University of Alberta

Integer-Delay Equalization and Interpolation Equalization Techniques for Filterbank-Based DMT Systems

by

Luqing Wang ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

Department of Electrical and Computer Engineering

Edmonton, Alberta
Fall 2002

0-612-81495-5

University of Alberta

Library Release Form

Name of Author: Luqing Wang

Title of Thesis: Integer-Delay Equalization and Interpolation Equalization Techniques for Filterbank-Based DMT Systems

Degree: Master of Science

Year this Degree Granted: 2002

Wang Luqing
Luqing Wang
2nd Floor,
Electrical and Computer Engineering
Research Facility (ECERF)
University of Alberta
Edmonton, AB
Canada, T6G 2V4

Date: Sep. 6, 2002

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Simple Delay Equalization and Interpolation Equalization Techniques for Filterbank-Based DMT Systems** submitted by Luqing Wang in partial fulfillment of the requirements for the degree of **Master of Science**.

B. N. Nowrouzian

Dr. B. Nowrouzian (Supervisor)

I. Filanovsky

Dr. I. Filanovsky

Dr. A. Amirfazli

Date: August 30, 2002

Abstract

This thesis is concerned with the development of three novel equalization techniques for DMT systems employing critically decimated filterbanks, namely, the Integer-Delay Equalization, One-Tap Interpolation Equalization and Multi-Tap Interpolation Equalization techniques. Based on the investigation of various DMT systems and equalization techniques, this thesis focuses on the analysis of the dominant effect of channel delay in DMT systems. The proposed equalization techniques equalize the channel (integer or fractional) delay by combining one or multiple polyphase components of the analysis filter output, leading to high Signal-to-Noise Ratios (SNRs) (for example, about 15 dB higher than the output combiner technique) while requiring a small number of equalizer taps. Tradeoff can be made between various equalization parameters, leading to high computational flexibility. Two suboptimal solutions are also proposed to simplify the equalizer training with a small loss in the system output SNR.

I dedicate this thesis to my wife Wenxin Zhang and both our families. All my successes are based on their endless encouragement and support.

Acknowledgements

I would like to thank my wife Wenxin Zhang and our families. Every time in my life, they are always the source of my power and courage.

During the past two years, Dr. Nowrouzian has given me countless advice, guidance, and encouragement. I would like to appreciate all his supervision.

I would also like to thank Ling Zhao who is currently the Ph.D. candidate in the Department of Computer Science, University of Alberta. When talking with him, I can always obtain some good ideas in computer science and LaTeXwriting.

I would further like to thank all my friends in China. Their friendship gives me so much happiness and I am deeply missing them.

Last, I would like to thank NSERC, Micronet, and Nortel Networks for their financial support.

Contents

List of Figures

List of Tables

Acronyms

A/D Analog-to-Digital Conversion

ADSL Asymmetric Digital Subscriber Line.

AWGN Additive White Gaussian Noise

bps bit per second

CMFB Cosine-Modulated Filterbank

CSA Carrier Serving Area

D/A Digital-to-Analog Conversion

DFT Discrete Fourier Transform

DMT Discrete Multitone

DWMT Discrete Wavelet Multitone

DWT Discrete Wavelet Transform

FDM Frequency Division Multiplexing

FEQ Frequency-Domain Equalizer

FEXT Far-End Crosstalk

FFT Fast Fourier Transform

FIR Finite Impulse Response

FMT Filtered Multitone

FWT Fast Wavelet Transform

ICI Interchannel Interference

IDFT Inverse Discrete Fourier Transform

IDWT Inverse Discrete Wavelet Transform

IFFT Inverse Fast Fourier Transform

IFI Interframe Interference

IFWT Inverse Fast Wavelet Transform

IIR Infinite Impulse Response

ISDN Integrated Services Digital Network

ISI Intersymbol Interference

ITU International Telecommunication Union

LOT Lapped Orthogonal Transform

MCM Multicarrier Modulation

MFB Matched Filter Bound

MMSE Minimum Mean-Squared Error

MSE Mean-Squared Error

NEXT Near-End Crosstalk

OFDM Orthogonal Frequency Division Multiplexing

P/S Parallel-to-Serial Conversion

PAM Pulse-Amplitude Modulation

PCFB Principal Component Filterbank

POTS Plain Old Telephone Services

PR Perfect Reconstruction

PSD Power Spectral Density

PSTN Public Switched Telephone Network

QAM Quadrature Amplitude Modulation

RHS Right-Hand-Side

S/P Serial-to-Parallel Conversion

SIR Signal-to-Interference Ratio

SNR Signal-to-Noise Ratio

SSNR Shortening Signal-to-Noise Ratio

TEQ Time-Domain Equalizer

TIR Target Impulse Response

UEC Unit-Energy Constraint

UTC Unit-Tap Constraint

VDSL Very-high-speed Digital Subscriber Line

xDSL various Digital Subscriber Lines

Notations

- $x(n)$: Lowercase letter represents the time-domain signal or symbol,
- $X(z)$, $X(e^{j\omega})$: Uppercase letter represents the transform-domain (z-domain or Fourier transform domain) signal or symbol,
- $\boldsymbol{X}(n)$: Uppercase bold letter represents a matrix or a vector,
- $\boldsymbol{X}_{r\times c}$: A matrix $\boldsymbol{X}$ with r rows and c columns,
- $x^{(k)}(n)$ or $X^{(k)}(z)$: The k-th polyphase component of $x(n)$ or $X(z)$,
- $x(n) * y(n)$: Linear convolution of $x(n)$ and $y(n)$,
- $x(n) \otimes y(n)$: Circular convolution of $x(n)$ and $y(n)$,
- $(\cdot)_{\downarrow M}$: Decimating $(\cdot)$ by a factor of M,
- $(\cdot)_{\uparrow M}$: Expanding $(\cdot)$ by a factor of M,
- $\text{row}_i\{\mathcal{F}_N\}$: The i-th row of the DFT matrix $\mathcal{F}_N$,
- $(\cdot)^*$: Complex conjugate operation,
- $(\cdot)^T$: Matrix transpose operation,
- $(\cdot)'$: Conjugate transpose operation,
- $(\cdot)^{-1}$: Matrix inverse operation,
- $\boldsymbol{I}$: The unity matrix,
- $\boldsymbol{u}_i$: The i-th unit vector, where the i-th element of $\boldsymbol{u}_i$ is equal to 1 while other elements are 0, i.e. $\boldsymbol{u}_i = \begin{bmatrix}0 & \cdots & 1 & \cdots & 0\end{bmatrix}^T$.

Chapter 1

Introduction

Wireline communication is the most widely used communication technology all over the world. Originally, twisted-pair copper line was the only media for wireline communication. With the increasing requirement of fast Internet access and real-time multimedia communication, more and more new media, such as coaxial cables and optical fibres, were employed in wireline communication. However, the twisted-pair copper line is not obsolete. Various new techniques make the twisted-pair copper line suitable for high speed digital communication. Usually, these techniques are referred to as various Digital Subscriber Lines (xDSL) [14].

An important modulation/demodulation technique family widely used in xDSL applications is the Discrete Multitone (DMT) modulation/demodulation technique which is a multicarrier modulation technique employing multirate filterbanks to do modulation and demodulation. DMT techniques can be used not only in xDSL applications but also in other communication applications such as cable TV communications or even wireless communications [24] [31].

A major issue in DMT systems is the channel equalization. Based on the investigation of various DMT systems and equalization techniques, this thesis focuses on the analysis of the channel delay effect in DMT systems, and proposes three novel equalization techniques.

1.1 Background

Although wireless communication was first used in the world in ancient time when a person communicated with other person by shouting, wireline communication has been the most widely used and the most important communication method since the invention of telephone in 1876. Nowadays, the telephone network is available almost everywhere people live, and is the largest communication network in the world. Originally, it was designed only for voice communication, as called Plain Old Telephone Services (POTS) today. However, the POTS system has also been used for various types of data communications such as facsimile, audio and video communications, and Internet applications. It has partly been, and in the near future, will be totally incorporated into the Internet network with the trend involving the convergence of telecommunications and computing [40].

The increasing requirement of various data communications makes the POTS system necessary to be used for broadband communications. The backbone of the telephone network has been (mostly, if not completely) replaced with optical fiber, which is the highest bit-rate wireline communication media today. However, the so-called "last one mile" of the telephone network, which refers to the telephone line between the end user and the nearest central office, is still using the twisted-pair copper line. It is not economical to upgrade every "last one mile" to the optical fiber. Therefore, the reuse of existing twisted-pair copper line for high speed data communications presents a feasible and relatively cheaper way to solve the "last one mile" problem.

The voice modem is the first commercial device to transfer bitstreams through the twisted-pair copper line. Its speed varies from ordinary 300 bit per second (bps) to today's 56k bps. The main standards of voice modems made by International Telecommunication Union (ITU) are listed[1] in Table 1.1.

Standard	Bit Rate (bps)
V.21	300
V.22	1200
V.22 bis	2400
V.27 ter	4800/2400
V.32	9600
V.32 bis	14400
V.34	33600
V.90	56000 (downstream) 33600 (upstream)
V.92	56000 (downstream) 48000 (upstream)

Table 1.1: Voice modem standards

In POTS system, the frequency band of the twisted-pair copper line used for voice communication (called voice-frequency-band) is from 0 to 4k Hz. The voice-frequency-band now is standardized as a channel of 8k Hz sampling rate and 8 bits/sample, i.e. 64k bps. The voice modem also uses the same voice-frequency-band for data communications (this is

[1] Data are obtained from http://www.itu.int

why people cannot use voice modems at the same time when making phone calls). Because usually 1 bit in each sample is used for error detection, the highest bit-rate of the voice modem is 56k bps [50].

The voice-frequency-band is only 4k Hz wide. However, the whole frequency bandwidth of the twisted-pair copper line that suitable for data communication can be much higher (for example, up to 1.1M Hz for Asymmetric Digital Subscriber Line (ADSL)). Therefore, the frequency spectrum higher than 4k Hz can be exploited for data communications.

The Integrated Services Digital Network (ISDN) [50] is the first technology to utilize the unused bandwidth of the twisted-pair copper line for data communications. It provides several 64k bps bearer channels[2] (B channels) for voice and data communications[3], and one 16k or 64k bps data channel (D channel) for signalling and call set-up information. There are two user-network interfaces of ISDN. One is the Basic Rate Interface (BRI) while the other one is the Primary Rate Interface (PRI). Table 1.2 gives some specifications of these two interfaces for comparison.

User-Network Interface	Channels	Bit Rate (bit/s)
BRI	2B+D (D channel is 16k bps)	144k
PRI (T1 in North America)	23B+D (D channel is 64k bps)	1536k
PRI (E1 in Europe)	30B+D (D channel is 64k bps)	1984k

Table 1.2: ISDN user-network interfaces

Cable Internet is a Internet access service provided by cable TV companies. It uses a cable modem to provide high bit-rate transmission. A splitter is needed in the cable modem to split the digital data from analog television signals. The cable modem can provide 27–56M bps digital transmission [10]. However, usually each communication channel (coaxial cable) between the cable TV company and the end users is shared by a group of cable Internet subscribers. Therefore, the actual transmission bit-rate depends on the number of users sharing the same cable.

In recent years, xDSL technologies have been used for high bit-rate transmission over twisted-pair copper line. The transmission bit-rate depends highly on the distance between the central office and the end user. Therefore, these technologies are used mainly to solve the "last one mile" bottleneck. The downstream[4] bit-rate of a xDSL system is higher than

[2] The definition of a bearer channel is that "an ISDN communication channel that bears or carries voice, circuit, or packet conversations" [15].

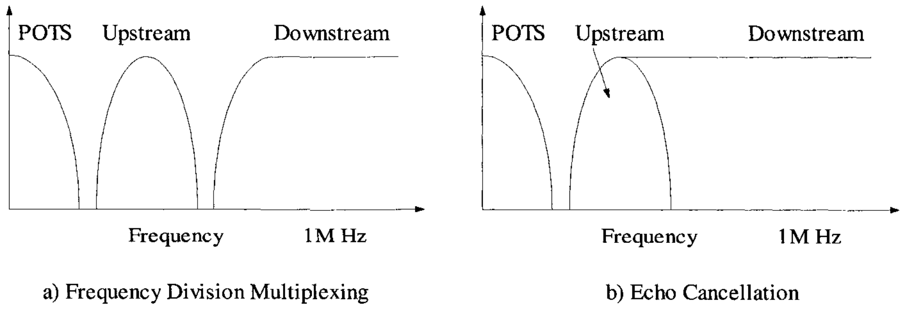
[3] One B channel occupies the voice-frequency-band and can be used for telephone communications if needed.

[4] Downstream and upstream are defined in the next section.

T1 (1.5M bps) and will be up to 52M bps in the future [51]. Because of the star network structure of the Public Switched Telephone Network (PSTN) between the central office and the end users, each communication channel is exclusively used by a single user. Nowadays, with xDSL technologies, the twisted-pair copper line is capable of fast Internet access and multimedia applications such as full motion video and Video-on-Demand.

1.2 Digital Subscriber Lines

xDSL applications use higher frequency band of the twisted-pair copper line for data communications and leave the lower frequency band for use by POTS and/or ISDN. The frequency band used by xDSL applications can be divided into two parts [20], namely, the upstream and downstream channels. The downstream refers to the bitstream transmitted from the central office to the end user while the upstream refers to the bitstream transmitted from the end user to the central office. Because the Internet applications usually transmit much more information in downstream than that in upstream (i.e. for Internet applications, downloading is more dominant than uploading), the downstream usually requires much higher bit-rate than the upstream. An illustration of the frequency band allocation is shown in Figure 1.1. In this figure, two approaches can be used to create the upstream and the downstream signals. The Frequency Division Multiplexing (FDM) approach (Figure 1.1a) assigns different bands to the upstream and the downstream signals. On the other hand, by using Echo Cancellation approach (Figure 1.1b), the upstream band overlaps with the downstream band, and local echo cancellation techniques are needed to separate the upstream and the downstream signals. In both cases, normally a splitter has to be used to separate the POTS/ISDN channels from xDSL channels.



a) Frequency Division Multiplexing

b) Echo Cancellation

Figure 1.1: Frequency band allocation for xDSL applications

There are several types of xDSL techniques. Some of them[5] as shown in Table 1.3 are:

[5] Data are obtained from http://www.dslforum.org

SDSL (Symmetric DSL or Single line DSL), HDSL (High bit-rate DSL), ADSL (Asymmetrical DSL), G.Lite (the simplified ADSL without using splitters), and VDSL (Very-high-speed DSL), where VDSL is still under development.

Name	Bit-rate (bps)	Mode	Media
SDSL	128k–2.32M	Symmetric	1 pair
HDSL	1.544M (T1) 2.3M (E1)	Symmetric	2 pairs (T1) 3 pairs (E1)
ADSL	1.5M–6M (Downstream) 16k–640k (Upstream)	Asymmetric	1 pair
G.Lite	1.5M (Downstream) 500k (Upstream)	Asymmetric	1 pair
VDSL	13M–55M (Downstream) 1.6M–6M (Upstream)	Asymmetric	1 pair

Table 1.3: xDSL techniques

As mentioned above, the bit-rate of a xDSL system varies in accordance with the distance between the central office and the end user. As an example, Table 1.4 shows the relationship between the bit-rate and the distance in ADSL systems.

Bit-rate	Distance (ft)	Distance (km)
1.544 Mbps	18,000	5.5
2.048 Mbps	16,000	4.8
6.312 Mbps	12,000	3.7
8.448 Mbps	9,000	2.7

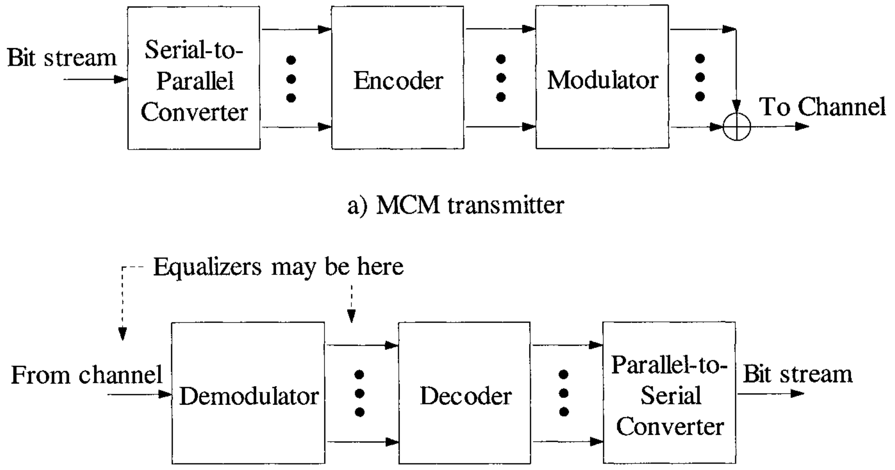
Table 1.4: Relationship between the bit-rate and distance between the central office and the end user for ADSL systems

In xDSL applications, various modulation/demodulation techniques are used to obtain high transmission bit-rates. Most of these techniques are Multicarrier Modulation (MCM) techniques. For example, the conventional DMT technique has been selected by ANSI and ITU as the standard modulation/demodulation technique for ADSL applications [4]. In the next section, some important concepts of MCM techniques are discussed as the foundation for the coming chapters.

1.3 Multicarrier Modulation

1.3.1 Structure of Multicarrier Modulation

MCM technique is a type of orthogonal FDM technique that transmits data "by dividing it into several interleaved bit streams, and using these to modulate several carriers" [7]. It divides the communication channel into several orthogonal subchannels by using a set of carriers, and transmits bitstreams through these subchannels in parallel. The block diagram of general MCM systems is shown in Figure 1.2.



a) MCM transmitter

b) MCM Receiver

Figure 1.2: Block diagram of general MCM systems

At the transmitter, the serial input bitstream is parsed into several subchannels by using a Serial-to-Parallel (S/P) converter. Then, an encoder is used to group the bitstream in each subchannel into bit blocks and encode these bit blocks into symbols (as explained later). After that, the generated symbols are modulated into different subchannels by using a modulator, and eventually summed together and sent through the channel. At the receiver, the received symbols are demodulated by using the demodulator. Equalizers may be used before or after the demodulator to compensate for the channel distortion to obtain the recovered symbols. After demodulation and equalization, the recovered symbols are decoded into parallel bitstreams by using a decoder. Then, after Parallel-to-Serial (P/S) conversion, the serial input bitstream is recovered at the system output.

1.3.2 Communication Channel

An ideal communication channel is a distortionless and noiseless channel such that its output signal is exactly the same as its input signal. In discrete-domain, the impulse response $c(n)$ of an ideal channel can be written as

$$c(n) = \delta(n)\,, \tag{1.1}$$

where $\delta(n)$ is defined as

$$\delta(n) \triangleq \begin{cases} 1\,, & n=0\,, \\ 0\,, & \text{otherwise.} \end{cases} \tag{1.2}$$

However, in practical situations, the channel is non-ideal and may introduce distortion and noise into the transmitted signal. Usually, the channel noise is assumed to be additive[6]. In this case, an actual (noisy) channel can be represented by a noiseless channel[7] $C(z)$ with a noise input $e(n)$, as shown in Figure 1.3.

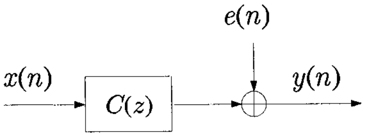


Figure 1.3: Channel model

By assuming that the channel noise is an Additive White Gaussian Noise (AWGN) [28], i.e. additive noise with uniformly distributed Power Spectral Density (PSD), C. E. Shannon [48] gave an upper-bound for the system capacity in bps (called Shannon system capacity) as

$$b = \mathrm{BW}\log_2(1+\mathrm{SNR})\,, \tag{1.3}$$

where BW represents the channel bandwidth, and SNR represents the channel Signal-to-Noise Ratio (SNR). Eqn (1.3) can also be written in bit per symbol as

$$b = \log_2(1+\mathrm{SNR})\,. \tag{1.4}$$

Usually in xDSL systems, the Shannon system capacity cannot be reached because of the existence of multitudes of non-Gaussian noise components in the twisted-pair copper line channel, such as the Near-End Crosstalk (NEXT), Far-End Crosstalk (FEXT) [8], and narrow-band noise (for example, radio frequency interferences).

1.3.3 Receiver Biases and Signal-to-Noise Ratio

In Communication system, the SNR is often used to measure the system performance. Depending on two different receiver settings, there are two types of SNRs: the biased SNR and the unbiased SNR [13]. Figure 1.4 illustrates the two receiver settings.

[6] This is true when the channel input is decorrelated with the channel noise and the power of the channel input is larger than that of the channel noise. When the channel noise is not additive, the communication channel model has to be changed, for example, using the multiplicative noise model or distributive noise model.

[7] $C(z)$ represents the z-transform of the channel impulse response $c(n)$.

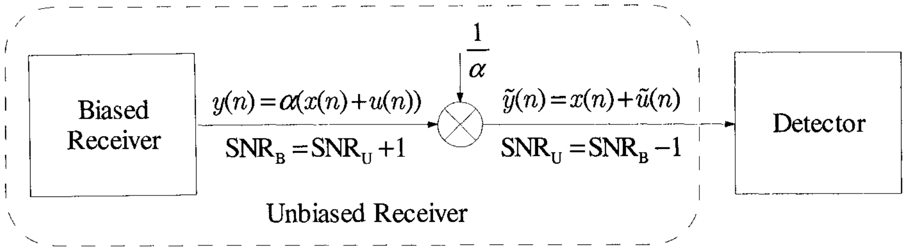

Biased Receiver
$y(n)=\alpha(x(n)+u(n))$
$\mathrm{SNR_B}=\mathrm{SNR_U}+1$
$\frac{1}{\alpha}$
$\tilde{y}(n)=x(n)+\tilde{u}(n)$
$\mathrm{SNR_U}=\mathrm{SNR_B}-1$
Detector
Unbiased Receiver

Figure 1.4: Biased receiver and unbiased receiver

In this figure, the output of the biased receiver is

$$y(n) = \alpha \left(x(n) + u(n)\right) , \tag{1.5}$$

where α is a positive constant, and $u(n)$ are the summation of noise and interference components contained in $y(n)$. This receiver is called as "biased" because the system input component $\alpha x(n)$ contained in the receiver output $y(n)$ is not equal to $x(n)$. In this case, the output SNR is the biased SNR in accordance with

$$\mathrm{SNR_B} \triangleq \frac{E\left[|x(n)|^2\right]}{E\left[|x(n)-y(n)|^2\right]} = \frac{\sigma_x^2}{|\alpha|^2 \sigma_u^2 + |1-\alpha|^2 \sigma_x^2} , \tag{1.6}$$

where σ_x^2 and σ_u^2 represent the autocorrelation of $x(n)$ and $u(n)$, respectively.

On the other hand, the unbiased receiver (the dashed box in Figure 1.4) is the biased receiver cascaded by a scaler $\frac{1}{\alpha}$. The output of the unbiased receiver can be written as

$$\tilde{y}(n) = x(n) + \tilde{u}(n) , \tag{1.7}$$

where $\tilde{u}(n)$ are the summation of noise and interference components contained in $\tilde{y}(n)$. Therefore, the system input component contained in the receiver output $\tilde{y}(n)$ is equal to $x(n)$. In this case, the output SNR is the unbiased SNR in accordance with

$$\mathrm{SNR_U} \triangleq \frac{E\left[|x(n)|^2\right]}{E\left[|x(n)-\tilde{y}(n)|^2\right]} = \frac{\sigma_x^2}{\sigma_{\tilde{u}}^2} , \tag{1.8}$$

where $\sigma_{\tilde{u}}^2$ represents the autocorrelation of $\tilde{u}(n)$. It can be proved [13] that the biased SNR and the unbiased SNR have the relationship as

$$\mathrm{SNR_B} = \mathrm{SNR_U} + 1 . \tag{1.9}$$

1.3.4 PAM/QAM Encoding and Water-Filling Strategy

In the MCM system, the encoder is used to convert binary inputs into symbols. In each subchannel, different number of bits are grouped into blocks and then converted into symbols using Pulse-Amplitude Modulation (PAM) or Quadrature Amplitude Modulation (QAM) encoding. Therefore, each symbol belongs to a fixed finite set of values (or collectively called a constellation) [53]. These values may be real numbers in PAM constellation or complex numbers in QAM constellation. Examples of PAM and QAM constellations are given in Figure 1.5.

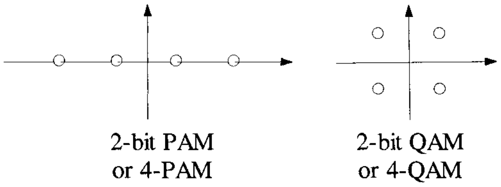


Figure 1.5: Constellation examples

In MCM systems, the number of bits allocated into each subchannel depends on the SNR of the corresponding subchannel. The bit allocation strategy is called as "water-filling strategy" [53]. Suppose the channel $C(z)$ can be fully equalized by using an equalizer $1/C(z)$. Then, the noise PSD at the equalizer output (the effective noise PSD) becomes

$$S_{qq}(e^{j\omega}) = \frac{S_{ee}(e^{j\omega})}{|C(e^{j\omega})|^2}, \tag{1.10}$$

where ω represents the real frequency variable, and $S_{ee}(e^{j\omega})$, $S_{qq}(e^{j\omega})$, and $C(e^{j\omega})$ represent the noise PSD at the equalizer input, the noise PSD at the equalizer output, and the frequency response of the channel, respectively. By using the water-filling strategy, the optimal input power distribution is

$$S_{xx}(e^{j\omega}) = \begin{cases} \lambda - S_{qq}(e^{j\omega}), & \text{when this} \geq 0\,, \\ 0, & \text{otherwise,} \end{cases} \tag{1.11}$$

where λ is a constant. The graphical illustration is shown in Figure 1.6, which looks like a bowl filling of water.

Eqn (1.11) means that in the channel frequency region where $S_{qq}(e^{j\omega}) > \lambda$, no bit should be allocated, in the channel frequency region where $S_{xx}(e^{j\omega}) > 0$ but small, small number of bits should be allocated, and in the channel frequency region where $S_{xx}(e^{j\omega}) > 0$ is large, a large number of bits should be allocated. With this strategy, the system capacity can be maximized.

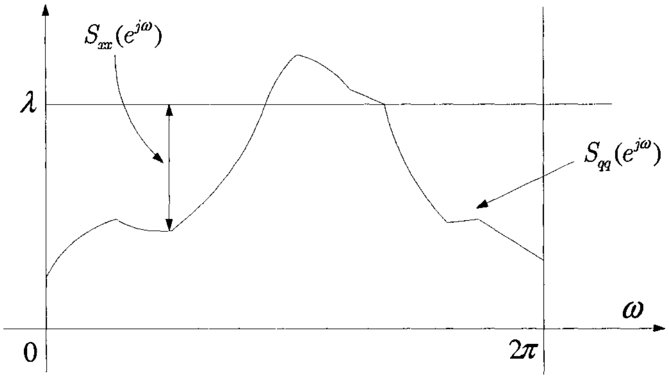


Figure 1.6: Illustration of water-filling strategy

1.3.5 Matched Filter Bound

If the channel noise is AWGN, from Eqns (1.10) and (1.11), one can conclude that the upper-bound of the system capacity is determined by the channel frequency response. In this case, the upper-bound of the system capacity is called as Matched Filter Bound (MFB).

Suppose the number of subchannels in a MCM system is M. When M is large, the frequency response of each subchannel can be approximated by a constant magnitude and a linear phase response. In this case, together with the assumption of AWGN channel noise, the MFB of the i-th subchannel ($i = 0$, ..., $M-1$) is defined as

$$\mathrm{SNR}_{\mathrm{MFB},i} \triangleq \frac{\varepsilon_{x,i} C_i^2}{\varepsilon_e}\,, \tag{1.12}$$

where $\varepsilon_{x,i}$, ε_e, and C_i represent the PSD of the input of the i-th subchannel, the PSD of the noise, and the approximated magnitude response of the i-th subchannel, respectively. Then, the Shannon system capacity becomes

$$\begin{aligned} b = \sum_{m=0}^{M-1} b_m &= \sum_{m=0}^{M-1} \log_2(1 + \mathrm{SNR}_{\mathrm{MFB},i}) \\ &= \sum_{m=0}^{M-1} \log_2\left(1 + \frac{\varepsilon_{x,i} C_i^2}{\varepsilon_e}\right). \end{aligned} \tag{1.13}$$

Usually, the margins of coding gain and SNR gap have to be considered. Then, the Shannon system capacity becomes

$$b = \sum_{m=0}^{M-1} \log_2(1 + 10^{\mathrm{SNR}_{\mathrm{marg.},i}})\,, \tag{1.14}$$

where $\mathrm{SNR}_{\mathrm{marg.},i}$ is conventionally defined in dB as

$$\mathrm{SNR}_{\mathrm{marg.},i}(\mathrm{dB}) \triangleq \mathrm{SNR}_{\mathrm{MFB},i}(\mathrm{dB}) + \gamma_c(\mathrm{dB}) - \gamma_n(\mathrm{dB}) - \Gamma_g(\mathrm{dB})\,, \tag{1.15}$$

with γ_c(dB), γ_n(dB), and Γ_g(dB) representing the coding gain, noise margin, and SNR gap in dB, respectively. Note that by replacing $\mathrm{SNR}_{\mathrm{MFB},i}$(dB) in Eqn (1.15) with the actual SNR of the i-th subchannel, Eqn (1.14) can also be used to calculate the actual system capacity. Moreover, after determining the system capacity b, the transmission bit-rate (in bps) is obtained as

$$\mathrm{BitRate} = \frac{b\,f_s}{M}\,, \tag{1.16}$$

where f_s represents the sampling frequency.

1.4 Problem Statement and Thesis Overview

The MFB discussed above corresponds to the case that only one symbol in each subchannel is transmitted. In practical situations, because the channel is non-ideal, interferences exist in system output when a series of symbols are transmitted[8]. To eliminate the interferences and recover the input symbols at the system output, channel equalization has to be employed to compensate for the channel distortion. In this thesis, various DMT systems and equalization techniques are investigated. Based on the investigation, the thesis mainly focuses on the study of the relationship between the channel phase response and the decimator used in DMT system. In light of the effect of channel delay, this thesis proposes three new equalization methods, namely, integer-delay equalization (resulting in relatively low SNR but having no additional equalization cost, suitable for binary or low bit-rate communication cases), one-tap interpolation equalization (resulting in moderate SNR with low equalization cost, suitable for moderate bit-rate communication cases), and multi-tap interpolation equalization (the generalization of the former two equalization techniques, resulting high SNR with relatively small equalization cost, suitable for high bit-rate communication cases). Moreover, tradeoff between various equalization parameters is allowed in the multi-tap interpolation equalization, leading to high computational flexibility. In addition, a significant simplification in the equalizer training can be achieved through constraining the search for the optimal solution to within the channel lower and upper group-delay bounds within each subchannel.

Chapter 2 is concerned with the investigation of various DMT systems. The general filterbank-based DMT system is introduced together with its properties. The optimal orthonormal DMT system is discussed, which gives a theoretical upper-bound for optimal DMT system design. Four commonly used DMT systems are also discussed, namely, the

[8] A detailed explanation of interferences is given in Section 2.3.3.

conventional DMT, zipper DMT, Discrete Wavelet Multitone (DWMT), and Filtered Multitone (FMT) systems.

Chapter 3 gives an investigation of various equalization techniques. Three equalization techniques that are widely used in various DMT systems are discussed in detail, namely, time-domain equalization, per tone equalization, and the output combiner.

In Chapter 4, the effect of channel phase response and the relationship between the channel delay and decimators are studied. The dominant role of the channel delay in output distortion is placed in evidence. Moreover, in filterbank-based DMT systems, the channel delay exhibits a periodic-like property, and can be represented by its principal-value. When channel delay mismatches the time synchronization of the decimator, symbols that contain important information about the system input are discarded by the decimator. In light of this, the integer-delay equalization technique for DMT systems using critically decimated filterbanks is proposed, which uses one of the M polyphase components of the analysis filter output as the recovered symbols. This equalization technique is suitable for the low bit-rate cases with no additional cost.

In Chapter 5, the study of the effect of channel delay is extended to the non-integer quantity case. In this case, no polyphase components of the analysis filter output can be used as the recovered symbols. However, some polyphase components of the analysis filter output still contain important information that can be used for equalization. With this study, the one-tap interpolation equalization technique for DMT systems using critically decimated filterbanks is proposed, which combines two carefully selected polyphase components of the analysis filter output to recover the system input. By using this equalization technique, moderate system output SNR can be obtained with significantly low equalization cost. In addition, a suboptimal solution is also proposed to simplify the equalizer training with a small loss in the system output SNR.

Chapter 6 proposes the multi-tap interpolation equalization technique as the generalization of the integer-delay equalization and one-tap interpolation equalization techniques. The multi-tap interpolation equalization technique can obtain high system output SNR with a small number of equalizer taps. It also has high computational flexibility because it permits a tradeoff between various equalization parameters. A further simplification for equalizer training is also proposed towards the end of the chapter. This simplification is proven (c.f. Appendix C) to be the generalization of the per tone equalization technique.

Chapter 7 presents general conclusions of this thesis, including the main contributions and the future work.

Chapter 2

Discrete Multitone Modulations

Originally, the name “DMT” was used for the special technique invented by Amati Corp. However, nowadays in the field of wireline communication, it is extended to name any type of multirate filterbank transmultiplexers (as a kind of technique, Amati’s DMT system can be interpreted as a special type of multirate filterbank transmultiplexers, which will be explained later.) Followed by this trend, therefore, in the remainder of this thesis, the name “DMT” is referred to the set of multirate filterbank transmultiplexers and the one invented by Amati Corp. is called “the conventional DMT”.

In this chapter, the conventional DMT system is investigated first. Then, followed by the concepts and investigation of the general multirate filterbank-based DMT systems, three other most important DMT systems will be investigated, which are: zipper DMT, DWMT, and FMT.

2.1 Conventional DMT System

The conventional Discrete Multitone (DMT) modulation/demodulation technique [12] is the standard technique adopted by ANSI and ITU for ADSL [4] applications while it is also widely used in xDSL applications, permitting high bit-rate transmissions over commonly used twisted-pair copper lines. DMT and its variation are currently candidates of the future Very-high-speed Digital Subscriber Line (VDSL) standard.

2.1.1 Modulation and Demodulation Procedure

The conventional DMT system uses the Inverse Discrete Fourier Transform (IDFT) and the Discrete Fourier Transform (DFT) for modulation and demodulation, respectively, to efficiently use the communication channel. It divides the channel into several orthogonal subchannels using IDFT and DFT, and allocates proper number of bits into each subchannel with accordance to the SNR of the corresponding subchannel. Bitstreams allocated to subchannels are transmitted simultaneously, therefore, the variation of the line attenuation can be compensated and the throughput can be maximized. In practical situations, the conventional DMT system is efficiently implemented using Inverse Fast Fourier Transform (IFFT) and Fast Fourier Transform (FFT). This makes it very easy to implement and be the most cost-efficient system among all DMT systems.

The idea of the conventional DMT system is to treat the input symbols as samples in frequency-domain, and multiplex input symbols using IDFT. The block diagram of the conventional DMT transmitter is shown in Figure 2.1, where N usually is an integer power of 2, and D/A represents the Digital-to-Analog (D/A) converter.

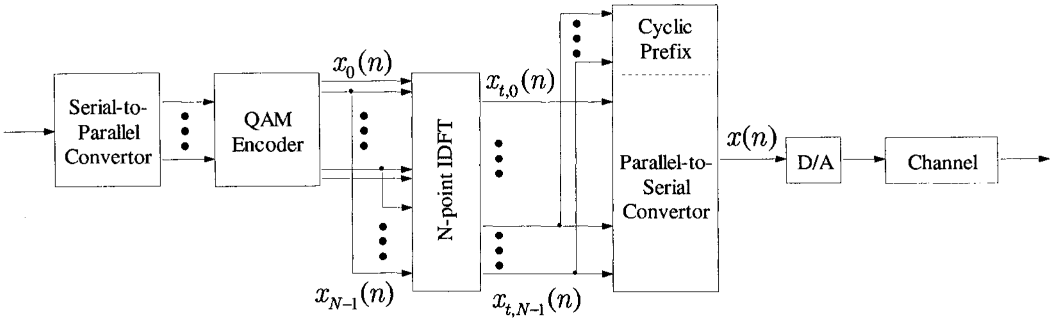


Figure 2.1: Block diagram of the conventional DMT transmitter

At the transmitter side, a S/P converter is used to parse the input bitstream into $N/2+1$ subchannels. The parsing consists of a bit-loading algorithm [30] to allocate a suitable number of bits into each subchannel with accordance to the SNR of the corresponding subchannel. Then, the subchannel bitstreams are encoded to a set of complex symbols[1] $x_0(n)$, ..., $x_{N/2}(n)$ (except $x_0(n)$ and $x_{N/2}(n)$ which must be real) by using a Discrete QAM encoder. These symbols are subsequently extended to a length-N Hermitian symmetric symbol set $x_0(n)$, ..., $x_{N-1}(n)$, such that

$$\begin{cases} x_0(n) \text{ and } x_{N/2}(n) \text{ are real,} \\ x_i(n) = x^*_{N-i}(n), \; i = 1, \ldots, N/2-1 \,, \end{cases} \tag{2.1}$$

where $(\cdot)^*$ represents complex conjugate operation. These N complex symbols are called a symbol frame, and will be transmitted in parallel. After the Hermitian symmetric extension,

[1] In those papers about the conventional DMT system, the QAM-coded symbols are denoted using capital letters ($X_0(n)$, ..., $X_{N-1}(n)$) to highlight the meaning of frequency-domain symbols. However, in this thesis, these symbols are denoted using lowercase letters ($x_0(n)$, ..., $x_{N-1}(n)$) to make them consistent with the notation of the input symbols in the general filterbank-based DMT systems (see Section 2.3), and use $x_{t,0}(n)$, ..., $x_{t,N-1}(n)$ to represent the corresponding time-domain symbols.

each symbol frame is transformed into time-domain by using N-point IDFT in accordance with

$$x_{t,k}(n) = \frac{1}{N}\sum_{i=0}^{N-1} x_i(n)e^{j2\pi ik/N}, \quad k = 0, ..., N-1\,. \tag{2.2}$$

Note that because of the Hermitian symmetric extension in Eqn (2.1), the resulting time-domain symbol frame $\{x_{t,0}(n), ..., x_{t,N-1}(n)\}$ is real. This time-domain symbol frame is then converted to serial by using a P/S converter. Then, a cyclic prefix (will be explained later) is added to the beginning of each time-domain symbol frame. After D/A conversion, symbol frames are sent to the channel.

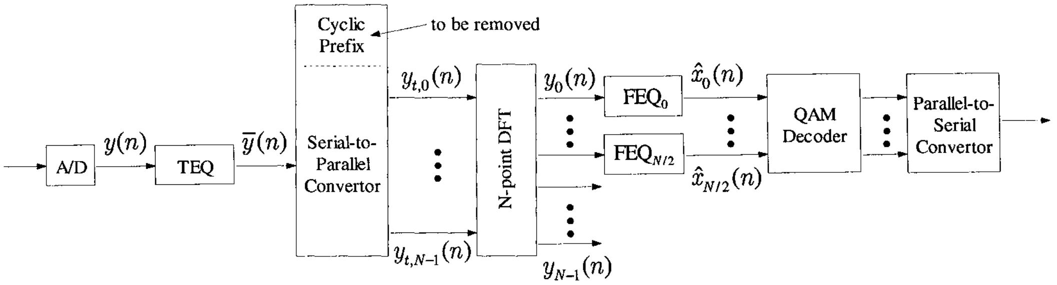


Figure 2.2: Block diagram of the conventional DMT receiver

The conventional DMT receiver is illustrated in Figure 2.2, where A/D represents the Analog-to-Digital (A/D) converter. After A/D conversion, the received time-domain symbols pass through a Time-Domain Equalizer (TEQ) to shorten the channel impulse response. Then, the cyclic prefix is removed from each time-domain symbol frame and the time-domain symbol frames are converted back to the frequency-domain frames using N-point DFT in accordance with

$$y_i(n) = \sum_{k=0}^{N-1} y_{t,k}(n)e^{-j2\pi ik/N}, \quad i = 0, ..., N-1\,. \tag{2.3}$$

Note that only the first $N/2+1$ samples of each frequency-domain frame are used. Subsequently, in each subchannel a one-tap Frequency-Domain Equalizer (FEQ) is used to compensate for the distortion caused by the channel. Finally, symbol frames are QAM-decoded and a parallel-to-serial converter is used to obtain the serial bitstream.

When the channel is ideal and noiseless, IDFT and DFT construct an orthogonal basis, and the input symbols modulated by IDFT operation can be perfectly recovered by DFT operation. However, when the channel is non-ideal, interferences caused by channel distortion will destroy the orthogonal property between IDFT and DFT. In this case, cyclic prefix is used in the conventional DMT system to combat the interferences as discussed in the following.

2.1.2 Cyclic Prefix

Cyclic prefix is an effective method used in Orthogonal Frequency Division Multiplexing (OFDM) [1][8] and the conventional DMT to combat interferences caused by the actual non-ideal channel. In order to know the effect of the cyclic prefix, one has to analyze the interferences in the conventional DMT.

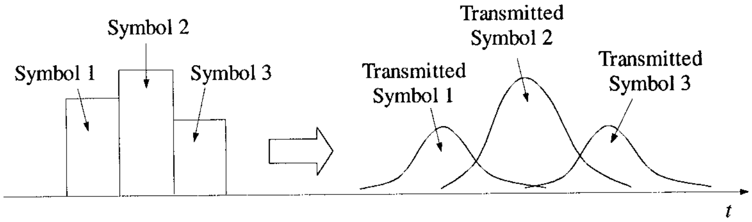


Figure 2.3: Dispersion and overlapping in signal transmission

Interferences can be interpreted as the result of signal dispersing and overlapping in transmission (see Figure 2.3), which happens because of the different transmission speeds of different frequency components of the transmitted signal. This interpretation is usually used in optical fiber communication. On the other hand, interferences can also be interpreted mathematically as follows:

Refer to Figure 2.1, and assume that the channel impulse response is $c(n)$. Then, by ignoring the channel noise, the received symbol $y(n)$ is:

$$\begin{aligned} y(n) = x(n) * c(n) &= \sum_{k=0}^{L_c-1} x(n-k)c(k) \\ &= x(n)c(0) + \sum_{k=1}^{L_c-1} x(n-k)c(k)\,, \end{aligned} \tag{2.4}$$

where $*$ represents linear convolution, L_c is the length of the channel impulse response, and $x(n)$ is the transmitted signal which is generated from converting the parallel symbols $x_{t,i}(n)$ $(i = 0, ..., N-1)$ to serial in accordance with

$$x(Nn+i) = x_{t,i}(n)\,. \tag{2.5}$$

The second item in the Right-Hand-Side (RHS) of Eqn (2.4) represents interference terms in the received signal $y(n)$. If $x(n-k)$ and $x(n)$ belong to the same time-domain symbol frame, the interference from $x(n-k)$ is called Intersymbol Interference (ISI)[2]. On the other

[2] Note that, precisely, the ISI here should be called Inter time-domain Symbol Interference (ITSI), which is different to the ISI in section 2.3, which should be precisely called Inter frequency-domain Symbol Interference (IFSI). ITSI is the composition of IFSI and Interchannel Interference (ICI). In this thesis, the name "ISI" instead of "ITSI" or "IFSI" is used to consist with the convention, and for simplifying notations.

hand, if $x(n-k)$ and $x(n)$ belong to different time-domain symbol frames, the interference from $x(n-k)$ is called Interframe Interference (IFI).

The main idea of the conventional DMT technique is based on the expectation that the following relationship holds (suppose $N \geq L_c$):

$$y_i(n) = x_i(n)C(i), \quad i = 0, \ldots, N-1\,, \tag{2.6}$$

where $x_i(n)$ is the i-th complex input symbol in the n-th frame, $y_i(n)$ is the i-th complex symbol in the n-th frame at the DFT output (see Figure 2.1 and 2.2), and $C(i)$ is the i-th sample of the N-point DFT of the channel impulse response $c(n)$. If Eqn (2.6) holds, then the input symbols can be recovered at the system output:

$$\hat{x}_i(n) = y_i(n)/C(i) = x_i(n), \quad i = 0, \ldots, N/2+1\,, \tag{2.7}$$

and $1/C(i)$ is the coefficient of the one-tap FEQ in the i-th subchannel.

In order to make Eqn (2.6) hold, in time-domain, the n-th output symbol frame $\{y_{t,i}(n)\}$ has to be the circular convolution (with respect to the index i instead of conventionally the index n) of the n-th input symbol frame $\{x_{t,i}(n)\}$ and channel impulse response $\{c(i)\}$. Therefore, given an arbitrary n and noticing that $c(k) = 0$ when $k > L_c - 1$, we expect that

$$\begin{aligned} y_{t,i}(n) &= x_{t,i}(n) \otimes c(i) \\ &= \sum_{k=0}^{i} x_{t,(i-k)}(n)c(k) + \sum_{k=i+1}^{N-1} x_{t,(N+i-k)}(n)c(k)\,, \end{aligned} \tag{2.8}$$

where $\otimes$ represents circular convolution, $i = 0, \ldots, N-1$, and the second item in RHS is equal to zero when $i \geq L_c - 1$.

In practical situations, the channel output is the linear convolution of the channel input and the channel impulse response. Note that $y_{t,i}(n) = y(Nn+i)$ when temporarily ignoring the TEQ. Then, Eqn (2.4) can be rewritten as:

$$\begin{aligned} y_{t,i}(n) &= \sum_{k=0}^{L_c-1} x(nN+i-k)c(k) \\ &= \sum_{k=0}^{i} x_{t,(i-k)}(n)c(k) + \sum_{k=i+1}^{N-1} x_{t,(N+i-k)}(n-1)c(k)\,, \end{aligned} \tag{2.9}$$

where $i = 0, \ldots, N-1$, and the second item in RHS is equal to zero when $i \geq L_c - 1$.

Comparing Eqn (2.9) with Eqn (2.8), one can say that, in time-domain, only the last $N - L_c + 1$ symbols (i.e., when $i = L_c - 1, \ldots, N-1$) of each output symbol frame can be presented as the circular convolution of the corresponding input symbol frame and the channel impulse response. In order to make the first $L_c - 1$ symbols in an output frame also be the circular convolution of the corresponding input symbol frame and the channel impulse response, one can insert a cyclic prefix, which is the last L_v ($L_v \geq L_c - 1$) symbols of an time-domain input frame, to the beginning of the corresponding input frame. Let the resulting time-domain input and output frames be, respectively, $\tilde{x}_{t,i}(n)$ and $\tilde{y}_{t,i}(n)$, $i = 0, ..., N + L_v - 1$. Then,

$$\tilde{x}_{t,i}(n) = \begin{cases} x_{t,(i-L_v)}(n), & i = L_v, ..., N + L_v - 1\,, \\ x_{t,(i+N-L_v)}(n), & i = 0, ..., L_v - 1\,, \end{cases} \tag{2.10}$$

and the last N symbols of the n-th output frame (i.e., $\tilde{y}_{t,i}(n)$, $i \geq L_v$) become the circular convolution of the input symbol frame $\{x_{t,(i-L_v)}(n)\}$ and channel impulse $\{c(i)\}$ in accordance with

$$\begin{aligned} \tilde{y}_{t,i}(n) &= \sum_{k=0}^{i} \tilde{x}_{t,(i-k)}(n)c(k) \\ &= \sum_{k=0}^{i-L_v} x_{t,(i-L_v-k)}(n)c(k) + \sum_{k=i-L_v+1}^{N-1} x_{t,(N+i-L_v-k)}(n)c(k)\,, \end{aligned} \tag{2.11}$$

where $i = L_v, ..., N+L_v-1$, and the second item in RHS is equal to zero when $i-L_v \geq L_c-1$.

Now, by inserting a length L_v ($L_v \geq L_c - 1$) cyclic prefix to each of the corresponding time-domain input frame, interferences can be eliminated, and Eqn (2.7) can be used to recover the input symbols. The effect of the cyclic prefix can also be succinctly illustrated as shown in Figure 2.4.

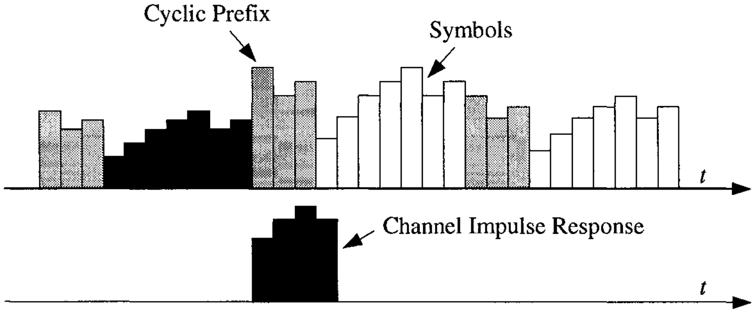


Figure 2.4: Effect of cyclic prefix

The use of cyclic prefix will decrease the transmission efficiency by a factor of $L_v/(N + L_v)$. In order to maintain a reasonable transmission efficiency, two approaches can be used. One is to increase the symbol frame length N, such that $N \gg L_v$. For example, in the conventional DMT system, usually N is from 256 to 2048. In addition to increasing

N, another approach is to decrease L_v. However, note that for a perfect elimination of Intersymbol Interference (ISI) and Interframe Interference (IFI), L_v has to be greater than or equal to $L_c - 1$. When L_c is large, L_v has to be large, and the transmission efficiency will be significantly lowered. Therefore, the length of channel impulse response, L_c, has to be shortened in order to obtain a relatively small L_v and increase the transmission efficiency. A TEQ [12] can be used to shorten L_c. When moving the TEQ to the frequency-domain and incorporate the TEQ with the FEQ, a new equalization scheme, called the per tone equalization [57], for the conventional DMT can be derived. All of these equalization schemes will be discussed in the next chapter.

2.2 Zipper DMT

Zipper DMT [22] [23] [49] is an extension of the conventional DMT system. It was proposed to combat the NEXT problem by allocating different orthogonal subchannels for upstream and downstream signals. The upstream subchannels usually are dynamically interleaving with downstream subchannels in accordance with the channel frequency response. In addition to using the cyclic prefix, the zipper DMT also uses the cyclic suffix to ensure the orthogonality between the transmitted and received signal. Figure 2.5 illustrates the subchannel allocation of zipper DMT.

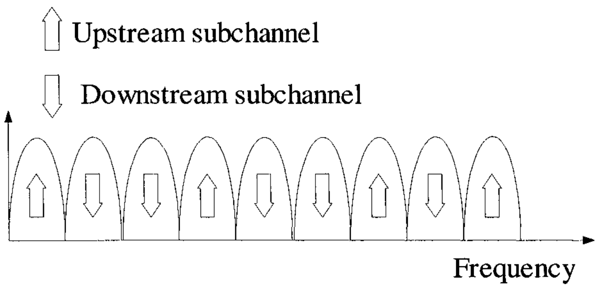


Figure 2.5: Subchannel allocation of zipper DMT

In this figure, the upstream signal in one symbol frame can be expressed as

$$x_{\text{up}}(t) = \begin{cases} \sum_{k\in\mathcal{I}_{\text{up}}} x_k e^{j\frac{2\pi k f_s}{2N}t}, & t \in [0, \frac{2N+\text{CP}+\text{CS}}{f_s}], \\ 0, & \text{otherwise}, \end{cases} \tag{2.12}$$

where $\mathcal{I}_{\text{up}}$ represents the index set of the upstream subchannels, N represents the total subchannel number, and CP, CS represent the length of the cyclic prefix and cyclic suffix, respectively. The downstream signal can also be expressed in a similar form of Eqn (2.12). In order to ensure the transmitted signal real, the input symbols have to satisfy the Hermitian symmetric property in accordance with

$$x_k = x^*_{2N-k} \, . \tag{2.13}$$

The index set $\mathcal{I}_{\text{up}}$ has to satisfy the following condition:

$$\text{If } k \in \mathcal{I}_{\text{up}}, \text{ then } 2N - k \in \mathcal{I}_{\text{up}} \, . \tag{2.14}$$

By using the cyclic suffix, the orthogonality between the upstream and downstream are ensured, and the NEXT is prevented. This can be explained in a similar manner of Section 2.1.2. The idea of cyclic suffix comes from the universal ADSL that using a longer cyclic extension to ensure subchannel orthogonality [17].

Both the conventional DMT and the zipper DMT use DFT/IDFT as the modulator/demodulator. From the view of multirate signal processing, DFT/IDFT can be interpreted as a special type of multirate filterbank [52]. Therefore, the conventional DMT and the zipper DMT systems can also be treated as two special types of multirate filterbank-based DMT systems, which use DFT and IDFT filterbanks as their modulator and demodulator. In the next section, some important concepts used in the multirate signal processing will be given for future use, the general multirate filterbank-based DMT system will be introduced, and the drawback of the conventional DMT as well as the zipper DMT will be discussed.

2.3 Multirate Filterbank-Based DMT

A multirate filterbank-based DMT system is a MCM system using multirate filterbanks as its modulator and demodulator. It is the counterpart of the subband coder [55]. Before giving a detailed description of the multirate filterbank-based DMT system, first let us examine some important concepts used in multirate signal processing.

2.3.1 Expander, Decimator and Polyphase Representation

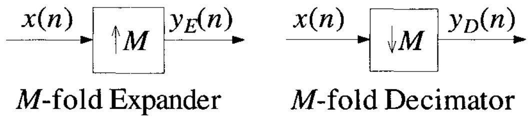


Figure 2.6: M-fold expander and decimator

Expanders and decimators are widely used in the multirate filterbank-based DMT systems. Figure 2.6 shows the M-fold expander ($\uparrow M$) and M-fold decimator ($\downarrow M$) used in the multirate filterbank-based DMT systems. The output of the M-fold expander is the upsampled version of the input with respect to M [52], which in time-domain is

$$y_E(n) = \begin{cases} x(n/M), & n \text{ is integer-multiple of } L, \\ 0, & \text{otherwise,} \end{cases} \tag{2.15}$$

or in z domain,

$$Y_E(z) = X(z^M)\,. \tag{2.16}$$

If the sampling frequency of $x(n)$ is $1/T$, then, from Eqn (2.15), the sampling frequency of $y_E(n)$ is M/T.

The output of the M-fold decimator is obtained by downsampling the input by a factor of M [52], represented by

$$y_D(n) = (x(n))_{\downarrow M} = x(Mn) \tag{2.17}$$

in time-domain, where $(\cdot)_{\downarrow M}$ represents downsampling $(\cdot)$ by a factor of M. In z domain, Eqn (2.17) becomes

$$\begin{aligned} Y_D(z) &= (X(z))_{\downarrow M}\,, \\ &= \frac{1}{M}\sum_{k=0}^{M-1} X(z^{1/M}e^{-j2k\pi/M})\,, \end{aligned} \tag{2.18}$$

Similarly, if the sampling frequency of $x(n)$ is $1/T$, then the sampling frequency of $y_D(n)$ is $1/(MT)$.

In multirate signal processing, the polyphase representation [52] of a signal or a filter is widely used. Given a filter $H(z)$ with the impulse response $h(n)$ (or a signal $h(n)$ with the z-transform $H(z)$), its type 1 polyphase representation (with respect to M) is:

$$H(z) = \sum_{k=0}^{M-1} Z^{-k} H^{(k)}(z^M)\,, \tag{2.19}$$

where $H^{(k)}(z)$ is the k-th polyphase component of $H(z)$, as determined in terms of

$$H^{(k)}(z) = \left(H(z)z^k\right)_{\downarrow M}, \quad k = 0, \ldots, M-1\,, \tag{2.20}$$

or in time-domain, as

$$h^{(k)}(n) = h(Mn+k), \quad k = 0, \ldots, M-1\,, \tag{2.21}$$

where $h^{(k)}(n)$ represents the k-th polyphase component of $h(n)$ in time-domain, and $H^{(k)}(z)$ is the z-transform of $h^{(k)}(n)$.

Similarly, the type 2 polyphase representation (with respect to M) of $H(z)$ is

$$H(z) = \sum_{l=0}^{M-1} Z^{-(M-1-l)} \tilde{H}^{(l)}(z^M), \tag{2.22}$$

where $\tilde{H}^{(l)}(z)$ is the l-th polyphase component of $H(z)$, and $\tilde{H}^{(l)}(z) = H^{(M-1-l)}(z)$.

2.3.2 Structure of Filterbank-Based DMT Systems

By using expanders and decimators, the block diagram of a multirate filterbank-based DMT system is shown in Figure 2.7, where N represents the expansion/decimation ratio, $F_k(z)$ and $H_k(z)$ represent synthesis and analysis filterbanks, respectively, $C(z)$ represents the communication channel, and $e(n)$ represents the additive channel noise.

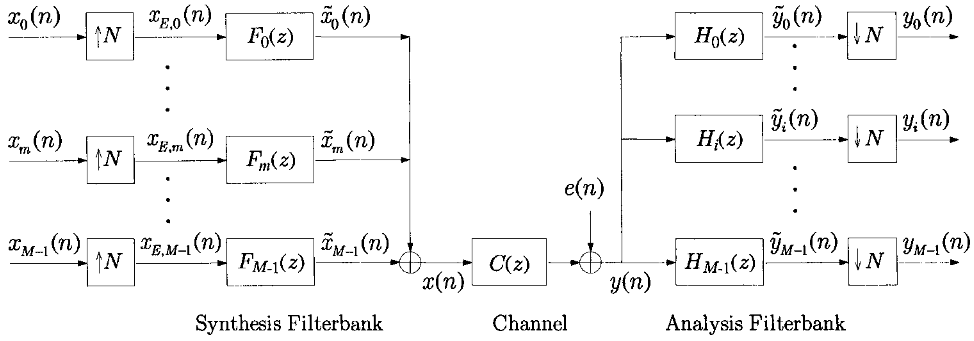


Figure 2.7: Block diagram of filterbank-based DMT systems

At the transmitter, the input bitstream is parsed into M branches or subchannels and PAM or QAM encoding[3] is used to obtain the M parallel input symbols $x_0(n)$, ..., $x_{M-1}(n)$. The input symbols are then expanded by a factor of N, yielding the expanded symbols $x_{E,0}(n)$, ..., $x_{E,M-1}(n)$. After expansion, the expanded symbols pass through the synthesis filterbank, and are subsequently added together to obtain the transmitted signal $x(n)$ before passing through the channel.

At each branch of the receiver, the received signal first passes through the analysis filter, yielding $\tilde{y}_i(n)$ (i = 0, ..., $M - 1$). After decimation operation (by the same factor N), the system output symbols $y_i(n)$ are then obtained. To obtain an output bitstream, the system output symbols are PAM or QAM decoded, and then converted to serial bitstream by using a Parallel-to-Serial converter (not shown in Figure 2.7).

[3] When the impulse responses of the analysis and synthesis filterbanks are real numbers, usually PAM encoding is used. On the other hand, if the impulse responses of the analysis and synthesis filterbanks are complex numbers, QAM encoding is used.

The analysis and synthesis filterbanks partition the whole transmission frequency band into M subchannels. In this thesis, uniform bandwidth partition is assumed, i.e. each subchannel has the same bandwidth $2\pi/M$. When $N = M(N > M)$, the analysis and synthesis filterbanks are said to be critically (non-critically) decimated filterbanks [52]. In digital communication area, DMT and FMT techniques use non-critically decimated filterbanks while DWMT technique and wavelet packet transmultiplexers use critically decimated filterbanks. Because redundancy is introduced in the non-critically decimated filterbank-based DMT systems, the channel equalization of these systems is relatively easy while the critically decimated filterbank-based DMT systems exhibit the maximum throughput because no redundancy is introduced.

Non-critically decimated filterbank systems can be treated as the extension of critically decimated filterbank systems. In the next sub-section, the latter systems will be discussed.

2.3.3 Properties and Input/Output Relationships of DMT Systems Using Critically Decimated Filterbanks

Usually, the filterbank used in a filterbank-based DMT system is carefully designed such that it has the property of biorthogonality [52]. Given a critically decimated filterbank-based DMT system (i.e. $N = M$ in Figure 2.7 and M will be used as the expansion/decimation ratio in this subsection), define $G_{mi}(z)$ as the product of $F_m(z)$ and $H_i(z)$, then the biorthogonal synthesis/analysis filterbanks satisfy the following condition:

$$(G_{mi}(z))_{\downarrow M} = (F_m(z)H_i(z))_{\downarrow M} = \begin{cases} 1, & m = i\,, \\ 0, & m \neq i\,, \end{cases} \tag{2.23}$$

or in time-domain,

$$(g_{mi}(n))_{\downarrow M} = (f_m(n) * h_i(n))_{\downarrow M} = \begin{cases} \delta(n), & m = i\,, \\ 0, & m \neq i\,, \end{cases} \tag{2.24}$$

where $*$ represents linear convolution. In Eqn (2.23) and (2.24), $g_{ii}(n)$ is a called as a Nyquist (M) filter [52], i.e.

$$g_{ii}^{(0)}(n) = g_{ii}(Mn) = \delta(n)\,. \tag{2.25}$$

The biorthogonality means that, in absence of channel effect, i.e. $c(n) = \delta(n)$, and $e(n) = 0$, a DMT system using biorthogonal filterbanks is a Perfect Reconstruction (PR) system with its output equal to its input.

Usually, orthonormal filterbanks are used in DMT systems, i.e.

$$\sum_{n=-\infty}^{\infty} f_{k_1}(n - i_1 M) f_{k_2}(n - i_2 M) = \delta(k_1 - k_2)\delta(i_1 - i_2)\,, \tag{2.26}$$

$$\sum_{n=-\infty}^{\infty} h_{k_1}(n - i_1 M) h_{k_2}(n - i_2 M) = \delta(k_1 - k_2)\delta(i_1 - i_2)\,, \tag{2.27}$$

where $0 \leq k_1, k_2 \leq M - 1$, and $-\infty < i_1, i_2 < \infty$. Then, the analysis filter has a simple relationship to the synthesis filter, called time reversed-conjugation [53], as following

$$h_k(n) = f_k^*(-n), \quad k = 0, ..., M - 1\,. \tag{2.28}$$

The time reversed-conjugation property significantly simplifies the design of filterbanks because only one filterbank (either $f_k(n)$ or $h_k(n)$) has to be designed while the other filterbank can be derived simply by using Eqn (2.28).

When a channel is used, in Figure 2.7, let us define $s_{mi}(n)$ as the impulse response of the overall transfer function from the m-th expander to the i-th decimator, in accordance with

$$s_{mi}(n) \overset{\triangle}{=} \tilde{y}_i(n)|_{\substack{x_{E,i}(n)=\delta(n) \\ x_{E,m}(n)=0\,(m\neq i) \\ e(n)=0}} \quad . \tag{2.29}$$

Then, $s_{mi}(n)$ can be written as

$$s_{mi}(n) = f_m(n) * c(n) * h_i(n) = g_{mi}(n) * c(n)\,, \tag{2.30}$$

or in z-domain,

$$S_{mi}(z) = G_{mi}(z)C(z)\,, \tag{2.31}$$

where $S_{mi}(z)$ and $C(z)$ are z-transforms of $s_{mi}(n)$ and $c(n)$, respectively.

With above definitions, in Figure 2.7,

$$\tilde{Y}_i(z) = \sum_{m=0}^{M-1} X_m(z^M) S_{mi}(z) + E(z)H_i(z)\,. \tag{2.32}$$

where $\tilde{Y}_i(z)$, $X_m(z)$ and $E(z)$ are z-transforms of $\tilde{y}_i(n)$, $x_m(n)$ and $e(n)$, respectively. The polyphase components of $\tilde{Y}_i(z)$ with respect to M are

$$\tilde{Y}_i^{(k)}(z) = \sum_{m=0}^{M-1} X_m(z) S_{mi}^{(k)}(z) + \left(E(z)H_i(z)z^k\right)_{\downarrow M}, \quad k = 0, ..., M - 1\,. \tag{2.33}$$

The polyphase components of $\tilde{Y}_i(z)$ will be used in Chapter 5 to develop the proposed Interpolation Equalization scheme.

By using the polyphase representation, the system output in Figure 2.7 can be written as

$$Y_i(z) = \left(\tilde{Y}_i(z)\right)_{\downarrow M} = \tilde{Y}_i^{(0)}(z) = \sum_{m=0}^{M-1} X_m(z)S_{mi}^{(0)}(z) + (E(z)H_i(z))_{\downarrow M}\,, \tag{2.34}$$

or in time-domain,

$$\begin{aligned} y_i(n) &= \sum_{m=0}^{M-1} x_m(n) * s_{mi}^{(0)}(n) + (e(n) * h_i(n))_{\downarrow M} \\ &= x_i(n) * s_{ii}^{(0)}(n) + \sum_{\substack{m=0 \\ m\neq i}}^{M-1} x_m(n) * s_{mi}^{(0)}(n) + (e(n) * h_i(n))_{\downarrow M}\,. \end{aligned} \tag{2.35}$$

The second term in the RHS of Eqn (2.35) is the Interchannel Interference (ICI) in the system output. When $s_{mi}^{(0)}(n)$ is not a δ function, ISI exists in the system output.

In practical situation, the channel is non-ideal, and $s_{mi}(n) \neq g_{mi}(n)$. In this case, biorthogonal systhesis/analysis filterbanks would not ensure a filterbank-based DMT system to be a PR system, and the system output will be distorted from the system input because of the ICI and ISI. In this case, the PR property should be ensured by either redesigning the synthesis and/or analysis filterbanks such that

$$s_{mi}^{(0)}(n) = \begin{cases} \delta(n), & m = i\,, \\ 0, & m \neq i\,, \end{cases} \tag{2.36}$$

($g_{mi}(n)$ would not satisfy Eqn (2.24) then), or by using an equalizer at the receiver side while keeping Eqn (2.24) hold.

Before further discussion, let us consider the effective implementation of the filterbank-based DMT system.

2.3.4 Effective Matrix Implementation

Because expanders and decimators are used, a drawback of the filterbank-based DMT system is that the synthesis and analysis filterbanks have to work at higher sampling frequencies than those of the input and output symbols. Specifically, suppose the sampling frequency of the input symbols $x_i(n)$ in Figure 2.7 is $1/T$. Then, after expansion operation, the sampling frequency of $x_{E,i}(n)$ is N/T. Therefore, the synthesis and analysis filterbanks have to work with the sampling frequency of N/T, which is N times higher than the input and output symbols. This leads to a high hardware requirement for the filterbank-based DMT system, specially when the sampling frequencies of the input/output symbols are very high.

However, an effective implementation structure of the filterbank-based DMT system can be derived by using matrix representation of the analysis/synthesis filterbanks [52]:

The transmitted signal $x(n)$ in Figure 2.7 can be written in z-domain as

$$X(z) = \sum_{m=0}^{M-1} X_m(z^N) F_m(z)\,, \tag{2.37}$$

which can be rewritten in matrix form as

$$X(z) = \begin{bmatrix} F_0(z) & F_1(z) & \cdots & F_{M-1}(z) \end{bmatrix} \begin{bmatrix} X_0(z^N) \\ X_1(z^N) \\ \vdots \\ X_{M-1}(z^N) \end{bmatrix}\,. \tag{2.38}$$

By decomposing $F_m(z)$ ($m = 0, ..., M-1$) into its type 1 polyphase components with respect to N, one can write $F_m(z)$ in matrix form as

$$F_m(z) = \begin{bmatrix} 1 & z^{-1} & \cdots & z^{-N+1} \end{bmatrix} \begin{bmatrix} F_m^{(0)}(z^N) \\ F_m^{(1)}(z^N) \\ \vdots \\ F_m^{(N-1)}(z^N) \end{bmatrix}\,. \tag{2.39}$$

Substitute Eqn (2.39) into Eqn (2.38), we have

$$X(z) = \begin{bmatrix} 1 & z^{-1} & \cdots & z^{-N+1} \end{bmatrix} \begin{bmatrix} F_0^{(0)}(z^N) & \cdots & F_{M-1}^{(0)}(z^N) \\ F_0^{(1)}(z^N) & \cdots & F_{M-1}^{(1)}(z^N) \\ \vdots & \ddots & \vdots \\ F_0^{(N-1)}(z^N) & \cdots & F_{M-1}^{(N-1)}(z^N) \end{bmatrix} \begin{bmatrix} X_0(z^N) \\ X_1(z^N) \\ \vdots \\ X_{M-1}(z^N) \end{bmatrix}\,. \tag{2.40}$$

Define $\boldsymbol{F}(z)$ as

$$\boldsymbol{F}(z) = \begin{bmatrix} F_0^{(0)}(z) & \cdots & F_{M-1}^{(0)}(z) \\ F_0^{(1)}(z) & \cdots & F_{M-1}^{(1)}(z) \\ \vdots & \ddots & \vdots \\ F_0^{(N-1)}(z) & \cdots & F_{M-1}^{(N-1)}(z) \end{bmatrix}\,. \tag{2.41}$$

Then,

$$\begin{aligned} X(z) &= \begin{bmatrix} 1 & z^{-1} & \cdots & z^{-N+1} \end{bmatrix} \boldsymbol{F}(z^N) \begin{bmatrix} X_0(z^N) & X_1(z^N) & \vdots & X_{M-1}(z^N) \end{bmatrix}^T \\ &= \begin{bmatrix} 1 & z^{-1} & \cdots & z^{-N+1} \end{bmatrix} \left(\boldsymbol{F}(z) \begin{bmatrix} X_0(z) & X_1(z) & \vdots & X_{M-1}(z) \end{bmatrix}^T \right)_{\uparrow N}\,, \end{aligned} \tag{2.42}$$

where $[\cdot]^T$ represents the matrix transpose operation, and $(\cdot)_{\uparrow N}$ represents expansion operation by a factor of N.

Similarly, by decomposing $H_i(z)$ $(i = 0, ..., M-1)$ into its type 1 polyphase components with respect to N, the system output can be expressed as

$$\begin{bmatrix} \hat{X}_0(z) \\ \hat{X}_1(z) \\ \vdots \\ \hat{X}_{M-1}(z) \end{bmatrix} = \left(\boldsymbol{H}(z^N) \begin{bmatrix} 1 \\ z^{-1} \\ \vdots \\ z^{-N+1} \end{bmatrix} Y(z) \right)_{\downarrow N} = \boldsymbol{H}(z) \left(\begin{bmatrix} 1 \\ z^{-1} \\ \vdots \\ z^{-N+1} \end{bmatrix} Y(z) \right)_{\downarrow N} , \qquad (2.43)$$

where

$$\boldsymbol{H}(z) = \begin{bmatrix} H_0^{(0)}(z) & H_0^{(1)}(z) & \cdots & H_0^{(N-1)}(z) \\ \vdots & \vdots & \ddots & \vdots \\ H_{M-1}^{(0)}(z) & H_{M-1}^{(1)}(z) & \cdots & H_{M-1}^{(N-1)}(z) \end{bmatrix} , \qquad (2.44)$$

and

$$Y(z) = X(z)C(z) + E(z) . \qquad (2.45)$$

Combining Eqn (2.42) and (2.43), an effective implementation of Figure 2.7 can be derived as shown in Figure 2.8.

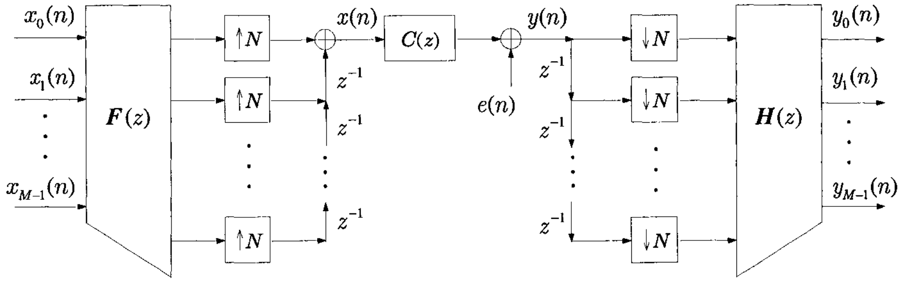


Figure 2.8: Efficient implementation of filterbank-based DMT systems

This effective implementation has three advantages. First, instead of originally working with a high sampling frequency N/T, the synthesis and analysis filter matrices $\boldsymbol{F}(z)$ and $\boldsymbol{H}(z)$ now are working at the same sampling frequency $(1/T)$ of the input symbols. The lowered sampling frequency makes the hardware implementation of a DMT system easier. Moreover, the transmitted signal $x(n)$ is generated by simply converting the output of $\boldsymbol{F}(z)$ into serial using a P/S converter (instead of originally adding together the output of the synthesis filterbank, which is more expensive in hardware implementation). Also, the input of the analysis filter matrix $\boldsymbol{H}(z)$ is obtained by parsing the received signal $y(n)$ in parallel using a S/P converter. Furthermore, this effective implementation structure is suitable for fast implementation using some kinds of fast computation algorithms such as Fast Wavelet Transform (FWT) or FFT.

2.3.5 Drawbacks of the Conventional and Zipper DMT Systems

The conventional and the zipper DMT systems belong to a special case of the DMT systems using orthonormal filterbanks, where the synthesis and analysis filterbanks are IDFT and DFT filterbanks [52], respectively. IDFT and DFT filterbanks satisfy the time reversed-conjugation property shown in Eqn (2.28), and all filters of them are modulated versions of a single prototype filter $p_0(n)$ that (ignoring the cyclic prefix)

$$p_0(n) = \begin{cases} 1, & n = 0, \ldots, N-1\,, \\ 0, & \text{otherwise,} \end{cases} \tag{2.46}$$

and

$$f_k(n) = \frac{1}{N} p_0(n) e^{j\omega_k n}\,, \tag{2.47}$$

$$h_k(n) = p_0(n) e^{-j\omega_k n}\,, \tag{2.48}$$

where $\omega_k = 2\pi/N$ is the k-th center-frequency.

The magnitude response of the IDFT filterbank is shown in Figure 2.9, where the first sidelobe of the magnitude response of each synthesis filter is only about -13 dB. The magnitude responses of the DFT filterbank is the same as in Figure 2.9. Because of the large sidelobes (only -13 dB) associated with the synthesis and analysis filterbank, substantial spectral overlapping takes place particularly between the adjacent subchannels. This spectral overlap leads to poor subchannel isolation and makes the conventional DMT system sensitive to channel distortion and narrowband noise, which is the main drawback of the conventional DMT system. The other two drawbacks of the conventional DMT and the zipper DMT are the lowered transmission efficiency, which has been discussed in Section 2.1, and drawbacks (e.g. the difficulty to find an optimal decision delay) inherited in the design of TEQ, which will be discussed in the next chapter.

In order to improve the spectral isolation, other DMT systems employing multirate filterbanks that own smaller or none spectral overlapping have become popular. In [56] and [54], Vaidyanathan indicated the theoretical upper bound of the optimal orthonormal filterbanks used in DMT systems. Cherubini, *et al* proposed the Filtered Multitone (FMT) system in [11], which uses non-critically decimated filterbanks as the modulator/demodulator. Although the non-critically decimated filterbanks make the spectral overlapping negligible, the resulting redundancy leads to a lowered throughput. In [45], Sandberg and Tzannes presented another filterbank-based DMT system called Discrete Wavelet Multitone (DWMT)

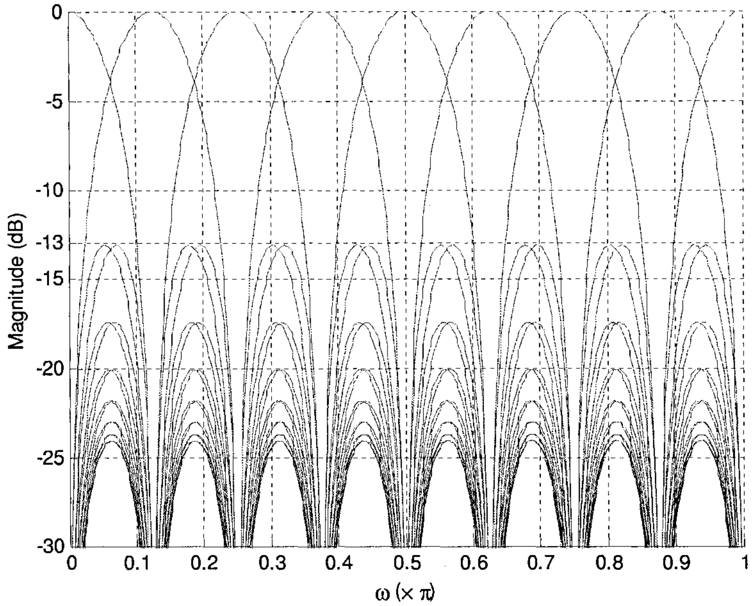


Figure 2.9: Magnitude response of the IDFT filterbank

[45]. This system uses critically decimated filterbanks that have small spectral overlapping, such as the Cosine-Modulated Filterbanks (CMFBs), as the modulator/demodulator. Unlike the conventional and zipper DMT systems, all of these systems need no cyclic prefix.

2.4 Optimal Orthonormal DMT Systems

In Section 1.3.4, it has been shown that for a given channel frequency response and a fixed channel noise power spectrum, the maximum transmission bit-rate is determined by the effective noise PSD $S_{qq}(e^{j\omega})$ (c.f. Eqn (1.10)). Based on this fact, Vaidyanathan *et al.* [56] [54] gave a theoretical proof that for orthonormal DMT systems in a noisy environment, the optimal receiver filterbank $\{H_k(z)\}$ maximizing the total bit-rate is the Principal Component Filterbank (PCFB) if a PCFB solution exists.

The PCFB is a special type of orthonormal PR filterbank explained as follows. Let us denote a class of M-subchannel orthonormal PR filterbanks as $\mathcal{C}$, i.e.

$$\mathcal{C} = \{\mathbb{H}_i\}, \quad i = 1, ..., \Omega \tag{2.49}$$

where Ω represents the number of filterbanks in $\mathcal{C}$, and $\mathbb{H}_i$ represents the i-th filterbank in accordance with

$$\mathbb{H}_i = \{H_{i,k}(z)\}, \quad k = 0, ..., M-1\,, \tag{2.50}$$

with $H_{i,k}(z)$ representing the k-th filter of the i-th filterbank $\mathbb{H}_i$.

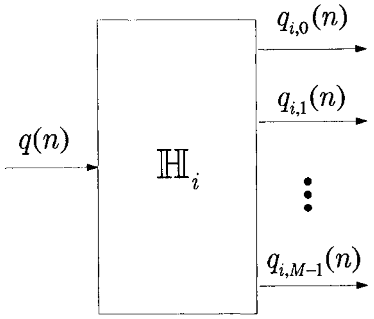


Figure 2.10: Receiver filterbank $\mathbb{H}_i$

Given an input signal $q(n)$ of power spectrum $S_{qq}(e^{j\omega})$, as shown in Figure , for each filterbank $\mathbb{H}_i$ ($i = 1, ..., \Omega$), let us denote its outputs as $\{q_{i,k}(n)\}$ ($k = 0, ..., M-1$), and denote the variance of $q_{i,k}(n)$ as $\{\sigma^2_{i,k}\}$. Moreover, without loss of generality, suppose the index k of the outputs $\{q_{i,k}(n)\}$ is rearranged such that $\{\sigma^2_{i,k}\}$ are in non-increasing order (i.e. $\sigma^2_{i,k} \geq \sigma^2_{i,k+1}$). Then, the partial sum of each filterbank $\mathbb{H}_i$ can be defined as

$$\mathcal{S}_{i,l} \triangleq \sum_{k=0}^{l} \sigma^2_{i,k}, \quad 0 \leq l \leq M-1\,. \tag{2.51}$$

Obviously, $\mathcal{S}_{i_1,M-1} = \mathcal{S}_{i_2,M-1}$ for any $1 \leq i_1, i_2 \leq \Omega$.

Now, the PCFB $\mathcal{C}_{i_0}$ is a filterbank in $\mathcal{C}$ such that

$$\mathcal{S}_{i_0,l} = \max_i\{\mathcal{S}_{i,l}\}, \quad \text{for any } l\,, \tag{2.52}$$

or equivalently,

$$\sigma^2_{i_0,0} \geq \sigma^2_{i,0},\ \sigma^2_{i_0,0} + \sigma^2_{i_0,1} \geq \sigma^2_{i,0} + \sigma^2_{i,1},\ ... \tag{2.53}$$

A PCFB divides the whole frequency band into M subchannels. Each subchannel contains the frequency regions that have same SNR. For example, all "best" frequency regions that have large SNRs are in one subchannel while all "worst" frequency regions that have small SNRs are in another subchannel. Figure 2.11 illustrates the frequency subdivision by using a PCFB (Figure 2.11a) and by using a brick-wall filterbank (Figure 2.11b). By using a PCFB, the frequency regions in a subchannel have same SNR. Then, a "bad" frequency region that has small SNR would not affect other "good" frequency region that has large SNR. Therefore, the transmission bit-rate can be maximized. On the other hand, by using a brick-wall filterbank, a subchannel may contain both "good" and "bad" frequency regions (c.f. Figure 2.11b). In this case, the effective SNR of this subchannel is the average of the SNRs of both frequency regions. Therefore, the subchannel SNR, and then the transmission bit-rate, would be lowered.

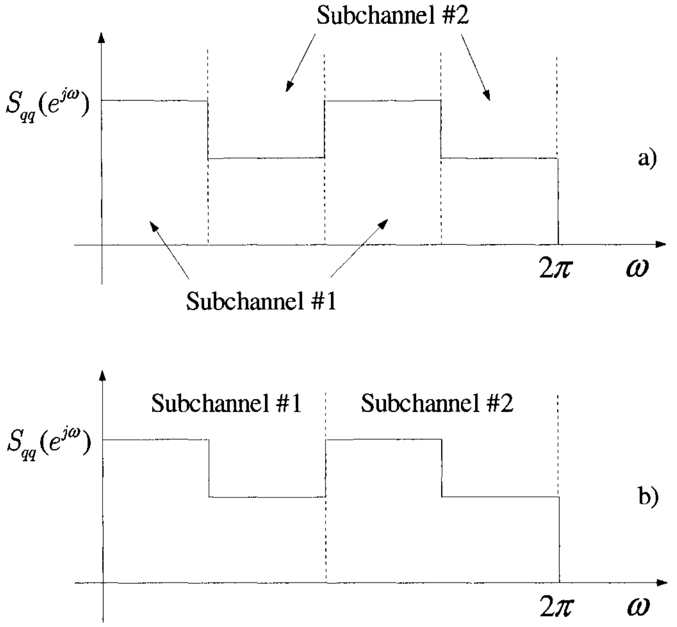


Figure 2.11: Frequency subdivision for a) PCFB and b) brick-wall filterbank

The above discussion implies that each filter in a PCFB may have multiple passbands. Also, the frequency location and the width of each passband are determined by the channel frequency response. Moreover, the subdivision of the whole frequency band is not uniform anymore, and the widths of passbands are different to each other. On the other hand, because the DFT and Cosine-Modulated Filterbank (CMFB) filterbanks are modulated filterbanks (i.e. each filter is a modulated version of the same prototype filter), their frequency subdivision is uniform and the widths of their passbands are the same. Therefore, the PCFB solution is not exist in DFT and CMFB filterbank classes.

Although, generally, the PCFB solution for DMT systems is difficult to implement (or not exist), it can be used as a guideline for performance comparisons of other possible suboptimal solutions. Moreover, an approximation of the PCFB solution can be made by using brick-wall filterbank when the number of subchannel are significantly large. In this case, each subchannel is narrow enough. Then, the "bad" frequency region would not belong to the same subchannel as the "good" frequency region. Now, by treating each group of subchannels that have the same SNR as a single (multi-passband) subchannel, the PCFB solution is approximated.

2.5 Discrete Wavelet Multitone Modulations

2.5.1 Modulation and Demodulation Procedure

S.D. Sandberg and M.A. Tzannes introduced the DWMT system in 1995 [45] by replacing the IDFT and DFT in the conventional DMT system with the Inverse Discrete Wavelet Transform (IDWT) and Discrete Wavelet Transform (DWT), respectively, where IDWT and DWT can be efficiently implemented in practical situations by using the Inverse Fast Wavelet Transform (IFWT) and DWT. As shown in Figure 2.12, the DWMT system divides the channel frequency band into M equal-width subchannels. Note that spectral overlap still occurs in adjacent subchannels. However, this spectral overlap can be significantly small by carefully designing the filterbank used in the DWMT system.

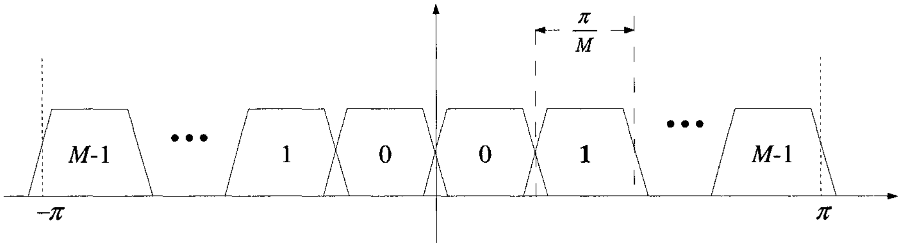


Figure 2.12: Frequency band subdivision of the DWMT system

The block diagram of the DWMT system is shown in Figure 2.13. At the transmitter, the serial input bitstream is first parsed into several parallel bitstreams by using a S/P converter. Then, these parallel bitstreams are converted into symbol sequences at the constellation encoder outputs. The resulting symbol sequences are then transformed into time-domain by using the IDWT operation. Finally, the time-domain signal is converted to analog signal by a D/A converter and then sent to the communication channel.

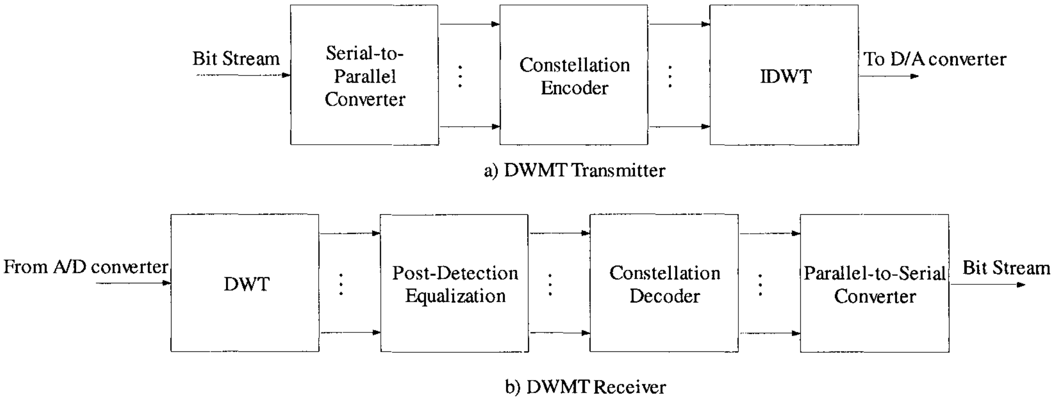


Figure 2.13: Block diagram of the DWMT system

At the receiver, after A/D conversion, the received symbols are demodulated by the DWT operation and then pass through the post-detection equalizers to equalize the channel distortion and obtain the recovered symbols. After that, the corresponding constellation decoders and a P/S converter are used to convert the recovered symbols back to a serial bitstream.

2.5.2 Filterbank Structure and Pulse Overlap

Similar to the conventional DMT system, the DWMT system is also a special type of DMT system that employs critically decimated filterbanks as its modulator and demodulator. Several types of critically decimated filterbanks can be used in DWMT systems, for example, the Fourier transforms (DFT and IDFT), M-band wavelet trasnforms [60], the Lapped Orthogonal Transform (LOT) and extended LOT [37][38], and CMFBs [32]. These filterbanks can be carefully designed such that the sidelobes of the frequency response of the filters used are significantly small. Therefore, better spectral isolation can be achieved compared to DFT and IDFT filterbanks. In [45], the authors employed the Cosine IV bases CMFB to design the DWMT system.

When treated as a type of filterbank-based DMT system, the block diagram of the DWMT system is the same as Figure 2.7 with $N = M$. All properties discussed in Section 2.3.3 also hold for the DWMT system. Specifically, the analysis and synthesis filterbanks ($\{F_m(z)\}$ and $\{H_i(z)\}$ where $m, i = 0, ..., M-1$) used in DWMT systems satisfy the biorthogonality expressed in Eqns (2.23) and (2.24), and the time reversed-conjugation property in Eqn (2.28). When CMFBs are employed, all filters used in the CMFB are modulated from a single prototype filter $p_0(n)$ in accordance with

$$f_m(n) = 2p_0(n)\cos\left(\frac{\pi}{M}(m+0.5)(n-\frac{L}{2}) - (-1)^m\frac{\pi}{4}\right), \tag{2.54}$$

$$h_i(n) = 2p_0(n)\cos\left(\frac{\pi}{M}(i+0.5)(n-\frac{L}{2}) + (-1)^i\frac{\pi}{4}\right), \tag{2.55}$$

where $0 \leq m, i \leq M-1$, and L is the length of the prototype filter $p_0(n)$.

The length of each filter in the CMFB is equal to the length of the prototype filter $p_0(n)$. For a M-band CMFB, usually, the length of $p_0(n)$ is chosen to be $L = gM$, where $g > 1$ is an integer[4]. In this case, "the pulse waveforms for different symbol blocks overlap in time" ([45], section II), and usually, g is called as the overlap factor. This "pulse overlap" can be clearly explained as follows.

[4] For conventional DMT systems, the length of the prototype filter $L = N$ (N is the number of subchannels), which means that $g = 1$.

Referring to Figure 2.7, in the m-th subchannel of the transmitter, the output $\tilde{x}_m(n)$ of the synthesis filter $F_m(z)$ can be written as

$$\begin{aligned}\tilde{x}_m(n) = x_{E,m}(n) * f_m(n) &= \sum_{k=-\infty}^{\infty} x_{E,m}(k) * f_m(n-k) \\ &= \sum_{k=-\infty}^{\infty} x_m(k) * f_m(n-Mk) \\ &= \sum_{k=\lfloor \frac{n+1}{M} - g \rfloor}^{\lfloor \frac{n}{M} \rfloor} x_m(k) * f_m(n-Mk)\,,\end{aligned} \qquad (2.56)$$

where $\lfloor \cdot \rfloor$ represents rounding to zero. From Eqn (2.56), it is clear that any sample of $\tilde{x}_m(n)$ is the convolution of the synthesis filter $f_m(n)$ and g samples of the corresponding system input (i.e. $\{x_m(k)\}$, where $k = \lfloor \frac{n+1}{M} - g \rfloor, ..., \lfloor \frac{n}{M} \rfloor$). Therefore, one can say that each transmitted symbol is generated from overlapped input symbol blocks. This procedure can be graphically illustrated in Figure 2.14.

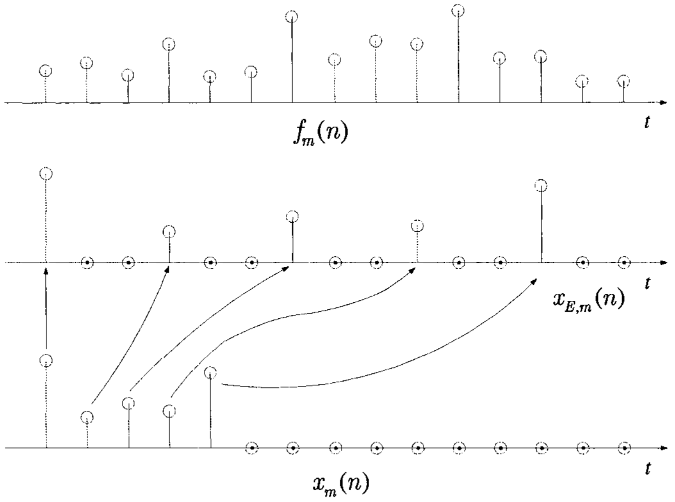


Figure 2.14: Graphical illustration of pulse overlap

2.5.3 Design of CMFB

There are a lot of techniques to design the CMFB [25] [26] [33] [43] [47]. In this thesis, the Kaiser window approach proposed in [33] is used. In general, in order to design a PR CMFB, the prototype filter $p_0(n)$ has to satisfy the following conditions [52]:

$$\left|P(e^{j\omega})\right| \approx 0, \quad \text{for } |\omega| > \pi/M\,, \qquad (2.57)$$

$$T(e^{j\omega}) \approx 1, \quad \text{where } T(e^{j\omega}) = \sum_{k=0}^{2M-1} \left|P(e^{j(\omega - k\pi/M)})\right|^2. \qquad (2.58)$$

Now, if the stopband edge ω_s is set to $\omega_s = \pi/M$, the first condition (Eqn (2.57)) can be achieved. Note that because $T(e^{j\omega})$ is periodic with period π/M, only the $T(e^{j\omega})$ in the interval $[0,\ \pi/M)$ has to be considered. By letting $\omega_s = \pi/M$, $T(e^{j\omega})$ can be approximated as

$$T(e^{j\omega}) \approx \left|P(e^{j\omega})\right|^2 + \left|P(e^{j(\omega-\pi/M)})\right|^2, \quad \text{for } 0 \leq \omega < \pi/M\,. \tag{2.59}$$

With this approximation, the objective function can be formed from Eqn (2.58) as

$$\min_{\omega_p} \{\phi\}\,, \tag{2.60}$$

where ω_p is the passband edge of the frequency response of $p_0(n)$, and ϕ is defined as

$$\begin{aligned}\phi &\triangleq \max_{\omega} \left\{\left|T(e^{j\omega}) - 1\right|\right\} \\ &= \max_{\omega} \left\{\left|\left|P(e^{j\omega})\right|^2 + \left|P(e^{j(\omega-\pi/M)})\right|^2 - 1\right|\right\}, \quad 0 \leq \omega < \pi/M\,.\end{aligned} \tag{2.61}$$

A linear phase prototype filter can be obtained by using Kaiser window approach. With this approach, a filter $p_0(n)$ of length $L+1$ is designed as

$$p_0(n) = h(n)w(n)\,, \tag{2.62}$$

where $h(n)$ is an ideal filter with cutoff frequency ω_c, and $w(n)$ is a Kaiser window in accordance with

$$h(n) = \frac{\sin(\omega_c(n-0.5L))}{\pi(n-0.5L)}\,, \tag{2.63}$$

$$w(n) = \begin{cases} \frac{I_0(\beta)\sqrt{1-\left(\frac{n-0.5L}{0.5L}\right)^2}}{I_0(\beta)}, & 0 \leq n \leq L\,, \\ 0, & \text{otherwise}\,, \end{cases} \tag{2.64}$$

where $I_0(\cdot)$ is the zeroth-order modified Bessel function. For given stopband attenuation A_s, the parameter β is [52]

$$\beta = \begin{cases} 0.1102(A_s - 8.7), & A_s > 50\,, \\ 0.5842(A_s-21)^{0.4} + 0.07886(A_s-21), & 21 < A_s < 50\,, \\ 0, & A_s < 21\,. \end{cases} \tag{2.65}$$

The relationship between the order of the window L, the stopband attenuation A_s, and an appropriately chosen transition bandwidth $\Delta\omega$ is

$$L \approx \frac{A_s - 7.95}{14.36\Delta\omega/2\pi}\,. \tag{2.66}$$

By using Eqns (2.62) – (2.66), the prototype filter $p_0(n)$ can be generated and its frequency response can be adjusted by a single parameter: the cut-off frequency ω_c. Now define a filter

$$g(n) = p_0(n) * p_0(-n)\,, \tag{2.67}$$

which means that in frequency-domain

$$G(e^{j\omega}) = \left|P(e^{j\omega})\right|^2 . \tag{2.68}$$

By substituting Eqn (2.68) into Eqn (2.61), one can say that the filter $g(n)$ that satisfies the objective function in Eqn (2.60) is approximately a Nyquist $(2M)$ filter, i.e. $g(2Mn) \approx \delta(n)$. In light of this, the objective function can be simplified as

$$\min_{\omega_c} \{\phi_{\text{new}}\} , \tag{2.69}$$

where

$$\phi_{\text{new}} = \max_{\substack{n \\ n\neq 0}} \{|g(2Mn)|\} . \tag{2.70}$$

This optimization problem can be solved by using region elimination methods [44].

2.5.4 Advantages and Disadvantages

Compared to the conventional DMT system, DWMT systems have the advantage of higher throughput because no redundancy is introduced in it. Also, DWMT systems are robust to narrow-band channel noise because of its better spectral isolation. Usually, post-detection equalizers are used in DWMT systems to equalizer each subchannel separately. Therefore, the equalization can directly related to the optimization of the transmission bit-rate. Moreover, because each subchannel is equalized separately, no tradeoff has to be made between different subchannels in equalization procedure, and the maximum transmission bit-rate can be achieved.

On the other hand, DWMT systems have two main disadvantages. First, because the overlap factor $g > 1$, longer filters are used in DWMT systems, which means that more memory units have to be used "to maintain a symbol buffer of size gM" [45]. Another disadvantage of DWMT systems is that the equalization is more complicated compared to the conventional DMT systems and other DMT systems using non-critically decimated filterbanks.

2.6 Filtered Multitone Modulations

Although the spectral isolation of DWMT systems is significantly better that the conventional DMT system, spectral overlap of DWMT systems still exists between the neighboring subchannels. In order to further increase the spectral isolation, the FMT technique was introduced in [11].

The FMT system uses complex-exponential-modulated filter banks with

$$f_k(n) = p_0(n)e^{j2\pi nk/M}, \tag{2.71}$$

$$h_k(n) = p_0^*(-n)e^{-j2\pi nk/M}, \tag{2.72}$$

where $k = 0, ..., M-1$.

Usually, FMT systems employ non-critically decimated filterbanks (Figure 2.7 with $N > M$) so as to introduce redundancy to combat the channel distortion. The frequency division of FMT systems is illustrated in Figure 2.15. In this figure, the frequency responses of neighboring subchannel pairs cross at the stopband frequencies instead of the cutoff frequencies in DMT systems using critically decimated filterbanks. Therefore, there is (approximately[5]) no spectral overlap in the FMT system.

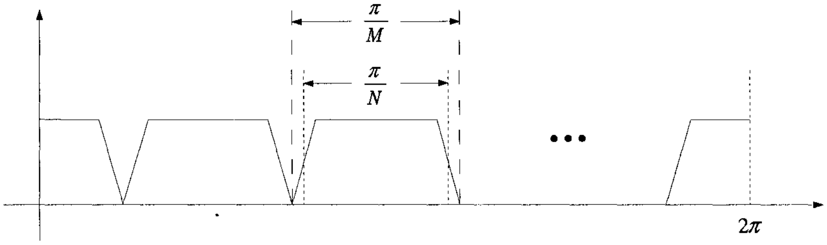


Figure 2.15: Frequency subdivision for FMT systems

The penalty for the added redundancy is the reduction in the bandwidth efficiency. However, the reduction in the bandwidth efficiency can be minimized by making N close to M, which requires filters with sharper roll-offs (increasing their implementation complexity and the system latency).

An efficient implementation of FMT systems using FFT and IFFT is shown in Figure 2.16, where $(F^{(0)}(z), \ldots, F^{(M-1)}(z))$ are M polyphase components of the prototype filter $p_0(n)$, and $(H^{(0)}(z), \ldots, H^{(M-1)}(z))$ are M polyphase components of $p_0^*(-n)$.

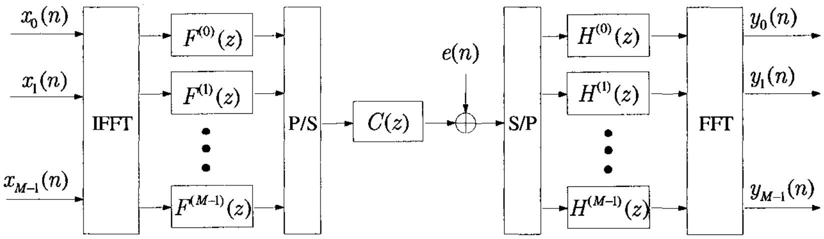


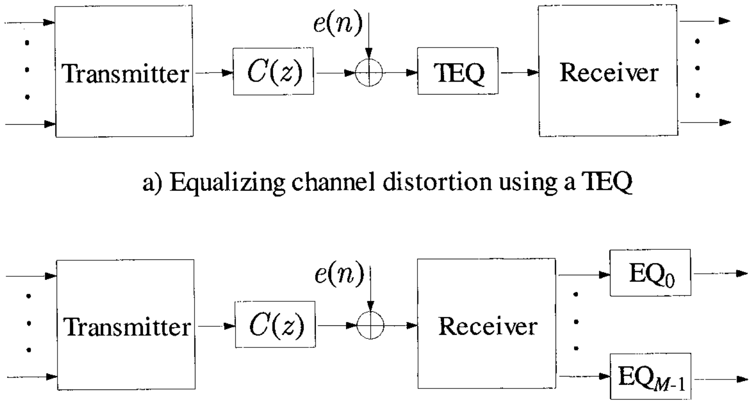
Figure 2.16: Efficient implementation of FMT systems

[5] Here, we say "approximately" because in practical situations, the stopband of each filter is not precisely equal to 0.

Chapter 3

Channel Equalization

Channel equalization plays an important role in DMT systems for combatting the distortion caused by the communication channel. Generally, channel equalization can be divided into two categories as illustrated in Figure 3.1.



a) Equalizing channel distortion using a TEQ

b) Equalizing channel distortion subchannel-by-subchannel

Figure 3.1: Two channel equalization categories

In the first category (Figure 3.1a), a TEQ is cascaded at the input of the system receiver to equalize the channel distortion in the whole channel frequency band. In the other category (Figure 3.1b), on the other hand, each subchannel has a separate equalizer (EQ_0, ..., EQ_{M-1}) at the system output end, which equalizes the channel distortion only within the frequency band of the corresponding subchannel. Usually, the latter category is called as "post-detection equalization". Most of the conventional DMT systems use TEQ to equalize (or shorten) the channel to a small length and then use the cyclic prefix to combat the remaining channel distortion (including ISI and ICI). On the other hand, other DMT systems such as FMT and DWMT systems usually employ the post-detection equalization to equalize the channel distortion in each subchannel separately.

3.1 Time-domain Equalizer in Conventional DMT System

A TEQ [12] is usually an Finite Impulse Response (FIR) filter cascaded with the channel to equalize the channel distortion, as shown in Figure 3.2, where $C(z)$ and $W(z)$ represent the channel and TEQ, respectively.

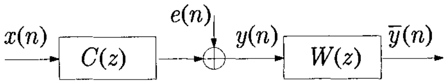


Figure 3.2: TEQ cascaded with the channel

If the channel is fully equalized (in which case $\bar{y}(n) = x(n)$), then a zero-forcing equalizer[1] $W(z) = 1/C(z)$ has to be used. However, the zero-forcing equalizer has two problems. First, generally it cannot be implemented using an FIR filter. In other words, a FIR zero-forcing equalizer cannot fully eliminate the ISI. Another problem of the zero-forcing equalizer is the noise enhancement. In practical situations, the channel attenuation in high frequency band is high. Then, the gain in the high frequency band of the zero-forcing equalizer $W(z)$ has to be large, which will make the high-frequency noise component at the system output intolerably large and significantly lower the transmission bit-rate. A solution to this problem is used in the conventional DMT system where, instead of fully equalizing the channel distortion, an equalizer is used to shorten the channel impulse response to a small number of taps (i.e. partially equalizing the channel distortion), while using the cyclic prefix to combat the remaining channel distortion. In the other word, as mentioned in Section 2.1, in order to fully eliminate the ISI and IFI, the length L_v of the cyclic prefix has to be greater or equal to $L_c - 1$, where L_c represents the length of the channel impulse response. However, L_v has to be small so as to maintain a reasonable transmission efficiency. Therefore, the length of the channel impulse response has to be shortened by using a TEQ [12].

3.1.1 Minimum Mean-Squared Error Design Model for TEQ

The block diagram in Figure 3.3 is the model for the design of TEQ equalizers [2] [29] [3]. In this model, $c(n)$ represents the channel[2] impulse response, $w(n)$ represents the

[1] The zero-forcing equalizer is named because the ISI is "forced to zero" [13].

[2] The channel used in this thesis is a discrete model of a practical continuous-time channel and the additional A/D, D/A converters, the splitter (if used), and so on.

impulse response of the TEQ, $x(n)$ and $\bar{y}(n)$ represent the transmitted signal at the input of the channel and the received signal at the output of the channel, respectively, and $e(n)$ represents the additive channel noise. The TEQ $w(n)$ shortens the channel impulse response $c(n)$ approximately to the desired short-length Target Impulse Response (TIR) $b(n)$ with a decision delay of $\varDelta$ and an approximation error of $r(n)$. This model is employed to find the optimal TEQ $w(n)$ that leads to the Minimum Mean-Squared Error (MMSE) of $r(n)$.

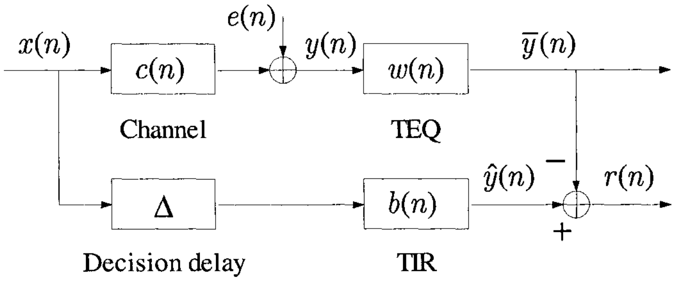


Figure 3.3: TEQ structure

A comment has to be mentioned before the mathematical calculation of the optimal TEQ settings. Note that the lower path in Figure 3.3 (i.e. the combination of the TIR $b(n)$ and the decision delay $\varDelta$) is drawn only for the optimization of the TEQ settings and would not be present in an actual xDSL modem. Moreover, in an actual xDSL modem, the TEQ output has to be advanced by $\varDelta$ samples[3] to compensate for the decision delay.

Let L_c, L_w, and L_b represent the length of the channel, the TEQ, and the TIR, respectively. The channel output can be written as

$$y(n) = x(n) * c(n) + e(n) = \sum_{k=0}^{L_c-1} c(k)x(n-k) + e(n)\,. \tag{3.1}$$

Then, L_w successive symbols of the channel output, $y(n)$, $y(n-1)$, ..., $y(n-L_w+1)$, can be expressed in matrix form as

$$\boldsymbol{y}(n) = \boldsymbol{C}\,\boldsymbol{x}(n) + \boldsymbol{e}(n)\,, \tag{3.2}$$

where $\boldsymbol{y}(n)$, $\boldsymbol{x}(n)$, and $\boldsymbol{e}(n)$ are defined as

$$\boldsymbol{y}(n) \triangleq \begin{bmatrix} y(n) & y(n-1) & \cdots & y(n-L_w+1) \end{bmatrix}^T\,, \tag{3.3}$$

$$\boldsymbol{x}(n) \triangleq \begin{bmatrix} x(n) & x(n-1) & \cdots & x(n-L_c-L_w+1) \end{bmatrix}^T\,, \tag{3.4}$$

$$\boldsymbol{e}(n) \triangleq \begin{bmatrix} e(n) & e(n-1) & \cdots & x(n-L_w+1) \end{bmatrix}^T\,, \tag{3.5}$$

[3] Because of the successive S/P operation (see Figure 2.2), here the operation of advancing the TEQ output by $\varDelta$ samples is equivalent to delaying the TEQ output by $N + L_v - \varDelta$ samples in a causal system with the only difference of the number of leading zeros in the output of the S/P converter. In this thesis, the noncausal representation is often used to simplify the notation.

with $(\cdot)^T$ representing matrix transpose operation, and $\boldsymbol{C}$ is defined as

$$\boldsymbol{C} \triangleq \begin{bmatrix} c(0) & c(1) & \cdots & c(L_c-1) & 0 & \cdots & 0 \\ 0 & c(0) & c(1) & \cdots & c(L_c-1) & 0 & \cdots \\ \vdots & & \ddots & & & & \vdots \\ 0 & \cdots & 0 & c(0) & c(1) & \cdots & c(L_c-1) \end{bmatrix}. \tag{3.6}$$

In Figure 3.3, the output of the TEQ is

$$\bar{y}(n) = y(n) * w(n) = \sum_{k=0}^{L_w-1} w(k)y(n-k), \tag{3.7}$$

which can be rewritten in matrix form as

$$\bar{y}(n) = \boldsymbol{w}'\boldsymbol{y}(n), \tag{3.8}$$

where $(\cdot)'$ denotes the conjugate transpose operation, and

$$\boldsymbol{w} \triangleq \begin{bmatrix} w(0) & w(1) & \cdots & w(L_w-1) \end{bmatrix}'. \tag{3.9}$$

By substituting Eqn (3.2) into Eqn (3.8), one has

$$\bar{y}(n) = \boldsymbol{w}'\boldsymbol{C}\boldsymbol{x}(n) + \boldsymbol{w}'\boldsymbol{e}(n). \tag{3.10}$$

On the other hand, $\hat{y}(n)$ in Figure 3.3 can be expressed as

$$\hat{y}(n) = \sum_{k=0}^{L_b-1} b(k)x(n-k-\Delta), \tag{3.11}$$

or in matrix form, as

$$\hat{y}(n) = \boldsymbol{b}_E'\boldsymbol{x}(n), \tag{3.12}$$

where $\boldsymbol{b}_E$ is an extended version of the TIR vector $\boldsymbol{b}$, where

$$\boldsymbol{b} \triangleq \begin{bmatrix} b(0) & b(1) & \cdots & b(L_b-1) \end{bmatrix}', \tag{3.13}$$

$$\boldsymbol{b}_E \triangleq \begin{bmatrix} \boldsymbol{0}_{1\times\Delta} & \boldsymbol{b}' & \boldsymbol{0}_{1\times(L_c+L_w-\Delta-L_b)} \end{bmatrix}', \tag{3.14}$$

with the notation $\boldsymbol{0}_{n_r\times n_c}$ representing a n_r-by-n_c all-zero matrix.

Then, the approximation error $r(n)$ is

$$r(n) = \hat{y}(n) - \bar{y}(n) = \boldsymbol{b}_E'\boldsymbol{x}(n) - \boldsymbol{w}'\boldsymbol{C}\boldsymbol{x}(n) - \boldsymbol{w}'\boldsymbol{e}(n). \tag{3.15}$$

Assuming the channel noise $e(n)$ is independent of the channel input $x(n)$, the Mean-Squared Error (MSE), defined as MSE $\triangleq E\left[|r(n)|^2\right]$, becomes

$$\begin{aligned} \text{MSE} &= E\left[\left(\boldsymbol{b}_E'\boldsymbol{x}(n) - \boldsymbol{w}'\boldsymbol{C}\boldsymbol{x}(n) - \boldsymbol{w}'\boldsymbol{e}(n)\right)\left(\boldsymbol{b}_E'\boldsymbol{x}(n) - \boldsymbol{w}'\boldsymbol{C}\boldsymbol{x}(n) - \boldsymbol{w}'\boldsymbol{e}(n)\right)'\right] \\ &= \boldsymbol{b}_E'\boldsymbol{R}_x\boldsymbol{b}_E - \boldsymbol{b}_E'\boldsymbol{R}_x\boldsymbol{C}'\boldsymbol{w} - \boldsymbol{w}'\boldsymbol{C}\boldsymbol{R}_x\boldsymbol{b}_E + \boldsymbol{w}'\left(\boldsymbol{C}\boldsymbol{R}_x\boldsymbol{C}' + \boldsymbol{R}_e\right)\boldsymbol{w}, \end{aligned} \tag{3.16}$$

where $E[\cdot]$ represents the expectation operator, $\boldsymbol{R}_x$ and $\boldsymbol{R}_e$ represent the autocorrelation matrices of $x(n)$ and $e(n)$, respectively, and

$$\boldsymbol{R}_x \triangleq E\left[\boldsymbol{x}(n)\boldsymbol{x}(n)'\right] , \tag{3.17}$$

$$\boldsymbol{R}_e \triangleq E\left[\boldsymbol{e}(n)\boldsymbol{e}(n)'\right] . \tag{3.18}$$

Note that in Eqn (3.16), MSE is a scaler. In this way, each item in the RHS of Eqn (3.16) is also a scaler, and $\boldsymbol{b}_E'\boldsymbol{R}_x\boldsymbol{C}'\boldsymbol{w} = \boldsymbol{w}'\boldsymbol{C}\boldsymbol{R}_x\boldsymbol{b}_E$. Therefore, Eqn (3.16) can be written as

$$\text{MSE} = \boldsymbol{b}_E'\boldsymbol{R}_x\boldsymbol{b}_E - 2\boldsymbol{w}'\boldsymbol{C}\boldsymbol{R}_x\boldsymbol{b}_E + \boldsymbol{w}'\left(\boldsymbol{C}\boldsymbol{R}_x\boldsymbol{C}' + \boldsymbol{R}_e\right)\boldsymbol{w} . \tag{3.19}$$

The MSE is minimized when

$$\frac{\partial\left(E\left[|r(n)|^2\right]\right)}{\partial \boldsymbol{w}} = \boldsymbol{0} , \tag{3.20}$$

which leads to

$$\boldsymbol{C}\boldsymbol{R}_x\boldsymbol{b}_E = \left(\boldsymbol{C}\boldsymbol{R}_x\boldsymbol{C}' + \boldsymbol{R}_e\right)\boldsymbol{w} . \tag{3.21}$$

By substituting Eqn (3.21) into Eqn (3.19), the minimized MSE becomes

$$\text{MSE} = \boldsymbol{b}_E'\boldsymbol{R}_\perp\boldsymbol{b}_E , \tag{3.22}$$

where

$$\boldsymbol{R}_\perp = \boldsymbol{R}_x - \boldsymbol{R}_x\boldsymbol{C}'\left(\boldsymbol{C}\boldsymbol{R}_x\boldsymbol{C}' + \boldsymbol{R}_e\right)^{-1}\boldsymbol{C}\boldsymbol{R}_x , \tag{3.23}$$

with $(\cdot)^{-1}$ representing the matrix inverse operation.

Note that the position of $\boldsymbol{b}$ in its extended version $\boldsymbol{b}_E$ depends on the decision delay $\varDelta$. In order to explicitly use $\boldsymbol{b}$ to calculate the MSE, let us define $\boldsymbol{R}_\varDelta$ as a L_b-by-L_b sub-matrix of $\boldsymbol{R}_\perp$ starting from the $\varDelta$-th row and $\varDelta$-th column of $\boldsymbol{R}_\perp$, i.e. let

$$\boldsymbol{R}_\varDelta \triangleq \begin{bmatrix}\boldsymbol{0}_{L_b\times\varDelta} & \boldsymbol{I}_{L_b} & \boldsymbol{0}_{L_b\times(L_c+L_w-\varDelta-L_b)}\end{bmatrix}\boldsymbol{R}_\perp\begin{bmatrix}\boldsymbol{0}_{\varDelta\times L_b}\\ \boldsymbol{I}_{L_b}\\ \boldsymbol{0}_{(L_c+L_w-\varDelta-L_b)\times L_b}\end{bmatrix} , \tag{3.24}$$

where $\boldsymbol{I}_{L_b}$ represents a L_b-by-L_b unity matrix. Then, Eqn (3.22) becomes

$$\text{MSE} = \boldsymbol{b}'\boldsymbol{R}_\varDelta\boldsymbol{b} , \tag{3.25}$$

where $\boldsymbol{R}_\varDelta$ is a positive-definite matrix because MSE > 0 in practical (noisy) situations.

A constraint on $\boldsymbol{b}$ is needed to avoid the trivial solution $\boldsymbol{b} = \boldsymbol{w} = 0$. Either the Unit-Tap Constraint (UTC) or Unit-Energy Constraint (UEC) can be applied to the coefficients of $\boldsymbol{b}$. The UTC is that one tap of $\boldsymbol{b}$ is selected as the reference tap and set to unity, i.e. $\boldsymbol{b}'\boldsymbol{u}_i = 1$ with $\boldsymbol{u}_i$ representing the i-th unit vector ($\boldsymbol{u}_i = \begin{bmatrix}0 & \cdots & 1 & \cdots & 0\end{bmatrix}^T$, where the i-th element of $\boldsymbol{u}_i$ is equal to 1 while other elements are 0). On the other hand, the UEC is that the energy of $\boldsymbol{b}$ is set to unity, i.e. $\boldsymbol{b}'\boldsymbol{b} = 1$.

Before introducing the optimal TEQ solution under UTC or UEC constraint, one has to note that the optimal solution highly depends on the choice of the decision delay Δ. To find the best choice of Δ that leads to the minimum MSE, an exhaustive search of Δ is needed, which increases the training time of the TEQ. On the other hand, heuristic methods for the determination of Δ can be used to simplify the search of Δ. In [36], a windowing method was used to estimate the optimal Δ in accordance with

$$\Delta_{\text{ratio}} = \arg\max_{\Delta} R_E(\Delta), \tag{3.26}$$

where

$$R_E(\Delta) \triangleq \frac{\text{energy inside a window}}{\text{energy outside a window}}, \tag{3.27}$$

with the window of length $(L_v + 1)$ samples beginning at the index $(\Delta + 1)$.

3.1.2 Optimal TEQ Solution under Unit-Tap Constraint

The Lagrangian function can be formed to minimize the MSE subject to the UTC constraint:

$$L^{\text{UTC}}(\boldsymbol{b}, \lambda) = \boldsymbol{b}'\boldsymbol{R}_\Delta\boldsymbol{b} + \lambda(\boldsymbol{b}'\boldsymbol{u}_i - 1). \tag{3.28}$$

The minimum of $L^{\text{UTC}}(\boldsymbol{b}, \lambda)$ can be found by letting its partial derivatives with respect to $\boldsymbol{b}$ and λ equal to 0,

$$\begin{cases} \frac{\partial L^{\text{UTC}}(\boldsymbol{b},\lambda)}{\partial \boldsymbol{b}} = \boldsymbol{0}, \\ \frac{\partial L^{\text{UTC}}(\boldsymbol{b},\lambda)}{\partial \lambda} = \boldsymbol{0}. \end{cases} \tag{3.29}$$

Then, we have

$$\begin{cases} 2\boldsymbol{R}_\Delta\boldsymbol{b} = -\lambda\boldsymbol{u}_i, \\ \boldsymbol{u}_i'\boldsymbol{b} = 1. \end{cases} \tag{3.30}$$

By solving the above simultaneous equations, the optimal TIR coefficients and the MMSE are given by

$$\boldsymbol{b}_{\text{opt.}} = \frac{\boldsymbol{R}_\Delta^{-1}\boldsymbol{u}_{i_{\text{opt.}}}}{\boldsymbol{R}_\Delta^{-1}(i_{\text{opt.}}, i_{\text{opt.}})}, \tag{3.31}$$

$$\text{MMSE}^{\text{UTC}} = \frac{1}{\boldsymbol{R}_\Delta^{-1}(i_{\text{opt.}}, i_{\text{opt.}})}, \tag{3.32}$$

where $\boldsymbol{R}_\Delta^{-1}(i_{\text{opt.}}, i_{\text{opt.}})$ represents the element of $\boldsymbol{R}_\Delta^{-1}$ in the $i_{\text{opt.}}$-th row and $i_{\text{opt.}}$-th column, and $i_{\text{opt.}}$ is the index that achieves the MMSE, i.e.

$$i_{\text{opt.}} = \arg\max_{0\le i\le L_b} \left\{\boldsymbol{R}_\Delta^{-1}(i,i)\right\}. \tag{3.33}$$

From Eqn (3.21), the optimal coefficients of the TEQ $\boldsymbol{w}_{\text{opt.}}$ are

$$\boldsymbol{w}_{\text{opt.}} = \left(\boldsymbol{C}\boldsymbol{R}_x\boldsymbol{C}' + \boldsymbol{R}_e\right)^{-1}\boldsymbol{C}\boldsymbol{R}_x\boldsymbol{b}_E\,. \tag{3.34}$$

3.1.3 Optimal TEQ Solution under Unit-Energy Constraint

When minimizing the MSE subject to the UEC constraint ($\boldsymbol{b}'\boldsymbol{b} = 1$), the Lagrangian function can be formed as

$$L^{\text{UEC}}(\boldsymbol{b}, \lambda) = \boldsymbol{b}'\boldsymbol{R}_{\Delta}\boldsymbol{b} - \lambda(\boldsymbol{b}'\boldsymbol{b} - 1)\,. \tag{3.35}$$

By letting the partial derivatives of $L^{\text{UEC}}(\boldsymbol{b}, \lambda)$ with respect to $\boldsymbol{b}$ and λ equal to 0, one has

$$\begin{cases} \boldsymbol{R}_{\Delta}\boldsymbol{b} = \lambda\boldsymbol{b}\,, \\ \boldsymbol{b}'\boldsymbol{b} = 1\,, \end{cases} \tag{3.36}$$

which means that $\boldsymbol{b}_{\text{opt.}}$ is an eigenvector of $\boldsymbol{R}_{\Delta}$. Then, the MSE becomes

$$\text{MSE} = \boldsymbol{b}'\boldsymbol{R}_{\Delta}\boldsymbol{b} = \lambda\boldsymbol{b}'\boldsymbol{b} = \lambda\,. \tag{3.37}$$

Now the MMSE is equal to the minimum eigenvalue (denoted as $\lambda_{\text{min.}}$) of $\boldsymbol{R}_{\Delta}$, i.e.

$$\text{MMSE}^{\text{UEC}} = \lambda_{\text{min.}}\,. \tag{3.38}$$

Then, the optimal TIR $\boldsymbol{b}_{\text{opt.}}$ is equal to the eigenvector corresponding to $\lambda_{\text{min.}}$. The closed form of the optimal TEQ $\boldsymbol{w}_{\text{opt.}}$ is the same as in Eqn (3.34).

It can be shown that $\text{MMSE}^{\text{UEC}} \leq \text{MMSE}^{\text{UTC}}$. When the channel is memoryless and the input and the channel noise are white, all the eigenvalues of $\boldsymbol{R}_{\Delta}$ are equal and $\text{MMSE}^{\text{UEC}} = \text{MMSE}^{\text{UTC}}$, which means the TEQ achieves the MFB under either the UTC or the UEC (See [2]).

3.1.4 Geometric SNR Method

An equalization method based on Geometric SNR was presented in [3] for the maximization of the bit-rate of the conventional DMT system. The total number of bits transmitted in one symbol is given by

$$\begin{aligned} b &= \sum_{i=1}^{N/2} b_i = \sum_{i=1}^{N/2} \log_2\left(1 + \frac{\text{SNR}_i}{\mathit{\Gamma}_i}\right) \\ &= \frac{N}{2}\log_2\left(1 + \frac{\text{SNR}_{\text{geom}}}{\mathit{\Gamma}}\right)\,, \end{aligned} \tag{3.39}$$

where SNR_i represents the SNR of the i-th subchannel, Γ_i represents the SNR gap for the i-th subchannel, and $\Gamma_i = \Gamma$ for all i (meaning that all subchannels have the same error probability). Moreover, SNR_{geom} represents the geometric SNR given by

$$\text{SNR}_{\text{geom}} \triangleq \Gamma \left\{ \left[\prod_{i=1}^{N/2} \left(1 + \frac{\text{SNR}_i}{\Gamma} \right) \right]^{2/N} - 1 \right\} . \tag{3.40}$$

Obviously, when SNR_i is high enough for all i, SNR_{geom} can be approximated as

$$\text{SNR}_{\text{geom}} \approx \left[\prod_{i=1}^{N/2} (\text{SNR}_i) \right]^{2/N} . \tag{3.41}$$

Therefore, the maximum bit-rate is achieved by maximizing SNR_{geom}, which entails a nonlinear optimization problem.

In [6], an improvement to the geometric SNR was made by incorporating the ISI term in SNR_i in accordance with

$$\text{SNR}_i = \frac{S_{x,i} \left| H_{\text{signal},i} \right|^2}{S_{n,i} \left| H_{\text{noise},i} \right|^2 + S_{x,i} \left| H_{\text{ISI},i} \right|^2} , \tag{3.42}$$

where $S_{x,i}$, $S_{n,i}$, $H_{\text{signal},i}$, $H_{\text{ISI},i}$ are the signal power, noise power, signal path gain, noise path gain, and ISI path gain in the i-th subchannel, respectively.

3.1.5 Shortening Signal-to-Noise Ratio Method

Melsa *et al* [39] use a different argument to measure the effectiveness of the channel impulse response shortening. Recall that the shortened channel impulse response $b(n)$ is the convolution of the channel impulse response $c(n)$ and the TEQ $w(n)$, and that perfect channel shortening may not be possible in practical situations. Therefore, define the Shortening Signal-to-Noise Ratio (SSNR) as

$$\text{SSNR} \triangleq 10 \log_{10} \frac{\text{energy in } b_{\text{win}}(n)}{\text{energy in } b_{\text{wall}}(n)} , \tag{3.43}$$

where $b_{\text{win}}(n)$ is a windowed part of $b(n)$, beginning at the Δ-th sample of $b(n)$, and having a length of $L_v + 1$. Moreover, $b_{\text{wall}}(n)$ denotes the remainder of $b(n)$ (all zeros for perfect channel shortening).

The main objective is to maximize the SSNR subject to a unit-energy constraint on $b_{\text{win}}(n)$ (so as to avoid the trivial solution $w(n) = 0$). This, in turn, is equivalent to minimizing the energy in $b_{\text{wall}}(n)$. By denoting $b_{\text{wall}}(n)$ and $w(n)$ in matrix form as $\boldsymbol{b}_{\text{wall}}$ and $\boldsymbol{w}$, respectively, the constrained objective function is

$$\min_{\boldsymbol{w}} \left\{ \boldsymbol{b}_{\text{wall}}^T \boldsymbol{b}_{\text{wall}} \right\} \quad \text{s.t. } \boldsymbol{b}_{\text{win}}^T \boldsymbol{b}_{\text{win}} = 1 . \tag{3.44}$$

The SSNR method has the same disadvantage as the MMSE method, for example, the highest bit-rate capacity may not be achieved.

3.1.6 Main Disadvantages of TEQ

TEQ technique has the advantage of simple structure and relatively low computation complexity. However, it has several disadvantages listed as follows.

1. As discussed in Section 3.1, generally, the channel cannot fully be equalized by a FIR TEQ. Therefore, TEQ technique can only be used to shorten the channel impulse response, and other techniques (for example, the cyclic prefix) have to be used together with TEQ to recover the system input.

2. Before of the same reason, when a FIR TEQ is used to shorten the channel, the energy leakage (i.e. the samples outside the TIR window in the equalized channel impulse response) causes the ISI and ICI remain in the equalized channel. The remaining ISI and ICI would not cause any severe problem in the conventional DMT system because redundancy (the cyclic prefix) is introduced in the system. However, these remaining interferences may lead to severe distortion in DMT systems using critically decimated filterbanks.

3. The TEQ technique equalizes the whole channel frequency band by using a single equalizer. Then, the characteristics of the equalized channel is the sum of the characteristics of the whole channel frequency band (weighted by the TEQ). Therefore, a severely distorted channel frequency band would affect the equalization of other channel frequency band, and lower the overall equalization effect.

4. Usually, the optimization of TEQ requires an UTC or UEC constraint. However, either constraint has no relation to the system transmission bit-rate.

5. The optimization of TEQ is based on maximizing the SNR at the TEQ output. However, this objective has no direct relation to the system transmission bit-rate. Therefore, the highest transmission bit-rate may not be achieved when the SNR of TEQ output is maximized.

6. The equalized channel frequency response may have some notches, which do not exist in the original channel frequency response.

3.2 Per Tone Equalization in Conventional DMT system

The per tone equalization was introduced by K. Van Acker *et al.* [57] for channel equalization of the conventional DMT system. It modifies the receiver structure of the conventional DMT system such that the TEQ operation is transferred to the frequency-domain, and leads to a post-detection equalization technique that equalizes each subchannel separately. While it is only a modification of the conventional DMT receiver, the cyclic prefix is still needed in the whole system.

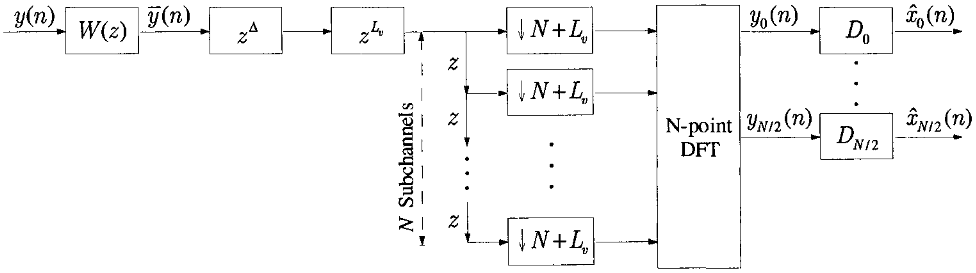


Figure 3.4: Explicit form of the conventional DMT receiver

The explicit form of the conventional DMT receiver (with TEQ) is redrawn in Figure 3.4, where $W(z)$ represents the TEQ and D_0, ..., $D_{N/2}$ represent the $N/2+1$ 1-tap FEQs. The advance component z^{Δ} is used to compensate the decision delay. Compared to Figure 2.2, the S/P converter is replaced by an advance chain and a set of decimators. The QAM decoder and the P/S converter are disregarded here to concentrate on the channel equalization and the relationship between the TEQ and the per tone equalization.

The output of the i-th FEQ can be expressed as

$$\begin{aligned}\hat{x}_i(n) &= D_i y_i(n) = D_i \sum_{k=0}^{N-1} \bar{y}(N_E n + L_v + \Delta + k) e^{-j2\pi ki/N} \\ &= D_i \begin{bmatrix} 1 & e^{-j2\pi i/N} & \cdots & e^{-j2\pi(N-1)i/N} \end{bmatrix} \begin{bmatrix} \bar{y}(N_E n + L_v + \Delta) \\ \bar{y}(N_E n + L_v + \Delta + 1) \\ \vdots \\ \bar{y}(N_E n + L_v + \Delta + N - 1) \end{bmatrix},\end{aligned} \tag{3.45}$$

where $N_E = N + L_v$. The TEQ output can also be written in matrix form as

$$\begin{aligned}\bar{y}(n) &= \sum_{k=-\infty}^{\infty} y(n-k)w(k) = \sum_{k=0}^{L_w-1} y(n-k)w(k) \\ &= \begin{bmatrix} y(n) & y(n-1) & \cdots & y(n-L_w+1) \end{bmatrix} \begin{bmatrix} w(0) \\ w(1) \\ \vdots \\ w(L_w-1) \end{bmatrix}.\end{aligned} \tag{3.46}$$

Substituting Eqn (3.46) into Eqn (3.45), one has

$$\begin{aligned}\hat{x}_i(n) &= D_i \begin{bmatrix}1 & e^{-j2\pi i/N} & \cdots & e^{-j2\pi(N-1)i/N}\end{bmatrix} (\boldsymbol{Y}\boldsymbol{w}) \\ &= D_i \mathrm{row}_i\{\mathcal{F}_N\}(\boldsymbol{Y}\boldsymbol{w}) \\ &= D_i \mathrm{row}_i\{\underbrace{\mathcal{F}_N(\boldsymbol{Y}\boldsymbol{w})}_{1 \text{ FFT}}\} ,\end{aligned} \tag{3.47}$$

where $\mathrm{row}_i\{\cdot\}$ represents the i-th row of the matrix $(\cdot)$, and $\mathcal{F}_N$, $\boldsymbol{Y}$, and $\boldsymbol{w}$ represent the N-by-N DFT matrix, the N-by-L_w Toeplitz matrix of the received symbols, and the TEQ vector, in accordance with

$$\mathcal{F}_N \triangleq \begin{bmatrix}1 & 1 & \cdots & 1 \\ 1 & e^{-j2\pi/N} & \cdots & e^{-j2\pi(N-1)/N} \\ \vdots & \vdots & & \vdots \\ 1 & e^{-j2\pi(N-1)/N} & \cdots & e^{-j2\pi(N-1)(N-1)/N}\end{bmatrix} , \tag{3.48}$$

$$\boldsymbol{Y} \triangleq \begin{bmatrix} y(N_E n + L_v + \Delta) & \cdots & y(N_E n + L_v + \Delta - L_w + 1) \\ y(N_E n + L_v + \Delta + 1) & \cdots & y(N_E n + L_v + \Delta - L_w + 2) \\ \vdots & \ddots & \vdots \\ y(N_E(n+1) + \Delta - 1) & \cdots & y(N_E(n+1) + \Delta - L_w) \end{bmatrix} , \tag{3.49}$$

and

$$\boldsymbol{w} \triangleq \begin{bmatrix}w(0) & w(1) & \cdots & w(L_w - 1)\end{bmatrix}^T . \tag{3.50}$$

It is always possible to exchange the positions of the TEQ and the FFT processor in Figure(3.4). By changing the computing sequence of Eqn (3.47), the TEQ $\boldsymbol{w}$ can be moved to the frequency-domain and combined with the original FEQ D_i to form a per tone equalizer $\boldsymbol{w}_i$ in accordance with

$$\begin{aligned}\hat{x}_i(n) &= D_i \mathrm{row}_i\{\mathcal{F}_N\}(\boldsymbol{Y}\boldsymbol{w}) \\ &= \mathrm{row}_i\{\mathcal{F}_N \boldsymbol{Y}\}(D_i \boldsymbol{w}) \\ &= \mathrm{row}_i \underbrace{\{\mathcal{F}_N \boldsymbol{Y}\}}_{L_w \text{ FFT's}} \boldsymbol{w}_i ,\end{aligned} \tag{3.51}$$

where $\boldsymbol{w}_i \triangleq D_i \boldsymbol{w}$ is a (complex) L_w-tap per tone equalizer for the i-th subchannel. With this modification, the per tone equalization is achieved with the cost of L_w N-point DFTs (instead of the original one N-point DFT). The modified receiver incorporating with per tone equalizers is shown in Figure 3.5. Note that this figure is modified from Fig.3 in [57] to give hints of the relationship between the per tone equalization and the proposed interpolation equalization.

The structure in Figure 3.5 can be further simplified to employ one N-point DFT with additional symbol differences. By decomposing the matrix $\boldsymbol{Y}$ into a column-circulant matrix

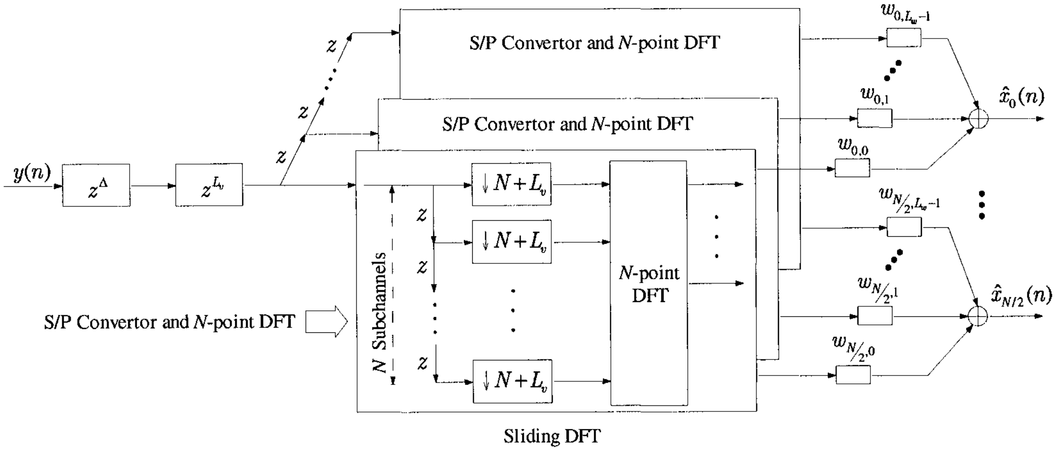


Figure 3.5: Per tone equalization with sliding DFT

and a difference matrix, one can arrive at a new set of per tone equalizer coefficients $\boldsymbol{v}_i = \begin{bmatrix} v_{i,0} & \cdots & v_{i,L_w-1} \end{bmatrix}^T$ for per tone equalization in accordance with

$$\begin{bmatrix} v_{i,0} \\ v_{i,1} \\ \vdots \\ v_{i,L_w-1} \end{bmatrix} = \begin{bmatrix} 1 & \alpha^{i-1} & \cdots & \alpha^{(i-1)(L_w-1)} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \alpha^{i-1} \\ 0 & \cdots & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} w_{i,0} \\ w_{i,1} \\ \vdots \\ w_{i,L_w-1} \end{bmatrix}, \tag{3.52}$$

where $\alpha = e^{-j2\pi/N}$. Then, Eqn (3.51) can be rewritten as

$$\hat{x}_i(n) = \bar{\boldsymbol{v}}_i^T \underbrace{\left[\begin{array}{c|c} \boldsymbol{I}_{L_w-1} & \begin{array}{c|c} \boldsymbol{0} & -\boldsymbol{I}_{L_w-1} \end{array} \\ \hline \boldsymbol{0} & \mathcal{F}_N(i,:) \end{array} \right]}_{\tilde{\mathcal{F}}_i} \boldsymbol{y}, \tag{3.53}$$

where $\bar{\boldsymbol{v}}_i^T = \begin{bmatrix} v_{i,L_w-1} & \cdots & v_{i,0} \end{bmatrix}$, $\boldsymbol{I}_{L_w-1}$ is the $(L_w-1) \times (L_w-1)$ unity matrix, $\mathcal{F}_N(i,:)$ is the i-th row of $N \times N$ DFT matrix, and

$$\boldsymbol{y} = \begin{bmatrix} y(N_E n + L_v + \Delta - L_w + 1) & \cdots & y(N_E(n+1) + \Delta - 1) \end{bmatrix}^T . \tag{3.54}$$

Therefore, the calculation of $\hat{x}_i(n)$ involves one DFT calculation (c.f. $\mathcal{F}_N(i,:)$) with additional symbol differences (c.f. the first block row in $\tilde{\mathcal{F}}_i$ multiplying with $\boldsymbol{y}$), as implemented in the block diagram shown in Figure 3.6.

In this way, the per tone equalizer coefficients can be optimized separately with the MMSE objective function defined as

$$\min_{\boldsymbol{v}_i} \left\{ E\left[\left| \hat{x}_i(n) - x_i(n) \right|^2 \right] \right\} . \tag{3.55}$$

The per tone equalization exhibits superior performance in terms of the bit-rate as compared to schemes employing TEQ. In addition, it exhibits a smooth bit-rate distribution as a

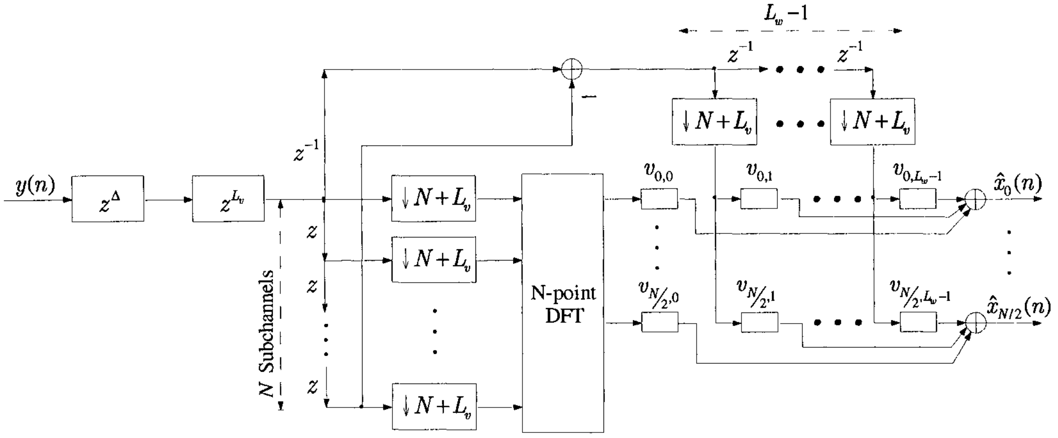


Figure 3.6: Per tone equalization with symbol differences

function of Δ, making it possible to achieve the optimum capacity easily. The computational complexity of the per tone equalization is comparable to that of TEQ schemes and can be further simplified by tone grouping [57].

3.3 Output Combiner in DMT Systems Employing Critically Decimated Filterbanks

The output combiner proposed by S. D. Sandberg and M. A. Tzannes [45] is a type of post-detection equalization technique for the DWMT system. This equalization technique can also be used in other DMT systems using critically decimated filterbanks. The block diagram of the DWMT system incorporating with output combiners is shown in Figure 3.7, where OC$_0$, ..., OC$_{M-1}$ represent output combiners.

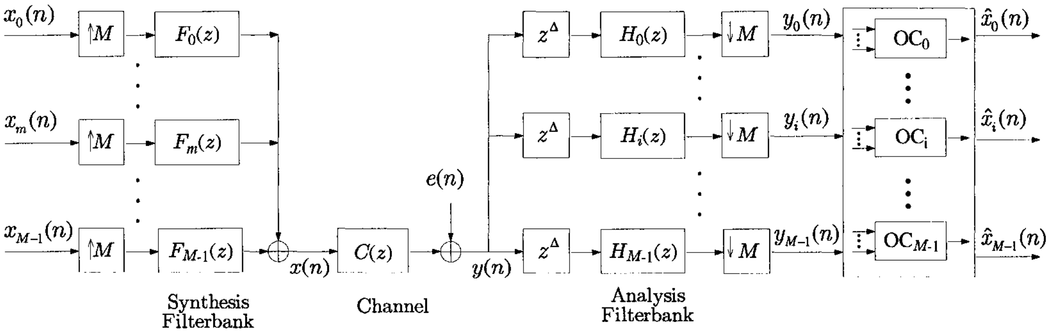


Figure 3.7: DWMT system incorporating output combiners

For a given subchannel, the idea of the output combiner is based on combining the outputs of several neighboring subchannels to recover the input symbols of the given subchannel. The block diagram of the output combiner used in the i-th subchannel ($i = 0, ..., M-1$) is shown in Figure 3.8. In this figure, the i-th subchannel and its $2N_o$ neighboring subchannels (symmetrically centered around the i-th subchannel, i.e. subchannels with indices of $\{i - N_o, ..., i + N_o\}$) are used to recover the system input[4], where the output of the decimator in each subchannel is filtered by an equalizer $W_{i,k}(z)$ ($k = i - N_o, ..., i + N_o$), and then, these $2N_o + 1$ filtered outputs are summed together to form the recovered symbols.

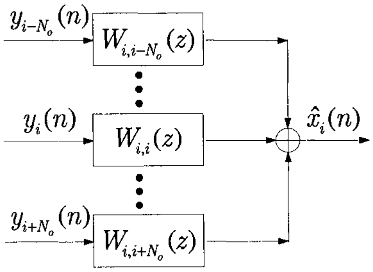


Figure 3.8: Block diagram of the i-th output combiner OC_i

Originally in [45], a TEQ equalizer (not shown in Figure 3.8) is also used for performance comparison of the DWMT and the conventional DMT systems. The advance component[5] z^{Δ} used in each subchannel of Figure 3.8 was employed in [45] to compensate the decision delay Δ of TEQ. Because the TEQ is not a necessary part of the output combiner, it is ignored in Figure 3.8. However, our experiments show that the advance component z^{Δ} strongly affects the SNR of the output combiner regardless of the use of TEQ. Therefore, it cannot be ignored even if the TEQ is not used.

The use of TEQ in [45] makes it become a tedious procedure to find the optimal output combiner settings. In this section, the TEQ is neglected and the procedure of finding the optimal solution is rewritten in a more straightforward manner. If for some reasons a TEQ is needed, one can still use the conclusion of this section by simply replacing the channel $C(z)$ with the TIR (c.f. section 3.1.1 for the definition of TIR).

[4] Here, we say $2N_o$ neighboring subchannels only for the sake of simplifying the notation. Note that some subchannels do not have $2N_o$ neighbors. Therefore, precisely, the neighboring subchannels used for the equalization of the i-th subchannel are subchannels with indices of $\{\max\{i-N_o, 0\}, ..., \min\{i+N_o, M-1\}\}$.

[5] Actually, [45] employed a delay component $z^{\Delta-gM}$, which is the causal counterpart of the advance component z^{Δ}.

First, from Figure 3.7, $y_i(n)$ (where $i = 0, ..., M-1$) can be expressed as

$$\begin{aligned} y_i(n) &= \sum_{m=0}^{M-1} x_m(n) * (f_m(n) * c(n) * \delta(n+\Delta) * h_i(n))_{\downarrow M} \\ &\quad + (e(n) * \delta(n+\Delta) * h_i(n))_{\downarrow M} \\ &= \sum_{m=0}^{M-1} x_m(n) * s_{mi}^{(\Delta)}(n) + (e(n+\Delta) * h_i(n))_{\downarrow M} \,, \end{aligned} \tag{3.56}$$

where $s_{mi}^{(\Delta)}(n)$ is the Δ-th polyphase component of $s_{mi}(n)$, and $s_{mi}(n)$ is defined in Eqn (2.29). Next, from Figure 3.8, the recovered symbols of the i-th subchannel can be written as

$$\hat{x}_i(n) = \sum_{k=i-N_o}^{i+N_o} y_k(n) * w_{i,k}(n) \,, \tag{3.57}$$

where $w_{i,k}(n)$ represents the impulse response of the k-th equalizer in the i-th output combiner. Then, by substituting Eqn (3.56) into Eqn (3.57), one has

$$\begin{aligned} \hat{x}_i(n) = &\sum_{k=i-N_o}^{i+N_o} \sum_{m=0}^{M-1} x_m(n) * s_{mk}^{(\Delta)}(n) * w_{i,k}(n) \\ &+ \sum_{k=i-N_o}^{i+N_o} (e(n+\Delta) * h_k(n))_{\downarrow M} * w_{i,k}(n) \,. \end{aligned} \tag{3.58}$$

Suppose all equalizers $\{w_{i,k}(n)\}$ ($k = i - N_o, ..., i + N_o$, $i = 0, ..., M-1$) have the same length L_w, and all $\{h_k(n\})$ have the same length L_h. Then Eqn (3.58) can be rewritten as

$$\begin{aligned} \hat{x}_i(n) &= \sum_{k=i-N_o}^{i+N_o} \sum_{m=0}^{M-1} \sum_{t=0}^{L_w-1} \sum_{l=-\infty}^{\infty} x_m(l) s_{mk}^{(\Delta)}(n-l-t) w_{i,k}(t) \\ &\quad + \sum_{k=i-N_o}^{i+N_o} \sum_{l=-\infty}^{\infty} \sum_{t=0}^{L_w-1} e(l) h_k(M(n-t)+\Delta-l) w_{i,k}(t) \\ &= \sum_{m=0}^{M-1} \sum_{l=-\infty}^{\infty} x_m(l) \sum_{k=i-N_o}^{i+N_o} \sum_{t=0}^{L_w-1} s_{mk}^{(\Delta)}(n-l-t) w_{i,k}(t) \\ &\quad + \sum_{k=i-N_o}^{i+N_o} \sum_{t=0}^{L_w-1} w_{i,k}(t) \sum_{l=-\infty}^{\infty} e(l) h_k(M(n-t)+\Delta-l) \,. \end{aligned} \tag{3.59}$$

The first term in the RHS of Eqn (3.59) represents the signal components (denoted as $\hat{x}_{i,s}(n)$) in the recovered symbols while the second term represents the noise component (denoted as $\hat{x}_{i,n}(n)$).

First, consider the signal component $\hat{x}_{i,s}(n)$. It can be expressed in matrix form as

$$\hat{x}_{i,s}(n) = \sum_{m=0}^{M-1} \sum_{l=-\infty}^{\infty} x_m(l) \sum_{k=i-N_o}^{i+N_o} \boldsymbol{w}_{i,k}' \boldsymbol{s}_{mk}(n-l) \,, \tag{3.60}$$

where $\boldsymbol{w}_{i,k}$ and $\boldsymbol{s}_{mk}(n-l)$ are defined as

$$\boldsymbol{w}_{i,k} = \begin{bmatrix} w_{i,k}(0) \\ w_{i,k}(1) \\ \vdots \\ w_{i,k}(L_w-1) \end{bmatrix}, \; \boldsymbol{s}_{mk}(n-l) = \begin{bmatrix} s_{mk}^{(\Delta)}(n-l) \\ s_{mk}^{(\Delta)}(n-l-1) \\ \vdots \\ s_{mk}^{(\Delta)}(n-l-L_w+1) \end{bmatrix}. \tag{3.61}$$

Eqn (3.60) can further be written in compact form as

$$\hat{x}_{i,s}(n) = \sum_{m=0}^{M-1} \sum_{l=-\infty}^{\infty} x_m(l) \boldsymbol{w}_i' \boldsymbol{s}_m(n-l)\,, \tag{3.62}$$

where $\boldsymbol{w}_i$ and $\boldsymbol{s}_m(n-l)$ are two new vectors that contain all $\boldsymbol{w}_{i,k}$ and $\boldsymbol{s}_{mk}(n-l)$, respectively, in accordance with

$$\boldsymbol{w}_i = \begin{bmatrix} \boldsymbol{w}_{i,i-N_o} \\ \boldsymbol{w}_{i,i-N_o+1} \\ \vdots \\ \boldsymbol{w}_{i,i+N_o} \end{bmatrix}, \; \boldsymbol{s}_m(n-l) = \begin{bmatrix} \boldsymbol{s}_{m(i-N_o)}(n-l) \\ \boldsymbol{s}_{m(i-N_o+1)}(n-l) \\ \vdots \\ \boldsymbol{s}_{m(i+N_o)}(n-l) \end{bmatrix}. \tag{3.63}$$

By defining $k = n - l$, Eqn (3.62) becomes

$$\hat{x}_{i,s}(n) = \sum_{m=0}^{M-1} \sum_{k=-\infty}^{\infty} x_m(n-k) \boldsymbol{w}_i' \boldsymbol{s}_m(k)\,. \tag{3.64}$$

Now, suppose all $\{s_{mk}^{(\Delta)}(n)\}$ have the same length L_s. Then, note that $\boldsymbol{s}_m(k)$ becomes all-zero matrix when $k < 0$ and $k > L_s + L_w - 2$, therefore, the infinite range of k in Eqn (3.64) can be confined to a finite range $0 \le k \le L_s + L_w - 2$, and Eqn (3.64) becomes

$$\begin{aligned} \hat{x}_{i,s}(n) &= \sum_{m=0}^{M-1} \sum_{k=0}^{L_s+L_w-2} x_m(n-k) \boldsymbol{w}_i' \boldsymbol{s}_m(k) \\ &= x_i(n-k_i) \boldsymbol{w}_i' \boldsymbol{s}_i(k_i) + \sum_{\substack{m=0 \\ m \neq i}}^{M-1} \sum_{\substack{k=0 \\ k \neq k_i}}^{L_s+L_w-2} x_m(n-k) \boldsymbol{w}_i' \boldsymbol{s}_m(k)\,, \end{aligned} \tag{3.65}$$

where k_i is the relative delay between the recovered symbols and the corresponding system input symbols. In Eqn (3.65), the first term of the RHS is the desired symbol while the second term represents interferences (both ICI and ISI).

Next, consider the noise component $\hat{x}_{i,n}(n)$. It can be written in matrix form as

$$\begin{aligned}\hat{x}_{i,n}(n) &= \sum_{k=i-N_o}^{i+N_o} \sum_{t=0}^{L_w-1} w_{i,k}(t) \sum_{l=-\infty}^{\infty} e(l) h_k(M(n-t)+\Delta-l) \\ &= \sum_{k=i-N_o}^{i+N_o} \boldsymbol{w}_{i,k}' \boldsymbol{q}_k(n)\,,\end{aligned} \tag{3.66}$$

where $\boldsymbol{w}_{i,k}$ is defined in Eqn (3.61), and $\boldsymbol{q}_k(n)$ is defined as

$$\boldsymbol{q}_k(n) = \begin{bmatrix} \sum_{l=-\infty}^{\infty} e(l) h_k(Mn+\Delta-l) \\ \sum_{l=-\infty}^{\infty} e(l) h_k(M(n-1)+\Delta-l) \\ \vdots \\ \sum_{l=-\infty}^{\infty} e(l) h_k(M(n-L_w+1)+\Delta-l) \end{bmatrix}.$$

Eqn (3.66) can further be written in compact form as

$$\hat{x}_{i,n}(n) = \boldsymbol{w}_i' \boldsymbol{q}(n)\,, \tag{3.67}$$

where $\boldsymbol{w}_i$ is defined in Eqn (3.63), and $\boldsymbol{q}(n)$ is a new vector that contains all $\boldsymbol{q}_k(n)$ in accordance with

$$\boldsymbol{q}(n) = \begin{bmatrix} \boldsymbol{q}_{i-N_o}(n) \\ \boldsymbol{q}_{i-N_o+1}(n) \\ \vdots \\ \boldsymbol{q}_{i+N_o}(n) \end{bmatrix}.$$

Now, substitute Eqns (3.65) and (3.67) in to Eqn (3.59), one has

$$\begin{aligned}\hat{x}_i(n) = x_i(n-k_i) \boldsymbol{w}_i' \boldsymbol{s}_i(k_i) + \sum_{\substack{m=0 \\ m\neq i}}^{M-1} \sum_{\substack{k=0 \\ k\neq k_i}}^{L_s+L_w-2} x_m(n-k) \boldsymbol{w}_i' \boldsymbol{s}_m(k) \\ + \boldsymbol{w}_i' \boldsymbol{q}(n)\,.\end{aligned} \tag{3.68}$$

Then, with the assumption that $\{x_i(n)\}$ are mutually uncorrelated for any i and n, and $x_i(n)$ is uncorrelated to the noise $e(n)$, by using the MMSE criterion, the SNR in $\hat{x}_i(n)$ becomes

$$\mathrm{SNR}_i = \frac{|\boldsymbol{w}_i' \boldsymbol{s}_i(k_i)|^2}{\sum_{\substack{m=0 \\ m\neq i}}^{M-1} \sum_{\substack{k=0 \\ k\neq k_i}}^{L_s+L_w-2} |\boldsymbol{w}_i' \boldsymbol{s}_m(k)|^2 + E\left[|\hat{x}_{i,n}(n)|^2\right]}\,, \tag{3.69}$$

where $E\left[|\hat{x}_{i,n}(n)|^2\right] = E\left[|\boldsymbol{w}_i' \boldsymbol{q}(n)|^2\right]$ is the autocorrelation of the noise component $\hat{x}_{i,n}(n)$.

Note that

$$E\left[|\hat{x}_{i,n}(n)|^2\right] = E\left[\boldsymbol{w}_i' \boldsymbol{q}(n) \boldsymbol{q}(n)' \boldsymbol{w}_i\right] = \boldsymbol{w}_i' \boldsymbol{C} \boldsymbol{w}_i\,, \tag{3.70}$$

where $\boldsymbol{C}$ is the covariance matrix of vector $\boldsymbol{q}(n)$ defined as

$$\boldsymbol{C} = E\left[\boldsymbol{q}(n)\boldsymbol{q}(n)'\right] .$$

and its element $\boldsymbol{C}_{i_1,i_2}$ can be calculated by

$$\boldsymbol{C}_{i_1,i_2} = \sum_{l_1=\infty}^{\infty} \sum_{l_2=\infty}^{\infty} R_e(l_1 - l_2) h_{k_{i_1}}(l_1 + \varDelta - Mt_{i_1}) h_{k_{i_2}}(l_2 + \varDelta - Mt_{i_2}) ,$$

where $R_e(l_1 - l_2) \triangleq E\left[e(l_1)e(l_2)\right]$ is the autocorrelation of $e(n)$. k_{i_1} and k_{i_2} represent the subscript index of $h_k(l+\varDelta-Mt)$ of the i_1- and i_2-th elements of the vector $\boldsymbol{q}(n)$, respectively, while t_{i_1} and t_{i_2} represent the corresponding index t of $h_k(l + \varDelta - Mt)$ of the i_1- and i_2-th elements of the vector $\boldsymbol{q}(n)$, respectively. Then, SNR_i becomes

$$\mathrm{SNR}_i = \frac{\boldsymbol{w}_i' \boldsymbol{s}_i(k_i) \boldsymbol{s}_i(k_i)' \boldsymbol{w}_i}{\sum_{\substack{m=0 \\ m \neq i}}^{M-1} \sum_{\substack{k=0 \\ k \neq k_i}}^{L_s+L_w-2} \boldsymbol{w}_i' \boldsymbol{s}_m(k) \boldsymbol{s}_m(k)' \boldsymbol{w}_i + \boldsymbol{w}_i' \boldsymbol{C} \boldsymbol{w}_i} . \tag{3.71}$$

The optimal equalizer coefficient vector $\boldsymbol{w}_i$ can be found by maximizing SNR_i under the constraint of $\boldsymbol{w}_i' \boldsymbol{s}_i(k_i) = 1$.

Chapter 4

Channel Phase Response and Integer-Delay Equalization Technique

Chapter 3 introduced several equalization techniques for filterbank-based DMT systems. By examining such equalization techniques, one can notice an interesting fact that no matter what kind of equalization structure and optimization technique is used, every equalization technique has to deal with the same design parameter: the decision delay Δ. Simulation results show that the decision delay plays a critical role in the system output SNR and the transmission bit-rate. An improper choice of decision delay may significantly decrease the system output SNR and transmission bit-rate.

The decision delay is determined by the channel phase response. In the hitherto literature, the effect of channel phase response in channel equalization for filterbank-based DMT systems has been considered only occasionally. For example, the TEQ technique for DWMT system [18] was proposed to linearize the channel phase response. The wavelet packet-based equalization algorithm [21] was proposed for the case of binary input, which is based on replacing the equalizers by a set of delays to compensate for the channel delay. However, neither the interplay between the channel delay and the multirate decimator, nor the effect of the fractional channel delay was considered. Note that because of the use of decimators, most samples of the filterbank outputs are discarded. When the time synchronization of decimators mismatch that of the filterbank outputs because of the channel phase distortion, the samples discarded by decimators may contain information highly useful or relevant to the characteristic of input symbols. Based on this fact, a new equalization technique can be developed at essentially little or no additional cost.

In this chapter, the effect of channel phase response is investigated to place in evidence the dominant role of the channel delay. Then, an integer-delay equalization technique is proposed for binary or low bit-rate input cases.

The DMT system used in this chapter employs critically decimated filterbanks. Its block diagram is the same as that in Figure 2.7 with the condition $N = M$. With this condition, M will be used as the expansion/decimation ratio. Moreover, the synthesis and analysis filterbanks ($F_m(z)$ and $H_i(z)$ where $m, i = 0, ..., M-1$) are assumed to satisfy the biorthogonality property.

4.1 Effect of Channel Phase Response

In MCM systems including filterbank-based DMT systems, the channel phase response (compared to the channel magnitude response) plays a dominant role in system output distortion. This can be revealed by inspecting the signal flow in a DMT system using a set of ideal brick-wall filters forming its critically decimated filterbanks.

A brick-wall filter is an ideal (but non-practical) linear phase filter with nonzero constant passband magnitude response, zero stopband magnitude response, and zero transition bandwidth. In the ideal case where all filters in the synthesis and analysis filterbanks are brick-wall filters, the whole transmission frequency band is ideally partitioned into M subchannels and no spectral overlapping exists between any two subchannels. This[1] is illustrated in Figure 4.1, where the bandwidth of each subchannel is $2\pi/M$.

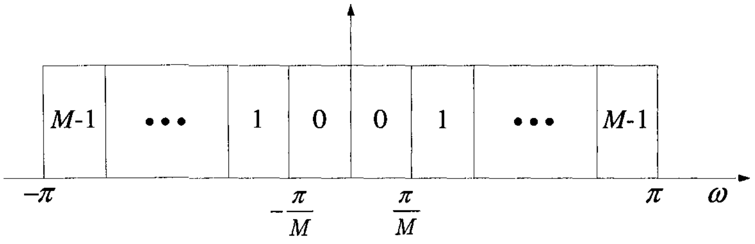


Figure 4.1: Ideal frequency subdivision of a M-subchannel filterbank-based DMT system

In this case (ignoring the channel noise), $S_{mi}(z) = 0$ when $m \neq i$, implying that no ICI exists in the system output. Then, Eqn (2.34) can be simplified as

$$Y_i(z) = \tilde{Y}_i^{(0)}(z) = X_i(z) S_{ii}^{(0)}(z) \,. \tag{4.1}$$

The block diagram of the i-th subchannel can also be simplified as Figure 4.2.

[1] Depending on the design of the synthesis and analysis filterbanks, the frequency subdivision may be different from Figure 4.1. However, the bandwidth of each subchannel is still $2\pi/M$ for the M-band frequency subdivision and the discussion in this chapter still holds.

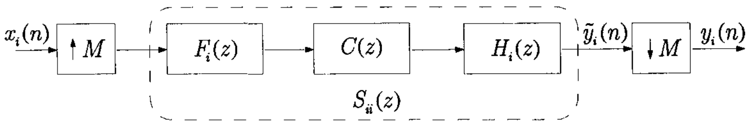


Figure 4.2: Block diagram of the simplified i-th subchannel

In filterbank-based DMT systems, the bandwidth of each subchannel is sufficiently narrow provided that M is sufficiently large. In this case, the frequency response of each subchannel can be approximated as a piecewise flat magnitude and piecewise linear phase, i.e. the channel frequency response $C(e^{j\omega})$ can be approximated as

$$C(e^{j\omega}) = |C(e^{j\omega})|e^{-j\phi(\omega)} \approx C_i e^{-jd_i\omega}, \quad \omega 1 \leq \omega \leq \omega 2\,, \tag{4.2}$$

where $|C(e^{j\omega})|$ represents the channel magnitude response (approximated as a constant number C_i), $\phi(\omega)$ represents the channel phase response (approximated as linear phase $d_i\omega$). Here, $d_i \geq 0$ is an integer or fractional constant number representing the subchannel delay, and ω_1 and ω_2 are the left and right cut-off frequencies of the i-th subchannel, respectively.

In the above DMT system, C_i can be easily compensated for by using a 1-tap equalizer $1/C_i$ in each subchannel at the receiver[2]. However, the subchannel delay d_i plays dominant role in the output distortion of a system using decimators, and is more difficult to compensate than the channel magnitude C_i. In the following discussion, we will assume that $C_i = 1$ so as to concentrate on the effect of the subchannel delay d_i only.

Eqn (4.2) means that for the i-th subchannel of a DMT system (see Figure 4.2), one can use z^{-d_i} as an approximation[3] for $C(z)$ without changing the relationship between $y_i(n)$, $\tilde{y}_i(n)$ and $x_i(n)$. With this approximation, the relationship between $\tilde{y}_i(n)$ and $x_i(n)$ in z-domain can be written as

$$\tilde{Y}_i(z) = X_i(z^M)S_{ii}(z) = X_i(z^M)G_{ii}(z)z^{-d_i}\,, \tag{4.3}$$

where $G_{ii}(z)$ satisfies the biorthogonality, i.e. $G_{ii}^{(0)}(z) = (G_{ii}(z))_{\downarrow M} = 1$. Then, the k-th polyphase component of $\tilde{Y}_i(z)$ becomes

$$\tilde{Y}_i^{(k)}(z) = X_i(z)S_{ii}^{(k)}(z) = X_i(z)\left(G_{ii}(z)z^{-d_i+k}\right)_{\downarrow M}\,, \quad k = 0, ..., M-1\,, \tag{4.4}$$

[2] If the subchannel gain is zero, the corresponding demodulator gain becomes infinity. But in practical situations, a subchannel with zero-gain is usually not used. Therefore, the case of zero-gain subchannel can be ignored in subsequent discussions.

[3] Although the case of d_i being a non-integer is not defined in discrete domain, we can relax the conventional integer delay restriction to accommodate the general channel delay cases. The meaning of non-integer delay will be explained by using the concept of Continuous-Time Processing of Discrete-Time Signals ([41], Chapter 4).

Now the effect of the subchannel delay d_i in the output distortion can be seen by considering the following cases.

Case 1: $d_i = m_i M$, where $m_i \geq 0$ is an integer.

In this case, the input symbols can be perfectly recovered at the system output (except for the delay m_i), in accordance with

$$\begin{aligned} Y_i(z) = \tilde{Y}_i^{(0)}(z) &= X_i(z)\left(G_{ii}(z)z^{-m_i M}\right)_{\downarrow M} \\ &= X_i(z)z^{-m_i}G_{ii}^{(0)}(z) \\ &= X_i(z)z^{-m_i}\,. \end{aligned} \tag{4.5}$$

Note that $Y_i(z)$ is but the 0-th polyphase component of $\tilde{Y}_i(z)$. Moreover, in general, $G_{ii}^{(k)}(z) \neq 1$ when $k \neq 0$. Therefore, other polyphase components of $\tilde{Y}_i(z)$ are distorted versions of $X_i(z)$ because

$$\tilde{Y}_i^{(k)}(z) = X_i(z)z^{-m_i}G_{ii}^{(k)}(z) \neq X_i(z), \quad k = 1, ..., M-1\,. \tag{4.6}$$

Eqns (4.5) and (4.6) indicate that in time-domain, $\tilde{y}_i(n)$ is a combination of the desired symbols $\big(\, x_i(n-m_i)\,\big)$ and a large number of other unwanted (and conventionally unimportant) symbols interleaving together.

Case 2: $d_i > 0$ is an integer but not an integer multiple of M.

In this case, let

$$d_i = m_i M + \bar{d}_i\,, \tag{4.7}$$

where $m_i \geq 0$ is an integer, and $1 \leq \bar{d}_i \leq M-1$. Then,

$$\begin{aligned} Y_i(z) &= X_i(z)\left(G_{ii}(z)z^{-m_i M - \bar{d}_i}\right)_{\downarrow M} \\ &= X_i(z)\left(G_{ii}(z)z^{-(m_i+1)M}z^{M-\bar{d}_i}\right)_{\downarrow M} \\ &= X_i(z)z^{-m_i-1}G_{ii}^{(M-\bar{d}_i)}(z)\,. \end{aligned} \tag{4.8}$$

Because $G_{ii}^{(M-\bar{d}_i)}(z) \neq 1$, the system output is distorted from the system input. However, note that

$$\tilde{Y}_i^{(\bar{d}_i)}(z) = X_i(z)z^{-m_i}G_{ii}^{(0)}(z) = X_i(z)z^{-m_i}\,. \tag{4.9}$$

Therefore, in time-domain, the $\bar{d}_i$-th polyphase component of $\tilde{y}_i(n)$ is the desired symbol $x_i(n-m_i)$, but, unfortunately, it is conventionally discarded at the decimator output.

By recalling that the only difference between Cases 1 and 2 is the different values of d_i, it is clear that the **shape** of $\tilde{y}_i(n)$ in Case 2 is the same as that in Case 1 (c.f. Eqn

(4.3)). Therefore, it is obvious that the channel delay d_i makes the decimator work at improper time synchronization, as a result of which the desired symbols are discarded while the decimator output is unwanted symbols.

With the above discussions, it is clear that the channel delay (or phase response) is a more dominant factor than the channel magnitude response. The desired symbols $x_i(n)$ can be extracted from $\tilde{y}_i(n)$ if and only if the time synchronization of the decimator matches that of $\tilde{y}_i(n)$. However, the channel delay may put the decimator out of the time synchronization, and make the decimator discard the undistorted symbols (or important information that can be used to recover the input symbols, when considering the nonlinear magnitude and phase distortions). A small channel delay may cause the the decimator output become highly distorted. This distortion is much more severe in high-frequency than in low-frequency subchannels because $\tilde{y}_i(n)$ of high-frequency subchannels fluctuate much more rapidly than that of low-frequency subchannels. In light of this fact, more attention should be paid to the channel phase response when equalizing a channel in filterbank-based DMT systems. If the characteristics of the channel phase response can be fully exploited, efficient equalization techniques can be found by compensating the channel delay to restore correct time synchronization of the decimator so as to match the channel delay.

Another useful conclusion can also be drawn from the above discussion. Note that the channel delay can be decomposed into two parts, an integer multiple of M (m_iM) and a principle-value delay $\bar{d}_i$ (c.f. Eqn (4.7)). Because decimators are used in the filterbank-based DMT system, m_iM will only delay the system output by m_i samples, and would not distort the system output. It is the principle-value delay $\bar{d}_i$ which leads to the distortion in the system output (c.f. Eqn (4.9)). Therefore, the effect of the channel delay d_i exhibits the periodic-like property, i.e. after subtraction of suitable integer multiples of M, d_i can always be represented by ("mapped to") its principle-value delay $\bar{d}_i$, which can be related to one or some of the M polyphase components of $\tilde{y}_i(n)$. Then, the channel equalization can be restricted in a limited range of delay (0, ..., $M-1$) and simplify the equalization. Specifically, the channel equalization can be done by using one or some of the M polyphase components of $\tilde{y}_i(n)$.

Note that the above discussion does not consider the case of d_i being a non-integer. This case will be discussed in Chapter 5, where a new interpolation equalization technique is proposed.

4.2 Channel Phase Response – Linear or Nonlinear?

The channel used in the filterbank-based DMT system is often assumed to be a linear phase channel. However, this assumption does not hold true in practical situations. As an example, the frequency response of the standard channel Carrier Serving Area (CSA) loop[4] #1 is shown in Figure 4.3.

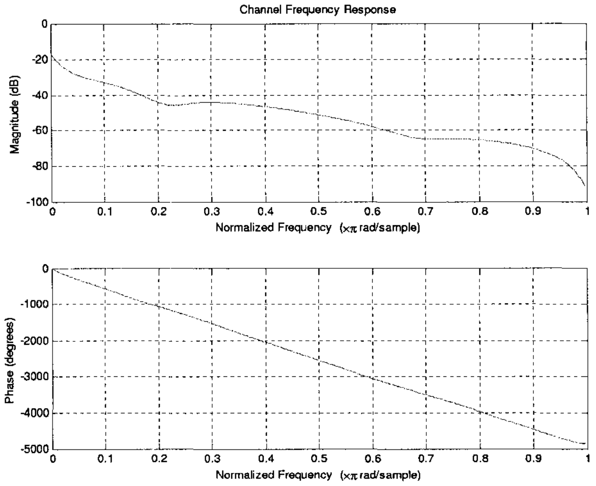


Figure 4.3: Frequency response associated with CSA loop #1

The channel phase response shown in Figure 4.3 looks as if it is linear, but actually it is not. A channel is linear phase if and only if its group-delay is a constant number. The group-delay of CSA loop #1 is shown in Figure 4.4, together with an enlarged portion in Figure 4.5. It is clear that even in view of the large frequency scale, this channel cannot be treated as linear phase because there is steep decrease at the low frequency band[5], as well as two dips around $\omega = 0.22\pi$ and $\omega = 0.68\pi$, respectively, and rapid fluctuations at high-frequency band. Moreover, even in the frequency region which is roughly flat in view of the large frequency scale, for example the area around 0.5π, the group-delay is also fluctuating and cannot be treated as linear from the view of small frequency scale (c.f. Figure 4.5).

Same conclusion can be drawn regarding other practical channels. Generally, the phase response of an actual channel cannot be treated as linear because the channel impulse response is neither symmetric nor anti-symmetric [5].

Generally, it is difficult to equalize the channel phase response using a single pre-detection equalizer when the channel phase response is nonlinear. However, because the

[4] The data for CSA loops as used in this thesis comes from the *DMT Toolbox* designed by Guner Arslan, *et al.*, Embedded Signal Processing Laboratory, Department of Electrical and Computer Engineering, The University of Texas at Austin, under the terms of the GNU General Public License.

[5] This is because of the AC property inherited in the twisted copper line.

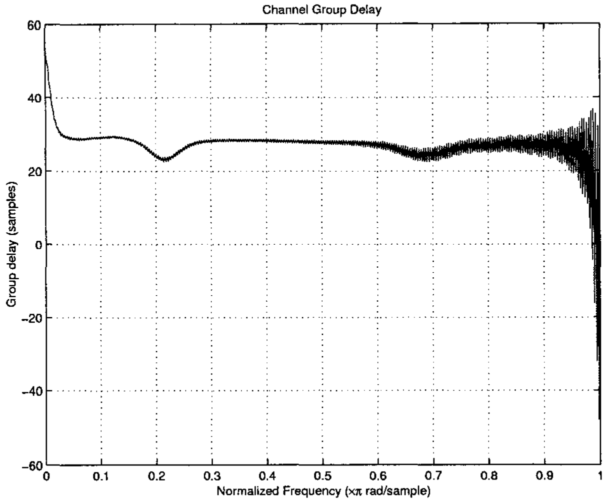


Figure 4.4: Group-delay associated with CSA loop #1

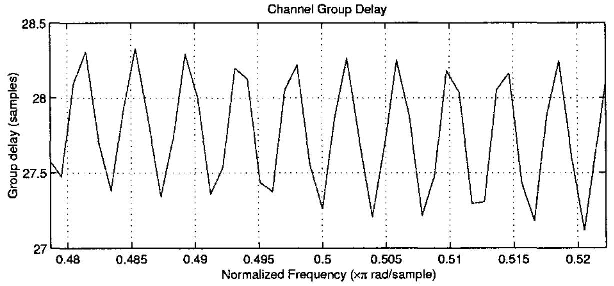


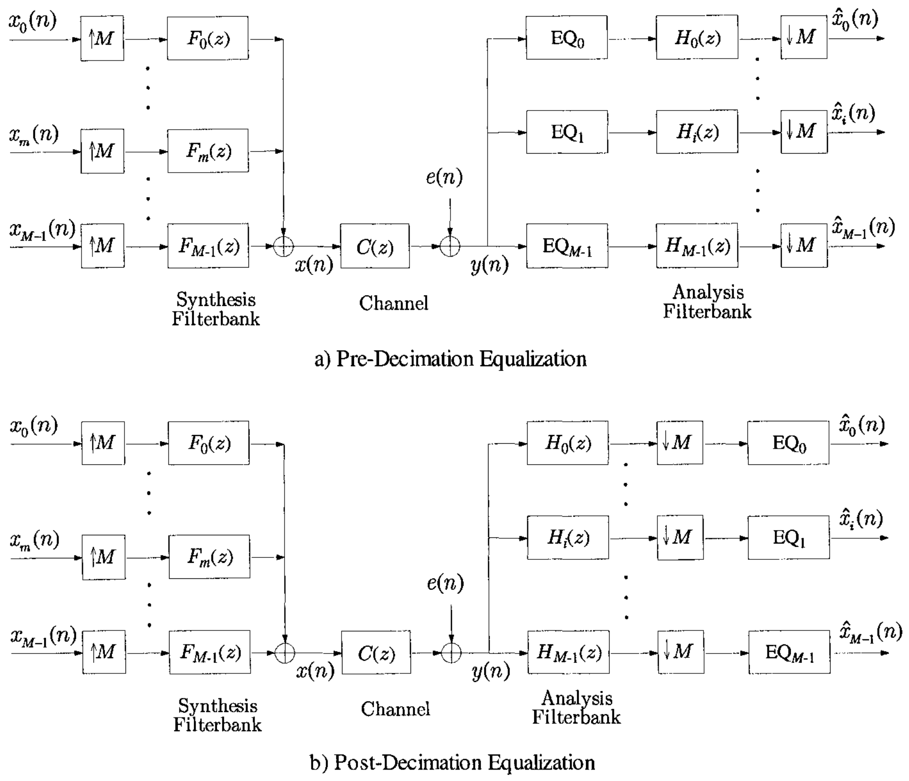
Figure 4.5: Enlarged illustration of the group-delay for CSA loop #1 around frequency $\omega = 0.5\pi$

channel phase response can be approximated as piecewise linear, it can easily be equalized piece by piece, or subchannel by subchannel, in filterbank-based DMT systems.

4.3 Pre-Decimation or Post-Decimation Equalization

As discussed above, each subchannel needs a separate equalizer at the receiver to equalize the channel distortion. Generally, there are two locations at the receiver to put the equalizer. One location is before the decimator, which is referred to as Pre-Decimation Equalization. The other location is after the decimator, which is referred to as Post-Decimation Equalization. The pre- and post-decimation equalizations are shown in Figure 4.6, where EQ_0, ..., EQ_{M-1} represent equalizers.

The main drawback of the pre-decimation equalization is that there are too many unwanted symbols mixed with the desired symbols taking part in the equalization, leading to high computation complexity. On the other hand, the main drawback of the post-decimation



a) Pre-Decimation Equalization

b) Post-Decimation Equalization

Figure 4.6: Pre-decimation and post-decimation equalizations

equalization is that the output of the decimator may be unwanted symbols that are severely distorted from the input symbols, which leads to a lowered output SNR. Therefore, a better approach is to combine the pre- and the post-decimation equalization together by fully exploiting the characteristic of the channel phase response, paving the way to the development of efficient equalization techniques.

4.4 Integer-Delay Equalization

In Case 2 in Section 4.1, the channel was assumed to have a (piecewise) constant integer delay. Then, in each subchannel, one polyphase component of the analysis filter output contains undistorted input symbols. If the assumption of the constant integer subchannel delay is acceptable, then, instead of using a decimator, a certain polyphase component of the analysis filter output can be picked out to recover the input symbols. This leads to the integer-delay equalization technique with no additional cost.

The integer-delay equalization technique is the direct implementation counterpart of the wavelet packet-based equalization algorithm proposed in [21] where the channel is unknown and the equalization only depends on the statistics of the output. The integer-delay equalization technique has the same disadvantage as the wavelet packet-based equalization technique, i.e., the performance measured by the Signal-to-Interference Ratio (SIR) in noiseless environment may be low and can only be used for binary or low bit-rate input cases. However, this equalization technique has the advantage of very simple implementation. It is a good starting point to derive the interpolation equalization technique which can obtain high SIR with limited complexity. The interpolation equalization technique will be introduced in Chapter 5.

The block diagram of the integer-delay equalization technique is shown in Figure 4.7. In each subchannel, a 1-tap equalizer $a_i z^{d_i}$ is inserted before the decimator[6], where a_i and d_i (d_i is an integer) are used to compensate the channel magnitude distortion and the channel delay in the i-th subchannel, respectively.

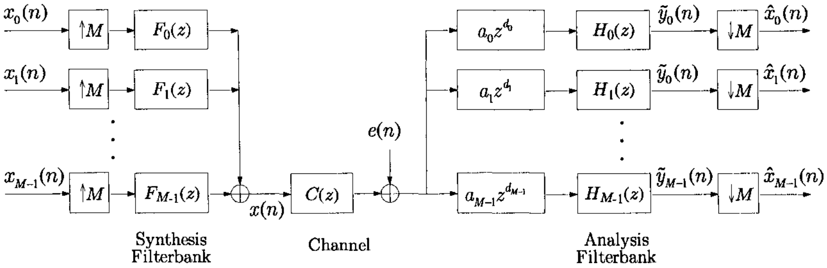


Figure 4.7: Integer-delay equalization technique

In the i-th subchannel, by ignoring the channel noise, the z-transform of $\tilde{y}_i(n)$ can be written as:

$$\tilde{Y}_i(z) = \sum_{m=0}^{M-1} X_m(z^M) S_{mi}(z) a_i z^{d_i} . \tag{4.10}$$

When synthesis and analysis filterbanks are carefully designed, ICI can be ignored and Eqn (4.10) can be simplified as

$$\tilde{Y}_i(z) = a_i X_i(z^M) S_{ii}(z) z^{d_i} . \tag{4.11}$$

Decomposing d_i by using Eqn (4.7), the z-transform of the system output $\hat{x}_i(n)$ can be expressed as

$$\hat{X}_i(z) = a_i X_i(z) z^{m_i} \left(S_{ii}(z) z^{d_i} \right)_{\downarrow M} = a_i X_i(z) z^{m_i} S_{ii}^{(d_i)}(z) . \tag{4.12}$$

If the channel effect in the i-th subchannel is fully equalized by the 1-tap equalizer $a_i z^{d_i}$, the system output $\hat{X}_i(z)$ is equal to the system input $X_i(z)$. Then, by letting Eqn (4.12) be equal to $X_i(z)$, one can say that $S_{ii}(z) z^{d_i}$ is a delayed and scaled Nyquist (M) filter [52]

[6] Without loss of generality, suppose $d_i \geq 0$.

in accordance with

$$\left(S_{ii}(z)z^{d_i}\right)_{\downarrow M} = S_{ii}^{(d_i)}(z) = \frac{1}{a_i}z^{-m_i}\,. \tag{4.13}$$

Therefore, in time-domain, the d_i-th polyphase component of $s_{ii}(n)$ is a delayed (and scaled) δ function ($\frac{1}{a_i}\delta(n - m_i)$). If d_i is known, one can use the d_i-th polyphase component of $\tilde{y}_i(n)$ as the recovered symbols[7]. In light of this, the equalization objective now becomes finding which polyphase component of $s_{ii}(n)$ is a delayed δ function.

Suppose the channel impulse response is known. Then, $s_{ii}(n)$ is also known. By decomposing $s_{ii}(n)$ into its M polyphase components $\{s_{ii}^{(m)}(n)\}$ ($m = 0, \ldots, M-1$), the objective of the integer-delay equalization technique is to find an optimal polyphase components of $s_{ii}(n)$ which is most similar to a (delayed and scaled) δ function. The modified filterbank-based DMT system incorporating the integer-delay equalization technique is shown in Figure 4.8.

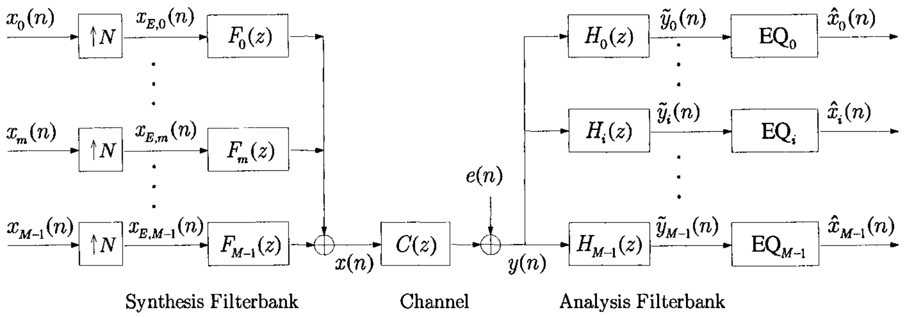


Figure 4.8: The modified filterbank-based DMT system incorporating equalizers

In each branch of the modified filterbank-based DMT system receiver, the decimator used in the original filterbank-based DMT receiver (see Figure 2.7) is replaced by an integer-delay equalizer. The block diagram of the i-th integer-delay equalizer EQ_i is shown in Figure 4.9. In this figure, a S/P converter is used to parse $\tilde{y}_i(n)$ (see Figure 4.8) into its M polyphase components. Then, one polyphase component, say the d_i-th polyphase component ($\tilde{y}_i^{(d_i)}(n)$), that corresponds the optimal polyphase component of $s_i i(n)$ is picked out as the recovered symbols after being scaled by a_i.

The algorithm for finding d_i and the corresponding a_i is as follows:

1. Calculate $s_{ii}(n)$ using Eqn (2.30) where $m = i$. Then parse $s_{ii}(n)$ into its M polyphase components, $\{s_{ii}^{(m)}(n)\}$, where $s_{ii}^{(m)}(n) = s_{ii}(nM + m)$ and $m = 0\,, \ldots, M-1$.

[7] Note that in original filterbank-based DMT systems (c.f. Figure 2.7), the system output is the 0-th polyphase component of $\tilde{y}_i(n)$.

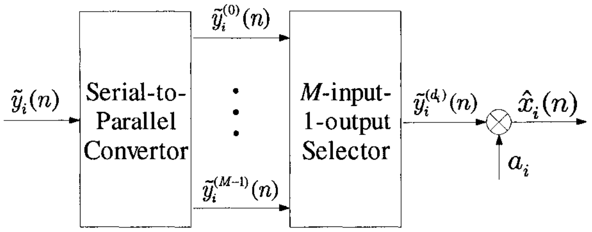


Figure 4.9: The i-th integer-delay equalizer

2. Find the maximum of each $s_{ii}^{(m)}(n)$, and denote it as $s_{ii}^{(m)}(n_m)$. Then $1/s_{ii}^{(m)}(n_m)$ is a candidate for a_i.

3. Normalize each $s_{ii}^{(m)}(n)$ such that its maximum sample value is equal to 1, i.e. calculate $\{s_{ii}^{(m)}(n)/s_{ii}^{(m)}(n_m)\}$.

4. Search the above normalized polyphase components of $s_{ii}(n)$ to find an optimal one, say the d_i-th polyphase component, which mostly resembles a delayed δ function.

5. After finding d_i, pick up the d_i-th polyphase component of $\tilde{y}_i(n)$ and scale it by the corresponding a_i $\big(= 1/s_{ii}^{(l_i)}(n_{d_i}) \big)$ as the recovered symbols.

In Step 4, the optimal d_i can be found by minimizing the ∞-norm or 2-norm of the error sequence $r_{m,i}(n)$, where $r_{m,i}(n)$ is defined as

$$r_{m,i}(n) \triangleq \frac{s_{ii}^{(m)}(n)}{s_{ii}^{(m)}(n_m)} - \delta(n - n_m)\,, \tag{4.14}$$

where

$$s_{ii}^{(m)}(n_m) = \max_n s_{ii}^{(m)}(n)\,, \tag{4.15}$$

with n_m representing the sample index of the maximum of $s_{ii}^{(m)}(n)$.

Minimizing the ∞-norm of $r_i(n)$ corresponds to the Minimax optimization [9], having the objective function

$$\min_m \left\{ \max_n \{|r_{m,i}(n)|\} \right\} . \tag{4.16}$$

On the other hand, minimizing the 2-norm of $r_i(n)$ corresponds to the MMSE optimization, having the objective function

$$\min_m \left\{ \sum_n |r_{m,i}(n)|^2 \right\} . \tag{4.17}$$

4.5 Simulation Results

In this section, it is assumed that system inputs $\{x_i(n)\}$ for various subchannels are mutually uncorrelated. Also, it is assumed that all input symbols within each subchannel are mutually uncorrelated. The filterbank-based DMT system uses critically decimated CMFB as its modulator/demodulator, where the design parameters of the CMFB are as listed in Table 4.1. Channels used in this simulation are standard CSA loops #1–#8 and are assumed to be noiseless. Their impulse and frequency responses are shown in Appendix A.

DMT prototype filter	Design approach	Kaiser windowing [33]
	Sidelobe attenuation A_s (dB)	> 50
	Filter length	$12M$

Table 4.1: CMFB design parameters for simulation of integer-delay equalization

In the following, the output SIR for different channels and for different M are compared, where the output SIRs are defined as

$$\mathrm{SIR}_i \triangleq \frac{E\left\{|x_i(n)|^2\right\}}{E\left\{|x_i(n)-\hat{x}_i(n)|^2\right\}}, \quad \text{for } e(n)=0\,. \tag{4.18}$$

Figures 4.11 and 4.12 show the output SIRs for CSA loops #1–#8 when $M = 64$. As comparison, the SIR for the case of an ideal channel $\left(\ c(n) = \delta(n)\ \right)$ is shown in Figure 4.10. In this case, no equalizer is needed and the output SIR is 42 dB[8]. Although the SIRs in Figures 4.11 and 4.12 are much lower than 42dB, the SIRs of most subchannels are around 15–30dB, which means that the integer-delay equalization technique is at least suitable for binary and low bit-rate input cases.

To compare the relationship between the SIR and the subchannel number M, Figure 4.13 shows the output SIRs for CSA loop #1–#4 when $M = 128$. Compared to the SIRs in Figure 4.11, one can see that the SIR ratio becomes higher when M increases. This is because the error of the piecewise linear phase approximation decreases when the number of subchannels increases.

A question arises when comparing Figures 4.11 and Figure 4.13: why are there several subchannels whose SIRs remain significantly low even for large M? Candidate examples for this question are subchannels #11, #22, #32, #33 and #39 of CSA loop #1 in Figure 4.11a, which are the counterparts of subchannels #22, #43, #64, #65, #77 of CSA loop #1

[8] The SIR for the ideal channel is only 42dB because here the CMFB used in the filterbank-based DMT system is a pseudo PR filterbank [52]. For PR filterbank, the SIR for ideal channel is infinity, i.e. no interference exists in the system output.

in Figure 4.13a. To answer this question, the channel phase response and the assumption of integer d_i used in the integer-delay equalization technique has to be re-examined, and the effect of non-integer channel delay has to be considered.

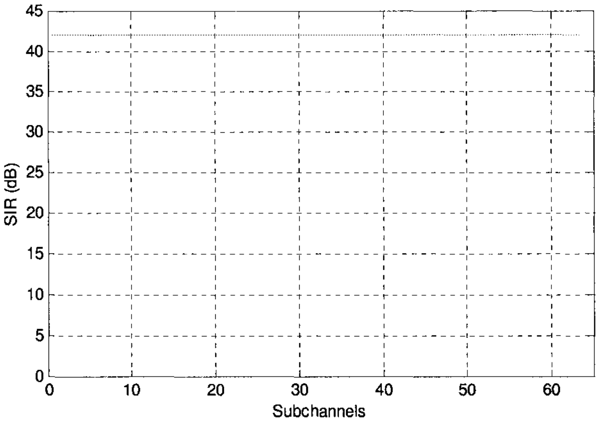


Figure 4.10: SIRs for the ideal channel when $M = 64$

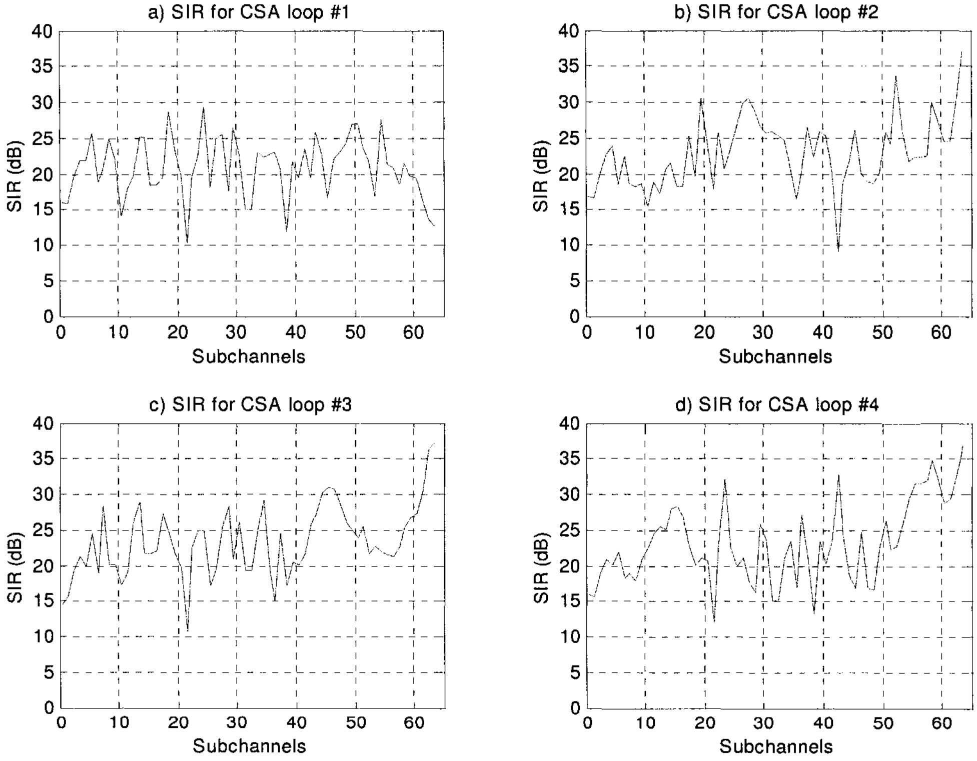


Figure 4.11: SIRs for integer-delay equalization for CSA loops #1–#4 when $M = 64$

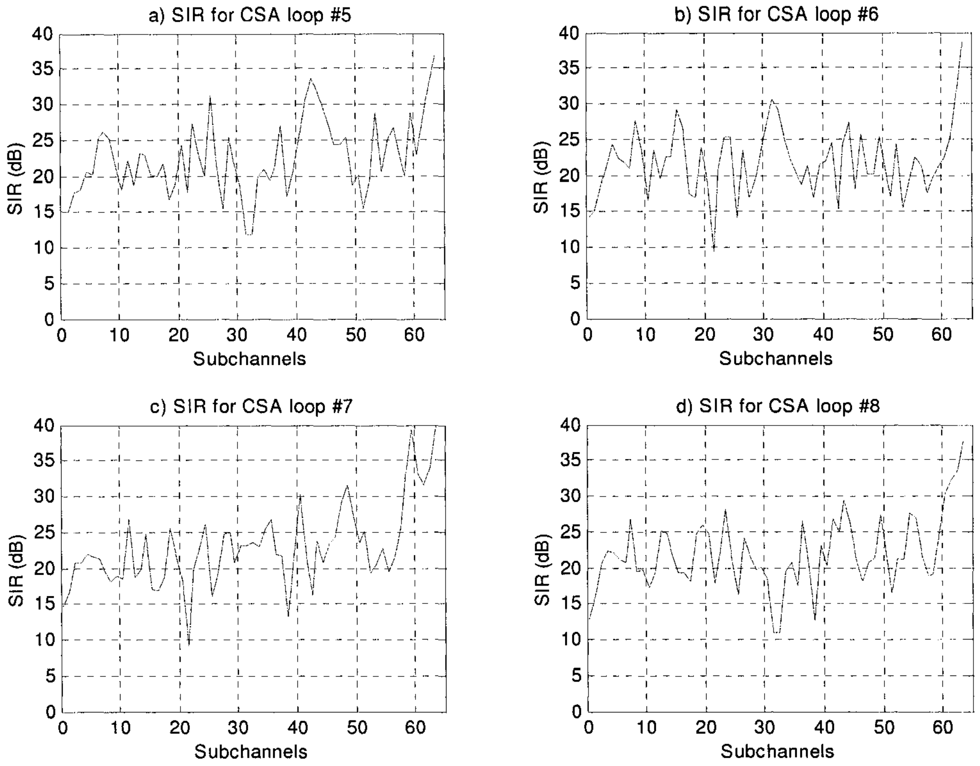


Figure 4.12: SIRs for integer-delay equalization for CSA loops #5–#8 when $M = 64$

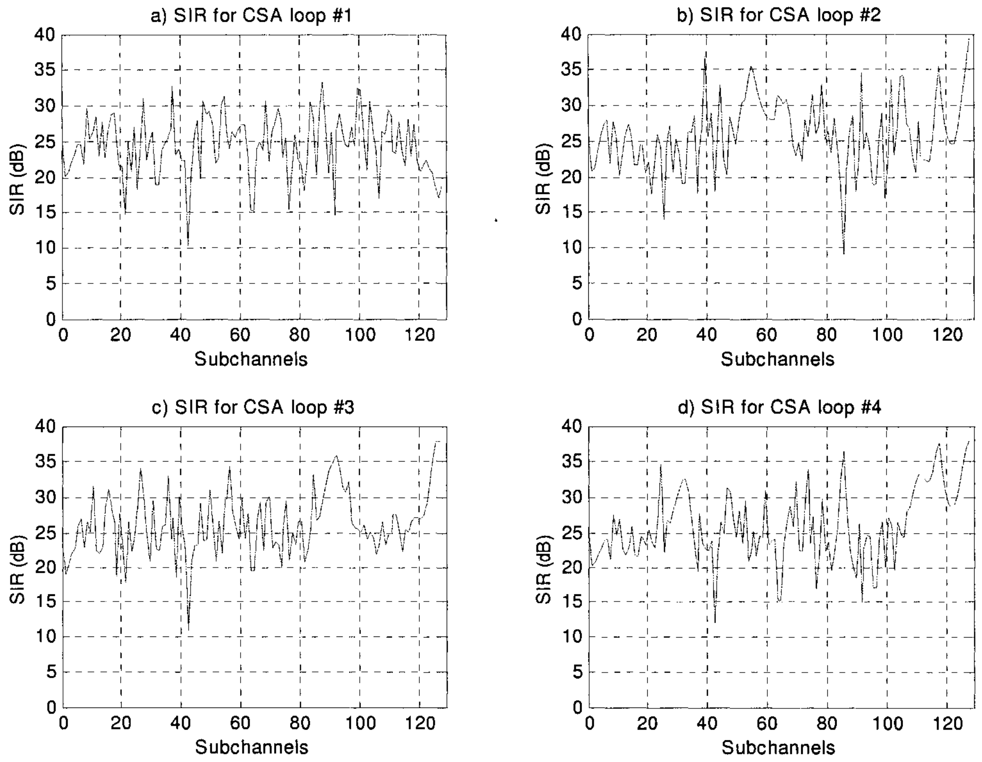


Figure 4.13: SIRs for integer-delay equalization for CSA loops #1–#4 when $M = 128$

Chapter 5

Channel Non-Integer Delay and One-Tap Interpolation Equalization

In the investigation of the effect of channel phase response in Chapter 4, the case where the subchannel delay d_i is a non-integer number was temporarily deferred. In practical situations, the subchannel delay is usually non-integer. In this chapter, the effect of non-integer delay will be investigated in detail. Then, a novel one-tap interpolation equalization technique [59] and its generalization, the multi-tap interpolation equalization technique [58] (Chapter 6), are proposed for DMT systems employing critically decimated filterbanks. This technique is based on the equalization of the channel fractional delay in each subchannel in time synchronization with the constituent decimator at the receiver end. This time synchronization is achieved through the exploitation of a subset (i.e. several polyphase components) of the signal samples of the analysis filter output (before decimation). The resulting equalization gives rise to a high SNR, while requiring a small number of equalizer taps. Moreover, it permits a tradeoff between various equalization parameters, leading to high computational flexibility. Generally, a complete search for the optimal polyphase components has to be done to obtain the maximum SNR. However, a significant simplification of this search can be achieved through constraining the search to within the channel lower and upper group-delay bounds within each subchannel.

As in Chapter 4, the DMT system under consideration employs critically decimated filterbanks (c.f. Figure 2.7 with the condition $N = M$), while the synthesis and analysis filterbanks ($F_m(z)$ and $H_i(z)$ where $m, i = 0, ..., M - 1$) are assumed to satisfy the biorthogonality property. Also, M will be used as the expansion/decimation ratio. The block diagram of the DMT system using critically decimated filterbanks is drawn in Figure 5.1. Some useful notations from Section 2.3 are repeated in the following for ease of reference.

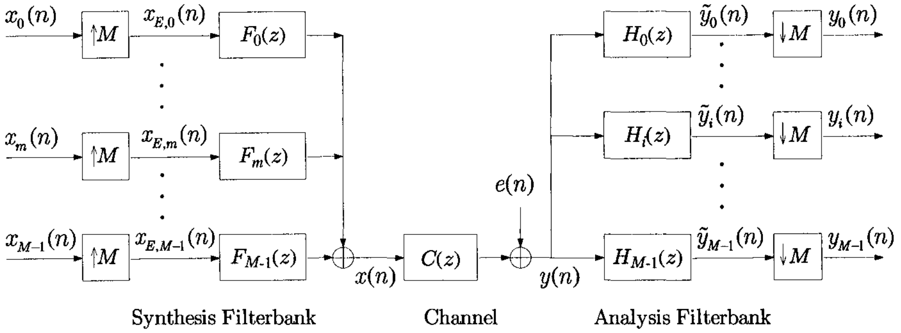


Figure 5.1: Block diagram of the DMT system using critically decimated filterbanks

$$G_{mi}(z) \triangleq F_m(z)H_i(z)\,, \tag{5.1}$$

$$S_{mi}(z) \triangleq F_m(z)C(z)H_i(z) = G_{mi}(z)C(z)\,, \tag{5.2}$$

$$\tilde{Y}_i(z) = \sum_{m=0}^{M-1} X_m(z^M)S_{mi}(z) + E(z)H_i(z)\,. \tag{5.3}$$

By ignoring the ICI and channel noise, Eqn (5.3) is approximated as

$$\tilde{Y}_i(z) \approx X_i(z^M)S_{ii}(z)\,. \tag{5.4}$$

By decomposing $\tilde{Y}_i(z)$ into its M polyphase components, the k-th polyphase component can be written as

$$\tilde{Y}_i^{(k)}(z) = X_i(z)S_{ii}^{(k)}(z)\,. \tag{5.5}$$

Once again, the channel is approximated as having piecewise flat magnitude response and piecewise linear phase response[1]. In the i-th subchannel ($i = 0$, ..., $M-1$), the approximated subchannel magnitude and subchannel delay are denoted as C_i and d_i, respectively. Then, Eqn (5.4) becomes

$$\tilde{Y}_i(z) \approx X_i(z^M)S_{ii}(z) \approx C_iX_i(z^M)G_{ii}(z)z^{-d_i}\,. \tag{5.6}$$

5.1 Interpretation of Fractional Delay in Discrete Domain

A non-integer delay is not defined in discrete domain. However, by using the concept of Continuous-Time Processing of Discrete-Time Signals ([41], Chapter 4), a non-integer delay

[1] If any of these two assumptions is not satisfied, the one-tap interpolation equalization technique being proposed in this chapter will give rise to relatively low output SNR (c.f. the simulation results in section 5.3.2). However, if the above assumptions are not satisfied, one can instead use the multi-tap interpolation equalization technique to obtain the desired high output SNR (c.f. Chapter 6).

becomes meaningful and helpful in practical applications.

A signal $x(n)$ may be visualized as discrete samples of an underlying continuous-time signal $x_c(t)$ in accordance with

$$x(n) = x_c(nT) = x_c(t)|_{t=nT} , \tag{5.7}$$

where T represents the sampling period. Then, if $x_c(t)$ is delayed by t_0, the corresponding discrete-time signal, denoted by $\tilde{x}(n)$ becomes

$$\begin{aligned}\tilde{x}(n) &= x_c(t-t_0)|_{t=nT} = x_c(nT-t_0) = x_c((n-\frac{t_0}{T})T) \\ &= x(n-d) ,\end{aligned} \tag{5.8}$$

where $d \triangleq t_0/T$ is the delay between $\tilde{x}(n)$ and $x(n)$. Obviously, if $t_0 \neq kT$ (k is an integer), then d becomes a non-integer delay.

5.2 Effect of Channel Non-Integer Delay

Consider the case where $d_i > 0$ is a non-integer number. In this case, no polyphase component of $\tilde{y}_i(n)$ is equal to $x_i(n)$. Specifically, let $d_i = \hat{d}_i + \xi_i$, where $\hat{d}_i$ is an integer and $|\xi_i| \leq 0.5$. Then, an integer-delay equalizer can only compensate the integer part $\hat{d}_i$, and the fractional part ξ_i will cause the system output to be distorted from the corresponding system input. This distortion depends on the amount of the remaining subchannel delay as well as on how rapidly $s_{ii}(n)$ fluctuates. Generally, if $|\xi_i| \ll 0.5$ and/or $s_{ii}(n)$ fluctuates relatively slowly, the output distortion is relatively small. Otherwise, the output distortion is large. Figure 5.2 illustrates the effect of the non-integer channel delay when $M = 2$ and the channel magnitude response is assumed to be 1. The dashed curve represents $g_{ii}(n)$ and the solid curve represents $s_{ii}(n)$. The circles on the dash curve represent $(g_{ii}(n))_{\downarrow M}$ while the circles on the solid curve represent $(s_{ii}(n))_{\downarrow M}$. The delay between $g_{ii}(n)$ and $s_{ii}(n)$ is roughly 0.5 sample, i.e. $d_i \approx 0.5$.

From Figure 5.2, it is clear that $(g_{ii}(n))_{\downarrow M}$ is a delayed δ function $\big(= \delta(n-4) \big)$. However, $(s_{ii}(n))_{\downarrow M}$ is far from such a δ function and no polyphase component of $s_{ii}(n)$ can be approximated as a (delayed) δ function, although $d_i \approx 0.5$ is quite small.

When d_i is a non-integer number, all polyphase components of $\tilde{y}(n)$ are distorted from $x_i(n)$. However, some of the polyphase components of $\tilde{y}(n)$ still embody important information about $x_i(n)$ (will be explained later in this section). By taking this into account, the system input can still be recovered from the combination of several polyphase components of $\tilde{y}_i(n)$.

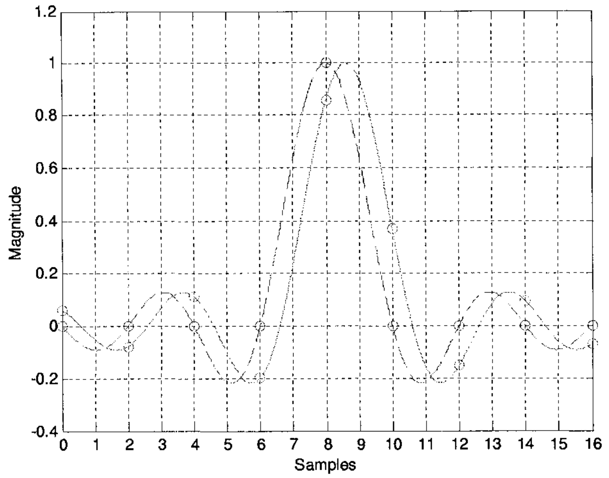


Figure 5.2: Effect of non-integer channel delay

First, suppose that $\tilde{Y}_i(z)$ can be advanced by a non-integer advance component[2] z^{d_i} to fully compensate for the subchannel delay d_i. Then, from Eqn (5.6) one has

$$\left(\tilde{Y}_i(z)z^{d_i}\right)_{\downarrow M} = C_i X_i(z)\,, \tag{5.9}$$

or in time-domain

$$\left(\tilde{y}_i(n+d_i)\right)_{\downarrow M} = \tilde{y}_i^{(0)}(n+d_i) = C_i X_i(z)\,, \tag{5.10}$$

i.e. the 0-th polyphase component of $\tilde{y}_i(n+d_i)$ is a scaled version of the input symbols $x_i(n)$.

Generally, a fractional delay (or, equivalently, a fractional advance) all-pass filter is difficult to implement. Numerous techniques were proposed in this topic [16] [19] [27] [34] [35] [42]. Most of these techniques are based on interpolation of time-domain samples. However, all of them are impractical to be used in DMT systems. On the other hand, the fractional delay component is not a necessary part in our equalizer because the only purpose of equalization is to recover the input signal. Any approach can be used to substitute the fractional delay component as long as the input signal can be recovered. Also, in xDSL applications, usually, the channel impulse response can be found during the initialization of the xDSL modem[3]. Therefore, instead of implementing a fractional delay all-pass filter, one can exploit the idea of sample interpolation as well as the effect of channel delay in filterbank-based DMT systems to find an efficient equalization technique.

[2] Although the advance component is noncausal, one can employ it through adding an extra delay z^{-M}.

[3] For those applications that the channel impulse response cannot be known in advance, blind equalization techniques have to be used [46].

From the view of Continuous-Time Processing of Discrete-Time Signals, one has

$$\tilde{y}_i(n) = \tilde{y}_{c,i}(nT) = \tilde{y}_{c,i}(t)|_{t=nT} \,, \tag{5.11}$$

$$\tilde{y}_i(n+d_i) = \tilde{y}_{c,i}(nT + d_iT) = \tilde{y}_{c,i}(t + d_iT)|_{t=nT} \,, \tag{5.12}$$

where T is the sampling period, and $\tilde{y}_{c,i}(t+d_iT)$ is a continuous-time signal advanced from the continuous-time signal $\tilde{y}_{c,i}(t)$. Note that the **shape** of $\tilde{y}_{c,i}(t+d_iT)$ is the same as that of $\tilde{y}_{c,i}(t)$ regardless of the value of d_i. Then, $\tilde{y}_i(n)$ and $\tilde{y}_i(n+d_i)$ can also be treated as **two different sample sets of the same continuous-time signal** $\tilde{y}_{c,i}(t)$, which are sampled at different time instants. Therefore, $\tilde{y}_i(n+d_i)$ can be obtained through interpolation of $\tilde{y}_i(n)$ samples.

Now suppose $d_i = d_{i0} + d_{i1}$, where $d_{i0} \geq 0$ is an integer number, and $0 < d_{i1} < 1$. Then, given an arbitrary integer sample index n_0, $\tilde{y}_i(n_0 + d_{i0})$ and $\tilde{y}_i(n_0 + d_{i0} + 1)$ are two neighboring samples of the $\tilde{y}_i(n_0 + d_i)$ sample. Therefore, $\tilde{y}_i(n_0 + d_i)$ can be obtained by using two-point interpolation of $\tilde{y}_i(n_0 + d_{i0})$ and $\tilde{y}_i(n_0 + d_{i0} + 1)$. An illustration of interpolation is shown in Figure 5.3. In this figure, two sample series $\tilde{y}_i(n)$ and $\tilde{y}_i(n+d_i)$, as well as the corresponding continuous time signal $\tilde{y}_{c,i}(t)$, are drawn together. On the curve $\tilde{y}_{c,i}(t)$, "o" represents the sample of $\tilde{y}_i(n)$ while "◇" represents the sample of $\tilde{y}_i(n+d_i)$.

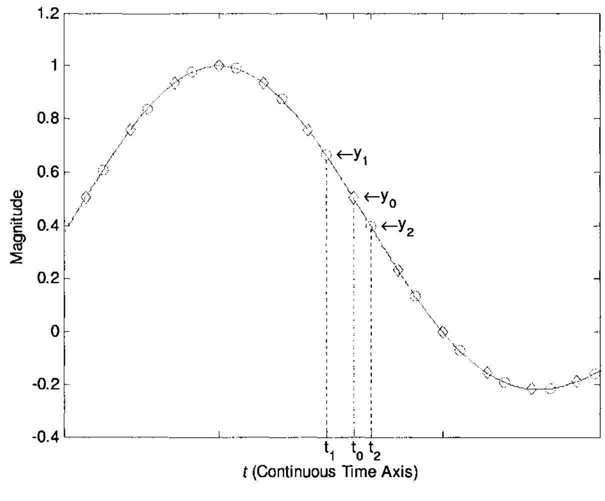


Figure 5.3: Illustration of sample interpolation

Consider the sample point y_0 and its two neighbors y_1 and y_2, which correspond to $\tilde{y}_i(n_0+d_i)$, $\tilde{y}_i(n_0+d_{i0})$ and $\tilde{y}_i(n_0+d_{i0}+1)$, respectively. By using two-point interpolation[4]

[4] Generally, interpolation involving more than two points leads to nonlinear equalization.

of $\tilde{y}_i(n_0 + d_{i0})$ and $\tilde{y}_i(n_0 + d_{i0} + 1)$, $\tilde{y}_i(n_0 + d_i)$ can be approximated as

$$\begin{aligned} \tilde{y}_i(n_0 + d_i) &\approx \tilde{y}_i(n_0 + d_{i0})\frac{t_0 - t_2}{t_1 - t_2} + \tilde{y}_i(n_0 + d_{i0} + 1)\frac{t_0 - t_1}{t_2 - t_1} \\ &= \tilde{a}_{i0}\tilde{y}_i(n_0 + d_{i0}) + \tilde{a}_{i1}\tilde{y}_i(n_0 + d_{i0} + 1)\,, \end{aligned} \tag{5.13}$$

where $\tilde{a}_{i0}$ and $\tilde{a}_{i1}$ are defined as

$$\tilde{a}_{i0} \triangleq \frac{t_0 - t_2}{t_1 - t_2}\,, \tag{5.14}$$

$$\tilde{a}_{i1} \triangleq \frac{t_0 - t_1}{t_2 - t_1}\,, \tag{5.15}$$

and $|\tilde{a}_{i0}|\,, |\tilde{a}_{i1}| \leq 1$.

After finding $\tilde{y}_i(n + d_i)$ by sample interpolation, the system input can be recovered using Eqn (5.10). However, not all samples of $\tilde{y}_i(n+d_i)$ need to be obtained. As mentioned before, $\tilde{y}_i(n + d_i)$ is the combination of desired symbols $x_i(n)$ and a large number of other unwanted symbols interleaved together. Therefore, from Eqn (5.9), only samples of the 0-th polyphase component of $\tilde{y}_i(n + d_i)$ have to be obtained.

In accordance with the above discussion, the input symbols can be recovered as

$$x_i(n) = \left(\frac{1}{C_i}\tilde{y}_i(n + d_i)\right)_{\downarrow M} = a_{i0}\left(\tilde{y}_i(n + d_{i0})\right)_{\downarrow M} + a_{i1}\left(\tilde{y}_i(n + d_{i0} + 1)\right)_{\downarrow M}\,, \tag{5.16}$$

where

$$a_{i0} \triangleq \frac{1}{C_i}\tilde{a}_{i0}\,, \tag{5.17}$$

$$a_{i1} \triangleq \frac{1}{C_i}\tilde{a}_{i1} \tag{5.18}$$

are weighting coefficients (to be determined later). Now let us decompose d_{i0} into two parts, $d_{i0} = m_iM + l_i$, where $m_i, l_i \geq 0$ are two integers. Then, Eqn (5.16) becomes

$$\begin{aligned} x_i(n) &= a_{i0}\left(\tilde{y}_i(n + m_iM + l_i)\right)_{\downarrow M} + a_{i1}\left(\tilde{y}_i(n + m_iM + l_i + 1)\right)_{\downarrow M} \\ &= a_{i0}\tilde{y}_i^{(l_i)}(n + m_i) + a_{i1}\hat{y}_i^{(l_i+1)}(n + m_i)\,, \end{aligned} \tag{5.19}$$

or in causal form

$$x_i(n - m_i) = a_{i0}\tilde{y}_i^{(l_i)}(n) + a_{i1}\hat{y}_i^{(l_i+1)}(n)\,, \tag{5.20}$$

where

$$\hat{y}_i^{(l_i+1)}(n) = \begin{cases} \tilde{y}_i^{(l_i+1)}(n), & l_i = 0, \ldots, M - 2, \\ \tilde{y}_i^{(0)}(n + 1), & l_i = M - 1, \end{cases} \tag{5.21}$$

i.e. when $l_i = M-1$, $\tilde{y}_i^{(l_i+1)}(n)$ has to map to the 0-th polyphase component with 1 sample advanced ($\tilde{y}_i^{(0)}(n+1)$).

Eqn (5.20) indicates that the input symbols can be recovered by summing together two scaled polyphase components of $\tilde{y}_i(n)$, leading to a one-tap interpolation equalization technique.

5.3 One-Tap Interpolation Equalization

The one-tap interpolation equalization technique uses the interpolation of two certain neighboring polyphase components of the analysis filter output to compensate for the channel non-integer delay and recover the system input. It may result in relatively high performance quantified by the system output SNR. With the structural modification of the original filterbank-based DMT system, the computational complexity of this equalization technique is very low. In this section, the optimal equalizer coefficients are found by using MMSE optimization. Then, the relationship between the channel group-delay and the optimal index of the polyphase components chosen for equalization is obtained. This relationship can be exploited to accelerate the equalizer training procedure.

5.3.1 Optimal Equalizer Settings under MMSE Criterion

The modified structure of the DMT system employing critically decimated filterbanks and a set of one-tap interpolation equalizers is the same as Figure 4.8 in Section 4.4. Presently, the equalizers $\{\mathrm{EQ}_i\}$ ($i = 0, ..., M-1$) in Figure 4.8 are one-tap interpolation equalizers. The block diagram of the i-th one-tap interpolation equalizer is shown in Figure 5.4.

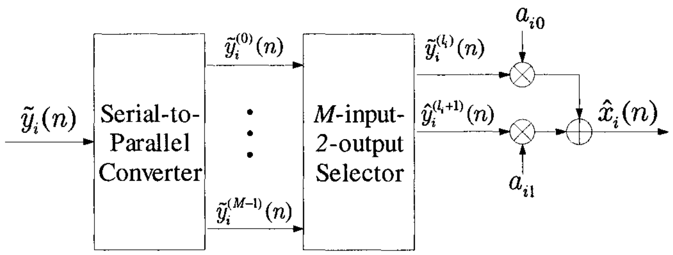


Figure 5.4: The i-th one-tap interpolation equalizer

In this figure, the output $\tilde{y}_i(n)$ of the i-th analysis filter is parsed into its M polyphase components. By using a M-input-2-output selector, two polyphase components ($\tilde{y}_i^{(l_i)}(n)$, and[5] $\hat{y}_i^{(l_i+1)}(n)$) of $\tilde{y}_i(n)$ are selected, weighted by the corresponding 1-tap equalizers a_{i0}

[5] $\hat{y}_i^{(l_i+1)}(n)$ is defined in Eqn (5.21).

and a_{i1}, and subsequently summed together to produce the recovered symbol $\hat{x}_i(n)$. Three parameters have to be determined during equalizer initialization, which are the polyphase component index l_i, and the equalizer coefficients a_{i0} and a_{i1}.

The equalizer output $\hat{x}_i(n)$ can be written in z-domain as

$$\hat{X}_i(z) = a_{i0}\tilde{Y}_i^{(l_i)}(z) + a_{i1}\hat{Y}_i^{(l_i+1)}(z)\,. \tag{5.22}$$

In this section, it is assumed that the filterbanks used in the DMT system are well designed such that no spectral overlap exists and the ICI can be neglected. Also, suppose the channel is noiseless[6]. Then, by substituting Eqns (5.5) and (5.21) into Eqn (5.22), one has

$$\hat{X}_i(z) = X_i(z)\left(a_{i0}S_{ii}^{(l_i)}(z) + a_{i1}\hat{S}_{ii}^{(l_i+1)}(z)\right)\,, \tag{5.23}$$

where

$$\hat{S}_{ii}^{(l_i+1)}(z) = \begin{cases} S_{ii}^{(l_i+1)}(z), & l_i = 0,\ ...,\ M-2, \\ zS_{ii}^{(0)}(z), & l_i = M-1. \end{cases} \tag{5.24}$$

Rewriting Eqn (5.23) in time-domain, one has

$$\hat{x}_i(n) = x_i(n) * \left(a_{i0}s_{ii}^{(l_i)}(n) + a_{i1}\hat{s}_{ii}^{(l_i+1)}(n)\right)\,, \tag{5.25}$$

where

$$\hat{s}_{ii}^{(l_i+1)}(n) = \begin{cases} s_{ii}^{(l_i+1)}(n), & l_i = 0,\ ...,\ M-2, \\ s_{ii}^{(0)}(n+1), & l_i = M-1, \end{cases} \tag{5.26}$$

i.e. $s_{ii}^{(l_i+1)}(n)$ also has to map back to the 0-th polyphase component with 1 sample advanced $\left(\, s_{ii}^{(l_i+1)}(n+1)\,\right)$ when $l_i = M-1$.

Now, define the error sequence between the recovered symbols and the corresponding input symbols as

$$r_i(n) \triangleq \hat{x}_i(n) - x_i(n-k_i)\,, \tag{5.27}$$

where k_i represents the relative delay between recovered and input symbols. Without loss of generality, suppose $s_{ii}^{(l_i)}(n)$ and $\hat{s}_{ii}^{(l_i+1)}(n)$ have the same lengths L_s. Then, $0 \leq k_i \leq L_s$.

The optimal a_{i0}, a_{i1} and l_i can be found by minimizing MMSE $r_i(n)$. Note that when $r_i(n)$ is minimized, the recovered symbol $\hat{x}_i(n)$ is approximately equal to the corresponding delayed system input symbol $x_i(n-k_i)$. Then, from Eqn (5.25), one has

$$a_{i0}s_{ii}^{(l_i)}(n) + a_{i1}\hat{s}_{ii}^{(l_i+1)}(n) \approx \delta(n-k_i)\,, \tag{5.28}$$

[6] The generalization to the case when both ICI and channel noise are considered will be deferred to Section 6.

which means that the ISI is approximately eliminated by using two-point interpolation of two polyphase components of the overall transfer function $S_{ii}(z)$. As mentioned above, the ICI is also eliminated with the assumption of no spectral overlap. Therefore, the system input can be recovered after equalization.

First, let us rewrite Eqn (5.25) in matrix form, in accordance with

$$\begin{aligned}\hat{x}_i(n) &= a_{i0}\begin{bmatrix} s_{ii}^{(l_i)}(0) & \cdots & s_{ii}^{(l_i)}(L_s-1)\end{bmatrix}\begin{bmatrix} x_i(n) \\ \vdots \\ x_i(n-L_s+1)\end{bmatrix} \\ &\quad + a_{i1}\begin{bmatrix} \hat{s}_{ii}^{(l_i+1)}(0) & \cdots & \hat{s}_{ii}^{(l_i+1)}(L_s-1)\end{bmatrix}\begin{bmatrix} x_i(n) \\ \vdots \\ x_i(n-L_s+1)\end{bmatrix} \\ &= \begin{bmatrix} a_{i0} & a_{i1}\end{bmatrix}\begin{bmatrix} s_{ii}^{(l_i)}(0) & \cdots & s_{ii}^{(l_i)}(L_s-1) \\ \hat{s}_{ii}^{(l_i+1)}(0) & \cdots & \hat{s}_{ii}^{(l_i+1)}(L_s-1)\end{bmatrix}\begin{bmatrix} x_i(n) \\ \vdots \\ x_i(n-L_s+1)\end{bmatrix},\end{aligned} \tag{5.29}$$

or in compact form as

$$\hat{x}_i(n) = \boldsymbol{A}_i'\boldsymbol{S}_{ii}\boldsymbol{x}_i(n)\,, \tag{5.30}$$

where

$$\boldsymbol{A}_i = \begin{bmatrix} a_{i0} \\ a_{i0}\end{bmatrix}\,,\ \boldsymbol{S}_{ii} = \begin{bmatrix} s_{ii}^{(l_i)}(0) & \cdots & s_{ii}^{(l_i)}(L_s-1) \\ \hat{s}_{ii}^{(l_i+1)}(0) & \cdots & \hat{s}_{ii}^{(l_i+1)}(L_s-1)\end{bmatrix},$$

and

$$\boldsymbol{x}_i(n) = \begin{bmatrix} x_i(n) & \cdots & x_i(n-L_s+1)\end{bmatrix}^T\,.$$

By substituting Eqn (5.30) into Eqn (5.27), $r_i(n)$ becomes

$$r_i(n) = \hat{x}_i(n) - x_i(n-k_i) = \boldsymbol{A}_i'\,\boldsymbol{S}_{ii}\,\boldsymbol{x}_i(n) - \boldsymbol{u}_{k_i}'\boldsymbol{x}_i(n)\,, \tag{5.31}$$

where $\boldsymbol{u}_{k_i}$ representing the k_i-th unit vector $\begin{bmatrix}\mathbf{0}_{1\times k_i} & 1 & \mathbf{0}_{1\times(L_s-k_i-1)}\end{bmatrix}'$. Then, the MSE of $r_i(n)$ can be obtained as

$$\begin{aligned}\mathrm{MSE}_i &\triangleq E\left[|r_i(n)|^2\right] \\ &= E\left[\left(\boldsymbol{A}_i'\,\boldsymbol{S}_{ii}\,\boldsymbol{x}_i(n) - \boldsymbol{u}_{k_i}'\boldsymbol{x}_i(n)\right)\left(\boldsymbol{A}_i'\,\boldsymbol{S}_{ii}\,\boldsymbol{x}_i(n) - \boldsymbol{u}_{k_i}'\boldsymbol{x}_i(n)\right)'\right]) \\ &= \boldsymbol{A}_i'\boldsymbol{S}_{ii}\boldsymbol{R}_{xx}^{(ii)}\boldsymbol{S}_{ii}'\boldsymbol{A}_i - 2\boldsymbol{A}_i'\boldsymbol{S}_{ii}\boldsymbol{R}_{xx}^{(ii)}\boldsymbol{u}_{k_i} + \boldsymbol{u}_{k_i}'\boldsymbol{R}_{xx}^{(ii)}\boldsymbol{u}_{k_i}\,,\end{aligned} \tag{5.32}$$

where $\boldsymbol{R}_{xx}^{(ii)}$ represents the autocorrelation matrix of $\boldsymbol{x}_i(n)$ in accordance with

$$\boldsymbol{R}_{xx}^{(ii)} \triangleq E[\boldsymbol{x}_i(n)\boldsymbol{x}_i'(n)]\,. \tag{5.33}$$

The optimal solution of Eqn (5.32) can be found by letting the partial derivative of MSE_i with respect to $\boldsymbol{A}_i$ equal to zero, yielding

$$\boldsymbol{S}_{ii}\boldsymbol{R}_{xx}^{(ii)}\boldsymbol{S}_{ii}'\boldsymbol{A}_i = \boldsymbol{S}_{ii}\boldsymbol{R}_{xx}^{(ii)}\boldsymbol{u}_{k_i} \, . \tag{5.34}$$

Then, $\boldsymbol{A}_i$ and MSE_i become

$$\boldsymbol{A}_i = (\boldsymbol{S}_{ii}\boldsymbol{R}_{xx}^{(ii)}\boldsymbol{S}_{ii}')^{-1}\boldsymbol{S}_{ii}\boldsymbol{R}_{xx}^{(ii)}\boldsymbol{u}_{k_i} \, , \tag{5.35}$$

$$\begin{aligned} \text{MSE}_i &= \boldsymbol{u}_{k_i}'\boldsymbol{R}_{xx}^{(ii)}\boldsymbol{u}_{k_i} - \boldsymbol{A}_i'\boldsymbol{S}_{ii}\boldsymbol{R}_{xx}^{(ii)}\boldsymbol{u}_{k_i} \\ &= \boldsymbol{u}_{k_i}'\left(\boldsymbol{R}_{xx}^{(ii)} - \boldsymbol{R}_{xx}^{(ii)}\boldsymbol{S}_{ii}'\left(\boldsymbol{S}_{ii}\boldsymbol{R}_{xx}^{(ii)}\boldsymbol{S}_{ii}'\right)^{-1}\boldsymbol{S}_{ii}\boldsymbol{R}_{xx}^{(ii)}\right)\boldsymbol{u}_{k_i} \, . \end{aligned} \tag{5.36}$$

Note that because $\boldsymbol{S}_{ii}$ is determined by l_i, the MMSE of $r_i(n)$ depends on the choice of l_i. Moreover, for a given l_i, the MMSE of $r_i(n)$ also depends on k_i. The optimal k_i can be obtained in a straightforward fashion as follows.

Let us define $\boldsymbol{A}_{i,\,\text{all}}$ as

$$\boldsymbol{A}_{i,\,\text{all}} = (\boldsymbol{S}_{ii}\boldsymbol{R}_{xx}^{(ii)}\boldsymbol{S}_{ii}')^{-1}\boldsymbol{S}_{ii}\boldsymbol{R}_{xx}^{(ii)} \, . \tag{5.37}$$

Then, Eqn (5.35) can be rewritten in the form

$$\boldsymbol{A}_i = \boldsymbol{A}_{i,\,\text{all}}\,\boldsymbol{u}_{k_i} \, , \tag{5.38}$$

while Eqn (5.36) becomes

$$\begin{aligned} \text{MSE}_i &= \boldsymbol{u}_{k_i}'(\boldsymbol{I} - \boldsymbol{A}_{i,\,\text{all}}'\boldsymbol{S}_{ii})\boldsymbol{R}_{xx}^{(ii)}\boldsymbol{u}_{k_i} \\ &= \boldsymbol{u}_{k_i}'\mathcal{E}_i\boldsymbol{u}_{k_i} \, , \end{aligned} \tag{5.39}$$

where $\boldsymbol{I}$ is the $L_s \times L_s$ identity matrix and $\mathcal{E}_i$ is defined as:

$$\mathcal{E}_i = (\boldsymbol{I} - \boldsymbol{A}_{i,\,\text{all}}'\boldsymbol{S}_{ii})\boldsymbol{R}_{xx}^{(ii)} \, . \tag{5.40}$$

Now, the MMSE of $r_i(n)$ is the entry with the minimum value on the main diagonal of $\mathcal{E}_i$, i.e.

$$\text{MMSE}_i = \min\{\text{diag}(\mathcal{E}_i)\} \, , \tag{5.41}$$

where diag$(\cdot)$ represents the main diagonal of the matrix $(\cdot)$. The optimal k_i becomes the index that achieves the MMSE, in accordance with

$$k_{i,\text{opt.}} = \arg\min_{0\leq k_i\leq L_s}\{\text{diag}(\mathcal{E}_i)\} \, . \tag{5.42}$$

After finding $k_{i,\text{opt.}}$, the optimal equalizer coefficient matrix $\boldsymbol{A}_i$ becomes

$$\boldsymbol{A}_{i,\text{opt.}} = \boldsymbol{A}_{i,\,\text{all}}\,\boldsymbol{u}_{k_{i,\text{opt.}}} \, . \tag{5.43}$$

Now assume that all inputs for various subchannels are mutually uncorrelated, and that all inputs within each subchannel are mutually uncorrelated having a constant PSD σ_x^2 over full channel bandwidth. With these assumptions, $\boldsymbol{R}_{xx}^{(ii)}$ can be simplified as:

$$\boldsymbol{R}_{xx}^{(ii)} = \sigma_x^2 \boldsymbol{I} . \tag{5.44}$$

Then, Eqn (5.37) and Eqn (5.40) can be simplified as:

$$\boldsymbol{A}_{i,\,\text{all}} = (\boldsymbol{S}_{ii}\boldsymbol{S}_{ii}')^{-1}\boldsymbol{S}_{ii} , \tag{5.45}$$

and

$$\mathcal{E}_i = \sigma_x^2(\boldsymbol{I} - \boldsymbol{A}_{i,\,\text{all}}'\boldsymbol{S}_{ii}) . \tag{5.46}$$

Therefore, the optimal equalizer coefficient matrix $\boldsymbol{A}_{i,\text{opt.}}$ and the MMSE of $r_i(n)$ are simplified as

$$\boldsymbol{A}_{i,\text{opt.}} = (\boldsymbol{S}_{ii}\boldsymbol{S}_{ii}')^{-1}\boldsymbol{S}_{ii}\boldsymbol{u}_{k_{i,\text{opt.}}} , \tag{5.47}$$

and

$$\text{MMSE}_i = \sigma_x^2(1 - \boldsymbol{A}_i'\boldsymbol{S}_{ii}\boldsymbol{u}_{k_{i,\text{opt.}}}) . \tag{5.48}$$

To find the optimal equalizer settings, a search for all possible $l_i \in \{0, ..., M-1\}$ is needed to find the MMSE between the system input and the corresponding system output. When M is large, the search for l_i involves a heavy computational task. But as the channel is known, its characteristics can be fully exploited to narrow down the search space for l_i. A discussion of the relationship between the optimal l_i and channel group-delay is given in Section 5.3.3.

It should be noted that as $\boldsymbol{A}_i$ consists of only two coefficients a_{i0} and a_{i1}, its computational overhead is very low. It should also be noted that the adverse effect of subchannel magnitude and phase distortions becomes relatively more pronounced as the number of subchannels M becomes smaller, mainly due to the fact that the subchannel cannot be approximated as a delayed ideal channel any longer. In such situations, the one-tap interpolation equalization technique should be extended to multi-tap equalization.

5.3.2 Simulation Results

In this section, all assumptions and design parameters are the same as those in Section 4.5. Also, the output SIRs for different channel cases are compared in this section with the assumption of noiseless channel. Figures 5.5–5.8 show the output SIRs for CSA loops #1, #2, #3, and #6 when $M = 64$ and $M = 128$. The SIRs for other CSA loops are shown in

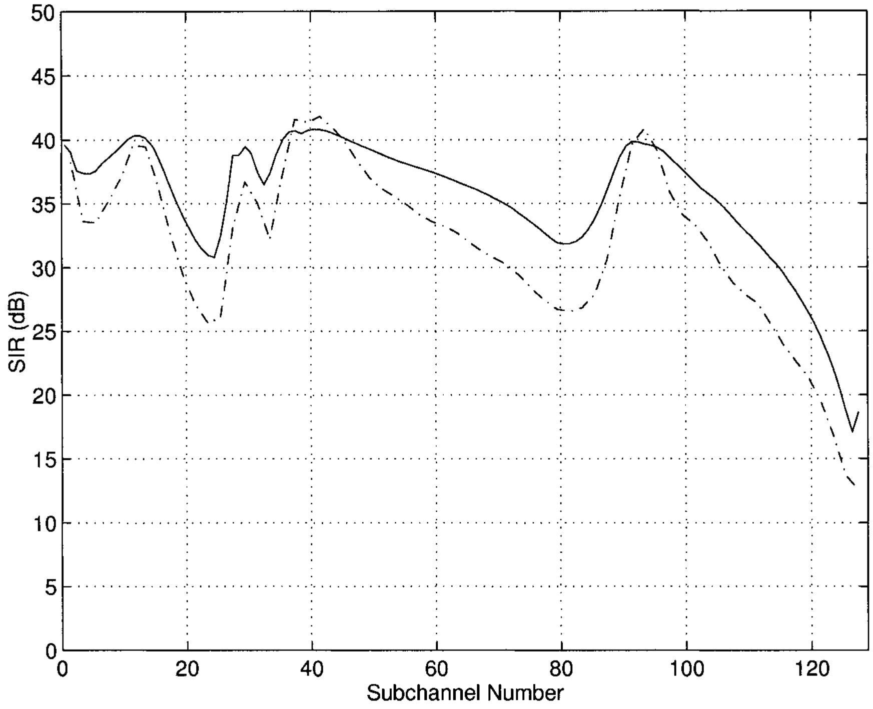


Figure 5.5: SIRs for one-tap interpolation equalization for CSA loop #1 when $M = 64$ and $M = 128$

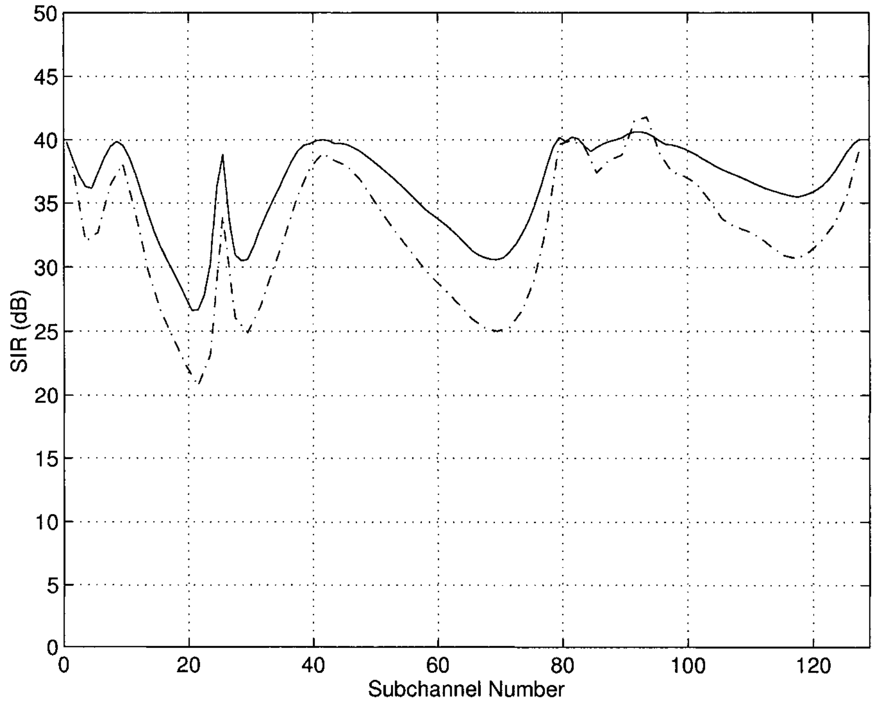


Figure 5.6: SIRs for one-tap interpolation equalization for CSA loop #2 when $M = 64$ and $M = 128$

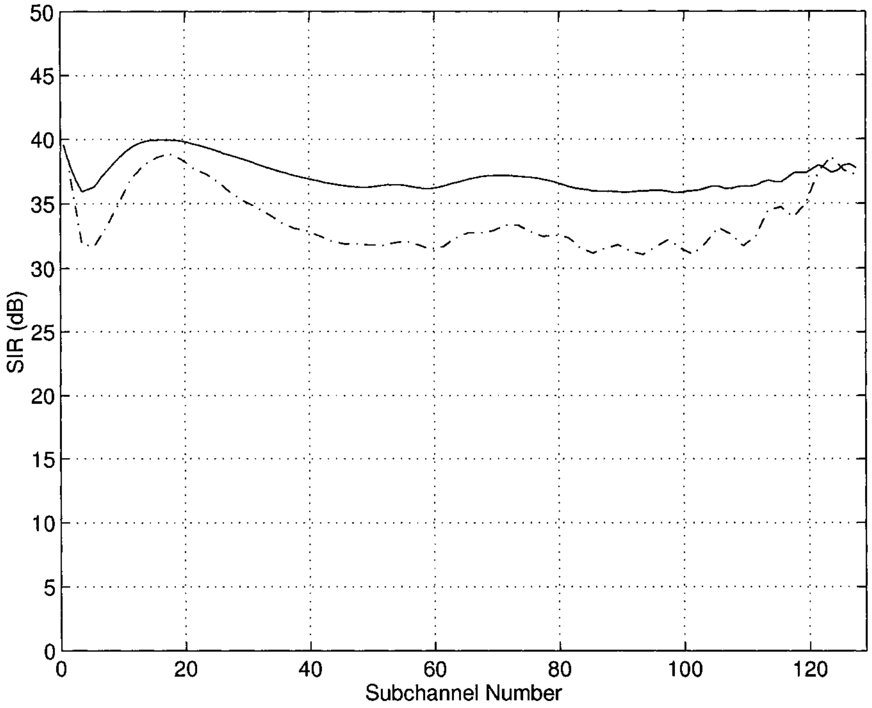


Figure 5.7: SIRs for one-tap interpolation equalization for CSA loop #3 when $M = 64$ and $M = 128$

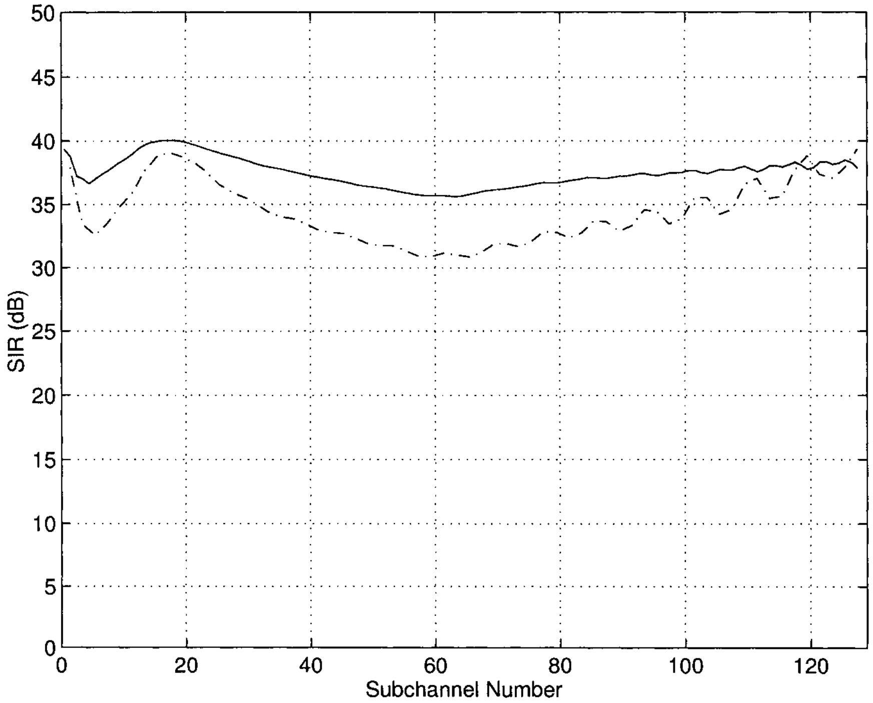


Figure 5.8: SIRs for one-tap interpolation equalization for CSA loop #6 when $M = 64$ and $M = 128$

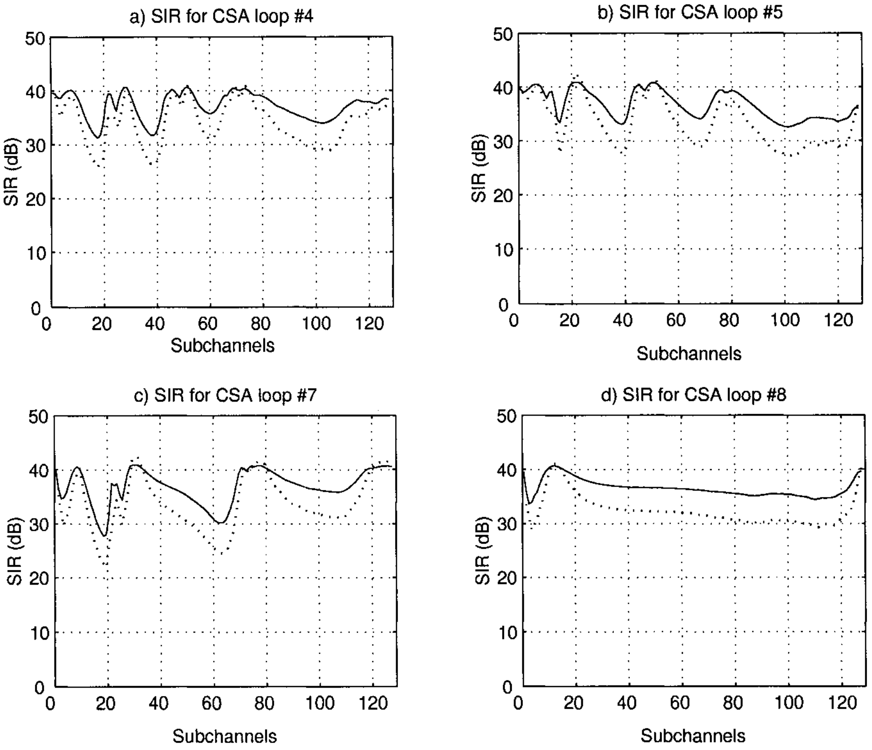


Figure 5.9: SIRs for one-tap interpolation equalization for CSA loops #4, #5, #7, and #8 when $M = 64$ and $M = 128$

Figure 5.9. In these figures, the dash-dot curves (or dotted curves in Figure 5.9) represent the SIRs for $M = 64$ while the solid curves represent the SIRs for $M = 128$. Note that the subchannel numbers for $M = 64$ are scaled by 2 in order to be consistent with the subchannel numbers for $M = 128$. In case of an ideal channel ($c(n) = \delta(n)$), the SIRs for both $M = 64$ and $M = 128$ are about 42 dB (c.f. Figure 4.10).

From these figures, one can see that the one-tap interpolation equalization can obtain a moderate SIR ratio. In each channel case, the SIRs for most subchannels are higher than 30dB. Some of them are close to the SIR of the ideal channel case, which are determined by the design of the CMFB[7]. Also, with increasing M, the output SIR increases. One can expect that with a better design for CMFB that makes the CMFB closer to a PR filterbank, the output SIR will be higher.

Next, by comparing SIRs in Figures 5.5–5.9 with channel group-delays shown in Appendix A, it is clearly noticed that the output SIRs are also determined by the channel group-delay. For example, in Figure A.1c, the group-delay of CSA loop #1 has two dips around $\omega = 0.22\pi$ and $\omega = 0.68\pi$, respectively. In the regions around these two frequency

[7] Recall that the CMFB used here is a type of pseudo PR filterbank. Therefore, distortions (although very small) exist in the system output.

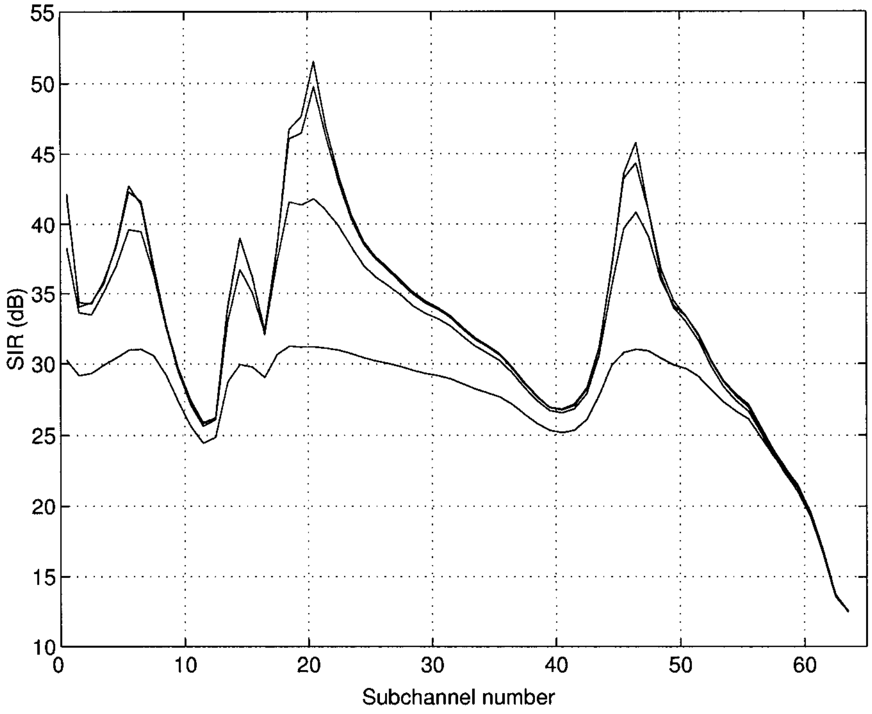


Figure 5.10: SIRs for one-tap interpolation equalization for CSA loop #1 where $M = 64$, and $g = 12, 8, 6, 4$ (top to bottom)

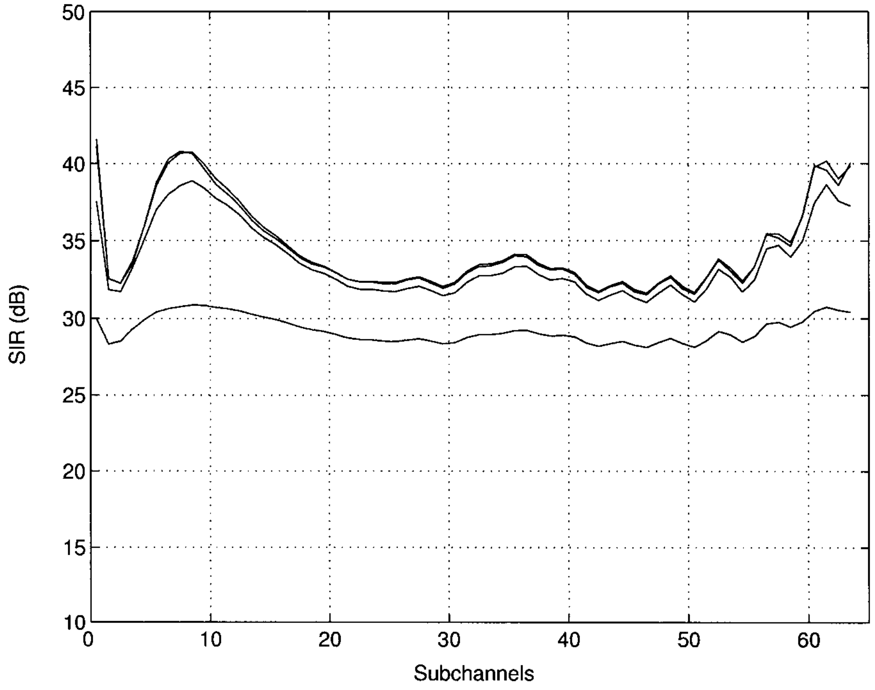


Figure 5.11: SIRs for one-tap interpolation equalization for CSA loop #3 where $M = 64$, and $g = 12, 8, 6, 4$ (top to bottom)

points, the channel group-delay has large fluctuation. This means that the assumption of the subchannel linear phase response is not satisfied in these frequency regions. Therefore, in Figure 5.5, the SIRs of subchannels #20–#30 and #70–#90 for $M = 128$ (or subchannels #10–#15 and #35–#45 for $M = 64$, which are subchannels whose center frequencies are around $\omega = 0.22\pi$ and $\omega = 0.68\pi$, respectively) are relatively low. On the other hand, the channel group-delays of CSA loops #3, #6, and #8 (c.f. Figures A.3c, A.6c, and A.8c, respectively) are relatively flat (i.e. the assumption of the channel linear phase response is satisfied very well[8]). Therefore, the corresponding SIRs (c.f. Figures 5.7, 5.8, and 5.9d, respectively) are relatively high and have no dips. With this comparison, one can conclude that the performance of the one-tap interpolation equalization (measured by the output SIR) is highly determined by the channel group-delay. Moreover, there is also a simple relationship between the channel group-delay and the optimal l_i (the index of the optimal polyphase components of the analysis filter output used for equalization). This relationship will be shown in Section 5.3.3.

Besides the number M of subchannels and the channel group-delay discussed above, the output SIR is also affected by the ICI. Note that the ICI is determined by the level of the spectral overlap between two subchannels. The larger the spectral overlap, the more severe is the ICI. For CMFB system, recall that the spectral overlap is determined by the overlap factor g (c.f. Section 2.5.2). Then, the output SIR is affected by the overlap factor g. Figures 5.10 and 5.11 show the relationship between the output SIR and various g for CSA loops #1 and #3 when $M = 64$. In each figure, the four curves, from top to bottom, represent the SIRs for $g = 12, 8, 6, 4$. One can see that with g increasing (i.e. the length of the prototype filter increasing and the spectral overlap decreasing), the output SIR increases. Note that in both figures, the output SIR for $g = 12$ is very close to that for $g = 8$. This is because in the frequency domain, two neighboring filters of CMFB cross at their cut-off frequencies, which means the spectral overlap always exists no matter how sharp the transition band of the prototype filter is. Therefore, after the length of the prototype filter exceeds a special limit, the spectral overlap (and then the ICI) would not significantly decrease.

5.3.3 Relationship between Optimal l_i and Channel Group-Delay

As mentioned in Section 4.1, in each subchannel of a filterbank-based DMT system, the channel effect can be approximated as piecewise flat magnitude and piecewise linear phase

[8] A similar conclusion can also be drawn for these channels by examining the magnitude responses of these channels.

response in accordance with Eqn (4.2). Although the subchannel approximation of $|C(e^{j\omega})|$ by C_i is quite acceptable, experimental investigations indicate that the subchannel approximation of $\phi(\omega)$ by $d_i\omega$ maybe quite inaccurate. This is due to the fact that although $\phi(\omega)$ is approximately a straight line, it possesses ripples that cannot be neglected.

Let $\omega_1 \leq \omega_0 \leq \omega_2$ represents an arbitrary frequency in the i-th subchannel frequency band, where ω_1 and ω_2 are the left and right cut-off frequencies of the i-th subchannel, respectively. Then, the channel phase response within this subchannel can be written in the form

$$\begin{aligned}\phi(\omega) &= \phi(\omega_0) + \phi'(\omega_0)(\omega - \omega_0) + o((\omega - \omega_0)^2) \\ &= \phi'(\omega_0)\omega + \left(\frac{\phi(\omega_0)}{\omega_0} - \phi'(\omega_0)\right)\omega_0 + o((\omega - \omega_0)^2)\,,\end{aligned} \tag{5.49}$$

where $\omega_1 \leq \omega \leq \omega_2$, $\phi'(\omega_0)$ is the first order derivative of $\phi(\omega)$ with respect to ω at $\omega = \omega_0$. By definition, $\phi'(\omega_0)$ represents the channel group-delay at $\omega = \omega_0$. Similarly, $\phi(\omega_0)/\omega_0$ represents the channel phase delay at ω_0. In this way, once the frequency ω_0 has been fixed, the second term in the right hand side (RHS) of Eqn (5.49)becomes a constant (independent of ω). Based on the specific choice of ω_0, the second and third terms in the RHS of Eqn (5.49) can be ignored and the subchannel phase response can be approximated as:

$$\phi(\omega) \approx \phi'(\omega_0)\omega\,. \tag{5.50}$$

In this way, the equalization has to compensate for $\phi'(\omega_0)$.

Eqn (5.50) means that the optimal l_i can be represented by $\phi'(\omega_0)$, which implies that the optimal l_i is within the range of subchannel group-delay. Now, if the range of the subchannel group-delay is known, only those l_i within this range need to be searched to find the optimal l_i (instead of searching through all M possible l_i). Thus, the training of equalizers can be significantly simplified.

Figure 5.12 illustrates the relationship between the optimal l_i and the channel group-delay for CSA loop #1. The grey curve represents the channel group-delay for CSA loop #1, and the black curve represents the optimal l_i. It is clear that the optimal l_i of most subchannels are within the range of channel-group-delay. To further avoid the possible deviation between the optimal l_i and the range of channel group-delay, one can enlarge the search range from the both end of the range of subchannel group-delay. For example, if the range of a subchannel group-delay is $[d_{i1},\, d_{i2}]$, then, the search range can be chosen as $[d_{i1} - d_\varepsilon,\, d_{i2} + d_\varepsilon]$, where d_ε is an arbitrary positive constant, and if a delay d_{ik} ($\in [d_{i1} - d_\varepsilon,\, d_{i2} + d_\varepsilon]$) is greater than $M - 1$ or less than 0, it should be represented by its principal-value by subtracting or adding suitable integer multiples of M.

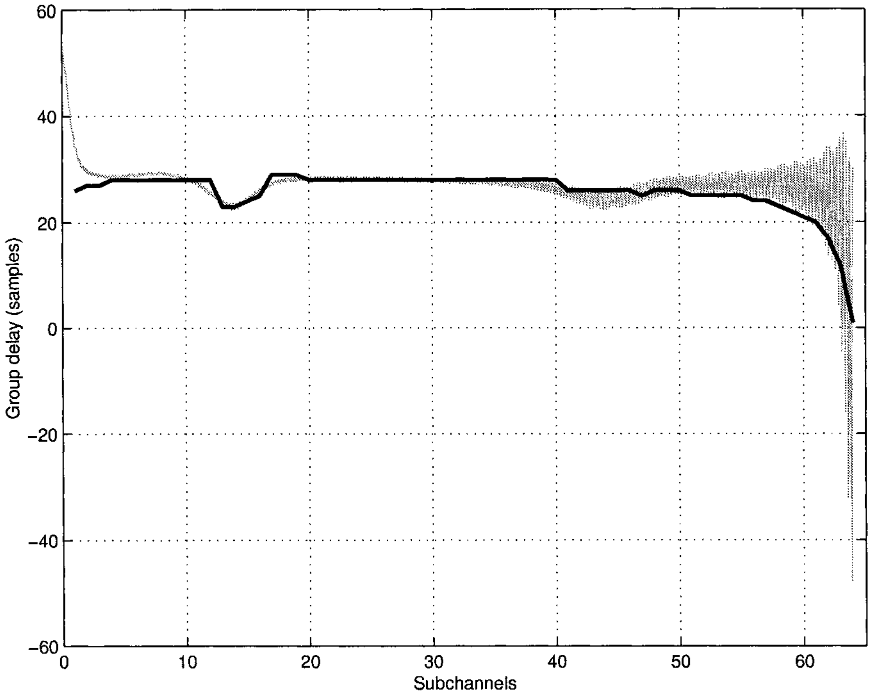


Figure 5.12: Comparison of optimal l_i and channel group-delay for CSA loop #1 where $M = 64$

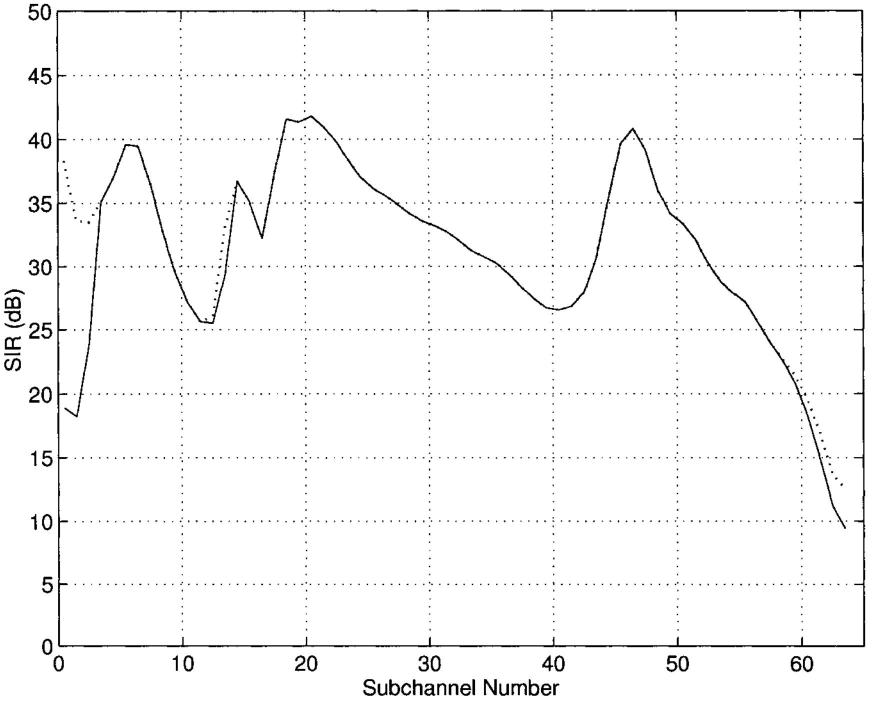


Figure 5.13: SIRs for one-tap interpolation equalization ($M = 64$) for CSA loop #1 when calculating optimal l_i within an enlarged range of subchannel group-delay

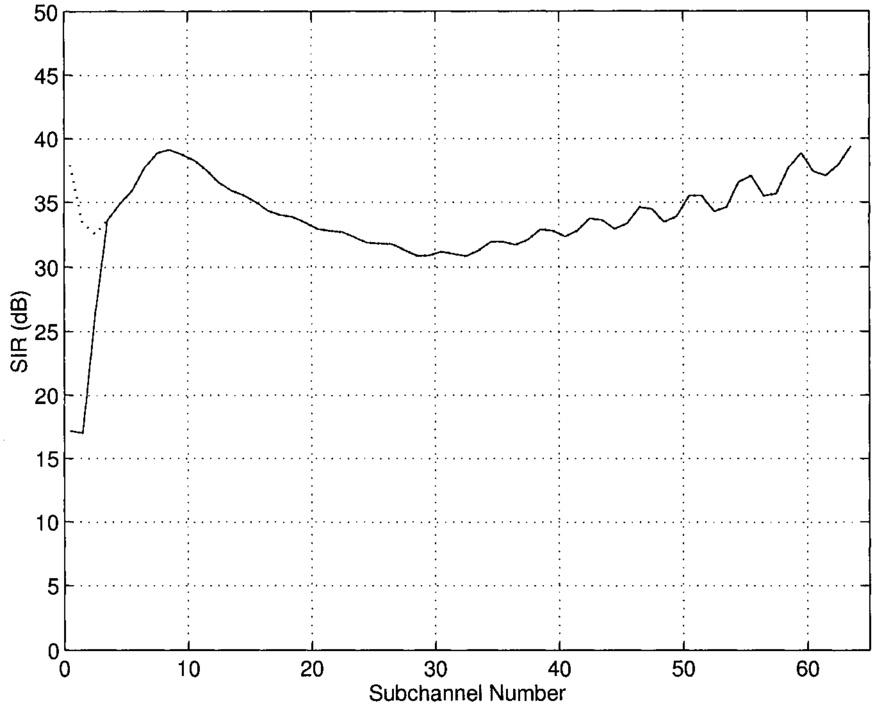


Figure 5.14: SIRs for one-tap interpolation equalization ($M = 64$) for CSA loop #6 when calculating optimal l_i within an enlarged range of subchannel group-delay

Figures 5.13 and 5.14 show the comparison of SIRs for CSA loops #1 and #6 when searching the optimal l_i among all M possible choices, and within an enlarged range based on the range of subchannel group-delay ($d_\varepsilon = 5$). In each figure, the dotted curve represents the SIR when searching the optimal l_i among all M possible choices while the solid curve represents the SIR when searching the optimal l_i within the enlarged range based on the range of subchannel group-delay. From these two figures, one can see that the deviation only occurs in several lower frequency subchannels. However, by recalling that usually the lower frequency subchannels are reserved for telephone communication, this deviation would not affect the transmission bit-rate of a practical DMT system.

Chapter 6

Multi-Tap Interpolation Equalization

The one-tap interpolation equalization technique in the previous chapter is based on channel approximation of piecewise constant magnitude and piecewise linear phase response. When M is relatively small and/or channel phase response fluctuates rapidly, channel magnitude and phase responses cannot be adequately regarded as being piecewise flat. In this case, one can combine more polyphase components of $\tilde{y}_i(n)$ and/or use multi-tap equalizers for each polyphase component to equalize the channel, leading to the multi-tap interpolation equalization technique [58].

Multi-tap interpolation equalization is the generalization or extension of the integer-delay equalization and one-tap interpolation equalization technique. It gives rise to a high system output SNR, while requiring a small number of equalizer taps. Moreover, it permits a tradeoff between various equalization parameters, leading to high computational flexibility.

6.1 Structure of Multi-Tap Interpolation Equalizer

The structure of the DMT system employing critically decimated filterbanks and a set of multi-tap interpolation equalizers is the same as Figure 4.8 in Section 4.4, while each equalizer EQ$_i$ ($i = 0$, ..., $M - 1$) becomes a multi-tap interpolation equalizer. Similarly, the block diagram of the i-th multi-tap interpolation equalizer is shown in Figure 6.1.

In this figure, the output $\tilde{y}_i(n)$ of the i-th analysis filter is decomposed into its M polyphase components $\tilde{y}_i^{(0)}(n)$, ..., $\tilde{y}_i^{(M-1)}(n)$, among which L successive polyphase components $\{\tilde{y}_i^{(l_i+k)}(n)\}$, ($k = 0$, ..., $L - 1$) are selected by an M-input-L-output selector. Each selected polyphase component $\tilde{y}_i^{(l_i+k)}(n)$ passes a corresponding multi-tap equalizer $A_{i,k}(z)$ (associating with the impulse response $a_{i,k}(n)$, where $k = 0$, ..., $L-1$). Then, the L filtered

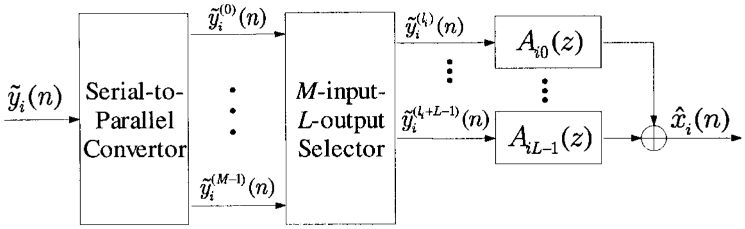


Figure 6.1: Multi-tap interpolation equalization scheme

polyphase components are summed together to obtain the recovered symbols.

Before further discussion, let us decompose the recovered symbols $\hat{x}_i(n)$ into two parts as

$$\hat{x}_i(n) = \hat{x}_{i,s}(n) + \hat{x}_{i,n}(n)\,, \tag{6.1}$$

where

$$\hat{x}_{i,s}(n) = \hat{x}_i(n)|_{e(n)=0} \tag{6.2}$$

represents the output of the signal path in Figure 4.8 and 6.1. On the other hand,

$$\hat{x}_{i,n}(n) = \hat{x}_i(n)|_{x_i(n)=0,\, i=0,\ldots,M-1} \tag{6.3}$$

represents the output of the corresponding noise path in Figure 4.8 and 6.1. With this decomposition, the signal path and the noise path will be examined separately in the following sections. A closed form solution is developed for the optimal equalizer settings. Moreover, In order to give a more general discussion and to obtain a more precise solution of the multi-tap equalizer settings, ICI terms are considered in the optimization of equalizer coefficients.

6.2 Analysis of Signal Path

By considering the signal path in Figure 4.8 and 6.1, $\hat{x}_{i,s}(n)$ can be written as

$$\begin{aligned} \hat{x}_{i,s}(n) &= \sum_{k=0}^{L-1} a_{i,k}(n) * \tilde{y}_{i,s}^{(l_i+k)}(n) \\ &= \sum_{m=0}^{M-1}\sum_{k=0}^{L-1} x_m(n) * a_{i,k}(n) * s_{mi}^{(l_i+k)}(n)\,, \end{aligned} \tag{6.4}$$

where $\tilde{y}_{i,s}^{(l_i+k)}(n)$ is the (l_i+k)-th polyphase component of $\tilde{y}_{i,s}(n)$, and $\tilde{y}_{i,s}(n) = \tilde{y}_i(n)|_{e(n)=0}$. Note that with a discussion similar to that in section 5.3, when $l_i + L - 1 \geq M$, $s_{mi}^{(l_i+k)}(n)$ $(l_i + k \geq M)$ in Eqn (6.4) has to be mapped to $s_{mi}^{(l_i+k-M)}(n + l_i + k - M + 1)$.

Suppose all $\{s_{mi}^{(l_i+k)}(n)\}$ have the same lengths L_s, and all $\{a_{i,k}(n)\}$ have the same lengths L_a. Then, Eqn (6.4) can be expressed in matrix form as

$$\hat{x}_{i,s}(n) = \sum_{m=0}^{M-1} \sum_{k=0}^{L-1} \boldsymbol{A}'_{ik} \boldsymbol{S}_{mi}^{(k)} \boldsymbol{x}_m(n)\,, \tag{6.5}$$

where

$$\boldsymbol{A}_{ik} = \begin{bmatrix} a_{i,k}(0) \\ a_{i,k}(1) \\ \vdots \\ a_{i,k}(L_a-1) \end{bmatrix}, \quad \boldsymbol{x}_m(n) = \begin{bmatrix} x_m(n) \\ x_m(n-1) \\ \vdots \\ x_m(n-L_s-L_a+2) \end{bmatrix},$$

and

$$\boldsymbol{S}_{mi}^{(k)} = \begin{bmatrix} s_{mi}^{(l_i+k)}(0) & \cdots & s_{mi}^{(l_i+k)}(L_s-1) & 0 & \cdots & 0 \\ 0 & s_{mi}^{(l_i+k)}(0) & \cdots & s_{mi}^{(l_i+k)}(L_s-1) & 0 & \cdots \\ \vdots & & \vdots & & & \vdots \\ 0 & \cdots & 0 & s_{mi}^{(l_i+k)}(0) & \cdots & s_{mi}^{(l_i+k)}(L_s-1) \end{bmatrix}.$$

To find the optimal equalizer settings, let us form a new equalizer coefficient vector $\boldsymbol{A}_i$ which contains the coefficients of all equalizers, and a new matrix $\boldsymbol{S}_{mi}$ which contain all polyphase components of $s_{mi}(n)$ used in Eqn (6.5) in accordance with

$$\boldsymbol{A}_i = \begin{bmatrix} \boldsymbol{A}_{i0} \\ \boldsymbol{A}_{i1} \\ \vdots \\ \boldsymbol{A}_{i(L-1)} \end{bmatrix}, \text{ and } \boldsymbol{S}_{mi} = \begin{bmatrix} \boldsymbol{S}_{mi}^{(0)} \\ \boldsymbol{S}_{mi}^{(1)} \\ \vdots \\ \boldsymbol{S}_{mi}^{(L-1)} \end{bmatrix}.$$

Then, Eqn (6.5) can be rewritten in a more compact form as

$$\hat{x}_{i,s}(n) = \sum_{m=0}^{M-1} \boldsymbol{A}'_i \boldsymbol{S}_{mi} \boldsymbol{x}_m(n)\,. \tag{6.6}$$

In Eqn (6.6) $\big($ as well as in Eqns (6.4) and (6.5) $\big)$, the terms with $m \neq i$ are ICI terms. In practical situations, ICI between the i- and m-th subchannels decreases with increasing the "distance" $|i-m|$ between these two subchannels. Therefore, some of the ICI terms in Eqn (6.6) can be neglected[1]. In this case, Eqn (6.6) can be modified as Eqn (6.7), taking into account ICI terms from $2N_t$ neighboring subchannels symmetrically centered around the i-th subchannel.

$$\hat{x}_{i,s}(n) = \sum_{m=i-N_t}^{i+N_t} \boldsymbol{A}'_i \boldsymbol{S}_{mi} \boldsymbol{x}_m(n)\,, \tag{6.7}$$

where $N_t \leq M/2$ is an integer, and $\boldsymbol{S}_{mi}$ has to map to $\boldsymbol{S}_{(m+M)i}$ $\big($or $\boldsymbol{S}_{(m-M)i}\big)$ when $m < 0$ (or $m \geq M$).

[1] In actual circumstances, it is sufficient to consider ICI between adjacent subchannels only.

6.3 Analysis of Noise Path

The i-th subchannel of the modified DMT receiver employing multi-tap interpolation equalizers is redrawn in Figure 6.2. In this figure, for the sake of simplicity, only two polyphase components of the analysis filter output are used to recover the system input (i.e. $L = 2$). The cases when $L \neq 2$ can be obtained through the extension of the case $L = 2$. In Figure 6.2, two pathes are drawn, where each path corresponds to a polyphase component of the analysis filter output filtered by the corresponding multi-tap interpolation equalizer. In order to find the noise model at the system output, the S/P converter and the M-input-L-output selector in Figure 6.1 are replaced by an advance component followed by a M-fold decimator. In this figure, $e(n)$ represents the additive channel noise while $\tilde{y}_{i,n}(n)$, $\tilde{y}_{i,n}^{(l_i)}(n)$, $\tilde{y}_{i,n}^{(l_i+1)}(n)$, and $\hat{x}_{i,n}(n)$ represent the channel noise path components of $\tilde{y}_i(n)$, $\tilde{y}_i^{(l_i)}(n)$, $\tilde{y}_i^{(l_i+1)}(n)$, and $\hat{x}_i(n)$, respectively.

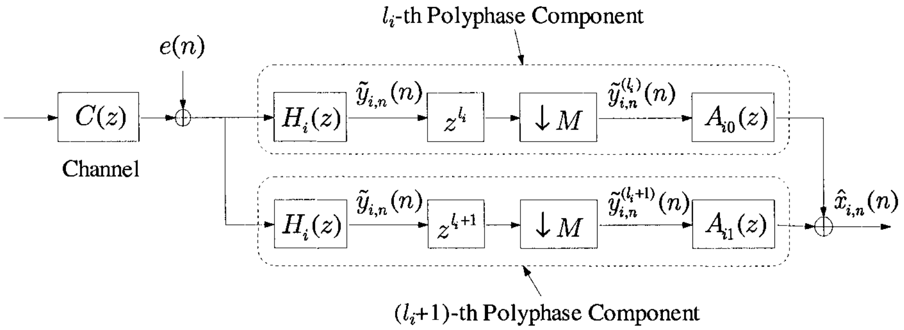


Figure 6.2: The noise path in the i-th subchannel of DMT receiver employing multi-tap interpolation equalizer

Now consider the channel noise path component $\tilde{y}_{i,n}^{(l_i)}(n)$ in the l_i-th polyphase component of $\tilde{y}_i(n)$ (c.f. the dashed box region in Figure 6.2). First, rewrite $H_i(z)$ using its Type 1 polyphase representation with respect to M in accordance with

$$H_i(z) = \sum_{m=0}^{M-1} z^{-m} H_i^{(m)}(z^M) . \tag{6.8}$$

Then the z-transform of $\tilde{y}_{i,n}^{(l_i)}(n)$ can be written as:

$$\begin{aligned} \tilde{Y}_{i,n}^{(l_i)}(z) &= \left(E(z) z^{l_i} H_i(z)\right)_{\downarrow M} \\ &= \left(E(z) z^{l_i} \sum_{m=0}^{M-1} z^{-m} H_i^{(m)}(z^M)\right)_{\downarrow M} \\ &= \sum_{m=0}^{M-1} \left(E(z) z^{l_i - m}\right)_{\downarrow M} H_i^{(m)}(z) . \end{aligned} \tag{6.9}$$

Eqn (6.9) means that $\tilde{y}_{i,n}^{(l_i)}(n)$ can be obtained by using a modified structure[2] as shown in Figure 6.3. This modified structure can also be derived graphically by using noble identities [52]. $\tilde{y}_{i,n}^{(l_i+1)}(n)$ can be obtained in a similar manner.

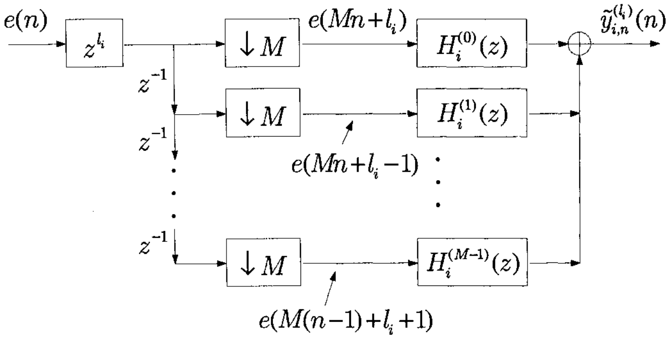


Figure 6.3: Modified structure for the l_i-th polyphase component of analysis filter output in Figure 6.2

It is easy to show that the noise component at the output of m-th decimator ($m = 0, ..., M-1$) in Figure 6.3 is $e(Mn+l_i-m)$. If the noise $e(n)$ is AWGN, all $\{e(Mn+l_i-m)\}$ (for any l_i and m) are also AWGN and independent of each other because only delay chain and decimators are passed [52]. Now by incorporating Figure 6.3 in Figure 6.2, $\hat{x}_{i,n}(n)$ can be written as

$$\begin{aligned}\hat{x}_{i,n}(n) &= \sum_{m=0}^{M-1} e(Mn+l_i-m) * h_i^{(m)}(n) * a_{i0}(n) \\ &+ \sum_{m=0}^{M-1} e(Mn+l_i+1-m) * h_i^{(m)}(n) * a_{i1}(n)\,.\end{aligned} \tag{6.10}$$

By factoring the terms in the first and second summations in Eqn (6.10), $\hat{x}_{i,n}(n)$ can be

[2] This structure can also be used in signal path.

rewritten as

$$
\begin{aligned}
\hat{x}_{i,n}(n) &= \sum_{m=1}^{M} e(Mn+l_i+1-m) * h_i^{(m-1)}(n) * a_{i0}(n) \\
&\quad + \sum_{m=0}^{M-1} e(Mn+l_i+1-m) * h_i^{(m)}(n) * a_{i1}(n) \\
&= \sum_{m=1}^{M-1} e(Mn+l_i+1-m) * \left(h_i^{(m-1)}(n) * a_{i0}(n) + h_i^{(m)}(n) * a_{i1}(n)\right) \\
&\quad + e(M(n-1)+l_i+1) * h_i^{(M-1)}(n) * a_{i0}(n) \\
&\quad + e(Mn+l_i+1) * h_i^{(0)}(n) * a_{i1}(n) \\
&= \sum_{m=1}^{M-1} e(Mn+l_i+1-m) * \left(h_i^{(m-1)}(n) * a_{i0}(n) + h_i^{(m)}(n) * a_{i1}(n)\right) \\
&\quad + e(Mn+l_i+1) * \Big(h_i^{(M-1)}(n-1) * a_{i0}(n) \\
&\quad + e(Mn+l_i+1) * h_i^{(0)}(n) * a_{i1}(n)\Big) \\
&= \sum_{m=0}^{M-1} e(Mn+l_i+1-m) * \left(\hat{h}_i^{(m)}(n) * a_{i0}(n) + h_i^{(m)}(n) * a_{i1}(n)\right),
\end{aligned}
\tag{6.11}
$$

where

$$
\hat{h}_i^{(m)}(n) = \begin{cases} h_i^{(M-1)}(n-1), & m=0, \\ h_i^{(m-1)}(n), & m=1, \ldots, M-1. \end{cases} \tag{6.12}
$$

In the last two steps of Eqn (6.11), the following identity is used:

$$
x(n-1) * y(n) * z(n) = x(n) * y(n-1) * z(n). \tag{6.13}
$$

Eqn (6.11) can be easily extended to the case when $L > 2$ polyphase components of $\tilde{y}_i(n)$ are used in equalization. In this case, $\hat{x}_{i,n}(n)$ becomes

$$
\hat{x}_{i,n}(n) = \sum_{m=0}^{M-1} \sum_{k=0}^{L-1} e(Mn+l_i+1-m) * h_{ik}^{(m)}(n) * a_{ik}(n), \tag{6.14}
$$

where

$$
h_{ik}^{(m)}(n) = \begin{cases} h_i^{(m-L+k+1)}(n), & m-L+k+1 \geq 0, \\ h_i^{(m-L+k+1+M)}(n+m-L+k+1), & m-L+k+1 < 0. \end{cases} \tag{6.15}
$$

From Eqn (6.14) (or Eqn (6.11) when $L = 2$), the contribution of the channel AWGN to the i-th subchannel of the system output can be considered as the sum of M AWGN components.

Suppose the length of $h_i(n)$ is L_h. Then, Eqn (6.14) can be rewritten in matrix form as

$$\hat{x}_{i,n}(n) = \sum_{m=0}^{M-1} \boldsymbol{A}_i' \boldsymbol{H}_{mi} \boldsymbol{e}(Mn + l_i + 1 - m)\,, \tag{6.16}$$

where

$$\boldsymbol{e}(Mn + l_i + 1 - m) = \begin{bmatrix} e(Mn + l_i + 1 - m) \\ e(Mn + l_i + 1 - m - 1) \\ \vdots \\ e(Mn + l_i + 1 - m - L_s - L_a + 2) \end{bmatrix}, \boldsymbol{H}_{mi} = \begin{bmatrix} \boldsymbol{H}_{i0}^{(m)} \\ \boldsymbol{H}_{i1}^{(m)} \\ \vdots \\ \boldsymbol{H}_{i(L-1)}^{(m)} \end{bmatrix},$$

and

$$\boldsymbol{H}_{ik}^{(m)} = \begin{bmatrix} h_{ik}^{(m)}(0) & \cdots & h_{ik}^{(m)}(L_{h|M}-1) & 0 & \cdots & 0 \\ 0 & h_{ik}^{(m)}(0) & \cdots & h_{ik}^{(m)}(L_{h|M}-1) & 0 & \cdots \\ \vdots & & \vdots & & & \vdots \\ 0 & \cdots & 0 & h_{ik}^{(m)}(0) & \cdots & h_{ik}^{(m)}(L_{h|M}-1) \end{bmatrix},$$

where $k = 0, \ldots L-1$, $L_{h|M}$ is the length of $h_{ik}^{(m)}$ and $L_{h|M} = \lceil L_h/M \rceil$, with $\lceil \cdot \rceil$ representing rounding to infinity.

6.4 Closed Forms of Optimal Equalizer Settings

By substituting Eqns (6.7) and (6.16) into Eqn (6.1), the system input/output relationship can be expressed in vector form as

$$\begin{aligned} \hat{x}_i(n) &= \hat{x}_{i,s}(n) + \hat{x}_{i,n}(n) \\ &= \sum_{m=i-N_t}^{i+N_t} \boldsymbol{A}_i' \boldsymbol{S}_{mi} \boldsymbol{x}_m(n) + \sum_{m=0}^{M-1} \boldsymbol{A}_i' \boldsymbol{H}_{mi} \boldsymbol{e}(Mn + l_i + 1 - m)\,. \end{aligned} \tag{6.17}$$

Next, by assuming that all system inputs $\{x_i(n)\}$ for various subchannels are mutually uncorrelated, the channel noise $e(n)$ is AWGN with a constant PSD σ_e^2, and each $x_i(n)$ is uncorrelated with $e(n)$ $\big($ thereby $e(Mn + l_i + 1 - m)$ $\big)$, one has

$$\boldsymbol{R}_{xx}^{(m_1 m_2)} \triangleq E[\boldsymbol{x}_{m_1}(n) \boldsymbol{x}_{m_2}(n)'] = \boldsymbol{0}\,, \quad \text{when } m_1 \neq m_2\,, \tag{6.18}$$

$$R_{xe} \triangleq E[\boldsymbol{x}_i(n) \boldsymbol{e}(Mn + l_i + 1 - m)'] = \boldsymbol{0}\,, \tag{6.19}$$

$$\begin{aligned} R_{ee}^{m_1 m_2} &\triangleq E[\boldsymbol{e}(Mn + l_i + 1 - m_1) \boldsymbol{e}(Mn + l_i + 1 - m_2)'] \\ &= \begin{cases} \sigma_e^2 \boldsymbol{I}, & m1 = m2\,, \\ \boldsymbol{0}, & m1 \neq m2\,. \end{cases} \end{aligned} \tag{6.20}$$

By using Eqns (6.17)–(6.20), the MSE between the system input and the corresponding system output becomes

$$\begin{aligned}
\mathrm{MSE}_i &\triangleq E\left[|r_i(n)|^2\right] = E[|\hat{x}_i(n) - x_i(n-k_i)|^2] \\
&= E\left[\left(\sum_{m=i-N_t}^{i+N_t} \boldsymbol{A}_i'\boldsymbol{S}_{mi}\boldsymbol{x}_m(n) + \sum_{m=0}^{M-1} \boldsymbol{A}_i'\boldsymbol{H}_{mi}\boldsymbol{e}(Mn+l_i+1-m) - \boldsymbol{u}_{k_i}'\boldsymbol{x}_i(n)\right) \cdot \right. \\
&\quad \left. \left(\sum_{m=i-N_t}^{i+N_t} \boldsymbol{A}_i'\boldsymbol{S}_{mi}\boldsymbol{x}_m(n) + \sum_{m=0}^{M-1} \boldsymbol{A}_i'\boldsymbol{H}_{mi}\boldsymbol{e}(Mn+l_i+1-m) - \boldsymbol{u}_{k_i}'\boldsymbol{x}_i(n)\right)'\right] \\
&= \boldsymbol{A}_i'\left(\sum_{m=i-N_i}^{i+N_i} \boldsymbol{S}_{mi}\boldsymbol{R}_{xx}^{(mm)}\boldsymbol{S}_{mi}' + \sigma_e^2 \sum_{m=0}^{M-1} \boldsymbol{H}_{mi}\boldsymbol{H}_{mi}'\right)\boldsymbol{A}_i \\
&\quad - 2\boldsymbol{A}_i'\boldsymbol{S}_{ii}\boldsymbol{R}_{xx}^{(ii)}\boldsymbol{u}_{k_i} + \boldsymbol{u}_{k_i}'\boldsymbol{R}_{xx}^{(ii)}\boldsymbol{u}_{k_i}\,,
\end{aligned} \tag{6.21}$$

where $\boldsymbol{u}_{k_i}$ representing the k_i-th unit vector, and k_i represents the relative delay between recovered and input symbols (c.f. Section 5.3.1).

Following the same procedure of section 5.3.1, by minimizing the MSE_i in Eqn (6.21), the optimal solution of $\boldsymbol{A}_i$ is

$$\boldsymbol{A}_{i,\mathrm{opt.}} = \left(\sum_{m=i-N_t}^{i+N_t} \boldsymbol{S}_{mi}\boldsymbol{R}_{xx}^{(mm)}\boldsymbol{S}_{mi}' + \sigma_e^2 \sum_{m=0}^{M-1} \boldsymbol{H}_{mi}\boldsymbol{H}_{mi}'\right)^{-1} \boldsymbol{S}_{ii}\boldsymbol{R}_{xx}^{(ii)}\boldsymbol{u}_{k_{i,\mathrm{opt.}}}\,, \tag{6.22}$$

and the MMSE of $r_i(n)$ becomes

$$\begin{aligned}
\mathrm{MMSE}_i &\triangleq \min\{\mathrm{MSE}_i\} \\
&= \boldsymbol{u}_{k_{i,\mathrm{opt.}}}'\boldsymbol{R}_{xx}^{(ii)}\boldsymbol{u}_{k_{i,\mathrm{opt.}}} - \boldsymbol{A}_i'\boldsymbol{S}_{ii}\boldsymbol{R}_{xx}^{(ii)}\boldsymbol{u}_{k_{i,\mathrm{opt.}}} \\
&= \boldsymbol{u}_{k_i}'\,\mathcal{E}_{i,\mathrm{multi.}}\,\boldsymbol{u}_{k_i}\,,
\end{aligned} \tag{6.23}$$

where

$$\mathcal{E}_{i,\mathrm{multi.}} = \boldsymbol{R}_{xx}^{(ii)} - \boldsymbol{R}_{xx}^{(ii)}\boldsymbol{S}_{ii}'\left(\sum_{m=i-N_t}^{i+N_t} \boldsymbol{S}_{mi}\boldsymbol{R}_{xx}^{(mm)}\boldsymbol{S}_{mi}' + \sigma_e^2 \sum_{m=0}^{M-1} \boldsymbol{H}_{mi}\boldsymbol{H}_{mi}'\right)^{-1} \boldsymbol{S}_{ii}\boldsymbol{R}_{xx}^{(ii)}\,,$$

with $k_{i,\mathrm{opt.}}$ representing the index of the minimum value of the main diagonal of $\mathcal{E}_{i,\mathrm{multi.}}$ $\big($ c.f. Eqn (5.42) $\big)$.

Finally, let us assume that all inputs within each subchannel are mutually uncorrelated and have a constant PSD σ_x^2 over full channel bandwidth. Then, $\boldsymbol{R}_{xx}^{(ii)}$ can be simplified as in Eqn (5.44). In this case, the optimal equalizer coefficient matrix $\boldsymbol{A}_{i,\mathrm{opt.}}$ is simplified to

$$\boldsymbol{A}_{i,\mathrm{opt.}} = \left(\sum_{m=i-N_t}^{i+N_t} \boldsymbol{S}_{mi}\boldsymbol{S}_{mi}' + \frac{\sigma_e^2}{\sigma_x^2} \sum_{m=0}^{M-1} \boldsymbol{H}_{mi}\boldsymbol{H}_{mi}'\right)^{-1} \boldsymbol{S}_{ii}\boldsymbol{u}_{k_{i,\mathrm{opt.}}}\,, \tag{6.24}$$

while the MMSE of $r_i(n)$ is still calculated by using Eqn (6.23) with $\mathcal{E}_{i,\text{multi.}}$ simplified to

$$\mathcal{E}_{i,\text{multi.}} = \sigma_x^2 \left(\boldsymbol{I} - \boldsymbol{S}'_{ii} \left(\sum_{m=i-N_t}^{i+N_t} \boldsymbol{S}_{mi}\boldsymbol{S}'_{mi} + \frac{\sigma_e^2}{\sigma_x^2} \sum_{m=0}^{M-1} \boldsymbol{H}_{mi}\boldsymbol{H}'_{mi} \right)^{-1} \boldsymbol{S}_{ii} \right) .$$

The multi-tap interpolation equalization technique owns high flexibility in computation because tradeoff can be made between L_a (the length of equalizers), L (the number of polyphase components of the analysis filter output used in equalization), and N_t ($2N_t$ is the number of ICI terms considered in equalization). Moreover, the multi-tap interpolation equalization technique is the generalization of the integer-delay equalization and one-tap interpolation equalization techniques. When $L = L_a = N_t = 1$, the multi-tap interpolation equalization technique becomes the integer-delay equalization. When $L = 2$ and $L_a = N_t = 1$, it becomes the one-tap interpolation equalization.

6.5 Simulation Results

In this section, the filterbank-based DMT system uses 64-subchannel critically decimated CMFB as its modulator/demodulator. The design parameters of CMFB are the same as those in Table 4.1. Moreover, it is assumed that system inputs $\{x_i(n)\}$ for various subchannels are mutually uncorrelated, and all input symbols within each subchannel are also mutually uncorrelated. Furthermore, it is assumed that the channel noise is AWGN and all system inputs $\{x_i(n)\}$ are mutually uncorrelated with the channel noise.

6.5.1 Noiseless Channel Cases

First, the simulation results for noiseless channel cases are compared. Figure 6.4–6.7 illustrate the SIRs for CSA loop #1, #2, #3, and #6 for various L_a (the length of equalizers) and L (the number of polyphase components of the analysis filter output used in equalization). The SIRs for other CSA loops are shown in Figure 6.8. The four curves in each figure, from top to bottom, represent the SIR for $L = 3$ and $L_a = 7$, $L = 2$ and $L_a = 7$, $L = 2$ and $L_a = 3$, as well as $L = 2$ and $L_a = 1$, respectively. As a comparison, the SIR for the case using ideal channel is also drawn as the straight line (42 dB) in these figures

From these figures, it is clear that with increasing L and/or L_a, the SIR increases. Note that because the filterbank used in this simulation is the pseudo PR filterbank, interference exists in the system output even when an ideal channel is used. However, by using the multi-tap interpolation equalization technique with moderate L and L_a, the interference

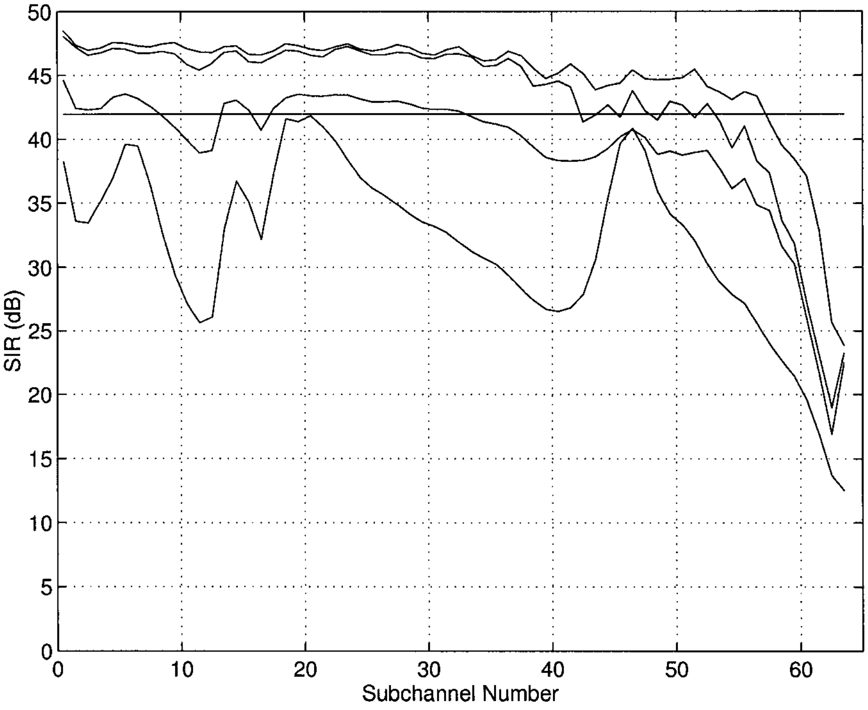


Figure 6.4: SIRs for multi-tap interpolation equalization for CSA loop #1 for various L and L_a

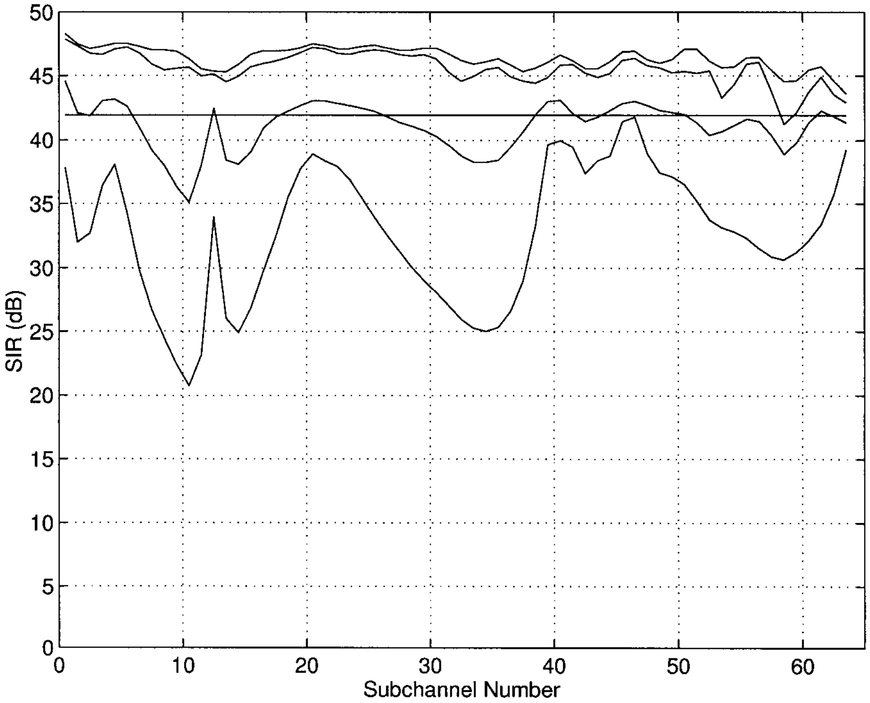


Figure 6.5: SIRs for multi-tap interpolation equalization for CSA loop #2 for various L and L_a

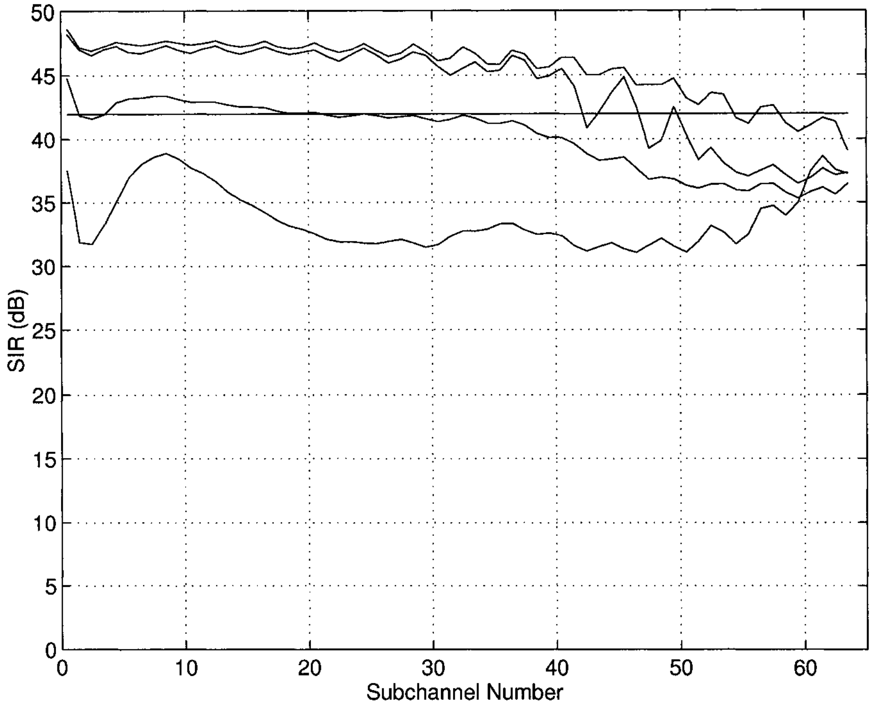


Figure 6.6: SIRs for multi-tap interpolation equalization for CSA loop #3 for various L and L_a

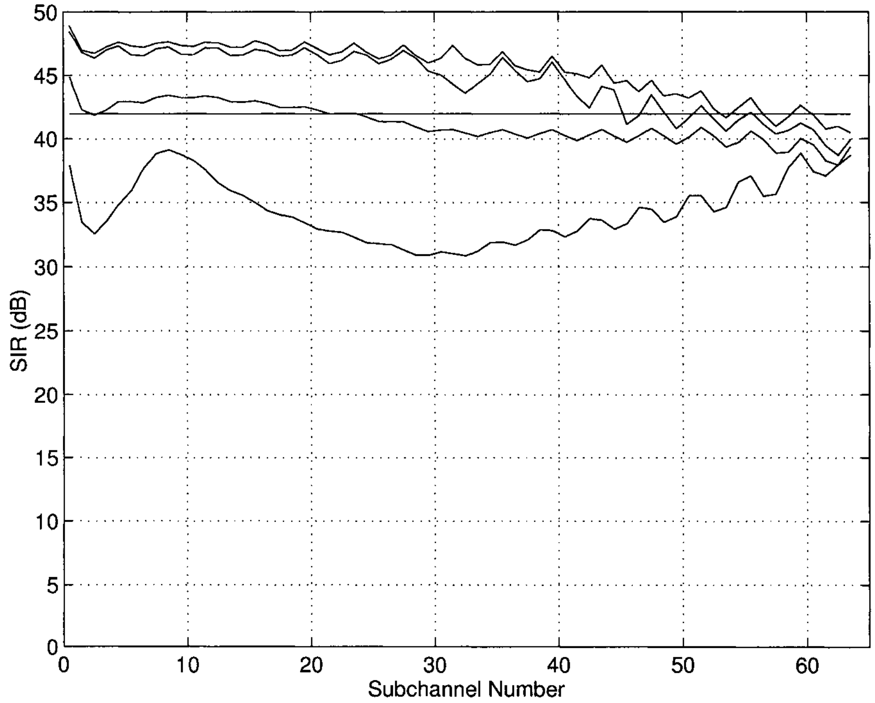


Figure 6.7: SIRs for multi-tap interpolation equalization for CSA loop #6 for various L and L_a

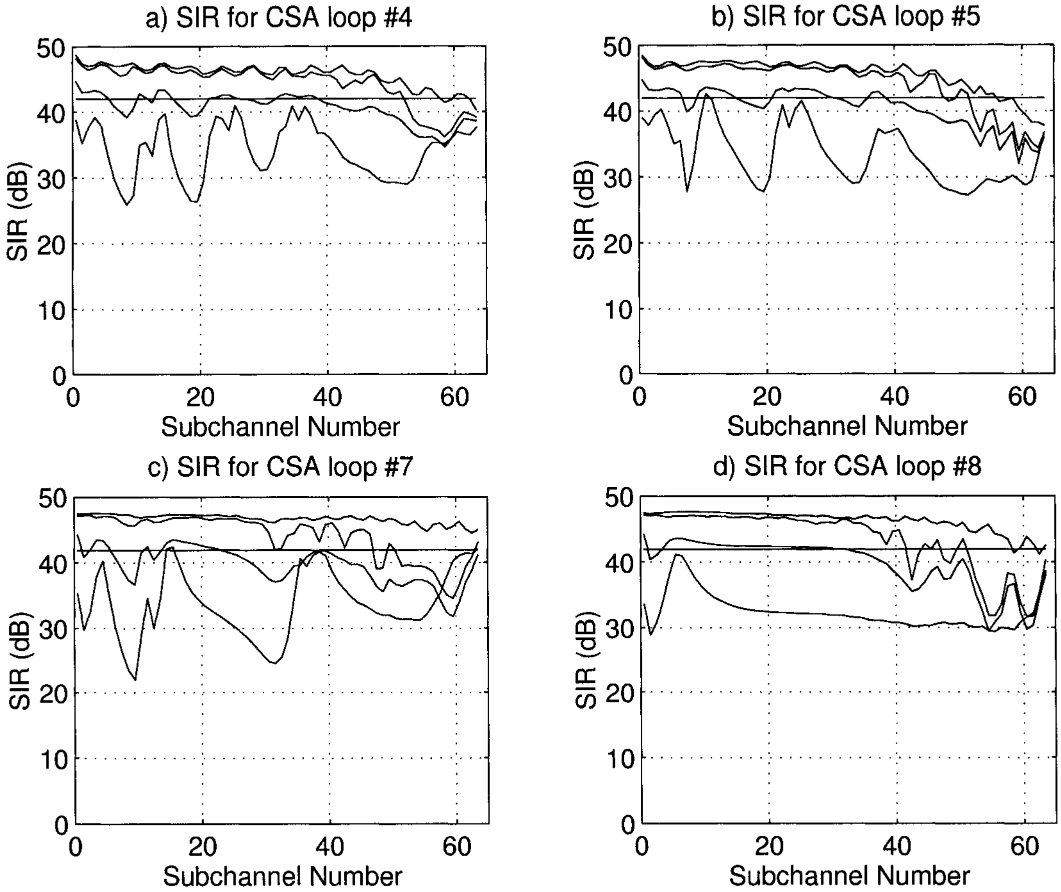


Figure 6.8: SIRs for multi-tap interpolation equalization for CSA loops #4, #5, #7, and #8 for various L and L_a

caused by the filterbank can also be compensated for, and the output SIR can exceed the SIR (42 dB) for the case using the pseudo PR filterbank and an ideal channel.

6.5.2 Noisy Channel Cases

In this subsection, the SNRs for noisy channel cases are compared. The length of the prototype filter of the CMFB is $12M = 768$. Other simulation parameters are as listed in Table 6.1.

Subchannel number M		64
Sampling frequency f_s		2.208M Hz
System inputs (same for all subchannels)	Mean	0
	PSD (dBm/Hz)	-40
Channel noise	Mean	0
	PSD (dBm/Hz)	-140

Table 6.1: Simulation parameters for multi-tap interpolation equalization

The output SNR of the i-th subchannel SNR_i is defined as

$$\mathrm{SNR}_i \triangleq \frac{E\left\{|x_i(n)|^2\right\}}{E\left\{|x_i(n) - \hat{x}_i(n)|^2\right\}} . \tag{6.25}$$

Note that because the MMSE optimization is used in this chapter, the receiver is a biased receiver [13] (c.f. Appendix B). Then, SNR_i defined in Eqn (6.25) is the biased SNR. However, by comparing to the definition of the unbiased SNR (c.f. Eqn (1.8)), the MFB defined in Eqn (1.12) corresponds to the unbiased SNR. Therefore, in order to compare the output SNR to the MFB, the MFB used in this section is modified to biased SNR as

$$\mathrm{SNR}_{\mathrm{MFB},i} \triangleq \frac{\varepsilon_{x,i} C_i^2}{\varepsilon_e} + 1 . \tag{6.26}$$

Figure 6.9–6.12 illustrate the SNRs for CSA loops #1, #2, #3, and #6 for various L_a and L. The SNRs for other CSA loops are shown in Figure 6.13. The four solid curves in each figure, from top to bottom, represent the SNR for $L = 2$ and $L_a = 16$, $L = 3$ and $L_a = 7$, $L = 2$ and $L_a = 3$, as well as $L = 2$ and $L_a = 1$, respectively. As a comparison, in each figure, the MFB for the corresponding channel case is also drawn as the dash-dot curve (or dotted curves in Figure 6.13).

From these figures, one can see that with increasing L and/or L_a, the SNR increases. In each channel case, when $L = 2$ and $L_a = 16$, the SNR is close to the MFB of the corresponding channel. Note that in each channel case, the SNRs of the high-frequency subchannels are close to the channel MFB even when L and L_a are relatively small (for example, $L = 2$ and $L_a = 3$). The reason of this can be interpreted as follows.

When L and L_a are relatively small, the interference in the recovered symbols is relatively large. However, compared to the low-frequency subchannels, because of the channel high attenuation in high-frequency subchannels, the desired signal components (as well as the interference) in the output symbols of high-frequency subchannels are significantly small and the channel noise components in the output symbols of these subchannels are dominant. Therefore, in the high-frequency subchannels, the interferences can be neglected, and the output SNR is close to the MFB. By knowing this, it is sufficient to measure the performance of the equalization by using the output SIR (i.e. assuming noiseless channel).

6.5.3 SIR Comparisons between Multi-Tap Interpolation Equalization, Output Combiner, and Per Tone Equalization

In this subsection, the multi-tap interpolation equalization technique will be compared with the output combiner and per tone equalization techniques in accordance with the output

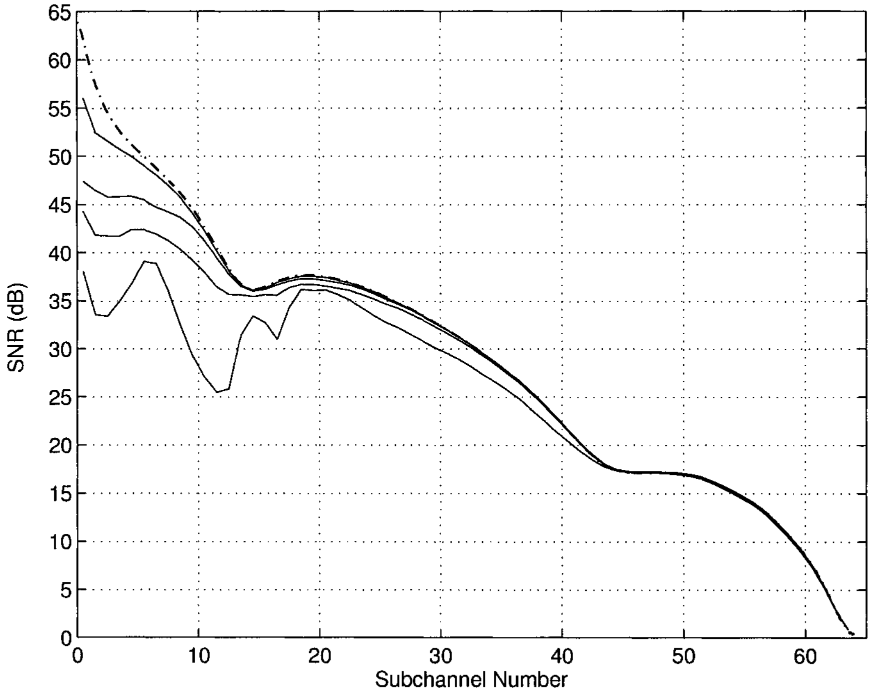


Figure 6.9: SNRs for multi-tap interpolation equalization for CSA loop #1 for various L and L_a

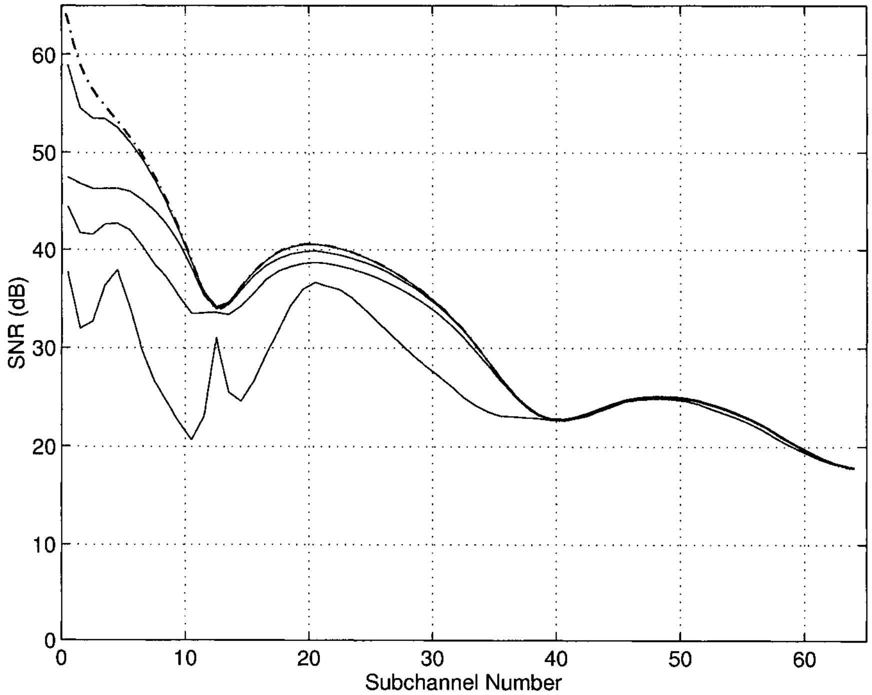


Figure 6.10: SNRs for multi-tap interpolation equalization for CSA loop #2 for various L and L_a

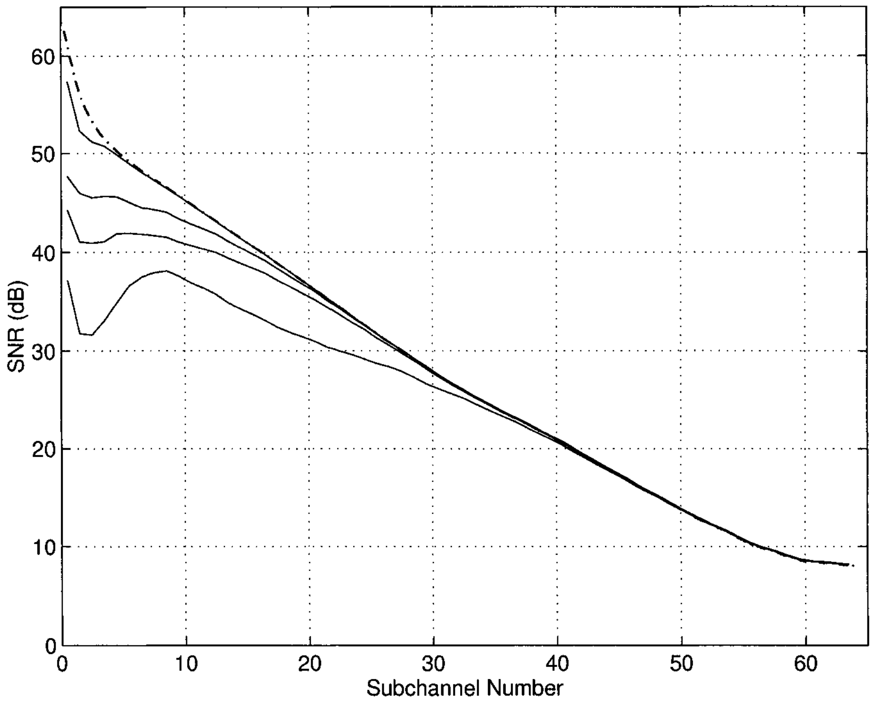


Figure 6.11: SNRs for multi-tap interpolation equalization for CSA loop #3 for various L and L_a

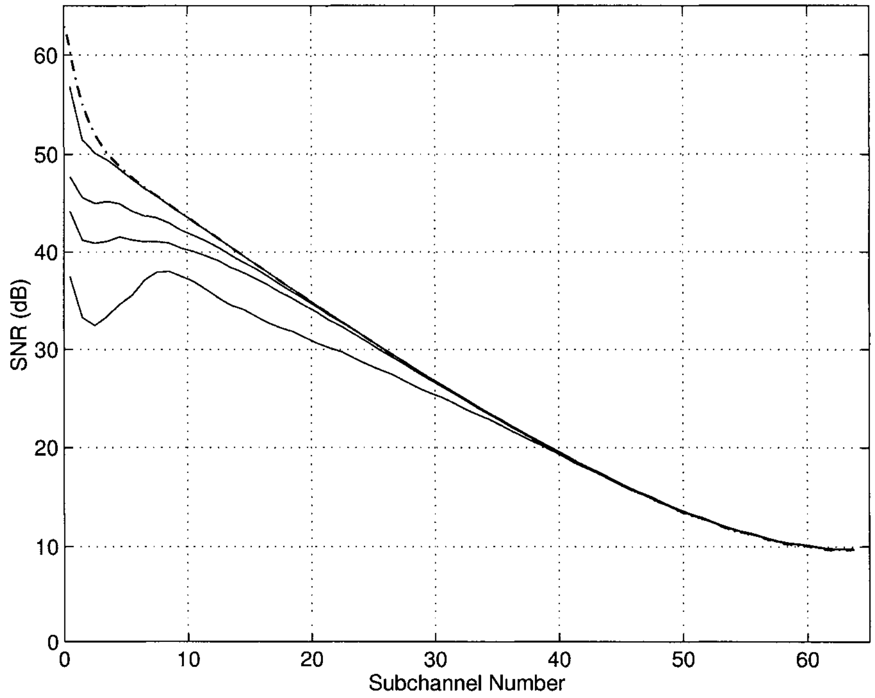


Figure 6.12: SNRs for multi-tap interpolation equalization for CSA loop #6 for various L and L_a

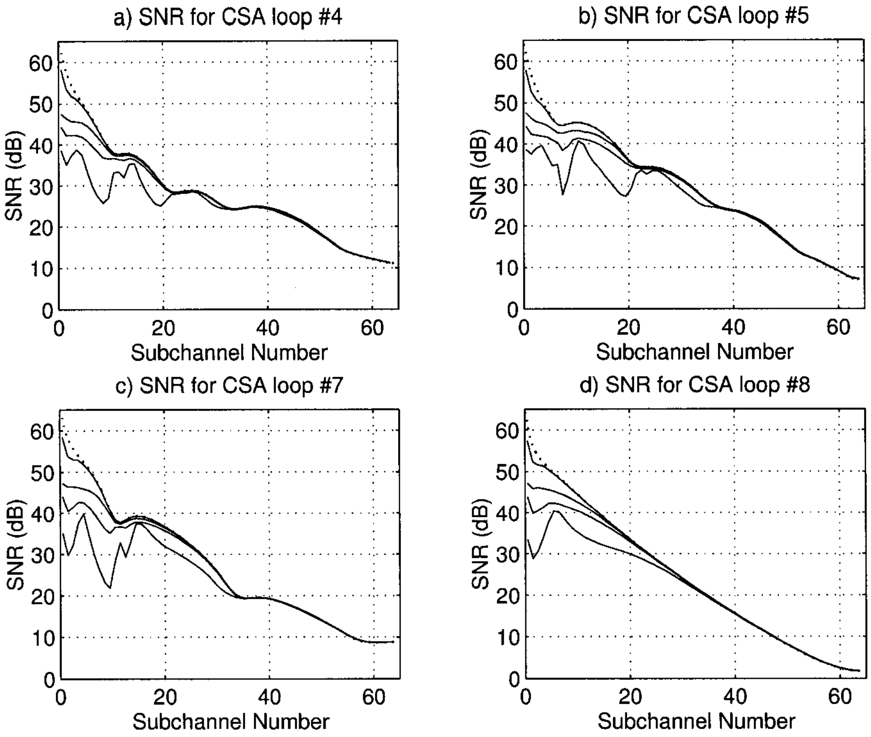


Figure 6.13: SNRs for multi-tap interpolation equalization for CSA loops #4, #5, #7, and #8 for various L and L_a

SIR. In the following comparisons, two channels are used, which are CSA loop #1 and #3. The group-delay of CSA loop #1 has violent fluctuations while the group-delay of CSA loop #3 is relatively smooth. Therefore, these two channels can be used as the worst and best channel cases.

First, the SIR comparison for the multi-tap interpolation equalization and the output combiner is shown in Figures 6.14 (for CSA loop #1) and 6.15 (for CSA loop #6). The DMT system for both equalization techniques uses 64-subchannel CMFB as the modulator/demodulator, with the 384-tap prototype filter. The multi-tap interpolation equalization combines 3 polyphase components of the analysis filter output for equalization, and the output combiner also combines 3 receiver outputs for equalization. In Figures 6.14 and 6.15, the two solid curves, from top to bottom, are SIRs for the multi-tap interpolation equalization with equalizer length of 15 and 7, respectively. The two dotted curves, from top to bottom, are SIRs for the output combiner with equalizer length of 15 and 7, respectively. Therefore, the total equalizer length for the multi-tap interpolation equalization and the output combiner are the same. From Figures 6.14 and 6.15, it has been shown that the multi-tap interpolation equalization leads to higher SIR than the output combiner.

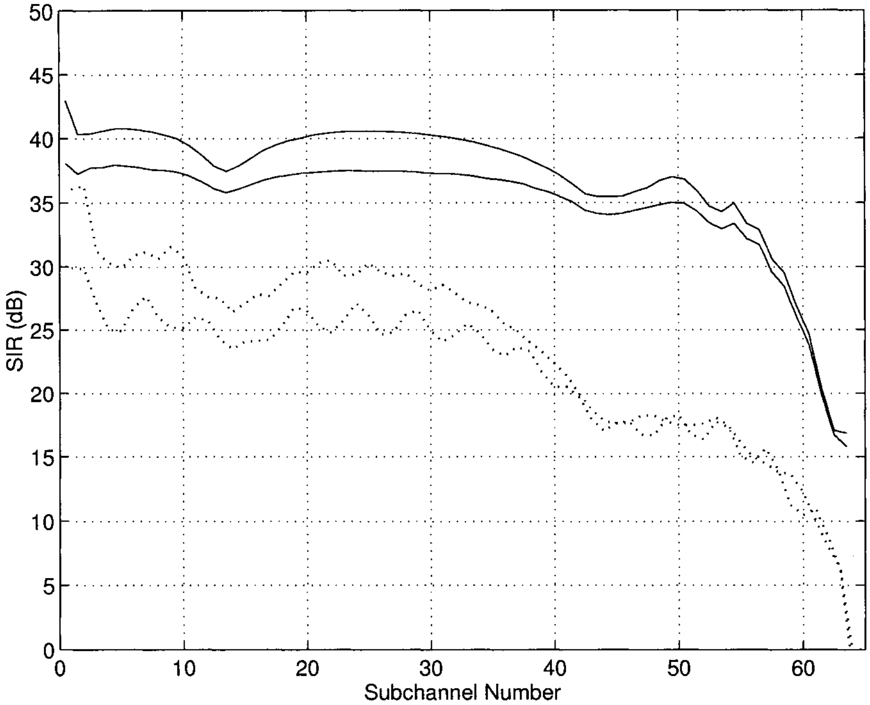


Figure 6.14: SIR comparison for multi-tap interpolation equalization and output combiner for CSA loop #1

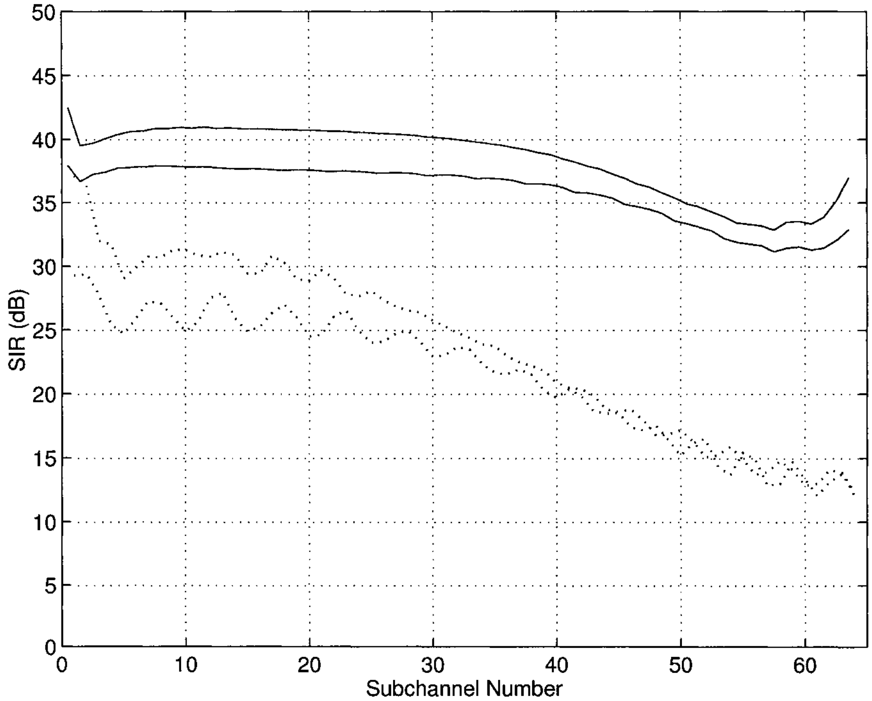


Figure 6.15: SIR comparison for multi-tap interpolation equalization and output combiner for CSA loop #6

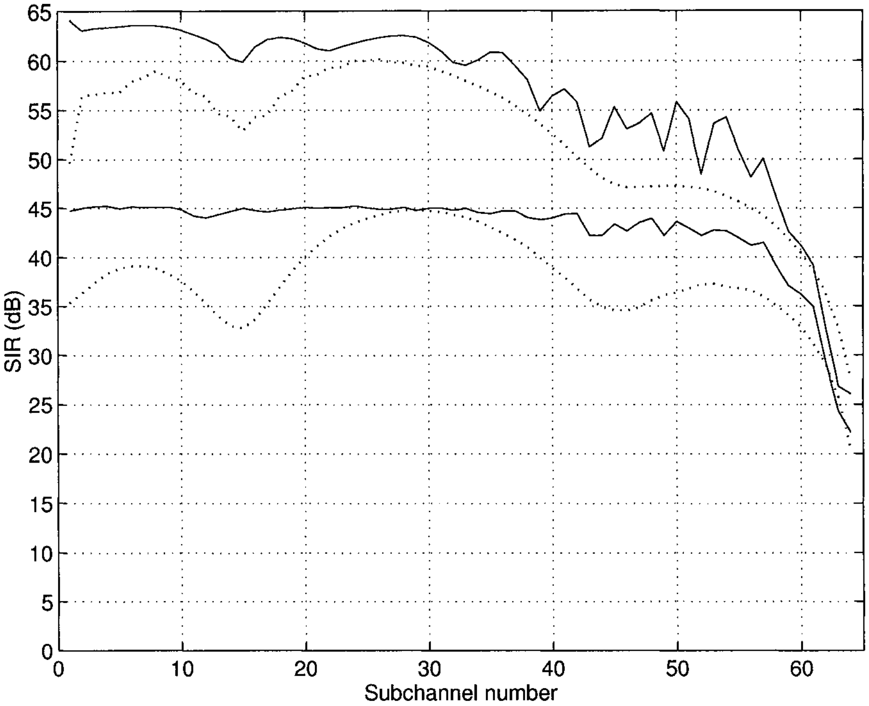


Figure 6.16: SIR comparison for multi-tap interpolation equalization and per tone equalization for CSA loop #1

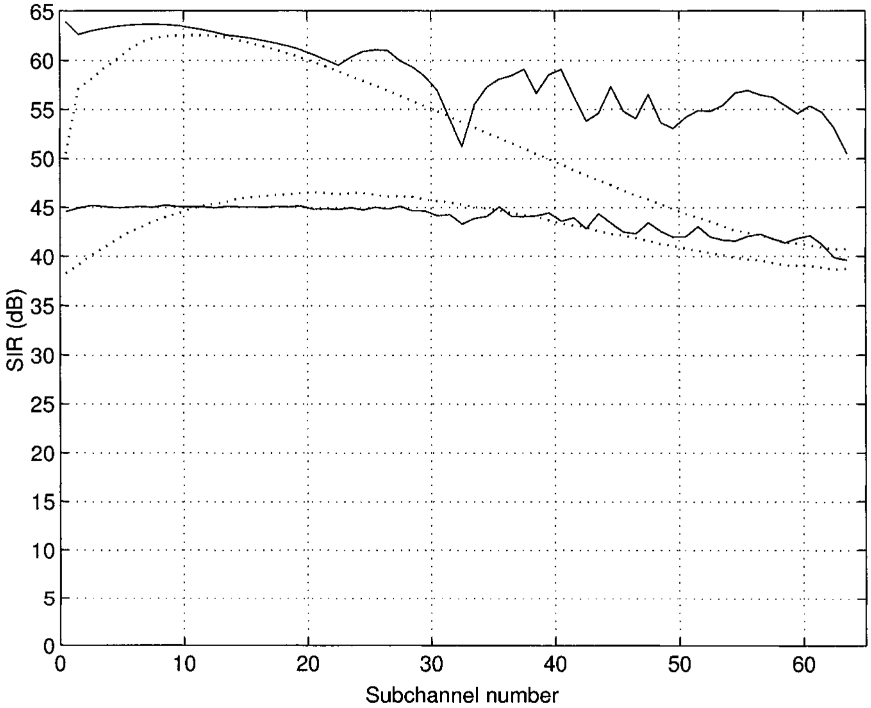


Figure 6.17: SIR comparison for multi-tap interpolation equalization and per tone equalization for CSA loop #6

Moreover, note that the 3 polyphase components used for the multi-tap interpolation equalization are all from the same analysis filter output. Therefore, in the equalizer training stage, only one overall transfer function ($s_{ii}(n)$) has to be used for multi-tap interpolation equalization. On the other hand, because the output combiner uses 3 receiver outputs, in the equalizer training stage, 3 overall transfer functions ($s_{i(i-1)}(n)$, $s_{ii}(n)$, and $s_{i(i+1)}(n)$) have to be used. Also note that these 3 overall transfer functions may be used for the equalizer training of other subchannels (for example, $s_{ii}(n)$, and $s_{i(i+1)}(n)$ are used for the equalizer training of the $(i+1)$-th subchannel), each overall transfer function is needed to be calculated once. Therefore, the computational complexities in equalizer training stage for multi-tap interpolation equalization and the output combiner are the same. However, the storage space used for equalizer training of the output combiner is 3 times as that of the multi-tap interpolation equalization.

Next, the SIR comparison for the multi-tap interpolation equalization and the per tone equalization is shown in Figures 6.16 (for CSA loop #1) and 6.17 (for CSA loop #6). The DMT system for the multi-tap interpolation equalization uses 128-subchannel CMFB as the modulator/demodulator, with the 384-tap prototype filter. The multi-tap interpolation equalization combines 2 polyphase components of the analysis filter output for equalization with various equalizer lengths. On the other hand, the per tone equalization uses 512-point IDFT/DFT as the modulator/demodulator, with the 4-sample cyclic prefix (this leads to a transmission throughput loss of 0.8%). In Figures 6.16 and 6.17, the two solid curves, from top to bottom, are SIRs for the multi-tap interpolation equalization with equalizer lengths of 16 and 4, respectively. The two dotted curves, from top to bottom, are SIRs for the per tone equalization with equalizer lengths of 32 and 8, respectively. Therefore, the total equalizer length for the multi-tap interpolation equalization and the per tone equalization are the same. From Figures 6.16 and 6.17, it has been shown that the multi-tap interpolation equalization leads to comparable SIR to the per tone equalization.

Because the DMT system for the multi-tap interpolation equalization employs longer filters than the DMT system for the per tone equalization, the computational complexity of the former is higher than that of the latter. However, the former system has higher spectral containment and maximum transmission throughput. Moreover, the per tone equalization is based on the assumption that the length of the IDFT/DFT is longer than the length of the channel impulse response while the multi-tap interpolation equalization does not need this assumption. Therefore, the multi-tap interpolation equalization can be used for more channel cases than the per tone equalization.

6.6 A Sub-Optimal Interpolation Equalization Approach

6.6.1 A Sub-Optimal Solution for Interpolation Equalization

The multi-tap interpolation equalization technique combines several polyphase components of the analysis filter output to recover the input symbols. The optimal polyphase component set has to be searched in order to obtain a high output SNR. Therefore, one can say that among the M polyphase components, some (i.e. the optimal polyphase component set) are suitable for channel equalization (let us call them "good" polyphase components) while others are not suitable for channel equalization (let us call them "bad" polyphase components). Once the "good" polyphase components are included in the polyphase component set used for multi-tap interpolation equalization, the output SNR will be high (although may not be the highest) no matter how many and which "bad" polyphase components are also included. Then, by combining more polyphase components, the index set of the optimal polyphase components will have a larger range, and the output SNR will be less sensitive to the choice of the used polyphase component set. For example, suppose the index set of the "good" polyphase components is $\{10,\ 11\}$, then, by combining 3 polyphase components, only 2 polyphase component index sets ($\{9,\ 10,\ 11\}$ and $\{10,\ 11,\ 12\}$) can lead to high SNR. However, if combining 6 polyphase components, 5 polyphase component index sets ($\{6, \ldots, 11\}$–$\{10, \ldots, 15\}$) will lead to high SNR. Moreover, from Figure 5.12, when 2 polyphase components are used in equalization, the difference of the optimal polyphase component index set between most subchannels is relatively small. Therefore, by combing a large number of polyphase components in equalization, one can expect that the index sets for the optimal polyphase components for all subchannels are same (or similar).

Based on the above analysis, a suboptimal (but simplified) interpolation equalization approach can be proposed, where a large L (the number of used polyphase components) is used and all subchannels use the same polyphase component index set, i.e. $l_i = l$ for all $0 \le i \le M-1$. In this case, the recovered signal $\hat{x}_i(n)$ (c.f. Figure 6.1 with $l_i = l$) becomes

$$\hat{x}_{i,s}(n) = \sum_{k=0}^{L-1} a_{i,k}(n) * \tilde{y}_i^{(l+k)}(n)\,. \tag{6.27}$$

With this suboptimal solution, the optimal polyphase index set does not need to be searched subchannel-by-subchannel. Therefore, the computational complexity in equalizer training stage can be significantly simplified.

6.6.2 Simulation Results

In this subsection, the DMT system uses 64-subchannel CMFB with the prototype filter of length $12M = 768$. Other simulation parameters are the same as in Table 6.1. In the following simulation, the transmission bit-rates for various equalizer settings and two channel cases (CSA loops #1 and #6) are compared. For each channel case, the sub-optimal interpolation equalization combines various number L of polyphase components of the analysis filter output, with each polyphase component passing through a 4-tap equalizer (i.e. $L_a = 4$). The transmission bit-rate is calculated as

$$b = \frac{f_s}{M} \sum_{m=0}^{M-1} \log_2(1 + 10^{\mathrm{SNR}_{\mathrm{marg.},i}}), \tag{6.28}$$

where $\mathrm{SNR}_{\mathrm{marg.},i}$ is conventionally defined in dB as

$$\mathrm{SNR}_{\mathrm{marg.},i}(\mathrm{dB}) \triangleq \mathrm{SNR}_i(\mathrm{dB}) + \gamma_c(\mathrm{dB}) - \gamma_n(\mathrm{dB}) - \mathit{\Gamma}_g(\mathrm{dB}), \tag{6.29}$$

where SNR_i is defined in Eqn (6.25) representing the output SNR of the i-th subchannel, and the coding gain $\gamma_c(\mathrm{dB}) = 3\mathrm{dB}$, noise margin $\gamma_n(\mathrm{dB}) = 6\mathrm{dB}$, and the SNR gap $\mathit{\Gamma}_g(\mathrm{dB}) = 9.8\mathrm{dB}$.

Figures 6.18 and 6.19 illustrate the transmission bit-rates in function of the optimal polyphase index set l for CSA loops #1 and #6, respectively. In each figure, from top to bottom, the three curves represent the transmission bit-rates for the cases when $L = 16$, 8, and 3, respectively. Also as a comparison, the corresponding transmission bit-rates for the multi-tap interpolation equalization technique (i.e. using different optimal polyphase component index set for each subchannel) are listed in Table 6.2.

Equalizer settings	CSA loop #1	CSA loop #6
$L = 16$, $L_a = 4$	11.84 Mb/s	10.38 Mb/s
$L = 8$, $L_a = 4$	11.26 Mb/s	10.17 Mb/s
$L = 3$, $L_a = 4$	11.02 Mb/s	9.94 Mb/s

Table 6.2: Transmission bit-rates for multi-tap interpolation equalization

From Figures 6.18 and 6.19, and Table 6.2, it is clear that the transmission bit-rate for the sub-optimal interpolation equalization is close to the bit-rate for the multi-tap interpolation equalization (i.e. the optimal solution) when combing a large number L of polyphase components. Also, with increasing L, the transmission bit-rate is less sensitive to selection of the optimal polyphase component index set l. This observation leads to a further simplification to the sub-optimal interpolation equalization, where the optimal l can be arbitrarily chosen from a pre-selected index range, instead of searching for the optimal l.

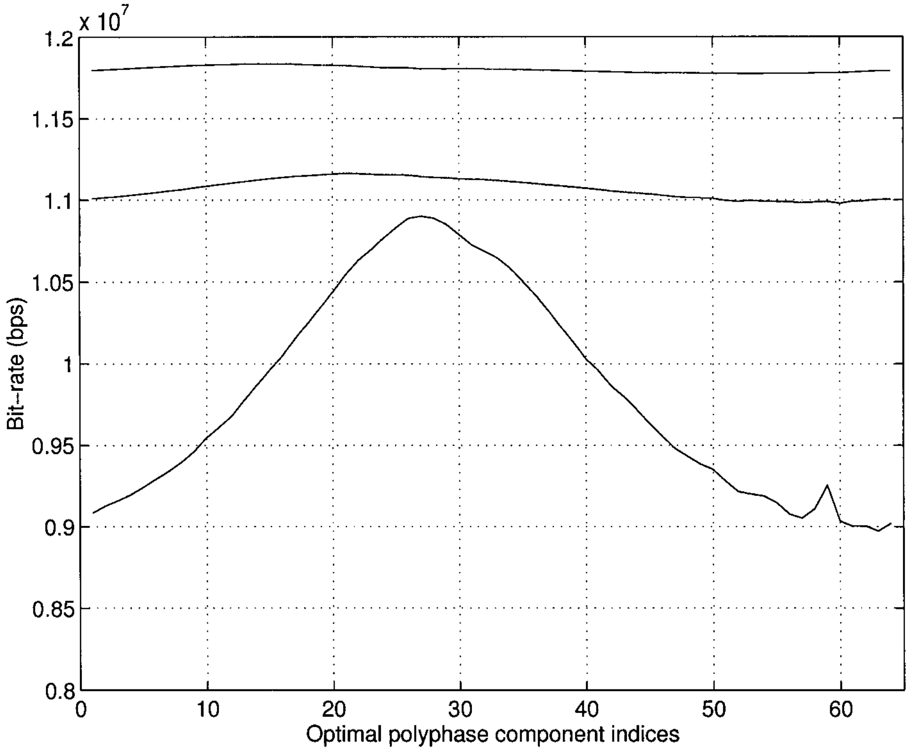


Figure 6.18: Bit-rate in function of optimal polyphase index set l for CSA loop #1

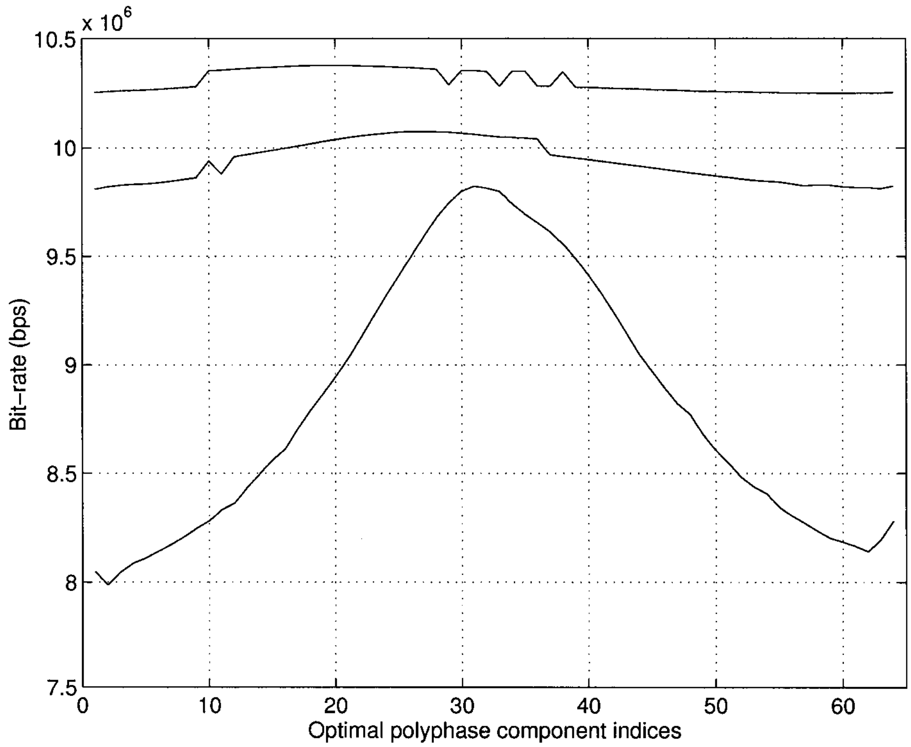


Figure 6.19: Bit-rate in function of optimal polyphase index set l for CSA loop #6

It can also be shown (c.f. Appendix C) that the per-tone equalization technique used in the conventional DMT system is a special case of this sub-optimal interpolation equalization approach, where the DFT filterbanks are used as the synthesis/analysis filterbanks, and the equalization of each subchannel combines multiple (equal to the length of a per tone equalizer) polyphase components and each polyphase component is weighted by a one-tap interpolation equalizer.

Chapter 7

Conclusion and future work

7.1 Conclusion

This thesis has been concerned with the design of Integer-Delay Equalization, One-Tap and Multi-Tap Interpolation Equalization Techniques for Filterbank-Based Discrete Multitone (DMT) Systems used in various Digital Subscriber Line (xDSL) applications. The main contributions of this thesis are as follows:

1. The dominant effects of channel delay has been investigated. It has been shown that when the time synchronization of decimators mismatch that of the filterbank outputs because of the channel phase distortion, the samples discarded by decimators may contain information highly useful or relevant to the characteristic of input symbols. It has also been shown that the channel delay exhibits the periodic-like property. With this property, after subtraction of suitable integer multiples of M, the channel delay can always be represented by its principle-value delay, which can be related to one or some of the M polyphase components of the analysis filter output.

2. An integer-delay equalization technique has been proposed for the case of channels characterized by integer delays. Instead of using the 0-th polyphase component of the analysis filter output, this equalization technique selects an optimal polyphase component as the recovered symbols. This equalization technique can be used for binary and low bit-rate communication cases with no additional cost.

3. The effect of channel non-integer delay has been analyzed. When channel delay is a non-integer, no polyphase component of the analysis filter output can be used as the recovered symbols. However, it has been proved that the interpolation approach can be used to recover the system input by combining two carefully selected polyphase components of the analysis filter output.

4. The one-tap interpolation equalization technique has been proposed. This technique combines two carefully selected polyphase components of the analysis filter output, and weights each polyphase component by a one-tap equalizer to recover the system input. It has been shown that this technique leads to moderate SNR with small equalization cost (only two equalizer taps for each subchannel).

5. A suboptimal (but simplified) solution for one-tap equalization technique has been proposed. This suboptimal solution restricts the index of the candidates of the optimal polyphase components within the upper and lower bounds of the corresponding subchannel group-delay. With this solution, the search of the optimal polyphase components can be significantly simplified.

6. As a generalization of the integer-delay equalization and the one-tap interpolation equalization, the multi-tap interpolation equalization has been proposed. This equalization technique employs multiple polyphase components, and weights each polyphase component by a multi-tap equalizer to recover the system input. Generally, this technique has higher computational complexity compared to the other two equalization techniques proposed in this thesis. However, it leads to high system output SNRs (for example, about 15 dB higher than the output combiner technique) and permits high flexibility in computation because a tradeoff can be made between the length of equalizers, the number of polyphase components used in equalization, and the number of ICI terms considered in equalization. It has comparable computational complexity and higher SNR compared to other equalization techniques such as the output combiner and the per tone equalization.

7. Another suboptimal (but simplified) solution for the multi-tap interpolation equalization technique has been proposed. By combining a large number of polyphase components, the equalizers of all subchannels can use the same optimal polyphase component index set, instead of searching it subchannel-by-subchannel. It has also been shown that the per tone equalization is a special case of this suboptimal solution.

7.2 Future Work

The one-tap interpolation equalization techniques use linear interpolation to recover the system input. The multi-tap interpolation equalization is also based on linear interpolation. Other interpolation approaches such as high-order Lagrange interpolation and Newton interpolation may lead to higher SNR. These interpolation approaches have not been studied yet.

In this thesis, the effect of channel phase response has been studied. This study is based on the assumption that the channel impulse response is known, which is true in wireline communication cases. On the other hand, when the channel is not known, an adaptive equalization technique is needed. This technique can be derived by studying the statistics of different polyphase components of the analysis filter output. The work of this topic can extend the use of multi-tap interpolation equalization into wireless communication area (where the channel is time-variant fading channel, and is usually supposed to be unknown).

The equalization techniques proposed in this thesis are used for DMT systems employing critically decimated filterbanks. A future work can be done by extending these equalization techniques for DMT systems employing non-critically decimated filterbanks, for example, the FMT system, to achieve high output SNR with relatively low equalization cost.

Appendix A

Impulse and Frequency Responses of CSA Loop #1–#8

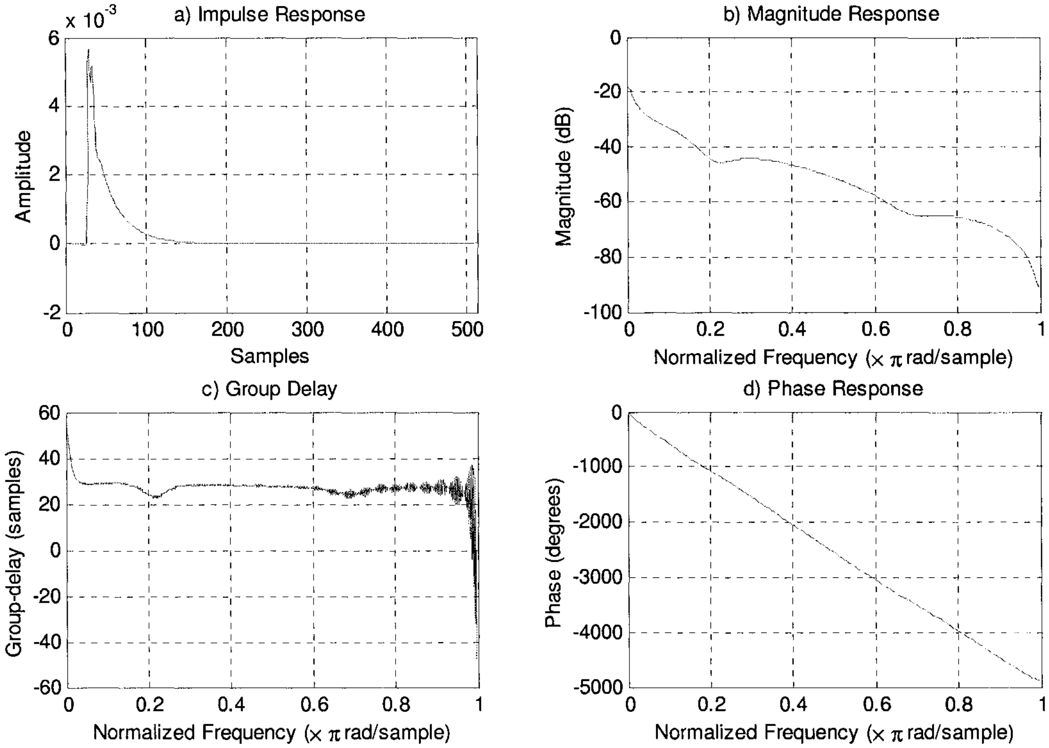


Figure A.1: CSA loop #1

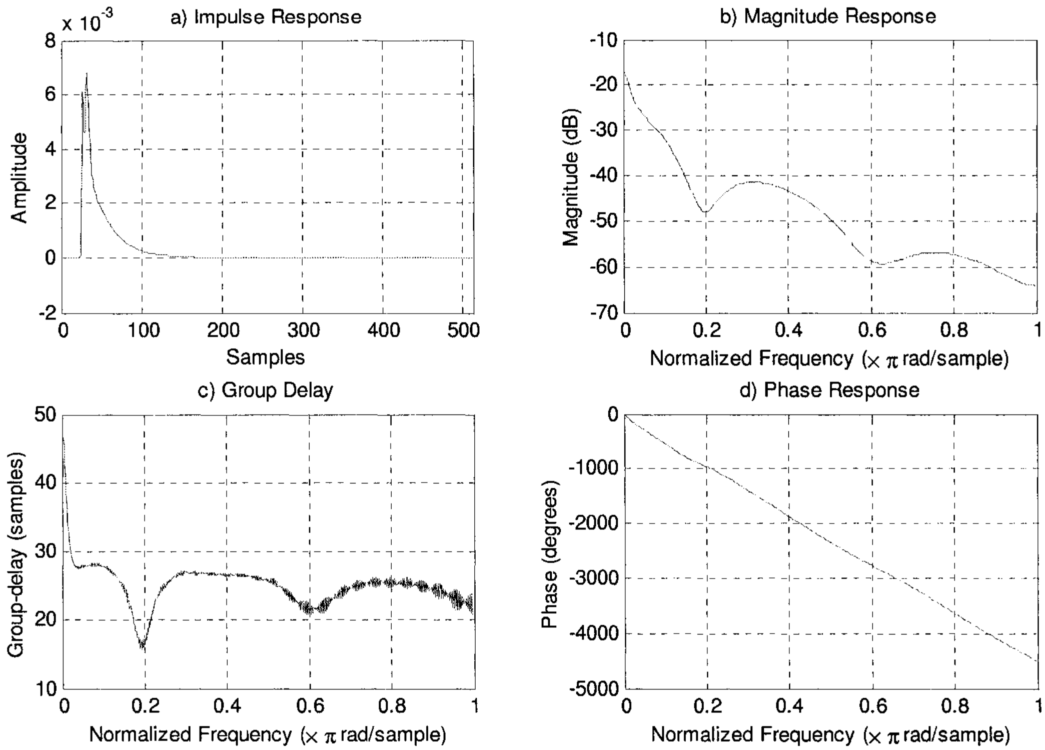


Figure A.2: CSA loop #2

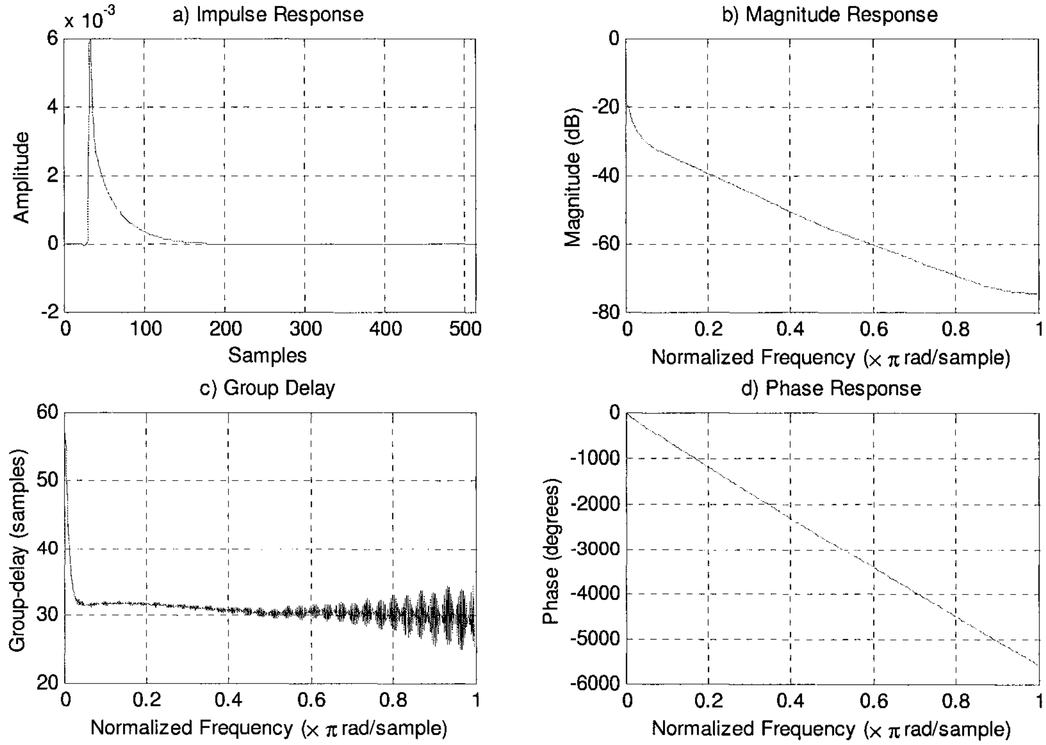


Figure A.3: CSA loop #3

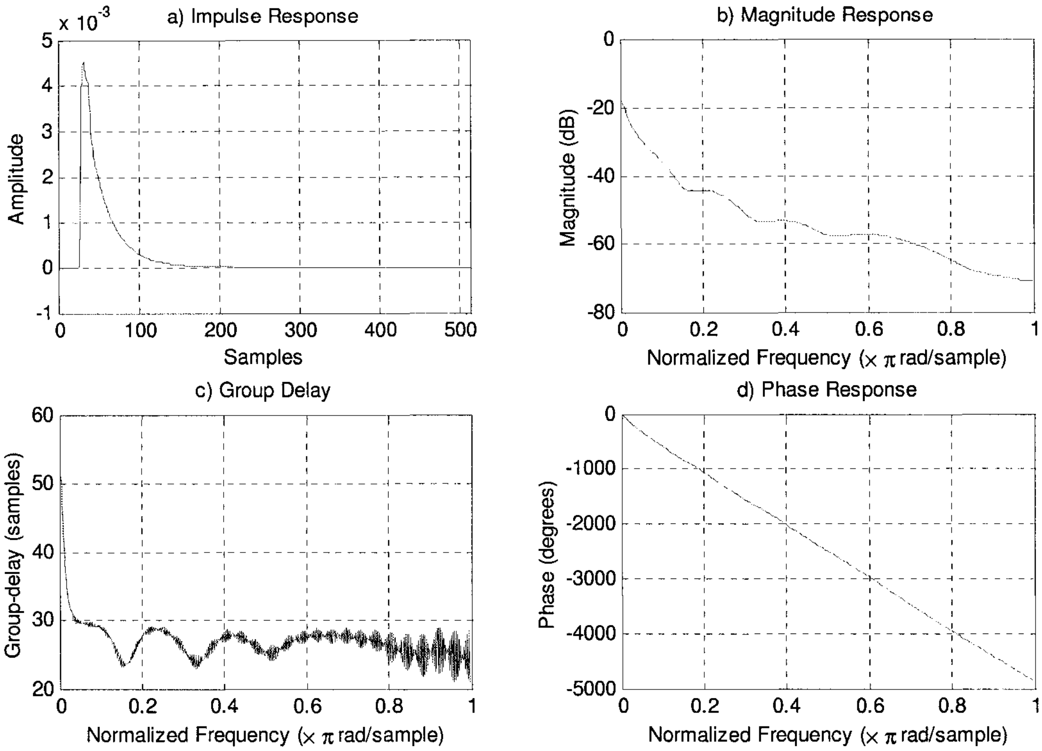


Figure A.4: CSA loop #4

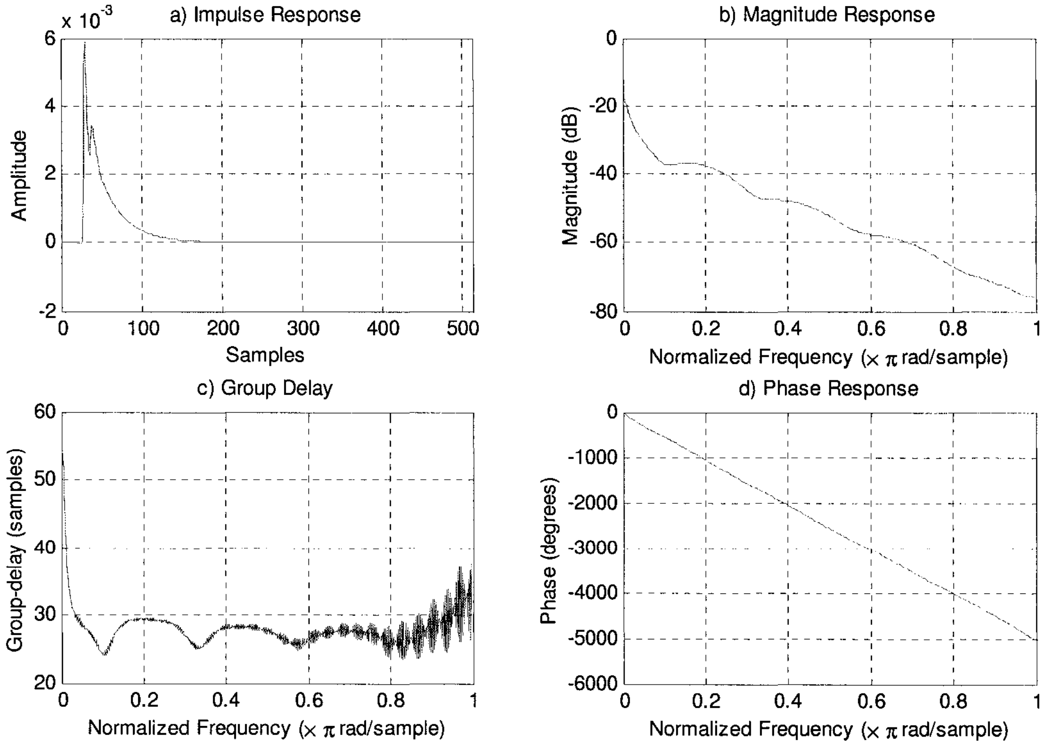


Figure A.5: CSA loop #5

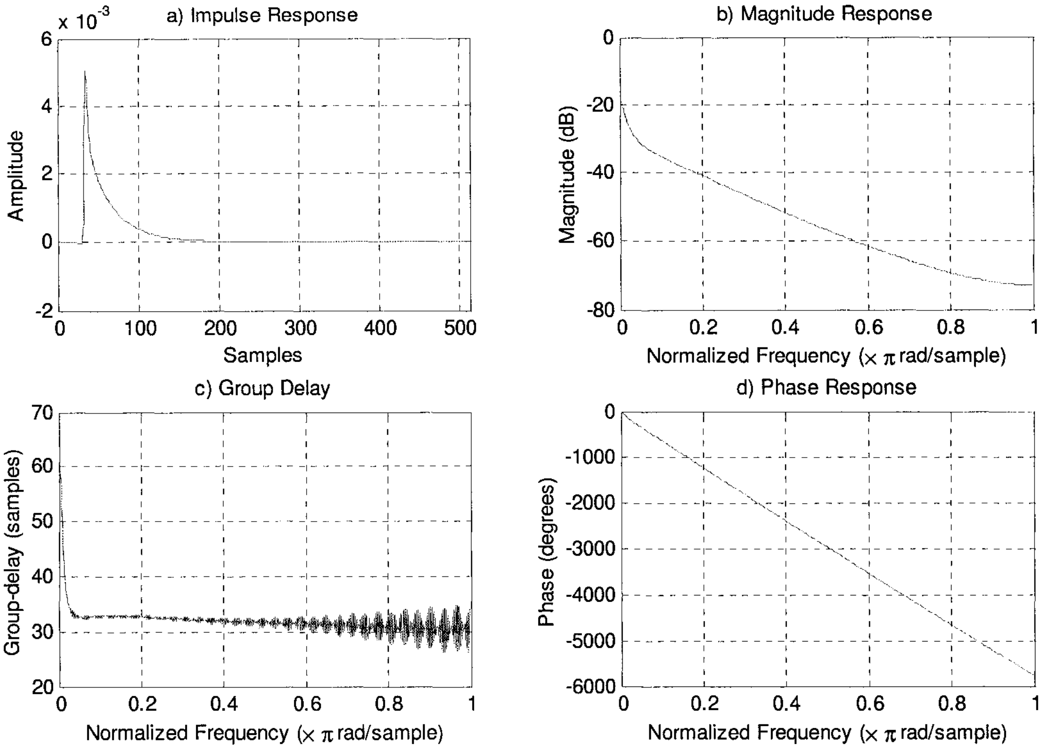


Figure A.6: CSA loop #6

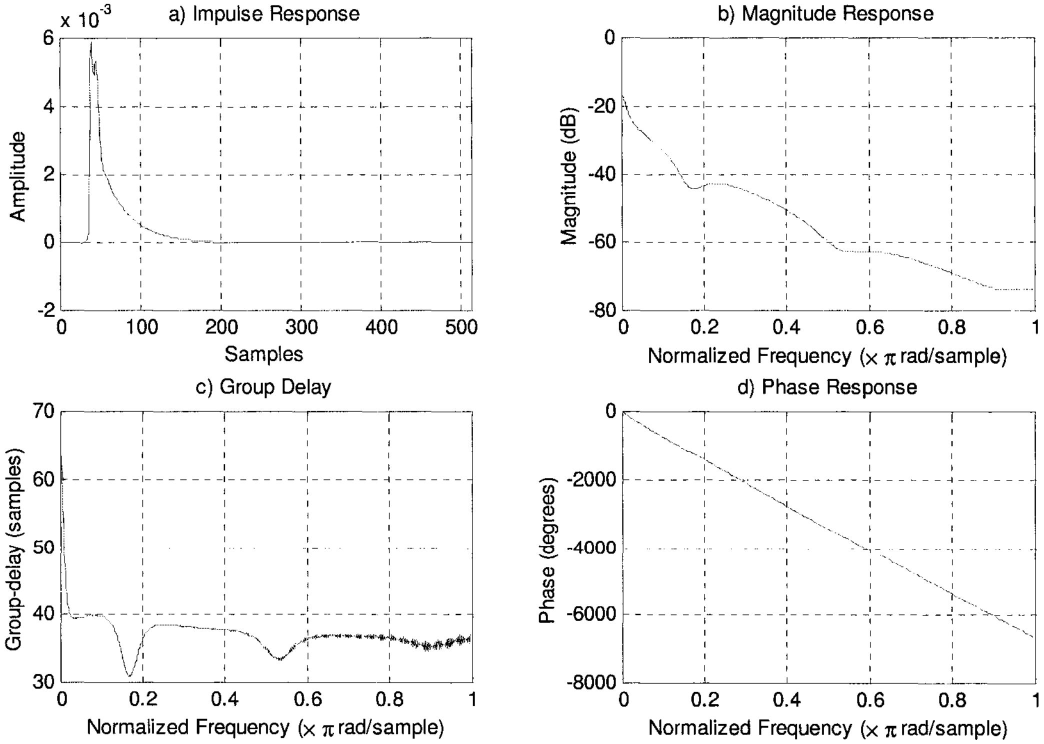


Figure A.7: CSA loop #7

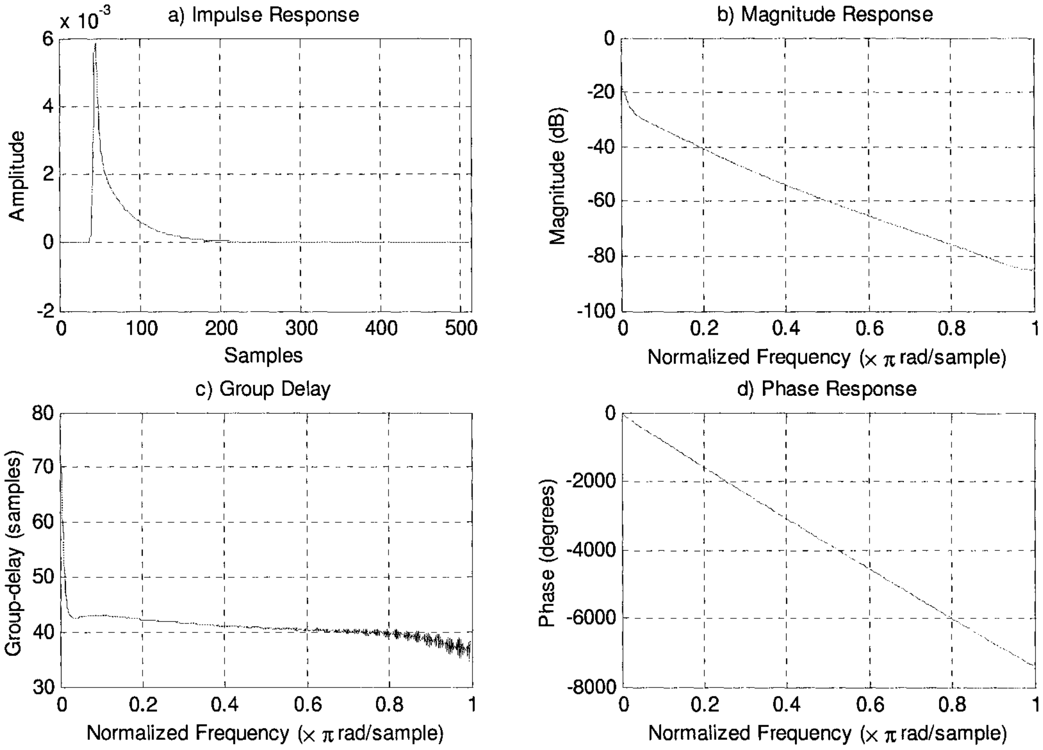


Figure A.8: CSA loop #8

Appendix B

Signal-to-Noise Ratio when Using MMSE Optimization

In this thesis, the MMSE optimization is employed to determine the optimal equalizer settings. Figure B.1 shows the illustration of equalization. In this figure, $S(z)$, $W(z)$ represent the transfer function between the system input and the corresponding receiver output, and the equalizer, respectively. $x(n)$, $q(n)$, $y(n)$, and $\hat{x}(n)$ represent the system input, the filtered channel noise, the receiver output, and the recovered signal, respectively.

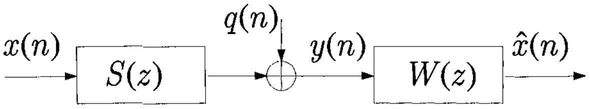


Figure B.1: Illustration of equalization

The recovered signal $\hat{x}(n)$ can be written as

$$\begin{aligned}\hat{x}(n) &= x(n) * s(n) * w(n) + w(n) * q(n) \\ &= \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} x(n-k-l)s(k)w(l) + \sum_{k=-\infty}^{\infty} q(n-k)w(k) \\ &= \alpha\left(x(n) + u(n)\right),\end{aligned} \tag{B.1}$$

where

$$\alpha \triangleq s(0)w(0)\,, \tag{B.2}$$

and

$$u(n) \triangleq \frac{1}{s(0)w(0)} \left(\sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} \sum_{\substack{l=-\infty \\ l\neq 0}}^{\infty} x(n-k-l)s(k)w(l) + \sum_{k=-\infty}^{\infty} q(n-k)w(k) \right). \tag{B.3}$$

The MMSE optimization is based on minimizing the MSE of $r(n)$, where $r(n)$ is the error between the system input $x(n)$ and the corresponding equalizer output $\hat{x}(n)$ in accordance with

$$r(n) \triangleq x(n) - \hat{x}(n)\,. \tag{B.4}$$

Then, the SNR becomes

$$\mathrm{SNR}_{\mathrm{MMSE}} \triangleq \frac{E\left[|x(n)|^2\right]}{E\left[|r(n)|^2\right]} = \frac{\sigma_x^2}{|\alpha|^2\,\sigma_u^2 + |1-\alpha|^2\,\sigma_x^2}\,, \tag{B.5}$$

where σ_x^2 and σ_u^2 represent the autocorrelation of $x(n)$ and $u(n)$, respectively. By comparing Eqn (B.5) to Eqn (1.6), it is clear that $\mathrm{SNR}_{\mathrm{MMSE}}$ is the biased SNR.

Appendix C

Relationship between Multi-Tap Interpolation Equalization and Per Tone Equalization

The per tone equalization is a special case of the multi-tap interpolation equalization when no cyclic prefix is used. This can be proved as follows.

From Eqn (3.51), the per tone equalization modifies the conventional DMT receiver such that a L_w-fold sliding DFT is used as the demodulator. Note that in Eqn (3.51), $N_E = N + L_v$. Then, by assuming that no cyclic prefix is used, one has $L_v = 0$, and $N_E = N$. With this assumption, Eqn (3.51) can be rewritten as

$$\begin{aligned}\hat{x}_i(n) &= \begin{bmatrix}\tilde{y}_{i,0}(n) & \tilde{y}_{i,1}(n) & \cdots & \tilde{y}_{i,L_w-1}(n)\end{bmatrix}\begin{bmatrix} w_{i,0} \\ w_{i,1} \\ \vdots \\ w_{i,L_w-1}\end{bmatrix} \\ &= \sum_{k=0}^{L_w-1} \tilde{y}_{i,k}(n) w_{i,k}\,,\end{aligned} \tag{C.1}$$

where

$$\tilde{y}_{i,k}(n) = \begin{bmatrix}1 & e^{j\frac{2\pi}{N}i} & \cdots & e^{j\frac{2\pi}{N}i(N-1)}\end{bmatrix}\begin{bmatrix} y(Nn+\Delta-k) \\ y(Nn+\Delta+1-k) \\ \vdots \\ y(N(n+1)+\Delta-1-k)\end{bmatrix}. \tag{C.2}$$

By denoting $e^{j\frac{2\pi}{N}i(N-t)}$ as h_{it} ($t = 0, ..., N-1$), Eqn (C.2) can be rewritten as

$$\tilde{y}_{i,k}(n) = \sum_{t=0}^{N-1} y(N(n+1)+\Delta-1-k-t)h_{it}\,. \tag{C.3}$$

Now by defining a filter $h_i(n)$ with the impulse response as

$$h_i(n) = \begin{cases} h_{in}, & 0 \le n \le N-1, \\ 0, & \text{otherwise,} \end{cases} \tag{C.4}$$

Eqn (C.3) becomes

$$\begin{aligned}\tilde{y}_{i,k}(n) &= \sum_{t=-\infty}^{\infty} y(N(n+1)+\Delta-1-k-t)h_i(t) \\ &= (y(n+N+\Delta-1-k)*h_i(n))_{\downarrow N} \\ &= \tilde{y}_i^{(\Delta-1-k)}(n+1)\,,\end{aligned} \tag{C.5}$$

where $\tilde{y}_i^{(\Delta-1-k)}(n+1)$ is the $(\Delta-1-k)$-th polyphase component of $\tilde{y}_i(n+1)$, and

$$\tilde{y}_i(n) = y(n) * h_i(n) \tag{C.6}$$

is the output of the filter $h_i(n)$ with the input of $y(n)$.

Substituting Eqn (C.5) into Eqn (C.1), one has

$$\hat{x}_i(n) = \sum_{k=0}^{L_w-1} \tilde{y}_{i,k}(n) w_{i,k} = \sum_{k=0}^{L_w-1} w_{i,k}\tilde{y}_i^{(\Delta-1-k)}(n+1)\,. \tag{C.7}$$

Then, by defining $l = \Delta - L_w$, Eqn (C.7) can be rewritten as

$$\hat{x}_i(n-1) = \sum_{k=0}^{L_w-1} w_{i,k}\tilde{y}_i^{(l+k)}(n)\,. \tag{C.8}$$

Moreover, by re-index the recovered symbols $\hat{x}_i(n)$, Eqn (C.8) becomes

$$\hat{x}_i(n) = \sum_{k=0}^{L_w-1} w_{i,k}\tilde{y}_i^{(l+k)}(n)\,, \tag{C.9}$$

which is exactly the same as Eqn (6.27).

With the above discussion, a conventional DMT system employing per tone equalization and without cyclic prefix is equivalent to the filterbank-based DMT system employing the interpolation equalization with the specifications as following.

- The filterbank-based DMT system using critically decimated filterbank as the modulator/demodulator,
- The synthesis filterbank is the IDFT filterbank, and the analysis filterbank is $\{h_i(n)\}$,
- The number of subchannels is N,
- The interpolation equalization combines L_w polyphase components of the analysis filter output, with each polyphase component weighted by a one-tap equalizer $w_{i,k}$.

Bibliography

[1] A.N. Akansu, P. Duhamel, X. Lin, and M. de Courville, "Orthogonal Transmultiplexers in Communication: A Review", *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 979–995, April 1998.

[2] N. Al-Dhahir and J. Cioffi, "Efficiently-Computed Reduced-Parameter Input-Aided MMSE Equalizers for ML Detection: A Unified Approach", *IEEE Transactions on Information Theory*, vol. 42, pp. 903–915, May, 1996.

[3] N. Al-Dhahir and J. M. Cioffi, "Optimum Finite-Length Equalization for Multicarrier Transceievers", *IEEE Trans. on Comm.*, vol. 44, pp. 56–93, Jan. 1996.

[4] ANSI T1.413-1995, "Network and Customer Installation Interfaces: Asymmetric Digital Subscriber Line (ADSL) Metallic Interface".

[5] Andreas Antoniou, *Digital Filters: Analysis, Design, and Applications*, McGraw-Hill, Inc., second edition, 1993.

[6] G. Arslan, B. L. Evans, and S. Kiaei, "Optimum Channel Shortening for Discrete Multitone Transceivers", *IEEE ICASSP'00*, vol. 5, pp. 2965–2968, 2000.

[7] John A. C. Bingham, "Multicarrier modulation for data trasnmission: an idea whose time has come", *IEEE Communications Magazine*, vol. 28, pp. 5–14, May 1990.

[8] John A.C. Bingham, *ADSL, VDSL, and Multicarrier Modulation*, John Wiley & Sons, Inc., 2000.

[9] Richard L. Burden and J. Douglas Faires, *Numerical Analysis*, Brooks-Cole Publishing, seventh edition, January 2001.

[10] Cable-Modems.Org, *Cable Modem Tutorial*, http://www.cable-modems.org/tutorial/.

[11] G. Cherubini, E. Eleftheriou, and S. Ölçer, "Filtered Multitone Modulation for VDSL", *IEEE GLOBECOM'99*, pp. 1139–1144, Dec. 1999.

[12] J. S. Chow, J. C. Tu, and J. M. Cioffi, "A Discrete Multitone Transceiver System for HDSL Applications", *IEEE Journal on Selected Areas in Comm.*, vol. 9, no. 6, pp. 895–908, Aug. 1991.

[13] J. M. Cioffi, *Course Reader of EE379A - Digital Communication: Signal Processing*, Stanford University, Winter, 2001 - 02.

[14] J. M. Cioffi, T. Starr, and P. Silverman, "Digital Subscriber Lines", *Comp. Networks and ISDN Sys., Special Issue on Broadband Access*, Dec. 1998.

[15] CISCO systems, *ISDN Glossary*, http://www.cisco.com/warp/public/129/27.html.

[16] R.E. Crochiene, I.R. Rabiner, and R.R. Shively, "A novel implementation of digital phase shifters", *Bell Syst. Tech. J.*, vol. 54, pp. 1497–1502, 1975.

[17] N. Dagdeviren, R. Laroia, R. Gitlin, and S. Celebi, "High-Performance, Low Complexity Discrete Multi-Tone (DMT) ADSL Systems", Tech. Rep., TG/98-033, UADSL, Pleasanton, CA, Feb. 1998.

[18] B. Farhang-Boroujeny and W. H. Chin, "Time domain equaliser design for DWMT multicarrier transceivers", *Electronics Letters*, vol. 36, no. 18, pp. 1590–1592, Aug. 2000.

[19] C.W. Farrow, "A continuously Variable Digital Delay Element", *Proc. IEEE Int. Symp. Circuits and Systems,* Espoo, Finland, pp. 2461–2465, 1988.

[20] DSL Forum, *VDSL Tutorial*, http://www.dslforum.org/.

[21] S. Gracias and V. U. Reddy, "An Equalization Algorithm for Wavelet Packet Based Modulation Schemes", *IEEE Transactions on Signal Processing*, vol. 46, no. 11, pp. 3082–3087, Nov. 1998.

[22] M. Isaksson, D. Bengtsson, P. Deutgen, M. Sandell, F. Sjöberg, P. Ödling, and H. Öhman, "Zipper – a duplex scheme for VDSL based on DMT", Tech. Rep., T1E1.4/97-016, ANSI, TX, Jan. 1997.

[23] M. Isaksson, P. Deutgen, F. Sjöberg, S. K. Wilson, P. Ödling, and P. O. Börjesson, "Zipper – a flexible duplex method for VDSL", *Proc. Intern. Workshop on Copper Wire Access Systems (CWAS),* Budapest, Hungary, pp. 95–99, Oct. 1997.

[24] K.S. Jacobsen and J.M. Cioffi, "An Efficient Digital Modulation Scheme for Multimedia Transmission on the Cable Television Network", *Technical Papers, 43rd Annual National Cable Television Association (NCTA) Convention and Exposition,* New Orleans, LA, May 1994.

[25] T. Karp and A. Mertins, "Biorthogonal cosine-modulated filter banks without DC leakage", *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, Seattle, WA, USA*, vol. 3, pp. 1457–1460, May 1998.

[26] R.D. Koilpillai and P.P. Vaidyanathan, "New results on cosine-modulated FIR filter banks satisfying perfect reconstruction", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.,* (Toronto, Canada), pp. 1793–1796, May 1991.

[27] Peter J. Kootsookos and Robert C. Williamson, "FIR Approximation of Fractional Sample Delay Systems", *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 43, no. 3, pp. 269–271, March, 1996.

[28] B. P. Lathi, *Modern Digital & Analog Communication Systems*, Oxford University Press, 1998.

[29] Inkyu Lee, Jacky Chow, and J. M. Cioffi, "Performance Evaluation of a Fast Computation Algorithm for the DMT in High-Speed Subscriber Loop", *IEEE Journal on Selected Areas in Comm.*, vol. 13, no. 9, pp. 1564–1570, Dec. 1995.

[30] A. Leke and J. M. Cioffi, "A Maximum Rate Loading Algorithm for Discrete Multitone Modulation", *IEEE GLOBECOM'97*, vol. 3, pp. 1514–1518, 1997.

[31] Yuan-Pei Lin and See-May Phoong, "Optimal DMT Transceivers Over Fading Channels", *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, pp. 1397–1400, Mar. 1999.

[32] Yuan-Pei Lin and P. P. Vaidyanathan, "Linear phase cosine modulated maximally decimated filter banks with perfect reconstruction", *IEEE Transactions on Signal Processing*, vol. 43, no. 11, pp. 2525–2539, 1995.

[33] Yuan-Pei Lin and P. P. Vaidyanathan, "A Kaiser Window Approach for the Design of Prototype Filters of Cosine Modulated Filterbanks", *IEEE Signal Processing Letters*, vol. 5, no. 6, June, 1998.

[34] G. S. Liu and C. H. Wei, "Programmable fractional sample delay filter with Lagrange interpolation", *Electronic Letter*, vol. 26, no. 19, pp. 1608–1610, 1990.

[35] Ging-Shing Liu and Che-Ho Wei, "A New Variable Fractional Sample Delay Filter with Nonlinear Interpolation", *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 39, no. 2, pp. 123–126, Feb. 1992.

[36] B. Lu, L. D. Clark, G. Arslan, and B. L. Evans, "Fast Time-domain Equalization for Discrete Multitone Modulation Systems", *Proc. IEEE Digital Signal Processing Workshop,* Hunt, Texas., Oct. 15-18, 2000.

[37] H. S. Malvar, *Signal Processing with Lapped Transforms*, Boston, MA: Artech House, 1992.

[38] H.S. Malvar, "Extended lapped transforms: properties, applications, and fast algorithms", *IEEE Transactions on Signal Processing*, vol. 40, pp. 2703–2714, Nov. 1992.

[39] P. J. W. Melsa, R. C. Younce, and C. E. Rohrs, "Impulse Response Shortening for Discrete Multitone Transceivers", *IEEE Trans. on Comm.*, vol. 44, pp. 1662–1672, Dec. 1996.

[40] David G. Messerschmitt, "The Convergence of Telecommunications and Computing: What Are the Implications Today?", *PROCEEDINGS OF THE IEEE*, vol. 84, no. 8, pp. 1167–1186, Aug. 1996.

[41] Alan V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[42] Soo-Chang Pei, Bi-Ruei Chiou, and Peng-Hua Wang, "Programmable Fractional Sample Delay Filters with Flatness Compromise Between Magnitude Response and Group Delay", *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 47, no. 8, pp. 783–787, Aug. 2000.

[43] T.A. Ramstad and J. Tanem, "Cosine-modulated analysis-synthesis filterbank with critical sampling and perfect reconstruction", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.,* (Toronto, Canada), pp. 1789–1792, May 1991.

[44] G. V. Reklaitis, A. Ravindran, and K. M. Ragsdell, *Engineering Optimization: Methods and Applications*, John Wiley, Toronto, 1983.

[45] S. D. Sandberg and M. A. Tzannes, "Overlapped Discrete Multitone Modulation for High Speed Copper Wire Communications", *IEEE Journal on Selected Areas in Comm.*, vol. 13, no. 9, pp. 1571–85, Dec. 1995.

[46] Anna Scaglione, Georgios B. Giannakis, and Sergio Barbarossa, "Redundant Filterbank Precoders and Equalizers Part II: Blind Channel Estimation, Synchronization, and Direct Equalization", *IEEE Transactions on Signal Processing*, vol. 47, no. 7, pp. 2007–2022, Jul. 1999.

[47] G. Schuller and T. Karp, "Modulated filter banks with arbitrary system delay: Efficient implementations and the time-varying case", *IEEE Trans. on Signal Processing*, Mar. 2000.

[48] C. E. Shannon, "A Mathematical Theory of Communication", *The Bell System Technical Journal*, vol. 27, no. 10, pp. 379–423, 623–656, Oct. 1948.

[49] F. Sjöberg, M. Isaksson, R. Nilsson, P. Ödling, S. K. Wilson, and P. O. Börjesson, "Zipper: A Duplex Method for VDSL Based on DMT", *IEEE Transactions on Communications*, vol. 47, pp. 1245–1252, Aug 1999.

[50] Andrew S. Tanenbaum, *Computer Networks*, Prentice Hall, third edition, 1996.

[51] J. M. Cioffi *et al.*, "Very-High-Speed Digital Subscriber Lines", *IEEE Communications Magazine*, pp. 72–79, April 1999.

[52] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice Hall, 1993.

[53] P. P. Vaidyanathan, "Filter Banks in Digital Communications", *IEEE Circuits and Systems Magazine*, vol. 1, no. 2, pp. 4–25, Second Quarter, 2001.

[54] P. P. Vaidyanathan and S. Akkarakaran, "A review of the theory and applications of principal component filter banks", *Journal of Applied and Computational Harmonic Analysis*, to be published.

[55] P. P. Vaidyanathan and Ahmet Kirac, "Results on optimal biorthogonal filter banks", *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 45, pp. 932–947, 1998.

[56] P. P. Vaidyanathan, Yuan-Pei Lin, S. Akkarakaran, and See-May Phoong, "Optimality of principal component filter banks for discrete multitone communication systems", *Proc. IEEE Int. Symp. Circuits and Systems*, Geneva, May 2000.

[57] K. Van Acker, G. Leus, M. Moonen, O. van de Wiel, and T. Pollet, "Per Tone Equalization for DMT-based Systems", *IEEE Trans. on Comm.*, vol. 49, pp. 109–119, Jan. 2001.

[58] Luqing Wang and Behrouz Nowrouzian, "Development of a Post-Detection Equalization Technique for Multicarrier Modulation/Demodulation Systems", *The 45th IEEE Midwest Symposium On Circuits and Systems,* Tulsa, Oklahoma, August 4-7, 2002.

[59] Luqing Wang and Behrouz Nowrouzian, "A New Interpolation Equalization Scheme for Discrete Wavelet Multitone Modulation/Demodulation Systems", *The 45th IEEE Midwest Symposium On Circuits and Systems,* Tulsa, Oklahoma, August 4-7 2002.

[60] H. Zou and A.H. Tewfik, "Design and parameterization of M-band orthonormal wavelets", *IEEE International Symposium on Circuits and Systems, (ISCAS '92)*, vol. 2, pp. 983–986, 1992.