

Timing Recovery for DOCSIS 3.1 Upstream OFDMA Signals

A Thesis Submitted
to the College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in the Department of Electrical and Computer Engineering
University of Saskatchewan

by
Yayi Xiao

Saskatoon, Saskatchewan, Canada

© Copyright Yayi Xiao, June, 2016. All rights reserved.

Permission to Use

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, it is agreed that the Libraries of this University may make it freely available for inspection. Permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professors who supervised this thesis work or, in their absence, by the Head of the Department of Electrical and Computer Engineering or the Dean of the College of Graduate Studies and Research at the University of Saskatchewan. Any copying, publication, or use of this thesis, or parts thereof, for financial gain without the written permission of the author is strictly prohibited. Proper recognition shall be given to the author and to the University of Saskatchewan in any scholarly use which may be made of any material in this thesis.

Request for permission to copy or to make any other use of material in this thesis in whole or in part should be addressed to:

Head of the Department of Electrical and Computer Engineering
57 Campus Drive
University of Saskatchewan
Saskatoon, Saskatchewan, Canada
S7N 5A9

Acknowledgments

Foremost I would like to express my sincere gratitude to my supervisors, Professor Ha H. Nguyen, Professor J. Eric Salt and my industry supervisor, Dr. Brian Berscheid for their guidance and encouragement throughout my graduate study program. The invaluable knowledge they provided me have certainly helped inspire many of the ideas expressed in this thesis. The discussions that they have given to me have not only influenced the way I study and do research but also will keep enlightening my future career life.

I would also like to thank Vecima Networks Inc. for its funding support in my research in conjunction with the Natural Sciences and Engineering Research Council of Canada (NSERC).

A special thanks goes to the other IOM students, Tung Nguyen, Ben Fortosky, and Chad Holst for their generous help and valuable suggestions throughout my research.

Finally, I must thank my parents, Wen Xiao and Qi Zhang, and my husband, Zixiang Zhou, for their constant love and endless support during my study. I could not have done it without you.

Abstract

Data-Over-Cable Service Interface Specification (DOCSIS) is a global standard for cable communication systems. Before version 3.1, the standard has always specified single-carrier (SC) quadrature-amplitude modulation (QAM) as the modulation scheme. Given that the multi-carrier orthogonal frequency-division multiplexing (OFDM) technique has been increasingly popular and adopted in many wired/wireless communications systems, the newest cable communication standard, DOCSIS 3.1, also introduces OFDM as a major upgrade to improve transmission efficiency.

In any digital communication systems, timing synchronization is required to determine and compensate for the timing offset from the transmitter to the receiver. This task is especially crucial and challenging in an OFDM system due to its very high sensitivity to synchronization errors. Although there have been many studies on the topic of OFDM timing synchronization, none of the existing methods are not directly applicable to DOCSIS 3.1 systems. Therefore, the main objective of this research is to develop effective and affordable timing synchronization algorithms for the DOCSIS 3.1 upstream signal. Specifically, three timing synchronization algorithms are proposed to comply and take advantage of the structure of the ranging signal (i.e., the signal used for synchronization purpose) specified in DOCSIS 3.1 standard. The proposed methods are evaluated under a realistic multipath uplink cable channel using computer simulation. The first algorithm makes use of the repetitive pattern of the symbol pairs in the ranging signal. The locations of the symbol pairs are determined by calculating a correlation metric and identifying its maximum value. The second and third algorithms are developed so that they exploit the mirrored symmetry of the binary phase-shift keying (BPSK)-modulated time-domain samples, corresponding to the first non-zero symbol in the ranging signal, and look for the exact location of the symmetry point. The first algorithm, with very low hardware complexity, provides reasonable performance under normal traffic and channel conditions. However its performance under a severe channel condition and heavy traffic is not satisfactory. The second and third algorithms provide much more accurate timing estimation re-

sults, even under the severe channel condition and heavy traffic flow. Since the second algorithm requires an enormous increase in hardware complexity, a few options are proposed to reduce the hardware complexity but it is still much higher than the complexity of the first algorithm. Applying the same complexity reduction techniques it is demonstrated that the third algorithm has similar hardware complexity to the first algorithm, while its timing estimation performance remains excellent.

Table of Contents

Permission to Use	i
Acknowledgments	ii
Abstract	iii
Table of Contents	v
List of Tables	viii
List of Figures	ix
List of Abbreviations	xii
1 Introduction	2
1.1 Evolution of CATV Industry	3
1.2 Basic Signal Processing in Digital Communications	7
1.3 Scope of the Thesis	10
1.4 Thesis Outline	11
2 Background	13
2.1 Single-Carrier QAM Systems	13
2.2 OFDM Systems	22
2.3 Summary	29
3 Basic Principles of Timing Synchronization in OFDM Systems	30
3.1 Frequency, Phase and Timing Offsets in OFDM Systems	30
3.1.1 Frequency and Phase Offsets	31

3.1.2	Timing Offset	32
3.2	Review of Timing Synchronization Algorithms	36
3.2.1	Schmidl and Cox (S&C) Timing Algorithm	38
3.2.2	Shi and Serpedin (S&S) Timing Algorithm	39
3.2.3	Park and Cheon (P&C) Timing Algorithm	41
3.3	Summary	43
4	Timing Synchronization Algorithms for DOCSIS 3.1 Upstream Trans-	
	mission	44
4.1	Ranging Process in DOCSIS 3.1 Upstream Transmission	44
4.1.1	Ranging Request Signal (RNG-REQ)	46
4.1.2	Time Synchronization Procedure for CMTS in Fine Ranging Process	50
4.2	Timing Synchronization Algorithm I	51
4.2.1	Algorithm	51
4.2.2	Computational Complexity	57
4.2.3	Performance	59
4.3	Timing Synchronization Algorithm II	63
4.3.1	Algorithm	65
4.3.2	Complexity Reduction by Skipping Data Samples	69
4.3.3	Complexity Reduction by Truncating Data Bits	72
4.3.4	Performance	72
4.4	Algorithm III	74

4.4.1	Algorithm	74
4.4.2	Complexity Reduction by Skipping Data Samples	77
4.4.3	Complexity Reduction by Truncating Data Bits	78
4.4.4	Performance	80
4.5	Summary and Comparison	81
5	Summary and Conclusions	83
	References	86

List of Tables

1.1	Different DOCSIS versions.	5
2.1	Micro-reflection bound for upstream dominant single echo.	27
4.1	Number of errors outside the safety region for each peak.	61
4.2	Means and variances of the timing estimation errors obtained using different peaks and their average.	61
4.3	Key system parameters used in Algorithm I simulations.	65
4.4	Number of estimation failures over 1000 simulation runs using different numbers of multipliers and multiplier sizes under different SNR values for Algorithm II.	73
4.5	Number of estimation failures over 1000 simulation runs using different numbers of multipliers and multiplier sizes under different SNR values for Algorithm III.	80

List of Figures

1.1	A generic structure for a DOCSIS system.	4
1.2	An example of frequency spectrum allocation for upstream and downstream signals.	4
1.3	An example of OFDM DOCSIS 3.1 upstream spectrum.	7
1.4	A high-level block diagram for a digital communication system.	8
1.5	Signal processing tasks in the demodulator.	10
2.1	Basic block diagram for a QAM transmitter.	14
2.2	Constellations of BPSK, QPSK and 16-QAM.	15
2.3	Basic block diagram for a QAM receiver.	17
2.4	Frequency responses of raised cosine filters with $\beta=0, 0.5$ and 0.75	18
2.6	Impulse responses for SRRC and RC filters.	20
2.7	Output of the SRRC pulse shaping filter and matched filter when $\beta = 0.5$ and BPSK is employed	21
2.8	Spectrum of OFDM signals.	22
2.9	OFDM system overview with DFT/IDFT.	24
2.10	Illustration of CP extension.	25
2.11	Equivalent discrete-time channel.	26
2.12	Channel delay effect.	28
3.1	Timing offset in an OFDM system.	34

3.2	Frequency-domain phase shift due to a timing error that does not cause ISI.	36
3.3	Frequency-domain phase shift due to a timing error that causes ISI. .	37
3.4	Sliding-window correlator for Schmidl and Cox timing algorithm. . .	38
3.5	An example of timing metric for Schmidl and Cox timing algorithm. .	40
3.6	Timing metrics for S&C, S&S and P&C timing algorithms.	42
4.1	Ranging steps for DOCSIS 3.1 [1].	45
4.2	Cyclic prefix, roll-off period and windowing of OFDM frames in DOCSIS 3.1. [1].	47
4.3	Fine ranging signal structure in DOCSIS 3.1.	47
4.4	Symbol pair structure for fine ranging.	48
4.5	Overall block diagram for time synchronization in the ranging process.	50
4.6	Correlation over the sliding windows for Algorithm I.	51
4.7	Matching points for the first method.	52
4.8	Timing metric plateau window for method I.	53
4.9	Multiple peaks in Algorithm I.	55
4.10	Example of an asymmetrical peak for a timing metric.	56
4.11	Block diagram for the first timing synchronization method.	58
4.12	Illustration of the safety window for the simulation scenario.	59
4.13	Timing estimation errors of the first four peaks.	60
4.14	Timing error by averaging the timing offset values obtained from the four peaks.	62

4.15	Spectrum of the original and filtered signals in the worst case scenario.	63
4.16	Histogram plots of timing estimation errors for Algorithm I under the severe channel condition.	64
4.17	Matching points for the second method.	66
4.18	Comparison of the timing metric for Algorithm I and Algorithm II.	67
4.19	Block diagram for the matching filter.	68
4.20	Applying matching filter to the timing metric in Algorithm II.	69
4.21	Block diagram for Algorithm II.	70
4.22	Timing metric and matching filter's output when skipping different numbers of data samples in Algorithm II.	71
4.23	Timing metric comparison for Algorithm II and Algorithm III.	76
4.24	Applying a matching filter to the timing metric in Algorithm III.	77
4.25	Block diagram for Algorithm III.	78
4.26	Timing metric and matching filter's output when skipping different numbers of data samples in Algorithm III.	79

List of Abbreviations

ASIC	Application-Specific Integrated Circuit
AWGN	Additive White Gaussian Noise
CFO	Carrier Frequency Offset
CIR	Channel Impulse Response
CM	Cable Modem
CMTS	Cable Modem Termination System
CP	Cyclic Prefix
CS	Cyclic Suffix
DFT	Discrete Fourier Transform
DOCSIS	Data Over Cable System Interface Specification
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
IBI	Inter-Block Interference
ICI	Inter-Carrier Interference
IDFT	Inverse Discrete Fourier Transform
IFFT	Inverse Fast Fourier Transform
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISI	Inter-Symbol Interference

OFDM Orthogonal Frequency Division Multiplexing

OFMDA Orthogonal Frequency Division Multiple Access

QAM Quadrature Amplitude Modulation

RP Roll-Off Period

1. Introduction

Electronic communication is now an important part of human daily activities and it has gone through a long history of evolution. As early as in ancient times, people started to use simple symbols or mural to exchange information. Over thousands of years, people have been using languages, symbols, letters and many other different methods to deliver messages. All these communication exchange methods are based on human vision and hearing. As technology evolves, some of the old communication styles are preserved and some are no longer used. Some are still among us in another form such as sign language which is an extension of the old communication methods.

Since the 19th century, with the discovery of electromagnetic waves, the field of human communication has gone through a dramatic evolution. Information transmission via electromagnetic waves over metal cables has been realized. People no longer rely solely on the conventional methods using just hearing and vision for information delivery. With the use of electromagnetic waves as the means for information transfer, people can send and receive more information over a long distance in a short period of time. More and more new technologies were invented such as telegraphy, telephony, and more recently, cellular wireless communications. Cable television (CATV) is another new technology for information delivery, which has gone through decades of evolution. The next section gives a brief history of CATV evolution since information transmission in CATV networks is the main subject of this thesis.

1.1 Evolution of CATV Industry

Cable television has been around since late 1940s [2]. For the original cable networks, data transmission is unidirectional. The cable center office (head-end) broadcasts television signals to multiple end users and the end users do not transmit data back to the head-end. As the cable industry develops and the demand for high speed services increases, many cable television providers have been providing both cable TV and Internet services. Meanwhile, more features have been added to cable system services such as video-on-demand, video conferencing, etc. Such services are all interactive and require bidirectional IP (Internet Protocol) traffic transfer between the head-end and the end users. The design and operation of such modern bidirectional cable systems are governed by DOCSIS standards.

DOCSIS stands for *Data-Over-Cable Service Interface Specification* and it is an international telecommunication standard for cable systems developed by CableLabs.¹ The system architecture for a generic DOCSIS network is shown in Figure 1.1. There are two main components in a DOCSIS system: *cable modem* (CM) and *cable modem termination system* (CMTS). A CM is located at the end user and it is connected to subscriber premises equipment. A CMTS is located at the cable network head-end and it is used for distributing data streams to and receiving information from all CMs that are connected to the network. The CMs connect to the CMTS head-end operator through a hybrid fiber-coaxial (HFC) network.

All data transfer in DOCSIS cable systems is bi-directional. *Upstream* refers to the direction of data transmission from the CMs (end user) to the CMTS (head-end), whereas *downstream* refers to the direction of data transmission from the CMTS to the CMs. In practice, the system needs to support two-way communications simultaneously. To accomplish this requirement, the frequency spectrum of the cable is partitioned into two disjoint (non-overlapped) regions. Each region is dedicated to either the downstream or upstream transmission (see Figure 1.2). In this way down-

¹Cable Television Laboratories, Inc is a not-for-profit R&D consortium that has cable operators as members to determine the service requirements to be supported by new technologies.

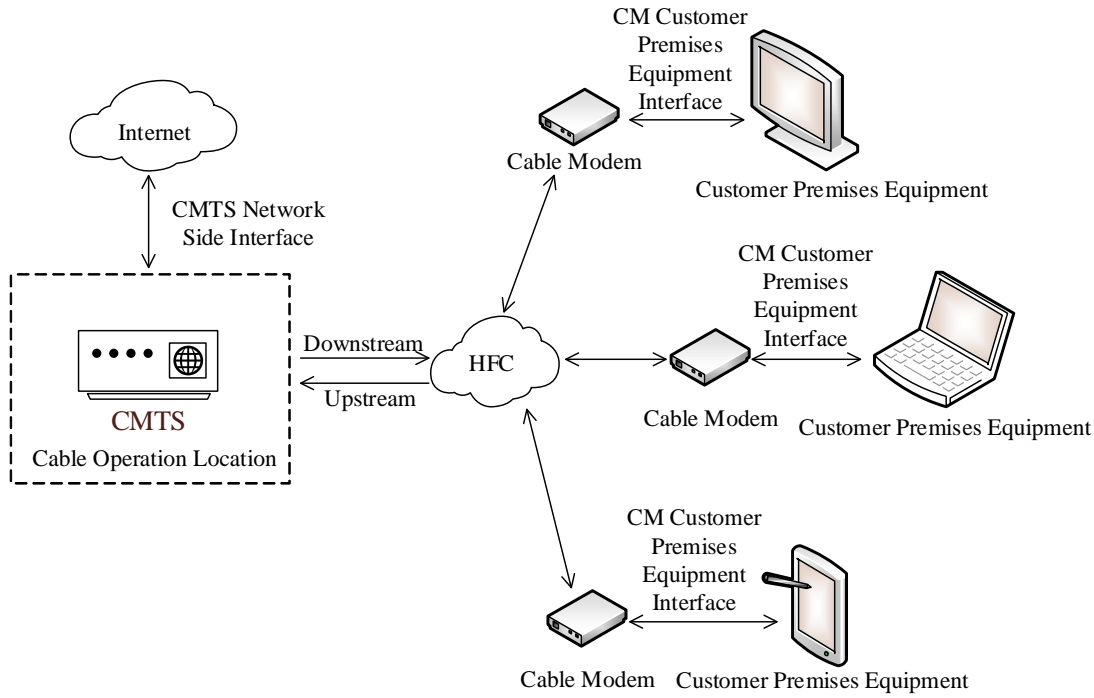


Figure 1.1: A generic structure for a DOCSIS system.

stream and upstream signals will transmit only on the allocated frequency regions without interfering each other.

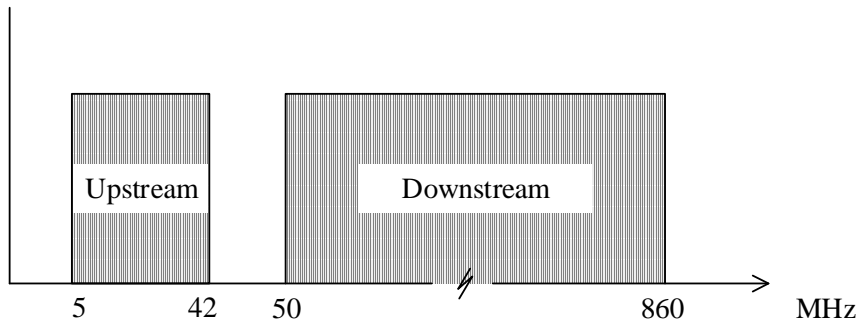


Figure 1.2: An example of frequency spectrum allocation for upstream and downstream signals.

The first DOCSIS version, DOCSIS 1.0, was initially released in 1997 [3]. Over the years, new versions were released to provide higher data throughput and improved efficiency. Table 1.1 shows the release date as well as the frequency boundaries for the upstream and downstream signals for each version. Compared to DOCSIS 1.0,

DOCSIS Version	Release Date	Downstream Frequency Range (MHz)	Upstream Frequency Range (MHz)	Minimum Number of Channels that Hardware Must Support
1.0	March 1997	50-860	5-42	1
2.0	December 2001	50-864	5-42	1
3.0	August 2006	50-1002	5-85 or 5-42	4
3.1	October 2013	258-1218	5-204	2

Table 1.1: Different DOCSIS versions.

in DOCSIS 2.0 [4] the upstream throughput was increased by employing more advanced modulation techniques. Specifically, while DOCSIS 1.0 only supports QPSK (quadrature phase-shift keying) and 16-QAM (quadrature amplitude modulation) for the upstream channel, DOCSIS 2.0 can support up to 128-QAM. The ideal downstream throughput for DOCSIS 1.0 is 42.88 Mbits/s and the upstream throughput is 10.24 Mbits/s for a 6 MHz channel. DOCSIS 2.0 maintained the already impressive downstream speed but tripled the upstream speed to 30.72 Mbits/s. This is in response to the increased demand for more symmetrical communications. After the introduction of DOCSIS 2.0, the cable networks have undergone a great increase in size such that the addressing constraint of IPv4 addressing scheme has become a big issue. Therefore DOCSIS 3.0 [5] added the compatibility with IPv6 standard, which uses 128 bits for IP addresses as opposed to 32 bits in IPv4. Also, DOCSIS 3.0 has slightly wider bandwidths for both upstream and downstream signals compared to the previous versions. While DOCSIS 1.0 & 2.0 can only support one channel per upstream or downstream, DOCSIS 3.0 is capable of supporting multiple channels per cable modem. This multi-channel structure significantly increases the throughput of the system. In particular, the equipment needs to support a minimum of 4 channels for both upstream and downstream signals and there is no maximum limit on the number of channels. In DOCSIS 3.0 and earlier versions, SC-QAM (single-carrier QAM) transmission is used for each channel that occupies a bandwidth of 6.4 MHz

or 3.2 MHz. Although multiple channels can be supported in DOCSIS 3.0, the available bandwidth cannot be fully utilized because of the need to have enough frequency separation between channels to avoid inter-carrier interference (ICI).

In recent years, the demand for a higher transmission speed and a better bandwidth efficiency has increased significantly. Operators are faced with the requirements to deploy reliable high-speed services to customers. In response to these needs, CableLabs introduced DOCSIS 3.1, the latest version of the DOCSIS standard, in October 2013 [1]. Communications in CATV under DOCSIS 3.1 is the focus of this thesis. The ultimate service goal of DOCSIS 3.1 is to achieve a full spectrum use and therefore provides a higher system throughput. As mentioned, single-carrier systems are used prior to DOCSIS 3.1 where only one carrier is used to transmit the information-bearing signals. To achieve a higher data rate requires a larger transmission bandwidth. As the signal bandwidth increases, it is more likely that the broadband signals will suffer from the fluctuation of the channel frequency response. The hardware cost to compensate for this effect with SC-QAM transmission would also increase significantly. To effectively deal with the effect of channel distortions experienced in high-speed transmission, orthogonal frequency-division multiplexing (OFDM) technique is introduced in DOCSIS 3.1. OFDM is a multi-carrier system that partitions a broadband channel into multiple narrow sub-channels, each assigned with a unique subcarrier. An example for one DOCSIS 3.1 upstream channel that occupies a bandwidth of 96 MHz is shown in Figure 1.3, in which the channel spans from 102 MHz to 198 MHz. There are two modes of operation for a DOCSIS 3.1 OFDM system depending on the size of the fast Fourier transform (FFT) operation: 2048 (2K) and 4096 (4K) modes. Note that in this example, the system operates in the 4K mode. There is a total of 4096 subcarriers inside the channel and the spacing between two adjacent carriers is 25 kHz. This spacing between subcarriers is different but fixed for each mode: it is 25 kHz in the 4K mode and 50 kHz in the 2K mode. DOCSIS 3.1 also supports high-order modulation schemes of up to 4096 QAM, and therefore theoretically can provide up to 10 Gbits/s downstream and 1 Gbits/s on the

upstream. More details on the advantages of OFDM systems over SC-QAM systems will be covered later in Section 2.2.

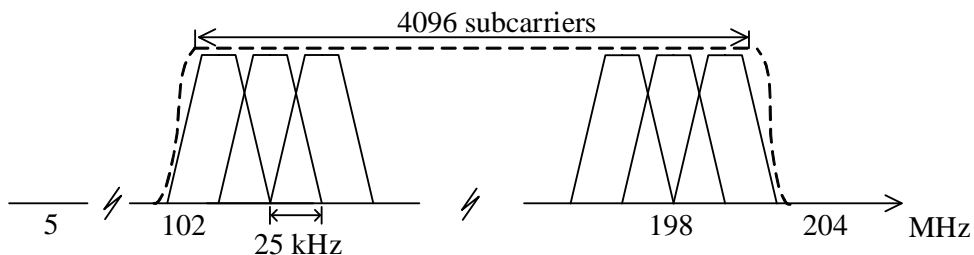


Figure 1.3: An example of OFDM DOCSIS 3.1 upstream spectrum.

1.2 Basic Signal Processing in Digital Communications

Data transfer in a DOCSIS system is by means of digital communications. It is essential to cover the basic signal processing and operations of a digital communication system first before presenting in more detail the subject of the thesis. As illustrated in Figure 1.4, there are three basic blocks in a digital communication system: the transmitter, the channel, and the receiver.

The transmitter first performs channel encoding on the information sequence. What channel encoding does is to add some amount of redundancy to the information so that the redundancy can be used to increase the reliability of information recovery at the receiver. The channel-encoded data is converted into a form that is suitable for transmission through the transmission medium by the process of modulation. For high-frequency transmission, the modulation process uses the encoded data to systematically vary the amplitude, phase, frequency, or any combinations of these parameters of a sinusoidal carrier signal. For example, in AM (Amplitude Modulation) radio, the information signal is contained in the amplitude envelope of the sinusoidal carrier. An important aspect of carrier modulation is to translate the information signal into a frequency band that matches the allocated frequency of the channel. In this way, the carrier modulation process makes it possible to allow multiple message signals from different transmitters to be transmitted over the same

physical channel.

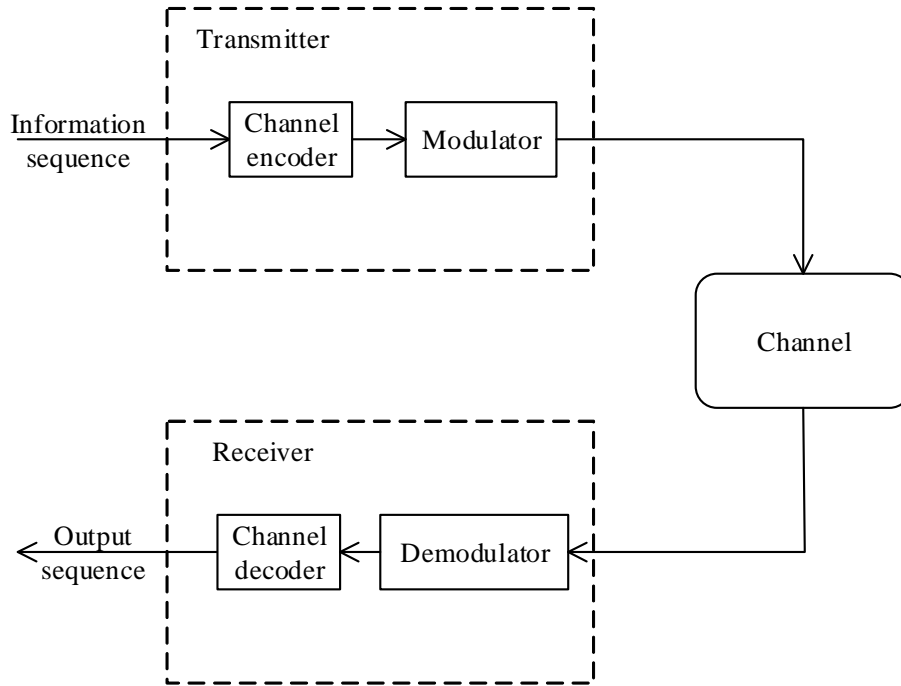


Figure 1.4: A high-level block diagram for a digital communication system.

The communication channel is the physical medium that is used to transfer the signal from the transmitter to the receiver. Common transfer mediums include the air for wireless communications, twisted-pair wires, coaxed cables and optical fiber, which are used for telephone and/or TV signals. Specifically, for CATV systems, the broadband signals are transmitted via a hybrid fibre-coaxial cable network. In most practical cases of digital communication systems, the channel will introduce multiple distortions to the transmitted signal. The most common form of signal degradation is due to additive noise. There are several causes for additive noise, the most prominent one being the thermal noise generated by the internal electrical components inside the receiver [6]. Other common additive noise sources include the interference from adjacent channels, and as in wireless communications, atmospheric noise picked up by the antenna. Additive noise corrupts the transmitted signal and degrades the signal integrity by causing uncertainties to the signal.

Many other channel degradations however are system specific. For example, in

a wireless transmission, the Doppler effect is an important factor since the end user might be moving and the distance between the transmitter and receiver is changing. Also there are many objects in the atmosphere that scatter the transmitted signal before it arrives at the receiver. As a result, the wireless channels exhibit a Rayleigh fading response.

The above impairments that are present in wireless channels do not apply to CATV systems since the transmitter and receiver are stationary and signals are transmitted through cables so there are no obstacles that scatter the signal along the way. Besides additive noise, some of the major signal distortions brought by the cable channel are *frequency offset*, *timing delay*, *phase offset* and *micro-reflections*. Frequency offset refers to the difference between the carrier frequency of the received signal and the nominal value of the transmitted carrier frequency. This offset normally originates from the difference of the transmitter and receiver clock signals for cable systems, and/or the Doppler effect for wireless transmissions. The carrier phase of the received signal also differs from the transmitted signal phase due to channel phase response and also the imperfect synchronization between the transmitter and receiver clocks. Timing offset is another major issue and it is caused by the unknown transmission delay introduced by the channel as the signal gets transmitted to the receiver. Micro-reflection occurs when there is an impedance mismatch along the line and the received signal will get reflected back and forth with reducing power. More details on channel impairments encountered in CATV systems will be discussed in Section 3.1.

The receiver is responsible for recovering the transmitted message contained in the received signal. It consists of two main functions: the demodulator and the channel decoder. The demodulator is responsible for extracting the channel-encoded signal from the sinusoidal carrier signal. Then the encoded signal is sent to the channel decoder to recover the original information signal. As discussed above, the received signal in a cable system is degraded by a number of impairments by the channel. These impairments introduce various distortions to the transmitted signals. It is essential to estimate and correct for these impairments in order to ensure reliable data

recovery and avoid severe system performance degradation. These errors must be corrected mathematically inside the demodulator module prior to the demodulation step as shown in Figure 1.5. The process to estimate and correct for these errors is called synchronization. This thesis is particularly concerned with *timing synchronization* and the terms timing synchronization and timing recovery are used interchangeably. The high-level structure of Figure 1.5 is generic and these blocks may be used in different orders for different applications [7]. As the figure suggests, the error correction process consists of different synchronization and equalization blocks. Each block estimates and corrects for a specific error that was introduced by the channel. The synchronization process plays a very important role in a digital communication system. The performance and cost of the receiver greatly depends on the design of these synchronization blocks.

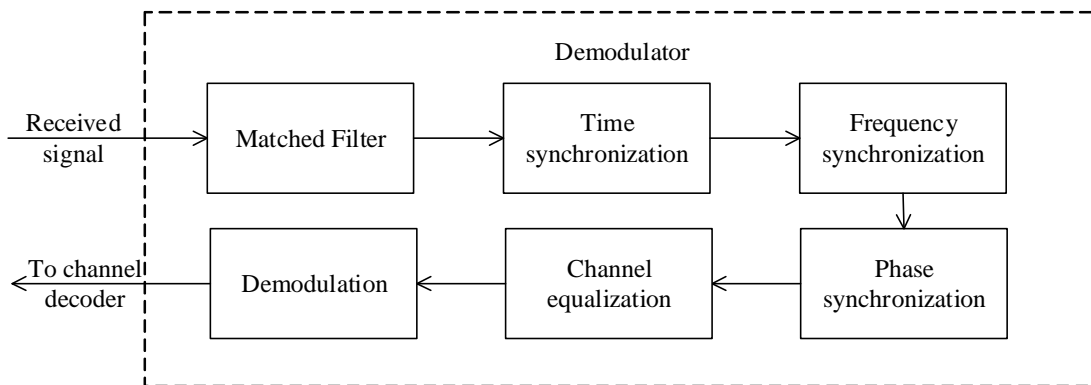


Figure 1.5: Signal processing tasks in the demodulator.

1.3 Scope of the Thesis

The quality and performance of the timing recovery operation is important as this is normally the first step in the synchronization process. Various timing synchronization techniques have been explored over the past decades [7–12]. The main purpose of this research is to investigate and analyze possible approaches to carry out timing synchronization for the latest DOCSIS 3.1 system under the cable channel environment, especially the upstream transmission which is particularly challenging due to the fact that it can have multiple simultaneous transmitters.

As mentioned before, timing error is due to the propagation delay introduced by the channel and it depends mainly on the distance between the transmitter and the receiver. Therefore in the upstream signal received at the CMTS side, the time delay is specific to each CM in the network. The timing recovery algorithm needs to be able to properly synchronize each specific CM in time. It is required that the proposed synchronization algorithm be suitable for FPGA (Field Programmable Gate Array) implementation. Being a re-configurable device, FPGA provides developers an economical solution in system design since it can potentially be reconfigured to realize firmware update needed to meet new standards. However, compared to other common embedded systems such as ASICs, FPGAs are relatively limited in the number of logic resources, especially the multipliers available. Therefore, it is important to ensure that the proposed algorithm requires a reasonable number of multipliers to keep the implementation cost low.

It is pointed out that this thesis is meant to serve as a proof of concept rather than the full hardware implementation. The approach in the thesis is to use MATLAB simulation to evaluate the feasibility and performance of the proposed timing recovery algorithms.

1.4 Thesis Outline

This thesis is organized into five chapters. This first chapter introduces CATV systems, their history, evolution and the DOCSIS standard that governs their operations. Some fundamental concepts in digital communication are provided. The chapter also briefly discusses challenges in cable system communications which lead to the motivation for this research.

The OFDM technique is introduced in Chapter 2. The chapter starts by presenting some basic principles of single-carrier QAM systems before extending to multi-carrier OFDM systems. It is also presented in Chapter 2 that OFDM systems offer advantages over single QAM systems.

The requirements and constraints for the OFDM systems to have advantages over single-carrier QAM systems are then further discussed in Chapter 3. The major distortions introduced by the cable channel to the received signal and their effects on OFDM signal integrity are discussed. Among all the distortions discussed, timing offset is emphasized which is also the main topic of the thesis. A few classical timing recovery methods are analyzed and comparison is made regarding their performances.

Chapter 4 first studies the DOCSIS 3.1 specific timing synchronization process, set-up and specifications. The signal used for timing estimation in DOCSIS 3.1 is discussed in detail. Then three algorithms are proposed to utilize the DOCSIS 3.1 specific signal structure. Hardware design complexity is also discussed and compared among the three methods. Several hardware complexity reduction options are given at the cost of performance loss. Simulations are performed with all the three methods proposed to measure the performance of timing estimation accuracy under a DOCSIS 3.1 cable system. Comparison of the simulation results and discussions are presented.

Finally, Chapter 5 summarizes the main contributions and concludes the thesis.

2. Background

As discussed in Chapter 1, DOCSIS 3.1 exploits multi-carrier OFDM in the physical layer as a new engine to greatly increase the system's throughput. The timing synchronization process for DOCSIS 3.1 upstream transmission involves many working principles of an OFDM system as well as cable channel conditions. To help understand the operations and performance of the timing recovery algorithms proposed in this thesis, it is necessary to acquire a general understanding of the DOCSIS upstream signal structure. In this chapter the basic principles of OFDM will be explained. Since OFDM is an extension of SC-QAM, it is useful to also review the QAM theory and discuss its drawbacks. There are various impairments in the received upstream signal introduced by the cable channel that impact the synchronization task. These channel impairments and their effects on the received signal will also be discussed in this chapter.

2.1 Single-Carrier QAM Systems

As specified in all versions of DOCSIS standards, data is transferred using QAM. Prior to DOCSIS 3.1, only one carrier frequency is used to transmit the information data and this technique is called SC-QAM. QAM utilizes two varying-amplitude sinusoidal carrier signals to convey information. The two carrier waves are of the same frequency F_c but with $\pi/2$ radians phase difference and they are called quadrature carriers. This phase difference ensures the two carrier signals are orthogonal and do not interfere with each other. Denote the two carrier signals by $\cos(2\pi F_c t)$ and $\sin(2\pi F_c t)$ and the transmitted symbol by a complex number $a_I + ja_Q$. Then the

transmitted signal will be of the form:

$$s(t) = a_I \cos(2\pi F_c t) + a_Q \sin(2\pi F_c t), \quad (2.1)$$

where a_I and a_Q are the real and imaginary parts of the complex symbol, respectively. Since the symbols are represented as complex numbers, they can be viewed as points on a two-dimensional complex plane. The real and imaginary axes are also called in-phase and quadrature, respectively. Plotting all possible modulation symbols in a two-dimensional plane produces the constellation diagram. The points on the constellation diagram are called constellation points, or symbols.

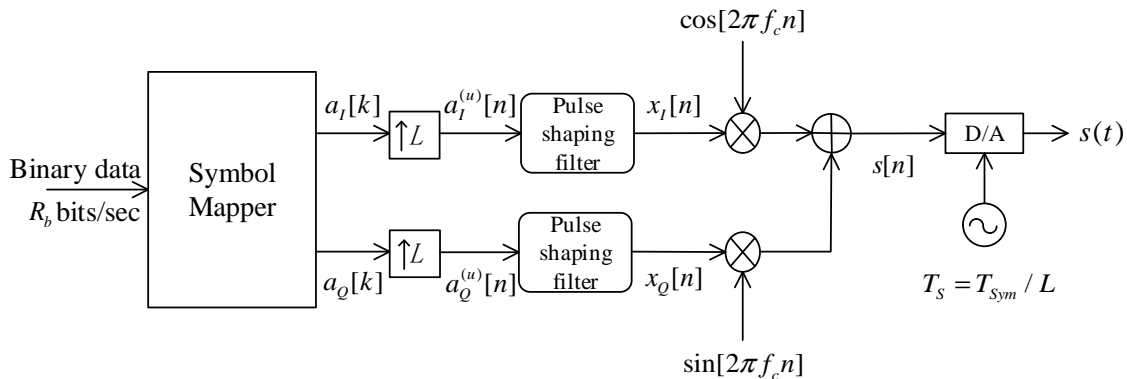


Figure 2.1: Basic block diagram for a QAM transmitter.

Figure 2.1 illustrates a basic block diagram of a QAM transmitter. The input binary sequence comes in at a rate of R_b bits/s and enters the QAM symbol mapper. In the symbol mapper, the binary data is grouped and mapped into QAM symbols in a manner that is known to both the transmitter and the receiver. The specific mapping rule maps every group of λ bits into one QAM symbol. Since there are 2^λ different patterns for λ bits, the number of all possible QAM symbols is $M = 2^\lambda$. Each QAM symbol is represented by a 2-tuple, the in-phase component $a_I[k]$ and quadrature component $a_Q[k]$, where k denotes the symbol index in time. This encoding rule is specific to each standard. An example of constellation diagrams for $\lambda = 1, 2, 4$ is shown in Figure 2.2. Note that the mapping rules for $\lambda = 1$ and $\lambda = 2$ are the same as phase-shift keying (PSK). Unlike amplitude modulation, phase-shift keying symbols have a constant amplitude and the symbols are modulated by changing the phase of

the sinusoidal carrier signal. As the figure shows, the constellations points for $\lambda = 1$ and $\lambda = 2$ are on the same unit circle and only differ in phase. These constellations are normally referred to as BPSK (binary phase shift keying) and QPSK (quadrature phase shift keying), respectively. The QAM symbols are obtained as the output of the symbol mapper in the form of in-phase and quadrature components, $a_I[k]$ and $a_Q[k]$. Each symbol has a period of $T_{\text{sym}} = \lambda T_b$ where $T_b = 1/R_b$ is the bit duration. Therefore the symbol rate is $R_{\text{sym}} = R_b/\lambda$.

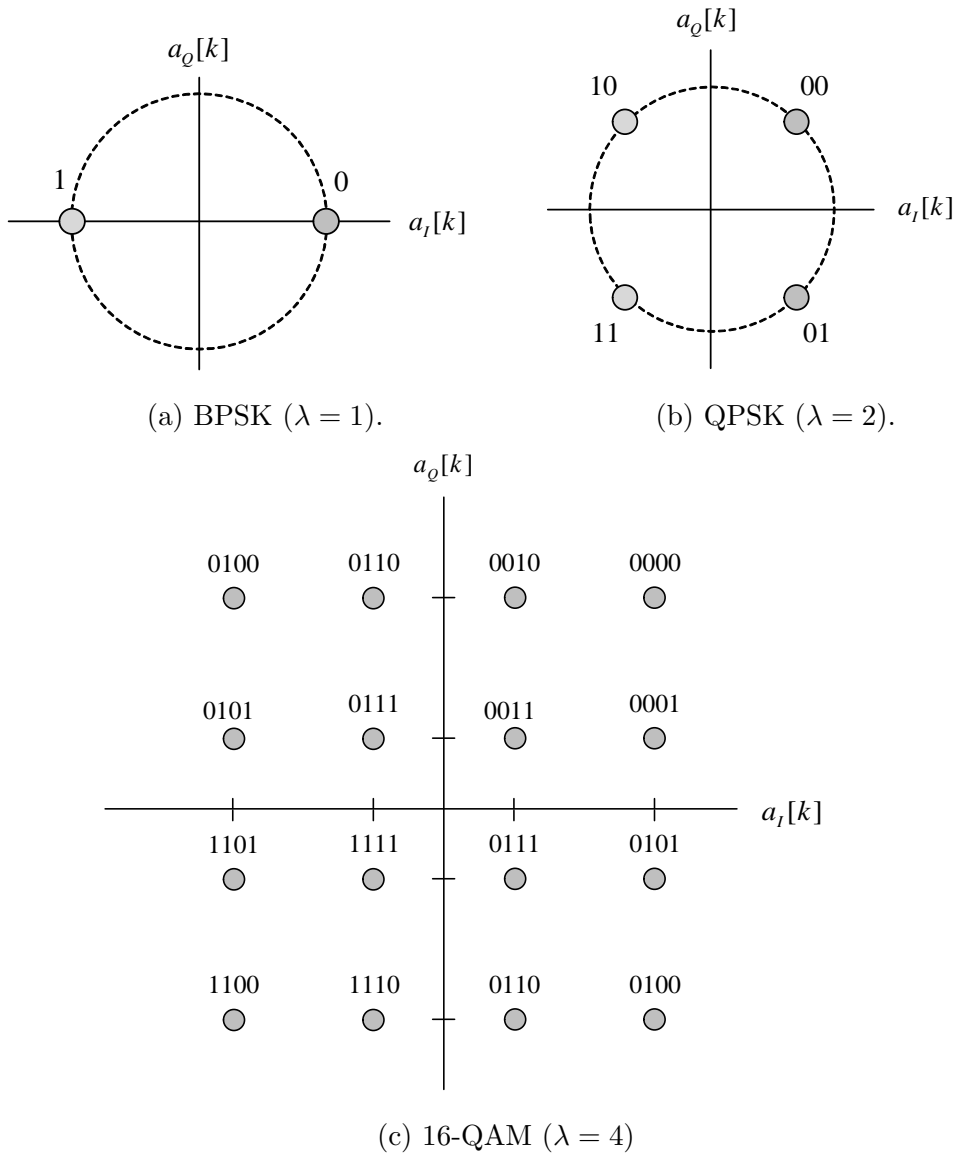


Figure 2.2: Constellations of BPSK, QPSK and 16-QAM.

Since frequency spectrum is a limited and valuable resource in communication systems, it is important to make an efficient use of the available spectrum. A low-pass filter is required in order to limit the effective bandwidth occupied by the signal and prevent possible interference to other channels. First $a_I[k]$ and $a_Q[k]$ are upsampled by inserting $L-1$ zeros between every pair of samples. This operation is usually called upsampling by a factor of L . Typical values of L are 4, 6, 8. Then the upsampled signals $a_I^{(u)}[n]$ and $a_Q^{(u)}[n]$ are passed through the low-pass filters to obtain the in-phase and quadrature signals, $x_I[n]$ and $x_Q[n]$. The specific low-pass filter for SC-QAM signals is also called a pulse shaping filter. A commonly-used pulse shaping filter is a square root raised cosine filter (SRRC). Denote the impulse response of a SRRC by $h_{\text{ps}}[n]$. Then the outputs of the filters are given as

$$x_I[n] = \sum_{m=-\infty}^{\infty} a_I^{(u)}[m] h_{\text{ps}}[n-m] \quad (2.2a)$$

$$x_Q[n] = \sum_{m=-\infty}^{\infty} a_Q^{(u)}[m] h_{\text{ps}}[n-m] \quad (2.2b)$$

Each of the two elements $x_I[n]$ or $x_Q[n]$ is amplitude modulated with one of the digital quadrature carriers with a carrier frequency f_c . The digital frequency f_c has unit cycles/sample. It can be converted from the analog carrier frequency F_c which has unit cycles/s (or Hz) by the relationship $f_c = F_c/F_S$ where F_S is the sampling frequency. Then the sum of the two modulated signals, $s[n]$, is converted to an analog signal $s(t)$ by the digital-to-analog converter of sampling time $T_S = 1/F_S = T_{\text{sym}}/L$, where T_{sym} is the symbol time before upsampling. The sum of the two modulated signals at the input of the digital-to-analog converter can be written as:

$$s[n] = x_I[n] \cos(2\pi f_c n) + x_Q[n] \sin(2\pi f_c n) \quad (2.3)$$

The most common digital-to-analog converter first outputs a piecewise-constant waveform where the previous voltage level is held at the output for the clock cycle until the next input number is latched. This piecewise-constant reconstruction causes multiple harmonics above the Nyquist frequency. A low pass reconstruction filter is

often used inside the digital-to-analog converter to remove these harmonics in order to meet the Nyquist criterion. In essence, the reconstruction filter will smooth out the step response into a continuous waveform $s(t)$.

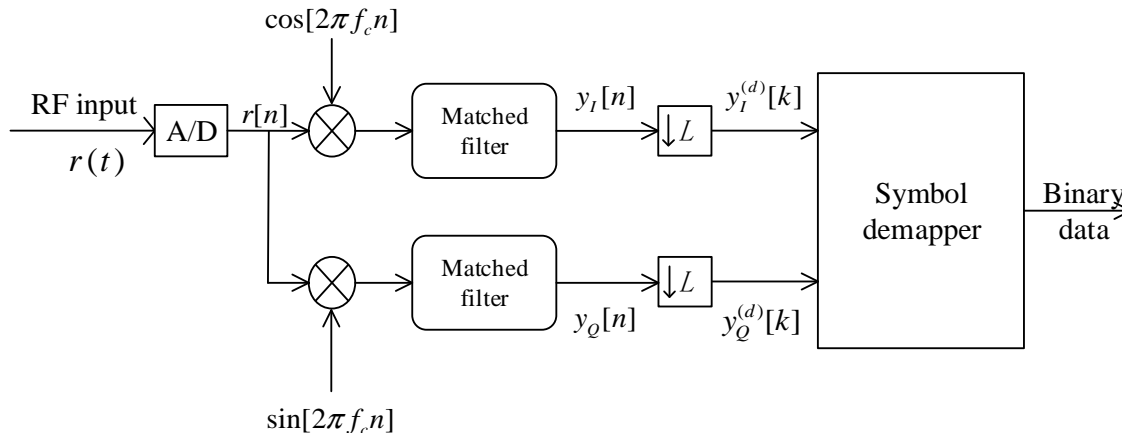


Figure 2.3: Basic block diagram for a QAM receiver.

At the other end of the communication channel, the binary information data is recovered with a QAM receiver. The structure for the QAM receiver is shown in Figure 2.3. Denote the received analog signal by $r(t)$. The signal $r(t)$ is first sampled into a digital signal. The digital signal $r[n]$ is then downconverted to baseband by multiplying with the two quadrature carriers of frequency f_c . The product signals then go through matched filters which are normally another SRRC filter that is identical to the pulse shaping filter used in the modulator. The main purpose of applying this filter is to maximize the signal-to-noise ratio at the correct sample point [13]. The filter also removes the high frequency components that are generated due to multiplication with sinusoids. The matched filter when combined with the pulse shaping filter results in a frequency response of a *raised cosine* function as given in Equation (2.4). In this equation, ω is the frequency in radians/sample, L is the upsampling factor. The parameter β is called the roll-off factor of the filter, which takes a value between 0 and 1, and it specifies the width of the transition band. As the equation indicates, the transition bandwidth of the filter is $\frac{2\pi\beta}{L}$ and a larger β value will result in a wider transition band. This is illustrated in Figure 2.4 in which three frequency responses are shown with $L = 4$ and $\beta = 0, 0.5$ and 0.75 , respectively.

The corresponding SRRC filters then have a frequency response which is the square root of the raised cosine spectrum.

$$H(e^{j\omega}) = \begin{cases} 1, & |\omega| \leq \frac{\pi(1-\beta)}{L} \\ \frac{1}{2} \left[1 + \cos \left(\frac{\pi}{2\beta} \left(\frac{|\omega|L}{\pi} - 1 + \beta \right) \right) \right], & \frac{\pi(1-\beta)}{L} < |\omega| \leq \frac{\pi(1+\beta)}{L} \\ 0, & \text{otherwise} \end{cases} \quad (2.4)$$

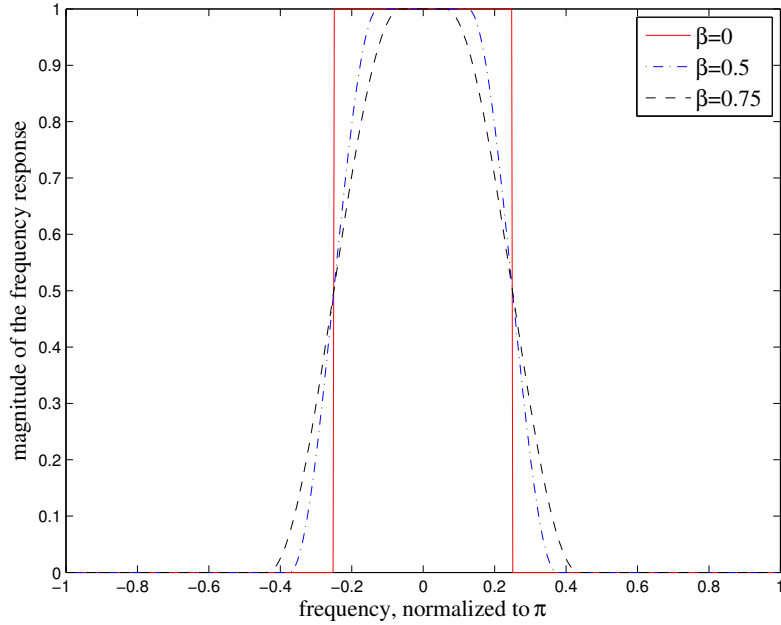


Figure 2.4: Frequency responses of raised cosine filters with $\beta=0, 0.5$ and 0.75 .

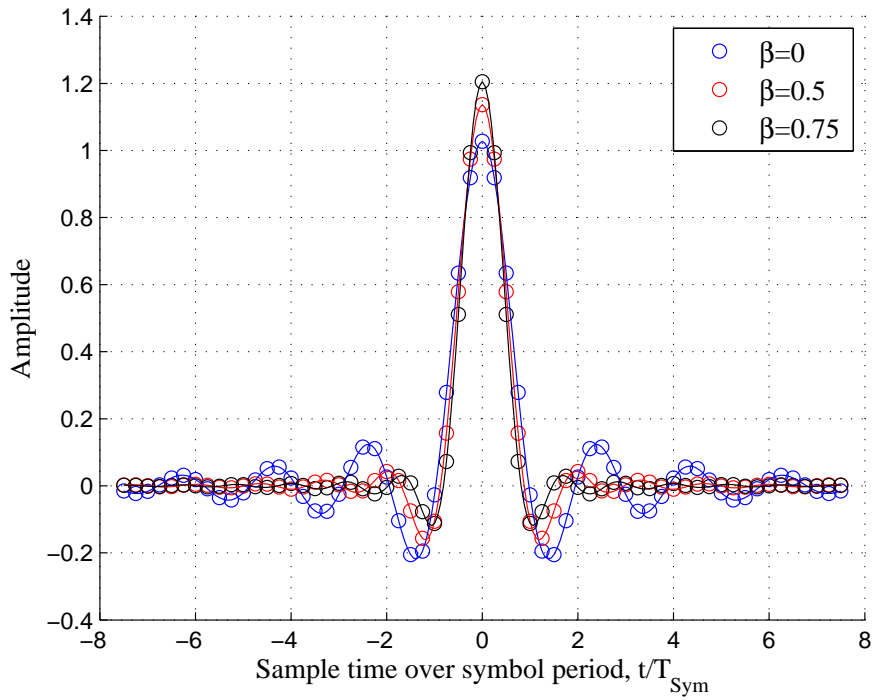
Since the low pass filters introduce a propagation delay to the signal, each information symbol will have a longer time duration at the filter output. This will likely cause the symbols to interfere with each other and this type of interference is called inter-symbol interference (ISI). ISI is an unwanted phenomenon since the smearing of the previous symbol acts like noise on the current symbol and makes the data recovery less reliable. Fortunately, a perfect raised cosine filter will eliminate this ISI since it meets the Nyquist filter criterion. A Nyquist filter has its impulse response to null at multiple symbol periods nT_{sym} , except $n = 0$. Figure 2.6 shows an example of the impulse responses for a square-root raised-cosine filter and its corresponding

raised-cosine filter with different roll-off values β . Note that as the β value decreases the impulse response approaches zero more slowly at both ends. Therefore to achieve narrower bandwidth (smaller β value), more filter coefficients are required in implementing a practical finite impulse response (FIR) filter, which increases the hardware cost. It is clear from the figure that if the receiver samples the input signal properly, then the sample contains contribution from only one symbol at a time. Thus the Nyquist criterion is met and no ISI occurs.

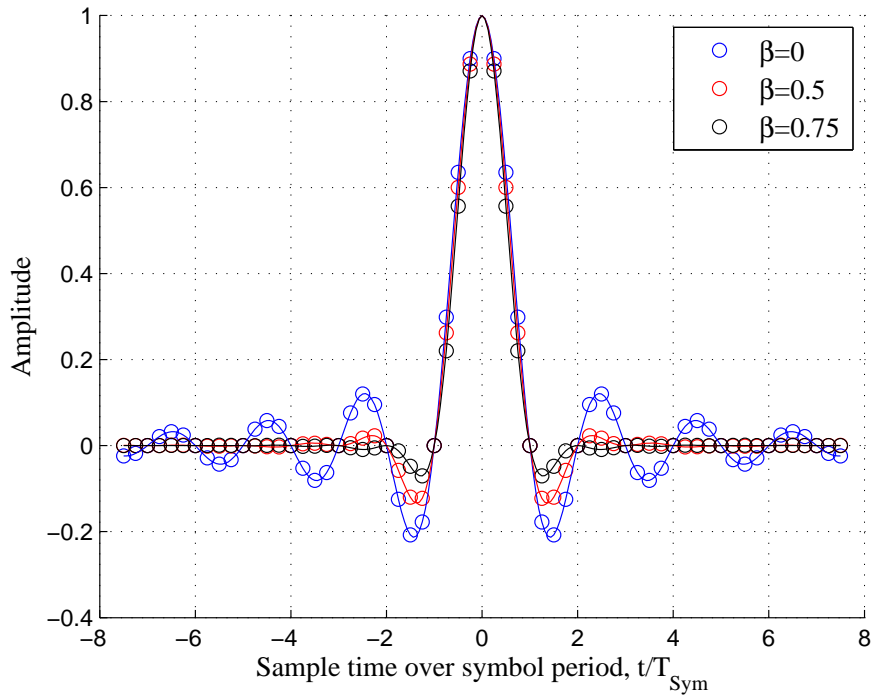
To illustrate the operations of the pulse shaping and matched filters, different signals are plotted in Figure 2.7 for SRRC filters with $\beta = 0.5$ and when BPSK is employed. First, $a_I^{(u)}[n]$ is obtained by upsampling the input sequence $a_I[k]$ by $L = 4$. The upsampled data gets passed into the SRRC pulse shaping filter whose output is plotted in the middle panel. The bottom panel illustrates how the signal is reconstructed with a matched filter at the receiver side.

The output of the matched filter is downsampled by L to keep the samples only at $k = Ln$ to obtain the in-phase and quadrature elements $y_I[k]$ and $y_Q[k]$. These are components of the complex signal $y_I[k] + jy_Q[k]$ and they are fed into a QAM demapper as decision variables. The transmitted symbols are recovered and demapped into the original binary data.

There is one major disadvantage of SC-QAM. As the data rate gets higher, the duration of one information symbol becomes smaller and the occupied bandwidth increases. As the symbol duration decreases, the delay in the micro-reflection becomes more prominent since the ratio between the micro-reflection delay and symbol period increases. Therefore micro-reflection causes more distortions to the signal and damages the signal integrity. In addition, other impairments introduced by the cable channel also have huge impacts on the transmitted signal such as impulsive noise. This complicates the process for channel equalization at the receiver. As the bandwidth increases, more complicated and expensive algorithms are required to realize channel equalization. The necessity for building complicated equalization in a high-speed SC-QAM receiver can be avoided by extending the concept of SC-QAM



(a) Square root raised cosine impulse response.



(b) Raised cosine impulse response.

Figure 2.6: Impulse responses for SRRC and RC filters.

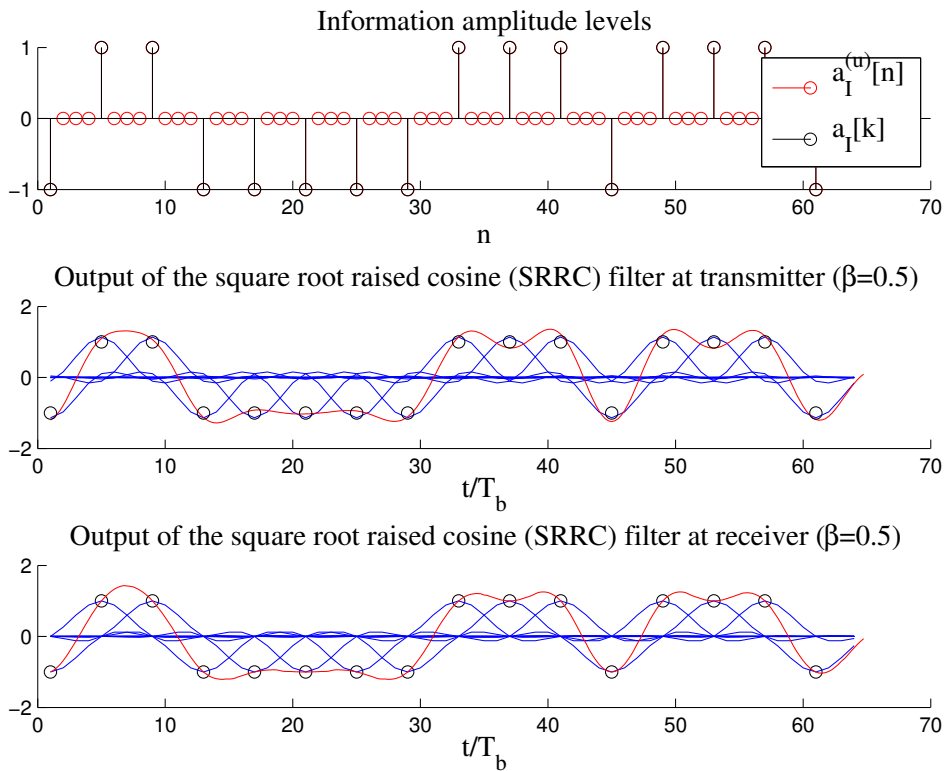


Figure 2.7: Output of the SRRC pulse shaping filter and matched filter when $\beta = 0.5$ and BPSK is employed

to frequency division multiplexing (FDM) which uses multiple sub-carriers within a single frequency band. As the entire occupied bandwidth is divided into multiple narrower subchannels, each subchannel is smaller and much easier to equalize. OFDM is one specific application of FDM.

2.2 OFDM Systems

To use the available frequency resource more efficiently, OFDM is adopted in DOCSIS 3.1 standard. In OFDM, the available spectrum is divided into N independent orthogonal subcarriers, each modulated with an individual symbol stream as illustrated in Figure 2.8. The following description of the OFDM transmitter and receiver follows closely references [14, 15].

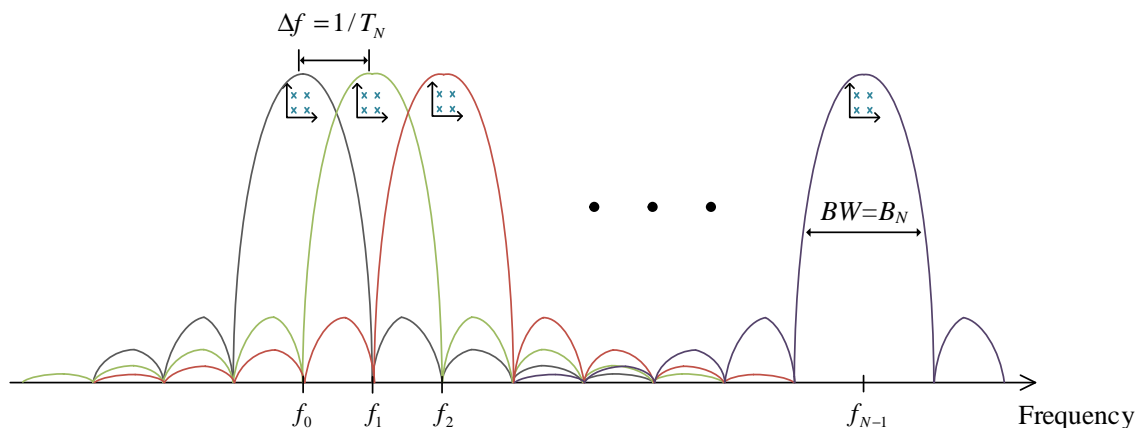


Figure 2.8: Spectrum of OFDM signals.

An OFDM transmitter can be viewed as a combination of N separate QAM transmitters. The information sequence is first sorted into N groups through a serial-to-parallel converter. Each binary data group is passed into a separate QAM transmitter. For the n^{th} QAM transmitter, the symbols are modulated with a sinusoidal carrier of frequency f_n in the same way as discussed in Section 2.1. Then the outputs of all N QAM modulators are summed and transmitted as $s(t)$. In this way, a high-rate data stream is divided into many low-rate parallel streams. Let λ_n represent the number of bits that the n^{th} subchannel carries in each symbol duration. Then the

total number of bits the system is able to transmit in every duration of $T_N = NT_{\text{sym}}$ is $\lambda = \sum_{n=0}^{N-1} \lambda_n$, where T_{sym} is the original symbol time of the system (i.e., the inverse of the symbol rate, R_{sym}).

To obtain the subcarrier orthogonality, the minimum spacing between two adjacent subchannels needs to be $\Delta f = \frac{1}{T_N}$, where $T_N = NT_{\text{sym}}$ is the OFDM symbol duration. By such a choice of carrier spacing the spectrum of each subcarrier will have a null at the center frequency of other subcarriers so that there is no inter-carrier interference (ICI). Normally, with a wide-band channel, the frequency response $H(f)$ is not flat and it is called frequency selective. The quantity B_C , called *coherence bandwidth*, is used to measure the frequency selectivity of a channel. It is the frequency range over which the frequency response of a channel is approximately constant. The bandwidth of each QAM subchannel, B_N , is chosen so that $B_N < B_C$. This will make sure that the frequency response is relatively constant over each subchannel's frequency range. Therefore channel equalization can be achieved by a simple division for each subcarrier. Compared with SC-QAM, OFDM channel equalization is much less complicated and hence cheaper.

However, the implementation as separate QAM modulators would require a large array of sinusoidal oscillators and is too expensive and complex for practical applications. An implementation with discrete Fourier transform (DFT) and inverse DFT (IDFT) was introduced as an effective solution to overcome the need for multiple oscillators. DFT/IDFT-based OFDM system requires only one oscillator at the transmitter and one at the receiver, which significantly lowers the system complexity.

The operations for DFT/IDFT-based OFDM transmitter and receiver are shown in Figure 2.9. The binary sequence is grouped and mapped into N QAM symbols the same way as discussed before. The outputs of the N QAM mappers form a complex symbol array $X[0], X[1], \dots, X[N-1]$. These QAM symbols are treated as if they are in the frequency domain and they are used as the input of the IDFT block to

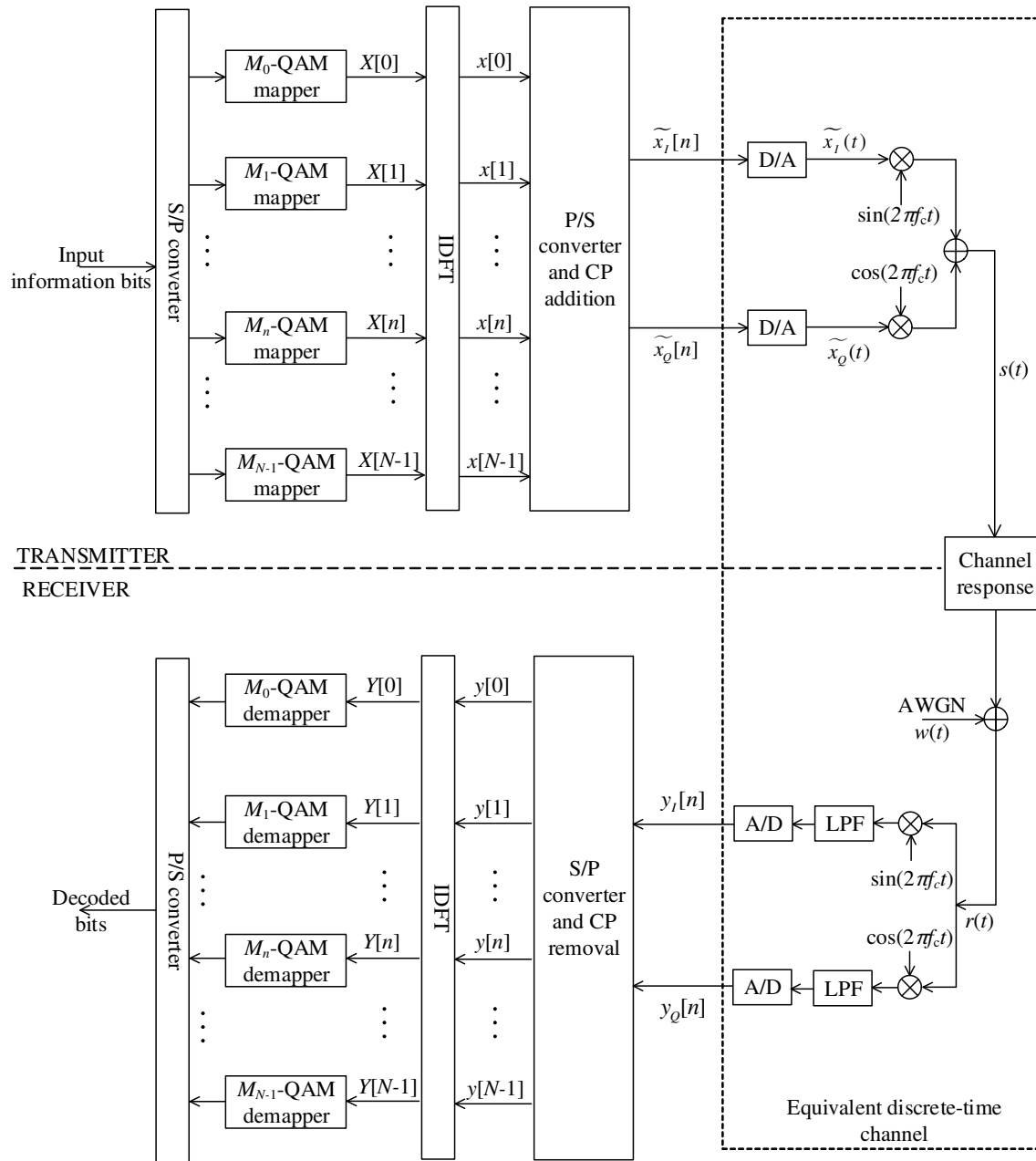


Figure 2.9: OFDM system overview with DFT/IDFT.

convert the signal into the time domain as in Equation (2.5).

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi nk}{N}}, \quad 0 \leq n \leq N-1. \quad (2.5)$$

As can be seen from the above equation, the IDFT takes N symbols at a time, each corresponding to a symbol period of T_{sym} . The output of IDFT is the summation of all N orthogonal sinusoids from all subcarriers. These sinusoids have different frequencies starting from baseband and their amplitudes and phases are determined by the whole symbol array. This method provides an easier way to modulate symbols on a large number of subcarriers. As pointed out before, the length of one OFDM block is $T_N = NT_{\text{sym}}$.

Next a *cyclic prefix* (CP) of length N_{CP} is added by replicating the last N_{CP} samples and prepending to the beginning of the sequence as shown in Figure 2.10. This step is critical since CP acts as a guard interval to eliminate ISI. A communication channel would normally introduce a delay to the transmitted signal. This delay is often measured by the length of the channel impulse response, L_{CIR} . If the guard interval N_{CP} is greater than the maximum delay of the channel, then the received cyclic prefix would contain all the interference from the previous OFDM symbol. Since this portion of data is redundant and can be discarded at the receiver with no information loss, ISI is eliminated between data blocks. The CP also provides the cyclic structure to the symbols and still maintains the orthogonality of the waveforms.

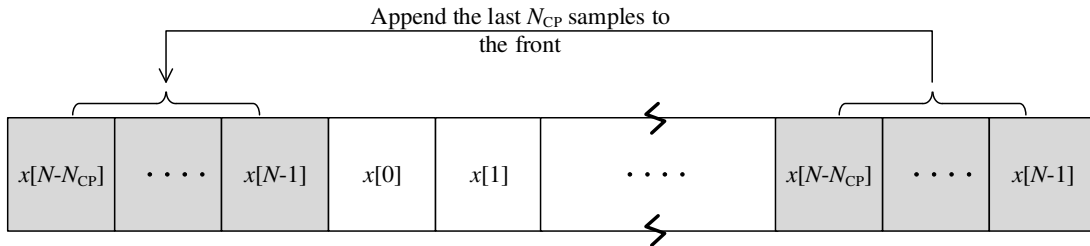


Figure 2.10: Illustration of CP extension.

The complete time-domain samples after the CP extension, denoted as $\{\tilde{x}[n]\}$,

then has the following structure:

$$\begin{aligned}\{\tilde{x}[n]\} &= \{\tilde{x}[-N_{\text{CP}}], \dots, \tilde{x}[-1], \tilde{x}[0], \dots, \tilde{x}[N-1]\} \\ &= \{x[N-N_{\text{CP}}], \dots, x[N-1], x[0], \dots, x[N-1]\}\end{aligned}\quad (2.6)$$

The time samples are grouped into inphase and quadrature components, $\tilde{x}_I[n]$ and $\tilde{x}_Q[n]$, through a parallel-to-serial converter. The inphase and quadrature components are then passed through a digital-to-analog converter to obtain the complex baseband OFDM signal $\tilde{x}(t) = \tilde{x}_I(t) + j\tilde{x}_Q(t)$. The quadrature components are upconverted to frequency F_c by the multiplying with two sinusoids that are out of phase by 90 degrees. The transmitted signal $s(t)$ is filtered by the channel impulse response $h(t)$ and gets disturbed by additive white Gaussian noise, $w(t)$. The received signal $r(t)$ at the receiver side can be represented as:

$$r(t) = s(t) * h(t) + w(t) \quad (2.7)$$

At the receiver side, the signal is downconverted to baseband to obtain the in-phase and quadrature components of the received signal. This is achieved by multiplying the input signal $r(t)$ by the quadrature carrier signals and passing through lowpass filters to remove the high frequency images caused by the multiplication. The outputs of the lowpass filters are sampled at $t = nT_s$ to obtain the discrete time signal $y[n]$, $-N_{\text{CP}} \leq n \leq N-1$. The relationship between the received samples $y[n]$ and the transmitted samples $\tilde{x}[n]$ can be characterized by an equivalent discrete-time channel of length L_{CIR} as shown in Figure 2.11.

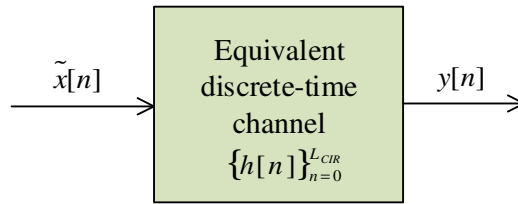


Figure 2.11: Equivalent discrete-time channel.

The multipath effect is the most prominent signal distortion that a cable channel introduces to the transmitted signal. It is caused by the connections that are not per-

fectly impedance matched along the transmission line. When there is an impedance mismatch through a connector, the transmitted signal will get reflected back and forth with reduced power after each reflection. Due to this, the CMTS will receive not just one single copy of the transmitted signal but also multiple delayed copies of the transmitted signal with different attenuations and phase shifts. The effect of a micro-reflection can be modeled as passing the transmitted signal through a linear filter $h[n]$. The filter coefficients are based on DOCSIS 3.1 which specifies the possible echo magnitudes and delays in the system. The micro-reflection specification for the upstream dominant single echo [1] is listed in Table 2.1:

−16 dBc for echo delay $\leq 0.5 \mu$ second
−22 dBc for echo delay $\leq 1.0 \mu$ second
−29 dBc for echo delay $\leq 1.5 \mu$ second
−35 dBc for echo delay $> 2.0 \mu$ second
−42 dBc for echo delay $> 3.0 \mu$ second
−51 dBc for echo delay $> 4.5 \mu$ second

Table 2.1: Micro-reflection bound for upstream dominant single echo.

The micro-reflections in the time domain translate to magnitude ripples in the frequency domain. For broadband SC-QAM transmission, this requires more complicated channel equalization. However, for the DOCSIS 3.1 OFDM system the subchannel bandwidth B_N is chosen to be smaller than the coherence bandwidth B_C which makes the frequency responses of all subchannels constant (i.e. flat). Meanwhile, due to the cyclic structure and the way CP is formed, the OFDM block now appears periodic when convolved with the channel. That is $\tilde{x}[n - m] = x[(n - m) \bmod N]$. The linear convolution is then converted into a circular convolution as shown in Equation (2.8) for data samples that are not corrupted by ISI, where $h[n]$ represents the

channel impulse response of a multipath channel.

$$\begin{aligned}
y[n] &= \tilde{x}[n] * h[n] = \sum_{m=0}^{L_{\text{CIR}}} h[m] \tilde{x}[n-m] \\
&= \sum_{m=0}^{L_{\text{CIR}}} h[m] x[(n-m) \bmod N] = x[n] \otimes h[n], \quad 0 \leq n \leq N-1
\end{aligned} \tag{2.8}$$

The length of $y[n]$ is $N + N_{\text{CP}} + L_{\text{CIR}} - 1$ and the last L_{CIR} samples act as interference to the next OFDM symbol as shown in Figure 2.12. Due to the redundancy of the added cyclic prefix, the first N_{CP} samples are not needed and can be discarded. The remaining N samples are converted back to the frequency domain through the DFT operation as in Equation (2.9).

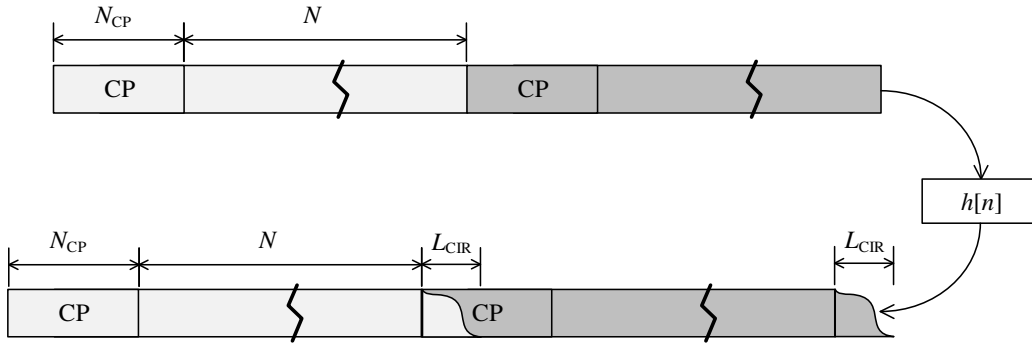


Figure 2.12: Channel delay effect.

$$\begin{aligned}
Y[k] &= \sum_{n=0}^{N-1} y[n] e^{-j\frac{2\pi nk}{N}} = \sum_{n=0}^{N-1} (x[n] \otimes h[n] e^{-j\frac{2\pi nk}{N}}), \quad 0 \leq k \leq N-1, \\
&= \sum_{n=0}^{N-1} \sum_{m=0}^{L_{\text{CIR}}} h[m] x[(n-m) \bmod N] e^{-j\frac{2\pi nk}{N}} \\
&= \sum_{m=0}^{L_{\text{CIR}}} \left(\sum_{n=0}^{N-1} x[(n-m) \bmod N] e^{-j\frac{2\pi nk}{N}} \right) h[m] \\
&= \sum_{m=0}^{L_{\text{CIR}}} \left(\sum_{n=0}^{N-1} x[(n-m) \bmod N] e^{-j\frac{2\pi(n-m)k}{N}} \right) h[m] e^{-j\frac{2\pi mk}{N}} \\
&= X[k] \sum_{m=0}^{L_{\text{CIR}}} h[m] e^{-j\frac{2\pi mk}{N}} = X[k] H[k]
\end{aligned} \tag{2.9}$$

As shown above, the DFT for the circular convolution of $x[n]$ and $h[n]$ returns element-wise multiplication between $X[k]$ and $H[k]$, where $H[k]$ is the DFT of $h[n]$, i.e., the channel frequency response. Since each element of $X[k]$ and $Y[k]$ correspond to one subcarrier, it can be seen that each subcarrier QAM symbol is multiplied with one complex channel frequency response element, namely $H[k]$. Therefore the effect of the channel can be compensated by a simple division in the frequency domain at the receiver side, namely $X[k] = Y[k]/H[k]$. Therefore channel equalization can be implemented using one-tap equalization. Once the transmitted symbols are obtained, the original transmitted data sequence can be recovered by demodulation.

2.3 Summary

In this chapter, some background knowledge on OFDM systems was presented. First, the basic operations of the single-carrier QAM system was reviewed. Then, the concept of multi-carrier QAM system in the form of OFDM is introduced. Compared to a single-carrier QAM system, the OFDM system benefits from requiring only one-tap channel equalization when used to provide high transmission rates. This is possible by an elegant use of the cyclic prefix. It should be pointed out, however, that the relationship in Equation (2.9) is only valid under the assumption of perfect timing, frequency and phase synchronization. Since timing, frequency and phase offsets destroy this relationship, synchronization is a critical step in the applications and implementation of a practical OFDM system. The following chapter discusses in detail the effects of these offsets and presents some basic timing synchronization principles.

3. Basic Principles of Timing Synchronization in OFDM Systems

A multipath channel introduces various signal distortions to an OFDM system such as frequency, phase and timing offsets, which need to be properly compensated in the synchronization process. In general, OFDM systems are much more sensitive to synchronization errors than single-carrier systems [16] since these offsets may destroy the orthogonality between the subcarriers. This makes timing and frequency synchronizations critical parts in OFDM receiver design. The main subject of this thesis is timing synchronization in DOCSIS 3.1 upstream transmission.

There are two main objectives for the timing synchronization process. First, it must be able to detect the presence of a new frame in the received data stream. Second, it needs to provide an estimation on the timing offset value to locate the correct position for the FFT window at the receiver side. Given the popularity of OFDM technology, there has been a lot of research done on the topic of OFDM timing synchronization in the past decades. This chapter first provides detailed analysis on the frequency and timing offsets, their causes and effects in OFDM systems. Then it gives an overview of the most common timing synchronization methods developed in the past for OFDM systems.

3.1 Frequency, Phase and Timing Offsets in OFDM Systems

To understand the timing synchronization process, it is essential to first get some insight on the timing offset itself and other closely related impairments. In most digital communications systems, timing synchronization is the first step in the syn-

chronization process. Therefore the timing synchronization must work with frequency and phase offsets present in the received signal. Thus it is helpful to obtain a general knowledge on these offsets first. Then in the second subsection, cause and effect of timing offset will be discussed in detail.

3.1.1 Frequency and Phase Offsets

Frequency and phase offsets are often due to the mismatch between the oscillators of the transmitter and receiver, which are used for upconversion and downconversion. Since the oscillators of the receiver and transmitter each work on its own and are not synchronized externally, there are likely to be mismatches in the frequency and phase of the sinusoidal carrier signals. Denote the argument of the transmitter oscillator sinusoid as $2\pi F_c t$ and the receiver oscillator sinusoid as $2\pi (F_c + \Delta F) t + \Phi_0$, where ΔF is the frequency offset and Φ_0 is the phase offset. The effect of frequency and phase offsets results in rotation of the constellation points. The phase offset causes a fixed rotation of Φ_0 radians for all symbols, whereas the frequency offset causes the constellation points to rotate at a rate of $2\pi\Delta F$ radians per second.

For a QAM system, the rotations of the constellation points caused by frequency and phase offsets complicate the task of symbol demapper at the receiver side. The modulation scheme depends on the position of the constellation points to acquire information on the signal. Furthermore, as a multi-carrier transmission technique, OFDM is more susceptible to the carrier frequency offset (CFO) and phase offset than single carrier systems [16]. As all subcarriers are closely spaced in frequency, it is important to maintain the orthogonality between carriers to avoid interference with each other.

With frequency and phase offsets, the received signal at the outputs of the A/D blocks in Figure 2.9 can be represented as $y_{\text{offset}}[n] = y[n] e^{j2\pi(\Delta f n + \phi_0)}$ where $y[n] = y_I[n] + jy_Q[n]$, $\Delta f = \Delta F/F_s$ and $\phi_0 = \Phi_0/F_s$. To illustrate the effect of Δf and ϕ_0 , the operation of the DFT block as shown in Equation(2.8) can be simplified by assuming $y[n]$ to be an unit-amplitude complex sinusoid signal with frequency k_1/N .

Therefore, $y[n] = e^{\frac{j2\pi k_1 n}{N}}$ where $k_1 \in 0, 1, \dots, N-1$ and it can be considered as a QAM symbol of 1 modulated with a single subcarrier of frequency k_1/N . Ideally, with no frequency or phase offset the DFT output would be:

$$Y[k] = \sum_{n=0}^{N-1} e^{\frac{j2\pi n(k_1-k)}{N}} = \begin{cases} \sum_{n=0}^{N-1} 1 = N, & \text{if } k = k_1 \\ 0, & \text{otherwise} \end{cases} \quad (3.1)$$

It can be seen that when $k_1 \neq k$, the time-domain received signal has zero contribution to other subcarrier frequencies. However with frequency and phase offsets, the received signal becomes: $y_{\text{offset}}[n] = y[n] e^{j2\pi(\Delta f n + \phi_0)} = e^{j2\pi(n(K_1/N + \Delta f) + \phi_0)}$ and the DFT equation is:

$$\begin{aligned} Y[k] &= \sum_{n=0}^{N-1} e^{j2\pi(n(k_1/N + \Delta f) + \phi_0)} e^{-\frac{j2\pi k n}{N}} \\ &= e^{j2\pi\phi_0} \sum_{n=0}^{N-1} e^{j2\pi n((k_1-k)/N + \Delta f)} \\ &= \begin{cases} e^{j2\pi\phi_0} \sum_{n=0}^{N-1} e^{j2\pi n\Delta f} = e^{j2\pi\phi_0} \frac{1 - e^{j2\pi\Delta f(N-1)}}{1 - e^{j2\pi\Delta f}}, & \text{if } k = k_1 \\ e^{j2\pi\phi_0} \frac{1 - e^{j2\pi((k_1-k)/N + \Delta f)(N-1)}}{1 - e^{j2\pi((k_1-k)/N + \Delta f)}}, & \text{otherwise} \end{cases} \end{aligned} \quad (3.2)$$

The above expression shows that with frequency offset, the DFT output of $y_{\text{offset}}[n]$ has non-zero components for other subcarrier frequencies as well. This introduces inter-carrier interference (ICI) and the signal transmitted in one subcarrier will be interfered by the signals transmitted on adjacent subcarriers. Therefore the receiver must be able to estimate and compensate for frequency and phase offsets accurately to be able to make proper decisions.

3.1.2 Timing Offset

Once the signal is downconverted to baseband and filtered at the receiver side, the signal needs to be sampled properly to recover the transmitted symbols. Ideally the correct sampling time is at $t = nT_{\text{sym}}$. However due to the unknown propagation delay of τ and the unsynchronized oscillators at the transmitter and receiver, there is a timing error or offset by θ . This timing offset θ is obtained by normalizing

the time delay τ to the system clock period T_S . It has no unit and it is equal to the number of delayed samples between the transmitted signal $s[n]$ and the received signal $s[n - \theta]$. As described in Chapter 2, IDFT and DFT functions are required in an OFDM system to implement the modulation and demodulation processes. In DOCSIS OFDM systems, the number of subcarriers is always an exponent of 2, therefore the IDFT/DFT process can be accomplished by IFFT/FFT process. From here on in this paper, IFFT/FFT will be used to represent IDFT/FFT process. The correct samples of the transmitted signal are needed in order to perform the IFFT/FFT operations. A timing offset of θ changes the location of the starting point of each OFDM symbol and therefore the FFT window position. It is more difficult to locate the starting position of the FFT window with timing offset. ISI will occur if there are samples from more than one symbol inside the FFT window.

Denote the sample indexes of a perfectly synchronized OFDM symbol by $\{-N_{CP}, \dots, -1, 0, 1, \dots, N - 1\}$ and the maximum channel delay spread by L_{CIR} . As shown in Figure 3.1, the beginning of the FFT window at the receiver side contains a few samples from the previous block due to the timing offset θ_1 (see the middle panel). Such ISI will likely result in errors in the process of symbol recovery. It is therefore necessary to design an effective timing synchronization circuitry to estimate the timing offset value. The estimated value is then fed into an interpolation filter which processes the incoming samples and return the correctly sampled values. In this thesis, estimation of the integer portion of θ is considered. Although θ can contain a fractional portion, its effect is treated as a phase shift and compensated in the channel equalization stage.

For a well designed system with CP length N_{CP} greater than the channel impulse response length L_{CIR} , there is a safety window inside the CP region where the samples are not affected by the tail of the previous symbol as shown in Figure 3.1 (see top panel). As long as the timing offset is within this safety region, i.e., $-N_{CP} + L_{CIR} - 1 \leq \theta \leq 0$, all the data samples in the FFT window still belong to the same OFDM symbol. This is illustrated in the bottom panel of Figure 3.1 where the timing offset θ_2 is within

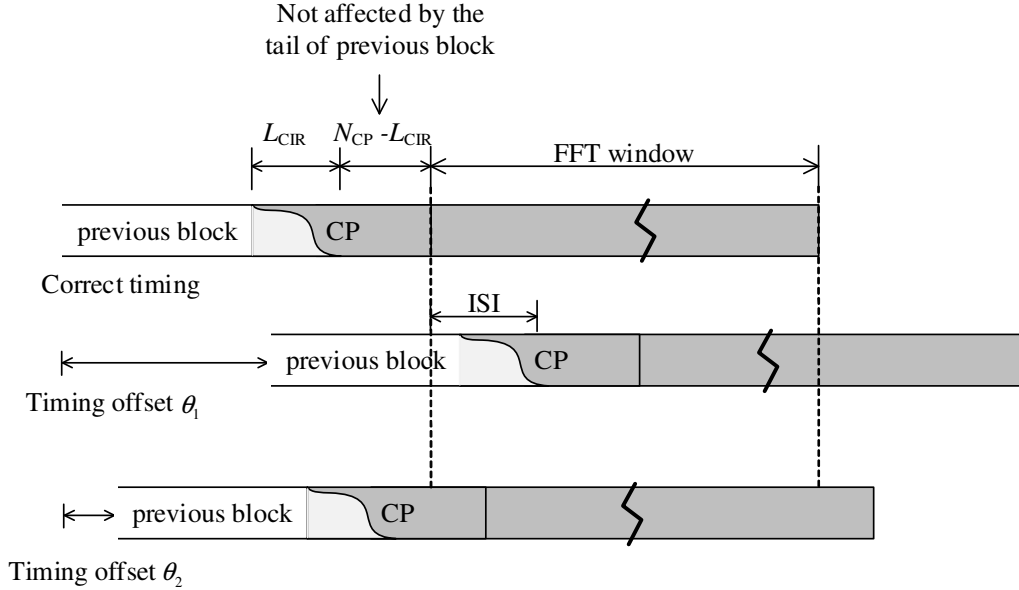


Figure 3.1: Timing offset in an OFDM system.

the safety window. Due to the cyclic structure, this type of timing estimation error results in a linear phase shift across all subcarriers. However, if the timing error is beyond this safety region, the FFT window contains interferences from other symbols and it increases the probability of symbol recovery error.

As discussed above, if the timing offset $\theta \in \{-N_{CP} + L_{CIR} - 1, \dots, 0\}$ the orthogonality among the subcarriers is not destroyed and no ISI is introduced. A timing offset θ in the time domain causes a phase shift of $2\pi k\theta/N$ in the frequency domain where k is the subcarrier index [18]. The timing offset θ introduces a phase rotation in every subcarrier symbol. This can be shown by taking the FFT of the time-domain received

samples $\{y_\theta[n]\}_{n=0}^{N-1} = \{y[n-\theta]\}_{n=0}^{N-1} = \{\{y[n+N-\theta]\}_{n=0}^{N-\theta-1}, \{y[n-\theta]\}_{n=\theta}^{N-1}\}$

$$\begin{aligned}
Y_\theta[k] &= \frac{1}{N} \sum_{n=0}^{N-1} y_\theta[n] e^{-\frac{j2\pi nk}{N}} \\
&= \frac{1}{N} \left(\sum_{n=0}^{\theta-1} y[n+N-\theta] e^{-\frac{j2\pi nk}{N}} + \sum_{n=\theta}^{N-1} y[n-\theta] e^{-\frac{j2\pi nk}{N}} \right) \\
&= \frac{1}{N} \left(\sum_{n=0}^{\theta-1} y[n+N-\theta] e^{-\frac{j2\pi(n+N-\theta)k}{N}} e^{\frac{j2\pi(N-\theta)k}{N}} + \sum_{n=\theta}^{N-1} y[n-\theta] e^{-\frac{j2\pi(n-\theta)k}{N}} e^{-\frac{j2\pi(\theta)k}{N}} \right) \\
&= \frac{e^{-\frac{j2\pi\theta k}{N}}}{N} \left(\sum_{p=N-\theta}^{N-1} y[p] e^{-\frac{j2\pi pk}{N}} + \sum_{q=0}^{N-\theta-1} y[q] e^{-\frac{j2\pi qk}{N}} \right) \\
&= e^{-\frac{j2\pi\theta k}{N}} \sum_{n=0}^{N-1} y[n] e^{-\frac{j2\pi nk}{N}} = e^{-\frac{j2\pi\theta k}{N}} Y[n]
\end{aligned} \tag{3.3}$$

The effect of such a rotation is illustrated in Figure 3.2 for QPSK constellation. For illustration purpose, only the first four subcarriers' transmitted and received symbols are shown in the figure. It is intentionally assumed that the positions of the four symbols for subcarriers 0 to 3 are next to each other in a counter-clockwise fashion. This gives a visual demonstration to show that the phase shift is proportional to the subcarrier index k . In the simulation, 2048 subcarriers are used to perform FFT and the cyclic prefix value is 32, channel impulse response has an order of 5. The safety window thus is $\{-27, \dots, 0\}$ and the timing offset is chosen to be 15 so that it lies into the safety window.

However if the time offset is outside the safety range, the orthogonality among the subcarriers will be destroyed, resulting in inter-symbol interference (ISI) and additional inter-carrier interference (ICI). Figure 3.3 illustrates the transmitted and received symbols for a channel with a timing offset that causes ISI. In such a case, there is no clear relationship between the transmitted and received symbols anymore.

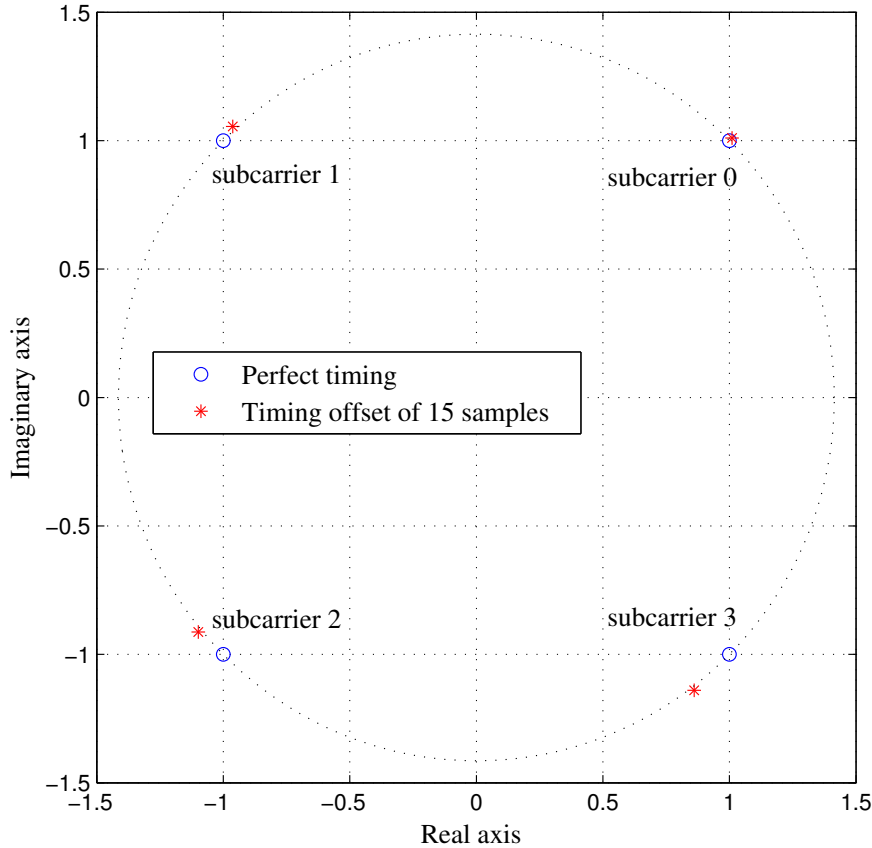


Figure 3.2: Frequency-domain phase shift due to a timing error that does not cause ISI.

3.2 Review of Timing Synchronization Algorithms

The OFDM synchronization techniques can be divided into data-aided and non-data-aided categories. In OFDM systems, the transmission is typically organized in frames. A transmission frame consists of multiple OFDM symbols/blocks. The data-aided techniques often use a training sequence or pilot symbols to help with timing estimation. In such data-aided transmission frames, the pilot/preamble symbols are appended in the front of the data symbols. The cyclic extension process provides a good autocorrelation property to the signal since the first N_{CP} samples are the same as the final N_{CP} samples at the end. The non-data aided techniques often use the cyclic prefix correlation such as in [19] [20] [11]. Data-aided methods have high

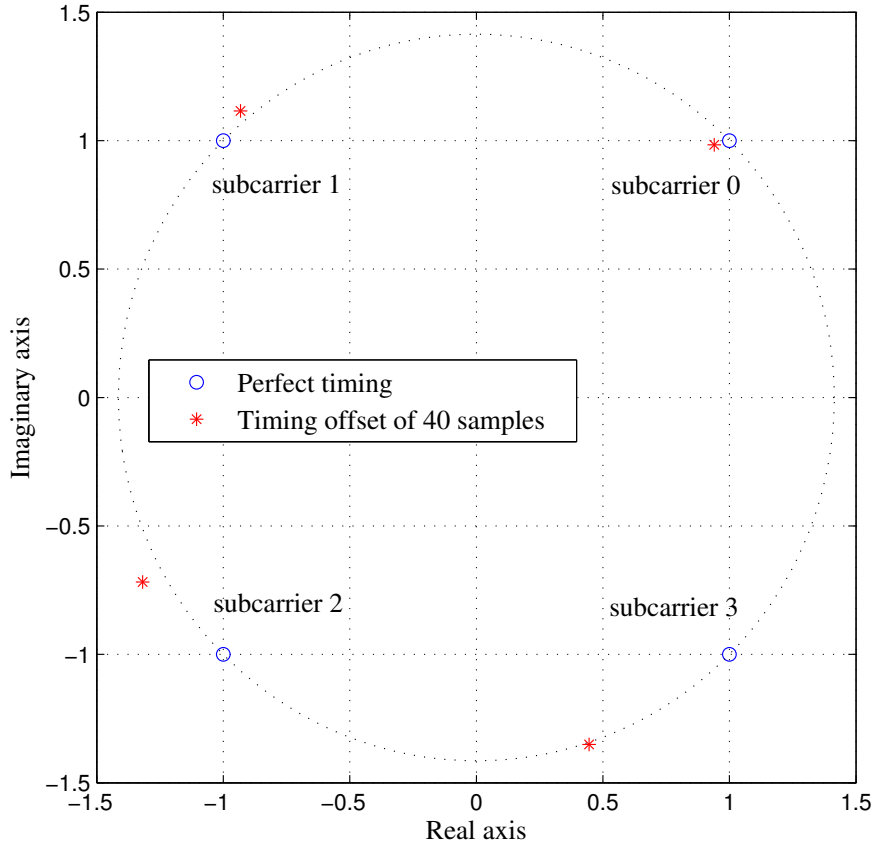


Figure 3.3: Frequency-domain phase shift due to a timing error that causes ISI.

accuracy and low computation cost, but reduce the data transmission rate due to the overhead of sending training sequence [7–10, 12, 18, 21, 22]. Non-data aided methods do not reduce the transmission rate since training symbols or pilot signals are not required, but they are typically less robust. They rely on the signal integrity of the cyclic prefix portion to be intact. In multipath channels, however, the guard interval (cyclic prefix) is heavily corrupted by ISI from the previous symbol. Therefore the performance of non-data aided methods is not good in the case of ISI and shall not be discussed any further in this thesis.

3.2.1 Schmidl and Cox (S&C) Timing Algorithm

One popular approach to estimate timing error is to use training blocks exhibiting a repetitive pattern in the time domain. In this way, timing estimation can be realized by searching for the peak of the autocorrelation among the repetitive parts. Schmidl and Cox proposed a method for timing synchronization with such a preamble [12]. The proposed preamble symbol consists of two identical halves of length $N/2$ in the time domain and it is placed at the beginning of a frame. It has the following pattern:

$$S_S = [A, A] \quad (3.4)$$

Such a repetitive pattern can be generated by transmitting a Pseudo-Noise (PN) sequence on subcarriers with even indexes and transmitting zero on subcarriers with odd indexes. As long as the cyclic prefix length N_{CP} is longer than the channel impulse response length L_{CIR} , the two halves of the reference will remain identical after transmitting over the channel, except for the phase shift induced by the carrier frequency offset as explained in Section 3.1. Then to detect a new frame, a correlation calculation is performed over the time domain samples in two consecutive windows of length $N/2$ where N is the symbol length as shown in Figure 3.4.

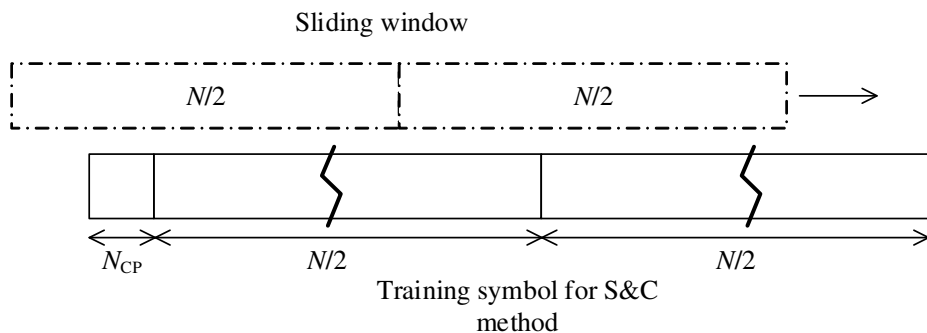


Figure 3.4: Sliding-window correlator for Schmidl and Cox timing algorithm.

The conjugate of a sample from the first window is multiplied by the corresponding sample from the second window. The windows are slid across the received signal. When the two windows are perfectly aligned with the received preamble symbol pair,

the products of each of these pairs of samples will have approximately the same phase and the magnitude of the sum will reach a maximum value. Denote the estimated timing offset by $\hat{\theta}$ and timing estimation error by $\Delta\theta = \hat{\theta} - \theta$. then timing offset $\hat{\theta}$ can be obtained as

$$\hat{\theta} = \operatorname{argmax}\{|\Gamma_S(\tilde{\theta})|\} \quad (3.5)$$

where $\tilde{\theta}$ is the index of received signal samples and $\Gamma_S(\tilde{\theta})$ is the autocorrelation timing metric function defined as in Equation (3.6):

$$\Gamma_S(\tilde{\theta}) = \frac{\sum_{m=\tilde{\theta}}^{\tilde{\theta}+N/2-1} y[m+N/2] y^*[m]}{\sum_{m=\tilde{\theta}}^{\tilde{\theta}+N/2-1} |y[m+N/2]|^2} \quad (3.6)$$

Figure 3.5 shows an example of the timing metric, $|\Gamma_S(\tilde{\theta})|$, as a function of the timing estimation error $\Delta\theta = \hat{\theta} - \theta$. The results are obtained over a multipath channel with CIR of $L_{\text{CIR}} = 5$. There is a total of $N = 256$ subcarriers and the signal-to-noise ratio is set to be 20 dB. As shown in the figure, the timing metric exhibits a large “plateau” that leads to uncertainty in determining the start of the symbol. The plateau is caused by the use of the cyclic prefix. Since the cyclic prefix is a duplicate of the end of a symbol, it is also identical to end of the first half of the training symbol. This prolongs the matching span where the two sliding windows are identical from 1 sample to N_{CP} samples. In this specific simulation example, N_{CP} value is 16.

3.2.2 Shi and Serpedin (S&S) Timing Algorithm

To overcome the problem with the plateau that was encountered with Schmidl and Cox’s timing metric, Shi and Serpedin [9] introduced a new training symbol by modifying the training sequence’s pattern proposed by Schmidl and Cox. To give a sharper timing metric, Shi and Serpedin proposed using a training block composed of four repetitive parts with the following pattern:

$$S_M = [A, A, -A, A]. \quad (3.7)$$

In the above pattern, A represents a sequence of samples of length $N/4$ generated by taking the $N/4$ point IFFT of a PN-sequence. This specific pattern leads to

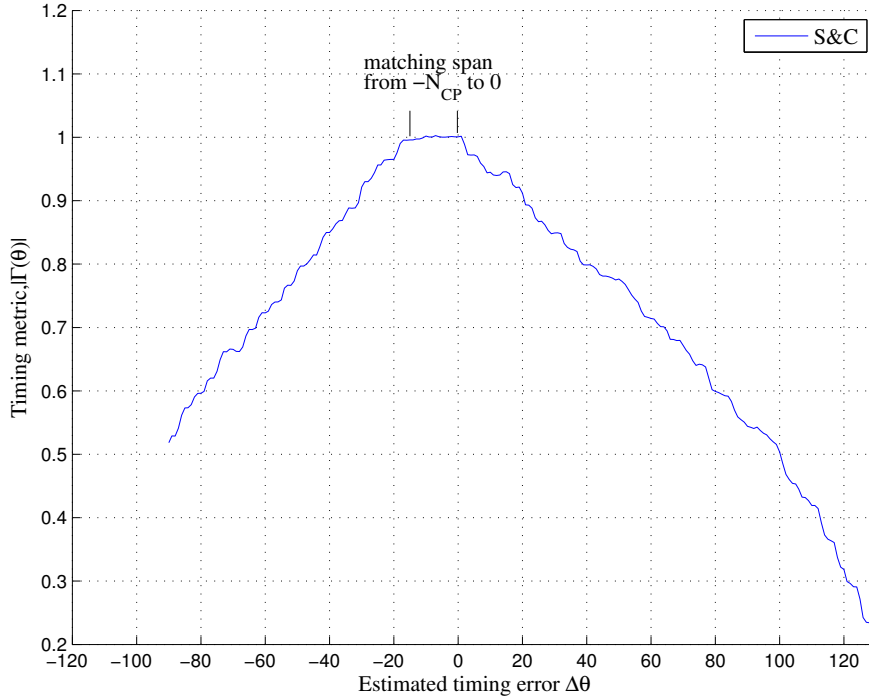


Figure 3.5: An example of timing metric for Schmidl and Cox timing algorithm.

a timing acquisition scheme that exhibits better detection properties. As a matter of fact the sign inversion on the third segment can be placed in any location without any loss in performance. Similarly, a correlation calculation is performed over samples in four consecutive windows. The four windows are defined by equally partitioning a single window of length N that includes the samples with indexes $\tilde{\theta} \leq n \leq \tilde{\theta} + N - 1$. The samples in the four mini windows can be represented as $y_j(\tilde{\theta}) = \{y[l + jN/4 + \tilde{\theta}]; 0 \leq l \leq N/4 - 1\}$ where $j = 0, 1, 2, 3$ and $\tilde{\theta}$ is the index of the received signal samples. Then the timing metric is computed as:

$$\Gamma_M(\tilde{\theta}) = \frac{|\Lambda_1(\tilde{\theta})| + |\Lambda_2(\tilde{\theta})| + |\Lambda_3(\tilde{\theta})|}{\frac{3}{2} \sum_{j=0}^3 \|y_j(\tilde{\theta})\|^2} \quad (3.8)$$

where

$$\begin{aligned}
\Lambda_1(\tilde{\theta}) &= y_0^H [\tilde{\theta}] y_1 [\tilde{\theta}] - y_1^H [\tilde{\theta}] y_2 [\tilde{\theta}] - y_2^H [\tilde{\theta}] y_3 [\tilde{\theta}] \\
\Lambda_2(\tilde{\theta}) &= y_1^H [\tilde{\theta}] y_3 [\tilde{\theta}] - y_0^H [\tilde{\theta}] y_2 [\tilde{\theta}] \\
\Lambda_3(\tilde{\theta}) &= y_0^H [\tilde{\theta}] y_3 [\tilde{\theta}]
\end{aligned} \tag{3.9}$$

Figure 3.6 shows the comparison of the two timing metrics for S&C and S&S algorithms. The red line indicates the timing metric for the S&S algorithm. The metrics are generated under the same system setup. As can be seen, the S&S method provides a much sharper peak compared to the S&C method.

However for the S&S method, the desired pattern involves four repetitive segments while the third segment has a reversed sign compared to the other three. This requires a manual sign inversion to be performed on the third segment which may not be possible in certain applications.

3.2.3 Park and Cheon (P&C) Timing Algorithm

This method was proposed as a modification of the S&C and S&S methods to further reduce the variation of the estimation error. Specifically, Park et al [24] proposed a new training symbol to increase the difference between the peak value and other values of the timing metric. The method involves modifying the training sequence and timing metric definition to maximize the different pairs of product between them. The pattern of the training sequence is given by:

$$S_P = [A, B, A^*, B^*], \tag{3.10}$$

where A represents the samples of length $L = N/4$ generated by IFFT of a PN sequence, A^* represents the conjugate of A and B is designed to be symmetric with A . A training symbol of such a pattern can be obtained by transmitting a real-valued PN sequence on the even frequencies while zeros on the odd frequencies. To make

use of the symmetry between A and B , the timing metric is defined as:

$$\Gamma_P [\tilde{\theta}] = \frac{\|P_3(\tilde{\theta})\|^2}{(R_3(\tilde{\theta}))^2}, \quad (3.11)$$

where

$$P_3(\tilde{\theta}) = \sum_{k=0}^{N/2} y[\tilde{\theta} - k] y[\tilde{\theta} + k] \quad (3.12a)$$

$$R_3(\tilde{\theta}) = \sum_{k=0}^{N/2} \|y[\tilde{\theta} + k]\|^2 \quad (3.12b)$$

The $P_3(\tilde{\theta})$ function is designed such that the proposed timing metric has its peak value at the correct symbol timing, while the values at all other positions are almost zero.

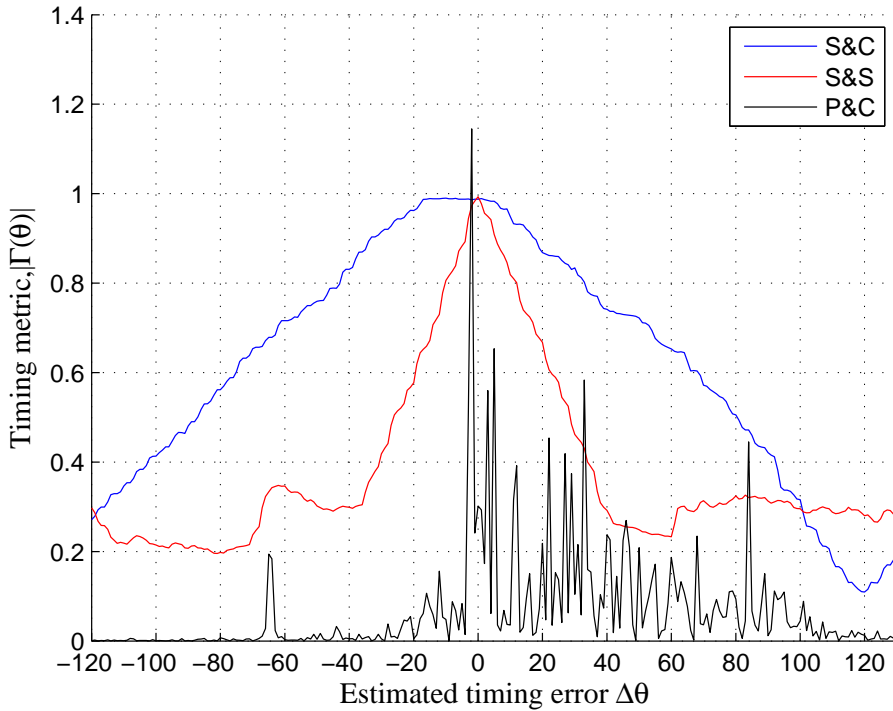


Figure 3.6: Timing metrics for S&C, S&S and P&C timing algorithms.

Figure 3.6 shows a comparison of the timing metrics for all three methods that have been discussed in this section. All three metrics are generated under the same

system setup. It is clear from the figure that the P&C method provides the sharpest peak by far.

3.3 Summary

This chapter discussed timing synchronization issues in OFDM system. Due to the high sensitivity to synchronization errors of OFDM systems, it is critical to make sure that the receiver is correctly synchronized with the transmitter in timing and frequency. Three classical data-aided approaches to accomplish timing synchronization in OFDM systems were discussed in this chapter. Some of the more recent research approaches to achieve timing synchronization have improved performance or reduced complexity, but they always require specific training symbol patterns [21] [18]. Given the specific constraints and requirements imposed by DOCSIS 3.1, none of the methods discussed in this chapter can be directly applied for DOCSIS 3.1 OFDM systems. In the next chapter, three new timing synchronization algorithms are proposed for DOCSIS 3.1 OFDM receiver.

4. Timing Synchronization Algorithms for DOCSIS 3.1 Upstream Transmission

The previous chapter discussed in detail the effect of timing offset on OFDM systems and reviewed several conventional methods to estimate the timing offset. However, as will be demonstrated later, those methods cannot be readily applied to DOCSIS 3.1 upstream transmission that is based on the Orthogonal Frequency Division Multiplexing Access (OFDMA) technique. For OFDMA, the available frequency spectrum is shared among all CMs connected to the CMTS and each CM only transmits on the subchannels that are assigned to it. Timing synchronization for upstream OFDMA is therefore more challenging than downstream OFDM because each CM connected to the network has a unique timing offset value and the CMTS needs to estimate and correct for each CM accordingly.

This chapter first discusses the ranging process in DOCSIS 3.1, which is used for timing synchronization. Three different methods are then proposed which utilize DOCSIS3.1 ranging signal structure to achieve timing offset estimation.

4.1 Ranging Process in DOCSIS 3.1 Upstream Transmission

In DOCSIS 3.1 upstream transmission, ranging is the process of acquiring the correct timing offset so that all the CMs' transmissions are aligned. When a new CM tries to connect to the network, the CMTS needs to perform a ranging process which is a combination of initial ranging, fine ranging and probing in order to calibrate for timing, frequency and power, as well as obtaining channel information for the new CM. Figure 4.1 is the process flow diagram for the ranging process in DOCSIS 3.1

system [1]. *Initial ranging* is used by the CMTS for identifying a new admitting CM and performing coarse power and timing calibration. After initial ranging has been completed, *fine ranging* is used for fine timing and power tuning. The fine ranging process may be repeated if needed to achieve the desired precision. *Wideband probe* is for channel pre-equalization configuration. Note that both fine ranging and wideband probe are used in two stages - admission stage and steady state stage. The CMTS periodically commands the CMs to send upstream probing and ranging to check the quality of the upstream OFDMA signal and adjust for timing shift and power.

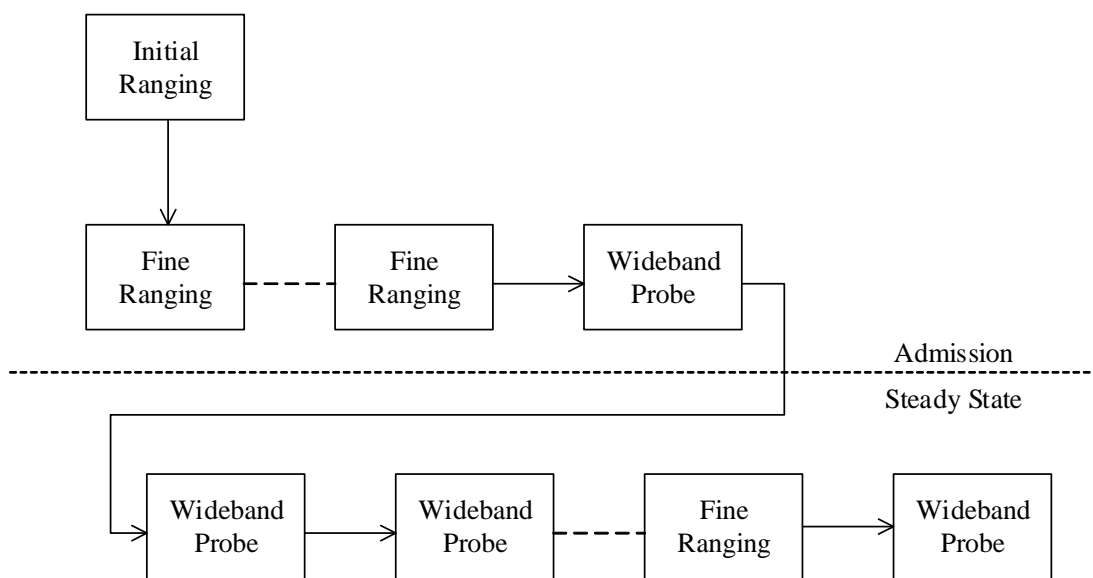


Figure 4.1: Ranging steps for DOCSIS 3.1 [1].

CM upstream frequency and timing of transmissions are based on downstream tracking, and in the case of timing, also based upon receiving and implementing timing adjustments from the CMTS. The process of interest here is fine ranging which is performed in both admission and steady states for fine time tuning. A CM sends out a ranging request (RNG-REQ) signal to the CMTS to perform fine ranging when needed. Based on the received ranging request signal, the CMTS will then calculate and send ranging response (RNG-RSP) back to the CM which includes information needed for time, frequency and power adjustment. Pre-equalization coefficients information will also be included in the ranging response signal. The CM then uses the

adjustment and pre-equalization values to calibrate for timing, phase and power to compensate for the distortions of a DOCSIS channel.

As Chapter 3 has introduced, timing synchronization is achieved by making use of the specific pattern of the training symbol. In order to obtain a better understanding of the possible training symbols DOCSIS 3.1 can provide, the detailed structure for the ranging signals is examined in the following subsections.

4.1.1 Ranging Request Signal (RNG-REQ)

To help understand the details of the RNG-REQ signal structure better, three new terms need to be introduced: *minislot*, *roll-off period* and *cyclic suffix*. Based on the FFT size, there are two transmission modes for DOCSIS OFDM system, 2k and 4k modes, corresponding to DFT sizes of 2048 and 4096 respectively. With the transmission mode known, the CMTS must allocate subcarriers to each CM in groups. Such a subcarrier group is called a minislot. A minislot contains a group of Q subcarriers, where the value of Q depends on the FFT size: Q is either 8 or 16 for 2k or 4k mode, respectively. It has been made clear in Section 2.2 that a cyclic prefix is always added to the beginning of an OFDM frame to avoid ISI. In addition to appending the cyclic prefix, for DOCSIS 3.1, a roll-off period is also appended to the symbol by duplicating the first N_{RP} samples of the symbol and paste to the end of the symbol as shown in the second panel in Figure 4.2. The CP&RP-extended sequence is now of length $(N + N_{RP} + N_{CP})$. The N_{RP} samples at both ends of this extended sequence are subject to tapering. This tapering is achieved using a raised-cosine window function. The window is applied to the entire extended sequence which has a flat top and raised-cosine tapering at both edges as shown in the two bottom panels in Figure 4.2.

The fine ranging signal that is discussed in this section is the ranging request signal transmitted by the CM. Based on the physical layer specifications for DOCSIS 3.1 [1], the structure of the fine ranging signal is shown as in Figure 4.3.

The vertical axis represents the frequency domain. The available frequency spec-

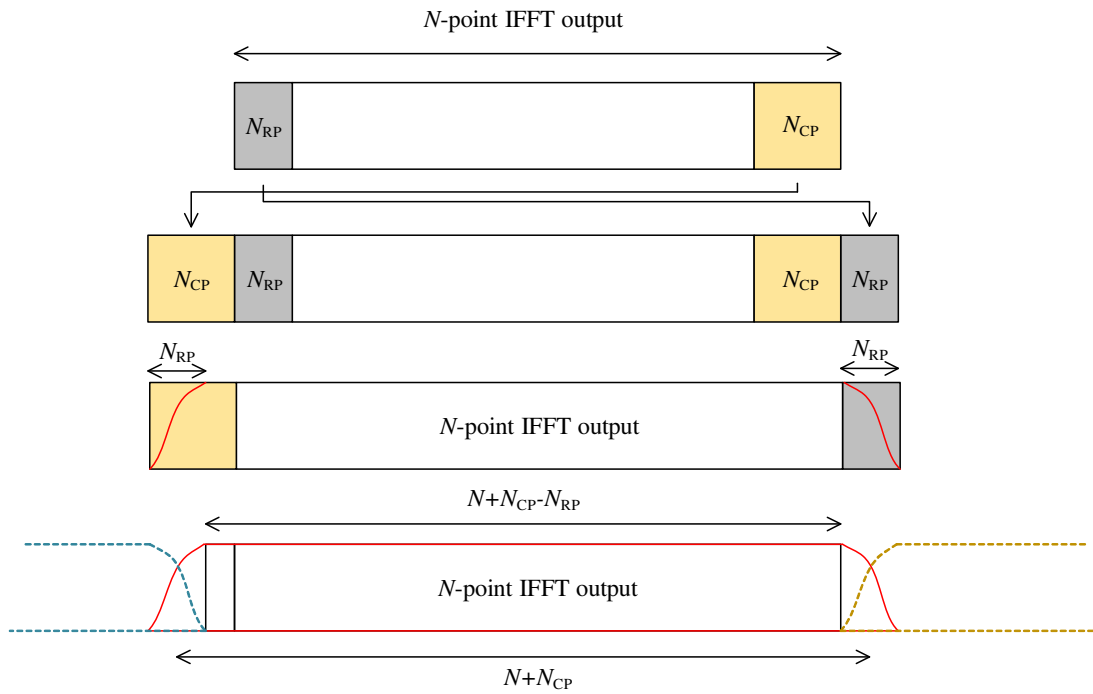


Figure 4.2: Cyclic prefix, roll-off period and windowing of OFDM frames in DOCSIS 3.1. [1].

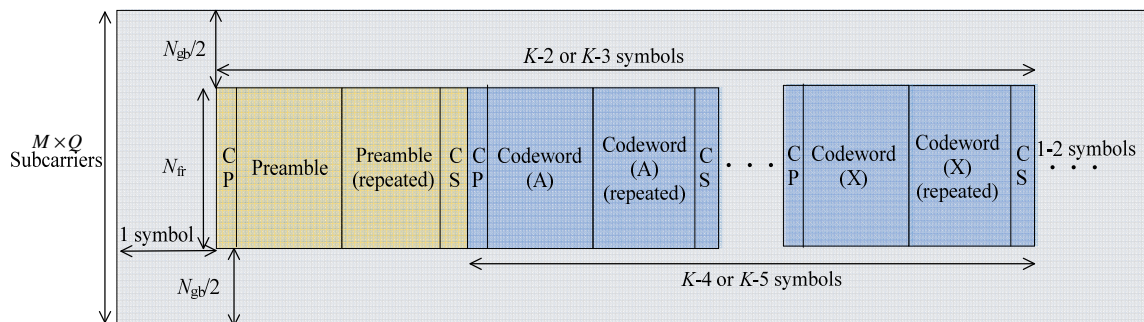


Figure 4.3: Fine ranging signal structure in DOCSIS 3.1.

trum is divided into a maximum of N subcarriers. The CMTS allocates M contiguous minislots within an OFDMA frame for fine ranging. Within this $M \times Q$ number of subcarriers, N_{fr} subcarriers are used for the actual ranging data transmission and $N_{\text{gb}}/2$ subcarriers act as a guard-band from each side of the data transmission subcarriers. The CM must transmit zero-valued subcarriers in the guard-band. The relationship between M , N_{fr} and N_{gb} is as follows:

$$M \times Q = N_{\text{gb}} + N_{\text{fr}}.$$

The horizontal axis represents the time domain. The ranging request frame consists of K OFDM symbols. The value of K is set by the CMTS. The ranging OFDM symbols are constructed differently from normal transmitting symbols. There is 1 empty symbol at the beginning of the ranging frame. The first non-zero symbol in the ranging frame is a preamble symbol. The ranging preamble symbol transmits a BPSK binary sequence on the assigned N_{fr} subcarriers. The length and content of the BPSK preamble sequence are configured by the CMTS and known by the CM. Then data symbols of the fine ranging signal are transmitted after the preamble symbol. The data symbols are QPSK-modulated and FEC encoded. Both the preamble and the coded data symbols are duplicated and transmitted as pairs [1] with no CP or RP between the paired symbols. The symbol pair structure is illustrated in Figure 4.4.

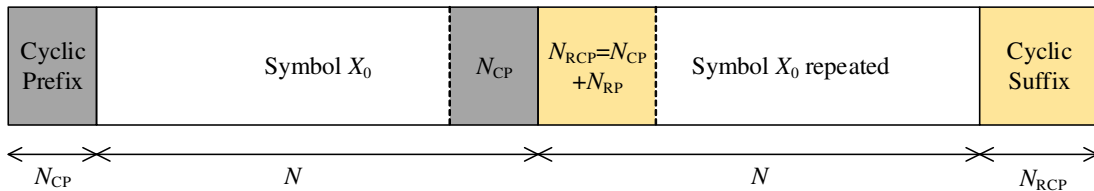


Figure 4.4: Symbol pair structure for fine ranging.

In the figure, N_{CP} and N_{RP} are the lengths of the cyclic prefix and roll-off period, respectively. Thus $N_{\text{RCP}} = N_{\text{CP}} + N_{\text{RP}}$ is the number of cyclic suffix samples. Similar to the operation of appending a roll-off period, a cyclic suffix is obtained by duplicating the N_{RCP} samples from the beginning of the symbol and appending them to

the end. A cyclic prefix of length N_{CP} is taken from the end of the first symbol and appended to the beginning of the symbol pair. The cyclic suffix is a unique feature of the ranging request signal. It guarantees that there is no transition between the two repeated symbols, yet making the two symbols of the same length as two ordinary OFDM symbols. After applying the windowing function which tapers the first and last N_{RP} samples of the symbol pair, the center $2(N + N_{CP})$ samples remain the same. Due to this structure, the safety window for the timing estimation error is now wider than in the ordinary OFDM structure which is $\{-(N_{CP} - N_{RP}), \dots, 0\}$ if the channel delay is zero. The new safety window is now $\{-(N_{CP} - N_{RP}), \dots, 0, \dots, N_{CP}\}$ for zero delay channels. It is assumed that the cyclic prefix length N_{CP} and the roll off period N_{RP} set by the CMTS are such that the channel micro-reflection does not exceed this safety range. The data in the ranging symbol can be obtained with no degradation as long as the timing error falls into this region.

As mentioned earlier, the preamble symbol in the ranging request signal consists of a sequence of BPSK modulated symbols. Each of the N_{fr} subcarriers that are allocated for fine ranging is assigned with a BPSK symbol. With the ranging frame structure in mind, the three methods discussed previously in Chapter 2 can be re-evaluated under the DOCSIS 3.1 setup. To obtain a preamble symbol with a time domain pattern matching that of the training training symbol in S&C method described in Chapter 2, the subcarriers with odd indexes have to transmit zeros which is not a BPSK symbol and therefore cannot be achieved. The S&S training symbol requires not only subcarriers that transmit zeros but also a manual sign inversion on one portion of the symbol. Hence the S&S method cannot be readily applied in this case either. Similarly to S&C, P&C method requires non-transmitting subcarriers at the odd indexes which is not possible in the preamble symbol of the ranging frame. Thus, none of the three methods described in Chapter 3 can be directly applied in DOCSIS 3.1 upstream timing synchronization process. However, the common idea of these three techniques, which is the exploitation of correlation generated by the repetitive structure of the training sequence, can be utilized. Section 4.2 presents a novel timing

estimation method which can be considered as a modified version of the S&C method.

4.1.2 Time Synchronization Procedure for CMTS in Fine Ranging Process

Once the CMTS received a ranging request signal, timing synchronization is the first procedure performed. A high-level block diagram for the timing synchronization procedure is shown in Figure 4.5. The upstream signal contains transmitted signals from multiple CMs that are connected to the network. It is necessary to isolate the fine ranging signal from other CM signals first with a band pass filter. As Figure 4.3 shows, fine ranging signal occupies a total of N_{fr} subcarriers with $N_{\text{gb}}/2$ subcarriers acting as a guardband on each side. Assuming a perfect FIR filter with passband width of N_{fr}/N and transition bandwidth of $N_{\text{gb}}/(2N)$ is applied to the received signal $y[n]$, the filter's output $y_{\text{fr}}[n]$ contains only the contribution from the fine ranging request signal. The output of the filter is passed into the time offset estimation block which returns a timing error estimation. This timing error is fed into a signal buffer which stores the original incoming signal to choose the correct data samples for further processing. This chapter mainly focuses on the algorithms used in the time offset estimation block.

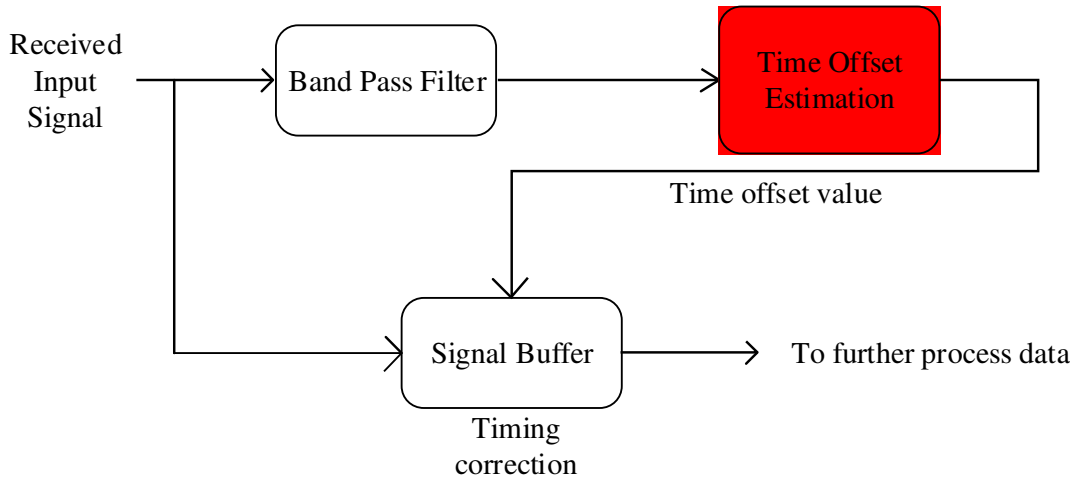


Figure 4.5: Overall block diagram for time synchronization in the ranging process.

4.2 Timing Synchronization Algorithm I

Since all symbols are transmitted as symbol pairs in the ranging request signal, the entire ranging frame has the repetitive pattern similar to that of the S&C method discussed in Section 3.2.2. Regardless of the actual data transmitted in each symbol, the ranging request frame always exhibits the structure of $[A, A, B, B, \dots, X, X]$.

4.2.1 Algorithm

Similar to the S&C method, a correlation calculation is performed over the time domain samples in two consecutive windows of length N where N is the symbol length or FFT size as shown in Figure 4.6. The two windows take samples from

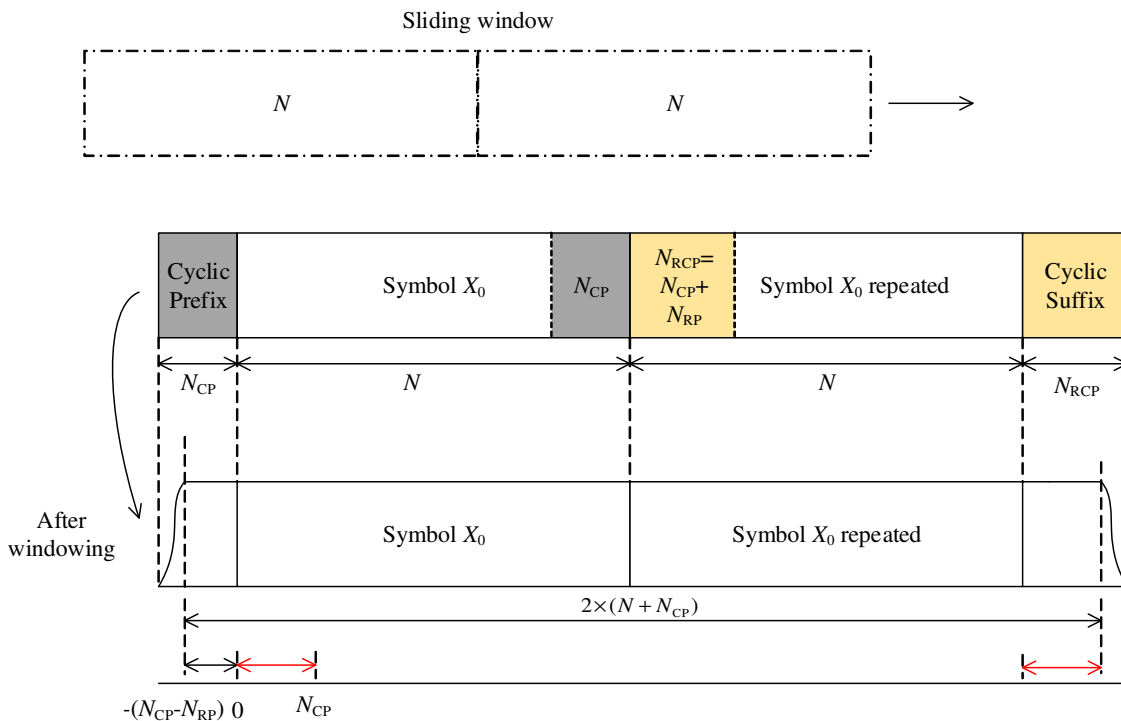


Figure 4.6: Correlation over the sliding windows for Algorithm I.

the incoming sequence and calculate the timing metric. The timing metric $\Gamma_I(\theta)$ is calculated as a ratio of two values as shown in Equation (4.1). The numerator is the sum of the products of the conjugates of samples from the first window and the corresponding samples from the second window. The sum of the squares of all

elements in each window is calculated and the greater sum is used as the denominator for normalization. This is to make sure that the empty symbol in the frame will not be used in the denominator for normalization.

$$\Gamma_I(\theta) = \frac{\sum_{m=\theta}^{\theta+N-1} y_{\text{fr}}[m+N] y_{\text{fr}}^*[m]}{\max \left\{ \sum_{m=\theta}^{\theta+N-1} |y_{\text{fr}}[m+N]|^2, \sum_{m=\theta}^{\theta+N-1} |y_{\text{fr}}[m]|^2 \right\}} \quad (4.1)$$

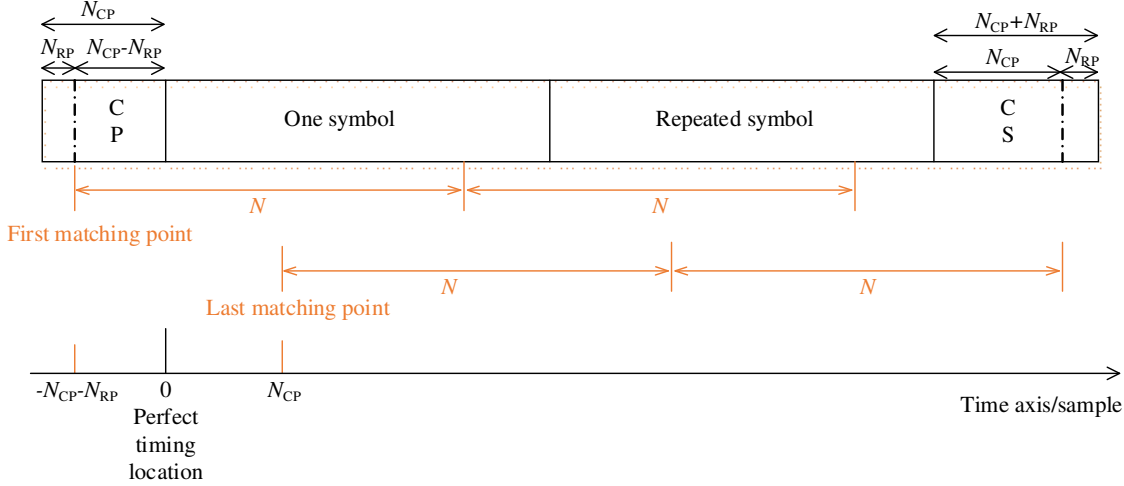


Figure 4.7: Matching points for the first method.

As the two windows are slid across the received signal, this timing metric $\Gamma_I(\theta)$ reaches its maximum value when the data inside the two windows are identical. Due to the addition of cyclic prefix, cyclic suffix and windowing, the two sliding windows match over a specific duration as shown in Figure 4.7. The starting point of the correlation match is at index $-(N_{\text{CP}} - N_{\text{RP}})$, and the last matching point is at index N_{CP} . Therefore, there is a peak plateau span of length $2N_{\text{CP}} - N_{\text{RP}}$ for every symbol pair in the RNG_REQ signal. For a perfect channel with only timing offset present, this plateau span should be 1.

An example of a plateau peak is shown in Figure 4.8. The channel used for the simulation is a perfect channel with just timing and frequency offsets. In the specific simulation, the value of N_{CP} is 96 and N_{RP} is 64. Therefore, theoretically the matching window starts from $-(N_{\text{CP}} - N_{\text{RP}}) = -32$ to $N_{\text{CP}} = 96$. The vertical solid lines indicate the theoretical positions of the plateau span for the timing metric. To

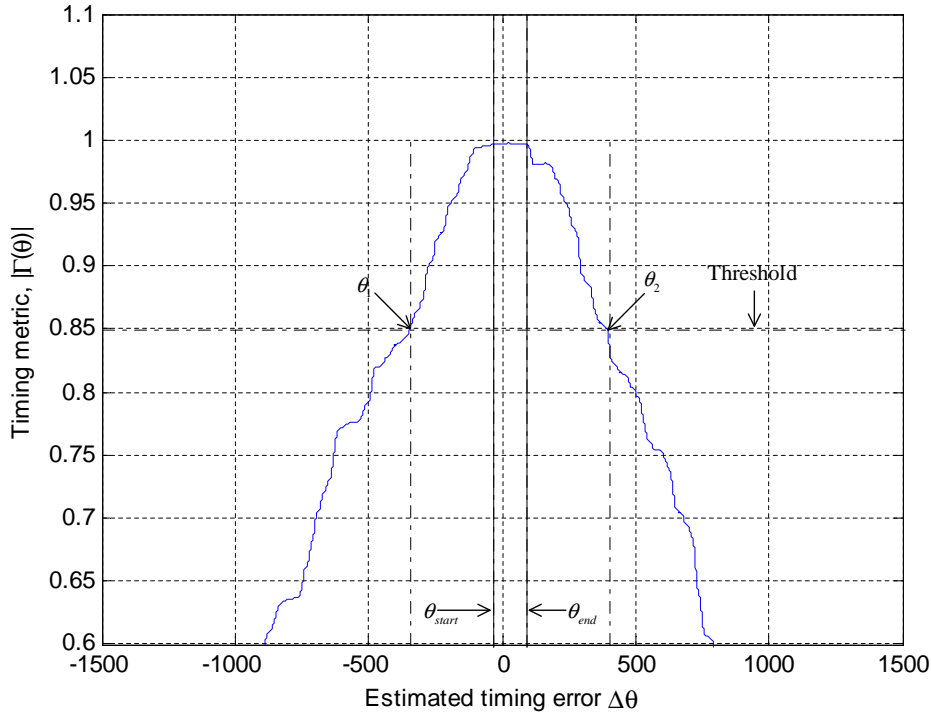


Figure 4.8: Timing metric plateau window for method I.

obtain the timing offset from the timing metric, one needs to identify the locations of the plateau span boundaries. The most effective way to locate the plateau is by looking for the triggering points for the rising and falling edges of the peak. To do so, a threshold value for the timing metric is chosen. Every time the timing metric crosses that threshold value, it marked either the rising or the falling triggering point of the peak plateau. In Figure 4.8, the threshold is set to be 85% of the maximum peak value, i.e., it is 0.85. The dashed horizontal threshold line intersects the timing metric at θ_1 and θ_2 as indicated.

Recall that the correlation of two uncorrelated random variables is zero. Therefore when each of the two sliding windows contain only samples from two different symbols respectively, every sample pair from the two windows is uncorrelated. As a consequence, the correlation sum for the sample pairs reaches 0 and so does the timing metric. As the two windows keep sliding and more samples of the same symbol pair start to get taken by the windows, the timing metric value increases until it reaches 1

when the samples inside the two windows are identical. This rising and falling trend is approximately linear with respect to the number of identical symbols in the windows. In the simulation shown in Figure 4.8, the FFT size is 2048. By choosing the threshold value to be 85% of the peak value, the timing metric theoretically would reach the threshold value when 85% of the symbol pairs were taken into the sliding window. Thus, the two triggering locations would be $(1 - 0.85) \times N = 307$ samples away from the closer end of the peak span as shown as the vertical dashed lines.

To be more specific, let the two triggering indexes be θ_1 and θ_2 and the start and end points of the peak plateau be θ_{start} and θ_{end} . Then the relationship between them can be represented as:

$$\theta_1 = \theta_{\text{start}} - (1 - 0.85) \times N \quad (4.2a)$$

$$\theta_2 = \theta_{\text{end}} + (1 - 0.85) \times N \quad (4.2b)$$

$$\theta_{\text{end}} - \theta_{\text{start}} = 2N_{\text{CP}} - N_{\text{RP}} \quad (4.2c)$$

These theoretical indexes match well with the actual intersection indexes as shown in the plot. The actual intersections θ_1 and θ_2 are 3 and 1 sample(s) away from the calculated theoretical values. Based on the relationship between θ_{start} , θ_{end} and θ_1 , θ_2 shown in Equation (4.2), the estimated start and end points θ_{start} , θ_{end} for the first peak can be represented as:

$$\theta_{\text{start}} = \theta_1 + (\theta_2 - \theta_1)/2 - N_{\text{CP}} + N_{\text{RP}}/2 \quad (4.3a)$$

$$\theta_{\text{end}} = \theta_1 + (\theta_2 - \theta_1)/2 + N_{\text{CP}} - N_{\text{RP}}/2 \quad (4.3b)$$

Since the ranging frame exhibits a pattern of $[A, A, B, B, \dots, X, X]$, there is more than one peak present in the timing metric. In Figure 4.9, there are a total of 4 peaks which are due to the use of 10 ranging symbols in the simulation. The channel used for simulation is a multipath channel with noise, frequency and time offsets. The dashed vertical lines show the theoretical plateau locations for each peak.

The peak plateau locating procedure is repeated for all peaks that are captured by the triggering condition to obtain the plateau starting and ending indexes for each

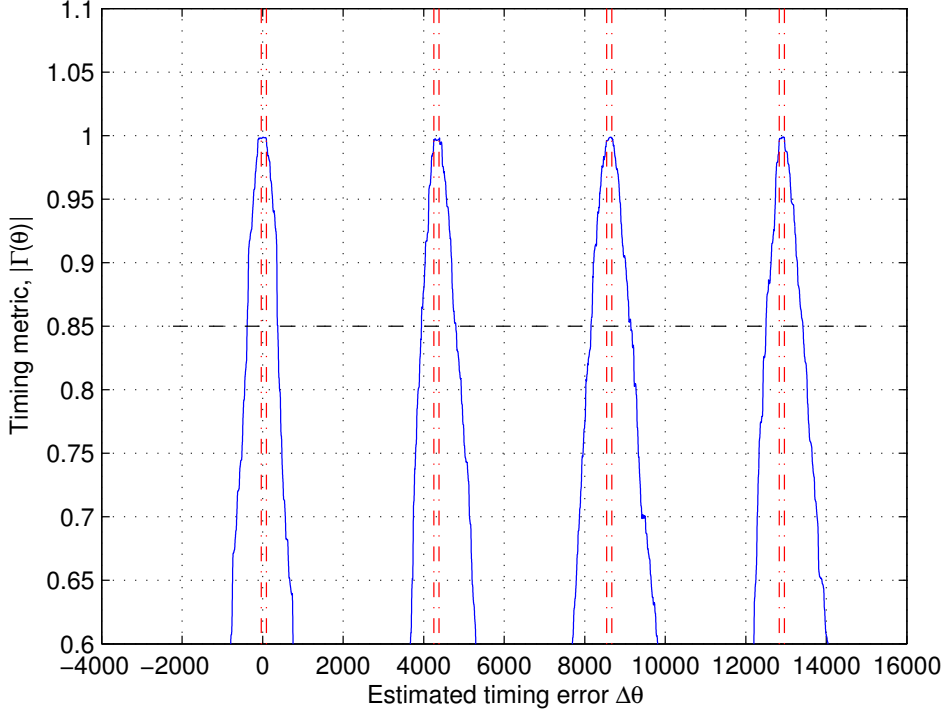


Figure 4.9: Multiple peaks in Algorithm I.

peak. Once the start index of the n th window $\theta_{\text{start}}^{(n)}$ is known, each peak plateau location can determine a unique time offset value. Represent the estimated time offset obtained based on the n th peak as $\hat{\theta}_n$ and it can be obtained as:

$$\hat{\theta}_n = \theta_{\text{start}}^{(n)} - N - N_{\text{CP}} - N_{\text{RP}} - (n - 1)(N + N_{\text{CP}}) \quad (4.4)$$

However, it should be pointed out that, due to the channel micro-reflections and noise, the symmetry between the triggering points and the peak plateau can be destroyed as indicated in the example in Figure 4.10. The theoretical peak plateau locations are indicated as the vertical dashed lines. The estimated locations using Equation (4.3) are illustrated as in the vertical solid lines. The threshold value used here is equal to 85% of the timing metric peak value. Since the symmetry of the timing metric is destroyed, this estimation results in a large variation on the result. In particular, the estimation error in this case is 62. This error requires a big safety window and therefore a large overhead, which reduces the transmission efficiency.

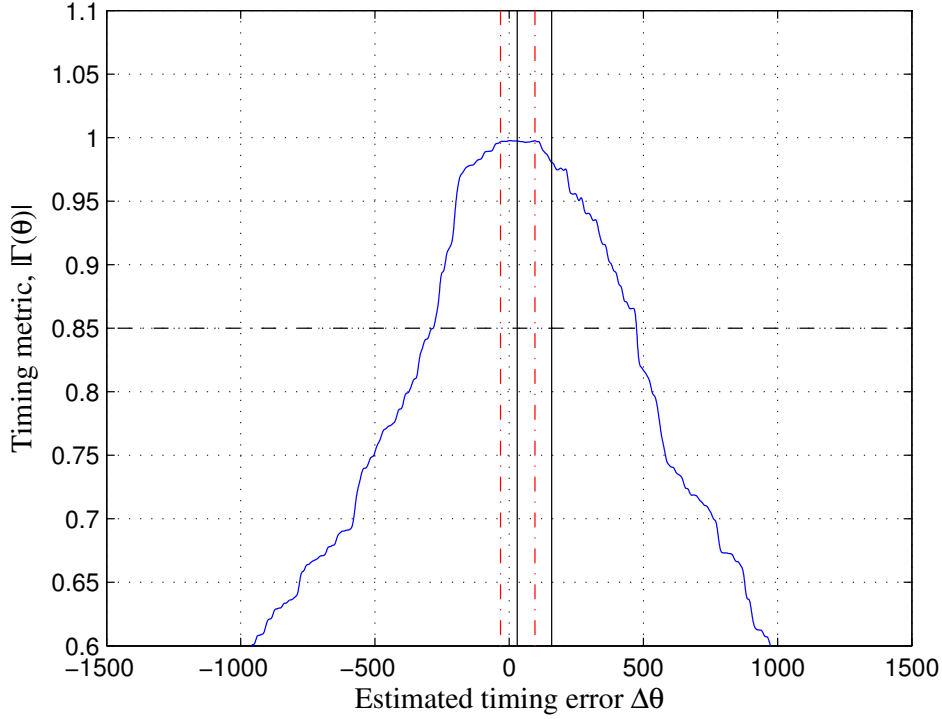


Figure 4.10: Example of an asymmetrical peak for a timing metric.

A more accurate plateau location estimation approach is now proposed. It was pointed out before that the plateau has a known width of $2N_{\text{CP}} - N_{\text{RP}}$ and it is bordered by two triggering points. Furthermore, the timing metric values between the triggering points are supposed to remain constant only for the duration of the plateau span between the triggering points. Based on these observations, in this new approach, a span of length $2N_{\text{CP}} - N_{\text{RP}}$ is slid across the timing metric between triggering points θ_1 and θ_2 . The new plateau location is obtained by checking for the index where there is the least amount of variation in the sliding span. Specifically, one can calculate and record the variation inside each span. The recorded variation array is calculated as $\Gamma_{\text{variation}}[\theta'] = \max(\Gamma_{\text{I}}[\theta' : (\theta' + 2N_{\text{CP}} - N_{\text{RP}} - 1)]) - \min(\Gamma_{\text{I}}[\theta' : (\theta' + 2N_{\text{CP}} - N_{\text{RP}} - 1)])$, where $\theta_1 \leq \theta' \leq \theta_2 - (2N_{\text{CP}} - N_{\text{RP}} - 1)$. Locate index θ'_{min} that returns the minimum value of the variation array and then θ'_{min} is the estimated plateau starting point for the given peak. Thus the estimated timing offset can be obtained as shown in Equation (4.4). When applying this approach to

the same example in Figure 4.10, the new identified span location returns a timing estimation error of -2 samples which is much smaller than the error incurred by using the previous approach.

Summary of Algorithm I

Input: Filtered received signal $y_{fr}[n]$

Output: Estimated timing offset value $\hat{\theta}$

1. Calculate the timing metric Γ_I based on Equation (4.1);
2. Identify all matching peak plateaus and locate the triggering points indexes for all peaks by setting a threshold of 85% of the peak timing metric value;
3. Calculate the timing metric variation array $\Gamma_{\text{variation}}[\theta']$ between the two triggering points of each peak;
4. The index of the variation array minimum value is the start point $\theta_{\text{start}}^{(n)}$ of the plateau for the n th peak and the estimated timing offset value is obtained by Equation (4.4).

4.2.2 Computational Complexity

Figure 4.11 shows the hardware block diagram for the timing estimation method described in this section. Illustrated is the case when the two duplicated symbols fall into the windows of operation, i.e., perfect timing location. The newest data that came in is A , which is multiplied by the conjugate of the data sample that is N samples away. There are two shift registers used in this algorithm. One register of size $2N$ is used to store the squared magnitude values of the data samples. The other one of size N is used to store the products of the two corresponding elements in the two windows. Note that in this method, only two multipliers are required as shown

in the diagram. One is for the product of the two different elements and one is for squaring.¹

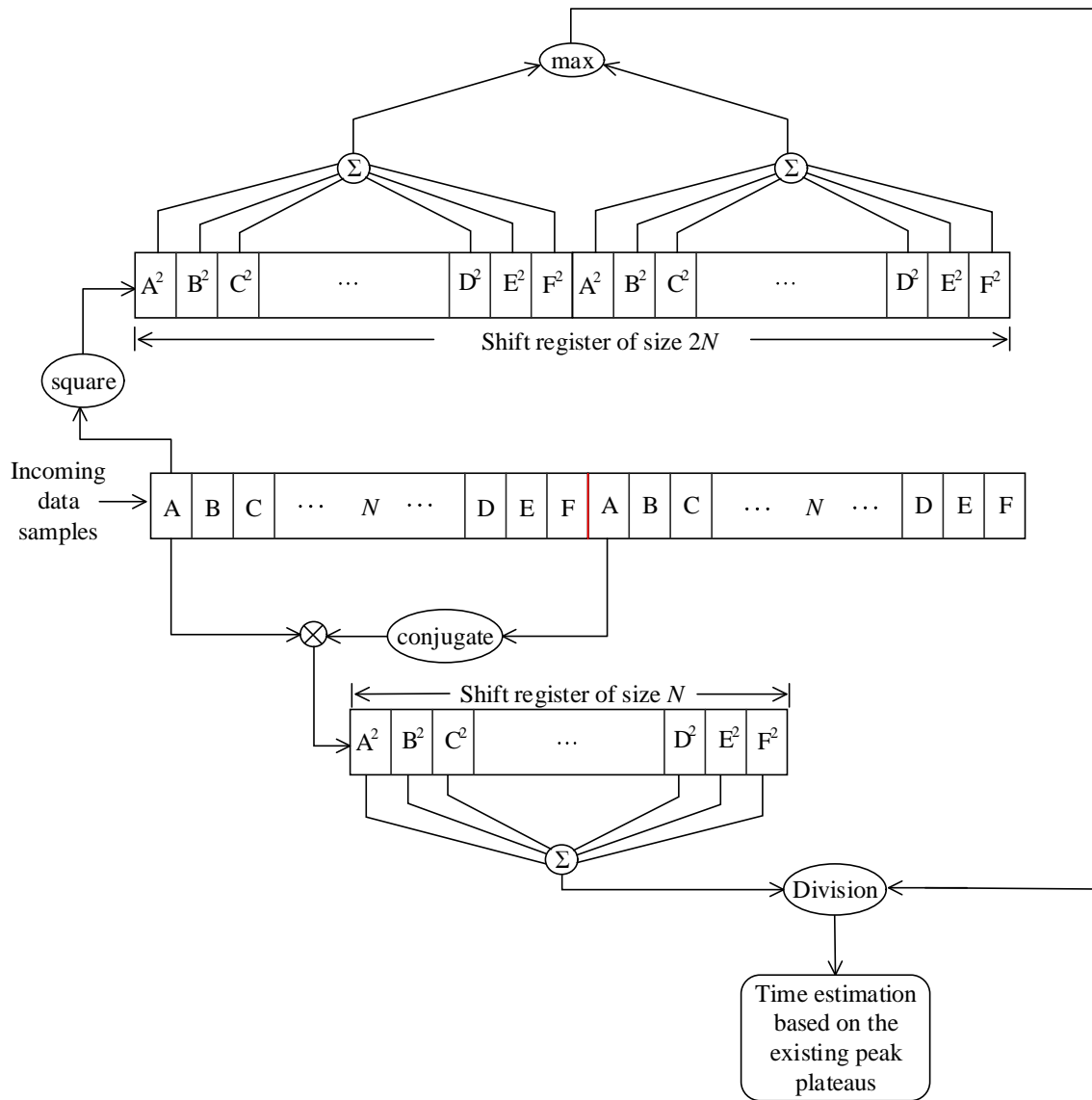


Figure 4.11: Block diagram for the first timing synchronization method.

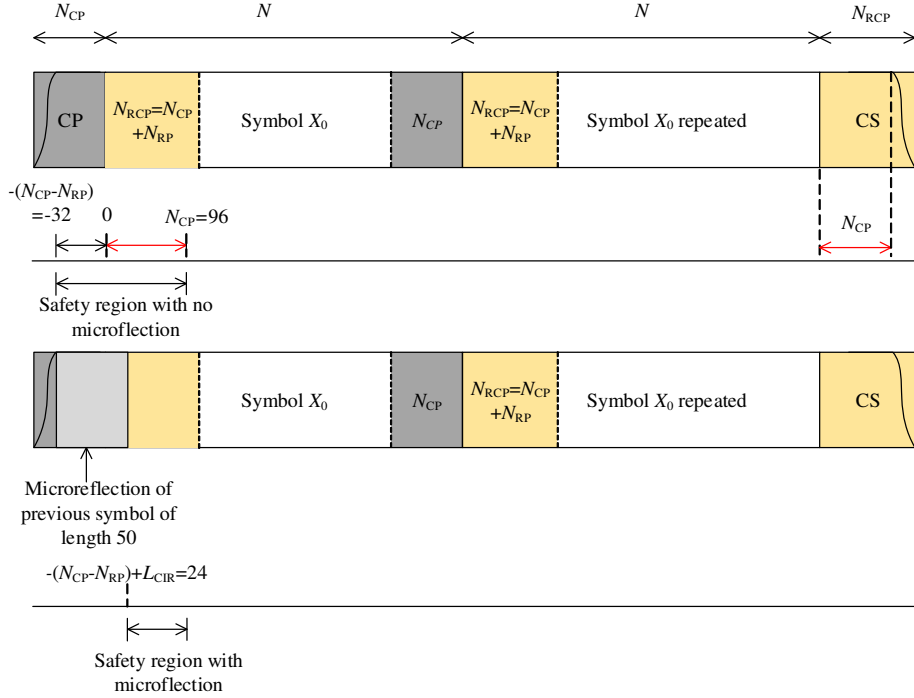


Figure 4.12: Illustration of the safety window for the simulation scenario.

4.2.3 Performance

The timing offset estimation errors are recorded over 1000 simulation runs. For each run, there are CMs that are either in ranging, normal data traffic mode or not transmitting at all. The number of ranging CM is fixed at 1 and the total number of CMs connected to the CMTS is 10. The ranging CM has a fixed guardband of 4 minislots, $N_{gb} = 4Q$ and a ranging bandwidth of $N_{fr} = N/16$. The timing offset is simulated as a random integer between N_{RP} and N , whereas the frequency offset $\Delta f/N$ is a random number between -0.3 and 0.3. The total number of normal traffic transmission bursts is 50 and each transmission burst is assigned to a non-ranging CM in a random fashion. Each burst occupies up to 5 minislots. The number of symbols in a frame, K , is fixed at 10 for all cases so that the study can focus on the timing metric with 4 peaks. In the simulation, the SNR is set to be 35dB, which is the minimum SNR

¹There is a limited number of multipliers in an FPGA, making multipliers a valuable resource. Thus, the number of multipliers required is one of the key metrics which determines the cost of an FPGA design.

value specified for a DOCSIS 3.1 cable system. The CP length is set to be $N_{\text{CP}} = 96$, which is the smallest CP value specified in DOCSIS 3.1. The redundant region of the frame is $\{-(N_{\text{CP}} - N_{\text{RP}}), \dots, 0, \dots, N_{\text{CP}}\}$ as illustrated in the upper portion in Figure 4.12. To test the case with the smallest safety window, the roll-off period value N_{RP} is set to be as close to N_{CP} as possible, which is $N_{\text{RP}} = 64$. A multipath channel is simulated where the microreflection of the previous symbol has a delay of 50 samples and a gain of 0.1, which is the maximum gain for the microreflection specified in the standard. Therefore the first 50 samples of the redundant region are also contaminated by the echo of the previous symbol as illustrated in the lower portion of Figure 4.12. This leaves a safety window of length 73 over the range $\{24, \dots, 96\}$. Instead of locating the exact beginning sample of the frame, the aim is to locate the middle of this safety region since the estimation error can be either positive or negative. In this way the safety window for the timing estimation error for the scenario tested in the simulation is $[-36, 36]$.

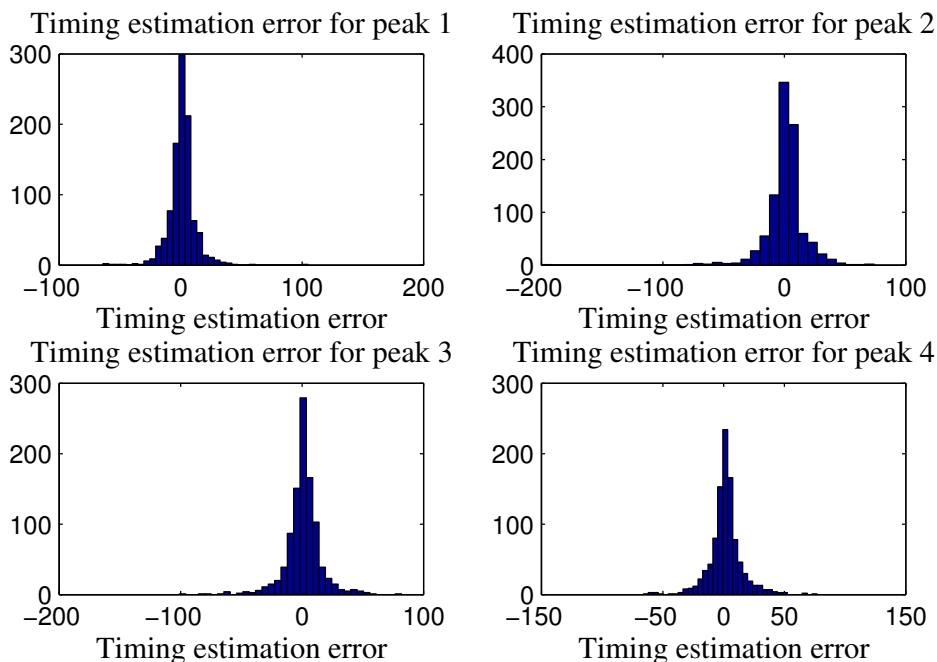


Figure 4.13: Timing estimation errors of the first four peaks.

Figure 4.13 shows the histograms of the timing offset estimation error using each

of the four peaks. The figure shows that the timing estimation errors for all four peaks have similar distributions. Over 95% of the cases, the peaks provide a timing estimation error within the safety region. A majority (70%) of the errors are within 10 samples. The number of times that the timing estimation errors do not fall into the safety region for each peak is listed in Table 4.1. The first peak has the minimum number of estimation failures while peaks 2, 3 and 4 have similar performance. The reason for this is that the first preamble pair is BPSK-modulated, while the remaining symbols in the preamble are QPSK modulated.

	Peak 1	Peak 2	Peak 3	Peak 4
Number of errors outside safety region	13	33	36	32

Table 4.1: Number of errors outside the safety region for each peak.

A simple step to improve the accuracy is to average the timing offsets obtained from all four peaks. The histogram plot of such an averaged error using the four peaks is shown in Figure 4.14. In this way, the number of times that the algorithm does not meet the accuracy requirement reduces to only 2. Table 4.2 compares the means and variance values for the timing estimation error obtained using different peaks and the averaged results. As the table shows, all five estimation errors have a small mean value of approximately 1. However, the variance is much smaller by using the averaged time offset error. It is about 1/3 to 1/5 of the variance when only one signal peak is used.

The above simulation was completed with the transmission bursts randomly allo-

	Peak 1	Peak 2	Peak 3	Peak 4	Averaged value
Mean	1.067	1.035	0.983	1.709	1.1985
Variance	122.775	255.333	222.255	189.8121	52.3681

Table 4.2: Means and variances of the timing estimation errors obtained using different peaks and their average.

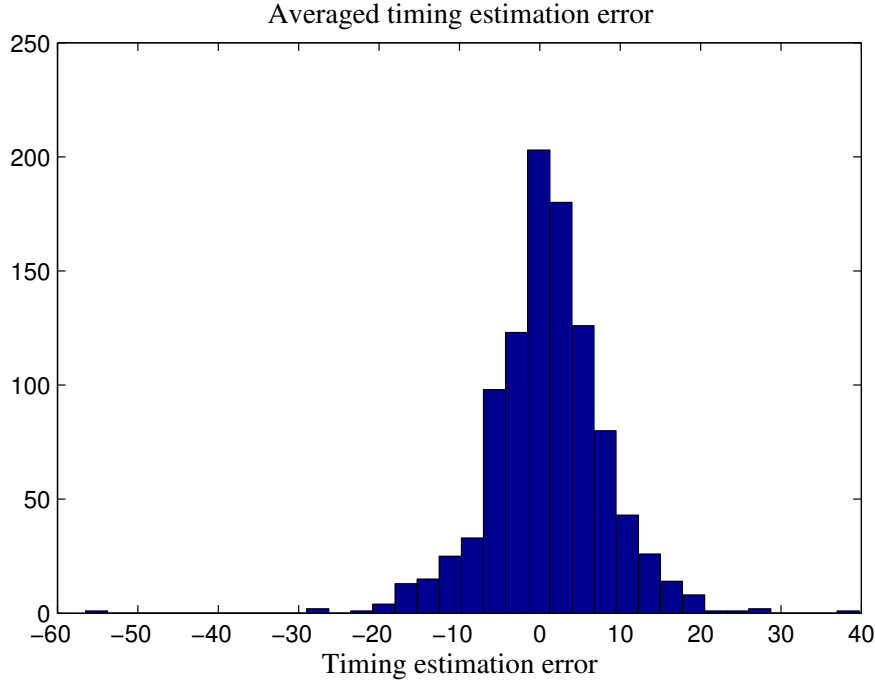


Figure 4.14: Timing error by averaging the timing offset values obtained from the four peaks.

cated for the transmitting CMs. To test the algorithm in the most critical setup, a new simulation was performed. Here, the system is configured to have two CMs transmitting over subcarriers that are next to both sides of the ranging CM's guardband. As the upstream signal has a power attenuation ranging from -9dB to +3dB, to create the strongest interference to the ranging signal as possible, the two CMs transmitting next to the ranging CM are configured to have a power attenuation of +3dB, while the ranging CM has a power attenuation of -9dB. This way, the difference between power of the ranging spectrum and power of the adjacent data-transmitting spectrum is maximized at 12 dB. Such an example of the spectrum is illustrated in Figure 4.15 where the upper panel of the figure shows the spectrum of the original received signal and the lower panel shows the spectrum of the signal after filtering. In the figure the fine ranging spectrum (FR) is surrounded by two data transmitting CM spectrum (DT) at each side. On the other hand, the frequency offset value $\Delta f/N$ is fixed at either -0.3 or 0.3, which is the maximum possible value as in the previous setup.

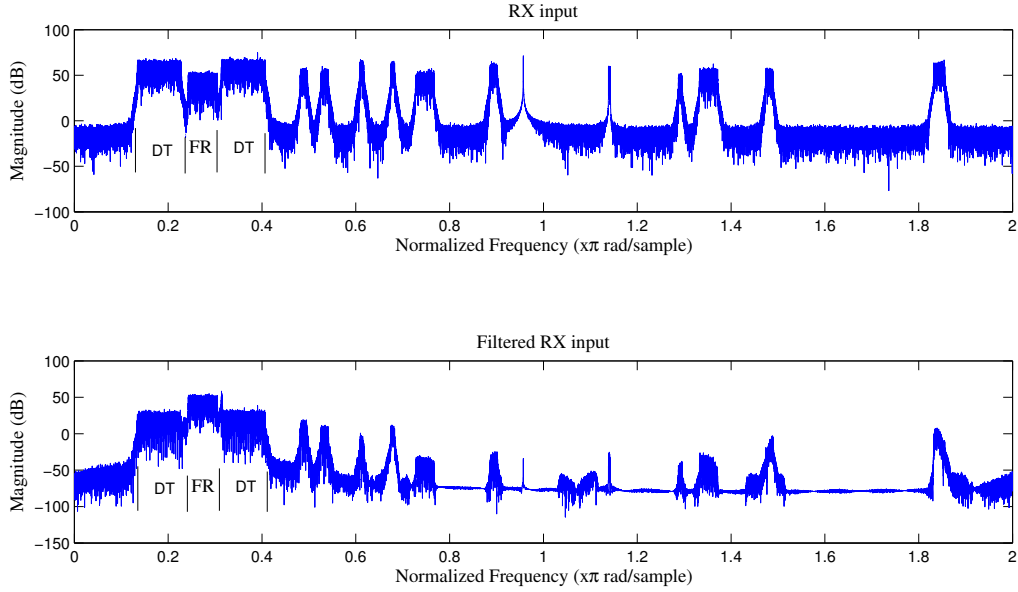


Figure 4.15: Spectrum of the original and filtered signals in the worst case scenario.

By simulating the above upstream signal spectrum and leaving everything else the same as in the previous setup, new simulation results are recorded and plotted in Figure 4.16. Using the averaged timing estimation value in this simulation, 370 of the 1000 simulation runs fail to provide a timing estimation value within the safety window. The average value of the timing estimation error increases to -5.66 and the variance is now 619. Thus Algorithm I does not provide satisfactory performance and cannot be used under the above severe channel condition.

To better compare the difference between the two simulation setups, Table 4.3 summarizes a few key system parameters. Specifically, the second column shows the parameters used for a practical channel condition and the third column shows the most severe channel condition.

4.3 Timing Synchronization Algorithm II

Algorithm I yields results with a reasonable accuracy at a typical channel scenario. Due to the existence of the plateau, the variance of the estimation error is quite

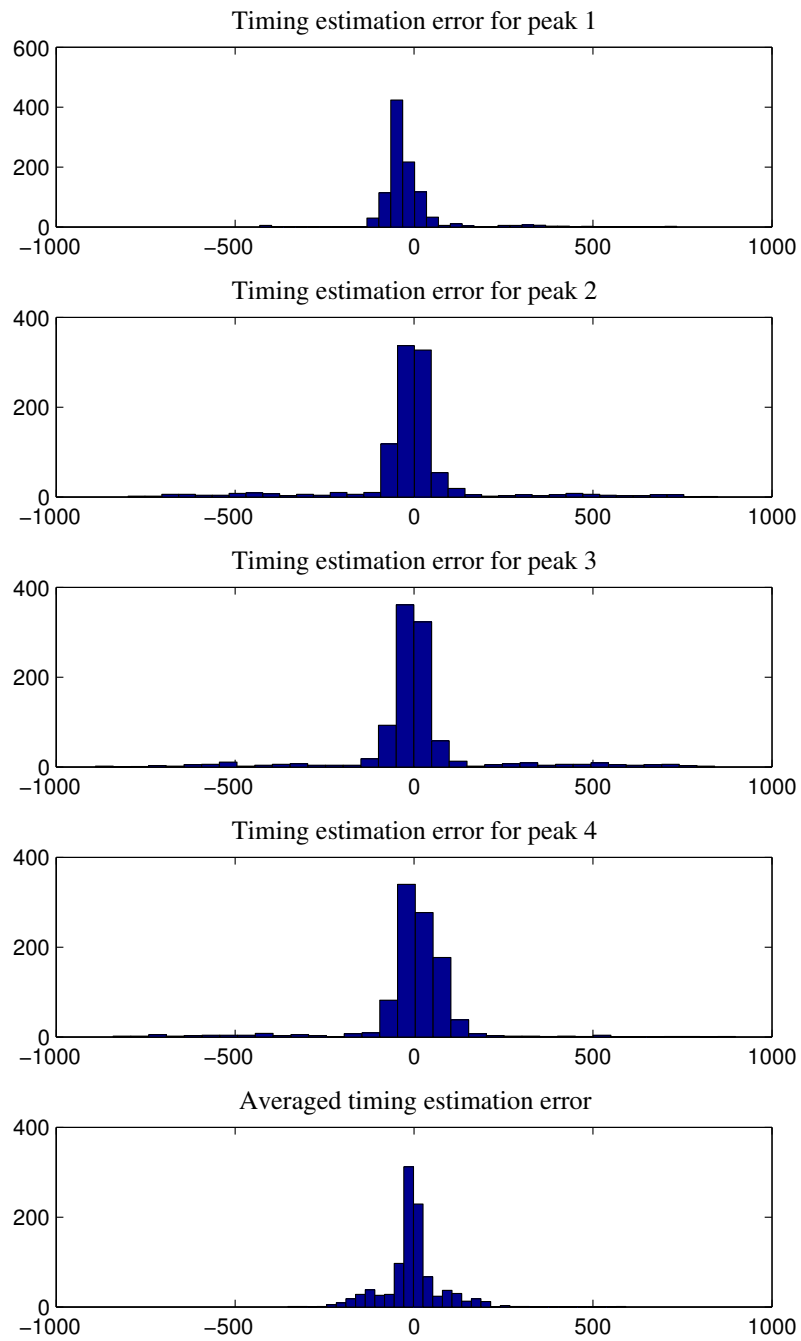


Figure 4.16: Histogram plots of timing estimation errors for Algorithm I under the severe channel condition.

	Practical Channel	Severe Channel
N_{CP}	96	96
N_{RP}	64	64
Number of data transmitting CMs	10	10
Spectrum location of the data transmitting CMs	randomized	2 out of 10 CMs are right next to the fine ranging spectrum
SNR	35	35
Time offset	uniform over $[N_{\text{RP}}, N_{\text{FFT}}]$	uniform over $[N_{\text{RP}}, N_{\text{FFT}}]$
Frequency offset	uniform over $[-0.3, 0.3]$	either -0.3 or 0.3
Guardband	4 minislots	4 minislots

Table 4.3: Key system parameters used in Algorithm I simulations.

large, at around 50 samples. However, under the worst channel condition and traffic, the performance Algorithm I becomes unacceptable. To further improve the timing estimation performance in such a case, a second algorithm is proposed in this section.

4.3.1 Algorithm

This second algorithm exploits the fact that the preamble symbols for the fine ranging process are BPSK modulated. For a BPSK-modulated signal, the time domain samples are mirrored and conjugate symmetrical as shown in Figure 4.17. One BPSK preamble symbol is of size N . The first half of the preamble symbol, $[A, B, C, \dots, D, E, F]$, is mirrored symmetrical with the conjugate of the second half, $[F^*, E^*, D^*, \dots, C^*, B^*, A^*]$.

Similar to the first algorithm, the timing metric is calculated first. A sliding-window correlator of lag $N/2$ is applied to the time domain samples $y_{\text{fr}}[n]$ where N is the symbol length. The two windows are now of length $N/2$. Unlike the first method, the second window is flipped with respect to the center to obtain a mirrored

image before performing the correlation calculation. Since the windows are conjugate symmetrical, conjugate function will not be needed for the product in the numerator as shown in Equation (4.5)

$$\Gamma_{II}(\theta) = \frac{\sum_{m=0}^{N/2-1} y_{fr}[\theta + N/2 - m] y_{fr}[\theta + N/2 + m]}{\sum_{m=\theta}^{\theta+N/2-1} |y_{fr}[m + N/2]|^2} \quad (4.5)$$

The timing metric obtained by Equation (4.5) results in three peaks at three different matching points, spaced $N/2$ apart. Figure 4.17 shows the BPSK preamble symbol pair structure indicating locations of the matching points. Note that the BPSK modulated preamble symbol is repeated with no transition between the two paired symbols. Therefore the three matching points are spaced by $N/2$ apart while the first one occurs at the beginning of the first preamble symbol.

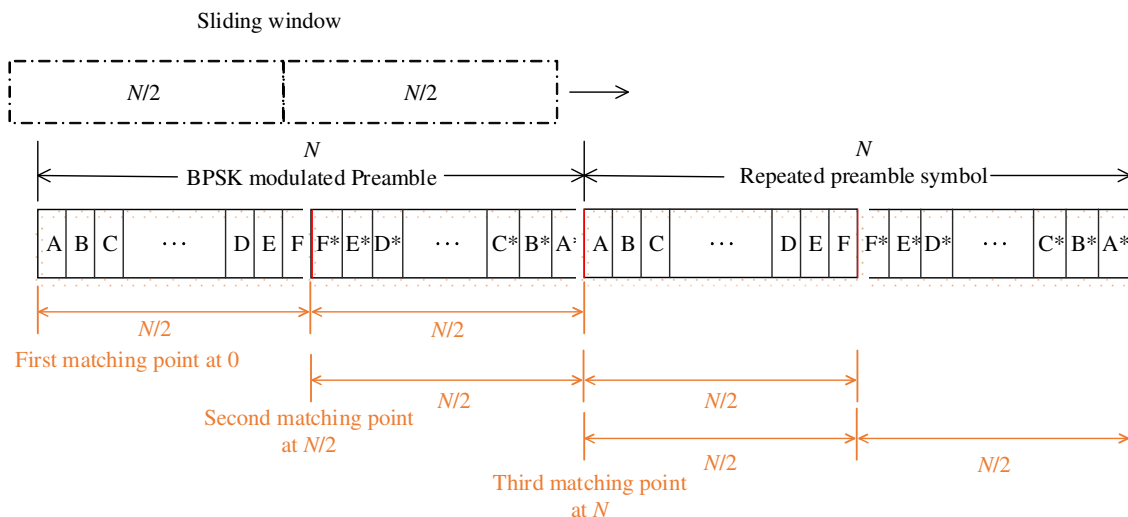


Figure 4.17: Matching points for the second method.

Unlike Algorithm I where there is a plateau region for the correlation match, the second algorithm provides a much sharper peak as shown in Figure 4.18. In the simulation, the system is in 2k mode, $N_{CP} = 96$ and $N_{RP} = 64$ which is the same as the simulation setup used for Algorithm I. The channel simulated is a multipath channel with an echo of 50 samples delay and gain of 0.1. The top part of the figure is a zoomed-in view of the first peak in the timing metric obtained using Algorithm I.

The bottom part of the figure is a zoomed in view of the same part of the timing metric obtained using Algorithm II. There are only three matching points in Algorithm II, while the values of the timing metric at all other positions are almost zero. Since the difference between the peak value and the other timing metric values around the peak is clearly enhanced, the algorithm is less likely to miss the peak.

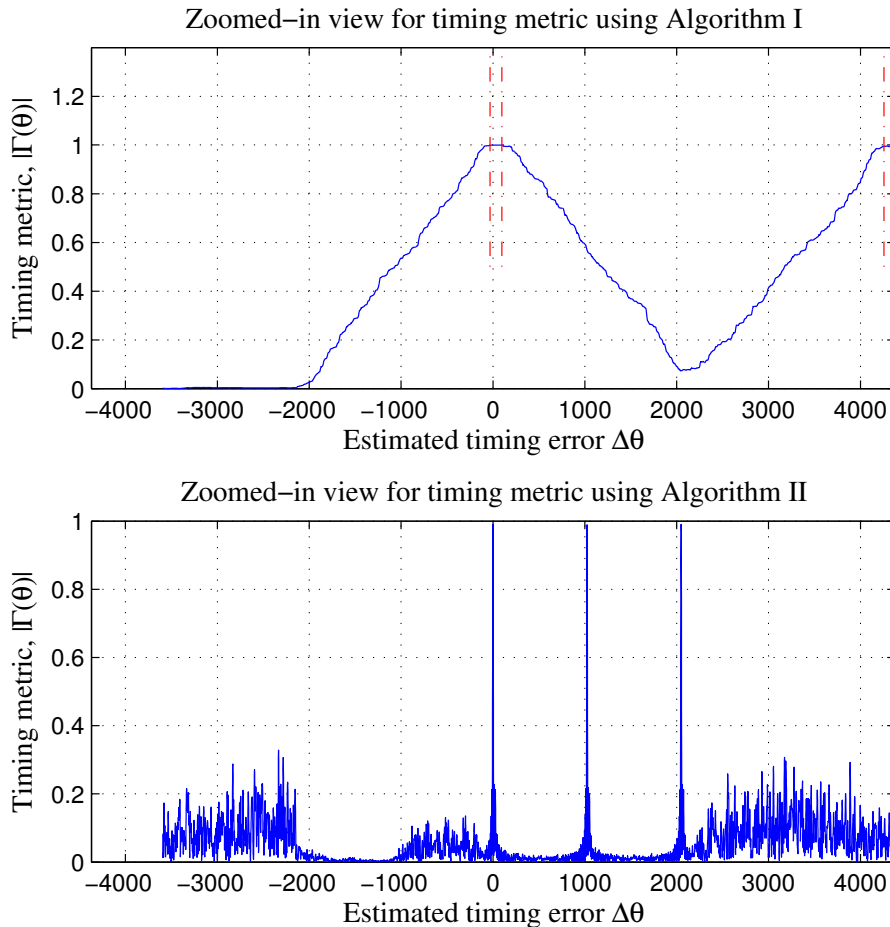


Figure 4.18: Comparison of the timing metric for Algorithm I and Algorithm II.

It can be seen from Figure 4.17 that the three matching peaks happen at 0, $N/2$ and N , respectively. To identify and obtain the indexes of the three peaks, a matching filter is applied to the timing metric. The matching filter stores three timing metric values that are spaced $N/2$ apart. The three elements are summed and returned as

the output of the matching filter as shown in Figure 4.19 and Equation (4.6).

$$\Gamma_{\text{II, MF}}(\theta) = \Gamma_{\text{II}}(\theta) + \Gamma_{\text{II}}(\theta - N/2) + \Gamma_{\text{II}}(\theta - N) \quad (4.6)$$

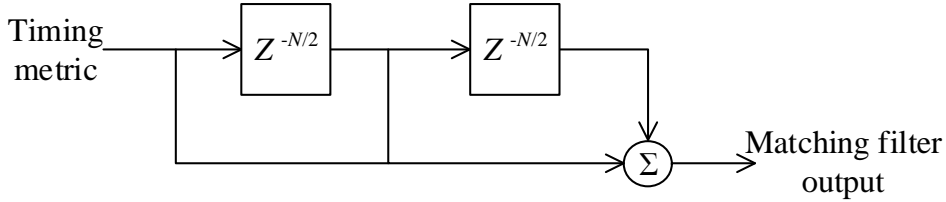


Figure 4.19: Block diagram for the matching filter.

Using the matching filter, only when all the three peaks are caught by the filter will the output be of the maximum value. This maximum value is equal to the sum of all three peaks in the timing metric $\Gamma_{\text{II}}(\theta)$. An example of the matching filter output is shown in the bottom part of Figure 4.20. Note that the x-axis of the two plots in the figure is the timing estimation error, not the time domain index. They are plotted and shifted in a way that the two perfect timing locations are aligned, and there is a lag of N in the matching filter output compared to the original timing metric. Instead of three peaks of equal value that are present in the timing metric as shown in the top figure, the matching filter output returns five peaks of different values with the maximum peak in the middle. Since the matching filter contains 2 delays of $N/2$ each, the maximum peak value happens where the last matching point occurs at index N . The other 4 smaller peaks occur when only one or two of the timing metric matches are caught by the matching filter. All five peaks are spaced $N/2$ apart. Therefore the perfect timing location index $\hat{\theta}$ is equal to the maximum value index of the matching filter output θ_{max} minus the FFT size N :

$$\hat{\theta} = \theta_{\text{max}} - N. \quad (4.7)$$

Figure 4.21 shows the block diagram of the second algorithm. Due to the mirrored symmetry of the conjugate samples, the cross products in the numerator of the

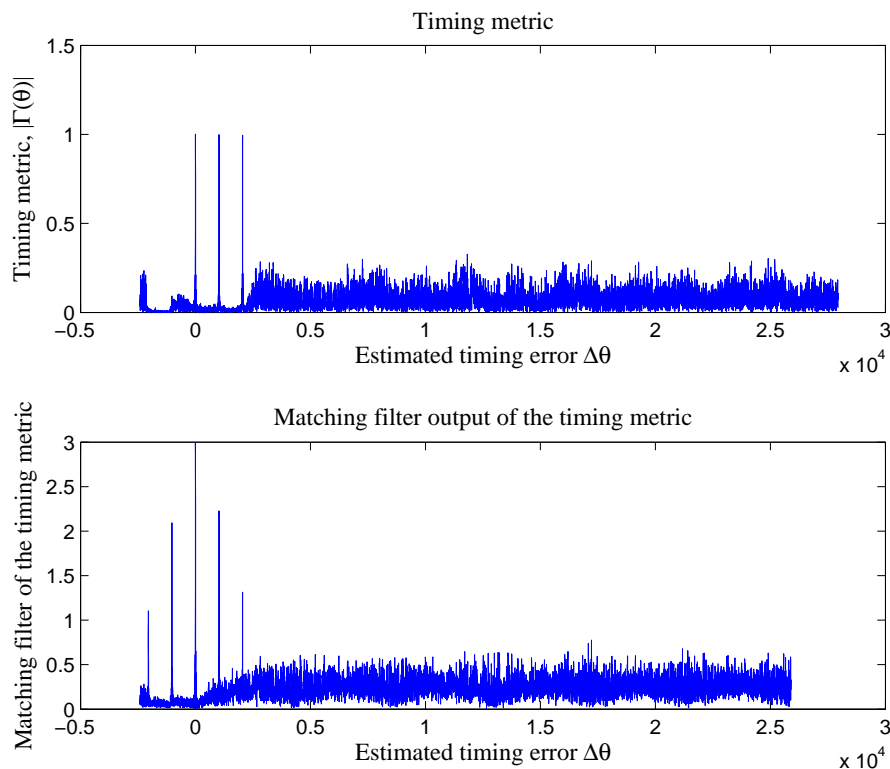


Figure 4.20: Applying matching filter to the timing metric in Algorithm II.

correlation metric equation cannot be stored and updated using a shift register as in Algorithm I. For each new data sample received, all $N/2$ cross products need to be recalculated. This means that $N/2$ multiplication operations need to be performed and $N/2$ multipliers are needed. Compared to just 2 multipliers in the first algorithm, the complexity of the second algorithm is significantly higher.

4.3.2 Complexity Reduction by Skipping Data Samples

To reduce the hardware cost, an approach is proposed to reduce the number of multipliers required by taking fewer samples for the cross products. Instead of taking the product of every sample pair as shown in the block diagram in Figure 4.21, only a fraction of the data samples is used for the cross products. The number of products required is determined by observing the pattern of the timing metric and matching filter output of the timing metric. The more samples that are discarded,

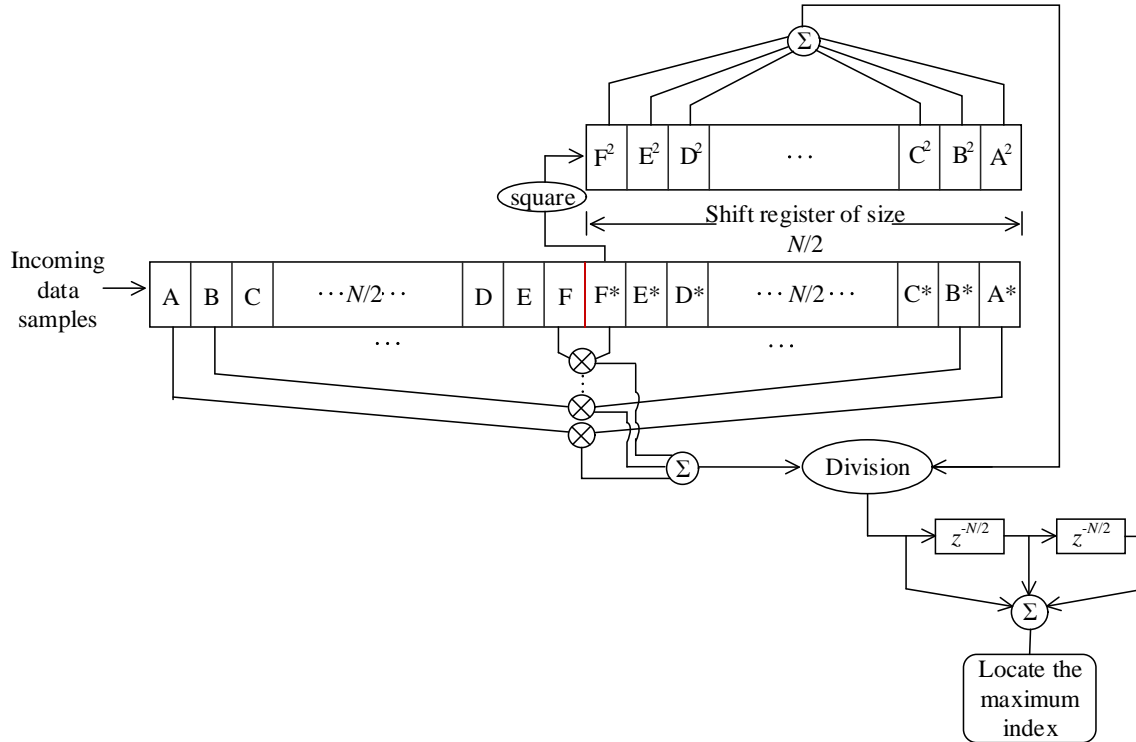


Figure 4.21: Block diagram for Algorithm II.

the more distorted the timing metric pattern would be. The sample sets used for calculation are obtained by skipping a certain number of samples on a regular basis. Figure 4.22 shows the timing metric and matching filter output for three examples of different number of sample sets. As the middle figure shows, the matching filter is able to output a well-defined peak at the correct timing index when 63 samples are discarded between every two retained samples. Once this value is increased to 127, that is taking every other 128th sample as the input sample, the matching filter output pattern is severely distorted and the difference between the correct peak and the other values around the peak is greatly reduced. Although the desired peak at the correct timing index remains the maximum value in this simulation setup, the task of identifying the correct peak is much more challenging. To be safe with the selection of the samples, it is recommended that a maximum number of 63 samples could be discarded between every two retained samples. This reduces the number of multipliers required to $N/128$.

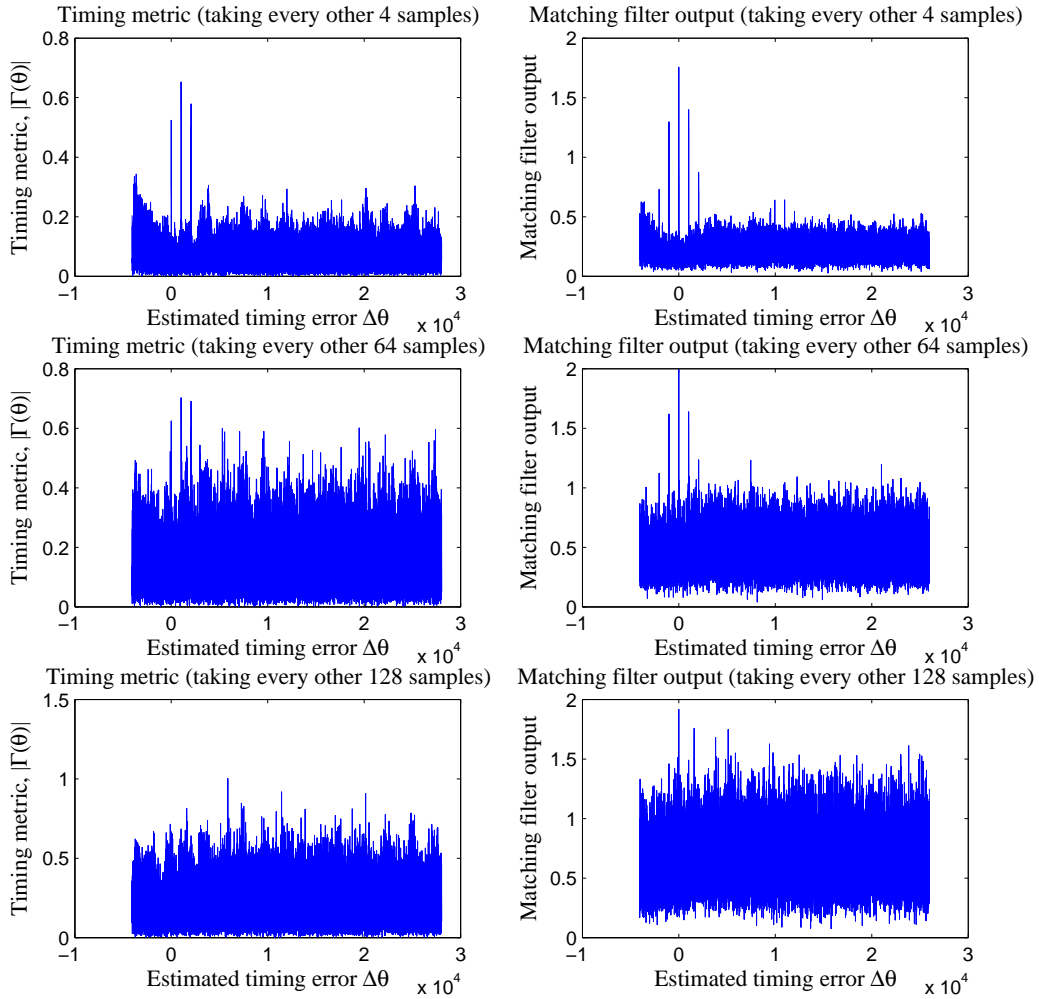


Figure 4.22: Timing metric and matching filter's output when skipping different numbers of data samples in Algorithm II.

4.3.3 Complexity Reduction by Truncating Data Bits

In real applications, hardware has limited precision and all calculations are achieved using fixed-point precision. This means that all numbers are represented and manipulated using a fixed number of binary bits. The data used for these calculations thus needs to be truncated to a certain precision level. This introduces in truncation noise. Thus achieving the desired precision and accuracy while keeping the hardware complexity as low as possible is a major challenge in FPGA designs. It has been mentioned that FPGA boards have a limited number of multipliers, and most of the FPGA multipliers are of 18 bits accuracy. Both the multiplicands and the product are represented with a fixed 18 bits in total including integer and fractional bits. However, a 18 bits multiplier can be split into 2 separate 9 bits multipliers if a precision of 9 bits is enough to achieve the desired performance. Therefore, it is preferred to use 9 bits multipliers if possible to reduce the number of multipliers required by half. On the other hand, for projects with a large production volume, it is more economical to develop and use ASIC (application-specific integrated circuit) over FPGA. ASIC can be optimized to have customized multiplier size to obtain the desired precision at a minimum cost. Therefore it is also of interest to have an idea about the minimum number of bits required for the multipliers without causing severe performance loss with our proposed Algorithm II.

4.3.4 Performance

As Algorithm II provides such a sharp peak, the timing estimation is much more precise in this case. However, as more samples are skipped and discarded, the less useful information is available. This makes the task to identify the desired peak more challenging. In addition, truncation noise also makes the difference between the peaks and noise floor of the timing metric smaller. To determine a good combination for the number of samples to be skipped and the number of bits to be used for the multipliers, 1000 simulations runs were completed with different combinations of the two quantities. The channel condition is kept the same as what was used in the second

simulation for Algorithm I, i.e., the worst case channel condition. Table 4.4a shows the performance in terms of the number of estimation failures (timing estimation error that does not fall into the safety region) for different numbers of multipliers used, and the sizes of the multipliers. Specifically, each row represents a set of simulation results using different multiplier sizes, whereas each column represents a set of simulation results with different precision levels for data representation.

	6 bits	8 bits	9 bits	18 bits		6 bits	8 bits	9 bits	18 bits
$N/2$	0	0	0	0	$N/2$	0	0	0	0
$N/8$	0	0	0	0	$N/8$	0	0	0	0
$N/16$	0	0	0	0	$N/16$	1	0	0	0
$N/32$	0	0	0	0	$N/32$	1	0	0	0
$N/64$	0	0	0	0	$N/64$	20	1	0	0
$N/128$	3	0	0	0	$N/128$	618	379	375	363
$N/256$	503	0	0	0	$N/256$	951	851	838	838

(a) Simulation results under SNR of 35dB. (b) Simulation results under SNR of 8dB.

Table 4.4: Number of estimation failures over 1000 simulation runs using different numbers of multipliers and multiplier sizes under different SNR values for Algorithm II.

From the results presented in Table 4.4a, it can be seen that under SNR of 35 dB multipliers of sizes 9 and 18 bits perform free of errors. However the results do not provide enough information regarding the relationship between hardware complexity and performance as the failure rates for most of the combinations are zero. Though SNR of 35dB is the worst noise level for a cable channel, to get a better understanding of the relationship between the hardware complexity and performance another 1000 simulations are completed with SNR of 8dB while keeping every other parameter unchanged. The new results are listed in Table 4.4b. It can be seen that multipliers of sizes 9 and 18 bits have similar performance, this means that the 18-bits multipliers on FPGA board can be split into 2 separate 9-bits multipliers for Algorithm II with-

out any performance loss. As the noise level increases, skipping more data samples greatly reduces the useful information available. Compared to Table 4.4a for SNR of 35dB, the failure rates for $N/128$ and $N/256$ number of multipliers have increased significantly. Therefore under SNR of 8dB the maximum number of data samples that can be discarded between every two retained samples is 32, leading to the number of multipliers required to be $N/64$. When the number of multipliers is kept at $N/64$, the multiplier size can be reduced to 9 bits. However, for a practical application in a cable channel the worst noise level is SNR of 35dB. The maximum number of data samples discarded between each retained sample pair can be increased to 128. This results in $N/256$ multipliers and the multiplier size can be reduced to 8 bits while still providing accurate timing estimation.

4.4 Algorithm III

Algorithm II provides a precise timing offset estimation but with a non-negligible increased hardware cost as compared with Algorithm I. To retain its performance advantage while further reducing the hardware cost, a new approach is proposed by further modifying Algorithm II.

4.4.1 Algorithm

This new algorithm still exploits the fact that the preamble symbol exhibits the mirrored conjugate symmetry unique to BPSK modulated time domain signals. Similar to Algorithm II, two sliding windows of lag $N/2$ are applied to the time domain samples. To avoid the use of multipliers, the sum of the differences is used rather than the correlation calculation to locate the matching points. In this way, adders are used in places where multipliers are used in Algorithm II. First the absolute values of all the samples are calculated and the samples in the second window are flipped with respect to the center to pair with the samples in the first window. The timing metric is obtained by summing up the absolute value of the difference between each corresponding sample pair from the two windows as shown in Equation (4.8). When

the two sliding windows are aligned with the preamble symbol, symbols in each pair have the same absolute value, and the sum of the differences is zero.

$$\Gamma_{\text{III}}(\theta) = \sum_{m=0}^{N/2-1} |y_{\text{fr}}[\theta + N/2 - m]^2 - y_{\text{fr}}[\theta + N/2 + m]^2|. \quad (4.8)$$

Instead of three peaks, the obtained timing metric results in three sharp pitches of zero at the matching location indexes as illustrated in Figure 4.23. The timing metric starts up with almost zero value because the first symbol of the ranging frame is an empty symbol. Then the timing metric value starts to increase once the first non-zero preamble symbol is captured by the sliding window. The timing metric reaches its maximum after $N/2$ when the first sliding window is filled with samples from the preamble symbol while all values from the second window remain zero. The timing metric then decreases in magnitude until another $N/2$ samples later when both windows are filled with non-zero symbol data. Then the timing metric value stays almost constant except for the three sharp pitches.

To help with locating the three pitches, a matching filter is applied to the timing metric similar to what performed in Algorithm II (described in Section 4.3.1). An example of the output of the matching filter is shown in Figure 4.24. Again the x-axis represents the timing estimation error and the two plots are shifted so that they are aligned at the perfect timing index. In the time domain, there should be a lag of N in the bottom plot compared to the top timing metric plot.

To locate the perfect timing index, the smallest middle pitch needs to be identified. It is not practical to simply look for the minimum valued location since when there is nothing transmitted or during the first empty symbol the matching filter's output is close to zero. As a result, simply looking for the minimum valued point will likely result in an incorrect index being selected. Thus, it is necessary to wait till the first non-zero symbol has been received before looking for the minimum valued point. To address this issue, a triggering condition is set so that the estimator triggers when the timing metric $\Gamma_{\text{III}}(\theta)$ has been continuously increasing for $N/2$ samples and

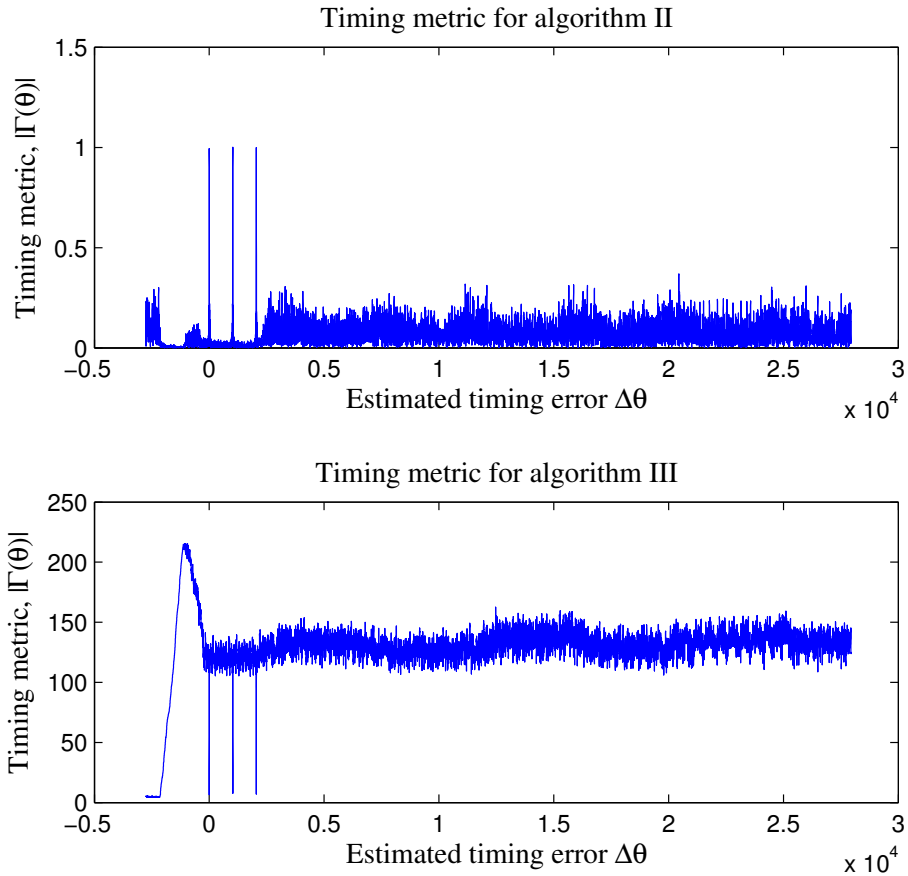


Figure 4.23: Timing metric comparison for Algorithm II and Algorithm III.

then decreasing for $N/2$ samples. This works on the premise that the first symbol in a ranging request signal frame is empty as discussed above. Ideally, this triggering position lies right where the first matching position is at. In Figure 4.24, this triggering condition is marked by the vertical dashed line. Once the triggering condition has been met, the index where the matching filter's output is at its minimum over the next $2N$ samples is returned. In this way, it will only pick up the desired pitch. Denote this minimum valued index as θ_{\min} . Then the estimated timing offset is equal to:

$$\hat{\theta} = \theta_{\min} - N \quad (4.9)$$

Figure 4.25 shows the block diagram for the third method. A shift register of

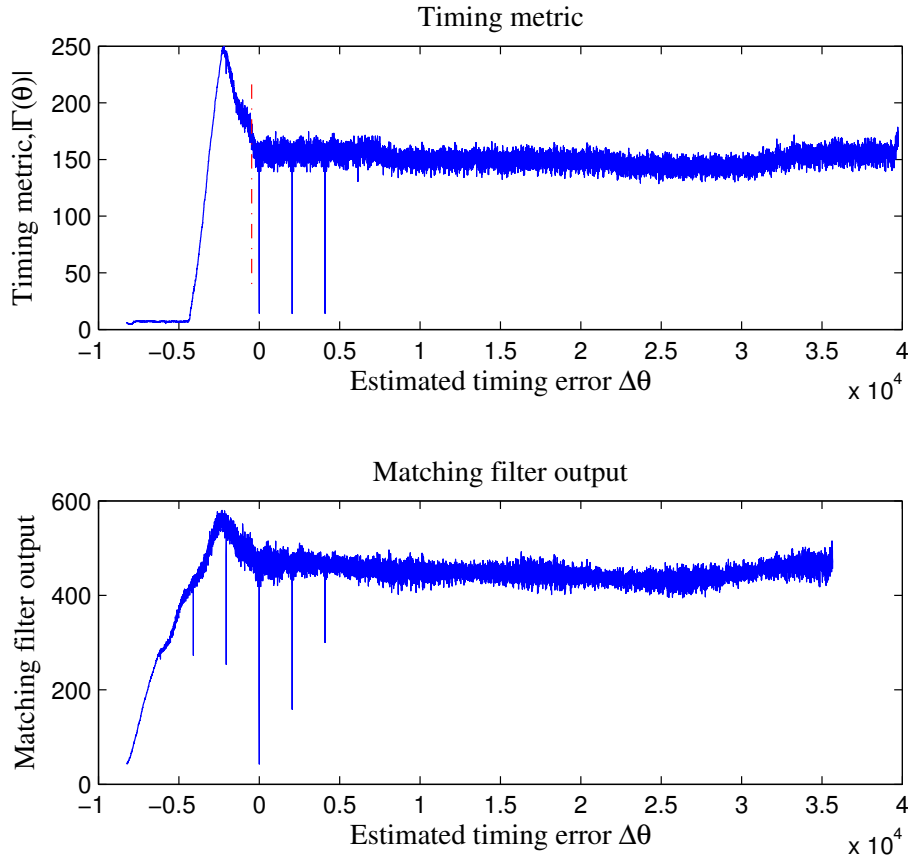


Figure 4.24: Applying a matching filter to the timing metric in Algorithm III.

size N is used to store the absolute values of the received data samples. Since the correlation is no longer calculated, no multipliers are required to calculate the cross products. Instead, adders are used for calculating the difference between the absolute values of the sample pair. This method eliminates a great number of multipliers compared to the second method and is by far the method with lowest hardware cost among the three methods discussed.

4.4.2 Complexity Reduction by Skipping Data Samples

Similar to the “skipping” technique that is applied to Algorithm II, the number of adders used in Algorithm III can be reduced as well. Instead of calculating the difference between the absolute sample values in each pair as shown in Figure 4.25,

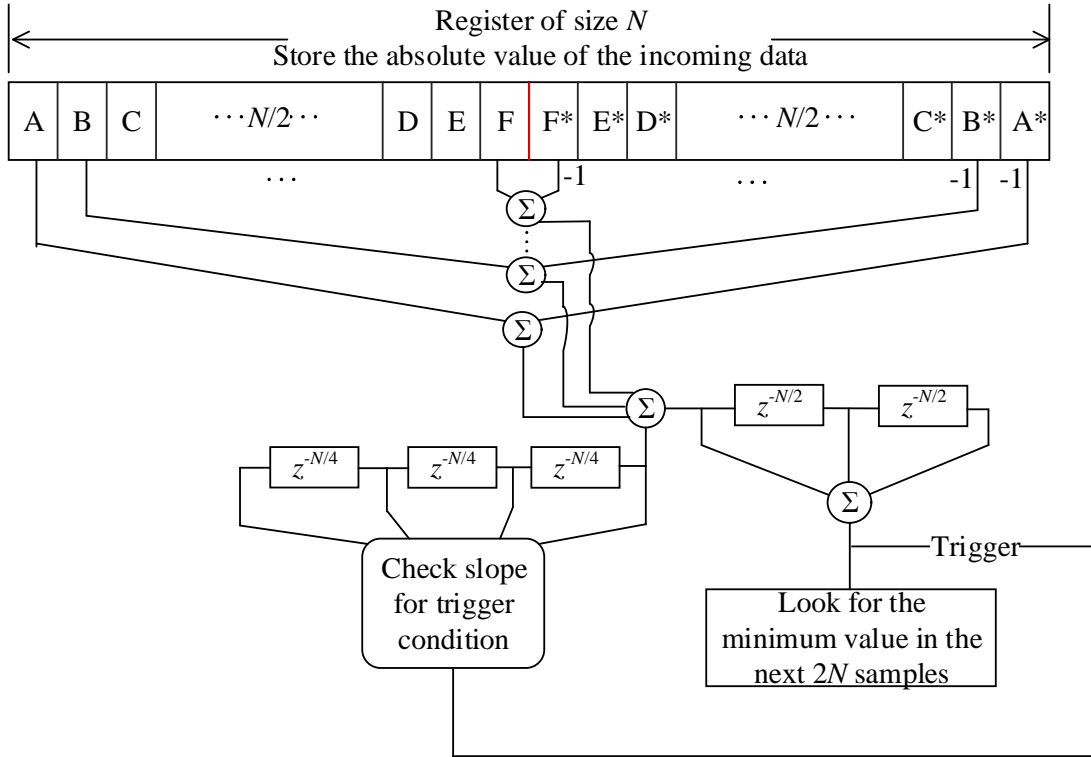


Figure 4.25: Block diagram for Algorithm III.

only a fraction of the data samples in the window is used for the difference calculation. The number of samples to be discarded is chosen so that the pitches in the timing metric remain distinct. Figure 4.26 shows the impact of the skipping factor on the timing metric and the matching filter's output of Algorithm III.

4.4.3 Complexity Reduction by Truncating Data Bits

As Section 4.3.3 has discussed, embedded systems (FPGA or ASIC) have limited precision and all numerical data is represented using a fixed number of binary bits. Since there is a large number of adders that Algorithm III requires, it is important to use a minimum number of bits possible without degrading the timing metric properties which causes performance loss. Adders can potentially have any size in terms of the number of bits. In this thesis, the adder sizes considered are 4, 6, 8, and 16 bits.

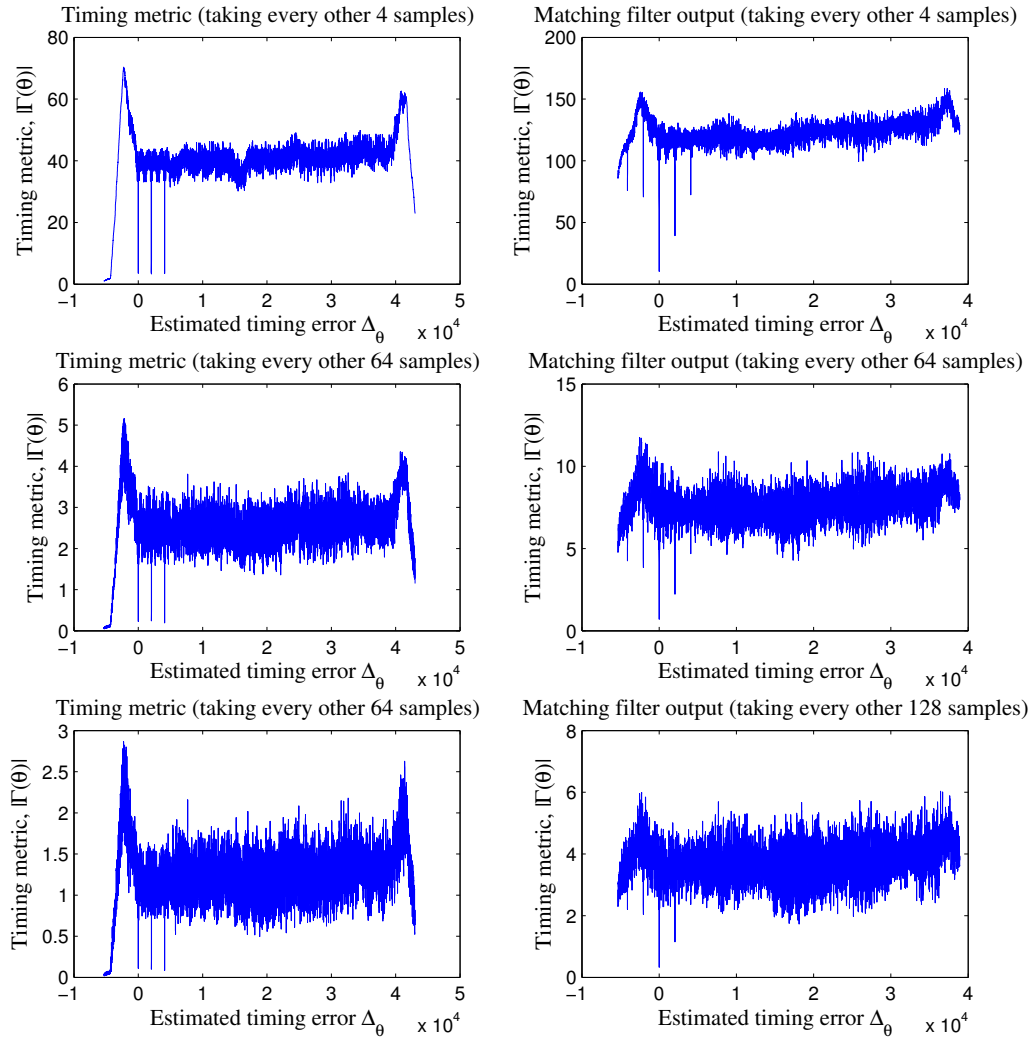


Figure 4.26: Timing metric and matching filter's output when skipping different numbers of data samples in Algorithm III.

4.4.4 Performance

Under the same severe channel setup as in Algorithm II, 1000 simulation runs were completed with various combinations of the numbers of multipliers required and adder sizes with various SNR values. Table 4.5 shows the simulation results under each noise level in terms of number of the failures in each simulation run for different numbers of adders and adder sizes. Specifically, each column represents timing estimation failure counts with a fixed adder size. Each row represents the number of timing estimation failure for a fixed number of available adders.

	6 bits	8 bits	12 bits	16 bits		6 bits	8 bits	12 bits	16 bits
$N/2$	0	0	0	0	$N/2$	0	0	0	0
$N/8$	0	0	0	0	$N/8$	0	0	0	0
$N/16$	0	0	0	0	$N/16$	0	0	0	0
$N/32$	0	0	0	0	$N/32$	0	0	0	0
$N/64$	0	0	0	0	$N/64$	6	1	1	1
$N/128$	10	1	0	0	$N/128$	194	70	28	30
$N/256$	352	18	2	2	$N/256$	832	448	328	324

(a) Simulation results under SNR of 35dB. (b) Simulation results under SNR of 18dB.

	6 bits	8 bits	12 bits	16 bits
$N/2$	0	0	0	0
$N/8$	0	0	0	0
$N/16$	3	0	0	0
$N/32$	12	1	0	0
$N/64$	104	20	8	8

(c) Simulation results under SNR of 15dB.

Table 4.5: Number of estimation failures over 1000 simulation runs using different numbers of multipliers and multiplier sizes under different SNR values for Algorithm III.

It can be seen from the simulation results shown in Table 4.5a that under SNR of 35dB a maximum of 64 data samples can be discarded between every two retained samples, leading to $N/128$ adders in total. With $N/128$ adders, the adder size can be reduced to 12 bits without any performance degradation. Once the noise level is increased to SNR of 18dB, the failure rates for the $N/128$ and $N/256$ multipliers cases have significantly increased as shown in Table 4.5b. The maximum number of samples that can be discarded between every two retained samples is decreased to 16 to ensure error free synchronization. The minimum adder size is 6 bits. Once the SNR value is decreased to 15dB, Algorithm III can hardly recognize the pattern when a ranging symbol has been received with $N/128$ and $N/256$ multipliers cases. This is because Algorithm III is only triggered when it detected a non-zero symbol after an empty symbol, that is when the timing metric $\Gamma_{\text{III}}(\theta)$ has been continuously increasing for $N/2$ samples and then decreasing for $N/2$ samples. With fewer data samples available to contribute to the timing metric $\Gamma_{\text{III}}(\theta)$ pattern, Algorithm III can no longer gets triggers properly and return a reasonable timing estimation result. Therefore, only five different number of multipliers are used for SNR of 15dB simulation as shown in Table 4.5c. The failure rate for the $N/64$ multipliers case has increased a bit compared to SNR of 18dB and the minimum number of multipliers required is $N/32$ with a minimum adder size of 12 bits.

4.5 Summary and Comparison

There has been other research done on the timing estimation algorithms for OFDM systems, but due to the special configuration of the DOCSIS 3.1 OFDM ranging signal structure, none of the methods can be directly applied to the DOCSIS 3.1 system.

A modified version of the S&C method is proposed as Algorithm I to work with DOCSIS 3.1 specifications. This algorithm makes use of the repetitive pattern of the symbols in the ranging frame. By determining the location of the symbol pairs, the timing offset estimate is obtained from the symbol pairs location information. Only two multipliers are needed for this algorithm. The algorithm returns reason-

able timing estimation results when the system and channel condition are randomly generated. However, in a severe channel condition and busy system where there are multiple other CMs transmitting next to the ranging CM and at a much larger power level, this algorithm's failure rate increases significantly, to about 37%.

A second algorithm is proposed by exploiting the special time-domain symmetry of the BPSK-modulated signal. This algorithm gives much more precise timing estimation results compared to Algorithm I, but its hardware complexity is much higher than the first algorithm, requiring additional $N/2$ multipliers. Techniques for reducing the hardware complexity of Algorithm II based on skipping or truncating the incoming samples were investigated. Simulations were completed using different combinations of these techniques with the same worst case scenario setup as in Algorithm I under a few different SNR values. The final results show that to have an accurate timing estimation with no estimation failures under a practical cable channel, the number of multipliers can be reduced to $N/256$ and the multiplier size can be as small as 8 bits.

Finally, Algorithm III is proposed as a further improvement of Algorithm II in terms of implementation. It reduces hardware complexity by replacing the multipliers with adders and at the expense of having a slightly worse performance under the same simulation setup compared to Algorithm II. It is also more sensitive to white noise and as the SNR value decreases, the performance decreases significantly. For a practical cable channel, the number of adders can be reduced to $N/128$ with an adder size of 12 bits while maintaining zero estimation failure.

5. Summary and Conclusions

Due to the rapidly increasing demand for high speed services over cable network transmission, the newest cable system standard, DOCSIS 3.1, has been released. In DOCSIS 3.1, the multi-carrier OFDM technique is introduced for the first time in the cable industry. OFDM systems are more robust against frequency-selective channel distortions as the multipath effects can be easily compensated with the use of cyclic prefix and IFFT/FFT. However, the advantages of OFDM over single-carrier QAM systems can only be fully realized when the receiver is perfectly synchronized because an OFDM system is much more sensitive to synchronization errors. In this thesis, a study on timing synchronization algorithms for DOCSIS 3.1 upstream OFDMA systems has been performed. Upstream synchronization is challenging as the signal received at the CMTS is made up of many signals transmitted from all other users (CMs) in the network. The purpose of this thesis is to develop accurate and economical timing synchronization algorithms tailored for the DOCSIS 3.1 specifications.

The basic operating principles of an OFDM system were first introduced. Then the effects of synchronization errors were discussed. OFDM systems are especially sensitive to frequency offset since it destroys the orthogonality among the subcarriers and introduces ICI. However, timing offsets are acceptable within a certain range which is called a safety window due to the addition of CP. Timing synchronization is the first step performed before frequency synchronization, hence it is crucial to keep the timing synchronization accurate for the frequency synchronization task that follows. Many algorithms have been proposed in the past to achieve timing synchronization

in OFDM systems. Three classical methods were discussed in this thesis. They all require the use of a training symbol which has a special pattern that is known to both the CMTS and CM. The timing offset is obtained by locating this training symbol which is always the first symbol transmitted utilizing a known pattern.

However, DOCSIS 3.1 has an unique symbol structure requirement and the transmitted symbols are limited to what they can be. Due to these limitations, none of the previously-proposed timing synchronization algorithms can be directly applied in DOCSIS 3.1 systems. After studying in detail the signal structure of the ranging signal that is used for timing synchronization in DOCSIS 3.1, three algorithms were proposed. The first algorithm is a variation of one of the previously developed algorithms. It utilizes the feature that all symbols in the ranging frame are repeated and transmitted as symbol pairs. It locates the location of each repetitive symbol pair by calculating the correlation of samples in two adjacent windows of the symbol size. Due to the addition of CP and CS, this method has a large plateau which makes locating the right symbol location difficult. This first method gives a reasonable performance while the system's parameters are randomly generated within the specified ranges for a cable system. However, when the system is under the most severe channel conditions, the failure rate of the algorithm increases to almost 37%. Two new algorithms were then proposed, both of which utilize the fact that BPSK-modulated symbols have mirrored conjugate symmetry in the time domain. Algorithms II and III are much more accurate compared to Algorithm I as they provide extremely sharp and definitive peaks of the timing metrics.

As the proposed timing recovery algorithms will most likely be built on FPGA development boards and because FPGAs have limited number of multipliers, the number of multipliers required in each algorithm is a key factor determining the cost of the algorithms. Algorithm I requires only two multipliers, Algorithm II requires $N/2$ multipliers and Algorithm III requires no multipliers but $N/2$ adders. Two options were then suggested to lower the hardware complexity of Algorithms II and III. The first option is to take fewer data samples (i.e., skipping data samples) for the

calculation of a timing metric. The second option is to sacrifice some calculation precision by truncating all data paths to a reduced bit length (i.e., truncating data bits). Simulations were conducted under the most severe channel condition and with different combinations of the two cost reduction options. To get a better understanding of the relationship between the hardware cost and performance, a few different SNR values are also used to test the robustness of the two algorithms under higher noise levels than in a cable channel. A key conclusion is that to achieve accurate timing estimation under the worst-case channel condition specified in DOCSIS 3.1, Algorithm II can use $N/256$ 8-bits multipliers and Algorithm III can reduce the number of adders to $N/128$ with the adder size of 12 bits.

References

- [1] Society of Cable Telecommunications Engineers, *Data Over Cable Service Interface Specification DOCSIS 3.1-Physical Layer Specification*. CableLabs, 2013.
- [2] E. Smith, “The emergence of CATV: A look at the evolution of a revolution,” *Proceedings of the IEEE*, vol. 58, pp. 967–982, July 1970.
- [3] Society of Cable Telecommunications Engineers, *Data Over Cable Service Interface Specification DOCSIS 1.0 Radio Frequency Interface*. CableLabs, 1997.
- [4] Society of Cable Telecommunications Engineers, *Data-Over-Cable Service Interface Specifications DOCSIS 2.0 Radio Frequency Interface Specification*. CableLabs, 2001.
- [5] Society of Cable Telecommunications Engineers, *Data Over Cable Service Interface Specification DOCSIS 3.0-Physical Layer Specification*. CableLabs, 2007.
- [6] S. O. Kasap, *Principles of Electrical Materials and Devices*. McGraw-Hill, 2000.
- [7] H. Minn, Vijay K. Bhargava, and Khaled Ben Letaief, “A robust timing and frequency synchronization for OFDM systems,” *IEEE Transactions Wireless Communications*, vol. 2, no. 4, pp. 822–838, 2003.
- [8] P. Xiao, C. Cowan, T. Ratnarajah, and A. Fagan, “Time synchronization algorithms for IEEE 802.11 OFDM systems,” in *IET International Conference on Wireless Mobile and Computing (CCWMC 2009)*, (Shanghai, China), pp. 287–290, Dec 2009.
- [9] K. Shi and E. Serpedin, “Coarse frame and carrier synchronization of OFDM systems: A new metric and comparison,” *IEEE Transactions Wireless Communication*, vol. 3, no. 4, pp. 1271–1284, 2004.

- [10] L. Sanguinetti, M. Morelli, and H. V. Poor, "Uplink synchronization in ofdma spectrum-sharing systems," *IEEE Transactions on Signal Processing*, vol. 58, pp. 2771–2782, May 2010.
- [11] Jan-Jaap van de Beek, Magnus. sandell, and Per Ola Borjesson, "ML estimation of time and frequency offset in OFDM systems," *IEEE Transactions on Signal Processing*, vol. 45, pp. 1800–1805, July 1997.
- [12] T. M. Schmidl and D. C. Cox, "Robust frequency and timing synchronization for ofdm," *IEEE Transactions on Communucations*, vol. 45, pp. 1613–1621, Dec. 1997.
- [13] H. H. Nguyen and E. Shwedyk, *A First Course in Digital Communications*. Cambridge University Press, 2009.
- [14] H. H. Nguyen, *Introduction to OFDM*. EE810 Lecture Notes, University of Saskatchewan, 2013.
- [15] H. H. Nguyen, *Matlab Simulation of an Orthogonal Frequency-Division Multiplexing (OFDM) System*. EE810 Lecture Notes, University of Saskatchewan, 2013.
- [16] T. Pollet and M. Moeneclaey, "The effect of carrier frequency offset on the performance of band limited single carrier and OFDM signals," *Proc. of IEEE Globecom*, pp. 719–723, 1996.
- [17] B. Berscheid, *FPGA-Based DOCSIS Upstream Demodulation*. Ph.D. Thesis, University of Saskatchewan, July, 2011.
- [18] S. Kaur, C. Singh, and A. S. Sappal, "Effects and estimation techniques of symbol time offset and carrier frequency offset in OFDM system: Simulation and analysis," *International Journal of Electronics and Computer Science*, pp. 1188–1196, 2012.

- [19] A. J. Al-Dweik, “A novel non-data-aided symbol timing recovery technique for ofdm systems,” *IEEE Transactions on Communications*, vol. 54, pp. 37–40, Jan 2006.
- [20] C. R. N. Athaudage and A. D. S. Jayalath, “Blind estimation of residual timing error in ofdm receivers: a non-data-aided maximum-likelihood approach,” in *2005 IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 4, pp. 2476–2480 Vol. 4, Sept 2005.
- [21] L. Nasraoui, L. N. Atallah, and M. Siala, “Encoding sequence design for a reduced complexity time synchronization approach for ofdm systems,” in *Electronics, Circuits and Systems (ICECS), 2012 19th IEEE International Conference on*, pp. 913–916, Dec 2012.
- [22] L. Nasraoui, L. Najjar Atallah, and M. Siala, “Genetic algorithm based optimization of encoding sequence for a reduced complexity ofdm time synchronization technique,” in *Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th*, pp. 1–5, Sept 2013.
- [23] F. Wu and Mosa Ali Abu-Rgheff, *Time and Frequency Synchronization Techniques for OFDM Systems operating in Gaussian and Fading Channels: A Tutorial*. University of Plymouth.
- [24] B. Park, H. Cheon, C. Kang, and D. Hong, “A novel timing estimation method for ofdm systems,” *IEEE Communications Letters*, vol. 7, pp. 239–241, May 2003.