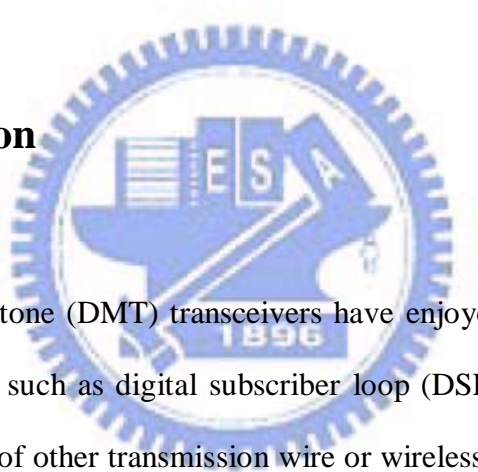# Chapter 1

Introduction

1.1 Introduction

The discrete multi-tone (DMT) transceivers have enjoyed great success in high speed data transmission such as digital subscriber loop (DSL). They also have been applied in a wide range of other transmission wire or wireless channels. It is typically called DMT for wired applications and OFDM (orthogonal frequency division multiplexing) for broadcasting or wireless applications, e.g., digital audio broadcasting (DAB) and digital video broadcasting (DVB). By inserting a cyclic prefix that is long enough, the frequency-selective channel is converted into M frequency non-selective parallel sub-channels. The input bit stream is parsed and coded as modulation symbols, e.g., PAM or QAM. High speed data transmission can be achieved using the DMT scheme at a relatively low cost.

The first part of this thesis is the time-domain equalizer (TEQ) in the DMT systems. In a DMT scheme, the serial bitstream is split in parallel streams in order to

modulate a carrier in each band. After modulation, implemented by IFFT, a cyclic prefix is added to each symbol. If the length of this prefix is equal or longer than the length of the channel impulse response minus one, the receiver can reconstruct the transmitted symbols without inter-carrier interference nor inter-symbol interference. The longer the channel impulse response is, the longer the cyclic prefix is needed to avoid ISI. For applications such as DSLs where channel response can be very long, a time-domain equalizer(TEQ) is used to shorten the effective channel response. In the past, many methods have been proposed for the design of the TEQ [2][4][7][11][13]. These methods can be categorized into 2 approaches, Single-Path TEQ and Per-Tone TEQ.

Single-Path TEQ also can be categorized into 3 types. First one is to design the TEQ to shorten the effective channel response or delay spread. This approach maximizes the energy (or weighted) of the effective channel impulse response within a certain window. It does not consider the effect of the channel noise. The second type minimizes the mean square error (or equivalently maximizes the SNR) at the TEQ output. Both of these approaches are not optimal. The resulting TEQ does not give the best bit error rate or the maximum bit rate performance. The third one optimizes the geometrical mean of SNRs of all tones. This approach involves highly nonlinear optimizations though it is optimal. Suboptimal solution by replacing the geometrical mean with arithmetic mean has been given in [2].

In per-tone TEQ, a T-taps FEQ is inserted for each tone separately. This scheme enables us to do real SNR-optimization per tone while complexity during data transmission is kept at the same level [4]. Moreover, the per-tone equalization has been shown to have a reduced sensitivity to the synchronization delay.

In this thesis, we propose a DMT scheme with modified per-tone TEQs. When all tone's weights are set to be equal, the proposed DMT scheme reduces to the DMT

scheme with a single TEQ studied in [2][7][13][14]. Tones are combined into groups and in each group, the per-tone equalizer is computed only for the center tone and is then reused for the whole group. By this method, We can reduce the memory and complexity during initialization but the bit rate will not decrease very much as we will demonstrate in simulation examples.

1.2 Organization of the thesis

This thesis is organized as follows. Chapter 2 is background. First part introduces the ADSL environment. In second part, DMT transceiver system is introduced. Chapter 3 describes several kinds of time domain equalizer are summarized. In chapter 4, we proposed the tone-grouping method and simulation results are presented with some discussion. Finally, conclusions are drawn in chapter 5.

Chapter 2

Background

In this thesis, we will apply DMT systems over ADSL environment. Therefore, in chapter 2.1, we will give a brief introduction to asymmetric digital subscriber line (ADSL). The channel model will also be introduced. At the end of the chapter 2.2, we will discuss the noises including crosstalk noise and additive white Gaussian noise (AWGN), which often exist in an ADSL environment. The second part of this chapter is the introduction of DMT system, which is commonly used in applications such as mobile communications, digital audio broadcasting, digital terrestrial TV, and xDSL systems.

2.1 ADSL System

2.1.1 ADSL Introduction

Asymmetric Digital Subscriber Lines (ADSL) is a local loop transmission

technology via one pair of wires. The system transports downstream signal, upstream signal, and Plain Old Telephone Service (POTS) over the same medium simultaneously. ADSL will support high bit rate services on standard twisted pair telephone lines, uni-directionally, from the central office to the customer premises with the intended application of compressed video distribution. The term *asymmetric* comes from the fact the transmission rates for upstream and downstream signals are different. The downstream bit rate (toward customer) is up to about 9 Mbps and the upstream (toward network) bit rate is up to 1Mbps.

Many ADSL systems use frequency division multiplexed (FDM) transmission techniques that places POTS signal, upstream signal and downstream signal at different frequency bands to prevent self-crosstalk. In Fig. 2.1 we can see the different frequency bands of downstream, upstream, and POTS. Here the guard band is necessary to facilitate filters to prevent POTS signal from interfering with the digital transmission. Some ADSL systems make the upstream frequency band resides within the downstream band.
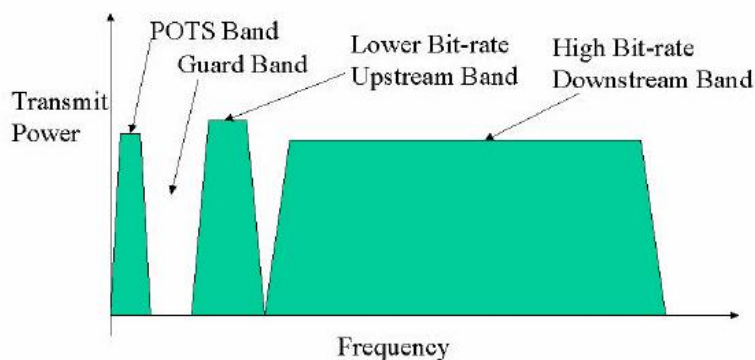

Fig 2.1 FDM ADSL

The way modulating the transmitted signal, competing transmission techniques for ADSL include CAP (carrier-less amplitude and phase modulation) and DMT (discrete multi-tone modulation). Among these techniques, only DMT system (also

called OFDM, orthogonal frequency division multiplexing) uses multi-tone technology and is the most competitive player.   In this thesis we will focus on DMT systems and DMT will be studied in detail in chapter 2.2. For transmissions on medium, like twist pairs, the digital modulated signals must be converted to analog signals. The signal will be distorted by the channel and corrupted by noises such as crosstalk and additive white Gaussian noise.   In the next section we will study the environment of twisted pair telephone lines. Channel transfer function and channel noise of twisted pair will be described. The receiver samples these analog signals and converts them into digital signals. Thus the transmitter and receiver systems use the same sampling frequency (in the degree of MHz for the ADSL system). The received symbols in general suffer from very severe Inter-symbol Interference (ISI). The twisted pair telephone lines have a very long ISI in ADSL application because of high sampling frequency. Therefore, an equalizer is developed to shorten this ISI.

2.1.2 ADSL Environment

In this section the loop characteristics over twisted pairs and their impairments will be introduced.

A. Channel Model

Twisted pairs are often characterized by their gauge using American Wire Gauge (AWG) designations and the gauge indicates the diameter of the copper wires making up the twisted pair. The AWG number N represents 1/Nth of an inch. For example as Table 2.1, the smaller gauge number is corresponding to the larger diameter. Gauges #24 and #26 are most common in DSL applications. The carrier serving area (CSA) is an identifiable subset of the current subscriber loop population in the U.S. Some

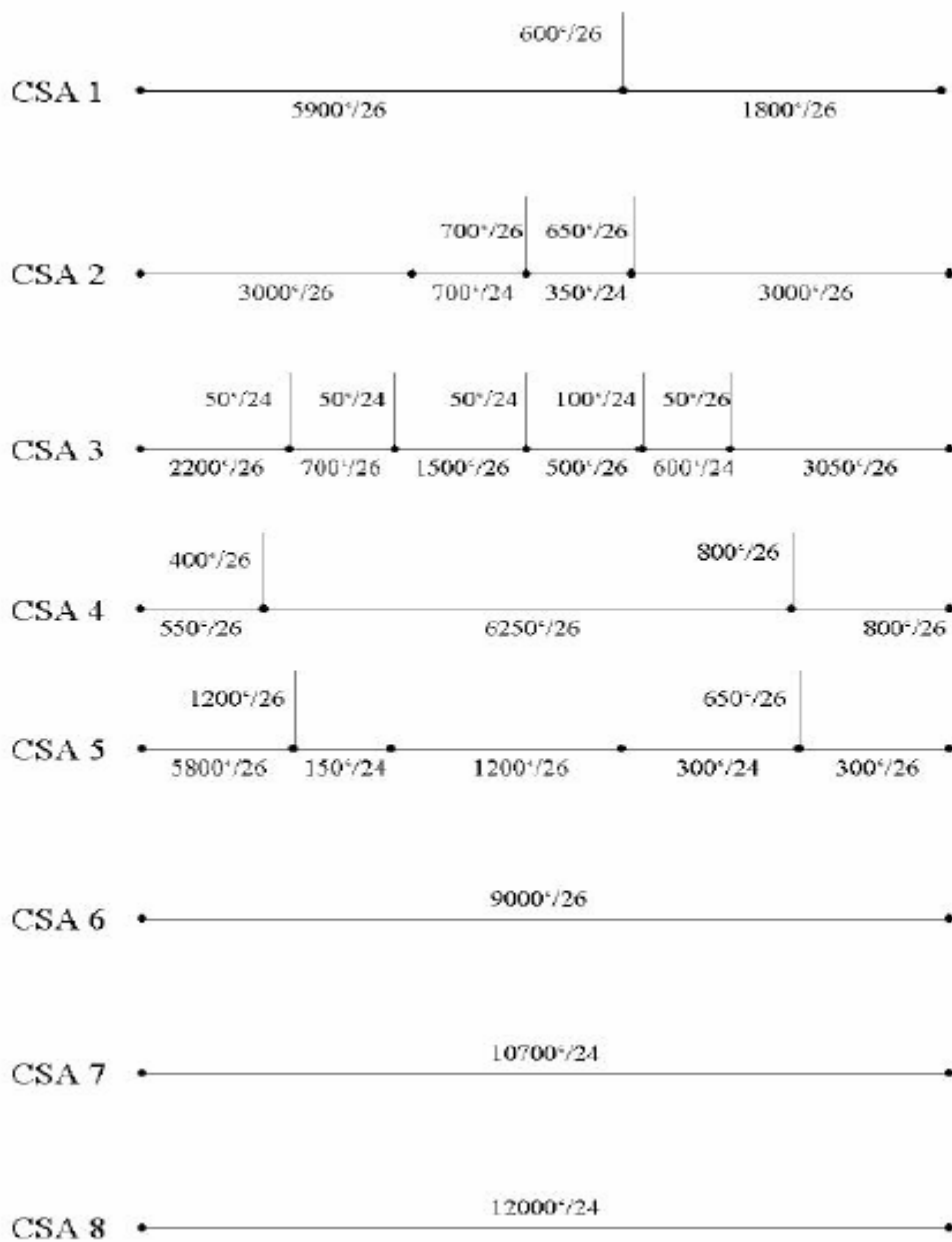| AWG	Diameter(in)
19	0.03589
22	0.02535
24	0.02010
26	0.01594

Table 2.1

Figure 2.2 ANSI CSA Loops (Feet/AWG)

standard CSA loops are shown in Fig. 2.2. By the ANSI T1.413 (ITU g.dmt) ADSL

standards, the downstream system employs a sampling rate of 1/T'=2.208MHz and

the upstream system employs a sampling rate of 1/T'=276kHz. The channel impulse
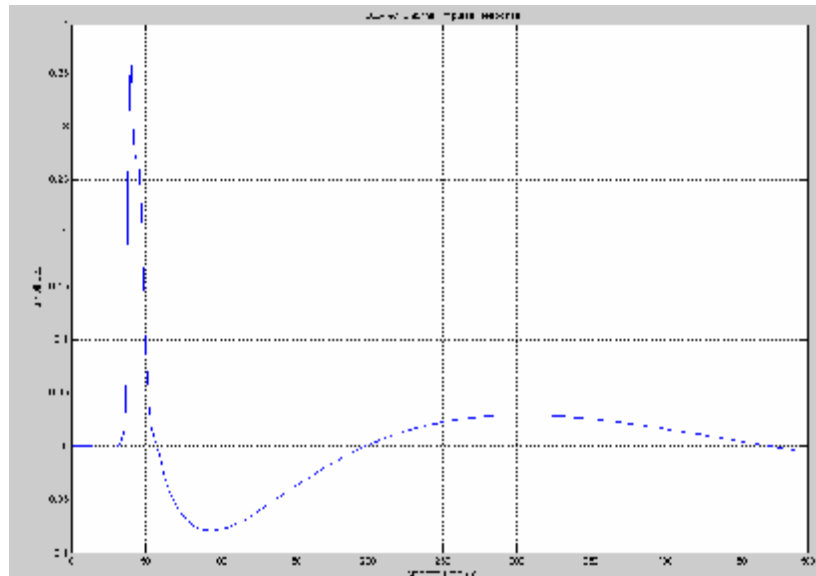

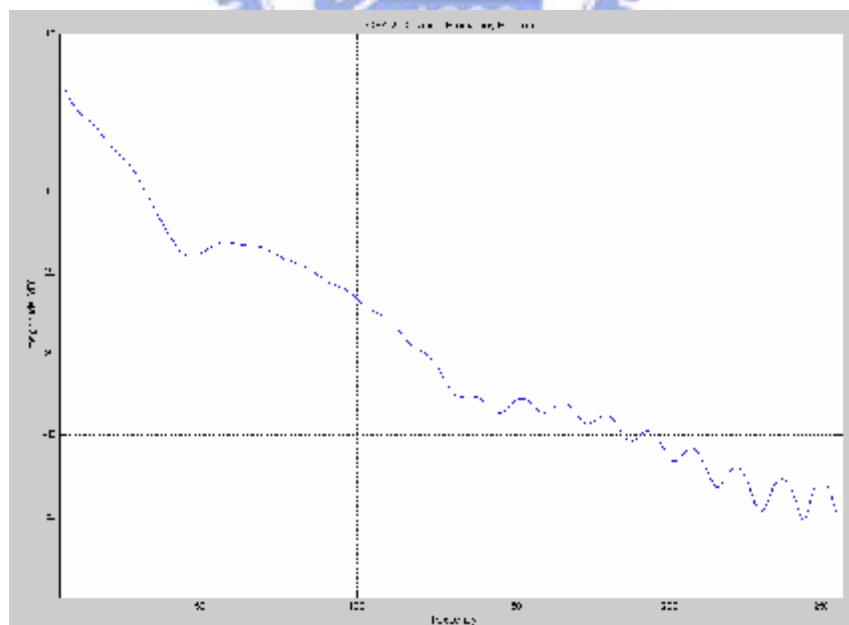Figure 2.3 CSA #7 channel impulse response with 2MHz sampling rate

Figure 2.4 CSA #7 channel magnitude response with 2MHz sampling rate

response for the CSA Loop #7 is shown in Fig. 2.3. With the sampling frequency of

2MHz, this impulse response will cause an ISI of several hundred samples. The magnitude response of the channel is shown in Fig. 2.4. As we can see, the distortion for data signal is much higher than that of the POTS signal. The higher the frequency is, the larger the distortion is.

B. Noise in the system

Besides Inter-symbol Interference (ISI) which will discuss in chapter 2.3, the data transmission system over ADSL loops are susceptible to various types of impairment, like the crosstalk noise, the Additive White Gaussian Noise (AWGN), etc.
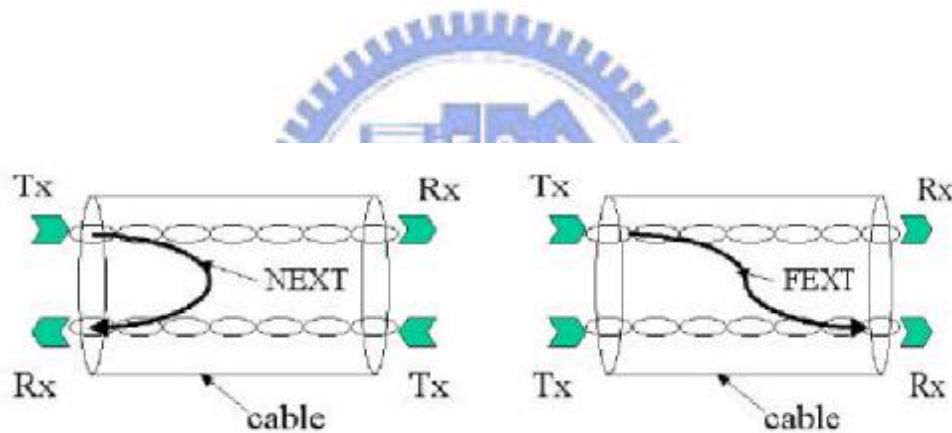

Fig 2.5 Near-end crosstalk(NEXT) and Far-end crosstalk(FEXT)

Crosstalk is most pronounced at the segment of cable near the interfering transmitters. A telephone cable contains up to several thousand separate wire pairs packed closely together. The crosstalk resulting from other transmission systems is a primary factor limiting the transmission rates of most DSL systems. Crosstalk Noise can be modeled by two terms, the near-end crosstalk (NEXT) and the far-end crosstalk (FEXT). We express these two crosstalk noises in Fig. 2.5.

Near-end crosstalk (NEXT). Next noise is seen by the receiver located at the

same end of the cable as the transmitter that is the noise source. NEXT is a major impairment for systems that share the same frequency band for upstream and downstream transmissions.

In this paper, we assume near-end crosstalk (NEXT) whose power spectral density (PSD) is given by

$$PSD_{NEXT}(f) = k_{NEXT} f^{3/2} \times PSD_2(f)$$

where $k_{NEXT} = 2.1581 \times 10^{-9}$ and $PSD_2(f)$ is the PSD of the transmitted signal.

Far-end crosstalk (FEXT). FEXT is detected by the receiver located at the far end of the cable from the transmitter that is the noise source. FEXT is less severe than NEXT because FEXT noise is attenuated by traversing the full length of the cable. The general expression for a FEXT disturber, based on the PSD of the disturbing signal, is given by[9]

$$PSD_{FEXT}(f) = K_{FEXT} \times f^2 \times l \times |H_{channel}(f)|^2 \times PSD_2(f) \qquad (2.1)$$

where $H_{channel}(f)$ is the response of the channel, l is the length of the cable in ft, and K_{FEXT} is a coupling constant determined by experiment.

AWGN. The AWGN channel is also used to simulate electric and thermal noises of the channel. Typically AWGN impairment is in the order -140 to -170 dBm/Hz. In this thesis, we assume AWGN is -140 dBm /Hz.

Here we only discuss about NEXT and AWGN noises in this paper and ignore other noises for convenience.

2.2 DMT System

In this chapter, the basic structure of the DMT systems will be discussed. In chapter 2.3, the interference structure created by a non-ideally equalized channel is analyzed.
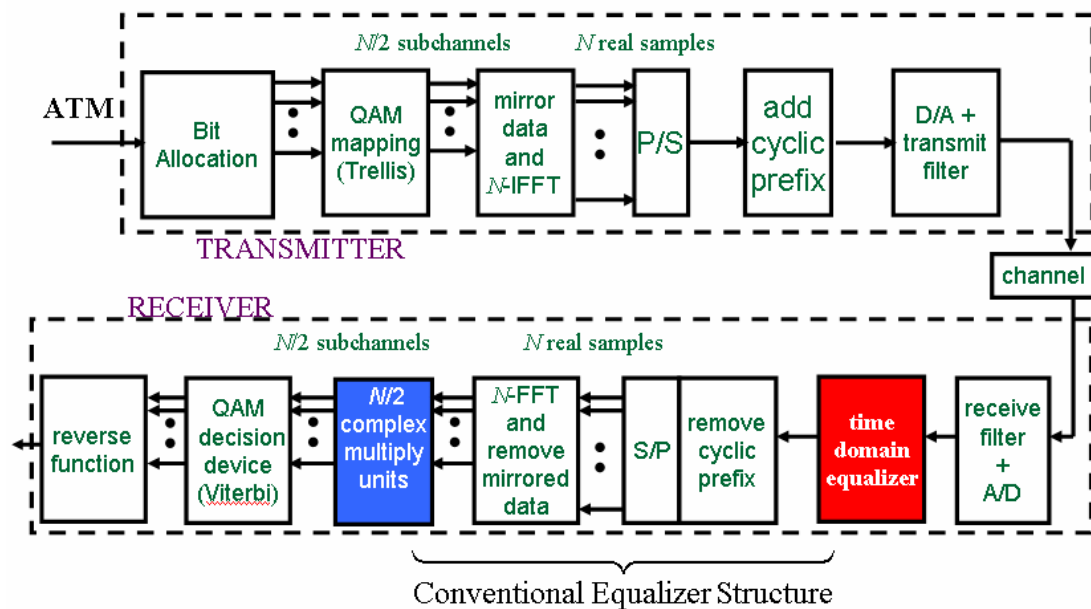

Figure 2.6 DMT scheme

Recently, various multi-carrier systems have been proposed to transmit data reliably in the presence of severe inter-symbol interference for digital subscriber line application. The DMT, one of the multi-channel transmission methods, achieves very high levels of performance and is used in ADSL. In the modulation for multi-channel, a transmission channel is partitioned into a multitude of sub-channels in frequency domain. We have to use frequency division multiplexing to transport a single–input data stream on several carriers within the frequency band of the channel. In the range of twisted pairs transmission, such as ADSL applications, the DFT-based DMT

system is widely adopted to realize the transmultiplexers. As shown in Fig. 2.6, the transmitting end of a DFT-based DMT system consists of a coding and serial-to-parallel conversion, a QAM (Quadrature Amplitude Modulation) encoder, an IFFT block, a cyclic prefix insertion, and a parallel-to-serial conversion. The receiving end of a DFT-based DMT system consists of an equalizer, a cyclic prefix removal, a blocking conversion, a DFT block, a QAM decoder, and a parallel-to-serial conversion. In the following, we will describe each of these blocks individually. The equalizer has been studied in Chapter 3.

Serial-to-Parallel and Parallel-to-Serial Conversions The concepts of serial-to-parallel and parallel-to-serial conversion are trivial. In S/P, a long stream of data is broken up into several equal-length (or approximately equal-length) "chunks" which can all be operated upon at the same time. In P/S, these pieces are concatenated one right after the another into a long data stream. Their implementation is also trivial; all one has to do is manipulate a vector into several columns of a matrix, or take a matrix and concatenate its columns. However, S/P conversion is very important in DMT. The blocks produced in S/P are the input to the constellation mapping, which is basically representing segments of bits as spectral coefficients. Also S/P is essential in choosing how many frequencies are to be used. Usually, the block length is a power of 2, which makes the IFFT and FFT algorithms most computationally efficient.

Constellation Mapping In the constellation mapping process, length *B* segments of the bitstreams are assigned a single complex value in a **constellation**. This is similar to quadrature amplitude modulation (QAM), because the IFFT of these complex points maps to various sinusoids with amplitudes proportional to the real part and phase shifts proportional to the complex part. We mostly used 2-bit (4 point) and 4-bit (16 point) constellations in our system, but we also coded a function that generates a mapping with an arbitrary number of points. In comparison, constellations

used in actual DMT systems can have as many as 240 points. Important characteristics of these constellations include the area occupied by the points and the spacing between points. The power necessary to transmit the actual signal is proportional to the area, so it is key not to let the points be too far apart. However, the channel introduces noise which results in the constellation points being received with error. If the points are too close together, it is impossible to correctly match the received points with the correct value in the constellation. We use Gray code to assign the bit streams to constellation points. The result mapping is shown in table2.2. The advantage of Gray code is that any two adjacent QAM symbols are differed only by 1 bit. Therefore the bit error rate can be reduced. The QAM decoder is known as a maximum likelihood detector which decision rule is

$$bit_{decoded} = bit_{pos} \text{ if } \| X_{decoded} - X_{pos} \| \leq \| X_{decoded} - X_{pos} \| \text{ for all } j \neq pos \text{ (2.2)}$$

and we can also see the decision region of QAM decoder in Fig. 2.7.

16-Point Mapping

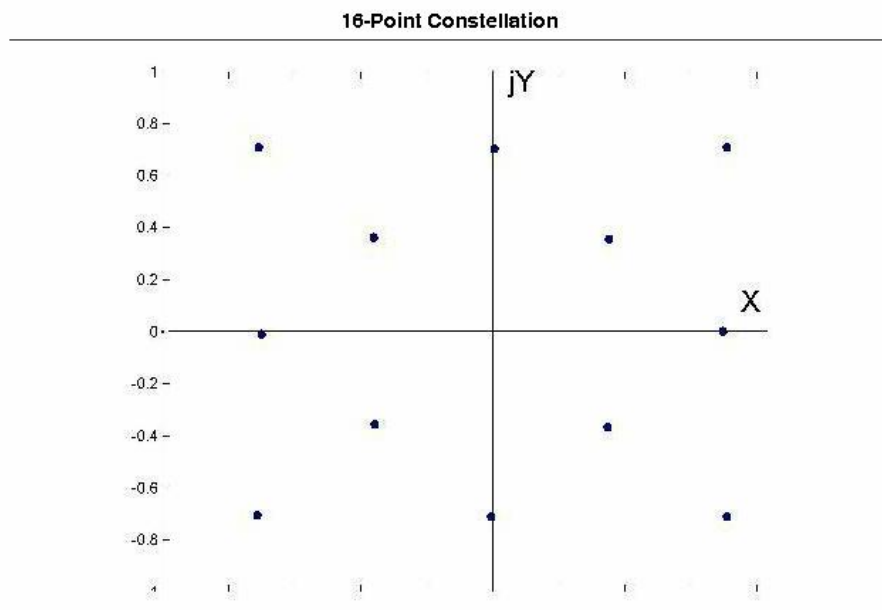| Bit Sequence	Value	Bit Sequence	Value
0000	.354+.354j	1000	1
0001	.707	1001	.707+.707j
0010	.707j	1010	j
0011	-.354+.354j	1011	-.707+.707j
0100	-.707j	1100	-1
0101	.354-.354j	1101	-.707-.707j
0110	-.354-.354j	1110	-j
0111	-.707	1111	.707-.707j

Table 2.2 16-point Mapping

Fig 2.7 16-point Constellation

Mirror/IFFT, De-Mirror/FFT Like serial/parallel conversion, mirroring is simple both in principle and implementation, but is essential to DMT. We need to eventually transmit a real-valued signal, but after constellation mapping, the blocks are all complex-valued. Mirroring ensures that the transmitted signal is real-valued by using the fact that the IFFT of a conjugate-symmetric vector is real; i.e., mirroring gives each block conjugate symmetry. Fig 2.8 shows a example. This is one block (length 128) of constellation points after mirroring (magnitude only). The mirrored points is shown in blue In mirroring, each block is flipped around, conjugated and tacked flipped around, conjugated and tacked onto the end of the original block. There are a few finer points, however. If the block is length M before mirroring, the $(M+1)$th element of the mirrored block takes any real value (we set it to zero) while the $(M+2)$th through $2M$th elements take the 1st through (M-1)th elements (flipped) of the original block. Also, the first element in any block cannot be complex valued; it must
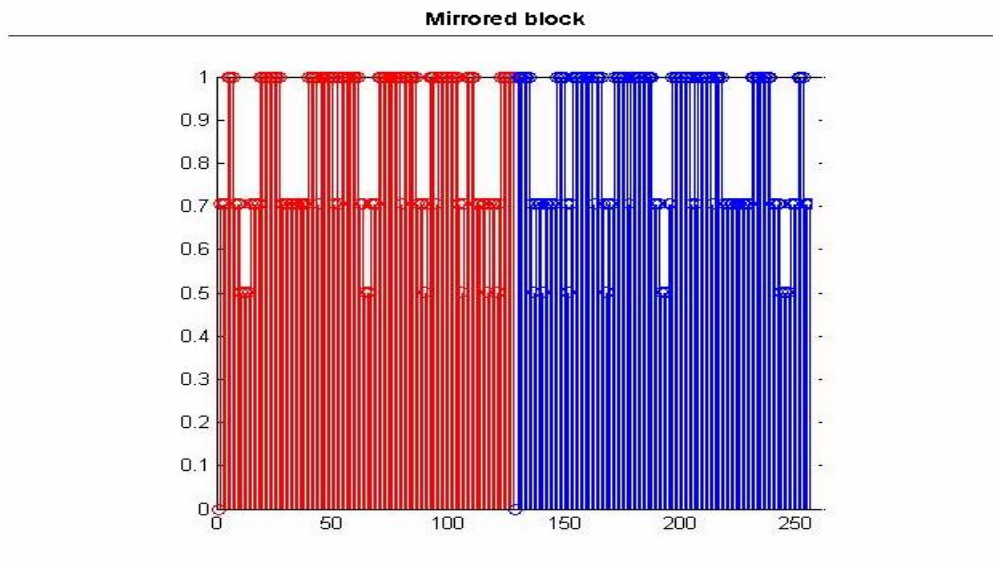
Fig 2.8 Mirrored block

be artificially set to some real value. This is because the first point in any block corresponds to a frequency of zero after the IFFT; if this element is complex, the resulting IFFT vector will have a complex DC offset and thus will also be complex. Although this would be a source of error, we assumed that the receiver knew the correct first elements of each block, eliminating this error. We felt we were justified in this regard because our main goal was to implement a DMT system resilient to *channel* error (We could also have coded it so that the first few bits in each block were assigned real-valued constellation points, but this way was easier).

The ***IFFT*** basically provides the modulation for DMT. As hinted at in the S/P module, the complex constellation points in each block (now mirrored) are interpreted as the spectral coefficients for a time-domain signal; the IFFT gives us this signal. By ensuring that our block length remains a power of 2 through the mapping and mirroring process, we make sure that the IFFT is most computationally efficient. We simply used the IFFT function built into MATLAB in our system. Fig 2.9 Shows the IFFT of Fig 2.8. It is real-valued and the same length as the mirrored block.
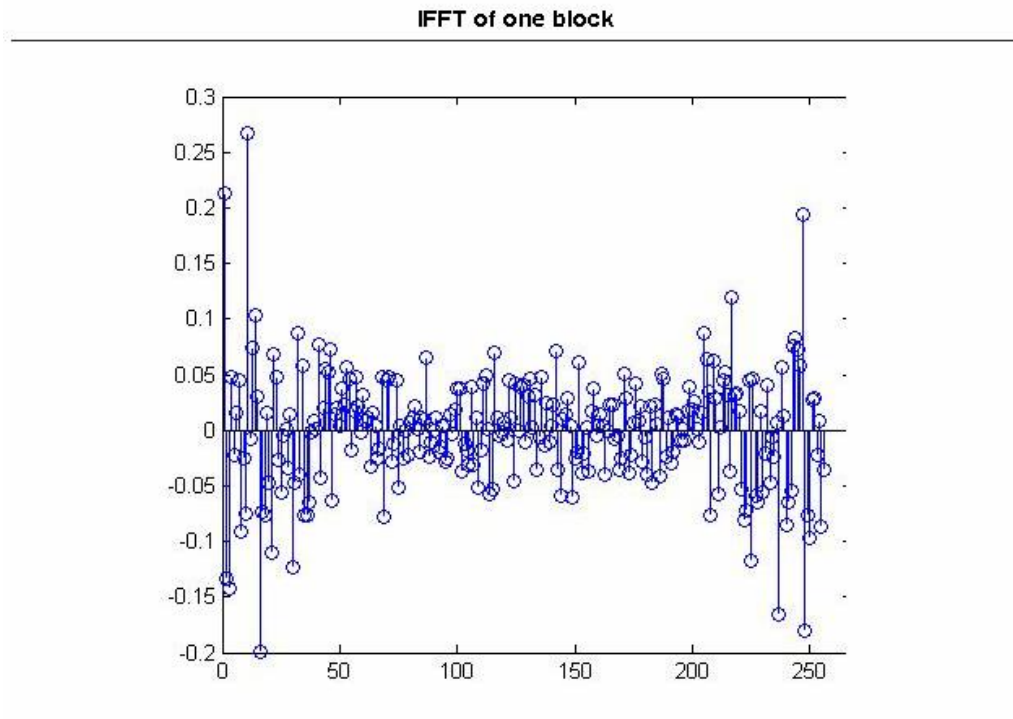
Fig 2.9 IFFT of the mirrored block in Fig 2.8

FFT/De-Mirror After going through the channel, the blocks undergo an inverse of the transmission process. Once the received time-domain blocks are separated, an FFT returns the spectral coefficients of each block, which are "noisy" versions of the post-mirroring values. This can be looked at as demodulating in a way; the FFT essentially multiplies the sinusoids by in the time-domain waveform by their conjugates. De-mirroring is straight forward; we want to get rid of the conjugate-symmetric part of each block so the last M values (for a block with pre-mirroring length M) are simply thrown away. Again, the FFT algorithm was built into MATLAB and the de-mirroring code was trivial.

Cyclic Prefix The addition of a cyclic prefix to each symbol solves both ISI and ICI. In our system, we assume the channel impulse response has a known length L.

The prefix consists simply of copying the last L-1 values from each symbol and appending them in the same order to the front of the symbol. By having this buffer of essentially junk data in the front, the convolution of the impulse response with the signal at the end of a symbol does not affect any of the actual data at the beginning of the next symbol. In addition, by repeating the last elements at the beginning, the first real "data" elements of each symbol experience overlap with the "end" of the symbol. Just as in cyclic convolution. This means the linear convolution of the channel
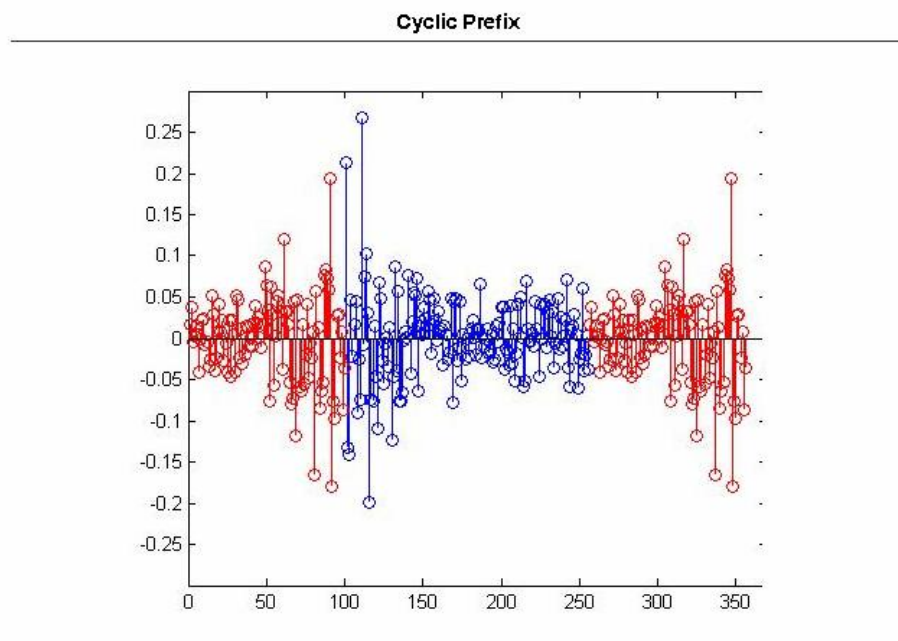

Fig 2.10 This is one t-domain symbol with the cyclic prefix shown in red

impulse response with the concatenated symbols becomes concatenated cyclic convolutions of the impulse response with the individual symbols. Since cyclic convolution directly corresponds to multiplication in the frequency domain, this has great import with respect to equalization, as we will see later. After the time-domain signal passes through the channel, it is broken back into the parallel symbols and the prefix is simply discarded.

Equalization In equalization, the received spectral coefficient blocks (i.e. after cyclic prefix removal and FFT) are adjusted to compensate for the frequency response of the channel (nothing can be done here about the additive noise). Due to the cyclic prefix, each block has essentially undergone cyclic convolution with the channel's impulse response. In the frequency domain, this is the same as if the spectral coefficients were point-wise multiplied by the frequency response of the channel. If the freq. response has no zeros and is known by the receiver, it is possible to perfectly remove the effect of the channel's filter. Since the channel point-wise multiplied the blocks by its freq. response, all that needs to be done is multiply the blocks point-wise by the 1 over the freq. response. Because we implemented the channel's impulse response as non-ideal low-pass, it's freq. response has no zeros and equalization is rather trivial. And we will focus on this in next chapter.
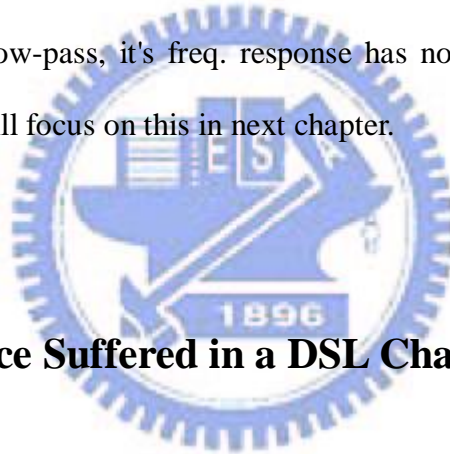
2.3   Interference Suffered in a DSL Channel

The useful signals at the outputs of the FFT at the receiver will suffer from perturbation which can be written as a weighted sum of the QAM symbols transmitted during the previous and next DMT symbol. In addition, because of the loss of orthogonality between the carriers in current symbol, extra ICI is generated. The following types of interference can be distinguished[1]: ICI_1, ICI_2, and ISI. These different interferences are shown in fig 2.11. Assume that a^k_m is the QAM symbol transmitted over carrier k during the mth DMT symbol.
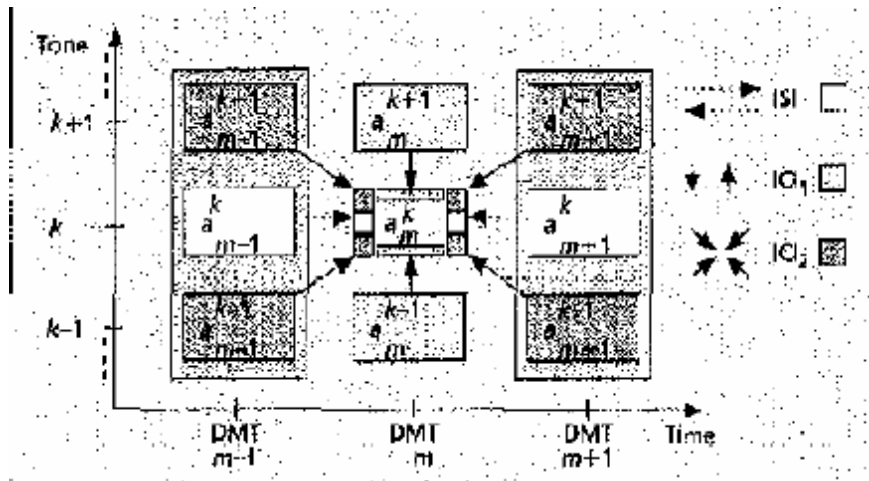
Fig 2.11 Types of interference

ICI₁ is the interference from QAM symbols transmitted over the other carriers of the *m*th symbol ($a^{k-1}{}_{m}$ & $a^{k+1}{}_{m}$).

ICI₂ is the interference caused by the QAM symbols in the previous and next DMT symbols, transmitted over carriers other than the *k*th carrier, over the other carriers of the *m*th symbol. ($a^{k-1}{}_{m-1}$ & $a^{k-1}{}_{m+1}$, $a^{k+1}{}_{m-1}$ & $a^{k+1}{}_{m+1}$)

ISI is the interference from $a^{k}{}_{m-1}$ and $a^{k}{}_{m+1}$, the QAM symbols modulating the kth carrier during the previous and next DMT symbols. Fig 2.12 is a example of the ISI. The linear convolution of the channel impulse response *h* with the time domain input to the channel *x*. Notice the overlap at the intersection of the two symbols.
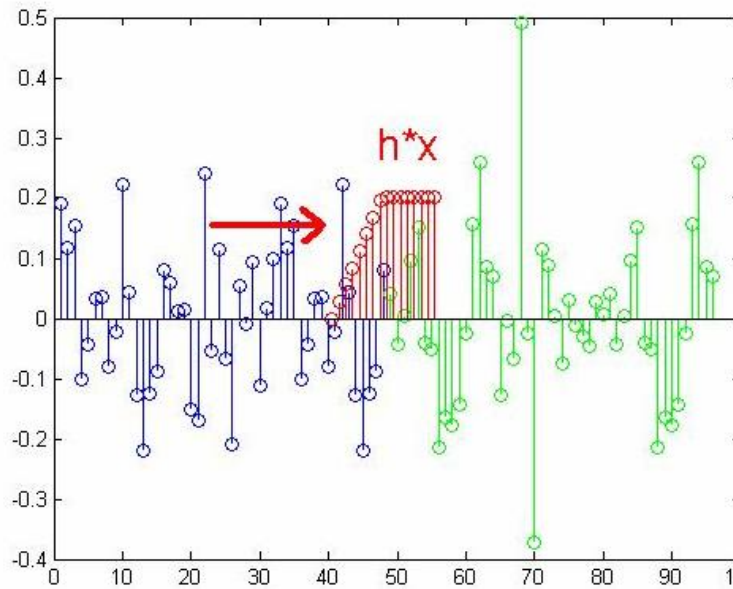
- 19 -

Fig 2.12 Inter-symbol-interference

It was shown that the interference one carrier creates on another could be expressed as a function of the channel impulse response. Two segments of the impulse response contribute to the interference: the *head* and *tail*. These are obtained from the channel impulse response by multiplying the response with a brick wall window, as in Fig 2.13. The zero-valued window length equals the guard time. The symbol synchronizer at the receiver determines the optimal shift of the window function with respect to the impulse response. It is obvious that when the impulse response length is shorter than the guard time, and appropriate symbol synchronization is performed, the tail and head are all zero, indicating that no interference is created. As a consequence, the shape of the head and tail dominates the interference structure.

For non-equalized FDM DMT transmission over the twisted pair, the used carriers that are close to the cutoff frequencies of the upstream/downstream splitting filters dominate by far the interference on any other carrier in the signal. It can be shown that at every carrier, ICI_1 power is approximately the same as ICI_2 power.
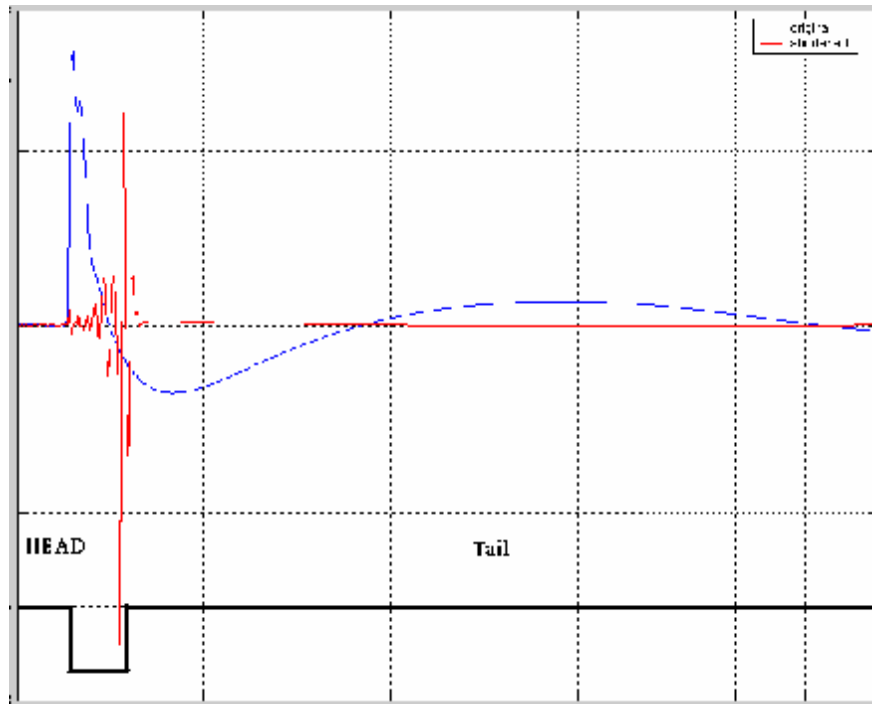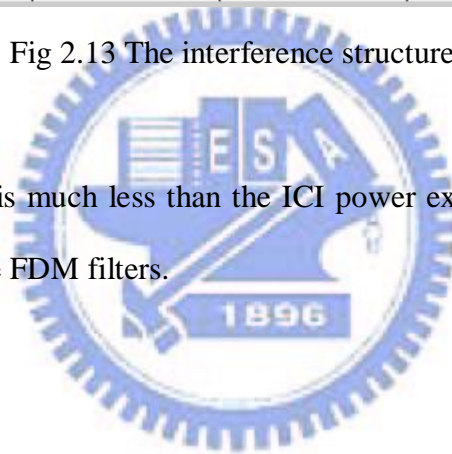
Fig 2.13 The interference structure of ICI

Usually, the ISI power is much less than the ICI power except for tones close to the cutoff frequencies of the FDM filters.

Chapter 3

Time Domain Equalizer

As we know, when the length of channel impulse response is no longer than that of cyclic prefix, the DMT system is ISI free. For channels with long impulse responses such as DSL loops, a time-domain equalizer (TEQ) is typically added at the receiver to shorten the effective impulse response. In this chapter, we first discuss some existing TEQ design methods. Chapter 3.1 focuses on Single-Path design, and chapter 3.2 is Per-tone equalizer design. We do some simulation to compare these two design ways. And the summary is located in chapter 3.3.

3.1   Single-Path Design Method of TEQ

A major advantage of the equalization method based on cyclic prefix insertion and single-tap frequency domain correction is its low computational complexity. The
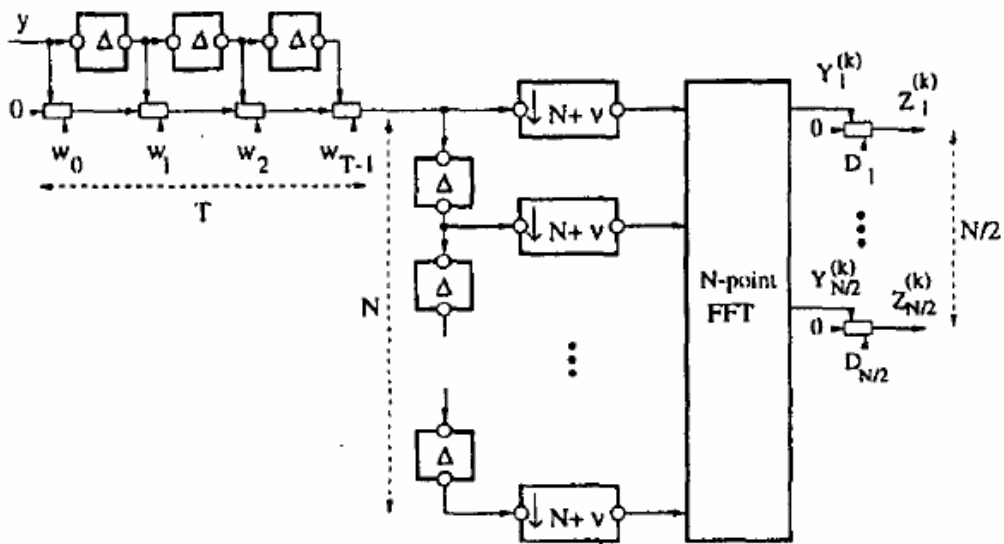
Figure 3.1: A TEQ-based Receiver

price to be paid is a reduction of the data rate cause by insertion of the cyclic prefix.

In time domain equalization schemes (Fig 3.1), a digital filter is inserted in the signal path before the FFT function. Time domain equalizer coefficient initialization for true capacity optimization leads to a highly nonlinear optimization problem[2]. Due to the local minima of the capacity as a function of the equalizer taps, such iterative coefficient optimization is not considered. As a result, a simplified procedure based on time domain channel shortening is usually resorted to. Fig 3.2 depicts a
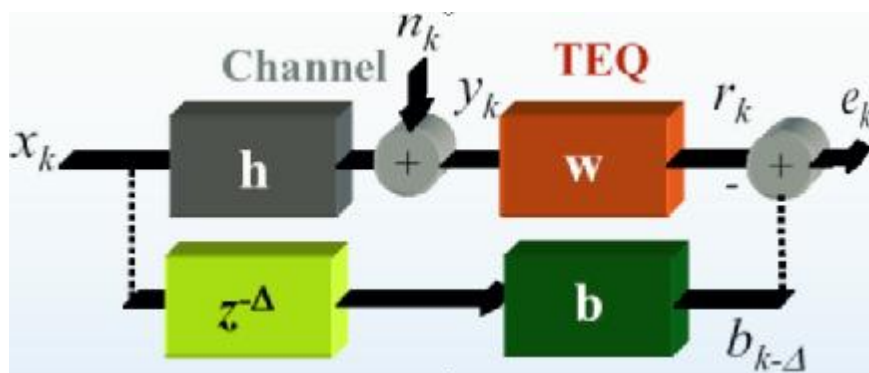

Fig 3.2 A model of TEQ and TIR

block diagram representation of this optimization problem. A TEQ ($\{w_k\}$) is sought for, such that the cascade of the TEQ and channel impulse response CIR ($\{h_k\}$) approximately forms an FIR target impulse response (TIR) ($\{b_k\}$), with an impulse response length shorter than the CP. The relative delay between the equalizer and the TIR is denoted by Δ. The unknown parameter Δ, ($\{w_k\}$), and ($\{b_k\}$) are computed based on several criterion, which will be introduce below.

In the following section, we will discuss three existing TEQs: MSSNR TEQ[11], MMSE TEQ[12], and GEO TEQ[2].

3.1.1 MSSNR TEQ

The channel and the TEQ is shown in Fig. 3.3, where $T(z)$ denotes transfer function of the TEQ, $C(z)$ denotes transfer function of the LTI channel and $v(n)$ represents the channel noise. Let $P(z) = C(z)T(z)$.
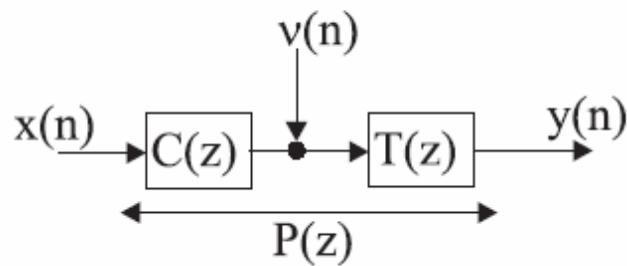

Figure 3.3: Channel, TEQ and effective channel.

Assume $T(z)$ is an order N_t FIR (finite impulse response) filter:

$$T(z) = t_0 + t_1 z^{-1} + t_2 z^{-2} + ... + t_{N_t} z^{-N_t} , \tag{3.1}$$

and the channel $C(z)$ is an order N_c FIR filter:

$$C(z) = c_0 + c_1 z^{-1} + c_2 z^{-2} + ... + c_{N_c} z^{-N_c} \text{ ,} \tag{3.2}$$

where N_c can be large in some applications such as DSLs. One of the purpose of $T(z)$ is to shorten the effective impulse response. Then the effective channel becomes

$$P(z) = T(z)C(z) = p_0 + p_1 z^{-1} + p_2 z^{-2} + ... + p_{N_p} z^{-N_p} \text{ ,} \tag{3.3}$$

where $N_p = N_t + N_c$. Note that thought the order N_p is large, only a small number of consecutive taps p_k will have significant values if $T(z)$ is designed properly. Define

$$\mathbf{t} = (t_0, t_1, \ldots, t_{N_t})$$
$$\mathbf{c} = (c_0, c_1, \ldots, c_{N_c})$$
$$\mathbf{p} = (p_0, p_1, \ldots, p_{N_p}). \tag{3.4}$$

Then the relation between $\mathbf{t}$, $\mathbf{c}$ and $\mathbf{p}$ can be written in a matrix multiplication form:

$$\mathbf{p} = \mathbf{Ct,} \tag{3.5}$$

where $\mathbf{C}$ is an $(N_p + 1) \times (N_t + 1)$ matrix:

$$
\mathbf{C} = \begin{pmatrix}
c_0 & 0 & 0 & \cdots & 0 \\
c_1 & c_0 & 0 & \cdots & 0 \\
\vdots & c_1 & c_0 & \ddots & \vdots \\
c_{N_c} & \vdots & c_1 & \ddots & 0 \\
0 & c_{N_c} & \vdots & \ddots & c_0 \\
0 & 0 & c_{N_c} & \ddots & c_1 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & c_{N_c}
\end{pmatrix}.
$$

(3.6)

In Section 3.1, It is known that the ISI free is obtained only when the effective channel order is no longer than the length of CP. Obviously, it is impossible to shorten the FIR channel with FIR TEQ because the convolution can only increase the order.

Instead, we design the TEQ such that the effective channel $\mathbf{p}$ will have most of its energy within a specific window of length $N_L + 1$, where $N_L \leq L$. Impulse responses that fall outside this window will generate ISI. Define

$$
\mathrm{p}_{\mathrm{signal}} = (0_{1 \times N_w} \ \mathrm{p}_{N_w} \ \mathrm{p}_{N_w+1} \cdots \mathrm{p}_{N_w+N_L} \ 0_{1 \times (N_p - N_w - N_L)})^{\mathrm{T}}
$$

$$
\mathrm{p}_{\mathrm{ISI}} = (\mathrm{p}_0 \ \mathrm{p}_1 \cdots \mathrm{p}_{N_w+1} \ 0_{1 \times (N_L+1)} \ \mathrm{p}_{N_w+N_L+1} \ \mathrm{p}_{N_w+N_L+2} \cdots \mathrm{p}_{N_p})^{\mathrm{T}},
$$

(3.7)

where N_ω is the starting location of the desired window. We have

$$
\mathrm{p}_{\mathrm{ISI}} = \mathrm{p}_{\mathrm{ISI}} + \mathrm{p}_{\mathrm{signal}}
$$

(3.8)

We can rewrite them into matrix multiplications:

$$p_{ISI} = DCt$$

$$p_{signal} = (I_{N_t+1} - D)Ct \, , \tag{3.9}$$

where **D** is the diagonal matrix

$$[\mathbf{D}]_{ij} = \begin{cases} 1, & \text{if } i = j < N_w \text{ or } N_w + N_L < i = j \\ 0, & \text{else} \end{cases} \tag{3.10}$$

for $i, j = 0, 1, \ldots, N_t$. When $N_L = L$, $\mathbf{p}_{signal}$ can be viewed as the part of channel that carries the signal and $\mathbf{p}_{ISI}$ can be considered as the part of channel that creates ISI (See Fig. 3.4). MSSNR TEQ is defined as
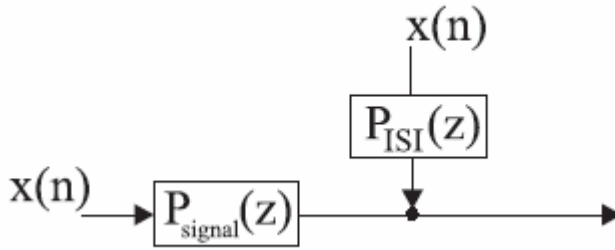

Figure 3.4 ISI as a noise source.

$$t_{MSSNR} = \arg \max_{t} \frac{\| p_{signal} \|^2}{\| p_{ISI} \|^2} \tag{3.11}$$

which maximizes the signal to ISI ratio. Then we have

$$t_{MSSNR} = \arg\min_{t} \frac{\| \mathrm{p}_{ISI} \|^2}{\| \mathrm{p}_{signal} \|^2} = \arg\min_{t} \frac{\| \mathrm{p}_{ISI} \|^2 + \| \mathrm{p}_{signal} \|^2}{\| \mathrm{p}_{signal} \|^2}$$

$$= \arg\max_{t} \frac{\| \mathrm{p}_{signal} \|^2}{\| \mathrm{p}_{ISI} \|^2 + \| \mathrm{p}_{signal} \|^2} = \arg\max_{t} \frac{\| \mathrm{p}_{signal} \|^2}{\| \mathrm{p} \|^2} \qquad (3.12)$$

$$= \arg\max_{t} \frac{t^H C^H (I_{N_t} - D)^H (I_{N_t} - D) C t}{t^H C^H C t}$$

Because $\mathbf{C}^H\mathbf{C}$ is an $(N_t + 1) \times (N_t + 1)$ positive definite matrix, we can decompose it as

$$C^H C = Q \Gamma Q^H, \qquad (3.13)$$

where $\mathbf{Q}$ is an unitary matrix and $\mathbf{\Gamma}$ is a diagonal matrix with positive diagonal entries. Define

$$U = Q \Gamma^{\frac{1}{2}} Q^H \qquad (3.14)$$

Then $\mathbf{U}$ is an $(N_t + 1) \times (N_t + 1)$ square matrix. Note that $\mathbf{U}$ is invertible and $\mathbf{U}^H = \mathbf{U}$. We have

$$C^H C = UU = U^H U \qquad (3.15)$$

Using Eq. 3.12 and Eq. 3.15, we have

$$t_{MSSNR} = \arg\max_{t} \frac{t^H C^H (I_{N_t+1}\text{-}D)^H (I_{N_t+1}\text{-}D)Ct}{t^H U^H Ut}$$

$$= \arg\max_{t} \frac{t^H U^H U^{-H} C^H (I_{N_t+1}\text{-}D)^H (I_{N_t+1}\text{-}D)CU^{-1}Ut}{t^H U^H Ut}$$

$$= U^{-1}\arg\max_{u} \frac{u^H U^{-H} C^H (I_{N_t+1}\text{-}D)^H (I_{N_t+1}\text{-}D)CU^{-1}u}{u^H u} \qquad (3.16)$$

$$= U^{-1}\arg\max_{u} \frac{u^H A u}{u^H u}$$

where $A = U^{-H} C^H (I_{N_t+1}\text{-}D)^H (I_{N_t+1}\text{-}D)CU^{-1}$. Because $\mathbf{A}$ is a positive semi definite matrix, the above optimization problem is well known and the optimal solution u_o is the eigenvector corresponding to the largest eigenvalue of $\mathbf{A}$. So we have

$$t_{MSSNR} = U^{-1}u_0 \qquad (3.17)$$

Example 3.1.1. Fig. 3.5 shows an example. The original channel is a typical DSL channel (CSA Loop 7, see Appendix A) with 500 taps. We see that the number of taps with significant value is more than 100. We use a TEQ with 32 taps (i.e. $N_t = 32$) and then set the desired window length $N_L = 32$. The MSSNR TEQ t_{MSSNR} will produce the effective channel response as shown in the below of Fig 3.5. From the figure, we see that only almost 30 taps have significant values. The SNR of the TEQ is 21.1dB.

The MSSNR TEQ concentrates the energy of the effective channel in a window of length $N_L + 1$. However, if the synchronization occurs, the receiver will get the wrong window starting location N_ω, say $N_\omega + 1$. The error on N_ω makes some of the impulse responses inside the window become ISI. In this case ISI will be very large because the MSSNR TEQ maximizes the energy inside the window. This problem can be solved by choose smaller N_L. For example, by choosing $N_L + 2e = L$,
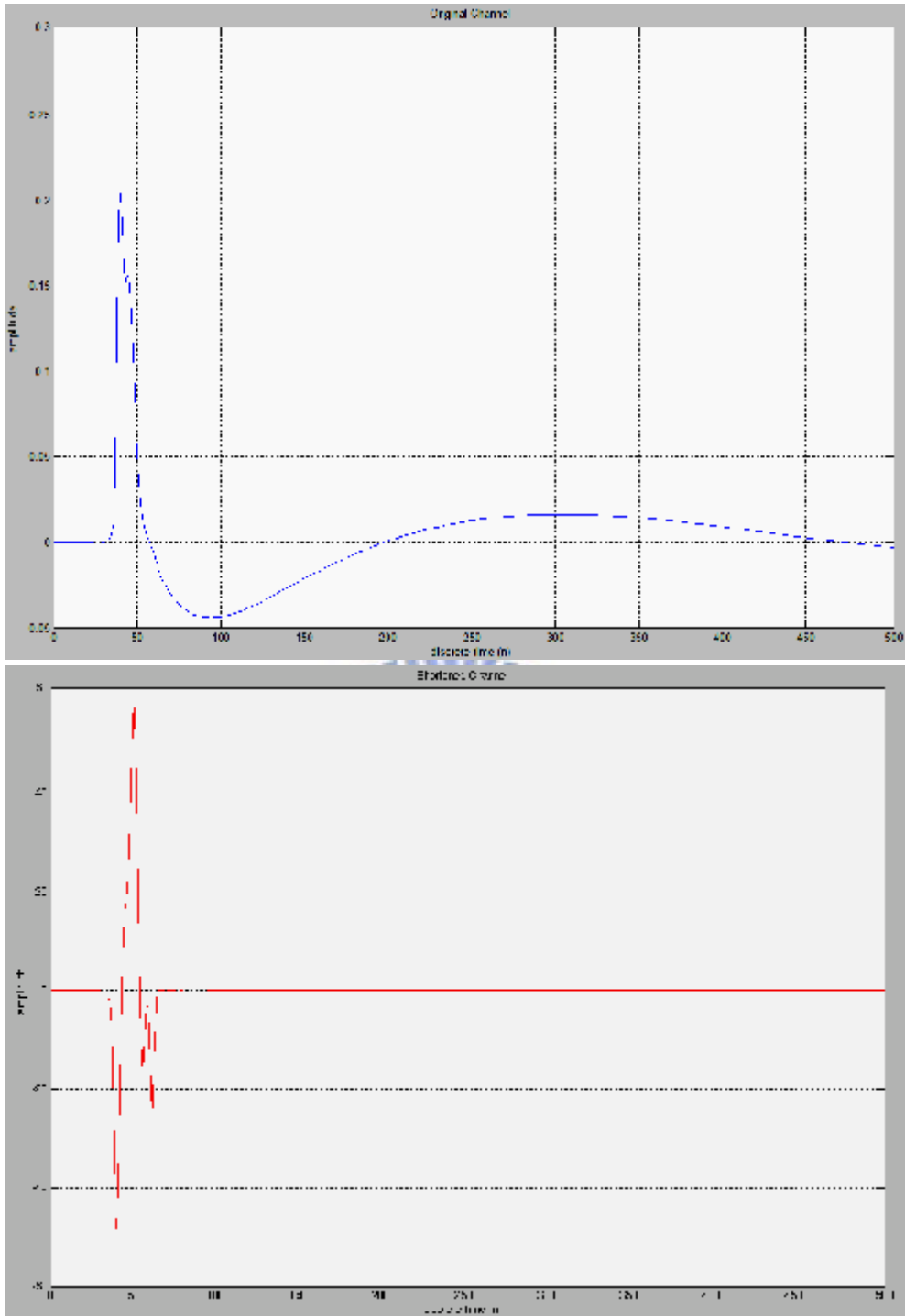
Figure 3.5: Impulse response of the CSA Loop 7 (top) and effective channel equalized

by MSSNR TEQ (bottom).

the system can tolerate synchronization error of e taps.

3.1.2 MMSE TEQ

From previous discussions, we see that the MSSNR TEQ can give a very good result. But the design of such TEQ is very sensitive to the window location N_ω. If it is not chose properly, the resulting SIR will be degraded significantly. Besides, MSSNR TEQ is designed for minimizing ISI. However, the channel noise $v(n)$ is passed through the TEQ (See Fig. 3.3). The total error should consist of ISI and channel
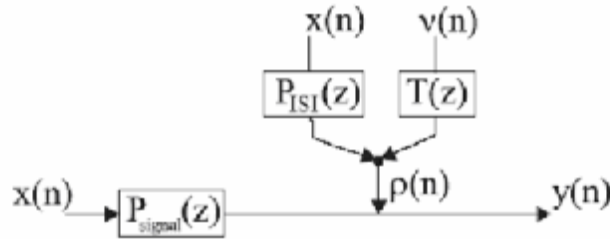
Figure 3.6: Two errors of TEQ.

noise (see Fig. 3.6). The total error of the TEQ output is

$$r(n) = n(n) * t(n) + p_{ISI}(n) * x(n). \qquad (3.18)$$

The MSSNR TEQ may amplify the channel noise. To avoid this, we can consider MMSE TEQ. The MMSE TEQ is defined as

$$t_{MMSE} = \arg\min_{t} \frac{E[\| r(n) \|^2]}{\| p \|^2} \qquad (3.19)$$

The MMSE TEQ is designed for minimizing the mean square error of the TEQ

output, including ISI and channel noise. From Eq. 3.18, we have

$$
\begin{aligned}
E[\| r(n)|^2] &= E[\| \sum_m x(n-m)p_{ISI}(m) + \sum_m n(n-m)t(m) \|^2] \\
&= \sum_m \sum_l p_{ISI}(m)E[x(n-m)x^*(n-l)]p_{ISI}^*(l) \\
&\quad + \sum_m \sum_l t(m)E[n(n-m)n^*(n-l)]t^*(l) \\
&= \sum_m \sum_l p_{ISI}(m)s_x^2 d(m-1)p_{ISI}^*(l) + \sum_m \sum_l t(m)r_n(l-m)t^*(l),
\end{aligned}
\tag{3.20}
$$

where $s_x^2 = E[\| x \|^2]$ is the power of $x(n)$ and $r_v(m) = E[v(n)v^*(n-m)]$ is the autocorrelation of $v(n)$. We have made the approximation that $x(n)$ is a wide sense stationary (WSS) process with a white power spectrum. Rewriting Eq. 3.20 in a matrix representation and using Eq. 3.9, we have

$$
\begin{aligned}
E[\| r(n)\|^2] &= s_x^2 p_{ISI}^H p_{ISI} + t^H R_n t \\
&= s_x^2 t^H C^H D^H DC t + t^H R_n t \\
&= t^H (s_x^2 C^H D^H DC + R_n) t,
\end{aligned}
\tag{3.21}
$$

where

$$
[R_n]_{ij} = E[n(i)n^*(j)]
\tag{3.22}
$$

is the autocorrelation matrix of $v(n)$. Then Eq. 3.19 can be written as

$$
t_{MMSE} = \arg\min_t \frac{t^H (s_x^2 C^H D^H DC + R_n) t}{t^H C^H C t}
\tag{3.23}
$$

Using the same derivation in Section 3.1.1, we have

$$t_{MMSE} = U^{-1} \arg \min_{u} \frac{u^H A_{MMSE} u}{u^H u}, \qquad (3.24)$$

where $A_{MMSE} = U^{-H}(s_x^2 C^H D^H DC + R_n)U^{-1}$ and $\mathbf{U}$ is defined in Eq. 3.15. Then we have

$$t_{MMSE} = U^{-1} u_{N_t}, \qquad (3.25)$$

where $\mathbf{u}_{N_t}$ is an eigenvector of $\mathbf{A}_{MMSE}$ corresponding to its minimum eigenvalue.

Example 3.1.2. Fig. 3.7 shows an example of MMSE TEQ. The original channel is the same DSL channel as in Example 3.3.1. Again, we use a TEQ with 32 taps (i.e. $N_t = 32$) and then set the desired window length $N_L = 32$. In this case, the SNR of the TEQ is 36.7dB.
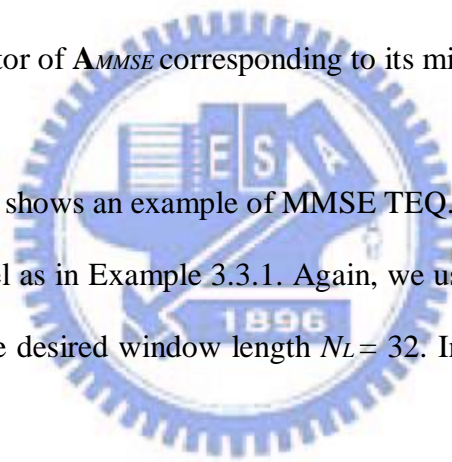
Figure 3.7: Impulse response of the CSA Loop 7 (top) and effective channel equalized

by MMSE TEQ (bottom).

3.1.3 Geometric SNR

Although the MMSE is the most popular equalization criterion since it is easy to analysis and since it lends itself to a simple adaptive implement (the performance surface has a unique global optimum), we shall argue next that it is not the optimum equalization criterion in conjunction with the DMT. In[2], Naofal and Cioffi proposed a new criterion for partially-equalizing severe ISI channels to reduce the cyclic prefix overhead of the DMT.

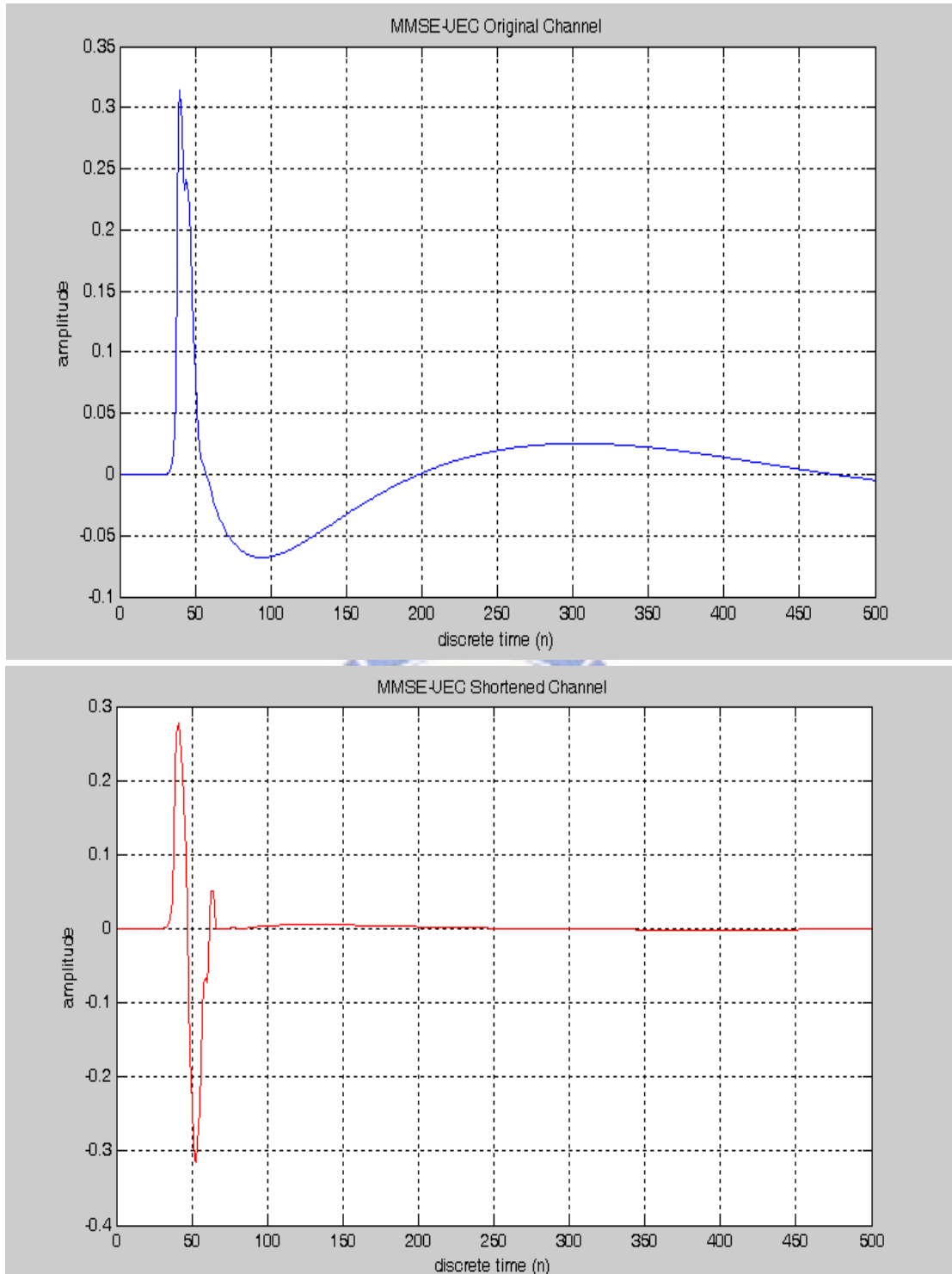Assuming that the subchannels of the DMT are equally spaced(each assumed to be 1/T Hz), independent, and memoryless, then the total number of bits transmitted in one DMT symbol is given by

$$
\begin{aligned}
b_{DMT} &= \sum_{i=1}^{\bar{N}} b_i \\
&= \sum_{i=1}^{\bar{N}} \log_2(1 + \frac{SNR_i}{\Gamma_i})
\end{aligned}
\tag{3.26}
$$

SNR_i is the signal-to-noise ratio of the ith subchannel and is defined by

$SNR_i = \dfrac{S_{x,i}|H_i|^2}{R_{nn,i}}$ where $S_{x,i}$, $R_{nn,i}$, and $|H_i|^2$ are the input energy, noise power spectral density, and channel gain of the ith subchannel. Γ_i is the gap that characterizes the distance (in dB) between SNR_i and the SNR required to achieve capacity, and is a function of the assumed probability of error. In addition, we shall assume a flat input energy distribution across the subchannels, in which case $SNR_i = \dfrac{S_{x,i}|H_i|^2}{R_{nn,i}}$.

Eq 3.26 can be expressed as follows

$$b_{DMT} = \bar{N} \log_2 (1 + \frac{SNR_{geom}}{\Gamma}), \tag{3.27}$$

where the geometric SNR is defined by

$$SNR_{geom} = \Gamma \{ [\prod_{i=1}^{\bar{N}} (1 + \frac{SNR_{geom}}{\Gamma})]^{\frac{1}{\bar{N}}} - 1 \},$$
$$\approx [\prod_{i=1}^{\bar{N}} (SNR_i)]^{\frac{1}{\bar{N}}}, \tag{3.28}$$

The approximation is the input power level and the sampling frequency are chosen such that the number of usable subchannels $\bar{N} = \frac{N}{2}$ (i.e., all the available bandwidth is used). Furthermore, the "1+" and "-1" terms can be typically ignored. This expression makes the name "geometric SNR" obvious.

Using Eq 3.27, we can compute the achievable bit rate of the DMT as follows

$$\begin{aligned} R_{DMT} &= \frac{b_{DMT}}{T} \frac{N}{(N + N_b)} \\ &= \frac{b_{DMT} f_s}{N} \frac{N}{(N + N_b)} \\ &= \frac{f_s}{(N + N_b)} \bar{N} \log_2 (1 + \frac{SNR_{geom}}{\Gamma}), \end{aligned} \tag{3.29}$$

where the f_s is the sampling frequency.

Therefore, assuming a fixed transmission bandwidth, maximizing the achievable bit rate of the DMT is equivalent to maximizing its geometric SNR.

For the equalized DMT, the geometric SNR is approximately given by

$$SNR_{geom} \approx S_x[\prod_{i=1}^{\bar{N}}(\frac{|B_i|^2}{R_{nn,i}|W_i|^2})]^{\frac{1}{N}},$$ (3.30)

where B_i and W_i are the ith FFT coefficients of the TIR and the TEQ, respectively.

Computing the optimum setting of the geometric TEQ is a two-step procedure. First , we need to compute the coefficients of the optimum TIR of a given lenth (N_b+1) that maximizes SNR_{geom}. Second, we compute the coefficients of the length N_f TEQ that results in the minimum mean square error when equalizing the original CIR to b_{opt}.

To simplify the optimization procedure, we shall assume that the input SNR is high enough so that we can ignore the dependence of SNR_{geom} on Γ and use the entire available bandwidth, i.e., $\bar{N} = \dfrac{N}{2}$. In this case, maximizing SNR_{geom} as given by Eq 3.30 is equivalent to maximizing the cost function L(b) defined as follows

$$L(b) = \frac{1}{\bar{N}}\sum_{i=1}^{\bar{N}}\ln|B_i|^2.$$ (3.31)

Now

$$B_i = \sum_{k=0}^{N_b}b_k^*e^{-j\frac{2p}{N}ik}$$
$$= b^*g_i^{(N_b+1)}$$ (3.32)

where $g_i^{(N_b+1)} = [1\ e^{-j\frac{2p}{N}i}\ ...\ e^{-j\frac{2p}{N}iN_b}]^t.$

Therefore, Eq 3.32 becomes

$$L(\mathbf{b}) = \frac{1}{\bar{N}} \sum_{i=1}^{\bar{N}} \ln(\mathbf{b}^* \mathbf{G}_i^{(N_b+1)} \mathbf{b})^2. \qquad (3.33)$$

where

$$\mathbf{G}_i^{(N_b+1)} = \mathbf{g}_i^{(N_b+1)} \mathbf{g}_i^{*(N_b+1)}$$

$$= \begin{bmatrix} 1 & e^{j\frac{2p}{N}i} & \cdots & e^{j\frac{2p}{N}iN_b} \\ e^{-j\frac{2p}{N}i} & 1 & e^{j\frac{2p}{N}i} & \vdots \\ \vdots & \ddots & \ddots & \ddots \\ e^{-j\frac{2p}{N}iN_b} & \cdots & e^{-j\frac{2p}{N}i} & 1 \end{bmatrix} \qquad (3.34)$$

To maximize L(b) as determined by Eq 3.33 while avoiding the impractical solution of an infinite-gain TIR, we need to impose additional constraints on **b**. One such constraint is the unit-energy constraint $\mathbf{b}^* \mathbf{b}$=1. However, with this constraint alone, it can be readily checked that the optimum TIR is memoryless channel, i.e., $|B_i|^2$=1, for all *i*. This requires "full" equalization of the CIR which could result in a large equalization MSE.

Therefore, we shall constrain the MSE of the TEQ to remain below a maximum tolerable value, call it MSE_{max}. This latter MSE condition is equivalent to

$$\mathbf{b}^* \mathbf{R}_\Delta \mathbf{b} \le MSE_{max} , \text{ where} \qquad (3.35)$$

$$R_\Delta = [0_{(N_b+1)\times\Delta} \ I_{N_b+1} \ 0_{(N_b+1)\times s}] \left(\frac{1}{S_x} I_{N_f+g} + H^* R_{nn}^{-1} H \right)^{-1} \begin{bmatrix} 0_{\Delta\times(N_b+1)} \\ I_{N_b+1} \\ 0_{s\times(N_b+1)} \end{bmatrix}.$$

In summary, we want to maximize the Lagrangian subject to the constraint

$$b^* R_\Delta b \le MSE_{\max} .$$

$$L(b) = \frac{1}{\bar{N}} \sum_{i=1}^{\bar{N}} \ln | b^* G_i^{(N_b+1)} b |^2 + (b^* b - 1). \tag{3.36}$$

After some approximation [2], an upper-bound on SNR_{geom} can be obtained by letting the DMT blocklength and the number of equalizer taps becomes infinite. This upper-bound can derived as follows.

$$SNR_{geom} \approx \frac{2}{f_s} \int_0^{\frac{f_s}{2}} (\frac{S_x | H(f) |^2}{R_{nn}(f)}) df : \text{for high input SNR}$$

$$| B_{opt}(f) |^2 = | W_{opt}(f) |^2 | H(f) |^2 (1 + \frac{R_{nn}(f)}{S_x | H(f) |^2})^2. \tag{3.37}$$

Example 3.1.3. Fig. 3.8 shows an example of GEO TEQ. The original channel is the same DSL channel as in Example 3.1.1. In this case, The SNR of the TEQ is 36.6dB.
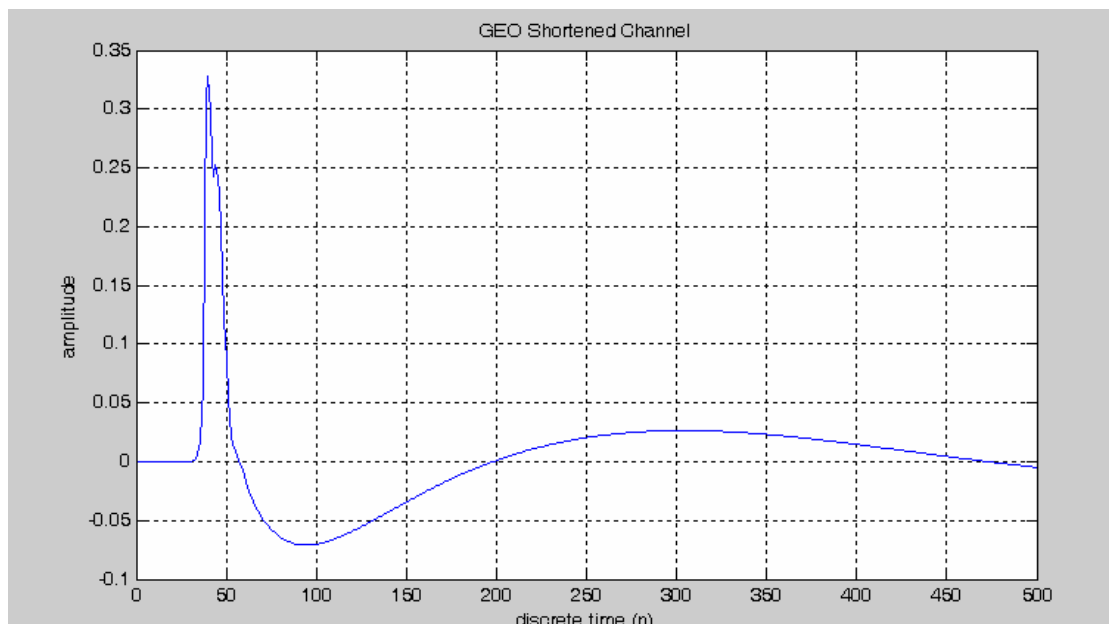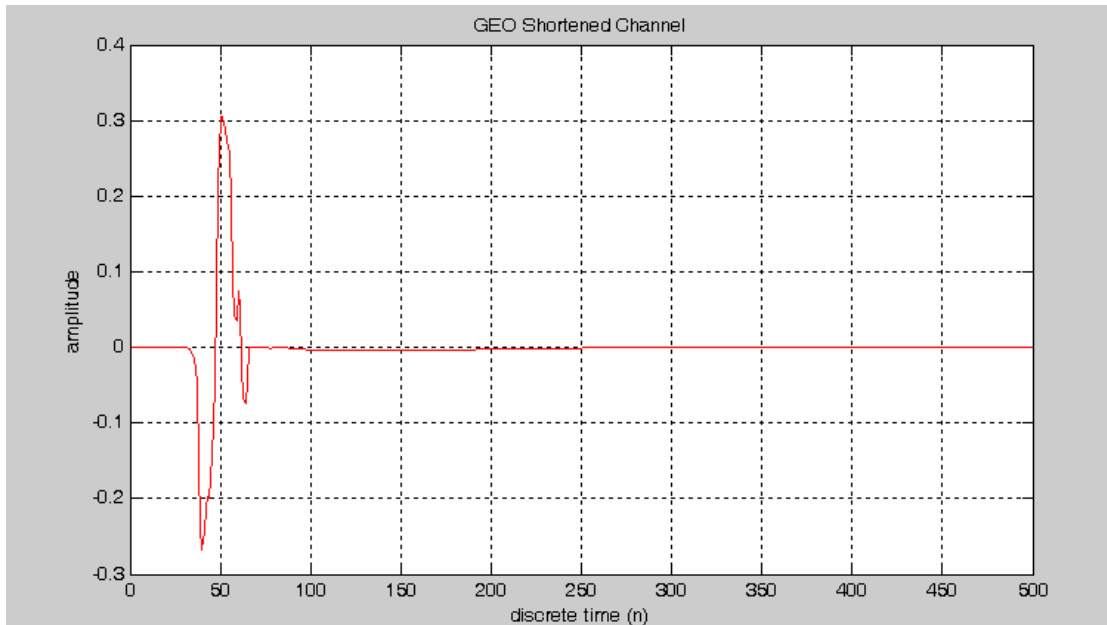
Figure 3.8: Impulse response of the CSA Loop 7 (top) and effective channel equalized

by GEO TEQ (bottom).

3.2   Per-tone TEQ

3.2.1 Introduction

While the single-tone TEQ DMT-receiver structure appears to have become the most popular receiver structure for DSL. Initializing the TEQ with an MSE criterion results in a simple eigenvalue problem. However, minimizing this criterion has two main disadvantages. First, the resulting capacity is not a smooth function of the synchronization delay which hampers optimal synchronization. Second, the resulting capacity is not necessarily the highest achievable capacity of the system. A general disadvantage, independent of the used criterion, is that the TEQ equalizes all tones "in

the same way" and as a result limits performance of the system.

In this section an alternative receiver structure is described based on so-called per-tone equalization, where an MSE optimization is performed for each carrier separately, leading to improved as well as more predictable and reproducible performance.

The structure of the per-tone equalizer is depicted in Fig. 3.9.

Fig 3.9 the Frequency Domain Equalizer

This receiver is based on the following operation:

$$\begin{bmatrix} Z_1^{(k)} \\ \vdots \\ Z_N^{(k)} \end{bmatrix} = \begin{bmatrix} D_1 & 0 & \cdots \\ 0 & \ddots & 0 \\ \vdots & 0 & D_N \end{bmatrix} \cdot \overbrace{F_N}^{1 \, FFT} \cdot (Y \cdot w) \tag{3.38}$$

where

$$Y = \begin{bmatrix} y_{k \cdot s+n+1} & y_{k \cdot s+n} & \cdots & y_{k \cdot s+n-T+2} \\ y_{k \cdot s+n+2} & y_{k \cdot s+n+1} & \cdots 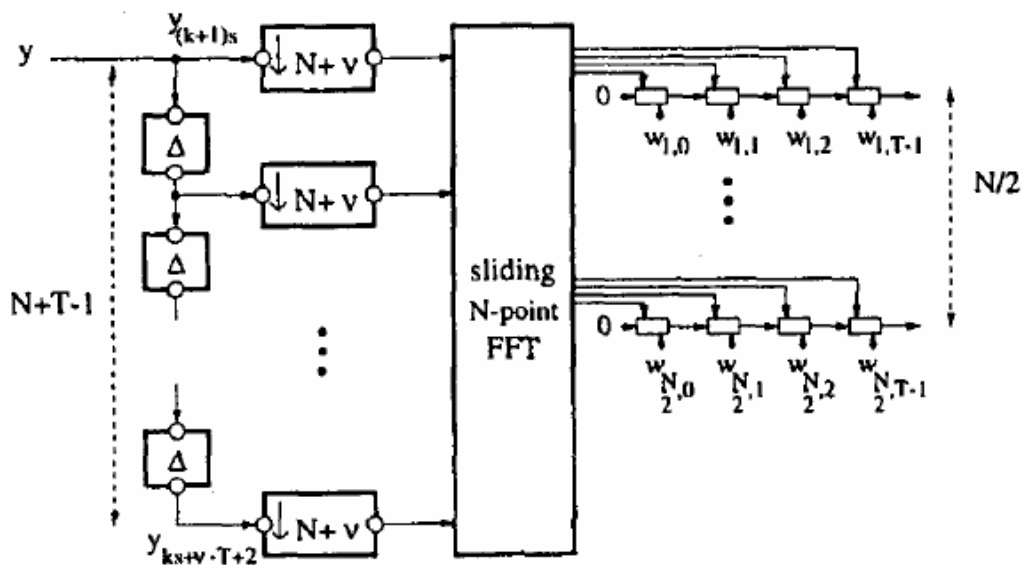& y_{k \cdot s+n-T+3} \\ \vdots & \ddots & \ddots & \vdots \\ y_{(k+1) \cdot s} & y_{(k+1) \cdot s-1} & \cdots & y_{(k+1) \cdot s-T+1} \end{bmatrix} \qquad (3.39)$$

and $w_{T \times 1} = [w_0 \ w_1 \cdots w_{T-1}\ ]^{\mathrm{T}}$ is the (real) T-tap TEQ, F_N an $N \times N$ FFT-matrix, D_i the (complex) 1-tap FEQ for tone i, and Y an $N \times T$ Toeplitz matrix which contains received signal samples. Now, we want to transfer the TEQ-operations to the frequency domain (i.e., after the FFT-demodulation). For a single tone i, we can rewrite the above operations as follows:

$$Z_i^{(k)} = D_i \cdot row_i(F_N) \cdot (Y \cdot w) = row_i \underbrace{(F_N \cdot Y)}_{T \text{ FFTs}} \cdot \underbrace{(w \cdot D_i)}_{w_i}. \qquad (3.40)$$

By putting D_i to the right, one has $w \cdot D_i = (w_i)_{T \times 1}$ which is a (complex) T-tap FEQ for tone i. The next step then is to allow each tone to have its own optimal TEQ, as a T-tap FEQ . The corresponding receiver structure is shown in Fig 3.9. From Eq 3.40, it follows that the latency of the system is the same for the TEQ-approach as for the per tone equalization.

Note that, as indicated in Eq 3.40, instead of one FFT-operation per symbol, we now apparently need T FFT-operations (one FFT for each column of Y). Fortunately, because of the Toeplitz structure of Y, these FFTs can be calculated efficiently by means of a sliding FFT[8]. Only one "full" FFT has to be calculated and the T-1 remaining FFTs can be reduced as follows, as is proven in Appendix A:

$$(F_N \cdot Y(:, m+1)) = (F_N \cdot Y(:, m)) \otimes p + \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{N \times 1} \cdot (y_{k \cdot s + n - (m-1)} - y_{k \cdot s + s + (m-1)}) \tag{3.41}$$

$$p^T = [a^0 \ a^1 \ \cdots \ a^{N-1}], \ a = e^{-j2p(1/N)}$$

Where $m = 1 \cdots T - 1$. The $\otimes$ represents a componentwise multiplication. $Y(:, m+1)$ is the $m+1$th column of Y, and $y_{k \cdot s + n - (m-1)}$ is its first element. $Y(:, m)$ is the mth column of Y, and $y_{k \cdot s + s + (m-1)}$ is its last element. For tone i, all relevant FFT-elements can thus be derived as linear combinations of the ith component of the FFT of the first column of Y and T-1 difference terms $y_{k \cdot s + n - (m-1)} - y_{k \cdot s + s + (m-1)}$ for $m = 1 \cdots T - 1$.

To reduce complexity even further, these linear combinations can then be incorporated in the FEQ coefficients such that the global FEQ for each tone i has as its input the ith output of the FFT and T-1(real) difference terms (see Fig 3.10). The modified per tone equalizers are denoted as and can be calculated from the original FEQs as follows:

$$\begin{bmatrix} v_{i,0} \\ v_{i,1} \\ \vdots \\ v_{i,T-1} \end{bmatrix} = \begin{bmatrix} 1 & a^{i-1} & \cdots & a^{(i-1)(T-1)} \\ 0 & 1 & \ddots & \ddots \\ 0 & \ddots & \ddots & a^{i-1} \\ 0 & \ddots & \cdots & 1 \end{bmatrix} \cdot \begin{bmatrix} w_{i,0} \\ w_{i,1} \\ \vdots \\ w_{i,T-1} \end{bmatrix} \tag{3.42}$$

which is proven in Appendix B. This can be calculated efficiently by the recursion formula

$$v_{i,t} = v_{i,t-1} \cdot a^{i-1} + w_{i,t} \tag{3.43}$$
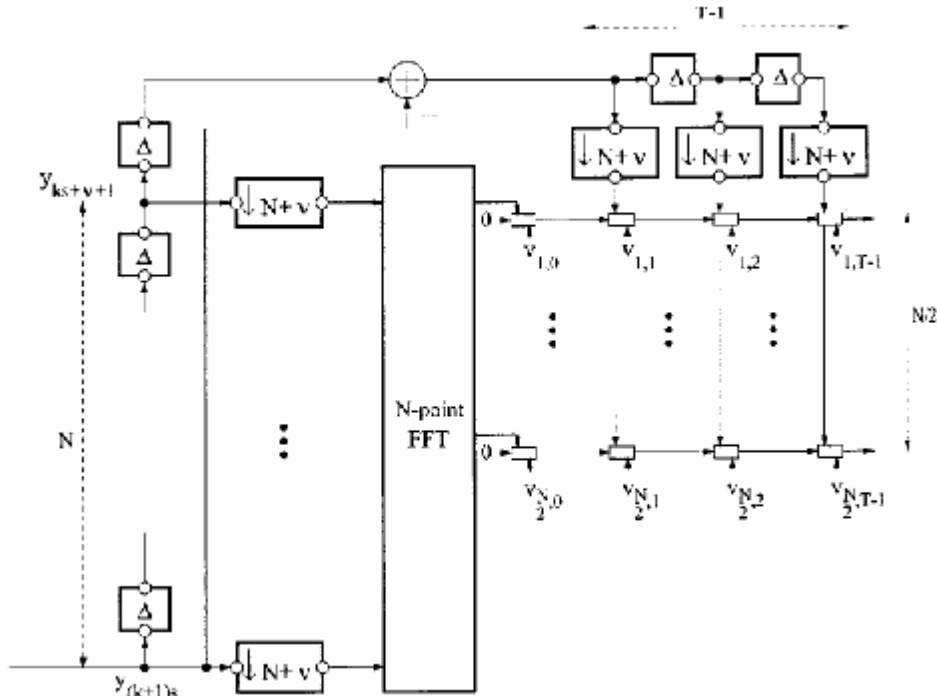
Fig 3.10 Modified T-tap FEQ per tone

for tone $i = 1 \cdots N$ with $t = 0 \cdots T - 2$, $v_{i,T-1} = w_{i,T-1}$, and $a = e^{-j2p(1/N)}$ (see also

Appendix B). These transformations are important when grouping tones, because the

equalizer v_i has two functions, it equalizes the channel for tone i and includes the

sliding Fourier transform corresponding to tone i. This equalizer w_i on the other hand,

can be used as such for other tones of the same group, because this equalizer does not

include the sliding Fourier transform for tone i. As indicated in Fig 3.10, the per tone

equalization structure works with the v_i 's, while the w_i 's are not computed

explicitly (except when tone grouping is used, then the w_i 's are computed for the

center tones).

3.2.2 Compare with Single-path design

In Fig 3.11, all equalizer has 32 taps of CP, 32 taps of equalizer, at loop 7, input

signal is 23 dB, and AWGN power is -140 dBm.
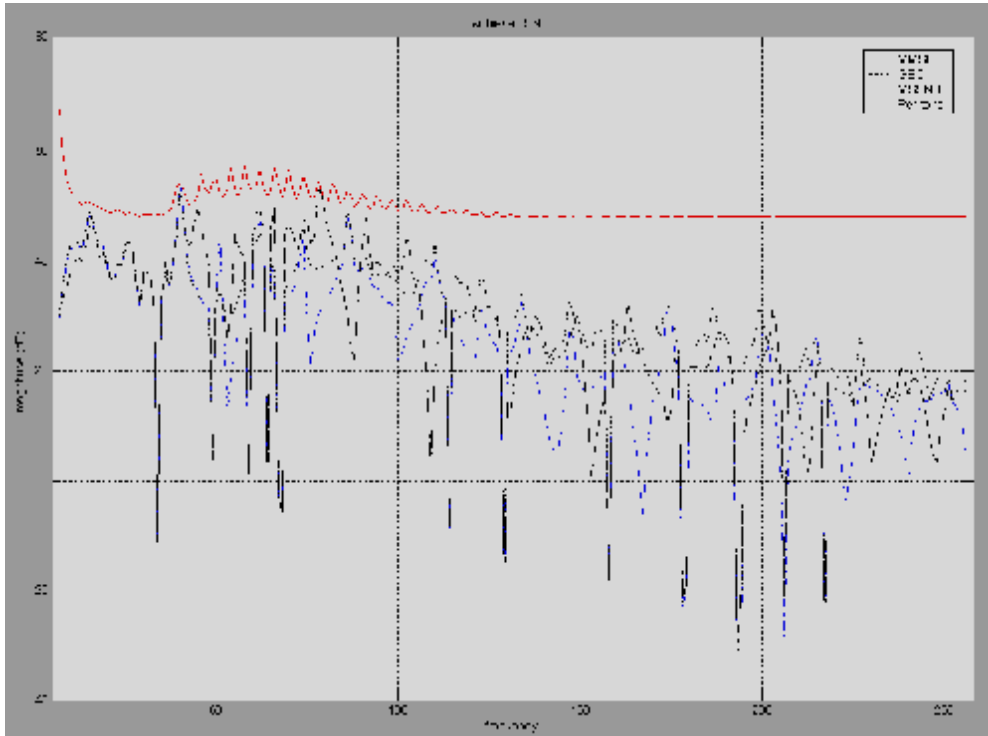
Fig 3.11 Achieved SNR of per tone
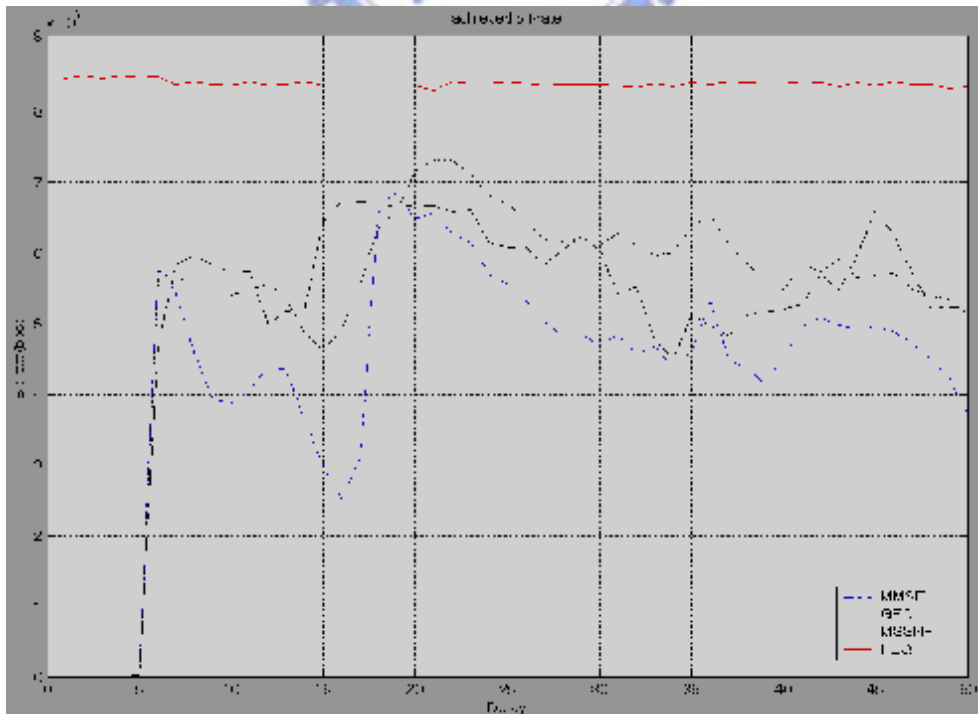
Fig 3.12 Bit-rate after equalization only as a function of synchronization delay

Fig 3.12 plots the bit-rate after equalization only as a function of synchronization delay. As last example, all equalizer has 32 taps of CP, 32 taps of equalizer, at loop 7, input signal is 23 dB, and AWGN power is -140 dBm.

3.3 Summary

As the simulation result in last section, Per-tone equalization has several advantages to the traditional method. It attends higher bit-rate, performances smooth with synchronization delay. Besides, the per-tone equalization approach is found to advantageously combine with other receiver DSP functionalities, such as those for echo cancellation and radio frequency interference (RFI) mitigation [1].
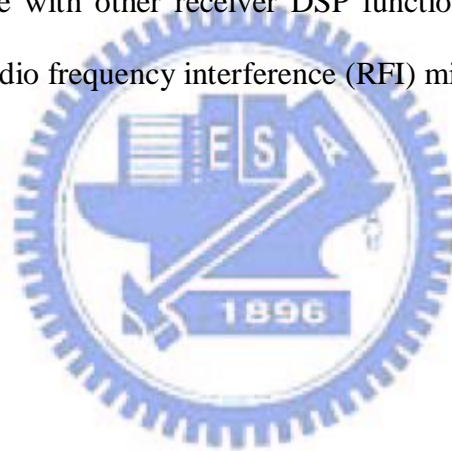
Chapter 4

Modified FEQ Design

It is shown that the per-tone equalization is an alternative way to equalize. Even though in this structure every single tone is given its own distinct and optimal equalizer, the computational requirement is roughly kept at the same level.

However, the per-tone equalization has several drawback, much more requirement of memory and much more computation during initialization competitive to the single-tone way. The number of taps is N/2 times of TEQ. And the initialization complexity also N/2 times more than the traditional way.

4.1 Tone Grouping

To reduce these two problems above, we can combine tones into groups. For each group, only one set of equalizer coefficients (v_{i_1} for tone i_1) is computed. Then the corresponding w_{i_1} is computed from v_{i_1}.

Per group, with center tone , we therefore execute the following steps:

1. Compute v_{i_1}.

2. compute w_{i_1} corresponding to v_{i_1} with the recursion formula

$$v_{i_1,t+1} \cdot a^{i_1-1} + w_{i_1,t} = v_{i_1,t}$$

with t = 0…T-2, $v_{i_1,T-1} = w_{i_1,T-1}$ and $a = e^{\frac{-j2p}{N}}$, which is proven in appendix B.

3. For $i_2 \neq i_1$, belonging to the group with center tone i_1, take $w_{i_2} = w_{i_1}$.

4. compute v_{i_2} from w_{i_2} with the recursion formula

$$v_{i_2,t+1} \cdot a^{i_2-1} + w_{i_2,t} = v_{i_2,t}$$

5. Take $v_{i_2} \leftarrow D_{i_2} \cdot v_{i_2}$, where D_{i_2} is an additional 1-taps FEQ for tone i_2,

   similar to the 1-taps FEQ in the TEQ-based implementation as Eq 3.40.

4.2 Simulation result

Simulations are made for downstream for the standard CSA Loop #1 to CSA

Loop #8. The used tones are from tone 6 up to tone 256. The Symbol size N is 512

and the prefix length is 32.

Per tone equalization is compared with per tone, per 8 tones, per 16 tones, per 32

tones, and per 64 tones in two equalizer length, Nw=16 and 32.

	Per tone	8 tones	16 tones	32 tones	64 tones
Nw=32	8.2303e+006	8.2139e+006	8.0808e+006	7.7831e+006	7.0619e+006
Compare to pertone(%)		99.8	98.2	93.1	85.8
Nw=16	1.6385e+005	1.4873e+005	1.1763e+005	98412	74016
Compare to pertone(%)		90.7	71.7	60.0	45.1

Table 4.1 Bitrate versus tone grouping over CSA loop #1

	Per tone	8 tones	16 tones	32 tones	64 tones
Nw=32	8.1793e+006	8.1683e+006	8.0591e+006	7.8338e+006	7.1338e+006
Compare to pertone(%)		99.3	97.9	93.8	86.7
Nw=16	1.9197e+005	1.7612e+005	1.4662e+005	1.1478e+005	84379
Compare to pertone(%)		91.7	76.4	59.8	43.9

Table 4.2 Bitrate versus tone grouping over CSA loop #2

	Per tone	8 tones	16 tones	32 tones	64 tones
Nw=32	8.293e+006	8.2644e+006	8.2029e+006	7.9177e+006	7.2095e+006
Compare to pertone(%)		99.7	98.9	95.5	87.0
Nw=16	2.1391e+005	1.8701e+005	1.4697e+005	1.0313e+005	96072
Compare to pertone(%)		87.4	68.7	48.2	44.9

Table 4.3 Bitrate versus tone grouping over CSA loop #3

	Per tone	8 tones	16 tones	32 tones	64 tones
Nw=32	8.4069e+006	8.4041e+006	8.348e+006	8.1185e+006	7.3412e+006
Compare to pertone(%)		99.9	99.3	96.5	87.3
Nw=16	1.8879e+005	1.6751e+005	1.3801e+005	97841	88322
Compare to pertone(%)		89.1	73.4	52.0	47.0

Table 4.4 Bitrate versus tone grouping over CSA loop #4

	Per tone	8 tones	16 tones	32 tones	64 tones
Nw=32	8.3081e+006	8.3012e+006	8.2075e+006	7.9133e+006	7.1982e+006
Compare to pertone(%)		99.9	98.8	93.8	86.6
Nw=16	1.6433e+005	1.4717e+005	1.2317e+005	83552	62342
Compare to pertone(%)		90.3	75.6	51.3	38.7

Table 4.5 Bitrate versus tone grouping over CSA loop #5

	Per tone	8 tones	16 tones	32 tones	64 tones
Nw=32	8.3472e+006	8.3233e+006	8.2275e+006	8.0534e+006	7.0351e+006
Compare to pertone(%)		99.7	98.5	96.4	84.3
Nw=16	1.5407e+005	1.3937e+005	1.1626e+005	83816	53166
Compare to pertone(%)		91.1	69.7	55.7	48.5

Table 4.6 Bitrate versus tone grouping over CSA loop #6

	Per tone	8 tones	16 tones	32 tones	64 tones
Nw=32	8.4504e+006	8.4408e+006	8.3295e+006	8.1831e+006	7.6619e+006
Compare to pertone(%)		99.9	98.6	96.8	90.7
Nw=16	1.4217e+005	1.28102e+005	98253	78584	68436
Compare to pertone(%)		90.9	69.7	55.7	48.5

Table 4.7 Bitrate versus tone grouping over CSA loop #7

	Per tone	8 tones	16 tones	32 tones	64 tones
Nw=32	8.4511e+006	8.4334e+006	8.3221e+006	8.1672e+006	7.4523e+006
Compare to pertone(%)		99.8	98.5	96.7	88.2
Nw=16	1.8661e+005	1.6536e+005	1.3245e+005	99135	76104
Compare to pertone(%)		88.9	71.2	53.3	40.9

Table 4.8 Bitrate versus tone grouping over CSA loop #8

4.3 Discussion

When the Nw=32, it is shown that the performance almost not reduces when we grouping 8 tones in a group. Bit-rate is starting to drop when grouping 32 tones. Grouping with 64 tones has a degraded performance. See the table 4.9.

As in Table 4.10, further simulation have shown that when Nw shorter to 16, it is clear that the performance becomes low and the dropping rate becomes more severe. As the grouping tones grows, the bit-rates reduce very fast. But as the Nw=32's

Nw=32	Per tone	8 tones	16 tones	32 tones	64 tones
Avg. BR	8.3333e+006	8.3187e+006	8.2222e+006	7.9213e+006	7.2617e+006
Avg.	1	99.7	98.6	94.9	87

Table 4.9 Utilization for each grouping number when Nw=32

Nw=32	Per tone	8 tones	16 tones	32 tones	64 tones
Avg. BR	1.7571e+005	1.5742e+005	1.2742e+005	94906	75355
Avg.	1	90.2	73.0	54.5	43.0

Table 4.10 Utilization for each grouping number when Nw=16

situation, combining tones into groups of 8 tones does not degrades the performance significantly. They also achieved 73% of maximum when the grouping tones number equals to 16. Compared the simulation result in last section,
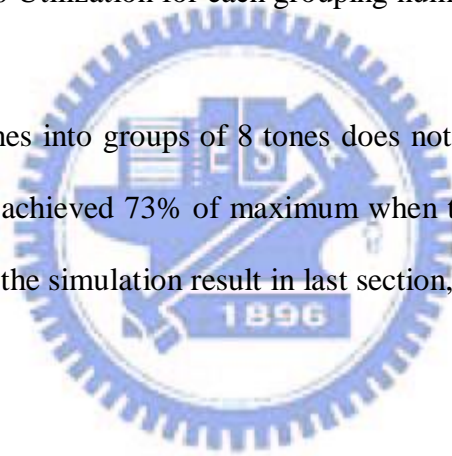
Chapter 5

Conclusions

Per-tone equalizer has several advantages to TEQ. In this thesis, we try to find the grouping number for per-tone equalizer in ADSL system. Combining tones into groups and computing only one optimal per tone equalizer in each group reduces memory and complexity during initialization. Also the cost of the implement will be less by tone grouping.

It does not reduce the capacity substantially if the group size is properly chosen. From the simulation result, it is shown that combines 8~16 tones equalization does not reduce the capacity substantially. It reaches a capacity close to the optimal per-tone capacity.

Finally, some possible further researches regarding per-tone equalization are suggested as follows.

1. Adaptive Per-tone equalization. Since per-tone equalization has advantage to combine Echo canceller or DSP, its design may be used for other system which has more variable channels, such as wireless. Then, the adaptive design will be important.

2. Change the Nw length for each tone to reduce the complexity. The tone with high SNR needs fewer taps. And give more taps to the tones have low SNRs. In this way, we can smooth the SNR of each tones which means a higher bit-rate to transmit data.

Appendix A

Prove Equation 3.41

Eq 3.41 is proven as follows. Given two vectors: $\mathbf{a}=[a_0 a_1 \cdots a_{N-1}]$ and $\mathbf{b}=[b_0 b_1 \cdots b_{N-1}]=[c a_0 a_1 \cdots a_{N-2}]$. The FFT of a is

$$A_{i-1} = \sum_{k=0}^{N-1} a_k e^{-j2p((i-1)k/N)}$$

where   is the tone index. The FFT of b is

$$
\begin{aligned}
B_{i-1} &= \sum_{k=0}^{N-1} b_k e^{-j2p((i-1)k/N)} \\
&= c + \sum_{k=1}^{N-1} a_k e^{-j2p((i-1)k/N)} \\
&= c + \left(\sum_{k=0}^{N-2} a_k e^{-j2p((i-1)k/N)} \right) e^{-j2p((i-1)/N)}
\end{aligned}
$$

with phase **p** defined as in Eq 3.41, one obtains

$$
\begin{aligned}
p(i-1) &= e^{-j2p(i-1)/N} \\
A_{i-1}\, p(i-1) &= \left(\sum_{k=0}^{N-2} a_k e^{-j2p((i-1)k/N)} \right) e^{-j2p(i-1)/N} + a_{N-1}
\end{aligned}
$$

By comparison of above two equations, one obtains.

$$B_{i-1} = A_{i-1}\, p(i-1) - a_{N-1} + c \text{ for } i = 1 \cdots N.$$

Appendix B

Prove Equation 3.42

Eq 3.42 is proven as follows. Without loss of generality, we take T=3 and N=4. Analogous to equation 3.40, we write the demodulated output for the ith tone as

$$Z_i^{(k)} = F_4(i,:) \cdot \begin{bmatrix} a & b & c \\ d & a & b \\ e & d & a \\ f & e & d \end{bmatrix} \cdot w_i$$

where the matrix contains received signal samples and w_i is a 3*1 vector. This expression can be split into a circulant part and a correction on this part, as follows:

$$Z_i^{(k)} = F_4(i,:) \cdot \left(\begin{bmatrix} a & f & e \\ d & a & f \\ e & d & a \\ f & e & d \end{bmatrix} + \begin{bmatrix} 0 & b-f & c-e \\ 0 & 0 & b-f \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right) \cdot w_i$$

Let A_{i-1} be $F_4(i,:)$ times the first column of the circulant matrix. The $F_4(i,:)$ operating on the other columns, which are circular shifts of the first column, gives rise to an additional phase. With $a = e^{-j2p(1/N)}$ (here N=4), the $F_4(i,:)$ operating on the second matrix is written as a "normal" FFT-operation: $F_N(i,:) = [1 | a^{(i-1)\cdot1} \cdots a^{(i-1)(N-1)}]$

$$Z_i^{(k)} = A_{i-1} \cdot [1 | a^{i-1} | a^{(T-1)(i-1)}] \cdot w_i + [0 | (b-f) | (c-e)+(b-f) \cdot a^{i-1}] \cdot w_i$$

One can now write the last expression in an other form by considering and the differences as inputs of the modified FEQ

$$Z_i^{(k)} = A_{i-1} \cdot \underbrace{[1 \mid a^{i-1} \mid a^{(T-1)(i-1)}] \cdot \mathrm{w}_i}_{v_{i,0}} + \underbrace{(b-f)[0 \mid 1 \mid a^{i-1}] \cdot \mathrm{w}_i}_{v_{i,1}} + \underbrace{(c-e)[0 \mid 0 \mid 1] \cdot \mathrm{w}_i}_{v_{i,2}}$$

The modified FEQ, i.e., the FEQ with these new inputs, is then found as

$$\begin{bmatrix} v_{i,0} \\ v_{i,1} \\ v_{i,2} \end{bmatrix} = \begin{bmatrix} 1 & a^{i-1} & a^{(T-1)(i-1)} \\ 0 & 1 & a^{i-1} \\ 0 & 0 & 1 \end{bmatrix} \cdot \mathrm{w}_i$$

or similarly

$$\begin{bmatrix} v_{i,0} & v_{i,1} & v_{i,2} \end{bmatrix} = F_4(i,:) \cdot \begin{bmatrix} w_{i,0} & w_{i,0} & w_{i,0} \\ w_{i,0} & w_{i,0} & 0 \\ w_{i,0} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

From

$$\begin{bmatrix} 1 & a^{i-1} & a^{(T-1)(i-1)} \\ 0 & 1 & a^{i-1} \\ 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & -a^{i-1} & 0 \\ 0 & 1 & -a^{i-1} \\ 0 & 0 & 1 \end{bmatrix}$$

It follows that a simple recursion formula may be obtained, namely

$$v_{i,t+1} \cdot a^{i-1} + w_{i,t} = v_{i,t}$$

For tone $i = 1 \cdots N$ with $t = 0 \cdots T-2$ and $v_{i,T-1} = w_{i,T-1}$.

Bibliography

[1] Marc Moonen, and Luc Vanden dorpe, "Equalization for DMT-Based Broadband Modems," *IEEE Communications Magazine, May. 2000.*

[2] Naofal Al-Dhahir, and John M. Cioffi, "Optimum Finite-Length Equalization for Multicarrier Transceivers," *IEEE trans. On Communications, VOL. 44, NO. 1, Jan 1996..*

[3] Katleen Van Acker, Geert Leus, Marc Moosen, and Thierry Pollet, "Frequency Domain Equalization with Tone Grouping in DMT/ADSL-Receivers," *Signals, Systems, and Computers. Conference Record of the Thirty-Third Asilomar Conference on , Volume: 2 , 24-27 Oct. 1999*

[4] Katleen Van Acker, Geert Leus, Marc Moosen, Olivier van de Wiel, and Thierry Pollet, "Per Tone Equalization for DMT-Based Systems," *IEEE Tran. On Communications, VOL. 49, NO. 1, Jan 2001*

[5] G. Arslan, M. Ding, B. Lu, M. Milosevic, Z. Shen, and B.L. Evans, The University of Texas at Austin. Matlab DMTTEQ Toolbox Version 3.1. [Online].   Available: http://www.ece.utexas.edu/~bevans/ projects/adsl/index.html

[6] R. K. Martin. Matlab code for papers by R. K. Martin. [Online]. Available: http://bard.ece.cornell.edu/matlab/martin/index.html

[7] G. Arslan, B.L. Evans, and S. Kiaei, "Equalization for discrete multitone transceivers to maximize bit rate," *IEEE Tran. On Signal Processing, Dec. 2001.*

[8] B. Fargang-Boroujeny and Y. C. Lim, "A comment on the computational complexity of sliding FFT," IEEE Trans. Circuits Syst., vol. 39, pp. 875-876, DEC. 1992.

[9]   Dennis J. Rauschmayer, "ADSL/VDSL principles: A practical and Precise Study of Asymmetric Digital Subscriber Lines and Very High Speed Digital Subscriber Lines," Macnillan Tech. Series, 1999.

[10] S. Haar, R. Zukunft, T. Magesacher, "Evaluation of Decision Feedback Equalization in VDSL Environment," IEEE ETS*2001*, Sept. 200.

[11] P.J.W. Melsa, R.C. Younce, and C.E. Rohrs, "Impulse Response Shortening for discrete multitone transceivers," *IEEE Trans. Commun., Dec, 1998*

[12] I. Djokovic, "MMSE equalizers for DMT systems with and without crosstalk," 31st Asilomar Conference, pp.545-549, 1997.

[13] B. Farhang-Bouroujeny and M. Ding, "Design methods for time-domain equalizers in DMT transceivers," *IEEE Trans. Commun*., Mar, 2001..

[14] M Van Bladel and M. Moeneclaey, "Time-domain equalization for multicarrier communication," in Proc. Int. Communications Conf. (ICC'96) pp. 1328-1334.