

國立交通大學

電信工程學系碩士班

碩士論文

應用於正交分頻多工系統之框碼同步方法

及其效能評估

Frame Synchronization Techniques and Performance
Evaluation for OFDM-Based Systems

研究生：詹克偉

指導教授：李程輝 教授

中華民國九十三年六月

應用於正交分頻多工系統之碼框同步方法
及其效能評估

Frame Synchronization Techniques and Performance
Evaluation for OFDM-Based Systems

研究生：詹克偉
指導教授：李程輝 教授

Student: Ke-Wei Jhan
Advisor: Prof. Tsern-Huei Lee

國立交通大學

電信工程學系碩士班

碩士論文

A Thesis

Submitted to Institute of Communication Engineering
College of Electrical Engineering and Computer Science
National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of
Master of Science

in

Communication Engineering

June 2004

Hsinchu, Taiwan, Republic of China.

中華民國九十三年六月

應用於正交分頻多工系統之框碼同步方法 及其效能評估

研究生：詹克偉

指導教授：李程輝 教授

國立交通大學

電信工程學系碩士班

中文摘要

在現今高速傳輸的通訊系統當中，正交分頻多工 (OFDM) 技術因其有效使用頻寬以及抵抗多重路徑干擾的特性而受到重視。OFDM 系統的主要問題在於時間與頻率的非同步效應會嚴重地降低系統的效能，所以能夠估測正確的時間與頻率偏移變得相當的重要，而本篇論文著重在探討框碼時間同步 (frame timing synchronization) 的問題。我們研究了數種基於循環前置 (cyclic prefix) 的框碼同步技術，並提出了兩個以整體搜尋法 (global search algorithm) 為基礎所改進的演算法。在局部搜尋法 (local search algorithm) 中，我們只搜尋前一個框碼所估測出之時間點附近的取樣點，如此可在高訊雜比的情況下增加框碼時序的同步準確率。另外一個整體搜尋改進法 (modified global search algorithm) 是一個較低複雜度，但可產生與其他演算法相似效能的演算法。本論文所提出的方法皆有自我學習的功能，也就是說，可以在緩慢變化的通道環境下自動追蹤時序。經由模擬的結果，我們所提出的演算法將與其他基於循環前置的框碼同步方法在運算複雜度以及同步效能上的表現逐一比較。

Frame Synchronization Techniques and Performance Evaluation for OFDM-Based Systems

Student: Ke-Wei Jhan

Advisor: Prof. Tsern-Huei Lee

Institute of Communication Engineering

National Chiao-Tung University

Abstract

Bandwidth efficiency and multipath immunity make orthogonal frequency division multiplexing (OFDM) an attractive modulation technique for modern high data rate communication systems. A major concern of OFDM systems is that symbol timing and frequency synchronization errors can seriously degrade the system performance. Therefore, it is important to obtain accurate estimate of time and frequency offsets. In this thesis, we focus on frame timing synchronization. Several cyclic prefix based techniques are investigated and two modified algorithms based on global search algorithm are proposed. In the local search algorithm, we only search the sample points near the estimated point of previous frame; this can increase the accuracy of estimated timing at high SNR. Another algorithm, called modified global search algorithm, yields a comparable performance and lower complexity compared with other techniques. Our proposed algorithms are all self-learning in the sense that they can automatically track slowly time varying channel conditions. The performance and complexity of our proposed algorithms will be compared with other cyclic prefix based algorithms from simulation results.

誌 謝

感謝我的指導教授 - 李程輝教授所給予的指導，啟發與協助，讓我在兩年的研究生涯學習到了解決問題的態度與方法。

感謝交大電信所蘇育德教授以及凌陽科技林仁政博士，在口試時所提出的寶貴建議與指教。

感謝網路技術實驗室的學長們：曾德功、吳佳龍、傅洪勛、羅天佑、謝坤宏、吳銘智，所給予我學業上的協助與經驗傳授。也感謝實驗室的同學們：柏均、柏成、思儒、冠州、程翔與麗雲這兩年來的相互切磋與鼓勵。

感謝我的父親詹孟東先生以及母親胡雪琴女士，從小到大所給予我無微不至的養育及照顧。感謝我的女友蔡宜均女士多年來的陪伴與支持。也感謝所有關心我的家人與朋友們。

這篇論文獻給所有我愛的人與愛我的人。

2004 年六月 于風城交大

Contents

中文摘要	i
English Abstract	ii
誌謝	iii
Contents	iv
List of Tables	viii
List of Figures	ix
Acronym	xiii
Chap 1 Introduction	1
1.1 Background.....	2
1.2 Classification of Synchronization Techniques.....	3
1.2.1 DA/ DD and NDA Synchronization.....	4
1.2.2 FF and FB Synchronization	4
1.2.3 Carrier, Sampling Clock and Frame Synchronization.....	4
1.2.4 Summary.....	5
1.3 Organization of the Thesis.....	6
Chap 2 Overview of OFDM	7
2.1 OFDM Transmission Basics.....	7



2.1.1 OFDM Signal Characteristics.....	9
2.1.2 Implementation Using IFFT / FFT.....	12
2.1.3 OFDM Bandwidth Efficiency.....	14
2.2 Guard Interval and Cyclic Prefix.....	15
2.2.1 ISI and ICI Avoiding.....	15
2.2.2 Linear Convolution Equivalent.....	20
2.3 Windowing.....	21
2.3.1 Common Used Window Type.....	22
2.3.2 Choice of Roll-Off Factor.....	23
2.3.3 Decision between Windowing and Filtering.....	25
2.4 Choice of OFDM Parameters.....	25
2.4.1 Guard Time and Symbol Duration.....	26
2.4.2 Number of Subcarriers.....	26
2.4.3 A System Design Example.....	27
Chap 3 Frame Synchronization Techniques	28
3.1 OFDM System Model.....	28
3.1.1 System Description.....	28
3.1.2 Synchronization Task.....	29
3.2 Correlation Frame Synchronization Techniques.....	31

3.2.1 ML Estimation Based on Received Signal.....	31
3.2.2 Peak-Picking Algorithm.....	34
3.2.3 Averaging and Peak-Picking Algorithm.....	36
3.3 Low-Complex Frame Synchronization Techniques.....	36
3.3.1 Complex-Quantization Algorithm.....	36
3.3.2 Smoothing Complex-Quantization Algorithm.....	39
3.3.3 Global Search Algorithm.....	41
3.4 Proposed Frame Synchronization Techniques.....	43
3.4.1 Local Search Algorithm.....	43
3.4.1 Modified Global Search Algorithm.....	44
Chap 4 Simulations and Performance Evaluation	47
4.1 Description of Simulation Model.....	47
4.1.1 Rayleigh Fading Channel Model.....	48
4.1.2 Multipath Channel Model.....	52
4.1.3 Simulation Platform and Parameters.....	54
4.2 Simulation Results.....	55
Part I: The Correlation Functions $G(n)$ and $G_c(n)$	55
Part II: PP / APP / CQ / ACQ Algorithms.....	57
Part III: GS / LS / MGS Algorithms.....	62

Chap 5	Conclusions	71
Appendix A		
	Derivation of the Log-Likelihood Function (3.4)	72
Appendix B		
	Derivation of the Complex-Quantization ML Estimator (3.25)	73
Bibliography		75



List of Tables

4.1	Common simulation parameters.....	54
4.2	MSE of the frame timing against the average channel SNR for PP / APP / CQ / ACQ algorithms.....	61
4.3	Comparison of computational complexity and MSE of the frame timing against the average channel SNR for PP, APP, GS and MGS algorithms.....	68
B.1	Expression of the look-up table in <i>Figure 3.6</i>	74
B.2	The look-up table after convex mapping.....	74



List of Figures

2.1	Function blocks of OFDM system.....	8
2.2	Structure of modulator in an OFDM system with N subcarriers.....	9
2.3	Example of four subcarriers within one OFDM symbol.....	10
2.4	Spectra of individual subcarriers.....	11
2.5	Structure of correlator-based OFDM demodulator.....	12
2.6	Structure of transmitter using IDFT (IFFT).....	13
2.7	Structure of receiver using DFT (FFT).....	14
2.8	Illustration of OFDM bandwidth efficiency: (a) conventional multi-band system, (b) OFDM multi-band system.....	15
2.9	Channel dispersion causes ISI between successive OFDM signals...	16
2.10	OFDM signals with silent GI.....	16
2.11	A delayed OFDM signal with a silent GI caused ICI on next signal..	17
2.12	OFDM signal with cyclic prefix.....	17
2.13	Example of an OFDM signal with three subcarriers in a two-ray multipath channel. The dashed line represents a delayed multipath component.....	18
2.14	A digital implementation of appending CP into OFDM signal in transmitter.....	19
2.15	Structure of a complete OFDM signal with CP.....	19

2.16	Relation between linear convolution and circular convolution of and OFDM signal and channel impulse response.....	21
2.17	Power spectral density (PSD) without windowing for 16, 64, and 256 subcarriers.....	22
2.18	OFDM cyclic extension and raised cosine windowing. T_s is the symbol time, T the FFT interval, T_g the guard time, T_{prefix} the preguard interval, $T_{postfix}$ the postguard interval, and \mathbf{b} is the roll-off factor.....	23
2.19	Spectral of raised cosine windowing with roll-off factor of 0 (rectangular window), 0.025, 0.05, and 0.1.....	24
2.20	OFDM symbol windows for a two-ray multipath channel, showing ICI and ISI, because in the gray part, the amplitude of the delayed subcarrier is not constant.....	24
3.1	OFDM system, transmitting subsequent blocks of N complex data...	28
3.2	Structure of OFDM signal with CP symbols $s(k)$	31
3.3	Structure of the ML estimator.....	34
3.4	Computation of correlation function $G(n)$ using an L -length shift register.....	35
3.5	Geometric representation of the signal set \mathbf{A} , and the quadrants $Q_i, i = 0, 1, 2, 3$ of the complex plane.....	37
3.6	Look-up table implementation of the complex-quantization ML estimator.....	38
3.7	Equivalent implementation of the complex-quantization ML estimator.....	39
4.1	Jakes' Rayleigh fading channel model.....	50
4.2	Modified Jakes' Rayleigh fading channel model (the j^{th} path).....	51

4.3	An example of the simulated output waveform envelope within 0.1 sec.....	51
4.4	Envelope, in-phase component and quadrature component pdfs of modified Jakes' Rayleigh fading channel model output waveforms...	52
4.5	Autocorrelation function of modified Jakes' Rayleigh fading channel model output waveforms.....	52
4.6	Time-dispersive wireless channel model for simulation. (a) unfaded impulse response. (b) unfaded frequency domain channel transfer function.....	53
4.7	Simplified OFDM system platform for simulation.....	54
4.8	Simulated magnitude of $G(n)$ for 5 consecutive OFDM symbols at average channel SNR = 0, 5, 10, 15 and 20 dB.....	56
4.9	Simulated magnitude of $G_c(n)$ for 5 consecutive OFDM symbols at average channel SNR = 0, 5, 10, 15 and 20 dB.....	56
4.10	MSE of frame timing error against the average channel SNR for PP algorithm and APP algorithm using $M = 8, 16, 32, 64, 128$ and 256 ...	57
4.11	Histograms of the frame timing error for PP algorithm and APP algorithm at (a) SNR = 0 and (b) SNR = 10.....	58
4.12	MSE of frame timing error against the average channel SNR for CQ and ACQ algorithm using $M = 8, 16, 32, 64, 128$ and 256	59
4.13	Histograms of the frame timing error for CQ algorithm and ACQ algorithm at (a) SNR = 0 and (b) SNR = 10.....	60
4.14	MSE of frame timing error against the average channel SNR for PP, APP ($M = 8$), and GS algorithm using $M_{GS} = 64, 128$ and 256	63
4.15	(a) Histograms of the frame timing error for PP, APP ($M = 8$), and GS algorithm at SNR = 0.....	63
4.15	(b) Histograms of the frame timing error for PP, APP ($M = 8$), and	

GS algorithm at SNR = 10.....	64
4.16 MSE of frame timing error against the average channel SNR for PP, APP ($M = 8$), GS and LS algorithm using $R = 5$ and $R = 10$	65
4.17 (a) Histograms of the frame timing error for PP, APP ($M = 8$), GS and LS algorithm using $R = 5$ and $R = 10$ at SNR = 0.....	65
4.17 (b) Histograms of the frame timing error for PP, APP ($M = 8$), GS and LS algorithm using $R = 5$ and $R = 10$ at SNR = 10.....	66
4.18 MSE of frame timing error against the average channel SNR for GS, APP ($M = 8$ and 16) and MGS algorithm using $K = 2$	67
4.19 MSE of frame timing error against the average channel SNR for GS, APP ($M = 8$ and 16) and MGS algorithm using $K = 3$	67
4.20 MSE of frame timing error against the average channel SNR for GS, APP ($M = 8$ and 16) and MGS algorithm using $K = 4$	68
4.21 (a) Histograms of the frame timing error for GS, APP ($M = 8$ and 16), and MGS algorithm using $R = 5$ and $R = 10$ at SNR = 0.....	69
4.21 (b) Histograms of the frame timing error for GS, APP ($M = 8$ and 16), and MGS algorithm using $R = 5$ and $R = 10$ at SNR = 10.....	70

Acronym

A/D	analog-to-digital converter
ACQ	averaging and complex-quantization algorithm
APP	averaging and peak-picking frame synchronization algorithm
AWGN	additive white Gaussian noise
BER	bit error rate
BPSK	binary phase shift keying
BRAN	the Broadband Radio Access Networks group
CCK	complementary code keying
CP	cyclic prefix
CQ	complex-quantization algorithm
D/A	digital-to-analog converter
DA	data-aided
DAB	digital audio broadcasting
DD	decision-directed
DFT	discrete Fourier transform
DSP	discrete-time signal processing
DVB	digital terrestrial television broadcasting
ETSI	the European Telecommunication Standards Institute
EWA	exponentially weighted average
EWMA	exponentially weighted moving average
FB	feedback
FEC	forward error correction coding
FF	feedforward
FFT	fast Fourier transform
FIR	finite impulse response
FM	frequency modulation
GI	guard interval
GS	global search algorithm
HF	high frequency
HiperLAN/2	High Performance Local Area Network type 2
i.i.d	independent and identically distributed
ICI	intercarrier interference
IDFT	inverse discrete Fourier transform

IEEE	the Institute of Electrical and Electronics Engineers
IFFT	inverse fast Fourier transform
IFT	inverse Fourier transform
IIR	infinite impulse response
ISI	intersymbol interference
ISM	industrial, scientific, and medical
LMS	least mean square
LOS	line-of-sight
LS	local search algorithm
MA	moving average
MAC	medium access control
MCM	multicarrier modulation
MGS	modified global search algorithm
ML	maximum likelihood
MMAC	the Mobile Multimedia Access Communication
MSE	mean square error
NDA	non-data-aided
OFDM	orthogonal frequency division multiplexing
P/S	parallel-to-serial converter
PAPR	peak-to-average power ratio
pdf	probability density function
PHY	physical layer
PP	peak-picking algorithm
PSD	power spectral density
PSK	phase shift keying
QAM	quadrature amplitude modulation
QPSK	quadrature phase shift keying
r.m.s	root mean square
S/P	serial-to-parallel converter
SMA	shortened moving average
SNR	signal-to-noise ratio
VLSI	very large scale integrate circuit
WLAN	wireless local area network
xDSL	digital subscriber loops

Chapter 1

Introduction

Orthogonal frequency division multiplexing (OFDM) is a widely adopted modulation technique in modern high data rate communication systems. The basic idea of OFDM is to divide the transmission channel into a large number of parallel, orthogonal, low-rate subchannels. With rapid advancement in VLSI and DSP techniques in recent years, the huge computational complexity for implementation due to a large number of subchannels becomes feasible. Actually, the OFDM technique had been adopted for digital audio broadcasting (DAB), digital terrestrial television broadcasting (DVB), digital subscriber loops (xDSL), and wireless local area network (WLAN) applications.

It is known that, compared with single carrier modulation, the multicarrier modulation (MCM) scheme OFDM is more sensitive to the synchronization parameter errors, such as carrier frequency offset, sampling clock offset, and symbol timing offset. We focus on the issues of the symbol timing offset estimation, also called frame synchronization, in this thesis. Suitable frame synchronization techniques can extract the correct FFT window at the OFDM receiver and maintain the demodulation performance. In this thesis, we describe several frame synchronization algorithms for OFDM systems that have been proposed before, propose our new modified algorithms, and evaluate the performance of these techniques in AWGN and multipath Rayleigh fading channel.

1.1 Background

OFDM have recently gained increased interest. WLAN is an important application for OFDM technology. Many WLAN standards such as IEEE 802.11a [1], HiperLAN/2, and MMAC system have accepted OFDM as their physical layer specifications. This section introduces the brief history of OFDM.

OFDM is a special case of MCM, which is the principle of transmitting data by dividing the stream into several parallel bit streams and modulating each of these data stream onto individual subcarriers. Although the origin of MCM dates back to the 1950s and early 1960s with military HF radio links, R. W. Chang in the mid 60s first published a paper demonstrating the concept we today call OFDM [2]. Chang's paper demonstrated the principle of transmission of multiple messages simultaneously through a linear band-limited channel without intercarrier interference (ICI) and intersymbol interference (ISI). The OFDM system developed by Chang differed from traditional MCM system in that the spectra of the subcarriers were allowed to overlap under the restriction they were all mutually orthogonal.

Weinsten and Ebert [3] were the first to suggest the discrete Fourier transform (DFT) and inverse discrete Fourier transform (IDFT) to perform baseband modulation and demodulation in 1971. Currently, OFDM systems utilize fast Fourier transform (FFT) and inverse fast Fourier transform (IFFT) to perform modulating and demodulating of the information data. To combat ISI and ICI, Peled and Ruiz [4] introduced the concept of cyclic prefix (CP) rather than using an empty guard interval (GI) in 1980. This effectively simulates a channel performing circular convolution as long as the length of CP is longer than that of channel impulse response.

In the 1980s, OFDM was studied for high-speed modems, digital mobile communications, and high-density recording. In the 1990s, OFDM was exploited for wideband data communications over mobile radio FM channels, xDSL, DAB, and DVB.

Since the beginning of the nineties, WLAN for the 900-MHz, 2.4-GHz, and 5-GHz ISM bands have been available, based on a range of proprietary techniques. In June 1997, IEEE approved an international interoperability standard. The standard specifies both MAC procedures and three different PHYs. There are two radio-based PHYs using the 2.4-GHz band. The third PHY uses infrared light. All PHYs support a data rate of 1 Mbps and optionally 2 Mbps.

User demand for higher bit rates spurred the development of a higher speed extension to the 802.11 standard. In July 1998, IEEE 802.11 working group developed two new PHY standards. One is the IEEE 802.11b standard using complementary code keying (CCK) modulation scheme in the 2.4-GHz band and provides the data rates of up to 11 Mbps. The other is the IEEE 802.11a standard using OFDM modulation scheme in 5-GHz band and targeting a range of data rates from 6 up to 54 Mbps. This new standard is the first to use OFDM in packet-based communication. To make OFDM effectively a worldwide standard for 5-GHz band, the OFDM standard was developed jointly with ETSI BRAN and MMAC.

1.2 Classification of Synchronization Techniques

In general, symbol timing and carrier frequency offset of a received signal in OFDM systems are unknown in the receiver. In addition, the received signal is disturbed by multipath fading under practical communication environment. Therefore,

they must be estimated correctly through synchronization processes for correct detection. Three classifications of synchronization techniques are described in this section [5].

1.2.1 DA / DD and NDA Synchronization

When the data sequence is known, for example a preamble sequence during acquisition, we speak of *data-aided* (DA) synchronization techniques. When the detected sequence is used as if it were the true sequence, one speaks of *decision-directed* (DD) synchronization techniques. All DD techniques require an initial synchronization parameter estimate before starting the detection process. To obtain a reliable estimate, one may send a preamble of known symbols. The *non-data-aided* (NDA) synchronization techniques are obtained if one actually performs the averaging operation on the synchronization parameter estimate.



1.2.2 FF and FB Synchronization

The synchronization techniques can be categorized according how the synchronization parameter estimate is derived from the received signal. The techniques are called *feedforward* (FF) if the estimate is derived from the received signal before it is corrected. The other techniques are called *feedback* (FB) if they derive the estimate of the synchronization error and feed a corrective signal to the synchronization error corrector. FB structures inherently have the self-learning ability to automatically track slowly varying parameter changes.

1.2.3 Carrier, Sampling Clock and Frame Synchronization

The *carrier synchronization* techniques solve the following synchronization errors: the carrier phase error introduced by the propagation delay in the transmitted

signal, the carrier frequency or phase offset caused by the frequency or phase differences between the oscillators at the transmitter and receiver, the Doppler shift due to mobile movements, and the phase noise introduced by non-linear channels. The *sampling clock synchronization* technique, also called timing recovery scheme, extracts the periodic sampling clock signals at the receiver. There are two methods, named as synchronous and asynchronous, to deal with the timing error and the frequency mismatch in sampling clocks. *Frame synchronization* technique, which is essential for the block transmission of the OFDM signals in frame-packaged system, means finding an estimate of the symbol position where the frame starts. It is usually accomplished with the aid of some special synchronization words.

1.2.4 Summary

This thesis focuses on frame synchronization techniques. Since the OFDM demodulation procedure is collectively performed against all subcarriers, frame synchronization errors affect all carriers. Thus, frame synchronization must be performed correctly. A popular solution for the frame synchronization is to insert some reference symbols within the OFDM signals, and these symbols are then picked up by the receiver to generate the frame clock [6], [7]. For NDA frame synchronization, several guard interval (GI)-based, or cyclic prefix (CP)-based, schemes have been presented to estimate the start position of a new frame [8]-[16]. The basic idea of these schemes is to exploit the cyclic extension preceding a symbol frame. However, the CP-based techniques are considered more advantageous because the use of reference symbols lowers the achievable data rate (bandwidth inefficient). All of the frame synchronization techniques introduced later in this thesis is CP-based, some of them are FF and others are FB synchronization methods. We will give a full detail of the CP characteristics and the CP-based schemes in *Chapter 2* and *Chapter 3*,

respectively.

1.3 Organization of the Thesis

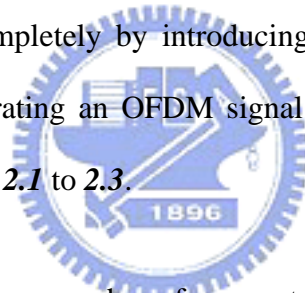
The rest of this thesis is organized as follows. *Chapter 2* introduces the basic concept of OFDM modulation. In *Chapter 3*, we will review some typical CP-based frame synchronization techniques presented in the literature, and describe our proposed modified algorithms. *Chapter 4* contains the simulation models, simulation results and comparison of the performance between the proposed and the previous synchronization techniques. Finally, we will give a conclusion of the thesis in *Chapter 5*.



Chapter 2

Overview of OFDM

The basic principle of OFDM is to split a high-rate data stream into a number of lower rate streams that are transmitted simultaneously over a number of subcarriers. Because the symbol duration increases for the lower rate parallel subcarriers, the relative amount of dispersion in time caused by multipath delay spread is reduced. ISI is eliminated almost completely by introducing a GI in every OFDM symbol. This whole process of generating an OFDM signal and the reasoning behind it are described in detail in *Section 2.1* to *2.3*.



In OFDM system design, a number of parameters are up for consideration, such as the number of subcarriers, guard time, symbol duration, subcarrier spacing, modulation type per subcarrier, and the type of forward error correction coding (FEC). The choice of parameters is influenced by system requirements such as available bandwidth, required bit rate, tolerable delay spread, and Doppler values. Some requirements are conflicting. These design issues are discussed in *Section 2.4*.

2.1 OFDM Transmission Basics

OFDM is a MCM technique based on DFT (FFT) and IDFT (IFFT), *Figure 2.1* shows the common OFDM system function blocks.

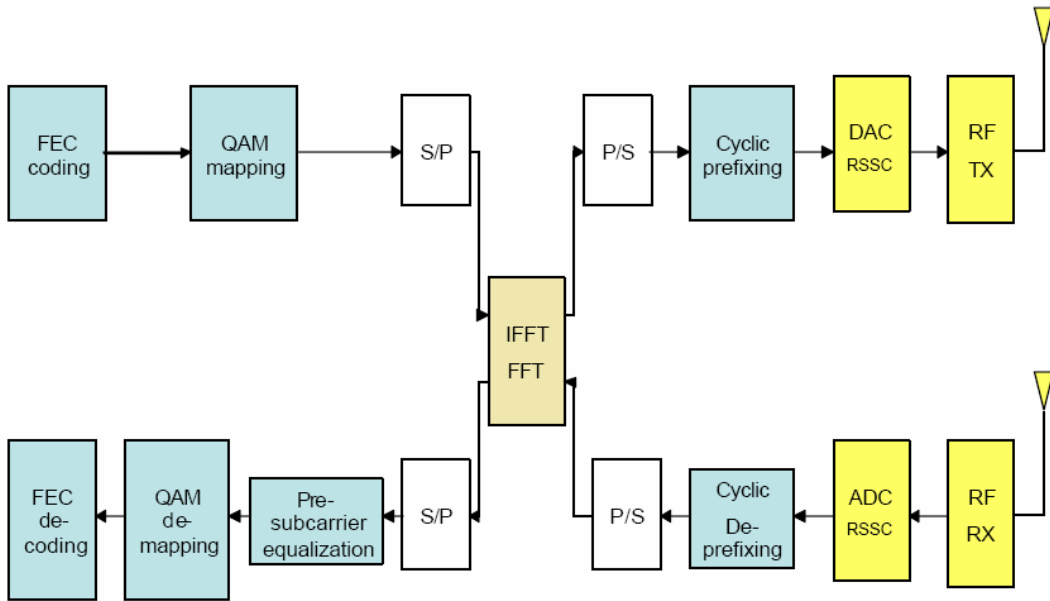


Figure 2.1 Function blocks of OFDM system [25].

An OFDM signal consists of a sum of subcarriers that are modulated by using phase shift keying (PSK) or quadrature amplitude modulation (QAM). The basic structure of OFDM modulator with N subcarriers is shown in **Figure 2.2**. As shown in **Figure 2.2**, the OFDM modulator consists of a serial-to-parallel converter (S/P) and an N -subcarrier modulators bank with different subcarrier frequencies. Firstly, the original data symbol streams are fed into the modulator in a serial way and the S/P divides these symbol streams into N parallel subsymbol streams, and then these subsymbols at each branch are used to modulate the different subcarriers. Assume that the original data symbol rate fed into the OFDM modulator is R_s , the reciprocal of the symbol duration T_s . After the S/P, the symbol duration T at each branch is increased to NT_s , and the symbol rate is down to R_s/N . Hence, the symbol period is N times longer than that of the symbol in a conventional single carrier communication system. This property benefits an OFDM signal transmitted in a multipath channel environment, because the relative amount of dispersion in time can be reduced. In the following, we will introduce the mathematical description of OFDM signals and discuss some characteristics of OFDM systems.

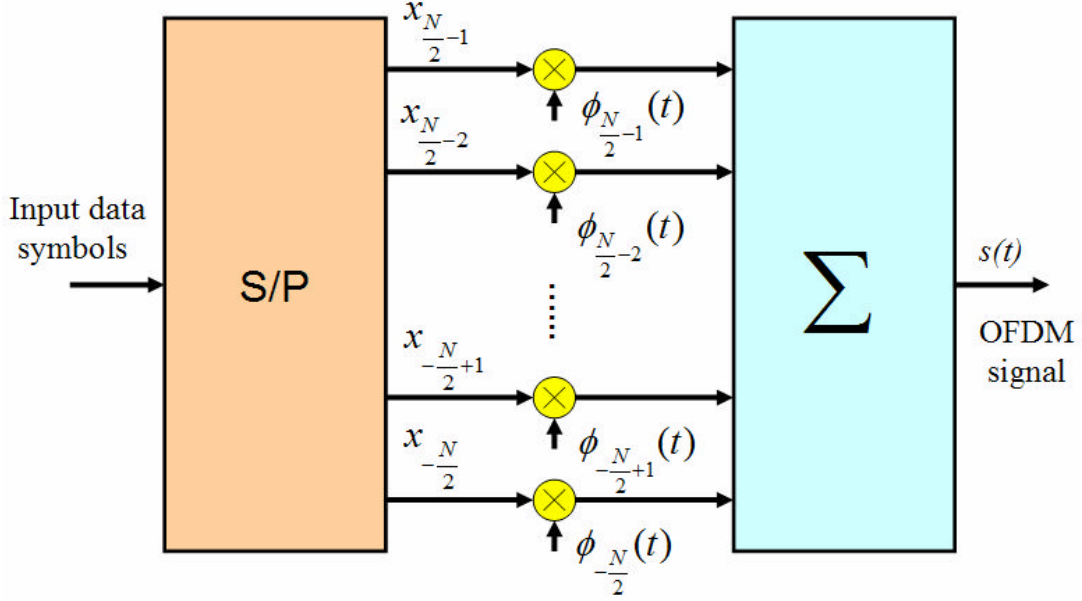


Figure 2.2 Structure of modulator in an OFDM system with N subcarriers [22].

2.1.1 OFDM Signal Characteristics

In **Figure 2.2**, we denote x_k as the transmitted subsymbol at the $(k + \frac{N}{2} + 1)$ -th branch, where k is an integer value from $-\frac{N}{2}$ to $\frac{N}{2} - 1$, which is chosen according to the representation of the subcarrier frequency f_k . In OFDM systems, the transmitted subsymbols x_k usually are PSK or QAM symbols. An OFDM signal with symbol period T generated by the OFDM modulator can be expressed as follows

$$s(t) = \begin{cases} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} x_k f_k(t) & 0 \leq t \leq T \\ 0 & \text{otherwise} \end{cases} \quad (2.1)$$

for $k = -\frac{N}{2}, \dots, 0, \dots, \frac{N}{2} - 1$, where

$$f_k(t) = \frac{1}{\sqrt{T}} e^{j2\pi f_k t} \quad (2.2)$$

is the subcarrier used to modulate the subsymbol x_k at the $(k + \frac{N}{2} + 1)$ -th branch,

and the frequency of the k -th subcarrier f_k is

$$f_k = \frac{k}{T} \quad (2.3)$$

From (2.3), we can see that each subcarrier has exactly an integer number of cycles in the interval T , and the number of cycles between adjacent subcarriers differs by exactly one. This property implies that there is orthogonality among the subcarriers used in OFDM systems. If we multiply the i -th subcarrier $\mathbf{f}_i(t)$ with the complex conjugated version of another subcarrier $\mathbf{f}_j^*(t)$, and integrate the result over the interval of T , we will get

$$\int_0^T \mathbf{f}_i(t) \mathbf{f}_j^*(t) dt = \frac{1}{T} \int_0^T e^{j2\pi \frac{(i-j)}{T} t} dt = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \quad (2.4)$$

In equation (2.4), the result is zero for all other subcarriers except for $i = j$, because the frequency difference $\frac{i-j}{T}$ produces an integer number of cycles within the integration interval T . As an example, **Figure 2.3** shows four subcarriers within one OFDM symbol.

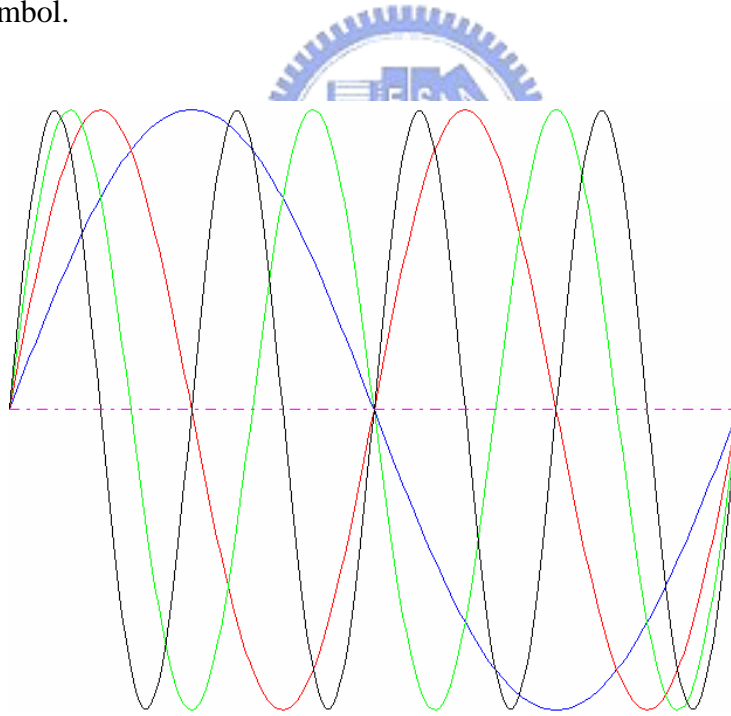


Figure 2.3 Example of four subcarriers within one OFDM symbol [18].

The orthogonality of the different OFDM subcarriers can also be demonstrated in another way. According to (2.1), each OFDM symbol contains subcarriers that are nonzero over an interval T . Hence, the spectrum of a single symbol is a convolution of a group of Dirac pulses located at the subcarrier frequencies with the spectrum of a

square pulse that is one in period T and zero otherwise. The amplitude spectrum of the square pulse is equal to $\text{sinc}(p f T)$, which is zero for all frequencies f that are an integer multiple of $1/T$. This effect is shown in **Figure 2.4**, which shows the overlapping sinc spectra of individual subcarriers. At the maximum of each subcarrier spectrum, all other subcarrier spectra are zero. Because an OFDM receiver essentially calculates the spectrum values at those point that correspond to the maxima of individual subcarriers, it can demodulate each subcarrier free from any interference from the other subcarriers.

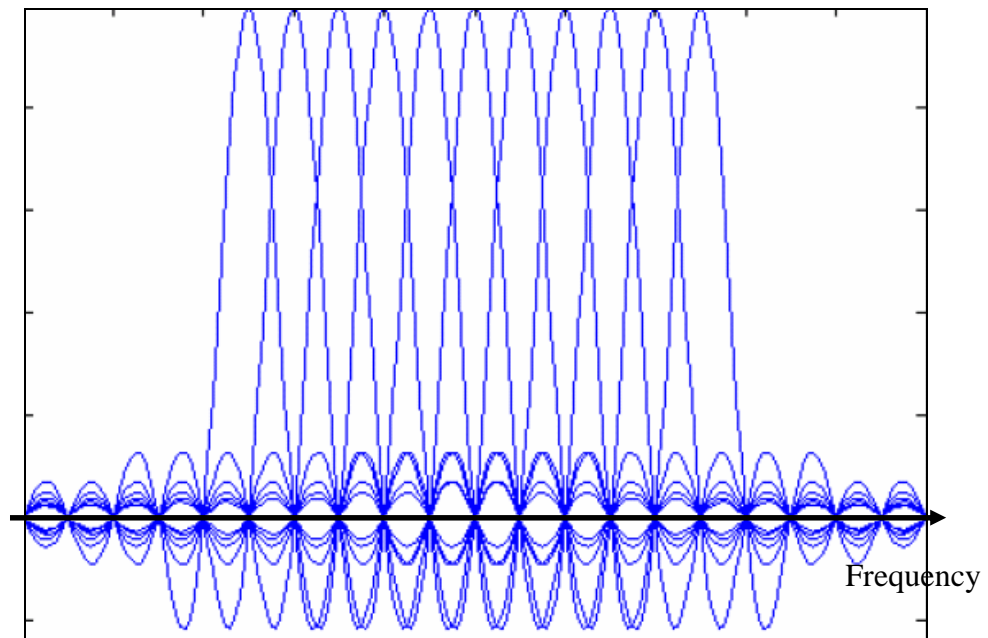


Figure 2.4 Spectra of individual subcarriers [25].

The orthogonal property is useful for the OFDM demodulator to easily demodulate the subsymbol at any of the subcarriers by using the correlator. The basic structure of the correlator-based OFDM demodulator is illustrated in **Figure 2.5**.

The correlator output at the j -th branch in the OFDM demodulator is denoted as

$$y_j: \quad y_j = \int_0^T s(t) \mathbf{f}_j^*(t) dt = \frac{1}{T} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} x_k \int_0^T e^{j2p \frac{(k-j)t}{T}} dt = x_j \quad (2.5)$$

In this way, we can recover the transmitted subsymbols correctly.

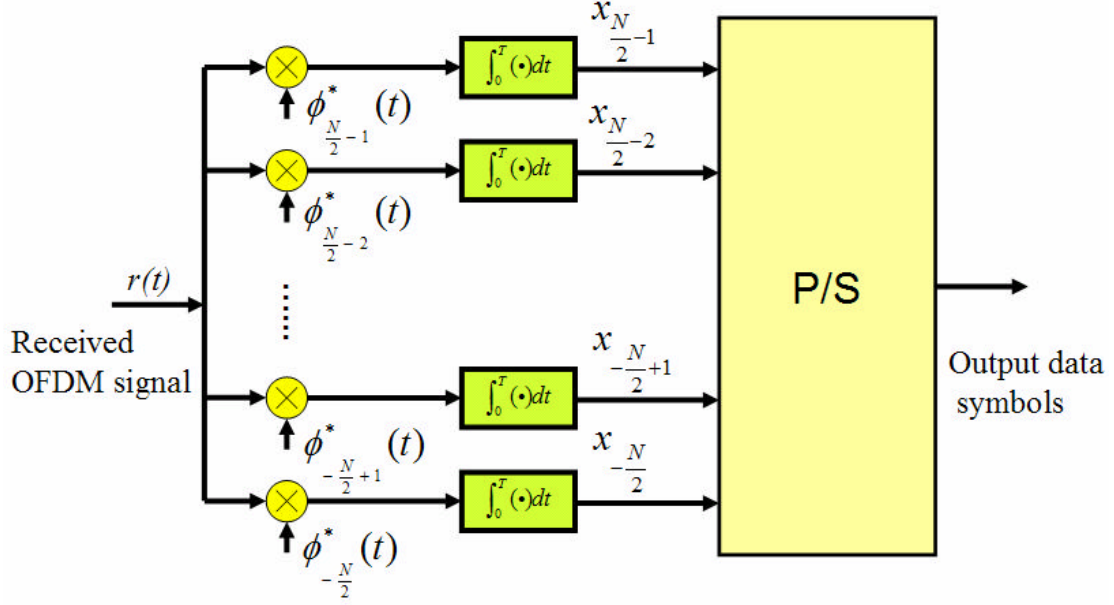


Figure 2.5 Structure of correlator-based OFDM demodulator [22].

2.1.2 Implementation Using IFFT / FFT

The complex baseband OFDM signal as defined by (2.1) is in fact nothing more than the inverse Fourier transform (IFT) of N input subsymbols. If we sample $s(t)$ by the sampling period $T_s = \frac{T}{N}$, then the time discrete equivalent, or IDFT signal model $s[n]$ is given by

$$s[n] = s(t)|_{t=nT_s} = \begin{cases} \frac{1}{N} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} x_k e^{j2\pi \frac{k}{N} n} & 0 \leq n \leq N-1 \\ 0 & \text{otherwise} \end{cases} \quad (2.6)$$

In the receiver, the DFT, the reverse operation of IDFT, can be used to recover the transmitted subsymbols at the OFDM subcarriers. The demodulation result of the j -th subcarrier by using the DFT can be expressed as

$$y_j = \text{DFT}\{s[n]\} = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} s[n] e^{-j2\pi \frac{j}{N} n} = \frac{1}{N} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} x_k N \mathbf{d}[k-j] = x_j \quad (2.7)$$

The demodulation result shown in (2.7) is the same as that in (2.5). Hence, we can

replace the subcarrier oscillators and the correlators used in the OFDM transmitter and receiver by the IDFT and DFT, respectively. In practice, the DFT / IDFT can be implemented very efficiently by the IFFT / FFT. An N point DFT / IDFT require a total of N^2 complex multiplications — which are actually phase rotations. The FFT / IFFT drastically reduce the amount of calculations by exploiting the regularity of the operations in the DFT / IDFT. Using the radix-2 algorithm, an N -point FFT / IFFT require only $(N/2) \cdot \log_2(N)$ complex multiplications [19].

The structure of the OFDM system using the IDFT (IFFT) modulation and the DFT (FFT) demodulation is shown in **Figure 2.6** and **2.7**, where the block "D/A" denotes the digital-to-analog converter that converts the discrete time signal to the continuous time signal, and "A/D" is the analog-to-digital converter which performs the reverse operation of the D/A.

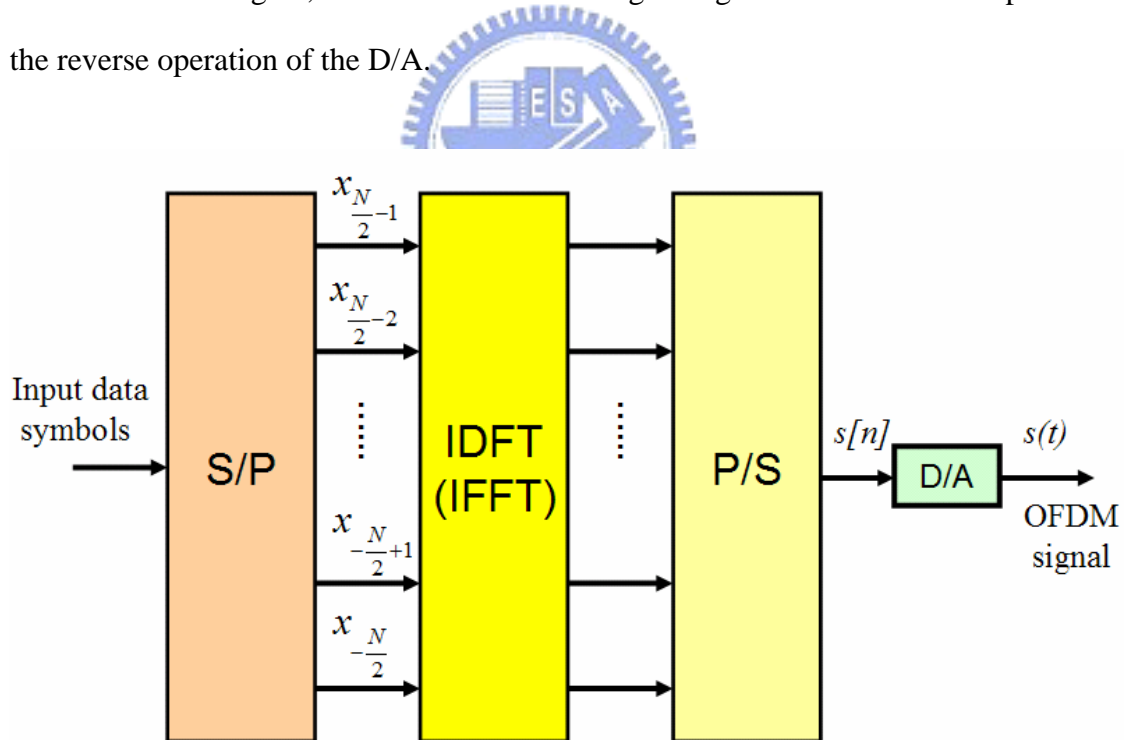


Figure 2.6 Structure of transmitter using IDFT (IFFT) [22].

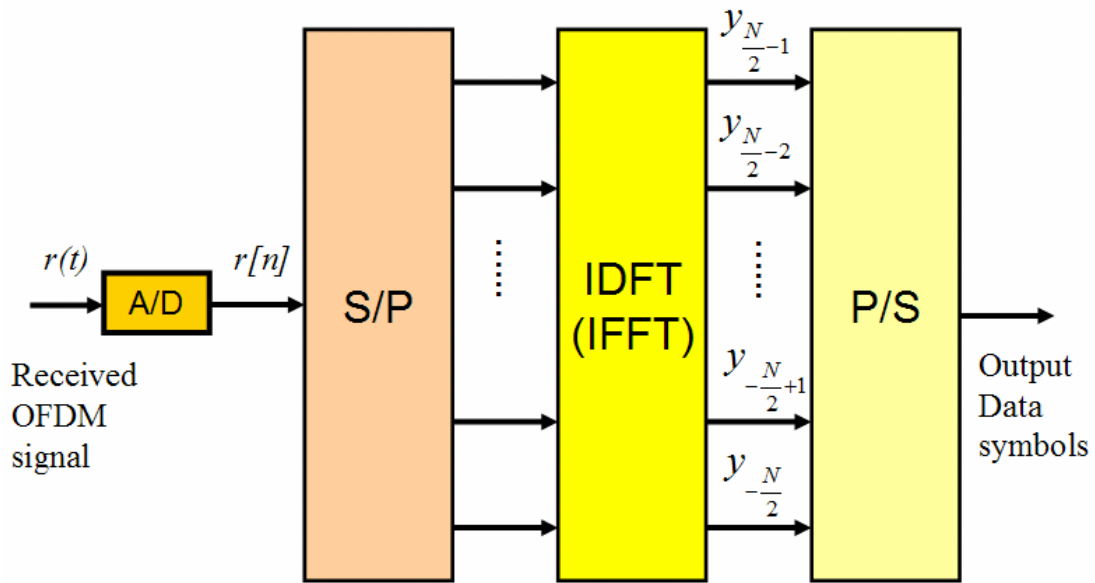


Figure 2.7 Structure of receiver using DFT (FFT) [22].

2.1.3 OFDM Bandwidth Efficiency

For simplicity, we assume that the signal spectra can be band-limited to the bandwidth of its main spectral lobe. In a classical parallel data system, called frequency division multiplexing (FDM), the total signal frequency band is divided into non-overlapping frequency channels. It seems good to avoid spectral overlap of channels to eliminate ICI. As seen in **Figure 2.8 (a)**, the null-to-null bandwidth is $W_B \cong \frac{2}{T_s}$ because the spectrum of the rectangular pulse is represented by the sinc function with first zero at $\frac{1}{T_s}$. Since the bit rate is $R = \frac{\log_2 M}{T_s}$ bits per second, where M is the alphabet, bit-rate-to-bandwidth ratio is $\frac{R}{W} = \frac{1}{2} \log_2 M$. The conventional multi-band system uses the available bandwidth inefficiently.

OFDM is the overlapping multicarrier modulation scheme, as shown in **Figure 2.8 (b)**, the approximate bandwidth of a N subcarrier becomes $W_B \cong (N+1) \frac{1}{NT_s}$ because the frequency separation between adjacent subcarriers is $\frac{1}{NT_s}$ for the signal

orthogonality. Therefore, the bit-rate-to-bandwidth ratio is $\frac{R}{W} = \frac{N}{N+1} \log_2 M$.

When N is large enough, the efficiency of OFDM systems is almost twice as that of FDM systems. Thus, OFDM multi-band systems can more efficiently use the available bandwidth compared to the conventional multi-band systems.

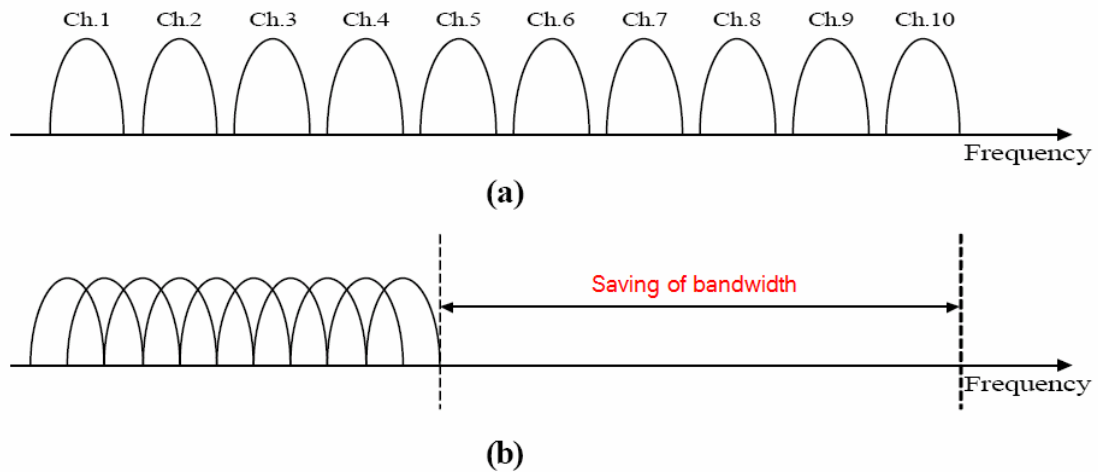


Figure 2.8 Illustration of OFDM bandwidth efficiency: (a) conventional multi-band system, (b) OFDM multi-band system [18].

2.2 Guard Interval and Cyclic Prefix

This section introduces the ideas of guard interval and cyclic prefix, and explains the reason to use CP in OFDM transmission systems.

2.2.1 ISI and ICI Avoiding

One of the most important reasons to do OFDM is the efficient way it deals with multipath delay spread. By dividing the input data stream in N subcarriers, the symbol duration is made N times longer, which also decreases the relative multipath delay spread, relative to the symbol time, by the same factor. In a multipath channel, the delayed replicas of the previous OFDM signal will cause the ISI between

successive OFDM signals as show in **Figure 2.9**. To eliminate ISI almost completely, a guard interval (GI) is introduced for each OFDM symbol. The GI is chosen larger than the expected delay spread, such that multipath components from one symbol cannot interfere with the next symbol. A GI consists no signal at all is inserted between successive OFDM signal as shown in **Figure 2.10**. In this case, however, the problem of ICI would arise. This effect is illustrated in **Figure 2.11**. In this example, a subcarrier 1 and a delayed subcarrier 2 are shown. When an OFDM receiver tries to demodulate the first subcarrier, it will encounter some interference from the second subcarrier, because within the FFT interval, there is no integer number of cycle differences between subcarrier 1 and 2. At the same time, there will be crosstalk from subcarrier 1 for the same reason.

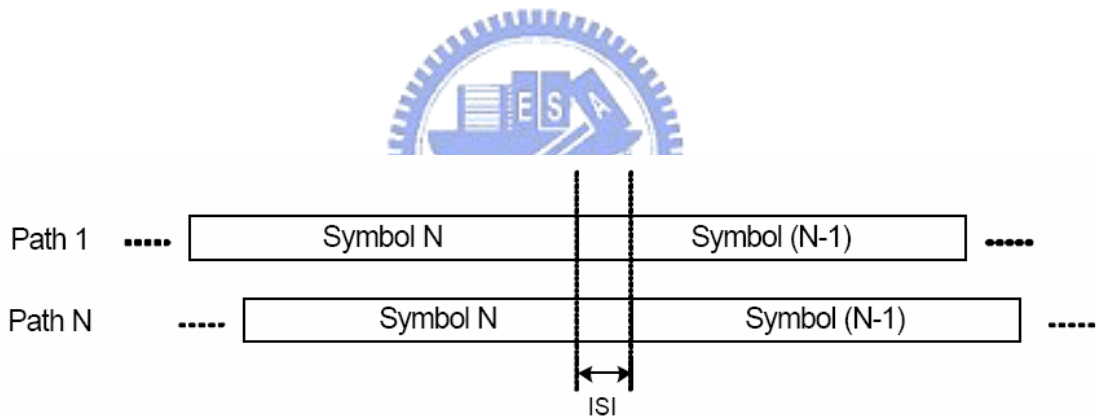


Figure 2.9 Channel dispersion causes ISI between successive OFDM signals [26].

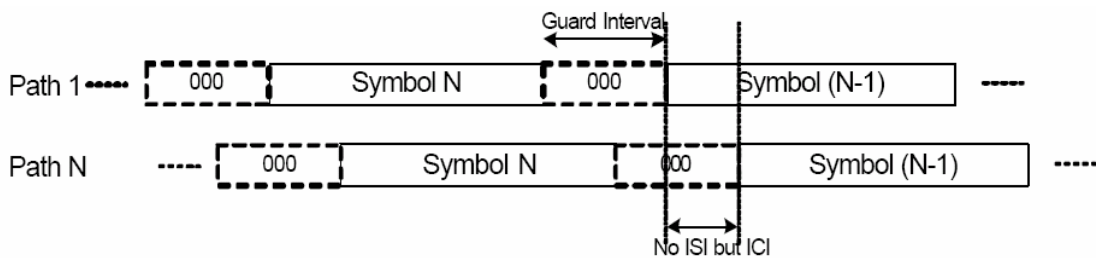


Figure 2.10 OFDM signals with silent GI [26].

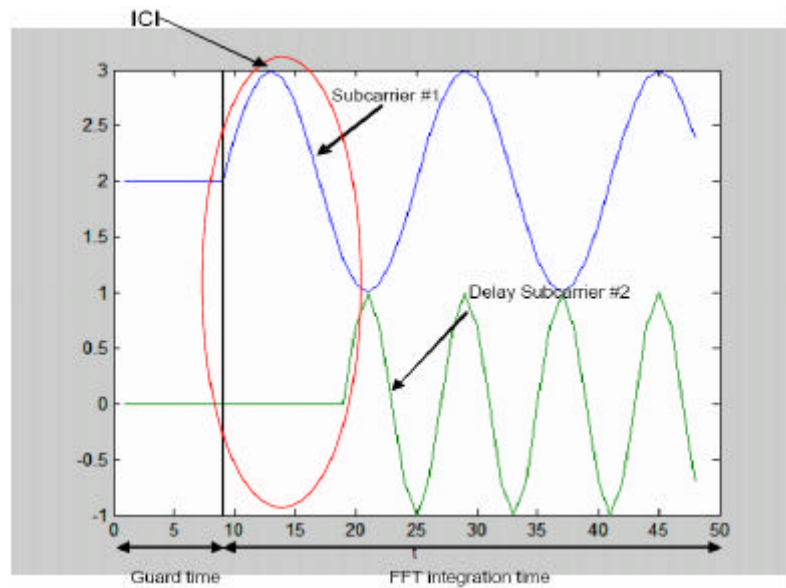


Figure 2.11 A delayed OFDM signal with a silent GI caused ICI on next signal [25].

In 1980, Peled and Ruiz [4] solved the ICI problem with the introduction of a cyclic prefix (CP), a copy of the last part of the OFDM signal attached to the front of the transmitted signal as shown in **Figure 2.12**. This ensures that delayed replicas of the OFDM symbol always have an integer number of cycles within the FFT interval, as long as the delay is smaller than the GI. As a result, multipath signals with delays smaller than the GI cannot cause ICI.

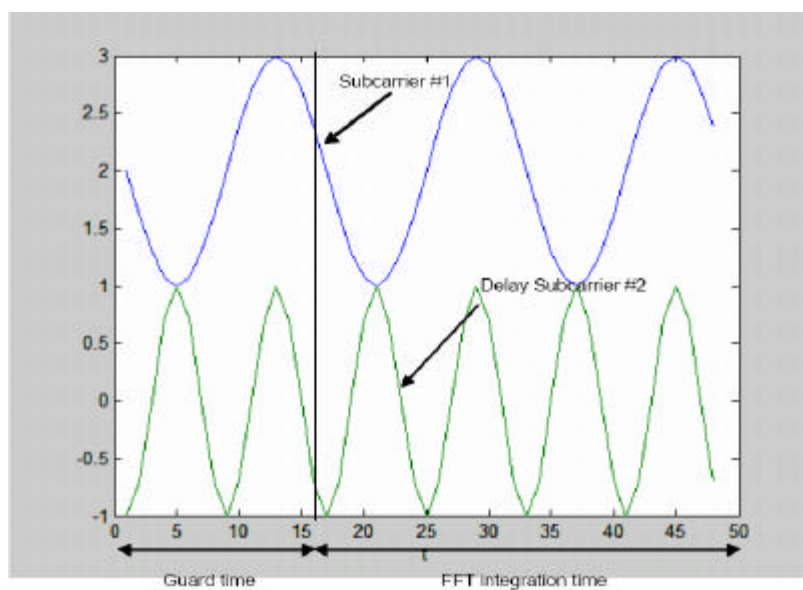


Figure 2.12 OFDM signal with cyclic prefix [25].

As an example of how multipath affects OFDM, **Figure 2.13** shows received signals for a two-ray channel, where the dotted curve is a delayed replica of the solid curve. Three separate subcarriers are shown during three symbol intervals. From the figure, we can see that the OFDM subcarriers are BPSK modulated, which means that there can be 180-degree phase jumps at the symbol boundaries. In this particular example, this multipath delay is smaller than the GI, which means there are no phase transitions during the FFT interval. Hence, an OFDM receiver “sees” the sum of pure sine waves with some phase offsets. This summation does not destroy the orthogonality between the subcarriers, it only introduces a different phase shift for each subcarrier. The orthogonality does become lost if the multipath delay becomes larger than the GI. In that case, the phase transitions of the delayed path fall within the FFT interval of the receiver.

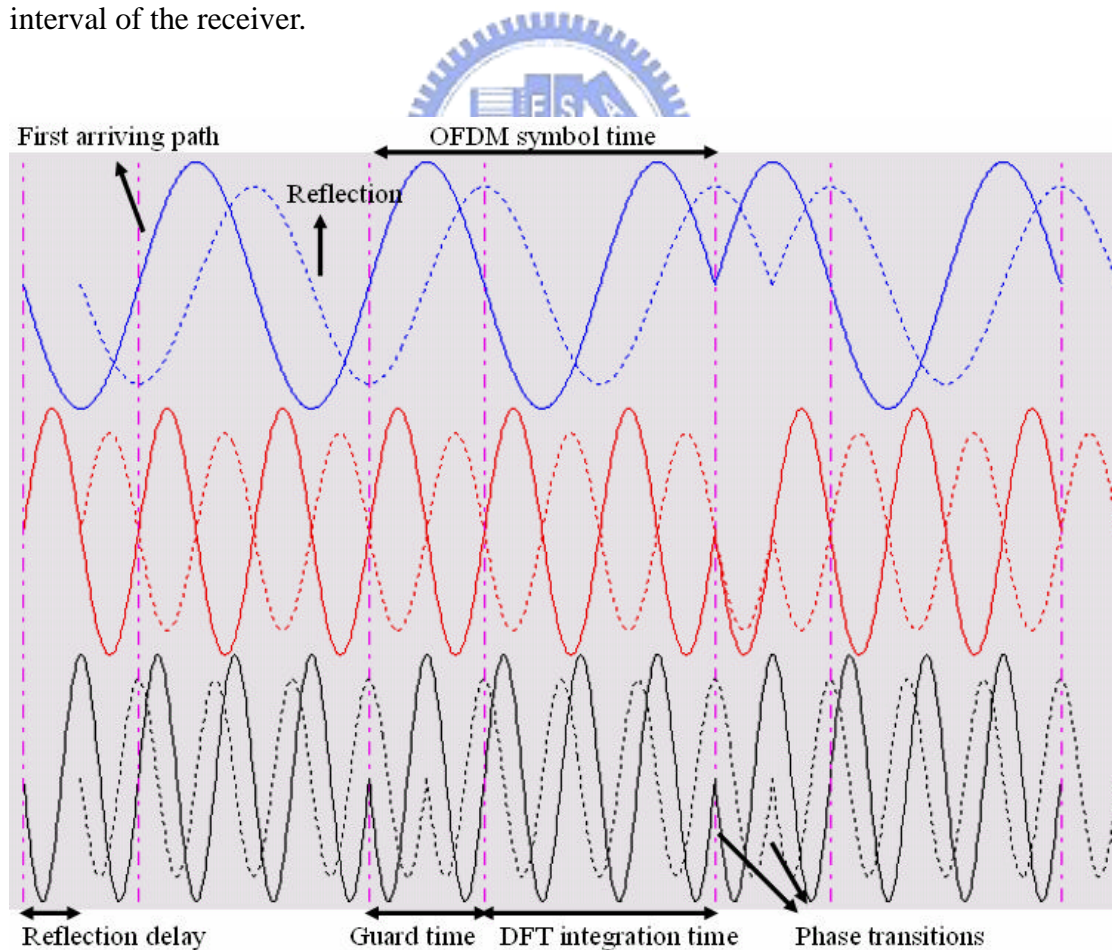


Figure 2.13 Example of an OFDM signal with three subcarriers in a two-ray multipath channel. The dashed line represents a delayed multipath component [18].

Figure 2.14 shows the digital implementation of the OFDM transmitter which appends the CP in the front of the original OFDM signal $s[n]$, and the structure of the OFDM signal with the guard period L is shown in Figure 2.15. We denote the signal with CP as the complete OFDM signal $\tilde{s}[n]$ and call the original $s[n]$ with length of N as the useful part of $\tilde{s}[n]$. The $\tilde{s}[n]$ can be expressed as

$$\tilde{s}[n] = \begin{cases} \frac{1}{\sqrt{N}} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} x_k e^{j2\pi \frac{k}{N}(n-L)} & 0 \leq n \leq N + L - 1 \\ 0 & \text{otherwise} \end{cases} \quad (2.8)$$

In the receiver, we only require the useful part of the complete OFDM signal to perform the FFT demodulation. Hence, we will remove the CP of the complete OFDM signal before the FFT demodulation in the receiver.

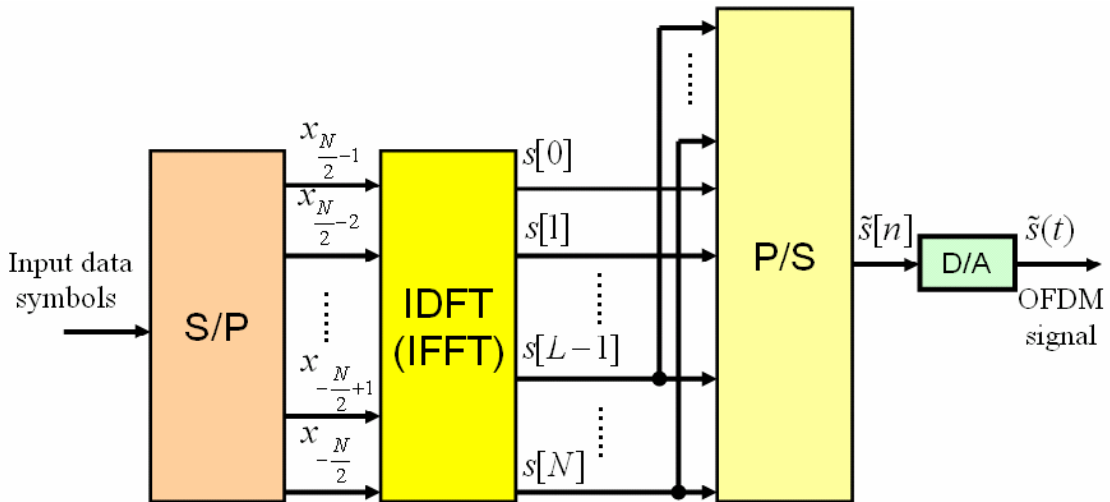


Figure 2.14 A digital implementation of appending CP into OFDM signal in transmitter [22].

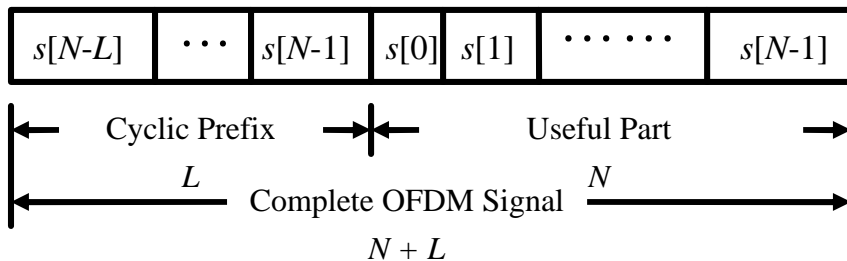


Figure 2.15 Structure of a complete OFDM signal with CP [22].

2.2.2 Linear Convolution Equivalent

In addition to avoid the ICI and ISI introduced by channel dispersion, the CP used in the OFDM system has another special purpose. As $\tilde{s}[n]$ is transmitted through the channel, the received complete OFDM signal $\tilde{r}[n]$ is the linear convolution of $\tilde{s}[n]$ and the impulse response of the channel:

$$\tilde{r}[n] = \tilde{s}[n] * h[n] \quad 0 \leq n \leq N + L + L_h - 2 \quad (2.9)$$

where $*$ denotes the linear convolution and $h[n]$ denotes the impulse response of the channel with the length L_h . We assume that L_h is smaller than the guard period L here. As mentioned, the CP is the last part of the original OFDM signal $s[n]$, so the result of the linear convolution described in (2.9) for $n = L, \dots, L + N - 1$ is the N -point circular convolution of $s[n]$ and $h[n]$ given by

$$r[n] = s[n] \otimes_N h[n] \quad (2.10)$$

for $n = 0, \dots, N - 1$, where \otimes_N denotes the N -point circular convolution. **Figure 2.16** shows the relation between (2.9) and (2.10), respectively. After removing the CP in the receiver, we use the DFT demodulation to recover the subsymbols in the received OFDM signal $r[n]$. According to the discrete time linear system theory [19], the DFT of $r[n]$ in (2.10) is equivalent to multiplying the frequency response of the OFDM signal $s[n]$ with that of the channel $h[n]$, and the result is given by

$$y_j = \text{DFT}(r[n]) = \text{DFT}(s[n] \otimes_N h[n]) = x_j H[j] \quad (2.11)$$

for $j = 0, \dots, N - 1$, where $H[j]$ is the frequency response of the channel at the frequency of the j -th subcarrier. Notes that $H[j]$ in (2.11) is the DFT of channel impulse response $h[n]$. From (2.11), we see that the demodulation result at the j -th subcarrier is the product of the original data subsymbol x_j and the frequency response of the channel at the same frequency, $H[j]$. This property states that the receiver in the OFDM system does not require the complex adaptive channel equalization technique used in conventional single carrier systems. In the OFDM systems, the

subsymbol x_j can be recovered in the receiver by dividing the demodulation result y_j by simply the weight equal to $H[j]$. This is the reason why the CP is a copy of the last part of the original rather than a copy of any part of the signal.

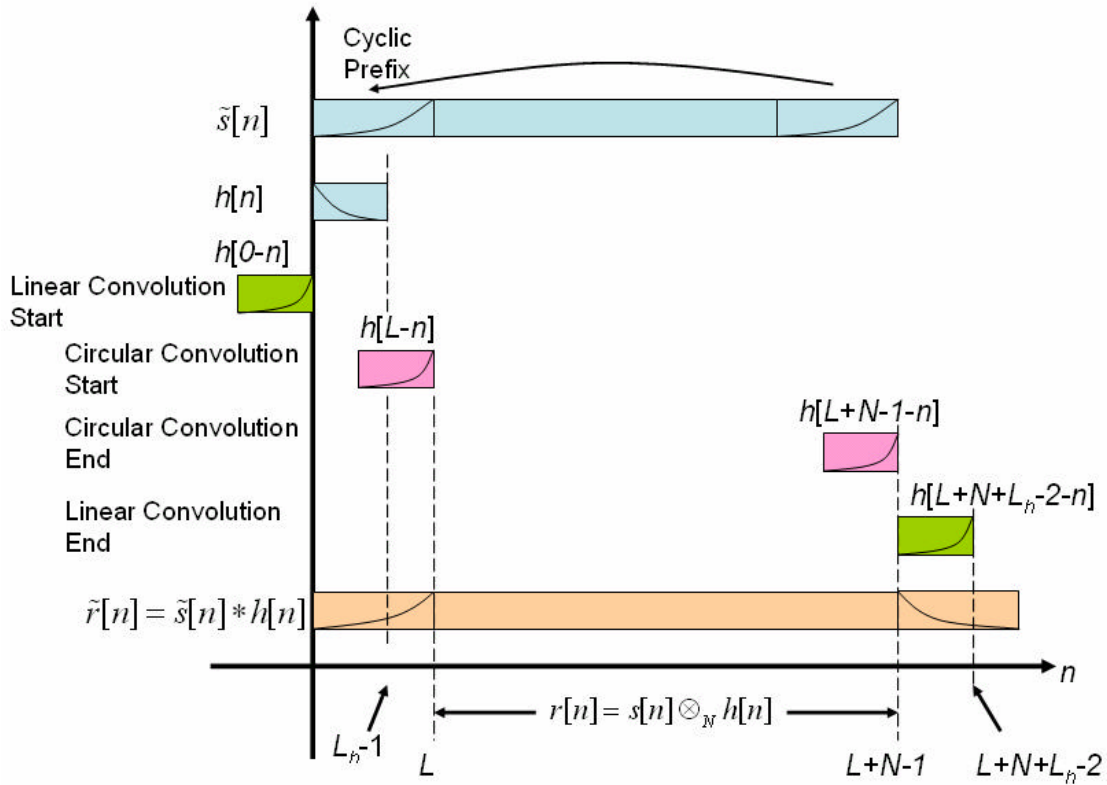


Figure 2.16 Relation between linear convolution and circular convolution of an OFDM signal and channel impulse response [22].

2.3 Windowing

Looking at an example OFDM signal like in **Figure 2.13**, sharp phase transitions caused by the modulation can be seen at the symbol boundaries. Essentially, an OFDM signal like the one depicted in **Figure 2.13** consists of a number of unfiltered QAM subcarriers. As a result, the out-of-band spectrum decreases rather slowly, according to a sinc function. As an example of this, the spectra for 16, 64, and 256 subcarriers are plotted in **Figure 2.17**. For larger number of subcarriers, the spectrum goes down more rapidly in the beginning, which is caused by the fact that

the sidelobes are closer together. However, even the spectrum for 256 subcarriers has a relatively small -40 dB bandwidth that is almost four times the -3 dB bandwidth.

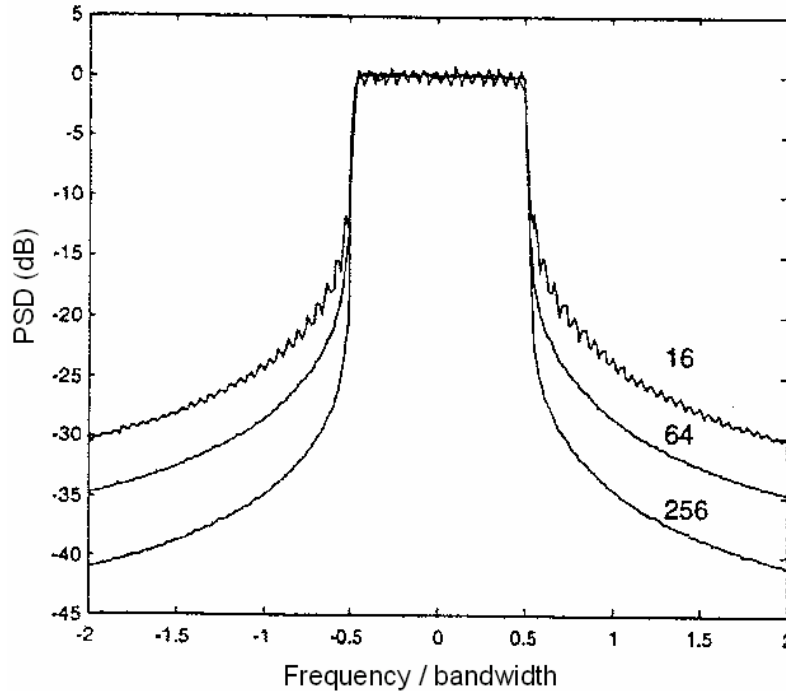


Figure 2.17 Power spectral density (PSD) without windowing for 16, 64, and 256 subcarriers [18].

2.3.1 Common Used Window Type

To make the spectrum go down more rapidly, windowing can be applied to the individual OFDM symbols. Windowing an OFDM symbol makes the amplitude go smoothly to zero at the symbol boundaries. A commonly used window type is the raised cosine window, which is defined as

$$w(t) = \begin{cases} 0.5 + 0.5 \cos(\mathbf{p} + \mathbf{p}t / \mathbf{b}T_s) & 0 \leq t \leq \mathbf{b}T_s \\ 1.0 & \mathbf{b}T_s \leq t \leq T_s \\ 0.5 + 0.5 \cos(\mathbf{p}(t - T_s) / \mathbf{b}T_s) & T_s \leq t \leq (1 + \mathbf{b})T_s \end{cases} \quad (2.12)$$

where \mathbf{b} is roll-off factor and T_s is symbol interval. The factor \mathbf{b} means that we allow adjacent OFDM symbols to partially overlap in the roll-off region and the transition between the consecutive symbol intervals are smoothed. The time

structure of the OFDM signal now looks like **Figure 2.18**.

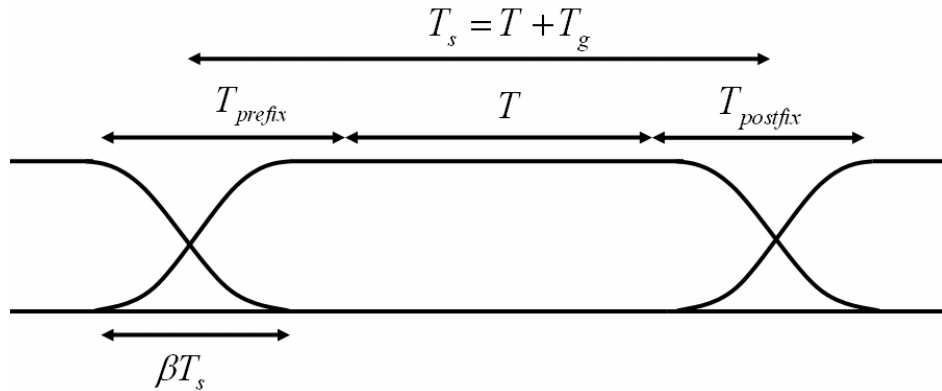


Figure 2.18 OFDM cyclic extension and raised cosine windowing. T_s is the symbol time, T the FFT interval, T_g the guard time, T_{prefix} the preguard interval, $T_{postfix}$ the postguard interval, and β is the roll-off factor [18].

In practice, the OFDM signal is generated as follows: first, N_c input QAM values are padded with zeros to get N input samples that are used to calculate an IFFT. Then, the last T_{prefix} samples of the IFFT output are inserted at the start of the OFDM symbol, and the first $T_{postfix}$ samples are appended at the end. The OFDM symbol is then multiplied by a raised cosine window $w(t)$ to more quickly reduce the power of out-of-band subcarriers. The OFDM symbol is then added to the output of the previous OFDM symbol with a delay of T_s , such that there is an overlap region of βT_s .

2.3.2 Choice of Roll-Off Factor

Figure 2.19 shows spectra for 64 subcarriers and different values of the roll-off factor β . It can be seen that a roll-off of 0.025 - so the roll-off region is only 2.5% of the symbol interval - already makes a large improvement in the out-of-band spectrum. Larger β improve the spectrum further, at the cost, however, of a decreased delay spread tolerance. The latter effect is demonstrated in **Figure 2.20**, which shows the signal structure of an OFDM signal for a two-ray multipath channel.

The receiver demodulates the subcarriers between the dotted lines. Although the relative delay between the two multipath signals is smaller than the GI, ICI and ISI are introduced because of the amplitude modulation in the gray part of the delayed OFDM symbol. The orthogonality between subcarriers holds when amplitude and phase of the subcarriers are constant during the entire T -second interval. Hence, a roll-off factor of b reduces the effective GI by $b T_s$.

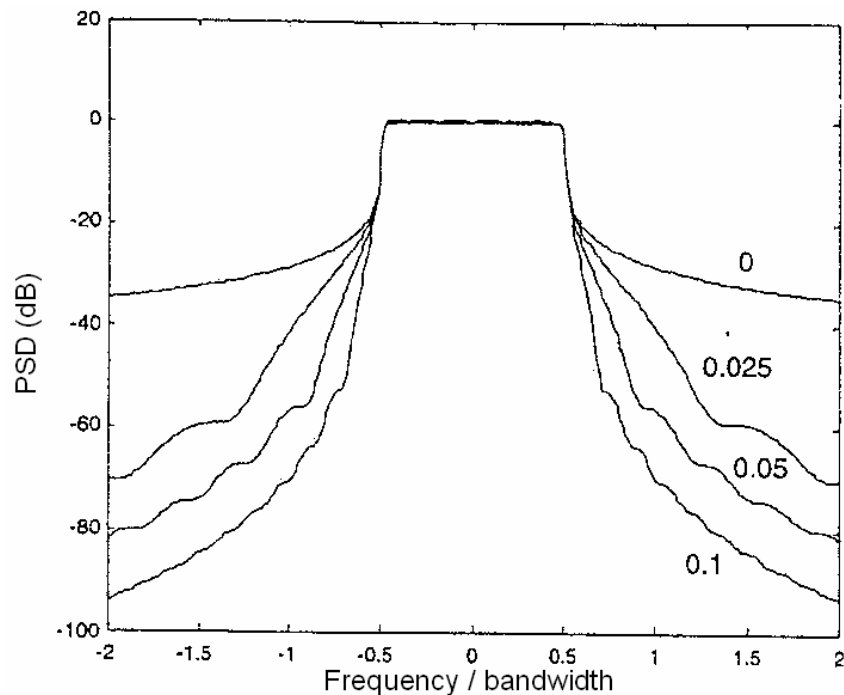


Figure 2.19 Spectral of raised cosine windowing with roll-off factor of 0 (rectangular window), 0.025, 0.05, and 0.1 [18].

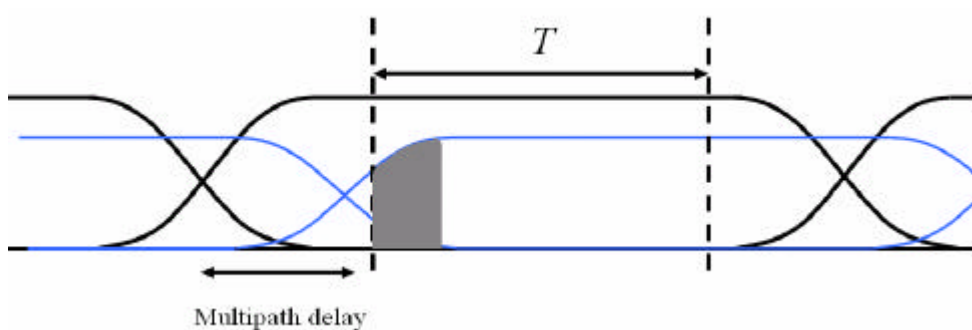
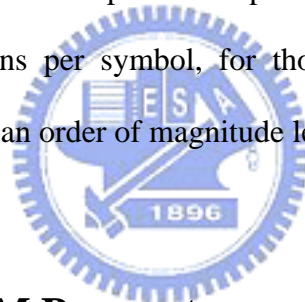


Figure 2.20 OFDM symbol windows for a two-ray multipath channel, showing ICI and ISI, because in the gray part, the amplitude of the delayed subcarrier is not constant [18].

2.3.3 Decision between Windowing and Filtering

Instead of windowing, it is also possible to use conventional filtering techniques to reduce the out-of-band spectrum. Windowing and filtering are dual techniques, multiplying an OFDM signal by a window means the spectrum is going to be a convolution of the spectrum of the window function with a set of impulse at the subcarrier frequencies. When using filters, care has to be taken not to introduce ripples on the envelope of the OFDM symbols over a timespan that is larger than the roll-off region of the windowing approach. Too much rippling means the undistorted part of the OFDM envelope is smaller, and this directly translates into less delay spread tolerance. The windowing technique is more feasible because a digital filter requires at least a few multiplications per sample, while windowing only requires a few multiplications per symbol, for those samples which fall into the roll-off region, windowing is an order of magnitude less complex than digital filtering.



2.4 Choice of OFDM Parameters

The choice of OFDM parameters is a tradeoff between various, often conflicting requirements. Usually, there are three main requirements to start with: bandwidth, bit rate, and delay spread. Generally, the following condition should be satisfied:

$$\mathbf{t} \leq \frac{N}{B} \leq \frac{1}{f_d} \quad (2.13)$$

where \mathbf{t} is the channel r.m.s delay spread, f_d is the maximum Doppler frequency spread, N is the number of subcarriers, and B is the total occupied bandwidth in the system. Therefore, the procedures of selecting system parameters are described in this section.

2.4.1 Guard Time and Symbol Duration

The delay spread directly dictates the time duration of the GI since the CP length must exceed the maximum delay spread. As a rule, the guard time T_g should be about two to four times the r.m.s delay spread of the channel t . T_g also depends on the type of coding and QAM modulation. Higher order QAM (like 64-QAM) is more sensitive to ICI and ISI than QPSK, while heavier coding obviously reduces the sensitivity to such interference.

Now the guard time has been set, the symbol duration T_s can be fixed. To minimize the SNR loss caused by the GI, it is desirable to have T_s much larger than the guard time. It cannot be arbitrarily large because a larger T_s means more subcarriers with a smaller subcarrier spacing, a larger implementation complexity, and more phase noise and frequency offset, as well as an increased peak-to-average power ratio (PAPR). Hence, a practical design choice is to make the symbol duration at least five times the guard time, which implies a 1-dB SNR loss because of the GI.

2.4.2 Number of Subcarriers

After the symbol duration and guard time are fixed, the number of subcarriers N follows directly as the required -3dB bandwidth B divided by the subcarrier spacing Δf , which is the inverse of the symbol duration less the guard time T .

$$N = \frac{B}{\Delta f} = B * \frac{1}{T} \quad (2.14)$$

Alternatively, N may be determined by the required bit rate divided by the bit rate per subcarrier. The bit rate per subcarrier is defined by the modulation type, coding rate, and symbol rate.

2.4.3 A System Design Example

As an example, suppose we want to design a system with the following requirements: the bit rate is 20 Mbps, the tolerable delay spread is 200 ns, and the maximum bandwidth is 15 MHz operating at $f_c = 5$ GHz.

The delay spread tolerance 200 ns suggests that 800 ns is a safe value for the guard time. By choosing the OFDM symbol duration 6 times the guard time (4.8 μ s), the guard time loss is made smaller than 1 dB. The subcarrier spacing is now the inverse of $4.8 - 0.8 = 4 \mu$ s, which gives 250 kHz. To determine the number of subcarriers needed, we can look at the ratio of the required bit rate and the OFDM symbol rate. To achieve 20 Mbps, each OFDM symbol has to carry 96 bits of information ($96/4.8 \mu$ s = 20 Mbps). To do this, there are several options. One is to use 16-QAM together with rate 1/2 coding to get 2 bits per symbol per subcarrier. In this case, 48 carriers are needed to get the required 96 bits per symbol. Another option is to use QPSK with rate 3/4 coding, which gives 1.5 bits per symbol per subcarrier. However, 64 subcarriers means a bandwidth of $64 \cdot 250$ kHz = 16 MHz, which is larger than the target bandwidth. To achieve a bandwidth smaller than 15 MHz, the first option with 48 subcarriers and 16-QAM fulfills all the requirements. It has the additional advantage that an efficient 64-point radix-4 FFT/IFFT can be used, leaving 16 zero subcarriers to provide oversampling necessary to avoid aliasing. If we assume that the moving speed of the mobile v is no more than 100 km/hr, (2.13) is

$$\text{satisfied. } (200 \cdot 10^{-9} \ll \frac{64}{20 \cdot 10^6} \ll \frac{1}{f_d} = \frac{c}{v} \cdot \frac{1}{f_c} = \frac{3 \cdot 10^8}{\frac{100 \cdot 10^3}{3600} \cdot 5 \cdot 10^9})$$

$$\Rightarrow 2 \cdot 10^{-11} \ll 3.2 \cdot 10^{-6} \ll 2.16 \cdot 10^{-3}, \text{ where } c \text{ is the velocity of light.})$$

Chapter 3

Frame Synchronization Techniques

This chapter is organized as follows. We will introduce the OFDM system model and the synchronization task in *Section 3.1*. Several typical CP-based frame synchronization techniques are described in *Section 3.2* and *3.3*, and our proposed modified techniques are presented in *Section 3.4*.

3.1 OFDM System Model

3.1.1 System Description

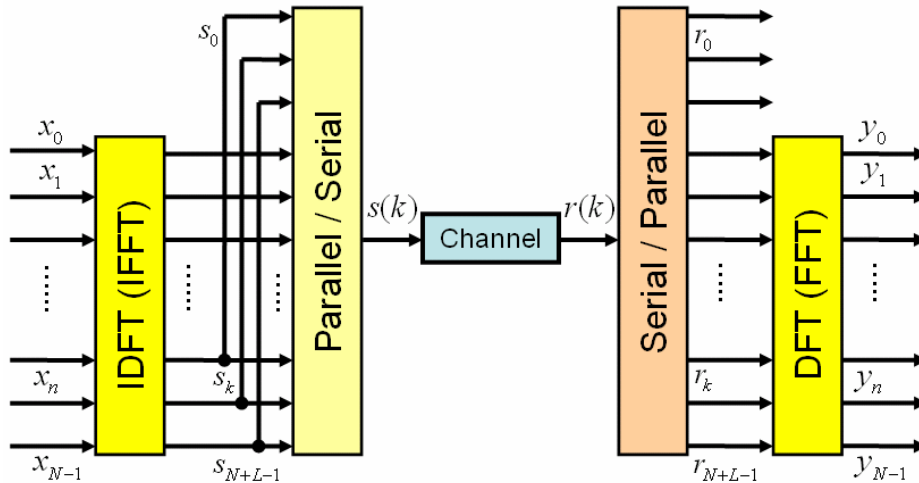


Figure 3.1 OFDM system, transmitting subsequent blocks of N complex data.

Figure 3.1 illustrates the baseband, discrete-time OFDM system model we investigate. The complex data subsymbols are modulated by means of an IDFT (IFFT) on N parallel subcarriers. The resulting OFDM symbol is serially transmitted

over a discrete-time channel, whose impulse response we assume is shorter than L samples. At the receiver, the data are retrieved by means of DFT (FFT).

An accepted means of avoiding ISI and preserving orthogonality between subcarriers is to copy the last L samples of the body of the OFDM symbol – the cyclic prefix (CP) – to form the complete OFDM symbol, as mentioned in **Subsection 2.2.1**. The effective length of the OFDM symbol as transmitted is this CP plus the body ($L+N$ samples long). The insertion of CP can be shown to result in an equivalent parallel orthogonal channel structure that allows for simple channel estimation and equalization, as mentioned in **Subsection 2.2.2**. In spite of the loss of transmission power and bandwidth associated with the CP, these properties generally motivate its use.

3.1.2 Synchronization Task

Consider two uncertainties in the receiver of the OFDM symbol: the uncertainty in the arrival time of the OFDM symbol and the uncertainty in carrier frequency. The first uncertainty, also called the frame error, is modeled as a delay in the channel impulse response $\mathbf{d}(k - \mathbf{q})$, where \mathbf{q} is the integer-valued unknown arrival time of a symbol. The latter is modeled as a complex multiplicative distortion of the received data in the time domain $e^{j2\mathbf{p}ek/N}$, where \mathbf{e} denotes the difference in the transmitter and receiver oscillators as a fraction of the subcarrier spacing. Notice that all subcarriers experience the same shift \mathbf{e} . These two uncertainties and the AWGN thus yield the received signal

$$r(k) = s(k - \mathbf{q})e^{j2\mathbf{p}ek/N} + n(k) \quad (3.1)$$

Two other synchronization parameters are not accounted here. First, an offset in the carrier phase may affect the symbol error rate in coherent modulation. If the data is differentially encoded, however, this effect is eliminated. An offset in the sampling

frequency will also affect the system performance. We assume that such an offset is negligible. The effect of non-synchronized sampling is investigated in [17].

Now, consider the transmitted signal $s(k)$. This is the IDFT of the data symbols x_k , which we assume are independent. Hence, $s(k)$ is a linear combination of independent and identically distributed (i.i.d) random variables. If the number of subcarriers is sufficiently large, we know from the central limit theorem that $s(k)$ approximates a complex Gaussian process whose real and imaginary parts are independent. This process, however, is not white since the appearance of a CP yields a correlation between some pairs of samples that are spaced N samples apart. Hence, $r(k)$ is not a white process either, but because of its probabilistic structure, it contains information about the time offset \mathbf{q} and carrier frequency offset \mathbf{e} . This is the crucial observation that offers the opportunity for joint estimation of these parameters based on $r(k)$.

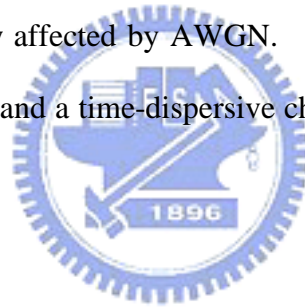


Next, we investigate the influence of the frame errors on the FFT output symbols while AWGN channel is used. If the estimated start position of the frame is located within the guard interval, each FFT output symbol within the frame will be rotated by a different angle. From subcarrier to subcarrier, the angle increases proportionally to the frequency offset. If the estimated start position of the frame locates within the data interval, the sampled OFDM frame will contain some samples that belong to other OFDM frame. Therefore, each symbol at the FFT output is rotated and dispersed due to the ISI from other OFDM frame. The phase rotation imposed by frame synchronization error can thus be corrected by appropriately rotating the received signal, but the dispersion of signal constellation caused by ISI forms a BER floor. Another effect that we must take into account is the channel impairment. The OFDM symbols are dispersed in time axis due to the multipath effect.

Consequently, the guard interval used to estimate the frame location is interfered by the previous symbol.

A synchronizer cannot distinguish between phase shifts introduced by the channel and those introduced by symbol time delays. Time error requirements may range from the order of one sample (wireless applications, where the channel phase is tracked and corrected by the channel equalizer) to a fraction of a sample (in, e.g., high bit-rate xDSL, where the channel is static and essentially estimated only during startup). The effect of a frequency offset is a loss of orthogonality between the tones. The resulting ICI has been investigated in [21].

In the following sections, we assume that the channel is non-dispersive and that the transmitted signal is only affected by AWGN. We will evaluate our techniques for both the AWGN channel and a time-dispersive channel by computer simulation in *Chapter 5*.



3.2 Correlation Frame Synchronization Techniques

3.2.1 ML Estimation Based on Received Signal [9]

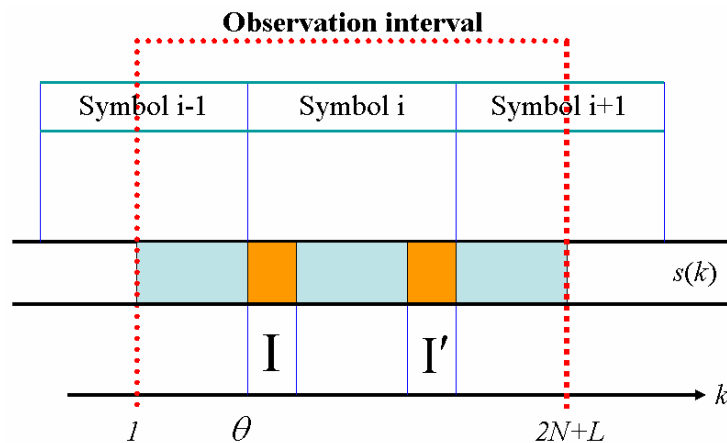


Figure 3.2 Structure of OFDM signal with CP symbols $s(k)$.

Assume that we observe $2N+L$ consecutive samples of $r(k)$, as shown in **Figure 3.2**, and that these samples contain one complete $(N+L)$ -sample OFDM symbol. The position of this symbol within the observed block of samples, however, is unknown because the channel delay \mathbf{q} is unknown to the receiver. Define the index sets

$$\begin{aligned} \mathbf{I} &\triangleq \{\mathbf{q}, \dots, \mathbf{q} + L - 1\} \quad \text{and} \\ \mathbf{I}' &\triangleq \{\mathbf{q} + N, \dots, \mathbf{q} + N + L - 1\} \end{aligned}$$

(see **Figure 3.2**). The set \mathbf{I}' thus contains the indices of the data samples that are copied into the CP, and the set \mathbf{I} contains the indices of this CP. Collect the observed samples in the $(2N+L) \times 1$ -vector $\hat{\mathbf{r}} \triangleq [r(1) \cdots r(2N+L)]^T$. Notice that the samples in the CP and their copies $r(k)$, $k \in \mathbf{I} \cup \mathbf{I}'$ are pairwise correlated, *i.e.*,

$$\forall k \in \mathbf{I}: E\{r(k)r^*(k+m)\} = \begin{cases} \mathbf{s}_s^2 + \mathbf{s}_n^2 & m = 0 \\ \mathbf{s}_s^2 e^{-j2\pi e} & m = N \\ 0 & \text{otherwise} \end{cases} \quad (3.2)$$

where $\mathbf{s}_s^2 = E\{|s(k)|^2\}$ and $\mathbf{s}_n^2 = E\{|n(k)|^2\}$. The remaining samples $r(k)$, $k \notin \mathbf{I} \cup \mathbf{I}'$ are mutually uncorrelated.

The log-likelihood function for \mathbf{q} and \mathbf{e} , $\Lambda(\mathbf{q}, \mathbf{e})$ is the logarithm of the probability density function (pdf) of the $2N+L$ observed samples in $\hat{\mathbf{r}}$ given the arrival time \mathbf{q} and the carrier frequency offset \mathbf{e} . In the following, we will drop all additive and positive multiplicative constants that show up in the expression of the log-likelihood function since they do not affect the maximizing argument. Moreover, we drop the conditioning on for notational clarity. Using the correlation properties of the observations $\hat{\mathbf{r}}$, the log-likelihood function can be written as

$$\begin{aligned} \Lambda(\mathbf{q}, \mathbf{e}) &= \log f(\hat{\mathbf{r}}|\mathbf{q}, \mathbf{e}) \\ &= \log \left(\prod_{k \in \mathbf{I}} f(r(k), r(k+N)) \prod_{k \notin \mathbf{I} \cup \mathbf{I}'} f(r(k)) \right) \\ &= \log \left(\prod_{k \in \mathbf{I}} \frac{f(r(k), r(k+N))}{f(r(k))f(r(k+N))} \prod_k f(r(k)) \right) \end{aligned} \quad (3.3)$$

where $f(\cdot)$ denotes the pdf of the variables in its argument. Notice that it is used

for both 1-D and 2-D distributions. The second product $\prod_k f(r(k))$ in (3.3) is independent of \mathbf{q} (since the product is over all k) and \mathbf{e} (since the density $f(r(k))$ is rotationally invariant). Since the ML estimation of \mathbf{q} and \mathbf{e} is the argument maximizing $\Lambda(\mathbf{q}, \mathbf{e})$, we may omit this factor. Under the assumption that \hat{r} is a jointly Gaussian vector, (3.3) is shown in the *Appendix A* to be

$$\Lambda(\mathbf{q}, \mathbf{e}) = |\mathbf{g}(\mathbf{q})| \cos(2p\mathbf{e} + \angle \mathbf{g}(\mathbf{q})) - \mathbf{r}\Phi(\mathbf{q}) \quad (3.4)$$

where \angle denotes the argument of a complex number

$$\mathbf{g}(\mathbf{q}) \triangleq \sum_{k=q-(L-1)}^q r(k)r^*(k+N), \quad (3.5)$$

$$\Phi(\mathbf{q}) = \frac{1}{2} \sum_{k=q-(L-1)}^q (|r(k)|^2 + |r(k+N)|^2) \quad (3.6)$$

$$\text{and } \mathbf{r} = \left| \frac{E\{r(k)r^*(k+N)\}}{\sqrt{E\{|r(k)|^2\}}E\{|r(k+N)|^2\}}} \right| = \frac{\mathbf{s}_s^2}{\mathbf{s}_s^2 + \mathbf{s}_n^2} = \frac{SNR}{SNR + 1} \quad (3.7)$$

is the magnitude of the correlation coefficient between $r(k)$ and $r(k+N)$, the asterisk $*$ indicates the conjugate of a complex value and $SNR = \mathbf{s}_s^2 / \mathbf{s}_n^2$. The first term in (3.4) is the weighted magnitude of $\mathbf{g}(\mathbf{q})$, which is a sum of L consecutive correlations between pairs of samples spaced samples apart. The weighting factor depends on the frequency offset. The term $\Phi(\mathbf{q})$ is an energy term, independent of the frequency offset \mathbf{e} . Notice that its contribution depends on the SNR (by the weighting-factor \mathbf{r}). The maximization of the log-likelihood function can be performed in two steps:

$$\max_{(\mathbf{q}, \mathbf{e})} \Lambda(\mathbf{q}, \mathbf{e}) = \max_{\mathbf{q}} \max_{\mathbf{e}} \Lambda(\mathbf{q}, \mathbf{e}) = \max_{\mathbf{q}} \Lambda(\mathbf{q}, \hat{\mathbf{e}}_{ML}(\mathbf{q})). \quad (3.8)$$

The maximum with respect to the frequency offset \mathbf{e} is obtained when the cosine term in (3.4) equals one. This yields the ML estimation of \mathbf{e}

$$\hat{\mathbf{e}}_{ML}(\mathbf{q}) = -\frac{1}{2p} \angle \mathbf{g}(\mathbf{q}) + n \quad (3.9)$$

where n is an integer. Notice that by the periodicity of the cosine function, several maxima are found. We assume that an acquisition, or rough estimate, of the

frequency offset has been performed and that $|\mathbf{e}| < 1/2$; thus, $n = 0$. Since $\cos(2p\hat{\mathbf{e}}_{ML}(\mathbf{q}) + \angle \mathbf{g}(\mathbf{q})) = 1$, the log-likelihood function of \mathbf{q} (which is the compressed log-likelihood function with respect to \mathbf{e}) becomes

$$\Lambda(\mathbf{q}, \hat{\mathbf{e}}_{ML}(\mathbf{q})) = |\mathbf{g}(\mathbf{q})| - r\Phi(\mathbf{q}) \tag{3.10}$$

and the joint ML estimator of \mathbf{q} and \mathbf{e} given $r(k)$ becomes

$$\hat{\mathbf{q}}_{ML} = \arg \max_{\mathbf{q}} \{|\mathbf{g}(\mathbf{q})| - r\Phi(\mathbf{q})\} \tag{3.11}$$

$$\hat{\mathbf{e}}_{ML} = -\frac{1}{2p} \angle \mathbf{g}(\hat{\mathbf{q}}_{ML}). \tag{3.12}$$

Notice that only two quantities affect the log-likelihood function (and thus the performance of the estimator): the number of the CP samples L and the correlation coefficient r given by the SNR. The former is known at the receiver, and the latter can be fixed. Basically, the quantity $\mathbf{g}(\mathbf{q})$ provides the estimates of \mathbf{q} and \mathbf{e} . The structure of the estimator in an OFDM receiver is shown in **Figure 3.3**.

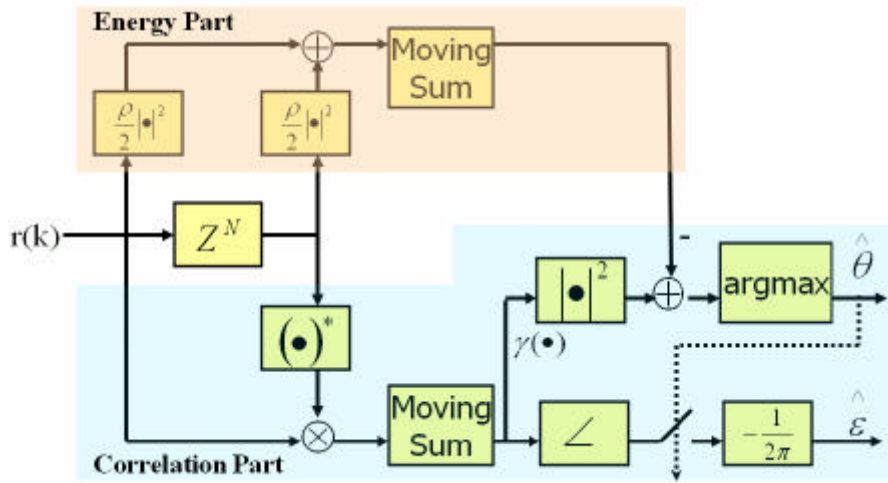


Figure 3.3 Structure of the ML estimator.

3.2.2 Peak-Picking Algorithm [9]

The peak-picking (*PP*) algorithm we introduce in this subsection is based on ML estimation describe in 3.2.1. We can see from (3.10), the first term (correlation part)

$|g(\mathbf{q})|$ dominates the log-likelihood function because the second term (energy part) is almost the same for different \mathbf{q} , then we can reformulate the ML estimator by a correlation function $G(n)$, which is given by

$$G(n) = \sum_{k=0}^{L-1} r(n-k) r^*(n-k-N). \quad (3.13)$$

The correlation function $G(n)$ is used for both frequency synchronization and frame timing synchronization. It represents the correlation of two sequences of L samples length, separated by N samples, in the received sample sequences as shown in **Figure 3.4**. The maximum magnitude sample of $G(n)$ is expected to coincide with the first sample of the current OFDM symbol. At this position, samples of CP and their copies in the current OFDM symbol are perfectly aligned in the summation window.

Therefore, the estimation $\hat{\mathbf{q}}_m$ of the frame timing for the m^{th} OFDM symbol can be given as

$$\hat{\mathbf{q}}_m = \arg \max_{\mathbf{q} \in \Theta} |G_m(\mathbf{q})| \quad (3.14)$$

The maximum value of the correlation function is found over a window of $\Theta = \{\mathbf{q} \mid 1 \leq \mathbf{q} \leq N + L\}$ for each OFDM symbol (window boundaries are not normally aligned with that of OFDM symbols) in the receiver. The estimation frequency error $\hat{\mathbf{e}}_m$ is estimated using the phase of the correlation function at $\mathbf{q} = \hat{\mathbf{q}}_m$,

$$\hat{\mathbf{e}}_m = -\frac{1}{2p} \angle G_m(\hat{\mathbf{q}}_m). \quad (3.15)$$

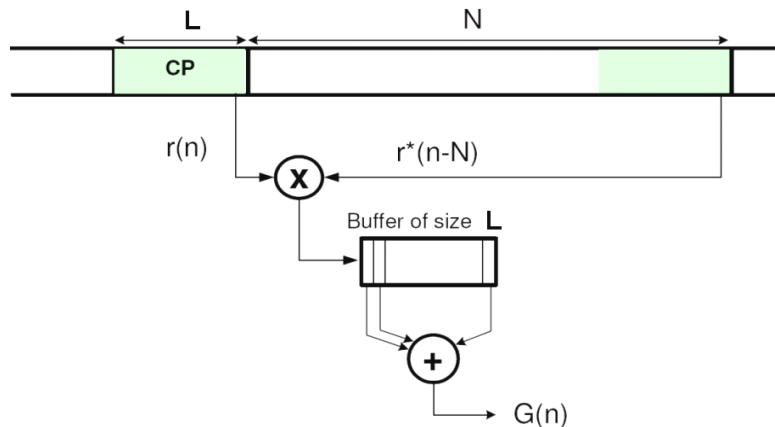


Figure 3.4 Computation of correlation function $G(n)$ using an L -length shift register.

3.2.3 Averaging and Peak-Picking Algorithm [11][15]

As suggested in [9], the accuracy of *PP* algorithm described in 3.2.2 can be improved by averaging $|G_m(\mathbf{q})|$ over several OFDM symbols.

$$G_{av}(\mathbf{q}) = \frac{1}{M} \sum_{m=1}^M |G_m(\mathbf{q})|, \quad \text{for } \mathbf{q} \in \Theta \quad (3.16)$$

where $G_{av}(\mathbf{q})$ is the correlation function averaged over M windows, each of size $L+N$, and $G_m(\mathbf{q})$ is the correlation function evaluated for the m^{th} window. The choice of M , the number of windows (symbols) to average over, in averaging and peak-picking (*APP*) algorithm mainly depends on the following two factors,

- (i) Time interval (number of OFDM symbols) over which the arrival time \mathbf{q} and frequency offset \mathbf{e} can be considered to be constant,
- (ii) Restriction on computational complexity.

The above factor (i) is tightly constrained in time fading channels because of the time variant channel delay \mathbf{q} and frequency offset \mathbf{e} . However, in a non-time fading channel (still with multipath and frequency selective fading), this constraint can be significantly relaxed, and \mathbf{q} and \mathbf{e} can be assumed to be constant over significantly long periods. Although this scenario allows large M values for averaging, the computational complexity becomes a major problem. We will introduce several low complexity solutions to this problem in the following sections.

3.3 Low-Complex Frame Synchronization Techniques

3.3.1 Complex-Quantization Algorithm [12]

In complex-quantization (*CQ*) algorithm, we quantize the in-phase and quadrature components of $r(k)$ to form the complex sequence $c(k)=Q[r(k)]$,

$k = 1, \dots, 2N + L$ where $Q[\cdot]$ denotes the complex quantizer

$$Q[x] \triangleq \text{sign}(\text{Re}\{x\}) + j \text{sign}(\text{Im}\{x\}), \quad (3.17)$$

$$\text{sign}(x) \triangleq \begin{cases} +1, & x \geq 0, \\ -1, & x < 0. \end{cases} \quad (3.18)$$

The signal $c(k)$ is a complex bitstream, *i.e.*, $c(k)$ can only take one of the four different values in the alphabet

$$\mathcal{A} = \{a_0, a_1, a_2, a_3\}, \{1 + j, -1 + j, -1 - j, 1 - j\}, \quad (3.19)$$

see **Figure 3.5**. The sequence $c(k)$ can thus be represented by 2 bits, one for its real and one for its imaginary part. In spite of this quantization, $c(k)$ still contains information about \mathbf{q} . A sample $c(k)$, $k \in I$, is correlated with $c(k + N)$, while all samples $c(k)$, $k \notin I \cup I'$, are independent.

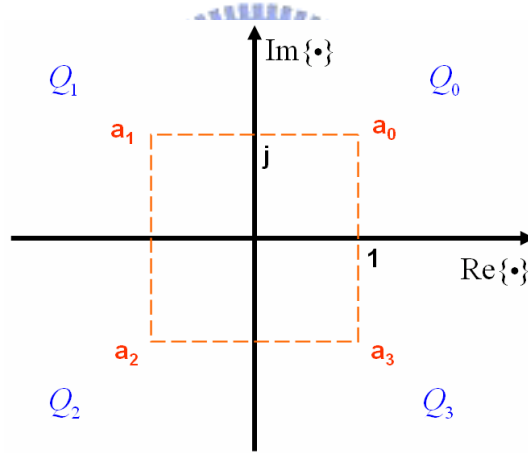


Figure 3.5 Geometric representation of the signal set \mathcal{A} , and the quadrants $Q_i, i = 0, 1, 2, 3$ of the complex plane.

The probability of all $2N + L$ samples of $c(k)$ to be observed simultaneously, given a certain value of \mathbf{q} , can be separated in the marginal probabilities for its sample to be observed, except for those samples $c(k)$, $k \in I \cup I'$, which are pairwise correlated. Denote the joint pdf for $c(k)$ and $c(k - N)$, $k \in I$, by $p_1(\cdot)$, and the pdf for $c(k)$, $k \notin I \cup I'$, by $p_2(\cdot)$. Then, the log-likelihood function of \mathbf{q} given $c(k)$ becomes

$$\Lambda_c(\mathbf{q}) = \log p_{\mathbf{q}}(c) = \log \left\{ \prod_{k \in I} p_1(c(k), c(k - N)) \cdot \prod_{k \notin I \cup I'} p_2(c(k)) \right\}. \quad (3.20)$$

The ML estimator of \mathbf{q} given $c(k)$, $\hat{\mathbf{q}}_c$, maximizes this function with respect to \mathbf{q} . For $k \notin I \cup I'$, $p_2(c(k)) = 1/4$, since $r(k)$ is zero-mean Gaussian process with independent real and imaginary parts. Hence, the second product of (3.20) is a constant, which can be omitted. The ML estimate $\hat{\mathbf{q}}_c$ becomes

$$\begin{aligned} \hat{\mathbf{q}}_c &= \arg \max_{\mathbf{q}} \Lambda_c(\mathbf{q}) \\ &= \arg \max_{\mathbf{q}} \sum_{k \in I} \log p_1(c(k), c(k-N)) \\ &= \arg \max_{\mathbf{q}} \sum_{k=q-L+1}^q \log p_1(c(k), c(k-N)) \\ &= \arg \max_{\mathbf{q}} (g * h)(\mathbf{q}), \end{aligned} \quad (3.21)$$

$$\text{where } g(k) = \log p_1(c(k), c(k-N)), \quad (3.22)$$

$$h(k) = \begin{cases} 1, & 0 \leq k \leq L-1, \\ 0, & \text{otherwise,} \end{cases} \quad (3.23)$$

and $*$ denotes convolution. To obtain the log-likelihood function we thus feed the resulting sequence by means of a moving sum of length L , see **Figure 3.6**. The ML estimation of \mathbf{q} selects the peaks of this function.

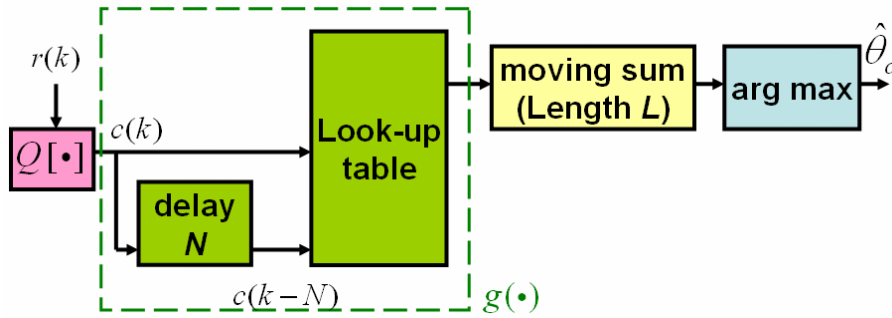


Figure 3.6 Look-up table implementation of the complex-quantization ML estimator.

In the **Appendix B**, the complex-quantization ML estimator based on $c(k)$ is determined by calculating the $p_1(\cdot)$. Moreover, it is shown that taking the real part of the correlation between $c(k)$ and $c(k-N)$, instead of applying the non-linearity $g(k)$ yields an equivalent and attractive structure for the ML estimator, as illustrated in

Figure 3.7. The ML estimate \hat{q}_c becomes

$$\hat{q}_c = \arg \max_q G_c(\mathbf{q}), \quad (3.24)$$

where

$$G_c(n) = \sum_{k=0}^{L-1} \text{Re}\{c(n-k) c^*(n-k-N)\}. \quad (3.25)$$

In most applications the arrival time \mathbf{q} is approximately constant over several, say M , received frames. This essentially means that instead of just one frame $r(k)$, M frames are observed simultaneously containing information about the unknown \mathbf{q} . Generalizing the discussion preceding, it can be shown that the log-likelihood function for \mathbf{q} given $c_i(k)$, $i=1, \dots, M$, becomes

$$\Lambda_a(\mathbf{q}) \sim \sum_{i=1}^M \Lambda_c^i(\mathbf{q}) \quad (3.26)$$

where $\Lambda_c^i(\mathbf{q})$ represents the log-likelihood function (3.20) of \mathbf{q} given frame $c_i(k)$. The ML estimate \hat{q}_a given $c_i(k)$, $i=1, \dots, M$, is the argument maximizing (3.26).

We call this method averaging and complex-quantization (ACQ) algorithm.

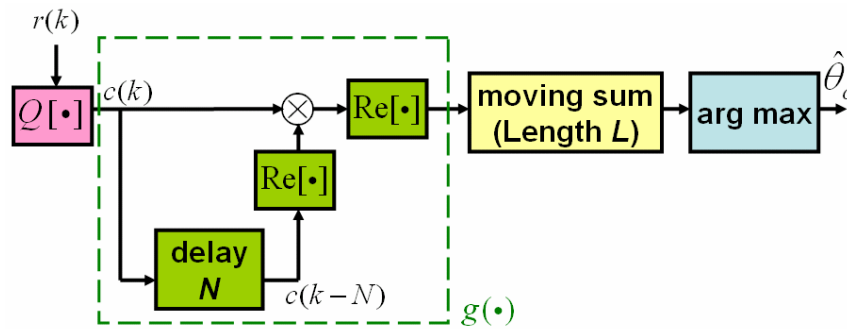


Figure 3.7 Equivalent implementation of the complex-quantization ML estimator.

3.3.2 Smoothing Complex-Quantization Algorithms [13]

As mentioned in *Subsection 3.1.2*, if the estimated start position of the frame locates within the data interval or multipath effect exists, the sampled OFDM frame

will contain some samples that belong to other OFDM frame. Therefore, each symbol at the FFT output is rotated and dispersed due to the ISI from other frame.

A solution to remedy this problem is to use different smoothing algorithms in place of the moving sum scheme shown in **Figure 3.7**. Instead of using the moving sum shown in **Figure 3.7** that weights $\text{Re}\{c(k) \cdot c^*(k-N)\}$ equally, an exponentially decaying weighted function is applied to $\text{Re}\{c(k) \cdot c^*(k-N)\}$. We consider four smoothing algorithms here. Let L be the number of samples in a guard interval, the log-likelihood functions at time instant \mathbf{q}_c for these algorithms are given as follows.

Moving average (**MA**):

$$\Lambda_c(\mathbf{q}_c) = \sum_{k=\mathbf{q}_c-L+1}^{\mathbf{q}_c} \text{Re}\{c(k) \cdot c^*(k-N)\}. \quad (3.27)$$

Shortened moving average (**SMA**):

$$\Lambda_c(\mathbf{q}_c) = \sum_{k=\mathbf{q}_c-L'+1}^{\mathbf{q}_c} \text{Re}\{c(k) \cdot c^*(k-N)\}, \text{ where } L' \leq L. \quad (3.28)$$

Exponentially weighted moving average (**EWMA**):

$$\Lambda_c(\mathbf{q}_c) = \sum_{k=\mathbf{q}_c-L+1}^{\mathbf{q}_c} w^{\mathbf{q}_c-k} \text{Re}\{c(k) \cdot c^*(k-N)\}. \quad (3.29)$$

Exponentially weighted average (**EWA**):

$$\Lambda_c(\mathbf{q}_c) = \sum_{k=0}^{\mathbf{q}_c} w^{\mathbf{q}_c-k} \text{Re}\{c(k) \cdot c^*(k-N)\}. \quad (3.30)$$

Note that the moving average (**MA**) scheme is identical to the **CQ** algorithm presented in 3.3.1. The **MA**, **SMA**, and **EWMA** algorithms can be realized as FIR filters, and the **EWA** algorithm can be realized as an IIR filter. The weighting factor w for both **EWMA** and **EWA** scheme can be intentionally chosen such that $w = 1 - 2^{-M}$, where M is a positive integer. By appropriately choosing the weighting factor, the multiplication operation within the summing scheme can be replaced by an adder and a shifter.

3.3.3 Global Search Algorithm [15]

A low complexity frame synchronization based on discrete stochastic approximation algorithms, which can be considered as the modification of the **PP** algorithm, will be described in this subsection. This technique avoids evaluating the correlation function $G(\mathbf{q})$ for all samples within a window in **PP** algorithm, thereby achieving a significant reduction in computational cost. It applies the idea of discrete stochastic optimization to reduce the number of complex multiplications while achieving synchronization. This iterative method is given below in detail. Here, after m iterations, \mathbf{q}_m is the current point, $W_m(\mathbf{q})$ for all $\mathbf{q} \in \Theta$ represents the number of times the synchronization algorithm has visited the point \mathbf{q} so far, and \mathbf{q}_m^* is the point that the algorithm has visited most often so far.

Algorithm: Global Search (**GS**) Algorithm

Step 0. Select a starting point $\mathbf{q}_0 \in \Theta$. Let $W_0(\mathbf{q}_0) = 1$ and $W_0(\mathbf{q}) = 0$ for all $\mathbf{q}_0 \in \Theta$, $\mathbf{q} \neq \mathbf{q}_0$. Let $m = 0$ and $\mathbf{q}_m^* = \mathbf{q}_0$. Go to Step 1.

Step 1. Given the value of \mathbf{q}_m , generate a uniform random variable \mathbf{q}'_m independently of the past so that for all $\mathbf{q}'_m \in \Theta$, $\mathbf{q} \neq \mathbf{q}_m$, we have $\mathbf{q}'_m = \mathbf{q}$ with probability $1/(|\Theta|-1)$. Go to Step 2.

Step 2. Given the values of \mathbf{q}_m and \mathbf{q}'_m , generate an observation $R_m = |G_m(\mathbf{q}'_m)| - |G_m(\mathbf{q}_m)|$ independently of the past. If $R_m > 0$, then let $\mathbf{q}_{m+1} = \mathbf{q}'_m$. Otherwise, let $\mathbf{q}_{m+1} = \mathbf{q}_m$. Go to Step 3.

Step 3. Let $m = m + 1$, $W_m(\mathbf{q}_m) = W_{m-1}(\mathbf{q}_m) + 1$, and $W_m(\mathbf{q}) = W_{m-1}(\mathbf{q})$ for all $\mathbf{q}_0 \in \Theta$, $\mathbf{q} \neq \mathbf{q}_m$. If $W_m(\mathbf{q}_m) > W_m(\mathbf{q}_{m-1}^*)$, then let $\mathbf{q}_m^* = \mathbf{q}_m$. Otherwise, let $\mathbf{q}_m^* = \mathbf{q}_{m-1}^*$. Go to Step 1.

The above algorithm resembles an adaptive filtering (LMS) algorithm in the sense that it generates a sequence of parameter values where each new parameter

value is obtained from the old one by moving in a good direction, and in the sense that it converges to the global optimizer of the objective function.

For the above procedure, iterations can be performed for $m = 1, 2, \dots, M_{GS}$, where M_{GS} is the number of OFDM symbols over which \mathbf{q} and \mathbf{e} can be assumed to be constant. In Step 0, the initial value of $\mathbf{q} = \mathbf{q}_0$ can be made equal to the peak position of $|G(\mathbf{q})|$ (obtained using a full search) for the first window. Note that in order to ensure real-time demodulation of OFDM symbols, estimation of \mathbf{q} and \mathbf{e} should be performed in real-time for each symbol. Accordingly, the estimation of \mathbf{q} and \mathbf{e} by the **GS** algorithm becomes,

$$\hat{\mathbf{q}}'_m = \mathbf{q}_m^*, \quad \text{for } m = 1, 2, \dots, M_{GS} \quad (3.31)$$

$$\hat{\mathbf{e}}'_m = -\frac{1}{2\mathbf{p}} \angle G_m(\mathbf{q}_m^*), \quad \text{for } m = 1, 2, \dots, M_{GS} \quad (3.32)$$

If **PP** algorithm is used, the correlation function $G(\mathbf{q})$ in (3.13) has to be evaluated for all point within the window of size $L+N$. According to (3.13), evaluating of $G(\mathbf{q})$ for a given \mathbf{q} involves L complex multiplications. A brute force computation, therefore, requires $L(L+N)$ complex multiplications to obtain the complete correlation functions for a window of $L+N$ sample points. However, when $G(\mathbf{q})$ is evaluated for consecutive points \mathbf{q} within the window $1 \leq \mathbf{q} \leq L+N$, it can be performed as depicted in **Figure 3.4** using a buffer of size L , thus reducing the per point cost to a one complex multiplication. Therefore, complexity C of **PP** algorithm becomes $L+N$ multiplications per symbol. In the **GS** algorithm, correlation function $G(\mathbf{q})$ has to be evaluated for only two points within a window of $1 \leq \mathbf{q} \leq L+N$. This is associated with the evaluation of $R_m = |G_m(\mathbf{q}'_m)| - |G_m(\mathbf{q}_m)|$ in Step 2 of the **GS** algorithm. However, since these two points can generally be far apart within the window, evaluation of $G(\mathbf{q})$ will cost L multiplications per point. Therefore, the complexity C' of the **GS** algorithm

becomes $2L$ multiplications per symbol. Thus, the percentage reduction of computational cost E_{GS} of the **GS** algorithm can be given as,

$$E_{GS} = \frac{C - C'}{C} \times 100\% = \left(\frac{1-r}{1+r}\right) \times 100\% \quad (3.33)$$

where $r = L/N$. However, it needs $L+N$ counters to maintain the values of $W_m(\mathbf{q})$.

3.4 Proposed Frame Synchronization Techniques

3.4.1 Local Search Algorithm

Consider the **GS** algorithm in 3.3.3, the accuracy of the **GS** algorithm heavily depends on the initial estimate, the reason is shown in 3.3.1. Assume that the estimate \mathbf{q}_m is incorrect but close the correct sample point. If we select the new sample point \mathbf{q}'_m , which is different from \mathbf{q}_m , in $[\mathbf{q}_m - R \quad \mathbf{q}_m + R]$, where the variable R is an integer depending on the search range we want, then the probability that the correct sample point is selected will higher than that in **GS** algorithm. Besides, $W_m(\mathbf{q}_m)$ will not tend to be a large value when the correct sample point is selected. Consequently, it takes a shorter time for the incorrect estimate to be replaced by the correct sample point than that in **GS** algorithm. This local search (**LS**) algorithm is given below in detail. Note that only the Step 1 of the **LS** algorithm is different from the **GS** algorithm.

Algorithm: Local Search (**LS**) Algorithm

Step 0. Select a starting point $\mathbf{q}_0 \in \Theta$. Let $W_0(\mathbf{q}_0) = 1$ and $W_0(\mathbf{q}) = 0$ for all $\mathbf{q}_0 \in \Theta, \mathbf{q} \neq \mathbf{q}_0$. Let $m = 0$ and $\mathbf{q}_m^* = \mathbf{q}_0$. Go to Step 1.

Step 1. Given the value of \mathbf{q}_m , generate a uniform random variable \mathbf{q}'_m independently of the past in $[\mathbf{q}_m - R \quad \mathbf{q}_m + R]$, where R is an integer, so that

for all $\mathbf{q}'_m \in [\mathbf{q}_m - R, \mathbf{q}_m + R]$, $\mathbf{q} \neq \mathbf{q}_m$, we have $\mathbf{q}'_m = \mathbf{q}$ with probability $1/2R$.
Go to Step 2.

Step 2. Given the values of \mathbf{q}_m and \mathbf{q}'_m , generate an observation $R_m = |G_m(\mathbf{q}'_m)| - |G_m(\mathbf{q}_m)|$ independently of the past. If $R_m > 0$, then let $\mathbf{q}_{m+1} = \mathbf{q}'_m$. Otherwise, let $\mathbf{q}_{m+1} = \mathbf{q}_m$. Go to Step 3.

Step 3. Let $m = m+1$, $W_m(\mathbf{q}_m) = W_{m-1}(\mathbf{q}_m) + 1$, and $W_m(\mathbf{q}) = W_{m-1}(\mathbf{q})$ for all $\mathbf{q}_0 \in \Theta$, $\mathbf{q} \neq \mathbf{q}_m$. If $W_m(\mathbf{q}_m) > W_m(\mathbf{q}_{m-1}^*)$, then let $\mathbf{q}_m^* = \mathbf{q}_m$. Otherwise, let $\mathbf{q}_m^* = \mathbf{q}_{m-1}^*$. Go to Step 1.

3.4.2 Modified Global Search Algorithm

As mentioned in 3.3.3, the *GS* algorithm, compared with the *PP* algorithm, saves up a lot of computation cost. Moreover, the accuracy of the *GS* algorithm heavily depends on the initial estimate. If the initial estimate is incorrect, then it is likely to take a long time to achieve accurate synchronization. This can be seen as follows. Assume that the initial estimate \mathbf{q}_0 is incorrect. Since a new sample point is selected randomly in each iteration, the probability that the correct sample point is selected is $1/(L+N-1)$, which implies the expected number of iterations to select the correct sample point is $L+N-1$. Besides, $W_m(\mathbf{q}_m)$ tends to be a large value when the correct sample point is selected. Consequently, it takes a long time for the incorrect estimate to be replaced by the correct sample point. This phenomenon may make the algorithm become unacceptable for time varying channels. In this subsection, we present a modified global search (*MGS*) algorithm that yields better performance than the *PP* algorithm and requires only slightly more computational cost.

The basic idea of *MGS* algorithm is to maintain a few good candidates, in addition to the current best estimate. A candidate becomes the best estimate if and

only if its correlation function is greater than or equal to that of the current best estimate at least K times in L windows. If more than one candidate satisfies the condition in the same iteration, then the one that has the largest correlation function in the current window becomes the best estimate. Let S denote the number of candidates maintained in the algorithm. To implement the K out of L algorithm, a register of L bits is associated to each maintained candidate. In addition to the current best estimate and the S maintained candidates, P other sample points are randomly selected at the beginning of each iteration. The correlation functions are evaluated and compared only for the randomly picked sample points, the current best estimate, and the maintained candidates. The steps for each iteration are described below.

Algorithm: Modified Global Search (*MGS*) Algorithm

Step 0. Select P sample points that are different from the current best estimate and the S maintained candidates. Go to Step 1.

Step 1. Evaluate the correlation functions for the current best estimate, maintained candidates, and the randomly picked sample points. Go to Step 2.

Step 2. Sort the correlation functions for the S maintained candidates and the P randomly picked sample points. Discard the P sample points with the P smallest correlation functions. Assume that R of the original S candidates are discarded. As a result, R of the P randomly picked sample points become new candidates. Reset and re-associate the registers that were associated to the discarded candidates to the new candidates. Go to Step 3.

Step 3. For each candidate, compare its correlation function with that of the current best estimate. Shift the associated register to the left by one bit. The bit inserted from the right is 1 if the correlation function of the candidate is larger than that of the current best estimate; otherwise, it is 0. Go to Step 4.

Step 4. Examine the shift register for each candidate. If there are K 1's, then the candidate becomes the new current best estimate and the original best estimate becomes a candidate. If there is more than one candidate satisfying the condition to become the best estimate, then the one with the largest correlation function wins the competition. Reset all shift registers associated to the candidates if a new best estimate is generated. Go to Step 0.

Note that in *MGS* algorithm, no counters are needed to store the number of visits to each sample point. Instead, the algorithm needs $S+1$ registers to record the current best estimate and the maintained candidates as well as S L -bit shift registers. The number of complex multiplications required in each iteration is given by $(P+S+1)L$, assuming that the evaluated sample points are separated so that each one requires L complex multiplications. For iteration 0, one evaluates correlation functions for all the $L+N$ sample points, chooses $S+1$ sample points with the largest $S+1$ correlation values, and sets the point that has the largest value to be the current best estimate and the others the candidates. It is clear that, after synchronization is lost, fast re-synchronization is possible because the exact synchronized point is a candidate with great probability. In the next chapter, we perform simulations for various values of P , S , K and L . Note that $K = L$ means a candidate becomes the best estimate if and only if its correlation function is greater than that of the best estimate for consecutively K (or L) times. For this simple realization, the shift register becomes either a flag or a two-bit counter and therefore, hardware cost is very low. The percentage reduction of computational cost E_{MGS} of the *MGS* algorithm, compared to the *PP* algorithm, can be given as,

$$E_{MGS} = \frac{(L+N) - (P+S+1)L}{L+N} \times 100\% = \left(\frac{1 - (P+S)r}{1+r} \right) \times 100\% \quad (3.34)$$

The percentage reduction E_{MGS} in (3.34) is smaller than E_{GS} in (3.33) because $P+S \geq 2$.

Chapter 4

Simulations and Performance Evaluation

This chapter is organized as follows. The fading channel model is described in *Section 4.1*. In *Section 4.2*, simulation results for several CP-based frame synchronization algorithms mentioned in *Chapter 3* are given, and the performance of our proposed modified techniques will be compared with other CP-based algorithms presented before.

4.1 Description of Simulation Model

In wireless communications, the fading channel can be classified as either frequency-nonselective or frequency-selective. In frequency-nonselective fading channels, the transmission path without line-of-sight (LOS) introduces an uncorrelated Rayleigh fading distortion to the corresponding transmitted signal for the worst case. The uncorrelated Rayleigh fading distortion is often modeled as a complex Gaussian fading process,

$$f(t) = f_I(t) + j f_Q(t) \quad (5.1)$$

where $f_I(t)$ and $f_Q(t)$ are two statistically independent real-valued baseband Gaussian waveforms. The PSD of $f(t)$ is given by

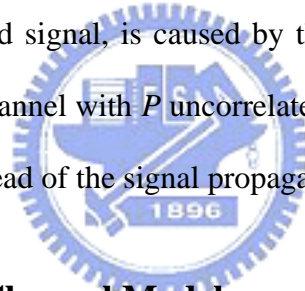
$$S(f) = \begin{cases} \frac{\mathbf{s}_f^2}{\mathbf{p} \sqrt{f_d^2 - f^2}}, & |f| \leq f_d \\ 0, & \text{elsewhere} \end{cases}, \quad (5.2)$$

where \mathbf{s}_f^2 is the average power of the faded carrier, and f_d denotes the maximum Doppler shift of the fading signal caused by the mobile motion. In general, the normalized maximum Doppler shift $f_d T$ is used as the measure for the fading rate of the channel variation. The envelope and phase of the fading process possess Rayleigh distribution and uniform distribution, respectively. The autocorrelation function of the fading process can be calculated as

$$R_f(\mathbf{t}) = \mathbf{s}_f^2 \cdot J_0(2\mathbf{p} f_d \mathbf{t}) \quad (5.3)$$

Jakes' Rayleigh fading channel model is often used to generate the fading signal in computer simulation for the frequency-nonselective fading channel.

In frequency-selective fading channel, the delay spread problem, which introduces ISI in the received signal, is caused by the channel dispersion. A P -ray multipath Rayleigh fading channel with P uncorrelated Rayleigh fading signals can be used to express the delay spread of the signal propagation paths. [5]



4.1.1 Rayleigh Fading Channel Model

In the Rayleigh fading channel without the LOS signal, the angle of the arrival signal in the horizontal plane is assumed to be uniformly distributed in the interval $[0, 2\mathbf{p})$. Jakes [21] modeled the Rayleigh fading channel by a bank of the oscillators with frequencies equal to the maximum Doppler frequency f_d and its fractions. The baseband simulator of Jakes' channel model is illustrated in **Figure 4.1**. It contains N_0 , which is must be large enough to satisfy the central limit theorem, low-frequency oscillators with the frequencies equal to the Doppler shifts $w_d \cos(2\mathbf{p}n/N)$, for $n=1, \dots, N_0$, plus one with the maximum Doppler frequency $w_d = 2\mathbf{p} f_d$. The amplitudes of all the components are equal to unity except that the one with maximum Doppler frequency is set to $1/\sqrt{2}$. The phases \mathbf{a} and \mathbf{b}_n are

chosen appropriately so that the pdf of the resultant phase will be very close to a uniform distribution over $[0, 2\mathbf{p})$. From Jakes' model, we have

$$f_I(t) = 2 \sum_{n=1}^{N_0} \cos \mathbf{b}_n \cdot \cos w_n t + \sqrt{2} \cos \mathbf{a} \cdot \cos w_d t, \quad (5.4)$$

$$f_Q(t) = 2 \sum_{n=1}^{N_0} \sin \mathbf{b}_n \cdot \cos w_n t + \sqrt{2} \sin \mathbf{a} \cdot \cos w_d t, \quad (5.5)$$

where $N = 2(2N_0 + 1)$, $N_0 \geq 8$,

$w_n = w_d \cos \mathbf{a}_n =$ Doppler shifts, $n = 1, 2, \dots, N_0$,

$\mathbf{a}_n = 2\mathbf{p}n / N =$ arrival angle of the n th arrival wave, $n = 1, 2, \dots, N_0$,

$\mathbf{b}_n =$ phase of the n th arrival wave, $n = 1, 2, \dots, N_0$,

Then the autocorrelation function of $f_I(t)$ and $f_Q(t)$ are

$$\langle f_I^2(t) \rangle = N_0 + \cos^2 \mathbf{a} + \sum_{n=1}^{N_0} \cos 2\mathbf{b}_n, \quad (5.6)$$

$$\langle f_Q^2(t) \rangle = N_0 + \sin^2 \mathbf{a} - \sum_{n=1}^{N_0} \cos 2\mathbf{b}_n. \quad (5.7)$$

The cross-correlation between $f_I(t)$ and $f_Q(t)$ is

$$\langle f_I(t) \cdot f_Q(t) \rangle = \sum_{n=1}^{N_0} \sin 2\mathbf{b}_n + \frac{1}{2} \sin 2\mathbf{a}. \quad (5.8)$$

These correlation functions depend on the parameters \mathbf{a} and \mathbf{b}_n . In general, the following three cases are chosen.

Case 1 : $\mathbf{a} = \mathbf{p}/4$ and $\mathbf{b}_n = \mathbf{p}n / N_0$,

$$\langle f_I^2(t) \rangle = \langle f_Q^2(t) \rangle = N_0 + 1/2, \quad \langle f_I(t) \cdot f_Q(t) \rangle = 1/2. \quad (5.9)$$

Case 2 : $\mathbf{a} = 0$ and $\mathbf{b}_n = \mathbf{p}n / N_0$,

$$\langle f_I^2(t) \rangle = N_0 + 1, \quad \langle f_Q^2(t) \rangle = N_0, \quad \langle f_I(t) \cdot f_Q(t) \rangle = 0. \quad (5.10)$$

Case 3 : $\mathbf{a} = 0$ and $\mathbf{b}_n = \mathbf{p}n / (N_0 + 1)$,

$$\langle f_I^2(t) \rangle = N_0, \quad \langle f_Q^2(t) \rangle = N_0 + 1, \quad \langle f_I(t) \cdot f_Q(t) \rangle = 0. \quad (5.11)$$

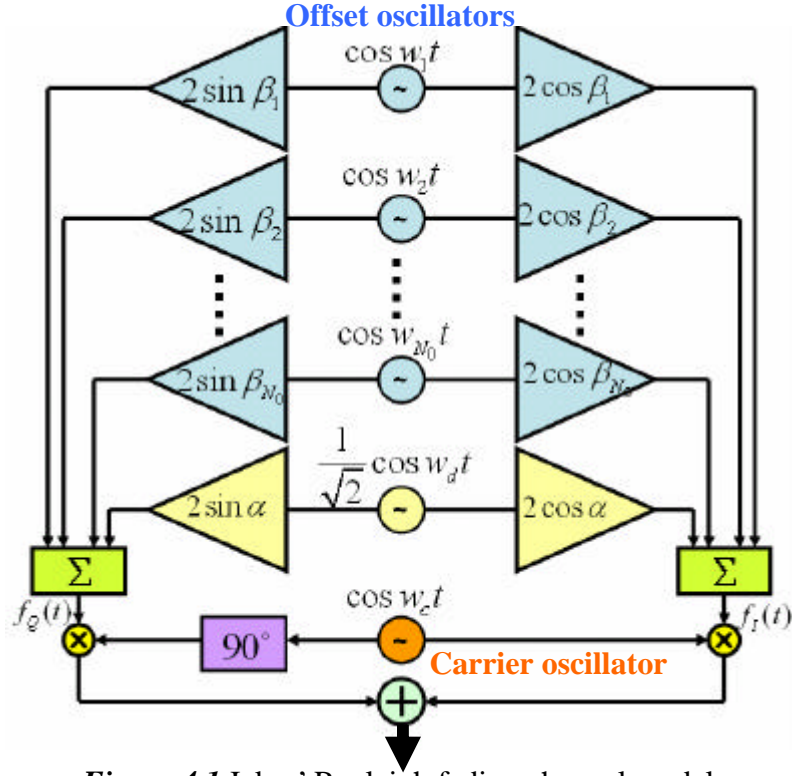


Figure 4.1 Jakes' Rayleigh fading channel model.

The Jakes' deterministic fading model is an established method for simulating time-correlated Rayleigh fading waveforms. For simulation of frequency selective and diversity combined fading channels, the modified Jakes' Model presented in [23] can produce multiple uncorrelated fading waveforms. There are two modified models in [23] and we adopt the model II and the simulator is shown in **Figure 4.2**. This model uses the same set of oscillators as suggested by Jakes but with adjusted initial phase shifts, \mathbf{b}_n and \mathbf{a}_j , the in-phase and quadrature components of j^{th} path ($j = 1, 2, \dots, N_0$) signal are:

$$f_{I_j}(t) = 2 \sum_{n=1}^{N_0} \cos \mathbf{b}_{nj} \cdot \cos(w_n t + \mathbf{p} j / 2) + 2 \left| \cos \mathbf{a}_j \right| \cdot \cos(w_d t + \mathbf{p} j / 2), \quad (5.12)$$

$$f_{Q_j}(t) = 2 \sum_{n=1}^{N_0} \sin \mathbf{b}_{nj} \cdot \cos(w_n t + \mathbf{p} j / 2) + 2 \sin \mathbf{a}_j \cdot \cos(w_d t + \mathbf{p} j / 2), \quad (5.13)$$

where $N = 2(2N_0 + 1)$, $N_0 \geq 8$, $\mathbf{a}_j = \mathbf{p} j$,

$w_n = w_d \cos \mathbf{a}_n =$ Doppler shifts, $n = 1, 2, \dots, N_0$,

$\mathbf{a}_n = 2\mathbf{p}n / N =$ arrival angle of the n th arrival wave, $n = 1, 2, \dots, N_0$,

$\mathbf{b}_{nj} = \mathbf{p}nj / (N_0 + 1) =$ phase of the n th arrival wave in j th path, $n = 1, 2, \dots, N_0$,

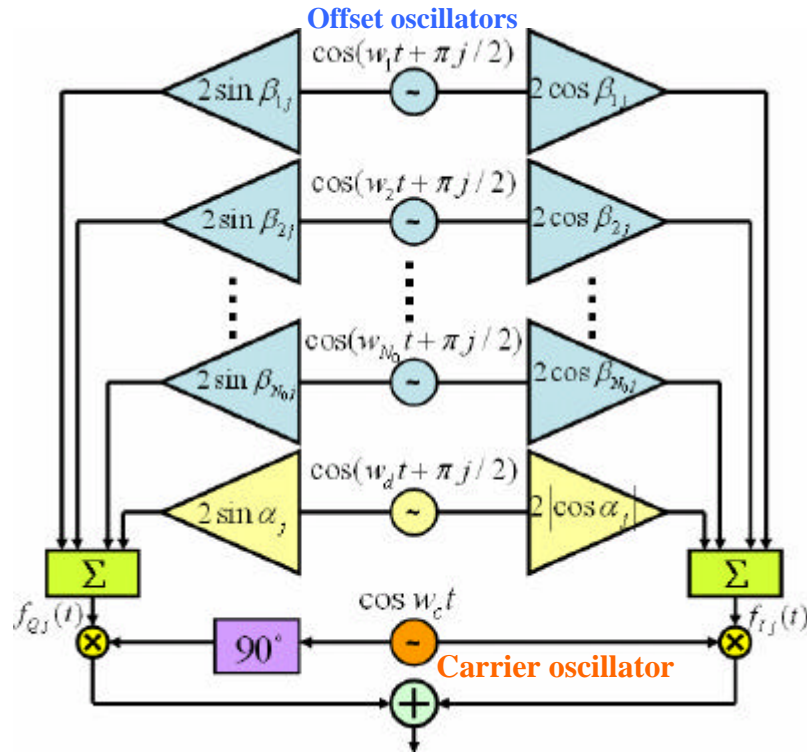


Figure 4.2 Modified Jakes' Rayleigh fading channel model (the j^{th} path).

In the following simulations, we use the Modified Jakes' Rayleigh fading channel model with the following settings: maximum Doppler frequency $w_d = 2\mathbf{p} \times 100$ Hz, the number of oscillators $N_0 = 16$, sample size = 1,000,000, sampling period $T = 10^{-5}$ sec. Then the normalized maximum Doppler shift $f_d T = 0.001$, can be regarded as slow fading. **Figure 4.3** shows an example of the envelope of our simulated output waveform within 0.1 second (10000 samples). The pdf and autocorrelation function of output waveforms are given in **Figure 4.4** and **Figure 4.5** for verification.

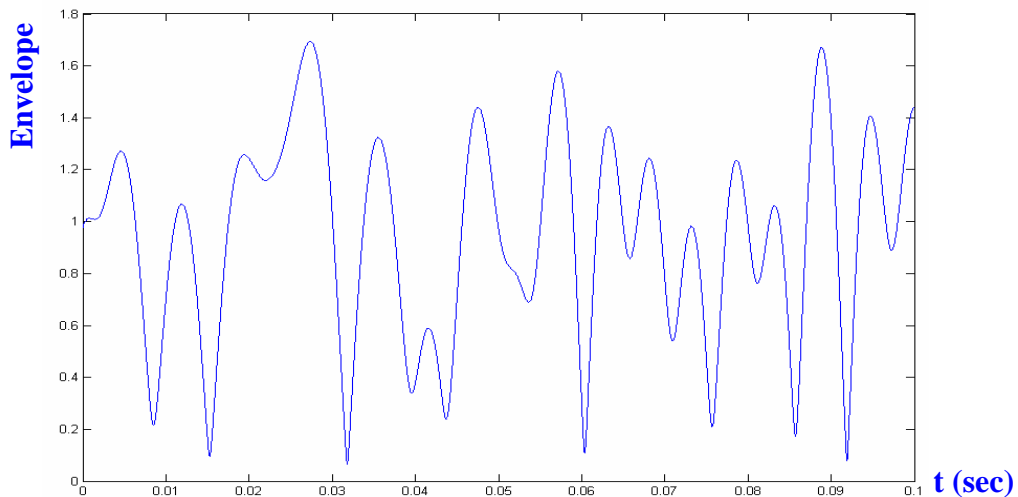


Figure 4.3 An example of the simulated output waveform envelope within 0.1 sec.

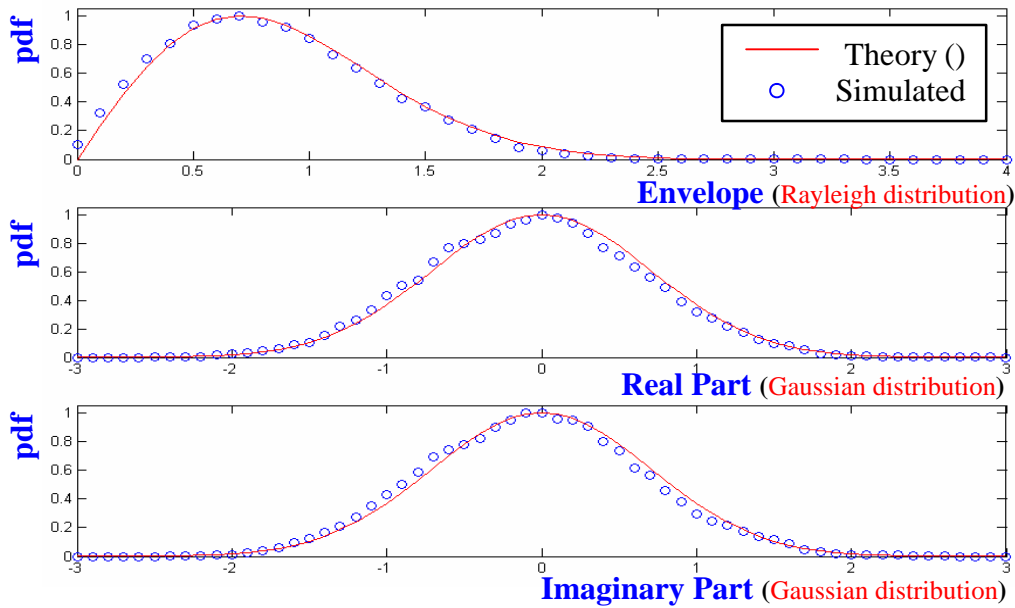


Figure 4.4 Envelope, in-phase component and quadrature component pdfs of modified Jakes' Rayleigh fading channel model output waveforms

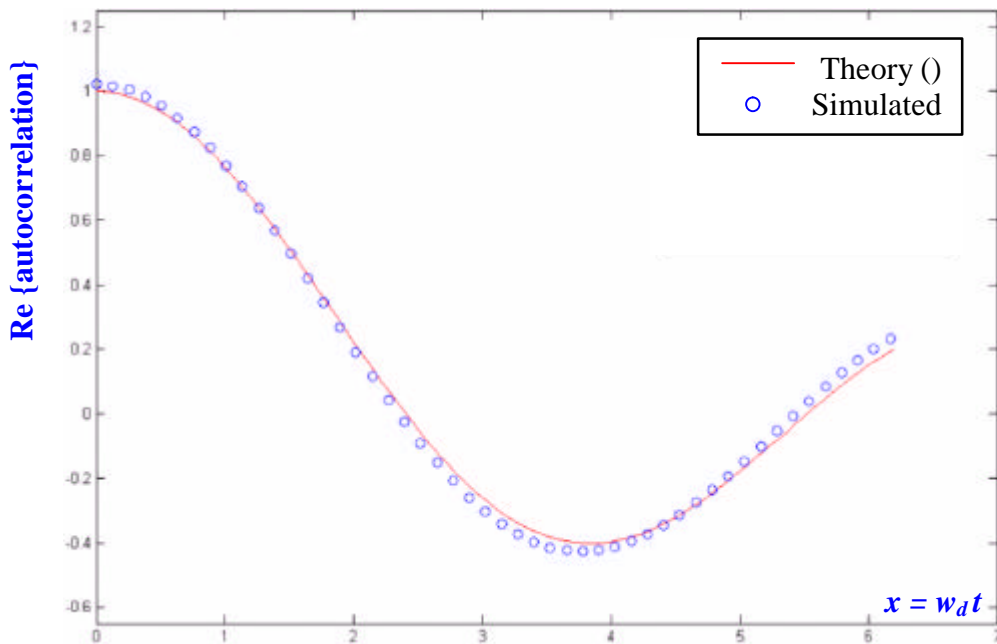


Figure 4.5 Autocorrelation function of modified Jakes' Rayleigh fading channel model output waveforms.

4.1.2 Multipath Channel Model

In P -ray multipath Rayleigh fading channel, there are one main path and $P-1$ delayed-paths in the propagation paths. We assume that the transmission signal is $s(t)$, and then we have the received signal

$$r(t) = \sum_{p=0}^{P-1} f_p(t) \cdot s(t - \mathbf{t}_p) + n(t), \quad (5.14)$$

where $n(t)$ is the complex AWGN term, $f_p(t)$ and \mathbf{t}_p denote the fading signal and the propagation delay of the p^{th} propagation path, respectively. The path with $p = 0$ is the main path, and the other paths with $p > 0$ are the delayed paths. In computer simulations, the $f_p(t)$, uncorrelated with the fading signals of other propagation paths, is generated by the aforementioned Rayleigh fading model independently. A wireless channel model with frequency selective fading in AWGN is employed in the following simulations. The unfaded finite impulse response (FIR) channel model consisting of three paths is given as,

$$h(n) = \sum_{i=1}^3 a_i \mathbf{d}(n - n_i). \quad (5.15)$$

the values $a_1 = 0.9$, $a_2 = 0.36$, $a_3 = 0.29$ and $n_1 = 0$, $n_2 = 6$, $n_3 = 11$ approximate the impulse response of a wireless channel in a warehouse-type indoor environment [13]. Each of the paths is multiplied by an independent Rayleigh fading function modeled in 4.1.1. The time domain delay profile and the corresponding frequency domain channel transfer function for 512 subcarriers are shown in **Figure 4.6**.

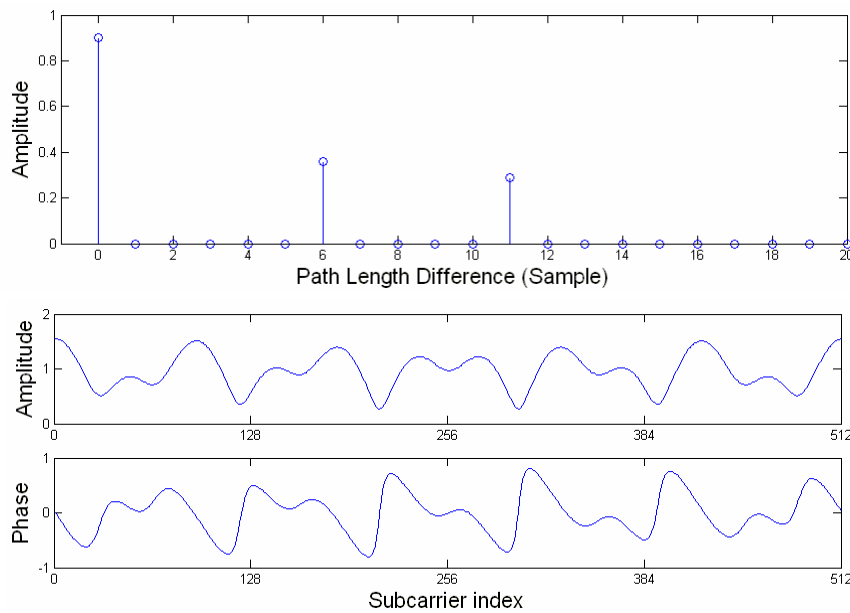


Figure 4.6 Time-dispersive wireless channel model for simulation. (a) unfaded impulse response. (b) unfaded frequency domain transfer function.

4.1.3 Simulation Platform and Parameters

A simplified OFDM system platform for simulations of frame synchronization techniques is shown in **Figure 4.7**. In a transmitter, data symbols are converted into parallel and fed to each port of IFFT. After adding the CP, the resultant OFDM signals are converted into serial and transmitted. In a receiver, the received OFDM signal is distorted into parallel and fed to FFT. The received signal is distorted by delayed signals due to multipath fading channel effect and AWGN. To demodulate the OFDM signal correctly, FFT should be started at the timing where the signal is not disturbed by the preceding symbol; the ideal timing is the end of GI.

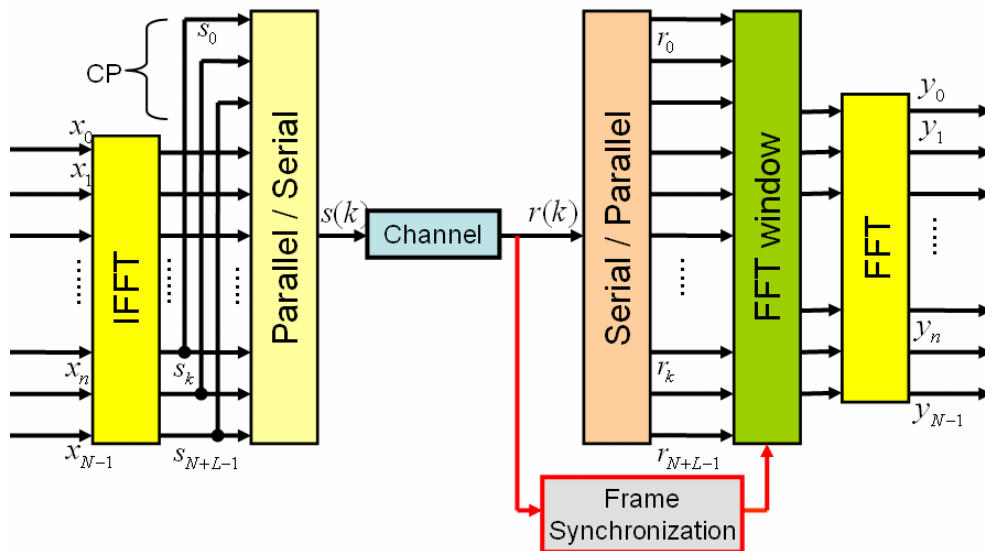


Figure 4.7 Simplified OFDM system platform for simulation.

The number of subcarriers / FFT(IFFT) size	512
Guard interval (CP length)	64 samples
The number of total transmitted frame	50000 frames
Modulation scheme	QPSK-OFDM
Delay profile of multipath fading channel	As shown in Figure 4.6(a)
Rayleigh fading channel model	Modified Jakes' model

Table 4.1 Common simulation parameters.

Table 4.1 shows the common parameters used in the following simulations. Note that the settings of Modified Jake's Rayleigh fading channel model are the same as we mentioned in **Section 4.1.1**. In the following **Section 4.2**, MATLAB simulations are performed to illustrate the synchronization accuracy of the proposed technique in terms of mean square error (MSE) of the frame timing and the histograms of frame timing estimation errors, the performance and complexity of the proposed algorithms are compared with other CP-based frame synchronization algorithms.

4.2 Simulation Results

Part I: The Correlation Functions $G(n)$ and $G_c(n)$

Figure 4.8 shows the correlation function $G(n)$ defined in (3.13), which is used for *PP* / *APP* / *GS* algorithms. **Figure 4.9** shows the complex-quantized correlation function $G_c(n)$ defined in (3.25), which is used for *CQ* / *ACQ* algorithms. In these two figures, correlation functions for 5 consecutive OFDM symbols at different average channel SNR are plotted.

We can see that the correlation peak value occurs at each frame boundary, and we can use this property to estimate the frame timing. Note that the correlation peak may not occur at the frame boundaries when the channel is severely faded or the average channel SNR is low, this will cause the frame timing estimation error if we just pick the peaking timing of each frame (*PP* algorithm). Thus there are several algorithms presented to improve the accuracy of frame timing estimation, as described in **Chapter 3**. These algorithms are simulated and the results will be shown in the following parts of this section.

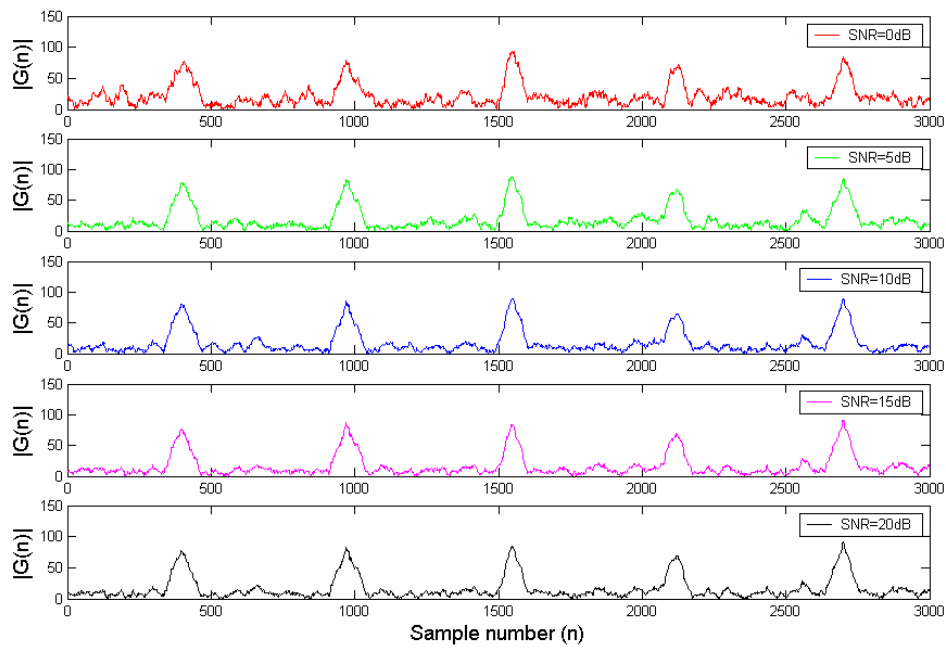


Figure 4.8 Simulated magnitude of $G(n)$ for 5 consecutive OFDM symbols at average channel SNR = 0, 5, 10, 15 and 20 dB.

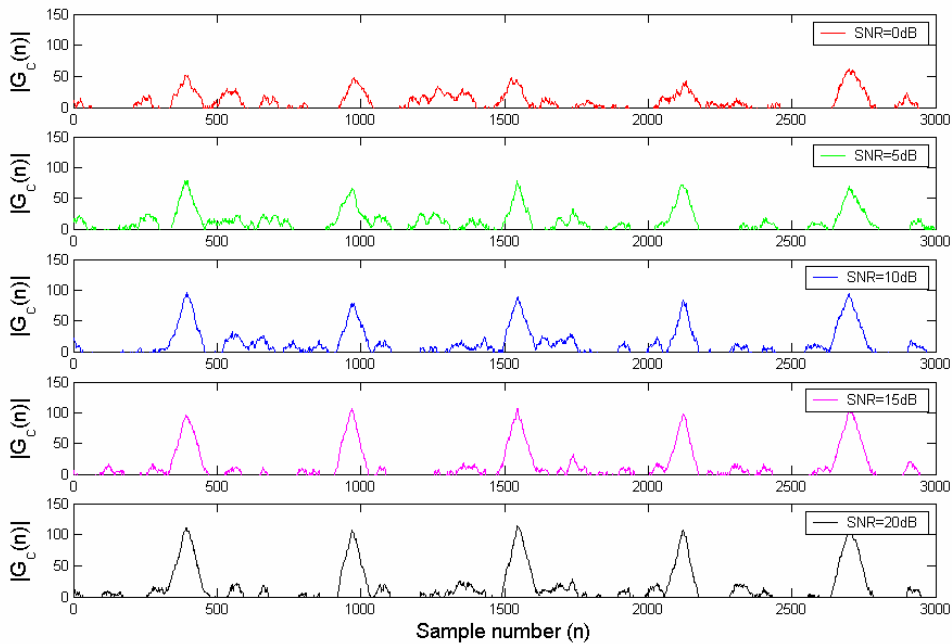


Figure 4.9 Simulated magnitude of $G_c(n)$ for 5 consecutive OFDM symbols at average channel SNR = 0, 5, 10, 15 and 20 dB.

Part II: *PP* / *APP* / *CQ* / *ACQ* Algorithms

Figure 4.10 shows the MSE of the frame timing against the average channel SNR for *PP* algorithm and *APP* algorithm using $M = 8, M = 16, M = 32, M = 64, M = 128$ and $M = 256$, where M is the averaging window length defined in (3.16). Note that the *PP* algorithm can be considered as the *APP* algorithm with $M = 1$. We can see that the *APP* algorithm performs better than the *PP* algorithm in terms of frame timing accuracy.

Figure 4.11 shows the histograms of the frame timing estimation errors for *PP* algorithm and *APP* algorithm at SNR = 0 and SNR = 10. At high SNR, $M = 8$ yields roughly the same performance as other values of M . When SNR is low, the pdf of frame timing errors is more concentrated toward 0 as M increases.

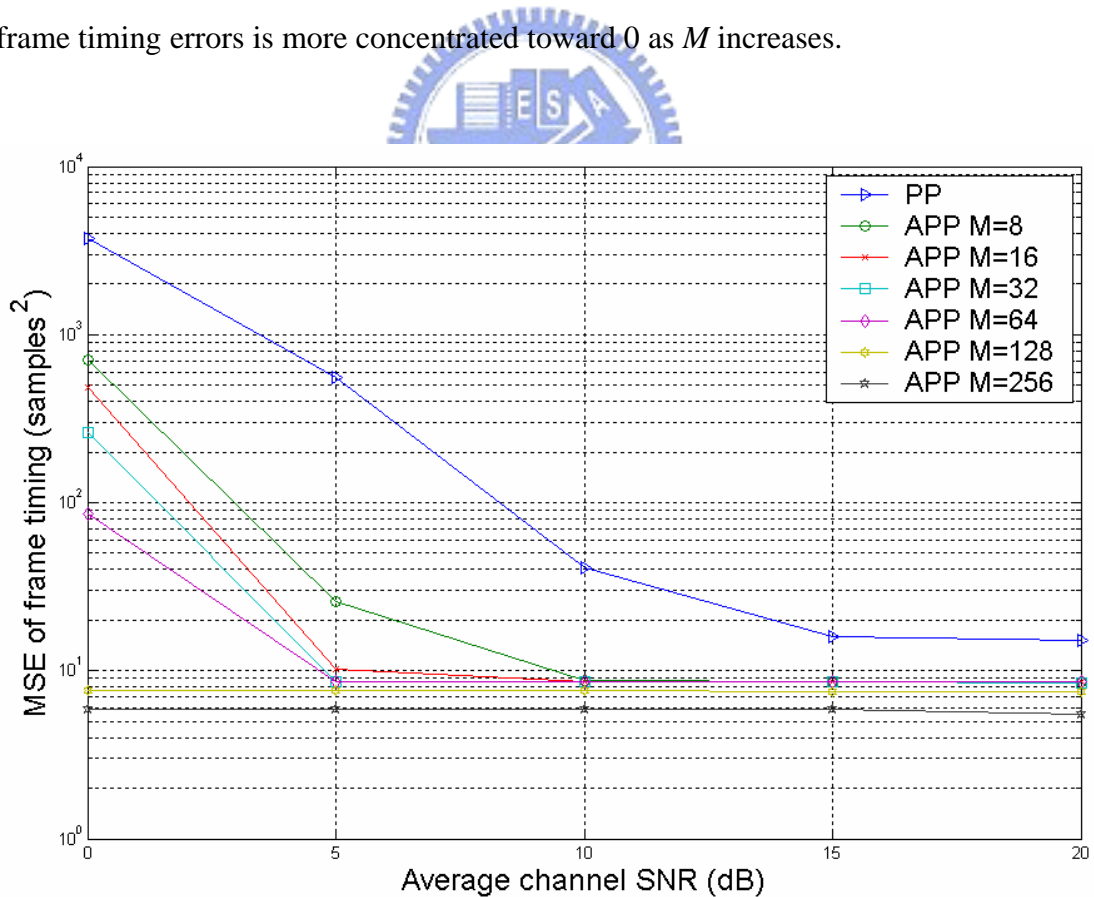
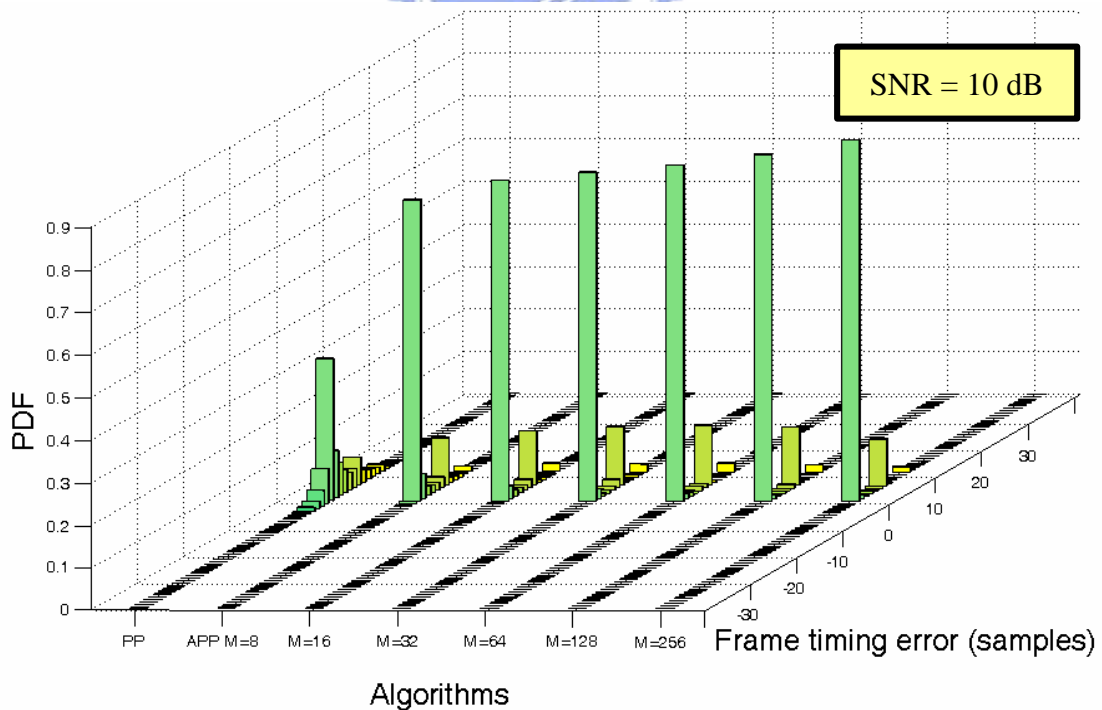
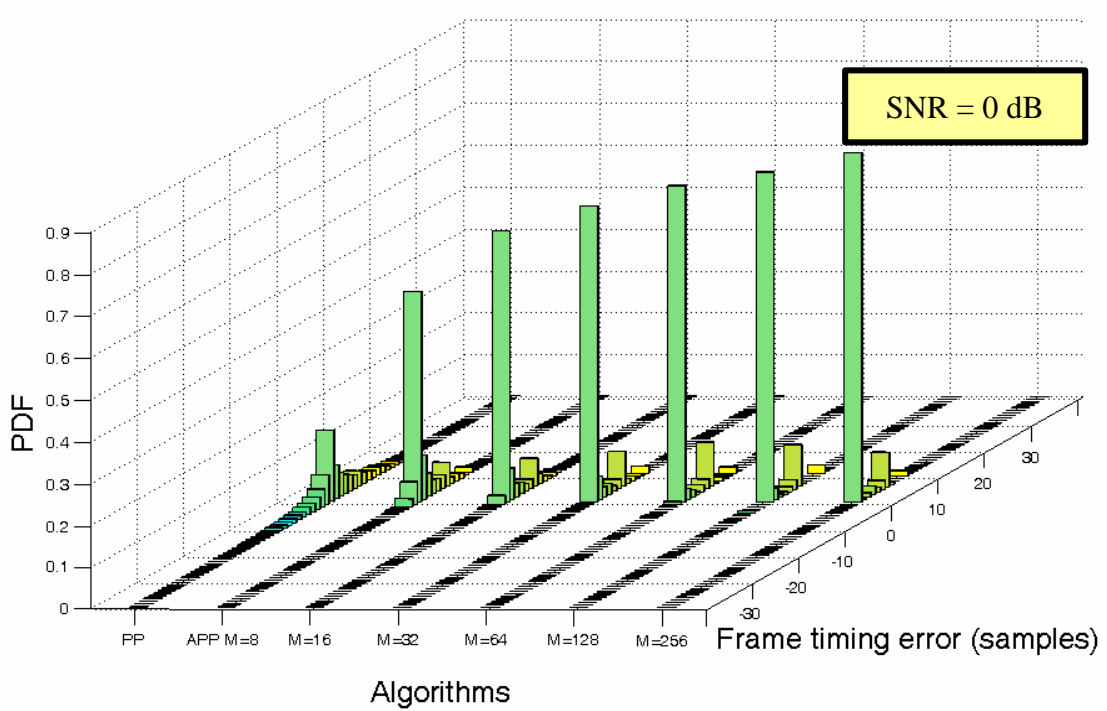


Figure 4.10 MSE of frame timing error against the average channel SNR for *PP* algorithm and *APP* algorithm using $M = 8, 16, 32, 64, 128$ and 256 .



(b)

Figure 4.11 Histograms of the frame timing error for *PP* algorithm and *APP* algorithm at (a) SNR = 0 and (b) SNR = 10.

Figure 4.12 shows the MSE of the frame timing against the average channel SNR for **CQ** algorithm and **ACQ** algorithm using $M = 8, M = 16, M = 32, M = 64, M = 128$ and $M = 256$, where M is the averaging window length defined in (3.26). Note that the **CQ** algorithm can be considered as the **ACQ** algorithm with $M = 1$. We can see that the complex-quantized correlation function also has the ability to find the frame timing. By using averaging method, the **ACQ** algorithm performs better than the **CQ** algorithm.

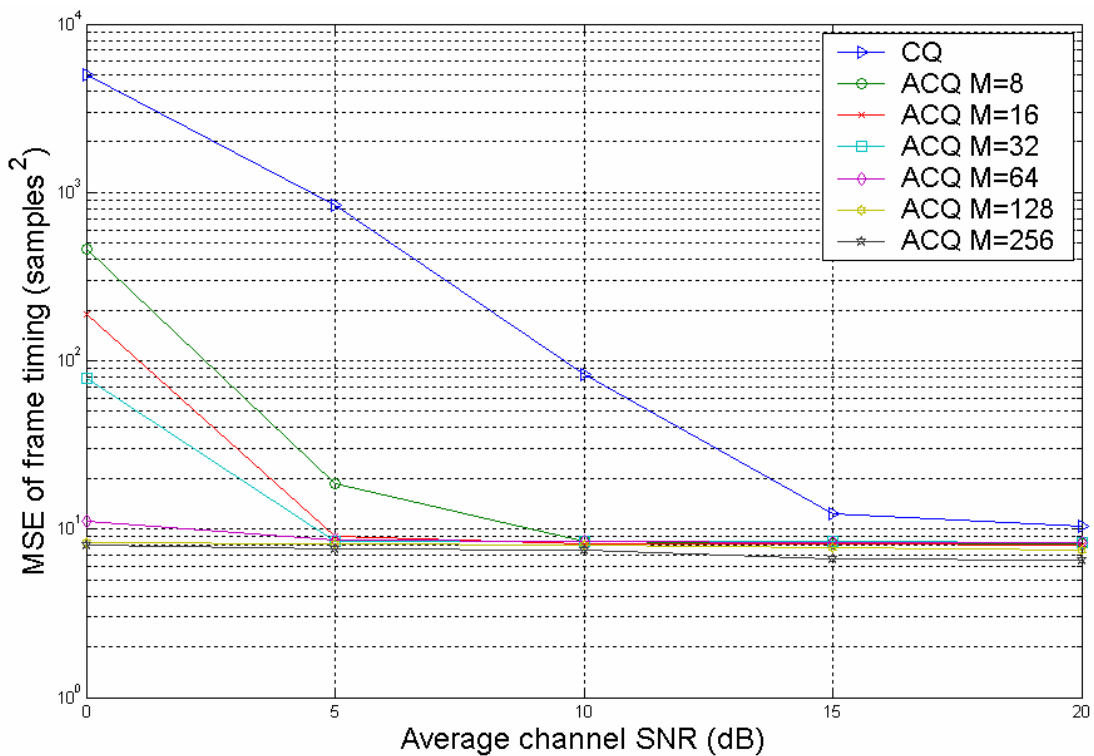
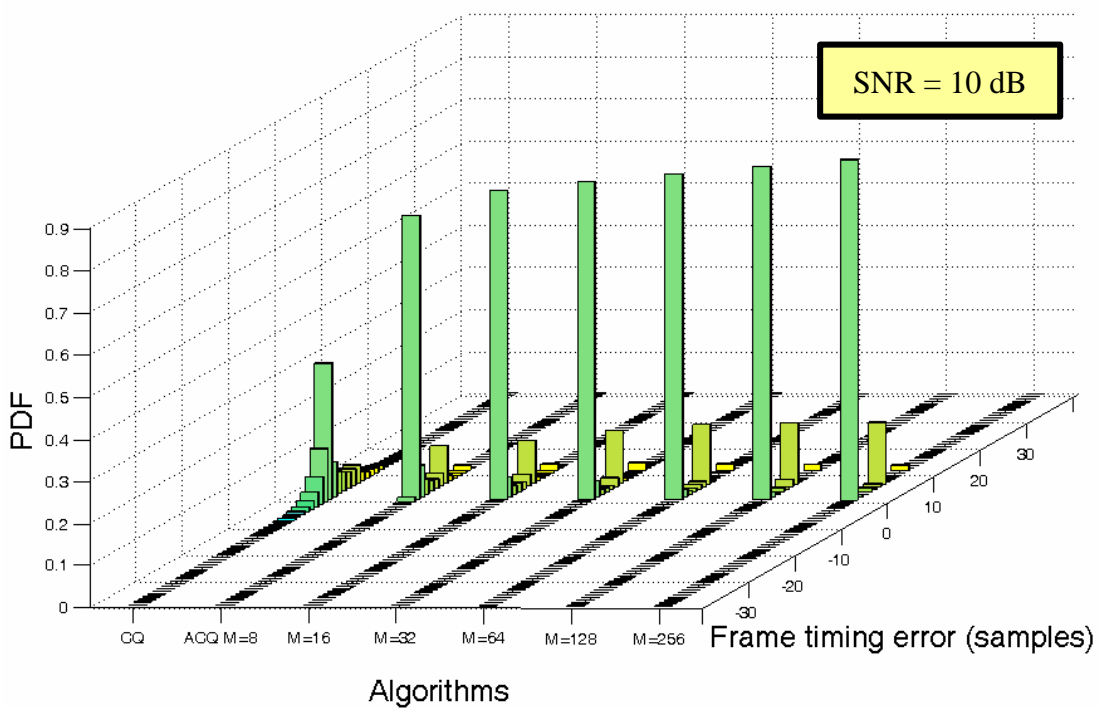
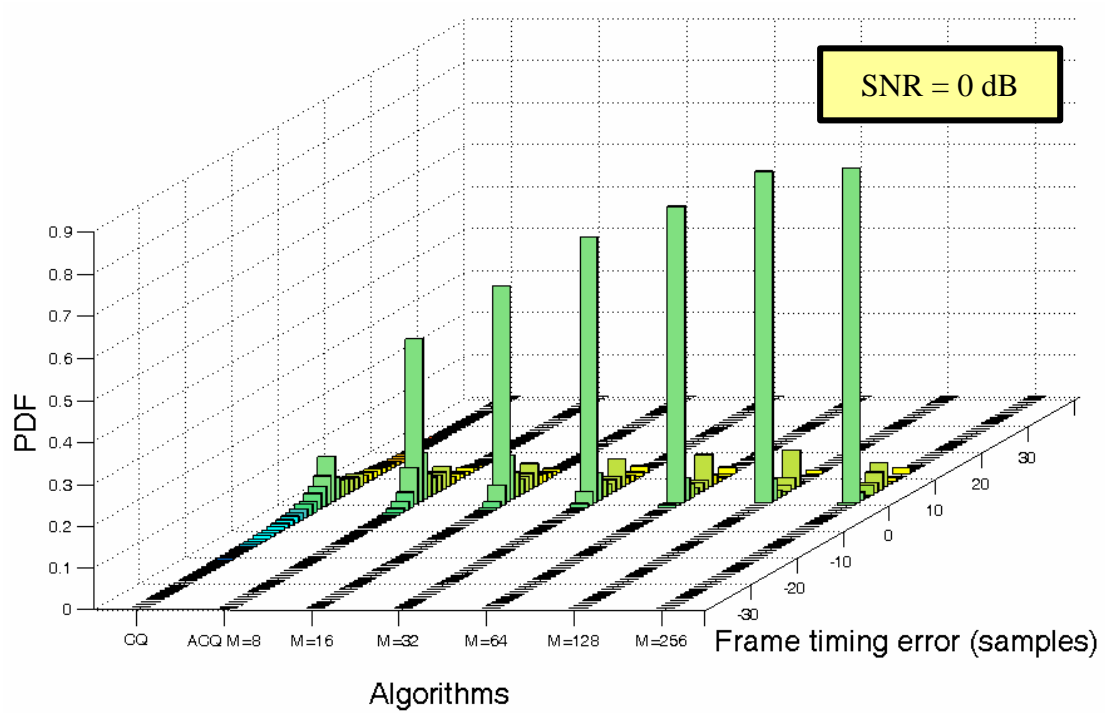


Figure 4.12 MSE of frame timing error against the average channel SNR for **CQ** and **ACQ** algorithm using $M = 8, 16, 32, 64, 128$ and 256 .

Figure 4.13 shows the histograms of the frame timing estimation errors for **CQ** algorithm and **ACQ** algorithm at $\text{SNR} = 0$ and $\text{SNR} = 10$. At high SNR, $M = 8$ yields roughly the same performance as other values of M . When SNR is low, the pdf of frame timing errors is more concentrated toward 0 as M increases.



(b)

Figure 4.13 Histograms of the frame timing error for *CQ* algorithm and *ACQ* algorithm at (a) SNR = 0 and (b) SNR = 10.

Now we consider the *PP* / *APP* and *CQ* / *ACQ* algorithms together. *Table 4.2* shows the MSE of the frame timing against the average channel SNR for *PP* / *APP* / *CQ* / *ACQ* algorithms. Compared with *PP* algorithm, *CQ* algorithm performs worse at low SNR but better at high SNR, because the sign bits of the received signal almost immune at high SNR. After averaging, *ACQ* algorithm's performance is better than *APP*'s when $M = 8, 16, 32$ and 64 , especially at low SNR. If M is increased to 128 and 256 , *APP* and *ACQ* performs almost the same, but *ACQ* has lower computational complexity because only the sign bits of the received signal are computed.

Algorithm		Average channel SNR (dB)				
		0	5	10	15	20
<i>PP</i>		3718.06	552.16	41.33	15.94	15.27
<i>CQ</i>		4954.89	843.49	83.51	12.29	10.36
<i>APP</i>	$M=8$	704.15	26.00	8.69	8.68	8.64
<i>ACQ</i>		458.20	18.75	8.42	8.16	8.13
<i>APP</i>	$M=16$	487.71	10.30	8.64	8.63	8.60
<i>ACQ</i>		189.73	9.01	8.18	8.16	8.06
<i>APP</i>	$M=32$	260.84	8.61	8.61	8.54	8.51
<i>ACQ</i>		78.91	8.52	8.50	8.44	8.34
<i>APP</i>	$M=64$	85.58	8.68	8.64	8.60	8.57
<i>ACQ</i>		11.10	8.60	8.52	8.34	8.28
<i>APP</i>	$M=128$	7.63	7.62	7.58	7.56	7.55
<i>ACQ</i>		8.25	8.17	8.06	7.80	7.53
<i>APP</i>	$M=256$	5.93	5.92	5.90	5.88	5.52
<i>ACQ</i>		8.05	7.67	7.50	6.60	6.55

Table 4.2 MSE of the frame timing against the average channel SNR for *PP* / *APP* / *CQ* / *ACQ* algorithms.

Simulation results about the *Smoothing-CQ* algorithms (*SMA*, *EWMA* and *EWA*) are not given here, because the performance improvement from *CQ* to *Smoothing-CQ* is not remarkable in our simulation environment.

Part III: *GS / LS / MGS* Algorithms

In this part, we consider the *GS* algorithm and two proposed algorithms, *LS* and *MGS* algorithms, based on *GS* algorithm. Replacing L ($=64$) and N ($=512$) of our simulation settings into (3.33), the percentage reduction of computational cost of the *GS* algorithm, compared with the *PP* algorithm, is 77.78 %. The *LS* algorithm we proposed in *Subsection 3.4.1* has almost the same computational complexity as *GS* algorithm, except the extra calculation for local searching range. From (3.34), the computational complexity of another proposed *MGS* algorithm is $\frac{8-(P+S)}{9} \times 100\%$, which depends on the values of S , the number of candidate points, and P , the number of randomly-chosen points. Simulation results for *GS / LS / MGS* algorithms are shown as follows.

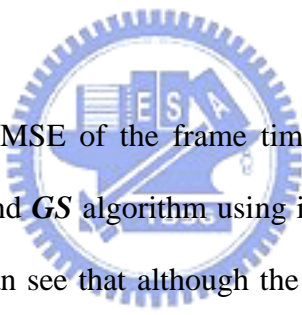


Figure 4.14 shows the MSE of the frame timing against the average channel SNR for *PP*, *APP* ($M = 8$) and *GS* algorithm using iteration period $M_{GS} = 64$, $M_{GS} = 128$, and $M_{GS} = 256$. We can see that although the performance of *GS* algorithm is much worse than *APP* algorithm even the M is only 8, it performs slightly better than the *PP* algorithm and the complexity is much lower than *PP* and *APP* algorithms. The MSE of $M_{GS} = 64$, 128, and 256 are roughly the same at high SNR, but when SNR is low, the *GS* estimator with larger M_{GS} performs better than that with lower M_{GS} .

Figure 4.15 shows the histograms of the frame timing estimation errors for *PP*, *APP* ($M = 8$) and *GS* algorithms at SNR = 0 and SNR = 10. We can see that when SNR is low, if the iteration period M_{GS} is too small, the pdf of the estimated frame timing errors will disperse.

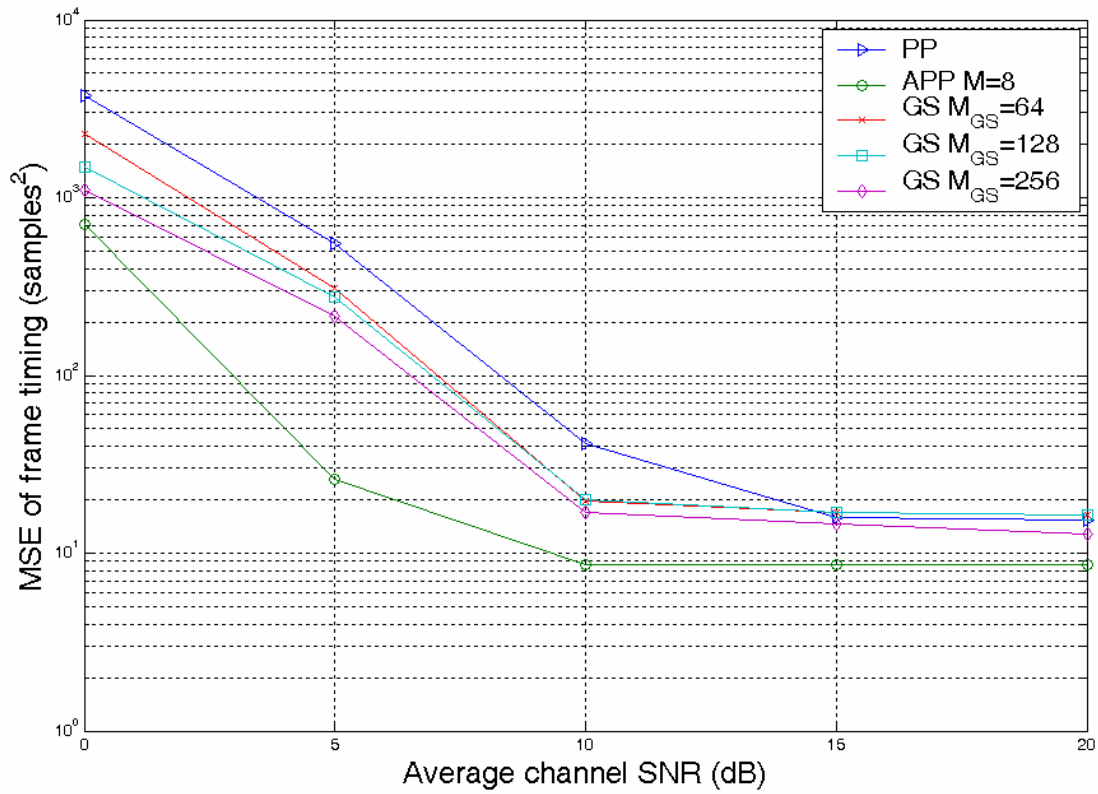


Figure 4.14 MSE of frame timing error against the average channel SNR for *PP*, *APP* ($M = 8$), and *GS* algorithm using $M_{GS} = 64, 128$ and 256 .

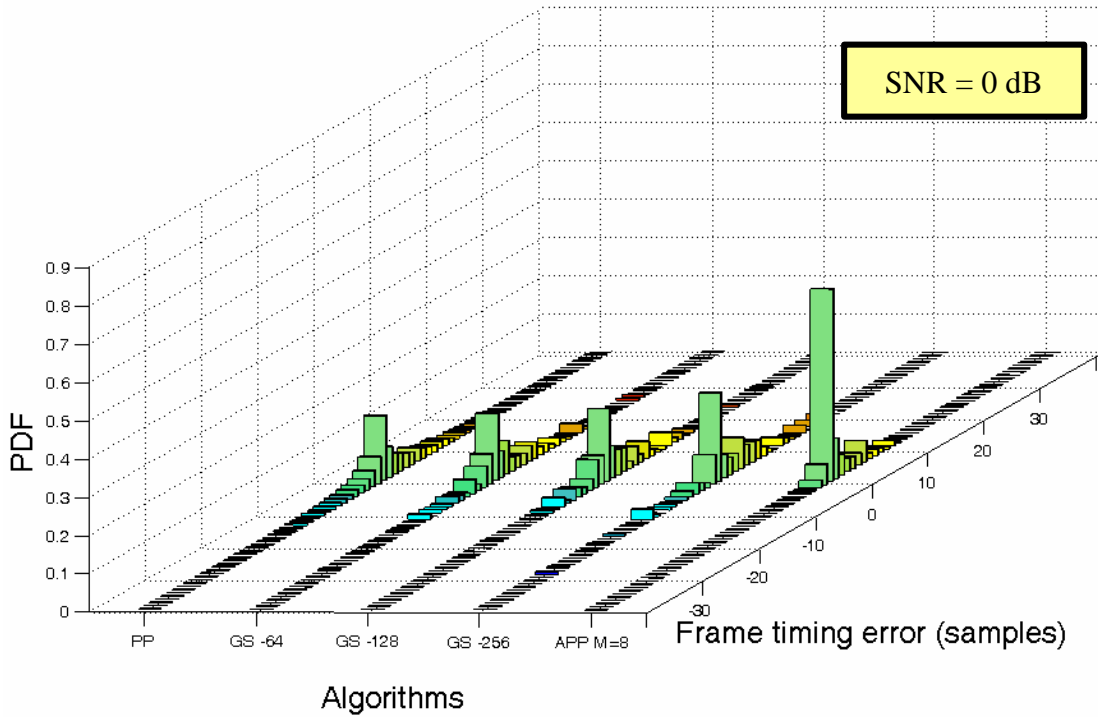


Figure 4.15(a) Histograms of the frame timing error for *PP*, *APP* ($M = 8$), and *GS* algorithm at SNR = 0.

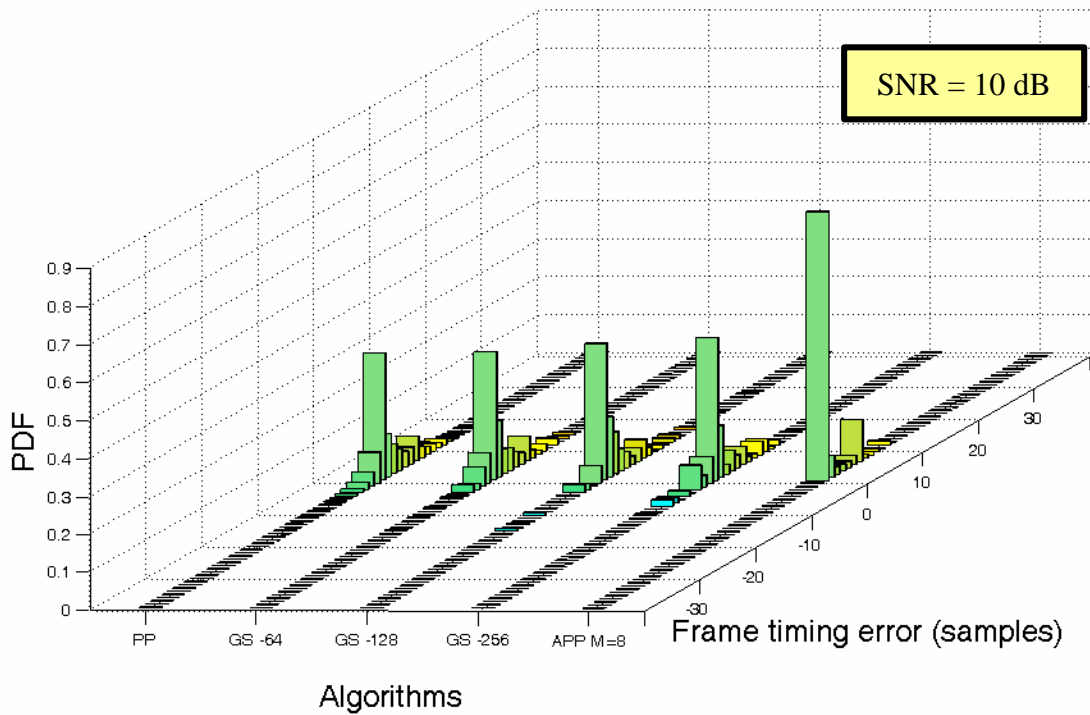


Figure 4.15(b) Histograms of the frame timing error for *PP*, *APP* ($M = 8$), and *GS* algorithm at SNR = 10.

In the simulations about *LS* algorithm, we choose the searching range parameter R , which is defined in *Subsection 3.4.1*, as 5 and 10 to observe the estimation performance. **Figure 4.16** illustrates the MSE of the frame timing against the average channel SNR for *PP*, *APP* ($M = 8$), *GS* and *LS* algorithm using $R = 5$ and $R = 10$. Note that the iteration period of *GS* algorithm, M_{GS} , and *LS* algorithm, M_{LS} , are all set to be 256. The figure shows that the *LS* algorithm performs almost the same with *APP* ($M = 8$) algorithm and much better than *PP* algorithm at high SNR, but the performance at low SNR is worse than the *GS* algorithm. This is because that when SNR is low, if the initial estimate is incorrect, the correct sample point may not within the searching range, thus it may takes a long time (even more than 256 iterations) to track the correct sample point. **Figure 4.17** shows the histograms of the frame timing estimation errors for *PP*, *APP* ($M = 8$), *GS* algorithm and *LS* algorithm using $R = 5$ and 10 at SNR = 0 and 10.

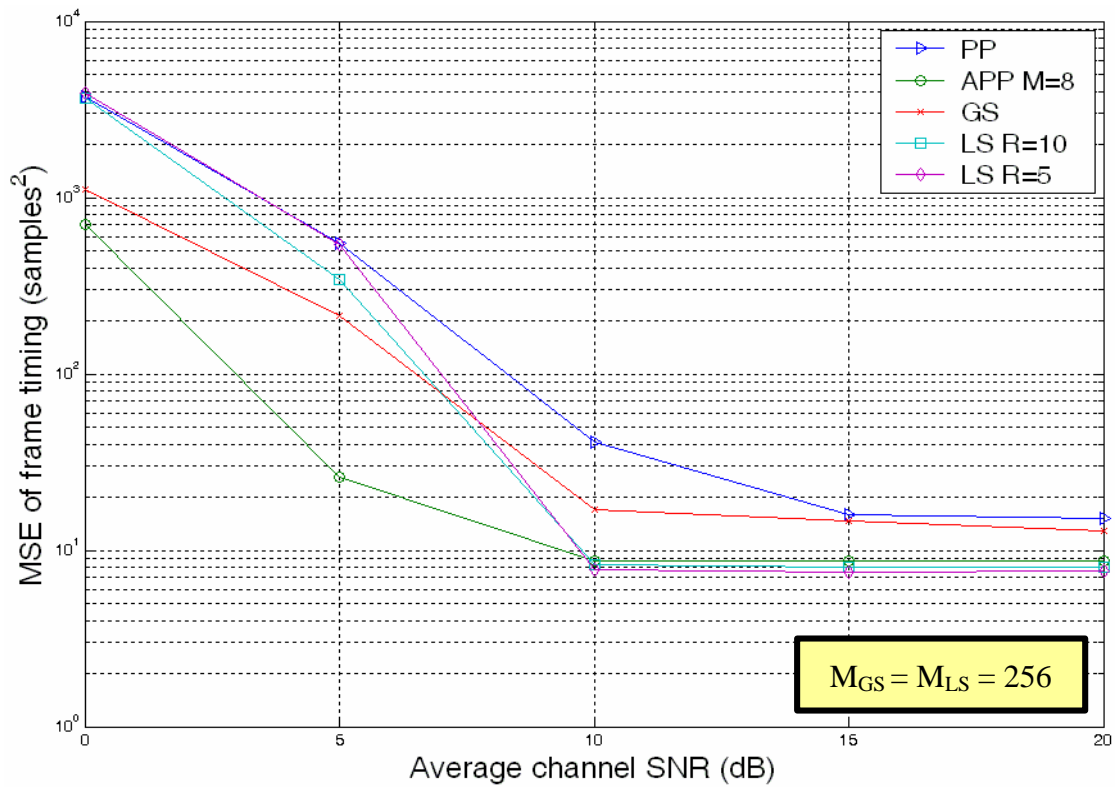


Figure 4.16 MSE of frame timing error against the average channel SNR for PP, APP ($M = 8$), GS and LS algorithm using $R = 5$ and $R = 10$.

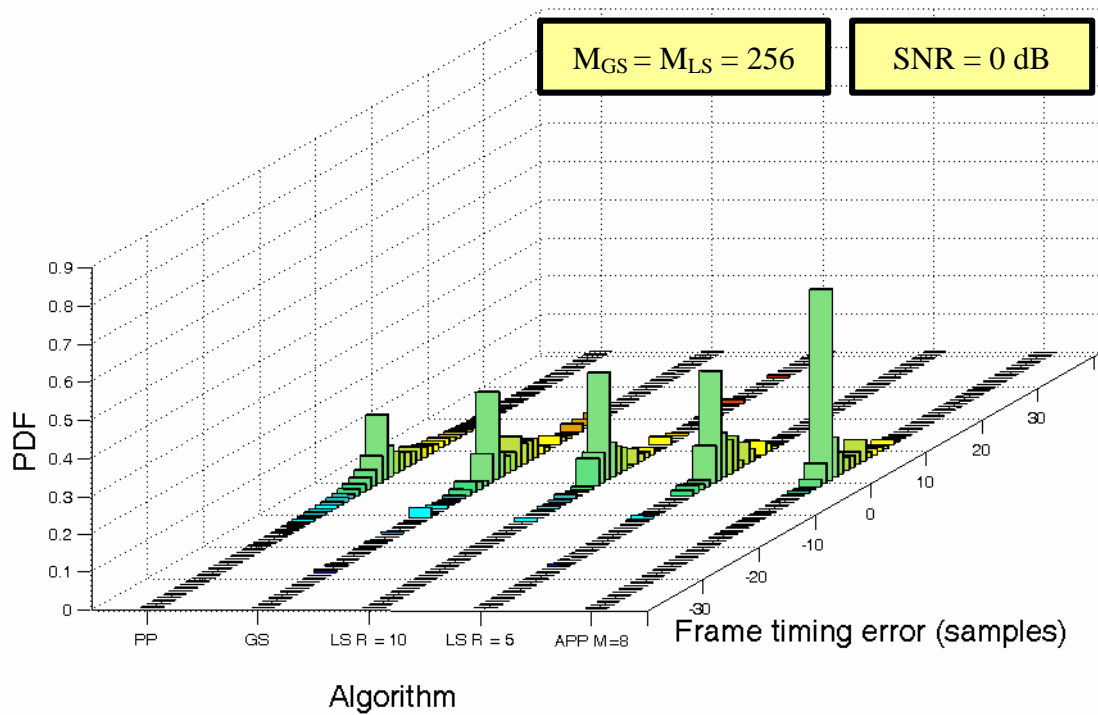


Figure 4.17(a) Histograms of the frame timing error for PP, APP ($M = 8$), GS and LS algorithm using $R = 5$ and $R = 10$ at SNR = 0.

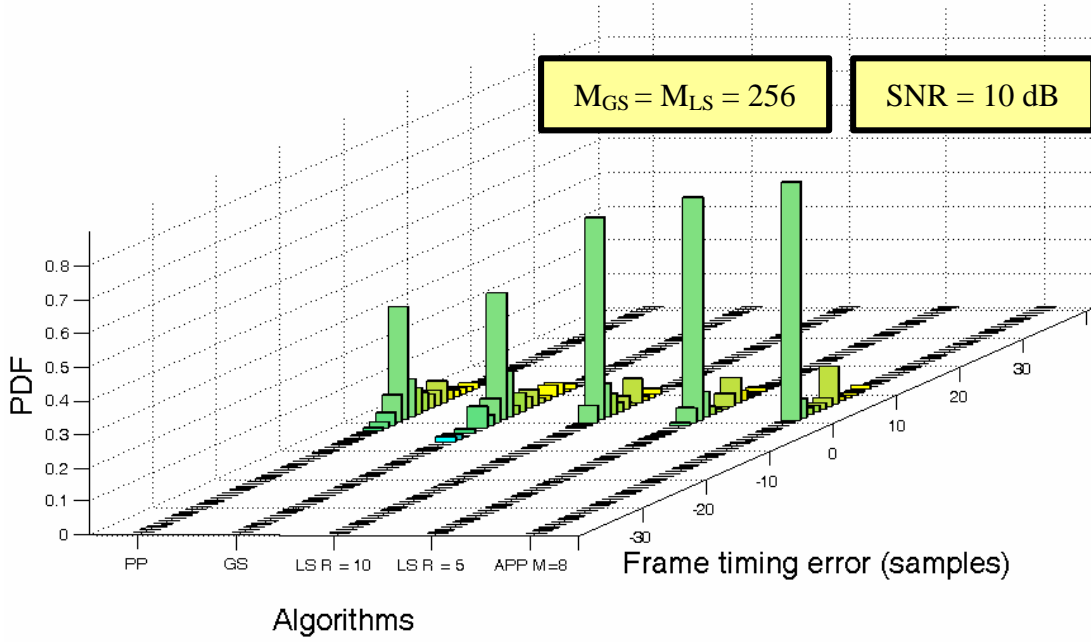


Figure 4.17(b) Histograms of the frame timing error for *PP*, *APP* ($M = 8$), *GS* and *LS* algorithm using $R = 5$ and $R = 10$ at $\text{SNR} = 10$.

For another proposed *MGS* algorithm, we only consider $P+S$ from 2 to 7 because, when $P+S = 8$, the computational complexity of *MGS* algorithm ($(P+S+1)L=576$) will be equal to that of the *PP* algorithm ($L+N=576$), the derivations of complexity of these two algorithm are shown in *Subsection 3.3.3* and *Subsection 3.4.2*, respectively. Also we set $K = L$, that means a candidate becomes the best estimate if and only if its correlation function is greater than that of the best estimate for consecutively K (or L) times, for simple realization, and K is simulated from 2 to 4.

Figure 4.18, *4.19*, and *4.20* show the MSE of the frame timing against the average channel SNR for *GS*, *APP* ($M = 8$ and 16) and *MGS* algorithm using $K = 2, 3$, and 4, respectively. Also we set the iteration period of *GS* algorithm, M_{GS} , and *MGS* algorithm, M_{MGS} , are all 256. Note that there are many different combinations of P and S when $P + S > 2$, the one with the minimum MSE is plotted in the figure. We can see that the *MGS* algorithm performs better than *GS* algorithm and yields roughly the same performance as *APP* algorithm using $M = 8$ and 16.

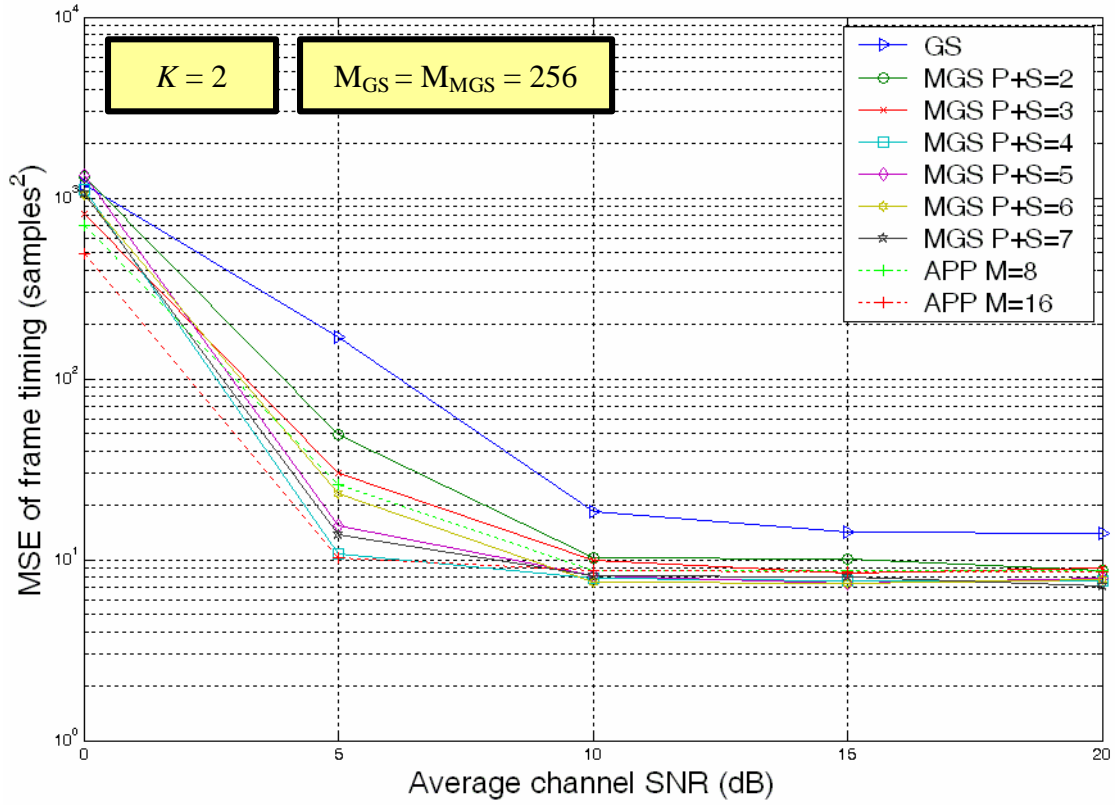


Figure 4.18 MSE of frame timing error against the average channel SNR for *GS*, *APP* ($M = 8$ and 16) and *MGS* algorithm using $K = 2$.

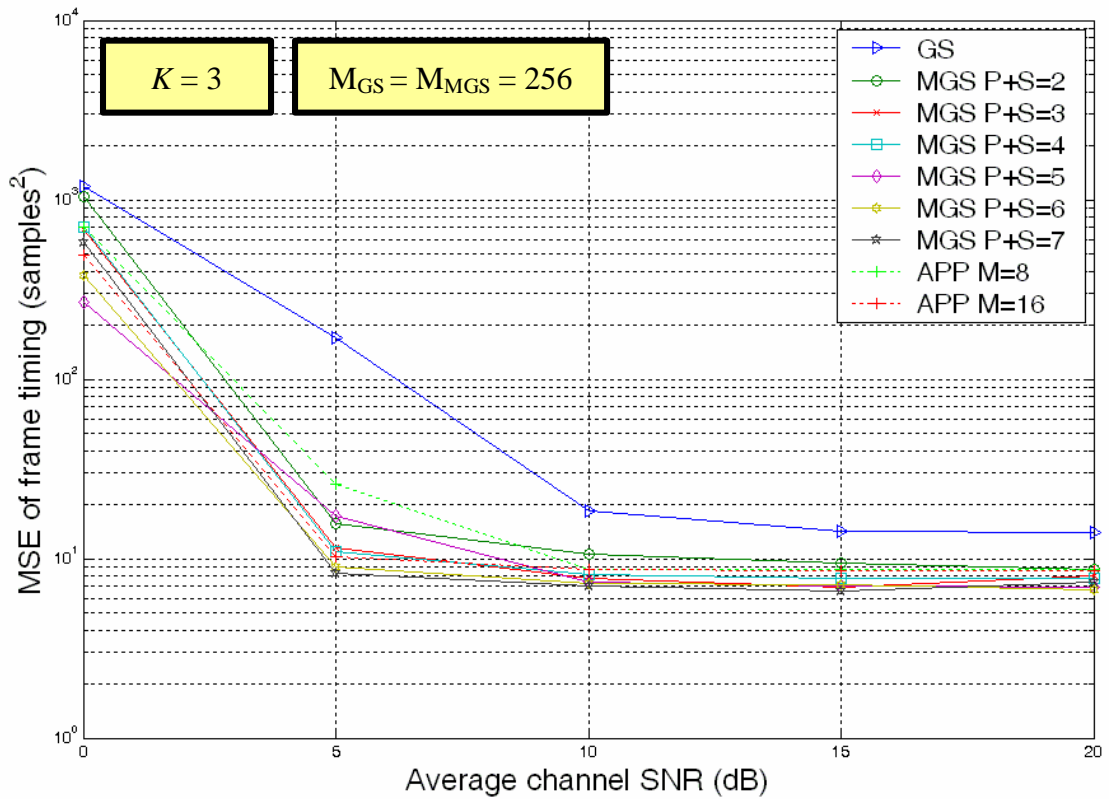


Figure 4.19 MSE of frame timing error against the average channel SNR for *GS*, *APP* ($M = 8$ and 16) and *MGS* algorithm using $K = 3$.

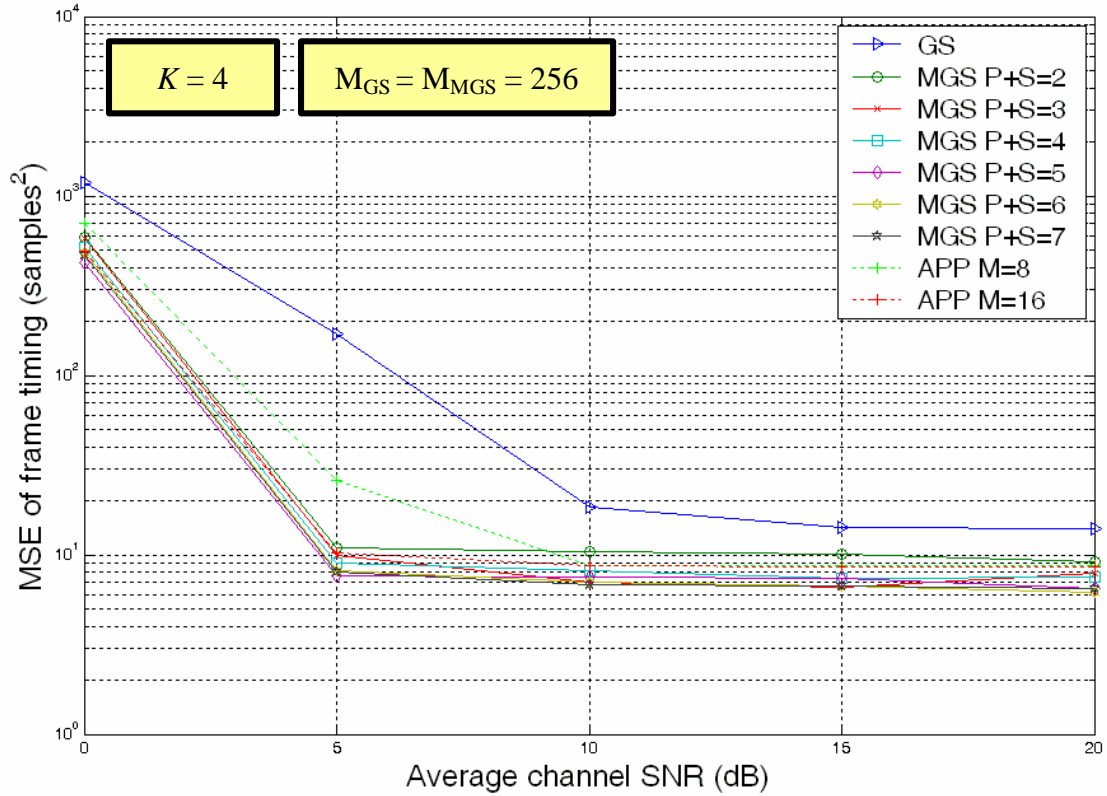


Figure 4.20 MSE of frame timing error against the average channel SNR for *GS*, *APP* ($M = 8$ and 16) and *MGS* algorithm using $K = 4$.

Algorithms	Complexity	Average channel SNR (dB)				
		0	5	10	15	20
<i>PP</i>	576	3718.06	552.16	41.33	15.94	15.27
<i>GS</i>	128	1111.16	214.51	16.98	14.59	12.84
<i>APP M = 8</i>	576	704.15	26.00	8.69	8.68	8.64
<i>APP M = 16</i>	576	487.71	10.30	8.64	8.63	8.60
<i>MGS, P+S = 2</i>	192	589.27	10.94	10.31	9.49	8.66
<i>MGS, P+S = 3</i>	256	580.72	9.87	7.84	8.57	7.89
<i>MGS, P+S = 4</i>	320	517.23	8.97	7.42	7.34	7.47
<i>MGS, P+S = 5</i>	384	267.20	7.70	7.04	7.07	6.48
<i>MGS, P+S = 6</i>	448	378.81	8.13	7.09	6.69	6.49
<i>MGS, P+S = 7</i>	512	465.51	8.01	6.76	6.59	6.18

Table 4.3 Comparison of computational complexity and MSE of the frame timing against the average channel SNR for *PP*, *APP*, *GS* and *MGS* algorithms.

Table 4.3 summarizes the computational complexity (number of complex multiplications required in each iteration) and MSE of the frame timing against the average channel SNR for *PP*, *APP*, *GS* and *MGS* algorithms. Note that given $P + S$ and SNR, there are 3 kinds of MSE for $K = 2, 3$, and 4. The one with the minimum MSE is listed in the table. When SNR is high, the performance of *MGS* algorithm improves as $P + S$ increases, therefore, $P + S = 7$ is the best among the various value of P and S . When SNR is at 0 and 5 dB, $P + S = 5$ is the best.

It can be seen from **Figure 4.19** and **4.20** that when $K = 3$ or 4, $P + S = 2$ yields roughly the same performance as $P + S = 7$, especially at high SNR. This also can be observed from the histograms of the frame timing estimation errors for *GS*, *APP* ($M = 8$ and 16) and *MGS* algorithm using $K = 4$ at SNR = 0 and 10 shown in **Figure 4.21**. Thus we can choose $P + S = 2$ or 3 to save a lot of computational complexity.

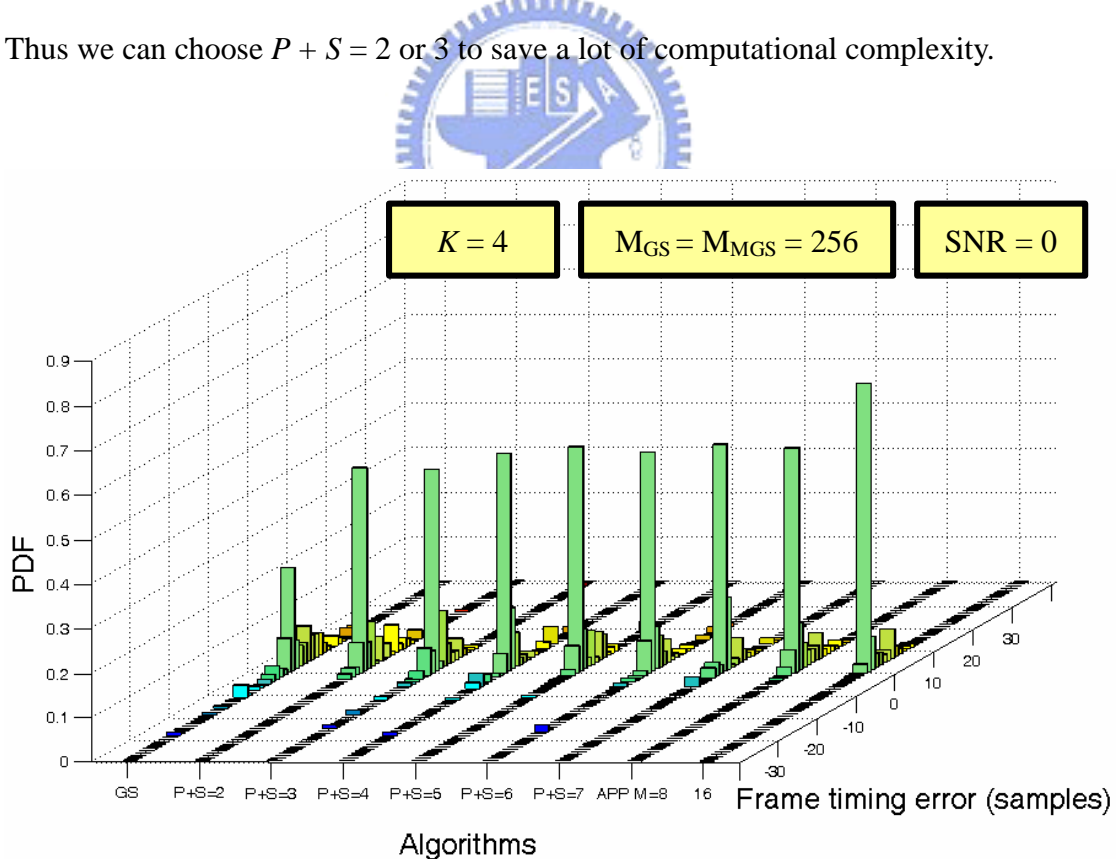


Figure 4.21(a) Histograms of the frame timing error for *GS*, *APP* ($M = 8$ and 16), and *MGS* algorithm using $K = 4$ at SNR = 0.

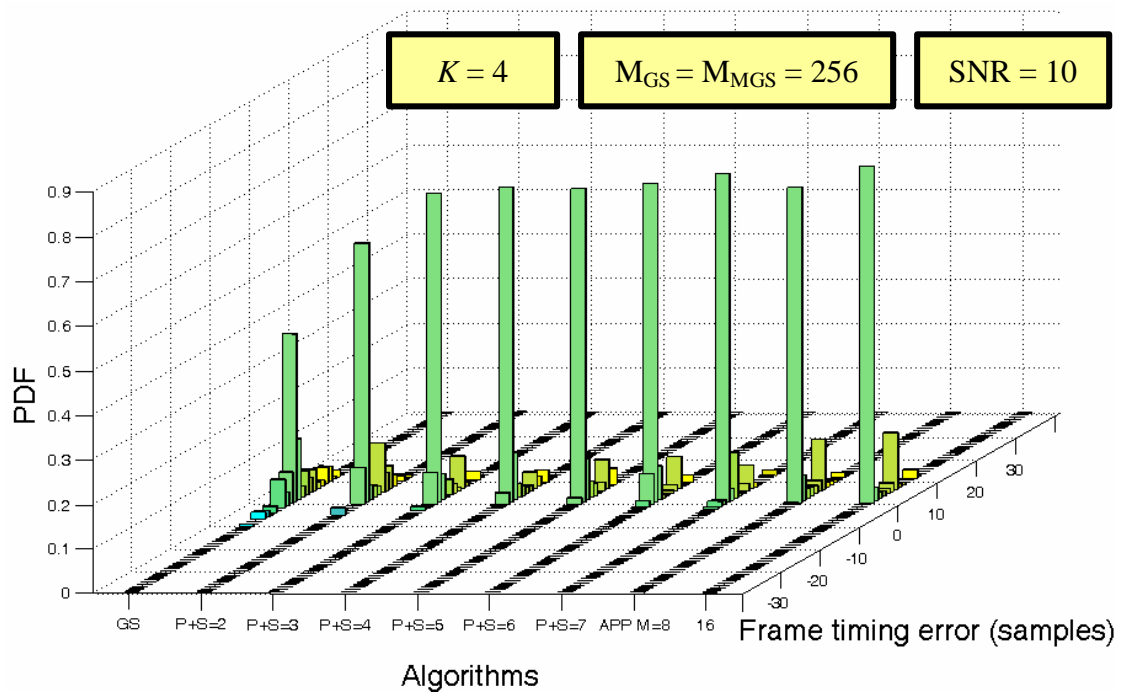
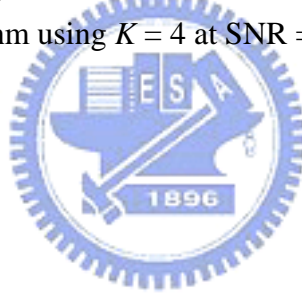


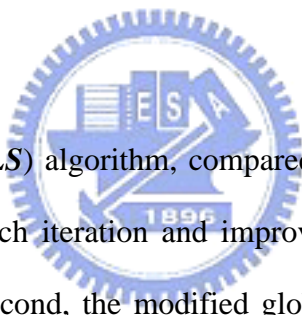
Figure 4.21(b) Histograms of the frame timing error for *GS*, *APP* ($M = 8$ and 16), and *MGS* algorithm using $K = 4$ at $SNR = 10$.



Chapter 5

Conclusions

In this thesis, we have introduced several cyclic prefix (CP) based frame synchronization techniques in the literature and presented two modified algorithms for OFDM systems. The modifications of these two proposed algorithms are based on the global search (*GS*) algorithm presented in [15] and all of them possess self-learning capability.



First, the local search (*LS*) algorithm, compared with the *GS* algorithm, narrows down the search range in each iteration and improves the accuracy of frame timing estimation at high SNR. Second, the modified global search (*MGS*) algorithm only evaluates the correlation functions of 3 to 8 sample points for each OFDM symbol for realization. Compared with the averaging and peak-picking (*APP*) algorithm, the *MGS* algorithm requires much less computational complexity. Compare with the *GS* algorithm, the *MGS* algorithm requires much less hardware, yields a better performance, and possesses much better re-synchronization capability with some increase in computation cost.

From the simulation results, we conclude that in the two algorithms we proposed, the *LS* algorithm is suitable at higher SNR, and the *MGS* algorithm is suitable at lower SNR. Considering the computational complexity and the accuracy of frame timing estimation together, the *MGS* algorithm is preferable to other CP-based algorithms.

Appendix A

Derivation of the Log-Likelihood Function (3.4)

The derivation of (3.4) is shown as follows. The log-likelihood function (3.3) can be written as

$$\Lambda(\mathbf{q}, \mathbf{e}) = \sum_{k=q}^{q+L-1} \log \left(\frac{f(r(k), r(k+N))}{f(r(k))f(r(k+N))} \right) \quad (\text{A.1})$$

The numerator is a 2-D complex-valued Gaussian distribution, which, using the correlation properties in (3.2), becomes

$$f(r(k), r(k+N)) = \frac{\exp\left(-\frac{|r(k)|^2 - 2\mathbf{r} \operatorname{Re}\{e^{j2p\mathbf{e}} r(k)r^*(k+N)\} + |r(k+N)|^2}{(\mathbf{s}_s^2 + \mathbf{s}_n^2)(1 - \mathbf{r}^2)}\right)}{\mathbf{p}^2(\mathbf{s}_s^2 + \mathbf{s}_n^2)(1 - \mathbf{r}^2)}, \quad (\text{A.2})$$

where \mathbf{r} is the correlation coefficient between $r(k)$ and $r(k+N)$ as defined in (3.7). The denominator of (A.1) consists of two 1-D complex Gaussian distributions

$$f(r(k)) = \frac{\exp\left(-\frac{|r(k)|^2}{(\mathbf{s}_s^2 + \mathbf{s}_n^2)}\right)}{\mathbf{p}(\mathbf{s}_s^2 + \mathbf{s}_n^2)} \quad (\text{A.3})$$

and the log-likelihood function (A.1), after some algebraic manipulations, becomes

$$\Lambda(\mathbf{q}, \mathbf{e}) = c_1 + c_2(|\mathbf{g}(\mathbf{q})| \cos(2p\mathbf{e} + \angle \mathbf{g}(\mathbf{q})) - \mathbf{r}\Phi(\mathbf{q})) \quad (\text{A.4})$$

where $\mathbf{g}(\mathbf{q})$ and $\Phi(\mathbf{q})$ are defined in (3.5) and (3.6), and c_1 and c_2 are constants, independent of \mathbf{q} and \mathbf{e} . Since the maximizing argument of $\Lambda(\mathbf{q}, \mathbf{e})$ in (A.4) is independent of the constants c_1 and c_2 , and $c_2 > 0$, the ML-estimate $(\hat{\mathbf{q}}_{ML}, \hat{\mathbf{e}}_{ML})$ also maximizes (3.4). [9]

Appendix B

Derivation of the Complex-Quantization ML Estimator (3.25)

Notice first that

$$c(k) = Q[r(k)] = a_l \Leftrightarrow r(k) \in Q_l \quad (\text{B.1})$$

where $Q_l, l = 0, 1, 2, 3$ are quadrants of the complex plane as depicted in **Figure 3.5**.

Since the real and imaginary part of $r(k)$ are independent, we can write the probability

$$\Pr(r(k) \in Q_l) = \Pr(\text{Re}\{r(k)\} \in H_l) \cdot \Pr(\text{Im}\{r(k)\} \in H'_l) \quad (\text{B.2})$$

(see **Figure 3.5**), where H_l and H'_l are the half-planes

$$H_l = \begin{cases} \mathfrak{R}^+, & l = 0, 3 \\ \mathfrak{R}^-, & l = 1, 2 \end{cases}, \quad H'_l = \begin{cases} \mathfrak{R}^+, & l = 0, 3 \\ \mathfrak{R}^-, & l = 1, 2 \end{cases}. \quad (\text{B.3})$$

The joint probability $p_l(\bullet)$ in (3.20) can be written as

$$\begin{aligned} \Pr(c(k) = a_l, c(k-N) = a_n) &= \\ \Pr(r(k) \in Q_l, r(k-N) \in Q_n) &= \\ \Pr(\text{Re}\{r(k)\} \in H_l, \text{Re}\{r(k-N)\} \in H_n) &\times \\ \Pr(\text{Im}\{r(k)\} \in H'_l, \text{Im}\{r(k-N)\} \in H'_n). & \end{aligned} \quad (\text{B.4})$$

By applying the symmetry expressions for two real, zero-mean, jointly Gaussian variables x and y with correlation coefficient ρ [31],

$$\begin{aligned} P^+ &\triangleq \Pr((x, y) \in \mathfrak{R}^+ \times \mathfrak{R}^+) = \Pr((x, y) \in \mathfrak{R}^- \times \mathfrak{R}^-) = \frac{1}{4} + \frac{\rho}{2\sqrt{1-\rho^2}} \\ P^- &\triangleq \Pr((x, y) \in \mathfrak{R}^- \times \mathfrak{R}^+) = \Pr((x, y) \in \mathfrak{R}^+ \times \mathfrak{R}^-) = \frac{1}{4} - \frac{\rho}{2\sqrt{1-\rho^2}}, \end{aligned} \quad (\text{B.5})$$

where $\mathbf{a} = \frac{1}{2} - \tan^{-1}\left(\frac{\mathbf{r}}{\sqrt{1-\mathbf{r}^2}}\right)$. The look-up table in *Figure 3.6* becomes

$c(k) \backslash c(k-N)$	a_0	a_1	a_2	a_3
a_0	$\log P^+ P^+$	$\log P^- P^+$	$\log P^- P^-$	$\log P^+ P^-$
a_1	$\log P^+ P^-$	$\log P^+ P^+$	$\log P^- P^+$	$\log P^- P^-$
a_2	$\log P^- P^-$	$\log P^+ P^-$	$\log P^+ P^+$	$\log P^- P^+$
a_3	$\log P^- P^+$	$\log P^- P^-$	$\log P^+ P^-$	$\log P^+ P^+$

Table B.1 Expression of the look-up table in *Figure 3.6*.

The ML estimate (3.21) that uses this look-up table is not affected by an affine scaling $f(x) = ax + b$, ($a > 0$) of the values in the table, since it is a convex mapping.

If we choose

$$f(x) = \frac{x - \log P^+ P^-}{\log P^+ P^+ - \log P^+ P^-} \tag{B.6}$$

the non-linearity $f[g(l, n)]$ becomes

$c(k) \backslash c(k-N)$	a_0	a_1	a_2	a_3
a_0	1	0	-1	0
a_1	0	1	0	-1
a_2	-1	0	1	0
a_3	0	-1	0	1

Table B.2 The look-up table after convex mapping.

Since

$$\text{Entry}(a_n, a_l) = \frac{1}{2} \text{Re}\{a_n a_l^*\}, \tag{B.7}$$

the look-up table may be implemented by taking real part of $c(k)c^*(k-n)$, as we shown in (3.25). [12]

Bibliography

- [1] IEEE 802.11a, IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, July 1999.
- [2] R.W. Chang, "Synthesis of Band-limited Orthogonal Signals for Multichannel Data Transmission," *Bell Syst. Tech. J.*, vol. 46, pp. 1775-1796, December 1966.
- [3] S.B. Weinstein and P.M. Ebert, "Data Transmission by Frequency Division Multiplexing Using the Discrete Fourier Transform," *IEEE Trans. Commun. Technol.*, vol. COM-19, pp. 628-634, October, 1971.
- [4] A. Peled and A. Ruiz, "Frequency Domain Data Transmission Using Reduced Computational Complexity Algorithms," *Proc. ICASSP*, Denver, CO, pp. 964-967, April, 1980.
- [5] D.Z. Liu, *Synchronization and Channel Compensation Techniques for Digital Mobile and Personal Communication Systems*, Doctor Thesis, National Chiao Tung University, 2001.
- [6] T.M. Schmidl and D.C. Cox, "Robust Frequency and Timing Synchronization for OFDM," *IEEE Trans. Communications*, vol. 45, No. 12, pp. 1613-1621, December 1997.
- [7] M. Luise and R. Reggiannini, "Carrier Frequency Acquisition and Tracking for OFDM Systems," *IEEE Trans. Communications*, vol. 44, pp. 1590-1598, November 1996.
- [8] J.-J. van de Beek, M. Sandell, and P.O. Borjesson, "ML Estimation of Time and Frequency Offset in OFDM Systems," *IEEE Trans. Signal Processing*, vol. 45, No. 7, pp. 1800-1805, July 1997.
- [9] M. Sandell, J.-J. van de Beek, and P.O. Borjesson, "Timing and Frequency Synchronization in OFDM Systems Using the Cyclic Prefix," *Proc. Int. Symp. Synchronization*, Essen, Germany, pp. 16-19, December 1995.

- [10] T. Keller, L. Piazzo, P. Mandarini and L. Hanzo, "Orthogonal Frequency Division Multiplex Synchronization Techniques for Frequency-Selective Fading Channels," *IEEE Journal on Selected Areas in Communications*, vol. 19, No. 6, pp. 999-1007, June 2001.
- [11] D.Z. Liu, C-H Wei and C.J. Chang, "An Extension of Guard-Interval Based Symbol and Frequency Synchronization Technique for Wireless OFDM Transmission," *Proc. IEEE Int. Conf. Vehicular Technology*, vol. 4, pp. 2324-2328, October 2001.
- [12] J.-J. van de Beek, M. Sandell, M. Isaksson and P.O. Borjesson, "Low Complex Frame Synchronization in OFDM Systems," *Proc. IEEE Int. Conf. Universal Personal Communication*, Toronto, Canada, pp. 982-986, September 1995.
- [13] M.H. Hsieh and C.H. Wei, "A Low-Complexity Frame Synchronization and Frequency Offset Compensation Scheme for OFDM Systems over Fading Channels," *IEEE Trans. Vehicular Technology*, vol. 48, No. 5, pp. 1596-1609, September 1999.
- [14] S.T. Lin and C.H. Wei, "Precursor ISI-Free Frame Synchronization for DMT VDSL System," *IEICE Trans. Communication*, vol. E85-B, No. 8, pp. 1447-1454, August 2002.
- [15] C.R.N. Athaudage and V. Krishnamurthy, "A Low Complexity Timing and Frequency Synchronization Algorithm for OFDM Systems," *Proc. IEEE Int. Conf. Globecom.*, Taipei, Taiwan, vol. 1, pp. 244-248, November 2002.
- [16] T.H. Lee and K.W. Jhan, "Performance Evaluation for a Class of Synchronization Techniques in OFDM Systems," *Int. Symp. Wireless Personal Multimedia Communications*, Yokosuka, Kanagawa, Japan, October 2003.
- [17] T. Pollet, P. Spruyt, and M. Moeneclaey, "The BER Performance of OFDM System Using Non-synchronized Sampling," *Proc. IEEE Int. Conf. Globecom.*, San Francisco, CA, vol. 1, pp. 253-257, November 1994.
- [18] R. Nee and R. Prasad, *OFDM for Wireless Multimedia Communications*, Artech House, 2000.
- [19] A.V. Oppenheim, R.W. Schaffer and J.R. Buck, *Discrete-Time Signal Processing*, 2nd edition. Prentice-Hall, 1999.

- [20] J. Terry and J. Heiskala, *OFDM Wireless LANs: A Theoretical and Practical Guide*, SAMS, 2002.
- [21] P.H. Moose, "A Technique for Orthogonal Frequency Division Multiplexing Frequency Offset Correction," *IEICE Trans. Communication*, vol. 42, pp. 2908-2914, October 1994.
- [22] T.M. Lin, *Algorithms and Performance Evaluation of Frame Detector and Frequency Estimator for OFDM Wireless LAN Systems*, Master Thesis, National Chiao Tung University, 2002.
- [23] R.Y. Jung, *Carrier Frequency Offset Estimation for OFDM-Based WLANs*, Master Thesis, National Chiao Tung University, 2002.
- [24] W.T. Huang, *Performance Evaluation of the IEEE 802.11a OFDM-based Wireless Local Area Networks*, Master Thesis, National Chiao Tung University, 2002.
- [25] M.C. Chen, *Design of 802.11a Baseband Transmitter and Synchronization*, Master Thesis, National Chiao Tung University, 2003.
- [26] C.L. Huang, *Unified Datapath Design for Inner Baseband Receiver for OFDM-Based Wireless Communication System*, Master Thesis, National Chiao Tung University, 2002.
- [27] J.G. Proakis, *Digital Communications*, 4th edition. McGraw-Hill, 2001.
- [28] S. Nakamura, *Numerical Analysis and Graphic Visualization with MATLAB*, 2nd edition. Prentice-Hall, 2002.
- [29] W.C. Jakes, *Microwave Mobile Communications*. IEEE Press, 1994.
- [30] Y. Li and Y.L. Guan, "Modified Jakes' Model for Simulating Multiple Uncorrelated Fading Waveforms," *Proc. IEEE Int. Conf. Vehicular Technology.*, Tokyo, Japan, vol. 3, pp. 1819-1822, May 2000.
- [31] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 2nd edition, McGraw-Hill, 1984.