Design and Performance Analysis of VDSL Systems with Per-tone Equalization

Peng Fei Wang

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montreal, Quebec, Canada

November 2006

Your file Votre référence
ISBN: 978-0-494-28931-0
Our file Notre référence
ISBN: 978-0-494-28931-0

ABSTRACT

Design and Performance Analysis of VDSL Systems with Per-tone Equalization

Peng Fei Wang

Digital subscriber line (DSL) is considered as an attractive approach to transmit high-speed data due to its advantage in the provision of affordable broadband services in a wide range of end users without massive capital outlays. Very-high-speed DSL (VDSL) is the latest generation of the xDSL family. It is a promising internet access technique for broadband services, which is capable of supporting high-definition television (HDTV), Voice over IP, and general Internet access over a single twisted pair of wires. In order to overcome the harsh VDSL transmission environment and to realize a reliable high-speed data transmission, equalization technique is required at the receiver to compensate for the channel attenuation and distortion.

In this thesis, we focus on the design and performance study of VDSL systems using the per-tone equalization (PTEQ). Some background material pertaining to VDSL systems and equalization techniques is first presented. The per-tone equalization technique is then investigated. Based on a two-port model of the subscriber loop, namely, the so-called ABCD theory, the transfer function of various VDSL test loops are well formulated and simulated. A practical VDSL system with configurations following the standard as well as the PTEQ in the receiver is designed and simulated. Performance study is conducted over several VDSL test loops with various channel conditions, showing a great improvement on the SNR of the received signal and the overall transmission bit-rate of the VDSL

system as a result of employing the PTEQ. In order to reduce the computational complexity of the training-based PTEQ initialization, a few adaptive schemes based on the NLMS and SR-RLS algorithms are proposed. Various simulation results regarding the convergence speed of the two adaptive algorithms, the number of equalizer taps and different VDSL test loops are presented. A comparison of the performances of the two adaptive initialization schemes is also presented.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to Dr. Wei-Ping Zhu who has been my supervisor for the thesis work. Dr. Zhu has been a great support during my study at the university. I am grateful for his constant guidance and stimulating suggestions through the thesis development. His extremely careful and thorough review as well as the constructive criticism has contributed to the technical quality of the manuscript. I would also like to thank all the professors with whom I have interacted during my studies at Concordia University. Finally, the deepest gratitude goes to my family. It is their love and support that accompany with me throughout all these years.

Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND SYMBOLS

ADC	Analog-to-Digital Converter
ADSL	Asymmetric DSL
AM	Amplitude Modulation
ANSI	American National Standards Institute
AWG	American Wire Gauge
AWGN	Additive White Gaussian Noise
CIR	Channel Impulse Response
CO	Central Office
CP	Cyclic Prefix
DAC	Digital-to-Analog Converter
DFT	Discrete Fourier Transform
DMT	Discrete Multi-tone
DSL	Digital Subscriber Line
EVD	Eigenvalue Decomposition
FDD	Frequency Division Duplexing
FEQ	Frequency Domain Equalization
FEXT	Far-end Crosstalk
FFT	Fast Fourier Transform
FTTCab	Fiber to the Cabinet
FTTEx	Fiber to the Exchange

HAM	Amateur Radio
HDSL	High-speed DSL
HDTV	High-definition Television
ICI	Inter-channel Interference
IDFT	Inverse DFT
ISDN	Integrated Services Digital Network
ISI	Inter-symbol interference
LMS	Least-mean-square
MC	Multi-carrier
MCH	Multi-channel
MMSE	Minimum Mean Square Error
NEXT	Near-end Crosstalk
NLMS	Normalized LMS
ONU	Optical Network Unit
POTS	Plain-old Telephone Service
PSD	Power Spectral Density
PTEQ	Per-tone Equalization
QAM	Quadrature Amplitude Modulation
RADSL	Rate-adaptive DSL
RFI	Radio Frequency Ingress
RLCG	Resistance, Inductance, Capacitance and Conductance
RLS	Recursive Least-squares
SDSL	Symmetric DSL

SHDSL	Symmetric High-speed DSL
SNR	Signal-to-noise Ratio
SR	Square-root
SR-RLS	Square-root RLS
TEQ	Time Domain Equalization
VDSL	Very-high-speed DSL
VoD	Video on Demand

b_n	Number of bits per DMT symbol for tone n
C	Channel capacity
B	Channel bandwidth
f	Frequency (Hz)
$\mathcal{F}_n$	n-th row of $\mathcal{F}_N$
$\mathcal{F}_N$	$N \times N$ DFT matrix
f_s	DMT symbol rate (in Hz)
f_s'	DMT symbol rate (in Hz); $f_s' = (N+v)f_s$
$h(t)$	Channel impulse response (CIR)
L	Channel impulse response length
$\mathbf{h}$	$= [h_0 \cdots h_L]^T$; Discrete-time CIR
$\tilde{h}_n$	Channel frequency response on tone n
k	DMT (block) symbol index
K	Number of training data symbols
$\mathbf{H}_C$	$N \times N$ circulant channel matrix
l	Sample index
n	Tone index
N	(I)DFT size
N_a	Number of active tones
R	Transmission bit rate
M	Number of TEQ taps; Number of PTEQ taps per tone

T_g	Guard period (in seconds)
T_h	Channel delay spread (in seconds)
T_s	DMT symbol time (in seconds); $T_s = \frac{1}{f_s}$
$T_s^{'}$	DMT symbol time (in seconds); $T_s^{'} = \frac{1}{f_s^{'}}$
$\mathbf{v}_n$	Per-tone equalizer tap-weight vector
$\mathbf{Y}_k$	$N \times M$ Toeplitz matrix of the k-th received signal
$\mathbf{w}$	Time-domain equalizer tap-weight vector
x	Transmitter related signals, symbols and samples
y	Receiver related signals, symbols and samples
z	Noise related signals, symbols and samples
$\mathbf{u}_{k,n}$	$= [\Delta \mathbf{y}_k^T \quad \tilde{y}_{k,n}]^T$; PTEQ input vector on tone n
Δ	Synchronization delay parameter
Ψ	The two-port matrix
Γ_n	SNR gap for tone n
λ	Forgetting factor of RLS algorithm
$\mathbf{\Phi}_k$	Correlation matrix of the tap-input signal
$\tilde{\mu}$	NLMS step size
σ_{ab}	Crosscorrelation of a and b
σ_a^2	Autocorrelation of a
v	Cyclic prefix length

$\tilde{x}_{k,n}$	k-th transmitted signal on tone n in frequency domain
$x_{k,n}$	k-th transmitted signal on tone n in time domain
$\mathbf{A}^*$	Complex conjugate of $\mathbf{A}$
$\mathbf{A}^H$	Hermitian of $\mathbf{A}$
$\mathbf{A}^T$	Transpose of $\mathbf{A}$
$\|\mathbf{A}\|$	Two-norm of $\mathbf{A}$
$\varepsilon\{\}$	The mean operation
$a*b$	Linear convolution of a and b

Chapter 1

Introduction

1.1 Need for DSL

Nowadays Internet access plays such an essential role in our daily activities that we need it to work in coordination, acquire useful information, fulfill business and financial services, and support home entertainment, etc. Moreover, the speed requirement of internet access is becoming higher and higher to accommodate various emerging applications, such as interactive multimedia communications, remote access to corporate LANs, digital telephony, video on demand (VoD) and video retrieval or streaming, etc [1]. Broadly speaking, to transmit high-speed data from and to customer premises, four principal media can be used, namely, digital subscriber line (DSL), coaxial cable, optical fiber and wireless. Among the several candidates, DSL is considered as an attractive approach, since it provides a simple, affordable mechanism to give broadband services to a wide range of end users, varying from residences to small/medium-size businesses. In addition, by using copper loops, it enables carriers to offer value-added high-speed networking services without massive capital outlays.

Over the last two decades, the access techniques using DSL have been well developed. Several types of DSL based high-speed access techniques have been proposed and some of them have been widely put into services, such as asymmetric DSL (ADSL). There are four main variations in the xDSL family which are briefly described as follows:

Integrated services digital network (ISDN) DSL (IDSL)—It is expected to transmit data from a user to a destination at 144 kbps using digital transmission. ISDN is capable of delivering two simultaneous connections, in any combination of data, voice, and fax, over a single line. IDSL is superior to ISDN in that IDSL provides always-on connections and transmits data via a data network rather than the traditional voice network.

High-speed DSL (HDSL), symmetric DSL (SDSL) and symmetric high-speed DSL (SHDSL)—They offer the same bit-rate in both upstream and downstream. HDSL requires two phone lines to deliver the basic data rate (1,544 kbps), and it can reach a maximum rate of 2,312 kbps when three lines are used; SDSL outperforms the HDSL by achieving up to 2,312 kbps while requiring one telephone line only; SHDSL further improves HDSL and SDSL, and features symmetrical data rates from 192 kbps to 2,312 kbps with one line and 384 kbps to 4,624 kbps with two lines [47].

ADSL—It is presently the most common form of DSL used in home networking. It has been designed to provide a higher bit-rate up to 8 Mbps for downstream while the bit-rate is limited to 1Mbps for upstream, since many typical uses of the web, for example, file downloading and web browsing, need more bandwidth in downstream than upstream. As a simplified form of ADSL, G.Lite requires less installation at the expense of providing a lower bit-rate (maximum of 1,554 kbps in downstream and that of 384 kbps in upstream). Like G.Lite, rate-adaptive DSL (RADSL) supports a much lower maximum downstream data rate (1,088 kbps) than regular ADSL, but RADSL is more tolerant of errors caused by noise and signal loss through adjusting the upstream bandwidth to create a greater frequency band for the downstream traffic.

Very-high-speed DSL (VDSL)—It is the latest generation of xDSL family. VDSL is

designed for use in a neighborhood where optical fiber connection to the central office (CO) is available and end users are reachable via copper wires spanning about 1.5 km from the fiber end. It is capable of providing data transmission up to a theoretical limit of 52 Mbps for downstream and 12 Mbps for upstream in asymmetric mode, or 26 Mbps in both directions in symmetric mode. Table 1.1 gives comparisons of the four representative DSL techniques in xDSL family regarding bit-rate, bandwidth and the distance covered.

Table 1.1 The xDSL family

	Bit-rate	Bandwidth	Coverage
IDSL	144 kbps	80 kHz	10 km
HDSL	2.312 Mbps	392 kHz	5 km
ADSL	8/1 Mbps	1.104 MHz	5 km
VDSL	52/12 Mbps	12 MHz	2 km

Due to its fast speed, VDSL is a promising technique for broadband services and applications through a single twisted pair of wires, including high-definition television (HDTV), Voice over IP, and general Internet access.

1.2 VDSL Transmission Environment

VDSL has been proposed as a broadband transmission technique primarily for use in hybrid fiber/copper systems connecting optical network units (ONUs) to customer premises. There are two configurations for connection of ONUs to the central office, that is, fiber to the exchange (FTTEx) and fiber to the cabinet (FTTCab). In FTTEx systems, the customer premises are close to the local exchange and can be served directly from the ONUs inside the CO, which is the case suitable for business service. In the second

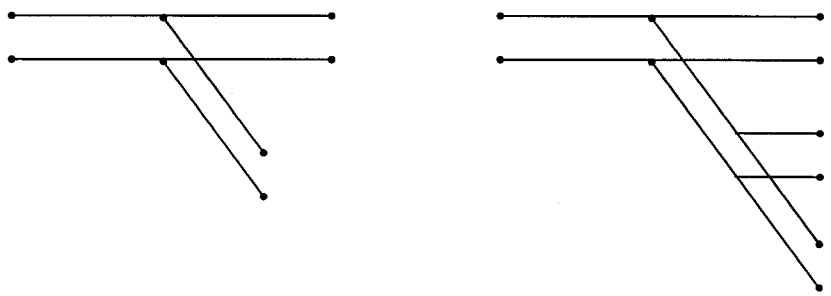
configuration, i.e., FTTCab systems, the ONUs are outside the CO such that residential premises can be served from a street cabinet. The length of the loop from ONU to a customer varies from 300 meters to 1.5 km. With FTTEx and FTTCab configurations, a large VDSL coverage can be guaranteed regardless of the distance between customer premises and the central office.

Subscriber loops are used to connect end users to ONUs, each of which consists of a pair of insulated copper wires of gauges ranging from 26 AWG (American wire gauge) to 19 AWG (approximately 0.4 to 0.91 mm in diameter) [2]. A typical loop plant consists of hundreds to thousands of twisted pairs deployed from ONU. These pairs of wires are then grouped into a maximum of 50 cable binders, each containing 10, 25, or 50 pairs, which are eventually dropped to customer premises.

As the subscriber loop was designed and deployed originally for voice transmission, the transmission of high bit-rate data using the same twisted pairs is a nontrivial task. Below, we give a brief overview of the main impairments that a VDSL system has to deal with.

- **Attenuation**. Signals transmitted over twisted pairs would be attenuated due to the impedance of wires. This attenuation depends on several factors including the type of dielectric used, the form of twisting, and the gauge as well as the length of the wire. It increases with the frequency of the transmitted signal and the length of the wire. Therefore, the attenuation reduces the bit-rate or capacity of the VDSL system, especially when the loop is very long and the transmitted signal has a wide bandwidth.
- **Bridge taps**. Bridge taps are open-circuited twisted pairs that are connected to a transmission pair. They could be left from the previous installation and maintenance

work or they are reserved for future extension of the deployment. Figure 1.1 shows a few typical configurations of bridge taps, where (a) represents a simple bridge tap, and (b) illustrates a more complicated but less common case, namely, a bridge tap exists on another bridge tap. The main impairment caused by bridge taps is the big notches in the transfer function of the transmission line, since the DSL signal is



(a) (b)

Figure 1.1 Bridge taps: (a) a simple tap; (b) a bridge tap on another tap

reflected back from the end of a bridge tap and the reflected signal is mixed with the original signal, which results in very low transmission signal strength at certain frequencies.

- **Crosstalk**. Crosstalk between pairs in the same cable binder is a dominant impairment in DSL systems, and it is caused by electromagnetic radiation of other telephone lines physically located in a close proximity. As shown in Figure 1.2, two types of crosstalk can be distinguished: the near-end crosstalk (NEXT) caused by signals traveling in opposite directions, and the far-end crosstalk (FEXT) caused by signals traveling in the same direction.

As other types of DSL signals, such as ADSL and HDSL, may be transmitted through

the same cable binder, the crosstalk can further be classified into self-crosstalk (S-NEXT, S-FEXT), and alien-crosstalk (A-NEXT, A-FEXT). The self-crosstalk is caused by other VDSL signals, while the alien-crosstalk is due to other types of DSL signals. As the VDSL downstream and upstream transmissions occur in different frequency bands, the self near-end crosstalk (S-NEXT) is normally ignored in VDSL systems.

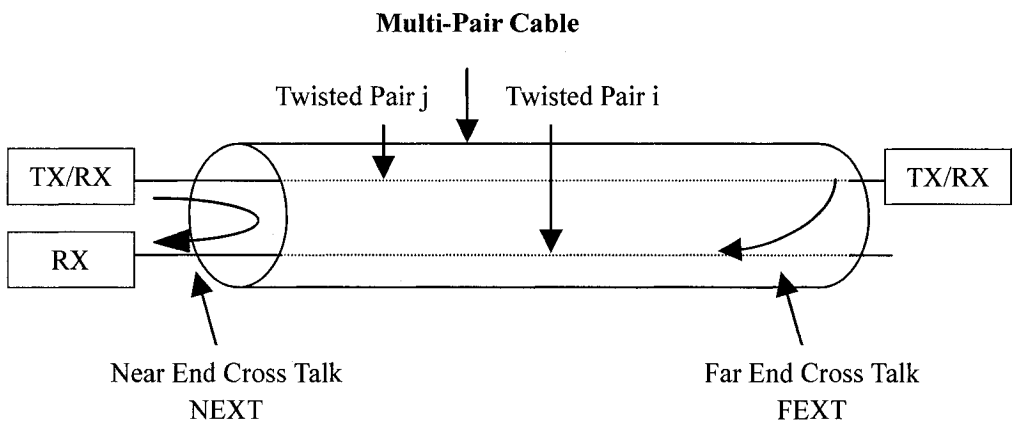


Figure 1.2 NEXT and FEXT in a cable binder

- **Radio frequency ingress (RFI), background noise and impulsive noise**. The RFI noise appears as a continuous narrow band interferer to VDSL signals whose peak envelope power could be as high as -10 dBm. It is caused by amplitude modulation (AM) radio signal or amateur radio (HAM) signal that is transmitted in close proximity of a VDSL transceiver. The background noise is a common additive white Gaussian noise (AWGN) with a fixed PSD level of -140 dBm as defined in [3]. The impulsive noise is a temporary and random interfering signal that is caused by switching of electronic devices such as a refrigerator, a computer, etc.

1.3 Equalization in VDSL

To implement a reliable high-speed data transmission in the harsh VDSL channel environment, equalization techniques are required at the receiver to compensate for the channel attenuation and distortion. More specifically, the equalization of VDSL systems is based on a particular modulation scheme, which is called the discrete multi-tone (DMT) modulation. DMT is an appropriate and powerful transmission scheme adopted to cope with the dispersive transmission characteristics of the VDSL channel. By partitioning the overall communication channel into a set of parallel and orthogonal sub-channels (tones), the equalization of a VDSL transmission system is reduced to a set of smaller and independent equalization problems at different tones. If the number of tones is sufficiently large, each separate tone becomes narrowband and the channel frequency response can be considered flat within the tone bandwidth and can be represented by a simple complex-valued scalar. Under the condition that no inter-symbol interference (ISI) is introduced and each sub-channel is ideally separated in the frequency domain, the equalization problem is then simplified as multiplying a scalar to each tone individually at the receiver. The further details of DMT are given in Section 2.2.

In practical situations, however, the channel impulse response is usually longer than the guard period inserted between symbols, and the sinc-shaped spectra of DMT sub-carriers have big side-lobes (see Section 2.2.1 for detail). As a result, ISI and inter-channel interference (ICI) are introduced inevitably. To mitigate the interferences, advanced equalization strategies have to be adopted sequentially. As summarized in [4], a number of equalization techniques have been proposed for DMT based systems. In these techniques, different criteria have been used to optimize the DSL system performance,

such as shortening the channel impulse response, maximizing the signal-to-noise ratio (SNR), minimizing the ISI, and maximizing the bit-rate, out of which achieving the maximum bit-rate is always the ultimate goal of DSL transmission. Among several equalization methods aimed at maximizing the bit-rate, the per-tone equalization (PTEQ) proposed in [5] outperforms all the others [6, 7, 8, 12], which will be the main research focus of this thesis.

1.4 Objective and organization of the thesis

PTEQ can achieve a large overall channel capacity by means of maximizing the capacity for each tone individually. Simulation results of PTEQ under data model for ADSL have shown its superiority in system performance over other TEQs. It cannot only achieve a higher transmission bit-rate, but also exhibit a better robustness in the presence of RFI or bridge taps. In this thesis, we investigate the PTEQ technique for the VDSL system and carry out detailed simulation study for different types of VDSL loops in various transmission environments. The frequency response of VDSL loops is obtained by modeling the twist pair transmission line, and a practical simulation system is constructed based on the standard T1.424 given by ANSI (American National Standards Institute). The computational complexity of PTEQ during data transmission is presented as well. In order to efficiently initialize the PTEQ tap weights, besides the basic minimum mean square error (MMSE) algorithm, this thesis focuses on the design of adaptive PTEQ algorithms using recursive least-squares (RLS) and the least-mean-square (LMS) criteria. The rest of the thesis is organized as follows.

In Chapter 2, we will provide background materials on the equalization issues of

DMT-based VDSL systems. We will show what makes multi-channel modulation attractive for broadband DSL communication systems, how DMT modulation is related to multi-carrier modulation, and why additional signal processing is required to make DMT modulation suitable for DSL systems. The thought behind the time domain equalization (TEQ) and the frequency domain equalization (FEQ) will also be revealed.

In Chapter 3, based on previous TEQ and FEQ techniques, we will present the concept and the structure of PTEQ. PTEQ is the first DMT equalization technique to guarantee bit-rate maximization and hence outperforms other TEQs. The memory and computational cost of PTEQ will also be discussed in this chapter. A simulation study of PTEQ in VDSL environment is carried out, in which the parameters and processes as required by the VDSL standard will be followed.

Chapter 4 is concentrated on the adaptive initialization of PTEQ. Although the computational cost of PTEQ during data transmission stage is comparable to traditional TEQ methods, it is very heavy in the initialization stage of the equalizer. In order to reduce the computational burden in initialization, LMS and RLS algorithms are implemented in PTEQ. The performance and convergence speed of the PTEQ using different adaptive schemes are studied and compared through computer simulations. Simulation results showing the improved performance of VDSL systems with PTEQ are presented.

Finally in Chapter 5, some concluding remarks on PTEQ in VDSL systems as well as suggestions for future study are provided.

Chapter 2

Background

As mentioned in the previous chapter, in order to implement the desired high bit-rate transmission, a VDSL system has to overcome several impairments due to transmission environments by using proper modulation schemes as well as advanced equalization techniques. In this chapter, we introduce some fundamental concepts behind the discrete multi-tone (DMT) modulation, as well as basic principles of channel equalization applied in VDSL systems.

In Section 2.1, multi-carrier (MC) modulation and bit loading are introduced. The concept of MC modulation is presented in Section 2.1.1. Based on the channel capacity definition in Section 2.1.2, which represents the maximum bit-rate that a channel could transmit, it is argued in Section 2.1.3 that the MC modulation is capable of achieving the channel capacity by means of a special bit loading scheme, namely, water-filling bit loading.

In Section 2.2, DMT modulation is reviewed in detail. A modification is introduced in Section 2.2.1 by adding a cyclic prefix (CP) to the transmitted DMT symbols, which leads to an efficient DMT transceiver. The structure of VDSL transmitter and that of receiver are presented in Section 2.2.2 and Section 2.2.3, respectively.

In Section 2.3, some guidelines and concepts on equalization techniques are given. Equalization is required to recover the performance loss with DMT modulation, such as inter-symbol interference (ISI) and inter-channel interference (ICI). Section 2.3.1 presents

several TEQ techniques based on different optimization criteria, whereas Section 2.3.2 discloses the common mathematical basis shared by almost all TEQs that is based on the maximization of a product of generalized Rayleigh quotients.

2.1 Multi-carrier Modulation and Bit Loading

2.1.1 Multi-carrier modulation concept

To transmit reliable high bit-rate signals in the harsh VDSL environment, one option is to employ a single carrier system, in which the signal sequence is transmitted serially at a certain specified rate, and the receiver will require much complexity to overcome the ISI introduced by time dispersion and non-ideal frequency response of the channel. Alternatively, another modulation scheme, namely, multi-channel (MCH) modulation can be used to reduce the complexity of the equalization in the receiver.

MCH modulation forms a general concept to conceive a reliable communication system by partitioning the transmission channel into N parallel and independent (or orthogonal) sub-channels [13]. It is a natural choice to deal with a harsh transmission environment, the constraints of limited available channel bandwidth, and the transmitting power and receiver complexity. As a special case of MCH modulation, multi-carrier modulation implements the sub-channel orthogonality by dividing the transmission bandwidth into N independent frequency bands, referred to as sub-carriers or tones.

The transceiver structure of an MC system, which realizes this bandwidth partitioning, is depicted in Figure 2.1 [14]. The incoming bit stream is fed to a channel encoder for error-correction purposes, followed by an interleaver. The interleaver reorders the

transmitted bytes to distribute a burst of errors, e.g., errors caused by an impulse noise,

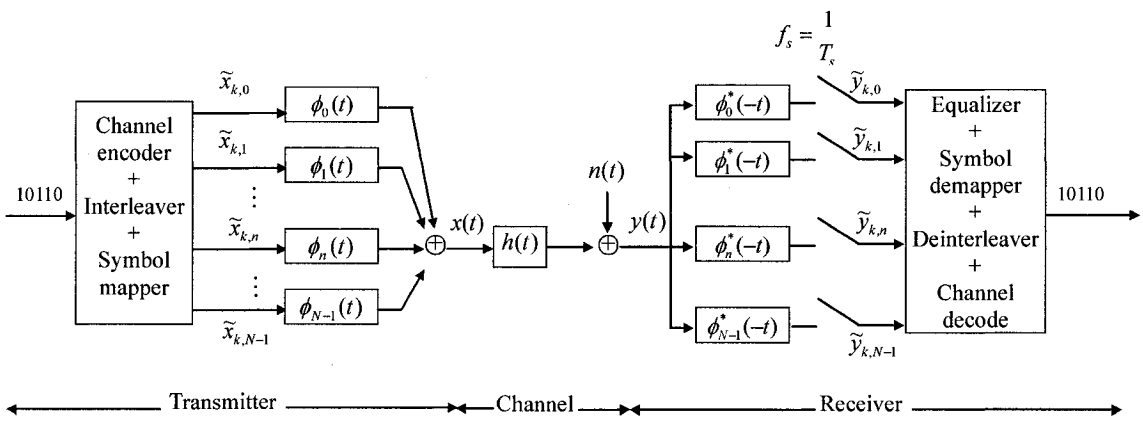


Figure 2.1 Concept of MC transmission [14]

over a longer time interval, thus avoiding the situation where data segments contain more errors than can be corrected [15]. Then, the bit stream is split into N parallel bit streams that are assigned to N tones, indexed from $n=0$ to $n=N-1$. The bits in each tone could be different. It will be explained in Section 2.1.3 how the number of bits, b_n, for each tone n is determined. After splitting, a symbol mapper translates the bits of each tone to a complex-valued transmit (TX) symbol $\tilde{x}_{k,n}$, where k represents the time index and n the tone index, and the tilde "~" above x indicates a frequency-domain version of x. Each transmitted symbol $\tilde{x}_{k,n}$ corresponds to a point in a so-called quadrature amplitude modulation (QAM) constellation [16], whose size (i.e., the number of constellation points M_n) is related to the number of bits b_n by $M_n = 2^{b_n}$.

The modulator bank is fed with a sequence of $N \times 1$ transmitted symbol vectors

$$\tilde{\mathbf{x}}_{k,N} = \left[\tilde{x}_{k,0} \quad \tilde{x}_{k,1} \cdots\cdots \tilde{x}_{k,N-1}\right]^T \tag{2.1}$$

in order to generate the k-th DMT symbol

$$x_k(t) = \sum_{n=0}^{N-1} \tilde{x}_{k,n} \phi_n(t - kT_s) \tag{2.2}$$

which is a superposition of N QAM symbols. Each term in the right-hand side of (2.2) corresponds to the output of one individual orthogonal modulator function $\phi_n(t) = \phi(t)e^{j2\pi f_n t}$ at a symbol rate of $f_s = \frac{1}{T_s}$ Hz. All modulator functions use the same pulse shaping filter $\phi(t)$, and the modulators $e^{j2\pi f_n t}$ produce a frequency shift of $f_n = nf_s$ Hz in frequency domain. Using different pulse shaping filters in time domain would give rise to different forms of bandwidth division in frequency domain. Generally speaking, there are two ways to divide the sub-channels. As shown in Figure 2.2, scheme (a) is to obtain sub-channels by dividing equally the overall frequency band. Note that the subcarrier frequencies are orthogonal even though there is an overlap between two consecutive sub-channels. The scheme (b) is again to divide the whole channel evenly but without leaving overlap between sub-channels. Obviously, the first scheme is more efficient in terms of the use of bandwidth, whereas the second scheme is advantageous in avoiding inter-channel interference.

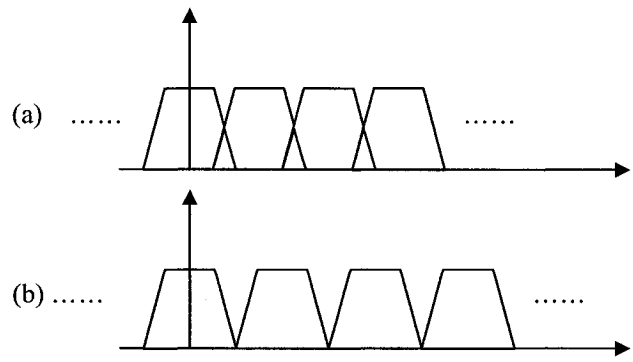


Figure 2.2 Frequency division of subchannels: (a) overlapped (b) non-overlapped

The transmitted signal $x(t)$ is sent over the channel with impulse response $h(t)$ and then corrupted by additive noise $z(t)$, giving the received (RX) signal

$$y(t) = h(t) * x(t) + z(t) \tag{2.3}$$

where $*$ denotes the convolution operation. The demodulation of $y(t)$ can be done by means of a bank of demodulator functions $\phi_n^*(-t)$, which is matched to the transmitter modulator bank. The N outputs of the demodulators are sampled at the symbol rate f_s and fed into the demapper to get the bits represented by a QAM symbol at each tone. Then, the N parallel outputs are serialized to be deinterleaved and decoded eventually.

If the orthogonality of sub-channels is well preserved, no inter-carrier interference would be introduced between the tones. Moreover, for a large number N of tones, the symbol period T_s is larger than the channel delay spread T_h. Hence the tones can be considered narrowband and the cannel frequency response can be assumed flat within a tone period such that the inter-symbol interference can then be ignored. This leads to simple 1-tap equalization for each sub-channel at the receiver which is often called a frequency-domain equalizer (FEQ). The details of FEQ will be presented in Section 2.2.3.

In what follows, we will introduce the definition of channel capacity as a measure for optimal system performance and explain how the MC system is capable of achieving the channel capacity for a given transmitter power budget.

2.1.2 Channel capacity

The capacity C of a communication channel forms the upper bound on the number of

bits that can be transmitted reliably over the channel. According to the noisy channel coding theorem [16] given by Shannon, there exist channel codes (and decoders) that make it possible to achieve reliable communication, with as small an error probability as desired, if the transmission rate R is smaller than the channel capacity C. If $R > C$, it is not possible to make the probability of error tend toward zero with any code. In other words, one can always find a coding method for the error-free transmission of signals as long as the transmission rate is lower than the channel capacity C.

Shannon derived an expression for the capacity of a flat channel in the presence of AWGN in [17]:

$$C = Bb \quad \text{with} \quad b = \log_2(1 + SNR) \tag{2.4}$$

where C is the channel capacity in bits per second (bps), b the maximum number of bits (per transmitted symbol) that can be transmitted reliably, B the channel bandwidth and SNR the signal-to-noise ratio at the receiver. Shannon also suggested partitioning the channel into a large number of narrowband AWGN sub-channels, in order to deal with frequency selective channels. The MC system described in Section 2.1.1 is a possible implementation of this idea. For a given channel with an overall bandwidth of B and the signal-to-noise ratio of SNR_n at the n-th tone in the receiver, the capacity of an MC system over N parallel and independent tones can be obtained by summing the capacities of all individual tones, i.e.,

$$C = \sum_{n=0}^{N-1} Bb_n \text{ with } b_n = \log_2(1 + SNR_n) \tag{2.5}$$

Under the condition of a large number N, SNR_n is given by:

$$SNR_n = \frac{\sigma_{\tilde{x},n}^2 \left|\tilde{h}_n\right|^2}{\sigma_{\tilde{z},n}^2} \tag{2.6}$$

where $\tilde{h}_n$ is the channel frequency response on tone n, $\sigma_{\tilde{x},n}^2 = \varepsilon\{\left|\tilde{x}_{k,n}\right|^2\}$ the transmit power on tone n and $\sigma_{\tilde{z},n}^2 = \varepsilon\{\left|\tilde{z}_{k,n}\right|^2\}$ the power spectral density (PSD) of the noise on tone n.

2.1.3 Water-filling bit loading

In practice, it is not possible to design a modulation scheme with limited implementation complexity that is capable of achieving the channel capacity. In other words, practical modulation schemes are not capable of realizing an error-free data transmission, even when bit-rate R is lower than the channel capacity C. A certain level of bit error and a gap between C and R have to be tolerated, leading to a feasible modification of (2.6) by introducing an SNR gap Γ_n for each tone in order to measure the actual bit-rate of an MC system:

$$C = \sum_{n=0}^{N-1} Bb_n \quad \text{with} \quad b_n = \log_2(1 + \frac{SNR_n}{\Gamma_n}) \tag{2.7}$$

where b_n is the actual number of bits per transmitted symbol that is transferred on tone n. The SNR gap Γ_n (which is often quantified in decibel (dB)) is determined by the target BER, the finite-alphabet symbol mapping (e.g., QAM symbol mapping), the channel coding and a noise margin [18]. The noise margin is a safety factor that accounts for unmodeled noise sources, such as nonlinearities and impulse noise [19].

From (2.6) and (2.7), it is clear that the bit-rate increases with the transmit power $\sigma_{\tilde{x},n}^2$

that is allocated to each tone. However, one cannot increase the power on each tone unlimitedly because of the total transmit power budget in practical systems. The total transmit power budget is given as:

$$\sigma_{\tilde{x}}^2 = \varepsilon\{\tilde{x}_{k,N}^H \tilde{x}_{k,N}\} = \sum_{n=0}^{N-1} \sigma_{\tilde{x},n}^2 \tag{2.8}$$

with $\tilde{x}_{k,N}$ defined in (2.1). Then, we have to face the question as how the transmit power is allocated in each tone so that the bit-rate R is maximized while keeping the total transmit power within the budge. In [20], an answer has been given to this question. The transmit power at tone n is determined from

$$\sigma_{\tilde{x},n}^2 + \frac{\Gamma_n \sigma_{\tilde{z},n}^2}{|\tilde{h}_n|^2} = K \tag{2.9}$$

where K is a constant and is not known a priori. The tones that would require a negative transmit power $\sigma_{\tilde{x},n}^2$ to satisfy (2.9) are turned off. That is, the power on these tones is set to zero, since the SNR at these tones is too low to support a reliable

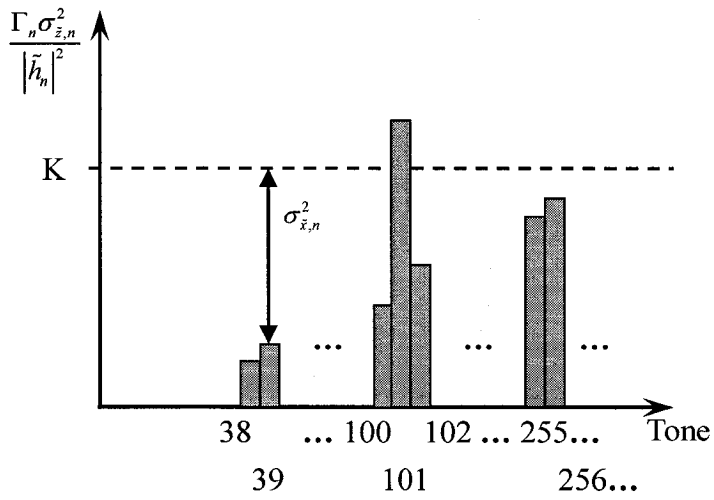


Figure 2.3 The water-filling principle

transmission. Then, K can be derived from the total transmit power budget $\sigma_{\tilde{x}}^2$, the number of tones with a positive transmit power $\sigma_{\tilde{x},n}^2$ and the noise power on each tone. This solution is referred to as the water-filling principle, as shown in Figure 2.3. The name comes from the appearance of allocating the transmit power $\sigma_{\tilde{x},n}^2$, which is like the way of filling the water to a basin to reach a flat surface. When $N \to \infty$, the MC system with transmit power allocation based on the water-filling principle is capable of achieving the channel capacity for a given transmit power budget.

Once the transmit power allocation has been determined, the number of bits b_n on each tone can be computed according to (2.5) directly, and the corresponding constellation size of each tone can then be determined as well. Obviously, the largest number of bits is allocated to the tones with the highest SNR. The number of bits allocated on a single tone in VDSL ranges from 2 to 15, as defined in [3].

2.2 Discrete Multi-tone Transmission

In this section, we will refine the previously introduced multi-carrier modulation to the discrete multi-tone modulation, by taking advantage of the discrete Fourier transform (DFT) and the inverse DFT (IDFT). Cyclic prefix is added to a DMT symbol in order to remove the ISI between DMT symbols as well as to function along with the DFT/IDFT pair for DMT transmission. The structure of DMT transmitter and that of the receiver are also discussed.

2.2.1 Cyclic prefix and DMT

Due to channel delay spread, the first T_h seconds of the k-th transmitted DMT symbol

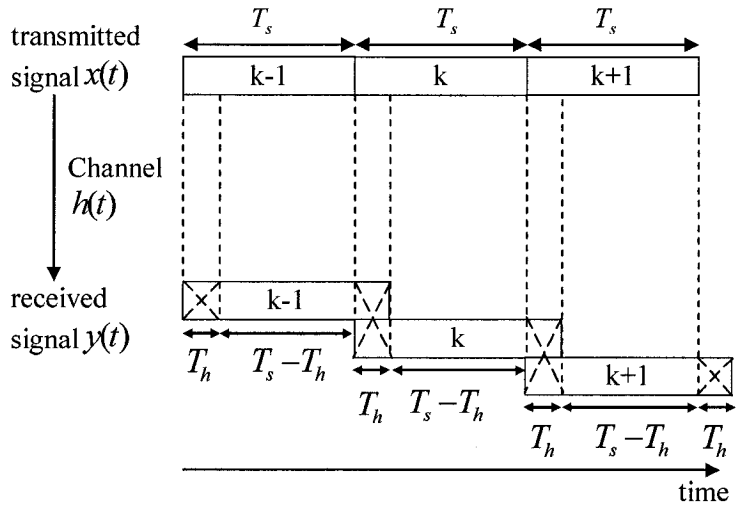


Figure 2.4 ISI between consecutive transmitted symbols

are corrupted by ISI from the $(k-1)$-th symbol, as illustrated in Figure 2.4. When $T_h << T_s$, a so-called guard period $T_g \geq T_h$ can be inserted between the transmitted symbols, such that the receiver may simply ignore the first T_g seconds of the received signal and demodulate the received signal $y(t)$ during the ISI-free time slot $((k-1)T_s + T_g, kT_s)$. The guard period can be implemented by inserting a prefix before each transmitted symbol.

We now derive an equivalent discrete-time model of the transmission system. By sampling the received signal $y(t)$, the transmitted signal $x(t)$, the noise signal $z(t)$ and the channel impulse response (CIR) $h(t)$ at a sufficiently high rate $f_s' = \frac{1}{T_s'}$ and denoting the corresponding samples as $y_l = y(lT_s')$, $x_l = x(lT_s')$, $z_l = z(lT_s')$ and $h_l = h(lT_s')$, respectively, $y(t)$ in (2.3) can be written in discrete form as

$$y_l = \sum_{m=0}^{L} h_m x_{l-m} + z_l \tag{2.10}$$

where the discrete-time CIR $\mathbf{h} = [h_0 \cdots h_L]^T$ has a length of $L+1 = \left\lfloor \frac{T_h}{T_s'} \right\rfloor + 1$ samples and the symbol $\lfloor \ \rfloor$ denotes the floor operation. Since we need N received samples within the ISI-free interval $((k-1)T_s + T_g, kT_s)$ to recover the $N \times 1$ transmitted symbol vector $\tilde{\mathbf{x}}_{\mathbf{k,N}}$ in (2.1), the sampling period T_s' should not be larger than $\frac{T_s - T_g}{N}$ seconds, and the guard period then corresponds to $\nu = \left\lfloor \frac{T_g}{T_s'} \right\rfloor$ samples. Given the notation for a vector that

$$\mathbf{v}_{k,i:j} = [v_{k,i} \ \ v_{k,i+1} \cdots v_{k,j}]^T \tag{2.11}$$

where k is the DMT symbol index, the received and transmitted vectors with prefix can be represented by $\mathbf{y}_{k,-\nu:N-1}$ and $\mathbf{x}_{k,-\nu:N-1}$, respectively. As the first L received samples $\mathbf{y}_{k,-\nu:-\nu+L-1}$ are corrupted by the ISI from $\mathbf{y}_{k-1,N-L:N-1}$, a guard period of ν samples ($\nu \geq L$) must be adopted and $\mathbf{y}_{k,-\nu:-1}$ at the receiver will be ignored. Obviously, the use of guard interval has reduced the bandwidth efficiency, and in turn decreased the bit-rate by a factor of $\frac{N}{N+\nu}$. Consequently, the choice of the length of the guard period ν requires a proper compromise between the bandwidth efficiency and the complexity of equalization.

Assuming that for the k-th DMT symbol a minimum guard period of $\nu = L$ samples is used and the remaining N ISI-free received samples y_l are stacked in vector $\mathbf{y}_{k,0:N-1}$,

(2.10) can be rewritten as the following matrix form:

$$\begin{bmatrix} y_{k,0} \\ \vdots \\ y_{k,N-1} \end{bmatrix} = \begin{bmatrix} h_L & \cdots & h_0 & 0 & \cdots & 0 \\ 0 & h_L & \cdots & h_0 & \ddots & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h_L & \cdots & h_0 \end{bmatrix} \begin{bmatrix} x_{k,-v} \\ \vdots \\ x_{k,N-1} \end{bmatrix} + \begin{bmatrix} z_{k,0} \\ \vdots \\ z_{k,N-1} \end{bmatrix} \tag{2.12}$$

The k-th transmitted signal vector $\mathbf{x}_{k,-v:N-1}$ is the discrete-time equivalent of the continuous-time transmitted signal $x(t)$ in (2.2) in the time-interval $((k-1)T_s, kT_s)$ and it can be written as:

$$\mathbf{x}_{k,-v:N-1} = \sum_{n=0}^{N-1} \mathbf{g}_n \tilde{x}_{k,n} = \mathbf{G}\tilde{\mathbf{x}}_{k,N} \tag{2.13}$$

Comparing (2.13) with (2.2), it is clear that the $(N+v)\times 1$ column vector $\mathbf{g}_n$ is the discrete-time version of the modulator function $\phi_n(t)$. The $(N+v)\times N$ modulation matrix $\mathbf{G} = [\mathbf{g}_0 \cdots \mathbf{g}_{N-1}]$ maps the transmitted symbol vector $\tilde{\mathbf{x}}_{k,N}$ onto the transmitted signal vector $\mathbf{x}_{k,-v:N-1}$. At the receiver, the demodulation of the received signal vector $\mathbf{y}_{k,0:N-1}$ can also be represented by a matrix operation:

$$\tilde{\mathbf{y}}_{k,N} = \mathbf{F}\mathbf{y}_{k,0:N-1} \tag{2.14}$$

where the rows of $\mathbf{F}$ can be equivalently taken as the replacement for the demodulator functions $\phi_n^*(-t)$ of the MC system in Figure 2.1. Thus, we have to find a proper modulator $\mathbf{G}$ and a demodulator $\mathbf{F}$ such that the N parallel and independent sub-channels are created without introducing interferences between them.

An efficient solution has been given in [21], which is obtained by constraining the first and the last v elements of $\mathbf{x}_{k,-v:N-1}$ in (2.12) to be equal, i.e., $\mathbf{x}_{k,-v:-1} = \mathbf{x}_{k,N-v:N-1}$. The

sample vector $\mathbf{x}_{k,-\nu:-1}$ is called a cyclic prefix (CP), as shown in Figure 2.5. The data

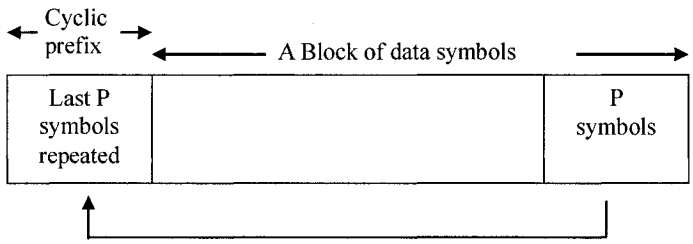


Figure 2.5: Cyclic prefix

model (2.12) can then be rewritten as:

$$\mathbf{y}_{k,0:N-1} = \mathbf{H}_C \mathbf{x}_{k,0:N-1} + \mathbf{z}_{k,0:N-1} \tag{2.15}$$

where the channel matrix $\mathbf{H}_C$ is an $N \times N$ circulant matrix:

$$\mathbf{H}_C = \begin{bmatrix} h_0 & 0 & \cdots & 0 & h_L & \cdots & h_1 \\ \vdots & \ddots & \ddots & & \ddots & \ddots & \vdots \\ h_{L-1} & \cdots & h_0 & 0 & \cdots & 0 & h_L \\ h_L & \cdots & h_1 & h_0 & 0 & \cdots & 0 \\ 0 & h_L & \cdots & h_1 & h_0 & \cdots & 0 \\ \vdots & \ddots & \ddots & & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h_L & \cdots & h_1 & h_0 \end{bmatrix} \tag{2.16}$$

From the property of circulant matrix [22], $\mathbf{H}_C$ can be represented by the discrete Fourier transform (DFT) based eigenvalue decomposition (EVD):

$$\mathbf{H}_C = \mathcal{F}_N^H \mathbf{H}_{D,N} \mathcal{F}_N \tag{2.17}$$

where $\mathcal{F}_N$ is the $N \times N$ DFT matrix, $\mathcal{F}_N^H$ the $N \times N$ inverse DFT (IDFT) matrix, and $\mathbf{H}_{D,N}$ a diagonal matrix with the diagonal elements representing the discrete channel frequency response:

$$\tilde{\mathbf{h}}_N = [\tilde{h}_0 \cdots \tilde{h}_{N-1}]^T = \mathcal{F}_N \begin{bmatrix} \mathbf{h} \\ \mathbf{0}_{(N-L-1)\times 1} \end{bmatrix}. \tag{2.18}$$

Thus, one can replace the modulation matrix $\mathbf{G}$ in (2.13) with the IDFT matrix $\mathcal{F}_N^H$ and the demodulation matrix $\mathbf{F}$ in (2.14) with the DFT matrix $\mathcal{F}_N$, i.e.,

$$\mathbf{x}_{k,0:N-1} = \mathcal{F}_N^H \tilde{\mathbf{x}}_{k,N} \tag{2.19}$$

$$\tilde{\mathbf{y}}_{k,N} = \mathcal{F}_N \mathbf{y}_{k,0:N-1} \tag{2.20}$$

such that the modulation and demodulation processes can effectively decouple the channel into N parallel and independent tones, represented as follows:

$$\tilde{\mathbf{y}}_{k,N} = \mathbf{H}_{D,N} \tilde{\mathbf{x}}_{k,N} + \tilde{\mathbf{z}}_{k,N} \tag{2.21}$$

where $\tilde{\mathbf{z}}_{k,N} = \mathcal{F}_N \mathbf{z}_{k,0:N-1}$. That is, the output signal at the n-th tone can be written as:

$$\tilde{y}_{k,n} = \tilde{h}_n \tilde{x}_{k,n} + \tilde{z}_{k,n} \tag{2.22}$$

By using the fast Fourier transform (FFT) algorithm for the DFT and IDFT, a discrete multi-tone modulation (DMT) scheme can be well constructed.

The DMT modulation is a digital version of the original MC scheme in Section 2.1.1. This is because the n-th row of the modulation IDFT matrix, denoted as $\mathcal{F}_n^H$ where

$$\mathcal{F}_n = \begin{bmatrix} 1 & e^{\frac{j2\pi n}{N}} & \cdots & e^{\frac{j2\pi n(N-1)}{N}} \end{bmatrix}^T \tag{2.23}$$

is equivalent to a bank of modulation functions $\phi_n(t)$ in (2.2) when the shaping filter is a rectangular time-window and in the discrete form of N samples. However, the tones are now overlapped in frequency, since the rectangular time-window has the sinc-shaped spectrum with one main lobe and many side lobes. This is one of the methods to divide the bandwidth as described in 2.1.1 and it is conceptually shown in Figure 2.2 (a).

2.2.2 DMT transmitter

The discussions in the previous subsection form the basis of DMT modulation which has been adopted in VDSL systems. We now describe in detail the structure of the DMT transmitter. That of the receiver will be given in the next subsection.

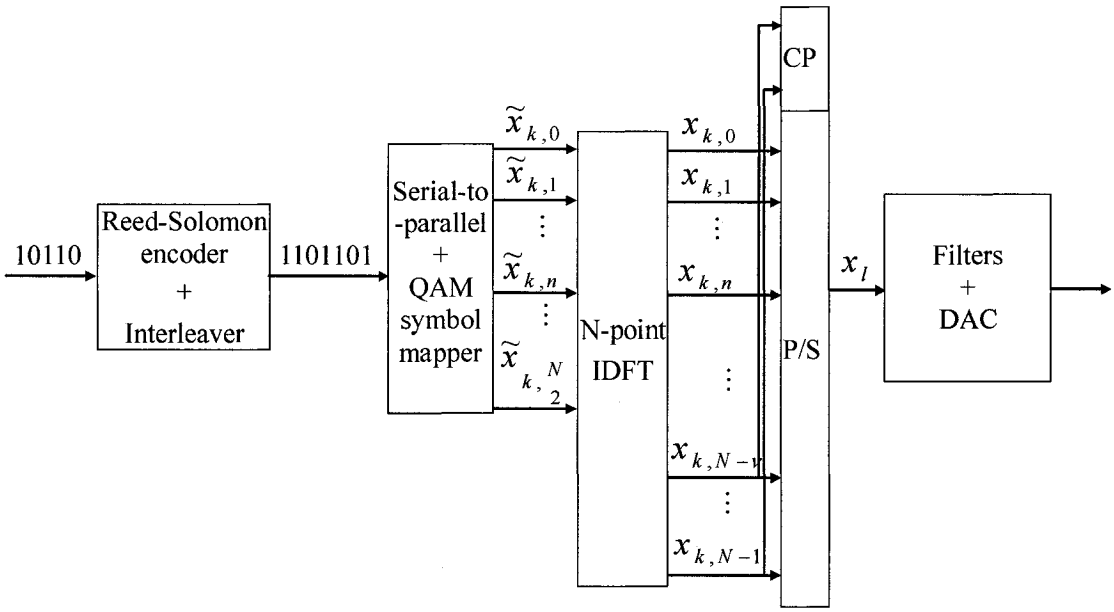


Figure 2.6: The DMT transmitter

The structure of a DMT transmitter is shown in Figure 2.6, which has a similar form as the MC transmitter in Figure 2.1. The incoming bit stream is first encoded (e.g., using a Reed-Solomon channel encoder [23] in VDSL), interleaved and split into parallel bit streams. The number of bits in each parallel stream may be different, which can be decided during the channel estimation and bit loading stage using the algorithm described in Subsection 2.2.3. The bit streams are then mapped to QAM symbols $\tilde{x}_{k,n}$. At each DMT symbol instant k, a block of QAM symbols, denoted as an $N\times 1$ transmitted

symbol vector $\tilde{\mathbf{x}}_{k,N}$, is fed to the modulation IDFT at a DMT symbol rate $f_s = \frac{1}{T_s}$. The IDFT output at DMT symbol time k is given by $x_{k,n}$. We need to mention that, since DSL systems use baseband transmission, the transmitted samples $x_{k,n}$ should be real. This is accomplished by imposing complex conjugate symmetry to $\tilde{\mathbf{x}}_{k,N}$, namely,

$$\begin{bmatrix} \tilde{x}_{k,1} & \cdots & \tilde{x}_{k,\frac{N}{2}-1} \end{bmatrix}^T = \begin{bmatrix} \tilde{x}^*_{k,N-1} & \cdots & \tilde{x}^*_{k,\frac{N}{2}+1} \end{bmatrix}^T \tag{2.24}$$

Note that N denotes the DFT/IDFT size, and $\frac{N}{2}$ is the number of actual parallel streams (i.e., tones) that can be used. Finally, a CP is added to obtain $\mathbf{x}_{k,-v:N-1}$, and the $N+v$ samples are then fed serially to the front-end of transmitter filter at a sample rate equal to $f_s' = (N+v)f_s$. The discrete-time sample x_l is transformed into an analog signal via a digital-to-analog converter (DAC). The front-end filters include low-pass and high-pass filters to separate the plain-old telephone service (POTS) signal and the DSL upstream and downstream transmissions.

2.2.3 DMT receiver

Figure 2.7 depicts a DMT receiver that performs the reverse operation of the transmitter. The received signal $y(t)$ is input to the front-end of a receive filter and then sampled by an analog-to-digital converter (ADC). The TEQ block will be explained in the next section. The samples of the received signal, y_l, are stacked as the $(N+v)\times 1$ received vector $\mathbf{y}_{k,-v:N-1}$, where k is the DMT symbol index. The first v samples of $\mathbf{y}_{k,-v:N-1}$,

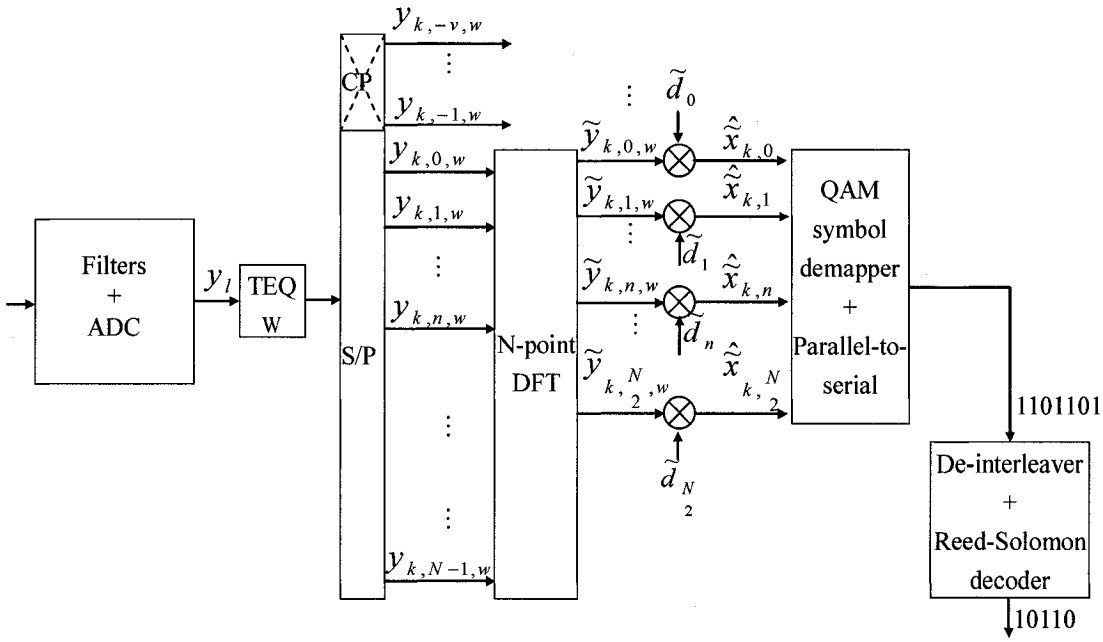


Figure 2.7: The DMT receiver

i.e., $\mathbf{y}_{k,-v:-1}$, correspond to the guard period to absorb the ISI between DMT symbols, and thus are discarded. The remaining samples are split into N parallel streams, and demodulated by means of a DFT to deliver $\tilde{\mathbf{y}}_{k,N}$. Since the last $\frac{N}{2}$ output samples, namely, $\tilde{\mathbf{y}}_{k,\frac{N}{2}:N-1}$, are complex conjugate of the first $\frac{N}{2}$ samples, as stated in Section 2.2.2. Therefore, only $\tilde{\mathbf{y}}_{k,0:\frac{N}{2}-1}$ needs to be further processed. In addition, the DMT receiver equalizes the received signal to compensate for the channel attenuation. Since we have assumed that the signals are transmitted through N parallel and independent tones, the equalization is straightforward. That is to say, each tone just needs to be scaled by the inverse of the discrete channel frequency response, i.e., $\tilde{d}_n = \tilde{h}_n^{-1}$. The estimate of the transmitted symbol $\tilde{x}_{k,n}$ is then given by

$$\hat{\tilde{x}}_{k,n} = \tilde{d}_n \tilde{y}_{k,n} = \tilde{h}_n^{-1}(\tilde{h}_n \tilde{x}_{k,n} + \tilde{z}_{k,n}) = \tilde{x}_{k,n} + \tilde{h}_n^{-1}\tilde{z}_{k,n} \quad (2.25)$$

The one-tap equalization per tone is called the frequency-domain equalizer (FEQ). Finally, the output of the FEQ is de-interleaved and decoded to recover the original information bit stream.

2.3 Equalization for DMT-based Systems

2.3.1 Time domain equalization

In Section 2.2.1, it has been stated that the CP length v should be larger than or equal to the channel dispersion delay L (the length of the channel impulse response) in order to avoid ISI and ICI. In practice, however, the CP length is usually not sufficiently large considering the bandwidth efficiency and the channel CIR characteristics. As a consequence, the received tones at the receiver are not orthogonal and there is a significant amount of ISI/ICI at the input of the decision devices.

Therefore, additional equalization is required in order to render the DMT modulation suitable for VDSL systems. It has been suggested in [19] that a digital filter $\mathbf{w}$ could be inserted before DFT demodulation, as shown in Figure 2.7, which is called a time-domain equalizer (TEQ). It acts as a channel shortener such that the convolution of the M-tap TEQ filter $\mathbf{w}$ and the CIR $\mathbf{h}$ of length L, $\mathbf{c} = \mathbf{h} * \mathbf{w}$, has an actual length smaller than or equal to the CP length v. Then the orthogonality between the tones is restored and the ISI/ICI vanishes. In theory, however, the convolution $\mathbf{c} = \mathbf{h} * \mathbf{w}$ has a length of $L+M-1$, which is normally larger than v. In order to shorten the channel impulse

response, one can design the TEQ **w** such that most of the energy of the overall impulse response is concentrated on the first v samples. We can denote the output of the TEQ **w** as:

$$\mathbf{y}_{k,-v,N-1,w} = \left[y_{k,-v,w} \cdots y_{k,N-1,w} \right]^T \tag{2.26}$$

where the subscript w denotes signals have been filtered by the TEQ **w.** After discarding the CP $\mathbf{y}_{k,-v,-1,w}$, the n-th equalized DFT output is given by

$$\tilde{y}_{k,n,w} = \breve{c}_n \tilde{x}_{k,n} + \tilde{z}_{k,n,w} + \mathrm{ISI}_{k,n,w} + \mathrm{ICI}_{k,n,w} \tag{2.27}$$

where $\breve{c}_n$ is the scalar of the overall frequency response on tone n and the residual terms $\mathrm{ISI}_{k,n,w}$ and $\mathrm{ICI}_{k,n,w}$ depend on the equalizer design. The SNR_n in (2.6) can then be modified as

$$SNR_n = \frac{\sigma_{\tilde{x},n}^2 \left| \breve{c}_n \right|^2}{\sigma_{\tilde{z},n,w}^2 + \varepsilon\{\left| ISI_{k,n,w} \right|^2\} + \varepsilon\{\left| ICI_{k,n,w} \right|^2\}} \tag{2.28}$$

The design of TEQ is crucial and non-trivial. It has a considerable impact on all terms in (2.28). Moreover, the equalization is performed before the DFT module, while the SNR_n (2.28) is determined after the DFT. Hence, the TEQ should act in such a way that it reduces jointly the ISI/ICI as well as the noises on all tones. The bad spectral containment of the sinc-shaped tone spectrum complicates the TEQ task, as it causes leakage of the desired signal, the residual ISI/ICI, and the noise to neighboring tones.

2.3.2 Common framework for TEQ design

Optimal TEQ design has turned out to be difficult and costly. Different TEQ design criteria have been proposed for the design of TEQs, yielding a number of TEQ methods.

Resorting to simplified procedures that are primarily based on time domain channel shortening, the TEQ has been designed so that the convolution of the channel CIR and the impulse response of the TEQ produces an overall impulse response with almost all of its energy concentrated within a window. Two out of the most popular and earliest channel-shortening TEQ design methods are the minimum mean-square-error (MMSE) TEQ and the maximum shortening-SNR (MSSNR) TEQ. In the DSL context, however, the maximization of bit-rate in a quasi-stationary environment is targeted. The above-mentioned two designs are based on a time domain criterion that aims at low computational complexity, and therefore, do not provide a maximum bit-rate.

Some TEQ design methods aiming at the bit-rate maximization have been presented, including the maximum geometric SNR (MGSNR) TEQ design proposed by Al-Dhahir [24], the maximum bit-rate (MBR) TEQ design given by Arslan [25] and the maximum data rate (MDR) TEQ given by Milosevic [26]. These methods use some approximations and assumptions in their adopted SNR models and therefore, they are suboptimum without achieving a true maximum bit-rate.

All the TEQ design methods mentioned above fall under a common TEQ design framework [4] that can be represented as a unified mathematical notation for different equalizer designs, i.e., the maximization problem of a single generalized Rayleigh quotient or that of a product of several generalized Rayleigh quotients

$$\max_{\mathbf{w}} \prod_{n=0}^{N_a-1} \frac{\mathbf{w}^T B_n \mathbf{w}}{\mathbf{w}^T A_n \mathbf{w}} \tag{2.29}$$

where $\mathbf{w}$ is typically (but not always) the weight vector of the M-tap TEQ, N_a typically the number of active tones, and the matrices $\mathbf{A}_n$ and $\mathbf{B}_n$ are defined for the

specific TEQ designs. Three categories of TEQ design approaches have been summarized in [4], i.e., the single Rayleigh quotient case (N_a=1), to which the MMSE and MSSNR TEQ design methods belong; the multiple Rayleigh quotient case (N_a>1) based on which the MGSNR, MBR and MDR methods were developed; and the case of multiple filters each representing a single quotient, of which the PTEQ is a proper example.

PTEQ consists of a set of complex-valued equalizers that are designed separately for each tone and gives a true maximum bit-rate. The MMSE optimization is performed in PTEQ for each tone. The per-tone approach exhibits a significantly improved performance compared to other TEQs. For a certain range of synchronization delay, the per-tone approach gives a larger capacity for each single tone and hence results in a larger total capacity. Moreover, the capacity is not sensitive to the synchronization delay within this range, implying that if an exhaustive search over the synchronization delay range cannot be afforded, it is easy to pick an appropriate synchronization delay when using PTEQ [5]. Even though each single tone is treated individually with an optimal equalization in the frequency domain, the total computational effort for data transmission still remains comparable with other equalization techniques. However, the PTEQ requires a high memory cost and a large computational amount for initialization. Therefore, it is imperative to develop efficient and direct PTEQ initialization algorithms, preferably, adaptive initialization schemes, such as the least-mean-square (LMS) algorithm and the square-root (SR) recursive least-squares (RLS) algorithm.

2.4 Conclusions

In this chapter, we have introduced the basic concepts of MC and DMT modulations as well as the basic principles of channel equalization applicable to DSL communication systems. First, we have shown how MC modulation is theoretically capable of achieving the channel capacity. Then, some restrictions and approximations were discussed in order to obtain an MC modulation scheme that can be implemented in practice, namely, the DMT modulation. We have explained why a CP has been introduced in the transmitter of DMT modulation and how it can be used to avoid ISI and ICI. It has been shown that the DMT scheme results in an efficient and channel-independent modulation and demodulation by using DFT, IDFT and a simple 1-tap equalizer. It has also been pointed out that the DMT modulation scheme alone is not sufficient for high-speed DSL communication. As the use of an insufficiently long CP along with overlapped sinc-shaped tone spectrum results in a bit-rate far below the channel capacity, additional equalization is needed to enhance the bit-rate. For the purpose of equalization, the TEQ has also been discussed, disclosing a common framework for TEQ designs. The PTEQ as an important equalization technique for VDSL has been briefly introduced in this chapter, which will be detailed in the next chapter with practical performance studies.

Chapter 3

Per-tone Equalization Technique for VDSL

As discussed in Chapter 2, the commonly used equalization structure consists of a time-domain equalizer (TEQ) and a 1-tap frequency-domain equalizer (FEQ) for each tone. In this chapter, an alternative equalization structure will be described, namely, the per-tone equalization (PTEQ), which implements a multi-tap FEQ for each tone individually. In the new structure, the SNR of each tone is maximized by solving a minimum-mean-square error (MMSE) problem, such that a maximum bit-rate of the overall system can be achieved. A performance study of the VDSL system using PTEQ is carried out by considering several VDSL test loops and typical impairments in the VDSL transmission environment according to the ANSI standard T1.424.

In Section 3.1, PTEQ is presented in detail. Mathematical basis leading to the formulation of PTEQ is given in Section 3.1.1, along with the PTEQ structure. It is shown in Section 3.1.2 that the complexity of PTEQ during data transmission is comparable to other TEQ/FEQ methods.

In Section 3.2, the simulation of VDSL test loops is conducted. Section 3.2.1 introduces the method of obtaining the channel impulse response (CIR) of VDSL test loops by means of modeling RLCG (resistance, inductance, capacitance and conductance) characteristics of the transmission line. The simulation results of CIR of different test loops are shown in Section 3.2.2.

In Section 3.3, simulation study of VDSL systems using PTEQ in the standard

transmission environment is performed. The standard signal and transceiver configurations in practical VDSL systems as defined in [3] are introduced in Section 3.3.1. Simulation results for various test loops, channel noise models and equalizers with different taps are given in Section 3.3.2.

3.1 Per-tone Equalization Technique

3.1.1 PTEQ concept and structure

In order to derive the PTEQ algorithm, the TEQ and FEQ operations in Figure 2.7 need to be presented mathematically. Given the received signal vector at DMT symbol time k as $\mathbf{y}_k = \left[y_{k,-v} \quad \cdots \quad y_{k,0}\, y_{k,1} \quad \cdots \quad y_{k,N-1} \right]^T$ and the TEQ tap-weight vector

$$\mathbf{w} = \left[w_0 \quad \cdots \quad w_{M-1} \right]^T \tag{3.1}$$

where M is the number of TEQ taps, the k-th equalized DMT symbol is the convolution of the received signal $\mathbf{y}_k$ with the TEQ $\mathbf{w}$. Since for every DMT symbol we only need N equalized signals at the TEQ output and the added prefix is discarded, the convolution can be represented in matrix operation by defining an $N \times M$ Toeplitz matrix $\mathbf{Y}_k$ of the k-th received signal vector as

$$\mathbf{Y}_k = \begin{bmatrix} y_{k,0} & \cdots & y_{k,-M+1} \\ \vdots & \ddots & \vdots \\ y_{k,N-1} & \cdots & y_{k,N-M} \end{bmatrix} \tag{3.2}$$

Then, the $N \times 1$ equalized signal vector at the TEQ output can be written as

$$\mathbf{y}_{k,w} = \mathbf{Y}_k \mathbf{w} \tag{3.3}$$

where the subscript w is added to identify the equalized signal. The DFT of $\mathbf{y}_{k,w}$ can be

written as

$$\begin{aligned}\tilde{\mathbf{y}}_{k,w} &= \mathcal{F}_N(\mathbf{y}_{k,w}) = \mathcal{F}_N(\mathbf{Y}_k\mathbf{w}) \\ &= (\mathcal{F}_N\mathbf{Y}_k)\mathbf{w} = \tilde{\mathbf{Y}}_k\mathbf{w}\end{aligned} \tag{3.4}$$

or its n-th element corresponding to the output of on tone n is given by:

$$\tilde{y}_{k,n,w} = \mathcal{F}_n^T(\mathbf{Y}_k\mathbf{w}) = (\mathcal{F}_n^T\mathbf{Y}_k)\mathbf{w} = \tilde{\mathbf{p}}_{k,n}^T\mathbf{w} \tag{3.5}$$

In (3.4), $\tilde{\mathbf{Y}}_k$ is an $N\times M$ matrix comprised of the DFTs of the columns of $\mathbf{Y}_k$, and $\tilde{\mathbf{p}}_{k,n}^T = \mathcal{F}_n^T\mathbf{Y}_k$ in (3.5) is a $1\times M$ vector comprised of M consecutive DFT coefficients for tone n. It follows from (2.25) that the transmitted symbol $\tilde{x}_{k,n}$ at the n-th FEQ output can then be estimated as:

$$\hat{\tilde{x}}_{k,n} = \tilde{d}_n^*\tilde{y}_{k,n,w} = \tilde{\mathbf{p}}_{k,n}^T\left(\tilde{d}_n^*\mathbf{w}\right) = \tilde{\mathbf{p}}_{k,n}^T\mathbf{w}^*{}_n \tag{3.6}$$

where $\tilde{d}_n^*$ is the conjugate of the complex-valued FEQ coefficient $\tilde{d}_n$ on tone n. It is seen from (3.6) that the TEQ vector $\mathbf{w}$ and the FEQ parameter $\tilde{d}_n$ have been combined into a single tone-dependent $M\times 1$ PTEQ coefficient vector $\mathbf{w}_n$.

It is also clear from (3.5) and (3.6) that the estimation of each transmitted symbol $\tilde{x}_{k,n}$ requires the calculation of M consecutive DFTs plus a matrix-vector multiplication, which is computationally intensive. An efficient way of computing these M DFTs has been proposed [27], which is based on the Toeplitz structure of $\mathbf{Y}_k$. Each item in the $1\times M$ vector $\tilde{\mathbf{p}}_{k,n}^T$, denoted as $\tilde{p}_{k,n}^T[m], m\in[0,M-1]$, is related by the following recursion:

$$\tilde{p}_{k,n}^T[m] = \alpha_n\tilde{p}_{k,n}^T[m-1] + (y_{k,-m} - y_{k,N-m}),\quad m = 1,....,M-1 \tag{3.7}$$

where $\alpha_n = \exp(-j2\pi(n-1)/N)$. Thus, the DFT of a column of $\mathbf{Y}_k$ can then be

derived from the DFT of its previous column plus the correction term. It follows from (3.7) that

$$\tilde{\mathbf{p}}_{k,n} = \mathbf{T}_n \begin{bmatrix} \Delta\mathbf{y}_k \\ \tilde{y}_{k,n} \end{bmatrix} \tag{3.8}$$

where

$$\mathbf{T}_n = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & \alpha_n \\ \vdots & \vdots & \iddots & \iddots & \vdots \\ 0 & 1 & \iddots & \iddots & \alpha_n^{M-2} \\ 1 & \alpha_n & \cdots & \alpha_n^{M-2} & \alpha_n^{M-1} \end{bmatrix},$$

$\Delta\mathbf{y}_k$ is an $(M-1)\times 1$ vector of difference terms, namely,

$$\Delta\mathbf{y}_k = \begin{bmatrix} y_{k,-M+1} - y_{k,-M+N+1} \\ y_{k,-M} - y_{k,-M+N} \\ \vdots \\ y_{k,-1} - y_{k,N-1} \end{bmatrix} = \begin{bmatrix} \Delta y_{k,-M+1} \\ \Delta y_{k,-M} \\ \vdots \\ \Delta y_{k,-1} \end{bmatrix} \tag{3.9}$$

and $\tilde{y}_{k,n}$ is given by

$$\tilde{y}_{k,n} = \mathcal{F}_n^T \mathbf{y}_{k,0:N-1} = \tilde{p}_{k,n}[0], (n = 0,1,\cdots,N-1) \tag{3.10}$$

In this way, only one DFT has to be performed to obtain $\tilde{y}_{k,n}$ $(n = 0,1,\cdots,N-1)$, and the $M-1$ remaining DFTs can be calculated from $\tilde{y}_{k,n}$ along with a linear combination of $M-1$ tone-independent $\Delta\mathbf{y}_k$. The computation of $\tilde{\mathbf{p}}_{k,n}$ for N_a active tones thus requires one DFT given by (3.10), $M-1$ subtractions in (3.9) and $N_a(M-1)$ recursions in (3.7). Using (3.6) and (3.8), the symbol estimate $\hat{\tilde{x}}_{k,n}$ can be written as:

$$\hat{\tilde{x}}_{k,n} = [\Delta\mathbf{y}_k^T \;\; \tilde{y}_{k,n}]\mathbf{v}^*_n = \mathbf{v}^H{}_n[\Delta\mathbf{y}_k^T \;\; \tilde{y}_{k,n}]^T = \mathbf{v}^H{}_n\mathbf{u}_{k,n} \tag{3.11}$$

where $\mathbf{v}_n = \mathbf{T}_n \mathbf{w}_n$ and $\mathbf{u}_{k,n} = [\Delta \mathbf{y}_k^T \ \tilde{y}_{k,n}]^T$. As shown in Figure 3.1, the PTEQ is now implemented as an M-tap filter $\mathbf{V}_n$ for each tone separately.

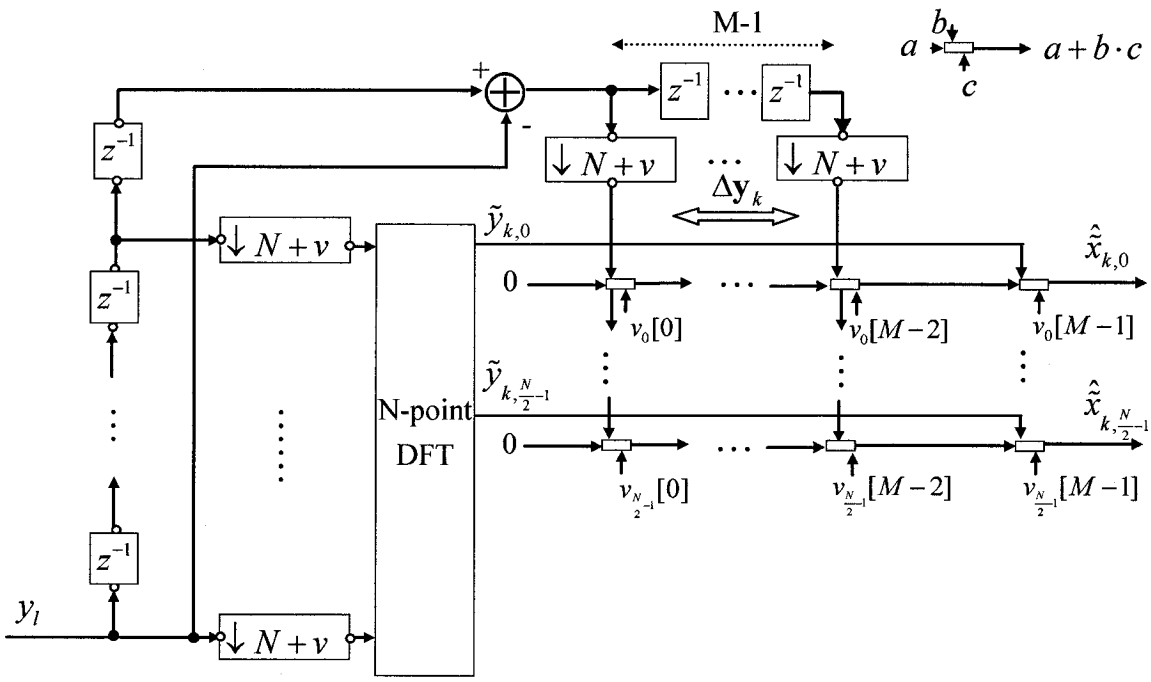


Figure 3.1 The per-tone equalizer

The SNR on tone n can then be maximized by determining the coefficient $\mathbf{v}_n$, which is equivalent to the minimization of the mean-square-error (MMSE) at the PTEQ output of each tone. The estimation error of tone n is given by

$$e_{k,n} = \hat{\tilde{x}}_{k,n} - \tilde{x}_{k,n} = \mathbf{v}^H{}_n \mathbf{u}_{k,n} - \tilde{x}_{k,n} \tag{3.12}$$

According to the MMSE criterion, we need to find the PTEQ coefficient vector for each tone n, $\mathbf{v}_n$, such that the mean of the squared error $|e_{k,n}|^2$ is minimized. That is, we need to solve the following cost function for tone n:

$$\min_{\mathbf{v}_n} \varepsilon\{|\mathbf{v}^H{}_n \mathbf{u}_{k,n} - \tilde{x}_{k,n}|^2\} \tag{3.13}$$

where $\varepsilon\{\}$ denotes the mean operation. Using Wiener-Hoff equations for linear transversal filters [28], the solution to (3.13) can be given by

$$\mathbf{v}_n = \varepsilon\{\mathbf{u}_{k,n}\mathbf{u}^H{}_{k,n}\}^{-1}\varepsilon\{\mathbf{u}_{k,n}\tilde{x}^*_{k,n}\} = (\sigma^2_{u,n})^{-1}\sigma_{u\tilde{x},n} \tag{3.14}$$

where $\sigma^2_{u,n} = \varepsilon\{\mathbf{u}_{k,n}\mathbf{u}^H_{k,n}\}$ is the auto-correlation matrix of the input signal $\mathbf{u}_{k,n}$, and $\sigma_{u\tilde{x},n} = \varepsilon\{\mathbf{u}_{k,n}\tilde{x}^*_{k,n}\}$ the cross-correlation vector of the input signal $\mathbf{u}_{k,n}$ and the desired response $\tilde{x}_{k,n}$. They can be obtained based on the knowledge of the channel impulse response, the channel noise and a series of training symbols sent by the transmitter.

3.1.2 Complexity of PTEQ

The computational complexity of PTEQ is similar to that of the traditional TEQ/FEQ during data transmission. When TEQ/FEQ is used in the receiver, for each DMT symbol, we first need to calculate the convolution of the received signal vector $\mathbf{y}_{k,-M+1:N-1}$ with the M-tap real-valued TEQ $\mathbf{w}$, which requires NM real multiplications and $N(M-1)$ real additions. We then need to undertake one N point DFT of the TEQ output, plus $\frac{N}{2}$ complex multiplications to get the product of the complex-valued DFT output and the coefficient of the 1-tap complex-valued FEQ. Note that one multiplication of two complex-valued numbers is counted as four real multiplications and two additions, while one multiplication of a complex-valued number with a real-valued number involves two real multiplications. The implementation of the FEQ requires $\frac{N}{2}\times 4$ real multiplications and $\frac{N}{2}\times 2$ additions. For the implementation of PTEQ, we need one

DFT of the un-equalized signal vector $\mathbf{y}_k$ plus the operations required for $\frac{N}{2}$ linear combinations of the complex-valued vector $\mathbf{v}_n$ with one complex-valued DFT output $\tilde{y}_{k,n}$ as well as $M-1$ real-valued difference terms $\Delta\mathbf{y}_k^T$ (see (3.11)). Therefore, a total of $\frac{N}{2}[4+2(M-1)]$ multiplications and $\frac{N}{2}[2+2(M-1)]+(M-1)$ additions are required at the PTEQ receiver. In both cases, it is assumed that all tones are used for transmission, that is, the number of used tones is $\frac{N}{2}$.

The computational complexity of the TEQ/FEQ technique and that of the PTEQ are shown in Table 3.1 and Table 3.2, respectively, in terms of the number of real multiplications and additions. In both cases, DFT is realized by the FFT algorithm which

Table 3.1 Complexity of equalization with TEQ/FEQ

	Number of real multiplications	Number of real additions
TEQ	NM	$N(M-1)$
FEQ	$\frac{N}{2}\times 4$	$\frac{N}{2}\times 2$
Total	$N(M+2)$	NM

Table 3.2 Complexity of PTEQ

	Number of real multiplications	Number of real additions
difference terms $\Delta\mathbf{y}_k^T$	0	$M-1$
PTEQ	$\frac{N}{2}[4+2(M-1)]$	$\frac{N}{2}[2+2(M-1)]$
Total	$N(M+1)$	$NM+M-1$

is at the computational complexity of $\mathcal{O}(N\log(N))$ [29]. As the FFT module is common for both techniques, its implementation cost is not counted in the tables.

It is seen from Table 3.1 and Table 3.2 that the total complexity of PTEQ and that of TEQ/FEQ during data transmission are comparable and both are of order NM. However, computing the MMSE solution (3.13) for all tones to initialize the PTEQ coefficient $\mathbf{v}_n$ requires heavy computations, since it is based on the statistical result of the received large -length signals. Therefore, we will present adaptive algorithms for the initialization of PTEQ coefficients to reduce the computational complexity in Chapter 4.

3.2 Modeling of VDSL Test Loops

In order to conduct a performance study of PTEQ in VDSL, we need first to formulate the channel impulse response of VDSL transmission lines. In [3], seven types of VDSL test loops are given for various transmission environments in terms of different loop lengths and gauges of wire as well as bridge taps, etc. In this section, we first introduce the method of modeling the twisted pair, and then present the simulation results of the frequency response of several test loops.

3.2.1 Modeling of twisted pair

Most twisted-pair phone lines can be well-modeled for the transmission of frequencies up to 30 MHz by using a two-port model, namely, the so-called ABCD theory. The ABCD theory has been well covered in basic electromagnetic texts, and has been presented in [30] to model the DSLs. A general circuit exploiting such a two-port model is shown in

Figure 3.2. The relationship between the input and the output of the two-port model is given in terms of the voltage-current pair,

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_2 \\ I_2 \end{bmatrix} = \Psi \begin{bmatrix} V_2 \\ I_2 \end{bmatrix} \tag{3.15}$$

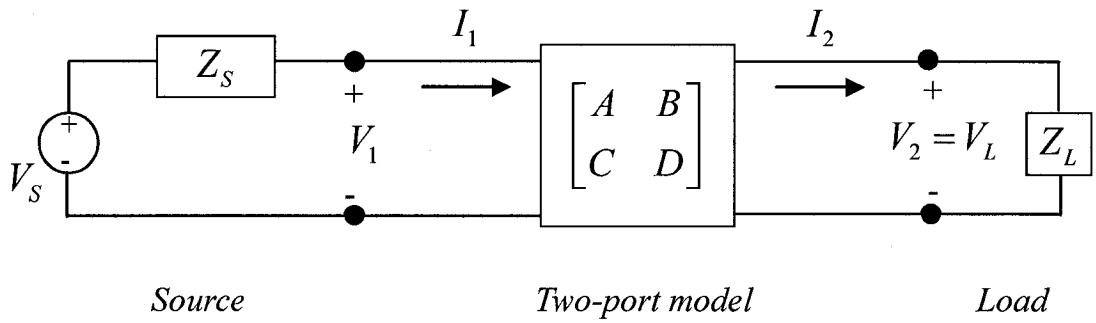


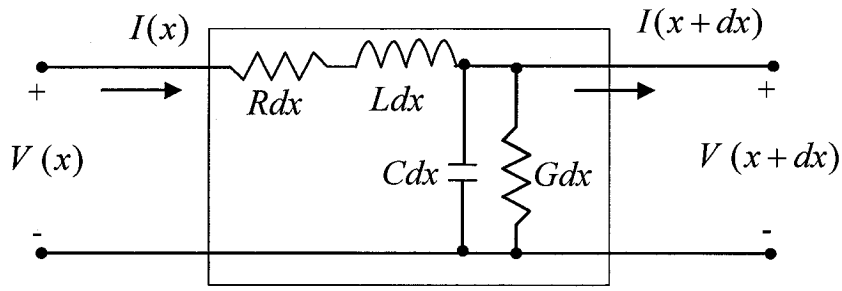
Figure 3.2 A general circuit exploiting a two-port model

where $\Psi = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ is the two-port matrix. From (3.15), the transfer function of the two-port model can be given by $\frac{V_2}{V_1}$. The four frequency-dependent parameters A, B, C and D fully describe the characteristic of the two-port model. Moreover, a cascade of multiple two-port models can be represented by a product of a number of the two-port matrices,

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \Psi_1 \Psi_2 \cdots \Psi_{N-1} \begin{bmatrix} V_N \\ I_N \end{bmatrix} = \Psi \begin{bmatrix} V_N \\ I_N \end{bmatrix} \tag{3.16}$$

which allows for the calculation of the transfer function of a more complicated network. The two-port modeling is suitable for the twisted-pair phone line, since the line can be characterized by the resistance, inductance, capacitance and conductance (i.e., RLCG) per unit length, and the whole phone line can be regarded as a cascade of many segments

with unit length. Figure 3.3 shows the two-port model of a segment of twisted-pair phone line.



$$Z = R + j\omega L$$
$$Y = G + j\omega C$$

Figure 3.3 Two port model for a segment of twisted-pair phone line

For a segment of phone line of length d, the two-port matrix can be represented as [9]:

$$\Psi_d = \begin{bmatrix} \cosh(\gamma d) & Z_0 \sinh(\gamma d) \\ \frac{1}{Z_0}\sinh(\gamma d) & \cosh(\gamma d) \end{bmatrix} \tag{3.17}$$

where $Z_0 = \sqrt{\frac{R+j\omega L}{G+j\omega C}} = \sqrt{\frac{Z}{Y}}$ and $\gamma = \sqrt{(R+j\omega L)(G+j\omega C)} = \sqrt{ZY}$. More specifically, Z_0 is the characteristic impedance of the twisted pair and γ the propagation constant for the twisted pair, both of which characterize the segment of transmission line and can be determined by measurement. Standard values of these parameters for different types of twisted-pair phone lines can be found in [46]. As a special case, a bridge tap of length d can be modeled as:

$$\Psi_d = \begin{bmatrix} 1 & 0 \\ \frac{1}{Z_0}\tanh(\gamma d) & 1 \end{bmatrix} \tag{3.18}$$

Based on the above modeling method and the deployment of test loops, the transfer function of VDSL channels can then be calculated.

3.2.2 Simulation of VDSL test loops

Seven VDSL test loops have been given in [3], which are shown in Figure 3.4. There are four types of telephone lines introduced in the VDSL test loops, including TP1, TP2, TP3 and FP, whose RLCG values can be found in ANSI T1.417-2001 spectrum management

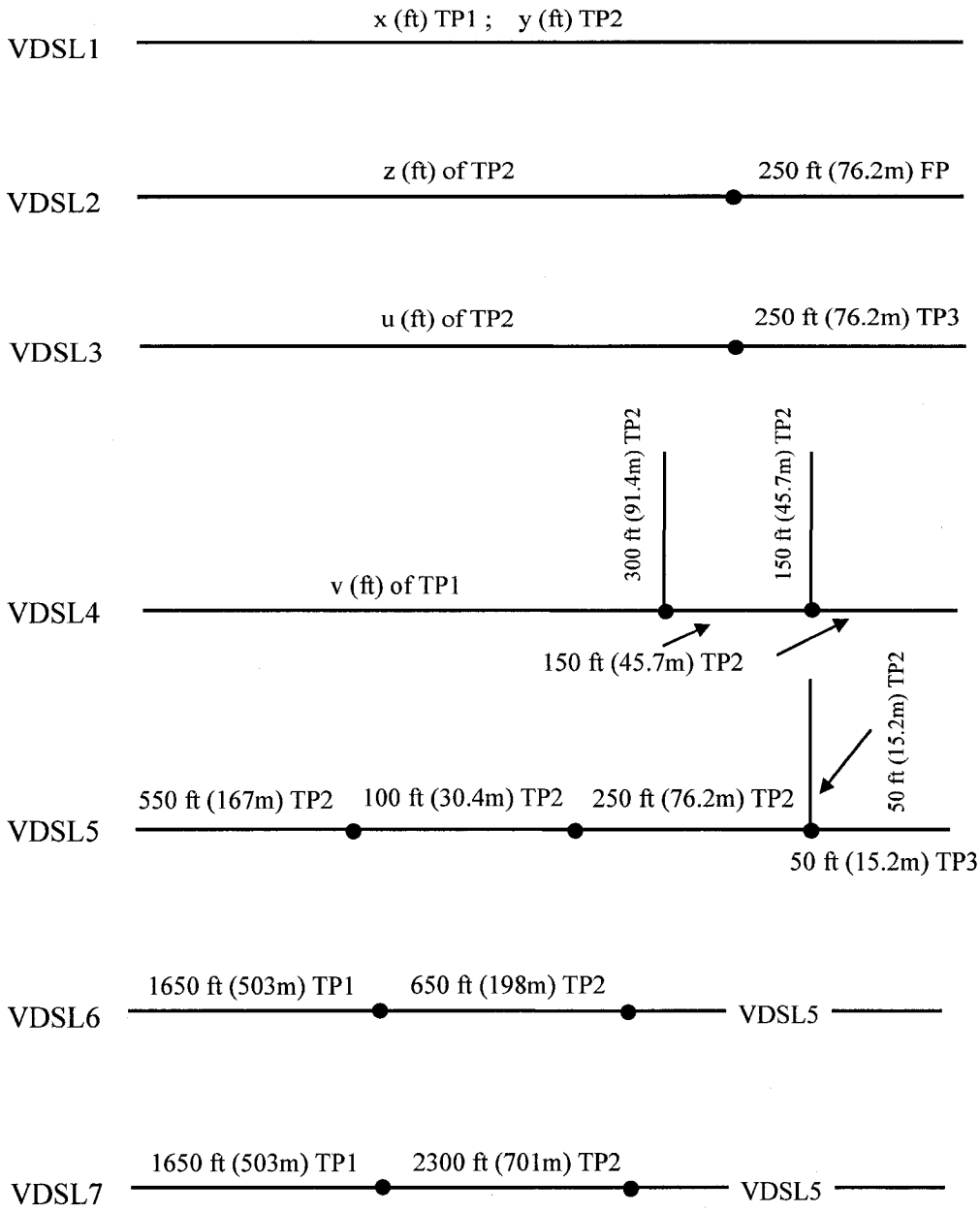


Figure 3.4 VDSL test loops [3]

standard [46]. The loop length parameters x, y, z, u and v shown in Figure 3.4 are categorized into short, medium and long sizes individually, as given in Table 3.3.

Table 3.3 VDSL test loop length [3]

Variable name	Short value	Medium value	Long value
x (VDSL1)	1000 ft. (304.8m)	3000 ft. (914.4m)	4500 ft. (1371.6m)
y (VDSL1)	1500 ft. (457.2m)	3000 ft. (914.4m)	4500 ft. (1371.6m)
z (VDSL2)	1500 ft. (457.2m)	3000 ft. (914.4m)	4500 ft. (1371.6m)
u (VDSL3)	1500 ft. (457.2m)	3000 ft. (914.4m)	4500 ft. (1371.6m)
v (VDSL4)	1000 ft. (304.8m)	3000 ft. (914.4m)	4500 ft. (1371.6m)

Based on the modeling method introduced in Section 3.2.1, the frequency response of the VDSL test loops are shown in Figure 3.5. It is clear from the figures that the channel attenuation is increased with the frequency, which intensively influences on the transmitted signals occupying a wide bandwidth. As a result, equalization technique is required at the receiver to compensate for the channel attenuation. From all the test loops, it is also seen that the attenuation is increased with the length of the loop. Thus the achievable transmission bit-rate on a long loop is lower than that on a short loop. Test loop 1 represents the situation that an underground wire is used to link the customer to a ONU, while test loop 2 and test loop 3 denote a aerial wire (z ft. or u ft. TP2) connected with a drop wire of different types (FP in loop 2 and TP3 in loop 3). Test loop 4 shows the situation where bridge taps are introduced on a transmission line, and test loops 5 through 7 give more complex situations to cover more joints and bridge taps. As we have stated in Section 1.2 and observed from Figure 3.5, for the loops with bridge taps (VDSL test loop 4-7), there are notches in the transfer function at certain frequencies. At those frequencies the bounced back signal is approximately 180 degrees different in

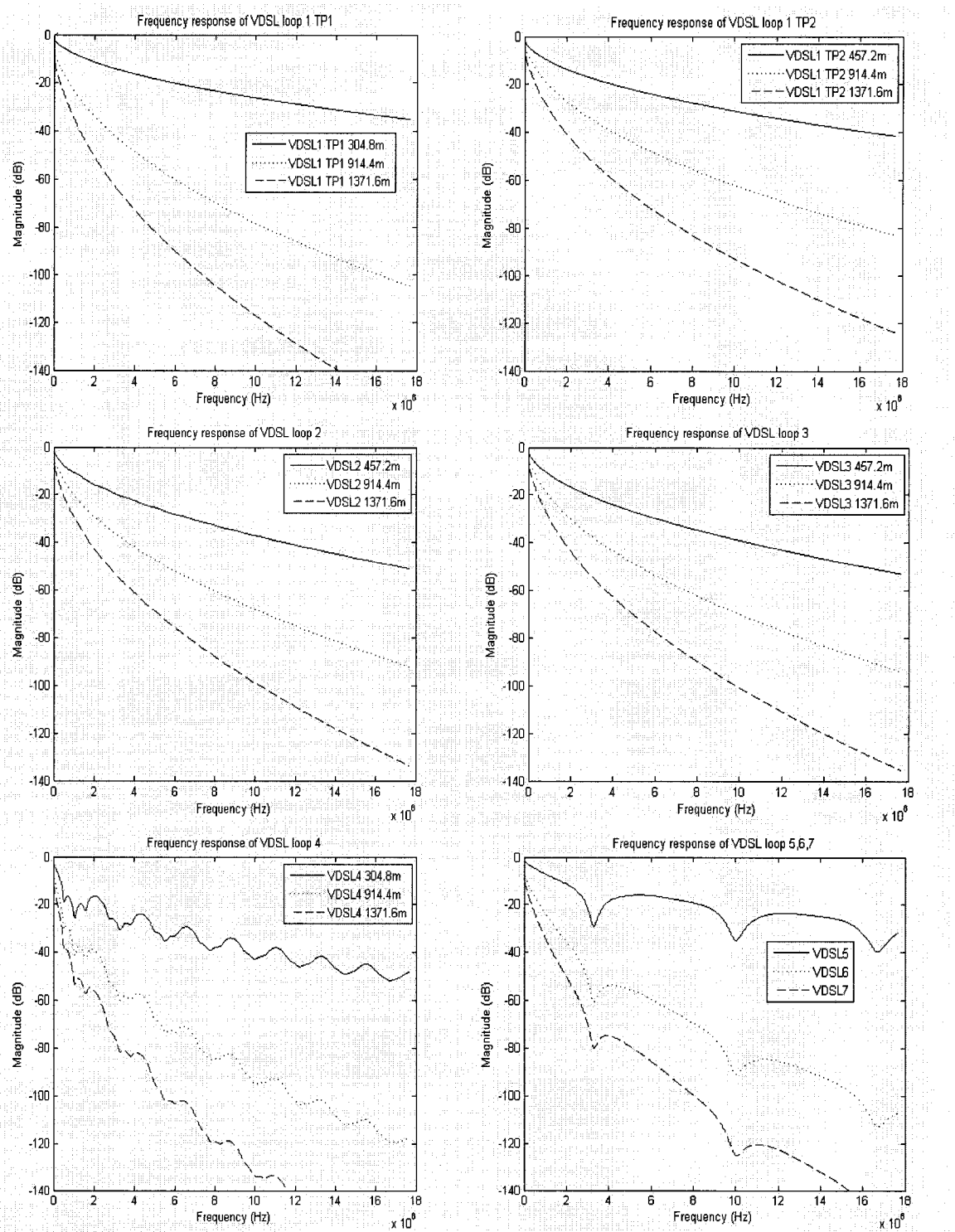


Figure 3.5 Frequency responses of VDSL test loops

phase from the original transmission signal, meaning that the length of the bridge tap is an odd multiple of one-quarter of the signal's wavelength. It can be represented as:

$$d = (2m+1)\frac{\lambda}{4} \quad meters \quad m = 0,1,2\cdots \tag{3.19}$$

where d is the length of the bridge tap and λ the wavelength of signal. Since the wavelength is equal to the velocity divided by frequency, i.e., $\lambda = v/f$, and in the telephone wire v can be approximated by $c/1.5$, where $c = 3\times10^8$ m/s is the speed of light, we can get the notch frequencies as:

$$f_{notch} = \frac{50}{d}(2m+1) \quad MHz \quad m = 0,1,2\cdots \tag{3.20}$$

Note that they are quite matched for the simulation results of VDSL test loop 4-7 as shown in Figure 3.5.

3.3 Simulation of PTEQ in VDSL

In this section, the per-tone equalization is simulated in a VDSL system through computer programs, trying to show the performance improvement of VDSL systems after using the PTEQ at the receiver. In [5], a data model is constructed for the study of the PTEQ performance based on ADSL channel settings, which uses a matrix-based computation to avoid the implementation of a practical DMT transceiver. However, this data model is based on some simplified assumptions and omits the constraints on the transmitted signals. Moreover, since the number of tones is greatly increased from 256 in ADSL to 4096 in VDSL, a much more memory cost is incurred for the matrix-based computation. Hence, this data model is not suitable for computer simulation of a VDSL system. In our simulation study, we try to implement a practical VDSL system according

to ANSI standard T1.424, using the specified signals, noises, channels and modulation schemes. In Section 3.3.1, some conditions for the simulation are shown to build a VDSL system. Simulation results as well as some remarks are given in Section 3.3.2.

3.3.1 Simulation conditions

The received signal can be got by means of a convolution of the channel impulse response and the transmitted signals plus channel noises. In Section 3.2, we have got the channel frequency responses of VDSL loops. The VDSL channel impulse response can then be obtained by transferring the response in frequency domain into time domain using Fourier transforms. To practically simulate the VDSL system, we need further information about the transmitted signals, noises and other parameters, which will be described as follows.

Training signals. A wire line channel is typically slowly time-varying as channel variations are caused by weather conditions and temperature [32]. This allows for a training-based equalizer initialization during the connection set-up phase. A specified DMT symbol training sequence is transmitted for initializing the equalizer. The training signal is made of all the allowable downstream tones modulated in 4 QAM, which means that the symbol of each tone shall be mapped from 2 bits. The 2-bit pseudo-random number is the output of a pseudo-random bit generator defined by the following equation:

$$d_n = d_{n-9} \oplus d_{n-11} \tag{3.21}$$

The bit generator is illustrated in Figure 3.6. Initially, all the registers of the bit generator are set to one. For a VDSL system that uses $\frac{N}{2}$ tones, N bits shall be generated for every DMT-symbol by the scrambler shown in Figure 3.6, denoted as $d_0 d_1 \cdots d_{N-2} d_{N-1}$.

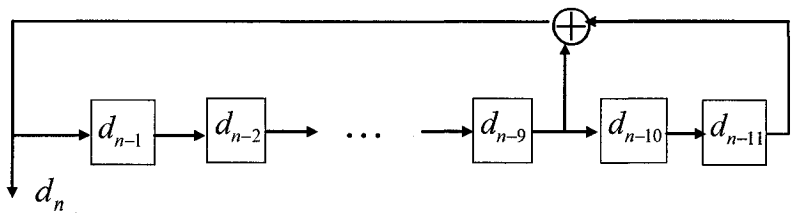


Figure 3.6 Bit generator for training signal

The first two bits $d_0 d_1$ correspond to tone 0, the next two bits $d_2 d_3$ to tone 1, ...In general, bits $d_{2j} d_{2j+1}$ correspond to tone j. The two bits on each tone are then mapped to one of the four QAM symbols (i.e., $1+1i, 1-1i, -1+1i, -1-1i$). If the scrambler is reset at the start of every symbol (i.e., all registers are reset to one), the same N bits will be used each symbol. This periodic training sequence is called a REVERB signal. On the other hand, if the scrambler is not reset, but keeps running for the subsequent symbols, a

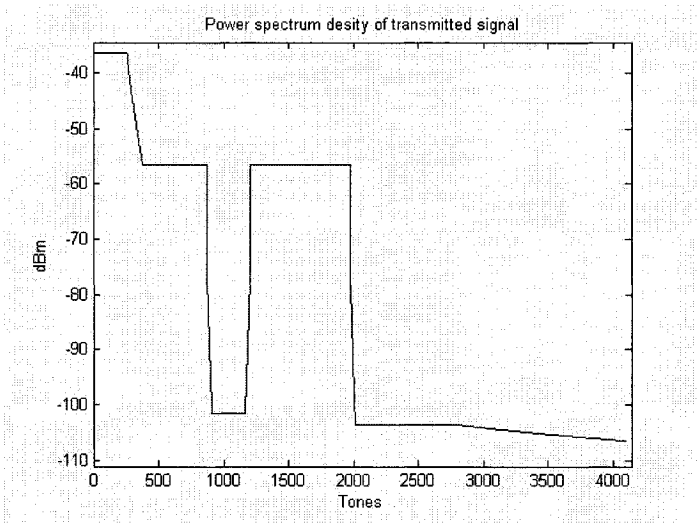


Figure 3.7 PSD of transmitted signal (each tone corresponds to 4.3125kHz)

non-periodic training sequence, called the MEDLEY signal, will be generated. Additionally, the signal transmitted on each tone has to comply with the transmission

PSD mask. Figure 3.7 illustrates a PSD mask of VDSL downstream for FTTEx deployment. As a result, the transmitted training signal of each tone has to multiply by a spectrum shaping coefficient to restrict the signal power on each tone.

NEXT and FEXT. As we have introduced in Section 1.2, FEXT and NEXT are two types crosstalk in VDSL transmission environment. The power spectrum density (PSD) of NEXT and FEXT is given by:

$$PSD(f) = PSD_{Upstream} \times \left|H_{NEXT}(f,n)\right|^2 + PSD_{Downstream} \times \left|H_{FEXT}(f,n,l)\right|^2 \qquad (3.22)$$

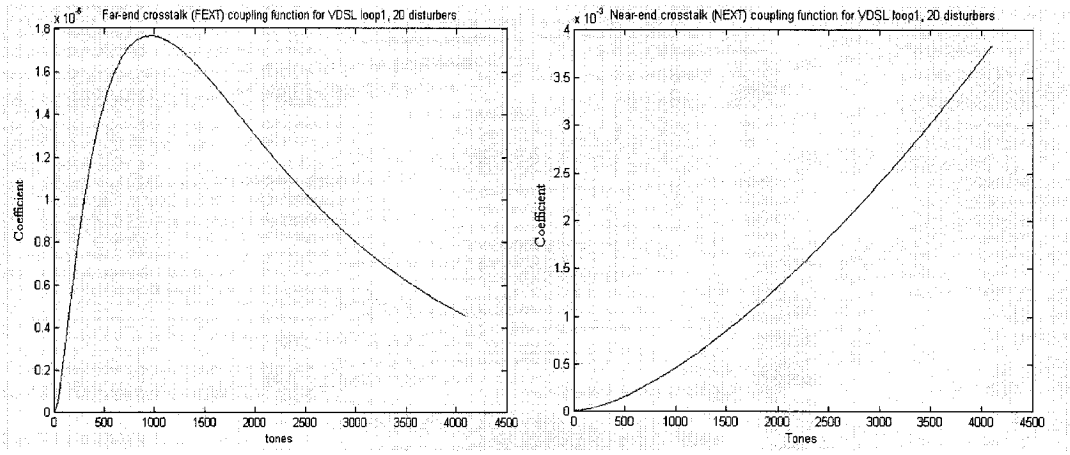
where $PSD_{Upstream}$ is the disturber's upstream PSD, $PSD_{Downstream}$ the disturber's downstream PSD, and $\left|H_{NEXT}(f,n)\right|^2$ and $\left|H_{FEXT}(f,n,l)\right|^2$ are given by:

$$\left|H_{NEXT}(f,n)\right|^2 = K_{NEXT} \times (n/49)^{0.6} \times f^{1.5} \times (1-\left|H_{channel}(f)\right|^4) \qquad (3.23)$$

and

$$\left|H_{FEXT}(f,n,l)\right|^2 = K_{FEXT} \times (n/49)^{0.6} \times l \times f^2 \times \left|H_{channel}(f)\right|^2 \qquad (3.24)$$

respectively, where $K_{NEXT} = 8.818 \times 10^{-14}$, $K_{FEXT} = 7.999 \times 10^{-20}$, n is the number of



(a) FEXT (b) NEXT

Figure 3.8 FEXT and NEXT coupling function for VDSL loop 1, 20 disturbers

disturbers and l the length of the pairs. Figure 3.8 plots the NEXT and FEXT coupling functions of VDSL test loop 1 with 20 disturbers, where (a) represents the FEXT coupling function and (b) the NEXT coupling function. As we can see from Figure 3.8, NEXT is usually more harmful than FEXT because NEXT is not attenuated by the transfer function of the line. By combining the downstream and/or upstream PSD of disturbers (e.g., ADSL, HDSL, VDSL) with the coupling functions, the FEXT and NEXT can be well constructed through (3.22) to (3.24).

Parameters. Table 3.4 summarizes the parameters for the simulation of a VDSL system.

Table 3.4 Parameters for the simulation of a VDSL system

Number	Description	Parameter
1	Tones	4096
2	DFT/IDFT size	8192
3	Cyclic prefix length	640
4	DMT symbols per second	4000
5	Tone spacing	4.3125 kHz
6	Coding gain	4.2 dB
7	Noise Margin	6 dB
8	Total transmitter power	14.5 dBm
9	Effective downstream tones	32~865 and 1210~1966
10	Background noise PSD	-140 dBm/Hz
11	Bits per tone	2~15 bits
12	Duplexing method	Frequency division duplexing (FDD)

A VDSL system uses $\frac{N}{2}=4096$ tones to transmit data, with each tone carrying 2 to 15 bits depending on the SNR of the tone. The DFT/IDFT size is accordingly twice double of the number of tones, i.e., $N=8192$. An appropriate CP length has been chosen as $v=640$ to avoid the ISI between DMT symbols. Every second 4000 DMT symbols are transmitted, giving a sampling frequency of $f'=4000\times(8192+640)=35.328\,MHz$, and

a tone spacing of $f'/N = 4.3125\ kHz$. The bandwidth of the transmitted signal is thus $4.3125 \times 4096 = 17.664\ MHz$. However, the bandwidth of VDSL systems is limited up to 12 MHz. The water-filling method introduced in Section 2.1.3 is used to calculate the bit-rate, which requires to determine a SNR gap Γ_n. In VDSL, the same SNR gap Γ_n is chosen for all tones. For a QAM symbol mapping and a target BER of 10^{-7}, a gap of 9.8 dB is obtained [31]. This gap can be reduced by means of channel coding. When Reed-Solomon coding is employed, a coding gain of 4.2 dB is obtained. Additionally, a noise margin is typically set to 6 dB [19], giving an overall SNR gap of $\Gamma_n = 9.8 - 4.2 + 6 = 11.6\,\text{dB}$. In order to transmit bidirectional signals on the single line, an upstream and downstream duplexing method is required to separate the transmitter and receiver signals. In VDSL, frequency division duplexing (FDD) is used, with different bands of downstream and upstream as shown in Figure 3.9 and Table 3.5. As a

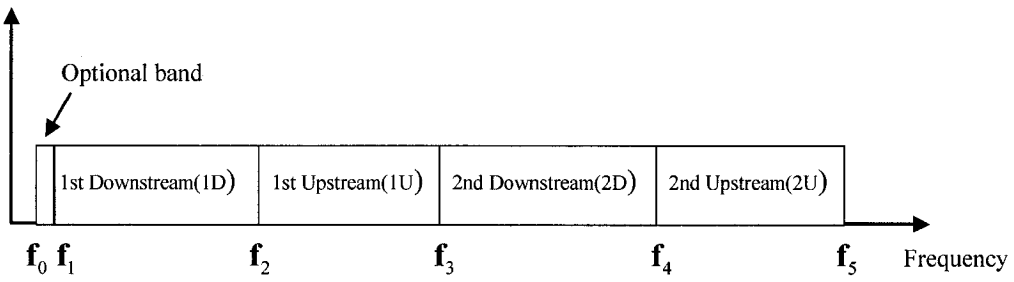


Figure 3.9 VDSL band allocations

Table 3.5 VDSL band separation frequencies

Separating Frequencies (MHz)	f_0	f_1	f_2	f_3	f_4	f_5
	0.25	0.138	3.75	5.2	8.5	12

consequence, the effective tones used for downstream are approximately from #32 to

#865 and from #1210 to #1966. It should also be mentioned that the synchronization is assumed perfect in the simulation, such that the delay of the received signal in reference to the transmitted one is omitted.

3.3.2 Simulation results

In this section, based on the transmitted MEDLEY training signal, whose PSD is shown in Figure 3.10, simulation results for various VDSL test loops are obtained and shown in comparison with the calculated PSD. We first give a simulation study of the channel attenuation and distortion. Then, the SNR of test loops with different number of equalizer taps is presented based on three channel noise models. Last, the transmission bit-rate of the simulated VDSL system is obtained, with respect to various test loops and different number of equalizer taps.

***Experiment 1**, Channel attenuation and distortion.*

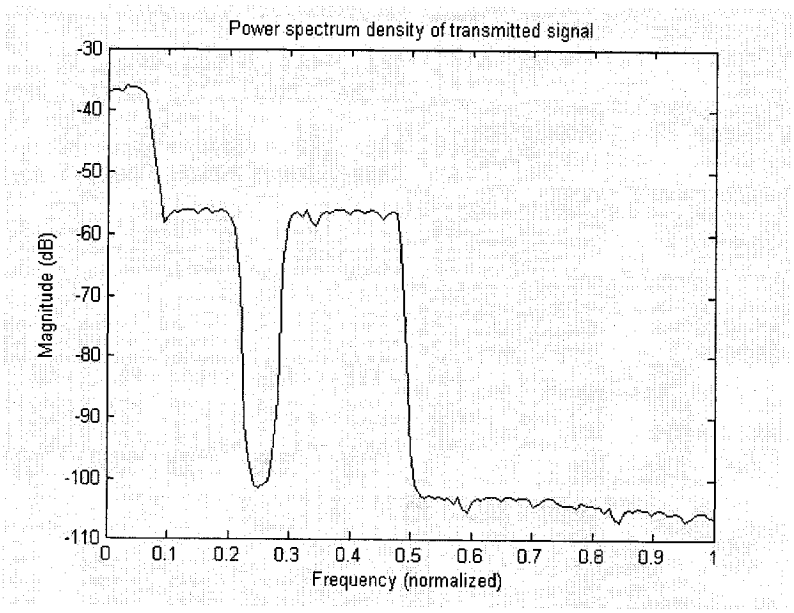


Figure 3.10 PSD of transmitted signal

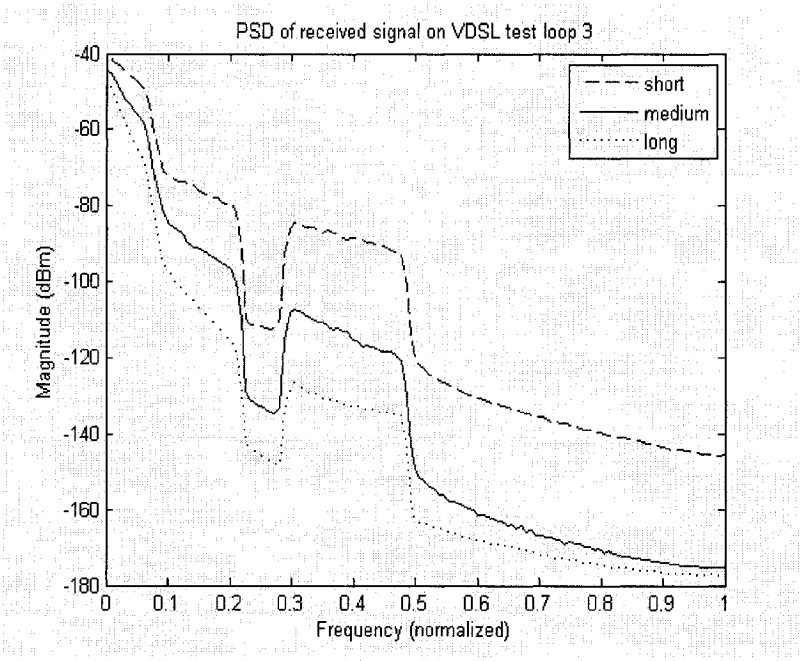


Figure 3.11 PSD of received signal on VDSL loop 3

Channel noises are not generated in this experiment, in order to evaluate the pure channel influences on the received signal. Figure 3.11 depicts the PSD of the received signal for loop 3 without channel noises, regarding the short, medium and long loop length as listed in Table 3.3. The frequency has been normalized to one. This plot clearly shows that the transmitted signal has been greatly attenuated, especially at high frequencies, and the attenuation is increased with loop length. As we mentioned in the previous chapter, if the cyclic extension is longer than the channel impulse response, the inter-symbol and inter-channel interference will be eliminated, and then the received signal on each tone is just equal to the transmitted signal on that tone multiplying the tone's attenuation. In Figure 3.12, we give a comparison of the PSD of the received signal and that of the multiplication result on VDSL loop 1. As shown in Figure 3.12, for the short loop, the received signal PSD is well matched to the results of a simple multiplication of the

transmitted signal PSD with the channel frequency response. However, for the long loop, the PSD of the received signal, as represented by the dotted lines in Figure 3.12, is higher

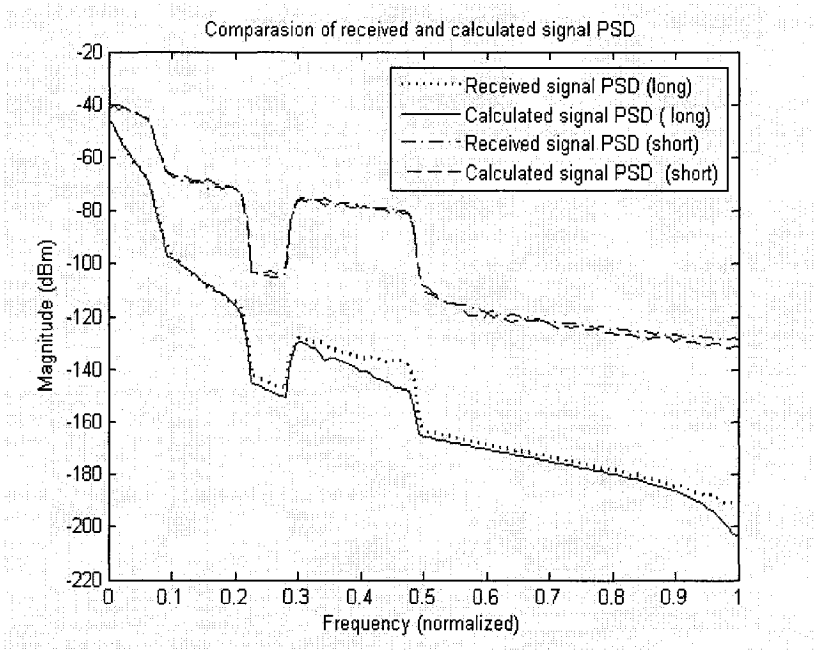


Figure 3.12 Comparison of received signal PSD and calculated signal PSD for VDSL test loop 1

than the calculated signal PSD. It suggests that a certain ISI/ICI be introduced in the received signal or certain ISI/ICI power be counted in the PSD of the received signal. Hence, in the presence of channel noises, the calculation of SNR requires counting in both the channel noises and the residual ISI and ICI due to the long channel impulse response.

***Experiment 2**, SNR of various test loops with different number of taps and different channel noise models.*

Three noise models representing several typical VDSL system channel environments [3], as given in Table 3.6, are adopted in this simulation. Figure 3.13 depicts the PSD of the three noises for the long length VDSL test loop 3. The RFI in noise model 2 is the AM

radio noise centered at 710 KHz, with the noise power of -55dBm.

Table 3.6 Test noises models

Model number	Description	Noises
1	Range test	20 VDSL self crosstalk and AWGN
2	RFI test	20 VDSL self crosstalk ,RFI and AWGN
3	Crosstalk test	10 ADSL, 16 ISDN, 4 HDSL and AWGN

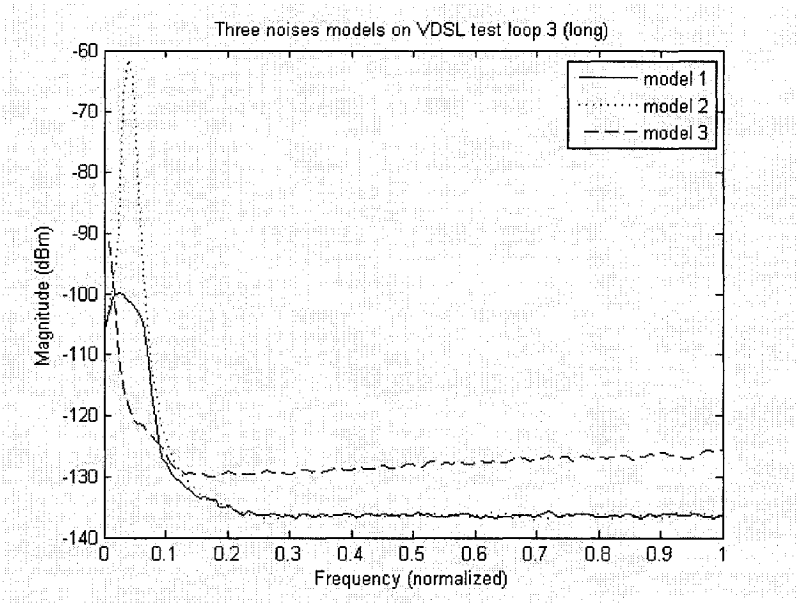


Figure 3.13 PSD of three noise models on VDSL loop 3 (long)

An efficient way to calculate the SNR on tone n is given by:

$$SNR_n = \frac{Power\ of\ desired\ signal_n}{Power\ of\ (received\ signal_n - desired\ signal_n)} \tag{3.25}$$

Note that the denominator in (3.25) has taken into account all possible noise contributions. Figure 3.14 through Figure 3.16 present the SNR of the received signal on VDSL test loop 3 with the noise model 1, 2 and 3, respectively, each including the result without using PTEQ and that with PTEQ of 4, 8 or 16 taps.

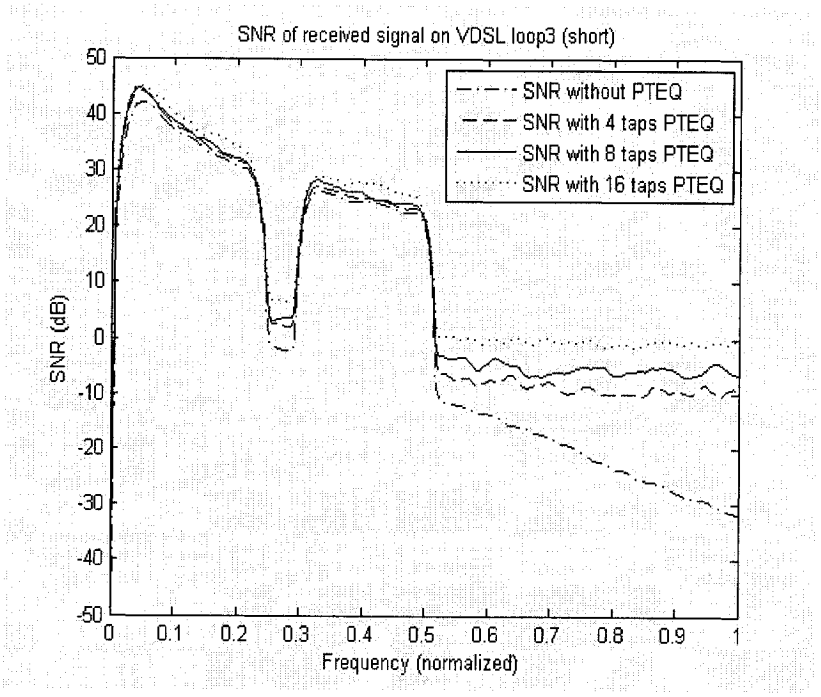


Figure 3.14 SNR of received signal on VDSL test loop 3 (short) with noise model 1

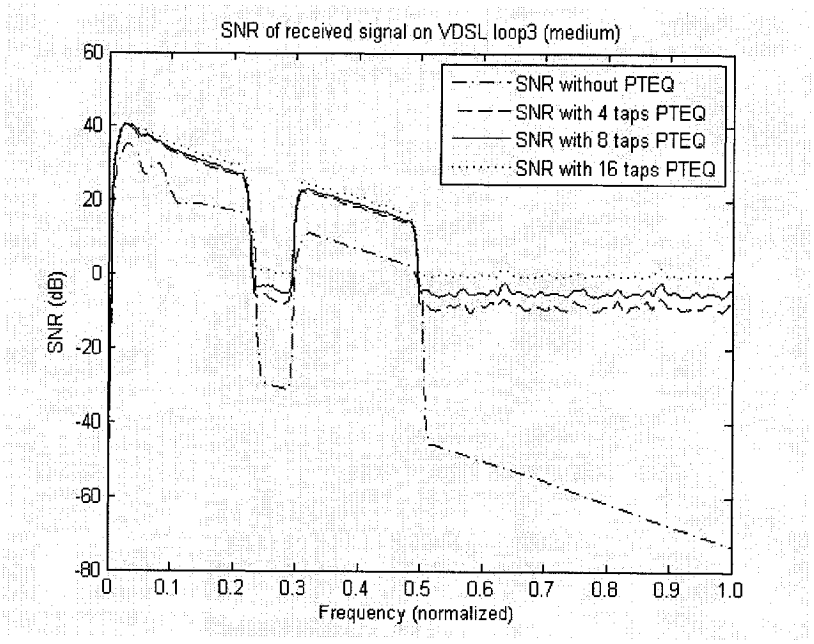


Figure 3.15 SNR of received signal on VDSL test loop 3 (medium) with noise model 2

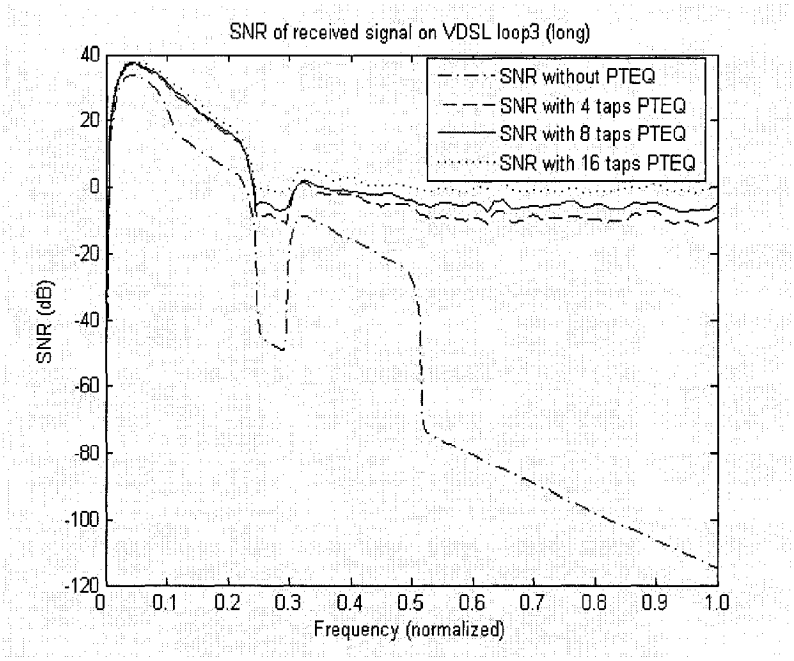


Figure 3.16 SNR of received signal on VDSL test loop 3 (long) with noise model 3

***Experiment 3**, Bit-rate of different test loops with different number of equalizer taps.*

Based on the water-filling algorithm (see Section 2.1.3), we can calculate the corresponding bit-rate with different loop conditions. A bit-rate table regarding different test loops and different number of taps is listed in Table 3.7, which is considered under

Table 3.7 Bit-rate of VDSL test loops with noise model 2

VDSL test loop number	Bit-rate (Mbps)			
	Without PTEQ	4-Tap PTEQ	8-Tap PTEQ	16-Tap PTEQ
1 (short TP1)	39.96	41.91	43.12	46.76
2 (short)	36.36	38.17	39.39	42.71
3 (medium)	17.72	26.79	29.84	32.77
4 (long)	1.01	3.22	3.628	4.608
5	40.66	42.48	43.82	47.44
6	10.82	18.84	20.14	23.49
7	3.53	8.12	8.48	9.5

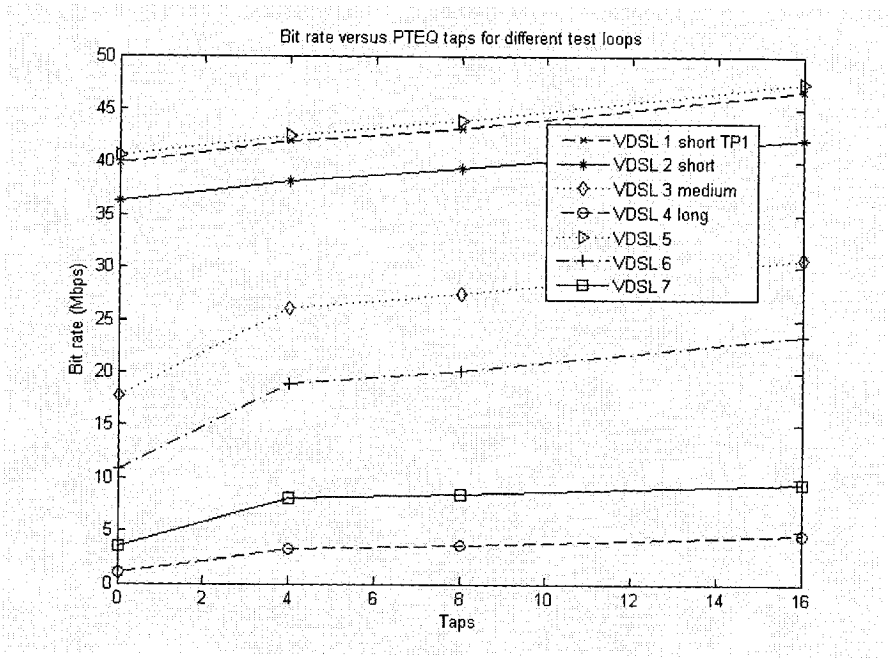


Figure 3.17 Bit-rate versus PTEQ taps for different VDSL test loops

the noise model 2 (see Table 3.6). The bit-rate plot versus the number of PTEQ taps corresponding to the parameter in Table 3.7 is shown in Figure 3.15.

From the above simulation results, some observations can be made:

1. A better bit-rate can be achieved for short loops than long ones. Since the attenuation is increased greatly with the length of loops, the achievable bit-rate of short loops could reach as high as 40 Mbps, whereas that of long loops is only about several Mbps. This observation complies with the VDSL system requirement, namely, a very-high speed can only be achieved on short loops (300m), which implies that a longer service range is possible at the price of reducing the bit-rate. In addition, the bit-rate on a long loop with bridge taps would be greatly reduced, e.g., the long VDSL test loop 4 can only achieve about 1 Mbps.
2. Bit-rate can be increased by employing PTEQ. Due to the compensation of the

channel attenuation and distortion, SNR on each tone is maximized, thus increasing the bit-rate. Figure 3.14 through Figure 3.16 have clearly indicated that after the equalization, the SNR of the received signal is significantly improved, and it exhibits a nearly constant within a certain range of frequency. Thus, a better transmission bit-rate is obtained according to the water-filling bit loading method.

3. Using more PTEQ taps would increase the bit-rate. As we can see in Table 3.7, the bit-rate is becoming higher when more PTEQ taps are applied. However, short loops are improved less compared to long loops. For example, the bit-rate on short length VDSL test loop 2 is raised about 17.5% after applying a 16-tap PTEQ, while the bit-rate on long length VDSL test loop 4 is raised about 360% after applying a 16-tap PTEQ. This is because the channel attenuation of short loops is not as high as that of long loops, which results in a larger improvement space for long loops. To compensate for this large attenuation and distortion on long loops, a large number of equalizer taps is needed, and thus, for a given bit-rate goal, less equalization taps are required for a short loop comparing to a long loop.

3.4 Conclusions

In this chapter, we have discussed the per-tone equalization (PTEQ) technique and various VDSL test loops, and simulated the SNR and bit-rate of VDSL systems using the PTEQ. The mathematic bases of PTEQ and its structure have been presented. Then, the complexity of PTEQ during data transmission has been analyzed. It is drawn that the complexity of PTEQ is comparable to that of other TEQ/FEQ methods. In order to

evaluate the performance of PTEQ in VDSL, the frequency response of seven VDSL test loops have been obtained by means of the two-port modeling. The simulation results of these loops have also been given. Furthermore, the training signals, noises and other parameters that are required to construct a practical VDSL system were also discussed. Simulation results of VDSL using PTEQ are given, including the PSD of the transmitted and received signals, the SNR and the achievable bit-rate. It has been shown that the PTEQ is able to well compensate the channel attenuation and distortion, thus resulting in a higher transmission bit-rate. Simulation results regarding various VDSL test-loops and different number of taps in PTEQ were also listed. The bit-rate for a short length loop is not increased as significantly as it is for a long loop. It has also been pointed out that the complexity of PTEQ is similar to that of TEQ/FEQ during the transmission of data. However, more computational effort is needed to initialize the PTEQ parameters. To further the initialization cost of PTEQ, adaptive equalization techniques are required, which will be discussed in the next chapter.

Chapter 4

Adaptive per-tone equalization in VDSL

In the previous chapter, we have discussed the PTEQ in a VDSL system. During the system's setup stage, a training sequence can be transmitted to initialize the PTEQ tap weights, trying to maximize the whole system transmission bit-rate. After the initialization, the PTEQ tap weights are determined and fixed, and the system steps into data transmission stage. We have shown that, during data transmission, the computational complexity of PTEQ is similar to that of the traditional TEQ/FEQ. However, the PTEQ initialization based on the MMSE criterion is a heavy burden. As seen from equation (3.13), in order to maximize the SNR of each tone, a computation of the auto correlation $\sigma_{u,n}^2$ and the crosscorrelation $\sigma_{u\tilde{x},n}$ is required. Since the computation is based on the entire received training signal during initialization stage, a large computational complexity as well as memory storage is required.

In this chapter, adaptive algorithms are used to reduce the computational complexity of the initialization of PTEQ. The equalizer tap weights are updated based on the previous equalizer tap weights as new training data come in, thus avoiding the need to store and compute all available data. Furthermore, the channel and noise characteristics of a VDSL system may vary with time due to the changing weather conditions, temperature, crosstalk and RFI. An adaptive equalizer in data-driven mode is capable of tracking these changing environments in data transmission stage, and thus keeps the system performance as high as possible [33]. In this chapter, we focus on two adaptive

equalization algorithms, namely, the least-mean-square (LMS) and the recursive least-squares (RLS). In particular, the normalized LMS (NLMS) algorithm and QR-based RLS algorithm for the initialization process are presented in detail. Performance study of using these adaptive algorithms for several VDSL test loops is carried out, according to standard VDSL system configurations.

In Section 4.1, LMS adaptive PTEQ initialization is discussed. The basic LMS algorithm is given in Section 4.1.1. To overcome the gradient noise amplification problem in LMS [28], a modified LMS algorithm, which is called the normalized LMS (NLMS) is also investigated. The application of the NLMS algorithm to the initialization of PTEQ is presented in Section 4.1.2.

In Section 4.2, RLS adaptive PTEQ initialization is presented. The basic RLS algorithm as well as the square-root RLS (SR-RLS) algorithm developed to improve the numerical stability of the basic RLS, is presented in Section 4.2.1. The adaptive initialization process employing QR-RLS algorithm is given in Section 4.2.2.

In Section 4.3, a simulation study of both NLMS and QR-RLS for PTEQ initialization is carried out. The simulation is performed in the standard transmission environment introduced in Section 3.3.1. Various simulation results regarding different equalizer taps, iteration numbers and test loops are presented. A comparison of the performances of these two adaptive algorithms is also given in this section.

4.1 LMS-based PTEQ Initialization

4.1.1 LMS and NLMS algorithms

The least-mean-square (LMS) is a widely used adaptive algorithm due to its simplicity

and other important features. It requires neither the measurement of pertinent correlation functions, nor the matrix inversion. The relatively low computational complexity in comparison to other adaptive algorithms makes it a more feasible and cost-effective method in practical applications.

Figure 4.1 gives a common structure of an adaptive equalizer, and Figure 4.2 illustrates

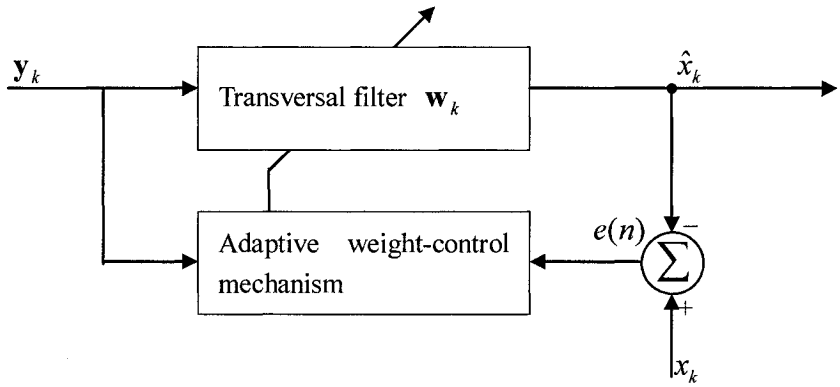


Figure 4.1 A commonly used equalizer structure

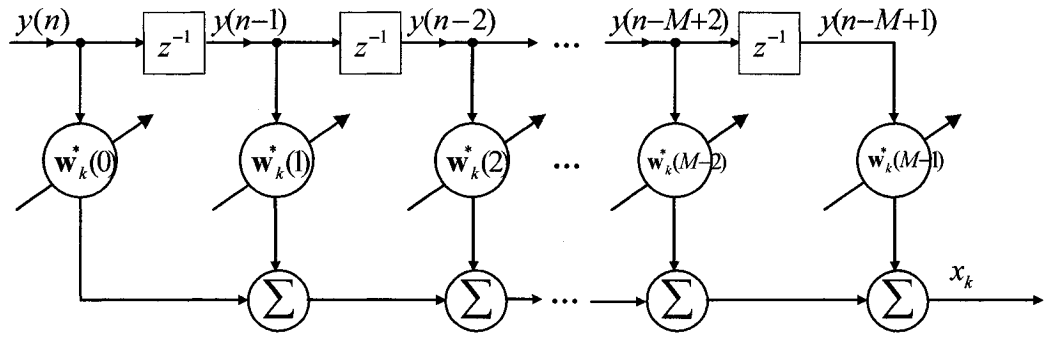


Figure 4.2 The structure of transversal filter

the structure of a transversal filter. In general, an adaptive filter consists of two processes:

1. Filtering process. The input signal is fed into a transversal filter containing a set of taps to generate an estimate of the transmitted signal. An estimation error is

computed by comparing the filter output to the desired signal.

2. Adapting process. It is to adjust the tap weights in accordance with the estimation error and the input signal, based on a specific algorithm.

The LMS algorithm can be described as three basic relations:

Filter output:

$$\hat{x}_k = \mathbf{w}^H{}_k \mathbf{y}_k \tag{4.1}$$

Estimation error:

$$e_k = x_k - \hat{x}_k \tag{4.2}$$

Tap weights adaptation:

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu \mathbf{y}_k e^*{}_k \tag{4.3}$$

An estimate of the desired signal x_k, denoted as $\hat{x}_k$, is generated by filtering the input signal $\mathbf{y}_k$ with a transversal filter $\mathbf{w}_k$. An estimation error e_k is then obtained by comparing the desired signal x_k with the estimated signal $\hat{x}_k$. The new tap-weight vector $\mathbf{w}_{k+1}$ is got by modifying the current tap weights. The second term on the right side of (4.3), $\mu \mathbf{y}_k e^*{}_k$, represents the adjustment that is made to the current tap-weight vector $\mathbf{w}_k$. The parameter μ is called step-size which controls the adjustment speed and in turn the convergence speed of the tap-weight vector as the iteration proceeds from one cycle to the next. The iterative procedure is normally started with an initial guess $\mathbf{w}_0$.

In the above standard LMS algorithm, the correction term $\mu \mathbf{y}_k e^*{}_k$ applied to the tap-weight vector $\mathbf{w}_k$ at iteration $k+1$ is directly proportional to the tap-input signal $\mathbf{y}_k$. As a result, when $\mathbf{y}_k$ is large, a gradient noise amplification will be introduced. To

overcome this problem, a modified LMS algorithm, namely, the normalized LMS (NLMS) algorithm can be employed, in which the correction term is normalized with respect to the squared Euclidean norm of the tap-input signal at iteration k.

The NLMS algorithm can be viewed as the solution to a constrained optimization problem [34], which leads the tap-weight adaptation to

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \frac{\tilde{\mu}}{\|\mathbf{y}_k\|_2^2}\mathbf{y}_k e^*_k \tag{4.4}$$

where $\tilde{\mu}$ is a positive real scaling factor to control the step-size from one iteration to the next, and $\|\mathbf{y}_k\|_2^2$ is the Euclidean norm of the tap-input signal $\mathbf{y}_k$. It is clearly seen from (4.4) that the product vector $\mathbf{y}_k e^*_k$ is normalized with respect to $\|\mathbf{y}_k\|_2^2$.

The NLMS in form of equation (4.4), however, still has a numerical difficulty, namely, when the tap-input signal $\mathbf{y}_k$ is small, $\mathbf{y}_k e^*_k$ will be divided by the small squared norm $\|\mathbf{y}_k\|_2^2$. To overcome this problem, a slight modification is made to equation (4.4), giving:

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \frac{\tilde{\mu}}{a + \|\mathbf{y}_k\|_2^2}\mathbf{y}_k e^*_k \tag{4.5}$$

where $a > 0$ is a small constant. The NLMS algorithm is convergent in the mean-square sense if $0 < \tilde{\mu} < 2$, as shown in [36]. In addition, the normalized LMS algorithm exhibits a rate of convergence that is potentially faster than that of the basic LMS algorithm [35]. Thus, we use the NLMS algorithm instead of the basic LMS algorithm for PTEQ initialization, as discussed in the next section.

4.1.2 NLMS-based PTEQ initialization

Based on the NLMS algorithm introduced in the previous section, an adaptive method to initialize the PTEQ tap-weight vector can be developed as follows.

The estimate of the transmitted training signal of tone n, as given in (3.11), is a linear combination of the equalizer tap vector $\mathbf{v}_n$ and the tap-input signal vector $\mathbf{u}_{k,n}$, which can be rewritten in the iteration form as:

$$\hat{\tilde{x}}_{k,n} = \mathbf{v}^H{}_{k,n}\mathbf{u}_{k,n} = \mathbf{v}^H{}_{k,n}[\Delta\mathbf{y}_k^T \quad \tilde{y}_{k,n}]^T \tag{4.6}$$

where the subscript k has been added to $\mathbf{v}_n$, representing the tap weights at the k-th iteration. The estimation error of the transmitted signal can then be given as

$$e_{k,n} = \tilde{x}_{k,n} - \hat{\tilde{x}}_{k,n} \tag{4.7}$$

Using (4.7), the tap-weight vector can be updated as follows

$$\mathbf{v}_{k+1,n} = \mathbf{v}_{k,n} + \frac{\tilde{\mu}}{a + \left\|\mathbf{u}_{k,n}\right\|_2^2}\mathbf{u}_{k,n}e^*{}_{k,n} \tag{4.8}$$

Note that the norm of the input vector $\left\|\mathbf{u}_{k,n}\right\|_2^2$ can be calculated from the norm of the tone-independent difference terms $\Delta\mathbf{y}_k$ and the norm of the DFT output $\tilde{y}_{k,n}$. The first norm $\left\|\Delta\mathbf{y}_k\right\|_2^2$ needs to be computed only once for all tones at each iteration. The NLMS-based PTEQ initialization process is summarized in Table 4.1.

The parameter K in Table 4.1 denotes the maximum iteration number, i.e., the number of training symbols transmitted at setup phase. The simulation of this adaptive algorithm for PTEQ in VDSL systems will be given at Section 4.3. A major problem of NLMS is the low convergence speed [28]. In the next section, the recursive least-squares (RLS) algorithm will be presented, which has a much higher convergence speed at the expense of more computational burden.

Table 4.1 NLMS-based PTEQ initialization process

Initialize PTEQ tap-weight vector $\mathbf{v}_{0,n}$ for each tone n

For $k = 0,1,2\cdots\cdots K$

Compute $\Delta\mathbf{y}_k^T$ and $\tilde{y}_{k,n}$, Construct $\mathbf{u}_{k,n} = [\Delta\mathbf{y}_k^T \quad \tilde{y}_{k,n}]^T$

Compute the norm of tone-independent difference terms

$$A = \|\Delta\mathbf{y}_k\|_2^2 \tag{4.9}$$

For each tone n

Compute the equalized output, as given by (4.6)

$$\hat{\tilde{x}}_{k,n} = \mathbf{v}^H{}_{k,n}\mathbf{u}_{k,n}$$

Compute the estimation error, as given by (4.7)

$$e_{k,n} = \tilde{x}_{k,n} - \hat{\tilde{x}}_{k,n}$$

Compute the norm of the DFT output $\tilde{y}_{k,n}$

$$B = \tilde{y}_{k,n}^*\tilde{y}_{k,n} \tag{4.10}$$

Update the tap-weight vector according to (4.8)

$$\mathbf{v}_{k+1,n} = \mathbf{v}_{k,n} + \frac{\tilde{\mu}}{a + \|\mathbf{u}_{k,n}\|_2^2}\mathbf{u}_{k,n}e^*{}_{k,n} = \mathbf{v}_{k,n} + \frac{\tilde{\mu}}{a + A + B}\mathbf{u}_{k,n}e^*{}_{k,n} \tag{4.11}$$

End

End

4.2 RLS-based PTEQ Initialization

The number of training symbols transmitted during setup phase is limited due to the constraint initialization time for practical VDSL system. As a consequence, a fast converge adaptive algorithm is desired. The RLS adaptive PTEQ initialization can then

be used instead to overcome the slow convergence of NLMS. In this section, we will first discuss the RLS algorithm as well as the square-root RLS (SR-RLS) algorithm in Section 4.2.1. An adaptive QR-RLS based PTEQ initialization technique is then given in Section 4.2.2.

4.2.1 RLS and SR-RLS algorithm

Suppose at time index k, an input data vector $\mathbf{y}_k$ is fed into a transversal filter $\mathbf{w}$. An estimate of the desired signal x_k, denoted as $\hat{x}_k$, is generated at the filter output. The estimation error e_k is then given by $e_k = x_k - \hat{x}_k$. From the least squares criterion, the goal is to find the tap weights of the transversal filter, such that the following total squared error

$$\begin{aligned} \mathbb{C}(\mathbf{w}) &= \sum_{i=0}^{k} |e_i|^2 \\ &= \sum_{i=0}^{k} |x_i - \mathbf{y}_i \mathbf{w}|^2 \end{aligned} \tag{4.12}$$

is minimized. In recursive implementation of the least squares method, we need to use the input data at new iteration to update the estimate of the transmitted signal. Also, it is natural to give a larger weight to the most recent error. Therefore, the cost function is modified as

$$\begin{aligned} \mathbb{C}(\mathbf{w}) &= \sum_{i=0}^{k} \lambda^{k-i} |e_i|^2 \\ &= \sum_{i=0}^{k} \lambda^{k-i} |x_i - \mathbf{y}_i \mathbf{w}_k|^2 \end{aligned} \tag{4.13}$$

where an exponential weighting factor $\lambda^{k-i}, i = 1, 2 \cdots k; 0 < \lambda < 1$ is introduced to mitigate the contribution of the past data, which affords the possibility of following the

statistical variations of the data when the filter operates in a non-stationary environment [28]. As such, λ is often called the forgetting factor. Note that the tap-weight vector $\mathbf{w}$ in (4.12) is modified to $\mathbf{w}_k$ in (4.13), which represents the tap-weight vector at iteration k. $\mathbf{w}_k$ remains fixed during the iteration for the computation of the cost function $\mathbb{C}(\mathbf{w})$ as given in (4.13). The optimum value of the tap-weight vector $\mathbf{w}_k$, for which the cost function $\mathbb{C}(\mathbf{w})$ attains its minimum value, is defined by the equation in the matrix form:

$$\mathbf{\Phi}_k \mathbf{w}_k = \mathbf{D}_k \tag{4.14}$$

where $\mathbf{\Phi}_k$ is the correlation matrix of the tap-input signal defined by:

$$\mathbf{\Phi}_k = \sum_{i=1}^{k} \lambda^{k-i} \mathbf{y}_i \mathbf{y}^H{}_i \tag{4.15}$$

and $\mathbf{D}_k$ is the cross-correlation vector between the tap-input signal and the desired signal, defined by

$$\mathbf{D}_k = \sum_{i=1}^{k} \lambda^{k-i} \mathbf{y}_i x^*{}_i \tag{4.16}$$

The symbol H in (4.15) denotes the Hermitian transpose and the asterisk * in (4.16) denotes the complex conjugation. In order to compute the tap-weight vector $\mathbf{w}_k$ in the adaptive fashion, we need to find the recursive formation of $\mathbf{D}_k$ and $\mathbf{\Phi}^{-1}{}_k$. According to the definition of $\mathbf{D}_k$ in (4.16), it can be represented recursively as

$$\mathbf{D}_k = \lambda \mathbf{D}_{k-1} + \mathbf{y}_k x^*{}_k \tag{4.17}$$

The recursive computation of $\mathbf{\Phi}^{-1}{}_k$ can be obtained based a matrix inversion lemma [37], giving

$$\boldsymbol{\Phi}^{-1}{}_{k} = \lambda^{-1}\boldsymbol{\Phi}^{-1}{}_{k-1} - \frac{\lambda^{-2}\boldsymbol{\Phi}^{-1}{}_{k-1}\mathbf{y}_k\mathbf{y}^{H}{}_{k}\boldsymbol{\Phi}^{-1}{}_{k-1}}{1+\lambda^{-1}\mathbf{y}^{H}{}_{k}\boldsymbol{\Phi}^{-1}{}_{k-1}\mathbf{y}_k} \tag{4.18}$$

To simplify the computation of (4.18), let

$$\mathbf{K}_k = \frac{\lambda^{-1}\boldsymbol{\Phi}^{-1}{}_{k-1}\mathbf{y}_k}{1+\lambda^{-1}\mathbf{y}^{H}{}_{k}\boldsymbol{\Phi}^{-1}{}_{k-1}\mathbf{y}_k} \tag{4.19}$$

which is called the Kalman gain. Also note that $\mathbf{K}_k = \boldsymbol{\Phi}^{-1}{}_{k}\mathbf{y}_k$, that is, the gain vector is defined as the tap-input signal vector $\mathbf{y}_k$ transformed by the inverse of the correlation matrix $\boldsymbol{\Phi}_k$. Using (4.19), (4.18) can be rewritten as:

$$\boldsymbol{\Phi}^{-1}{}_{k} = \lambda^{-1}\boldsymbol{\Phi}_{k-1} - \lambda^{-1}\mathbf{K}_k\mathbf{y}^{H}{}_{k}\boldsymbol{\Phi}^{-1}{}_{k-1} \tag{4.20}$$

Thus the tap-weight vector can be given by

$$\begin{aligned}\mathbf{w}_k &= \mathbf{w}_{k-1} + \mathbf{K}_k(x^*{}_k - \mathbf{y}^{H}{}_{k}\mathbf{w}_{k-1}) \\ &= \mathbf{w}_{k-1} + \mathbf{K}_k\xi^*{}_k\end{aligned} \tag{4.21}$$

With the equations (4.19) to (4.21), the RLS algorithm is well constructed. The adaptive computation can start from some initial values of $\mathbf{w}_0$ and $\boldsymbol{\Phi}^{-1}{}_{0}$. Comparing (4.21) and (4.3), it is clear that the tap weights update equations of RLS and LMS are similar. For both of them, the new tap weights are obtained from the previous tap weights plus one correction term, and the correction term is calculated based on the estimation error. However, there are two differences between them. In (4.3), the estimation error of the LMS algorithm is multiplied by $\mu\mathbf{y}_k$, while in the RLS, the estimation error is multiplied by the Kalman gain $\mathbf{K}_k$. Since $\mathbf{K}_k$ is a vector with the dimension equal to the number of taps M, each tap weight in effect is controlled by one of the elements of $\mathbf{K}_k$; whereas the LMS algorithm has only the single adjustable parameter μ. Consequently, a much faster convergence is obtained by the RLS algorithm. The other

difference is the computation of the estimated signal. In the LMS algorithm, the input signal $\mathbf{y}_k$ is equalized by the current tap-weight vector $\mathbf{w}_k$, i.e., $\mathbf{y}_k\mathbf{w}_k$; whereas in the RLS algorithm, the estimated signal is the output of input signal $\mathbf{y}_k$ equal ized by the previous tap-weight vector $\mathbf{w}_{k-1}$, i.e., $\mathbf{y}_k\mathbf{w}_{k-1}$. The estimation error denoted by ξ_k in (4.21) is thus sometimes called a priori estimation error.

The RLS algorithm stated above has a numerical robust problem [38]. More specifically, since $\mathbf{\Phi}_k$ denotes the correlation matrix of the tap-input signal, as shown in (4.15), the $\mathbf{\Phi}^{-1}{}_k$ is nonnegative definite. However, the recursive computation of $\mathbf{\Phi}^{-1}{}_k$ in (4.20) is computed as the difference between two nonnegative definite matrices, hence the resulting $\mathbf{\Phi}^{-1}{}_k$ may not be nonnegative definite any more. Unless the numerical accuracy employed at every iteration of the algorithm is high enough, unacceptable result may be introduced. This problem can be overcome by using some numerically stable transformations of the correlation matrix [40]. In particular, $\mathbf{\Phi}^{-1}{}_k$ can be propagated in a square-root form, by using the Cholesky factorization

$$\mathbf{\Phi}^{-1}{}_k = \mathbf{\Phi}^{-\frac{H}{2}}{}_k \mathbf{\Phi}^{-\frac{1}{2}}{}_k \tag{4.22}$$

where $\mathbf{\Phi}^{-\frac{1}{2}}{}_k$ is an upper triangular matrix, and $\mathbf{\Phi}^{-\frac{H}{2}}{}_k$ is its Hermitian transpose. The Cholesky factor $\mathbf{\Phi}^{-\frac{1}{2}}{}_k$ is commonly referred to as the square root of the matrix $\mathbf{\Phi}^{-1}{}_k$. Since the product of any square matrix and its Hermitian transpose is always positive definite, the nonnegative definite character of $\mathbf{\Phi}^{-1}{}_k$ is preserved. Consequently, the numerical robust of the RLS algorithm using the Cholesky factor $\mathbf{\Phi}^{-\frac{1}{2}}{}_k$ for recursive

computation is generally much better than that of using $\mathbf{\Phi}^{-1}_k$ itself. Similarly, one can also use the square-root of $\mathbf{\Phi}_k$ for the adaptive computation.

The square-root adaptive filtering algorithms for RLS are known as the QR-RLS algorithm [41], the extended QR-RLS [43], and the inverse QR-RLS [42]. They all use an orthogonal triangularization process known as the QR decomposition [37] on the correlation matrix of the tap-input signal $\mathbf{\Phi}_k$ or its inverse $\mathbf{\Phi}^{-1}_k$. In [39], a unified treatment of the QR-RLS algorithms for exponentially weighted RLS is given. In the next section, the initialization of PTEQ using the QR-RLS algorithm is presented.

4.2.2 QR-RLS based PTEQ initialization

Since the square-root RLS is numerically more stable, it is commonly more suitable for adaptive equalizers rather than the standard RLS algorithm. In this section, the initialization of PTEQ using the QR-RLS algorithm is presented.

As has been given in Section 4.1.2, an estimate of the transmitted training signal on tone n is obtained iteratively as a linear combination of the equalizer tap vector $\mathbf{v}_{k,n}$ and the tap-input vector $\mathbf{u}_{k,n}$, as given by:

$$\hat{\tilde{x}}_{k,n} = \mathbf{v}^H{}_{k,n}[\Delta\mathbf{y}_k^T \quad \tilde{y}_{k,n}]^T = \mathbf{v}^H{}_{k,n}\mathbf{u}_{k,n} \tag{4.23}$$

The correlation matrix of the tap-input signal is given by:

$$\begin{aligned}\mathbf{\Phi}_{k,n} &= \sum_{i=1}^{k} \lambda^{k-i}\mathbf{u}_{i,n}\mathbf{u}^H{}_{i,n} \\ &= \mathbf{R}_{k,n}\mathbf{R}^H{}_{k,n}\end{aligned} \tag{4.24}$$

where $\mathbf{R}_{k,n}$ is the lower triangular Cholesky factor of $\mathbf{\Phi}_{k,n}$. The QR-RLS adaptive

algorithm is based on iteratively updating the square root of $\mathbf{\Phi}_{k,n}$, i.e., the lower triangular matrix $\mathbf{R}_{k,n}$. Substituting the tap-weight vector $\mathbf{v}_{k,n}$ for $\mathbf{w}_k$ in (4.14) and multiplying both sides of the equation with $\mathbf{R}^{-1}{}_{k,n}$, a new vector can then be obtained as:

$$\mathbf{p}_{k,n} = \mathbf{R}^{H}{}_{k,n}\mathbf{v}_{k,n} = \mathbf{R}^{-1}{}_{k,n}\mathbf{D}_{k,n} \tag{4.25}$$

As given in the QR-RLS algorithm, a pre-array to post-array transformation of $\mathbf{R}_{k,n}$ can be found as [28]:

$$\begin{bmatrix} \lambda^{\frac{1}{2}}\mathbf{R}_{k-1,n} & \mathbf{u}_{k,n} \\ \lambda^{\frac{1}{2}}\mathbf{p}^{H}{}_{k-1,n} & \tilde{x}_{k,n} \\ \mathbf{0} & 1 \end{bmatrix} \Theta_{k,n} = \begin{bmatrix} \mathbf{R}_{k,n} & \mathbf{0} \\ \mathbf{p}^{H}{}_{k,n} & \xi_k \gamma^{\frac{1}{2}}{}_k \\ \mathbf{u}^{H}{}_{k,n}\mathbf{R}^{-H}{}_{k,n} & \gamma^{\frac{1}{2}}{}_k \end{bmatrix} \tag{4.26}$$

where $\Theta_{k,n}$ is a unitary rotation that operates on the elements of the tap-input data vector $\mathbf{u}_{k,n}$ in the pre-array, annihilating its elements one by one, so as to produce a block zero entry in the first row of the post-array. From (4.26), we can see the recursive relation between $\mathbf{R}_{k-1,n}$ and $\mathbf{R}_{k,n}$. Furthermore, with the updated block values of $\mathbf{R}_{k,n}$ and $\mathbf{p}^{H}{}_{k,n}$ in the post-array, the tap-weight vector $\mathbf{v}_{k,n}$ can be computed by

$$\mathbf{v}^{H}{}_{k,n} = \mathbf{p}^{H}{}_{k,n}\mathbf{R}^{-1}{}_{k,n} \tag{4.27}$$

Thus, as long as we can find a unitary rotation $\Theta_{k,n}$ that can zero out the tap-input data vector $\mathbf{u}_{k,n}$, an update formula for the lower triangular $\mathbf{R}_{k,n}$ can be well constructed. One method to generate the unitary rotation $\Theta_{k,n}$ is through the product of a series of Givens rotations [28], each of which is given as

$$\mathbf{Q}_m = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & \cos\phi_m & \cdots & 0 & \cdots & e^{j\psi}\sin\phi_m \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & 1 & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & -e^{-j\psi}\sin\phi_m & \cdots & 0 & \cdots & \cos\phi_m \end{bmatrix} \tag{4.28}$$

where the Givens rotation $\mathbf{Q}_m$, $(m = 0,1\cdots M-1)$ zeroes out the m-th element of the vector $\mathbf{u}_{k,n}$. Thus, through the equations (4.23) to (4.28), an adaptive computation can start from $\mathbf{R}_{0,n}$ and $\mathbf{p}_{0,n}$ for each tone n. The QR-RLS based PTEQ initialization process is summarized in Table 4.2.

As shown in (4.26), the QR-RLS algorithm operate directly on the tap-input data $\mathbf{u}_{k,n}$. In [11], an inverse QR-RLS algorithm based PTEQ initialization method is given, which can be easily implemented with a systolic structure. However, in the inverse QR-RLS the tap-input data vector $\mathbf{u}_{k,n}$ is transformed by the $\mathbf{\Phi}^{-\frac{1}{2}}_{k,n}$ before the recursive calculation, resulting in a more computational complexity.

The $\mathbf{u}_{k,n}$ can be split into the tone-independent difference terms $\Delta\mathbf{y}_k$ and the DFT output $\tilde{y}_{k,n}$ of each tone. By virtue of this, a split SR-RLS algorithm [10] can be used, such that the memory and computational cost is reduced, on the other hand, its convergence speed of is lower than that of the full SR-RLS algorithm.

Table 4.2 QR-RLS based PTEQ initialization process

Initialize $\mathbf{R}_{0,n}$ and $\mathbf{p}_{0,n}$ for each tone n

For $k = 0,1,2\cdots\cdots K$

Compute $\Delta\mathbf{y}_k^T$ and $\tilde{y}_{k,n}$, Construct $\mathbf{u}_{k,n} = [\Delta\mathbf{y}_k^T \quad \tilde{y}_{k,n}]^T$

For each tone n

For $m = 1,2\cdots M$, determine Givens rotations $\mathbf{Q}_1\cdots\mathbf{Q}_M$, such that

$$\begin{bmatrix} \mathbf{R}_{k,n} & \mathbf{u}_{k,n} \\ \mathbf{p}^H{}_{k,n} & \tilde{x}_{k,n} \end{bmatrix} \mathbf{Q}_1\cdots\mathbf{Q}_M = \begin{bmatrix} \mathbf{R}_{k+1,n} & \mathbf{0}_{M\times 1} \\ \mathbf{p}^H{}_{k+1,n} & \delta \end{bmatrix} \tag{4.29}$$

Apply exponential weighting to $\mathbf{R}_{k+1,n}$ for next iteration

$$\mathbf{R}_{k+1,n} = \frac{\mathbf{R}_{k+1,n}}{\lambda^{\frac{1}{2}}} \tag{4.30}$$

Apply exponential weighting to $\mathbf{p}^H{}_{k+1,n}$ for next iteration

$$\mathbf{P}_{k+1,n} = \frac{\mathbf{P}_{k+1,n}}{\lambda^{\frac{1}{2}}} \tag{4.32}$$

Update the tap-weight vector

$$\mathbf{v}^H{}_{k+1,n} = \mathbf{p}^H{}_{k+1,n}\mathbf{R}^{-1}{}_{k+1,n} \tag{4.33}$$

End

End

4.3 Simulation of Adaptive PTEQ Initialization for VDSL

In this section, the adaptive per-tone equalization technique is investigated through computer simulations. Two adaptive algorithms, namely, the NLMS and QR-RLS, are considered in the simulation. The simulation is performed on several VDSL test loops as given in Section 3.2, as well as using standard system conditions specified in the ANSI standard T1.424. Configurations of the simulation system are given in Section 4.3.1. In Section 4.3.2, we present the simulation results, including the convergence behaviour of the two algorithms, the convergence speed versus equalizer taps, and the adaptive initialization performance for different VDSL channel models.

4.3.1 Simulation configurations

As has been introduced in Section 3.3.1, during the adaptive initialization of PTEQ for a VDSL system, a prescribed training sequence is sent by the transmitter, which can be further categorized into the non-periodic MEDLEY signal and the periodic REVERB signal. A PSD mask applies on the transmitted signal by means of multiplying a spectrum shaping coefficient to each tone. The received signal is obtained through the convolution of the channel impulse response and the transmitted signal plus some channel noises. Three channel noise models are introduced in the simulation, as given in Table 3.6, representing the range test, RFI test and crosstalk test, respectively. Based on the received noisy signal, the PTEQ tap weights can be computed iteratively according to the adaptive algorithms. In our simulation, the NLMS and QR-RLS algorithms are adopted, whose

training processes have already been presented in Table 4.1 and Table 4.2, respectively. The parameters for the simulation of adaptive PTEQ initialization in VDSL systems are summarized in Table 4.3. The length of the training sequence is set to 512, i.e., 512 DMT symbols. The number of training symbols is set according to the standard and it is suitable for the computer simulation cost. The value of the forgetting factor λ (see

Table 4.3 Simulation parameters of adaptive PTEQ initialization for VDSL

Number	Description	Parameter
1	Training sequence length	512
2	Tones	4096
3	DFT/IDFT size	8192
4	Cyclic prefix length	640
5	DMT symbols per second	4000
6	Tone spacing	4.3125 kHz
7	Coding gain	4.2 dB
8	Noise Margin	6 dB
9	Total transmitter power	14.5 dBm
10	Effective downstream tones	32~865 and 1210~1966
11	Background noise PSD	-140 dBm/Hz
12	Bits per tone	2~15 bits
13	Duplexing method	Frequency division duplexing (FDD)
14	Forgetting factor λ of RLS	0.99
15	Step size $\bar{\mu}$ of NLMS	0.1

Section 4.2) for the RLS initialization is chosen as 0.99. The step size $\bar{\mu}$ in the NLMS initialization is set to 0.1. Other parameters used in the simulation system have been described in Section 3.3.1. Also, synchronization is assumed perfect in the simulation such that the delay of received signal and transmitted signal is omitted. Simulation results based on these system configurations are depicted in the next section.

4.3.2 Simulation results

In this section, various simulation results are presented. First, we investigate the convergence behaviour of the NLMS and the QR-RLS based schemes based on two different training sequences, i.e., the MEDLEY and REVERB signals. Then, for both algorithms, the SNR evolution of different tones is depicted. As a performance measure for the adaptive equalization, the convergence in terms of bit-rate is presented afterwards, the relation between the convergence speed and equalizer taps is shown as well. Finally, we illustrate the performance of the adaptive algorithms for a wide range of VDSL test loops.

***Experiment 1**, Convergence behaviour based on different training sequence.*

As we have introduced previously, the REVERB training signal is periodic with the DMT

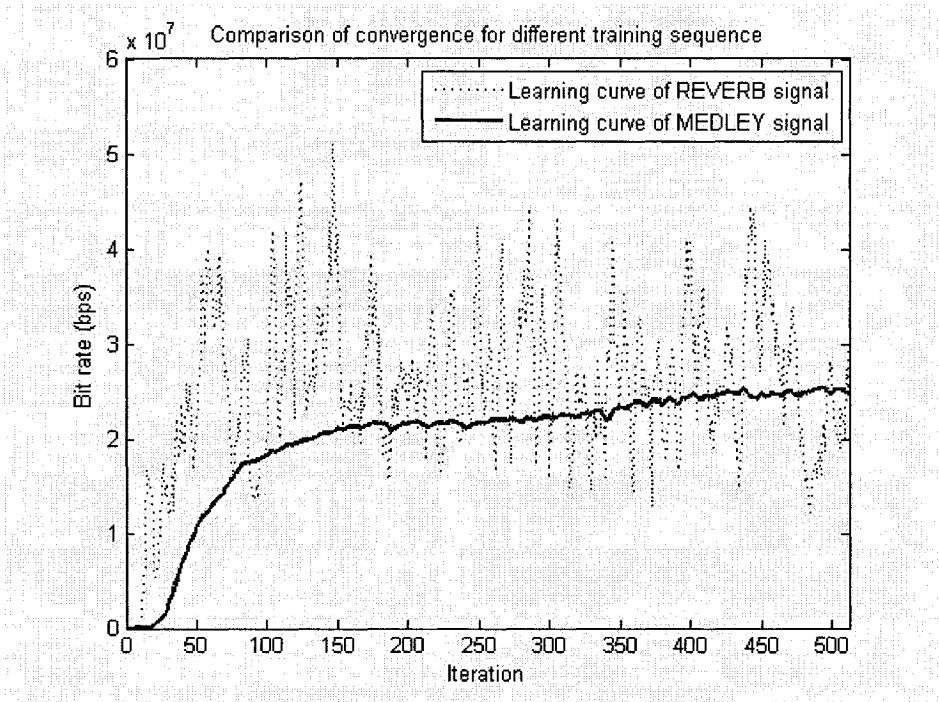


Figure 4.3 Comparison of different training sequences for the NLMS-based PTEQ initialization of VDSL test loop 2 (medium) with 8 taps and noise model 1

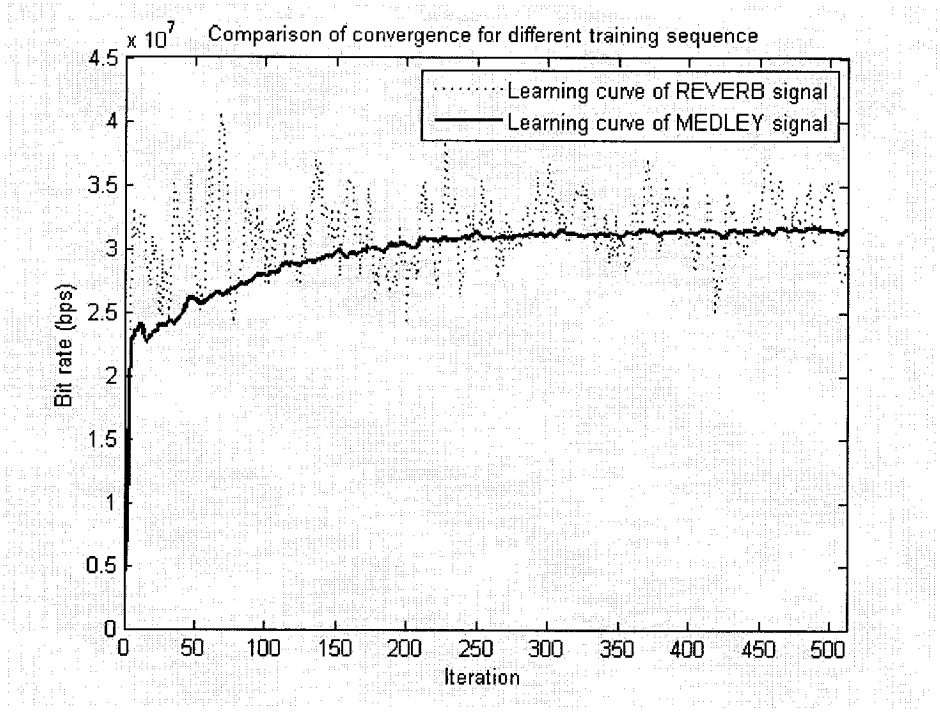


Figure 4.4 Comparison of different training sequences for the QR-RLS-based PTEQ initialization of VDSL test loop 2 (medium) with 8 taps and noise model 1

symbol length. which means that each tone always has the same two bits assigned to it for successive DMT symbols. Whereas the non-periodic MEDLEY signal brings no correlation between the two bits mapped on tone j during the m-th DMT symbol and those mapped on the same tone during the $m+1$-th DMT symbol. Figure 4.3 and Figure 4.4 depict the bit-rate convergence plots of NLMS and QR-RLS, respectively, based on the two different training sequences. They clearly show that the MEDLEY signal archives reasonable convergence behaviour while the REVERB signal cannot obtain a stable and smooth learning curve. This is because the bit loading algorithm is based on the assumption that the signal and noise are both white. In other words, there should be no correlations between the signal and noise. Therefore, we use the MEDLEY training sequence for the consequent simulations to give a proper performance study.

***Experiment 2**, SNR evolution of adaptive PTEQ initialization.*

Figure 4.5 and Figure 4.6 depict the SNR evolution during the update of the PTEQ tap weights when using the NLMS and the QR-RLS, respectively. The simulation is performed for the VDSL test loop 2 (medium) with an 8 taps PTEQ and noise model 1. The plots indicate that for both schemes, as more iteration goes, the SNR on the effective transmission tones become higher. Thus, a more desirable shape of the SNR is obtained, rendering a better transmission performance of the system.

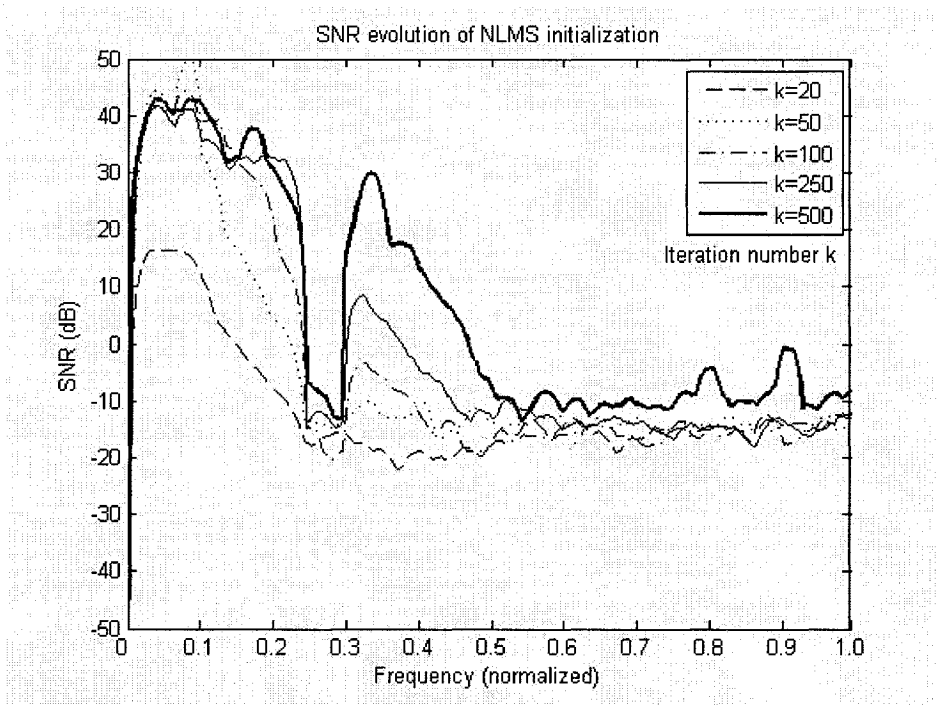


Figure 4.5 Evolution of the SNR during the NLMS-based PTEQ initialization of VDSL test loop 2 (medium) with 8 taps and noise model 1

***Experiment 3**, Bit-rate convergence of adaptive PTEQ initialization.*

The convergence behaviour in terms of the overall transmission bit-rate for an 8-tap PTEQ is depicted in Figure 4.7. The VDSL test loop 3 (medium length) is employed in conjunction with the channel noises of model 2 as given in Table 3.6. The bit-rate of the

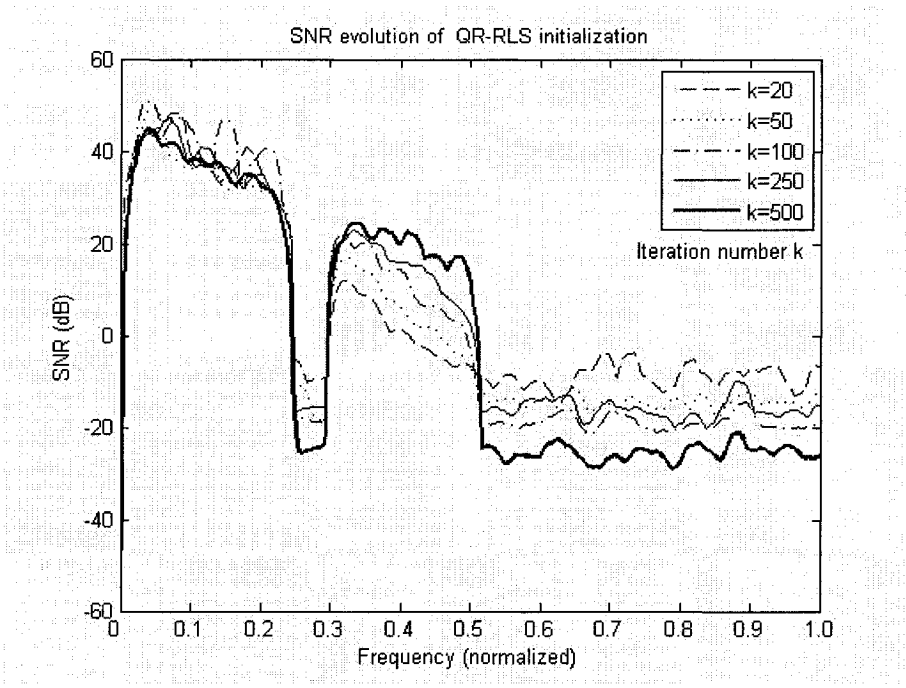


Figure 4.6 Evolution of the SNR during the QR-RLS-based PTEQ initialization of VDSL test loop 2 (medium) with 8 taps and noise model 1

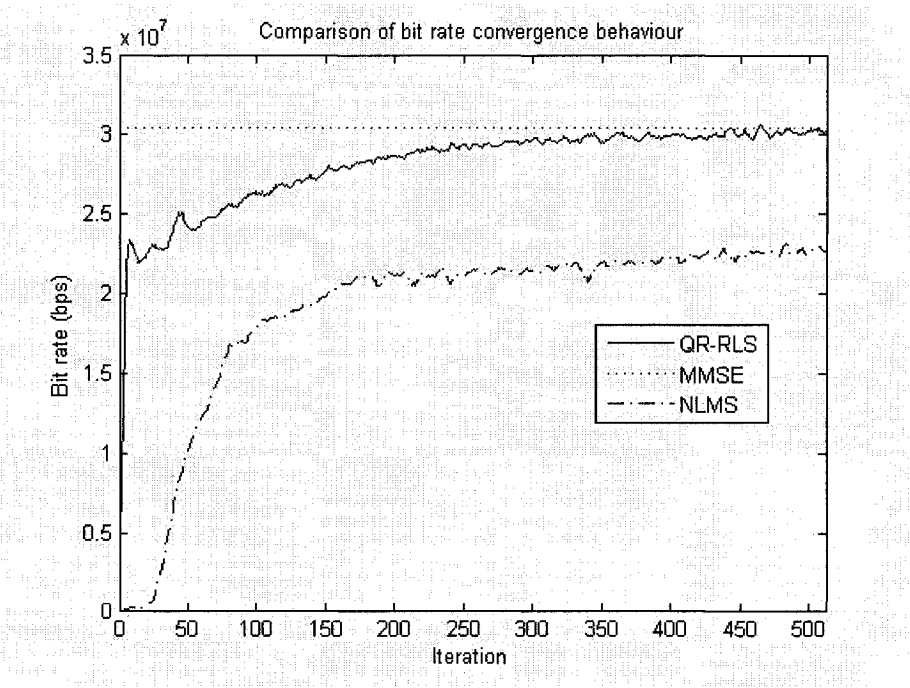


Figure 4.7 Comparison of different bit-rate convergence curve for the PTEQ initialization of VDSL test loop 3 (medium) with 8 taps and noise model 1

MMSE solution resulting from a 512 length training sequence is shown in the figure as an upper bound for the comparison of the adaptive algorithms. The plot clearly shows that the convergence of the RLS algorithm is much faster than that of the NLMS algorithm. The overall transmission bit-rate is calculated based on the SNR of each tone, using the Water-filling algorithm given in Section 2.1.3. The performance of bit-rate incorporates the tone-dependent SNR into one measure representing all the tones. Figure 4.8 depicts the bit-rate convergence of different noise models for the NLMS and the QR-RLS initialization schemes, which is simulated for medium length VDSL test loop 3. It is shown that the noise model 3 achieves the fastest bit-rate and the noise model 2 leads to the lowest one. For all the noise models, the QR-RLS achieves a faster convergence

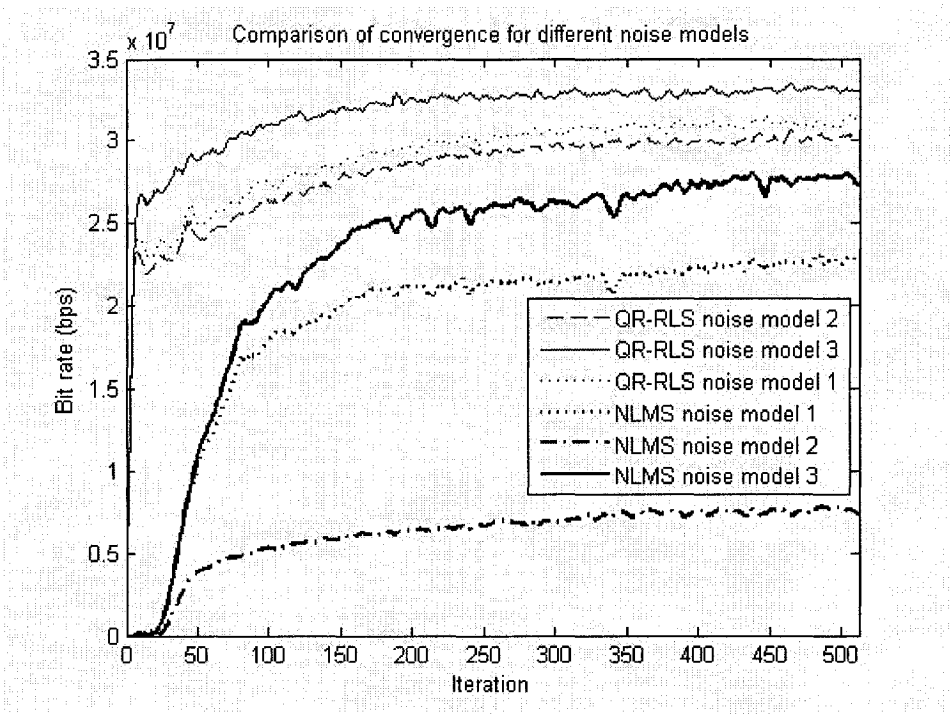


Figure 4.8 Comparison of bit-rate convergence of different noise models for the NLMS and QR-RLS -based PTEQ initialization of VDSL test loop 3 (medium), with 8 taps

than the NLMS method. In addition, the learning curve of noise model 2 is much lower than noise model 1 for NLMS, which implies that the increased channel noise reduces the convergence speed of NLMS.

***Experiment 4**, Comparison of convergence speed of different number of PTEQ taps.*

Figure 4.9 depicts the bit-rate learning curve for the NLMS and QR-RLS initialization processes. The plots present the convergence speed for VDSL test loop3 (medium), noise model 2, when the number of adaptive filters is 4 and 8. It is clearly seen that the convergence speed of NLMS is reduced as the number of taps increases. For the QR-RLS algorithm, the learning curve of 4-tap PTEQ and that of 8-tap PTEQ just have a slightly

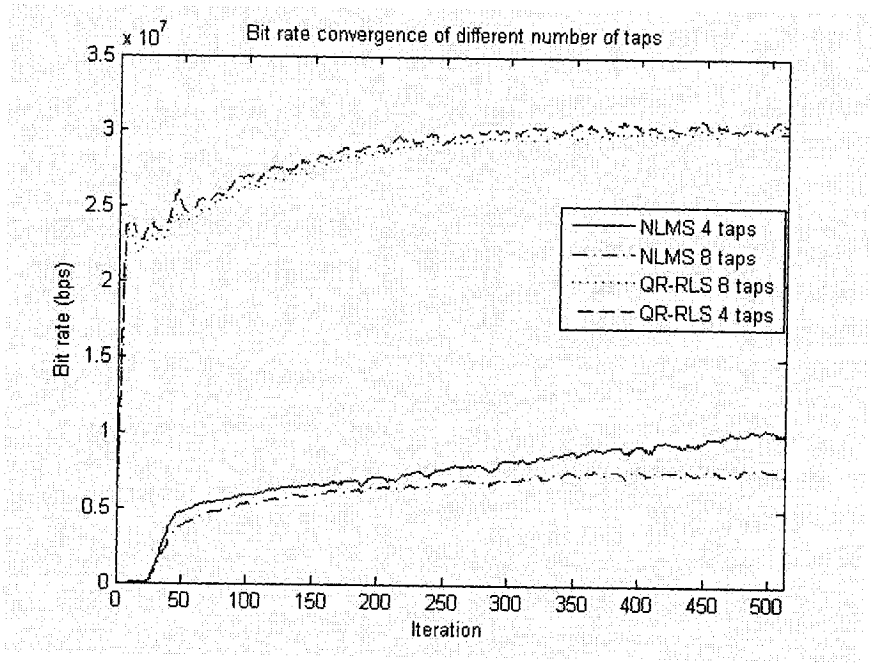


Figure 4.9 Bit-rate convergence of VDSL test loop 3 (medium) using NLMS and QR-RLS for PTEQ initialization with noise model 2

difference. As we have stated in Table 3.7, the PTEQ with a larger number of taps will achieve a higher maximum bit-rate. As a consequence, the convergence speed of QR-RLS is also reduced as the number of taps increases. In order to clearly show the

relation between the convergence speed and the number of taps, Figure 4.10 depicts the number of training symbols is required as a function of the number of taps, such that the bit-rate can reach a specific percentage of the MMSE solution. Due to the different convergence speed of QR-RLS and NLMS, we choose the target bit-rate as 98% and 70% of the MMSE bit-rate for the QR-RLS and NLMS, respectively. The number of PTEQ taps ranges from 4 to 16. It is clear from this figure that the convergence speed decreases with the increase of the number of taps. Hence, besides the consideration of memory and computational cost, the convergence speed is another factor that affects the choice of an appropriate number of taps

***Experiment 5**, Bit-rate convergence of different VDSL test loops.*

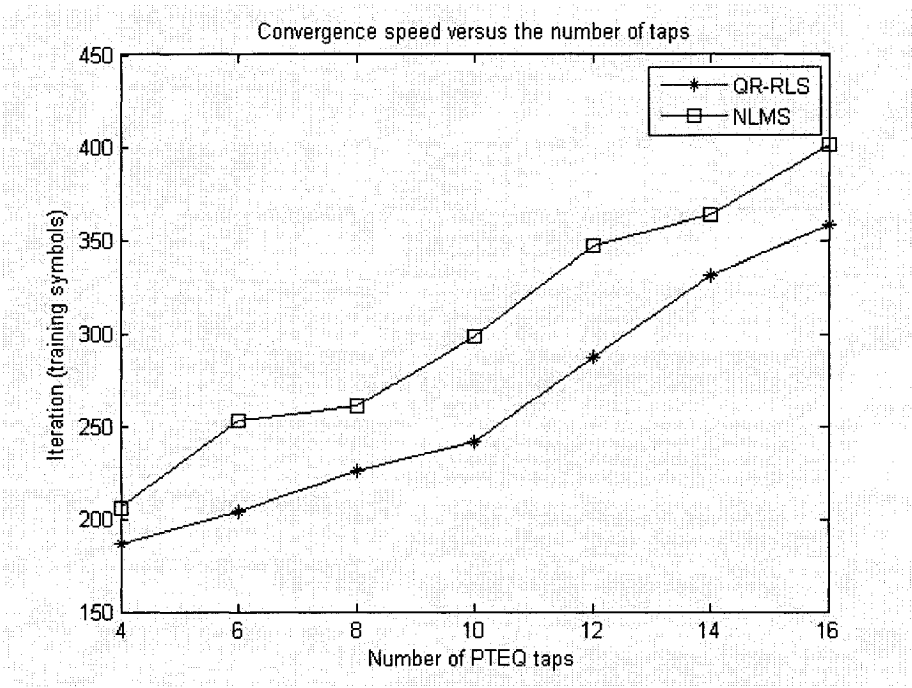


Figure 4.10 Convergence speed of QR-RLS and NLMS as a function of the number of PTEQ taps for VDSL test loop 3 (medium) with noise model 1

In order to illustrate the performance of the NLMS and QR-RLS initialization for various VDSL channels, simulation is carried out for the VDSL test loops 1 to 7. Table 4.4

indicates the number of iterations required for QR-RLS to reach 98% of the bit-rate obtained from the MMSE PTEQ, and that for NLMS to reach 70% of MMSE bit-rate. The noise model 1 is used in this simulation, and the number of PTEQ taps is set to 8. It

Table 4.4 Convergence behaviour of QR-RLS and NLMS for VDSL test loops 1 to 7 (noise model 1, 8 taps)

VDSL test loop number	Symbols to reach 98% of MMSE bit-rate of RLS	Symbols to reach 70% of MMSE bit-rate of NLMS	MMSE bit-rate (Mbps)
1 (short TP1)	212	206	45.14
2 (short)	236	225	41.36
3 (medium)	262	232	30.44
4 (long)	224	212	3.79
5	284	276	45.32
6	252	237	21.69
7	211	215	8.65

is shown in this figure that the transmission speed converges for all the VDSL test loops. For the QR-RLS initialization process, the transmission speed on all the VDSL test loops is convergent within 512 iterations.

From the simulation results given above, we can see that both the QR-RLS and the NLMS algorithms can well initialize the PTEQ tap weights for different VDSL test loops. However, the QR-RLS algorithm yields a much faster convergence speed than the NLMS algorithm. Within a training sequence of 512 symbols, all VDSL loops can almost achieve the MMSE bit-rate by the QR-RLS initialization, whereas the NLMS initialization can only achieve about one-half of the MMSE bit-rate. In addition, the convergence speed of both of them decreases with the increase of the number of taps, which imposes another factor on the choices of the equalizer tap size.

4.4 Conclusions

In this chapter, adaptive algorithms have been discussed for the initialization of PTEQ in VDSL systems to reduce the memory and computational cost. In particular, two adaptive algorithms, namely, the NLMS and the QR-RLS have been investigated for applications in VDSL equation. It has been shown that the NLMS can overcome the gradient noise amplification problem of the standard LMS algorithm and have a better convergence performance. The QR-RLS is one of the square-root RLS algorithms, which is numerically more stable than the original RLS algorithm, because of the factorization of the correlation matrix of the tap-input signal. Two initialization processes for the PTEQ are presented, based on the NLMS and QR-RLS algorithms, respectively. Simulations using these initialization processes in the VDSL systems have been conducted, where the configurations as specified in the standard are adopted in the simulation system. A comparison of the convergence speed based on these two initialization processes is carried out, in which the bit-rate of QR-RLS shows a much faster convergence than that of the NLMS. Simulation results based on different VDSL test loops and different noise models have also been given. For both of the adaptive algorithms, the PTEQ tap weights well converge to the MMSE solution. The relation between the convergence speed and the number of equalizer taps was also examined, showing that the convergence speed decreases with the increase of the number of taps for both algorithms.

Chapter 5

Conclusions and Future Work

In this chapter, we draw general conclusions in Section 5.1 and provide suggestions for future work in Section 5.2.

5.1 Conclusions

The realization of high bit-rate data transmission through existing digital subscriber loops (DSL) is being pursued by telecommunication companies due to its advantage in providing broad-band services to various end-users without massive capital outlays. Per-tone equalization (PTEQ) is considered as one of the most efficient equalization techniques in the DSL receiver to tackle the harsh transmission environment and support a reliable high-speed data transmission. In this thesis, we have carried out the design and performance study of practical VDSL systems with per-tone equalization. We have also investigated the performance of adaptive PTEQ techniques using NLMS and SR-RLS algorithms. The objective of the thesis has been to examine the performance of practical VDSL systems with the configurations as required by the ANSI standard as well as the performance improvement of VDSL systems using the PTEQ technique in terms of the SNR of the received signal on each tone as well as the overall transmission bit rate.

We have first introduce the current position of DSL as one of the most widely deployed broadband access media, including the brief history of xDSL family and the main

challenges in the development of practical xDSL systems. A number of impairments that render harsh DSL transmission environment were then enumerated, necessitating the adoption of advanced equalization techniques to increase the DSL coverage and offer more broadband services to as many customers as possible.

The basic concepts of multi-carrier (MC) and discrete multi-tone (DMT) modulations have been introduced. We showed that the MC modulation is capable of achieving the channel capacity while the DMT modulation is a feasible, complexity-constrained transmission scheme to approximate the ideal MC modulation. DMT modulation makes use of the very efficient DFT/IDFT-based modulation and demodulation scheme, leading to a simple equalizer structure in the receiver, namely, a bank of 1-tap frequency equalizers (FEQs). We have explained that a cyclic prefix (CP) needs to be added between transmitted DMT symbols to avoid the ISI and ICI. However, the length of CP specified in the ANSI standard is not sufficiently large, which, in conjunction with the bad spectral containment of sinc-shaped DMT symbols, results in a bit rate that is far below the channel capacity. Therefore, additional time-domain equalization (TEQ) is required. The per-tone equalization as a result of combing the TEQ and FEQ into a multi-tap complex-valued equalizer for each individual tone has been developed, which can maximize the capacity for each tone and lead to a larger total capacity than any other TEQ designs. The diagram of PTEQ has been depicted and the computational complexity of PTEQ during data transmission has been discussed.

We have designed and simulated a practical VDSL system according to the ANSI standard T1.424. The frequency response of seven VDSL test loops has been obtained by means of the two-port modeling. The simulation results of the frequency response of

these loops have been depicted. The training signals, noises and other parameters that are required to construct a real VDSL system have been employed. Simulation results, including the PSD of the transmitted and received signals, the SNR and the achievable bit-rate, have been presented, showing that the PTEQ can well compensate the channel attenuation and distortion and yield a higher transmission bit-rate.

In order to reduce the memory and computational cost for the initialization of PTEQ in VDSL systems, two adaptive algorithms NLMS and SR-RLS have been adopted. The equalizer tap weights are updated at each iteration by using the training signal. Simulation results have shown that the system converges very well with the MEDLEY training signals and the bit-rate of SR-RLS converges much faster than that of the NLMS. The relation between the convergence speed and the number of equalizer taps has also been examined, confirming that the convergence speed decreases with the increase of the number of taps for both NLMS and SR-RLS.

5.2 Suggestions for Future Work

The channel and noise characteristics of VDSL systems may vary with time due to the changing environment. As proposed in the newest VDSL standard, the bits assigned to each tone should be adjustable seamlessly with the changing SNR on that tone during data transmission, which is called the bit swapping. In order to keep the system performance as high as possible during data transmission, the equalizer taps should be adjustable to track these channel variation. A further study for a VDSL system utilizing bit swapping can be conducted to examine the tracking performance of the PTEQ during data transmission.

Although the RLS algorithm shows a fast convergence speed, its computational cost is relatively high. Further study is suggested to adopt the complexity reduced algorithms, such as the order-recursive adaptive filters, whose computational complexity is increased linearly with the number of adjustable filter parameters while a fast convergence speed is retained [28].

In this thesis, we have focused on the DSL communication in a single-input single-output (SISO) context, that is, one DMT transmitter communicates with one DMT receiver over a single twisted pair. Recently, there has been an increased research interest in the so-called multiple-input multiple-output DSL (MIMO-DSL) [44, 45]. By exploiting the crosstalk among the twisted pairs in one binder, the co-located modems are able to cooperate at the end of the binder. Additional performance gains can be obtained by considering a binder of twisted pairs as a MIMO system. The MIMO-DSL techniques may be combined with an appropriate equalizer to deliver a better performance. Therefore, it is worth while studying the equalization issues in the MIMO-DSL systems in the near future.

Appendix

Amateur Radio Bands Recognized by ANSI

Table A.1 Amateur radio bands recognized by ANSI

Start frequency (MHz)	End frequency (MHz)
1.810	2.000
3.500	4.000
7.000	7.300
10.100	10.150
14.000	14.350
18.068	18.168
21.000	21.450
24.890	24.990
28.000	29.700

References

[1] John A. C. Bingham, *ADSL, VDSL, and Multicarrier Modulation,* Palo Alto, California, John Wiley and Sons, Inc, 2000.

[2] Mario Diaz Nava and Christophe Del-Toso, "A Short Overview of the VDSL System Requirements", *IEEE Communications Magazine*, pp. 82-90, Dec. 2002.

[3] ANSI standard T1.424, "Very-high-bit-rate Digital Subscriber Line (VDSL) Metallic Interface," 2002.

[4] R.K.Martin, K.Vanbleu, B.L.Evans, G.Ysebaert, M.Moonen, M.Ding, M.Milosevic and C.R.Johnson, "Unification and evaluation of equalization structures and design algorithms for discrete multitone modulation systems," *IEEE Trans. Signal Process,* vol.53, no.10, pp. 3880 –3894, Oct. 2005.

[5] K.V.Acker, G.Leus, M.Moonen, O. van de Wiel and T.Pollet, "Per tone equalization for DMT-based systems," *IEEE Trans. Commun.,* vol.49, no.1, pp.109 – 119, Jan. 2001.

[6] N. Al-Dhahir and J. M. Cioffi, "Efficiently computed reduced-parameter input-aided MMSE equalizers for ML detection: A unified approach," *IEEE Trans. Information Theory,* vol.42, no.3, pp.903–915, May 1996.

[7] P. J. W. Melsa, R. C. Younce, and C. E. Rohrs, "Impulse response shortening for discrete multitone transceivers," *IEEE Trans. Communications,* vol.44, no.12, pp.1662–1672, Dec.1996.

[8] D. Daly, C. Heneghan, and A. D. Fagan, "A minimum mean-squared error

interpretation of residual ISI channel shortening for discrete multitone transceivers," *In Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, pp. 2065–2068, May 2001.

[9] ANSI standard T1E1.4/98-043R3, "Very-high-bit-rate Digital Subscriber Lines System Requirements," 1998.

[10] Geert Ysebaert, Koen Vanbleu, Gert Cuypers, and Marc Moonen, "Split SR-RLS for the joint initialization of the per-tone equalizers and per-tone echo cancellers in DMT-based receivers," *EURASIP Journal on Applied Signal Processing,* pp. 1433–1445, Oct. 2004.

[11] K.V.Acker, G.Leus, M.Moonen and T.Pollet, "RLS-based initialization for per-tone equalizers in DMT receivers," *IEEE Trans. Commun.,* vol.51, no.6, pp. 885-889, Jun. 2003.

[12] R. K. Martin, D. Ming, B. L. Evans, and C. R. Johnson, "Efficient channel shortening equalizer design," *EURASIP Journal on Applied Signal Processing*, vol.13, pp.1279–1290, Dec. 2003.

[13] J. M. Cioffi. Advanced digital communication: *course notes EE379C.* Technical Report Available viahttp://www.stanford.edu/class/ee379c/readerfiles/chap4.pdf, Stanford University, 2002.

[14] E. A. Lee and D. G. Messerschmitt, *Digital Communication.* Kluwer Academic Publishers, second edition, 1994.

[15] T. Starr, J. M. Cioffi, and P. J. Silvermann, *Understanding Digital Subscriber Line Technology.* Englewood Cliffs, NJ: Prentice Hall, 1999.

[16] John G.Proakis, *Digital Communications*, fourth edition, The McGraw-Hill

Companies, Inc. 2001.

[17] C. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol.27, no.5, pp.379–423 and 623–656, Jul. and Oct. 1948.

[18] J. A. C. Bingham, "Multicarrier modulation for data transmission: An idea whose time has come," *IEEE Commun. Mag.*, vol.28, no.5, pp.5–14, May 1990.

[19] J. S. Chow, J. C. Tu, and J. M. Cioffi, "A discrete multitone transceiver system for HDSL applications," *IEEE Journal Select. Areas Commun.*, vol.9, no.6, pp.895–907, Aug. 1991.

[20] I. Kalet, "The multitone channel," *IEEE Trans. Communications*, vol.37, no.2, pp.119–124, Feb. 1989.

[21] A. Peled and A. Ruiz, "Frequency domain data transmission using reduced computational complexity algorithms," *In Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP),* pp. 964–967, Denver, Colorado, Apr. 1980.

[22] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Third Edition. The John Hopkins University Press, 1996.

[23] Bernard Sklar, *Digital communications Fundamentals and Application,* second edition, 2002.

[24] N. Al-Dhahir and J. M. Cioffi, "Optimum finite-length equalization for multicarrier transceivers," *IEEE Trans. Communications*, vol.44, no.1, pp.56–64, Jan. 1996.

[25] G. Arslan, B. L. Evans, and S. Kiaei, "Equalization for discrete multitone transceivers to maximize bit rate," *IEEE Trans. Signal Processing,* vol.49, no.12, pp.3123–3135, Dec. 2001.

[26] M. Milosevic, L. F. C. Pessoa, B. L. Evans, and R. Baldick, "Optimal time domain

equalization design for maximizing data rate of discrete multitone systems," *IEEE Trans. Signal Processing*, 2003.

[27] B. Farhang-Boroujeny and Y. C. Lim, "A comment on the computational complexity of sliding FFT," *IEEE Trans. Circuits System,* vol. 39, pp. 875–876, Dec. 1992.

[28] Simon Haykin, *Adaptive Filter Theory,* fourth edition, Prentice Hall, Upper Saddle River, New Jersey 07458,2002.

[29] A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing,* second edition, Prentice-Hall, Inc. 1999.

[30] Jean-Jacques Werner, "The HDSL Environment," *IEEE Journal on Selected Areas in Communications,* vol.9, no.6, Aug. 1991.

[31] Irving Kalet, "The Multitone Channel," *IEEE Transactions on Communications,* vol. 37, no.2, Feb. 1989.

[32] Technical report, "ADSL2 and ADSL2+ - the new ADSL standards," Aware Inc. White paper, March 2003.

[33] S. Haykin, "Adaptive digital communication receivers," *IEEE Commun. Mag.,* vol.38, no.12, pp.106–114, Dec. 2000.

[34] G.C. Goodwin and K.S. Sin, *Adaptive filtering prediction and control.* Prentice-Hall, 1984.

[35] Douglas, S.C, and T.H.-Y.Meng, "Normalized data nonlinearities for LMS adaptation," *IEEE Trans. Acoust. Speech Signal Process,* vol. 42, pp. 1352-1365, 1994.

[36] Weiss, A., and D.Mitra, "Digital adaptive filter: conditions for convergence, rates of convergence, effects of noise and errors arising from the implementation," *IEEE*

Trans. Information Theory, vol. IT-25, pp. 637-652, 1979.

[37] Gene H. Golub and Charles F. Van Loan, *Matrix computations (3rd ed.),* page 50, John Hopkins University Press, 1996.

[38] Bierman, G.J., and C.L.Thornton, "Numerical comparison of Kalman filter algorithms orbit determination case study," *Automatica,* vol. 13, pp. 23-35, 1977.

[39] Sayed, A.H., and T.Kailath, "A state-space approach to adaptive RLS filtering," *IEEE Signal Process, Mag.*, vol. 11, pp. 18-60, 1994.

[40] Morf, M., and T.Kailath, "Square-root algorithms for least-squares estimation," *IEEE Trans. Autom. Control,* vol. AC-20, pp. 487-497, 1975.

[41] Gentleman, W.M, and H.T.Kung, "Matrix triangularization by systolic arrays," *in Proc. SPIE, Real Time Signal Processing IV,* vol.298, pp. 298-303, 1981.

[42] Alexander, S.T., and A.L. Ghirnikar, "A method for recursive least-squares filtering based upon an inverse QR decomposition," *IEEE Trans. Signal Process,* vol. 41, pp. 20-30, 1993.

[43] Hudson, J.E., and T.J. Shepherd, "Parallel weight extraction by a systolic least squares algorithm," *in Proc. SPIE, Advanced Algorithms and Architectures for Signal Processing IV*, vol. 1152, pp. 68-77, 1989.

[44] G. Ginis and J. M. Cioffi. "Vectored transmission for digital subscriber line systems," *IEEE Journal on Select. Areas Commun.*, vol.20, no.5, pp. 1085–1104, Jun. 2002.

[45] R. Cendrillon, M. Moonen, G. Ginis, K. Van Acker, T. Bostoen, and P. Vandaele. "Partial crosstalk cancellation for upstream VDSL," *EURASIP Journal Applied Signal Processing,* vol.10, pp.1520–1535, Oct. 2004.

[46] American National Standard for Telecommunications, "Spectrum Management for Loop Transmission Systems," T1.417-2001, Jan. 2001.

[47] G. H. Dobrowski, S. Mukherjee, J. Engstrom, S. Lindecke, A. Nicholson, and B. Dropping, "Symmetric DSL White Paper", DSL Forum, 2002