Copyright
by
Biao Lu
2000

WIRELINE CHANNEL
ESTIMATION AND EQUALIZATION

by

BIAO LU, B.S., M.S.E.E

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2000

WIRELINE CHANNEL ESTIMATION AND EQUALIZATION

APPROVED BY
DISSERTATION COMMITTEE:

Brian L. Evans, supervisor

Alan C. Bovik

Joydeep Ghosh

Risto Miikkulainen

Lloyd D. Clark

To my parents with love

Acknowledgements

I would like to express my deepest appreciation and gratitude to Prof. Brian L. Evans, my dissertation advisor, for his guidance, encouragement, and enthusiasm throughout the course of my graduate research. For it was his judgment and trust that made my admission to Ph.D. program possible and introduced me to such an interesting field. I have been fortunate to be able to benefit from his experiences and his attitude towards work. His desire for perfection has always encouraged me to try to do everything as well as possible. Prof. Evans is the only person so far who thinks that I have leadership abilities. It has been my honor to be the lab manager for Embedded Signal Processing Laboratory (ESPL) for the last two years. It has been indeed great training and experience from which I will benefit throughout my career. It has been my great pleasure and privilege to be his student and work with him.

My sincere appreciation also goes to the other members of the committee — Prof. Alan C. Bovik, Prof. Joydeep Ghosh, Prof. Risto Miikkulainen, Prof. Guanghan Xu, and Dr. Lloyd D. Clark — for their helpful comments and understanding. I took or audited at least one course from each of the professors on my committee. Their serious attitude toward teaching has shown me their rich knowledge in their research fields and their hard work in transferring the knowledge to all of the students. I owe special thanks to Dr. Lloyd D. Clark. Dr. Clark has taught me a lot about time-domain equalizers when I worked with him in the summer of 1999.

I have been really grateful to all of the professors who have taught me different courses from my undergraduate study to graduate school. Among them, I thank sincerely to Prof. Kai Ouyang, my advisor for the senior project from the department of biomedical engineering at the Capital Institute of Medicine, Beijing, China. Prof. Ouyang introduced me to the field of neural networks. I was first admitted to the College of Pharmacy at UT Austin. When I struggled to understand drug design, Prof. Robert S. Pearlman told me that I would not beat out the competition by using my weak points. I appreciate his encouragement to transfer to Department of Electrical and Computer Engineering (ECE). I thank Prof. Gary Wise to recommend me to the ECE department and Prof. J. K. Aggarwal to support me financially for two semesters.

I am thankful for the opportunity to work on a challenging project as a summer intern at the Austin Technology Center at Schlumberger in Austin, Texas. I would like to thank Ms. Suzanne Richardson, Dr. Lloyd Clark, Dr. Terry Mayhugh, Mr. Joe Steiner, Mr. Steve Bissell, and all of the other people who have taught me so many things which I cannot learn from courses in school. What I have learned about designing time-domain equalizers in asymmetric digital subscriber line systems at Schlumberger has become my current research interest and a key part of this dissertation.

I would like to sincerely thank all of the members of ESPL and Laboratory for Image and Video Engineering (LIVE) for the great time I had at UT and great help from them while I was a lab manager of ESPL. They are: Gregory

Allen, Güner Arslan, Serene Banerjee, David Brunke, Young Cho, Niranjan Damera-Venkata, Amey Deosthali, Ming Ding, Srikanth Gummadi, Zhengting He, Tanmoy Mandal, Milos Milosevic, Wade Schwartzkopf, Clint Slatton, and Magesh Valliappan. I have realized from them that the fundamentals of being a leader in a group are to get help from them and also to help them when needed. I own my eternal thanks to Wade Schwartzkopf for his efforts and help to prepare the paperwork for my oral exam, bind my dissertation, etc. ESPL has a collection of excellent graduate students who work on various topics. Among them, I would like to thank Mr. Güner Arslan for being such a good system administrator so that I can use the computer without worry and for the rich discussions with him on research topics. David and Wade occasionally gave me lectures on God so that I could have a better understanding of God. Many thanks go to Serene, Niranjan, and Magesh for their help when I took the real-time DSP lab course. I was so lucky to take the course while there were two past TAs and one current TA in ESPL. Clint deserves special thanks since he made all of the banners for my conference posters. I would also thank the following people for the help I received over the years — Dr. Dong Wei, Dr. Tom Kite, Dr. Marios Pattichis, and Dr. Hung-Ta Pai from LIVE. Their success in finishing their Ph.D. studies has given me great encouragement. I owe special thanks to Dr. Dong Wei for introducing me to matrix pencils which are also a part of this dissertation.

I have my eternal gratitude to Dr. Jack Moncrief, his wife Betty and all of his family members for their love and encouragement. I became one of their family members right after I came to the United States. I met Jack and Betty in Beijing, China, when they attended a conference in 1993. It was they who

encouraged me to come to the United States for a higher degree. Their love and encouragement through these years have been strong support for me. Jack's hard work always encourages me to learn more things no matter how old I am.

I would also like to thank all of my friends for their encouragement and friendship. I owe special thanks to my close friends: Dr. Hongyun Wang, Ms. Haihong Zhuo, Ms. Li Jiang, Dr. Kuiyu Chang and his wife Junyu, and Ms. Xiaowei Wang for their love and help through so many years. Kuiyu and Junyu have been helping me buy and set up computers. They also take care of my pet while I am out of town. They have taught me what true friendship means.

I am especially indebted to my parents for their love, sacrifice, and support. They are my first teachers after I came to this world and have set great examples for me about how to live, study, and work.

Biao Lu

The University of Texas at Austin
December 2000

WIRELINE CHANNEL ESTIMATION AND EQUALIZATION

Publication No. ____________________

Biao Lu, Ph.D
The University of Texas at Austin, 2000

Supervisor: Brian L. Evans

Communication involves the transmission of information from one point to another through a series of processes. The three basic elements in each communication system are the transmitter, channel, and receiver. The transmitter and receiver are separated in space. A channel is the physical medium that connects the transmitter and receiver and distorts the transmitted signals in different ways. Severe distortions occur when data transmits through wireline channels. One way to counteract channel distortion in the transmission band is to employ an equalizer in the receiver.

This dissertation focuses on the design of channel equalizers in wireline communication systems. In particular, I consider equalization with and without channel estimation. When equalization is considered as a classification problem, neural networks can be used as equalizers without estimating the channel impulse response. I design a new neural network equalizer by cascading multilayer perceptron and radial basis function networks. In discrete multitone systems, the channel impulse response needs to be known at the receiver. Channel equalizers, a.k.a. *time-domain equalizers* (TEQs), are used to shorten the effective channel impulse response to a desired length. Channel impulse responses are generally infinite in extent. The long tails of the response are due to the poles of digital subscriber lines. I develop new matrix pencil methods to estimate the pole locations. Then, setting zeros of a TEQ at the locations of estimated poles is one way that I design a TEQ, which is possible with or without the knowledge of input training sequence. I also design divide-and-conquer TEQs which have lower computational cost than the current methods and give comparable performance in terms of shortening signal-to-noise ratio. The divide-and-conquer TEQs can be implemented on fixed-point digital signal processors.

Table of Contents

Acknowledgements **v**

Abstract **ix**

List of Tables **xiv**

List of Figures **xv**

Chapter 1. Introduction **1**

1.1 Wireline communication systems 2
  1.1.1 Voiceband modems 4
  1.1.2 Digital subscriber line modems 4
  1.1.3 Cable modems 6
1.2 Commercial wireline channels 9
1.3 Channel equalization 10
  1.3.1 Telephone lines 11
  1.3.2 Dedicated xDSL lines 14
  1.3.3 Cable lines 16
1.4 Goal and organization of the dissertation 16
1.5 Contributions 19
1.6 Abbreviations 20

Chapter 2. Channel Equalization **22**

2.1 Introduction 23
2.2 Channel estimation 25
  2.2.1 Training sequences 25
  2.2.2 Channel estimation methods 26
  2.2.3 Matrix pencil methods 27
  2.2.4 Hua and Sarkar's matrix pencil method 29

2.2.5 Estimation of the number of poles 31
2.3 Feedforward neural network equalizers 33
2.3.1 Model of a neuron 33
2.3.2 Training of multilayer perceptron (MLP) networks 36
2.3.3 Training of radial basis function (RBF) networks 38
2.3.4 Neural networks as channel equalizers 39
2.4 Discrete multitone (DMT) modulation 43
2.5 Time-domain equalization (TEQ) for DMT channels 44
2.5.1 Minimizing mean squared error (MSE) design 47
2.5.2 Maximizing shortening signal-to-noise ratio (SSNR) design . . . 50
2.5.3 Maximizing channel capacity 55
2.6 Frequency-domain equalizer for DMT channels 56
2.7 Conclusion 57

Chapter 3. Neural Network Equalizers 59
3.1 Introduction 59
3.2 Hybrid MLP-RBF equalizer 61
3.3 Simulation results for 2-PAM 62
3.4 Simulation results for 16-QAM 66
3.5 Conclusion 68

Chapter 4. A New Matrix Pencil Method for Channel Estimation 70
4.1 Introduction 70
4.2 Reduced-rank Hankel approximation 72
4.3 Modified matrix pencil methods 73
4.3.1 Method #1 74
4.3.2 Method #2 75
4.3.3 Method #3 77
4.3.4 Comparison 78
4.3.5 Matrix pencil for channel estimation 79
4.4 Simulation results 80
4.5 Conclusion 85

Chapter 5. Time-Domain Equalization for Discrete Multitone Modulation 87

5.1 Introduction 88

5.2 Divide-and-Conquer Method 90

  5.2.1 Finding the delay Δ 91

  5.2.2 Divide-and-Conquer TEQ by minimization 92

  5.2.3 Divide-and-Conquer TEQ to cancel the energy in $\mathbf{h}_{\mathrm{wall}}$ 95

5.3 Blind Channel Shortening 98

  5.3.1 Models of discrete multitone wireline channels 100

  5.3.2 Matrix pencil TEQ 101

5.4 Simulation Results 103

  5.4.1 Comparison of SSNR for known channels 104

  5.4.2 Comparison of SSNR for unknown channels 108

5.5 Conclusion 110

Chapter 6. Conclusion 113

6.1 Equalization without estimating the channel 113

6.2 Equalization based on channel estimation 115

6.3 Future research 117

Appendix 120

Appendix A. Proof that $J_{2,i}$ in (5.6) is not ill-conditioned 121

Appendix B. Proof of unique solutions to (5.8) 126

Bibliography 128

Vita 141

List of Tables

1.1 ITU-T V series transmission standards. 5
1.2 Transmission standards for xDSL family. 6
1.3 Comparison of speed between cable modem and other technologies. 8
1.4 Transmission standards for cable modems. 9

2.1 Implementation and computational cost of the maximum shortening signal-to-noise ratio (SSNR) method 54

3.1 Training time and symbol error rate 64

4.1 Comparison among Matrix Pencil method and Modified Matrix Pencil methods 79

5.1 Implementation and computational cost of Divide-and-Conquer-TEQ-minimization 94
5.2 Implementation and computational cost of Divide-and-Conquer-TEQ-cancellation with unit tap constraint 96
5.3 Implementation and computational cost of Divide-and-Conquer-TEQ-cancellation with unit norm constraint 99
5.4 Computational cost for maximum SSNR method and proposed methods: Divide-and-Conquer-TEQ-minimization and Divide-and-Conquer-TEQ-cancellation 99
5.5 The Matrix Pencil TEQ design method 102
5.6 Parameters for the nine digital subscriber line channels used in simulation. 111
5.7 Locations of poles in eight carrier-serving-area channels 112

List of Figures

1.1 Block diagram of a dial-up connection from a PC to Web server, where PSTN is the public switched telephone network and RAS is the remote access service. 3

1.2 System model of a modem. 3

1.3 Cable setup. 8

1.4 Intersymbol interference 11

1.5 A simple communication system 12

1.6 A feedforward equalizer 13

1.7 Impulse and frequency response of carrier-serving-area loop 1. . . 15

1.8 Cyclic prefix 15

2.1 Implementation of original matrix pencil method. 31

2.2 Models of a biological neuron and artificial neurons. 35

2.3 A feedforward neural network 37

2.4 Pulse amplitude modulation signals 40

2.5 Multicarrier modulation transmitter. 44

2.6 Multicarrier modulation receiver. 44

2.7 Discrete multitone modulation subchannels. 45

2.8 Time-domain equalizer 48

2.9 Implementation of the minimum mean squared error method. . . 49

2.10 Comparison of actual effective channel and ideal effective channel 51

3.1 Performance analysis of four equalizers. 65

3.2 Comparison of multilayer perceptron (MLP), radial basis function (RBF), and MLP-RBF equalizers 66

3.3 Comparison of multilayer perceptron (MLP), radial basis function (RBF), and MLP-RBF equalizers 68

4.1 Modified Matrix Pencil Method 1 74

4.2 Modified Matrix Pencil Method 2 76

4.3 Modified Matrix Pencil Method 3 78
4.4 Performance comparison of modified matrix pencil methods in estimating damping factors 82
4.5 Performance comparison of modified matrix pencil methods in estimating frequencies 83
4.6 Performance comparison of pole 1 at 0.8464 by five methods . . . 84
4.7 Performance comparison of pole 2 at 0.7146 by five methods . . . 84
4.8 Performance comparison of pole 3 at 0.2108 by five methods . . . 85

5.1 Impulse response of two wireline channels 105
5.2 Determination of Δ for carrier-serving-area digital subscriber loop 1 channel 106
5.3 Comparison of four different TEQ design methods in terms of SSNR for two known channels with 2–21 TEQ taps 107
5.4 Comparison among matrix pencil methods and MMSE method in terms of SSNR for unknown channels for 2–30 TEQ taps 109

Chapter 1

Introduction

In the 1970s, telecommunications was virtually synonymous with plain old telephone service. Technology primarily consisted of copper wires and electromechanical switches. In the 1980s, telecommunications services expanded to include voiceband data modems, facsimile machines, and analog cell phones. Now, through digitization and technological convergence, telecommunications involves the transfer of a wide variety of information — data, speech, audio, image, video, and graphics — over wireless and wireline channels.

Communication is the transmission of information from one point to another through a series of processes. The three basic elements in a communication system are a transmitter, channel, and receiver. The transmitter and receiver are separated in space. A channel is the physical medium that connects the transmitter to the receiver, and it distorts the transmitted signals in various ways. Data transmission through wireline channels with severe distortion can be made more reliable by using the following techniques:

1. Transmit and receive filters to reject distortions that fall outside the band of transmitted frequencies.

2. Equalizers in the receiver to counteract channel distortion in the transmis-

sion band.

3. Detection techniques to recover transmitted data from noisy data.

Channel equalizer design is the theme of the dissertation. In this chapter, Section 1.1 describes a general communication system and summarizes standards for modems. Section 1.2 discusses properties of commercial wireline channels. Section 1.3 introduces equalization of commercial wireline channels. Section 1.4 presents the goal and organization of this dissertation. Section 1.5 lists the publications related to this dissertation.

1.1 Wireline communication systems

Figure 1.1 shows a block diagram for connecting to the Internet through a dial-up connection. In a dial-up connection, the user's personal computer (PC) connects to a modem that dials up across the public switched telephone network (PSTN) to a remote access service (RAS) concentrator. The word "modem" is a concatenation of **mod**ulator and **dem**odulator, but there is a wide range of opinion as to what constitutes modulation and demodulation, and whether a modem comprises more than a *mod* and a *demod* [1]. Figure 1.2 shows a digital communication system model for modulation and demodulation [2]. A modem is the combination of a transmitter and receiver that is used to convey information in the form of digital signal, from one location to another over the appropriate channel. There are three commonly-used modems to make the connection from a PC to the Internet. They are voiceband modems, digital subscriber line (DSL) modems, and cable modems, which are discussed in Sections 1.1.1, 1.1.2, and 1.1.3, respectively.

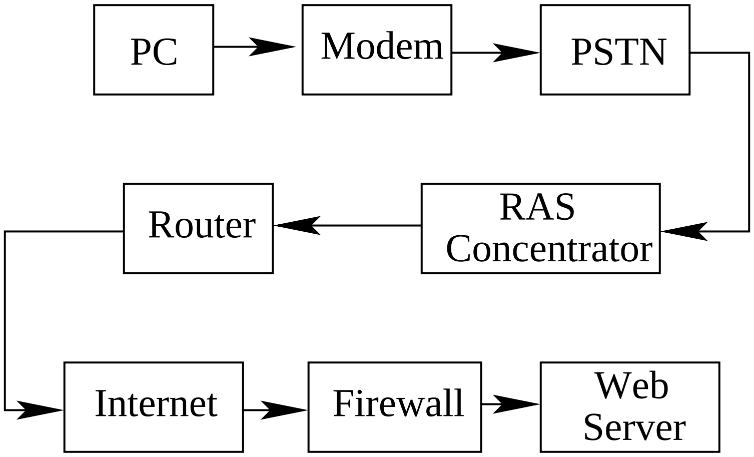


Figure 1.1: Block diagram of a dial-up connection from a PC to Web server, where PSTN is the public switched telephone network and RAS is the remote access service.

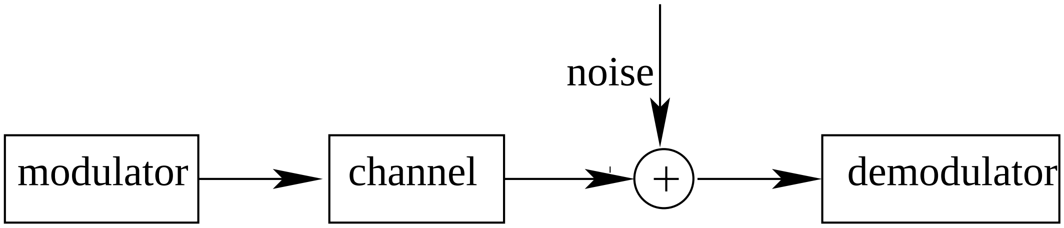


Figure 1.2: System model of a modem.

1.1.1 Voiceband modems

One device that makes the connection in Figure 1.1 possible across a wide geographical area is the voiceband modem. The modem at either end of the PSTN allows the PC and Web server to communicate with each other. Voiceband modems are used to carry digital data from a PC (or Web server) through the available infrastructure of a telephone network. However, the telephone network was initially designed to carry voice signals in analog form. Later, control offices in telephone companies were redesigned to digitize and transport speech sampled at 8 kHz with 8 bits/sample (i.e., 64 kbits/s). Therefore, voiceband connections have to go through multiple analog-to-digital and digital-to-analog converters, which will delay transmission. This delay is part of the propagation delay time that signals take to pass through the voice channel of the transmission link. Table 1.1 lists various standards for voiceband modems [3].

The voiceband is typically considered as the analog frequency band from 0 to 4 kHz [4]. The passband of a voiceband telephone channel is roughly 300 Hz to 3300 Hz. Modem technology is subject to these bandwidth limitations imposed by the analog voice network between the subscriber and the central office. Thus, dial-up access through the telephone network is slow and ill-suited to the Megabit bandwidth requirements of rich and dynamic multimedia content.

1.1.2 Digital subscriber line modems

Many possible solutions for the bandwidth limitation of the PSTN network exist. In fact, the copper loop used to carry voice traffic between a subscriber and the central office at a telephone company (Telco) is inherently capable of sufficient

Standard	*Transmission Rate*	*Duplex Capability*	*Mode*	*Media*
V.21	300 bit/s	Full	Asynchronous	2 wire PSTN
V.22	1200 bit/s	Full	Asynchronous	2 wire PSTN
V.22 bis	2400 bit/s	Full	Asynchronous	2 wire PSTN
V.23	1200 bit/s	Full	Asynchronous	4 wire leased
V.26 bis	2400 bit/s	Full Half	Synchronous Synchronous	2 wire leased 2 wire PSTN
V.27	4800 bit/s	Half	Synchronous	2 wire PSTN
V.29	9600 bit/s	Full	Synchronous	4 wire leased
V.32	9600 bit/s	Full	Asynchronous/ Synchronous	2 wire PSTN
V.32 bis	14400 bit/s	Full	Asynchronous/ Synchronous	2 wire PSTN
V.33	14400 bit/s	Full	Synchronous	4 wire leased
V.34	28800 bit/s	Full	Asynchronous/	2 wire PSTN

Table 1.1: ITU-T V series transmission standards.

bandwidth to carry Megabits of data. Since the voiceband utilizes only 4 kHz of the bandwidth, the copper loop has unused bandwidth that could be used to support high data rates depending on the loop length. This fact motivates *digital subscriber line (DSL)* technology. The technology forms a family called x-type digital subscriber line (xDSL), where “x” stands for one of many types of DSL technology. The following advantages make DSL technology an attractive choice for high-speed Internet access:

- DSL technology utilizes the existing infrastructure in the PSTN;
- DSL technology does not require replacement of network equipment;
- DSL technology builds upon the techniques developed for modem technology for modulation, error correction, and error detection.

Standard	*Meaning*	*Transmission Rate (Mbps)*	*Mode*	*Media*
ISDL	ISDN DSL	0.144	Symmetric	1 wire pair
SDSL	Single Line DSL	0.768	Symmetric	1 wire pair
HDSL	High data rate DSL	1.544	Symmetric	2 wire pairs
HDSL2		2.048	Symmetric	1 wire pair
CDSL	Consumer DSL	up to 1 0.016 to 0.128	Downstream Upstream	1 wire pair
ADSL	Asymmetric DSL	1.5 to 8 0.016 to 0.640	Downstream Upstream	1 wire pair
RADSL	Rate Adaptive DSL	1.5 to 8 Mbps 16 to 640 Kbps	Downstream Upstream	1 wire pair adapt data rate to line conditions
VDSL (proposed)	very high data rate DSL	13 to 52 Mbps 1.5 to 6.0 Mbps	Downstream Upstream	fiber feeder and ATM

Table 1.2: Transmission standards for xDSL family.

Table 1.2 lists the standards of current xDSLs [5]. In spite of possible confusion over the relationships among xDSLs, ADSL is the most standardized in terms of available documentation, service trials, and open specifications [6]. In this dissertation, I focus on ADSL which uses bandwidth from 25 kHz to 1.1 MHz for data transmission and 0 to 4 kHz for voice transmission.

1.1.3 Cable modems

Many providers of cable television services offer *cable modems* as their own solution for high-speed data communication to the home. In fact, the term “cable modem” is a bit misleading because it works more like a local area network interface than a modem. In some cable TV networks, cable modems allow computers to be connected to the same cable system that feeds the television set. Using a signal splitter, the coaxial cable hosts the modem on the PC side of the connection. The cable modem is connected to an Ethernet card that resides on the user's PC. The software installed along with the Ethernet card allows access to the Internet.

Figure 1.3 shows how the cable modem fits into the home cable setup (see www.cablemodems.com). The cable modem is located between the radio frequency (RF) module and the Ethernet card so that Internet and other data communications traffic can be managed separately from the video signals. Cable modems use the 5 to 50 MHz frequency band for upstream channels and the 50 to 550 MHz band for downstream channels. Cable modems take advantage of existing cable TV networks and are capable of operating at higher speeds than ADSL. Table 1.3 compares the transmission speed of different technologies. Although cable modems achieve high speed, they have several disadvantages [7]. The primary disadvantage is the cable link to a residence is shared among many users. Therefore, if some of the users log onto the Internet at the same time, then the achievable communication speeds decrease. Because the line is shared, security can also be a serious problem for some users. Second, the upstream bandwidth picks up signals and noise from other home services [5]. Third, cable modems may not be available in the vast majority of commercial districts since cable has been deployed primarily for residential use.

Table 1.4 lists the current standards for cable modems [8]. Multimedia Cable Network System Partners Ltd. (MCNS) was formed in 1996 and released its draft standard called the Data Over Cable Service Interface Specification (DOCSIS) in March 1997. In 1998, the International Telecommunications Union

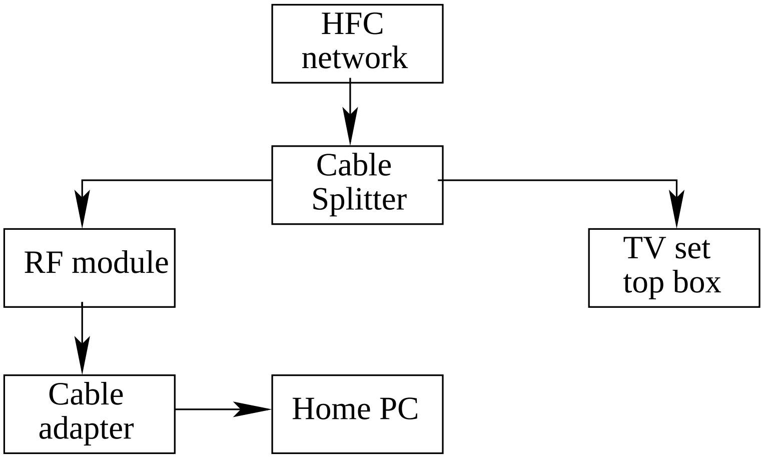


Figure 1.3: Cable setup.

Technology	*Downstream*	*upstream*
POTS	56.6 kbps	56.6kbps
ISDN	230 kbps	230 kbps
SDSL	384 kbps	384 kbps
HDSL	768 kbps	768 kbps
ADSL	8 Mbps	500 kbps
cable modems	**27 Mbps**	**10 Mbps**
Wireless (900 MHz)	28.8 kbps	28.8 kbps
Satellite	400 kbps	POTS line used

Table 1.3: Comparison of speed between cable modem and other technologies.

(ITU) accepted DOCSIS as a cable modem standard, called ITU J.112. The Institute of Electronic and Electrical Engineers (IEEE) 802.14 cable TV media access control and physical protocol working group was formed in 1994 but did not develop an international cable modem standard until 1998. Digital Video Broadcast (DVB)/Digital Audio Council (DAVIC) technology is the incumbent European standard for digital set-top boxes and is now starting to be employed for cable modems. As the name suggests, DVB focuses on digitized video delivery

Standard	*Transmission Rate*	*Mode*
MCNS/DOCSIS	up to 38 Mbps	Downstream
	up to 10 Mbps	Upstream
IEEE802.14	20.5632 Mbps, 30.8448 Mbps, or 41.1264 Mbps	Downstream
	132 kbps, 536 kbps, or 2.144 Mbps	Upstream
DVB/DAVIC	up to 51 Mbps	Downstream
	up to 6.178 Mbps	Upstream
ITU-T J.112	the same as MCNS/DOCSIS	

Table 1.4: Transmission standards for cable modems.

and requires MPEG II framing.

1.2 Commercial wireline channels

A channel is a transmission path from the transmitter to receiver. The analog channels corresponding to the three types of modems in Section 1.1 follow:

- **Telephone lines**

  The Plain Old Telephone Service (POTS) line is needed for remote broadcast consoles, telephone hybrids, analog telephones, cordless telephones, fax machines, and modems. The POTS line consists of two wires called tip and ring. The bandwidth of a POTS line is approximately from 300 Hz to 3300 Hz. The signal-to-noise ratio over the passband is approximately 45 dB.

- **Dedicated ADSL lines**

  ADSL service provides high-speed transmission over twisted-pair telephone lines. Copper twisted-pair is capable of carrying higher frequency signals up to approximately 2 MHz for distance up to 10 kft. However, high-frequency signals experience more attenuation with distance than do signals at voiceband frequencies. ADSL uses a guard frequency band to separate the voiceband POTS from ADSL frequencies. A set of eight test ADSL lines is discussed in detail in [6].

- **Cable lines**

  Cable TV networks operate via coaxial cable. Some cable TV systems are hybrid fiber/coax (HFC) systems. Both coaxial cable and HFC system are analog. Coaxial cables are RF transmission lines. The coaxial cables in HFC systems can carry an extensive bandwidth up to 1 GHz. The characteristic impedance of coaxial cables in HFC systems is 75 Ω. According to the IEEE 802.14 standard, the maximum distance between the farthest end-user and the fiber hub is 50 miles [8].

1.3 Channel equalization

Telephone lines, ADSL lines, and cable lines distort transmitted signals. The transmitted signals are bandpass. The channel's frequency response $C(f)$ can be expressed as [9]

$$C(f) = A(f)\ e^{j\theta(f)} \tag{1.1}$$

where $A(f)$ is the amplitude response and $\theta(f)$ is the phase response. An ideal channel that does not distort the transmitted signal is obtained when $A(f)$ is a constant and $\theta(f)$ is a linear function of frequency, f.

When signals are transmitted through a non-ideal channel, the channel may disperse the signal in such a way that a pulse interferes with adjacent

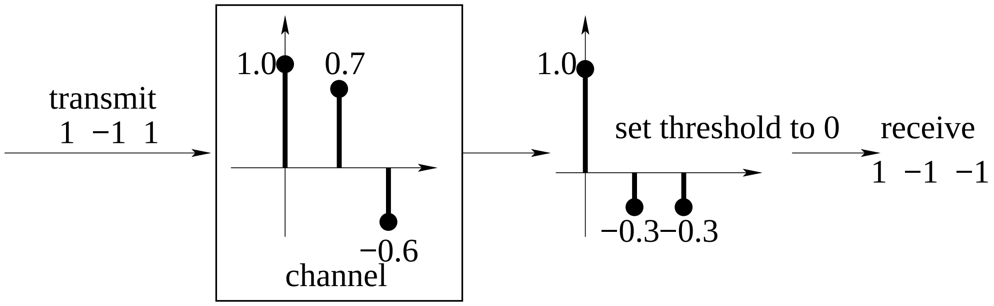


Figure 1.4: The communication channel distorts the transmitted signal by attenuating, delaying, and dispersing each pulse that represents a symbol. The dispersion causes intersymbol interference in the received signal.

pulses at the sample instant, causing *intersymbol interference* (ISI), as shown in Figure 1.4. In Figure 1.4, the transmitted signal is the binary signal $1, -1, 1$, and each pulse is distorted by a channel with impulse response $\mathbf{h} = [1.0, 0.7, -0.6]$. The channel contains no noise. The receiver recovers a sequence of $1, -1, -1$ which is different from the transmitted signal. At the receiver, we may employ a linear filter with adjustable coefficients to compensate for the channel distortion. The filter coefficients are adjusted on the basis of measurements of the channel characteristics. These adjustable filters are called *channel equalizers*. Figure 1.5 shows a basic model of a communication system that contains an equalizer to compensate for ISI. Section 1.3.1 discusses equalizers for telephone lines that are obtained by inverting the channel impulse response. Section 1.3.2 describes how to flatten the frequency response of a channel in DSL lines. Section 1.3.3 mentions equalizers for cable lines.

1.3.1 Telephone lines

The structure of a feedforward equalizer used in a telephone line transmission system is shown in Figure 1.6. If the frequency response of an equalizer is $E(f)$, then $\mid E(f) \mid$ must compensate for the channel distortion. A first solution would be to make the equalizer frequency response equal the inverse of the channel impulse response, i.e.,

$$E(f)C(f) = 1, \text{ or } E(f) = \frac{1}{C(f)} \tag{1.2}$$

Equation (1.2) implies that the equalizer is the *inverse channel filter* to the channel response so that the linear channel distortion can be fully compensated. This inverse channel filter can completely eliminate ISI caused by the channel. Since it forces the ISI to be zero at the sampling times, the equalizer based on (1.2) is called the *zero-forcing equalizer.*

One disadvantage of the zero-forcing equalizer is that it neglects the presence of additive noise. As a result, the use of a zero-forcing equalizer may amplify the noise significantly. If $C(f)$ in (1.2) is small in a frequency range, then the channel equalizer, $E(f) = 1/C(f)$, gives a large gain in the same frequency range. Therefore, the noise in this frequency range is amplified.

A second solution to compensate channel distortion is to use a minimum mean square error (MMSE) equalizer. If the output from the equalizer is $f(\mathbf{y}(k))$ as shown in Figure 1.6 and the desired response at the output of an equalizer

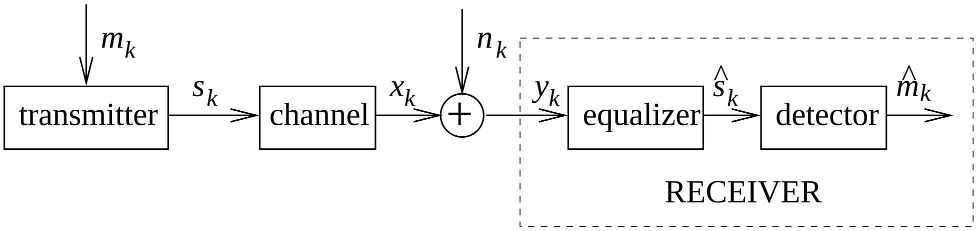


Figure 1.5: A simple communication system. A digital message m_k is transmitted through an analog channel. The received signal y_k is corrupted by additive noise represented by n_k. The receiver equalizes the distortion in the channel and then detects the symbols (sequence of bits) that were transmitted.

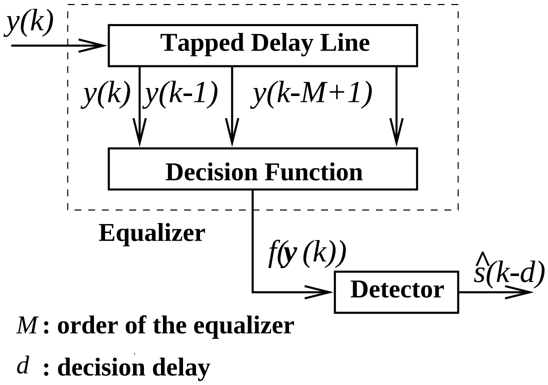


Figure 1.6: A feedforward equalizer. The detector is nonlinear.

is the transmitted symbol $\mathbf{s}(k)$ in Figure 1.5, then the kth sample of the error signal between $f(\mathbf{y}(k))$ and $\mathbf{s}(k)$ is

$$e(k) = f(\mathbf{y}(k)) - \mathbf{s}(k) \tag{1.3}$$

Then, the mean squared error (MSE) is defined as

$$\text{MSE} = E\left\{e^2(k)\right\} \tag{1.4}$$

where $E\{\cdot\}$ is the expectation operator. We can obtain the minimum MSE solution by differentiating the MSE in (1.4) with respect to the equalizer coefficients, setting the derivative to zero, and solving for the equalizer coefficients.

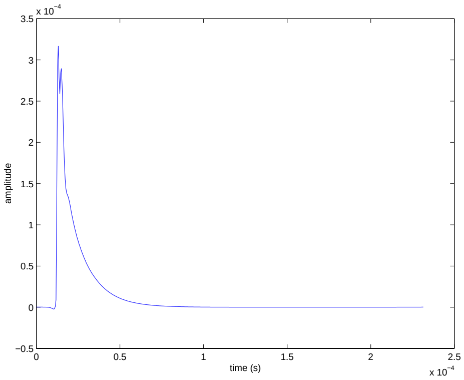
Both zero-forcing and MMSE equalizers are linear equalizers to combat ISI. When a channel has a spectral null (i.e., $C(f) = 0$), the linear equalizers may not compensate ISI sufficiently. An alternative is to use a nonlinear equalizer such as a decision-feedback equalizer. Neural networks may also be used as the decision function in Figure 1.6, as discussed in Section 2.3.

Both linear equalizers and neural network equalizers share one drawback. With respect to the channel length, the computation time to adapt the equalizer coefficients increases dramatically as the length of a channel increases. Therefore, shorter channels are desired. Channel shortening is discussed next.

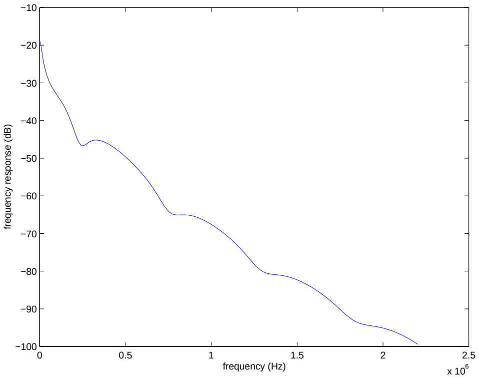
1.3.2 Dedicated xDSL lines

In an ideal channel, the frequency response $A(f)$ is constant. In general, the frequency response of a channel over the entire bandwidth range is not constant. Figure 1.7 shows the impulse response and frequency response of the carrier-serving-area (CSA) digital subscriber loop 1 [6]. Its frequency response, as shown in Figure 1.7(b), is not flat. The CSA loop 1 impulse response contains 512 taps. To invert this 512-tap channel, the channel equalizer mentioned in Section 1.3.1 is an all-pole IIR filter obtained from (1.2) with 511 poles which may not be stable (the poles would be outside the unit circle) and might enhance the channel noise. In general, an FIR filter is used to approximate the IIR filter. The FIR filter will have to be long enough, and small changes in IIR coefficients may require large changes in FIR coefficients. To make the equalizer problem tractable for a broadband channel, discrete multitone (DMT) modulation is proposed. DMT modulation is the standard for data transmission in ADSL [6]. Using DMT modulation, the channel is divided into a large number of parallel, independent, and approximately flat subchannels. The non-ideal channel causes ISI between two adjacent subchannels.

One popular method for combating ISI in DMT modulation is to use a guard sequence, called the *cyclic prefix* (CP). The CP is prepended to each symbol. In the ADSL standard [6], the CP is a copy of the last 32 samples of a symbol (see Figure 1.8). The symbol length is 512 samples.



(a) Impulse response of CSA loop 1



(b) Frequency response of CSA loop 1

Figure 1.7: Impulse and frequency response of carrier-serving-area loop 1.

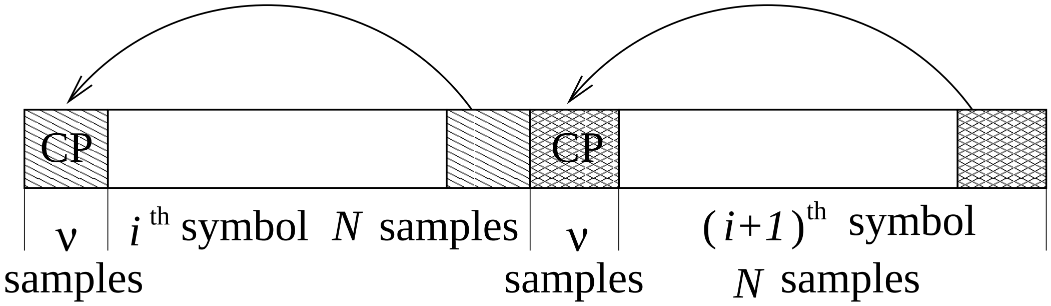


Figure 1.8: Cyclic prefix

1.3.3 Cable lines

High-speed data transmission through coaxial cable also suffers from ISI introduced by the cable. In the IEEE 802.14 standard for cable modems, quadrature amplitude modulation (QAM) is used. An adaptive equalizer in a cable modem for combating ISI consists of a feedforward equalizer combined with a decision feedback equalizer. There are 16 complex taps — the feedforward equalizer and decision feedback equalizer each have eight taps. There are no training sequences required so that equalization for cable modems is blind [8],[10].

The IEEE 802.14 standard suggests single-carrier modulation since it is market-ready and the technology is well understood and developed. Theoretically, multicarrier modulation such as DMT has advantages in performance. IEEE 802.14 officially formed a research group to investigate multicarrier modulation for future application for cable modems. Therefore, DMT modulation and equalization methods may be used in future cable modems [8].

1.4 Goal and organization of the dissertation

This dissertation focuses on new channel estimation and equalizer design methods. The goal of this dissertation is to investigate the applications of filters and parameter estimation in channel equalization in a digital communication system. The contributions of the dissertation are the following:

- Wireline channel equalization can be considered to be a classification problem. Previous uses of neural network classifiers as equalizers are described. I develop a new equalizer by cascading two neural networks in order to decrease the computational cost and reduce symbol error rate vs. SNR.

Since the number of symbols used to train neural networks is related to the length of the channel impulse response and the number of neurons in the input layer of the network, I develop methods to estimate channel impulse response and shorten the channel to a desired length.

- When a channel is not known to the receiver, I develop a matrix pencil method to estimate the channel by locating the poles of an IIR filter model of the channel.
- A channel shortening method is used to compensate for the intersymbol interference incurred during high-speed data transmission by discrete multitone modulation, e.g. in asymmetric digital subscriber lines. I propose two new methods for channel shortening. Both can be implemented in real-time software using fixed-point arithmetic and give comparable shortening signal-to-noise ratio to the optimum method in [11].

The three contributions of the dissertation are discussed in the following chapters:

1. Chapter 3 proposes a new neural network for channel equalization,
2. Chapter 4 modifies matrix pencil methods to improve channel estimation, and
3. Chapter 5 develops new time-domain equalizers for DMT systems based on the modified matrix pencil method and a divide-and-conquer algorithm.

Chapter 2 introduces channel estimation and equalization. Chapter 2 also explores applications of neural networks and matrix pencil methods to channel equalization in wireline channels. Finally, Chapter 2 describes time-domain equalizers for ADSL channels.

Channel equalization can be considered as a classification problem. Chapter 3 analyzes the use of neural networks as channel equalizers. I propose to cascade multilayer perceptron and radial basis function networks [12] to reduce noise and classify transmitted data.

Chapter 4 improves the matrix pencil method for channel estimation [13]. The data matrix from the received signal has *Hankel structure* and *rank deficiency* when noise is not present. When noise is present, the data matrix has Hankel structure but loses rank deficiency. Singular value decomposition (SVD) is a tool to reduce rank but it destroys Hankel structure. I propose three novel methods by applying a reduced rank Hankel approximation [14, 15].

Chapter 5 presents the applications of matrix pencil methods to design channel equalizers. When the channel is long, the linear filter needs more taps to equalize the distortion whereas neural networks require more data to train the weights. Linear filter methods have difficulty with more memory requirements to save the taps, where training a neural network may not be realizable in a real-time implementation. One practical approach is to shorten the extent of the channel impulse response to a desired window length [16], which is widely used in discrete multitone systems [17]. The shortening is effected by an FIR filter called a *time-domain equalizer* or *channel shortening equalizer*. I develop closed-form solutions for sub-optimal TEQ design using a divide-and-conquer method. The solutions have lower computational cost than previously proposed methods with comparable performance. I model a channel as an IIR filter and use matrix pencil methods to find the poles so that the channel impulse response can be shortened.

Chapter 6 concludes this dissertation, highlights the contributions, and points out future research directions.

1.5 Contributions

The material presented in this dissertation is discussed in the papers given below.

- The neural network equalizer presented in Chapter 3 is based on:
  B. Lu and B. L. Evans, "Channel Equalization by Feedforward Neural Networks", *Proc. IEEE Int. Sym. on Circuits and Systems*, May 31-Jun. 2, 1999, Orlando, FL, vol. 5, pp. 587-590.

- The modified matrix pencil methods presented in Chapter 4 are from:
  B. Lu, D. Wei, B. L. Evans, and A. C. Bovik, "Improved Matrix Pencil Methods", *Proc. IEEE Asilomar Conf. on Signals, Systems, and Computers*, Nov. 1-4, 1998, Pacific Grove, CA, vol. 2, pp. 1433-1437.
  The idea of modified matrix pencil method #2 comes from Prof. Dong Wei [18].

- The divide-and-conquer time-domain equalizer presented in Chapter 5 appears as a conference paper and has also been submitted as a journal paper:
  - B. Lu, L. D. Clark, G. Arslan, and B. L. Evans, "Divide-and-Conquer and Matrix Pencil Methods for Discrete Multitone Equalization," *IEEE Transactions on Signal Processing*, submitted.
  - B. Lu, L. Clark, G. Arslan, and B. L. Evans, "Fast Time-Domain

Equalization for Discrete Multitone Modulation Systems", *IEEE Digital Signal Processing Workshop*, Oct. 15-18, 2000, Hunt, Texas.

The idea for the Divide-and-Conquer method comes from Dr. Lloyd Clark [19]. The closed-form solution for the Divide-and-Conquer cancellation with unit tap constraint method is due to Mr. Güner Arslan [20].

1.6 Abbreviations

This section lists the acronyms which appear in this dissertation.

ADSL : Asymmetric Digital Subscriber Line
ANN : Artificial Neural Network
ANSI : American National Standards Institute
ATM : Asynchronous Transfer Mode
CDSL : Consumer Digital Subscriber Loop
CRB : Cramer-Rao Bound
CSA : Carrier-Serving-Area
DAVIC : Digital Audio Council
DC : Divide-and-Conquer
DFT : Discrete Fourier Transform
DMT : Discrete Multitone
DOCSIS : Data Over Cable Service Interface Specification
DSL : Digital Subscriber Loop
DVB : Digital Video Broadcast
EM : Expectation Maximization
FEQ : Frequency-domain Equalizer
FIR : Finite Impulse Response
HDSL : High-speed Digital Subscriber Loop
HFC : Hybrid Fiber/Coax
IEEE : Institute of Electrical and Electronic Engineers
IIR : Infinite Impulse Response
ISDL : ISDN Digital Subscriber Loop
ISDN : Integrated Services Digital Network
ISI : Intersymbol Interference
ITU : International Telecommunication Union

LM : Levenberg-Marquardt
LMS : Least Mean Squares
LRHA : Low Rank Hankel Approximation
MCNS : Multimedia Cable Network System
MKT : Modified Kumaresan-Tufts
MLP : Multilayer Perceptron
MMP : Modified Matrix Pencil
MMSE : Minimum Mean Squared Error
MP : Matrix Pencil
MPEG : Moving Picture Experts Group
MSE : Mean Squared Error
PAM : Pulse Amplitude Modulation
PC : Personal Computer
POTS : Plain Old Telephone Service
PSTN : Public Switched Telephone Network
QAM : Quadrature Amplitude Modulation
RADSL : Rate Adaptive Digital Subscriber Loop
RAS : Remote Access Service
RBF : Radial Basis Function
RRHA : Reduced Rank Hankel Approximation
SDSL : Single line Digital Subscriber Loop
SER : Symbol Error Rate
SNR : Signal-to-Noise Ratio
SSNR : Shortening Signal-to-Noise Ratio
SVD : Singular Value Decomposition
Telco : Telephone company
TEQ : Time-domain Equalizer
TIR : Target Impulse Response
UNC : Unit Norm Constraint
UTC : Unit Tap Constraint
VDSL : Very high data rate Digital Subscriber Loop

Chapter 2

Channel Equalization

Equalization is the process of applying a filter to a signal in order to remove or compensate for the effects of linear distortion. This filter can be defined in the frequency domain by frequency response parameters. It can also be defined in the time domain by its impulse response. In this chapter, I describe two types of channel equalizers:

1. **Equalizers that do not require channel estimation**

   A neural network can be employed as a feedforward equalizer. It has an advantage over linear feedforward equalizers since the neural network is nonlinear and has a generalization ability. It can learn the properties of an equalizer from the received signal and training sequence without estimating the channel impulse response.

2. **Equalizers that require channel estimation**

   DMT modulation has been approved as the standard modulation method for ADSL. At the receiver end in a DMT system, current technologies require estimation of the channel impulse response. The design of a time-domain equalizer can fully depend on the knowledge of the channel impulse response.

2.1 Introduction

Equalization is effective for compensating types of linear distortion including

- frequency distortions such as ripple, and
- nonlinear phase.

However, equalization is not effective on the following types of nonlinear distortion:

- **Noise**: The two primary sources of noise are electromagnetic interference and ambient noise. Electromagnetic interference is caused by a radio signal or other magnetic field including itself onto a medium (twisted/nontwisted-pair wire) or device (telephone or other electronics).
- **Adjacent channel interference**: Signals are assigned to different frequency bands. The adjacent channel interference is generated by transmitters assigned to adjacent frequency bands.
- **Spurious distortion**: The following changes in a signal involve the addition of spurious tones at frequencies not present in the original signal:
  - **Intermodulation**: In “intermodulation” distortion, discordant tones appear at the sums and differences of two original frequencies.
  - **Harmonic distortion**: In “harmonic” distortion, the spurious tones are at integral multiples of the original frequency.

Equalizers can be divided mainly into two classes: feedforward equalizers (linear filter with adjustable parameters compensate for the channel distortion) and decision feedback equalizers. In the later case, nonlinear equalizers employ previous decisions to eliminate the ISI caused by previously detected symbols on the current symbol to be detected. Figure 1.6 shows a block diagram of a feedforward equalizer which is a primary topic in this dissertation.

Equation (1.2) implies that if the channel impulse response were known, then the design of an equalizer would be straightforward. Therefore, channel estimation plays an important role in designing a channel equalizer. A channel can be estimated either in the frequency domain or in the time domain. The estimation can be performed by a direct estimation on the received signal or by a training sequence. The former method is called *blind estimation*.

Training sequences are not only used in channel estimation, but also used in equalizer training. Equalizer training adjusts the parameters of an equalizer filter and may involve either decision directed training or training sequences. Decision-directed equalizer training does not require any knowledge of the transmitted data *a priori*. The key disadvantages of decision directed training are slow convergence and the inability to track rapidly changing channel characteristics. The use of training sequences can overcome these drawbacks.

This chapter is organized as follows. Section 2.2 discusses methods for estimating the channel impulse response, including matrix pencil methods. Matrix pencil methods form the basis of a new blind estimation method for time-domain equalizer (TEQ) design in DMT systems that is presented in Chapter 5. Section 2.3 discusses feedforward neural network equalizers that can combat some types of nonlinear distortion. A neural network equalizer does not need to estimate the channel impulse response whereas a TEQ in DMT systems does. DMT modulation is described in Section 2.4 and current TEQ design methods are summarized in Section 2.5. A frequency domain equalizer is discussed in

Section 2.6. Section 2.7 concludes this chapter.

2.2 Channel estimation

Channel estimation is important in a digital communication system, especially when there is a little or no knowledge about the transmission channel. The equalization problem will be solved more easily if the channel impulse response is known to the equalizer. The channel estimation problem can be stated as follows: given samples of the received signal, $\{y_k, k = 1, 2, \cdots, N\}$, determine the channel impulse response $\mathbf{h}$. If samples of the input signal $\{s_k, k = 1, 2, \cdots, N\}$ are not available, then it is called blind channel estimation or blind identification.

2.2.1 Training sequences

In digital communication systems, the design of optimal receivers and fast startup equalizers requires channel estimation. Therefore, a known training sequence is transmitted to estimate the channel impulse response before data transmission. Training sequences are periodic or aperiodic. In either case, the power spectrum of the training sequence is approximately flat over the transmission bandwidth. The suggested training sequence for channel estimation in a DMT system is a pseudo-random binary sequence with N samples. The training sequence is made periodic by repeating N samples or adding a cyclic prefix. Tellambura, Guo, and Barton discuss aperiodic training sequences in [21].

The use of a training sequence reduces the transmission rate, especially when the training sequence has to be retransmitted often, e.g. for the fast channel variations that occur in mobile communications. Current research on training sequences includes the design of training sequences that optimize an objective function for a channel estimator. A time-domain optimization method is introduced in [22]. A disadvantage of the time-domain method is that an exhaustive search for the optimal training sequence of length N requires 2^N possible sequences. A frequency-domain method is proposed to reduce the computational cost by introducing a gain loss factor in [23]. However, the frequency-domain method cannot always find the optimal periodic training sequence in terms of the mean-squared channel estimation error [23].

2.2.2 Channel estimation methods

The goal in channel estimation is to estimate an L_h-tap channel impulse response

$$\mathbf{h} = [h(1), h(2), \cdots, h(L_h)]^T,$$

where ^{T} is the transpose. The received signal $\mathbf{y}$ in vector form is of length N and given by

$$\mathbf{y} = \mathbf{S}\mathbf{h} + \mathbf{n},$$

where $\mathbf{S}$ is a $N \times L_h$ matrix containing the transmitted symbols $\{s_k,\ k = 0, 1, \cdots, N-1\}$ given by

$$\mathbf{S} = \begin{bmatrix} s_0 & s_{N-1} & \cdots & s_{N-L_h+1} \\ s_1 & s_0 & \cdots & s_{N-L_h+2} \\ \vdots & \vdots & \ddots & \vdots \\ s_{N-1} & s_{N-2} & \cdots & s_{N-L_h} \end{bmatrix} \tag{2.1}$$

This matrix has Toeplitz symmetry. The vector $\mathbf{n}$ is a vector of samples of an additive white Gaussian noise process with variance σ^2 and is independent of the transmitted signal and the channel.

One time-domain method estimates the channel impulse response based on a least-squares approach [22]. The resulting channel estimate is

$$\hat{\mathbf{h}} = \left(\mathbf{S}^T\mathbf{S}\right)^{-1}\mathbf{S}^T\mathbf{y}.$$

The mean-squared error (MSE) for the time-domain case is given by

$$\text{MSE}_{\text{time-domain}} = \sigma^2 \ \text{Tr}\left[\left(\mathbf{S}^T\mathbf{S}\right)^{-1}\right],$$

where $\text{Tr}(\cdot)$ denotes the trace of the matrix. The trace of a square matrix is the sum of the entries down the leading diagonal [24] or the sum of the eigenvalues of the matrix. In comparison, a frequency-domain method estimates the channel impulse response as [25]

$$\widehat{h}_k = \frac{1}{N}\sum_{n=0}^{N-1}\left(\frac{Y_n}{S_n}\right)e^{j2\pi kn/N}, \ k = 0, 1, \cdots, L_h - 1,$$

where Y_n and S_n are the N-point discrete Fourier transforms (DFT) of y_k and s_k in Figure 1.5, respectively. The corresponding MSE is given by

$$\text{MSE}_{\text{frequency-domain}} = \sigma^2 \frac{L_h}{N}\sum_{n=0}^{N-1}\frac{1}{\mid S_n \mid^2}.$$

In general, both periodic and aperiodic training sequences can be designed to prevent S_n from being zero or near zero.

Since the least-squares time-domain method is sensitive to noise when the SNR is low, alternative reduced rank channel estimation methods have been proposed that use singular value decomposition (SVD) [26, 27]. The frequency-domain channel estimation method is currently used in the DMT system to estimate the channel impulse responses of the CSA DSL loops [6]. Since the cyclic prefix is used to combat ISI, Wang and Liu propose a time-domain joint channel estimation and equalization algorithm by using the cyclic prefix [28]. Their method is suitable to track variations in a moderately time-varying channel.

2.2.3 Matrix pencil methods

The matrix pencil method may be used for channel estimation. The matrix pencil method can estimate poles when a channel is modeled as having an infinite impulse response. The problem of estimating poles (or damping factors and frequencies) from exponentially damped/undamped sinusoids has drawn great attention for both practical and theoretical interests [13, 15, 29, 30, 31, 32]. Estimating signal parameters has many diverse applications, such as determination of direction-of-arrival plane waves at a uniform linear array of sensors [33] and high-resolution spectral estimation [34].

The characteristic impedance of a transmission line can be written as [35]

$$Z_0 = \sqrt{\frac{L}{C}} \left(1 - j \frac{R}{4\pi f L} \right),$$

where L is inductance, R is resistance, and C is capacitance. As the frequency f increases, the characteristic impedance decreases in absolute value. Hence, a transmission line has a lowpass response.

A wireline channel can be modeled as an infinite impulse response (IIR) filter [36]. The transfer function of an IIR filter is given by

$$H(z) = \frac{B(z)}{A(z)} = \frac{B(z)}{\prod_{m=1}^{M} \left(1 - e^{p_m} z^{-1} \right)} \tag{2.2}$$

where

$$p_m = -d_m + j 2\pi f_m \tag{2.3}$$

I assume that the IIR filter model of a wireline channel does not have duplicate poles. The all-pole portion of (2.2) is

$$h_{\text{ap}}(z) = \frac{1}{\prod_{m=1}^{M} \left(1 - e^{p_m} z^{-1} \right)} \tag{2.4}$$

The all-pole portion can also be rewritten as

$$h_{\text{ap}}(z) = \sum_{m=1}^{M} \frac{a_m}{1 - e^{p_m} z^{-1}} \tag{2.5}$$

In the time-domain, (2.5) gives

$$h_{\text{ap}}(n) = \sum_{m=1}^{M} a_m \ e^{p_m n} \ u(n) \tag{2.6}$$

The problem of estimating signal parameters from a noisy observed data sequence of K samples is considered as

$$r_k = s_k + n_k, \ k = 0, 1, \ldots, K-1. \tag{2.7}$$

where the noise term n_k is a complex white Gaussian random process, and the noise-free signal s_k is given by

$$s_k = \sum_{m=1}^{M} a_m e^{p_m k}, \text{ where } p_m = -d_m + j2\pi f_m. \tag{2.8}$$

for $k = 0, 1, \ldots, K-1$, where M is the number of exponentially damped sinusoids, and the a_m, d_m, and f_m terms in (2.3) represent the complex amplitudes, damping factors, and frequencies, respectively, which are the unknown signal parameters to be estimated. The set of amplitude terms $\{a_m\}$ can be estimated by solving a linear least-squares problem if all of the other parameters are known [13]. A matrix pencil method can be used to estimate the set of damping factors $\{d_m\}$ and frequencies $\{f_m\}$ (see Section 2.2.4). Section 2.2.5 discusses methods to estimate the number of sinusoids M.

2.2.4 Hua and Sarkar's matrix pencil method

The matrix pencil method is derived from the property of the underlying signal to estimate the damping factors $\{d_m, m = 1, 2, \cdots, M\}$ and frequencies $\{f_m, m = 1, 2, \cdots, M\}$. Let $\mathbf{A}$ and $\mathbf{B}$ be two matrices. The set of all matrices of the form $\mathbf{A} - \beta\mathbf{B}$, where β is an indeterminate and not a particular real value, is said to be a *matrix pencil.* From the received signal in (2.7), vector $\mathbf{r}_l$ is defined as

$$\mathbf{r}_l = \left[r_l^*, r_{l+1}^*, \cdots, r_{K-L+l-1}^* \right], \text{ for } l = 0, 1, \cdots, L \tag{2.9}$$

where * denotes the complex conjugate. The master matrix is formed as

$$\mathbf{R} = [\mathbf{r}_0, \mathbf{r}_1, \cdots, \mathbf{r}_L] = \begin{bmatrix} r_0 & r_1 & \cdots & r_L \\ r_1 & r_2 & \cdots & r_{L+1} \\ \vdots & \vdots & \ddots & \vdots \\ r_{K-L-1} & r_{K-L} & \cdots & r_{K-1} \end{bmatrix}_{(K-L)\times(L+1)} \tag{2.10}$$

by making full use of the available data samples. L is the *pencil parameter* which satisfies $M \leq L \leq K - M$. We partition $\mathbf{R}$ in (2.10) into the matrix pencil $\mathbf{R}_0 - \beta\mathbf{R}_1$ where

$$\begin{aligned} \mathbf{R}_0 &= [\mathbf{r}_0,\ \mathbf{r}_1,\ \cdots,\ \mathbf{r}_{L-1}]_{(K-L)\times L} \\ \mathbf{R}_1 &= [\mathbf{r}_1,\ \mathbf{r}_2,\ \cdots,\ \mathbf{r}_L]_{(K-L)\times L} \end{aligned} \tag{2.11}$$

Similarly, we can define the signal matrices $\mathbf{S}$, $\mathbf{S}_0$, and $\mathbf{S}_1$ from $\{s_k\}_{k=0}^{K-1}$, where s_k is given in (2.8). All three matrices $\mathbf{S}$, $\mathbf{S}_0$, and $\mathbf{S}_1$ have *Hankel* structure and *low-rank* with rank M. Then, $\{\exp(-p_m)\}_{m=1}^{M}$ is the set of M positive eigenvalues, $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_M$, of $\mathbf{S}_0^\dagger \mathbf{S}_1$, where † denotes the pseudoinverse, and

$$p_m = -\ln(\lambda_m),\ m = 1, 2, \cdots, M. \tag{2.12}$$

The eigenvalues $\{\lambda_i, i = 1, 2, \cdots, M\}$ are the values of β for which the columns of the matrix pencil, $\mathbf{S}_0 - \beta\mathbf{S}_1$, become dependent.

In the presence of noise, the matrices $\mathbf{R}$, $\mathbf{R}_0$, and $\mathbf{R}_1$ defined in (2.10) and (2.11), respectively, are full rank but still have Hankel structure. Hua and Sarkar [13, 37, 38, 39] perform an SVD and its rank M truncation of the M largest singular values of the noisy data matrices $\mathbf{R}_0$ and $\mathbf{R}_1$ to reduce the

Step 1	Form the master matrix $\mathbf{R}$ as in (2.10)
Step 2	Form matrices $\mathbf{R}_0$ and $\mathbf{R}_1$ as in (2.11)
Step 3	Compute $\widetilde{\mathbf{S}}_0^\dagger = \mathcal{L}^\dagger\{\mathbf{R}_0\}$ from (2.14)
Step 4	$\widetilde{\mathbf{S}}_1 = \mathcal{L}\{\mathbf{R}_1\}$ from (2.13)
Step 5	Calculate the M non-zero eigenvalues of $\widetilde{\mathbf{S}}_0^\dagger\widetilde{\mathbf{S}}_1$ as $\{\exp{(d_m - j2\pi f_m)}\}_{m=1}^M$

Figure 2.1: Implementation of original matrix pencil method.

effect of noise. For a given matrix $\mathbf{X}$ of rank L, I define the rank M ($M \leq L$) approximation operator $\mathcal{L}$ as

$$\mathcal{L}\{\mathbf{X}\} = \sum_{m=1}^{M} \sigma_m \mathbf{u}_m \mathbf{v}_m^H \tag{2.13}$$

where $\{\sigma_i,\ i = 1, 2, \cdots, M\}$ is the set of the M largest singular values of $\mathbf{X}$, $\mathbf{u}_m$ and $\mathbf{v}_m$ are the corresponding left and right singular vectors, respectively, and the superscript ^{H} is the matrix conjugate transpose. I also define the rank M ($M \leq L$) pseudoinverse operator $\mathcal{L}^\dagger$ as

$$\mathcal{L}^\dagger\{\mathbf{X}\} = \sum_{m=1}^{M} \frac{1}{\sigma_m} \mathbf{u}_m \mathbf{v}_m^H \tag{2.14}$$

The procedure for finding $\{\exp{(d_m - j2\pi f_m)}\}_{m=1}^M$ is given in Figure 2.1 [13].

2.2.5 Estimation of the number of poles

The matrix pencil method in Section 2.2.4 assumes that the number of sinusoids, M, in (2.13) and (2.14) is known. However, the rank M of the signal matrices is not known *a priori*. In the noise-free case, the three data matrices $\mathbf{S}$, $\mathbf{S}_0$, and $\mathbf{S}_1$ have low rank. Then, M is equal to the number of non-zero eigenvalues of matrix $\mathbf{S}$. When noise is present, the matrices $\mathbf{R}$, $\mathbf{R}_0$, and $\mathbf{R}_1$ are *full* rank and the accuracy of estimating the number of poles decreases as n_k in (2.7) gets large.

In the presence of noise, all of the current methods used to estimate the number of poles may be classified into two categories. One is a class of the subjective-based methods, and the other is a group of objective-based methods. A subjective-based method involves the designer's decision, which is difficult to automate. The representatives of subjective-based methods are Fuchs' criterion based on perturbation analysis [40]; Bartlett's χ^2 approximations [41]; and Lawley's test on significance of the latent roots [42]. The representatives of subjective-based methods are listed in [43]. Although objective-based methods can be implemented in practice, most have high computational cost and give good performance only at large SNR. The candidates of the objective-based methods include the model selection methods proposed by Akaike [44], Schwartz [45], and Rissanen [46] and the information theoretic criteria developed by Wax and Kailath [47] and Reddy and Biradar [48]. Sano and Tsuji [49] develop an objective-based method from perturbation analysis. Their method introduces a set of *regularization parameters* and estimates the number of poles more accurate than the methods in [44] and [47].

A set of regularization parameters, $\{\mu_i\}$, is added to (2.14) so that the regularized SVD truncated pseudoinverse is described by

$$\mathcal{L}\{\widehat{\mathbf{X}}_R^\dagger\} = \sum_{m=1}^{r} \frac{1}{\sigma_m + \mu_m} \mathbf{v}_m \mathbf{u}_m^H \tag{2.15}$$

where r is the rank of $\mathbf{R}_0$. If $\mu_i = 0$ for $i = 1, 2, \cdots, M$, and $\mu_i \to \infty$ for $i = M+1, \cdots, r$, then (2.15) is identical to (2.13). The MSE criterion is used to determine the optimal values of $\{\mu_i\}$. The number of μ_i's less than a threshold is the number of poles.

2.3 Feedforward neural network equalizers

A neural network is essentially an interconnected assembly of simple processing elements, called *units or nodes*, whose functionality is loosely based on human neurons. The processing ability (knowledge) of the network is stored in the inter-unit connection strengths, called *weights*, which are obtained by a process of adaptation to, or learning from, a set of training patterns. If learning is accomplished by presenting a sequence of training patterns (inputs), each of which is associated with a target (output), then the weights are adjusted by a learning algorithm. This process is called *supervised learning.* If no target is available, then the process is called *unsupervised learning.*

I describe the biological neuron and a mathematical model of a biological neuron in Section 2.3.1. Sections 2.3.2 and 2.3.3 discuss two feedforward neural networks — multilayer perceptron and radial basis function networks. The learning algorithms for the two neural networks are also discussed. Section 2.3.4 reviews the applications of neural networks as channel equalizers in telecommunication system.

2.3.1 Model of a neuron

Figure 2.2(a) shows a biological neuron. The dendrites carry the signals from other neurons. A chemical process occurs at the synaptic site to scale the signals. Once the signals are greater than a threshold, the neuron fires and broadcasts the output signal to other neurons. Figure 2.2(b) shows a mathematical model, where $x_1, x_2, \ldots, x_N$ are the input signals and $w_1, w_2, \ldots, w_N$ are the scaling factors (weights) at the synaptic site. Artificial neural networks (ANNs), which are mathematical models of human cognition or biological neural networks, are based on the following assumptions [50]:

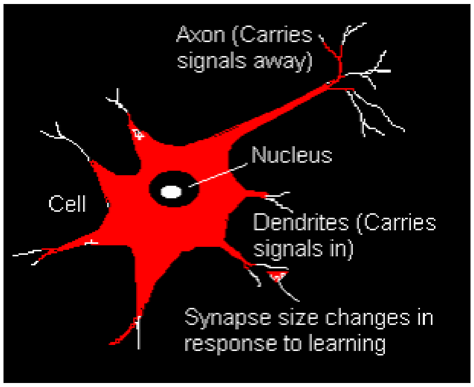
1. information processing occurs at many simple elements called neurons or nodes;
2. signals are passed between neurons over connection links;
3. each connection link has an associated weight; and
4. each neuron applies an activation function to determine the output.

The artificial neuron in Figure 2.2(b) sums the weighted inputs, applies an activation function f to the weighted sum, and passes the result to the other neurons on output y:

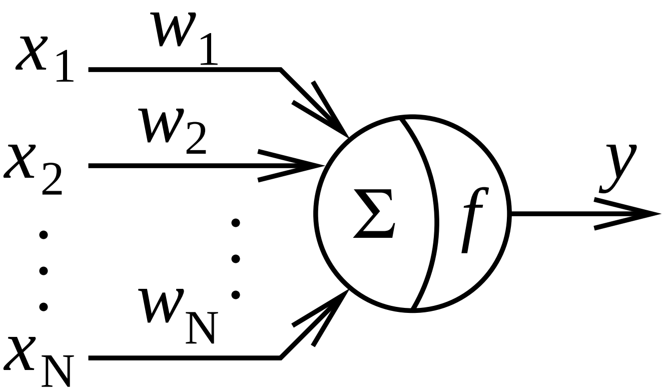
$$y(x_1, x_2, \cdots, x_N) = f\left(\sum_{i=1}^{N} w_i x_i\right) \tag{2.16}$$

The activation function f in (2.16) is used to limit the amplitude of the output of a neuron, e.g. to the interval $[0, 1]$ or $[-1, 1]$, which determines which activation function is used in an application. The activation function is required to be differentiable everywhere since the derivatives of the activation functions are used during the training process. Possible activation functions are

- linear: $f_{linear}(v) = k\, v$
- sigmoid: $f_{sigmoid}(v) = \dfrac{1}{1 + e^{-v}}$
- hyperbolic tangent: $f_{tanh}(v) = \tanh(v) = \dfrac{1 - e^{-v}}{1 + e^{-v}} = 2\, f_{sigmoid}(v) - 1$


Axon (Carries signals away)
Nucleus
Cell
Dendrites (Carries signals in)
Synapse size changes in response to learning

(a) Model of a biological neuron



(b) Mathematical model

Figure 2.2: Models of a biological neuron and artificial neurons.

A feedforward neural network consists of the nodes shown in Figure 2.2(b) arranged in different layers: input layer, hidden layer, and output layer (see Figure 2.3). A feedforward neural network can be described as $O - P_1 - P_2 - \cdots - P_k - Q$ where O is the number of neurons in the input layer, P_i is the number of neurons in the ith hidden layer, and Q is the number of neurons in the output layer.

The following two sections discuss two commonly-used feedforward neural networks — multilayer perceptron (MLP) networks and radial basis function (RBF) networks. MLP and RBF networks are similar. It is common that both

MLP networks and RBF networks employ supervised learning. On the output layer of each neural network, a target vector is presented. Minimum mean squared error and least mean-squares methods are used to minimize the difference between the target vector and output vector. Then, a learning algorithm adjusts weights between two adjacent layers. One major difference is the way in which hidden units combine values from preceding layers in the network. MLPs use inner products, as in (2.16), whereas RBFs use Euclidean distances. Another difference is that an MLP network may have more than one hidden layer while an RBF network has only one hidden layer.

2.3.2 Training of multilayer perceptron (MLP) networks

A multilayer perceptron (MLP) may have more than one hidden layer. The neurons in the hidden layer commonly use either sigmoid or hyperbolic tangent activation functions which introduce nonlinearity into the network. It is the nonlinearity of the activation functions that makes multilayer networks able to perform a variety of nonlinear mappings, e.g. in pattern recognition [51, 52]. The activation function for the output layer may be linear, sigmoid, or hyperbolic tangent, which are defined in Section 2.3.1.

Many training algorithms exist for MLPs such as back-propagation, second-order optimization, and hybrid linear-nonlinear training methods. Although batch back-propagation is widely used [12], it suffers from slow convergence and can be trapped in local minima. Modified back-propagation algorithms overcome both drawbacks [53]. The generalized back-propagation algorithm accelerates convergence by magnifying the backward propagated error [54, 55]. The Levenberg-Marquardt algorithm also increases the convergence rate [12].

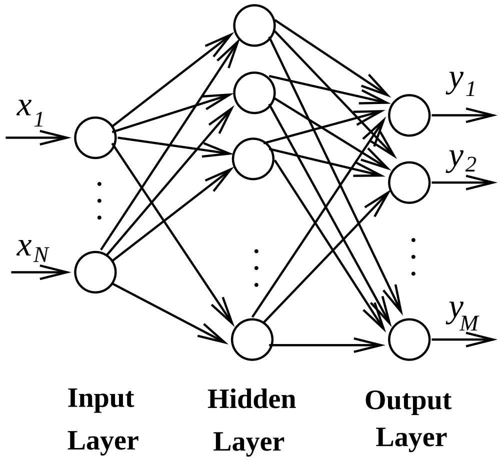


Figure 2.3: A feedforward neural network. Each circle represents a neuron which sums the inputs and passes the sum through an activation function. Each arc represents multiplication by a scalar weight.

Second-order optimization techniques, such as the conjugate gradient method, scaled conjugate gradient method, and quasi-Newton method, can also increase the convergence rate [12]. A hybrid linear-nonlinear training algorithm [53] is able to escape from poor local minima. All algorithms initialize the weights using random values which are commonly between -0.5 and 0.5, inclusive.

The activation function in real-valued neural networks is bounded and differentiable everywhere. However, in the complex case, any analytic function cannot be bounded unless it is constant [56]. Complex-valued backpropagation algorithms were developed independently by several researchers [57, 58, 59]. For an MLP, these algorithms suffer from slower convergence than in the real-valued case and may produce unpredictable learning skill or improper feature extraction. An additional problem of complex-valued MLP networks is to choose an activation function.

2.3.3 Training of radial basis function (RBF) networks

Since an MLP may not converge to a global minimum, radial basis function (RBF) networks offer an alternative. RBF networks have only three layers: one input, one hidden, and one output layers. The kth output is given by

$$y_k(\mathbf{x}) = \sum_{i=1}^{N_h} w_{k,i}\ \phi_i\left(\mathbf{x}\right) \tag{2.17}$$

where N_h is the number of neurons in the hidden layer, $\mathbf{x} = [x_1, x_2, \cdots, x_{N_h}]^T$, and $\phi_i\left(\mathbf{x}\right)$ is the radial basis function which is a radially symmetric scalar function with N_h centers. A commonly used radial basis function is a Gaussian function

$$\phi_i(\mathbf{x}) = \exp\left(-\frac{\parallel \mathbf{x} - \mathbf{c}_i \parallel^2}{2\sigma_i^2}\right) \tag{2.18}$$

where $\parallel \cdot \parallel$ is a vector norm (usually Euclidean), σ_i is the variance, and $\mathbf{c}_i$ is the center location. A radial basis function is local in character — its response to the input $\mathbf{x}$ drops off quickly for input values that are away from the center of the activation function's receptive field, $\mathbf{c}_i$.

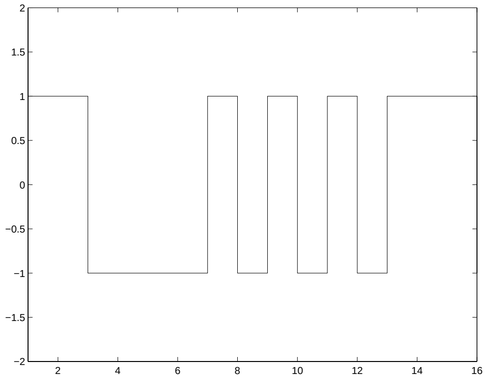
Training takes two steps [12]. In the first step, the variance terms σ_i's and center locations $\mathbf{c}_i$'s are calculated solely from input data. There are two kinds of training methods to find the σ_i's and $\mathbf{c}_i$'s [12]. One is unsupervised training which does not require target data. Subsets of data points, orthogonal least squares, clustering algorithms, and Gaussian mixture models belong to the unsupervised training methods. I calculate the σ_i's and $\mathbf{c}_i$'s by using expectation maximization (EM) which uses Gaussian mixture models. On the other hand, supervised training can find the parameters of the basis function as well as the weights between the hidden layer and output layer. However, the computation of the basis function parameters by supervised training requires a nonlinear optimization method which is computationally intensive [12].

In the second step, the weights between the hidden and output layers are determined while the parameters for the basis functions are kept fixed. In this case, (2.17) is the sum of weighted ϕ_i terms. The weights $w_{k,i}$ are calculated by minimizing a suitable function between y_k and target signal. I calculate them by using least mean squares (LMS). This two-step training method is referred as *hybrid* training. Hybrid training is much faster than typical nonlinear optimization techniques used in MLP training since the computation reduces to a linear or generalized linear model once the centers and widths are fixed. The output weights can be learned efficiently. Hybrid training, however, usually requires more hidden units than supervised learning. The number of hidden units needed by hybrid methods becomes a serious problem as the number of inputs increases. The required number of hidden units tends to increase *exponentially* with the number of inputs [12]. Therefore, the RBF networks are sensitive to massive perturbations (e.g., very noisy environments) in the training data.

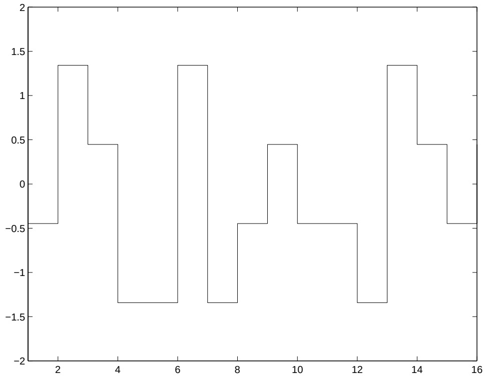
The complex RBF network is an extension of the real RBF network [60, 61]. Therefore, the training algorithms used in real RBF networks could be employed in complex RBF networks. The inputs and outputs of complex RBF networks are complex while each neuron in the hidden layer still keeps the real radially symmetric function [61].

2.3.4 Neural networks as channel equalizers

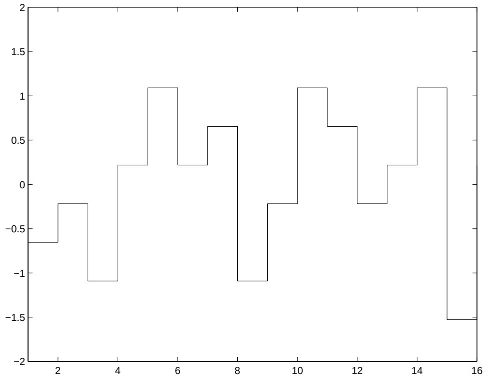
Consider a sequence of binary numbers taken from the set $\{1, -1\}$ that passes through a two-tap channel. The output from the channel shows two clusters



(a) 2-PAM



(b) 4-PAM



(c) 8-PAM

Figure 2.4: Pulse amplitude modulation signals, where n-PAM means that the amplitude of the message signal is quantized to n levels ($\log_2 n$ bits).

around $+1$ and -1 when noise is free [62, 63, 64, 65, 66]. However, as the SNR decreases, these two cluster may get closer to one large cluster. A channel equalizer classifies the received signal into different classes. Thus, channel equalization can be considered as a classification problem. This motivates the use of neural networks in channel equalization.

Gibson, Siu, and Cowan first demonstrated that an MLP equalizer works for 2-PAM [62, 66, 67], where PAM stands for *Pulse Amplitude Modulation.* 2-PAM means that the amplitude of the message signal is quantized to two levels (one bit). Figure 2.4 shows examples of 2-PAM, 4-PAM, and 8-PAM signals. Peng, Nikias, and Proakis develop new activation functions for multilayer perceptron equalizers for 4-PAM and 8-PAM constellation [68, 69, 70]. Their simulations show that MLP equalizers outperform linear equalizers in terms of symbol error rate (SER) vs. SNR. A *symbol error* occurs when the detector chooses a data symbol which is different from the actual transmitted symbol. The symbol error rate (SER) is given by

$$\mathrm{SER} = \frac{\text{total number of symbol errors}}{\text{total number of transmitted symbols}}$$

Peng, Nikias, and Proakis use MLPs with two hidden layers. The hidden neurons use nonlinear sigmoid activation function to create intricately curved partitions of space. However, using more than one hidden layer may not guarantee a local-minima-free network [71].

Chen, Gibson, Cowan, and Grant first showed that the RBF equalizer outperforms the linear equalizer [72] in terms of symbol error rate. Mulgrew gave a fundamental analysis on the application of RBF networks in equalization [64]. Placement of the centers of RBF equalizers determines their performance because the number of cluster centers equal the number of hidden units in RBF networks. Mulgrew concludes that the use of RBF networks provides more controllable training characteristics than MLP networks. However, the training period for both neural networks is still too long for a practical real-time implementation.

In the case of QAM, complex neural networks are required. You and Hong apply complex MLP networks to both training-based equalization [73] and blind equalization [74]. They discuss important properties for complex activation functions in the complex MLP and show that neural networks are most suitable to compensate for nonlinear distortion. For 4-QAM, they use 15 input neurons, 9 neurons in the hidden layer, and 1 neuron in the output layer. The training of this MLP network is performed off-line. Chen, McLaughlin, and Mulgrew study the complex-valued RBF network [75] and its application to equalizer design [76]. They show that the RBF equalizer is identical to the Bayesian solution to the symbol-based equalizer. For their complex-valued neural network, they use a real-valued activation function in their simulations.

Mulgrew [64] points out that the number of symbols required to train a neural network equalizer is

$$\text{number of symbols for training} = M^{L_h+N_{in}-1}, \tag{2.19}$$

where M stands for the number of constellations, L_h is the length of the channel impulse response, and N_{in} is the number of neurons in the input layer. Although it can help us decide the number of symbols to be used, it grows exponentially as length of channel impulse response increases. In general, the length of channel impulse response is an unknown parameter.

2.4 Discrete multitone (DMT) modulation

Discrete Multitone (DMT) modulation [6] divides the usable bandwidth of a channel into a number of independent subchannels over which data is modulated. A common subchannel modulation method is quadrature amplitude modulation. The American National Standards Institute (ANSI) has approved DMT as the standard modulation method for ADSL. DMT is also being considered for newer cable modem and digital broadcast radio systems [77].

DMT is a form of multicarrier modulation in which the original spectrum of the input signal is spread over numerous bands, called *subchannels*. Each of the subchannels is independently modulated to a carrier frequency. Figure 2.5 shows a multicarrier transmitter [78] and Figure 2.6 shows a block diagram of the receiver. In a DMT system, the frequency response of the channel is partitioned into equal bandwidth subchannels [79] as shown in Figure 2.7. The goal is to keep the frequency response across each subchannel approximately constant in order to eliminate the need for equalization. In practice, the subchannels are not completely independent.

A number of time-domain samples are prepended to each DMT symbol. This cyclic prefix is added to help the receiver remove intersymbol interference caused by the channel. In order for the receiver to remove ISI, the length of the channel impulse response has to be less than or equal to $(\nu + 1)$ samples where ν is the length of cyclic prefix. Time-domain equalization is used in ADSL receivers to shorten the channel impulse response length as discussed next.

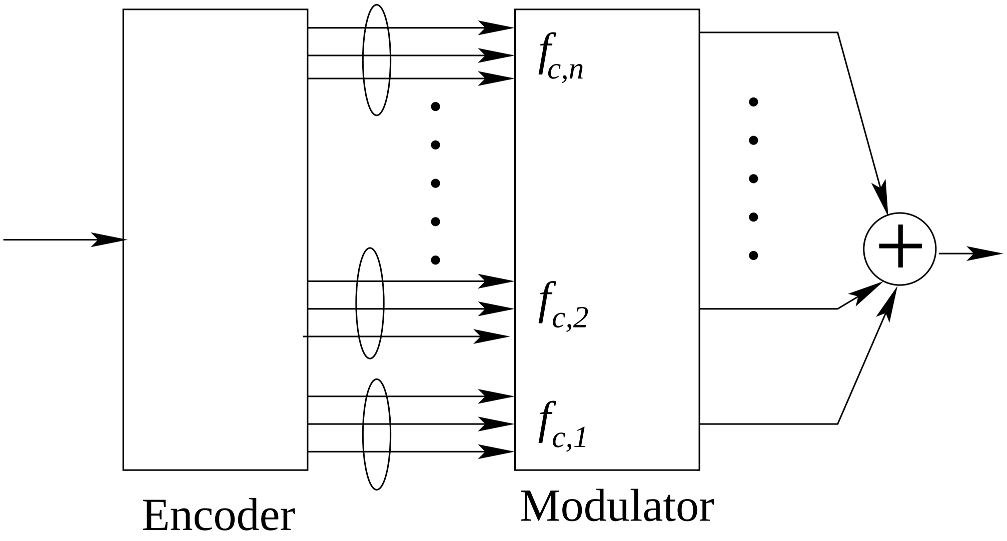


Figure 2.5: Multicarrier modulation transmitter.

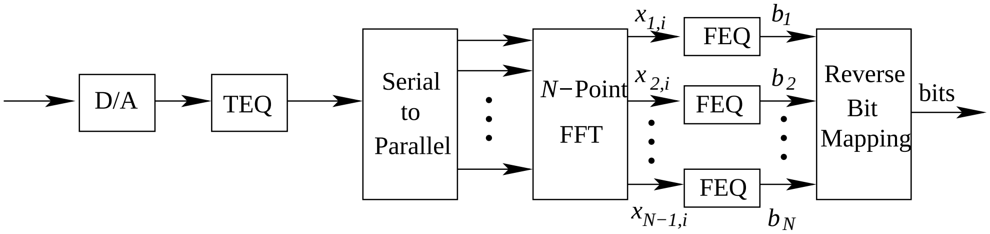


Figure 2.6: Multicarrier modulation receiver.

2.5 Time-domain equalization (TEQ) for DMT channels

A DMT symbol consists of N samples, and the cyclic prefix that is prepended to the symbol is a copy of the last ν samples of the DMT symbol. The following example motivates the need for channel shortening. Suppose that a DMT symbol contains $N = 4$ samples. They are $\mathbf{s}_k = [s_{k,0}, s_{k,1}, s_{k,2}, s_{k,3}]^T$ where $\mathbf{s}_k$ is shown in Figure 1.5. The channel impulse response $\mathbf{h}$ has an extent of two samples, i.e. $\mathbf{h} = [h_0, h_1]$. With the length of channel impulse response being $L_h = \nu + 1 = 2$

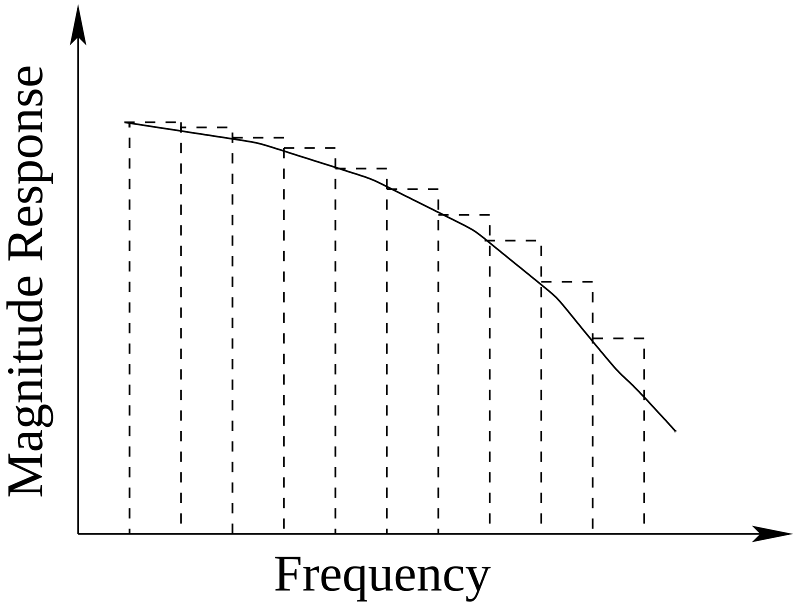


Figure 2.7: Discrete multitone modulation subchannels.

the cyclic prefix length ν is 1. Then, the DMT symbol $\mathbf{s}_k$ with the cyclic prefix is $[s_{k,3}, s_{k,0}, s_{k,1}, s_{k,2}, s_{k,3}]^T$. In Figure 1.5, $\mathbf{x}_k$ is the convolution between $\mathbf{s}_k$ with cyclic prefix and the two-tap channel, which can be written in matrix form as

$$\mathbf{x}_k = \begin{bmatrix} s_{k,3} & 0 \\ s_{k,0} & s_{k,3} \\ s_{k,1} & s_{k,0} \\ s_{k,2} & s_{k,1} \\ s_{k,3} & s_{k,2} \\ 0 & s_{k,3} \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \end{bmatrix} = \mathbf{S}_k \mathbf{h} \tag{2.20}$$

Extracting N rows from matrix $\mathbf{S}_k$ forms a circulant matrix $\mathbf{C}_k$

$$\mathbf{C}_k = \begin{bmatrix} s_{k,0} & s_{k,3} \\ s_{k,1} & s_{k,0} \\ s_{k,2} & s_{k,1} \\ s_{k,3} & s_{k,2} \end{bmatrix}$$

Then, $\mathbf{C}_k\mathbf{h}$ is equivalent to the circular convolution between the first column of $\mathbf{C}_k$ and $\mathbf{h}$. Based on the circular convolution property of discrete Fourier transform (DFT), the frequency-domain transform of $\mathbf{C}_k\mathbf{h}$ is $\mathbf{\Lambda}_k\mathbf{H}$ where $\mathbf{\Lambda}_k$ is a diagonal matrix with the DFT of the first column of $\mathbf{C}_k$ as its diagonal, and $\mathbf{H}$ is the N-point DFT of $\mathbf{h}$. The orthogonality of $\mathbf{\Lambda}_k$ ensures that the cyclic prefix serves as a guard sequence to combat ISI between two consecutive symbols. This example shows that $L_h = \nu + 1$ can guarantee that ISI can be eliminated, which can be generalized as $L_h \leq \nu + 1$ where L_h is the length of channel impulse response. In the ANSI ADSL standard, ν is 32 [6]. Most of the wireline telephone channels have impulse responses with significant energy beyond $\nu + 1 = 33$ samples at ADSL sampling rate of 2.208 MHz. Therefore, it is desirable to shorten the channel impulse response to be within $\nu + 1$ samples in length.

Since the ν samples in the cyclic prefix do not convey any new information about the transmitted symbol, the efficiency of a DMT transceiver is decreased by a factor of $N/(N+\nu)$. Efficiency increases with increasing N or decreasing ν. Increasing N increases memory and computational requirements in a transceiver, and choosing ν *a priori* is difficult because the channel length L_h is generally not known. For a smaller value of ν to be used, a time-domain equalizer (TEQ), a.k.a. a channel shortening equalizer, is required to shorten the effective length of a channel [17].

Section 2.5.1 presents the mean squared error (MSE) [80] objective function and TEQ design methods based on minimizing the MSE. Minimizing the MSE is equivalent to maximizing the SNR at the TEQ output. Section 2.5.2 reviews the shortening SNR (SSNR) objective function and methods to maximize the SSNR. Section 2.5.3 highlights TEQ design methods for maximizing channel capacity.

2.5.1 Minimizing mean squared error (MSE) design

Chow and Cioffi [81] propose an FIR TEQ to shorten the channel impulse response from L_h samples to $(\nu + 1)$ samples, as shown in Figure 2.8. The kth sample of the error signal is

$$e(k) = \mathbf{w}^T \mathbf{y}_k - \mathbf{b}^T \mathbf{x}_{k-\Delta} \tag{2.21}$$

where $\mathbf{w}$ is the column vector of N_w taps of the TEQ impulse response; $\mathbf{y}_k = [y(k), \cdots, y(k - N_w + 1)]^T$ is a vector of N_w samples of the received signal; $\mathbf{b}$ is the column vector of $(\nu + 1)$ values of the target impulse response (TIR); and $\mathbf{x}_{k-\Delta} = [x(k - \Delta), x(k - \Delta - 1), \cdots, x(k - \Delta - \nu)]^T$ is a vector of $(\nu + 1)$ samples of the input training sequence. In the ideal case, the convolution of the channel and TEQ impulse responses yields

$$\mathbf{b}_{\text{ideal}} = \begin{bmatrix} \mathbf{0}_{\Delta \times 1} \\ \mathbf{b}_{(\nu+1) \times 1} \\ \mathbf{0}_{(L_h + N_w - \Delta - \nu - 2) \times 1} \end{bmatrix} \tag{2.22}$$

Thus, $\mathbf{b}_{\text{ideal}}$ contains $(\nu + 1)$ consecutive samples of the TIR $\mathbf{b}$ with a delay of Δ samples. The TIR extent should be less than or equal to $(\nu + 1)$ samples. In practice, the lower path in Figure 2.8 containing the delay Δ and TIR $\mathbf{b}$ is not physically implemented. The values of ν and the TEQ length N_w may be set *a priori* and do not vary with the channel.

The mean squared error is defined as

$$\text{MSE} = E\left\{e^2(k)\right\} \tag{2.23}$$

where $E\{\cdot\}$ is the expectation operator. Substituting $e(k)$ in (2.21),

$$\text{MSE} = E\left\{\left(\mathbf{w}^T \mathbf{y}_k - \mathbf{b}^T \mathbf{x}_{k-\Delta}\right)^2\right\} \tag{2.24}$$

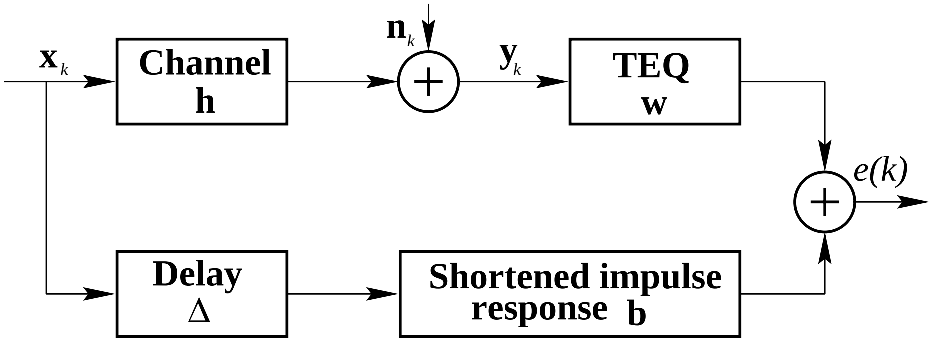


Figure 2.8: Time-domain equalizer with vectors: $\mathbf{x}$ of input training sequence data, $\mathbf{n}$ of additive noise samples, $\mathbf{y}$ of the the received signal, $\mathbf{h}$ of original channel impulse response data, $\mathbf{w}$ of N_w TEQ taps, and $\mathbf{b}$ of the shortened channel impulse response with ν taps, and Δ as the system delay of the overall response from both the channel and TEQ.

In order to minimize MSE in (2.24), the TEQ coefficient vector $\mathbf{w}$ and TIR vector $\mathbf{b}$ must satisfy

$$\mathbf{b}^T \mathbf{R}_{\mathbf{x}_{k-\Delta}\mathbf{y}_k} = \mathbf{w}^T \mathbf{R}_{\mathbf{y}_k\mathbf{y}_k} \tag{2.25}$$

where $\mathbf{R}_{\mathbf{x}_{k-\Delta}\mathbf{y}_k}$ is the $(\nu+1)\times(\nu+1)$ cross-correlation matrix between the delayed input sequence $\mathbf{x}_{k-\Delta}$ and the output sequence $\mathbf{y}_k$, and $\mathbf{R}_{\mathbf{y}_k\mathbf{y}_k}$ is the $N_w \times N_w$ output autocorrelation matrix. In this case,

$$\text{MSE} = \mathbf{b}^T \mathbf{R}_{\mathbf{x}_{k-\Delta}\mathbf{y}_k} \mathbf{b} \tag{2.26}$$

To avoid the trivial solution of $\mathbf{w} = \mathbf{0}_{N_w \times 1}$ and $\mathbf{b} = \mathbf{0}_{(\nu+1)\times 1}$ when minimizing the MSE in (2.26), one may apply either

1. a unit tap constraint [17, 82], in which one of the taps in $\mathbf{b}$ is set to 1, or
2. a unit norm constraint [16, 80, 83, 84], in which either $\| \mathbf{b} \|$ or $\| \mathbf{w} \|$ is normalized to 1, where $\| \cdot \|$ is the L_2-norm.

The unit norm constraint is usually preferred because the unit tap constraint adds another search direction which increases the computational complexity [80] and because designs produced by using a unit tap constraint have higher MSE than those produced by using a unit norm constraint [80].

With a unit norm constraint placed on $\mathbf{w}$, the design problem becomes

$$\min_{\Delta,\mathbf{b},\mathbf{w}} \mathbf{b}^T \mathbf{R}_{\mathbf{x}_{k-\Delta}\mathbf{y}_k} \mathbf{b}, \text{ subject to } \parallel \mathbf{w} \parallel = 1 \text{ and } \mathbf{b}^T \mathbf{R}_{\mathbf{x}_{k-\Delta}\mathbf{y}_k} = \mathbf{w}^T \mathbf{R}_{\mathbf{y}_k\mathbf{y}_k}$$

where $1 \leq \Delta \leq L_h + N_w - \nu - 2$. For a fixed value of Δ, the unique Minimum MSE (MMSE) solution follows the steps in Figure 2.9. As an alternative to direct eigenvalue decomposition, Nafie and Gatherer [83] use an iterative modified power method to reduce the computational cost.

Melsa, Younce, and Rohrs [11] define the effective channel impulse response $h_{\text{eff}}(k)$ as $h_{\text{eff}}(k) = h(k) * w(k)$, which in vector form becomes

$$\mathbf{h}_{\text{eff}} = [h_{\text{eff}}(1), h_{\text{eff}}(2), \cdots, h_{\text{eff}}(L_h + N_w - 1)] \tag{2.27}$$

Here, $\mathbf{h}_{\text{eff}}$ in (2.27) is ideally equal to $\mathbf{b}_{\text{ideal}}$ in (2.22) in that the samples of $\mathbf{h}_{\text{eff}}$ outside of a window of $(\nu + 1)$ samples starting at sample $(\Delta + 1)$ are zero. I rewrite the error signal as

$$\tilde{e}(k) = h_{\text{eff}}(k) - b_{\text{ideal}}(k), \; k = 1, 2, \ldots, L_h + N_w - 1 \tag{2.28}$$

Step 1	Compute $\mathbf{b}$ as the eigenvector corresponding to the minimum eigenvalue of $\mathbf{R}_{\mathbf{x}_{k-\Delta}\mathbf{y}_k}$ in (2.26)
Step 2	Calculate the TEQ taps $\mathbf{w}$ from (2.25)
Step 3	Normalize $\mathbf{w}$ to satisfy the unit norm constraint

Figure 2.9: Implementation of the minimum mean squared error method.

Minimizing $\tilde{e}(k)$ in (2.28) is equivalent to minimizing $e(k)$ in (2.21) if there is no noise in the channel [85].

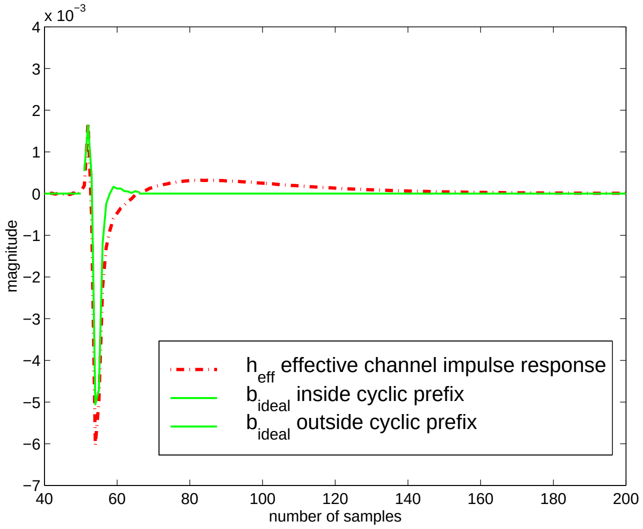
One can decouple the contribution of each component to the MSE as

$$\text{MSE} = \text{MSE}_{\text{win}} + \text{MSE}_{\text{wall}}$$

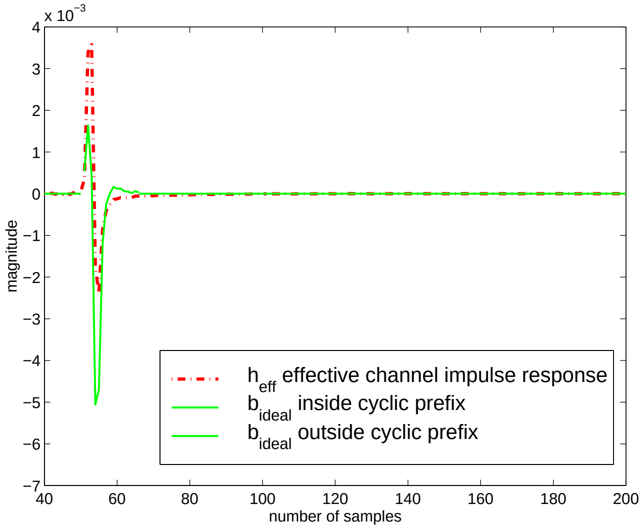
where MSE_{win} is the mean squared error inside the window of $(\nu + 1)$ samples and MSE_{wall} is that outside of the window. Although ISI is caused by the error outside of the window, the MMSE solution minimizes the error inside and outside the window. When compared to the MMSE solution, it is possible that there exists a solution with higher than or equal to MSE_{win} but with smaller MSE_{wall}, which would cause less ISI. We demonstrate this with the following example. Figure 2.10 shows an example of the equalization performed by a two-tap TEQ. In both cases, $\text{MSE} = 0.001$ and $\mathbf{w}$ satisfies the unit norm constraint. In Figure 2.10(a), $\text{MSE}_{\text{wall}} = 6.67 \times 10^{-9}$ with $\mathbf{w} = [0.4189, -0.9080]^T$, whereas in Figure 2.10(b), $\text{MSE}_{\text{wall}} = 6.83 \times 10^{-11}$ with $\mathbf{w} = [0.7226, -0.6913]^T$. Using the SSNR given by (2.30) below as the figure of merit, the MMSE method shown in Figure 2.10(a) yields 13.27 dB SSNR, whereas the method shown in Figure 2.10(b) yields 29.90 dB SSNR. The second method minimizes only MSE_{wall}. This is the maximum SSNR method given in the next section.

2.5.2 Maximizing shortening signal-to-noise ratio (SSNR) design

This section discusses an approach to maximize the *shortening signal-to-noise ratio* (SSNR) [11]. One of the advantages of the SSNR approach is that it does not rely on a TIR $\mathbf{b}$. If all of the samples of $\mathbf{h}_{\text{eff}}$ in (2.27) that are outside the window of $(\nu + 1)$ samples are negligible, then the impulse response of the



(a) $\mathrm{MSE}_{\mathrm{wall}} = 6.67 \times 10^{-9}$,
$\mathrm{SSNR} = 13.27$ dB, and $\mathbf{w} = [0.4189, -0.9080]^T$



(b) $\mathrm{MSE}_{\mathrm{wall}} = 6.83 \times 10^{-11}$,
$\mathrm{SSNR} = 29.90$ dB, and $\mathbf{w} = [0.7226, -0.6913]^T$

Figure 2.10: Comparison of effective channel $\mathbf{h}_{\mathrm{eff}}$ and $\mathbf{b}_{\mathrm{ideal}}$ between two cases with the same $\mathrm{MSE} = 0.001$. The channel is the dedicated data transmission channel with $L_h = 512$ and $\nu = 16$.

cascade of the channel and TEQ is effectively shortened. Splitting $\mathbf{h}_{\text{eff}}$ into two parts, $\mathbf{h}_{\text{win}}$ and $\mathbf{h}_{\text{wall}}$, separates the samples inside and outside the window [11]:

$$\begin{aligned} \mathbf{h}_{\text{win}} &= [h_{\text{eff}}(\Delta+1), h_{\text{eff}}(\Delta+2), \cdots, h_{\text{eff}}(\Delta+\nu+1)] \\ \mathbf{h}_{\text{wall}} &= [h_{\text{eff}}(1), \cdots, h_{\text{eff}}(\Delta), h_{\text{eff}}(\Delta+\nu+2), \cdots, h_{\text{eff}}(L_h+N_w-1)] \end{aligned} \tag{2.29}$$

The samples in $\mathbf{h}_{\text{wall}}$ include the samples before the window and the samples after the window (a.k.a. the tail). The SSNR objective function [11] is

$$\text{SSNR} = 10\log_{10}\frac{\text{Energy in } \mathbf{h}_{\text{win}}}{\text{Energy in } \mathbf{h}_{\text{wall}}} \tag{2.30}$$

If the energy in $\mathbf{h}_{\text{win}}$ is kept constant, then minimizing the energy in $\mathbf{h}_{\text{wall}}$ is the same as maximizing the SSNR in (2.30). We can write $\mathbf{h}_{\text{win}}$ and $\mathbf{h}_{\text{wall}}$ in (2.30) as the following matrix equations, respectively:

$$\begin{aligned} &\underbrace{\begin{bmatrix} h_{\text{eff}}(\Delta+1) \\ h_{\text{eff}}(\Delta+2) \\ \vdots \\ h_{\text{eff}}(\Delta+\nu+1) \end{bmatrix}}_{\mathbf{h}_{\text{win}}} \\ &= \underbrace{\begin{bmatrix} h(\Delta+1) & h(\Delta) & \cdots & h(\Delta-N_w+2) \\ h(\Delta+2) & h(\Delta+1) & \cdots & h(\Delta-N_w+3) \\ \vdots & \vdots & \ddots & \vdots \\ h(\Delta+\nu+1) & h(\Delta+\nu) & \cdots & h(\Delta+\nu-N_w+2) \end{bmatrix}}_{\mathbf{H}_{\text{win}}} \underbrace{\begin{bmatrix} w(0) \\ w(1) \\ \vdots \\ w(N_w-1) \end{bmatrix}}_{\mathbf{w}} \end{aligned} \tag{2.31}$$

$$\underbrace{\begin{bmatrix} h_{\text{eff}}(1) \\ \vdots \\ h_{\text{eff}}(\Delta) \\ h_{\text{eff}}(\Delta+\nu+2) \\ \vdots \\ h_{\text{eff}}(L_h+N_w-1) \end{bmatrix}}_{\mathbf{h}_{\text{wall}}}$$

$$= \underbrace{\begin{bmatrix} h(1) & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h(\Delta) & h(\Delta-1) & \cdots & h(\Delta-N_w+1) \\ h(\Delta+\nu+2) & h(\Delta+\nu+1) & \cdots & h(\Delta+\nu-N_w+3) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h(L_h-1) \end{bmatrix}}_{\mathbf{H}_{\text{wall}}} \underbrace{\begin{bmatrix} w(0) \\ w(1) \\ \vdots \\ w(N_w-1) \end{bmatrix}}_{\mathbf{w}} \tag{2.32}$$

The energy of $\mathbf{h}_{\text{win}}$ and $\mathbf{h}_{\text{wall}}$ in (2.30) can be written as

$$\begin{aligned} \mathbf{h}_{\text{wall}}^T\mathbf{h}_{\text{wall}} &= \mathbf{w}^T\mathbf{H}_{\text{wall}}^T\mathbf{H}_{\text{wall}}\mathbf{w} = \mathbf{w}^T\mathbf{A}\mathbf{w} \\ \mathbf{h}_{\text{win}}^T\mathbf{h}_{\text{win}} &= \mathbf{w}^T\mathbf{H}_{\text{win}}^T\mathbf{H}_{\text{win}}\mathbf{w} = \mathbf{w}^T\mathbf{B}\mathbf{w} \end{aligned} \tag{2.33}$$

where

$$\begin{aligned} \mathbf{A}_{N_w\times N_w} &= \mathbf{H}_{\text{wall}}^T\mathbf{H}_{\text{wall}} \\ \mathbf{B}_{N_w\times N_w} &= \mathbf{H}_{\text{win}}^T\mathbf{H}_{\text{win}} \end{aligned} \tag{2.34}$$

The two optimal shortening methods in [11] find $\mathbf{w}$ to minimize $\mathbf{w}^T\mathbf{A}\mathbf{w}$ while satisfying $\mathbf{w}^T\mathbf{B}\mathbf{w} = 1$. The method assumes that $\mathbf{B}$ is positive definite and factors $\mathbf{B} = \sqrt{\mathbf{B}}\sqrt{\mathbf{B}}^T$ by Cholesky decomposition [11]. Then, the method computes $\mathbf{l}_{\text{min}}$ is as the eigenvector associated with the smallest eigenvalue of the matrix $\left(\sqrt{\mathbf{B}}\right)^{-1}\mathbf{A}\left(\sqrt{\mathbf{B}^T}\right)^{-1}$. Finally, $\mathbf{w}_{\text{opt}} = \left(\sqrt{\mathbf{B}}^T\right)^{-1}\mathbf{l}_{\text{min}}$. Table 2.1 summarizes the method and its computational cost. The second more complicated

Step	*Description*	$\times$	$+$	$\div$
1	Fix a Δ, Compute $\mathbf{A}_{(N_w \times N_w)}$	$(L_h - \nu)N_w$	$(L_h - \nu)N_w$	0
2	Compute $\mathbf{B}_{(N_w \times N_w)}$	$\nu N_w +$ $(N_w + 1)N_w/2$	$\nu N_w +$ $(N_w + 1)N_w/2$	0
3	Take Cholesky decomposition of $\mathbf{B}$	N_w^3	N_w^3	0
4	Calculate $\left(\sqrt{\mathbf{B}}\right)^{-1}$ [86]	$(5N_w^3 + N_w)/3$	$(5N_w^3 + N_w)/3$	0
5	Calculate $\mathbf{C} = \left(\sqrt{\mathbf{B}}\right)^{-1} \mathbf{A} \left(\sqrt{\mathbf{B}^T}\right)^{-1}$	$2N_w^3$	$2N_w^2(N_w - 1)$	0
6	Use power method to find eigenvector $\mathbf{l}$ corresponding to the minimum eigenvalue of $\mathbf{C}$, λ			
6.1	Calculate $\mathbf{C}^{-1}$ [86]	$(5N_w^3 + N_w)/3$	$(5N_w^3 + N_w)/3$	0
6.2	Initialize $\mathbf{l}^{(0)}$			
6.3	$\mathbf{z}^{(k)} = \mathbf{C}^{-1}\mathbf{l}^{(k-1)}$	N_w^2	$(N_w - 1)N_w$	0
6.4	$\mathbf{l}_{\text{opt}}^{(k)} = \mathbf{z}^{(k)} / \parallel \mathbf{z}^{(k)} \parallel$	N_w	$N_w - 1$	N_w
6.5	$\lambda^{(k)} = \left[\mathbf{l}^{(k)}\right]^T \mathbf{C}^{-1}\mathbf{l}^{(k)}$	$(N_w + 1)N_w$	$N_w^2 - 1$	0
6.6	if $\lvert\lambda^{(k)} - \lambda^{(k-1)}\rvert >$ threshold, go to Step 6.3			
7	$\mathbf{w}_{\text{opt}} = \left(\sqrt{\mathbf{B}^T}\right)^{-1} \mathbf{l}_{\text{opt}}^{(k)}$	N_w^2	$(N_w - 1)N_w$	0

Table 2.1: Implementation and computational cost of the maximum SSNR method to find $\mathbf{w}_{\text{opt}}$ for a fixed Δ, where $\times$, $+$, and $\div$ are multiplication, addition, and division, respectively. The algorithm requires $\left(\frac{7}{6} + L_h\right) N_w + \frac{5}{2} N_w^2 + \frac{25}{3} N_w^3$ multiplications, $4N_w^2 + 2N_w$ fewer additions than multiplications, and N_w^2 divisions, assuming that N_w iterations for the inverse power method are needed for steps 6.3–6.6 to estimate the minimum eigenvalue of $\mathbf{C}$ (i.e., the maximum eigenvalue of $\mathbf{C}^{-1}$) and its associated eigenvectors.

method in [11] applies when $\mathbf{B}$ is singular. In order to avoid $\mathbf{B}$ from being singular, Yin and Yue [87] suggest an objective function to maximize $\mathbf{w}^T\mathbf{B}\mathbf{w}$ while satisfying the constraint $\mathbf{w}^T\mathbf{A}\mathbf{w} = 1$. In this case, they assume that $\mathbf{A}$ is positive definite since they perform a Cholesky decomposition on $\mathbf{A}$. All three methods [11, 87] require a Cholesky decomposition and an eigendecomposition of an $N_w \times N_w$ matrix to find $\mathbf{w}_{\text{opt}}$. Although we can replace the eigenvalue decomposition with a power method, Cholesky decomposition is computationally intensive (see Table 2.1) and generally requires floating-point arithmetic.

2.5.3 Maximizing channel capacity

The ultimate goal of an optimum TEQ design is to maximize channel capacity. MMSE and SSNR methods, however, do not necessarily maximize channel capacity. Al-Dhahir and Cioffi define the achievable channel capacity in one DMT symbol based on several assumptions [82, 88] as

$$b_{DMT} = N \log_2 \left(1 + \frac{\text{SNR}_{\text{geom}}}{\Gamma} \right) \text{ bits/symbol} \tag{2.35}$$

where Γ is the SNR gap between the actual SNR and the Shannon channel capacity due to the methods used for coding, modulation, etc., and SNR_{geom} is the *geometric SNR* [82, 88]. They propose to maximize the geometric SNR to maximize the channel capacity which requires the TIR $\mathbf{b}$.

In order to optimize (2.35), Al-Dhahir and Cioffi propose a complicated nonlinear optimization method [82, 88]. The method is sensitive to the initial guess for $\mathbf{b}$ and is not guaranteed to converge to the global optimum solution. Lashkarian and Kiaei [89] propose an iterative method that uses projection onto convex sets and is not sensitive to the initial guess for $\mathbf{b}$. Although their method has lower computational complexity than the method in [82, 88], their method is too complex for a practical real-time implementation. Arslan, Evans, and Kaiei [90] propose a method to maximize the channel capacity by developing a new model of the signal, noise, and ISI paths in the systems. Their solution generalizes the maximum SSNR method [11] by distributing the ISI into frequency bins with low SNRs.

2.6 Frequency-domain equalizer for DMT channels

For a DMT system, using a longer cyclic prefix can help receiver reduce more ISI, but at the price of reducing data throughput. To reduce the loss of throughput, the symbol length can be increased, which increases the FFT length. The length of the FFT, however, is limited by digital signal processing speed, cost, and receiver phase noise. To compensate for the frequency selectivity of the channel, a one-tap per frequency bin frequency-domain equalizer (FEQ) can be used to compensate for the channel amplitude and phase offsets [6]. One of the primary advantages of using a TEQ and an FEQ is that the TEQ essentially eliminates intercarrier interference in the frequency domain and the FEQ reduces to one complex tap per carrier. For 256 carriers, only 256 complex multiplications are required. The disadvantages of using the combination of TEQ and FEQ are [91]:

- The TEQ equalizes the entire channel and processes each subchannel with the same set of coefficients.
- The resulting channel capacity is not a smooth function of the synchronization delay. Therefore, an optimal synchronization delay cannot be selected easily, and instead, must be found by an exhaustive search over a large

range of delays.

- The obtained channel capacity may not be the highest achievable capacity of the DMT system.

A new direction of research [91] is called *per tone equalization*. The real-valued TEQ is replaced by a complex-valued FEQ for each carrier (tone) with the same number of taps. This new FEQ can overcome the drawbacks listed above. In this case, the FEQ becomes much more computationally complex because it has to compensate for a lot of intercarrier interference. For the case of 256 carriers, the general FEQ would consist of a 256×256 complex-valued matrix that must be multiplied by the length-256 vector at the output of the FFT.

2.7 Conclusion

In this chapter, I discuss channel estimation which aids the design of an equalizer to combat channel distortions. When channel estimation is not required, the parameters of an equalizer can be trained from the training sequence and the received signal. A neural network is a good candidate to learn the features from the training sequence and received signal in order to generalize its detection ability to other transmitted data. Two commonly used feedforward neural networks — multilayer perceptron network and radial basis function network — are discussed with respect to their architectures, training algorithms, and their application as channel equalizers. In theory, neural network equalizers can learn the features of received signal and classify the data to different clusters. However, neural network equalizers are still a research topic.

When channel estimation is necessary such as in DMT system, I introduce the matrix pencil method which can estimate the poles of channel impulse response modeled as an IIR filter. When a channel is known to the receiver at DMT system, I summarize current methods to design a TEQ for DMT channel. The equalized channel impulse response ideally is zeros for all of the samples except that $(\nu + 1)$ consecutive are non-zero values, which motivates the new proposed work in this dissertation.

Chapter 3

Neural Network Equalizers

A signal suffers from nonlinear, linear, and additive distortion when transmitted through a channel. Linear equalizers as shown in Figure 1.6 are commonly used in receivers to compensate for linear distortion in a channel. As an alternative, nonlinear equalizers have the potential to compensate for all three sources of channel distortion. Previous authors have shown that nonlinear feedforward equalizers based on either multilayer perceptron (MLP) or radial basis function (RBF) neural networks can outperform linear equalizers in terms of symbol error rate vs. SNR. In this chapter, I compare the performance of MLP vs. RBF equalizers in terms of symbol error rate vs. SNR. I design a reduced complexity neural network equalizer by cascading an MLP and a RBF network. In simulation, the new MLP-RBF equalizer outperforms MLP equalizers and RBF equalizers.

3.1 Introduction

Among feedforward equalizers, RBF [72, 92] and MLP [62, 66, 67, 68] equalizers outperform linear equalizers in symbol error rate vs. SNR, as mentioned in Section 2.3. RBF equalizers estimate the probability density function of the incoming signal to approximate the optimal Bayesian equalizer [64]. MLP equalizers can approximate a Bayesian discriminant function. The size and structure of the MLP limit the approximation accuracy of the networks [93]. MLP and RBF equalizers appear to be insensitive to the channel phase response, as demonstrated in Section 3.3.

MLP training algorithms are either fast, but get trapped in local minima (such as the Levenberg-Marquardt algorithm), or slow, but converge to a global minimum (such as simulated annealing). Here, "slow" can be several orders of magnitude slower than "fast". When using a "fast" algorithm, an MLP equalizer is trained several times and the best network is chosen.

In RBF equalizers, the number of hidden neurons increases exponentially with the length of the FIR model of the channel to provide the same SER [12]. To obtain a similar symbol error rate vs. SNR performance as MLP equalizers, RBF equalizers must use an increasingly larger number of hidden nodes than the MLP equalizer as SNR decreases. In order to reduce the number of neurons in an RBF equalizer, a modified k-mean algorithm [92] or a self-organizing map [94] may be used in the first step of RBF training to compute σ_i's and $\mathbf{c}_i$'s in (2.18).

When the SNR is low, the transmitted data has been scattered by the addition of strong noise. In an RBF equalizer, the data far away from each cluster is considered *irrelevant*. MLP reduces irrelevant data because the hidden layer performs a moving average by first calculating the weighted sum of the inputs. The linear combinations of the inputs confine the network's attention to the linear subspace spanned by the weight vectors. When the data is scattered due to low SNR, the number of hidden units in MLP is not required to increase. In RBF networks, the irrelevant data significantly degrades performance because activation functions have local receptive fields that can be adapted to a local pattern in the data.

3.2 Hybrid MLP-RBF equalizer

To improve performance and reduce complexity of MLP and RBF equalizers, I cascade an MLP network and RBF network to form an MLP-RBF equalizer. In the MLP-RBF equalizer, I set the number of MLP input neurons, MLP output neurons, and RBF input neurons equal to the number of taps L_h in the FIR channel model. The inputs to the MLP-RBF equalizer are the current received sample and the previous $L_h - 1$ samples. The MLP stage suppresses the irrelevant data (noise) and outputs cleaned values of the current received sample and previous $L_h - 1$ samples. The RBF stage takes these MLP outputs, performs a best fit, and outputs the symbol decision on its single output. Using the training sequence, I first train the MLP network using the Levenberg-Marquardt algorithm, then feed the trained MLP output into the RBF network, and finally train the RBF network using expectation maximization (EM) to compute the σ_i's and $\mathbf{c}_i$'s and least-mean squares (LMS) to train the weights (see Sections 2.3.2 and 2.3.3). The MLP-RBF equalizer requires far fewer neurons for the same symbol error rate vs. SNR performance than either an MLP equalizer or an RBF equalizer, as demonstrated next.

Because the probability that MLP networks fall into local minima during training increases with the number of hidden layers [95], I use only one hidden layer. For activation functions in the MLP equalizer, the hidden layer uses the hyperbolic tangent function and the output layers use an identity function (linear function with $k = 1$). The RBF equalizer uses a Gaussian radial basis function. The neural network inputs are delayed versions of the received signal.

3.3 Simulation results for 2-PAM

I compare symbol error rate vs. SNR of the MLP and RBF equalizers for the following settings: 6 training algorithms for MLP, 2 channel responses, 3–15 input nodes for MLP and 3–8 inputs for RBF, and different numbers of hidden units. The channel responses

$$\begin{aligned} H_{min}(z) &= 0.6963 + 0.6964z^{-1} + 0.1741z^{-2} \\ H_{linear}(z) &= 0.3482 + 0.8704z^{-1} + 0.3482z^{-2} \end{aligned}$$

are minimum and linear phase, respectively, and have the same magnitude response. The transmitted 2-PAM signals are chosen from $\{-1, 1\}$ with equal probability. They are independent and identically distributed. The additive distortion in the channel is modeled as white Gaussian noise. I vary the SNR from 0 dB to 30 dB SNR in increments of 3 dB. The training sequence is 1000 symbols (one symbol/bit). From (2.19), the number of neurons N_{in} in the input layer could be up to 8 because $L_h = 3$. After training, I test the equalizer using 2×10^6 symbols.

A minimum of 3 inputs is used for the MLP and RBF equalizers because each channel has 3 taps. The maximum number of inputs is chosen so that training would complete in a reasonable amount of time and diminishing performance returns are observed as the number of inputs increases. I use 1, 2, 4, 8, and 16 hidden units for MLP equalizers, and 10, 20, and 40 hidden units for RBF equalizers.

Table 3.1 compares six training algorithms [12] for the MLP equalizer: conjugate gradient, scaled conjugate gradient, quasi-Newton, Levenberg-Marquardt (LM), hybrid linear-nonlinear, and batch backpropagation. The LM algorithm gives the best symbol error rate vs. SNR performance yet requires the lowest computational complexity. Table 3.1 shows the results of using the EM/LMS method [12] for the RBF equalizer. Throughout the rest of this chapter, I use MLP-LM and RBF-EM/LMS equalizers.

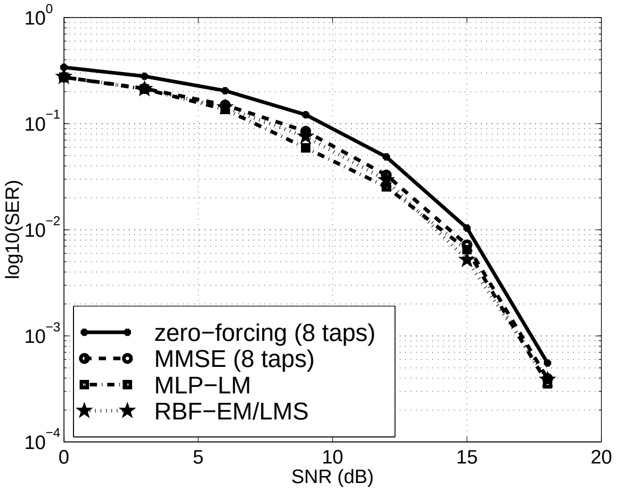
Figure 3.1 compares the simulation performance of the nonlinear MLP and RBF equalizers and the linear zero-forcing and minimum mean squared error (MMSE) equalizers. In the simulation, the two linear equalizers know the channel impulse response, whereas the nonlinear equalizers do not. The MLP and RBF equalizers obtain almost the same performance as the MMSE equalizer even though MLP and RBF equalizers have no knowledge of the channel impulse response.

Figure 3.2 compares the simulation performance of MLP, RBF, and MLP-RBF equalizers. For the MLP-RBF equalizer, I set the number of inputs to be the length of the FIR channel model. I use an MLP with 3 inputs, 4 hidden units, and 3 outputs, and an RBF network with 3 inputs, 4 hidden units, and 1 output. The MLP-RBF equalizer outperforms both MLP and RBF equalizers. Since the RBF equalizer in Figure 3.2 has the same structure as that in Figure 3.1(b), this MLP-RBF equalizer also outperforms linear feedforward equalizers. In Matlab 5, the training time was 3.59 s for the MLP (3-4-1), 170.20 s for the RBF (3-40-1) equalizer, and 14.95 s for the MLP (3-4-3)–RBF (3-4-1) equalizer. The simulations were run on a 167-MHz Ultra-2 workstation.

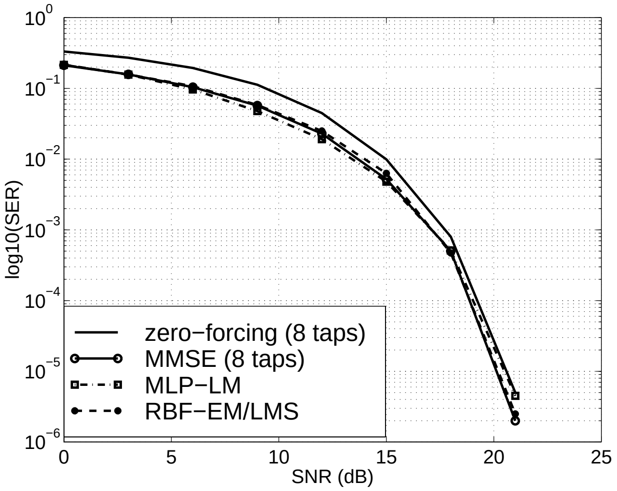
I compare the performance of two neural network equalizers (MLP and RBF) and two “best-case” linear equalizers (zero-forcing and MMSE). The linear equalizers have precise knowledge of the channel coefficients, which are unknown to the neural network equalizers. The MMSE equalizer is the optimal linear equalizer in the least squares sense. The order of symbol error rate vs. SNR per-

Training Method	Training CPU Time	Training GFLOPS	Testing SER
Batch back-propagation	2608.0 s	31.14	0.0084
Levenberg-Marquardt (LM)	65.3 s	2.50	0.0083
Conjugate Gradient	2213.0 s	39.20	0.0114
Scaled Conjugate Grad.	1428.0 s	8.38	0.0120
Quasi-Newton	9770.0 s	27.20	0.0121
Hybrid linear/ nonlinear	More than a day	n/a	0.0085
RBF-EM/LMS (5-20-1)	21.1 s	0.16	0.0074
MLP (3-4-3): RBF (3-4-1)	24.4 s	0.47	0.0033

Table 3.1: Training time and symbol error rate (SER) for MLP, RBF, and MLP-RBF equalizers for the minimum phase channel $H_{min}(z) = 0.6963 + 0.6964z^{-1} + 0.1741z^{-2}$. The first six methods are for an MLP (8-6-1) equalizer. Here, EM stands for expectation-maximization and LMS stands for least mean squares (LMS). 2-PAM modulation was used. Training consisted of 1000 symbols (one symbol/bit) at an SNR = 15 dB.



(a) Minimum phase channel
MLP-LM has 6 inputs, 8 hidden units, 1 output.
MLP-LM has 6 inputs, 16 hidden units, 1 output.



(b) Linear phase channel
RBF-EM/LMS has 6 inputs, 20 hidden units, 1 output.
RBF-EM/LMS has 3 inputs, 40 hidden units, 1 output.

Figure 3.1: Performance analysis of four equalizers for 2-PAM modulation. Zero-forcing and minimum mean square error (MMSE) equalizers know the channel coefficients, whereas the neural network equalizers do not. The MMSE equalizer is the optimal linear equalizer.

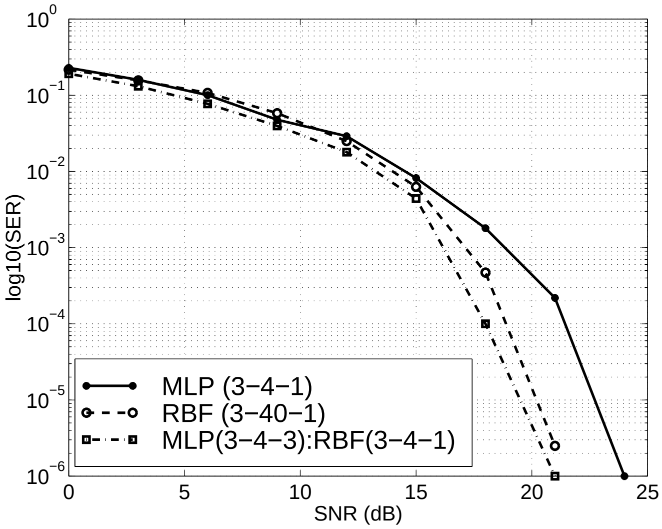


Figure 3.2: Comparison of MLP, RBF, and MLP-RBF equalizers for the linear phase channel $H_{linear}(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$. The MLP-RBF equalizer outperforms MLP and RBF equalizers yet has lower complexity.

formance from best to worst is MLP, RBF, MMSE, and zero-forcing, according to Figure 3.1. For some SNR values, the RBF equalizer outperforms the MLP equalizer in SER in Figure 3.2. The MLP network has to be trained several times, whereas the other equalizers are trained in one pass over the training data.

3.4 Simulation results for 16-QAM

M-ary QAM is a very effective technique to achieve a high bit-rate transmission without increasing the bandwidth [96]. In this simulation experiment, I compare symbol error rate vs. SNR of MLP, RBF, and MLP-RBF equalizers for a telephone channel from Ptolemy (http://ptolemy.eecs.berkeley.edu/). The linear distortion comes from a linear phase bandpass FIR filter with the transfer function

$$\begin{aligned} H(z) = \; & -0.04 - 0.0016z^{-1} + 0.1785z^{-2} + 0.3767z^{-3} + \\ & 0.3767z^{-4} + 0.1785z^{-5} - 0.0016z^{-6} - 0.04z^{-7}. \end{aligned}$$

The output of the FIR filter, $y_{in}(n)$ is then nonlinearly distorted as follows

$$y_{out}(n) = y_{in}(n) + 0.1y_{in}^2(n) + 0.05y_{in}^3(n)$$

where $y_{out}(n)$ is the output from the nonlinear distortion. The real and imaginary parts of transmitted 16-QAM signals are selected from $\{-3, -1, 1, 3\}$ with equal probability. They are also independent and identically distributed. The additive noise in the channel is complex white Gaussian noise and varies from 0 to 30 dB SNR with increments of 3 dB.

Since a complex MLP network cannot have complex-valued differentiable activation functions and a complex RBF network uses a real-valued activation function (see Sections 2.3.2, 2.3.3, and 2.3.4), I train the real and imaginary parts of received signals separately so that the real-valued MLP and RBF network can be used. As Mulgrew suggests [64], the number of symbols in (2.19) to train the MLP and RBF networks should be $4^8 = 65536$ symbols if the length of channel impulse response is considered only, where "4" is the number of constellations of the real and imaginary parts of signal and "8" is the number of taps in the simulation channel. Table 3.1 shows that this is impractical for real-time implementation. In this simulation, 1000 symbols were used for training and 2×10^6 symbols were used for testing. Based on the generalization property of neural networks, I train MLP and RBF networks with the real part of the received signals and assume that the neural networks can generalize the features to the imaginary part. The network structures are (5-10-1) for MLP equalizers,

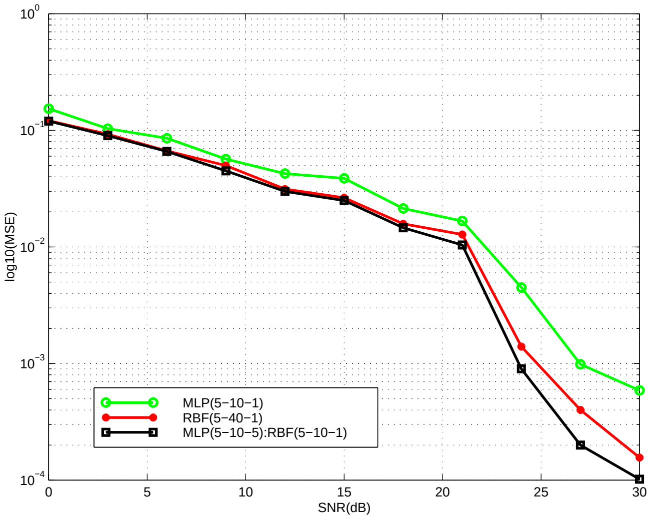


Figure 3.3: Comparison of MLP, RBF, and MLP-RBF equalizers for a simulated telephone channel with linear and nonlinear distortions

(5-40-1) for RBF equalizers, and (5-10-5)-(5-10-1) for MLP-RBF equalizers. The training algorithms are LM for the MLP networks and EM/LMS for the RBF networks.

Figure 3.3 shows the simulation performance of MLP, RBF, and MLP-RBF equalizers. The MLP-RBF equalizer outperforms both MLP and RBF equalizers with respect to SER. It uses fewer weights than the RBF equalizer and obtains lower SER than the MLP equalizer.

3.5 Conclusion

Since the problem of channel equalization can be considered as a classification problem, artificial neural networks can play an important role in telecommunication systems for designing a channel equalizer. The nonlinearity and generalization property are promising features of neural network equalizers. I have designed a new reduced complexity neural network equalizer by cascading an MLP and an RBF network. The simulation results from 2-PAM and 16-QAM show that an MLP-RBF equalizer can outperform MLP equalizers and RBF equalizers. In order to reduce the training time in the case of a complex signal, I use only the real part of the received signal to train the neural network. In the MLP-RBF equalizer, the MLP network suppresses noise and the RBF network performs the equalization. The new MLP-RBF equalizer outperforms the MLP and RBF equalizers in terms of symbol error rate vs. SNR.

The primary difficulty of neural network equalizers is the amount of training data required. Mulgrew [64] shows that the number of symbols used in the training is exponential in the length of the channel impulse response and the number of neurons in the input layer of a neural network. The number of neurons can be fixed while the number of taps in a channel still remains unknown. Therefore, it is desired to estimate a channel impulse response. Since the number of training symbols grows exponentially with the length of the channel impulse response, it is necessary to develop methods to shorten a channel to a desired length. These two topics will be covered in the next two chapters.

Chapter 4

A New Matrix Pencil Method for Channel Estimation

The matrix pencil method estimates the damping factors and frequencies of superimposed exponentially damped/undamped sinusoids, as discussed in Section 2.2.4. When no noise is present, the matrices of the received data $\mathbf{S}$, $\mathbf{S}_0$, and $\mathbf{S}_1$ are rank-deficient and have Hankel structure. When noise is present, the matrices of the received data, $\mathbf{R}$, $\mathbf{R}_0$, and $\mathbf{R}_1$, have full rank and Hankel structure. Hua and Sarkar propose to use singular value decomposition (SVD) to reduce the rank in order to mitigate noise effects [13]. The disadvantage of SVD is that it destroys the Hankel structure of the data matrices. The contribution of this chapter is to apply a reduced-rank Hankel approximation to the matrices $\mathbf{R}$, $\mathbf{R}_0$, and $\mathbf{R}_1$ in different steps of the original matrix pencil method. This results in three modified matrix pencil methods to retain the rank-deficient and Hankel properties.

4.1 Introduction

For a data sequence consisting of superimposed damped/undamped sinusoids, the matrix in the corresponding matrix prediction equation has the properties of rank deficiency and Hankel structure. The *Kumaresan-Tufts* (KT) method

[29] exploits the structure of backward linear prediction equations satisfied by the underlying signal to estimate the signal's parameters, and applies an SVD-based reduced-rank approximation to mitigate noise effects. However, such an approximation takes into account only the rank-deficient property of the signal matrix and does not preserve its Hankel structure. The resulting performance is unsatisfactory at low SNRs.

The *matrix pencil* (MP) method by Hua and Sarkar [13] exploits the structure of a matrix pencil of the underlying signal and makes use of reduced-rank approximation based on singular value decomposition (SVD) to suppress the noise. It ignores the Hankel structure of the matrix pencil and suffers from a degradation of estimation accuracy at low SNR. Ekstrom modifies the matrix pencil method by using "forward/backward averaging" [97]. Il'kiv [98] multiplies the pencil of matrices by a weight matrix. Although both Ekstrom's modified matrix pencil and Il'kiv's weighted matrix pencil methods yield slightly better parameter estimation than the original matrix pencil method, they both suffer from the same problem as Hua and Sarkar's matrix pencil method.

Cadzow proposes a signal enhancement algorithm for estimating the signal parameters of undamped sinusoids [14, 99] that exploits both the rank-deficient and Toeplitz-Hankel properties of the signal matrix. Such an algorithm is extended to an iterative *reduced-rank Hankel approximation* (RRHA) in [14, 15] to approximately retain rank deficiency and Hankel structure of the data matrix. We apply the RRHA to Hua and Sarkar's method and develop three modified matrix pencil methods [100]. One of the improved methods in [100] is similar to the *new matrix pencil* method proposed in [31, 32].

In this chapter, Section 4.1 summarizes the various matrix pencil methods derived from Hua and Sarkar's method [13]. Section 4.2 discusses the reduced-rank Hankel approximation. Section 4.3 proposes three modified matrix pencil methods and compares their improvement of matrix pencil properties. Applications of the matrix pencil method to channel estimation are also described. Simulation results in Section 4.4 show the estimation performance of the proposed methods. Section 4.5 concludes this chapter.

4.2 Reduced-rank Hankel approximation

For a given $P \times Q$ matrix $\boldsymbol{X}$, the Hankel approximation operator $\mathcal{H}$ is defined as

$$\boldsymbol{Y} = \mathcal{H}\{\boldsymbol{X}\}$$

in which the (p, q)-th element of $\boldsymbol{Y}$ is given by

$$y_{p,q} = \frac{1}{|\Lambda_{p+q}|} \sum_{(p',q') \in \Lambda_{p+q}} x_{p',q'} \tag{4.1}$$

for $p = 0, 1, \ldots, P-1$, and $q = 0, 1, \ldots, Q-1$, where Λ_t denotes the set of indices corresponding to the t^{th} anti-diagonal of a matrix; i.e.

$$\Lambda_t = \{(p', q') : 0 \leq p' \leq P-1,\ 0 \leq q' \leq Q-1,\ p' + q' = t\},$$

and $|\Lambda_t|$ denotes the cardinality of the set Λ_t. So, the Hankel approximation operator replaces each cross-diagonal with the average of the elements on the cross-diagonal.

In general, the operator $\mathcal{L}$ in (2.13) does *not* preserve the Hankel property, and the operator $\mathcal{H}$ does *not* preserve the rank-deficient property. Since there is a one-to-one correspondence between a data sequence consisting of superimposed damped sinusoids and a rank-deficient Hankel matrix [15], it is desirable to maintain both the rank-deficient and Hankel properties of the data matrix. Thus, I define $\mathcal{J}$ to be the *reduced-rank Hankel approximation* (RRHA) operator that generates a matrix that has both the rank-deficient and Hankel properties. Since no analytic form for the operator $\mathcal{J}$ exists to date, an iterative algorithm is used to approximate $\mathcal{J}$ [15]:

$$\begin{aligned}\mathcal{J}\{\boldsymbol{X}\} &= (\mathcal{HL})^{\infty}\{\boldsymbol{X}\} \\ &= \lim_{n\to\infty}(\mathcal{HL})^{n}\{\boldsymbol{X}\} \\ &= \lim_{n\to\infty}\underbrace{(\mathcal{HL}\cdots(\mathcal{HL}}_{n}(\mathcal{HL}\{\boldsymbol{X}\}))\cdots). \qquad (4.2)\end{aligned}$$

Cadzow shows that such an iteration converges for the data matrix $\boldsymbol{S}$ associated with the superimposed damped sinusoids buried in noise [99]. Theoretically, the matrix $\mathcal{J}\{\boldsymbol{R}\}$ possesses both the Hankel and rank-deficient properties and results in a better approximation of $\boldsymbol{S}$ than the reduced-rank but non-Hankel matrix $\mathcal{L}\{\boldsymbol{R}\}$.

In reality only a finite number of iterations can be implemented (thus, a stopping criterion is needed in practice) so that the resulting matrix is either roughly rank-deficient or roughly Hankel. Usually, we choose to force the matrix to be exactly rank-deficient and approximately Hankel in order to make the subsequent steps (e.g. finding non-zero eigenvalues) easier.

4.3 Modified matrix pencil methods

The MP method involves three matrices, $\boldsymbol{S}$, $\boldsymbol{S}_0$, and $\boldsymbol{S}_1$, as defined in the same way as in (2.10) and (2.11). It would be desirable to make use of rank-deficient and Hankel properties of all three matrices. Next, I develop three modified MP methods classified by the properties they exploit.

4.3.1 Method #1

A straightforward modification of the original matrix pencil method is to replace the reduced-rank approximation of $\boldsymbol{R}_0$ and $\boldsymbol{R}_1$ by the RRHA of the two matrices so that the Hankel structure of $\boldsymbol{S}_0$ and $\boldsymbol{S}_1$ is exploited, too. Figure 4.1 gives the steps to implement this method. According to Theorem 1 in [15], we infer that during each iteration of the RRHA, the preservation of the Hankel structure leads to a better approximation of the true signal matrices $\boldsymbol{S}_0$ and $\boldsymbol{S}_1$; i.e., in the lth iteration,

$$\|(\mathcal{H}\mathcal{L})^l\{\boldsymbol{R}_i\} - \boldsymbol{S}_i\|_F \leq \|\mathcal{L}(\mathcal{H}\mathcal{L})^{l-1}\{\boldsymbol{R}_i\} - \boldsymbol{S}_i\|_F \tag{4.3}$$

for $i = 0, 1$, where $\|\cdot\|_F$ denotes the Frobenius norm

$$\|\boldsymbol{X}\|_F = \left(\sum_{p=0}^{P-1}\sum_{q=0}^{Q-1} |x_{p,q}|^2\right)^{1/2}$$

for a $P \times Q$ matrix $\boldsymbol{X}$. Like the original matrix pencil method, a disadvantage of the MMP1 method is that, since the two matrices $\boldsymbol{R}_0$ and $\boldsymbol{R}_1$ are processed independently, the properties of $\boldsymbol{S}$ are ignored. Therefore, in general, $\widetilde{\boldsymbol{S}}_0$ and $\widetilde{\boldsymbol{S}}_1$ in the MMP1 method do not correspond to any valid reduced-rank and Hankel

Step 1	Form the $(K-L) \times (L+1)$ master matrix $\mathbf{R}$ as in (2.10)
Step 2	Form the $(K-L) \times L$ matrices $\mathbf{R}_0$ and $\mathbf{R}_1$ as in (2.11)
Step 3	Compute $\widetilde{\boldsymbol{S}}_0 = \mathcal{J}\{\boldsymbol{R}_0\}$ from (4.2)
Step 4	Compute $\widetilde{\boldsymbol{S}}_1 = \mathcal{J}\{\boldsymbol{R}_1\}$from (4.2)
Step 5	Compute the M non-zero eigenvalues of $(\widetilde{\boldsymbol{S}}_0)^\dagger \widetilde{\boldsymbol{S}}_1$ as $\{\exp(d_m - j2\pi f_m)\}_{m=1}^{M}$

Figure 4.1: Modified Matrix Pencil Method 1 for estimating M poles, using K samples for matrix pencil parameter L, where $\frac{1}{3}K \leq L \leq \frac{2}{3}K$ and $M \leq K - L$.

approximation of $\boldsymbol{S}$. This drawback can yield significantly biased estimates when the two signals corresponding to $\widetilde{\boldsymbol{S}}_0$ and $\widetilde{\boldsymbol{S}}_1$ are very different from each other.

4.3.2 Method #2

To avoid the highly biased estimates caused by the independent processing of $\boldsymbol{R}_0$ and $\boldsymbol{R}_1$ in method #1, I consider the rank-deficient property and/or the Hankel property possessed by $\boldsymbol{S}$ as well; i.e., I process the two matrices $\boldsymbol{R}_0$ and $\boldsymbol{R}_1$ *jointly* [18]. The key idea is to modify the RRHA operator by performing the Hankel operator on the two rank-reduced matrices *simultaneously* rather than individually so that the corresponding master matrix will also possess the Hankel structure. I extend the operator $\mathcal{H}$ to a pair of matrices $\{\boldsymbol{Y}_0, \boldsymbol{Y}_1\} = \mathcal{H}\{\boldsymbol{X}_0, \boldsymbol{X}_1\}$. The elements of $\boldsymbol{Y}_0$ and $\boldsymbol{Y}_1$ are

$$y^0_{p,q} = \frac{\sum\limits_{(p',q')\in\Lambda_{p+q}} x^0_{p',q'} + \sum\limits_{(p',q')\in\Lambda_{p+q-1}} x^1_{p',q'}}{|\Lambda_{p+q}| + |\Lambda_{p+q-1}|}$$

$$y^1_{p,q} = \frac{\sum\limits_{(p',q')\in\Lambda_{p+q+1}} x^0_{p',q'} + \sum\limits_{(p',q')\in\Lambda_{p+q}} x^1_{p',q'}}{|\Lambda_{p+q+1}| + |\Lambda_{p+q}|} \tag{4.4}$$

for $p = 0, 1, \ldots, P-1$, and $q = 0, 1, \ldots, Q-1$. Figure 4.2 summarizes the resulting "MMP2" method. The proof of the convergence of such an iterative algorithm remains open. However, it always approximately converges within a few iterations in simulation over a wide range of SNRs. That is, each iteration better approximates $\boldsymbol{S}_0$ and $\boldsymbol{S}_1$, as stated in the following theorem.

Theorem 1 *In the lth iteration,*

$$\sum_{i=0}^{1} \|(\mathcal{H}\mathcal{L})^l\{\boldsymbol{R}_i\} - \boldsymbol{S}_i\|_F^2 \le \sum_{i=0}^{1} \|\mathcal{L}(\mathcal{H}\mathcal{L})^{l-1}\{\boldsymbol{R}_i\} - \boldsymbol{S}_i\|_F^2 \tag{4.5}$$

Step 1	Form the $(K-L)\times(L+1)$ master matrix $\mathbf{R}$ as in (2.10)
Step 2	Form the $(K-L)\times L$ matrices $\mathbf{R}_0$ and $\mathbf{R}_1$ as in (2.11)
Step 3	Let $\widetilde{\boldsymbol{S}}_0 = \boldsymbol{R}_0$ and $\widetilde{\boldsymbol{S}}_1 = \boldsymbol{R}_1$
Step 4	Compute $\{\widehat{\boldsymbol{S}}_0, \widehat{\boldsymbol{S}}_1\} = \mathcal{H}\{\widetilde{\boldsymbol{S}}_0, \widetilde{\boldsymbol{S}}_1\}$ from (4.4)
Step 5	Compute $\widetilde{\boldsymbol{S}}_0^{\dagger} = \mathcal{L}\{\widehat{\boldsymbol{S}}_0\}$ from (2.14)
Step 6	Compute $\widetilde{\boldsymbol{S}}_1 = \mathcal{L}\{\widehat{\boldsymbol{S}}_1\}$ from (2.13)
Step 7	Go to Step 2 unless the stop criterion is satisfied
Step 8	Compute the M non-zero eigenvalues of $(\widetilde{\boldsymbol{S}}_0)^{\dagger}\widetilde{\boldsymbol{S}}_1$ as $\{\exp(d_m - j2\pi f_m)\}_{m=1}^{M}$

Figure 4.2: Modified Matrix Pencil Method 2 for estimating M poles, using K samples for matrix pencil parameter L, where $\frac{1}{3}K \leq L \leq \frac{2}{3}K$ and $M \leq K-L$.

Proof: Let $\tilde{\boldsymbol{S}}_i = \mathcal{L}(\mathcal{H}\mathcal{L})^{l-1}\{\boldsymbol{R}_i\}$ and $\widehat{\boldsymbol{S}}_i = (\mathcal{H}\mathcal{L})^{l}\{\boldsymbol{R}_i\}$, for $i = 0, 1$. We consider the approximation of a pair of diagonals in $\boldsymbol{S}_0$ and $\boldsymbol{S}_1$ given by $\{s^0_{p,q} : (p,q) \in \Lambda_z\}$ and $\{s^1_{p,q} : (p,q) \in \Lambda_{z-1}\}$, respectively,

$$\begin{aligned}
& \sum_{(p,q)\in\Lambda_z} |s_z^* - \widetilde{s}^0_{p,q}|^2 + \sum_{(p,q)\in\Lambda_{z-1}} |s_z^* - \widetilde{s}^1_{p,q}|^2 \\
= & \sum_{(p,q)\in\Lambda_z} \left(|s_z^* - \widehat{s}^0_{p,q}|^2 + |\widehat{s}^0_{p,q} - \widetilde{s}^0_{p,q}|^2 - 2\Re\{(s_z^* - \widehat{s}^0_{p,q})(\widehat{s}^0_{p,q} - \widetilde{s}^0_{p,q})^*\}\right) \\
& + \sum_{(p,q)\in\Lambda_{z-1}} \left(|s_z^* - \widehat{s}^1_{p,q}|^2 + |\widehat{s}^1_{p,q} - \widetilde{s}^1_{p,q}|^2 - 2\Re\{(s_z^* - \widehat{s}^1_{p,q})(\widehat{s}^1_{p,q} - \widetilde{s}^1_{p,q})^*\}\right) \\
= & \sum_{(p,q)\in\Lambda_z} \left(|s_z^* - \widehat{s}^0_{p,q}|^2 + |\widehat{s}^0_{p,q} - \widetilde{s}^0_{p,q}|^2\right) + \sum_{(p,q)\in\Lambda_{z-1}} \left(|s_z^* - \widehat{s}^1_{p,q}|^2 + |\widehat{s}^1_{p,q} - \widetilde{s}^1_{p,q}|^2\right) \\
\geq & \sum_{(p,q)\in\Lambda_z} |s_z^* - \widehat{s}^0_{p,q}|^2 + \sum_{(p,q)\in\Lambda_{z-1}} |s_z^* - \widehat{s}^1_{p,q}|^2.
\end{aligned}$$

Considering all of the corresponding pairs of diagonals in $\boldsymbol{S}_0$ and $\boldsymbol{S}_1$, we have proven (4.5). Q.E.D.

Even though the improvement in the matrix approximation described in (4.5) seems weaker than the one given in (4.3), one would expect that the MMP2 method would yield more accurate estimates of signal parameters than the MMP1 method, especially at low SNR. The better performance results because the MMP2 method makes use of the relationship between $\boldsymbol{S}_0$ and $\boldsymbol{S}_1$.

4.3.3 Method #3

Another way to exploit the properties of the three matrices $\boldsymbol{S}$, $\boldsymbol{S}_0$, and $\boldsymbol{S}_1$ is to perform the RRHA on the master data matrix $\boldsymbol{R}$ *directly*. Since in theory there exists a unique exponential data sequence corresponding to a rank-deficient Hankel matrix $\widehat{\boldsymbol{S}}$, the matrices $\widehat{\boldsymbol{S}}_0$ and $\widehat{\boldsymbol{S}}_1$ are both rank-deficient and Hankel. Therefore, in theory, the two properties are satisfied by all three matrices. I call this method the "MMP3" method, which may be viewed as applying a *preprocessing* procedure to the original MP method. It can be implemented according to Figure 4.3.

The MMP3 method achieves a better approximation of the true master matrix $\boldsymbol{S}$; i.e., in the lth iteration,

$$\|(\mathcal{HL})^l\{\boldsymbol{R}\} - \boldsymbol{S}\|_F \leq \|\mathcal{L}(\mathcal{HL})^{l-1}\{\boldsymbol{R}\} - \boldsymbol{S}\|_F. \tag{4.6}$$

Since in practice $\widehat{\boldsymbol{S}}$ is only approximately Hankel, $\widehat{\boldsymbol{S}}_0$ and $\widehat{\boldsymbol{S}}_1$ are neither exactly rank-deficient nor exactly Hankel. However, because of the reduced-rank operator embedded in the subsequent MP method, the rank-deficiency of $\widehat{\boldsymbol{S}}_0$ and $\widehat{\boldsymbol{S}}_1$ can be eventually achieved. Razavilar, Li, and Liu [31] propose such a method. However, their method requires the data matrices to be square in order to obtain the best performance. This requirement is not necessary since Hua and Sarkar point out that any values of K and L satisfying $K/3 \leq L \leq 2K/3$ are good choices [13].

Step 1	Form the $(K-L)\times(L+1)$ master matrix $\mathbf{R}$ as in (2.10)
Step 2	Compute $\widehat{\boldsymbol{S}} = \mathcal{J}\{\boldsymbol{R}\}$ from (4.4)
Step 3	Form $\widehat{\boldsymbol{S}}_0$ and $\widehat{\boldsymbol{S}}_1$ from $\widehat{\boldsymbol{S}}$ from (2.11)
Step 4	Compute $\widetilde{\boldsymbol{S}}_0^{\dagger} = \mathcal{L}^{\dagger}\{\widehat{\boldsymbol{S}}^0\}$ from (2.14)
Step 5	Compute $\widetilde{\boldsymbol{S}}_1 = \mathcal{L}\{\widehat{\boldsymbol{S}}_1\}$ from (2.13)
Step 6	Compute the M non-zero eigenvalues of $(\widetilde{\boldsymbol{S}}^0)^{\dagger}\widetilde{\boldsymbol{S}}^1$ as $\{\exp(d_m - j2\pi f_m)\}_{m=1}^{M}$

Figure 4.3: Modified Matrix Pencil Method 3 for estimating M poles, using K samples for matrix pencil parameter L, where $\frac{1}{3}K \le L \le \frac{2}{3}K$ and $M \le K - L$.

4.3.4 Comparison

Table 4.1 provides a summary of the properties of the matrices $\boldsymbol{S}$, $\boldsymbol{S}_0$, and $\boldsymbol{S}_1$, which are exploited by the original and modified MP methods. The symbol "YES" (or "NO") indicates whether a certain property is exploited (or not exploited) by a particular method. The superscript "$^{-}$" indicates that a property is exactly satisfied in theory and approximately satisfied in practice.

Among the four methods in Table 4.1, only the MMP3 method can exploit both the rank-deficient and Hankel properties of all three matrices. All three modified MP methods make use of more properties of the three matrices than the original MP method. Consequently, the modified MP methods should achieve better estimation performance than the original one. Since both the MMP2 and MMP3 methods take into account the relationship between $\widehat{\boldsymbol{S}}_0$ and $\widehat{\boldsymbol{S}}_1$, they should give more accurate estimates than either the original MP method or the MMP1 method.

Methods		$\widetilde{\boldsymbol{S}}$	$\widetilde{\boldsymbol{S}_0}$	$\widetilde{\boldsymbol{S}_1}$
MP	Rank-M	NO	YES	YES
	Hankel	NO	NO	NO
MMP1	Rank-M	NO	YES	YES
	Hankel	NO	YES$^-$	YES$^-$
MMP2	Rank-M	NO	YES	YES
	Hankel	YES$^-$	YES$^-$	YES$^-$
MMP3	Rank-M	YES	YES$^-$	YES$^-$
	Hankel	YES$^-$	YES$^-$	YES$^-$

Table 4.1: Comparison of the four methods in terms of the preservation of the rank-deficient and Hankel properties of the three data matrices involved in those methods (the notation YES$^-$ indicates that a property is exactly satisfied in theory and approximately satisfied in practice).

4.3.5 Matrix pencil for channel estimation

The received signal in (2.7) can be rewritten as

$$r_k = h_k + n_k,$$

where h_k is the impulse response of the rational transfer function given in (2.2), which has poles at e^{p_m}'s where p_m is given in (2.12). A communication channel can be modeled as an IIR filter as given in (2.2). All of the matrix pencil related methods can be used to estimate the channel poles. For example, an estimation method could be applied to the received signal assuming that the power spectrum of the transmitted signal is flat. Once the poles of a transfer function have been determined, the zeros may be obtained by many methods such as Shanks' method [101].

4.4 Simulation results

I first compare how well the proposed modified matrix pencil methods estimate the parameters of frequencies and damping factors. I use the same example as in [15] to compare the estimation performance of the modified Kumaresan-Tufts (MKT) method [15], the MP method, and the three modified MP methods. I choose $M = 2$ poles, $K = 25$ samples, $L = 17$, $a_1 = a_2 = 1$, $d_1 = 0.2$, $d_2 = 0.1$, $f_1 = 0.42$ rad/sample, and $f_2 = 0.52$ rad/sample. Up to $K - L = 8$ pairs of frequencies and damping factors could be estimated, but I only estimate $M = 2$ pairs. The SNR in dB is defined as

$$\text{SNR} = -10 \log_{10}(2\sigma_n^2), \tag{4.7}$$

where σ_n^2 is the variance of the zero-mean complex white Gaussian noise process n_k. I use mean squared error (MSE) to assess the estimation accuracy of the parameters d_1, d_2, f_1, and f_2. From each parameter estimate, Figures 4.4 and 4.5 show the MSE vs. SNR plots of the five methods as well as the Cramer-Rao Bound (CRB), which is the lower bound of the Cramer-Rao inequality [102]. Each data point is measured based on 500 independent runs of the random noise process.

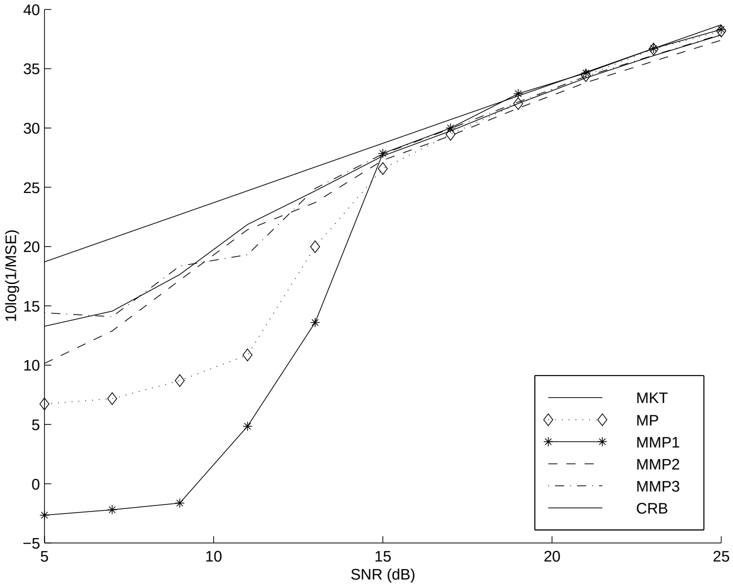
From Figures 4.4 and 4.5, all five methods yield about the same results at high SNRs but quite different results at low SNRs. The MMP1 method achieves comparable performance in estimating the frequencies as the MP method, but performs worse in estimating the damping factors. This may be caused by the negligence of the rank-deficient and Hankel properties possessed by the master matrix. Both MMP2 and MMP3 methods performed significantly better than the MP method in estimating f_2, d_1, and d_2, and MMP3 gives a better estimation for f_1 than the MP method. This is consistent with our analysis. On the other hand, both the MMP2 and MMP3 methods achieve performance comparable to the MKT method in terms of the estimation accuracy of the damping factors, and significantly outperform the MKT method in the estimation accuracy of the frequencies.

Second, I compare our proposed methods with MKT method to estimate a channel impulse response modeled as an IIR filter. The channel is the same as in [36] which is

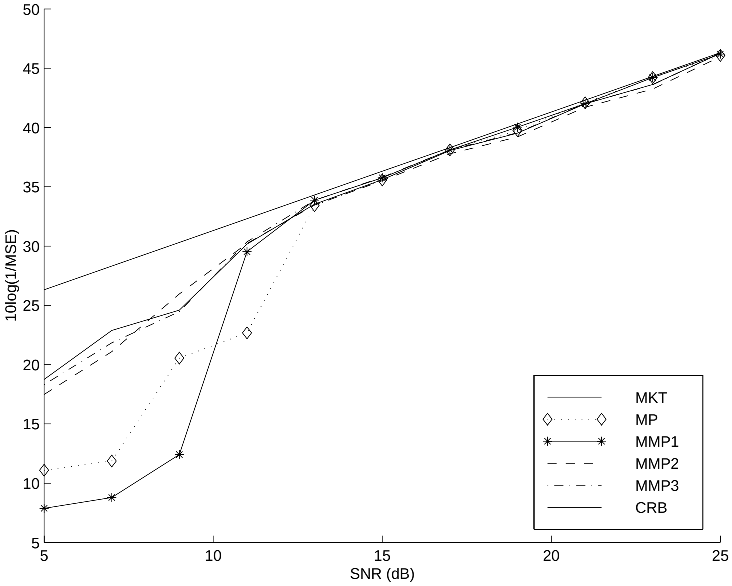
$$H(z) = \frac{B(z)}{A(z)} = \frac{-1 + 0.5354z^{-1} + 0.5056z^{-2}}{1 - 1.3501z^{-1} + 0.2757z^{-2} + 0.1275z^{-3}} \tag{4.8}$$

The three poles in (4.8) are at 0.8464, 0.7146, and 0.2108 while the two zeros are at 1.0275 and -0.4921. The noise-corrupted impulse response is used to estimate the poles by the proposed methods and MKT methods for comparison. The zeros are estimated from Shanks' method [101] after the poles are obtained. The MSE between the estimated poles and the original poles is the performance measure for the simulation. SNR varies from 0 dB to 30 dB with an increment of 3 dB. Each data point is calculated according to 500 independent runs of the random noise process.

Figures 4.6, 4.7, and 4.8 show MSE vs. SNR plot for the MKT, MP, and three modified MP methods to estimate the pole locations. All of the modified MP methods and original MP method outperform the MKT method with respect to MSE. The MMP3 method outperforms the original MP method.

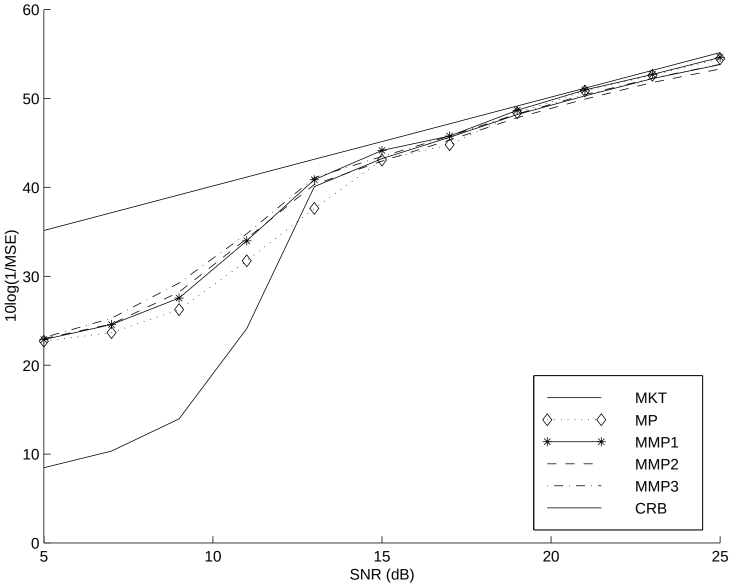


(a)

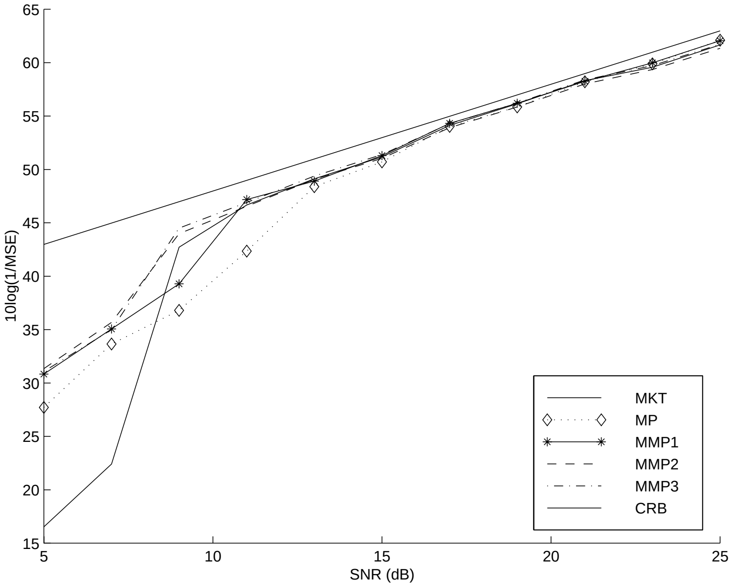


(b)

Figure 4.4: Performance comparison among the modified matrix pencil methods based on 500 trials in estimating damping factors: (a) d_1 and (b) d_2.



(a)



(b)

Figure 4.5: Performance comparison among the modified matrix pencil methods based on 500 trials in estimating frequencies: (a) f_1 and (b) f_2.

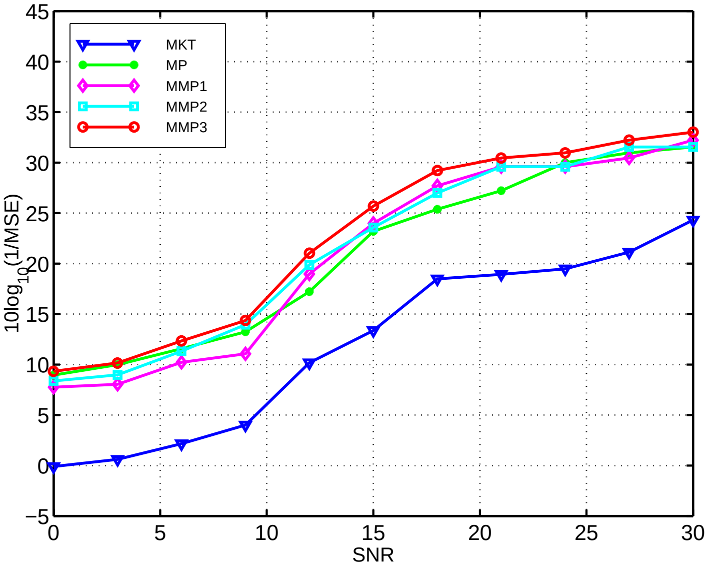


Figure 4.6: Performance comparison of pole 1 at 0.8464 by five parameter estimation methods: modified Kumaresan-Tufts (MKT), matrix pencil (MP), and three modified matrix pencil methods (MMP1 – MMP3).

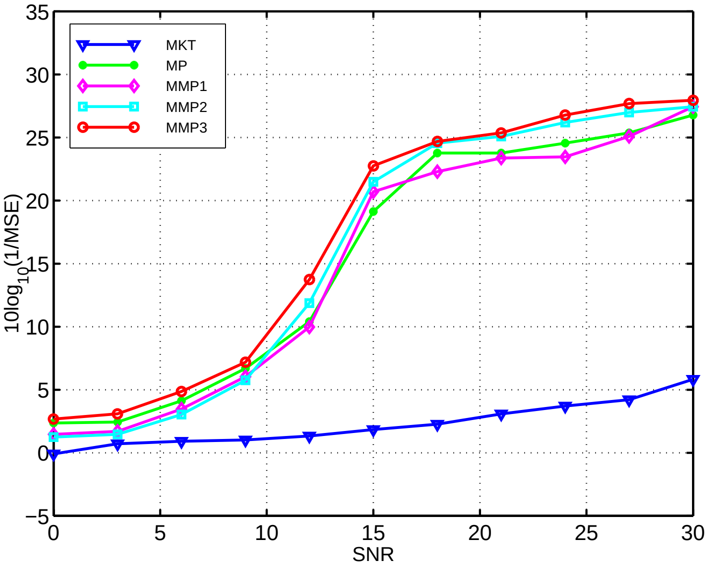


Figure 4.7: Performance comparison of pole 2 at 0.7146 by five parameter estimation methods: modified Kumaresan-Tufts (MKT), matrix pencil (MP), and three modified matrix pencil methods (MMP1 – MMP3).

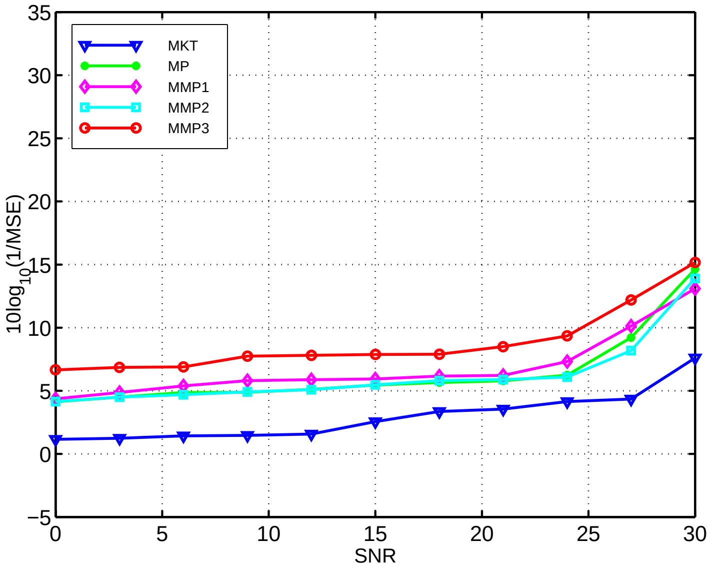


Figure 4.8: Performance comparison of pole 3 at 0.2108 by five parameter estimation methods: modified Kumaresan-Tufts (MKT), matrix pencil (MP), and three modified matrix pencil methods (MMP1 – MMP3).

4.5 Conclusion

This chapter develops and evaluates three modified versions of the MP method. The simulation results from estimating the frequencies and damping factors show that MMP2 and MMP3 obtained a lower MSE than the original MP method, MKT method, and MMP1 method. In addition, the modified MP methods can be easily extended to the forward-and-backward MP method [13] for estimating parameters of undamped sinusoids. The effect of noise in the data matrices can be attenuated by using a *reduced-rank joint Toeplitz-Hankel approximation*, which was developed in [14] for improving the Tufts-Kumaresan method [103].

In addition, I use the MKT, MP, MMP1, MMP2, and MMP3 methods to estimate the poles of an IIR channel from its impulse response. The MMP3 method again shows better performance in terms of MSE between the original poles and the estimated poles. The MMP3 method achieves significantly better performance than the original MP method, the MMP1 method, and the MMP2 method, and also outperforms the MKT method. The gap in performance increases as SNR decreases.

Poles cause a long tail in a channel impulse response. More memory and computational cost are needed to equalize a longer channel. If a filter were designed to have zeros at the pole locations of an IIR channel, then the convolution between the IIR channel and the filter would produce a shortened impulse response, which will be discussed in next chapter.

Chapter 5

Time-Domain Equalization for Discrete Multitone Modulation

In a DMT wireline receiver, a TEQ (see Section 2.5) is used to reduce ISI by decreasing the effective duration of the channel impulse response. This chapter develops new blind and non-blind methods for designing finite impulse response TEQs. The methods attempt to maximize the SSNR in (2.30). For a known channel, I develop two real-time sub-optimal divide-and-conquer (DC) methods that divide the TEQ into a cascade of two-tap TEQs and design each two-tap TEQ by a greedy approach. For an unknown channel, I model the channel as an infinite impulse response (IIR) filter in which the poles account for the long tail of the channel impulse response. Based on the IIR model, I develop two MP methods (see Sections 2.2.3 and 4.3.3) that first estimate the poles by using samples of the channel impulse response or received signal, and then use the pole locations as the zero locations of a TEQ to shorten the channel impulse response. I simulate the DC TEQ and MP TEQ designs for eight CSA DSL channels and a dedicated data channel to measure the resulting SSNR. When compared to previously reported TEQ design methods, the DC methods have better SSNR performance vs. complexity tradeoff for known channels, and the MP methods give higher SSNR for unknown channels.

5.1 Introduction

Discrete multitone (DMT) modulation is widely used in high-speed data transmission over a wireline channel and has been standardized for ADSL [6]. In ADSL, a guard sequence known as the *cyclic prefix* is prepended to each symbol to help the receiver eliminate ISI and perform symbol recovery. A DMT symbol consists of N samples, and the cyclic prefix is a copy of the last ν samples of the N samples. The length of the channel impulse response has to be less than or equal to $(\nu + 1)$ samples in order to remove ISI completely.

When using a cyclic prefix, the channel throughput of a DMT transceiver is reduced by a factor of $N/(N + \nu)$. The channel throughput increases with increasing N or decreasing ν. Increasing N increases memory and computational requirements in a transceiver. Moreover, choosing ν *a priori* is difficult because the channel length is generally not known. When a fixed value of ν is used, a time-domain equalizer (TEQ) may be required to shorten the effective length of the channel impulse response [17].

The ideal shortened channel impulse response would be zero-valued outside of a window of $(\nu + 1)$ samples starting at sample $(\Delta + 1)$, which would yield zero ISI. This ideal shortened channel impulse response would have infinite SSNR [11] as discussed in Section 2.5.2. SSNR is the ratio of the energy inside the window of $(\nu + 1)$ samples starting at sample $(\Delta + 1)$ to the energy outside of the same window of the shortened channel impulse response. The larger the SSNR, the closer the shortened channel impulse response is to the ideal. The SSNR is one objective function that could be optimized in TEQ design. Section 2.5 summarizes TEQ design methods and the objective functions they optimize.

In this chapter, the primary contributions are two non-blind and two blind FIR TEQ design methods to maximize the SSNR. Although the two non-blind Divide-and-Conquer TEQ design methods are sub-optimal, they are well suited for a real-time implementation on a digital signal processor. The Matrix Pencil TEQ design methods are blind in that they do not need a training sequence or knowledge of channel impulse response. The matrix pencil methods model the channel as an infinite impulse response (IIR) filter, estimate the number and locations of the poles, and cancel the poles with a TEQ which has zeros at the pole locations.

In this chapter, Section 5.2 proposes a low-complexity heuristic to search for the optimal delay Δ and derives two sub-optimal, low-complexity Divide-and-Conquer methods to maximize the SSNR. The idea for the Divide-and-Conquer method comes from Lloyd Clark [19]. In the Divide-and-Conquer cancellation with unit tap constraint method, the closed-form solution for g_i is due to Güner Arslan [20]. I compare the computational complexity of the maximum SSNR method [11] and Divide-and-Conquer methods. Section 5.3 develops two blind channel shortening methods that model the channel as an IIR filter and apply a Matrix Pencil method to estimate the poles of the channel. Section 5.4 simulates the proposed Divide-and-Conquer and Matrix Pencil equalization methods for nine different wireline channels. Section 5.5 concludes the paper. When compared to previously reported TEQ design methods,

- the Divide-and-Conquer methods have better performance vs. complexity tradeoff when the channel is known, and
- the Matrix Pencil methods do not require the knowledge of training se-

quence and give higher SSNR when the channel is unknown.

5.2 Divide-and-Conquer Method

In this section, I reduce the computational complexity when maximizing SSNR given by (2.30). The maximum SSNR method [11] applies the $\mathcal{O}(N_w^3 + L_h\, N_w^2)$ algorithm given in Table 2.1 to compute the N_w TEQ taps for each candidate value of the delay Δ, where $1 \leq \Delta \leq L_h + N_w - \nu - 2$. The best solution corresponds to the delay and TEQ taps that yield the highest SSNR.

Section 5.2.1 proposes a heuristic search for the optimum delay Δ so that the TEQ taps are only computed once. The search requires $\mathcal{O}(L_h)$ multiplications and additions for each value of Δ considered. Sections 5.2.2 and 5.2.3 propose two low-complexity sub-optimal *Divide-and-Conquer TEQ* (DC-TEQ) design methods that compute the TEQ taps for a given Δ. The DC-TEQ methods divide the N_w taps of the TEQ filter into $(N_w - 1)$ two-tap filters and iteratively design each two-tap filter to maximize the SSNR or minimize the energy in $\mathbf{h}_{\text{wall}}$. After dividing the TEQ filter into $(N_w - 1)$ two-tap FIR filters, I add either unit tap constraint (UTC) or unit norm constraint (UNC) to each two-tap filter.

- Unit tap constraint

  The first tap of each filter is set to one to prevent the trivial solution. For the ith two-tap filter, the coefficients are

  $$\mathbf{w}_i = [1, g_i]^T \tag{5.1}$$

- Unit norm constraint

  Each two-tap filter has the unit norm. For the ith two-tap filter, the coefficients are

$$\mathbf{w}_i = [\sin\theta_i, \cos\theta_i]^T \tag{5.2}$$

The heuristic search and the divide-and-conquer methods are well suited for implementation under fixed-point arithmetic, whereas the full SSNR method suffers from severe scaling problems under fixed-point arithmetic.

5.2.1 Finding the delay Δ

The original maximum SSNR method varies the delay Δ to maximize

$$\text{SSNR} = J_1 = \frac{\mathbf{w}^T\mathbf{B}\mathbf{w}}{\mathbf{w}^T\mathbf{A}\mathbf{w}} \tag{5.3}$$

subject to the constraint $\mathbf{w}^T\mathbf{B}\mathbf{w} = 1$. Instead of performing the optimization for each candidate value of Δ, I estimate the optimal delay Δ first and then solve for the TEQ taps once. The following heuristic for Δ_{ratio} estimates the optimal delay:

$$\Delta_{\text{ratio}} = \arg\max_{\Delta} \frac{\text{energy inside a window of the channel impulse response}}{\text{energy outside a window of the channel impulse response}} \tag{5.4}$$

Here, the window is of length $(\nu+1)$ samples beginning at index $(\Delta+1)$. The calculation of (5.4) requires L_h multiplications, $L_h - 2$ additions, and one division per value of Δ. The multiplications and additions can be reused if additional Δ values are considered. If Δ were enumerated from 0 to ν, then the calculation of (5.4) would require L_h multiplications, $2(\nu+1) + L_h - 2$ additions, and $(\nu+1)$ divisions. The calculation of (5.4) does not depend on N_w.

For each value of Δ being considered, the full SSNR algorithm requires $\left(\frac{7}{6} + L_h\right) N_w + \frac{5}{2} N_w^2 + \frac{25}{3} N_w^3$ multiplications, $4N_w^2 + 2N_w$ fewer additions than multiplications, and N_w^2 divisions, assuming that N_w iterations are used for the inverse power method in steps 6.3–6.6 in Table 2.1 to estimate the eigenvector associated with the minimum eigenvalue of $\mathbf{C}$. The heuristic always requires fewer multiplications, additions, and divisions if more than one Δ value is considered by the full SSNR algorithm. For example, with $N_w = 10$, $L_h = 512$, and $\nu = 32$, the heuristic requires 512 multiplications to find Δ and 37,295 multiplications to compute the TEQ taps for the estimated Δ. The full SSNR algorithm requires $33 \times 37,295 = 1,230,735$ multiplications, when searching $\Delta \in [0, \nu]$. Yet, the difference in SSNR between the heuristic search and the exhaustive search is less than 1 dB.

5.2.2 Divide-and-Conquer TEQ by minimization

The divide-and-conquer method minimizes

$$J_2 = \frac{\mathbf{w}^T \mathbf{A} \mathbf{w}}{\mathbf{w}^T \mathbf{B} \mathbf{w}} \tag{5.5}$$

which is equivalent to maximizing J_1 in (5.3). I call this method *DC-TEQ-Minimization.* I derive the DC-TEQ-minimization with unit tap constraint (DC-TEQ-minimization (UTC)) and DC-TEQ-minimization with unit norm constraint (DC-TEQ-minimization (UNC)).

In the case of DC-TEQ-minimization (UTC), I substitute (5.1) into (5.5). For the ith two-tap filter, the $\mathbf{A}_i$ and $\mathbf{B}_i$ matrices are 2×2, and the objective function becomes

$$J_{2,i}(\mathrm{UTC}) = \frac{\mathbf{w}_i^T \mathbf{A}_i \mathbf{w}_i}{\mathbf{w}_i^T \mathbf{B}_i \mathbf{w}_i} = \frac{\begin{bmatrix} 1 & g_i \end{bmatrix} \begin{bmatrix} a_{1,i} & a_{2,i} \\ a_{2,i} & a_{3,i} \end{bmatrix} \begin{bmatrix} 1 \\ g_i \end{bmatrix}}{\begin{bmatrix} 1 & g_i \end{bmatrix} \begin{bmatrix} b_{1,i} & b_{2,i} \\ b_{2,i} & b_{3,i} \end{bmatrix} \begin{bmatrix} 1 \\ g_i \end{bmatrix}} = \frac{a_{1,i} + 2a_{2,i} g_i + a_{3,i} g_i^2}{b_{1,i} + 2b_{2,i} g_i + b_{3,i} g_i^2} \tag{5.6}$$

Appendix A proves that the denominator in (5.6) is not equal to zero for all real values of g_i. By differentiating $J_{2,i}$ in (5.6) with respect to g_i, setting the derivative to zero, and simplifying the results,

$$(a_{3,i}b_{2,i} - a_{2,i}b_{3,i})g_i^2 + (a_{3,i}b_{1,i} - a_{1,i}b_{3,i})g_i + (a_{2,i}b_{1,i} - a_{1,i}b_{2,i}) = 0 \tag{5.7}$$

The solutions to the quadratic function of g_i in (5.7) are

$$\begin{aligned} g_{i(1,2)} &= \frac{-(a_{3,i}b_{1,i} - a_{1,i}b_{3,i})}{2(a_{3,i}b_{2,i} - a_{2,i}b_{3,i})} \\ &\pm \frac{\sqrt{(a_{3,i}b_{1,i} - a_{1,i}b_{3,i})^2 - 4(a_{3,i}b_{2,i} - a_{2,i}b_{3,i})(a_{2,i}b_{1,i} - a_{1,i}b_{2,i})}}{2(a_{3,i}b_{2,i} - a_{2,i}b_{3,i})} \end{aligned} \tag{5.8}$$

Appendix B proves that $g_{i(1)}$ and $g_{i(2)}$ are real-valued. I choose the value of g_i from $\{g_{i(1)}, g_{i(2)2}\}$ in (5.8) that gives the lower (minimum) value of J_2. Table 5.1 gives the DC-TEQ-minimization method.

In the case of DC-TEQ-minimization (UNC), I rewrite (5.2) as

$$\mathbf{w}_i = [\sin\theta_i, \cos\theta_i]^T = \sin\theta_i \left[1, \frac{\cos\theta_i}{\sin\theta_i}\right]^T = \sin\theta_i[1, \eta_i]^T \tag{5.9}$$

Then, I can substitute (5.9) into (5.5) to obtain

$$\begin{aligned} J_{2,i}(\text{UNC}) &= \frac{\mathbf{w}_i^T\mathbf{A}_i\mathbf{w}_i}{\mathbf{w}_i^T\mathbf{B}_i\mathbf{w}_i} = \frac{\sin\theta_i \begin{bmatrix} 1 & \eta_i \end{bmatrix} \begin{bmatrix} a_{1,i} & a_{2,i} \\ a_{2,i} & a_{3,i} \end{bmatrix} \sin\theta_i \begin{bmatrix} 1 \\ \eta_i \end{bmatrix}}{\sin\theta_i \begin{bmatrix} 1 & \eta_i \end{bmatrix} \begin{bmatrix} b_{1,i} & b_{2,i} \\ b_{2,i} & b_{3,i} \end{bmatrix} \sin\theta_i \begin{bmatrix} 1 \\ \eta_i \end{bmatrix}} \\ &= \frac{\begin{bmatrix} 1 & \eta_i \end{bmatrix} \begin{bmatrix} a_{1,i} & a_{2,i} \\ a_{2,i} & a_{3,i} \end{bmatrix} \begin{bmatrix} 1 \\ \eta_i \end{bmatrix}}{\begin{bmatrix} 1 & \eta_i \end{bmatrix} \begin{bmatrix} b_{1,i} & b_{2,i} \\ b_{2,i} & b_{3,i} \end{bmatrix} \begin{bmatrix} 1 \\ \eta_i \end{bmatrix}} \end{aligned} \tag{5.10}$$

From (5.10), I can calculate η_i in the same way as g_i in (5.8). That is, the DC-TEQ-minimization gives the same solution with unit tap constraint and unit norm constraint.

Step	*Description*	$\times$	$+$	$\div$
1	Initialize $\mathbf{w}_{\text{TEQ}} = [1]$			
2	Initialize $\mathbf{h}_0 = \mathbf{h}$			
3	Fix Δ. For $i = 1 \ldots N_w - 1$,			
3.1	Compute $\mathbf{A}_{i,(N_w \times N_w)}$, $N_w = 2$	$3(L_{h_i} - \nu - 1)$	$3(L_{h_i} - \nu - 1)$	0
3.2	Compute $\mathbf{B}_{i,(N_w \times N_w)}$, $N_w = 2$	$3(\nu + 1)$	$3(\nu + 1)$	0
3.3	Calculate $g_{i(1,2)}$ from (5.8)	12	7	1
3.4	Calculate $J_{2,i}$ in (5.6) from $g_{i(1)}$	6	4	1
3.5	Calculate $J_{2,i}$ in (5.6) from $g_{i(2)}$	6	4	1
3.6	Choose a g_i from $g_{i(1)}$ and $g_{i(2)}$ which gives the smaller $J_{2,i}$.			
3.7	If $g > 10^{-5}$, calculate $\mathbf{w}_{\text{TEQ}} = \mathbf{w}_{\text{TEQ}} * \mathbf{w}_i$	$2(i + 2)$	$2(i + 1)$	0
3.8	Calculate $\mathbf{h}_i = \mathbf{h}_{i-1} * \mathbf{w}_i$	$2(L_h + i)$	$2(L_h + i - 1)$	0
3.9	If $g_i < 10^{-5}$, stop.			

Table 5.1: Implementation and computational cost of the Divide-and-Conquer TEQ method to maximize the SSNR given by (5.3) for a fixed Δ where $L_{h_i} = L_h + i - 1 \leq L_h + N_w - 1$ and $\times$, $+$, and $\div$ are multiplication, addition, and division, respectively.

5.2.3 Divide-and-Conquer TEQ to cancel the energy in $\mathbf{h}_{\text{wall}}$

The DC-TEQ-minimization method requires the calculation of $\mathbf{A}_i$ and $\mathbf{B}_i$ in each iteration. In order to avoid calculating the matrices $\mathbf{A}_i$ and $\mathbf{B}_i$, I apply divide-and-conquer directly to the channel impulse response to cancel the energy in $\mathbf{h}_{\text{wall}}$. I call this method *DC-TEQ-cancellation.* For each iteration i of the method, $\tilde{\mathbf{h}}_{\text{wall}_i}$ is computed similarly to $\mathbf{h}_{\text{wall}}$ in (2.32). The equalized channel impulse response at the output of the ith filter is denoted as $\tilde{\mathbf{h}}_i$, which is the convolution of $\tilde{\mathbf{h}}_{i-1}$ and $\mathbf{w}_i$ where $\tilde{\mathbf{h}}_0$ is the channel impulse response $[h(1), h(2), \ldots \cdots h(L_h)]$.

In the case of DC-TEQ-minimization (UTC), the energy of $\tilde{\mathbf{h}}_{\text{wall}_i}$ at the ith iteration is given by

$$\begin{aligned}
\tilde{\mathbf{h}}_{\text{wall}_i} &= \begin{bmatrix} \tilde{h}_{i-1}(1) & 0 \\ \tilde{h}_{i-1}(2) & \tilde{h}_{i-1}(1) \\ \vdots & \vdots \\ \tilde{h}_{i-1}(\Delta) & \tilde{h}_{i-1}(\Delta-1) \\ \tilde{h}_{i-1}(\Delta+\nu+2) & \tilde{h}_{i-1}(\Delta+\nu+1) \\ \vdots & \vdots \\ \tilde{h}_{i-1}(L_{h_{i-1}}) & \tilde{h}_{i-1}(L_{h_{i-1}}-1) \end{bmatrix} \begin{bmatrix} 1 \\ g_i \end{bmatrix} \\
&= \begin{bmatrix} \tilde{h}_{i-1}(1) & +0 \\ \tilde{h}_{i-1}(2) & +\tilde{h}_{i-1}(1)\, g_i \\ \vdots & \vdots \\ \tilde{h}_{i-1}(\Delta) & +\tilde{h}_{i-1}(\Delta-1)\, g_i \\ \tilde{h}_{i-1}(\Delta+\nu+2) & +\tilde{h}_{i-1}(\Delta+\nu+1)\, g_i \\ \vdots & \vdots \\ \tilde{h}_{i-1}(L_{h_{i-1}}) & +\tilde{h}_{i-1}(L_{h_{i-1}}-1)\, g_i \end{bmatrix}
\end{aligned} \tag{5.11}$$

Here, $L_{h_{i-1}}$ is the length of $\tilde{\mathbf{h}}_{i-1}$ at the ith iteration. Then, the energy in $\tilde{\mathbf{h}}_{\text{wall}}$ can be expressed as

$$\tilde{\mathbf{h}}_{\text{wall}_i}^T \tilde{\mathbf{h}}_{\text{wall}_i} = \textstyle\sum_{k \in S} \left(\tilde{h}_{i-1}(k) + g_i\, \tilde{h}_{i-1}(k-1) \right)^2 \tag{5.12}$$

Step	*Description*	$\times$	$+$	$\div$
1	Initialize $\mathbf{w}_{\text{TEQ}} = [1]$			
2	Initialize $\mathbf{h}_0 = \mathbf{h}$			
3	Fix Δ. For $i = 1 \ldots N_w - 1$,			
3.1	Calculate g_i from (5.13)	$2(L_{h_i} - \nu - 1)$	$2(L_{h_i} - \nu - 2)$	1
3.2	If $g_i > 10^{-5}$, calculate $\mathbf{w}_{\text{TEQ}} = \mathbf{w}_{\text{TEQ}} * \mathbf{w}_i$	$2(i+2)$	$2(i+1)$	0
3.3	Convolve $\tilde{\mathbf{h}}_{i-1}$ with this two-tap, $\mathbf{w}_i$, to obtain $\tilde{\mathbf{h}}_i$	$2(L_h + i)$	$2(L_h + i - 1)$	0

Table 5.2: Implementation and computational cost in the Divide-and-Conquer-TEQ-cancellation (UTC) for a fixed Δ where $L_{h_i} = L_h + i - 1 \le L_h + N_w - 1$ and $\times$, $+$, and $\div$ are multiplication, addition, and division, respectively.

where $S = \{1, 2, \ldots, \Delta, \Delta + \nu + 2, \ldots, L_{h_{i-1}}\}$. By differentiating (5.12) with respect to g_i, setting the derivative to zero and solving for g_i yield the minimum of the quadratic function (5.12).

In the case of DC-TEQ-cancellation with unit norm constraint (DC-TEQ-cancellation (UNC)), the energy of $\tilde{\mathbf{h}}_{\text{wall}_i}$ at the ith iteration is given by

$$g_i = -\frac{\sum\limits_{k \in S} \tilde{h}_{i-1}(k-1)\, \tilde{h}_{i-1}(k)}{\sum\limits_{k \in S} \tilde{h}_{i-1}^2(k-1)} \tag{5.13}$$

This calculation requires one scalar division and two vector multiplications. Table 5.2 gives one iteration of the DC-TEQ-cancellation (UTC) method.

$$\begin{aligned}\tilde{\mathbf{h}}_{\text{wall}_i} &= \left[\begin{array}{ll} \tilde{h}_{i-1}(1) & 0 \\ \tilde{h}_{i-1}(2) & \tilde{h}_{i-1}(1) \\ \vdots & \vdots \\ \tilde{h}_{i-1}(\Delta) & \tilde{h}_{i-1}(\Delta-1) \\ \tilde{h}_{i-1}(\Delta+\nu+2) & \tilde{h}_{i-1}(\Delta+\nu+1) \\ \vdots & \vdots \\ \tilde{h}_{i-1}(L_{h_{i-1}}) & \tilde{h}_{i-1}(L_{h_{i-1}}-1) \end{array}\right] \left[\begin{array}{l} \sin\theta_i \\ \cos\theta_i \end{array}\right] \qquad (5.14) \\ &= \textstyle\sum_{k\in S} \left(\tilde{h}_{i-1}(k)\sin\theta_i + \tilde{h}_{i-1}(k-1)\cos\theta_i\right)^2\end{aligned}$$

By differentiating (5.14) with respect to θ_i, setting the derivative to zero, and solving for $\sin\theta_i$ and $\cos\theta_i$, I find the minimum of (5.14) to be

$$\sin\theta_i = \pm\sqrt{0.5\left(1 \mp \sqrt{\frac{a^2}{a^2+4b^2}}\right)} \qquad (5.15)$$

and

$$\cos\theta_i = \pm\sqrt{0.5\left(1 \pm \sqrt{\frac{a^2}{a^2+4b^2}}\right)} \qquad (5.16)$$

where

$$a = \sum_{k\in S}\left(\tilde{h}_{i-1}^2(k) - \tilde{h}_{i-1}^2(k-1)\right)$$

and

$$b = \sum_{k\in S}\tilde{h}_{i-1}(k-1)\tilde{h}_{i-1}(k)$$

Table 5.3 gives one iteration of the DC-TEQ-cancellation (UNC) method.

Unlike the maximum SSNR method in [11], both DC-TEQ-minimization and DC-TEQ-cancellation methods do not require Cholesky decomposition, eigenvalue decomposition, or matrix inversion. For a fair comparison, I replace the eigenvalue decomposition in the maximum SSNR method by the iterative power method [104] since only the minimum eigenvalue and its corresponding eigenvector are needed. For the power method, I use N_w iterations to find the minimum eigenvalue and its corresponding eigenvector. Table 5.4 compares the number of multiplications, additions, and divisions and memory requirements for a fixed value of Δ for the design of a 21-tap TEQ ($N_w = 21$) with $N = 512$, $\nu = 32$, and $L_h = 512$ for the SSNR and the two Divide-and-Conquer methods. Both proposed methods not only have reduced computational cost and memory requirements compared to the maximum SSNR method but are also well suited for real-time implementation on digital signal processors.

5.3 Blind Channel Shortening

The sub-optimum methods in Sections 5.2.2 and 5.2.3 require accurate knowledge of the channel impulse response. In some applications, a reliable estimate of the channel impulse response may be obtained in advance. For example, the ADSL standard [6] provides a periodic training sequence with period equal to the length of the DMT symbol. By averaging a large number of received data frames, noise can be smoothed out. Then, an estimate of the discrete Fourier transform (DFT) of the channel impulse response can be obtained by dividing the average by the DFT of the training sequence. Without a reliable estimate of the channel impulse response, the methods that require knowledge of the channel *a priori* may fail to shorten the effective channel impulse response.

This section proposes a new blind TEQ design method for channel shortening. Section 5.3.1 models the channel as an IIR filter. Section 5.3.2 develops two Matrix Pencil (MP) methods (see Sections 2.2.4 and 4.3.3) to estimate the number and values of the channel poles. The new methods can estimate the TEQ coefficients.

Step	*Description*	$\times$	$+$	$\div$
1	Initialize $\mathbf{w}_{\text{TEQ}} = [1]$			
2	Initialize $\mathbf{h}_0 = \mathbf{h}$			
3	Fix Δ. For $i = 1 \ldots N_w - 1$,			
3.1	Calculate $\sin\theta_i$ and $\cos\theta_i$ from (5.15) and (5.16)	$2(L_{h_i} - \nu + 2)$	$2(L_{h_i} - \nu + 1)$	1
3.2	If $g_i > 10^{-5}$, calculate $\mathbf{w}_{\text{TEQ}} = \mathbf{w}_{\text{TEQ}} * \mathbf{w}_i$	$2(i+2)$	$2(i+1)$	0
3.3	Convolve $\tilde{\mathbf{h}}_{i-1}$ with this two-tap, $\mathbf{w}_i$, to obtain $\tilde{\mathbf{h}}_i$	$2(L_h + i)$	$2(L_h + i - 1)$	0

Table 5.3: Implementation and computational cost in the Divide-and-Conquer-TEQ-cancellation (UNC) for a fixed Δ where $L_{h_i} = L_h + i - 1 \leq L_h + N_w - 1$ and $\times$, $+$, and $\div$ are multiplication, addition, and division, respectively.

Methods	*Multiplications*	*Additions*	*Divisions*	*Memory* (words)
Maximum SSNR	120379	118552	441	1899
DC-TEQ-minimization (UTC & UNC)	53240	52980	60	563
DC-TEQ-cancellation (UNC)	42280	42160	20	555
DC-TEQ-cancellation (UTC)	41000	40880	20	554

Table 5.4: Computational cost for maximum SSNR method and two proposed methods: Divide-and-Conquer-TEQ-minimization and Divide-and-Conquer-TEQ-cancellation with $\nu = 32$, $N_w = 21$, and $L_h = 512$ for a fixed value of Δ.

5.3.1 Models of discrete multitone wireline channels

Fig. 5.1 shows impulse responses of two wireline channels. Since the channel impulse responses have long decaying tails, one may consider the DMT channels as IIR filters. Also, motivated by the MMSE TEQ in Section 2.5.1, I write the transfer function of a wireline channel as

$$H(z) = \frac{B(z)}{W(z)} \tag{5.17}$$

where $W(z)$ is the TEQ transfer function and $B(z)$ is the transfer function of the cascade of the channel and the TEQ. The IIR filter would have ν zeros and $N_w - 1$ poles. Once the channel has been estimated, the channel may be modeled as an ARMA$(\nu, N_w - 1)$ process [105]. Al-Dhahir, Sayed, and Cioffi propose a pole-zero model by using the generalized ARMA-Levinson algorithm and apply it to the decision feedback equalizer [36].

Based on the ARMA channel model, the TEQ can be used to cancel the channel poles, which shortens the channel response. The transfer function in (5.17) can be rewritten as

$$H(z) = \frac{B(z)}{\prod_{i=1}^{N_w-1} (1 - C_i z^{-1})} \tag{5.18}$$

which is the same as (2.2). Wang and Smith also propose to model an FIR sequence as a truncated exponential sequence to reduce computational cost [106]. Without loss of generality, the ith pole C_i may be defined as a complex number. This motivates the application of the matrix pencil method to design the TEQ by estimating the locations of poles in (5.18). The poles cause the long tail of wireline channel impulse responses. Therefore, channel shortening is equivalent to finding the pole locations and canceling them out while the number of zeros of $B(z)$ in (5.18) is fixed to ν. This model leads to our proposed Matrix Pencil Method for TEQ design, since the Matrix Pencil method [37] is a robust way to extract poles from the received data as described in Chapter 4.

5.3.2 Matrix pencil TEQ

Many methods estimate the poles in (5.18). The least squares Prony method [101] can estimate sinusoids, which are the poles in the ARMA model in Section 5.3.1. The Prony method is sensitive to noise and cannot successfully estimate closely spaced poles [37]. The Matrix Pencil method proposed by Hua and Sarkar [13, 39] overcomes both disadvantages. Lu, Wei, Evans, and Bovik apply the RRHA to Hua and Sarkar's method in order to develop three modified Matrix Pencil methods [100]. One of the improved methods, MMP3 in Section 4.3.3 and in [100], which will be used in TEQ design, is similar to the new Matrix Pencil method proposed in [31, 32].

The Matrix Pencil TEQ (MP-TEQ) can shorten the channel by estimating the location of poles of the channel model in (5.18) in two ways. First, the poles can be located directly from the channel impulse response, i.e. $r_k = s_k$ in (2.7). Second, the poles are located from the received signal r_k given little or no knowledge of the channel. In this case, the received signal is

$$y(k) = x(k) * h(k) + n(k) = s(k) + n(k) \tag{5.19}$$

By substituting (5.17) into (5.19) and taking the z-transform,

$$Y(z) = X(z)\frac{B(z)}{W(z)} + N(z) \tag{5.20}$$

Based on (5.20), the received signal contains information about the channel poles. So, the Matrix Pencil method may be used to find the poles without knowing

Step	*Description*
1	Set $L = \nu$. $K \in [3L/2, 3L]$.
2	Choose K. Use $K = 25$ for $\nu = 16$ and $K = 50$ for $\nu = 32$.
3	Let the $(\Delta + 1)$st sample of the channel data be at index $k = 0$, and form the master matrix $\mathbf{Y}$ in (2.10) from K received samples.
4	From $\mathbf{Y}$, compute submatrices $\mathbf{Y}_0$ and $\mathbf{Y}_1$ given by (2.11).
5	Perform eigendecomposition of the $L \times L$ matrix $\mathbf{Y}_0^{\dagger}\mathbf{Y}_1$.
6	Use the largest N_w eigenvalues to estimate pole locations by (2.12).
7	Set the TEQ zero locations equal to the estimated pole locations.

Table 5.5: The Matrix Pencil TEQ design method will estimate $K - L$ poles where K is the number of received samples to process and $K/3 \leq L \leq 2K/3$. The dominant N_w poles become the N_w zeros of the TEQ ($N_w \leq K - L$).

the channel impulse response. The $y(k)$ and $x(k) * h(k)$ terms correspond to r_k and s_k in (2.7), respectively. I form the matrices $\mathbf{Y}$, $\mathbf{Y}_0$, and $\mathbf{Y}_1$ according to (2.10) and (2.11).

The Matrix Pencil method assumes that the number of poles M in (2.13) and (2.14) are known. For precise estimation of the channel, i.e. $y(k)$ is equal to $s(k)$ in (5.19), M is the number of non-zero eigenvalues of matrix $\mathbf{S}$. However, if the estimation error n_k is not negligible, then $\mathbf{Y}$ has full rank and the accuracy of estimating the number of poles decreases as the variance of n_k becomes large. When equalizing ADSL channels, I fix the number of zeros in $B(z)$ to be equal to ν so that the effective channel impulse response can be shortened to $(\nu + 1)$ taps after I estimate the locations of poles and cancel the poles. I use Matrix Pencil method to estimate the poles for eight CSA loops. When the SSNR in (2.30) changes only slightly after the N_w taps, the number of poles is set to be $N_w - 1$. The MP-TEQ method is given in Table 5.5.

5.4 Simulation Results

I am particularly interested in low-complexity TEQ design methods, especially those can be implemented in real-time on a digital signal processor. All of the methods in Section 2.5.3 to maximize the channel capacity have difficulty in meeting real-time constraints. Therefore, I do not include these methods in the comparisons. The simulations compare the proposed TEQ design methods in terms of SSNR as defined in (2.30) for the following cases:

- *Known channels*: it is possible that the channel impulse response is known to the receiver so that the TEQ coefficients can be calculated directly from the channel impulse response. In this case, the estimation error, n_k, in (5.19), is approximately zero. I compare the proposed methods of DC-TEQ-minimization (UTC) and DC-TEQ-cancellation (UTC and UNC) with the maximum SSNR method [11] since they all require the channel impulse response *a priori*. MP-TEQ can be used for blind and non-blind cases to cancel the tail by finding the poles of the IIR model of a channel. Therefore, I include the MP-TEQ in this comparison.

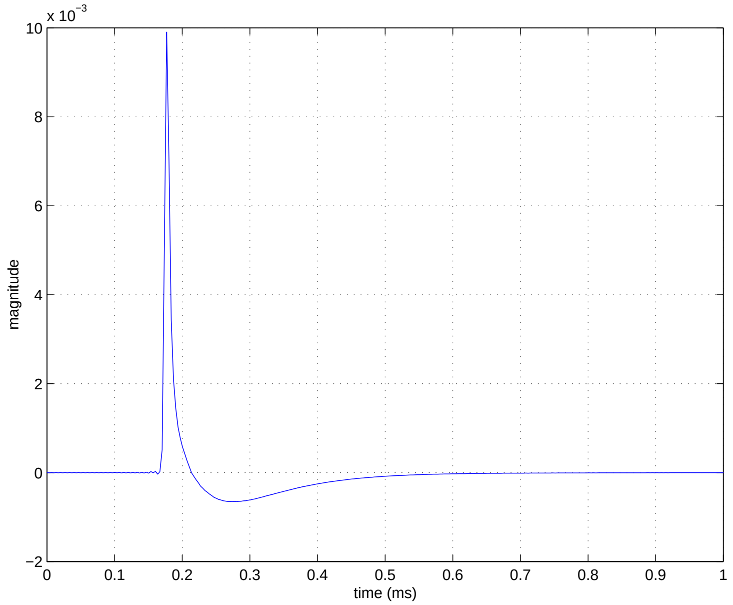
- *Unknown channel*: at the receiver, the training sequence and the received sequence are available to the equalizer. I generate a pseudo-random training sequence. The TEQ coefficients are estimated from the received data. The Matrix Pencil method can estimate the poles directly from the received data since it does not need to calculate the cross-correlation matrix between the training sequence and the received sequence. Therefore, the Matrix Pencil method serves as a *blind channel shortening* method for TEQ design. For unknown channels, I compare the MP-TEQ and modified matrix pencil TEQ (MMP3-TEQ) with the MMSE method [80, 83].

In the simulations, I use a dedicated data channel and the carrier-serving-area (CSA) digital subscribing loop (DSL) 1 channel in Fig. 5.1. Table 5.6 lists the simulation parameters.

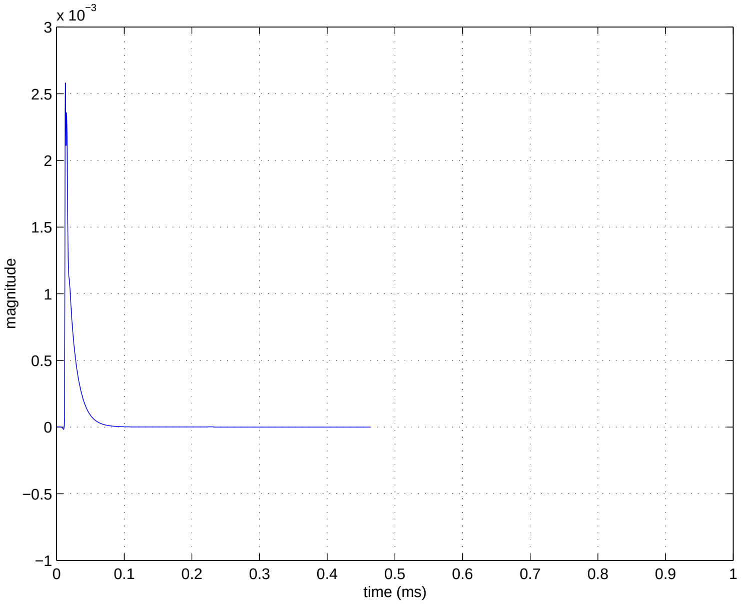
5.4.1 Comparison of SSNR for known channels

Fig. 5.2 shows SSNR obtained from the original maximum SSNR method with exhaustive search of delay [11] and the proposed sub-optimal maximum SSNR method with heuristic seach of delay (see Section 5.2.1). Fig. 5.3 compares the SSNR obtained by four of the proposed methods— DC-TEQ-minimization (unit tap constraint), DC-TEQ-cancellation (unit tap and unit norm constraints), and MP-TEQ— and the maximum SSNR [11] method. The MP-TEQ method models the channel as an IIR filter in which the poles model the tail of channel impulse response. Pole estimates are reliable when the Matrix Pencil parameter L satisfies $K/3 \leq L \leq 2K/3$, where K is the number of samples used and $K - L$ is the maximum number of poles that can be estimated [13]. The samples used may be either samples of the channel impulse response or received signal.

In Fig. 5.3(a), I choose $\nu = 16$ for the dedicated data channel and let $L = \nu$. For $L = 16$, any value of K such that $K \in [24, 48]$ would make the estimation of $K - L$ poles reliable [13]. For $K = 25$, $K - L = 9$ poles may be reliably estimated. Thus, the MP-TEQ would have up to 10 taps. In Fig. 5.3(b), I use $\nu = 32$ from the ADSL standard [6] for the CSA DSL 1 channel and let $L = \nu$ as before. For $L = 32$, $K \in [48, 96]$ and I choose $K = 50$. For this case, the MP-TEQ would have up to 18 taps. Yet, the maximum SSNR method can

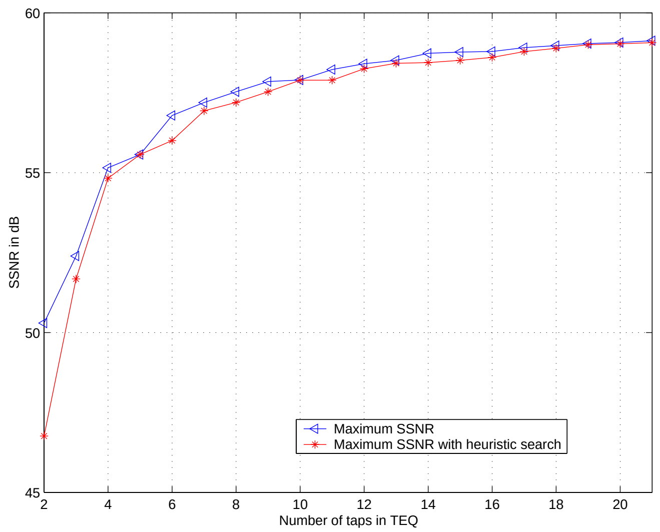


(a) Dedicated data transmission channel

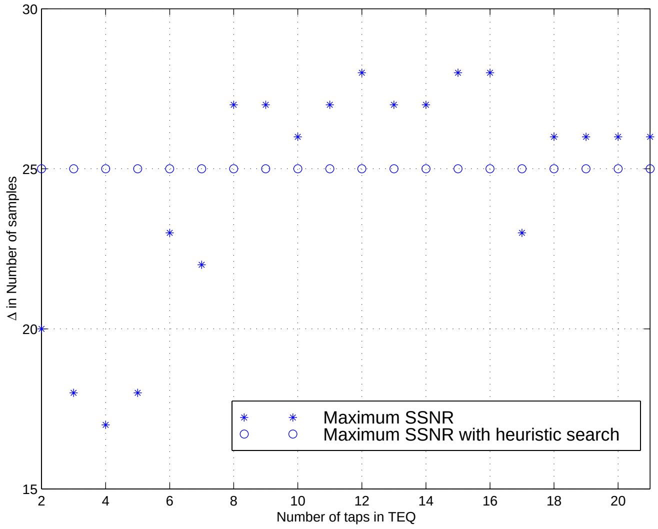


(b) CSA DSL 1 channel

Figure 5.1: Impulse response of two of the wireline channels used in the simulations.

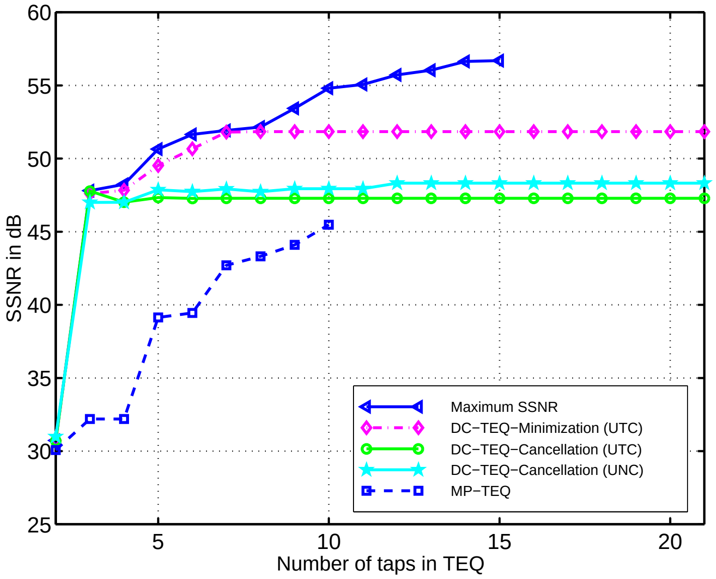


(a) SSNR between maximum SSNR [11]
and sub-optimal maximum SSNR with heuristic search

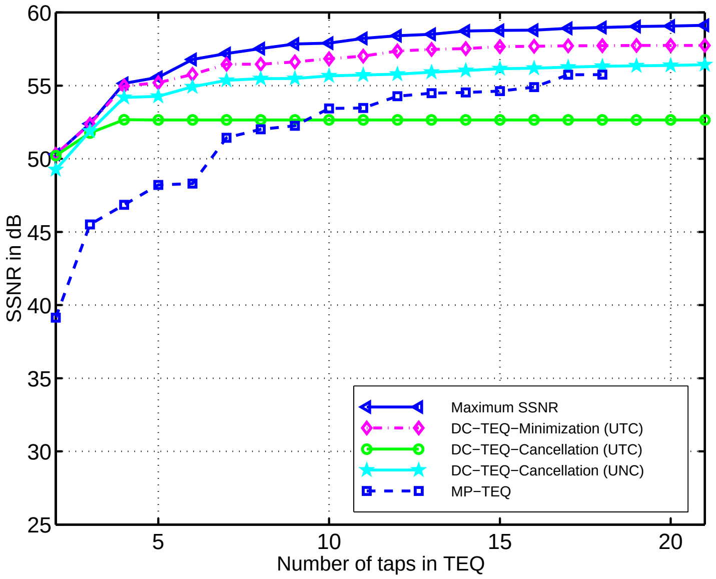


(b) Δ obtained from maximum SSNR [11]
and heuristic search

Figure 5.2: Determination of Δ for the CSA DSL 1 channel with $L_h = 512$, $\nu = 32$, and 2–21 TEQ taps.



(a) SSNR of the dedicated data channel with $L_h = 512$ and $\nu = 16$.



(b) SSNR of CSA DSL 1 channel with $L_h = 512$ and $\nu = 32$.

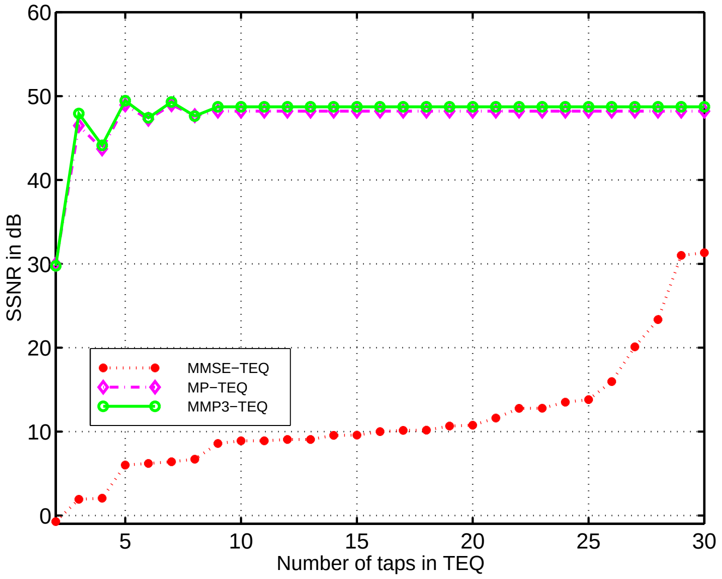
Figure 5.3: Comparison of four different TEQ design methods in terms of SSNR for two known channels with 2–21 TEQ taps.

only reach 15 taps since $\mathbf{B}$ becomes singular when $N_w \geq \nu$, as plotted in Fig. 5.3. All the other methods have 2 to 21 taps for both channels in Fig. 5.3. The SSNR for DC-TEQ saturates at four taps for the dedicated data channel and 10 taps in the CSA DSL 1 channel.

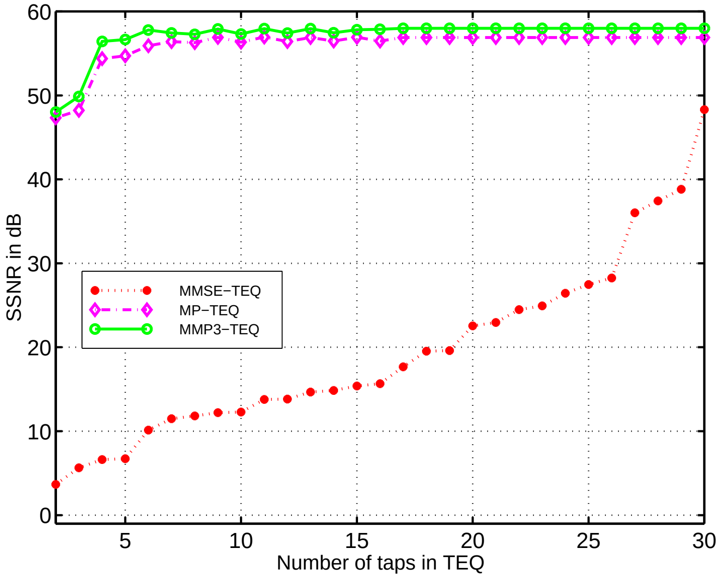
For eight CSA loops, I use the Matrix Pencil method [13] to estimate the poles, as shown in Table 5.7. The number of poles are computed according to Section 5.3.2. From Table 5.7, a TEQ may require fewer than 14 taps for all CSA loop channels.

5.4.2 Comparison of SSNR for unknown channels

I compare the MP-TEQ, MMP3-TEQ, and MMSE-TEQ design methods when the channel impulse response is unknown. Fig. 5.4 shows that MP-TEQ and MMP3-TEQ outperform MMSE-TEQ in terms of the SSNR. Both MP-TEQ and MMP3-TEQ use the same L and ν parameters as in the known channels for the dedicated data channel and CSA DSL 1 channel in Section 5.4.1. K is 47 for the dedicated data channel and 65 for the CSA DSL 1 channel so that the TEQ can have 30 taps to compare with MMSE-TEQ. Fig. 5.4 shows that both MP-TEQ and MMP3-TEQ can give higher SSNR than MMSE-TEQ when the TEQ coefficients are calculated from the received signal and/or training sequence. Both MP-TEQ and MMP3-TEQ with two taps can reach similar SSNR to MMSE-TEQ with 30 taps. This result is consistent with [100] in that the modified Matrix Pencil method provides better estimation than the original Matrix Pencil approach.



(a) SSNR of the dedicated data channel with $L_h = 512$ and $\nu = 16$.



(b) SSNR of CSA DSL 1 channel with $L_h = 512$ and $\nu = 32$.

Figure 5.4: Comparison among matrix pencil methods and MMSE method in terms of SSNR for unknown channels for 2–30 TEQ taps.

5.5 Conclusion

In discrete multitone modulation, the TEQ shortens the effective impulse response of the cascade of the channel and the TEQ. This chapter develops two new non-blind and two new blind TEQ design methods. To maximize the SSNR objective function, I develop two sub-optimum divide-and-conquer (DC) TEQ design methods with unit tap constraint and unit energy constraint. Both methods have lower computational complexity than the maximum SSNR method, but give comparable SSNR results.

I model the unknown channel as an IIR filter in which the tail of the channel response is modeled by the poles. Then, I derive two Matrix Pencil TEQ design methods that estimate the poles from the received signal without *a priori* knowledge of the channel impulse response. Finally, I use the poles of the model as the zeros of the TEQ to cancel the energy of $\mathbf{h}_{\text{wall}}$.

I test the proposed methods on eight CSA DSL channels (as shown in Table 5.7) and a dedicated data transmission channel. The sub-optimum solution to SSNR, DC-TEQ-minimization, has the best tradeoff of computational complexity vs. maximization of SSNR. For the unknown channel case, the proposed Matrix Pencil methods give higher SSNR than the MMSE-TEQ with the unit norm constraint. The Matrix Pencil methods do not have restrictions on matrix singularity.

Parameters	*Dedicated Data Channel*	*CSA DSL Channels 1–8*
sampling rate	300 kHz	2.208 MHz
samples per symbol	512	512
power of additive white Gaussian noise	−200 dBm/Hz	−113 dBm/Hz
transmitter power	1 W	1 W
transmitter termination resistance	50 Ω	100 Ω
fast Fourier transform size	512	512
cyclic prefix	16	32
bit error rate	10^{-7}	10^{-7}
field margin	6 dB	6 dB
coding gain	0 dB	0 dB
number of runs	50	50

Table 5.6: Parameters for the nine digital subscriber line channels used in simulation.

CSA loops		*loop 1*	*loop 2*	*loop 3*	*loop 4*
# of poles		7	10	8	7
complex poles pair	1	$0.57 \pm j0.25$	$0.75 \pm j0.26$	$0.94 \pm j0.19$	$0.52 \pm j0.81$
	2	$0.10 \pm j0.61$	$-0.69 \pm j0.37$	$-0.14 \pm j0.29$	$0.77 \pm j0.56$
	3	n/a	$0.13 \pm j0.72$	n/a	$0.91 \pm j0.29$
	4	n/a	$0.45 \pm j0.54$	n/a	n/a
real poles	1	-0.89	-0.93	-0.95	0.62
	2	0.89	0.92	0.93	n/a
	3	-0.87	n/a	0.88	n/a
	4	n/a	n/a	0.47	n/a
SSNR in dB	$\mathbf{h}$	10.84	11.09	6.63	10.13
	$\mathbf{h}_{\text{eff}}$	54.54	53.93	54.00	50.15

CSA loops		*loop 5*	*loop 6*	*loop 7*	*loop 8*
# of poles		6	5	8	12
complex poles pair	1	$-0.07 \pm j0.95$	$0.91 \pm j0.35$	$0.32 \pm j0.62$	$-0.87 \pm j0.17$
	2	$0.90 \pm j0.27$	$0.75 \pm j0.62$	$0.64 \pm j0.30$	$0.71 \pm j0.20$
	3	$0.75 \pm j0.51$	n/a	$-0.60 \pm j0.51$	$0.44 \pm j0.37$
	4	n/a	n/a	n/a	$-0.30 \pm j0.45$
	5	n/a	n/a	n/a	$0.09 \pm j0.49$
real poles	1	n/a	0.87	-0.94	0.91
	2	n/a	n/a	0.90	-0.89
	3	n/a	n/a	n/a	n/a
	4	n/a	n/a	n/a	n/a
SSNR in dB	$\mathbf{h}$	9.75	6.69	10.22	9.54
	$\mathbf{h}_{\text{eff}}$	50.12	53.45	48.47	62.35

Table 5.7: Locations of poles in eight CSA channels estimated by the matrix pencil method (see Sections 2.2.3 and 4.3.3) where $\mathbf{h}$ is the original channel impulse response and $\mathbf{h}_{\text{eff}}$ is the effective equalizerd channel impulse response.

Chapter 6

Conclusion

Reliable data transmission through wireline channels with severe distortion can be achieved in several ways. One way is to design an equalizer in the receiver to compensate for channel distortion. Equalization methods are used to mitigate the effects of intersymbol interference. This dissertation focuses on the design of time-domain equalizers for both voice telephone and DSL wireline channels. The contributions of the dissertation fall into two broad categories: channel equalization with and without estimating the channel impulse response.

6.1 Equalization without estimating the channel

A transmission channel is generally not known to the receiver *a priori*. In the case that an estimate of the channel is not available, the parameters of an equalizer can be obtained either from the training sequence and received signal or from the received signal only. This dissertation explores both methods by developing neural network and matrix pencil equalizers.

Chapter 3 proposes a new neural network equalizer after the discussion of feedforward neural network equalizers in Section 2.3. By considering channel equalization as a classification problem, neural networks are naturally applied to learn the behavior of an equalizer in a communication system. Both MLP and

RBF networks are employed as feedforward equalizers and have shown better performance than linear equalizers, respectively. My first contribution in Chapter 3 is to compare the performance in terms of symbol error rate between MLP equalizers and RBF equalizers. From my simulation results, RBF equalizers must use an increasingly larger number of hidden neurons than MLP equalizers in order to obtain a similar symbol error rate as SNR decreases. This is because RBF networks are more sensitive to noise than MLP networks. However, the training of MLP networks has the drawbacks of exhibiting slow convergence and falling into local minima. In order to overcome the above drawbacks from both MLP and RBF networks, I design a new nonlinear feedforward neural network equalizer by cascading two different neural networks. The first stage uses a MLP network to suppress noise and the second stage uses an RBF network to implement the decision logic. The neural network can learn the features from the received data by using the training sequence to train the neural networks. When using the neural network equalizers, it is not necessary to estimate the channel first. However, the supervised training methods for feedforward neural networks require the knowledge of both received signal and transmitted training sequence. One key drawback of this approach is the excessive computation time for training the equalizers.

Compared to the neural network equalizers, the matrix pencil method can estimate the locations of poles in wireline channel when channel is modeled as the infinite impulse response. The location of poles can be estimated either from the received signal or from the original channel impulse response. This dissertation explores both of the cases with DSL channels. Chapter 4 builds on equalizer training with the received signal only from Chapter 3 and becomes a prerequisite of Chapter 5 to design a blind channel shortening equalizer in Section 5.3. I review the matrix pencil method proposed by Hua and Sarkar [13] in Section 2.2.3. The formation of matrix pencil from a noise-free signal makes both the master matrix in (2.10) and submatrices in (2.11) be rank deficient but have Hankel structure. When noise is present, rank deficiency is no longer valid but the matrices still have Hankel structure. Performing SVD truncation would enforce rank deficiency but destroy Hankel structure.

The contribution in Chapter 4 is to apply rank reduced Hankel approximation to the matrices after SVD truncation in order for the matrices to have Hankel structure and approximate low rank. I first develop the new matrix pencil methods to achieve a more accurate estimation on the frequencies and damping factors. The new matrix pencil methods, MMP1, MMP2, and MMP3, are forced by applying RRHA to the received data matrices at different stages of the original matrix pencil method. Then, I apply the matrix pencil methods to design the time-domain equalizer for CSA DSL channels. The matrix pencil method can estimate the poles of channel directly from the received signal without the knowledge of the training sequence.

6.2 Equalization based on channel estimation

In an ADSL system, discrete multitone modulation is used as the line coding. It is possible to estimate the channel impulse response of CSA DSL loops at the receiver end. When the channel impulse response is known in the ADSL system, TEQ methods can directly obtain the coefficients. Section 2.5 reviews previous methods for designing TEQs. MMSE methods do not take the differences of samples inside and outside the cyclic prefix window into consideration, but they do not give satisfactory results in terms of SSNR in effective channel impulse response. Maximum SSNR and maximum channel capacity methods are computationally intensive. The maximum SSNR method is optimum in terms of SSNR. However, it requires Cholesky decomposition and eigenvalue decomposition to find the optimum TEQ coefficients, which are not suitable for real-time implementation on programmable digital signal processors. The complexity of maximum channel capacity methods is worse than that for the maximum SSNR method due to nonlinear optimization solver necessary to find the TEQ. My contributions in Chapter 5 are as follows.

- I propose a heuristic method to estimate the optimal delay of the window. The heuristic method estimates the optimal delay first and then applies the TEQ design method. In general, TEQ design methods perform an exhaustive search over the possible delays in the range from 1 to $L_h + N_w - \nu - 2$ samples. For the eight CSA loops, the loss in SSNR for the heuristic applied to the maximum SSNR method is less than 1 dB.

- I develop Divide-and-conquer methods that divide the TEQ filter into a cascade of two-tap filter. For each two-tap filter, the methods use a greedy approach to find the second tap with the first tap constrained to be one. Divide-and-conquer methods significantly reduce the computation cost and memory requirements. They are suitable for real-time implementation since they do not require the matrix decomposition methods of the maximum SSNR method. All divide-and-conquer methods yield SSNR that is comparable to maximum SSNR method.

    - I develop the divide-and-conquer minimization TEQ method to minimize the $\frac{1}{\text{SSNR}}$ objective function. The closed-form solution for each two-tap filter is the same for unit tap and unit norm constraints.

  - I develop the divide-and-conquer cancellation method that directly uses the channel impulse response to cancel the energy outside the window. This method has lower computation cost than the divide-and-conquer minimization method for either unit tap or unit norm constraints.

- I use the matrix pencil method to estimate the poles of an IIR channel model so that the zeros of a TEQ can be set to the locations of poles in order to cancel the tail of the channel impulse response. However, the matrix pencil method can be used not only in the case of a known channel, but also in the case of an unknown channel. Matrix pencil methods can estimate the number of poles directly from the received signal.

6.3 Future research

Although the work presented in this dissertation makes several contributions in the area of channel equalizers in wireline communication systems, many challenges remain.

The simulation result shows that both MLP and RBF equalizers may not be sensitive to channel phase response, which may be an interesting research topic to explore neural network equalizers. However, the reduction of computational complexity is the most important for neural network equalizers to be used in practice. Although neural networks can learn features of input data, training in neural networks is a serious problem that prevents neural networks from having practical real-time implementations when on-line training is required, such as in communication systems. Fast training algorithms for neural networks could be developed. One possible solution is a deeper understanding of how the human brain learns and generalizes.

Computational complexity is also an important issue for matrix pencil methods. Matrix pencil methods have the advantage of estimating the poles of a channel. However, the original matrix pencil method requires the implementation of singular value decomposition and eigenvalue decomposition, and the modified matrix pencil methods require an iterative process that applies the reduced-rank Hankel approximation within the original matrix pencil method. Parallel computation might be able to lower the computation cost of matrix pencil methods. However, implementing the matrix pencil methods in real time remains an open research topic.

Since the mid 1980s, multicarrier systems have provided data transmission over bandlimited channels. Their immunity to nonflat frequency response channels and the inherent parallelism in the transceiver designs make these systems a competitive and attractive candidate in high-speed data communication. However, this research field is far from being mature since a new wireline and wireless standard using multicarrier modulation has been approved every year since 1997, and dozens of multicarrier standards are being debated.

I have not addressed the problem of maximizing channel capacity for TEQ designs. Joint optimization methods to optimize a TEQ with other blocks in multicarrier transceivers (e.g. the FEQ) could increase the data transmission rate for the same SER. As I mention in Section 2.6, FEQ can offer better channel capacity over a TEQ [91]. Joint TEQ-FEQ FEQ design could be a direction of future research in multicarrier systems.

The TEQ design methods proposed in this dissertation are based on the assumption that the variation of channel impulse response with time is too small and can be neglected. When applying TEQ design techniques to time-varying dynamic fading channels, one may encounter complicated problems in channel estimation. Although matrix pencil methods can estimate the channel without the knowledge of training sequences, the intensive computation cost makes matrix pencil methods hard to track the variations in real time. It would be desirable to modify current TEQ design methods for time-varying channels.

Appendix

Appendix A

Proof that $J_{2,i}$ in (5.6) is not ill-conditioned

I prove that the denominator of $J_{2,i}$ in (5.6) is not zero for all g so that $J_{2,i}$ is not ill-conditioned. In this proof, I want to show

$$b_{1,i} + 2b_{2,i}g + b_{3,i}g^2 \neq 0, \text{ for } \forall g \in \Re$$

so that $J_{2,i}$ in (5.6) is well-defined. I write $\mathbf{H}_{i,\text{win}}$ and $\mathbf{H}_{i,\text{wall}}$ according to (2.31) and (2.32):

$$\mathbf{H}_{i,\text{win}} = \begin{bmatrix} h_i(\Delta+1) & h_i(\Delta) \\ h_i(\Delta+2) & h_i(\Delta+1) \\ \vdots & \vdots \\ h_i(\Delta+\nu+1) & h_i(\Delta+\nu) \end{bmatrix}_{(v+1)\times 2}$$

and

$$\mathbf{H}_{i,\text{wall}} = \begin{bmatrix} h_i(1) & h_i(0) \\ h_i(2) & h_i(1) \\ \vdots & \vdots \\ h_i(\Delta) & h_i(\Delta-1) \\ h_i(\Delta+\nu+2) & h_i(\Delta+\nu+1) \\ \vdots & \vdots \\ h_i(L_{h_i}) & h_i(L_{h_i}-1) \\ h_i(L_{h_i}+1) & h_i(L_{h_i}) \end{bmatrix}_{(L_{h_i}+1)\times 2}$$

where L_{h_i} is the channel impulse response at iteration i. Note that $h_i(0) = h_i(L_{h_i}+1) = 0$. Then, I can write matrices $\mathbf{A}_i$ and $\mathbf{B}_i$ to be explicitly related to $h_i(k),\ k = 1, 2, \ldots, (L_{h_i} + 1)$:

$$\begin{aligned}
\mathbf{A}_i = \mathbf{H}_{i,\text{win}}^T \mathbf{H}_{i,\text{win}} &= \begin{bmatrix} a_{1,i} & a_{2,i} \\ a_{2,i} & a_{3,i} \end{bmatrix} \\
&= \begin{bmatrix} \sum\limits_{k=\Delta+1}^{\Delta+\nu+1} h_i^2(k) & \sum\limits_{k=\Delta+1}^{\Delta+\nu+1} h_i(k)h_i(k-1) \\ \sum\limits_{k=\Delta+1}^{\Delta+\nu+1} h_i(k)h_i(k-1) & \sum\limits_{k=\Delta+1}^{\Delta+\nu+1} h_i^2(k-1) \end{bmatrix} \\
\mathbf{B}_i = \mathbf{H}_{i,\text{wall}}^T \mathbf{H}_{i,\text{wall}} &= \begin{bmatrix} b_{1,i} & b_{2,i} \\ b_{2,i} & b_{3,i} \end{bmatrix} \\
&= \begin{bmatrix} \sum\limits_{k \in S} h_i^2(k) & \sum\limits_{k \in S} h_i(k)h_i(k-1) \\ \sum\limits_{k \in S} h_i(k)h_i(k-1) & \sum\limits_{k \in S} h_i^2(k-1) \end{bmatrix}
\end{aligned}$$

where $S = \{1, 2, \ldots, \Delta, \Delta + \nu + 2, \ldots, L_{h_i+1}\}$. Thus,

$$\begin{aligned}
a_{1,i} &= \sum_{k=\Delta+1}^{\Delta+\nu+1} h_i^2(k) \\
a_{2,i} &= \sum_{k=\Delta+1}^{\Delta+\nu+1} h_i(k)h_i(k-1) \\
a_{3,i} &= \sum_{k=\Delta+1}^{\Delta+\nu+1} h_i^2(k-1) = \sum_{k=\Delta}^{\Delta+\nu} h_i^2(k) = \sum_{k=\Delta+1}^{\Delta+\nu+1} h_i^2(k) - h_i^2(\Delta+\nu+1) + h_i^2(\Delta) \\
&= a_{1,i} - h_i^2(\Delta+\nu+1) + h_i^2(\Delta), \\
&\quad \text{or,}\ \ a_{1,i} = a_{3,i} + h_i^2(\Delta+\nu+1) - h_i^2(\Delta)
\end{aligned}$$

$$\begin{aligned}
b_{1,i} &= \sum_{k \in S} h_i^2(k) \\
b_{2,i} &= \sum_{k \in S} h_i(k)h_i(k-1) \\
b_{3,i} &= \sum_{k \in S} h_i^2(k-1) = b_{1,i} + h_i^2(\Delta+\nu+1) - h_i^2(\Delta)
\end{aligned}$$

There are two steps to prove that $b_{1,i} + 2b_{2,i}g + b_{3,i}g^2 \neq 0,$ for $\forall g$.

- Step 1: $b_{1,i} \neq 0$

As we can see from above,

$$b_{1,i} = \sum_{k \in S} h_i^2(k) \geq 0$$

The equality holds when $h_i(k) = 0, \text{ for } \forall k \in S$, which is the ideal case that all the samples outside $(\nu + 1)$ samples of effective channel impulse response in ith iteration are zero. In general, $h_i(k) \neq 0, \text{ for } \forall k \in S$. Thus, $b_{1,i} \neq 0$ holds.

- Step 2: For $b_{1,i} > 0$, if

$$(2b_{2,i})^2 - 4b_{1,i}b_{3,i} < 0$$

then, $b_{1,i} + 2b_{2,i}g + b_{3,i}g^2 \neq 0, \text{ for } \forall g \in \Re$. Now, the task is to show that

$$\begin{aligned}
b_{2,i}^2 - b_{1,i}b_{3,i} &= \left(\sum_{k \in S} h_i(k)h_i(k-1)\right)^2 \\
&\quad - \left(\sum_{k \in S} h_i^2(k)\right)\left(\sum_{k \in S} h_i^2(k) + h_i^2(\Delta + \nu + 1) - h_i^2(\Delta)\right) \\
&< 0
\end{aligned}$$

where

$$\begin{aligned}
\sum_{k \in S} h_i^2(k) &= \frac{1}{2}h_i^2(1) + \left(\frac{1}{2}h_i^2(1) + \frac{1}{2}h_i^2(2)\right) + \left(\frac{1}{2}h_i^2(2) + \frac{1}{2}h_i^2(3)\right) \\
&\quad + \cdots + \left(\frac{1}{2}h_i^2(\Delta - 1) + \frac{1}{2}h_i^2(\Delta)\right) + \frac{1}{2}h_i^2(\Delta) \\
&\quad - \frac{1}{2}h_i^2(\Delta + \nu + 1) + \left(\frac{1}{2}h_i^2(\Delta + \nu + 1) + \frac{1}{2}h_i^2(\Delta + \nu + 2)\right) \\
&\quad + \cdots + \left(\frac{1}{2}h_i^2(L_{h_i}) + \frac{1}{2}h_i^2(L_{h_i} + 1)\right) + \frac{1}{2}h_i^2(L_{h_i} + 1) \\
&\quad \text{because of the inequality } a^2 + b^2 \geq 2ab \\
&\geq \frac{1}{2}h_i^2(1) + h_i(1)h_i(2) + h_i(2)h_i(3) + \cdots + h_i(L_{h_i} - 1)h_i(L_{h_i})
\end{aligned}$$

$$+\frac{1}{2}h_i^2(\Delta) - \frac{1}{2}h_i^2(\Delta+\nu+1) + \frac{1}{2}h_i^2(L_{h_i}+1)$$

" $=$" holds when any two consecutive samples are equal,

which is in general not true for channel impulse response.

$$= \sum_{k\in S} h_i(k)h_i(k-1) + \frac{1}{2}h_i(1)$$

$$+\frac{1}{2}h_i^2(\Delta) - \frac{1}{2}h_i^2(\Delta+\nu+1)$$

Thus,

$$
\begin{aligned}
& b_{2,i}^2 - b_{1,i} b_{3,i} \\
= \quad & \left(\sum_{k \in S} h_i(k) h_i(k-1) \right)^2 - \left(\sum_{k \in S} h_i^2(k) \right)^2 \\
& - \left(\sum_{k \in S} h_i^2(k) \right) \left(h_i^2(\Delta + \nu + 1) - h_i^2(\Delta) \right) \\
= \quad & \left(\sum_{k \in S} h_i(k) h_i(k-1) + \sum_{k \in S} h_i^2(k) \right) \left(\sum_{k \in S} h_i(k) h_i(k-1) - \sum_{k \in S} h_i^2(k) \right) \\
& - \left(\sum_{k \in S} h_i^2(k) \right) \left(h_i^2(\Delta + \nu + 1) - h_i^2(\Delta) \right) \\
< \quad & 2 \left(\sum_{k \in S} h_i^2(k) \right) \left(\frac{1}{2} h_i^2(\Delta + \nu + 1) - \frac{1}{2} h_i^2(1) - \frac{1}{2} h_i^2(\Delta) \right) \\
& - \left(\sum_{k \in S} h_i^2(k) \right) \left(h_i^2(\Delta + \nu + 1) - h_i^2(\Delta) \right) \\
= \quad & \left(\sum_{k \in S} h_i^2(k) \right) \times \\
& \left(h_i^2(\Delta + \nu + 1) - h_i^2(1) - h_i^2(\Delta) - h_i^2(\Delta + \nu + 1) + h_i^2(\Delta) \right) \\
= \quad & -h_i^2(1) \left(\sum_{k \in S} h_i^2(k) \right) \\
< \quad & 0
\end{aligned}
$$

Appendix B

Proof of unique solutions to (5.8)

We show that the solutions to (5.8) are unique in two steps:

- Step 1: $a_{3,i}b_{2,i} - a_{2,i}b_{3,i} \neq 0$, since minimizing J_2 in (5.5) is equivalent to maximizing J_1 in (5.3), the only requirement for $a_{3,i}b_{2,i} - a_{2,i}b_{3,i}$ is not equal to zero.

$$
\begin{aligned}
& a_{3,i}b_{2,i} - a_{2,i}b_{3,i} \\
= & \left[\sum_{k=\Delta+1}^{\Delta+\nu+1} h_i^2(k-1)\right]\left[\sum_{k\in S} h_i(k-1)h_i(k)\right] \\
& - \left[\sum_{k=\Delta+1}^{\Delta+\nu+1} h_i(k-1)h_i(k)\right]\left[\sum_{k\in S} h_i^2(k-1)\right] \\
= & \left[\sum_{k=\Delta+1}^{\Delta+\nu+1} h_i^2(k-1)\right]\left[\sum_{k=1}^{L_{h_i}} h_i(k-1)h_i(k) - \sum_{k=\Delta+1}^{\Delta+\nu+1} h_i(k-1)h_i(k)\right] \\
& - \left[\sum_{k=\Delta+1}^{\Delta+\nu+1} h_i(k-1)h_i(k)\right]\left[\sum_{k=1}^{L_{h_i}} h_i^2(k-1) - \sum_{k=\Delta+1}^{\Delta+\nu+1} h_i^2(k-1)\right] \\
= & \left[\sum_{k=\Delta+1}^{\Delta+\nu+1} h_i^2(k-1)\right]\left[\sum_{k=1}^{L_{h_i}} h_i(k-1)h_i(k)\right] \\
& - \left[\sum_{k=\Delta+1}^{\Delta+\nu+1} h_i(k-1)h_i(k)\right]\left[\sum_{k=1}^{L_{h_i}} h_i^2(k-1)\right]
\end{aligned}
$$

$$
\begin{aligned}
& \text{since } \left[\sum_{k=1}^{L_{h_i}} h_i^2(k-1)\right] > \left[\sum_{k=1}^{L_{h_i}} h_i(k-1)h_i(k)\right] \\
< \;& \left[\sum_{k=\Delta+1}^{\Delta+\nu+1} h_i^2(k-1)\right]\left[\sum_{k=1}^{L_{h_i}} h_i(k-1)h_i(k)\right] \\
& - \left[\sum_{k=\Delta+1}^{\Delta+\nu+1} h_i(k-1)h_i(k)\right]\left[\sum_{k=1}^{L_{h_i}} h_i(k-1)h_i(k)\right] \\
= \;& \left[\sum_{k=1}^{L_{h_i}} h_i(k-1)h_i(k)\right]\left(\left[\sum_{k=\Delta+1}^{\Delta+\nu+1} h_i^2(k-1)\right] - \left[\sum_{k=\Delta+1}^{\Delta+\nu+1} h_i(k-1)h_i(k)\right]\right) \\
\neq \;& 0
\end{aligned}
$$

- Step 2: From (5.7), the solutions are unique if

$$(a_{3,i}b_{1,i} - a_{1,i}b_{3,i})^2 - 4(a_{3,i}b_{2,i} - a_{2,i}b_{3,i})(a_{2,i}b_{1,i} - a_{1,i}b_{2,i}) \geq 0$$

Since $a_{1,i} = a_{3,i} + h_i^2(\Delta+\nu+1) - h_i^2(\Delta)$ and $b_{1,i} = b_{3,i}$, we have

$$
\begin{aligned}
& (a_{3,i}b_{1,i} - a_{1,i}b_{3,i})^2 - 4(a_{3,i}b_{2,i} - a_{2,i}b_{3,i})(a_{2,i}b_{1,i} - a_{1,i}b_{2,i}) \\
= \;& b_{1,i}^2(h_i^2(\Delta+\nu+1) - h_i^2(\Delta))^2 - 4(a_{3,i}b_{2,i} - a_{2,i}b_{1,i}) \\
& \times [a_{2,i}b_{1,i} - a_{3,i}b_{2,i} - b_{2,i}(h_i^2(\Delta+\nu+1) - h_i^2(\Delta))] \\
= \;& b_{1,i}^2(h_i^2(\Delta+\nu+1) - h_i^2(\Delta))^2 \\
& +4(a_{3,i}b_{2,i} - a_{2,i}b_{1,i})^2 + 4b_{2,i}(a_{3,i}b_{2,i} - a_{2,i}b_{1,i})(h_i^2(\Delta+\nu+1) - h_i^2(\Delta)) \\
& \text{ since } b_{1,i} \geq b_{2,i} \\
\geq \;& b_{2,i}^2(h_i^2(\Delta+\nu+1) - h_i^2(\Delta))^2 \\
& +4(a_{3,i}b_{2,i} - a_{2,i}b_{1,i})^2 + 4b_{2,i}(a_{3,i}b_{2,i} - a_{2,i}b_{1,i})(h_i^2(\Delta+\nu+1) - h_i^2(\Delta)) \\
= \;& [b_{2,i}(h_i^2(\Delta+\nu+1) - h_i^2(\Delta)) + 2(a_{3,i}b_{2,i} - a_{2,i}b_{1,i})]^2 \geq 0
\end{aligned}
$$

Bibliography

[1] J. A. C. Bingham, *The Theory and Practice of Modem Design.* John Wiley & Sons, 1988.

[2] A. P. Clark, *Equalizers for Digital Modems.* John Wiley & Sons, 1985.

[3] F. Mazda, ed., *Telecommunications Engineer's Reference Book.* Focal Press, 2nd ed., 1998.

[4] P. Warrier, "DSP solution for voiceband and ADSL modems," Application Resport SPAA005, Texas Instruments, 1998.

[5] W. Goralski, ed., *ADSL and DSL Technologies.* McGraw-Hill, 1998.

[6] "ANSI T1.413-1995, Network and customer installation interfaces: Asymmetrical digital subscriber line (ADSL) metallic interface." Printed from: *Digital Subscriber Line Technology* by T. Starr, J. M. Cioffi, and P. J. Silverman, Prentice Hall, 1999.

[7] M. J. Riezenman, "Bringing home the Internet," *IEEE Spectrum*, Mar. 2000.

[8] A. Azzam, ed., *High-Speed Cable Modem.* McGraw-Hill, 1997.

[9] J. D. Gibson, *The Mobile Communications Handbook.* CRC Press, 2nd ed., 1999.

[10] T. Wolf, C. Lemonds, and A. Gatherer, "Low complexity equalization for cable modems," in *Proc. IEEE Int. Sym. Circ. Syst.*, pp. 522–524, May 1998.

[11] P. J. W. Melsa, R. C. Younce, and C. E. Rohrs, "Impulse response shortening for discrete multitone transceivers," *IEEE Trans. Comm.*, vol. 44, pp. 1662–1672, Dec. 1996.

[12] C. M. Bishop, *Neural Networks for Pattern Classification.* New York: Oxford University Press, 1995.

[13] Y. Hua and T. K. Sarkar, "Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 38, pp. 814–824, May 1990.

[14] J. A. Cadzow, Y. Sun, and G. Xu, "Detection of multiple sinusoids in white noise: A signal enhancement approach," in *SVD and Signal Processing* (E. F. Deprettere, ed.), pp. 171–186, North-Holland: Elsevier, 1988.

[15] Y. Li, K. J. R. Liu, and J. Razavilar, "A parameter estimation scheme for damped sinusoidal signals based on low-rank Hankel approximation," *IEEE Trans. Signal Processing*, vol. 45, pp. 481–486, Feb. 1997.

[16] D. D. Falconer and F. R. Magee, "Adaptive channel memory truncation for maximum likelihood sequence estimation," *The Bell System Technical Journal*, vol. 52, pp. 1541–1562, Nov. 1973.

[17] J. S. Chow and J. M. Cioffi, "A cost-effective maximum likelihood receiver for multicarrier systems," in *Proc. IEEE Int. Conf. on Comm.*, pp. 948–952, June 1992.

[18] D. Wei, “Personal correspondence.” June 1997.

[19] L. D. Clark, “Personal correspondence.” June 1999.

[20] G. Arslan, “Personal correspondence.” Mar. 2000.

[21] C. Tellambura, Y. J. Guo, and S. K. Barton, “Channel estimation using aperiodic binary sequence,” *IEEE Comm. Letters*, vol. 2, pp. 140–142, May 1998.

[22] S. N. Crozier, D. D. Falconer, and S. A. Mahmoud, “Least sum of squared error (LSSE) channel estimation,” *Proc. of the IEE*, vol. 138, pp. 371–378, Aug. 1991.

[23] C. Tellambura, M. G. Parker, Y. J. Guo, S. J. Shepherd, and S. K. Barton, “Optimal sequence for channel estimation using discrete Fourier transform techniques,” *IEEE Trans. Comm.*, vol. 47, pp. 230–238, Feb. 1999.

[24] D. C. Lay, *Linear Algebra and Its Applications.* Addison-Wesley Publishing Company, 1994.

[25] W. Chen and U. Mitra, “Frequency domain versus time domain based training sequence optimization,” in *Proc. IEEE Int. Conf. Comm.*, pp. 646–650, June 2000.

[26] M. Barton and D. W. Tufts, “A suboptimum linear receiver derived from SVD based channel estimates,” in *Proc. IEEE Int. Conf. Comm.*, pp. 831–835, May 1989.

[27] E. Lindskog and C. Tidestav, “Reduced rank channel estimation,” in *Proc. IEEE Conf. on Vehicular Technology*, pp. 1126–1130, May 1999.

[28] X. Wang and K. J. R. Liu, “Joint channel estimation and equalization in multicarrier modulation system using cyclic prefix,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 2733–2736, Mar. 1999.

[29] R. Kumaresan and D. W. Tufts, “Estimating the parameters of exponentially damped sinusoids and pole-zero modeling in noise,” *IEEE Trans. Acoust. Speech Signal Processing*, vol. 30, pp. 833–840, Dec. 1982.

[30] B. Porat and B. Friedlanser, “A modification of the Kumaresan-Tufts method for estimating rational impulse responses,” *IEEE Trans. Acoust. Speech Signal Processing*, vol. 34, pp. 1336–1338, Oct. 1986.

[31] J. Razavilar, Y. Li, and K. J. R. Liu, “Spectral estimation based on structured low rank matrix pencil,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, vol. 5, pp. 2503–2506, May 1996.

[32] J. Razavilar, Y. Li, and K. J. R. Liu, “Structured low-rank matrix pencil for spectral estimation and system identification,” *Signal Processing*, vol. 65, pp. 363–372, Mar. 1998.

[33] A.-J. van der Veen and E. F. Deprettere, “Parallel VLSI matrix pencil algorithm for high resolution direction finding,” *IEEE Trans. Signal Processing*, vol. 39, pp. 383–394, Feb. 1991.

[34] J. E. F. del Rio and T. K. Sarkar, “Comparison between the matrix pencil method and the Fourier transform technique for high-resolution spectral estimation,” *Digital Signal Processing*, vol. 6, pp. 108–125, Apr. 1996.

[35] C. Christopoulos, *The Transmission-Line Modeling Method.* Oxford University Press, 1995.

[36] N. Al-Dhahir, A. H. Sayed, and J. M. Cioffi, “Stable pole-zero modeling of long FIR filters with applications to the MMSE-DFE,” *IEEE Trans. Signal Processing*, vol. 45, pp. 508–513, May 1997.

[37] Y. Hua and T. K. Sarkar, “Subspace linear prediction approach to extracting poles,” in *Proc. IEEE Workshop on Spectrum Estimation and Modeling*, (Minneapolis, MN), pp. 367–370, Aug. 1988.

[38] Y. Hua, “Parameter estimation of exponentially damped sinusoids using higher order statistics and matrix pencil,” *IEEE Trans. Signal Processing*, vol. 39, pp. 1691–1692, July 1991.

[39] Y. Hua and T. K. Sarkar, “On SVD for estimating generalized eigenvalues of singular matrix pencil in noise,” *IEEE Trans. Signal Processing*, vol. 39, pp. 892–900, Apr. 1991.

[40] J.-J. Fuchs, “Estimating the number of sinusoids in additive white noise,” *IEEE Trans. Acoust. Speech Signal Processing*, vol. 36, pp. 1846–1853, Dec. 1988.

[41] M. S. Bartlett, “A note on the multiplying factors for various χ^2 approximations,” *J. Roy. Stat. Soc., Ser B*, vol. 16, pp. 296–298, 1954.

[42] D. N. Lawley, “Tests of significance of the latent roots of the covariance and correlation matrices,” *Biometrica*, vol. 43, pp. 128–136, 1956.

[43] V. U. Reddy and L. S. Biradar, “SVD-based information theoretic criteria for detection of the number of damped/undamped sinusoids and their performance analysis,” *IEEE Trans. Signal Processing*, vol. 41, pp. 2872–2881, Sept. 1993.

[44] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Automatic Control*, vol. 19, pp. 716–723, Dec. 1974.

[45] G. Schwartz, "Estimating the dimension of a model," *Ann. Stat.*, vol. 6, no. 2, pp. 461–464, 1978.

[46] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, pp. 465–471, 1978.

[47] M. Wax and T. Kailath, "Detection of signals by information theoretic criteria," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 33, pp. 387–392, Apr. 1985.

[48] L. Biradar and V. U. Reddy, "SVD based information theoretic criteria for detection of the number of sinusoids," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, vol. 5, pp. 2519–2522, Apr. 1990.

[49] A. Sano and H. Tsuji, "Optimal determination of number of sinusoids in SVD-based spectral estimation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, vol. 5, pp. 2563–2566, Apr. 1990.

[50] L. Fausett, *Fundamental of Neural Networks: Architectures, Algorithms, and Applications.* Englewood Cliffs, NJ: Prentice Hall, 1994.

[51] Y. Pao, *Pattern Recognition and Neural Networks.* Reading, MA: Addison-Wesley, 1989.

[52] C. G. Looney, *Pattern Recognition Using Neural Networks: Theory and Algorithms for Engineers and Scientists.* Oxford University Press, 1997.

[53] S. McLoone, M. D. Brown, G. Irwin, and G. Lightbody, "A hybrid linear/nonlinear training algorithm for feedforward neural networks," *IEEE Trans. Neural Networks*, vol. 9, pp. 669–684, July 1997.

[54] S. C. Ng, S. H. Leung, and A. Luk, "The generalized back-propagation algorithm with convergence analysis," in *Proc. IEEE Int. Sym. Circ. Syst.*, (Orlando, FL), pp. 612 –615, 1999.

[55] S. C. Ng, S. H. Leung, and A. Luk, "A generalized back-propagation algorithm for fast convergence," in *Proc. IEEE Int. Conf. Neural Networks*, (Washington, DC), pp. 409–413, 1996.

[56] Z. Nehari, *Introduction to Complex Analysis.* Boston, MA: Allyn and Bacon, 1961.

[57] H. Leung and S. Haykin, "The complex backpropagation algorithm," *IEEE Trans. Signal Processing*, vol. 39, pp. 2101–2104, Sept. 1991.

[58] G. M. Georgiou and C. Koutsougeras, "Complex domain backpropagation," *IEEE Trans. Circuits Syst. II*, vol. 39, pp. 330–334, May 1992.

[59] N. Benvenuto and F. Piazza, "On the complex backpropagation algorithm," *IEEE Trans. Signal Processing*, vol. 40, pp. 967–969, Apr. 1992.

[60] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive network," *Complex Systems*, vol. 2, pp. 321–355, 1988.

[61] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Networks*, vol. 2, pp. 302–309, Mar. 1991.

[62] G. J. Gibson, S. Siu, and C. F. N. Cowan, "The application of nonlinear structures to the reconstruction of binary signals," *IEEE Trans. Signal Processing*, vol. 39, pp. 1877–1884, Aug. 1991.

[63] S. Siu, G. J. Gibson, and C. F. N. Cowan, "Decision feedback equalisation using neural network structures and performance comparison with standard architecture," *IEE Proc., Part I*, vol. 137, pp. 221–225, Aug. 1990.

[64] B. Mulgrew, "Applying radial basis functions," *IEEE Signal Proc. Magazine*, vol. 13, pp. 50–65, Mar. 1996.

[65] S. Theodordis, C. F. N. Cowan, C. P. Callender, and C. M. S. See, "Schemes for equalisation for communication channels with nonlinear impairments," *IEE Proc.- Comm.*, vol. 142, pp. 165–171, June 1995.

[66] S. Chen, G. J. Gibson, C. F. N. Cowan, and P. M. Grant, "Adaptive equalization of finite non-linear channels using multilayer perceptrons," *Signal Processing*, vol. 20, pp. 107–119, 1990.

[67] G. J. Gibson, S. Siu, and C. F. N. Cowan, "Multilayer perceptron structures applied to adaptive equalizer for data communications," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Proc.*, vol. II, (Glasgow, Scotland), pp. 1183–1186, May 1989.

[68] M. Peng, C. L. Nikias, and J. Proakis, "Adaptive equalization with neural networks: new multilayer perceptron structure and their evaluation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Proc.*, vol. II, (San Francisco, CA), pp. 301–304, Mar. 1992.

[69] M. Peng, C. L. Nikias, and J. G. Proakis, "Adaptive equalization for PAM and QAM signals with neural networks," in *Proc. IEEE Asilomar Conf. Signals, Systems, and Comp.*, (Pacific Grove, CA), pp. 496–500, Nov. 1991.

[70] M. Peng, H. Lev-Ari, C. L. Nikias, and J. G. Proakis, "Performance improvement of neural network equalizers," in *Proc. IEEE Asilomar Conf. Signals, Systems, and Comp.*, (Pacific Grove, CA), pp. 396–400, Nov. 1993.

[71] M. Bianchina and M. Gori, "Optimal learning in artificial neural networks: A review of theoretical results," *Neurocomputing*, vol. 13, no. 6, pp. 313–346, 1996.

[72] S. Chen, G. J. Gibson, C. F. N. Cowan, and P. M. Grant, "Reconstruction of binary signals using an adaptive radial-basis-function equalizer," *Signal Processing*, vol. 22, pp. 77–93, 1991.

[73] C. You and D. Hong, "Adaptive equalization using the complex backpropagation algorithm," in *Proc. IEEE Int. Conf. Neural Networks*, (Washington, DC), pp. 2136–2141, June 1996.

[74] C. W. You and D. S. Hong, "Nonlinear blind equalization schemes using complex-valued multilayer feedforward neural networks," *IEEE Trans. Neural Networks*, vol. 9, pp. 1442–1455, Nov. 1998.

[75] S. Chen, S. McLaughlin, and B. Mulgrew, "Complex-valued radial basis function network, Part I: network architecture and learning algorithm," *Signal Processing*, vol. 35, pp. 19–31, 1994.

[76] S. Chen, S. McLaughlin, and B. Mulgrew, “Complex-valued radial basis function network, Part II: application to digital communications channel equalization,” *Signal Processing*, vol. 36, pp. 175–188, 1994.

[77] K. S. Jacobsen, P. S. Chow, and J. M. Cioffi, “Discrete multitone modulation for transmission on the cable television network,” in *IEEE 802.catv Committee Contribution Number 94-024*, (Orlando, FL), July 1994.

[78] J. A. C. Bingham, “Multicarrier modulation for data transmission: an idea whose time has come,” *IEEE Personal Comm. Magazine*, vol. 28, pp. 5–14, May 1990.

[79] K. S. Jacobsen, P. S. Chow, and J. M. Cioffi, “Synchronized DMT for multipoint-to-point communications on HFC networks,” in *IEEE Global Comm. Conf.*, (Singapore), pp. 963–966, Nov. 1995.

[80] N. Al-Dhahir and J. M. Cioffi, “Efficiently computed reduced-parameter input-aided MMSE equalizers for ML detection,” *IEEE Trans. Inform. Theory*, vol. 42, pp. 903–915, May 1996.

[81] J. S. Chow, J. M. Cioffi, and A. C. Bingham, “Equalizer training algorithms for multicarrier modulation systems,” in *Proc. IEEE Int. Conf. on Comm.*, vol. 3, pp. 761–765, May 1993.

[82] N. Al-Dhahir and J. M. Cioffi, “Optimum finite-length equalization for multicarrier transceivers,” *IEEE Trans. Comm.*, vol. 44, pp. 56–64, Jan. 1996.

[83] M. Nafie and A. Gatherer, “Time-domain equalizer training for ADSL,” in *Proc. IEEE Int. Conf. on Comm.*, vol. 2, pp. 1085–1089, June 1997.

[84] J.-F. V. Kerckhove and P. Spruyt, "Adapted optimization criterion for FDM-based DMT-ADSL equalization," in *Proc. IEEE Int. Conf. on Comm.*, vol. 3, pp. 1328–1334, June 1996.

[85] M. Webster and R. Roberts, "Adaptive channel truncation for FFT detection in DMT system - error component partitioning," in *Proc. IEEE Asilomar Conf. Signals, Systems, and Computers*, vol. 1, pp. 669–673, Nov. 1997.

[86] G. H. Golub and C. F. V. Loan, *Matrix Computation.* The John Hopkins University Press, 3rd ed., 1996.

[87] C. Yin and G. Yue, "Optimal impulse response shortening for discrete multitone transceivers," *Electronics Letters*, vol. 34, pp. 35–36, Jan. 1998.

[88] N. Al-Dhahir and J. M. Cioffi, "A bandwidth-optimized reduced-complexity equalized multicarrier transceiver," *IEEE Trans. Comm.*, vol. 45, pp. 948–956, Aug. 1997.

[89] N. Lashkarian and S. Kiaei, "Optimum equalization of multicarrier systems via projection onto convex sets," in *Proc. IEEE Int. Conf. on Comm.*, pp. 968–972, June 1999.

[90] G. Arslan, B. L. Evans, and S. Kiaei, "Optimum channel shortening for discrete multitone transceivers," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, vol. V, (Istanbul, Turkey), pp. 2965–2968, June 2000.

[91] K. V. Acker, G. Leus, M. Moonen, O. van de Wiel, and T. Pollet, "Per tone equalization for DMT receivers," *IEEE Global Comm. Conf.*, pp. 2311–2315, Dec. 1999.

[92] J. S. Lee, C. D. Beach, and N. Tepedelenlioglu, "Channel equalization using radial basis function network," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Proc.*, vol. III, (Atlanta, GA), pp. 1719–1722, May 1996.

[93] D. W. Ruck, S. K. Rogers, M. Kabrisky, and M. E. Oxley, "The multilayer perceptron as an approximation to a Bayes discriminant function," *IEEE Trans. Neural Networks*, vol. 1, pp. 296–298, Dec. 1990.

[94] S. Bouchired, M. Ibnkahla, D. Roviras, and F. Castanie, "Equalization of satellite mobile communication channels using combined self-organizing maps and RBF networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Proc.*, vol. 6, (Seattle, WA), pp. 3377–3379, May 1998.

[95] J. de Villiers and E. Barnard, "Backpropagation neural networks," *IEEE Trans. Neural Networks*, vol. 4, pp. 136–141, Jan. 1993.

[96] J. G. Proakis and M. Saleshi, *Communication Systems Engineering.* Upper Saddle River, New Jersey: Prentice Hall, 1994.

[97] M. P. Ekstrom, "Dispersion estimation from borehole acoustic arrays using a modified matrix pencil algorithm," in *Proc. IEEE Asilomar Conf. Signals, Systems, and Computers*, vol. 1, pp. 449–453, Nov. 1995.

[98] V. S. Il'kiv, "Weighted matrix pencil method for estimating exponents in noise," *Radioelectronics and Communication Systems*, vol. 38, pp. 30–36, Jan. 1995.

[99] J. A. Cadzow, "Signal enhancement: A composite property mapping algorithm," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 36, pp. 49–62, Jan. 1988.

[100] B. Lu, D. Wei, B. L. Evans, and A. C. Bovik, “Improved matrix pencil methods,” in *Proc. IEEE Asilomar Conf. Signals, Systems, and Computers*, vol. 2, pp. 1433–1437, Nov. 1998.

[101] M. H. Hayes, *Statistical Digital Signal Processing and Modeling.* John Wiley and Sons, 1996.

[102] H. L. V. Trees, *Detection, Estimation, and Modulation Theory.* John Wiley & Sons, 1968.

[103] D. W. Tufts and R. Kumaresan, “Estimation of frequencies of multiple sinusoids: Making linear prediction perform like maximum likelihood,” *Proc. IEEE*, vol. 70, pp. 975–989, Sept. 1982.

[104] J. W. Demmel, *Applied Numerical Linear Algebra.* SIAM, 1997.

[105] J. M. Cioffi, *A Multicarrier Primer.* Amati Communication Corporation and Stanford University, T1E1.4/91-157, Nov. 1991.

[106] A. Wang and J. O. Smith, “On fast FIR filters implemented as tail-canceling IIR filters,” *IEEE Trans. Signal Processing*, vol. 45, pp. 1415–1427, June 1997.

Vita

Biao Lu was born in Beijing, P. R. China, on December 20, 1968, as the daughter of Xiuying Guo and Zhicheng Lu. She graduated from high school in 1987 from Beijing No. 10 School in Beijing, P. R. China. She received the degree of Bachelor of Engineering in Biomedical Engineering from the Capital Institute of Medicine in July 1992 in Beijing, P. R. China, and the Master of Science in Electrical Engineering from The University of Texas at Austin in December 1997. From July 1992 to January 1995, she was a research assistant and teaching assistant in the department of biomedical engineering in Peking Union Medical College in Beijing, P. R. China. She received her PhD EE degree in the area of Telecommunications and Information Systems Engineering, Department of Electrical & Computer Engineering, The University of Texas at Austin, Austin, TX USA in December of 2000.

Permanent address: Fengtai District Yungang, Building No. 9, Apt. mid-7,
Beijing, P. R. China, 100074

This dissertation was typeset with LaTeX[‡] by the author.

[‡] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.