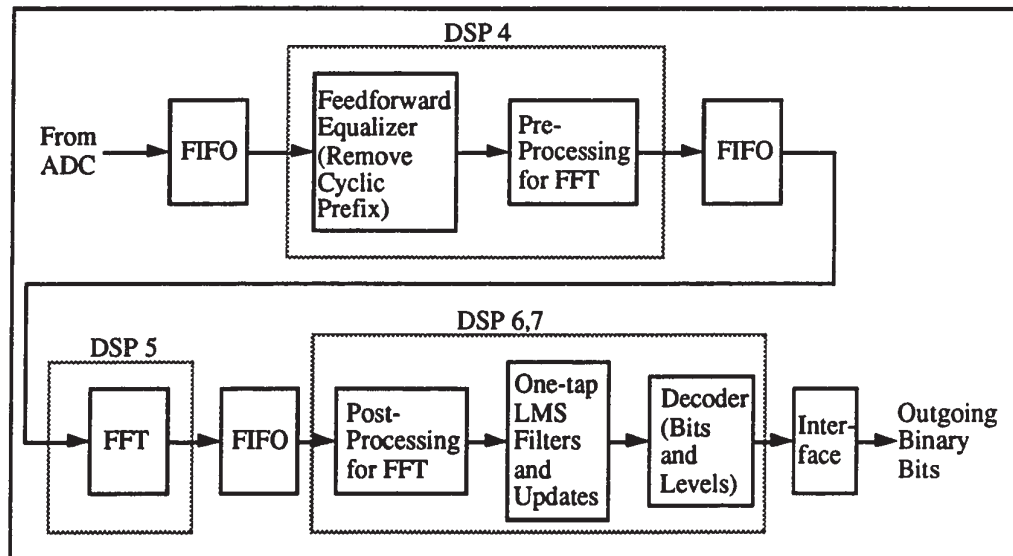


Figure 6.3: DMT receiver prototype software functions

The real-time constraint and the complexity of the DMT system at 640 kHz lead to the partitions as shown in Figure 6.2 and Figure 6.3 for the DMT transmitter prototype and the receiver prototype, respectively. Three DSPs are required for the transmitter and four DSPs are needed for the receiver. The initial goal of the prototype is to demonstrate the steady state performance of the DMT system. A set of separate software codes are written to perform system initialization off-line and is not described here. Furthermore, the DMT prototype does not include a trellis code, since the Viterbi decoder cannot be implemented efficiently using general purpose DSP, and a programmable custom design Viterbi signal processor is currently being designed by [60]. In this Section, we will describe each of the software functional block briefly.

6.3.1 Encoder

The function of the encoder is to group the binary bits according to the bit assignment, and then form the appropriate QAM symbol for each corresponding QAM subchannel. An arithmetic function is first written to perform the encoding, and due to real-time constraint, the encoder is limited to generate only square QAM signal constellations. As a result, only even number of bits per subchannel are implemented, and the granularity is considered too coarse for the DMT system. A look-up table which encompass the QAM cross constellations as well as the square constellations, enables using all integer bit allocation. Fewer instruction cycles are needed for encoding using a look-up table even with cross constellation, though some memory spaces are sacrificed to store the QAM constellation points.

6.3.2 Inverse FFT and FFT

The most computationally intensive functional blocks of the DMT transceiver system are the FFT/IFFT functions. Very efficient complex-to-complex FFT and IFFT has been previously written for the Motorola DSP 56001. Since the DMT system for DSL applications is not modulated by a carrier frequency, the output symbols of the DMT transmitter must be real. Instead of attempting to write an optimized

real-to-complex FFT function, we exploit the special properties of DFT and use the complex-to-complex FFT codes as the core of the FFT operation.

There are two methods to perform real-to-complex FFT using complex-to-complex FFT. The first method is based on the Hermitian symmetry properties of the discrete Fourier transform, and is described explicitly in [65]:

Double-Blocking FFT/IFFT

To understand the Hermitian symmetry property, we consider the N -point DFT of a real-valued sequence x_m

$$X_k = \sum_{m=0}^{N-1} x_m e^{-j\frac{2\pi mk}{N}} \quad k = 0, \dots, N-1 \quad (6.1)$$

$$= \sum_{m=0}^{N-1} x_m \cos\left(\frac{2\pi mk}{N}\right) - j \sum_{m=0}^{N-1} x_m \sin\left(\frac{2\pi mk}{N}\right), \quad k = 0, \dots, N-1. \quad (6.2)$$

Since x_m is real, equation (6.2) implies that the real part of X_k , $\Re[X_k]$, is even and that the imaginary part of X_k , $\Im[X_k]$, is odd

$$\Re[X_k] = \Re[X_{-k}] = \Re[X_{N-k}], \quad (6.3)$$

and

$$\Im[X_k] = -\Im[X_{-k}] = -\Im[X_{N-k}]. \quad (6.4)$$

Similarly, the N -point DFT of a purely imaginary sequence, jy_m , where y_m is an N -point real-valued sequence, is given by

$$Y_k = j \sum_{m=0}^{N-1} y_m e^{-j\frac{2\pi mk}{N}} \quad k = 0, \dots, N-1 \quad (6.5)$$

$$= j \sum_{m=0}^{N-1} y_m \cos\left(\frac{2\pi mk}{N}\right) + \sum_{m=0}^{N-1} y_m \sin\left(\frac{2\pi mk}{N}\right), \quad k = 0, \dots, N-1. \quad (6.6)$$

Now the real part of Y_k , $\Re[Y_k]$, is odd and the imaginary part of Y_k , $\Im[Y_k]$, is even (call Hermitian anti-symmetry)

$$\Re[Y_k] = -\Re[Y_{-k}] = -\Re[Y_{N-k}] , \quad (6.7)$$

and

$$\Im[Y_k] = \Im[Y_{-k}] = \Im[Y_{N-k}] . \quad (6.8)$$

With the above properties, we can perform N -point DFT on two N -point real-valued sequences x_m and y_m , using a single N -point complex-to-complex FFT. Assume that X_k is the N -point DFT of x_m , and Y_k is the N -point DFT of y_m , then we evaluate the N -point DFT of the sequence $x_m + jy_m$

$$Z_k = \sum_{m=0}^{N-1} (x_m + jy_m) e^{-j\frac{2\pi mk}{N}} , \quad k = 0, \dots, N-1 \quad (6.9)$$

$$= X_k + jY_k , \quad k = 0, \dots, N-1 . \quad (6.10)$$

So

$$\Re[Z_k] = \Re[X_k] - \Im[Y_k] , \quad (6.11)$$

and

$$\Im[Z_k] = \Im[X_k] + \Re[Y_k] . \quad (6.12)$$

Using the Hermitian symmetry and Hermitian anti-symmetry properties described, we have

$$\Re[X_k] = \frac{\Re[Z_k] + \Re[Z_{N-k}]}{2} , \quad (6.13)$$

$$\Im[X_k] = \frac{\Im[Z_k] - \Im[Z_{N-k}]}{2} , \quad (6.14)$$

$$\Re[Y_k] = \frac{\Im[Z_k] + \Im[Z_{N-k}]}{2} , \quad (6.15)$$

and

$$\Im[Y_k] = \frac{\Re[Z_{N-k}] - \Re[Z_k]}{2}. \quad (6.16)$$

Thus the preprocessing function before the FFT simply store the output block of equalized symbols alternatively into the real and imaginary memory spaces. The postprocessing function after the FFT then uses equations (6.13) - (6.16) to retrieve the DFT of the two blocks of symbols.

Similarly, the preprocessing function before the IFFT combines two N -point Hermitian symmetric input sequences U_k and V_k , where $\Re[U_k] = \Re[U_{N-k}]$, $\Im[U_k] = -\Im[U_{N-k}]$, $\Re[V_k] = \Re[V_{N-k}]$, and $\Im[V_k] = -\Im[V_{N-k}]$, into a single N -point sequence W_k by

$$W_k = U_k + jV_k, \quad k = 0, \dots, N-1. \quad (6.17)$$

The postprocessing function after the IFFT simply output the real and the imaginary memory space alternatively, since the IFFT of U_k is in the real memory space and the IFFT of V_k is in the imaginary memory space.

The N -point ($N = 512$) complex-to-complex FFT core function results in an execution time just below the maximum execution time allowed per DSP. Consequently, the extra pre- and post- processing required are passed on to the previous and next DSP, respectively.

Half-Size Single-Block FFT/IFFT

The disadvantage of the double-blocking FFT/IFFT is that it results in twice the latency, and the same cost, though the time to perform the FFT per block is roughly halved, than using an N -point FFT per block.

This second method exploits the nature of the radix-2 FFT algorithm, and uses an $N/2$ -point complex-to-complex FFT to perform an N -point real-to-complex FFT.

A radix-2 FFT algorithm, at the first stage, divides the N -point input sequence x_m into two $N/2$ -point sequences, the even samples, x_{2m} , and the odd sample, x_{2m+1} ,

$$X_k = \sum_{m=0}^{N/2-1} x_{2m} e^{-j \frac{2\pi(2mk)}{N}} + e^{-j \frac{2\pi k}{N}} \sum_{m=0}^{N/2-1} x_{2m+1} e^{-j \frac{2\pi(2mk)}{N}} . \quad (6.18)$$

This approach is called decimation in time, and each $N/2$ -point sequence is decomposed similarly until only 2 points are in each sequence. Each decomposition corresponds to one butterfly stage, and there is a total of $\log_2 N$ stages for an N -point complex-to-complex FFT. Assume the N -point real-valued sequence that we want to perform FFT is denoted by x_m , we define a new $N/2$ -point complex-valued sequence, y_m , by

$$\Re[y_m] = x_{2m} , \quad m = 0, \dots, N/2 - 1 , \quad (6.19)$$

$$\Im[y_m] = x_{2m+1} , \quad m = 0, \dots, N/2 - 1 . \quad (6.20)$$

We now consider the $N/2$ -point complex-to-complex FFT for the sequence y_m

$$Y_k = \sum_{m=0}^{N/2-1} y_m e^{-j \frac{2\pi mk}{N/2}} \quad (6.21)$$

$$= \sum_{m=0}^{N/2-1} x_{2m} e^{-j \frac{2\pi mk}{N/2}} + j \sum_{m=0}^{N/2-1} x_{2m+1} e^{-j \frac{2\pi mk}{N/2}} . \quad (6.22)$$

Compare equation (6.22) to equation (6.18), we found that

$$X_k = \sum_{m=0}^{N/2-1} x_{2m} e^{-j \frac{2\pi mk}{N/2}} + \left(\frac{e^{-j \frac{2\pi k}{N}}}{j} \right) j \sum_{m=0}^{N/2-1} x_{2m+1} e^{-j \frac{2\pi mk}{N/2}} . \quad (6.23)$$

To convert the $N/2$ -point FFT of y_m to the N -point FFT of x_m , we only need to add one more butterfly stage to the radix-2 algorithm.

This method maintains the latency and reduces the amount of execution time by about half, allowing the pre- and post- processing to be done within the same DSP, though the pre- and post- processing are more difficult to understand than the previous method.

6.3.3 Cyclic Prefix

Addition and removal of the cyclic prefix can simply be accomplished by manipulating the updating of the address pointer: the addition of cyclic prefix is done by pointing to the ν taps cyclic prefix before pointing to the whole block of symbols, and the cyclic prefix can be removed simply by skipping ν samples for every $N + \nu$ received samples.

6.3.4 Feedforward Equalizer

The Motorola DSP 56001 provides a multiply-accumulate (MAC) function that performs this type of sums of products extremely efficient. Neglecting overhead in setting up the appropriate pointers, each additional tap of the equalizer only requires one DSP instruction.

6.3.5 One-Tap LMS Filters and Decoder

In order to update the LMS taps, a symbol-level decision has to be performed. We thus cannot separate the decoding function from the one-tap filters into two separate DSPs, since the FIFO interface allows only uni-directional data transfer between DSPs. With real-time constraint, it is possible that these functions may exceed the capacity of one DSP. To solve this problem, once the output block from the FFT is available, the block of symbols (subchannels) is divided into two halves. The first DSP (DSP6) worked on the first half of the subchannels: multiplying the symbols to their corresponding one-tap filters, decoding the symbol levels and bits, and updating the one-tap filters. Similarly, the second DSP (DSP7) worked on the second half of the subchannels.

6.4 Testing of the DMT Prototype

All the software functions for the steady-state performance have been completed. Ten DSP and FIFO boards are fully operational. However, the interface between the data source/sink and the hybrid circuits have not been completed. A preliminary test

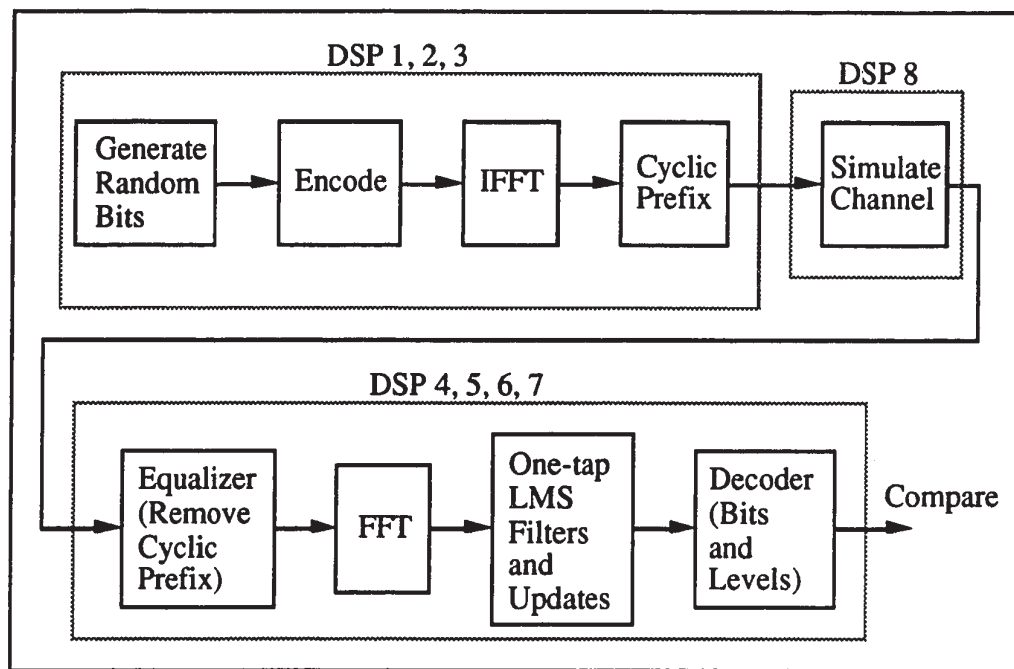


Figure 6.4: DMT prototype test set-up

set-up is shown in Figure 6.4, where channels were simulated digitally using one of the DSPs. Pseudo-random bits were generated internally by the DSP, and noise was not included in the test set-up. We verified that the hardware and software set-up as in Figure 6.4 operated error-free when left running continuously for days.

6.5 Summary

In this Chapter we documented the requirement and design of a real-time prototype of the DMT transceiver. The constraints that leads to the system requirement are described. We found that, in terms of hardware, multiple DSPs are required for the speed and performance we targeted. We identified some of the implemented software functions that needed to be optimized in order to meet the real-time constraint. Although the prototype has not been completed yet, preliminary testing on available hardware verified the functionality of the DMT system. This DMT prototype is especially invaluable for future commercial implementation.

Chapter 7

Conclusions

We began this dissertation by addressing the demand for using the full potential of existing communication channels to transmit higher data rates. Conventional digital communication systems that are currently in commercial use typically achieve a throughput or data rate that is well below the theoretical capacity of these channels. Attempts to achieve near-optimum performance with these conventional systems often results in prohibitive complexity and cost, especially when the channel has severe attenuation and intersymbol interference. This observation encouraged us to investigate alternative digital communication techniques with a better performance/cost tradeoff.

We proposed a specific multi-carrier modulation scheme called discrete multi-tone (DMT) modulation transceiver system. Multi-carrier modulation is based on a frequency-division multiplexing scheme that maximizes performance by selectively assigning different amounts of information to different frequency tones. We previously showed that multi-carrier systems achieve optimum performance with infinite block length or complexity. In this dissertation, we derived an algorithm to maximize data rate with any given finite complexity constraint, and developed a specific structure of the DMT transceiver that uses this algorithm and has reasonable complexity for actual implementation. Applying this finite-complexity DMT transceiver to a set of telephone loops, we found that the T1 transmission rate of 1.6 Mbps on one or two copper twisted-pair wires over 2-3 miles can be easily achieved with a moderate

complexity.

7.1 Summary of Results

In Chapter 2 we described basic multi-carrier modulation techniques that have been proposed previously. Basic baseband and passband communication systems are also described since multi-carrier modulation systems can be visualized as combining a set of passband modulated channels. The discrete Fourier transform (DFT) based multi-carrier system is preferred for actual implementation, since the computation required for modulating a block of N symbols is proportional only to $N \log_2 N$ instead of N^2 for other techniques, and the modulating vectors are fixed for all channels. A cyclic prefix is also added to this DFT-based system to ensure independence between successive blocks of transmitted symbols in the case of channels with intersymbol interference (ISI). Near-optimum performance can be achieved with this system for channels with insignificant ISI.

In Chapter 3 we presented a new multi-carrier modulation system called the discrete multi-tone (DMT) modulation transceiver, which is an enhanced version of the DFT-based system in Chapter 2. Unlike the DFT-based system that performs poorly for channels with severe ISI, the DMT system can achieve near-optimum performance with these channels, with a given complexity constraint for both systems. We proved that the DMT system is optimum with infinite complexity, and we derived an algorithm for finite complexity DMT system. The finite-length optimization algorithm maximizes achievable data rate for any given complexity constraint, and is an essential design tool for actual implementation of the DMT system. The structure of the DMT transmitter is identical to the DFT-based system. The key feature is the addition of the time-domain feedforward equalizer at the DMT receiver, which effectively shortens the channel response to allow a short cyclic prefix to be used and at the same time maintains near-optimum performance. A set of one-tap, complex, adaptive filters is also added to adjust the gain and phase of each frequency subchannel, so that a common decision element can be used for all subchannels. Moreover, the inaccuracy of the channel estimation and the variation of the channel and noise characteristics

can be accounted through this set of adaptive filters. The performance analysis of the DMT system is also derived and closely parallels the development of the passband system. A system start-up initialization is required to perform channel identification, to compute the equalizer taps, to calculate the signal to noise ratio (SNR), and to evaluate the bit and power allocation. The bit assignment is then sent back to the transmitter before actual data transmission begins. Upon estimating the complexity of the DMT system based on million instruction per second (MIPS) measurement, we showed that the DMT transceiver can be reasonably implemented for practical systems.

In Chapter 4 we demonstrated the desirability of the DMT system when we considered applying the DMT system, as an illustrative example, to a set of twisted-pair transmission lines that are representative of digital subscriber line (DSL) applications. The dominant noise impairment is the crosstalk noise, which is dependent on the input signal power of the crosstalking wires. An additive Gaussian white noise (AWGN) floor is also added to model other noise impairments including residual echo, thermal, and electronics noise. We found that the addition of a trellis code, higher input power, longer block length, higher sampling rate (but not excessively high) increase the throughput in general. We concluded that, with the DMT transceiver system, most of the subscriber loops can achieve the target data rate of 1.6 Mbps full-duplex on two twisted-pair wires with significant performance margin, and can achieve the target data rate on a single twisted-pair with a suitable trellis code.

In Chapter 5 we identified some of the finite-length effects that was not reflected in the idealized performance results in Chapter 4. The rate loss caused by the addition of a cyclic prefix is the most obvious effect and can easily be included after we fix the block length and the number of cyclic prefix. We showed that this rate loss is negligible (only about 0.2 dB) for the DSL application when a block length of 512 is used. We identified another important finite-length effects caused by a rapidly varying channel frequency response such as one with frequency notches. A sharp notch at D.C., for example, invalidates the ISI-free assumption in deriving the SNR for the subchannels near D.C.. We derived two techniques to model this effect and to evaluate the SNR in a more accurate way. A set of computer simulation programs is written to account

for this effect and other more subtle finite length effects. Based on the simulation results, we found that about 0.5 dB of the performance margin is loss caused by these effects, not including the rate loss due to cyclic prefix. With a block length of 512, the projected performance results for the HDSL application in Chapter 4 is within 1 dB of the actual performance as evaluated by computer simulations. We also compared the DMT system to previously proposed multi-carrier systems using computer simulation results and found that the DMT system has better achievable data rate and much lower latency and memory requirements, with comparable complexity.

In Chapter 6 we described a real-time prototype of the DMT system that we have implemented at Stanford University as a proof-of-concept system that goes beyond computer simulations. The prototype also gives a realistic picture of the tremendous details involved that is not apparent either in theoretical derivation or computer simulation. The prototype is based on multiple boards of general purpose digital signal processors, connected serially through a custom designed interface board. Most of the functional blocks of the DMT system are implemented through DSP software codes. The time and complexity constraints make the optimization of software code critical, and methods of optimizing the “bottleneck” function, the FFT/IFFT, has been developed. These methods allow the use of an optimized complex-to-complex FFT code to perform real-to-complex FFT, and reduces the run-time cycle by nearly a factor of two. Other functions are similarly optimized. We demonstrated the feasibility of actual implementation of the DMT system, and provided a valuable platform for converting the DMT system to commercial products.

7.2 Continuing Research Topics

In this dissertation we have presented a particular multi-carrier modulation system that achieves near optimum performance with reasonable complexity. When finite complexity constraint is imposed, there may exist other multi-carrier systems that achieve the same or even better performance, given the same complexity constraint. In this regard, the optimum finite complexity multi-carrier system remains to be an open question, though various methods to achieve near-optimum performance

with reasonable complexity have been proposed. In particular, we believe that the finite-length optimization algorithm presented in this dissertation can be applied to sequence detectors such as Viterbi decoder to optimize performance for single-carrier systems. The length of the channel target, ν , becomes critical since the complexity of the Viterbi decoder grows exponentially with respect to ν , as the complexity is proportional to M^ν , where M is the number of levels for the signal constellation. We can see that the finite-length algorithm is more valuable when $M = 2$, as in the case of binary communication channels such as magnetic recording channels.

We have assumed throughout this dissertation that the channel of interest can be correctly estimated. However, the area of channel identification remains to be solved. An important study would be to identify the performance degradation caused by channel mis-estimation, for both single- and multi- carrier systems. Although the set of adaptive filters at the receiver of the DMT system alleviates this problem somewhat, it is not clear how much channel mis-estimation the system can sustain.

We estimated the complexity of the DMT system in terms of million instruction per second (MIPS), based on the assumption that most of the functional blocks of the DMT system are implemented in general purpose DSPs. Depending on the actual method of implementation, other measures for estimating the complexity may be more appropriate. For example, if the DMT system is implemented in semi-custom VLSI designs such as using BiCMOS gate-array, then the complexity measure is likely to be a function of the number of chips required, power consumption, overall size, and cost.

The finite-length algorithm presented in this dissertation maximize achievable data rate based on minimum mean-square error criterion. Computer simulation results showed that the actual performance using the algorithm is indeed close to the optimum. Nevertheless, we do not preclude the possibility that an even better criterion may exist, though the improvement in throughput given a fixed complexity constraint will only be marginal, since the current performance level is already very close to the optimum level.

The optimum bit allocation requires using non-integer number of bits for the sub-channels, for a given bit error rate. If the dominant noise source is not crosstalk noise,

we can then adjust the input power of the subchannels (thereby modifying the SNR) to confine the bit assignment to integer bits. However, when the dominating noise impairment is crosstalk noise, then an increase in input power causes a corresponding increase in crosstalk noise power (SNR remains fixed). Although applying a trellis code partially solve this problem (since some trellis codes allow half bit and quarter bit per two dimensions), a general fractional bit algorithm remains to be solved.

We have focused on applying the DMT system to telephone lines application in this dissertation. The DMT system could potentially be applied to other applications such as magnetic storage channels, mobile wireless communications, and satellite communications. Different system requirements as well as different channels and noise characteristics may change the architecture of the DMT system and need to be carefully investigated.

Appendix A

Unequal SNR for Real and Imaginary Parts

Assuming we assign equal energy on the real and imaginary part of each frequency band, then the SNR for the real and imaginary part will be the same iff the variance of the noise in each component is the same for a particular frequency band.

Consider the effects of Additive white Gaussian Noise (AWGN) on the DMT system, as in Figure 3.5, by setting $x_k = 0$, i.e., $y_k = n_k$. Denote each noise sample by n_k , then $E[n_k] = 0$, and

$$E[n_k^2] = \sigma_n^2 \quad (\text{A.1})$$

$$E[n_k n_l] = 0, \quad k \neq l. \quad (\text{A.2})$$

Let the $(M+1)$ -tap feedforward equalizer be W , where

$$W = w_0 + w_1 D + w_2 D^2 + \dots + w_M D^M, \quad (\text{A.3})$$

then

$$z_k = \sum_i n_i w_{k-i} \quad (\text{A.4})$$

and we have

$$E[z_k] = 0 \quad (\text{A.5})$$

$$E[z_k^2] = \sigma_n^2 \sum_{i=0}^M w_i^2 \triangleq \sigma_z^2 \quad (\text{A.6})$$

$$E[z_k z_{k-l}] = \sigma_n^2 \sum_i w_i w_{i+l} \quad (\text{A.7})$$

Denote the input block to the N-point FFT as $\vec{z}$, where

$$\vec{z} = \begin{bmatrix} z_0 & z_1 & \dots & z_{N-1} \end{bmatrix}, \quad (\text{A.8})$$

Then the output block $\vec{Z}$ is:

$$\vec{Z} = \text{FFT}\{\vec{z}\} = \begin{bmatrix} Z_0 & Z_1 & \dots & Z_{N-1} \end{bmatrix}, \quad (\text{A.9})$$

where

$$Z_m = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} z_i e^{-j \frac{2\pi i m}{N}} \quad (\text{A.10})$$

and

$$Z_m = \frac{1}{\sqrt{N}} \left\{ \begin{aligned} & z_0 \cos \frac{2\pi(0)m}{N} + z_1 \cos \frac{2\pi(1)m}{N} + \dots + z_{N-1} \cos \frac{2\pi(N-1)m}{N} \\ & + j \left[-z_1 \sin \frac{2\pi m}{N} - \dots - z_{N-1} \sin \frac{2\pi(N-1)m}{N} \right] \end{aligned} \right\} \quad (\text{A.11})$$

Let $Z_m = RZ_m + jIZ_m$, the real and imaginary part, then

$$E[RZ_m] = E[IZ_m] = 0 \quad (\text{A.12})$$

and

$$E[RZ_m^2] = \frac{1}{N} \sum_{i=0}^{N-1} E[z_k z_{k-i}] U_{m,i} \quad (\text{A.13})$$

where

$$U_{m,0} = \cos^2 \left(\frac{2\pi(0)m}{N} \right) + \cos^2 \left(\frac{2\pi(1)m}{N} \right) + \dots + \cos^2 \left(\frac{2\pi(N-1)m}{N} \right) \quad (\text{A.14})$$

$$U_{m,i} = 2 \left[\cos \left(\frac{2\pi(0)m}{N} \right) \cos \left(\frac{2\pi(0+i)m}{N} \right) + \cos \left(\frac{2\pi(1)m}{N} \right) \cos \left(\frac{2\pi(1+i)m}{N} \right) \right. \\ \left. + \dots + \cos \left(\frac{2\pi(N-1-i)m}{N} \right) \cos \left(\frac{2\pi(N-1)m}{N} \right) \right], \quad 1 \leq i \leq N-1, \quad (\text{A.15})$$

similarly

$$\mathbb{E} [IZ_m^2] = \frac{1}{N} \sum_{i=0}^{N-1} \mathbb{E} [z_k z_{k-i}] V_{m,i} \quad (\text{A.16})$$

where

$$V_{m,0} = \sin^2 \left(\frac{2\pi(0)m}{N} \right) + \sin^2 \left(\frac{2\pi(1)m}{N} \right) + \dots + \sin^2 \left(\frac{2\pi(N-1)m}{N} \right) \quad (\text{A.17})$$

$$V_{m,i} = 2 \left[\sin \left(\frac{2\pi(0)m}{N} \right) \sin \left(\frac{2\pi(0+i)m}{N} \right) + \sin \left(\frac{2\pi(1)m}{N} \right) \sin \left(\frac{2\pi(1+i)m}{N} \right) \right. \\ \left. + \dots + \sin \left(\frac{2\pi(N-1-i)m}{N} \right) \sin \left(\frac{2\pi(N-1)m}{N} \right) \right], \quad 1 \leq i \leq N-1. \quad (\text{A.18})$$

It is easy to show that

$$U_{m,0} = V_{m,0} = \frac{N}{2}, \quad m \neq 0, \frac{N}{2}, \quad (\text{A.19})$$

$$U_{m,0} = N, V_{m,0} = 0, \quad m = 0, \frac{N}{2}, \quad (\text{A.20})$$

($m = 0$ or $\frac{N}{2}$ corresponds to D.C. and Nyquist respectively, and $IZ_m = 0$ exactly because of Hermitian symmetry), and using trigonometry identity,

$$U_{m,i} = (N-i) \cos \left(\frac{2\pi(i)m}{N} \right) + T_{m,i}, \quad 1 \leq i \leq N-1, \quad (\text{A.21})$$

$$V_{m,i} = (N-i) \cos \left(\frac{2\pi(i)m}{N} \right) - T_{m,i}, \quad 1 \leq i \leq N-1, \quad (\text{A.22})$$

where

$$T_{m,i} = \sum_{n=1}^{N-i} \cos \left(\frac{2\pi(i+2n-2)m}{N} \right) \quad 1 \leq i \leq N-1, \quad (\text{A.23})$$

We can thus conclude that

$$E[RZ_m^2] = E[IZ_m^2] \text{ iff } \sum_{i=1}^{N-1} E[z_k z_{k-i}] T_{m,i} = 0 \quad (\text{A.24})$$

Obviously if z_k is uncorrelated (that is, if we have an 1-tap FIR equalizer W), then condition in Equation (A.24) is true, and the noise variance is the same for both the real and the imaginary part. However, in general, the noise variance will not be equal on the real and imaginary part.

Similar results apply to crosstalk noise, and so in general the SNR for the real and imaginary part in any particular frequency band are not necessary the same.

The results above led to the following Theorem:

Theorem 1 *For any zero-mean i.i.d. process $\{z_k\}$, the variance of the real and imaginary part of Z_m are the same for $m \neq 0, N/2$, where $\{Z_m\}$ is the N -point DFT of $\{z_k\}$, and $m = 0$ corresponds to D.C.*

Theorem 2 *For any zero-mean correlated process $\{z_k\}$ with $E[z_k^2] = \sigma_z^2$, the variance of the real and imaginary part of Z_m are the same for $m \neq 0, N/2$ iff*

$\sum_{i=1}^{N-1} E[z_k z_{k-i}] \sum_{n=1}^{N-i} \cos \left(\frac{2\pi(i+2n-2)m}{N} \right) = 0$, where $\{Z_m\}$ is the N -point DFT of $\{z_k\}$, and $m = 0$ corresponds to D.C.

Appendix B

Average Energy with Cyclic Prefix

Let the input to the IFFT be denoted by u_k and v_k for the independent real and imaginary part, respectively. With a block length of N , the block of data symbol going to the Inverse FFT is denoted by $\vec{G}$:

$$\vec{G} = \begin{bmatrix} u_0 + jv_0 & u_1 + jv_1 & u_2 + jv_2 & \dots & u_{\frac{N}{2}} + jv_{\frac{N}{2}} & \dots & u_{N-1} + jv_{N-1} \end{bmatrix} \quad (\text{B.1})$$

Since the input data to the channel needs to be real, Hermitian symmetry is required:

$$v_0 = v_{\frac{N}{2}} = 0 \quad (\text{B.2})$$

$$u_i = u_{N-i}, \quad v_i = -v_{N-i}, \quad i = 1, 2, \dots, \frac{N}{2} - 1 \quad (\text{B.3})$$

$E[u_k]$ and $E[v_k]$ are zero, and for $k, l \leq \frac{N}{2}$:

$$E[u_k u_l] = E[v_k v_l] = 0, \quad k \neq l \quad (\text{B.4})$$

$$E[u_k v_l] = 0 \quad (\text{B.5})$$

We also denote the average energy as:

$$\mathbb{E}[u_k^2] = \sigma_{u_k}^2 \quad \mathbb{E}[v_k^2] = \sigma_{v_k}^2 \quad (\text{B.6})$$

The output of the Inverse FFT, $\vec{X}$, is then :

$$\vec{X} = \text{IFFT}\{\vec{G}\} = [X_0 \ X_1 \ \dots \ X_{N-1}] , \quad (\text{B.7})$$

where

$$X_m = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} (u_k + jv_k) e^{j\frac{2\pi km}{N}} \quad (\text{B.8})$$

Applying Equation (B.2) and (B.3) and simplifying, we obtain:

$$X_m = \frac{1}{\sqrt{N}} \left[u_0 + u_{\frac{N}{2}} e^{j\pi m} + 2 \sum_{i=1}^{\frac{N}{2}-1} \left(u_i \cos \frac{2\pi im}{N} - v_i \sin \frac{2\pi im}{N} \right) \right] , \quad (\text{B.9})$$

and so $\mathbb{E}[X_m] = 0$ from Equations (B.4) and (B.5). And

$$\mathbb{E}[X_m^2] = \frac{1}{N} \left\{ \sigma_{u_0}^2 + \sigma_{u_{\frac{N}{2}}}^2 + 4 \sum_{i=1}^{\frac{N}{2}-1} \left[\sigma_{u_i}^2 \cos^2 \left(\frac{2\pi im}{N} \right) + \sigma_{v_i}^2 \sin^2 \left(\frac{2\pi im}{N} \right) \right] \right\} \quad (\text{B.10})$$

Define

$$\mathcal{E}_x \triangleq \frac{1}{N} \sum_{i=0}^{N-1} \mathbb{E}[X_i^2] = \frac{1}{N} \left[\sigma_{u_0}^2 + \sigma_{u_{\frac{N}{2}}}^2 + 2 \sum_{i=1}^{\frac{N}{2}-1} (\sigma_{u_i}^2 + \sigma_{v_i}^2) \right] \quad (\text{B.11})$$

For general case, that is, if $\sigma_{u_k}^2 \neq \sigma_{v_k}^2$ for some k , then the sum of the energy of any ν consecutive samples of $\vec{X}$ is

$$\begin{aligned} \sum_{i=M}^{M+\nu-1} \mathbb{E}[X_i^2] &= \frac{1}{N} \left\{ \nu (\sigma_{u_0}^2 + \sigma_{u_{\frac{N}{2}}}^2) + 4 \sum_{l=1}^{\frac{N}{2}-1} \right. \\ &\quad \left. \left[\sigma_{u_l}^2 \sum_{n=M}^{M+\nu-1} \cos^2 \left(\frac{2\pi ln}{N} \right) + \sigma_{v_l}^2 \sum_{n=M}^{M+\nu-1} \sin^2 \left(\frac{2\pi ln}{N} \right) \right] \right\} \quad (\text{B.12}) \end{aligned}$$

where M is the starting point of the ν samples.

There is no closed form expression for arbitrary values of ν and M , however, for sufficient large N (such as 64 and above) and $\nu \geq 2$, then

$$\sum_{i=M}^{M+\nu-1} E[X_i^2] \approx \frac{1}{N} \left\{ \nu (\sigma_{u_0}^2 + \sigma_{u_{\frac{N}{2}}}^2) + 2\nu \sum_{l=1}^{\frac{N}{2}-1} [\sigma_{u_l}^2 + \sigma_{v_l}^2] \right\} = \nu \mathcal{E}_x \quad (\text{B.13})$$

so the average energy of a block of $N + \nu$ samples $\vec{X}_{CE}$ (after cyclic extension by ν samples) is given by:

$$E[\vec{X}_{CE} \vec{X}_{CE}^T] \approx (N + \nu) \mathcal{E}_x \quad (\text{B.14})$$

where $\vec{X}_{CE}$ is a row vector.

To ensure the average energy of the block $\vec{X}_{CE}$ is strictly less than a given energy constraint, we can define $\sigma_{t_k}^2$ as the larger of $\sigma_{u_k}^2, \sigma_{v_k}^2$:

$$\sigma_{t_k}^2 = \max \{ \sigma_{u_k}^2, \sigma_{v_k}^2 \} \quad (\text{B.15})$$

then

$$E[X_m^2] \leq \frac{1}{N} \left[\sigma_{t_0}^2 + \sigma_{t_{\frac{N}{2}}}^2 + 4 \sum_{i=1}^{\frac{N}{2}-1} \sigma_{t_i}^2 \right] \triangleq \mathcal{E}_t \quad (\text{B.16})$$

and

$$E[\vec{X}_{CE} \vec{X}_{CE}^T] \leq N \mathcal{E}_x + \nu \mathcal{E}_t \quad (\text{B.17})$$

A simple and fairly tight bound for sufficiently large N .

In the case of equal energy among the real and imaginary part, $\sigma_{u_k}^2 = \sigma_{v_k}^2$:

$$E[X_m^2] = \frac{1}{N} \left[\sigma_{u_0}^2 + \sigma_{u_{\frac{N}{2}}}^2 + 4 \sum_{i=1}^{\frac{N}{2}-1} \sigma_{u_i}^2 \right] = \mathcal{E}_x \quad (\text{B.18})$$

independent of m . This means that each sample at the output of the IFFT has the same average energy $\mathcal{E}_x$, and

$$\mathbb{E} [\vec{X}_{CE} \vec{X}_{CE}^T] = (N + \nu) \mathcal{E}_x, \quad (\text{B.19})$$

which means that the cyclic prefix does not change the energy constraint.

Appendix C

Listings of Matlab Programs

This Appendix lists the **Matlab** programs that are used to perform the Monte-Carlo simulations to accurately project the performance results of the DMT system (see Section 5.3).

C.1 Matlab program *nextfilter.m*

```
% Use this program to find a filter for simulation NEXT
% NEXT =  $10^{(-13)} * f^{(1.5)}$ 

f=[0:200:640e3];

F=f/max(f);

wsq= $10^{(-13)} * f.^{(1.5)}$ ;

w=sqrt(wsq);

[nextz,nextp]=yulewalk(79,F,w);

save nextfilter nextz nextp;
% nextz is the zero-polynomial of the NEXT filter
% nextp is the pole-polynomial of the NEXT filter
```

C.2 Matlab program *nexts.m*

```
% This script gives samples of crosstalk noise
%
% n is the blocklength
%
% Cyclic Extended by 8
%
% use with nextfilter.m
%

load nextfilter;

numofit=160000;
rand('seed',123456);

len=9;      % Number of cyclic prefix = len-1
n=512; % n = block length

sta1=zeros(1,length(nextz)-1);
temp=rand(1,numofit);
xnoise=zeros(1,numofit/n*(n+len-1));
for i=1:numofit,
    if temp(i)>=0.5,
        vin(i)=1;
    else
        vin(i)=-1;
    end;
end;
clear temp;
for j=1:numofit/n,

% Note that the DC and Nyquist channels are NOT used

tvin=vin(1+(j-1)*n:(n/2-1)+(j-1)*n)+sqrt(-1)*vin(n/2+(j-1)*n:j*n-2);
% the last vin above goes from n/2+(j-1)*n to n-2+(j-1)*n
vinidft=[0 tvin 0 conj(tvin((n/2-1):-1:1))];
cinbce=real(ifft(vinidft));
cin=[cinbce(n-len+2:n) cinbce];
% Here is the addition of cyclic prefix
```

```

    sta=sta1;

    [xout,sta1]=filter(nextz,nextp,cin,sta);

% xout is the NEXT noise for each block

    xnoise(1+(j-1)*(n+len-1):j*(n+len-1))=xout;
end;

clear vin;

save nexts xnoise;
% xnoise is the simulated NEXT noise

```

C.3 Matlab program *ftot.m*

```

% Convert frequency response to time response
% Blocklength = 512

load chres.freq;

n=512;

temp=chres(:,1)+sqrt(-1)*chres(:,2);
% The first column of chres contains the real part of the response
% The second column of chres contains the
% imaginary part of the response

ttemp=[temp.' conj(temp(n/2:-1:2)).'];

tres=real(ifft(ttemp));

save ftot1 tres;
% tres is the channel pulse response (in time-domain)

```

C.4 Matlab program *noiseauto.m*

```
% To calculate noise autocorrelation sequence
% Blocklength = 512

rt=640e3;
% rt = sampling rate

n=512;
% n = blocklength

px=10;
% Input power = 10 mW

inpsd640k=px/rt;           % Input PSD=Input Power/Rate;
f640k=[0:rt/2/512:rt/2];
m=max(size(f640k));

for kk=1:m,
    fxtmr640k(kk)=10^(-13)*(f640k(kk)^(1.5));    % NEXT
end;

tnpsd640k=1e-11*ones(fxtmr640k)+inpsd640k*fxtmr640k;
% The first part is for AWGN (10^(-11) mW/Hz),
% and the second part is the NEXT

tempf=[tnpsd640k tnpsd640k(512:-1:2)];
xwnoise=rt/px*real(iff(tempf));

save noiseauto xwnoise;
% xwnoise contains the noise autocorrleation (AWGN + NEXT)
```

C.5 Matlab program *computetaps.m*

```
% This program computes the taps settings for the
% feedforward equalizer
```

```

load ftot1;
load noiseauto;
M=9;
% M+1 = Number of Feedforward Taps
N=8;
% N = Number of Cyclic Prefix
noise=xwnoise(1:M+1);
p=tres;

clear tres xwnoise;

dfesnrdB=zeros(1,N+1);
taps=zeros(N+1,M+1+N);
delta=zeros(1,N+1);

dfesnrdB2=zeros(1,N+1);
taps2=zeros(N+1,M+1+N);
delta2=zeros(1,N+1);

for i=0:N,
    for d=0:30,

        [a,w]=dfe1(p,M,N,d,i,noise);
        if dfesnrdB(i+1)<a,
            dfesnrdB(i+1)=a;
            taps(i+1,:)=w;
            delta(i+1)=d;
        end;
    end;

    for dd=0:i,
        [aa,ww]=dfe2(p,M,N,dd,i,noise);
        if dfesnrdB2(i+1)<aa,
            dfesnrdB2(i+1)=aa;
            taps2(i+1,:)=ww.';
            delta2(i+1)=dd;
        end;
    end;
end;

```

```

save dfeeci1x;
% The Taps are now contain in taps and taps2,
% and the delay in delta and delta2

```

C.6 Matlab program *deltap.m* and *deltan.m*

C.6.1 Matlab program *deltap.m*

```

function [dfesnrdB,w]=dfe1(p,M,N,d,i,noise);
% inputs: pulse response
%   M+1 taps for w, N taps for b, d=delay
%   0 <= i <= N
%   i=0 corresponds to b completely causal (regular DFE)
%   i=N corresponds to b completely anticausal
%   one-sided noise correlation sequence (size M+1)
% outputs: dfesnr (in dBs), w
% Corresponds to e=bx{k-Delta}-wy{k}

nff=M+1;
nfb=N;
p=row2col(p);
L=length(p)-1;
P=toeplitz([p(1)';zeros(nff-1,1)],[p',zeros(1,nff-1)]);
Rn=toeplitz(noise);
R11=P*P' + Rn;
R22=eye(nfb);
top=d+1; bot=min(nff+L,nfb+d+1);
B=zeros(nff+L,nfb+1);
if ((nfb > 0) & (bot>=top)),
    preii=eye(bot-top+1);
    prei=[preii(1:min(i,bot-top+1),:); ...
zeros(1,bot-top+1); preii(i+1:(bot-top),:)]';
B(top:bot,1:bot-top+1)=prei(1:bot-top+1,1:bot-top+1);
end
R12=P*B(:,1:nfb);
R=[R11,R12;R12',R22];
temp=zeros(nff+L,1);

```

```

if (d+i)<(nff+L),
    temp(d+i+1)=1;
end;
rho=[P*temp;zeros(nfb,1)];
w=(inv(R)*rho)';
sig=1-rho'*inv(R)*rho;

dfesnrdB=10*log10((1-sig)/sig);

```

C.6.2 Matlab program *deltan.m*

```

function [dfesnrdB,w]=dfe2(p,M,N,d,i,noise);
% inputs: pulse response
%   M+1 taps for w, N taps for b, d=delay
%   0 <= i <= N, d <= i
%   i=0 corresponds to b completely causal (regular DFE)
%   i=N corresponds to b completely anticausal
%   one-sided noise correlation sequence (size M+1)
% outputs: dfesnr (in dBs), w
% Corresponds to e=bx{k}-wy{k-Delta}

nff=M+1;
nfb=N;
p=row2col(p);
L=length(p)-1;
P=toeplitz([p(1)';zeros(nff-1,1)],[p',zeros(1,nff-1)]);
Rn=toeplitz(noise);
R11=P*P' + Rn;
R22=eye(nfb);
B=zeros(nff+L+d,nfb);
prei=eye(nfb);
B(1:nfb+1,:)=[prei(1:i,:); zeros(1,nfb); prei(i+1:nfb,:)];
R12=P*B(d+1:d+nff+L,1:nfb);
R=[R11,R12;R12',R22];
temp2=zeros(nff+L+d,1);
if i<(nff+L+d),
    temp2(i+1)=1;
end;

```

```

temp=temp2(d+1:d+L+nff);
rho=[P*temp;zeros(nfb,1)];
w=inv(R)*rho;
sig=1-rho'*inv(R)*rho;

dfesnrdB=10*log10((1-sig)/sig);

```

C.7 Matlab program *findsnr.m*

```

% Using Channel 16 rate 640k
% This script uses IDFT and DFT vectors
% n is the blocklength
% With AWGN and NEXT,
% With cyclic extension at the beginning of the block
% to account for all zeros
% WITH SNR Calculation
% NOTE Two Passes
% Pass one: Identify optimal 1-Tap Settings
% Pass two: Obtain SNR
% 4-level input
% Note: D.C. and Nyquist channels are not used

load dfec16r640k;
load nexts;
i=0;
wf=taps(i+1,1:M+1);
delay=delta(i+1);

load rand4l;
load pz16r640k1;

pass1=40000;
pass2=40000;
% pass1 and pass2 are the loop numbers

% For 4-level PAM, with energy = 1 per dimension
fl=sqrt(0.8);

```

```

pts=[-3/2*f1:f1:3/2*f1];
% For 4-level

len=N+1; % Number of cyclic prefix + 1

n=512; % Blocklength
ff=length(wf); % Number of feedforward tap=M+1

scalesig=sqrt(1e-11*640e3*255/256/5); % AWGN, spst(Variance)

rand('normal'); % Normal Distribution
rand('seed',1);

snr=zeros(1,n); % this will be the final snr

h=filter(polyz, polyp, [1 zeros(1,n-1)]);
% h=channel time response.
% polyz, polyp are the zeros and poles polynomial for the channel

sta1=zeros(1,max(length(polyp),length(polyz))-1);
stb1=zeros(1,length(wf)-1);
% set the number of states for the channel

errs=zeros(1,n); % Initialization for computing SNR
fh=fft(h);
fw=fft([wf zeros(1,n-ff)]);

ntfcheck=ones(1,n/2-1)./(fh(2:n/2).*fw(2:n/2));
% Initial Taps setting for the 1-tap complex LMS filters

% invflmd(2:n/2)=ntfcheck;
invflmd(2:n/2)=zeros(1,n/2-1);
% invflmd is the 1-tap LMS filters
% we will compute its value at the end of first pass

% confact=0.02*ones(1,n);
% Stepsizes for the LMS filters
% we will NOT update the weights in this program

mb=130000;

```

```

ms=3*n/2;
% Base of Modulo, to save the length of vin (to save memory space)

% the following is for adding delay
firout=zeros(1,n+len-1);
vinidft=zeros(1,n);

for j=1:pass1+1,      % Note that DC and Nyquist channels are NOT used
% pass1 is an arbitrary number for the number of blocks simulated

    vinbuf=vinidft;
% First Pass does not Count

    if modrnz(1+(j-1)*n,mb)>modrnz(j*n-2,mb),

        rtvin=vin(modrnz(1+(j-1)*n+ms,mb):modrnz((n/2-1)+(j-1)*n+ms,mb));
        itvin=sqrt(-1)*vin(modrnz(n/2+(j-1)*n+ms,mb):modrnz(j*n-2+ms,mb));
    else
        rtvin=vin(modrnz(1+(j-1)*n,mb):modrnz((n/2-1)+(j-1)*n,mb));
        itvin=sqrt(-1)*vin(modrnz(n/2+(j-1)*n,mb):modrnz(j*n-2,mb));
    end;
    tvin=rtvin+itvin;
% rtvin=real part
% itvin=imaginary part

    vinidft=[0 tvin 0 conj(tvin((n/2-1):-1:1))];
% create Hermitian symmetry

% Here is the addition for delay
% These are used as desired response (Assume we have the actual bit,
% as during training period)

    cinbce=real(ifft(vinidft));
% Perform IFFT

    cin=[cinbce(n-len+2:n) cinbce];
% Here is the addition of cyclic prefix

    sta=sta1;
    stb=stb1;

```

```

% update the states

[cout,sta1]=filter(polyz,polyp,cin,sta);
% channel
% cout = [Yk-n+1 . . . Yk-1 Yk Yk+1 Yk+2],
% where the first len-1 are the cyclic extended output

gnoise=rand(1,n+len-1)*scalesig/sqrt(n);
% Divided by sqrt(n) since IFFT reduces energy by n

cout=cout+gnoise+xnoise;
% AWGN and NEXT is added to channel output

inada=firout;
[firout,stab1]=filter(wf,1,cout,stab);
% firout is the output of the FIR equalizer at the receiver
% The following is to take into account the effect of delay
inada=[inada firout];
% inada consists of two blocks, the first one from previous run
% the second from current run
outada=inada(len+delay:n+len-1+delay);

% Here is the Starting point for finding SNR

tindec=fft(outada);
% Performs FFT

if j>=2,
    for k=2:n/2,
        invf1md(k)=(vinbuf(k)/tindec(k))+invf1md(k);
    end;
end;
% the above lines are used to average the 1-Tap settings

end;
% end the for loop

invf1md=invf1md/pass1;

% at this point in time we have the 1-Tap equalizer settings

```

```

% coffset is a constant offset so that the input
% will not be identical
coffset=43219;

for j=1+coffset:coffset+pass2+1,
% Note that DC and Nyquist channels are NOT used
% pass2 is an arbitrary number for the number of blocks simulated

    vinbuf=vinidft;
% First Pass does not Count

    if modrnz(1+(j-1)*n,mb)>modrnz(j*n-2,mb),

        rtvin=vin(modrnz(1+(j-1)*n+ms,mb):modrnz((n/2-1)+(j-1)*n+ms,mb));
        itvin=sqrt(-1)*vin(modrnz(n/2+(j-1)*n+ms,mb):modrnz(j*n-2+ms,mb));
    else
        rtvin=vin(modrnz(1+(j-1)*n,mb):modrnz((n/2-1)+(j-1)*n,mb));
        itvin=sqrt(-1)*vin(modrnz(n/2+(j-1)*n,mb):modrnz(j*n-2,mb));
    end;
    tvin=rtvin+itvin;
% rtvin=real part
% itvin=imaginary part

    vinidft=[0 tvin 0 conj(tvin((n/2-1):-1:1))];
% create Hermitian symmetry

% Here is the addition for delay
    rvinidft=real(vinbuf);
    ivinidft=imag(vinbuf);
% These are used as desired response (Assume we have the actual bit,
% as during training period)

    cinbce=real(ifft(vinidft));
% Perform IFFT

    cin=[cinbce(n-len+2:n) cinbce];
% Here is the addition of cyclic prefix

    sta=sta1;
    stb=stb1;

```

```

% update the states

[cout,sta1]=filter(polyz,polyp,cin,sta);
% channel
% cout = [Yk-n+1 . . . Yk-1 Yk Yk+1 Yk+2],
% where the first len-1 are the cyclic extended output

gnoise=rand(1,n+len-1)*scalesig/sqrt(n);
% Divided by sqrt(n) since IFFT reduces energy by n

cout=cout+gnoise+xnoise;
% AWGN + NEXT is added to channel output

inada=firout;
[firout,stab1]=filter(wf,1,cout,stab);
% firout is the output of the FIR equalizer at the receiver
% The following is to take into account the effect of delay
inada=[inada firout];
% inada consists of two blocks, the first one from previous run
% the second from current run
outada=inada(len+delay:n+len-1+delay);

% Here is the Starting point for finding SNR

tindec=fft(outada);
% Performs FFT

indec(2:n/2)=tindec(2:n/2).*invfimd(2:n/2);
% Performs the 1-Tap Filters
% indec(1)=0, and we don't use the D.C. channel

rindec=real(indec);
iindec=imag(indec);

if j>=coffset+2,
    for k=2:n/2,
        errs(k)=(rindec(k)-rvinidft(k))^2+errs(k);
    end;
    for k=2:n/2,
        errs(n/2+k)=(iindec(k)-ivinidft(k))^2+errs(n/2+k);

```

```

        end;
    end;          % end the if
% the above lines are used to compute and accumulated the
% mean-square error (noise)

% We use the CORRECT input as decision
% This is for LMS update
% if j>1,
% ev(2:n/2)=vinbuf(2:n/2)-indec(2:n/2);
% invfimd(2:n/2)=invfimd(2:n/2)+
%   confact(2:n/2).*(ev(2:n/2).*conj(tindec(2:n/2)));
% end;
% Don't do Updating the weights

end;
% end the for loop

% This is the Beginning for finding SNR after the loop

errs(2:n)=errs(2:n)./pass2;
% Average the noise

snr2(2:n)=ones(1,n-1)./errs(2:n);
% Compute SNR

snr=(snr2(2:n/2)+snr2(n/2+2:n))/2;
% Average the Real and Imaginary SNRs

% This is the End for finding SNR after the loop

clear vin;

rrate=640e3/(512+len-1)*sum(log(1 + snr / (10^(0.98))))/log(2);

ssnr=sort(snr);
for i=1:255,
    nsnr=ssnr(255-i+1:255);
    nsnr=(255/i)*nsnr;
    orate(i)=640e3/(512+N)*sum(log(1 + nsnr / (10^(0.98))))/log(2);
end;

```

```

mrate=max(orate);

save mcssnr2 invf1md snr rrate orate mrate;

```

C.8 Matlab program *gen4pam.m*

```

% generate random 4-level number
% [1 2 3 4]
% the output vin is the 4-level PAM with energy=1

rand('uniform');
rand('seed',10000);
lvin=150000;
% Number of vin we want

% For 4-level
f1=sqrt(0.8);
pts=[-3/2*f1:f1:3/2*f1];
% For 4-level

for i=1:lvin,
    vin(i)=pts(round(4*rand+0.5));
end;

save rand4l vin;

```

C.9 Matlab program *bitassign.m*

```

% Compute Bit Assignment after obtaining SNR
% use in conjunction with bitpump.m
% Only Square QAM constellation is used

n=512;
asnrdB=10*log10(snr);

```

```

for i=1:n/2-1,
if asnrdb(i)>=40.1,
jba(i)=5;
elseif asnrdb(i)>=34,
jba(i)=4;
elseif asnrdb(i)>=27.9,
jba(i)=3;
elseif asnrdb(i)>=21.6,
jba(i)=2;
elseif asnrdb(i)>=14.5,
jba(i)=1;
else
jba(i)=0;
end;
end;

save jb jba;

```

C.10 Matlab program *datapump.m*

```

% This is a new file to Simulate Multitone Transceiver
% According to the Bit Assignment given in jb.mat
% Blocklength = 512
% Both DC and Nyquist channels are not used
% Bits are encoded according to the binary order
% Written by Jacky Chow
% March 20, 1991
% with AWGN and NEXT

% sampling rate = 640kHz

load jb;
% jb contains jba (the bit assignments)

load nexts;
% nexts contains xnoise (NEXT noise)

load tapset;

```

```

% tapset contains invfimd (the initial 1-tap settings)

load firtap;
% firtap contains taps (the FIR tap settings)
% and delta (the delay)
% i=0;
% wf=taps(i+1,1:M+1);
wf=taps;
% delay=delta(i+1);
delay=delta;

load pz16r640k1;
% pz16r640k1 contains polyz, polyp
% (the poly, zero model of the channel)

len=7+1; % Number of cyclic prefix + 1

n=512;          % Blocklength
nqam=n/2-1;
% Total number of complex subchannels (w/o D.C & Nyquist)
ff=length(wf); % Number of feedforward tap=M+1

h=filter(polyz, polyp, [1 zeros(1,n-1)]);
% h=channel time response.
% polyz, polyp are the zeros and poles polynomial for the channel

sta1=zeros(1,max(length(polyp),length(polyz))-1);
stb1=zeros(1,length(wf)-1);
% set the number of states for the channel

fh=fft(h);
fw=fft([wf zeros(1,n-ff)]);

ntfcheck=ones(1,n/2-1)./(fh(2:n/2).*fw(2:n/2));
% Initial Taps setting for the 1-tap complex LMS filters

% invfimd(2:n/2)=ntfcheck;
% invfimd is the 1-tap LMS filters
% invfimd is obtained from tapset

```

```

confact=0.01*ones(1,nqam+1);
% Stepsizes for the LMS filters

% the following is for adding delay, Initialization
firout=zeros(1,n+len-1);
vinidft=zeros(1,n);
rcbit=zeros(1,nqam);
icbit=zeros(1,nqam);

rand('normal'); % Normal Distribution
rand('seed',234);

rsnr=zeros(1,nqam);          % this will be the final snr
isnr=zeros(1,nqam);          % this will be the final snr

rberr=zeros(1,nqam); % Initial for computing BER
iberr=zeros(1,nqam);

rerrs=zeros(1,nqam);          % Initialization for computing SNR
ierrs=zeros(1,nqam);          % Initialization for computing SNR
tbit=0; % Total number of bits per block

iter=100000; % Number of block iterations, how many blocks of data

ene=zeros(1,nqam);          % Signal Energy of each QAM channel

for g=1:nqam,
if jba(g)>0,
tbit=tbit+2*jba(g);
seglen=2/(2^jba(g)-1);
for i=1:(2^jba(g))/2,
ene(g)=ene(g)+(1-(i-1)*seglen)^2;
end;
ene(g)=ene(g)/i;
% i=(2^jba(g))/2 at this point
end;
end;
atp=4*sum(ene)/n;
% atp is the average energy
% 2x for real and imag, 2x for the Hermitian counterpart;

```

```

scalesig=sqrt(1e-11*640e3*atp/10);
% AWGN, sprt(Variance)

% tbit is ready at this point
% This is the starting point of the iteration

for j=1:(iter+1),

    vinbuf=vinidft;
    rbit=rcbit;
    ibit=icbit;
% First Pass does not Count

    vin=zeros(1,nqam);
    for g=1:nqam,
        if jba(g)>0,
            temp=rand(1,jba(g));
            segtr=0;
            for i=1:jba(g),
                if temp(i)>0.0,
                    segtr=segtr+2^(i-1);
                end;
            end;
            rcbit(g)=segtr;
            temp=rand(1,jba(g));
            segti=0;
            for i=1:jba(g),
                if temp(i)>0.0,
                    segti=segti+2^(i-1);
                end;
            end;
            icbit(g)=segti;
            seglen=2/(2^jba(g)-1);
            vin(g)=-1+segtr*seglen+sqrt(-1)*(-1+segti*seglen);
        end;
    end;
    vinidft=[0 vin 0 conj(vin(nqam:-1:1))];
% create Hermitian symmetry

```

```

% Here is the addition for delay
    rvinidft=real(vinbuf);
    ivinidft=imag(vinbuf);
% These are used as desired response (Assume we have the actual bit,
% as during training period)

    cinbce=real(ifft(vinidft));
% Perform IFFT

    cin=[cinbce(n-len+2:n) cinbce];
% Here is the addition of cyclic prefix

    sta=sta1;
    stb=stb1;
% update the states

    [cout,sta1]=filter(polyz,polyp,cin,sta);
% channel
% cout = [Yk-n+1 . . . Yk-1 Yk Yk+1 Yk+2],
% where the first len-1 are the cyclic extended output

    gnoise=rand(1,n+len-1)*scalesig/sqrt(n);
% Divided by sqrt(n) since IFFT reduces energy by n

    cout=cout+gnoise+xnoise;
% AWGN and NEXT is added to channel output

    inada=firout;
    [firout,stb1]=filter(wf,1,cout,stb);
% firout is the output of the FIR equalizer at the receiver
% The following is to take into account the effect of delay
    inada=[inada firout];
% inada consists of two blocks, the first one from previous run
% the second from current run
    outada=inada(len+delay:n+len-1+delay);

% Here is the Starting point for finding SNR

    tindec=fft(outada);
% Performs FFT

```

```

        indec=tindec(2:n/2).*invf1md(2:n/2);
% Performs the 1-Tap Filters
% we don't use the D.C. channel

        rindec=real(indec);
        iindec=imag(indec);

% the following is new in order to compute the error vector (AA)

        if j>=2,
for k=1:nqam,
    if jba(k)>0,
        seglen=2/(2^jba(k)-1);
        for gg=1:(2^jba(k)),
            sigcon(gg)=-1+(gg-1)*seglen;
            rdd(gg)=abs(rindec(k)-sigcon(gg));
            idd(gg)=abs(iindec(k)-sigcon(gg));
        end;
        rdix=1;
        for hh=2:(2^jba(k)),
            if (rdd(hh)<rdd(1)),
                rdix=hh;
            end;
            rdd(1)=rdd(hh);
        end;
        routdec(k)=sigcon(rdix);
        idix=1;
        for hh=2:(2^jba(k)),
            if (idd(hh)<idd(1)),
                idix=hh;
            end;
            idd(1)=idd(hh);
        end;
        ioutdec(k)=sigcon(idix);

% rdix,idix = 1,2,..., (2^jba(k))

rbitout(k)=rdix-1;

```

```

ibitout(k)=idix-1;

% since rbit,ibit = 0,1,...,(2^jba(k)-1)

% This is the starting point for computing
% Bit error rate

rtemp=rbit(k);
itemp=ibit(k);
rtempout=rbitout(k);
itempout=ibitout(k);

for gg=jba(k):-1:1,
rbyb=0;
if((rtemp>0)&(rtemp>=(2^(gg-1)))),
rtemp=rtemp-2^(gg-1);
rbyb=1;
end;
rbybout=0;
if((rtempout>0)&(rtempout>=(2^(gg-1)))),
rtempout=rtempout-2^(gg-1);
rbybout=1;
end;
if(rbyb~=rbybout),
rberr(k)=rberr(k)+1;
end;

ibyb=0;
if((itemp>0)&(itemp>=(2^(gg-1)))),
itemp=itemp-2^(gg-1);
ibyb=1;
end;
ibybout=0;
if((itempout>0)&(itempout>=(2^(gg-1)))),
itempout=itempout-2^(gg-1);
ibybout=1;
end;
if(ibyb~=ibybout),
iberr(k)=iberr(k)+1;
end;

```

```

end;

rerrs(k)=(rindec(k)-rvinidft(k+1))^2+rerrs(k);
ierrs(k)=(iindec(k)-ivinidft(k+1))^2+ierrs(k);

end;
end;

ev(2:n/2)=vinbuf(2:n/2)-indec;
invf1md(2:n/2)=invf1md(2:n/2)+ ...
    confact(2:n/2).*(ev(2:n/2).*conj(tindec(2:n/2)));
% Updating the weights

    end;
    % end if j>=startererror

end; % ends j, the outermost loop

% This is the Beginning for finding SNR after the loop

rerrs=rerrs/iter;
ierrs=ierrs/iter;
for k=1:nqam,
if jba(k)>0,
rsnr(k)=ene(k)/rerrs(k);
isnr(k)=ene(k)/ierrs(k);
berg(k)=(rberr(k)+iberr(k))/(2*jba(k)*iter);
end;
end;
pbe=(sum(rberr)+sum(iberr))/(tbit*iter);

% pbe = bit error rate

save bitpump pbe rsnr isnr rberr iberr berg tbit;

```

Bibliography

- [1] S. Kasturia, J. Aslanis, and J.M. Cioffi. Vector Coding for Partial-Response Channels. *IEEE Trans. on Information Theory*, 36(4):741–762, July 1990.
- [2] A. Ruiz, J.M. Cioffi, and S. Kasturia. Discrete Multiple Tone Modulation with Coset Coding for the Spectrally Shaped Channel. To appear in *IEEE Transactions on Communications*.
- [3] G.D. Forney Jr. Maximum likelihood sequence detection in the presence of intersymbol interference. *IEEE transactions on Information Theory*, 18(3):363–378, May 1972.
- [4] J.M. Cioffi. Class notes for EE379, Stanford University. September, 1989.
- [5] J.M. Cioffi. A Multicarrier Primer. private communication, Stanford University, November 1991.
- [6] A.B. Carlson. *Communication Systems*. McGraw-Hill, third edition, 1986.
- [7] C.E. Shannon. A Mathematical Theory of Communications. *Bell Systems Technical Journal*, 27, July 1948.
- [8] M.L. Doelz, E.T. Heald, and D.L. Martin. Binary Data Transmission Techniques for Linear Systems. *Proceedings of the IRE*, 44:656–661, May 1957.
- [9] I. Kalet. The Multitone Channel. *IEEE Trans. on Communications*, 37(2):119–124, February 1989.
- [10] J.A.C. Bingham. Multicarrier Modulation for Data Transmission : An Idea Whose Time Has Come. *IEEE Communications Magazine*, 28(5):5–14, May 1990.
- [11] B. Hirosaki, S. Hasegawa, and A. Sabato. Advanced Groupband Data Modem Using Orthogonally Multiplexed QAM Technique. *IEEE trans on Communications*, 34(6):587–592, June 1986.

- [12] A. Peled and A. Ruiz. Frequency domain data transmission using reduced computational complexity algorithms. In *International Conference on Acoustics, Speech, and Signal Processing*, April 1980. Denver, pp. 964-967.
- [13] G.P. Dudevoir. *Equalization Techniques for High-Rate Digital Transmission on Spectrally Shaped Channels*. PhD thesis, Stanford University, August 1989.
- [14] J. Lechleider. The feasibility of using adaptive transmitters to suppress crosstalk. In *Proceedings ICC'89*, Boston, MA, June 1989. Paper No. 17.6.
- [15] A.V. Oppenheim and R.W. Schaffer. *Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1975.
- [16] J.M. Cioffi, P. Algoet, J.C. Tu, and P.S. Chow. Minimum Mean Squared Vector Coding. In *1990 International Symposium on Information Theory*, San Diego, January 1990.
- [17] P. Algoet and J.M. Cioffi. How to Achieve the Capacity of a Vector Channel with Additive Gaussian Noise. *in preparation*.
- [18] J.C. Tu. *Theory, Design and Application of Multi-Channel Modulation for Digital Communications*. PhD thesis, Stanford University, June 1991.
- [19] R.G. Gallager. *Information Theory and Reliable Communication*. Wiley, New York, NY, 1968.
- [20] J. Salz. A unified theory of data-aided equalization. *Bell Systems Technical Journal*, 60(9):2023-2035, November 1981.
- [21] J.M. Cioffi, G.P. Dudevoir, M.V. Eyuboglu, and G.D. Forney Jr. MMSE Decision-Feedback Equalizers and Coding - Part I: General Results. *IEEE Transactions on Communications*. submitted on 5/90.
- [22] J.S. Chow, J.C. Tu, and J.M. Cioffi. A Discrete Multitone Transceiver System for HDSL Applications. *IEEE Journal on Selected Areas in Communications*, 9(6):895-908, August 1991.
- [23] N. Zervos, S. Pasupathy, and A. Venetsanopoulos. Low Complexity Maximum Likelihood Equalizers. In *International Conference on Communications*, pages 1259-1262, Amsterdam, May 1984.
- [24] N. Zervos, S. Pasupathy, and A. Venetsanopoulos. The Unified Decision Theory of Non-Linear Equalization. In *Globecom '84*, pages 683-687, Atlanta, December 1984.

- [25] J. Wozencraft and I. Jacobs. *Principles of Communication Engineering*. John Wiley and Sons, New York, 1965.
- [26] G. Ungerboeck. Adaptive maximum likelihood receiver for carrier-modulated data-transmission systems. *IEEE trans. on Communications*, 22(5):624–636, May 1974.
- [27] J.M. Cioffi, G.P. Dudevoir, M.V. Eyuboglu, and G.D. Forney Jr. MMSE Decision-Feedback Equalizers and Coding - Part I: General Results. *IEEE Transactions on Communications*. submitted on 5/90.
- [28] G.D. Forney Jr. and M.V. Eyuboglu. Combined Equalization and Coding using Precoding. *IEEE Communications Magazine*, 29(12), December 1991.
- [29] W.U. Lee and F.S. Hill. A Maximum-likelihood Sequence Estimator with Decision-Feedback Equalization. *IEEE Trans. on Communications*, 25:971–979, September 1977.
- [30] L. Ljung and G.T. Soderstrom. *Theory and Practice of Recursive Identification*. MIT Press, Cambridge, MA, 1983.
- [31] C.A. Belfiore and J.H. Park Jr. Decision Feedback Equalization. *Proceedings of the IEEE*, 67(8):1143–1156, August 1979.
- [32] G. Ungerboeck. Fractional tap-spacing equalizer and consequences for clock recovery in data modems. *IEEE trans. on Communications*, 24(8):856–864, August 1976.
- [33] G.D. Forney Jr. Coset Codes I: Introduction and Geometrical Classification. *IEEE Transactions on Information Theory*, 34(5):1123–1151, September 1988.
- [34] G.D. Forney Jr. Coset Codes II: Binary Lattices and Related Codes. *IEEE transactions on Information Theory*, 34(5):1152–1187, September 1988.
- [35] G.A. Clark, S.K. Mitra, and S.R. Parker. Block implementation of adaptive digital filters. *IEEE trans on ASSP*, 29(3):744–752, June 1981.
- [36] J. S. Chow, J. C. Tu, and J.M. Cioffi. A Computationally Efficient Adaptive Transceiver for High-Speed Digital Subscriber Lines. In *1990 International Conference on Communications*, Atlanta, April 1990.
- [37] G.J. Pottie and M.V. Eyuboglu. Combined Coding and Precoding for PAM and QAM HDSL Systems. *IEEE Journal on Selected Areas in Communication*, 9(6):861–870, August 1991.

- [38] A. Aman, R. Cupo, and N. Zervos. Combined Trellis Coding and DFE through Tomlinson Precoding. *IEEE Journal on Selected Areas in Communication*, 9(6):876–884, August 1991.
- [39] J.C. Tu and J.M. Cioffi. A Loading Algorithm for the Concatenation of Coset Codes with Multichannel Modulation Methods. In *1990 Global Communications Conference*, San Diego, December 1990.
- [40] J.M. Cioffi. Least-Squares Storage Channel Identification. *IBM Journal of R&D*, 30(3):310–320, May 1986.
- [41] S.M. Kay. *Modern Spectral Estimation: Theory and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1988.
- [42] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, Englewood Cliffs, NJ, 1987.
- [43] J.A.C. Bingham J.M. Cioffi. A Data-Driven Multitone Echo Canceller. *IEEE Transactions on Communications*. Submitted in May 1991.
- [44] G. Davidson and D. Falconer. Reduced Complexity Echo Cancellation Using Orthonormal Functions. *IEEE Transactions on Circuits and Systems*, 38(1), January 1991.
- [45] M. Ho, J. Tu, and J.M. Cioffi. A Parametric Estimator for Reduced Complexity Echo Cancellation. *1991 IEEE HDSL Workshop*, June 1991. Sunnyvale, CA.
- [46] J.T. Aslanis and J.M. Cioffi. Achievable Information Rates on Digital Subscriber Loops: Limiting Information Rates with Crosstalk Noise. *IEEE Transactions on Communications*, 40(2):361–372, February 1992.
- [47] G.P. Dudevoir, J.S. Chow, S. Kasturia, and J.M. Cioffi. Combined Equalization and Coding for T1 Data Rates on Carrier Serving Area Subscriber Loops. In *1989 International Conference on Communications*, Boston, June 1989.
- [48] J.W. Lechleider. High Bit Rate Digital Subscriber Lines: A Review of HDSL Progress. *IEEE Journal on Selected Areas in Communication*, 9(6):769–784, August 1991.
- [49] Bell Communications Research. High Bit-Rate Digital Subscriber Line (HDSL) Systems that Support DS1 Rate Access in the Copper Loop Plant. *Bellcore Request for Information*, (RFI 90-03), March 1990.
- [50] P.M. Crespo. private communication. Bell Communications Research, August 1988.

- [51] T. Starr and M. Day. private communication. Ameritech, September 1989.
- [52] J.A.C. Bingham. private communication, October 1990.
- [53] D.G. Messerschmitt. Design Issues in the ISDN U-Interface Transceiver. *IEEE Journal on Selected Areas in Communications*, pages 1281–1293, November 1986.
- [54] S. Cox and P. Adams. An Analysis of Digital Transmission Techniques for the Local Network. *British Telecom Technical Journal*, 33, July 1985.
- [55] D.W. Lin. Wide-Band Digital Subscriber Access with Multidimensional Block Modulation and Decision-Feedback Equalization. *IEEE Journal on Selected Areas in Communications*, 7(6):996–1005, August 1989.
- [56] E. Arnon. Impulse Noise Immunity of HDSL Systems. ANSI Contribution number T1E1.4/90-170, September 24, 1990.
- [57] K. Kerpez. Near End Crosstalk is Almost Gaussian. *IEEE Transactions on Communications*. Submitted, February 1991.
- [58] L.F. Wei. Trellis-Coded Modulation with Multidimensional Constellations. *IEEE Trans. on Information Theory*, 33(4):483–501, July 1987.
- [59] G.D. Forney Jr. Trellis Constellations. *IEEE Transactions on Information Theory*. Submitted 1989.
- [60] H. Lou and J.M. Cioffi. A Programmable Parallel Processor Architecture for Viterbi Detection. In *1990 Global Communications Conference*, San Diego, December 1990.
- [61] N. Zervos and I. Kalet. Optimized Decision Feedback Equalization versus Optimized Orthogonal Frequency Division Multiplexing for High-Speed Data Transmission over the Local Cable Network. In *1989 International Conference on Communications*, Boston, June 1989.
- [62] A. Ghanem. private communication. Southwest Bell Corporation, November 1990.
- [63] P.S. Chow, J.C. Tu, and J.M. Cioffi. Performance Evaluation of a Multichannel Transceiver System for ADSL and VHDSL Services. *IEEE Journal on Selected Areas in Communications*, 9(6):909–919, August 1991.
- [64] R.N. Bracewell. *The Fourier Transform and Its Applications*. McGraw-Hill, second edition, 1986.
- [65] H.J. Nussbaumer. *Fast Fourier Transform and Convolution Algorithms*. Springer-Verlag, first edition, 1981.