



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**EXTENDING THE RANGE OF THE 802.11G WLAN
THROUGH IMPROVED SYNCHRONIZATION
TECHNIQUES**

by

Vikram Sardana

June 2007

Thesis Advisor:
Second Reader:

Frank E. Kragh
R. Clark Robertson

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE June 2007	3. REPORT TYPE AND DATES COVERED Master's Thesis		
4. TITLE AND SUBTITLE: Extending the Range of the 802.11g WLAN through Improved Synchronization Techniques			5. FUNDING NUMBERS	
6. AUTHOR(S) Sardana, Vikram				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (maximum 200 words) Orthogonal Frequency Division Multiplexing (OFDM) allows for a spectrally efficient means of obtaining high data rates while simultaneously combating the effects of fading. The multi-carrier spectrum of OFDM mandates that the receiver accomplish a number of synchronization tasks to successfully demodulate the OFDM signal, including the critical requirement to synchronize the carrier frequency. Additional synchronization tasks include frame synchronization (packet detection), synchronization of the carrier phase, and symbol timing. Improved receiver synchronization algorithms may hold the prospect of superior performance; specifically allowing successful demodulation by the receiver at an extended range. This thesis discusses several promising synchronization algorithms. Furthermore, a performance analysis of these algorithms is conducted at low signal to noise ratio (SNR) in an AWGN channel using MATLAB.				
14. SUBJECT TERMS OFDM, Orthogonal Frequency Division Multiplexing, 802.11, 802.11g, AWGN, OFDM, Synchronization, Carrier Synchronization, Frequency Synchronization, Phase Synchronization, Symbol Synchronization, FFT, IFFT, Cyclic Prefix, Guard Interval, wireless LAN			15. NUMBER OF PAGES 151	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18

THIS PAGE LEFT INTENTIONALLY BLANK

Approved for public release; distribution is unlimited

**EXTENDING THE RANGE OF THE 802.11G WLAN THROUGH IMPROVED
SYNCHRONIZATION TECHNIQUES**

Vikram Sardana
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1989
MSAE, Naval Postgraduate School, 1997

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

**NAVAL POSTGRADUATE SCHOOL
June 2007**

Author: Vikram Sardana

Approved by: Frank E. Kragh
Thesis Advisor

R. Clark Robertson
Second Reader

Jeffrey B. Knorr
Chairman, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Orthogonal Frequency Division Multiplexing (OFDM) allows for a spectrally efficient means of obtaining high data rates while simultaneously combating the effects of fading. The multi-carrier spectrum of OFDM mandates that the receiver accomplish a number of synchronization tasks to successfully demodulate the OFDM signal, including the critical requirement to synchronize the carrier frequency. Additional synchronization tasks include frame synchronization (packet detection), synchronization of the carrier phase, and symbol timing. Improved receiver synchronization algorithms may hold the prospect of superior performance; specifically allowing successful demodulation by the receiver at an extended range. This thesis discusses several promising synchronization algorithms. Furthermore, a performance analysis of these algorithms is conducted at low signal to noise ratio (SNR) in an AWGN channel using MATLAB.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	BACKGROUND	1
B.	THESIS OBJECTIVE	1
C.	RELATED WORK	2
D.	THESIS ORGANIZATION.....	2
II.	BASICS OF ORTHOGONAL FREQUENCY-DIVISION MULTIPLEXING (OFDM).....	3
A.	THE PROBLEM OF FADING AND HIGH DATA RATES	3
B.	FREQUENCY-DIVISION MULTIPLEXING AND DATA RATES	4
C.	SPECTRALLY EFFICIENT SPACING FOR SUB-CARRIERS	5
D.	INVERSE DISCRETE FOURIER TRANSFORM KERNEL FOR OFDM	8
III.	SYNCHRONIZATION VULNERABILITIES OF OFDM	13
A.	INTER-CARRIER INTERFERENCE	13
1.	Local Oscillator Offset.....	15
2.	Phase Noise	18
3.	Doppler Shift	24
4.	Summary.....	28
B.	INTER-SYMBOL INTERFERENCE	29
1.	Long Delay Spreads and ISI	29
2.	Symbol Timing Offset and ISI.....	31
3.	Summary.....	36
IV.	SYNCHRONIZATION OF THE OFDM SIGNAL.....	37
A.	INTRODUCTION.....	37
1.	OFDM Synchronization and Single Carrier Synchronization	37
2.	Synchronization for WLANs and Broadcast Systems.....	38
3.	Synchronization Methodology	38
a.	Non Data-Aided Synchronization	39
b.	Cyclic Prefix Based Synchronization.....	39
c.	Data-Aided Synchronization.....	40
B.	IEEE 802.11G PACKET STRUCTURE AND SYNCHRONIZATION...40	40
1.	Physical Layer Convergence Protocol Receive Procedure.....	40
2.	Short Training Symbols	42
3.	Long Training Symbols	45
4.	Pilot Tones	46
5.	Summary.....	47
C.	IEEE 802.11G PACKET DETECTION	49
1.	Introduction and Link Budget Analysis	49
2.	Packet Detection Theory	52
a.	Minimizing Error in Packet Detection.....	55
b.	Neyman-Pearson Criterion.....	56

c.	<i>Using Multiple Samples to Facilitate Packet Detection</i>	60
3.	Delay and Correlate Packet Detection	64
4.	Matched Filter Detector	69
5.	Summary and Comparison of Packet Detection Techniques	74
D.	FREQUENCY SYNCHRONIZATION	75
1.	Introduction.....	75
a.	<i>Single and Multi-Carrier Frequency Synchronization Differences</i>	76
b.	<i>General Frequency Synchronization Technique in OFDM</i>	76
c.	<i>Summary of Various Synchronization Algorithms</i>	77
2.	Maximum Likelihood Synchronization	77
a.	<i>Limits on the Operating Range of the ML Estimator</i>	80
b.	<i>Implementation of the ML Estimator</i>	81
3.	Non-linear Least Squares Synchronization.....	84
a.	<i>Implementation of the NLS Estimator</i>	87
4.	Summary of Frequency Synchronization Algorithms.....	89
V.	PERFORMANCE OF THE SYNCHRONIZATION ALGORITHMS.....	91
A.	INTRODUCTION.....	91
B.	PACKET DETECTION PERFORMANCE	91
1.	Delay and Correlate Packet Detection Performance.....	92
a.	<i>Improved Performance from the Delay and Correlate Detector</i>	94
2.	Matched Filter Packet Detection Performance.....	96
a.	<i>Improved Performance from the Matched Filter Detector</i> ..	97
3.	Summary of Packet Detection Performance	99
C.	FREQUENCY SYNCHRONIZATION PERFORMANCE	100
1.	Maximum Likelihood Frequency Synchronization Performance	100
a.	<i>Robustness of the ML Frequency Correction Algorithm</i>	103
2.	Non-linear Least Squares Frequency Synchronization Performance	104
a.	<i>Robustness of the NLS Frequency Correction Algorithm</i> ..	107
3.	Summary of Frequency Synchronization Performance.....	108
VI.	CONCLUSION	111
A.	RESULTS OF THE PERFORMANCE ANALYSIS	111
B.	RECOMMENDATIONS FOR FOLLOW ON RESEARCH.....	113
	APPENDIX	115
	LIST OF REFERENCES.....	127
	INITIAL DISTRIBUTION LIST	131

LIST OF FIGURES

Figure 1	Frequency Division Multiplexing.....	5
Figure 2	Orthogonal Frequency Division Multiplexing.....	6
Figure 3	Depiction of ICI – from Reference [20]	13
Figure 4	Reception without ICI.....	14
Figure 5	Sub-carrier reception with ICI	15
Figure 6	SNR degradation due to relative frequency error, $\Delta f_c T_s$	17
Figure 7	Single-sided LO PSD with Phase Noise ($\beta = 1\text{Hz}$)	22
Figure 8	SNR degradation caused by phase noise	23
Figure 9	Doppler shift due to relative motion between RX and TX	25
Figure 10	Guard interval and channel response – from Reference [20]	31
Figure 11	Ideal FFT window timing	32
Figure 12	3 sub-carriers in FFT interval – no ICI present.....	33
Figure 13	3 sub-carriers with ICI caused by sub-carrier delay	34
Figure 14	3 sub-carriers with CP added to delayed sub-carrier	35
Figure 15	Early vs. Late FFT window timing and ISI	36
Figure 16	PLCP receive procedure – from Reference [4].....	41
Figure 17	OFDM preamble and header structure – from Reference [4]	42
Figure 18	Coefficient mapping within the IDFT – from Reference [4].....	43
Figure 19	Short training symbols (time domain, in-phase).....	44
Figure 20	Long training symbols (time domain, in-phase).....	46
Figure 21	Time-frequency structure of OFDM 802.11g packet – from Reference [20] ..	48
Figure 22	Power received as a function of distance.....	51
Figure 23	Probabilities of False Alarm and Detection in an AWGN channel	53
Figure 24	Signal threshold given hypothesis H_0 and H_1	59
Figure 25	Signal flow structure of the Delay and Correlate algorithm – from Reference [8].....	65
Figure 26	Delay and Correlate algorithm using 802.11g preamble and showing C_M , P_M , X	66
Figure 27	Detection Threshold and P_{fa} for Delay and Correlate Detector when $P_d = .9$	69
Figure 28	Matched Filter Detector – from Reference [13]	70
Figure 29	Decision statistic output for Matched Filter Detector.....	71
Figure 30	Matched Filter y_T and P_{fa} for varying SNR.....	74
Figure 31	Frequency offset synchronization using the ML algorithm.....	82
Figure 32	ML Frequency correction for 125 kHz offset (10 dB SNR).....	83
Figure 33	ML Frequency Synchronizer	84
Figure 34	Frequency offset correction using NLS algorithm.....	88
Figure 35	NLS Frequency Synchronizer.....	89
Figure 36	ROC curves for Delay and Correlate Detector	93
Figure 37	32 sample Matched Filter Detector.....	98

Figure 38	Remaining Frequency offset using ML algorithm (10 and 6 dB SNR) using 10 short training symbols	101
Figure 39	Remaining frequency offset percentage using ML algorithm at 3 dB SNR ..	103
Figure 40	Remaining Frequency offset percentage using ML algorithm and reduced training symbols.....	104
Figure 41	Remaining Frequency offset percentage using NLS algorithm using 10 short and 2 long training symbols.....	106
Figure 42	Remaining Frequency offset percentage using NLS algorithm and reduced training symbols.....	107
Figure 43	Comparison of remaining Frequency offset percentage at 3 dB SNR with 4 short and 2 long training symbols.....	109

LIST OF TABLES

Table 1	Typical Values of RMS Delay Spread – from Reference [1]	30
Table 2	Receiver Performance Requirements – from Reference [4]	49
Table 3	Threshold range for evaluated SNR.....	92
Table 4:	P_d and P_{fa} for 80 sample Delay and Correlate Detector	95
Table 5:	P_d and P_{fa} for Matched Filter Detector	96
Table 6:	P_d and P_{fa} for Improved Matched Filter Detector.....	98

THIS PAGE LEFT INTENTIONALLY BLANK

EXECUTIVE SUMMARY

Orthogonal frequency-division multiplexing (OFDM) is a digital communication technique that employs the simultaneous transmission of multiple sub-carriers at low data rate to achieve an overall high data rate in a spectrally efficient manner. While using low data rate sub-carriers is effective in multi-path environments and in combating frequency selective fading, OFDM systems are sensitive to loss of synchronization. Lack of synchronization in frequency or time results in a noticeable degradation in performance of the OFDM communication system. Lack of frequency synchronization occurs due to oscillator mismatch, phase noise, and Doppler shift, resulting in loss of orthogonality between the sub-carriers. Lack of timing synchronization results in missed signals due to failure to detect and then properly demodulate the arriving packet. To maintain synchronization with the transmitter, the OFDM receiver must therefore detect the packet (packet or frame synchronization), estimate and correct any carrier frequency error, perform phase synchronization and achieve accurate symbol timing.

In this thesis, the sensitivity of OFDM to lack of synchronization is examined, and several synchronization algorithms are evaluated for a commercial, widespread application of OFDM –IEEE 802.11g Wireless Local Area Networks (WLANs). The governing 802.11 IEEE standards do not promulgate the type of algorithms that commercial vendors may use to obtain and maintain synchronization with the transmitted signal. Instead, training symbols and pilots are mandated by the IEEE standard as part of the 802.11g packet, and each vendor may implement the most efficient algorithm to achieve the best performance, in terms of reception range and data rate. Vendors who leverage these mandated symbols and pilots employ data-aided synchronization, and because the training symbols are located at the start of the 802.11g packet, this allows for quick synchronization in a WLAN environment. The underlying premise of this thesis is that the use of improved synchronization algorithms will result in greater performance – including greater detection range of the OFDM signal. In the following pages, no direct comparison is made to any commercial synchronization algorithms due to their proprietary nature – instead promising algorithms that result in detection of the 802.11g

packet at low signal-to-noise ratio (SNR) to simulate extended range are examined. Additionally, promising frequency synchronization algorithms are evaluated as a necessary step to successfully demodulate the OFDM signal in the low SNR environment. Successful detection and frequency synchronization of the OFDM signal in a low SNR environment points the way to follow on research in which a direct comparison between the promising algorithms and current commercial synchronization algorithms can be made using commercial-off-the-shelf (COTS) equipment.

The synchronization algorithms examined in this thesis are designed in MATLAB and then subject to a performance evaluation using multiple test signals at low signal-to-noise ratio to simulate detection and carrier frequency estimation and correction at extended range. Due to the necessity to synchronize quickly in a WLAN environment and because the training symbols must be transmitted in accordance with the IEEE standard, the evaluated algorithms are data-aided and focus on the training symbols found at the beginning of the 802.11g packet.

LIST OF SYMBOLS, ACRONYMS, AND ABBREVIATIONS

W	Bandwidth
R	Data rate
T	Symbol duration
R_b	Sub-carrier data rate
σ	Delay spread
σ_{RMS}	RMS Delay spread
μ	micro ($1 \cdot 10^{-6}$)
η	nano ($1 \cdot 10^{-9}$)
N	# of Sub-carriers
T_b	Sub-carrier bit duration
T_s	OFDM symbol duration
R_s	OFDM symbol rate
A	A constant, proportional to signal amplitude
i	Sub-carrier index
ω_i	Center frequency for sub-carrier i
ω_{Δ}	Sub-carrier spacing (in radians)
d	OFDM symbol values, modulation dependent
I	In-phase component
Q	Quadrature component
P_{T_s}	OFDM symbol windowing function
j	$\sqrt{-1}$
C	Correlation metric
T_{gi}	CP duration
T_{total}	OFDM symbol and CP duration
β	One-sided 3 dB linewidth of an oscillator

ϕ	Phase error
Φ	Average phase error
Δf	Deviation in center frequency
f_0	First sub-carrier frequency (in Hz)
Δf_c	Carrier frequency offset
c	Speed of light
v	Speed of relative motion between transmitter and receiver
f	Frequency
f_c	Carrier frequency
Δf_D	Change in frequency due to Doppler shift
ξ	Percentage change in frequency due to Doppler shift
T	Sampling period
t_{1-10}	Short training symbol
$T_{1 \text{ or } 2}$	Long training symbol
S	Short training sequence
L	Long training sequence
M_{sc}	Sub-carrier remapping function for pilot tones
PL	RF path loss
D	Distance, in meters
γ	Path loss exponent
H_0	Noise only hypothesis
H_1	Noise and packet hypothesis
$r(t)$	Received signal
$n(t)$	Received noise
$s(t)$	Received packet
D_0	Receiver decision for noise only hypothesis
D_1	Receiver decision for noise plus packet hypothesis

$Pr\{f g\}$	Conditional probability (for random variable f, given g, for example)
p	Probability density function curve
Λ	Neyman-Pearson likelihood ratio
y_T	Decision threshold
χ_m	Individual sample decision statistic (Gaussian random variable)
m_χ	Mean of random decision statistic χ_m
σ_χ^2	Variance of decision statistic χ_m
s_m	Packet component of received signal at some time $t = m$
σ_n^2	Band limited noise variance
X	Multiple sample decision statistic
m_X	Mean of multiple sample decision statistic
σ_X^2	Variance of multiple sample decision statistic
σ_s^2	Variance of the short training samples
Pr_d	Probability of detection
Pr_{fa}	Probability of false alarm
λ_T	Neyman-Pearson required threshold (dependent upon false alarm prob.)
M	Number of samples used in packet detection
τ	Propagation delay
$s_l(t)$	Equivalent low-pass received packet
f_{tx}	Transmitted carrier frequency
f_{rx}	Receiver center frequency
Δf	Frequency difference caused by oscillator mismatch
r_m	Received signal (complex baseband)
z	Frequency offset intermediate variable
t_s	Sample duration
$\Delta \hat{f}$	Frequency offset error

f_s	Sampling rate
$\angle$	Angle operator for complex numbers
M_s	Number of short training samples (16) in one symbol
M_L	Number of long training samples in one symbol (64)
WLAN	Wireless Local Area Network
MAC	Medium access control
PHY	Physical layer
ISM	Industrial, Scientific, and Medical
PDF	Probability Density Function
BLAWGN	Band limited Additive White Gaussian Noise
DAC	Digital-to-analog converter
RSSI	Received Signal Strength Indication
PLCF	Physical Layer Convergence Function
PLCP	Physical Layer Convergence Protocol
PDSU	PLCP Data Service Unit
MPDU	Medium Access Control Protocol Data Unit
OFDM	Orthogonal Frequency Division Multiplexing
BPSK	Binary Phase Shift Keying
QAM	Quadrature Amplitude Modulation
BER	Bit-error-rate
ML	Maximum likelihood
NLS	Non-linear least squares

I. INTRODUCTION

A. BACKGROUND

Wireless Local Area Networks (WLANs) are rapidly proliferating throughout commercial, governmental, and educational organizations. To satisfy increased consumer expectations, including multimedia content, engineers must strive to deliver the greatest data rates possible, using robust methodologies, while minimizing costs and bandwidth. Unfortunately, as data rates increase, wireless transmissions encounter increasing problems with fading [1]. To combat fading problems at high data rate, multi-carrier modulation is used. Multi-carrier modulation is based on the parallel transmission of data using sub-carriers and can be done in a spectrally efficient manner using orthogonal frequency-division multiplexing (OFDM) [2]. Consequently, OFDM has become the modulation of choice for many wireless networks.

The relevant governing standard is the IEEE 802.11 standard [3], including the IEEE 802.11a Medium Access Control (MAC) and Physical Layer (PHY) specification [4] and the IEEE 802.11g amendment governing the further higher data rate extension in the 2.4 GHz unlicensed Industrial, Scientific and Medical (ISM) band [5]. The IEEE 802.11g standard allows data rates of up to 54 Mbps (million bits per second) at strong signal to noise ratio (SNR) [5] but the range of these WLANs is limited due to imposed power constraints [5]. Performance of the OFDM system often is also impeded by sensitivity to loss of synchronization between the transmitter and receiver [6].

B. THESIS OBJECTIVE

The goal of this thesis is to analyze promising synchronization techniques that can be used to extend the range and performance of the 802.11g WLAN. These promising synchronization techniques are evaluated in a low SNR environment to simulate reception of the OFDM signal at extended range. If successful detection of the signal is accomplished then the packet detection synchronization algorithm is judged to be promising and may be reserved for future analysis in direct comparison with existing

commercial synchronization algorithms. The same standard is applied to promising frequency synchronization algorithms, a necessary step in the successful demodulation of the OFDM signal.

C. RELATED WORK

The OFDM system is vulnerable to synchronization errors [7]. At long range, and consequently low signal to noise ratio (SNR), the OFDM receiver must reliably detect the incoming packet, and synchronization is the first task performed [8]. Moreover, lack of frequency synchronization between transmitter and receiver can contribute to lower signal strength and destructive interference from adjacent carriers at the receiver, further reducing performance [9].

The use of multi-carriers allows the OFDM signal to achieve high data rates in the presence of multi-path propagation of the transmitted signal, but also exposes the OFDM signal to vulnerabilities with respect to frequency offset and phase noise. This loss of frequency synchronization leads to a phenomenon known as inter-carrier interference (ICI) [8]. Additionally, the multi-carrier system must address packet detection and symbol timing in the context of low SNR at extended range. Failure to properly synchronize in time with the OFDM symbol will result in inter-symbol interference (ISI) [8]. Therefore, to achieve optimum performance, the OFDM receiver must do both – time and frequency synchronization. The IEEE standard does not specify the specific synchronization techniques to be used, so a variety of time and frequency synchronization techniques will be analyzed at low SNR using MATLAB.

D. THESIS ORGANIZATION

Chapter II reviews the problems of fading in the presence of multi-path propagation, OFDM signal generation and various critical parts of the OFDM signal structure. Chapter III addresses in detail the difficulties of receiving the OFDM signal in the presence of ISI and ICI. Chapter IV comparatively analyzes various time and frequency synchronization techniques. Finally, Chapter V reviews analyzed data and provides a summary of results.

II. BASICS OF ORTHOGONAL FREQUENCY-DIVISION MULTIPLEXING (OFDM)

A. THE PROBLEM OF FADING AND HIGH DATA RATES

Fading of the transmitted signal occurs at the receiver when there is movement between transmitter and receiver (Doppler shift) and multi-path propagation. Multi-path effects are a natural phenomenon caused by reflectors and scatterers within the wireless channel. Multiple “copies” of the original transmitted symbol arrive at the wireless receiver. These “copies” or echoes then add destructively or constructively with the original transmitted symbol. [1]

The delay between the first reception of the transmitted symbol and the last echo is known as the delay spread of a particular channel and simply represents the time difference between the shortest (usually direct line of sight) and longest transmission path [10]. Equation (2.1) shows the relationship between delay spread (σ) and difference in transmission distance (Δd):

$$\Delta d = \sigma c \quad (2.1)$$

where c is the speed of light. Due to longer transmission distances, typical RMS values of delay spread in an outdoor environment are on the order of microseconds while indoor environments have average values of 100 nanoseconds but could exceed 1500 nanoseconds in a worst case situation [1].

A smaller delay spread is advantageous. When the delay spread is a significant fraction of the transmitted symbol duration, the multi-path effects can substantially interfere with the follow-on symbol. In this case of “inter-symbol interference”, the received pulse is unevenly distorted across its bandwidth, a situation known as frequency selective fading. [1]

Inter-symbol interference (ISI) is exacerbated by high data rates required by multimedia content. This is because increased data rates are often implemented by using

shorter duration symbols, resulting in greater symbol bandwidth [1]. If, for example, binary phase shift keying (BPSK) is the preferred modulation in an indoor wireless transmission, then:

$$W = \frac{1}{R} = T \quad (2.2)$$

where W is the signal equivalent noise bandwidth in Hz, R is the data rate in bits per second, and T is the bit duration in seconds. If 20 Mbps data rate is desired, 20 MHz bandwidth is required and bit duration is 50 nanoseconds. In this case bit duration is less than average delay spread values indoors therefore such a communications link would experience substantial ISI. To reduce ISI, the solution is either to reduce the data rate (increasing bit duration) [1], to provide channel equalizers to mitigate the ISI, or switch to an M – ary type of modulation [11]. Channel equalizers result in increased receiver complexity, M – ary modulation requires higher SNR per bit [11], and increasing bit duration (thereby decreasing bandwidth and decreasing data rate) sufficiently to eliminate ISI, would require bit durations much larger than the RMS values of delay spread. If, in a particular indoor channel, $\sigma_{RMS} = 100 \times 10^{-9}$, then to effectively mitigate ISI for a binary modulation scheme, bit duration must adhere to the following rule of thumb: [1]

$$T \geq 10\sigma_{RMS} \quad (2.3)$$

In such a case, bit duration can be no less than 1000 nanoseconds implying a maximum data rate of 1 Mbps vice the desired 20 Mbps. Thus, ISI imposes an upper limit on symbol rate and consequently on data rate for BPSK and various other modulations.

B. FREQUENCY-DIVISION MULTIPLEXING AND DATA RATES

In our previous example, if we instead use N sub-carriers vice a single carrier, and transmit data in parallel on all of the sub-carriers, the sub-carrier bit duration is now:

$$T_b = NT \quad (2.4)$$

The resulting increase in sub-carrier bit duration reduces individual sub-carrier data rates by a factor of N but the parallel transmission of all sub-carriers' signals yields

the original desired data rate. Thus each sub-carrier's signal experiences negligible ISI in a multi-path environment [12]. IEEE 802.11g and several other wireless standards utilize this multi-carrier technique to solve the ISI/frequency selective fading problem.

C. SPECTRALLY EFFICIENT SPACING FOR SUB-CARRIERS

Multi-carrier modulation (based on multiple carriers, called sub-carriers) is essentially frequency division multiplexing (FDM) [13]. In order for this scheme to work effectively, so that there is no interference between sub-carriers, the sub-carriers must be spaced apart in frequency. Figure 1 depicts a scheme that has very low interference between sub-carriers, but uses too much bandwidth, i.e. it is not spectrally efficient.

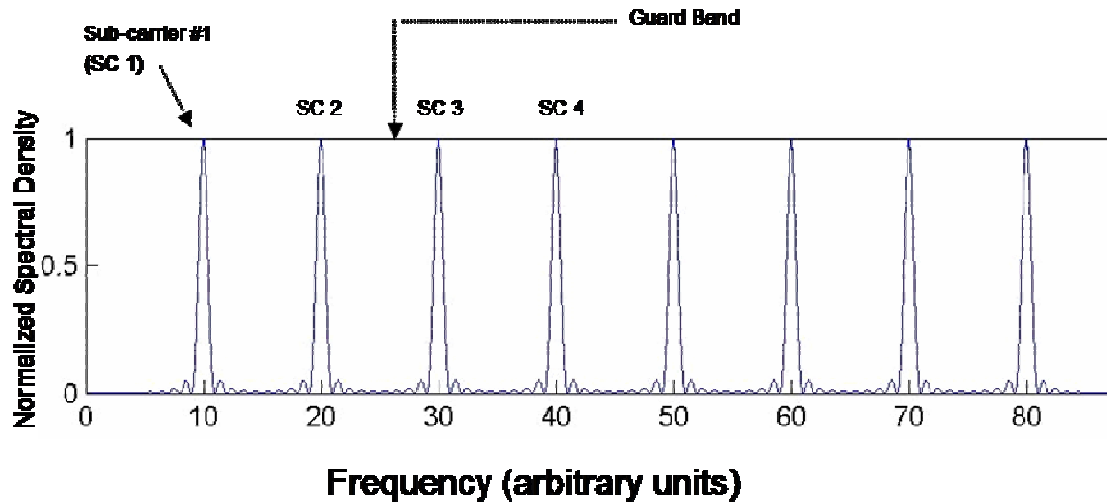


Figure 1 Frequency Division Multiplexing

With respect to spectral efficiency, there is a better way to manage the allotted bandwidth. If we use Orthogonal Frequency Division Multiplexing (OFDM), there would be significant savings in bandwidth while still maintaining our lower bit rate per sub-carrier. The resulting power spectral density (PSD) for the same eight sub-carriers depicted earlier would look like Figure 2. The increased spectral efficiency compared to simple frequency division multiplexing (previous figure) is obvious.

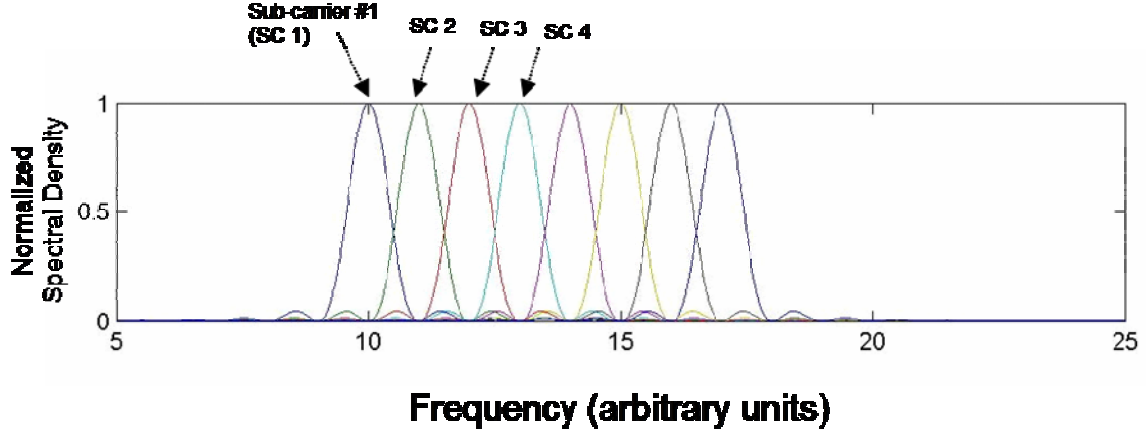


Figure 2 Orthogonal Frequency Division Multiplexing

Note that in Figure 2, each sub-carrier peak coincides with nulls in the power spectral density of all other sub-carriers. This is a practical result of orthogonality. In OFDM, orthogonality is maintained between the sub-carrier channels by properly spacing the sub-carrier center frequencies. Each sub-carrier channel's signal is a quadrature amplitude modulated (QAM) signal given as [2]

$$s_i(t) = Ad_{I_i}(t)\cos(\omega_i t) - Ad_{Q_i}(t)\sin(\omega_i t) \quad (2.5)$$

where A is a constant, proportional to signal amplitude, ω_i is the center frequency for sub-channel i , and

$$d_{I_i}(t) = \sum_{n=-\infty}^{n=+\infty} d_{I_{i,n}} P_{T_s}(t - nT_s). \quad (2.6)$$

The value of $d_{I_{i,n}}$ can take on the values $\pm 1, \pm 3, \pm 5, \dots, \pm \sqrt{M} - 1$; M is the number of symbols for a given modulation (M -QAM); T_s is the symbol duration;

$$P_{T_s}(t) = \begin{cases} 1 & \text{if } 0 \leq t \leq T_s \\ 0 & \text{otherwise;} \end{cases} \quad (2.7)$$

and the definition of $d_{Q_i}(t)$ is analogous to that of $d_{I_i}(t)$ as the quadrature component.

Using the trigonometric identity [14]:

$$\cos(\alpha + \beta) = \cos(\alpha) \cos \beta - \sin(\alpha) \sin \beta \quad (2.8)$$

$s_i(t)$ may be re-written as:

$$s_i(t) = a_i(t) \cos[\omega_i t + \theta_i(t)] \quad (2.9)$$

Where $a_i(t) = A\sqrt{d_{I,i}^2(t) + d_{Q,i}^2(t)}$ and $\theta_i(t) = \tan^{-1}\left[\frac{d_{Q,i}(t)}{d_{I,i}(t)}\right]$. To maintain

orthogonality between sub-carriers, the correlation between the sub-carriers must be zero.

The correlation between signals on sub-carriers p and q is

$$\begin{aligned} C_{p,q} &= \int_{-\infty}^{+\infty} s_p(t) s_q(t) dt = \int_{-\infty}^{+\infty} a_p(t) \cos[\omega_p t + \theta_p(t)] a_q(t) \cos[\omega_q t + \theta_q(t)] dt \\ &= \sum_{n=-\infty}^{n=+\infty} \int_{nT_s}^{(n+1)T_s} a_p(t) \cos[\omega_p t + \theta_p(t)] a_q(t) \cos[\omega_q t + \theta_q(t)] dt \\ &= \sum_{n=-\infty}^{n=+\infty} \int_{nT_s}^{(n+1)T_s} a_{p,n} \cos[\omega_p t + \theta_{p,n}] a_{q,n} \cos[\omega_q t + \theta_{q,n}(t)] dt \end{aligned} \quad (2.10)$$

where $a_{p,n} = A\sqrt{d_{I,p,n}^2 + d_{Q,p,n}^2}$, $\theta_{p,n} = \tan^{-1}\left[\frac{d_{Q,p,n}}{d_{I,p,n}}\right]$, and $d_{I,p,n}$ and $d_{Q,p,n}$ are the respective

in-phase and quadrature OFDM symbol values of the p^{th} sub-carrier over the interval

$nT_s \leq t < (n+1)T_s$. Therefore, using the following trigonometric identity [14],

$\cos \alpha \cos \beta = \frac{1}{2} [\cos(\alpha + \beta) + \cos(\alpha - \beta)]$, the correlation may be re-written as

$$C_{p,q} = \frac{1}{2} \sum_{n=-\infty}^{n=+\infty} a_{p,n} a_{q,n} \int_{nT_s}^{(n+1)T_s} \cos[(\omega_p + \omega_q)t + (\theta_{p,n} + \theta_{q,n})] + \cos[(\omega_p - \omega_q)t + (\theta_{p,n} - \theta_{q,n})] dt \quad (2.11)$$

if $(\omega_i + \omega_j) \gg \frac{1}{T_s}$, as is typically the case, then

$$C_{p,q} \approx \frac{1}{2} \sum_{n=-\infty}^{n=+\infty} a_{p,n} a_{q,n} \int_{nT_s}^{(n+1)T_s} \cos[(\omega_p - \omega_q)t + (\theta_{p,n} - \theta_{q,n})] dt \quad (2.12)$$

$= 0$ if $(\omega_p - \omega_q)T_s = 2\pi k$, where $k \in$ set of integers.

Considering adjacent sub-carriers, letting $k=1$ for closest spacing, and thereby maximizing spectral efficiency,

$$f_{p+1} - f_p = \frac{\omega_{p+1}}{2\pi} - \frac{\omega_p}{2\pi} = \frac{1}{T_s} = R_s$$

The above result shows that to achieve orthogonality between the sub-carriers, while maximizing spectral efficiency, the sub-carrier center frequency spacing must be equal to the OFDM symbol rate.

D. INVERSE DISCRETE FOURIER TRANSFORM KERNEL FOR OFDM

To properly transmit the OFDM symbols in a spectrally efficient manner, as discussed in the previous section, it is essential to create and maintain orthogonality between the multiple sub-carriers. The previous section demonstrated that the greatest spectral efficiency that could be achieved while maintaining orthogonality resulted when sub-carrier spacing in frequency was equal to the sub-carrier symbol rate, R_s . The most obvious solution might be to have a transmitter and receiver each having N local oscillators. Each oscillator would tune an assigned individual sub-carrier frequency and the oscillators would be separated in frequency by an amount equal to the sub-carrier symbol rate. Such a scheme, however, would be unreasonably expensive and complex [2]. The inevitable relative drift in oscillator frequencies would result in non-zero correlation between the sub-carriers, destroying orthogonality, and resulting in ICI.

There is a superior method that ensures that all oscillator frequencies are precisely R_s apart, thereby maintaining orthogonality between the N sub-carriers while maintaining spectral efficiency. This method involves using the Fourier transform [15]. Consider the resulting signal on the i^{th} sub-carrier,

$$s_i(t) = Ad_{I,i}(t)\cos(\omega_i t) - Ad_{Q,i}(t)\sin(\omega_i t) \quad (2.13)$$

where A is a constant, proportional to signal amplitude, ω_i is the i^{th} sub-carrier's center frequency, and $d_{Q,i}(t)$ and $d_{I,i}(t)$ are the respective quadrature and in-phase pulse amplitude modulated signals of the i^{th} sub-carrier for a given M -ary modulation. Alternatively, Equation (2.13) can be more conveniently written as

$$s_i(t) = A \operatorname{Re} \left\{ \left[d_{I,i}(t) + j d_{Q,i}(t) \right] e^{j\omega_i t} \right\}. \quad (2.14)$$

We can further simplify the representation of $s_i(t)$ by recognizing from Equation (2.6) that

$$d_{I,i}(t) + j d_{Q,i}(t) = \sum_{n=-\infty}^{+\infty} d_i(n) P_{T_s}(t - nT_s), \quad (2.15)$$

where $P_{T_s}(t)$ is the rectangular pulse function defined in Equation (2.7) and $d_i(n) \equiv d_{I,i,n} + j d_{Q,i,n}$. The new representation of $s_i(t)$ is

$$s_i(t) = A \operatorname{Re} \left\{ \sum_{n=-\infty}^{+\infty} d_i(n) P_{T_s}(t - nT_s) e^{j\omega_i t} \right\}. \quad (2.16)$$

If we only consider one symbol duration where $mT_s \leq t \leq (m+1)T_s$, and recognizing that

$\omega_i = 2\pi f_i$, the result is

$$s_i(t) = A \operatorname{Re} \left\{ d_i(m) e^{j2\pi f_i t} \right\}. \quad (2.17)$$

Considering all N sub-carriers,

$$s(t) = \sum_{i=0}^{N-1} s_i(t) = \sum_{i=0}^{N-1} A \operatorname{Re} \left\{ d_i(m) e^{j2\pi f_i t} \right\} \quad (2.18)$$

Additionally, if the i^{th} sub-carrier frequency is defined as $f_i = f_0 + i\Delta f$ then $s(t)$ can be written as

$$s(t) = A \operatorname{Re} \left\{ \sum_{i=0}^{N-1} d_i(m) e^{j2\pi i\Delta f t} e^{j2\pi f_0 t} \right\} \quad (2.19)$$

As previously discussed, orthogonality is achieved when $\Delta f = \frac{1}{T_s}$. If we sample $s(t)$ at samples $t = m\frac{T_s}{N}$, where $m = 0, 1, \dots, N-1$, the result is

$$s\left(m\frac{T_s}{N}\right) = N \text{ARE} \left\{ \frac{1}{N} \sum_{i=0}^{N-1} d_i(m) e^{j2\pi i \frac{m}{N}} e^{j2\pi f_0 m \frac{T_s}{N}} \right\}. \quad (2.20)$$

The Inverse Discrete Fourier Transform (IDFT) is defined as $IDFT[d_i(m)] = \frac{1}{N} \sum_{i=0}^{N-1} d_i(m) e^{j2\pi i \frac{m}{N}}$.

Consequently, Equation (2.20) may be re-written as

$$s\left(m\frac{T_s}{N}\right) = N \text{ARE} \left\{ IDFT[d_i(m)] e^{j2\pi f_0 t} \right\} \quad (2.21)$$

or, using the complex envelope,

$$\tilde{s}\left(m\frac{T_s}{N}\right) = NA \cdot IDFT[d_i(m)]. \quad (2.22)$$

Equation (2.22) shows that the IDFT algorithm forms the kernel for transmitting the serial data stream into N Orthogonal Frequency Division Multiplexed data streams. The IDFT algorithm ensures that all oscillator frequencies are precisely R_s apart, thereby maintaining orthogonality between the N signals. The IDFT algorithm is practical today due to faster signal processing and computational efficiency via application specific IFFT integrated circuits [16] and reconfigurable hardware such as field programmable gate arrays [8]. This makes the IDFT method less expensive than a conventional multi-oscillator technique. Additionally, the receiver simply implements an FFT to recover the data [17]. Another advantage is that the receiver can separate carriers by use of baseband signal processing techniques vice using more complex bandpass filtering techniques [18]. At the receiver, the sub-carrier signal can then be integrated over the symbol period to produce the desired sub-carrier decision statistic. All other sub-carrier signals are filtered out due to their orthogonal nature. This does not entirely solve the synchronization

vulnerabilities of the OFDM signal. For example, symbol timing becomes critical to ensure that only the appropriate OFDM symbols are included in the receiver's FFT processing. Failure to properly time the FFT processing could result in significant ISI by including values of either the previous or follow-on OFDM symbol as we shall discuss in section III [8].

In summary, this chapter addressed the necessity to use multi-carrier modulation to achieve high data rates in fading channels. OFDM accomplishes this in a spectrally efficient manner using the IDFT kernel to achieve orthogonality between the sub-carriers. In the next chapter, the drawbacks of this scheme, with respect to synchronization, are addressed.

THIS PAGE INTENTIONALLY LEFT BLANK

III. SYNCHRONIZATION VULNERABILITIES OF OFDM

As discussed in the previous section, high data rates may be achieved with per sub-carrier flat fading and minimal ISI by multiplexing and transmitting data in parallel via sub-carriers. The most efficient way to do this is through OFDM, using an IFFT-FFT processing algorithm [2]. This ensured orthogonality can be maintained between the sub-carriers while transmitting the sub-carriers in a spectrally efficient manner.

Unfortunately, because of the efficient use of available bandwidth and the simultaneous transmission of N sub-carriers, OFDM is extremely sensitive to imperfections in frequency synchronization. Additionally, the multiplexing of symbols onto multiple sub-carriers, in a multi-path environment, leads to a system that is vulnerable to imperfections in timing synchronization as well. Equation Section 3. [19]

A. INTER-CARRIER INTERFERENCE

Lack of frequency synchronization causes inter-carrier interference (ICI). OFDM is vulnerable to ICI because of the simultaneous and spectrally efficient means of transmitting the sub-carriers as seen below.

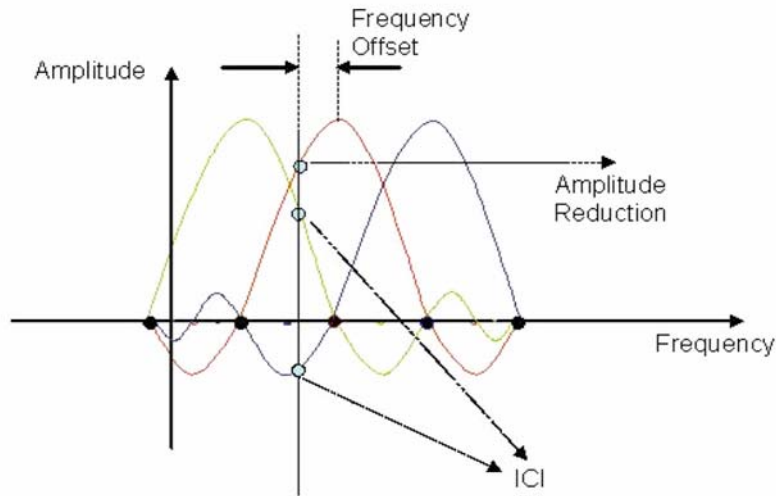


Figure 3 Depiction of ICI – from Reference [20]

In Figure 3, taken from [20], local oscillator (LO) drift has caused the receiver's center frequency of a particular sub-carrier to drift from the maximum value (peak of the sinc function) for the sub-carrier of interest. As the diagram shows, orthogonality between the sub-carriers has been destroyed. In the diagram above, if the intention is to detect the middle frequency sub-carrier there will be interference from the other two sub-carriers (ICI). Figure 4 is provided to show receiver sampling when ICI is not present. With multiple sub-carriers present, each sub-carrier is sampled by the receiver at its maximum amplitude ($\Delta f_c = 0$) while the remaining sub-carriers have zero amplitude at the sampling frequency, preserving orthogonality and maximizing SNR. This is much more desirable than the situation depicted in Figure 3.

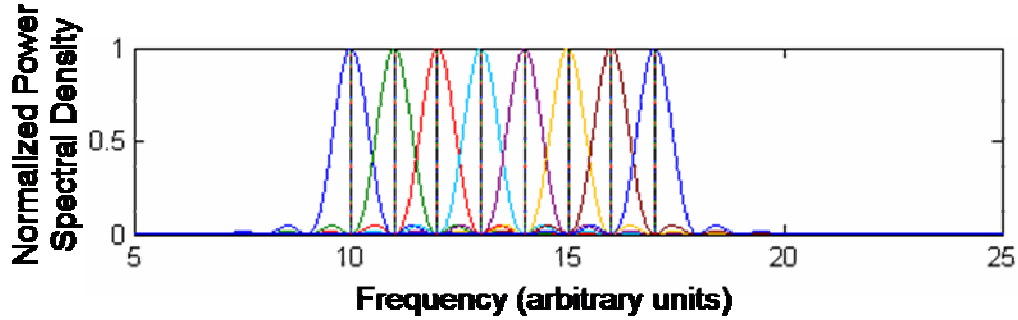


Figure 4 Reception without ICI

In Figure 3, a single sub-carrier frequency was shown to drift. The *IFFT/FFT* algorithm previously discussed eliminates the need for N individual oscillators for N sub-carriers, negating the problem of individual sub-carrier oscillator drift. This scheme, however, cannot address ICI caused by the possibility of a frequency offset of the receiver's local oscillator, phase noise inherent in the transmission system, or Doppler shift between receiver and transmitter.

1. Local Oscillator Offset

Local oscillator (LO) frequency offset is a frequency difference between the LO of the receiver and the LO of the transmitter. This will result in a frequency offset for every sub-carrier as is depicted in Figure 5.

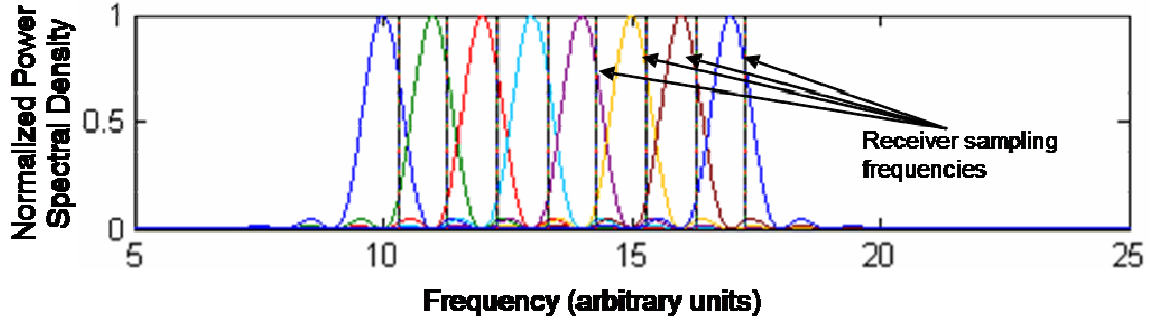


Figure 5 Sub-carrier reception with ICI

The frequency difference between the receiver's and transmitter's LO's is the mathematical equivalent of multiplying the post IFFT transmitter complex equivalent lowpass signal in the time domain by a complex exponential whose frequency is equal to the frequency offset (difference). This is the same as shifting the received signal in frequency. [21]

Additionally, close examination of Figure 5 shows that there is a reduction in received signal strength of the desired sub-carrier because the sub-carrier is no longer sampled at the peak of the sinc squared function. This reduction in signal strength, combined with the resulting ICI, results in lower detection rates [8]. The degradation in SNR, for relatively small frequency errors, can be approximated as [13]

$$D_{freq} \cong \frac{10}{3 \ln 10} (\pi \Delta f_c T_s)^2 \frac{E_s}{N_o} \text{ dB.} \quad (3.1)$$

In Equation (3.1), D_{freq} is the degradation in SNR, in dB, due to frequency offset, where Δf_c is the frequency offset in Hz, $\frac{E_s}{N_0}$ is the symbol energy to noise density ratio in Watts/Watt, and T_s is the symbol period in seconds.

Examination of Equation (3.1) shows that the degradation in SNR is proportional to the square of the frequency offset. Additionally, recognizing that the degradation, D_{freq} , is proportional to $\frac{E_s}{N_0}$, which will be largest when reception is strong and higher order M – ary modulations are used. Therefore we expect degradation in SNR due to the offset to be most pronounced for 64 – QAM and least pronounced for BPSK.

Figure 6 is an illustration of Equation (3.1). It shows the sensitivity of various modulation schemes to frequency offset at a given bit error rate (BER) of 10^{-6} . The $\frac{E_b}{N_0}$ required for BPSK/QPSK, 16-QAM, and 64-QAM modulation schemes to achieve a BER of 10^{-6} , uncoded, in an additive white Gaussian noise (AWGN) channel with no fading and perfect time, phase and frequency synchronization, are 10.5 dB, 14.7 dB, and 18.8 dB respectively [22]. The results are consistent with other sources, such as [8] and [13].

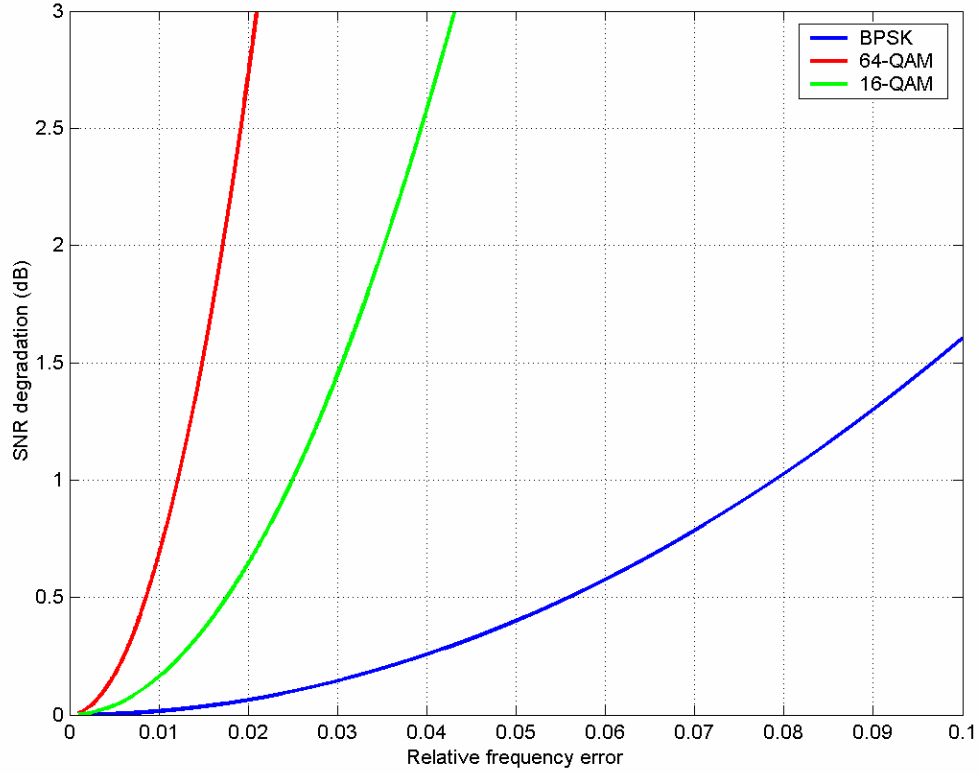


Figure 6 SNR degradation due to relative frequency error, $\Delta f_c T_s$

Figure 6 indicates that for a degradation of no more than 0.5 dB, the frequency offset cannot exceed 0.85% of sub-carrier spacing for 64-QAM, or 2.6565 kHz. For 16-QAM, frequency offset cannot exceed 1.76% of sub-carrier spacing, or 5.500 kHz. Finally, for BPSK/QPSK, frequency offset cannot exceed 5.58% of sub-carrier spacing, or 17.437 kHz. Hence, OFDM's sensitivity to frequency offset, as depicted in Figure 6, is apparent and confirmed by a variety of sources such as [8] and [9]. The IEEE 802.11g standard stipulates maximum center frequency tolerance of ± 25 PPM [5]. Under the IEEE standard, the maximum possible error between receiver and transmitter occurs if each oscillator experiences the maximum deviation of 25 PPM with opposite sign, resulting in a 50 PPM worst case frequency offset. At 2.4 GHz, this translates to a maximum allowable frequency error of

$$\Delta f_c = 50 \cdot 10^{-6} \cdot 2.4 \cdot 10^9 \text{ Hz} = 120 \text{ kHz}. \quad (3.2)$$

From Equation (3.2) and Figure 6, we can infer that despite the fact that commercial receivers' and transmitters' respective oscillators will meet the IEEE standard, a frequency offset of 60 kHz, or 19.2 % of sub-carrier spacing, will cause massive degradation of over 200 dB in receiver performance. The worst case scenario of 120 kHz, or 38.4% of sub-carrier spacing, will cause a degradation of over 1000 dB in receiver performance. In either case, 802.11g systems must correct for frequency offset of the respective local oscillator with frequency synchronization or performance will be severely affected.

2. Phase Noise

The additional problem of phase noise is related to the quality and cost of the local oscillators (LO) used. The ideal oscillator produces only the required frequency and its power spectral density is a Dirac delta function at the LO frequency. The output of an actual oscillator, not only produces the desired LO frequency, but also outputs power concentrated in a band about that frequency. These additional unwanted frequency components constitute phase noise and are present even after LO frequency offset correction is performed by the receiver [23]. Phase noise has two undesirable effects on receiver performance – ICI and constellation rotation [24].

Recalling from Equation (2.22), if the complex envelope of the transmitted OFDM signal for a given symbol is sampled at $t = m \frac{T_s}{N}$, where $m = 0, 1, \dots, N-1$, then the complex envelope of the transmitted signal becomes

$$\tilde{s}(m) = NA \left\{ \frac{1}{N} \sum_{i=0}^{N-1} d_i(m) \exp \left(j 2\pi i \frac{m}{N} \right) \right\}. \quad (3.3)$$

In Equation (3.3), the $\exp(\bullet)$ term is used for clarity vice $e^{(\bullet)}$ when the structure of the exponents would cause very small fonts to be used. If we sample instead at

$t = mT_s$ vice $t = m\frac{T_s}{N}$ for simplicity, and disregarding constants, Equation (3.3) can be re-written as

$$\tilde{s}(m) = \sum_{i=0}^{N-1} d_i \exp\left(j \frac{2\pi}{N} im\right). \quad (3.4)$$

The received complex envelope of the signal, neglecting additive noise, oscillator mismatch, and channel fading, is only affected by phase noise in the receiver as [25]

$$\tilde{r}(m) = \tilde{s}(m)e^{j\phi(m)} \quad (3.5)$$

At the receiver, the signal is Orthogonal Frequency Division de-multiplexed including the FFT as

$$y_k = \frac{1}{N} \sum_{p=0}^{N-1} \tilde{r}(p) \exp\left(-j \frac{2\pi}{N} kp\right) \quad (3.6)$$

which can be expanded to

$$y_k = \frac{1}{N} \sum_{p=0}^{N-1} e^{j\phi(p)} \tilde{s}(p) \exp\left(-j \frac{2\pi}{N} pk\right) \quad (3.7)$$

Using Equation (3.4), Equation (3.7) can be re-written, using different indices, as

$$\begin{aligned} y_k &= \frac{1}{N} \sum_{p=0}^{N-1} e^{j\phi(p)} \sum_{r=0}^{N-1} d_r \exp\left(j \frac{2\pi}{N} rp\right) \exp\left(-j \frac{2\pi}{N} pk\right) \\ &= \frac{1}{N} \sum_{p=0}^{N-1} e^{j\phi(p)} \sum_{r=0}^{N-1} d_r \exp\left(j \frac{2\pi}{N} p(r-k)\right). \end{aligned} \quad (3.8)$$

where d_r is the sub-carrier symbol defined with Equation (2.15). If the phase offset, $\phi(p)$, is small, it may be approximated as $e^{j\phi(p)} \approx 1 + j\phi(p)$ [25]. In that case, y_k becomes

$$y_k \approx \frac{1}{N} \sum_{r=0}^{N-1} d_r \sum_{p=0}^{N-1} \exp\left(j \frac{2\pi}{N} p(r-k)\right) + \frac{j}{N} \sum_{r=0}^{N-1} d_r \sum_{p=0}^{N-1} \phi(p) \exp\left(j \frac{2\pi}{N} p(r-k)\right). \quad (3.9)$$

The first term on the right hand side of Equation (3.9) can be simplified to

$$\begin{aligned}
\frac{1}{N} \sum_{r=0}^{N-1} d_r \sum_{p=0}^{N-1} \exp\left(j \frac{2\pi}{N} p(r-k)\right) &= \frac{1}{N} \sum_{p=0}^{N-1} \sum_{r=0}^{N-1} d_r \exp\left(j \frac{2\pi}{N} rp\right) \exp\left(-j \frac{2\pi}{N} kp\right) \\
&= \frac{1}{N} \sum_{p=0}^{N-1} \tilde{s}(p) \exp\left(-j \frac{2\pi}{N} kp\right) \\
&= DFT[\tilde{s}(p)] \\
&= DFT[IDFT(d_i)] \\
&= d_k.
\end{aligned}$$

Hence, Equation (3.9) becomes

$$y_k = d_k + \frac{j}{N} \sum_{r=0}^{N-1} d_r \sum_{p=0}^{N-1} \phi(p) \exp\left(j \frac{2\pi}{N} p(r-k)\right). \quad (3.10)$$

[26]

Equation (3.10) may be re-written as

$$y_k \approx d_k + \varepsilon_k, \quad (3.11)$$

where ε_k is the total error caused by phase noise and is added to every sub-carrier symbol. It represents the second term of Equation (3.10) and can be described as a complex exponential summed up for every sub-carrier. As a result, ε_k can be re-written as

$$\varepsilon_{k,r} = \frac{j}{N} \sum_{r=0}^{N-1} d_r \sum_{p=0}^{N-1} \phi(p) \exp\left(j \frac{2\pi}{N} p(r-k)\right). \quad (3.12)$$

Hence, each sub-carrier value has an error term in it due to the effects of phase noise, which in turn has two specific effects. The first effect occurs for the k^{th} term of $\varepsilon_{k,r(r=k)}$, or

$$\varepsilon_{k,r(r=k)} = \frac{j}{N} \sum_{r=k}^k d_r \sum_{p=0}^{N-1} \phi(p) \exp\left(j \frac{2\pi}{N} p(r-k)\right) = jd_k \Phi. \quad (3.13)$$

where the average phase error, Φ , is

$$\Phi = \sum_{p=0}^{N-1} \phi(p). \quad (3.14)$$

Equation (3.13) shows that a phase error of $j\Phi$ is added to every sub-carrier value and that this phase error is proportional to the received sub-carrier value d_k . Each sub-carrier, regardless of the value of d_k is equally affected by Φ . Hence, Φ represents the angular magnitude of the constellation's rotation due to phase noise [25]. When the rotation of the constellation's data points exceeds the modulation type's decision boundaries, symbols will be demodulated incorrectly [8]. Fortunately, every sub-carrier experiences equal constellation rotation and this can be leveraged to develop effective techniques to correct constellation rotation and maintain performance [13].

The second effect caused by the error term $\varepsilon_{k,r}$ is ICI. The ICI is caused by all of the remaining sub-carriers interfering with the sub-carrier being demodulated. In this case, $\varepsilon_{k,r(r \neq k)}$ becomes

$$\varepsilon_{k,r(r \neq k)} = \frac{j}{N} \sum_{\substack{r=0 \\ r \neq k}}^{N-1} d_r \sum_{p=0}^{N-1} \phi(p) \exp\left(j \frac{2\pi}{N} p(r-k)\right). \quad (3.15)$$

Equation (3.15) shows that the ICI-error is a frequency shifted version of $\phi(p)$, the frequency shift being proportional to $(r-k)$. This complex number is added to each sub-carrier's useful value, reducing its amplitude and destroying orthogonality [25].

To minimize degradations caused by phase noise, wireless engineers will model and simulate phase noise for various commercial oscillators to triangulate price, accuracy, and complexity of the receiver's synchronization technique to meet performance and cost requirements. To properly model the phase noise in an oscillator, the phase noise is treated as a random process which causes the oscillator frequency to deviate from the desired frequency. The strength of the undesired frequency deviations can be shown in a depiction of the power spectral density of the oscillator (including phase noise). Ideally, the oscillator would only generate the desired LO frequency, but

actually generates random signals about the desired frequency. This is shown in the PSD for a phase locked oscillator depicted in Figure 7 below.

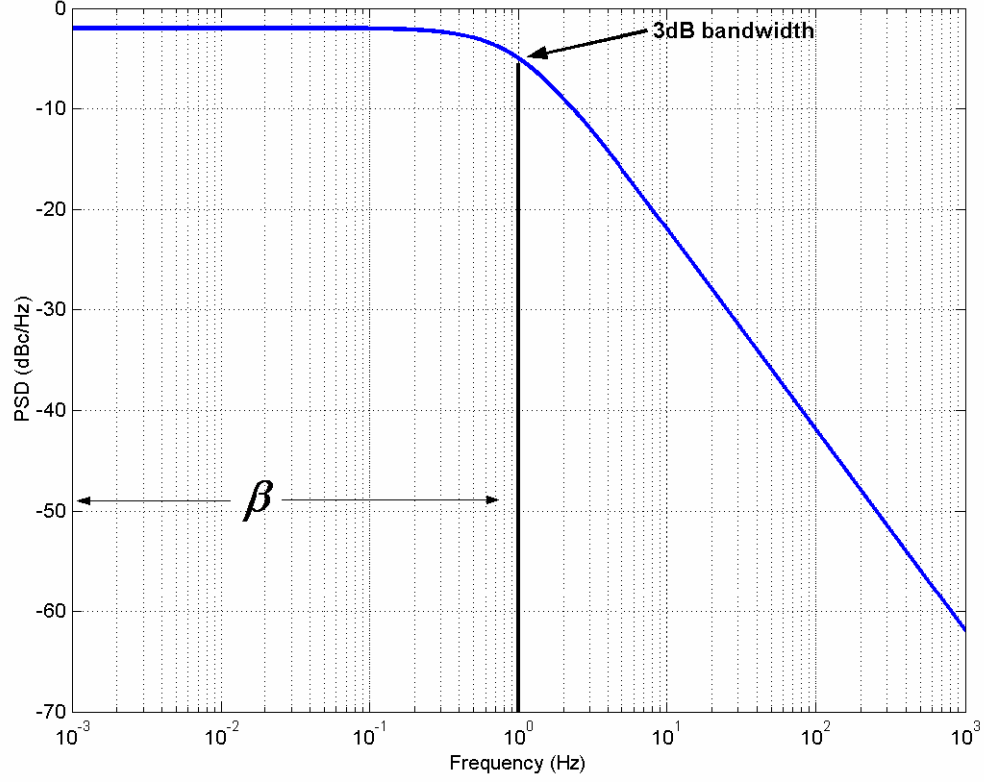


Figure 7 Single-sided LO PSD with Phase Noise ($\beta = 1\text{Hz}$)

The half power point, or Beta, as depicted in Figure 7, can be used to estimate the SNR degradation [9]. The degradation in SNR, caused by a failure to correct for phase noise of the LO, is approximated as [13]

$$D_{\text{phase}} \cong \frac{11}{6 \ln 10} (4\pi\beta T_s) \frac{E_s}{N_0} \text{ dB}, \quad (3.16)$$

where β has been defined above, T_s is the symbol period in seconds, and $\frac{E_s}{N_0}$ is the symbol energy to noise density ratio in Watts/Watt. Figure 8 depicts the degradation in

SNR due to phase noise, using the same modulations types and values of BER and $\frac{E_b}{N_0}$ that we used in the previous analysis for frequency offset.

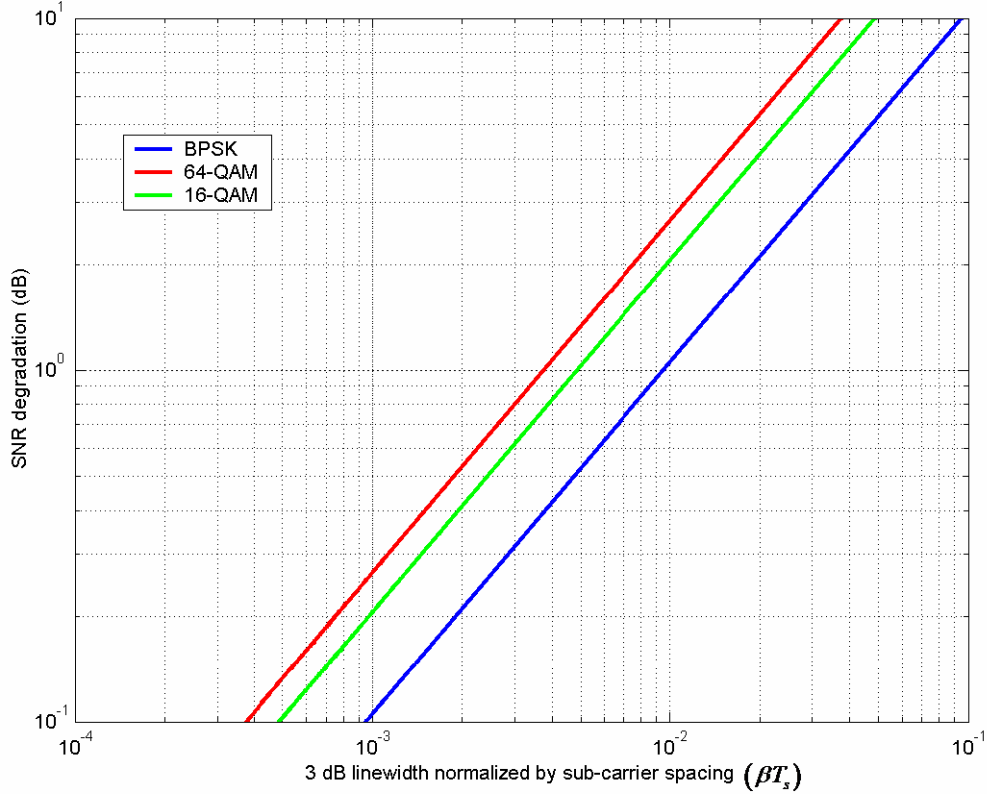


Figure 8 SNR degradation caused by phase noise

The horizontal axis of Figure 8 is normalized to the 802.11g sub-carrier spacing of 312.5 kHz for clarity. If an arbitrary SNR degradation of .5 dB is acceptable, then from Figure 8, the one-sided 3 dB oscillator half-power point or linewidth (β) to sub-carrier spacing ratio must be less than .0019 for 64-QAM. This translates to a maximum acceptable linewidth of 594 Hz. For 16-QAM and BPSK, the maximum acceptable linewidth is relaxed to 750 Hz and 1.50 kHz respectively.

The narrow linewidth limitations required to avoid noticeable degradations in SNR, and consequently receiver performance, suggest that an obvious solution would be

to purchase the highest quality LO for the 802.11g receiver and transmitter. The economic feasibility of any broadband fixed wireless access system, however, depends heavily on the cost of individual subscriber units and their components. Unfortunately, the cost of components, including oscillators, is directly related to their stability and accuracy. Hence, any commercial OFDM wireless access system shall have to meet acceptable performance standards with moderately priced components, necessitating receiver synchronization techniques to correct for phase noise inherent in the receiver's LO. [27]

3. Doppler Shift

The final phenomena causing ICI is Doppler shift. Doppler shift will occur when there is relative motion between receiver (Rx) and transmitter (Tx). The Doppler shift is the difference between the received signal center frequency and the transmitted signal center frequency. Considering the 802.11g system, the Doppler shift for a particular sub-carrier is given as

$$f_{Rx} - f_{Tx} = \Delta f_D = \frac{v}{c} f_s \cos \alpha. \quad (3.17)$$

In Equation (3.17), Δf_D represents the deviation from sub-carrier center frequency due to Doppler shift, f_s is the transmitted sub-carrier center frequency, c is the speed of light, and $v \cos \alpha$ is the relative velocity between transmitter and receiver. Equation (3.17) only considers either the Rx or Tx in motion. Most scenarios in a WLAN environment would involve a mobile unit (such as a laptop or hand held device) and a station unit (such as a wireless access point or router). Hence α represents the angular difference between the Rx's or Tx's velocity vector and a straight line connecting the two. Figure 9 below highlights four instances of Doppler shift in which the Tx is stationary while the Rx moves.

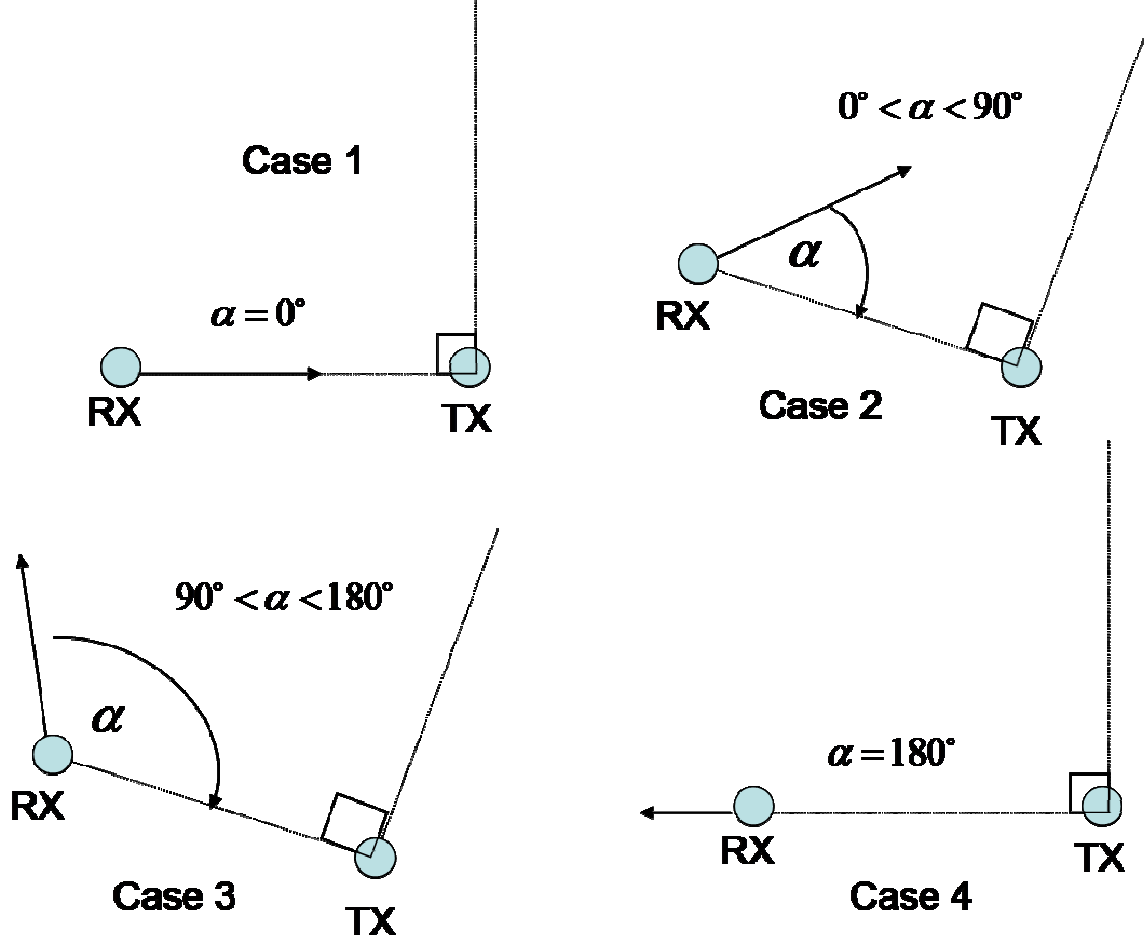


Figure 9 Doppler shift due to relative motion between RX and TX

If $|v|$, or the speed of the Rx, is the same for all four cases of Figure 9, the maximum Doppler shift occurs when $\alpha = 0^\circ$, as depicted in Case 1 of Figure 9. The minimum Doppler shift occurs when $\alpha = 180^\circ$, as depicted in Case 4. Cases 2 and 3 highlight how α can vary between $[0, \pi]$ and thereby change the value of Δf_D . Perpendicular axes are shown in Figure 9 in each case for clarity.

From Equation (3.17), Doppler shift has a greater effect at high frequencies. Consequently, when considering multiple sub-carriers, higher frequency sub-carriers will experience a greater frequency shift due to Doppler. Equation (3.17) may be re-written for multiple sub-carriers as

$$\Delta f_{D,i} = \frac{v}{c} f_i \cos \alpha, \quad (3.18)$$

where $i = 1 \dots N$ sub-carriers. The percentage change in frequency due to Doppler shift, however, will be the same for all sub-carriers, when each sub-carrier's Doppler shifted frequency is normalized to its respective transmitted sub-carrier center frequency [28]. In this case (for simplicity, let $\alpha = 0$),

$$\xi = \frac{\Delta f_{D,i}}{f_i} = \frac{v}{c} \quad (3.19)$$

where ξ is the percentage change in frequency due to Doppler shift. From Equation (3.19), the percentage change in frequency for all sub-carriers is only affected by the relative motion between the Tx and Rx. Further, if a particular sub-carrier is transmitted at frequency f_s , then the received frequency will be $f_s(1 + \xi)$ due to Doppler. For multiple sub-carriers, the received Doppler shifted frequencies becomes

$$(1 + \xi)(f_c + h\Delta f_s) = (1 + \xi)f_c + (1 + \xi)h\Delta f_s, \quad (3.20)$$

where f_c is the carrier frequency, Δf_s is the sub-carrier frequency spacing and $h = \left(\frac{-N}{2} \dots \frac{N}{2}\right)$. Unlike Equation (3.18), Equation (3.20) describes Doppler shifted frequencies in which a carrier frequency is bracketed above and below by $N/2$ sub-carriers. From Equation (3.20), it can be seen that the carrier frequency, sub-carrier frequencies, and sub-carrier spacing are affected by the same percentage ξ . For an OFDM signal, recall the following time-domain, transmitted OFDM signal from Equation (2.9):

$$s(t) = \sum_{i=0}^{N-1} a_i(t) \cos[(\omega_c + i\omega_\Delta)t + \theta_i(t)] \quad (3.21)$$

where ω_c is the carrier frequency, ω_Δ is the sub-carrier spacing, a_i is dependent upon signal amplitude, θ_i varies with the in-phase and quadrature modulation values and for

clarity, sub-carriers $i = 0 \dots N - 1$ are considered. When a Doppler shift is included, the received signal becomes

$$r(t) = \sum_{i=0}^{N-1} a_i(t) \cos[(1 + \xi)(\omega_c + i\omega_\Delta)t + \theta_i(t)], \quad (3.22)$$

where the effect of the Doppler shift on $\theta_i(t)$ has been ignored. Substituting Equation (3.20) and recalling Equation (2.8), Equation (3.22) may be re-written as

$$r(t) = \sum_{i=0}^{N-1} \left\{ a_i(t) \cos[(1 + \xi)i\omega_\Delta t + \theta_i(t)] \cos[(1 + \xi)\omega_c t] - a_i(t) \sin[(1 + \xi)i\omega_\Delta t + \theta_i(t)] \sin[(1 + \xi)\omega_c t] \right\} \quad (3.23)$$

If the factor $\cos[(1 + \xi)i\omega_\Delta t + \theta_i(t)]$ can be considered as the envelope of the carrier $\cos[(1 + \xi)\omega_c t]$ and similarly for the second term in Equation (3.23), the expression shows that both carrier frequency and envelope frequency are affected by the same Doppler percentage ξ . [28]

Finally, clock timing is derived from the OFDM symbol rate R_s . Recall from Equation (2.12), that to maintain orthogonality, sub-carrier spacing must be an integer multiple of R_s . When the sub-carrier spacing is shifted in frequency due to some Doppler percentage ξ , orthogonality between the sub-carriers is destroyed, creating ICI. [28]

Consequently, Doppler shift affects the frequencies of the carrier, sub-carriers, envelope, and symbol timing by the same Doppler percentage ξ , thereby creating ICI. The degradation in performance caused by Doppler shift has been estimated by many sources. It has been shown that the fractional degradation in received power due to ICI in an OFDM system with flat fading in any Doppler spectra is universally upper bounded by [29]

$$P_{ICI} \leq \frac{1}{12} (2\pi\Delta f_D T_s)^2, \quad (3.24)$$

where P_{ICI} , as a function of Δf_D and T_s , is the unitless degradation of received signal power caused by Doppler shift induced inter-carrier interference. In Equation (3.24), received signal power is normalized to 1 watt for clarity. Hence, a computed magnitude of .1 for P_{ICI} translates to a loss of 10% in received signal power. Although Equation (3.24) reveals that the P_{ICI} is a function of Δf_D and T_s , symbol period in 802.11g systems is fixed at $3.2 \mu\text{sec}$ [4]. Additionally, commercial WLANs are designed to be used inside buildings, or small outside areas, with fixed access points. Mobile subscribers are usually stationary or moving at pedestrian speeds. A brisk walking speed of 3 m/s yields a worst case Doppler shift of

$$\Delta f_D = \frac{3 \text{ m/sec}}{3 \cdot 10^8 \text{ m/sec}} \cdot 2.4 \cdot 10^9 = 24 \text{ Hz.} \quad (3.25)$$

The resulting maximum fractional degradation in received power due to ICI, using Equation (3.24), turns out to be

$$P_{ICI} \leq \frac{1}{12} (2\pi \cdot 24 \text{ Hz} \cdot 3.2 \mu\text{sec})^2 = .0000000194, \quad (3.26)$$

or .00000194 % loss in received signal power. This negligible effect is further confirmed by visual inspection of Figure 6 as 24 Hz represents only .007 % of sub-carrier spacing. At that amount of frequency offset, there is little effect on overall system performance. Therefore, ICI caused by Doppler shift will be much less than ICI caused by phase noise or LO offset in commercial systems [29].

4. Summary

In summary, inter-carrier interference (ICI) is caused by three phenomena - local oscillator offset, phase noise and Doppler shift. While Doppler shift, under normal conditions, was shown to have negligible effect on system performance, LO offset and phase noise exposed the vulnerabilities of OFDM systems. This is due to the spectrally efficient manner that the sub-carriers are transmitted over the wireless channel. Offset sampling by the OFDM receiver produced a reduction in signal strength and ICI from the remaining sub-carriers. In the case of LO offset, it was shown that even small deviations

in frequency offset caused noticeable degradations in performance, particularly for higher order modulations (e.g. 64-QAM). It was also shown that this degradation in performance could occur despite having met the IEEE 802.11g oscillator standard. Consequently, frequency synchronization is an essential task for the OFDM receiver. Phase noise was shown to produce noticeable degradation in system performance as well. The degradation from phase noise was directly related to the 3 dB linewidth of the receiver's oscillator. The obvious solution of purchasing high quality oscillators was discarded due to their high cost necessitating phase tracking by the OFDM receiver. Frequency [8] and phase synchronization are essential tasks for commercial OFDM receivers today and various techniques have been proposed [24]. These techniques shall be explored in Section IV.

B. INTER-SYMBOL INTERFERENCE

The previous section highlighted the vulnerabilities of OFDM to ICI. From the discussion in chapter II, and from Equations (2.3) and (2.4), it is shown that OFDM is particularly resistant to the effects of inter-symbol interference (ISI). Individual sub-carrier symbol duration is longer than if a single high data rate carrier was used, mitigating multi-path effects.

Regrettably, ISI is not entirely eliminated. Channel impulse response (delay spread of the channel) could affect follow-on OFDM symbols [8]. In addition, symbol timing offset can introduce ISI [30].

1. Long Delay Spreads and ISI

Long delay spreads for certain channels could induce ISI for the 802.11g system. Per the IEEE 802.11g standard, the sub-carrier symbol duration is $3.2 \mu\text{sec}$. This lengthy symbol duration will generally ensure flat fading within a particular sub-carrier, precluding ISI. Recall that Equation (2.3) stipulates a rule of thumb for flat fading. Inspection of Table 1 shows that for all but the longest channel delay spreads, the flat fading rule of thumb is adhered to for sub-carrier duration. This is further confirmed by the fact that, according to Table 1, a typical office building has a median RMS delay

spread of 25 η sec. This corresponds to only 0.7% greater path length than line of sight for 802.11g's sub-carrier symbol duration. However, for channels with extremely long RMS delay spreads, ISI could manifest itself when OFDM symbols overlap in time. [31]

The following table lists delay spreads for various channels between 900 MHz and 1.9 GHz but can be extrapolated to higher frequencies as an upper bound.

Table 1 Typical Values of RMS Delay Spread – from Reference [1]

Environment	RMS Delay spread (σ_{RMS})	Notes
Urban	1300 η sec average 600 η sec std. dev. 3500 η sec maximum	New York City
Urban	10 – 25 μ sec	Worst case, San Francisco, CA
Suburban	200 – 310 η sec	Averaged typical case
Indoor	10 – 50 η sec 25 η sec median	Office building
Indoor	270 η sec maximum	Office building
Indoor	70 – 94 η sec average 1470 η sec maximum	Three San Francisco buildings

From Table 1, we conclude that 802.11g systems will be resistant to ISI except in most channels with particularly long delay spreads. To ensure ISI is eliminated between follow-on OFDM symbols, a guard period is inserted between the symbols. The IEEE standard specifies the duration of the guard interval – .8 μ sec (T_{gi}). Figure 10 shows how the guard interval provides ISI protection for follow-on OFDM symbols.

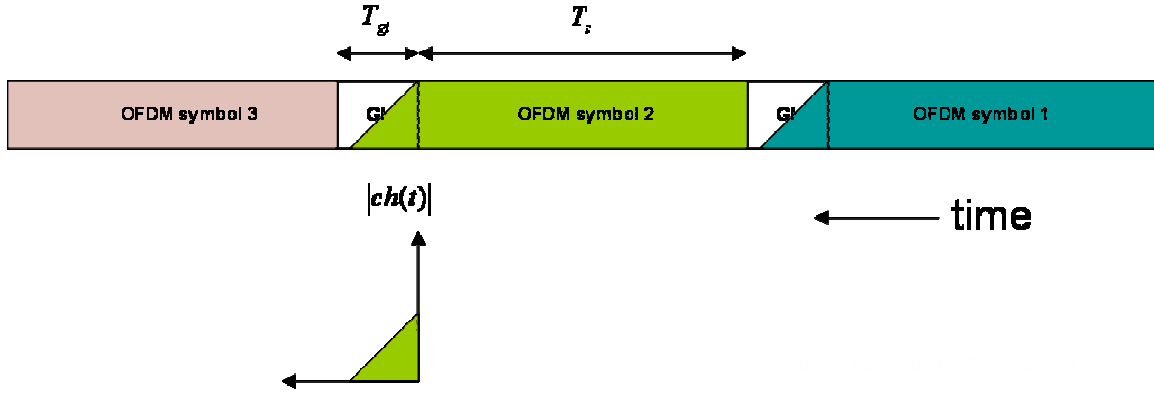


Figure 10 Guard interval and channel response – from Reference [20]

ISI from each OFDM symbol is depicted above by a generic channel impulse response (CIR), decreasing in intensity with time. Note that once a properly chosen guard interval has been placed before each follow-on OFDM symbol, the effects of ISI are essentially eliminated. As Table 1 depicts, ISI could appear in the symbol estimate if a channel has an extremely long delay spread, but a properly designed system will have minimum ISI due to the weak values of the channel response far into the guard interval. Critically, the length of the guard interval must be greater than the maximum acceptable delay spread value [13]. Total symbol duration (T_{total}) will now be:

$$T_{total} = T_{gi} + T_s \quad (3.27)$$

where T_s is the original OFDM symbol duration of 3.2 μsec and T_{gi} is the guard interval duration. In theory, the guard interval duration could be made to vary with the channel impulse response (time delay spread of the channel), however, per the 802.11a standard, from which the 802.11g standard adopts, guard interval duration is fixed at 20% of T_{total} . [4] The drawback in accommodating the guard interval is overhead with respect to data transmission rate, but well worth the price to mitigate ISI.

2. Symbol Timing Offset and ISI

Symbol timing offset can also introduce ISI within an 802.11g system. As the receiver begins its demodulation process, the FFT window length shall equal the length

of the OFDM symbol. Ideally, the FFT window could start with the first data sample of a particular symbol. The window would end with the last data sample of the same symbol. Figure 11 depicts this.

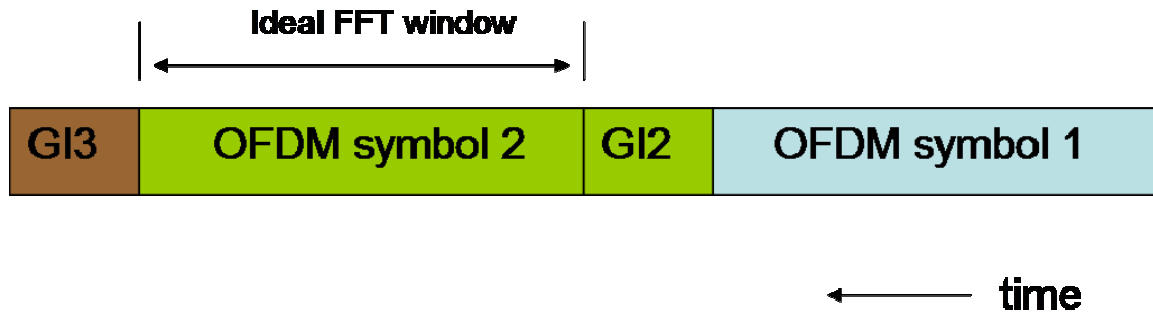


Figure 11 Ideal FFT window timing

Note that in Figure 11, when demodulating OFDM symbol 2, no FFT samples are taken from Guard Interval 2. This provides the maximum ISI protection, as samples of the guard interval may be affected by multi-path effects from OFDM symbol 1.

In practice, it is impossible to fix the symbol timing point at exactly the first sample of an OFDM symbol. There is, invariably, some deviation of symbol timing around a mean value. [8]

In Figure 11, early FFT window timing means including samples from guard interval 2, while omitting some of the last samples from OFDM symbol 2. Late FFT window sampling would mean including samples from guard interval 3 and omitting the initial samples of symbol 2.

The sampled guard interval could have been designed to contain no signal data but then, in a multi-path environment, orthogonality would have been lost between the sub-carriers, resulting in inter-carrier interference (ICI). Instead, the IEEE standard mandates that the OFDM symbol is cyclically extended into the guard interval. Hence the guard interval, serves two functions – protect the OFDM symbols against ISI and mitigate ICI in a multi-path environment by maintaining orthogonality between sub-carriers. [13]

From Equation (2.12), orthogonality between sub-carriers, is achieved when the frequencies of the sub-carriers are integer multiples of each other. To maximize spectral efficiency, the first sub-carrier is set to integer 1, yielding a sub-carrier spacing of $1/T_s$. Neglecting the carrier frequency, additional sub-carriers would be at frequency i/T_s , where $i = 1 \dots N$ sub-carriers. Figure 12 shows an OFDM receiver's FFT interval with three sub-carriers. Although, these sub-carriers would be summed together in an OFDM signal, they are depicted separately for clarity. The number of cycles in the FFT interval of sub-carriers 2 and 3 are integer multiples of the first sub-carrier, achieving orthogonality. Furthermore, a guard period exists prior to the FFT interval. In Figure 12, the guard interval is filled with zero-valued samples.

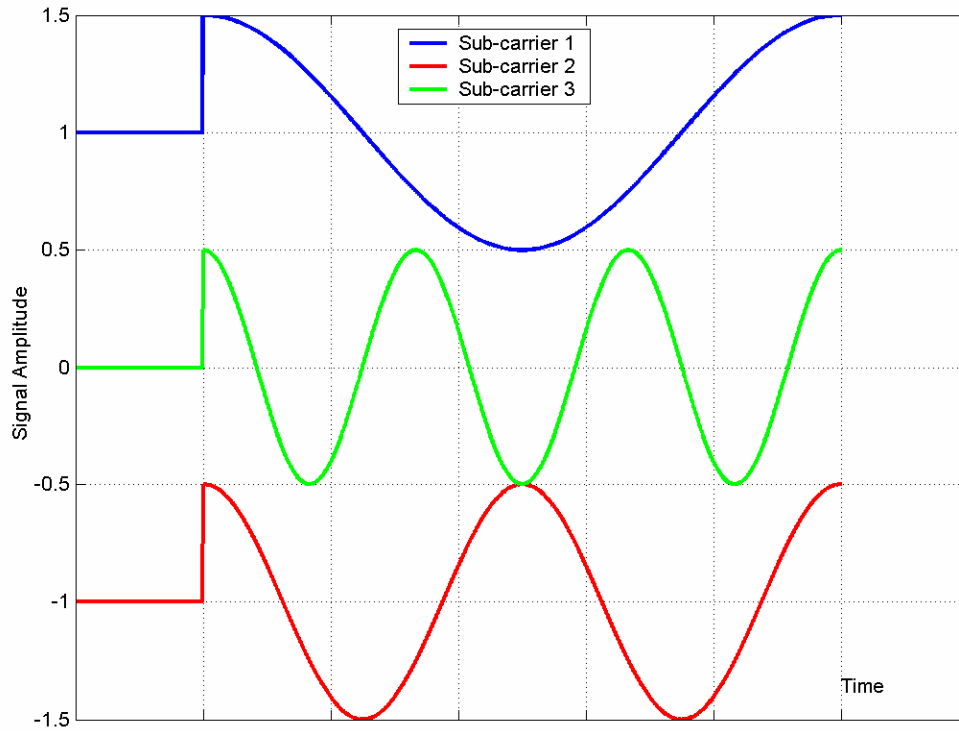


Figure 12 3 sub-carriers in FFT interval – no ICI present

If a sub-carrier's arrival at the receiver is delayed due to multi-path, there will no longer be integer multiples of sub-carrier cycles in the FFT interval, destroying orthogonality between the sub-carriers. This is depicted in Figure 13.

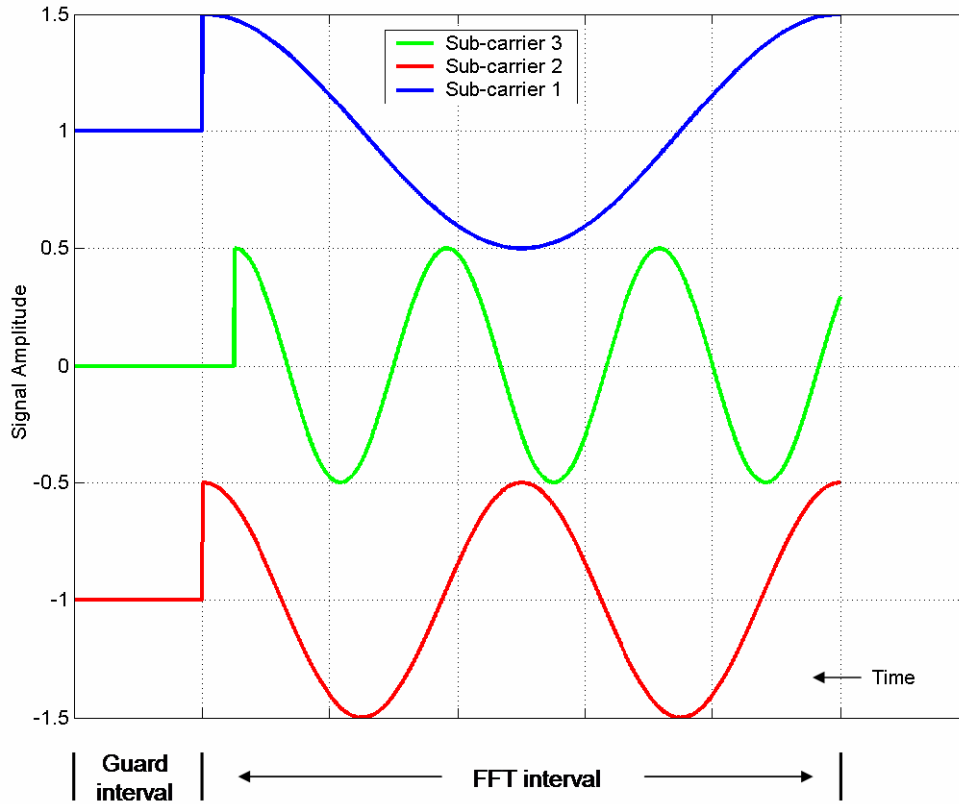


Figure 13 3 sub-carriers with ICI caused by sub-carrier delay

In Figure 13, sub-carrier 3 is delayed and no longer has an integer multiple number of cycles within the FFT interval relative to the other sub-carriers. Hence, sub-carrier 3 will cause cross-talk, or create ICI, with sub-carriers 1 and 2 and vice versa. The solution is to cyclically extend, or copy samples from sub-carrier 3 and append it to the beginning of sub-carrier 3. The result is shown in Figure 14 where delayed sub-carrier 3 now has an integer number of cycles within the FFT interval. [13]

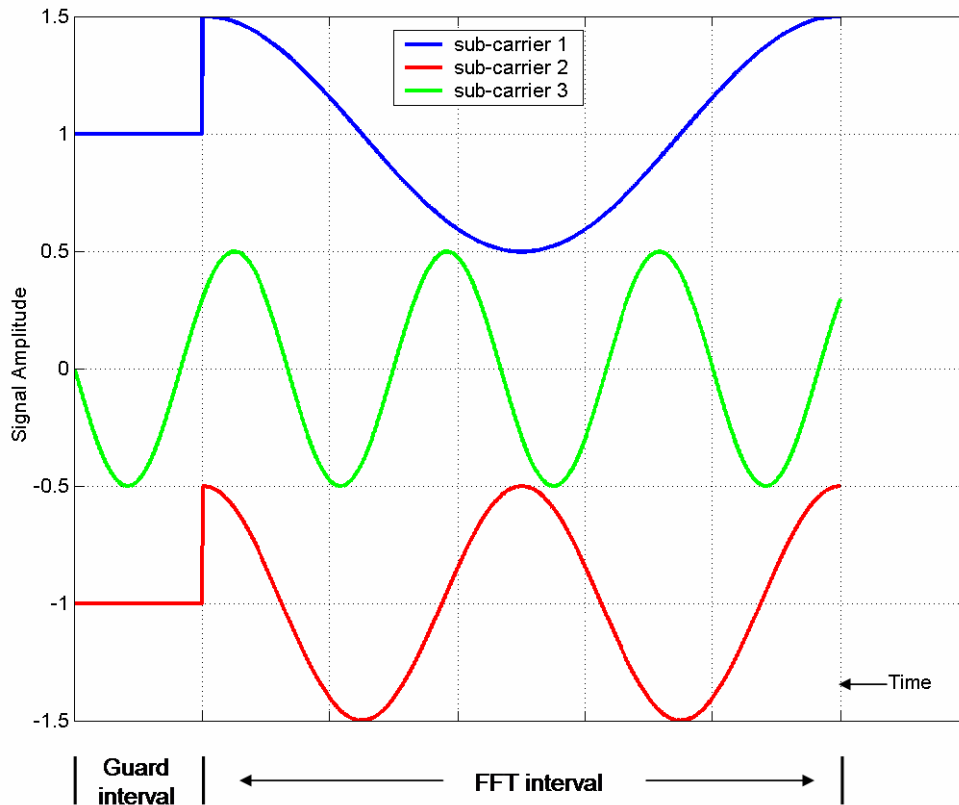
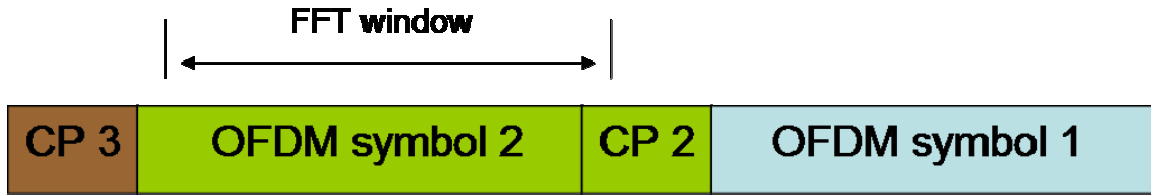
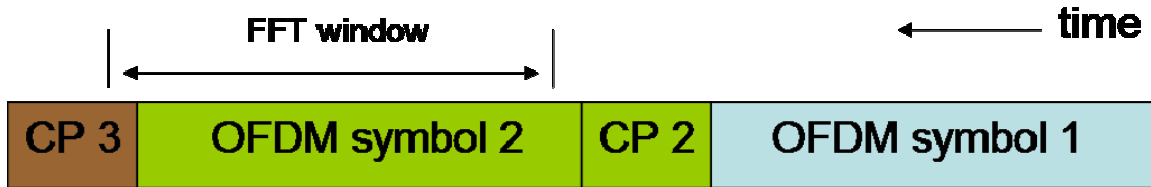


Figure 14 3 sub-carriers with CP added to delayed sub-carrier

Because the guard interval actually consists of samples of the follow on OFDM symbol it is more correctly known as the cyclic prefix (CP). In Figure 11 above, guard interval 2 actually consists of a repetition of a portion of OFDM symbol 2's sample values. This raises the potential for ISI contamination from OFDM symbol 1 but as mentioned previously, channel response values should be weak far into the guard interval. Samples from the guard interval, now cyclic prefix, contain the appropriate values for correct demodulation. Consider the case of late FFT window timing, as depicted in Figure 15 (b). The FFT window includes samples from CP 3 and omits the beginning of OFDM symbol 2. Cyclic Prefix 3, however, consists of samples from follow-on OFDM symbol 3 (not depicted). The result from this late symbol timing is severe ISI from CP 3, which contains samples from the follow on symbol. [8]



(a) Acceptable (early) FFT window – minimal ISI



(b) Unacceptable (late) FFT window – severe ISI

Figure 15 Early vs. Late FFT window timing and ISI

3. Summary

The solution to avoiding ISI caused by late symbol timing offset is to ensure that the FFT window starts early. This means including samples from the appropriate cyclic prefix. Orthogonality is still maintained because the CP contains the appropriate samples values for correct demodulation. If, as Table 1 suggests, the length of the CP is longer than the CIR, ISI is significantly reduced allowing the OFDM receiver to take advantage of the circular aspects of the FFT. With ISI mitigated, the linear convolution between the symbol and channel becomes circular. In other words, the FFT window should begin well inside the CP (when the values of the channel response due to the prior symbol are weak) and end prior to the CP of the follow on symbol to avoid ISI.[8]

In summary, this chapter showed clearly that 802.11g receivers must implement some form of symbol synchronization. Failure to properly synchronize in time and frequency resulted in ISI and ICI. The next chapter will address various synchronization algorithms to overcome these problems.

IV. SYNCHRONIZATION OF THE OFDM SIGNAL

A. INTRODUCTION

As shown in Chapter III, failure of the receiver to properly synchronize in time and frequency can seriously impair the performance of an OFDM system. To ensure ISI-free detection, precise timing synchronization must occur to determine the symbol boundaries within an OFDM packet. To ensure sub-carrier orthogonality and, hence, ICI-free detection and performance, frequency synchronization must occur [32]. Prior to these tasks, the packet must be detected by the receiver. This is the first synchronization task performed that future synchronization tasks are dependent upon [8]. Consequently, to properly detect and demodulate the OFDM signal, the receiver and transmitter must be properly synchronized. The algorithms that make up the OFDM synchronization process represent a major design problem that has to be solved to field a successful product [8].

1. OFDM Synchronization and Single Carrier Synchronization

Synchronization in the OFDM multi-carrier system can be very different from synchronization in a single carrier system. Specifically, the nature of the OFDM signal waveform makes the majority of the traditional single carrier synchronization algorithms unusable [8]. While the parallel transmission of N sub-carriers results in longer symbol duration, and consequently less sensitivity to timing offset than in a single carrier system [31], OFDM is very sensitive to frequency offset, as shown in Chapter III. Single carrier systems may experience a similar degradation in SNR from frequency offset, but do not, in general, suffer the loss of orthogonality and the resulting interference from neighboring sub-carriers. Also, many of the OFDM synchronization functions may be performed in the time or frequency domain, a level of flexibility that is not found in single carrier systems [8]. Ultimately, the variety of synchronization schemes available within OFDM presents a trade-off between performance and computational complexity. [8]

2. Synchronization for WLANs and Broadcast Systems

OFDM is used for both broadcast communication systems and ad hoc wireless networks. The nature of the packet switched WLAN will determine the appropriate synchronization algorithms. Broadcast systems, such as Digital Audio Broadcasting or Digital Terrestrial Television Broadcast, can employ techniques that may require a relatively long time to acquire and then track the signal. This luxury is not permitted for WLANs. The packet switched WLAN can expect random, short bursts of data [8]. In 802.11g the maximum permitted length of the Medium Access Control Protocol Data Unit (MPDU) is 4095 bytes [3]. The preamble and header fields comprise 20 μsec [4], resulting in a 627 μsec signal duration at the maximum data rate of 54 Mbps [12]. The header is required for successful demodulation of the OFDM data symbols, leaving only the preamble duration of 16 μsec for synchronization [12]. Therefore, synchronization must occur very rapidly with minimal overhead to maintain high throughput [12]. Consequently, the IEEE 802.11g preamble has been designed to facilitate synchronization [8]. This fundamentally differs from broadcast systems, in which the signal is always being transmitted.

3. Synchronization Methodology

No explicit synchronization technique is mandated by the IEEE 802.11g standard. Many different types of synchronization algorithms have been proposed to maximize performance [33].

For the OFDM WLAN, synchronization is usually accomplished in two phases: acquisition and tracking. A good acquisition technique should be robust with as wide an acquisition range as possible and should obtain a feasible timing or frequency estimate quickly. Furthermore, the better the estimate, the better the overall bit error performance. [33]

Synchronization algorithms can be divided into three types:

- Non data-aided synchronization that depends on the spectral characteristics of the received signal

- Cyclic prefix based synchronization that exploits the redundancy present in the OFDM packet.
- Data-aided synchronization that exploits training data or pilot symbols within the OFDM packet. [12]

a. *Non Data-Aided Synchronization*

Non data-aided synchronization does not need training symbols or pilot tones, reducing overhead and increasing data throughput. Because training symbols are not required, non data-aided synchronization appears well suited for broadcast type systems because the receiver does not have to wait for the training symbols to appear at the beginning of the data stream. Unfortunately, the performance of non data-aided synchronization algorithms is substandard in the presence of multi-path interference and their accuracy is generally not sufficient to guarantee orthogonality during OFDM packet based transmission [12].

b. *Cyclic Prefix Based Synchronization*

Cyclic prefix based synchronization exploits the redundancy within the OFDM data symbol. As described in chapter III, the cyclic prefix has been inserted to protect follow on data symbols from ISI but also contains copies of the data samples to preserve sub-carrier orthogonality. Since the cyclic prefix is a repetition of some fraction of the OFDM symbol, analyzing the change in the repeated data can be used to maintain synchronization. Generally, the complexity of cyclic prefix based synchronization is less than data-aided or non data-aided synchronization, allowing for lower cost hardware [12]. There is no additional overhead to be transmitted because the cyclic prefix is already mandated by the IEEE standard. If the OFDM receiver were to simply use this low complexity technique, one immediate problem would be packet detection. This initial synchronization task would be dependent upon a cyclic prefix at the start of the OFDM packet. [4] and [5] make no specific provision for a CP at the beginning of the preamble. This is confirmed by Figure 17 which depicts the 802.11g preamble with no CP for the start of the OFDM packet. Furthermore, unless cyclic prefix synchronization is

combined with other synchronization techniques, it will take several guard intervals before synchronization is obtained, something that is not practical for packet-based OFDM [13]. Because of this, cyclic prefix based synchronization is best suited for broadcast systems [12].

c. Data-Aided Synchronization

Data-aided synchronization algorithms depend on special synchronization symbols that must be added to the data packet and / or pilot tones embedded within the OFDM sub-carriers. The use of data-aided algorithms to achieve synchronization therefore, by definition, reduces data throughput and requires increased overhead. On the other hand, data-aided synchronization provides the widest acquisition range, best frequency tracking and can meet the requirement of the packet-based 802.11g system for synchronization within 16 μ sec of packet arrival [12]. Because the 802.11g IEEE standard mandates the use of synchronization symbols in the preamble and designates certain sub-carriers as pilots, the overhead associated with data-aided synchronization is already borne by the packet in order to meet the IEEE standard. A general communications engineering design principle is that the receiver should use all the available information to its advantage [8]. As a result, we can expect many synchronization algorithms to exploit these parts of the OFDM packet structure because of the requirement to achieve fast and accurate synchronization while minimizing additional overhead. Viable data-aided synchronization techniques, because they seem to be the likely candidates to improve performance of the 802.11g system, will be the focus of the remainder of this chapter and those parts of the OFDM packet that data-aided synchronization techniques exploit will be addressed in the following section.

B. IEEE 802.11G PACKET STRUCTURE AND SYNCHRONIZATION

1. Physical Layer Convergence Protocol Receive Procedure

The IEEE 802.11g standard promulgates an OFDM packet structure that allows for synchronization in time and frequency. The OFDM packet is sent from the transmitter to the receiver in accordance with the 802.11g physical layer convergence

function (PLCF), which is supported by a specified physical layer convergence protocol (PLCP). The PLCP defines the method of mapping 802.11g medium access control (MAC) sub-layer protocol data units (MPDUs) into a frame format suitable for sending data between transmitter and receiver given the characteristics of a particular wireless channel. The characteristics of the wireless channel, and the method of transmitting data, form the physical medium dependent (PMD) portion of the PLCF. [4]

The OFDM PLCP receive procedure is depicted below.

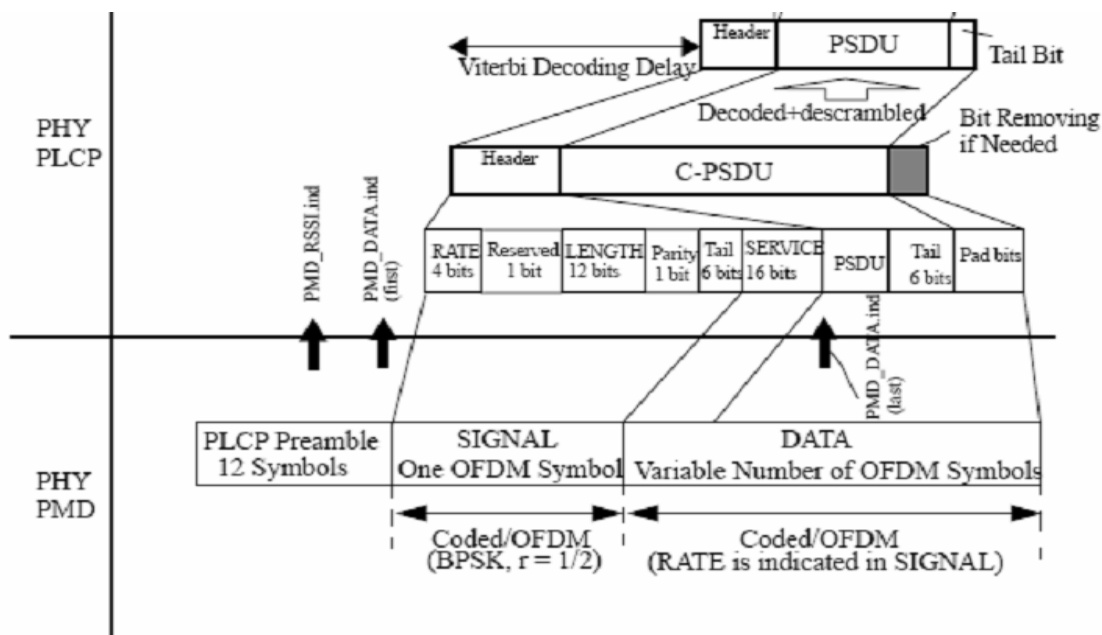


Figure 16 PLCP receive procedure – from Reference [4]

Figure 16 shows that the PLCP includes a 12 symbol preamble, a signal field that is convolutionally coded and modulated using BPSK, and data. Information about the data rate and length of the data field is included in the signal field, which, coupled with the service field, comprises the header of the OFDM packet. The coded data, consisting of OFDM symbols, pad, and tail bits, comprise the PLCP data service unit (PDSU). The OFDM receiver uses the preamble to generate the Received Signal Strength Indication (RSSI). In a wireless network, such as 802.11g, receivers will not know, a priori, when a packet will be received. A strong value for RSSI indicates the successful detection of an

OFDM packet, a necessary requirement before additional synchronization tasks can be attempted. Figure 17 closely depicts the preamble and header, including cyclic prefixes, and highlights the synchronization tasks that must be performed prior to successful demodulation of the transmitted data symbols.

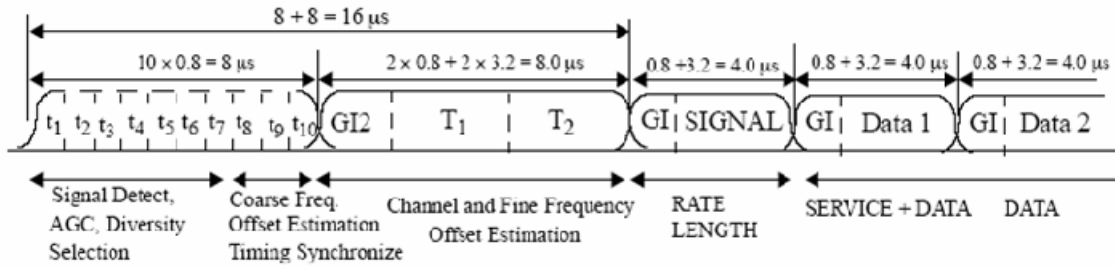


Figure 17 OFDM preamble and header structure – from Reference [4]

The preamble consists of 10 repetitions of the same symbol, labeled in the diagram as $t_1 \dots t_{10}$, 2 additional symbols repeated, labeled as T_1 and T_2 , and $GI2$, which serves as a cyclic prefix for symbol T_1 . Underneath the preamble symbols, the text in Figure 17 delineates the various synchronization tasks that can be performed to train the receiver. Due to the short duration of $0.8 \mu\text{sec}$ each for symbols $t_1 \dots t_{10}$, these are known as short training symbols while T_1 and T_2 , each with symbol duration $3.2 \mu\text{sec}$, are known as the long training symbols. [4]

It is these repeated symbols and the embedded pilot tones within the sub-carriers that many of the data-aided 802.11g synchronization algorithms exploit to quickly synchronize to the WLAN signal.

2. Short Training Symbols

The 10 short training symbols mentioned in the previous section form the beginning of the OFDM packet. Each of the symbols is 16 samples long and is used for packet detection, automatic gain control (AGC) and diversity selection. The symbols can also be exploited for coarse frequency synchronization and symbol timing. [12]

From Figure 17, each symbol has duration of $0.8 \mu\text{sec}$, resulting in a total duration of $8.0 \mu\text{sec}$ for the short training symbols. The symbols are generated by the 802.11g transmitter using 12 sub-carriers and modulated by the sequence, $S_{-26,26} = \sqrt{13/6} \{0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0\}$. The reason for multiplying the coefficient values of the sequence by $\sqrt{13/6}$ is to normalize the power of the resulting OFDM symbol because only 12 of 52 sub-carriers are used. The resulting symbol is mapped to the appropriate input of the 64 point IFFT that serves as the kernel for the transmitter OFDM processing. The mapping procedure is depicted below. [4]

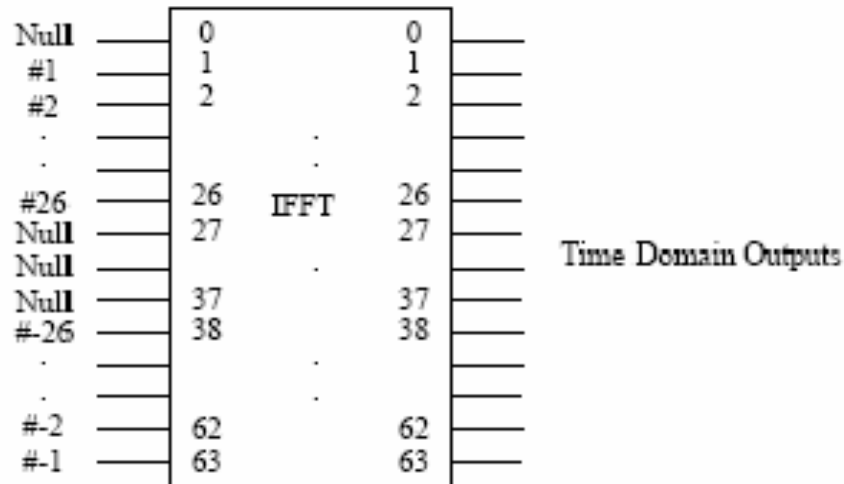


Figure 18 Coefficient mapping within the IDFT – from Reference [4]

In Figure 18, the values for coefficients 1...26 are mapped to the same numbered inputs while the coefficients $-26 \dots -1$ are mapped to inputs 38...63. The remaining inputs, 27...37 and 0 (dc) are set to zero. [4]

The in-phase component of the time domain output of the IFFT is shown in Figure 19.

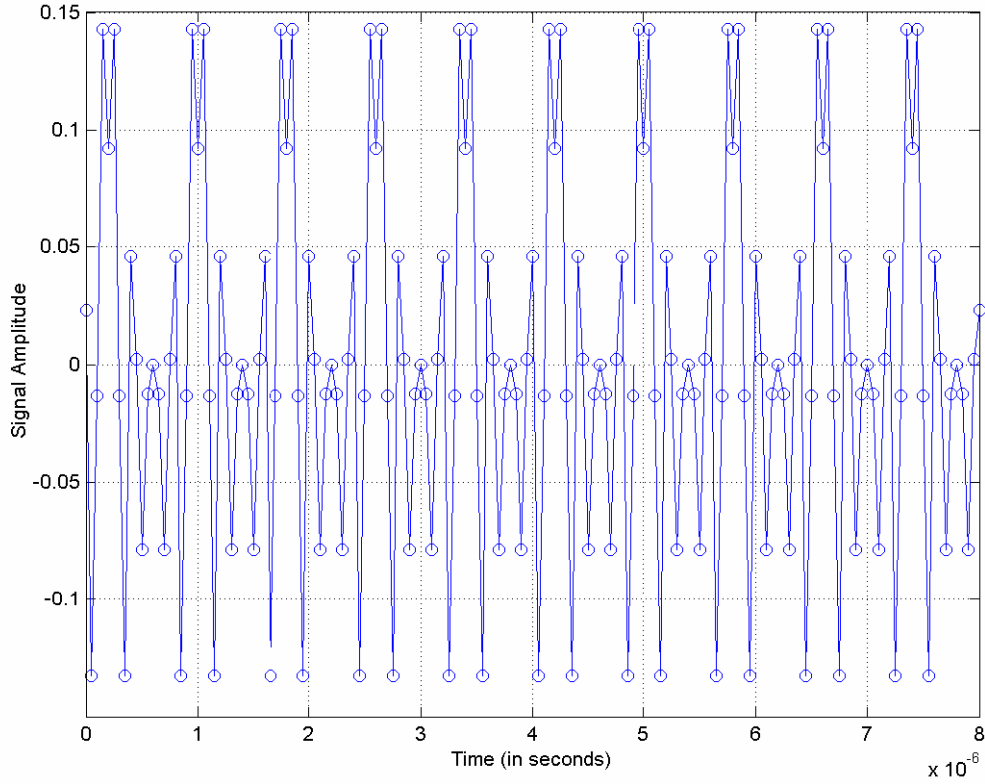


Figure 19 Short training symbols (time domain, in-phase)

As depicted in Figure 19 and in accordance with [4], the output of the IFFT has been cyclically extended to 161 samples, or 8 μsec , and the following windowing function has been applied:

$$P_T(n) = \begin{cases} .5 & n = 0 \\ 1 & 1 \leq n \leq 160 \\ .5 & n = 160 \end{cases} \quad (4.1)$$

In Equation (4.1), n represents the sample index and P_T is the windowing function discussed in Equation (2.6), now a function of sample index vice time. From Figure 19, the short training symbols are repetitive with ten repetitions, each 16 samples in duration. Since the 802.11g receiver knows a priori the sample values of the training symbols, this can be leveraged to help achieve synchronization [13].

3. Long Training Symbols

The long training symbols are included in the OFDM packet for fine frequency synchronization and are concatenated to the short training symbols [4]. Note that the long training symbols include a cyclic prefix designed to protect the long training symbols from ISI generated by the short training symbols. Figure 17 indicates that the duration of the cyclic prefix is $1.6 \mu\text{sec}$. This duration compares favorably to the values of delay spread in Table 1. The two long training symbols are identical and each symbol is $3.2 \mu\text{sec}$ long, resulting in a long training symbol total duration (including cyclic prefix) of $8.0 \mu\text{sec}$. The long training symbols are generated by the 802.11g transmitter using the sequence $L_{-26,26} = \{1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 0, 1, -1, -1, 1, 1, -1, 1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, -1, 1, -1, 1, 1, 1, 1\}$ [4]. After mapping to the appropriate IFFT input depicted in Figure 18, the time domain output is cyclically extended to 161 samples, creating the required cyclic prefix and ensuring a total duration of $8.0 \mu\text{sec}$. [4]

The time domain representation of the long training symbols appears in Figure 20.

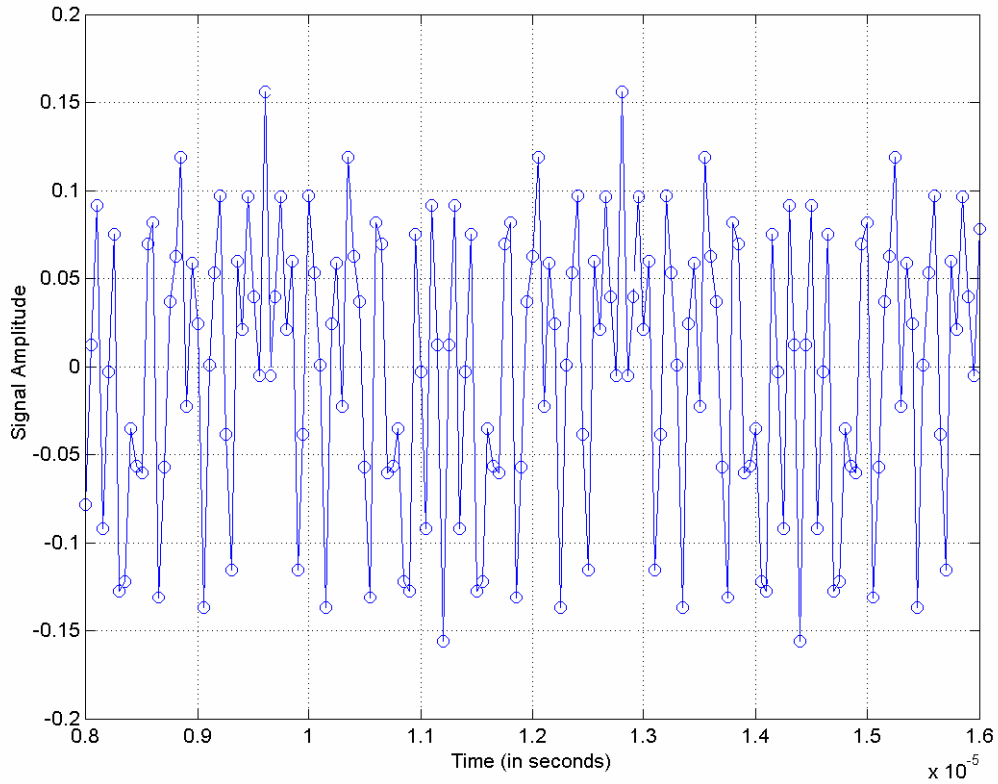


Figure 20 Long training symbols (time domain, in-phase)

In Figure 20, the time axis for the long training symbols reflects the position of the long training symbols within the OFDM packet. The last sample is at $16 \mu\text{sec}$, reflecting the last opportunity to use preamble data for synchronization. The 802.11g receiver uses these training symbols under the same principle as the short training symbols – a priori knowledge of the long training symbols can be used to achieve synchronization [8].

4. Pilot Tones

Each OFDM data symbol contains 4 dedicated sub-carriers that also carry a priori sample values. These pilot tones are used to prevent loss of phase and frequency synchronization. The pilots are inserted into sub-carriers -21, -7, 7, and 21 prior to IFFT processing. To reserve those sub-carrier positions, the sub-carriers carrying data, the logical sub-carriers 0...47, are remapped using the mapping function

$$M_{sc}(i) = \begin{cases} i-26 & 0 \leq i \leq 4 \\ i-25 & 5 \leq i \leq 17 \\ i-24 & 18 \leq i \leq 23 \\ i-23 & 24 \leq i \leq 29 \\ i-22 & 30 \leq i \leq 42 \\ i-21 & 43 \leq i \leq 47. \end{cases} \quad (4.2)$$

In Equation (4.2) i represents the sub-carrier index. Equation (4.2) shows that positions -21, -7, 7, and 21 are skipped to provide for pilot insertion. The pilot a priori values are $\{1,1,1,-1\}$ or $\{-1,-1,-1,1\}$ respectively. [4]

The polarity of the pilot tones is dependent upon the first chip of an internally generated pseudorandom scrambling sequence [4]. The scrambling is performed to avoid exceeding the average power spectral density of the OFDM spectrum during transmission [13].

5. Summary

Due to the nature of the packet-based WLANs, synchronization must occur rapidly and includes the detection of the actual packet as the first synchronization task. Furthermore, because of the nature of OFDM modulation, specific synchronization techniques must be employed that differ from techniques in single frequency communication. The governing IEEE standard does not explicitly mandate synchronization algorithms [8], but data-aided algorithms appear to be best suited for packet-based WLANs and generally provide good performance [12]. Furthermore, the IEEE standard does mandate various training symbols and pilot tones to aid data-aided synchronization. Since this overhead is mandatory, there is no additional cost for data-aided synchronization in terms of throughput. The remainder of this chapter will focus on synchronization algorithms that use training symbols in the OFDM packet's preamble and embedded pilot tones. Figure 21 depicts the time-frequency structure of an 802.11g OFDM packet, where all training symbols and pilots are shaded in red. It clearly illustrates the emphasis placed by the IEEE standard on data-aided synchronization.

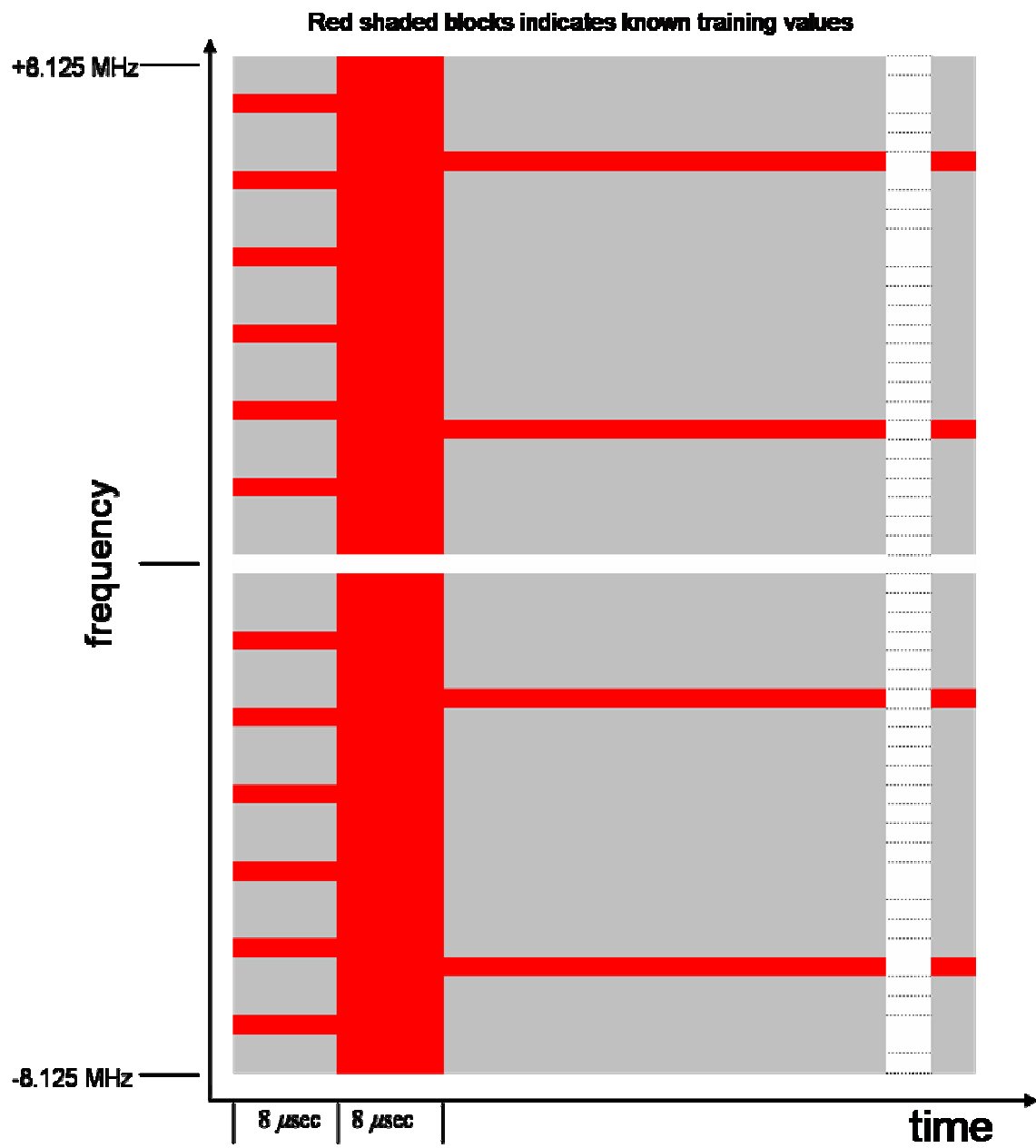


Figure 21 Time-frequency structure of OFDM 802.11g packet – from Reference [20]

C. IEEE 802.11G PACKET DETECTION

1. Introduction and Link Budget Analysis

Packet detection means finding the beginning of the OFDM packet. Without packet detection, the follow-on synchronization functions cannot be performed. [8]

Furthermore, as discussed previously, the packet-based nature of WLANs requires that synchronization be performed rapidly once the packet has been detected. The minimum average received signal power that the 802.11g OFDM receiver must detect before bit error rate becomes unacceptable is known as receiver sensitivity. The sensitivity is determined by the IEEE standard and is depicted in Table 2 [4].

Table 2 Receiver Performance Requirements – from Reference [4]

Data rate (Mbps)	Minimum Sensitivity (dBm)
6	-82
9	-81
12	-79
18	-77
24	-74
26	-70
48	-66
54	-65

These mandated receiver sensitivities will provide good bit-error-rate performance under typical indoor noise and channel conditions with the range increasing coupled with decreasing data rate [34]. Table 2's receiver performance requirement is predicated on a packet error rate of less than 10 % at a PSDU length of 1000 bytes [5]. The table implies that higher data rates are associated with greater signal strength at the receiver. Received signal strength decreases rapidly with increasing range and obstructions and is known as path loss [35].

A common expression for 2.4 GHz radio frequency path loss is

$$PL(\text{dB}) = PL(1) + 10\log(D^\gamma) \quad (4.3)$$

where PL is the path loss in dB, D is distance in meters, and γ is the path loss exponent. The value of γ depends on the attenuation characteristics of the wireless environment - $\gamma = 2$ for open space, $\gamma = 3.3$ for open office, and $\gamma = 4.5$ for home environments. $PL(1)$ is the path loss at 1 meter, which is equal to 41 dB. [35]

The power received for a typical 802.11g system can be written as

$$P_{rx} = \frac{P_{tx} G_{tx} G_{rx}}{PL}. \quad (4.4)$$

In Equation (4.4), P_{rx} is signal power out of the receiver's antenna, P_{tx} is transmitted power, G_{tx} is the transmitter's antenna gain, and G_{rx} is the receiver's antenna gain. Equation (4.4) may be re-written as

$$P_{rx}(\text{dB}) = P_{tx}(\text{dB}) + G_{tx}(\text{dBi}) + G_{rx}(\text{dBi}) - PL(\text{dB}). \quad (4.5)$$

Equation (4.5) shows that received power decreases with increasing path loss. Equation (4.3) showed that this path loss was dependent upon the characteristics of the wireless channel, but nonetheless, exponential with increasing distance. The detection range of the 802.11g system could be increased by improving receiver sensitivity. For indoor environments, Equation (4.4) can be re-written to solve for distance, D .

$$D = \sqrt[4.5]{\frac{P_{tx} G_{tx} G_{rx}}{P_{rx} PL(1)}} \quad (4.6)$$

In Equation (4.6), where D is in meters, if P_{tx} , G_{tx} , and G_{rx} are held constant, then to double reception range would require an improvement in receiver sensitivity, or decrease of acceptable P_{rx} , by a factor of approximately 22.6 ($= 2^{4.5}$), or 13 dB. For outdoor environments, the performance requirement would not be as onerous, due to the decreased path loss exponent. Figure 22 depicts the increased reception range with lower

minimum receiver sensitivity, assuming the bit-error-rate (BER) can be held to acceptable standards. The plot is based on the path loss model from Equation (4.3) using typical commercial sector values for P_{tx} , G_{tx} , and G_{rx} [36].

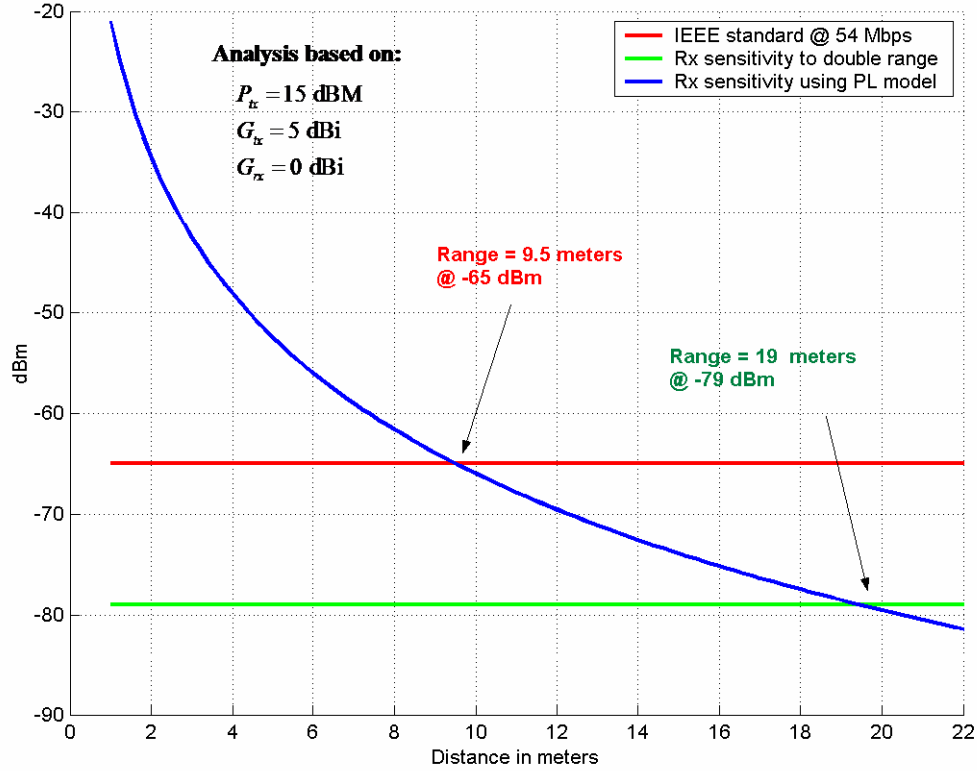


Figure 22 Power received as a function of distance

While doubling range performance requires a 13 dB decrease in sensitivity, the same analysis would show that a 3 dB decrease in receiver sensitivity yields, in an indoor environment, a commercially significant increase of 16 percent in reception range. The analysis depicted in Figure 22 only considers increasing receiver sensitivity to achieve increases in range performance. Commercial vendors obviously have the option to combine increased transmitter power or improved antenna gain to achieve the same effects. This thesis will only focus on synchronization techniques to improve receiver sensitivity and thereby increase reception range.

2. Packet Detection Theory

Packet detection is based on two complementary hypotheses – either a packet is present or it is not [8]. In order to build up to overall packet detection, we begin by considering the simpler detection problem of packet detection using the received signal value at single sample instant, t_m . This is can be mathematically represented as [12]

$$H_0 : r(t_m) = n(t_m) \quad (4.7)$$

$$H_1 : r(t_m) = s(t_m) + n(t_m) \quad (4.8)$$

where H_0 is hypothesis zero (i.e. no packet is present, only noise present) and H_1 is hypothesis one (a packet is present with noise), $r(t_m)$ is the received signal, $n(t_m)$ is noise, and $s(t_m)$ is the incoming packet. The receiver should choose the correct hypothesis with high probability, i.e. decide if a packet is present (D_1) or if a packet is not present (D_0).

The goal is for the receiver to maximize the probability that a packet is detected in the presence of noise, while minimizing the probability that a receiver mistakenly declares a packet present when only noise is present. These probabilities are depicted in Figure 23. The horizontal axis shows the values for decision statistic χ . Ordinarily χ is computed by the receiver, but for this initial step in the development of a packet detector, χ is merely a sample of the received signal. In other words, $\chi_m = r(t_m)$, and is dependent on whether $r(t_m)$ contains only noise (H_0) or the incoming packet and noise (H_1). χ_m is compared to threshold, y_T , to allow the receiver to make decision D_0 or D_1 . The blue curve on the left of Figure 23 represents the probability density function (PDF) for χ_m given hypothesis H_0 and is called the likelihood of H_0 . The red curve on the right represents the PDF for χ_m given hypothesis H_1 and is called the likelihood of H_1 . [10]

The curves in Figure 23 depict classic Gaussian probability density functions. This is because the values of $n(t_m)$ are Gaussian distributed. Because χ_m is simply a linear transformation of $n(t_m)$, it is also Gaussian distributed. The probability density function can be mathematically described as

$$p(\chi_m) = \frac{1}{\sqrt{2\pi\sigma_\chi^2}} \exp\left[-\frac{(\chi_m - m_\chi)^2}{2\sigma_\chi^2}\right],$$

where $p(\chi_m)$ is the notation used to describe the probability density function of Gaussian random variable χ_m , m_χ is the mean of χ_m , and σ_χ^2 is the variance of χ_m . Note that m_χ is zero under H_0 and non-zero under H_1 . [37]

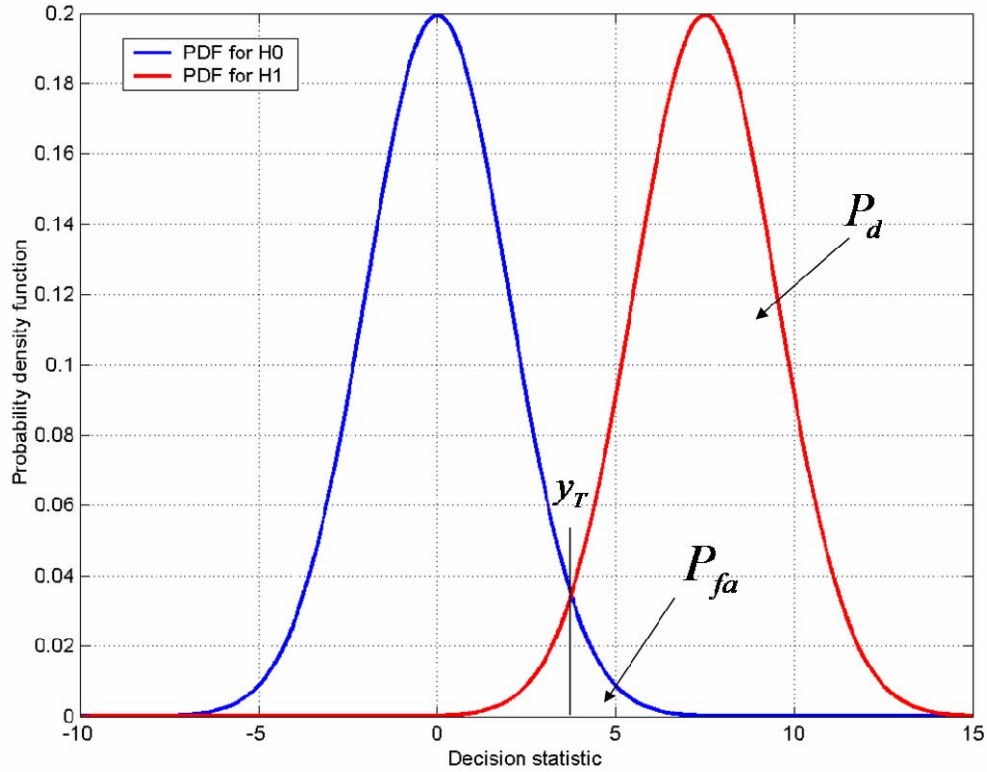


Figure 23 Probabilities of False Alarm and Detection in an AWGN channel

To maximize the probability that the receiver will detect a packet in the presence of noise is to maximize P_d , which is known as the probability of detection and is defined as

$$P_d = \Pr\{D_1|H_1\}. \quad (4.9)$$

To minimize the probability that the receiver will not mistakenly declare a packet present when only noise is present is to minimize the probability of false alarm, P_{fa} , which is defined as

$$P_{fa} = \Pr\{D_1|H_0\} \quad (4.10)$$

[37]

As Figure 23 shows, both probabilities cannot be optimized simultaneously. P_d is the area bounded by the likelihood of H_1 on the right and y_T on the left. P_{fa} is the area bounded by likelihood H_0 on the right and y_T on the left. Increasing P_d by lowering the detection threshold, y_T , results in an increase in the probability of a false alarm. Conversely, minimizing P_{fa} results in lowered P_d . The relationship between P_d and P_{fa} , and therefore D_0 and D_1 , is dependent upon χ_m and y_T . The relationship can be expressed as

$$\begin{array}{c} D_1 \\ > \\ \chi_m < y_T \\ < \\ D_0 \end{array} \quad (4.11)$$

[10]. The inequality relationship in Equation (4.11) indicates that the receiver makes decision D_1 if $\chi_m > y_T$ or D_0 if $\chi_m < y_T$. To achieve maximum performance from the receiver, χ_m must be computed with sufficient accuracy. Furthermore, a value of y_T

must be chosen to minimize error and optimize detection. While this thesis will explore various synchronization algorithms that are used to compute χ_m , we will first evaluate an optimal value for y_T .

a. Minimizing Error in Packet Detection

One popular criterion to optimize receiver performance is to minimize the probability of receiver error. To minimize the probability of error, the inequality

$$\Lambda = \frac{\frac{D_0}{Pr\{\chi < y_T | H_0\}}}{\frac{D_1}{Pr\{\chi > y_T | H_1\}}} > \frac{Pr\{H_1\}}{Pr\{H_0\}} \quad (4.12)$$

is formed. Equation (4.12) is known as the likelihood ratio test. Λ is the ratio of the conditional probabilities $Pr\{\chi_m < y_T | H_0\}$ and $Pr\{\chi_m > y_T | H_1\}$ for random variable χ at sample time $t = m$ given hypothesis H_0 or H_1 , and threshold y_T . $Pr\{H_0\}$ and $Pr\{H_1\}$ represent the a priori probabilities for H_0 and H_1 respectively. These conditional probabilities represent the likelihood of H_0 or H_1 . [10]

If the a priori probabilities of H_0 or H_1 are unknown, often the a priori probabilities are set equal to each other. In that case, Equation (4.12) may be re-written as

$$\frac{D_0}{Pr\{\chi_m < y_T | H_0\}} > \frac{D_1}{Pr\{\chi_m > y_T | H_1\}} \quad (4.13)$$

Receiver decision-making based on Equation (4.13) is known as the maximum likelihood criterion. [10]

The maximum likelihood criterion can be adjusted to fit other hypothesis testing algorithms, such as Bayesian, min-max, and Neyman-Pearson. The Bayesian

criterion assigns cost functions to incorrect decisions by the receiver with known a priori probabilities of the hypotheses. The min-max criterion is similar to the Bayesian criterion but is based on unknown a priori probabilities. The Neyman-Pearson criterion is based on maximizing P_d for a fixed P_{fa} . All three criteria result in a likelihood ratio test, similar to Equation (4.11) but with different y_T . In those cases, the value of y_T could be dependent on cost functions associated with incorrect decisions made by the receiver, a priori probabilities of H_0 or H_1 , or on a fixed value for P_{fa} . [37]

Because of the desire to maximize P_d to improve performance, the Neyman-Pearson criterion will be the focus of the remaining part of this section.

b. Neyman-Pearson Criterion

Because the Neyman-Pearson criterion assigns a fixed probability of false alarm and then attempts to maximize probability of detection using a likelihood ratio test, the likelihood ratio now becomes

$$\Lambda = \frac{\Pr\{\chi_m > y_T | H_1\}}{\Pr\{\chi_m < y_T | H_0\}} > \lambda_T, \quad (4.14)$$

where λ_T represents the required likelihood ratio threshold and Λ is the likelihood ratio [12]. λ_T is determined by false alarm or detection probabilities and is not equivalent to y_T , the signal threshold. If the receiver passes only RF in the band of interest, the noise terms in Equations (4.7) and (4.8) are assumed to be band limited AWGN (BLAWGN), with zero mean and a variance σ_n^2 , then the conditional probability density function of random variable χ_m given hypothesis H_1 can be described as [37]

$$p_{\chi_m > y_T | H_1}(\chi_m) = p_1(\chi_m) = \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{\frac{-(\chi_m - s_m)^2}{2\sigma_n^2}}. \quad (4.15)$$

Equation (4.15) shows that χ_m is Gaussian distributed with its mean value equal to the signal value s_m at sample time $t = m$ [12]. The probability density function, under hypothesis H_0 would be [37]

$$p_{\chi_m < y_T | H_0}(\chi_m) = p_0(\chi_m) = \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{(\chi_m)^2}{2\sigma_n^2}}. \quad (4.16)$$

In Equation (4.16), χ_m remains Gaussian distributed at sample time $t = m$, with the same variance but zero mean in accordance with H_0 . The likelihood ratio of the two density functions is

$$\Lambda(\chi_m) = \frac{p_1(\chi_m)}{p_0(\chi_m)} = \frac{\frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{(\chi_m - s_m)^2}{2\sigma_n^2}}}{\frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{(\chi_m)^2}{2\sigma_n^2}}} = e^{\frac{1}{2\sigma_n^2} [(\chi_m)^2 - (\chi_m - s_m)^2]}. \quad (4.17)$$

Taking the natural log of both sides and simplifying yields

$$\ln(\Lambda(\chi_m)) = \frac{1}{2\sigma_n^2} [s_m(2\chi_m - s_m)]. \quad (4.18)$$

Substituting Equation (4.18) into the left hand side of Equation (4.14) and solving for χ_m yields

$$\begin{array}{c} D_1 \\ \chi_m > \frac{\sigma_n^2}{s_m} \ln(\lambda_T) + \frac{s_m}{2} \\ D_0 \end{array} \quad (4.19)$$

Equation (4.19) shows that y_T does not equal λ_T [12]. Setting

$$y_T = \left(\frac{\sigma_n^2}{s_m} \right) \ln(\lambda_T) + \frac{s_m}{2} \quad (4.20)$$

yields the hypotheses depicted in Figure 23 and mathematically expressed as Equation (4.11). As Figure 23 depicts, the probability of false alarm may be defined as

$$P_{fa} = \int_{y_T}^{\infty} p_0(\chi_m) d\chi_m \quad (4.21)$$

and the probability of detection may be defined as

$$P_d = \int_{y_T}^{\infty} p_1(\chi_m) d\chi_m \quad (4.22)$$

[37]. Equations (4.21) and (4.22) show that a desired P_{fa} or P_d can be achieved by specifying a particular value of y_T . This is illustrated for clarity in Figure 24, where a detection threshold is depicted for a hypothetical decision statistic. The decision statistic is shown in blue before, during and after a packet is present at the receiver. Figure 24 suggests that a lower detection threshold will result in a greater P_d but also result in greater P_{fa} . The desired P_d will be driven by performance metrics for packet detection for a given signal strength in the presence of noise, which is based on the receiver's sensitivity.

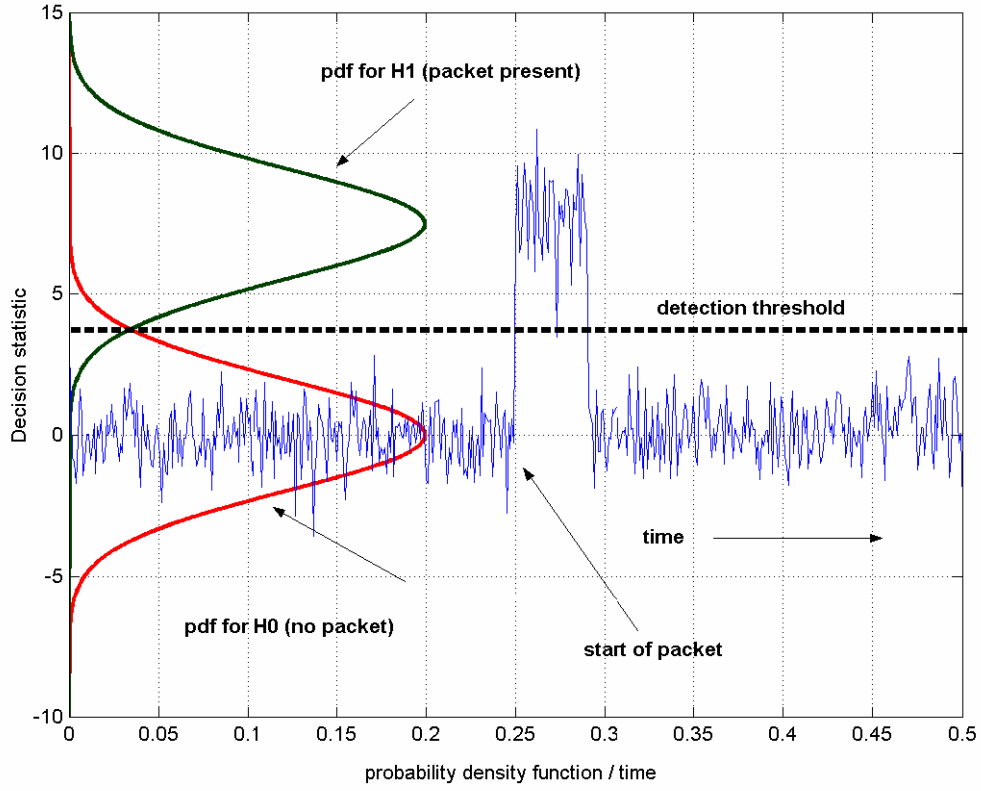


Figure 24 Signal threshold given hypothesis H_0 and H_1

Figure 24 also suggests that separating the probability density functions for hypotheses H_0 and H_1 would increase P_d while lowering P_{fa} . Changing the probability density function for H_1 could be easily done simply by increasing transmitter power. The IEEE standard, however, limits maximum transmitted power in accordance with government regulations [5]. Additionally, commercial vendors limit transmitted power to meet power management schemes [31], and must limit large peak-average power ratios within the OFDM transmitter to maintain performance and acceptable cost of the digital-to-analog converter (DAC) [31].

Another way to change the PDFs for H_0 and H_1 would be for the receiver to use multiple signal samples to compute the decision statistic. This would have the net effect of shifting the PDF for H_1 away from the PDF for H_0 while the detection

threshold could be maintained at the same level. The variance for H_1 and H_0 would be increased but P_d would also be increased without increasing P_{fa} . [12]

The OFDM 802.11g packet facilitates the use of multiple samples for packet detection. The “multiple samples” are the samples of the repetitive training symbols [31] that are mandated at the start of the packet and shall be the focus of the next section.

c. *Using Multiple Samples to Facilitate Packet Detection*

The previous section highlighted the fact that multiple samples of the incoming signal could be used to increase P_d while holding P_{fa} constant. For multiple samples, the noise is uncorrelated because it is assumed white [12].

With multiples samples, the two hypotheses from Equations (4.7) and (4.8) become

$$H_0 : r_m = n_m \quad (4.23)$$

$$H_1 : r_m = s_m + n_m \quad (4.24)$$

for all sample times $m = \{1, 2, \dots, M-1, M\}$. Equations (4.23) and (4.24) show that r_m is the sum of Gaussian random variable n_m and its mean value, s_m , when a packet is present. Hence, r_m is Gaussian distributed with a mean of zero under H_0 and a mean of s_m under H_1 . The variance of r_m under H_1 is given by

$$\sigma_{r_m}^2 = E[r_m - E(r_m)]^2 = E[s_m + n_m - s_m]^2 = E(n_m)^2 = \sigma_n^2, \quad (4.25)$$

where σ_n^2 is the noise variance under the same band limited conditions discussed previously. [37]

The joint probability density functions are

$$p_0(r_m) = \prod_{m=1}^M \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{(r_m)^2}{2\sigma_n^2}} \quad (4.26)$$

and

$$p_1(r_m) = \prod_{n=1}^M \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{(r_m - s_m)^2}{2\sigma_n^2}} \quad (4.27)$$

[37].

The likelihood ratio now becomes

$$\Lambda(r_m) = \frac{p_1(r_m)}{p_0(r_m)} = \frac{\prod_{n=1}^M \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{(r_m - s_m)^2}{2\sigma_n^2}}}{\prod_{n=1}^M \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{(r_m)^2}{2\sigma_n^2}}} = e^{\frac{-1}{2\sigma_n^2} \sum_{m=1}^M (-2s_m r_m + s_m^2)}. \quad (4.28)$$

Taking the natural log of both sides yields

$$\ln(\Lambda(r_m)) = \frac{-1}{2\sigma_n^2} \sum_{m=1}^M (-2s_m r_m + s_m^2). \quad (4.29)$$

Substituting Equation (4.29) into the left hand side of Equation (4.14) and solving for the summation of $\sum_{m=1}^M r_m s_m$ yields

$$X = \sum_{m=1}^M r_m s_m \underset{D_0}{\overset{D_1}{>}} \sigma_n^2 \ln(\lambda_T) + \frac{1}{2} \sum_{m=1}^M s_m^2 = y_T. \quad (4.30)$$

where X is the new decision statistic based on the sum of individual values of $r_m s_m$ taken from sample times $1, 2, \dots, M-1, M$. [37]

Note that Equation (4.30) simplifies to (4.11). Because the probability functions for H_0 and H_1 are known, y_T can be determined by specifying the desired probability of false alarm or the probability of detection [12].

Due to multiple sampling, Equation (4.21) becomes

$$P_{fa} = \int_{y_T}^{\infty} p_0(X) dX \quad (4.31)$$

and Equation (4.22) becomes

$$P_d = \int_{y_T}^{\infty} p_1(X) dX. \quad (4.32)$$

Considering X as a Gaussian random variable because it is the sum of Gaussian random variables [12], Equation(4.32) may be re-written as

$$P_d = \int_{y_T}^{\infty} \frac{1}{\sqrt{2\pi\sigma_X^2}} e^{-\frac{(X-m_X)^2}{2\sigma_X^2}} dX, \quad (4.33)$$

where m_X is the mean and σ_X^2 is the variance of X .

The area to the right of any threshold for a zero mean, normalized, Gaussian random variable is defined as

$$Q(y_T) = \frac{1}{\sqrt{2\pi}} \int_{y_T}^{\infty} e^{-\frac{u^2}{2}} du \quad (4.34)$$

which is sometimes referred to as the complementary cumulative distribution function for a zero mean, unit variance Gaussian random variable [34]. Substituting Equation (4.33)

into Equation (4.34) by letting $u = \frac{(X-m_X)}{\sigma_X}$ yields the following relationship:

$$P_d = Q\left(\frac{y_T - m_X}{\sigma_X}\right). \quad (4.35)$$

Solving for the threshold, Equation (4.35) becomes

$$y_T = m_X + \sigma_X Q^{-1}(P_d). \quad (4.36)$$

In Equation (4.36), m_X is defined as

$$m_X = E\{X\} = E\left\{\sum_{m=1}^M \chi_m\right\} = E\left\{\sum_{m=1}^M r_m s_m\right\} = E\left\{\sum_{m=1}^M (s_m + n_m) s_m\right\} = \sum_{m=1}^M \{s_m^2 + s_m E(n_m)\} = Ms^2 \quad (4.37)$$

where the last step holds if $s_m = s$ is a constant and M is the number of samples used in the detection algorithm. The variance of X is

$$\begin{aligned}
\sigma_X^2 &= E \left\{ \left[\sum_{m=1}^M (s + n_m) s - m_X \right]^2 \right\} \\
&= E \left\{ \left(\sum_{m=1}^M s^2 + s \sum_{m=1}^M n_m - M s^2 \right)^2 \right\} \\
&= E \left\{ \left(s \sum_{m=1}^M n_m \right)^2 \right\}
\end{aligned} \tag{4.38}$$

Equation (4.38) can be further simplified to

$$\begin{aligned}
\sigma_X^2 &= s^2 E \left\{ \left(\sum_{m=1}^M n_m \right) \left(\sum_{l=1}^M n_l \right) \right\} \\
&= s^2 \sum_{m=1}^M \sum_{l=1}^M E \{ n_m n_l \} \\
&= s^2 \sum_{m=1}^M \sum_{l=1}^M \left[E \{ n_m^2 \} \delta_{l,m} \right] \\
&= s^2 \left[M \left(\sigma_n^2 \right) \right] \\
&= s^2 M \sigma_n^2
\end{aligned} \tag{4.39}$$

where $\delta_{l,m}$ is the Kronecker delta function and is defined as

$$\delta_{l,m} = \begin{cases} 1 & \text{if } l = m \\ 0 & \text{if } l \neq m \end{cases} \tag{4.40}$$

[12]

Equation (4.37) shows that the mean value of X under H_1 can be increased by using multiple samples. Equation (4.39) shows that this does increase the variance of the decision statistic under H_1 . Using Equations (4.36), (4.37), and (4.39), the threshold, y_T , becomes

$$y_T = Ms^2 + \sqrt{M}s\sigma_n Q^{-1}(P_d) \quad (4.41)$$

[12].

For hypothesis H_0 , the mean of X is

$$m_X = E \left\{ \sum_{m=1}^M n_m \right\} = 0. \quad (4.42)$$

The variance of X under H_0 is the same as Equation (4.39). Substituting Equation (4.31) into Equation (4.34) yields

$$P_{fa} = Q \left(\frac{y_T}{\sigma_X} \right) \quad (4.43)$$

which, after making substitutions from Equations (4.39) and (4.41) simplifies to

$$P_{fa} = Q \left(\sqrt{M} \frac{s}{\sigma_n} + Q^{-1}(P_d) \right). \quad (4.44)$$

[12]

Equation (4.35) shows that a desired P_d can be obtained by adjusting the value of y_T . Using multiple samples to accomplish this, allows us to lower the threshold to obtain a desired P_d while minimizing P_{fa} . There are packet detection schemes that incorporate this algorithm. These schemes shall be covered in detail in the next sections.

3. Delay and Correlate Packet Detection

The Delay and Correlate packet detector is one example of a robust packet detection scheme that employs an algorithm similar to one discussed previously in this chapter. The Delay and Correlate detector is based on a double sliding window algorithm [8].

The double sliding window algorithm calculates received signal energy for two consecutive sliding windows of samples. A comparison of the ratio of received energy in both windows is used to compute the decision statistic. The Delay and Correlate

detector's windows are samples that comprise the signal and a delayed version of the signal. This detection scheme will exploit the repetitive preamble data (short training symbols) at the start of the OFDM packet. The signal flow and structure of the detector is illustrated below. [8]

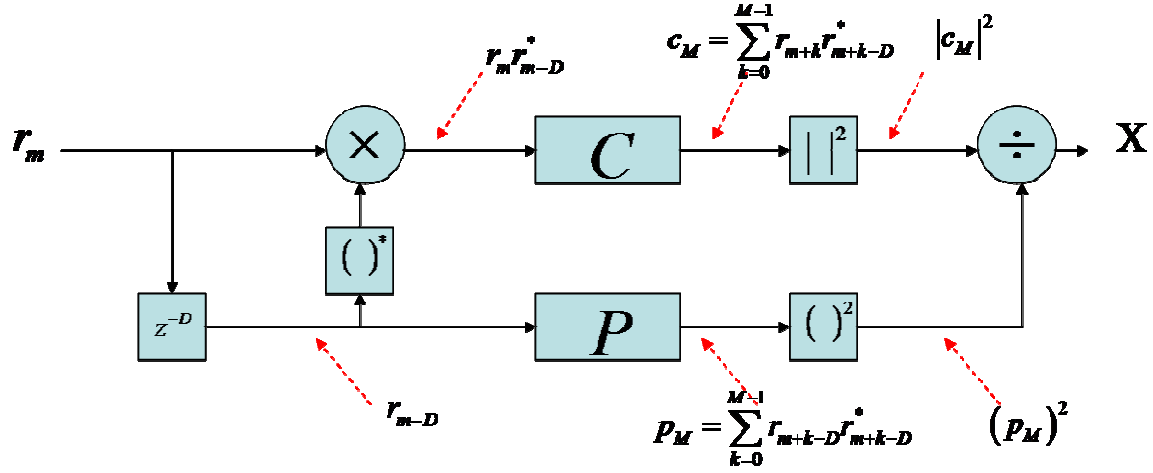


Figure 25 Signal flow structure of the Delay and Correlate algorithm – from Reference [8]

Figure 25 shows that the decision statistic is equal to

$$X = \frac{|c_M|^2}{(p_M)^2} \quad (4.45)$$

where M is the number of samples used to compute the decision statistic. The delay, z^{-D} in Figure 25, is used to create the delayed version of the received signal. To take advantage of the periodicity of the short training samples, the value of the delay is set to 16 samples. The C and P blocks are the two sliding algorithms used to compare whether a packet is present. The C algorithm, as shown in Figure 25, represents the cross correlation between the signal and the delayed version of the signal (hence the name Delay and Correlate detector). The P algorithm represents received signal energy during the cross correlation window and is used to normalize the cross correlation value, so it is not dependent upon the absolute received power level. [8]

Figure 26 shows the response of the decision statistic when the 802.11g short training samples are detected at $SNR = 10$ dB and $M = 8$. Included in Figure 26 are the results of the cross correlation window and received signal energy window for clarity. The short training samples are embedded in AWGN and are fully correlated at sample 1000, with the start of the preamble at sample 839.

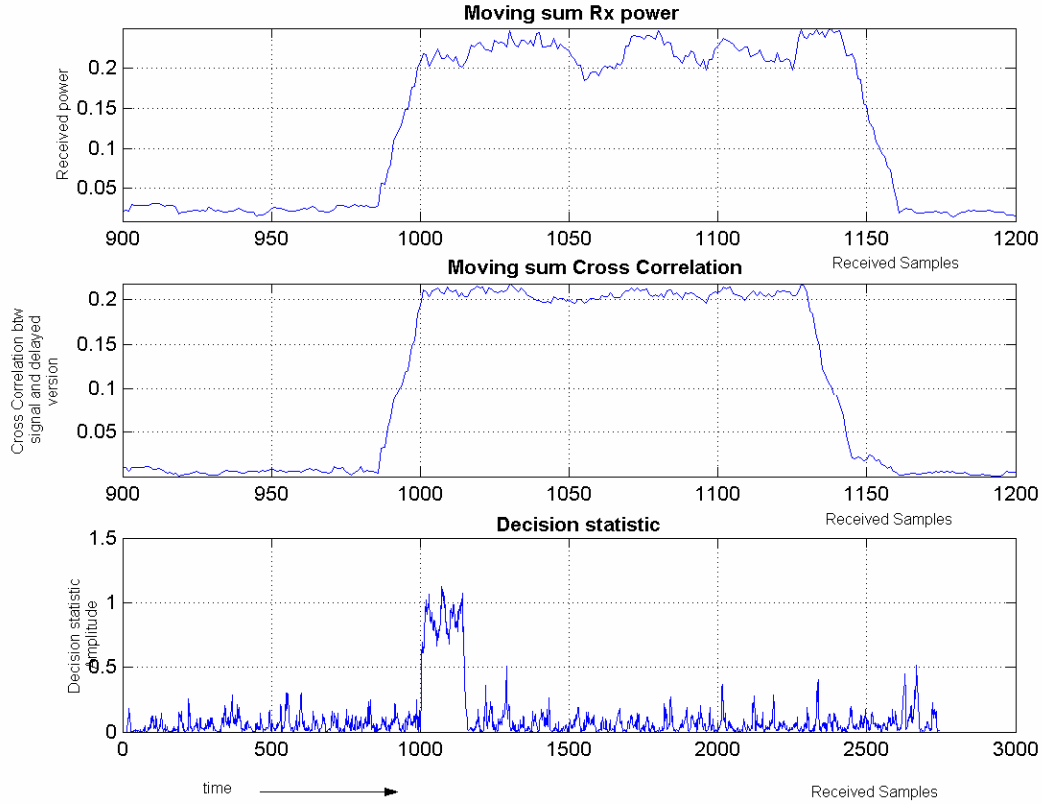


Figure 26 Delay and Correlate algorithm using 802.11g preamble and showing C_M , P_M , X

The low value of X prior to sample 1000 is explained by the fact that the cross correlation window consists only of uncorrelated noise, whose mean value is zero. At the start of the packet, the identical short training samples in the cross correlation window cause a sudden increase in the value of the decision statistic. This will give a good estimation of the start of the packet. [8].

Inspection of Figure 26 shows that the response of X permits the creation of a threshold to declare whether a packet is present or not. As previously discussed, the threshold can be determined from a desired P_d and acceptable P_{fa} . In this case, Equation (4.30) can be re-written as

$$X = \frac{\sum_{k=0}^{M-1} r_{m+k} r_{m+k-D}^*}{\sum_{k=0}^{M-1} |r_{m+k-D}|^2} \underset{D_0}{\overset{D_1}{>}} y_T. \quad (4.46)$$

In Equation (4.46), because the variance of the numerator and denominator on the left hand side of the inequality are much smaller than their means, the decision statistic can be approximated as a Gaussian random variable [38]. In that case, the probability distribution of X can be approximated as

$$f(X) = \frac{1}{\sqrt{2\pi\sigma_X^2}} e^{-\frac{(X-m_X)^2}{2\sigma_X^2}} \quad (4.47)$$

where m_X is the mean value and σ_X^2 is the variance of the Gaussian approximation of X . According to [38],

$$m_X = \frac{\sigma_s^4}{(\sigma_s^2 + \sigma_n^2)^2} \quad (4.48)$$

and

$$\sigma_X^2 = \frac{2\sigma_s^4 [(1+m_X)\sigma_s^2\sigma_n^2 + (1+2m_X)\sigma_n^4]}{M(\sigma_s^2 + \sigma_n^2)^2}. \quad (4.49)$$

In Equations (4.48) and (4.49), σ_s^2 is the variance of short training samples and σ_n^2 is the variance of the AWGN in the channel. Recognizing that

$$SNR = 10 * \log\left(\frac{\sigma_s^2}{\sigma_n^2}\right), \quad (4.50)$$

Equation (4.48) may be re-written as

$$m_x = \left[\frac{10^{SNR/10}}{10^{SNR/10} + 1} \right]^2 \quad (4.51)$$

where SNR is in units of dB. Equation (4.49) may be re-written as

$$\sigma_x^2 = \frac{2}{M} m_x \frac{[(1 + m_x)10^{SNR/10} + 1 + 2m_x]}{(10^{SNR/10} + 1)^2} \quad (4.52)$$

Equation (4.51) shows that the mean can be made to be only dependent upon SNR . Equation (4.52) shows that the variance of the decision statistic is inversely proportional to the number of samples used to compute the decision statistic. As more samples are used, this decreases σ_x^2 . From Equation (4.43), this shows that P_{fa} is reduced. Using Equations (4.51), (4.52), (4.36), and (4.43), a designer need only specify a desired P_d for the Delay and Correlate detector and for a given SNR , y_T and P_{fa} may be computed.

Figure 27 illustrates this below. The plots in Figure 27 depict y_T and P_{fa} when P_d is fixed and SNR is varied. The top plot shows y_T and P_{fa} when 8 samples ($M = 8$) are used to compute σ_x^2 . The bottom plot depicts the same when $M = 16$. Figure 27 clearly shows that using an increasing number of samples to compute X raises y_T and lowers P_{fa} . Close inspection of Figure 27 suggests that this appears most evident at single digit SNR when the performance of the 802.11g system is expected to be at its worst. This means that the Delay and Correlate detector is a promising synchronization technique for packet detection. The repetitive short training symbols, consisting of 161 samples, easily lend themselves to this type of algorithm and can be leveraged to increase range and performance.

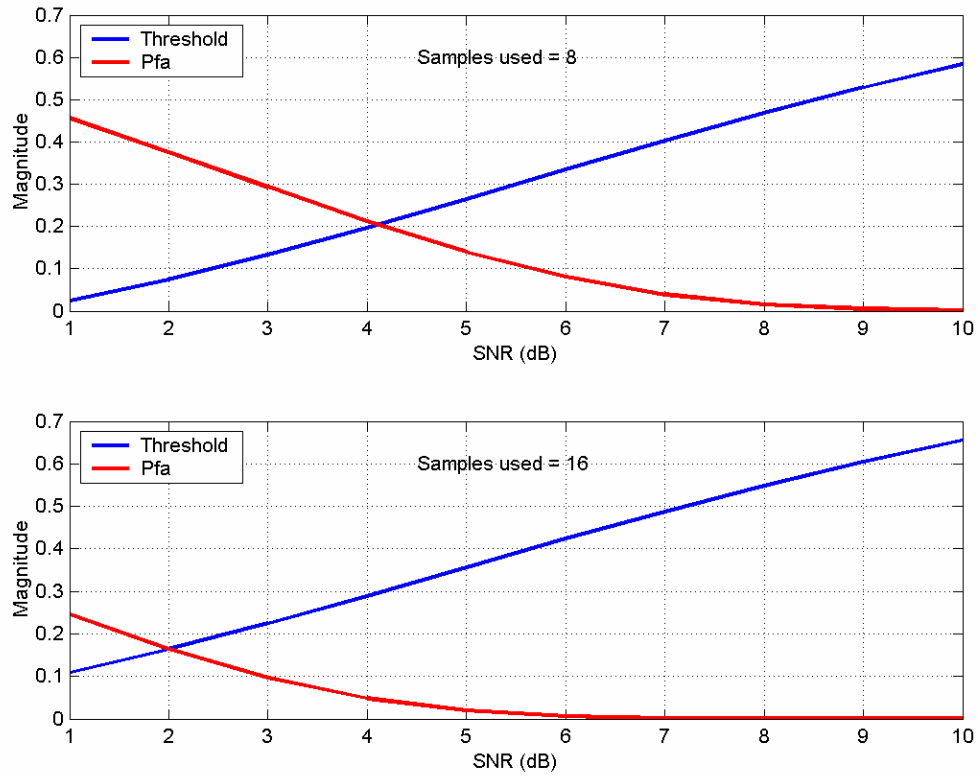


Figure 27 Detection Threshold and P_{fa} for Delay and Correlate Detector when

$$P_d = .9$$

4. Matched Filter Detector

Another type of detector that can be used to exploit the short training symbols is the Matched Filter detector. The Matched Filter detector algorithm is based on using the short training symbols sample values to detect the incoming signal. The signal structure and flow is depicted below.

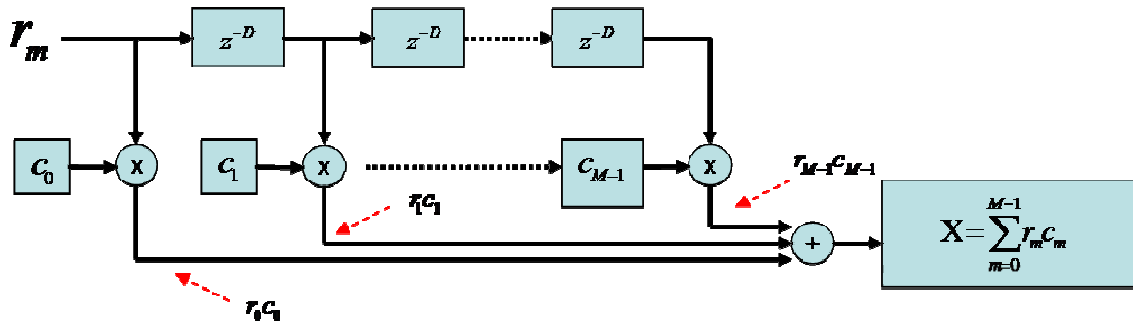


Figure 28 Matched Filter Detector – from Reference [13]

In Figure 28, the receiver uses delay blocks to multiply stored coefficients of short training sample values ($c_0, \dots, c_{M-1}$) with received values ($r_0, \dots, r_{M-1}$). The short training samples, as specified by the 802.11g standard, are periodic every sixteen samples [4], so in Figure 28, $M = 16$. These multiplied values are then summed to compute the decision statistic. There are 10 short training symbols, so in theory more samples could be used, thereby improving performance but at the cost of increased receiver complexity. This tradeoff will be addressed when synchronization performance is analyzed in the Chapter 0. Figure 29 depicts the absolute values of the decision statistic when $SNR = 10$ dB (as was done previously with the Delay and Correlate detector).

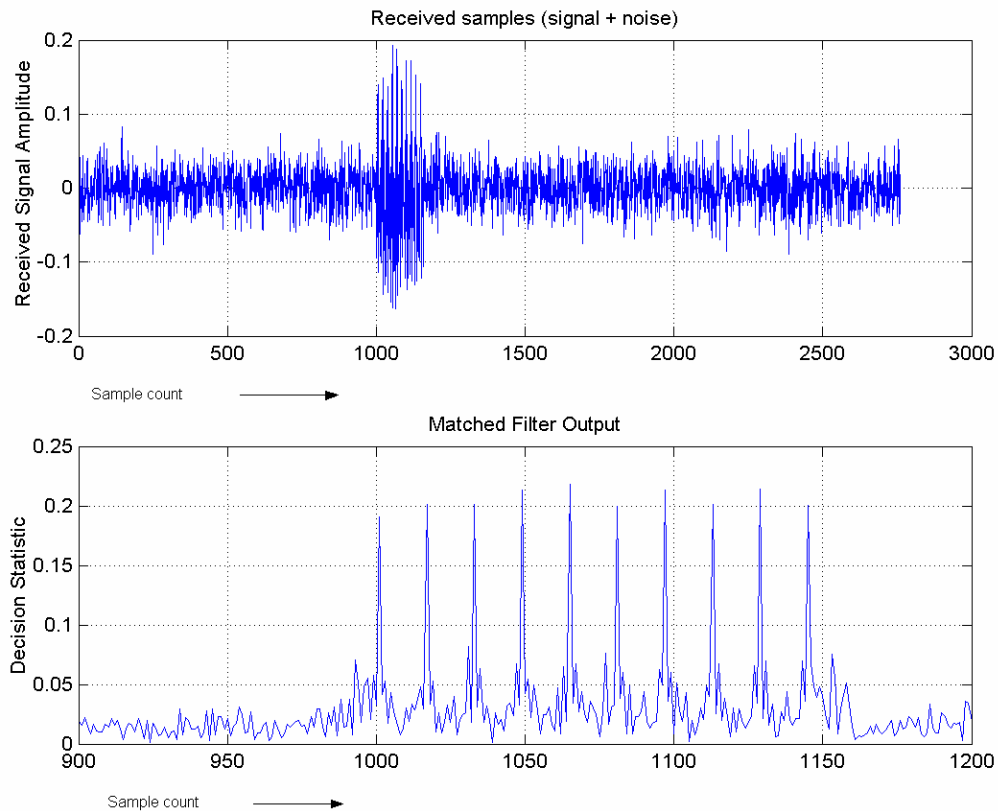


Figure 29 Decision statistic output for Matched Filter Detector

Figure 29 shows that there is a large increase in the value, or “spike” of the decision statistic when the portion of the received signal containing a short training symbol is correlated with the Matched Filter coefficients.

The Matched Filter coefficients are complex conjugates of the known training symbol values. As can be seen in Figure 29, the 10 short training symbols are easily detected at the tested SNR. The low values of the decision statistic between the “spikes”, or sidelobes, represent partial or no correlation between the Matched Filter coefficients and the signal. [13]

From Figure 29 it is apparent that the Matched Filter detector presents a viable technique to detect the 802.11g packet. The first spike will appear when the first sixteen

short training samples match the filter coefficients, indicating the start of the packet. Subsequent spikes could possibly be used to improve timing synchronization.

Using the same method of analysis from the previous section, the mean value of decision statistic, X , becomes

$$m_x = E \left\{ \sum_{m=1}^M X \right\} = E \left\{ \sum_{m=1}^M r_m s_m \right\} = E \left\{ \sum_{m=1}^M (s_m + n_m) s_m \right\}. \quad (4.53)$$

Because the noise in the channel is independent of the values of short training samples, Equation (4.53) can be simplified to

$$m_x = E \left\{ \sum_{m=1}^M (s_m + n_m) s_m \right\} = \sum_{m=1}^M \left\{ E(s_m^2) + E(s_m) E(n_m) \right\} = E \left\{ \sum_{m=1}^M s_m^2 \right\}. \quad (4.54)$$

Equation (4.54) is similar to Equation (4.37) except for the fact that the values of the short training samples are not constant during the window of summation. The values of the short training samples are known, however, and the expectation of their summation is simply the summation of the sample values. Hence, Equation (4.54) may be re-written as

$$m_x = \sum_{m=1}^M s_m^2. \quad (4.55)$$

Equation (4.55) is confirmed by [12]. The variance of the decision statistic can be expressed as

$$\begin{aligned}
\sigma_X^2 &= E[(X - m_X)^2] \\
&= E\left[\left(\sum_{m=1}^M (s_m + n_m)s_m - m_X\right)^2\right] \\
&= E\left[\left(\sum_{m=1}^M s_m^2 + \sum_{m=1}^M s_m n_m - m_X\right)^2\right] \\
&= E\left[\left(m_X + \sum_{m=1}^M s_m n_m - m_X\right)^2\right] \\
&= E\left[\left(\sum_{m=1}^M s_m n_m\right)^2\right] \\
&= E\left[\sum_{m=1}^M s_m n_m \sum_{p=1}^M s_p n_p\right] \\
&= E\left[\sum_{p=1}^M \sum_{m=1}^M s_m s_p n_m n_p\right] \\
&= \sum_{p=1}^M \sum_{m=1}^M s_m s_p E[n_m n_p] \\
&= \sum_{p=1}^M \sum_{m=1}^M s_m s_p \delta_{m,p} \sigma_n^2 \\
&= \sum_{m=1}^M s_m^2 \sigma_n^2 \\
\sigma_X^2 &= m_X \sigma_n^2.
\end{aligned} \tag{4.56}$$

In Equation (4.56), $\delta_{m,p}$ is the Kronecker delta function defined in Equation (4.40). The analysis in Equation (4.56) is similar to Equations (4.38) and (4.39), but once again, s_m is not a constant value. Now that the mean value and variance of the decision statistic for the Matched Filter detector have been established, Equations (4.36) and (4.43) can be used to compute detection threshold and P_{fd} for a given P_d .

In the case of the Matched Filter detector, the coefficients of the Matched Filter are the complex conjugates of the known periodic short training samples. Hence, an actual value can be computed for Equation (4.55). In this case, $m_X = .2031$. Furthermore, Equation (4.56) can be simplified to $\sigma_X^2 = .2031\sigma_n^2$. Figure 30 shows how

detection threshold and probability of false alarm varies with SNR when $P_d = .9$. If the mean of the signal strength is expressed in volts, then the computed threshold can be expressed in volts as shown below.

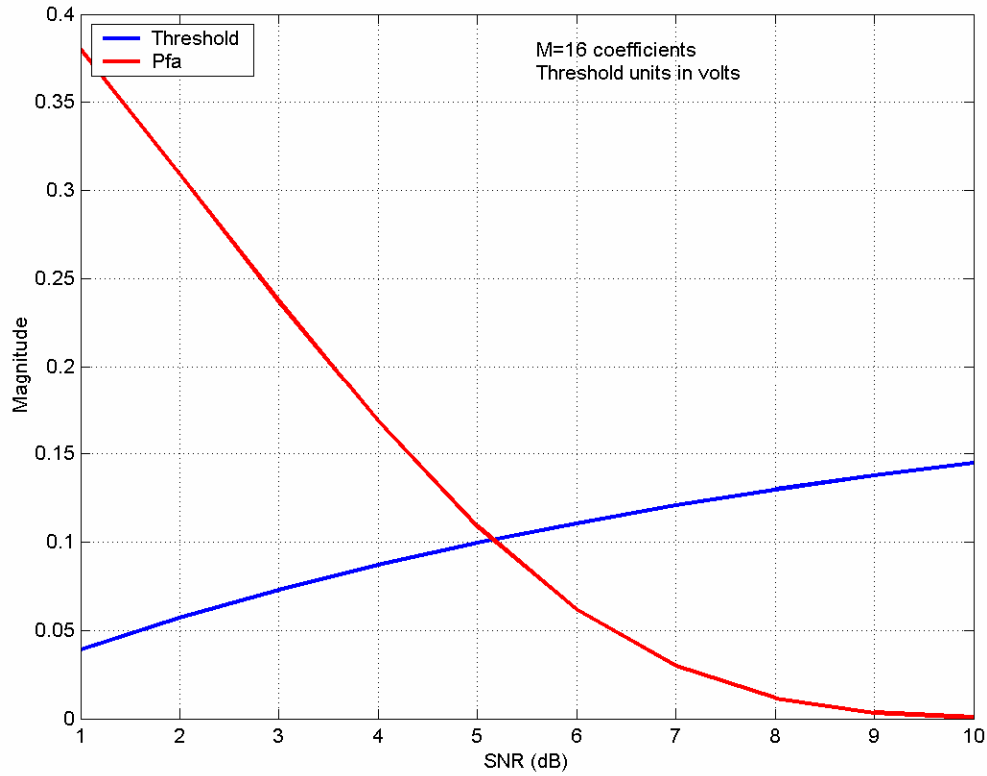


Figure 30 Matched Filter y_T and P_{fa} for varying SNR

Figure 30 suggests that at low SNR, the Matched Filter detector has a relatively high false alarm rate and a low threshold of detection is required to ensure desired P_d is achieved. As Figure 30 shows, with increasing SNR, P_{fa} decreases, as expected, while the detection threshold can be raised.

5. Summary and Comparison of Packet Detection Techniques

Link budget analysis suggested that improving packet detection would improve range and performance of the 802.11g system. This led to a review of basic detection

theory, the role that packet detection plays in synchronization, and a review of various packet detection algorithms. The Neyman-Pearson algorithm seems to be promising because it focuses on maximizing P_d for a particular acceptable P_{fa} . Costs associated with incorrect decision making need not be computed. Finally the use of multiple samples to compute the decision statistic was examined. High P_d could be specified without increasing P_{fa} by using multiple samples. The Delay and Correlate detector, as described in [8], was examined in detail. It employs multiple samples to optimize performance. Subsequent analysis depicted in Figure 27 showed that the Delay and Correlate detector could meet a desired P_d at low SNR. While threshold values remained low, P_{fa} was held to an acceptable value by using multiple samples. The Matched Filter detector, as described by [13], also uses multiple samples as it seeks to leverage the 16 sample periodicity of the short training samples. Unfortunately, inspection of Figure 30 revealed much higher values of P_{fa} at low SNR compared to the Delay and Correlate detector values shown in Figure 27. This suggests that the Matched Filter detector's performance lags that of the Delay and Correlate detector. This shall be confirmed in Chapter V, Synchronization Performance. In Chapter V, multiple simulations for both detectors shall be run at low SNR and detection and false alarm shall be verified. Additionally, the Delay and Correlate detector and the Matched Filter detector will be evaluated using more than one 16 sample short training symbol.

D. FREQUENCY SYNCHRONIZATION

1. Introduction

As shown in Chapter III, Equation (3.1), OFDM receivers suffer from degradation in SNR due to lack of frequency synchronization. This lack of frequency synchronization, or frequency offset, is a critical factor in OFDM system design. Failure to synchronize in frequency results in ICI and violates the orthogonality of the sub-carriers. [37]

a. *Single and Multi-Carrier Frequency Synchronization Differences*

At the heart of nearly all synchronization circuits is some version of a phase-locked loop or its functional equivalent [10]. Many single carrier systems use a narrow band phase-locked loop to acquire and track the carrier, thereby maintaining frequency synchronization [12]. The phase-locked loop is used to generate an error signal proportional to the difference between the received frequency and a reference frequency. This error signal is then used to drive a voltage-controlled oscillator to match the received frequency and drive the error signal to zero. Unfortunately, in OFDM systems, the multiple sub-carriers present in the received signal make it extremely difficult to create a single error signal that is proportional to the difference between the received signal and a reference signal [12]. Consequently, various frequency synchronization techniques have been proposed.

b. *General Frequency Synchronization Technique in OFDM*

To achieve frequency synchronization with minimal computational effort by the OFDM receiver, the synchronization process is divided into two phases – acquisition and tracking. This also serves to minimize the amount of redundant information that must be added to the transmitted packet. In the acquisition (or coarse) phase, an initial estimate of the frequency offset must be made. Consequently, the acquisition phase is characterized by more complex and robust synchronization algorithms which focus on those portions of the packet that contain data that can be leveraged for synchronization. Once acquisition has been achieved, tracking (or fine synchronization) can be maintained with a less complex algorithm that only needs to correct small frequency deviations. [39]

Furthermore, frequency synchronization techniques can be characterized as data-aided (relying on special training symbols embedded within the packet) or dependent on redundancy (e.g. cyclic prefix based) within the packet [33]. Data-aided techniques will achieve synchronization using the 802.11g mandated training symbols, as discussed in the previous chapter. Redundancy techniques exploit the cyclic prefix that is

part of every 802.11g OFDM symbol. The 802.11g standard does not specify a particular synchronization technique; however latter short training symbols are intended by the 802.11g standards group for coarse frequency synchronization while the two long training symbols are intended for fine frequency synchronization [40]. As discussed in earlier in this chapter, data-aided synchronization offers a strong candidate for OFDM WLAN applications because of the repetitive nature of the training symbols within the preamble. The cost in data efficiency must be borne anyway to remain in compliance with the IEEE standard. Cyclic-prefix based synchronization offers the advantage of not having to wait for the arrival of the OFDM preamble, but the 802.11g short training symbols lack a cyclic prefix. . Synchronization schemes exploiting the short training symbols are categorized as data-aided and will be the focus of this thesis.

c. Summary of Various Synchronization Algorithms

Per the IEEE 802.11g standard, any coarse algorithm must be sufficiently robust to correct for a maximum frequency offset of 120 kHz, as shown in Equation (3.2). Ideally, the frequency offset should be computed prior to FFT processing by the receiver [8]. Several frequency synchronization algorithms have been suggested for OFDM; Daffara and Chouly [41] and Moose [16] have suggested a maximum likelihood (ML) estimator to compute frequency offset. Li, Liu, and Giannakis [40] have recommended a non-linear least squares approach. Schmidl and Cox [38] and Van de Beek, Sandell, and Börjesson [42] have suggested a CP synchronization scheme that exploits the redundancy between samples in the data and CP portions of the OFDM frame. The first two algorithms shall be addressed in this chapter as the focus remains on data-aided synchronization schemes.

2. Maximum Likelihood Synchronization

The maximum likelihood estimator is one of several promising techniques that have been presented by a variety of researchers, in slightly different forms [8]. The underlying premise is to exploit the repetitive training symbols (data-aided) or repeated data symbols (redundancy – CP based). Any phase difference between the sub-carriers of

repeated OFDM symbols can be used to estimate the frequency difference. This is a reasonable assumption because from Proakis [11], the received signal may be expressed as

$$r(t) = s(t - \tau) + n(t) \quad (4.57)$$

where τ is the propagation delay, $n(t)$ is the AWGN of the channel and

$$s(t) = \text{Re} \left[s_l(t) e^{j2\pi f_{tx} t} \right]. \quad (4.58)$$

In Equation (4.58), $s(t)$ is the transmitted packet and $s_l(t)$ is the equivalent low-pass signal of $s(t)$, and f_{tx} is the carrier frequency of the transmitted signal. Consequently, Equation (4.57) may be re-written as

$$\begin{aligned} r(t) &= \text{Re} \left\{ [s_l(t - \tau) + n_l(t)] e^{j2\pi f_{tx}(t - \tau)} \right\} \\ &= \text{Re} \left\{ \left[(s_l(t - \tau) + n_l(t)) e^{-j2\pi f_{tx}\tau} \right] e^{j2\pi f_{tx}t} \right\} \\ &= \text{Re} \left\{ \left[s_l(t - \tau) e^{-j\phi} + n_l(t) e^{-j\phi} \right] e^{j2\pi f_{tx}t} \right\} \end{aligned} \quad (4.59)$$

where ϕ is the carrier phase error and is equal to $2\pi f_{tx}\tau$ [11]. If there is a difference between f_{tx} and f_{rx} , the receiver's center frequency, an additional error will result as the received signal is down converted to baseband at the receiver. This is caused by the oscillator mismatch, between transmitter and receiver [8]. The preceding statement ignores any type of Doppler effect between receiver and transmitter. As shown in Chapter 0, these effects were considered negligible at typical walking speeds and are not considered for the remainder of this thesis.

Disregarding noise (letting $n_l(t) = 0$) and letting $\tau = 0$ at sample time $t = mt_s$ for simplicity, once the signal in Equation (4.59) has been received and down converted to baseband, it could be multiplied by the complex conjugate of a repeated symbol. If there were no oscillator mismatch between repeated symbols, i.e. $f_{tx} - f_{rx} = 0 = \Delta f$, then the result of the correlation would be strictly real. If there were an oscillator mismatch, i.e. $\Delta f \neq 0$, then any remaining imaginary portion is dependent upon the phase error, which

would be proportional to the oscillator mismatch. This assumes that any fading is flat fading. Consider the sampled complex baseband signal with an oscillator mismatch between transmitter and receiver (disregarding noise again). In this case, the sampled signal, at $t = mt_s$, would be

$$\begin{aligned} r_m &= s_m e^{j2\pi f_{tx} mt_s} e^{-j2\pi f_{rx} mt_s} \\ r_m &= s_m e^{j2\pi (f_{tx} - f_{rx}) mt_s} \\ r_m &= s_m e^{j2\pi (\Delta f) mt_s} \end{aligned} \quad (4.60)$$

where Δf is the frequency offset between transmitter and receiver caused by oscillator mismatch and t_s is the sample duration [8]. Doppler effects are ignored.

Using intermediate variable z , an estimation of the frequency offset can be formed as

$$\begin{aligned} z &= \sum_{m=0}^{M-1} r_m r_{m-D}^* \\ z &= \sum_{m=0}^{M-1} s_m e^{j2\pi \Delta f m t_s} \left(s_{m-D} e^{j2\pi \Delta f (m-D) t_s} \right)^* \\ z &= \sum_{m=0}^{M-1} s_m s_{m-D}^* e^{j2\pi \Delta f m t_s} e^{-j2\pi \Delta f (m-D) t_s} \\ z &= e^{j2\pi \Delta f D t_s} \sum_{m=0}^{M-1} |s_m|^2 \\ z &= \left[\cos(2\pi \Delta f D t_s) + j \sin(2\pi \Delta f D t_s) \right] \sum_{m=0}^{M-1} |s_m|^2 \end{aligned} \quad (4.61)$$

Equation (4.61) shows that intermediate variable z consists of a real and imaginary component. If there were no frequency offset ($\Delta f = 0$) z would be purely real. Instead, the imaginary portion of z is dependent upon the frequency offset. If the sampling rate is high ($t_s \ll 1$), then the imaginary portion of z can be simplified because if $2\pi \Delta f D t_s \ll 1$, then $\sin(2\pi \Delta f D t_s) \approx 2\pi \Delta f D t_s$ and $\cos(2\pi \Delta f D t_s) \approx 1$. Consequently, we can estimate the frequency offset from the imaginary portion of z as

$$\hat{\Delta f} = \frac{1}{2\pi Dt_s} \angle z, \quad (4.62)$$

where $\angle$ is defined as the angle argument of complex number z . [8]

This estimator is referred to as the maximum likelihood estimator because the received values can be modeled using the conditional probability distribution of the received values given Δf , or $P\{r_n|\Delta f\}$. In this case, Δf is considered unknown, but deterministic. $\hat{\Delta f}$ is the value of Δf that maximizes the $P\{r_n|\Delta f\}$. [11]

a. Limits on the Operating Range of the ML Estimator

ML estimators, such as the one described above, can only correct up to a certain maximum frequency offset. This is because

$$\angle z = 2\pi Dt_s \hat{\Delta f} \quad (4.63)$$

which is only unambiguously defined in the range $[-\pi, \pi]$. If the frequency offset between repeated samples exceeds

$$|\hat{\Delta f}| \leq \frac{\pi}{2\pi Dt_s} = \frac{1}{2Dt_s}, \quad (4.64)$$

then the frequency offset estimate will be inaccurate. Typically, the maximum allowable frequency offset is normalized to sub-carrier spacing. If the delay, D , is equal to symbol length, which for immediately repeated samples means $D = 1$, Equation (4.64) may be re-written as

$$|\hat{\Delta f}| \leq \frac{1}{2} f_s. \quad (4.65)$$

Hence, this type of estimator is usually limited to a maximum frequency offset that is one half of the sample rate if the frequency correction is performed prior to the DFT demodulation by the receiver. [12]

From Equation (4.64), when D becomes larger, the range of the estimator is reduced. For data-aided synchronization, if the short training samples are used for the

ML estimator ($D = 16$, $t_s = 50 \text{ } \mu\text{sec}$) [8], then Equation (4.64) shows that the maximum frequency offset becomes 625 kHz. Redundancy synchronization algorithms that use the cyclic prefix have a delay that is longer than the symbol period and consequently have less range than the short training samples. This can be alleviated by increasing the sampling rate. The long training samples, as discussed previously, are periodic every 64 samples ($D = 64$), giving the estimator a range of up to 156.25 kHz. From the numbers discussed above, it is apparent that either data-aided synchronization technique meets the 802.11g standard. The performance such a time-domain ML estimator will be discussed in the next section.

b. Implementation of the ML Estimator

As discussed on page 75, frequency synchronization is usually accomplished by the OFDM receiver in two phases – acquisition and tracking. The short training symbols, with their 16 sample periodicity at the start of the frame, can be exploited to accomplish ML frequency synchronization [8]. Because the short training symbols are at the beginning of the frame, they lend themselves well to the acquisition task. Figure 31 depicts the output of MATLAB code, in which the ML algorithm has been used to generate a frequency offset correction that is applied to subsequent incoming short training samples. The top graph in Figure 31 shows the received short training symbols (161 samples) after passing through an AWGN channel (SNR is 10 dB). The middle graph shows those same received samples when the OFDM receiver's oscillator is experiencing a frequency offset of 40 percent of sub-carrier spacing. Recall from Equation (3.2) that the worst case tolerable frequency offset per the 802.11g standard is 120 kHz, which represents only 38.4 percent relative of the sub-carrier spacing. So the middle graph represents a worst case scenario for frequency offset, rounded up for simplicity. To facilitate comparison between the ML algorithm and other frequency synchronization techniques, a standardized frequency offset of -40% of sub-carrier spacing, or 125 KHz is employed. Such a frequency offset will cause a constellation rotation of 2π during the 160 short training symbol preamble, resulting in a severe degradation in performance. The bottom graph of Figure 31 shows the short

training samples with the ML frequency offset correction applied. To show the effects of the ML algorithm on short training symbol reception for clarity, the ML algorithm uses an “on-the fly” correction that applies the computed frequency offset correction to subsequent incoming short training symbols. This serves to significantly mitigate the effects of the original frequency offset despite the noisy channel.

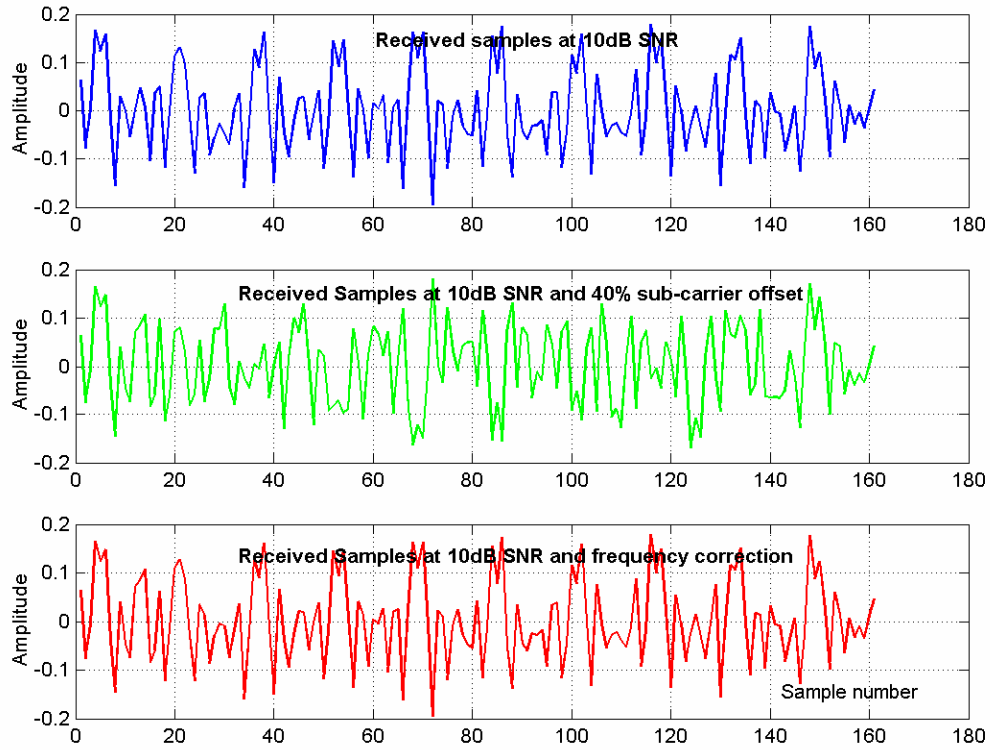


Figure 31 Frequency offset synchronization using the ML algorithm

This is evident because the samples depicted in the bottom third of Figure 31 more closely resemble the original noisy samples without the frequency offset (top third of the figure). The ML derived frequency offset correction is applied to drive the frequency offset ideally to exactly zero. This is never accomplished due to the effects of noise. The noise interference introduces inaccuracies in the initial frequency offset estimation and a continual, or “on the fly” frequency correction algorithm must be used for follow on short training symbols. Figure 32 shows the resulting frequency offset

correction relative to sub-carrier spacing for the short training symbols received in Figure 31 above. Figure 32 shows that the initial -40% frequency offset is detected and corrected within the arrival of the first two short training symbols. The residual frequency offset correction is applied to maintain the correction factor until completion of the short training symbols.

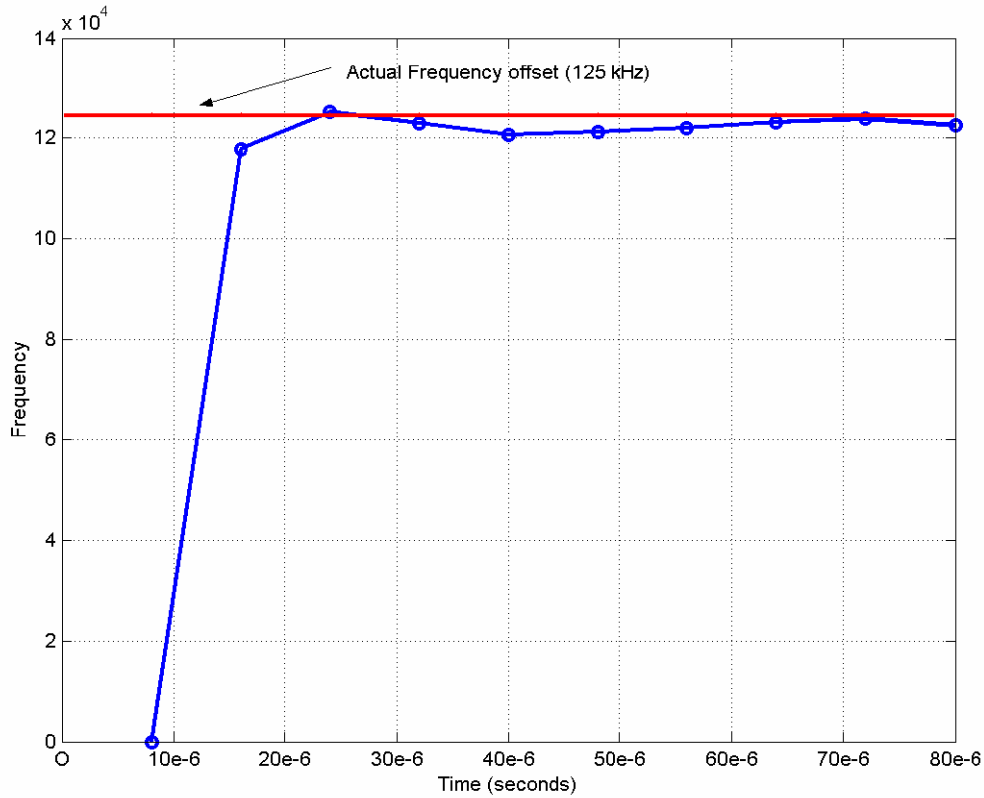


Figure 32 ML Frequency correction for 125 kHz offset (10 dB SNR)

Additionally, Figure 32 shows, the final frequency offset is less than 1% of the sub-carrier spacing, or 3125 KHz, when SNR = 10dB. The horizontal axis in Figure 32 depicts the frequency offset correction required based on the current and previous OFDM short training symbols. Data points in Figure 32 coincide with the arrival of short training symbols at which point the frequency offset estimate is refined and a new frequency correction is generated. The fidelity of the synchronizer depends on the number of short training symbols. If 4 or 5 symbols are only detected, vice the full

complement, performance of the synchronizer suffers. The signal flow and structure of this ML frequency synchronizer is depicted below.

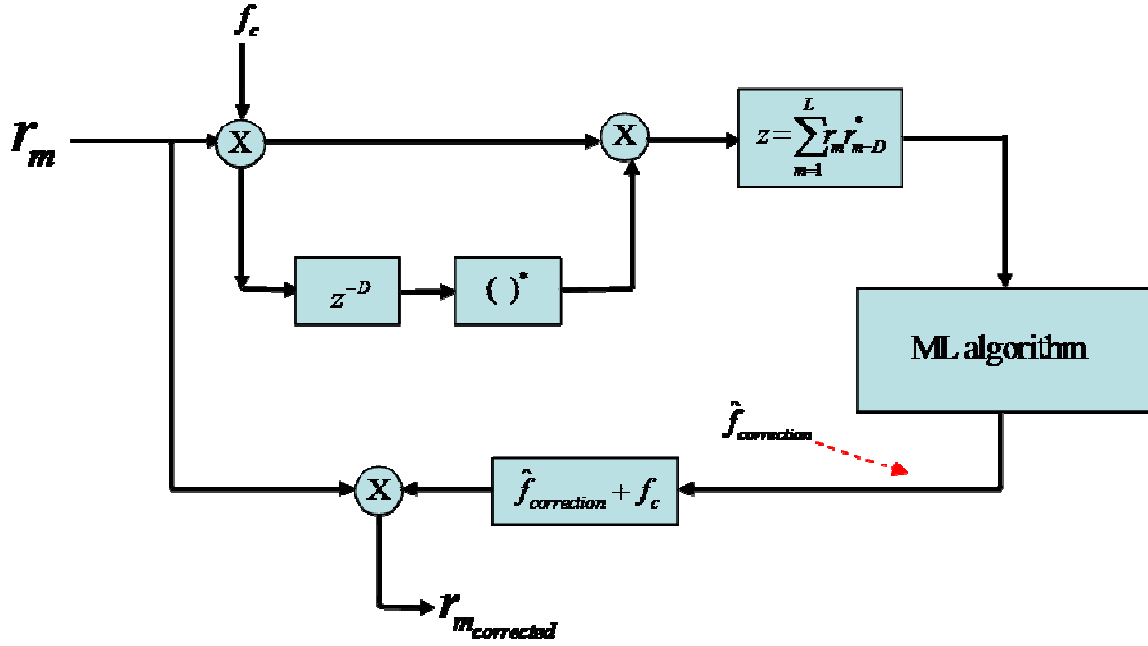


Figure 33 ML Frequency Synchronizer

The synchronizer depicted in Figure 33 will gradually improve or maintain the frequency offset estimate with each short training symbol received. Due to noise, a final residual frequency offset may remain but can be further mitigated using frequency tracking, or fine correction, by exploiting the long training symbols which follow later in the 802.11g preamble. In Chapter V, the performance of this synchronizer at low SNR and various frequency offsets will be examined.

3. Non-linear Least Squares Synchronization

Another approach to frequency synchronization involves a Non-linear Least Squares (NLS) approximation of the frequency offset. This method depends on the known signal structure of the preamble. In the case of 802.11g, the preamble consists of the 10 identical short training symbols used for packet detection and 2 identical long training symbols. [40]

Recall from Equation (4.60) that

$$r_m = s_m \exp(j2\pi\Delta f m t_s), \quad (4.66)$$

where r_m is the received sample at discrete time $t = m$, s_m is the equivalent low pass transmitted sample, and Δf is the frequency offset caused by a mismatch between the oscillators of the receiver and transmitter with all other causes of frequency offset ignored for simplicity. Because the 10 short training symbols are identical with 16 sample periodicity and if we assume flat fading conditions, the received short training samples may be written as

$$r_{m+nM_s} = s_{m+nM_s} \left[\exp(j2\pi t_s \Delta f n M_s) \right], \quad (4.67)$$

where n is the symbol number with the range $n = (0, \dots, N_s - 1)$, and m is now the sample index with range $m = (1, \dots, M_s)$. For the short training symbols, M_s is the duration of a training symbol in samples and N_s is the number of training symbols. Per the 802.11g standard, $M_s = 16$ and $N_s = 10$. Equation (4.67) can also be applied to the long training symbols, in which case, $N_L = 2$ and $M_L = 64$. [40]

The transmitted values of the short and long training samples are known a priori to the receiver and are assumed to be only affected by the frequency offset between transmitter and receiver. Therefore, Equation (4.67) can be re-written as

$$\begin{aligned} r_{m+nM_s} &= s_m \exp\{j2\pi\Delta f t_s [m + nM_s]\} \text{ if } 1 < m < M_s \text{ and } 0 < n < N_s \\ r_{m+nM_s} &= s_m \exp\{j2\pi\Delta f t_s [m + nM_L]\} \text{ if } 1 < m < M_s \text{ and } 0 < n < N_L \end{aligned} \quad (4.68)$$

for the short and long training samples respectively. From Equation (4.68), it can be seen that the frequency offset acts as a complex exponential multiplied by the first training symbol. If this can be considered as a multiplicative transfer function in the frequency domain, then in the time domain, this can be treated as an impulse response for the first training symbol, i.e. a convolution between the frequency offset and the first training symbol [10].

Time domain convolution can be easily expressed as matrix multiplication [8]. In this case, the matrix multiplication expression for Equation (4.68) becomes

$$\vec{r}_{M_s \times 1} = \vec{s}_{M_s \times M_s} \vec{c}_{M_s \times 1} \quad (4.69)$$

where $\vec{r}$ is the column vector of received training samples that constitute one symbol, $\vec{c}$ is the column vector comprised of complex exponentials that represent the frequency offset and $\vec{s}$ is the circular convolution matrix formed from the known training data. For the short training samples,

$$\vec{s} = \begin{bmatrix} s_1 & s_{16} & s_{15} & \cdots & s_2 \\ s_2 & s_1 & s_{16} & \cdots & s_3 \\ \vdots & \vdots & \ddots & & \vdots \\ s_{15} & s_{14} & s_{13} & \ddots & s_{16} \\ s_{16} & s_{15} & s_{14} & \cdots & s_1 \end{bmatrix} \quad (4.70)$$

A similar circular matrix can be constructed for the long training samples. [8].

From Equation (4.69), the column vector of complex exponentials caused by the frequency offset can be estimated as

$$\hat{\vec{c}} = \vec{s}^\dagger \vec{r} \quad (4.71)$$

where $\vec{s}^\dagger$ is the Moore-Penrose generalized inverse of $\vec{s}$ and represents the shortest least squares solution to Equation (4.69) [43]. The circular convolution matrix can be stored in the receiver's memory thereby reducing the quantity of computations done when a signal is detected. Follow-on, identical symbols can then be leveraged to refine the estimate of $\hat{\vec{c}}$. Equation (4.72) depicts such an algorithm.

$$\hat{\vec{c}}_n = \frac{1}{n} \vec{s}^\dagger (\vec{r}_1 + \dots + \vec{r}_n) \quad (4.72)$$

Equation (4.72)'s algorithm can be applied to both short and long training symbols, which then may be used to refine the estimate of the vector of complex exponentials. In this case, $n = (1, 2, \dots, N_s)$ or $n = (1, 2, \dots, N_L)$ for the short and long training symbols respectively. Once $\hat{\vec{c}}$ has been accurately estimated, the frequency offset can be

computed by taking the Fast Fourier Transform (FFT) of $\hat{\hat{c}}$ and then the magnitude of the frequency offset is found using Equation (4.62).

a. Implementation of the NLS Estimator

The NLS synchronizer described above was constructed in MATLAB. The algorithm exploits the periodicity of both the short and long training symbols. The frequency offset is -40 percent of sub-carrier spacing and SNR is 10 dB, which are the same conditions for the ML frequency synchronizer. Once again, “on-the-fly” frequency correction is implemented during the reception of the short and long training symbols to evaluate the performance of the algorithm. Figure 34 depicts the frequency offset correction value that is applied to subsequent incoming training symbols. As expected, due to noise in the channel, the offset is never completely eliminated, but the algorithm routinely computes corrections that result in a final frequency offset that is typically less than 1 percent of sub-carrier spacing using both the short and long training symbols.

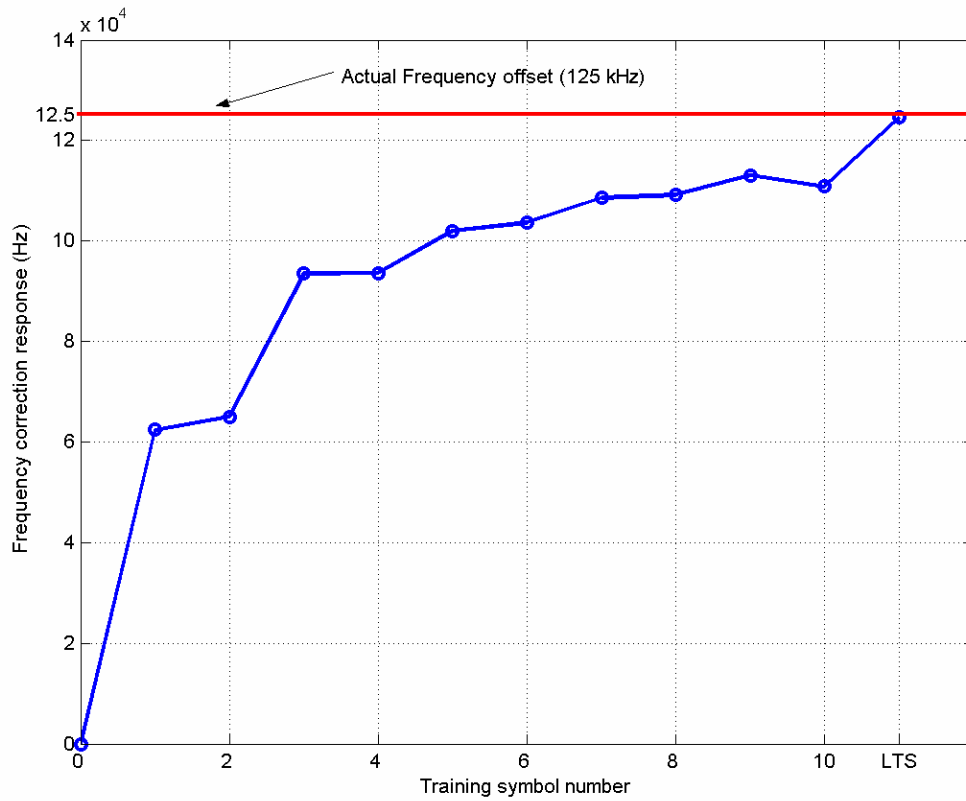


Figure 34 Frequency offset correction using NLS algorithm

In the Figure 34 above, “LTS” indicates the long training symbols per the 802.11g standard. These two long symbols, with the CP discarded, constitute the final correction to the frequency offset using the NLS algorithm. The CP is discarded to simulate what an actual 802.11g receiver may have to do when correcting for frequency offset with a real signal. The signal flow and structure of the NLS synchronizer is depicted in Figure 35. It depicts the implementation of the NLS algorithm coupled with the FFT of $\hat{c}$. It is then that the actual frequency offset is estimated using Equation (4.62) from section D2. This does not limit the algorithm to a maximum frequency offset of one-half of sub-carrier spacing apparently. Using the NLS algorithm, offsets of up to 95% sub-carrier spacing were detected and reasonably corrected. The NLS algorithm discussed not only depends on a priori knowledge of the training sample values but also

assumes perfect symbol timing. If symbol timing is unknown, or sample values are not known a priori, the NLS algorithm may require an iterative solution, where the receiver must make an initial guess of the frequency offset. The performance of the NLS algorithm will be explored more thoroughly in Section V under various SNR coupled with partial reception of the training symbols and imperfect symbol timing.

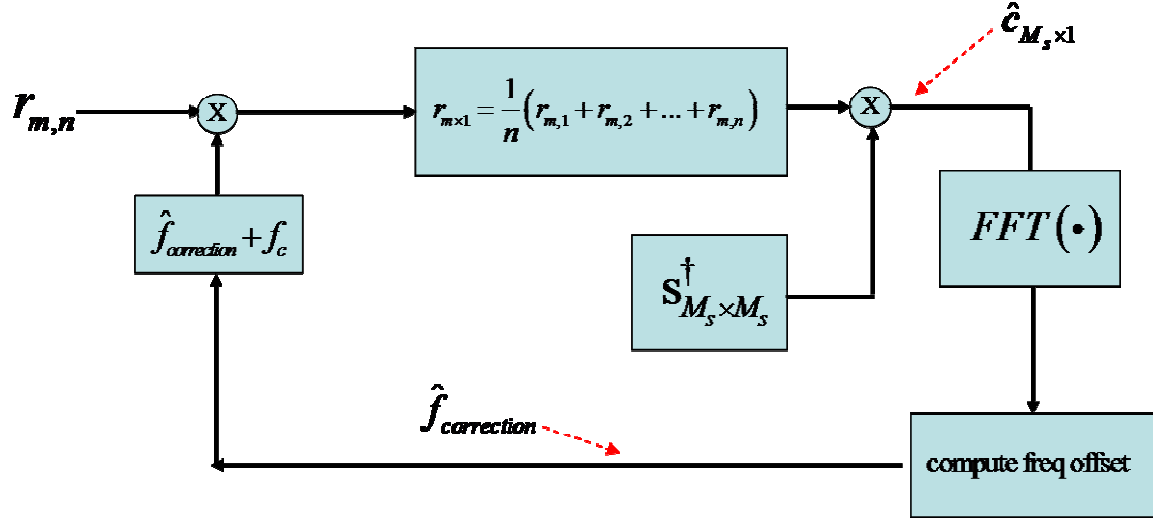


Figure 35 NLS Frequency Synchronizer

4. Summary of Frequency Synchronization Algorithms

Analysis in Chapter III established the necessity of frequency synchronization in OFDM systems. Many synchronization algorithms exist, but, as discussed in this section A3 of this chapter, data-aided synchronization seems to be the most applicable for OFDM WLANs because synchronization can occur rapidly with the widest acquisition range and best tracking [12]. The IEEE standard mandates the use of training symbols in the OFDM preamble anyway, discounting any argument against their use in terms of throughput. Two specific synchronization algorithms were examined in this sub-chapter – maximum likelihood (ML) and non-linear least squares (NLS). ML provided the advantage of a low complexity algorithm, with rapid correction to a predetermined frequency offset. The ML did suffer from a limited acquisition range, but the algorithm's

range, using either the long or short training symbols, is large enough to synchronize in frequency provided the oscillators used in the transmitter and receiver meet the IEEE standard. The NLS algorithm, seemed to provide less of a residual frequency offset than the ML algorithm with much greater acquisition range. This shall be verified in Chapter V. The NLS algorithm, however, was more complex and took a longer time to achieve frequency synchronization as a comparison between Figure 32 (ML) and Figure 34 (NLS) shows.

V. PERFORMANCE OF THE SYNCHRONIZATION ALGORITHMS

A. INTRODUCTION

This chapter focuses on evaluating the performance of the packet detection and frequency synchronization algorithms discussed in Chapter IV. Ultimately, the performance of any algorithm is dependent upon the nature of the channel and strength of the received signal [12]. The channel for all of the Monte Carlo simulations in this chapter will be an AWGN channel. The discussion in Chapter IV emphasized data-aided synchronization and the necessity of fast synchronization in a WLAN environment; hence the following performance evaluations will focus on the 802.11g 16 μ sec preamble.

B. PACKET DETECTION PERFORMANCE

The two packet detection algorithms analyzed in Chapter IV were called the Matched Filter detector and the Delay and Correlate detector. Both packet detection algorithms are evaluated in this Chapter by comparing their respective probability of detection against probability of false alarm while varying a single digit SNR (it is assumed that both algorithms perform adequately at double-digit SNR). Such a comparison, when depicted graphically is known as a receiver operating characteristic (ROC) curve [37]. To compute probability of detection, continuous short training symbols are transmitted in an AWGN channel. When a packet detection algorithm computes a decision statistic that is greater than the threshold value, a successful detection is judged to have occurred. To compute probability of false alarm, a noise-only signal is sent to the receiver, and once again, the decision statistic is computed and then compared to the appropriate threshold. When the decision statistic exceeds the threshold value, a false detection is judged to have occurred. The number of detections and false detections are tabulated and compared to the number of decision statistics to determine P_d and P_{fa} . The threshold range is determined by the mean value of the decision statistic which is either fixed or dependent upon the evaluated SNR values depending on the type

of detector being evaluated. For this performance evaluation, because any algorithm is expected to provide acceptable performance at medium to high levels of SNR, arbitrary values of SNR between -6 dB and +6 dB have been chosen. Once ROC data has been generated between the Delay and Correlate algorithm and the Matched Filter algorithm, they may be compared.

1. Delay and Correlate Packet Detection Performance

The threshold values used to determine P_d and P_{fa} for various low levels of SNR were based on the mean value of the decision statistic in the test signal. Equation (4.51) gave the mean value of the short training samples as a function of SNR. The threshold is then allowed to vary from $m_\chi - 0.5$ to $m_\chi + 0.5$, where m_χ is the mean value of the decision statistic when a signal is present. This threshold range is employed in order to detect the packet at low SNR. The use of multiple samples within the Delay and Correlate detector is used to keep P_{fa} at a relatively low value. The table below shows the evaluated SNR values, the mean value of the decision statistic as computed from Equation (4.51), and the resulting threshold range.

Table 3 Threshold range for evaluated SNR

SNR (dB)	Mean value of decision statistic (m_χ)	Threshold range (y_T)
-6	.0403	-.4597 $\rightarrow$.5403
-3	.1115	-.3885 $\rightarrow$.6115
0	.25	-.25 $\rightarrow$.75
3	.4437	-.0563 $\rightarrow$.9437
6	.6388	-.1388 $\rightarrow$ 1.1388

Once the threshold range has been determined for each evaluated SNR, the performance analysis is begun using the test signal discussed earlier. The test signal consists of 16,384 short training samples which, due to the 16 sample periodicity and

structure of the Delay and Correlate algorithm, resolves to 1023 decision statistics for each evaluated SNR. This is because, in the Delay and Correlate algorithm, the first symbol is compared to the second symbol; the second symbol is compared to the third, and so on. The test signal has a total of 1024 short training symbols, which resolves to 1023 decision statistics. The test signal is repeated to compute P_{fa} but in this case, the test signal consists only of noise. The resulting ROC curve is shown below for the various evaluated SNRs (shown as asterisk data points in Figure 36). Schmidl and Cox's [38] Gaussian approximation (GA) of the theoretical performance of the Delay and Correlate packet detector is depicted as solid lines. The curves are derived using the GA as it was discussed in Chapter 0 coupled with Equations (4.36), (4.43), (4.51), and (4.52). From Figure 36, the Gaussian approximation (GA) serves as a lower bound at higher tested SNRs while it closely approximates the computed values at low SNR. For clarity, the portion of the ROC curves for which $P_{fa} \leq .15$ is depicted.

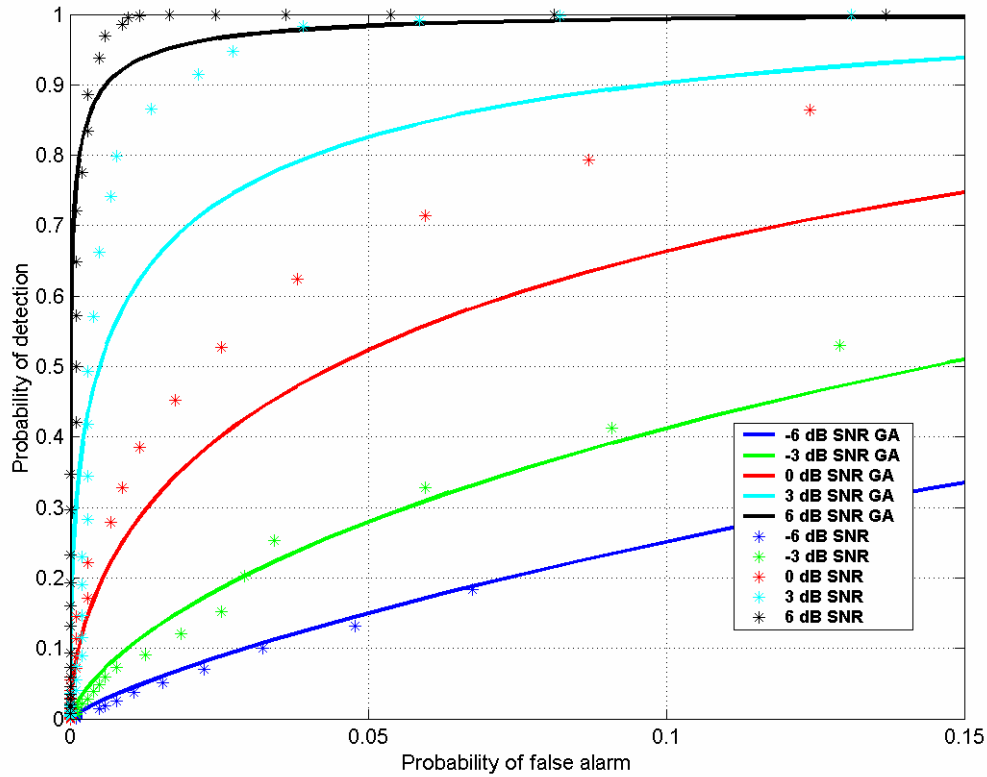


Figure 36 ROC curves for Delay and Correlate Detector

While the IEEE standard makes no mention of probability of false alarm requirements, it is certainly possible that designers would want to minimize the probability of false alarm because the receiver may be processing a false alarm and miss an actual packet. Furthermore, I speculate that this increased signal processing might result in a noticeable degradation in the time between subscriber unit battery recharges, particularly for small handheld devices. The IEEE standard does, however, mandate a maximum packet error rate of 10% for a PDSU length of 1000 bytes given the mandated receiver sensitivity for the various data rates [5]. If an arbitrary P_{fa} of .05 is acceptable, then Figure 36 shows that the Delay and Correlate detector, using 16 sample periodicity, in an AWGN channel, meets the IEEE packet error rate requirement when SNR is evaluated at 3 and 6 dB SNR. This assumes that the packet cannot be demodulated properly, and hence in error, if it is never detected. Therefore, for the purposes of this performance analysis, an acceptable baseline P_d is .9.

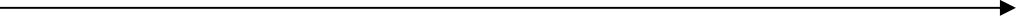
a. Improved Performance from the Delay and Correlate Detector

The performance of the Delay and Correlate detector can be further improved if additional samples are used to detect the packet. This should, as discussed in Chapter IV, allow for a low threshold while minimizing false alarms. The short training symbols, at the start of the 802.11g packet consist of 10 symbols, each 16 samples long. If the 802.11g receiver contained a buffer that could store more of the short training samples, then these additional samples could be used in the Delay and Correlate detector. The buffered sample values could also be used for other synchronization tasks as well. Such a detector, however, would cost more. Along with greater receiver complexity and additional signal processing, additional hardware may be required (e.g. a high gain antenna) to successfully demodulate the packet at very low SNR, thereby increasing unit cost. Such a specialized detector's performance, using the Delay and Correlate algorithm, is well illustrated in the table below. The Delay and Correlate algorithm and the simulation parameters remain the same as in the previous performance analysis, except that the size of the delay block has been changed to 80 samples from 16 samples.

Although there are 160 samples in the short training symbols, because the first half of the preamble is compared to the second half, the algorithm is limited to 80 repetitive samples.

Table 4: P_d and P_{fa} for 80 sample Delay and Correlate Detector

SNR (dB)	P_d / P_{fa}		P_d / P_{fa}		P_d / P_{fa}		P_d / P_{fa}		P_d / P_{fa}		P_d / P_{fa}		P_d / P_{fa}	
-6	0.98	0.77	0.96	0.65	0.95	0.52	0.95	0.47	0.92	0.39	0.91	0.35	0.91	0.30
-3	1.00	0.40	1.00	0.25	0.99	0.17	0.99	0.13	0.99	0.06	0.98	0.05	0.96	0.02
0	1.00	0.15	1.00	0.06	1.00	0.02	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
3	1.00	0.03	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
6	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

y_T 

In Table 4, P_d and P_{fa} are shown as a function of SNR and threshold. For this performance analysis each run is conducted at a tested SNR, which is adjusted by increasing the noise variance. The threshold, y_T , is gradually increased for the SNR values listed in the left most column of the table. As discussed in Chapter 0, the threshold can be varied to achieve a desired P_d . y_T is set at an initial value of $m_x/10$ and allowed to gradually increase to a value of $m_x/2$. In order to present the table data clearly, the probabilities have been rounded to 2-decimal accuracy. Table values that have a red background indicated a failure to meet the previously discussed standard of $P_d = .9$. Cells with a yellow background show where $P_d \geq .9$ but $P_{fa} > .05$. A blue background indicates the desired criteria of $P_d \geq .9$ and $P_{fa} \leq .05$ has been met. From Table 4, the performance of the 80 sample Delay and Correlate detector clearly exceeds the performance of the same algorithm using only 16 samples. As discussed earlier, the 16 sample detector met the desired criteria for +3 dB and +6 dB SNR. The 80 sample detector also met that criteria and meets the criteria at 0 dB and -3 dB for higher threshold levels. Additionally at 0 dB, the 80 sample detector missed the arbitrarily set P_{fa} by only a small amount at low thresholds.

2. Matched Filter Packet Detection Performance

The other packet detection algorithm discussed in Chapter IV was the Matched Filter detector. Because the values of the short training samples are known a priori to the receiver, this could be used to facilitate Matched Filter detection. For the performance analysis, the same test signal used in section B1 was also used to evaluate the Matched Filter detector's performance. The detector, which has the same signal structure as depicted in Figure 28, generates 16368 decision statistics from the test signal. The value of the decision statistic, χ_M , is dependent upon the tested SNR but closely matches the bottom half of Figure 29. As Figure 29 shows, the values of χ_M "spike" upwards when there is a perfect correlation between an incoming short training symbol and the Matched Filter coefficients. The "spikes" occur every 16 samples (the other values of χ_M represent partial or no correlation with the filter coefficients). These particular values of χ_M are the focus of the performance analysis for the Matched Filter detector. Hence, there are actually 1023 values of χ_M that can be used for detecting a short training symbol in noise. The performance analysis was run using the same tested values of SNR that were used in the evaluation of the Delay and Correlate detector. For the analysis, the threshold was allowed to vary in accordance with the mean value of the decision statistic for each tested SNR. For the 16 sample Matched Filter detector, the ROC data is best illustrated in the table below. To analyze P_{fa} , the test signal consists of only noise.

Table 5: P_d and P_{fa} for Matched Filter Detector

SNR (dB)	P_d / P_{fa}		P_d / P_{fa}		P_d / P_{fa}		P_d / P_{fa}		P_d / P_{fa}		P_d / P_{fa}		P_d / P_{fa}		P_d / P_{fa}		P_d / P_{fa}		P_d / P_{fa}	
-6	0.99	0.37	0.99	0.31	0.98	0.25	0.97	0.20	0.96	0.15	0.95	0.12	0.92	0.09	0.89	0.06	0.85	0.05		
-3	1.00	0.32	1.00	0.25	1.00	0.17	1.00	0.11	0.99	0.07	0.98	0.04	0.96	0.02	0.95	0.01	0.93	0.00		
0	1.00	0.24	1.00	0.15	1.00	0.08	1.00	0.04	1.00	0.02	1.00	0.00	1.00	0.00	0.99	0.00	0.98	0.00		
3	1.00	0.16	1.00	0.07	1.00	0.02	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00		
6	1.00	0.09	1.00	0.02	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00		

y_T —————→

In Table 5, the left most columns represent low threshold and the right most columns represent high threshold. For the analysis, $y_T = m_x/8$ to $m_x/1.5$. The performance of the 16 sample Matched Filter detector shows good performance at higher tested SNRs but fails to meet the arbitrary standard of $P_{fa} \leq .05$ at low SNR for the various thresholds. The Matched Filter Detector also fails to meet the arbitrary standard of $P_d \geq .9$ and $P_{fa} \leq .05$ only at the lowest tested SNR and the highest values of y_T . The performance of this detector, like the Delay and Correlate detector can be improved by using more samples. The drawback will be greater complexity of the detector.

a. *Improved Performance from the Matched Filter Detector*

An improved Matched Filter detector can be built by using more samples. The improved detector would require an additional buffer or delay blocks to store the samples so they can be compared against an increased number of filter coefficients. A modified structure is shown in Figure 37. This structure is specific for exploiting the repetitive short training symbols. The blocks annotated as z^{-D} represent single sample delay blocks while the block annotated as z^{-M} represents a 16 sample delay block. This detector represents a significant increase in complexity relative to the Matched Filter detector analyzed previously. For comparison, the original detector's block diagram is depicted in Figure 28.

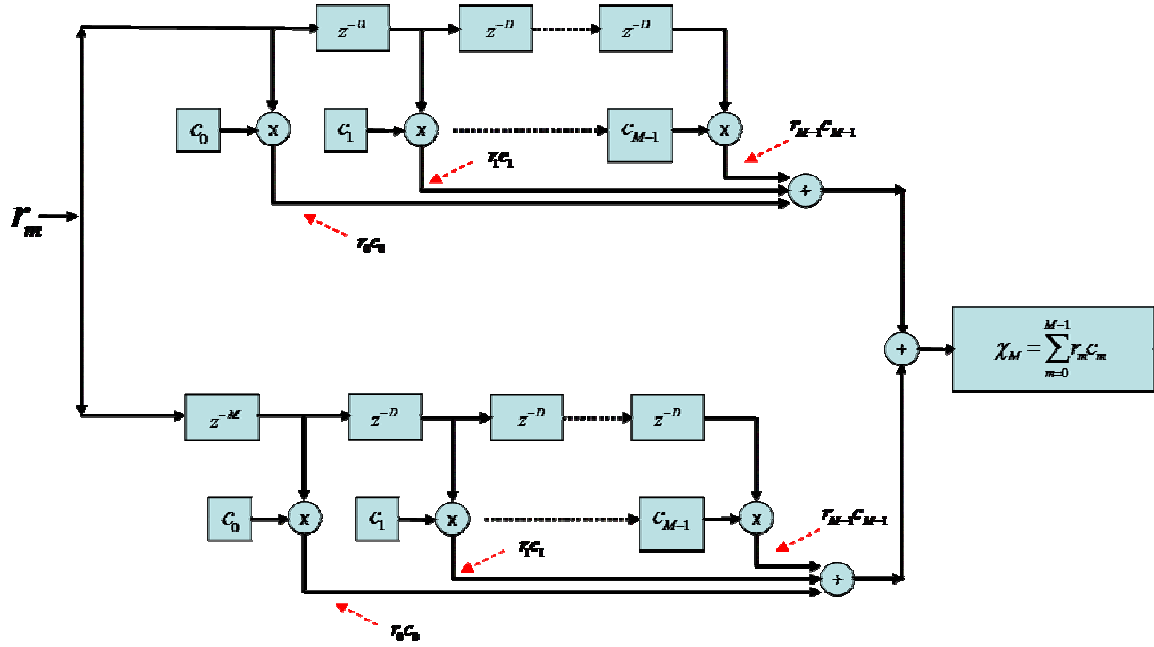


Figure 37 32 sample Matched Filter Detector

The resulting performance analysis of the new or “double” Matched Filter detector shows that the use of more samples allows for a higher threshold, thereby reducing the false alarm rate. For the performance analysis, the new mean of the decision statistic becomes twice the old mean, or $m_\chi = .4$. The threshold, y_T , is allowed to vary from $m_\chi/8$ to $m_\chi/1.5$. The results of the performance analysis are shown in Table 6.

Table 6: P_d and P_{fa} for Improved Matched Filter Detector

SNR (dB)	P_d / P_{fa}	P_d / P_{fa}	P_d / P_{fa}	P_d / P_{fa}	P_d / P_{fa}	P_d / P_{fa}	P_d / P_{fa}	P_d / P_{fa}	P_d / P_{fa}
-6	1.00 0.28	1.00 0.21	1.00 0.15	1.00 0.09	0.99 0.06	0.99 0.03	0.97 0.02	0.95 0.01	0.93 0.00
-3	1.00 0.23	1.00 0.14	1.00 0.08	1.00 0.03	1.00 0.01	1.00 0.00	1.00 0.00	0.99 0.00	0.99 0.00
0	1.00 0.14	1.00 0.06	1.00 0.02	1.00 0.01	1.00 0.00	1.00 0.00	1.00 0.00	1.00 0.00	1.00 0.00
3	1.00 0.08	1.00 0.02	1.00 0.00	1.00 0.00	1.00 0.00	1.00 0.00	1.00 0.00	1.00 0.00	1.00 0.00
6	1.00 0.02	1.00 0.00	1.00 0.00	1.00 0.00	1.00 0.00	1.00 0.00	1.00 0.00	1.00 0.00	1.00 0.00

y_T
→

3. Summary of Packet Detection Performance

Sub-Sections B1 and B2 discussed the performance analysis of the Delay and Correlate detector and the Matched Filter detector respectively. It is shown that the Delay and Correlate Detector had difficulty meeting the arbitrary standard of $P_d \geq .9$ and $P_{fa} \leq .05$ at low SNR. Although some improvement was experienced by using additional samples (with the drawback of likely greater cost and complexity), performance still lagged at very low SNR. The “improved” Delay and Correlate detector was dependent upon receiving all ten short training samples because it incorporated a delay buffer equivalent to 80 samples. Although it is beyond the scope of this thesis, the performance of the 80 sample detector could be evaluated with only partial receipt of the ten short training samples. It is expected that the performance of the detector would further deteriorate.

The Matched Filter detector’s performance is initially analyzed using 16 filter coefficients due to the 16 sample periodicity of the short training symbols. The performance of the detector is comparable to the Delay and Correlate Detector using 80 samples. This is graphically evident in the similar color schemes of Table 4 and Table 5. To determine if the desired values of P_d and P_{fa} could be obtained, additional samples were used to create more filter coefficients and then placed in series with the single Matched Filter. In this case, the receipt of at least two consecutive short training symbols would be required for detection. The resulting Double Matched Filter detector, as depicted in Figure 37, would be more complex and likely more costly than a single Matched Filter detector, but met the performance requirements across the tested range of SNR for middle to high threshold values.

In conclusion, the Double Matched Filter detector provided the best performance at the tested SNR range but meeting the performance requirements meant using additional filter coefficients, leading to greater detector complexity. This result is not completely surprising. The Matched Filter is the best filter for maximizing SNR [10]. Intuitively, it makes sense that the same filter would provide the best performance at a given SNR. For a commercial vendor, such a detector may not be economically viable

due to the requirement to minimize subscriber unit production costs. On the other hand, such high performance detectors may find specialized uses, particularly when coupled with hardware solutions, such as high gain antennas.

C. FREQUENCY SYNCHRONIZATION PERFORMANCE

In Chapter IV, the Maximum Likelihood (ML) and Non-linear Least Squares (NLS) frequency synchronization techniques were introduced. This section focuses on the performance analysis and comparison of those techniques. As discussed previously, the necessity to detect and correct any frequency offset error suggests leveraging the short and long training symbols that are placed at the beginning of the 802.11g preamble in accordance with the IEEE standard. For the performance evaluation, the test signal consists of the ten short training symbols sent to the receiver in an AWGN channel. As in the detection analysis, multi-path effects are ignored for the purpose of this thesis. The test signal is sent multiple times to the receiver. In each case, the tested algorithm must correct the frequency offset. Although in practice, there will always be some residual frequency offset due to phase noise, it is not considered in this evaluation. Hence, the final frequency offset should be zero in theory after the application of the frequency correction algorithm. The results of each algorithm's multiple frequency correction attempts are displayed as histograms in the following sub-sections.

1. Maximum Likelihood Frequency Synchronization Performance

The performance analysis of the ML algorithm consists of the test signal, depicted in Figure 31, corrupted by noise and a significant frequency offset (40 percent of sub-carrier spacing). In each case, the ML is able to detect and correct the frequency offset, using only the short training symbols. For the initial analysis, all ten short training symbols are used to refine the frequency estimate. Additionally, the ML algorithm is examined for robustness by using only 4 short training symbols and 2 long training symbols. These results will be compared to the performance data when the algorithm uses all ten symbols. The expectation is that the ML algorithm will be relatively immune to the reduction in the number of short training symbols available for synchronization. This is suggested from the implementation results displayed in Figure 32. If less training

symbols are required for frequency synchronization, then the initial training symbols may be discarded after being used for packet detection, perhaps reducing complexity and cost of the receiver. The results, for 500 trials at 10 dB SNR and 6 dB SNR, are shown in the histograms of Figure 38.

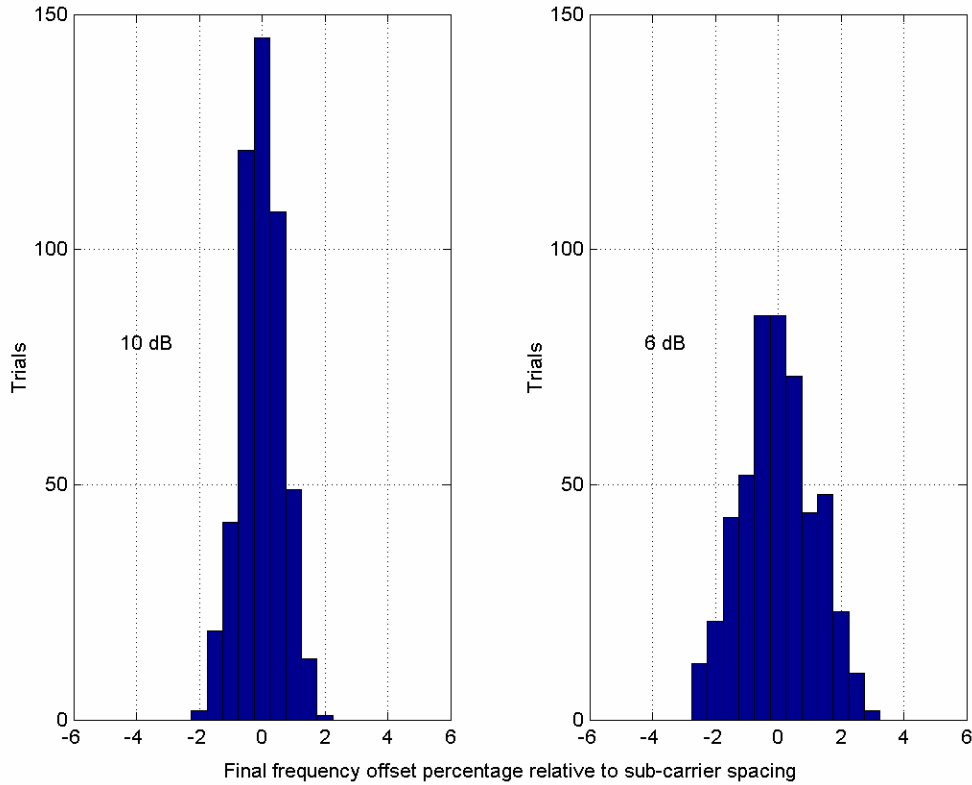


Figure 38 Remaining Frequency offset using ML algorithm (10 and 6 dB SNR) using 10 short training symbols

The performance analysis results depicted in the histograms suggest that the ML algorithm is an accurate technique to synchronize in frequency. The average final frequency offset, after 10 short training symbols, was less than 1 percent frequency offset at 10 dB SNR. For all 500 trials, the final frequency offset never exceeds two percent of sub-carrier spacing for the histogram on the left despite an initial offset of 40 percent of sub-carrier spacing (or about 125 kHz). As noted previously, the maximum allowable frequency offset for the ML algorithm is 50 percent relative to sub-carrier spacing. As

expected, the performance of the ML algorithm degrades as SNR is reduced. This is evident from the greater variance in the final frequency offset percentage visibly present in the histogram on the right side of Figure 38. Additionally, there are far fewer trials that resulted in a zero or near zero offset condition.

Furthermore, as SNR is reduced, performance continues to degrade. The degradation is graphically illustrated by the histogram depicting final frequency offset at 3 dB SNR in Figure 39. As shown in Figure 39, the ML algorithm may also employ the long training symbols in addition to the short training symbols for frequency synchronization. The purpose of showing the use of the long training symbols in frequency synchronization on the left hand side of Figure 39 is to highlight the performance advantage inherent in the use of the long training symbols to “fine” tune the frequency offset as recommended by the IEEE standard [4]. This is clearly evident when the two histograms in Figure 39 are compared. The histogram on the left hand side of Figure 39 depicts ML algorithm performance without the long training symbols. Another reason is because the next section will examine the “robustness” of the ML algorithm. The robustness of the algorithm is analyzed by limiting the number of training symbols available to the synchronization algorithm. This performance analysis involves simulating the reception of only a few of the short training symbols and the long training symbols vice the full complement.

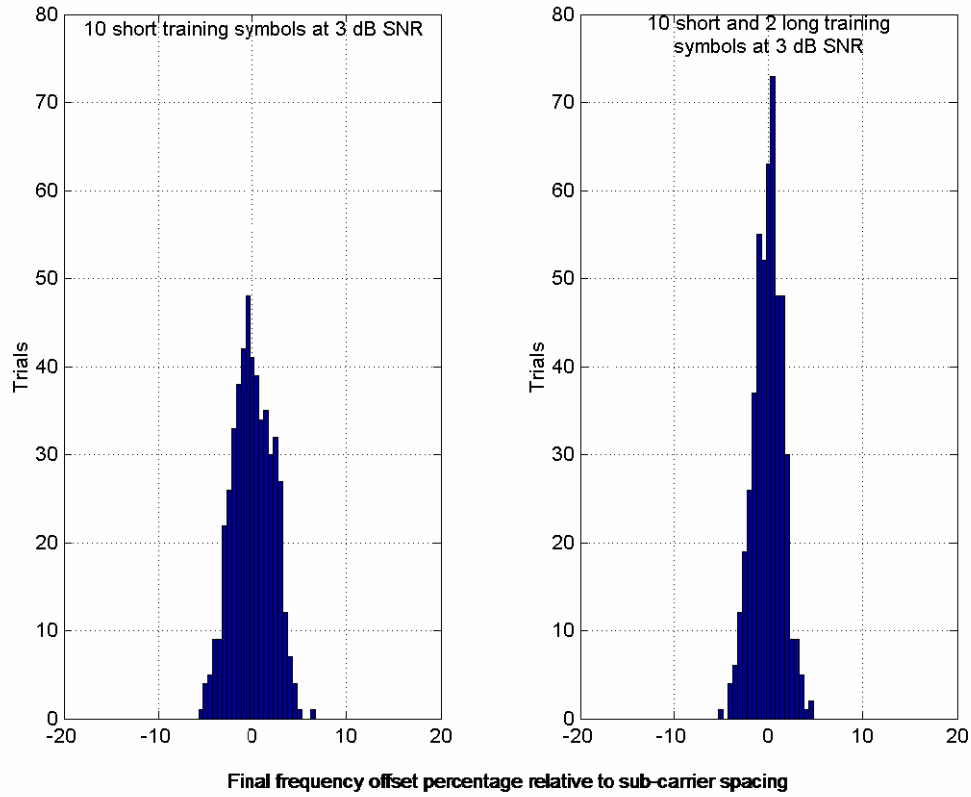


Figure 39 Remaining frequency offset percentage using ML algorithm at 3 dB SNR

a. Robustness of the ML Frequency Correction Algorithm

To test the robustness of the ML algorithm, the analysis was re-run using only four short training symbols and then again with four short training symbols and the two long training symbols to highlight the critical role the long training symbols play in synchronization. In each case, the new test signal was re-sent 500 times to the receiver at 10 dB SNR. The results, displayed in Figure 40, can be contrasted with the left hand side of Figure 38. While both Figures are histograms that depict the performance of the ML algorithm at 10 dB SNR, the results in Figure 38 are based on the reception of 10 short training symbols and the two long training symbols. The results in Figure 40 show only a slight degradation in performance. This robustness of the ML algorithm is not surprising

because of the ability of the ML algorithm to quickly detect and correct the frequency offset as suggested from the implementation results depicted in Figure 32.

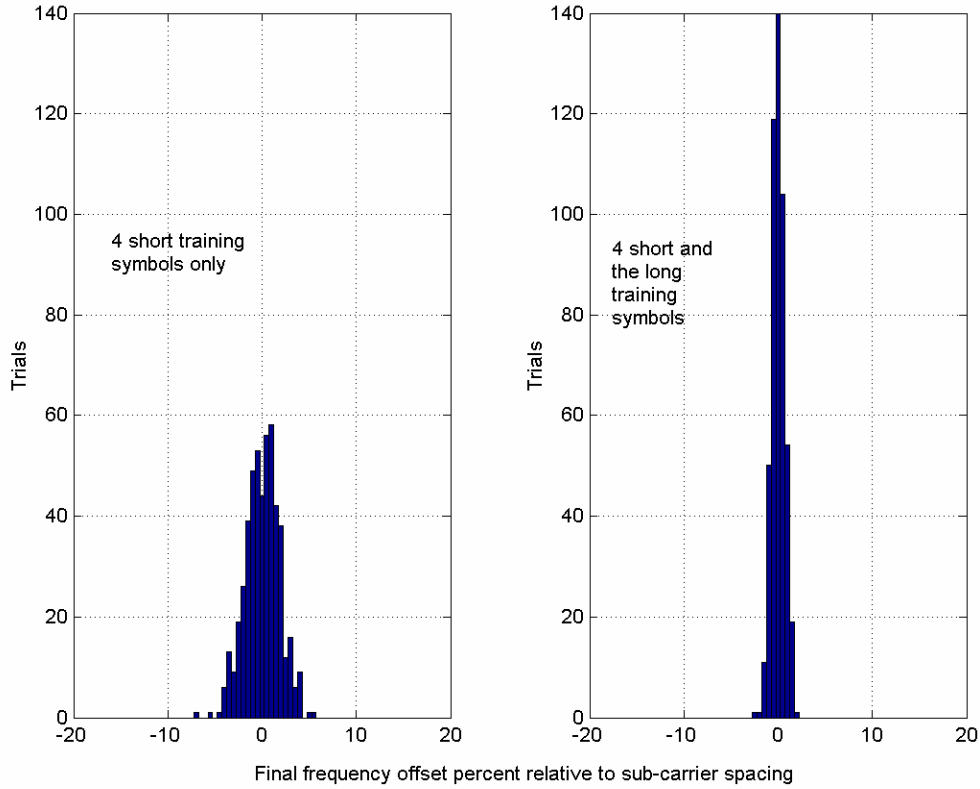


Figure 40 Remaining Frequency offset percentage using ML algorithm and reduced training symbols

2. Non-linear Least Squares Frequency Synchronization Performance

The second frequency technique that was discussed in Section IV was Non-linear Least Squares (NLS). Although this technique could be used in a non data-aided synchronization scheme as discussed in Chapter IV, the necessity in the WLAN environment to synchronize in frequency required focusing on a data-aided scheme, using both the short and long training symbols. This will also allow for a straight forward comparative analysis with the ML technique. The implementation results depicted in Figure 34 show that the NLS technique could synchronize in frequency, but the long training symbols are needed to approach the same performance standard as that of the

ML algorithm. I judge this to be acceptable because the IEEE standard specifically recommends using the two long training symbols for fine frequency correction [4]. The performance of the NLS algorithm is evaluated using the same conditions that governed the analysis of the ML technique – large initial frequency offset, low SNR, multi-path effects disregarded, and eventually examining the robustness of the NLS algorithm by using only 4 short training symbols. The initial frequency offset is -40% of sub-carrier spacing (or 125 kHz less than the sub-carrier's center frequency). In the last condition (robustness), it is expected that the NLS algorithm will lag in performance relative to the ML algorithm. To graphically depict the performance of the NLS technique, histograms are produced (as in the ML algorithm analysis) by using a test signal re-sent to the receiver 500 times. In this case, the test signal consists of the 10 short training symbols, the cyclic prefix of the long training symbols (which is discarded by the receiver) and the two long training symbols themselves. The first of these histograms, in Figure 41, shows the remaining frequency offset percentage relative to sub-carrier spacing at 10 dB SNR (left hand side) and 6 dB SNR (right hand side).

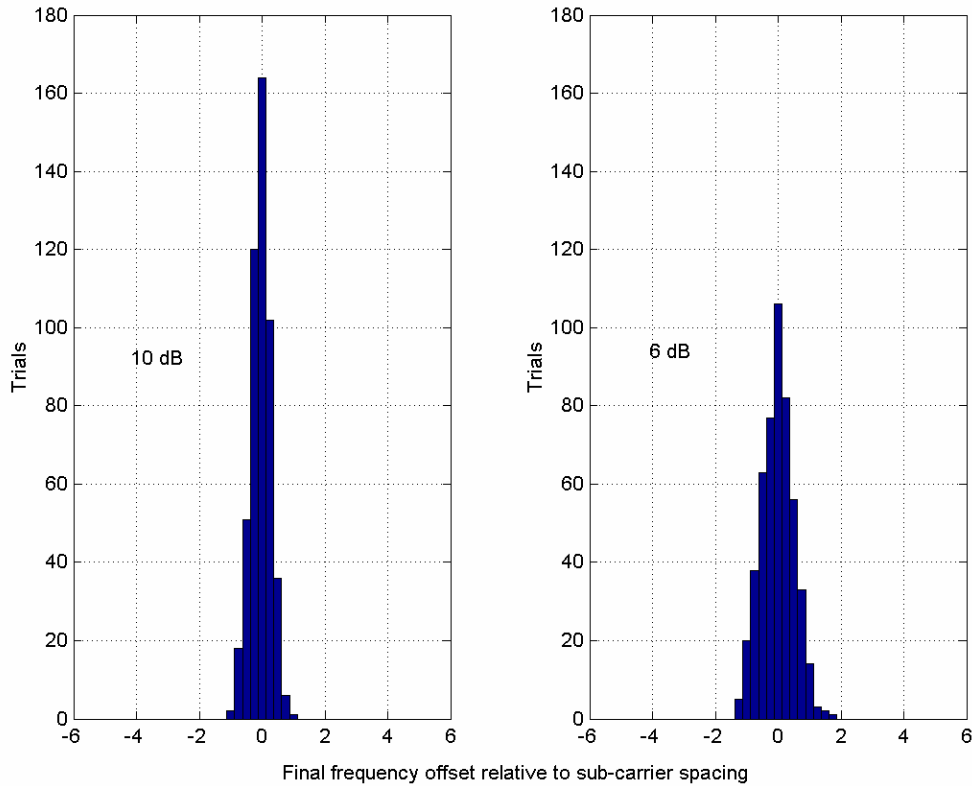


Figure 41 Remaining Frequency offset percentage using NLS algorithm using 10 short and 2 long training symbols

From Figure 41, it is evident that the NLS algorithm effectively performs the frequency synchronization task. All of the trials resulted in a frequency offset of less than two percent for both 10 dB and 6 dB SNR. As expected, as SNR is reduced, the performance of the NLS is degraded. Results for 6 dB SNR are depicted as the right histogram in Figure 41. The histogram shows a similar performance degradation that was apparent for the ML algorithm – greater variance in the final frequency offset percentage (relative to sub-carrier spacing) and less trials that results in zero or near zero remaining frequency offset. Similar decreasing performance can be observed when the SNR is reduced to 3 dB. That histogram will be highlighted in the final section of this chapter when a direct comparison with the ML algorithm is made.

a. Robustness of the NLS Frequency Correction Algorithm

The performance results shown in the NLS histograms suggest similar performance relative to the ML algorithm. The NLS histograms presented so far depict performance of the algorithm when all ten short training symbols and two long training symbols are used for frequency synchronization. To examine the robustness of the algorithm, additional analysis is performed under the same conditions as the ML algorithm. Once again, the test signal consists of four short training symbols and the two long training symbols. To further test the robustness of the algorithm, a second test signal consisting only 4 short training symbols is used. The results for 500 trials are shown in Figure 42.

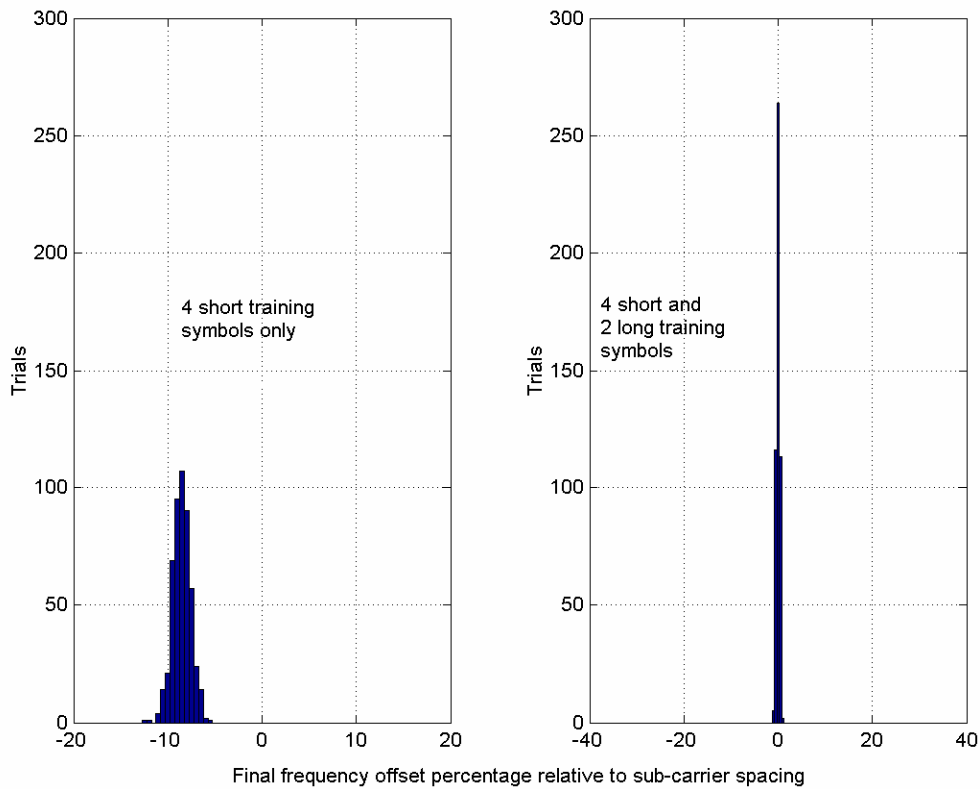


Figure 42 Remaining Frequency offset percentage using NLS algorithm and reduced training symbols

The histograms in Figure 42 reinforce the power of the long training symbols in frequency synchronization. This is especially true for the NLS algorithm, as a 64×64 matrix is used to for “fine” frequency adjustment. In this case, fine synchronization using the long training symbols quickly eliminates any residual frequency offset. The Figure 42 histograms can also be viewed, conversely, as showing that the NLS algorithm is highly dependent upon the long training symbols for frequency synchronization. Unlike the ML algorithm, the NLS algorithm slowly synchronizes in frequency with each received short training symbol. This result is not unexpected and was evident in the implementation results shown in Figure 34. Hence, the NLS algorithm is robust only if the long training symbols are received and used for frequency synchronization. The algorithm will not sufficiently synchronize the receiver using only 4 short training symbols.

3. Summary of Frequency Synchronization Performance

Two types of frequency synchronization algorithms were evaluated at low SNR – the Maximum Likelihood algorithm and the Non-linear Least Squares algorithm. Histograms were produced from multiple test signals sent to receivers using the two algorithms. Both algorithms performed effectively. The robustness of each algorithm was also examined. Here, the performance of the algorithms contrasted sharply. The ML algorithm was able to quickly eliminate most of the frequency offset using only the short training symbols (only 4 symbols in the tested case). The long training symbols were an important facet of the synchronization scheme in the ML algorithm but the algorithm could correct a frequency offset without them if required. The NLS algorithm was highly dependent upon the use of the long training symbols to eliminate the frequency offset. Without the long training symbols, the offset could not be eliminated with only 4 short training symbols. As discussed previously, this was because the NLS algorithm only slowly corrects the frequency offset as opposed to the ML algorithm, which quickly eliminates the offset. **Although the NLS scheme was less robust, due to the dependency of the long training symbols, the performance of the NLS algorithm was nevertheless superior to the ML algorithm.** This is evident in a final comparison of

both synchronization algorithms at very low SNR (3 dB), using only 4 short training symbols and the long training symbols as shown in Figure 43.

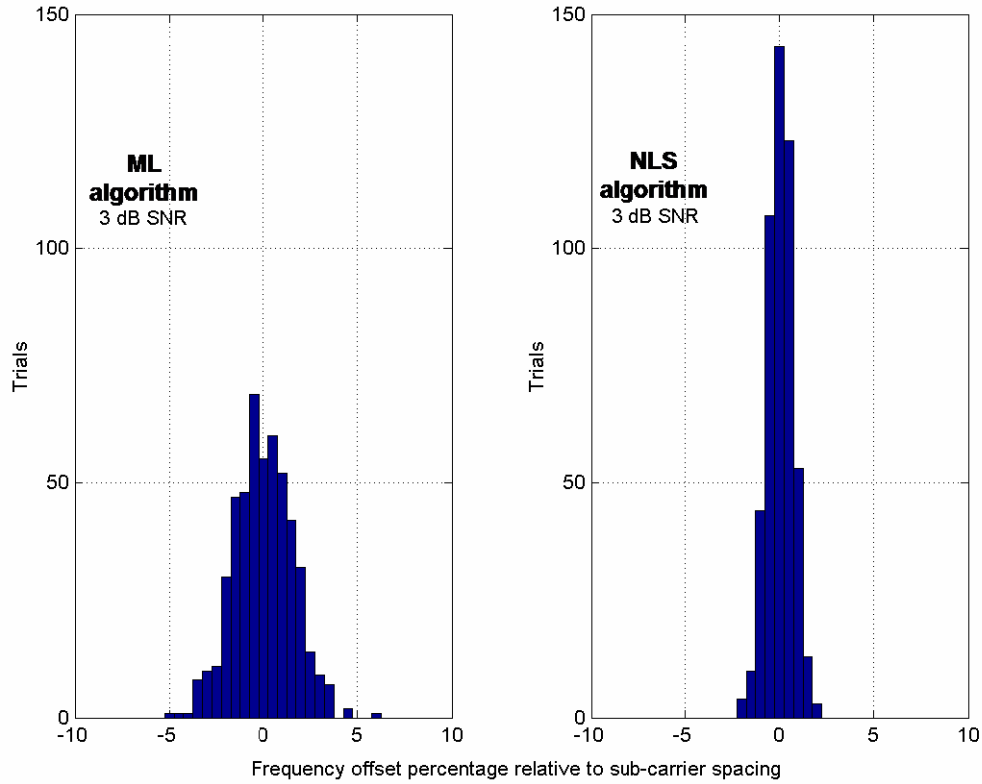


Figure 43 Comparison of remaining Frequency offset percentage at 3 dB SNR with 4 short and 2 long training symbols

In summary, this chapter analyzed the synchronization algorithms discussed in the Chapter IV. The various algorithms for packet detection and frequency synchronization were comparatively analyzed. The following conclusion chapter will summarize the results of this performance analysis.

THIS PAGE LEFT INTENTIONALLY BLANK

VI. CONCLUSION

This thesis addresses promising synchronization algorithms that may be used to extend the range of 802.11g WLANs. The scope of the thesis centers on the analysis of packet detection (frame synchronization) and frequency synchronization algorithms. Furthermore, to focus on the implementation of those algorithms, only performance in an AWGN channel is considered.

Frame synchronization algorithms that seem to hold promise are the Delay and Correlate packet detector and the Matched Filter Detector. Promising frequency synchronization algorithms highlighted in the thesis were the Maximum Likelihood algorithm and the Non-Linear Least Squares algorithm. The implementation of all of these algorithms were successfully completed in MATLAB and discussed in Chapter IV. The MATLAB code is included as Appendix A. The performance analysis of these algorithms, under test conditions of low SNR, is covered in detail in Chapter V. The results are discussed in the following section.

A. RESULTS OF THE PERFORMANCE ANALYSIS

The performance analysis of Chapter V focuses on two specific packet detection synchronization algorithms using data-aided synchronization at low SNR. Specifically, the packet detection algorithms are evaluated in Chapter V with an SNR range of +6 dB to -6 dB, using the short training symbols at the beginning of the 802.11g preamble. The long training symbols are not used, recognizing their necessity for the follow-on task of frequency synchronization. While Receiver Operating Characteristic curves of P_d vs P_{fa} are generated, the results for both algorithms are best displayed in color-coded table format. The results for the 80-sample Delay and Correlate detector, displayed in Table 4, showed that the detector could not meet the arbitrary standard of $P_d \geq .9$ and $P_{fa} \leq .05$ at lowest tested SNR (-6 dB) regardless of the threshold setting. Of note, the Delay and Correlate detector suffered from a very high probability of false alarm at the lowest tested SNR. This was the most robust of the Delay and Correlate detectors implemented in this thesis. The 80 sample Delay and Correlate detector uses

the most samples available, assuming full reception of all ten short training symbols. Although beyond the scope of this thesis, it is likely that fewer received symbols would result in a degradation of performance for this detector.

The Matched Filter detector's performance was similar to that of the Delay and Correlate detector across the tested SNR range. There was one difference: P_{fa} was consistently lower than that of the 80-sample Delay and Correlate detector at the lowest tested SNR. Despite that, the Matched Filter's P_{fa} was still above the arbitrary standard at the lowest tested SNR. As discussed in Chapter IV, the use of more samples within a detector algorithm would allow for a lower threshold while maintaining the same false alarm rate or would lower the false alarm rate for a given threshold. This led to an implementation and evaluation of a 32 sample (or double) Matched Filter detector. The results, shown in Table 6, highlight the superior performance of this detector throughout the range of tested SNR values. The 32 sample Matched Filter detector outshines the maximum sample (80) Delay and Correlate detector. Furthermore, its superior performance is enhanced because it requires the reception of only two vice 10 short training symbols, which the 80-sample Delay and Correlate detector is dependent upon.

The performance results of the frequency synchronization algorithms are best displayed as histograms. Once again, the emphasis is on data-aided synchronization. In this case, both short and long training symbols are employed. Each frequency synchronization algorithm was initially tested using 10 short training symbols and the two long training symbols. Although the effect of multi-path was beyond the scope of this thesis, the cyclic prefix of the first long training symbol was discarded by the receiver to more realistically simulate the performance of these algorithms in a WLAN environment. After initial testing, the algorithms were tested for "robustness" – in which the number of available short training symbols was reduced from 10 to 4. This was done to simulate that short training symbols used for packet detection are not buffered by the receiver and therefore not available for frequency synchronization. Although it is

certainly technically possible for an OFDM receiver to buffer the short training symbols, the receiver may not buffer the samples because, as discussed in Chapter III, it may not be economically viable to do so.

The histograms in Chapter V show that both the Maximum Likelihood and Non-linear Least Squares algorithms adequately correct a large frequency offset at 10 dB SNR. The histograms for both algorithms show very similar performance, with both algorithms suffering performance degradation as SNR is reduced to the minimum tested SNR (3 dB). The most telling results are contained in Figure 43. The histograms depicted there provide a direct comparison between the ML and NLS frequency synchronization algorithms at the lowest tested SNR coupled with a reduced number of training symbols available to correct a frequency offset. The NLS algorithm offers superior performance relative to the ML algorithm. The disadvantage of the NLS algorithm is its complexity. Use of the NLS algorithm requires that the receiver store a 16×16 circular convolution matrix for short training symbols. For the long training symbols, the receiver must store a 64×64 matrix! While the NLS algorithm's performance was superior, such complexity may not be economically viable for a commercial-off-the-shelf (COTS) receiver.

B. RECOMMENDATIONS FOR FOLLOW ON RESEARCH

The use of OFDM as a viable modulation technique to achieve high data rates, while mitigating ISI, suggests its continued proliferation within WLANs. This thesis highlighted several promising synchronization techniques that successfully detected and frequency synchronized the 802.11g packet using the IEEE mandated training symbols at low SNR. No direct comparison was made to any existing commercial synchronization techniques however. The software analysis of the promising techniques discussed in Chapter V naturally suggests a follow on comparison between those techniques and the synchronization algorithms currently in use in commercial products to verify the improved performance and to accurately assess tradeoffs between performance and cost. Due to the proprietary nature of commercial synchronization algorithms, this could be done using COTS equipment.

Also, because multiple synchronization tasks are required to successfully demodulate OFDM, a wide variety of synchronization algorithms have been developed. This wide variety of algorithms and tasks cannot be adequately addressed within the limited scope of a single thesis. This offers three additional areas for follow on research.

Although phase noise correction and symbol timing were reviewed in Chapter III, analysis of synchronization algorithms that address those issues offer opportunity for follow-on research. Next, because the IEEE standard does not specify the types of synchronization algorithms to be used for the various synchronization tasks, new algorithms are being developed that require comparative analysis. Finally, the performance analysis of the evaluated synchronization algorithms was conducted in a simulated AWGN channel. While incorporating multi-path into the performance analysis would be technically more difficult, a more realistic simulation could be performed because WLANs often operate in a multi-path environment.

APPENDIX

The following MATLAB code implements and provides output for the various detectors and synchronizers used throughout this thesis. Use of ‘functions’ is employed, allowing the recycling of usable code.

delay_correlate_detector.m – This m-file implements the Delay and Correlate detector that was extensively analyzed in this thesis. The output of this file is a graph of the detector’s decision statistic along with the ‘C’ and ‘P’ windows.

```
clear all
% this file implements the Delay and Correlate detector. It uses the functions st_sym and
% short_trng_samples.
% author: Vik Sardana, MSEE
% 22 DEC 2006
% output of this file is a graph of the decision statistic, P and C windows.

% get short training symbols
rn1=st_sym;
rn2=ifft(rn1);
%
% add noise to the delay buffer (or else divide by zero) and training samples
noise_variance=6.3477e-4; %average signal power is 6.3477e-3
nsd=sqrt(noise_variance); %standard deviation of noise, used in randn function
%
% cyclically extend short trng symbols to 161 samples and add noise
rn3=st_trng_samples(rn2)+nsd*randn(1,161)+nsd*randn(1,161)*j;

buff1=nsd*randn(1,1000)+nsd*randn(1,1000)*j;
buff2=nsd*randn(1,1600)+nsd*randn(1,1600)*j;
%
% %create incoming signal with noise **short trng samples start at sample 1000 **
rx_signal=[buff1 rn3 buff2];
%
%
%
% search_window=1600; %scan from samples 1 to 1600
% D=16; %size of delay window = periodicity of short trng samples
```

```

%*****

D=16; %size of delay window, should equal periodicity of training samples
L=16; %number of multiple samples to use to compute decision statistic
nn=1+D;
rx_len=length(rx_signal);
rx_cross=rx_signal(nn:rx_len).*conj(rx_signal(nn-D:rx_len-D));
rx_pwr=(abs(rx_signal(nn-D:rx_len-D))).^2;

% compute moving sums of rx power and cross corr using delayed signal and L samples

n=1;
%
while n<=length(rx_pwr)-L-1
    ms_rx_pwr(n)=sum(rx_pwr(n:n+L-1));
    ms_rx_cross(n)=abs(sum(rx_cross(n:n+L-1)));
    ds(n+L)=(ms_rx_cross(n)^2)/(ms_rx_pwr(n))^2; %compute decision statistic
    n=n+1;
end

%
tt=[1:1:length(ms_rx_pwr)];
%
figure(2)
subplot(3,1,1)
plot(tt,ms_rx_pwr)
grid
axis([900 1200 min(ms_rx_pwr) max(ms_rx_pwr)])
title('Moving sum Rx power')
%
subplot(3,1,2)
plot(tt,ms_rx_cross)
grid
axis([900 1200 min(ms_rx_cross) max(ms_rx_cross)])
title('Moving sum Cross Correlation')
%
vv=[1:1:length(ds)];
subplot(3,1,3)
plot(vv,ds)
grid
% axis([900 1200 min(ds) max(ds)])
title('Decision statistic')

```

st_sym.m – This function is used to generate the short training symbols used by the Delay and Correlate detector and the Matched Filter detector. This function also performs the necessary coefficient mapping as shown in Figure 18.

```
function[mapped_st_trng_symbols]=st_sym
```

```
% this function returns the row vector of mapped short trng symbols per the  
% 802.11a standard
```

```
%short training symbols:(-26:26) ref IEEE 802.11a std 17.3.3
```

```
st_trng_symbols_initial=sqrt(13/6)*[0 0 1+j 0 0 0 -1-j 0 0 0 1+j 0 0 0 -1-j 0 0 0 -1-j 0 0 0  
1+j 0 0 0 0 0 0 -1-j 0 0 0 -1-j 0 0 0 1+j 0 0 0 1+j 0 0 0 1+j 0 0 0 1+j 0 0];
```

```
% mapped short training symbols per figure 109, 802.11 standard (inputs of the idft)
```

```
mapped_st_trng_symbols=[0 st_trng_symbols_initial(28:53) 0 0 0 0 0 0 0 0 0 0  
st_trng_symbols_initial(1:26)];
```

short_trng_samples.m – This function creates the 161 values of the short training samples and applies the windowing function per the IEEE standard [4].

```
function[mapped_short_trng_symbols]=st_trng_samples(short_trng_symbols)
```

```
% this function cyclically extends the 64 short training symbols to 161
```

```
% samples per the 802.11 standard (REF:table G.3)
```

```
mapped_short_trng_symbols=[short_trng_symbols short_trng_symbols  
short_trng_symbols(1:33)];
```

```
% 802.11 windowing function is also applied here
```

```
mapped_short_trng_symbols(1)=mapped_short_trng_symbols(1)*.5;
```

```
mapped_short_trng_symbols(161)=mapped_short_trng_symbols(161)*.5;
```

match_filter_detector.m – This m-file implements the matched filter detector that was extensively analyzed in this thesis. The output of this file is a graph of the detector's decision statistic and a depiction of the embedded short training samples in an AWGN channel.

```
clear all
```

```
% this file is the match filter detector. It uses the functions st_sym and
```

```
% short_trng_samples.
```

```
% author: Vik Sardana, MSEE student
```

```
% 22 DEC 2006
```

```
% get short training symbols
```

```
rn1=st_sym;
```

```

rn2=ifft(rn1);
%
% add noise to the delay buffer (or else divide by zero) and training samples
noise_variance=6.3477e-4; %average signal power is 6.3477e-3
nsd=sqrt(noise_variance); %standard deviation of noise, used in randn function

%cyclically extend short trng symbols to 161 samples and add noise
sig_plus_noise=st_trng_samples(rn2)+nsd*randn(1,161)+nsd*randn(1,161)*j;

% sig_plus_noise=st_trng_samples(rn2); %test only NO NOISE for now

%create 1600 samples of noise
buff1=nsd*randn(1,1000)+nsd*randn(1,1000)*j;
buff2=nsd*randn(1,1600)+nsd*randn(1,1600)*j;
%
%create incoming signal with noise **short trng samples start at sample 1000 **
rx_signal=[buff1 sig_plus_noise buff2];

%*****

D=16; %size of delay window, should equal periodicity of training samples

short_trng_sample_values=st_trng_samples(rn2);
stored_values=conj(short_trng_sample_values(17:32));

%compute matched filter decision stat

nn=[1:1:length(rx_signal)-(D+1)];
ds=zeros(1,length(nn));

nn=1;
while nn<=length(rx_signal)-D-1
    ds(nn)=abs(sum(rx_signal(nn:nn+(D-1)).*stored_values));
    nn=nn+1;
end

% length(rx_signal)-D-1

tt=[1:1:length(ds)];

```

```

gg=[1:1:length(rx_signal)];

figure(1)
subplot(2,1,2)
plot(tt,ds)
grid
title('Matched Filter Output')
ylabel('Decision Statistic')
axis([900 1200 0 .25])
subplot(2,1,1)
plot(gg,rx_signal)
grid
ylabel('Received Signal Amplitude')

```

ml_cfo_corrector_withLTS.m– This m-file implements the maximum likelihood (ML) frequency synchronizer that was extensively analyzed in this thesis. This file uses only 4 short training symbols and the long training symbols to evaluate the algorithm’s “robustness”. The number of the training symbols and the SNR can be modified at the user’s discretion. The output of this file is a series of histograms that depict the final frequency offset percentage (relative to sub-carrier spacing) for each trial run of the test signal.

```

% This m-file represents the data-aided maximum likelihood frequency offset corrector
% for 802.11g OFDM systems

```

```

% author Vik Sardana
% used for author's MSEE thesis 2006
clear all

```

```

% step 1 : get short training symbols to be used in data-aided cfo correction
short_training_samples_64=st_sym;

```

```

% step 2: take ifft in the transmitter
post_ifft_short_training_samples=ifft(short_training_samples_64);

```

```

% step 3: cyclically extend 64 short trng samples to 161 samples
short_training_samples_161=st_trng_samples(post_ifft_short_training_samples);

```

```

% step 4: step up short trng samples to transmit carrier frequency
tx_packet=transmitted_signal(short_training_samples_161);

```

```

% tx_packet=[tx_packet(1:64) zeros(1,97)];
tx_packet=tx_packet(1:64);

```

```

% step 5: corrupt transmitted signal with noise in channel
% noise_variance=6.3477e-4; % for 10 dB - average signal power is .10156/16=6.3477e-
3
% noise_variance=.00159447; % for 6 dB
noise_variance=.00317385; % .00317385 for 3 dB
% noise_variance=6.3477e-3; % for 0 dB

% noise_variance=.02527065; % .02525871 for -6 dB
nsd=sqrt(noise_variance); % standard deviation of noise, used in randn function

% step 5a: loop to collect data on frequency offset with correction
loop_count=1;
while loop_count<=500

tx_packet_plus_channel=tx_packet+nsd*randn(1,length(tx_packet))+nsd*randn(1,length(
tx_packet))*j;

CFO_correction_short_symbols(loop_count)=received_signal_with_correction(tx_packet
_plus_channel);
    frequency_offset_with_correction(loop_count)=long_symbol_corrector...
        (CFO_correction_short_symbols(loop_count),nsd);
    rx_packet=received_signal(tx_packet_plus_channel);

frequency_offset_without_correction(loop_count)=cfo_calculator_short_symbols(rx_pac
ket);
    loop_count=loop_count+1;
end

bins=[30:.5:50];
bins1=[-10:.5:10];

figure(1)
hist(CFO_correction_short_symbols,bins1)
grid
xlabel('Final frequency offset percent relative to sub-carrier spacing')
title('WITHOUT long training samples')

figure(3)
hist(frequency_offset_with_correction,bins1)
grid
xlabel('Final frequency offset percent relative to sub-carrier spacing')

```

```
title('With long training samples')
```

```
mean_percent_offset=mean(frequency_offset_with_correction)
var_percent_offset=var(frequency_offset_with_correction)
mean_percent_offset_uncorrected=mean(frequency_offset_without_correction)
var_percent_offset_uncorrected=var(frequency_offset_without_correction)
```

```
figure(4)
subplot(1,2,1)
hist(CFO_correction_short_symbols,bins1)
grid
xlabel('Final frequency offset percent relative to sub-carrier spacing')
title('WITHOUT long training samples')
ylabel('Trials')
subplot(1,2,2)
hist(frequency_offset_with_correction,bins1)
grid
xlabel('Final frequency offset percent relative to sub-carrier spacing')
title('With long training samples')
ylabel('Trials')
```

nls_receiver_with_correction6.m – This m-file implements the non-linear least squares (NLS) frequency synchronizer that was extensively analyzed in this thesis. This file uses only 4 short training symbols and the long training symbols to evaluate the algorithm’s “robustness”. The number of the training symbols and the SNR can be modified at the user’s discretion. The output of this file is a histogram that depicts the final frequency offset percentage (relative to sub-carrier spacing) for each trial run of the test signal.

```
% this is an m-file which implements the NLS
% frequency synchronization algorithm as described in my thesis
% author: Vik Sardana, DEC 2006
% THIS PARTICULAR VERSION USES ONLY 4 SHORT TRAINING SYMBOLS
```

```
clear all
% step 1 : get short training symbols to be used in data-aided cfo correction
short_training_samples_64=st_sym;

% step 2: take ifft in the transmitter
post_ifft_short_training_samples=ifft(short_training_samples_64);

% step 3: cyclically extend 64 short trng samples to 161 samples
short_training_samples=st_trng_samples(post_ifft_short_training_samples);

%now get long training samples
```



```

        XL(row,col)=stored_long_training_samples(nn); %circular convultion matrix for
long trng samples
        nn=nn+1;
    end
    nn=nn-1;
end

XL_inv=pinv(XL); %creates psuedo inverse matrix of long training symbol convultion
matrix

%now simulate frequency offset
% step 1: initialize variables
ts=50e-9; %sample time
f_tx=2.9e9; % transmitted carrier frequency
freq_spacing=312.5e3;
D=16; %periodicity of the short training samples
carrier_freq_offset=freq_spacing*-.40; %freq offset of -40% of sub-carrier spacing
f_rx=f_tx+carrier_freq_offset; % received carrier frequency=tx_freq + some freq offset.
f_delta=f_rx-f_tx;

n=1;
while n<=161
    yn(n)=received_noisy_samples(n).*(exp(j*2*pi*ts*n*f_delta)); %creating freq
offset for noisy samples
    n=n+1;
end

% step 3a: now form circular convolution matrix (a 16x16 matrix)
% based on short trng sample values. Convultion matrix called 'X'
stored_values=short_training_samples(17:48);
rxx=[stored_values stored_values stored_values stored_values stored_values
stored_values... %buffering short training samples to create matrix
    stored_values stored_values stored_values stored_values];

nn=1;
for col=[1:1:D]
    for row=[1:1:D]
        X(row,col)=rxx(nn);
        nn=nn+1;
    end
    nn=nn-1;
end
end

```

```

X_inv=pinv(X); %creates psuedo inverse matrix of short training symbols

% implement loop here to compute distribution of final percentage offset
no_of_trials=1;
while no_of_trials<=500

received_noisy_samples=short_training_samples+nsd*randn(1,length(received_noisy_sam
ples))...
    +nsd*randn(1,length(received_noisy_samples))*j;
    n=1;
    while n<=length(received_noisy_samples)
        rn(n)=received_noisy_samples(n).*(exp(j*2*pi*ts*n*f_delta)); %creating freq offset
for noisy samples
        n=n+1;
    end
    symbol1=rn(1:16); %create offset symbols 1 and 2
    received_sig=conj(symbol1)';
    n=17;
    symbol_number=1;
    while symbol_number<=4 %CHANGE #1 FROM 10 TO 4 TO TEST FOR
ROBUSTNESS (4 SHORT SYMBOLS ONLY)
        xx=1/symbol_number;
        h_hat=X_inv*received_sig*xx;
        H_HAT=fft(h_hat);
        mean_H_HAT=mean(H_HAT);
        angle=atan(imag(mean_H_HAT)/real(mean_H_HAT));
        fo=angle/(2*pi*ts*D);
        f_corr(symbol_number)=-1*fo;

percent_corrected(symbol_number)=100*(f_delta+f_corr(symbol_number))/freq_spacing
;
        nn=1;
        initial_sample=n;
        if n<=64 % CHANGE #2 FROM 160 TO 64
            while n<initial_sample+D

next_symbol_corrected(nn)=received_noisy_samples(n).*(exp(j*2*pi*ts*n*(f_delta+f_c
orr(symbol_number))));

rn_corrected(n)=received_noisy_samples(n).*(exp(j*2*pi*ts*n*(f_delta+f_corr(symbol_
number))));
            n=n+1;
            nn=nn+1;
        end
    end
end

```

```

        received_sig=received_sig+conj(next_symbol_corrected)';
        symbol_number=symbol_number+1;
    end

    final_percent_corrected_st(no_of_trials)=percent_corrected(length(percent_corrected));
    n=[1:1:128]; %now apply short samples freq correction to long trng samples

    noisy_long_samples=long_training_samples_no_cp+nsd*randn(1,128)+nsd*randn(1,128
    )*j;
    rn_long=noisy_long_samples(n).*(exp(j*2*pi*ts*n*(f_delta+f_corr(symbol_number-
    1))));
    received_long_samples=conj(rn_long(1:64)+rn_long(65:128))';
    h_est=.5*XL_inv*received_long_samples;
    H_est=fft(h_est);
    H_est_mean=mean(H_est);
    angle_long=atan(imag(H_est_mean)/real(H_est_mean));
    fo_long=angle_long/(2*pi*ts*Dlong);
    final_freq_offset_correction=f_corr(symbol_number-1)-fo_long;
    f_corr(symbol_number)=final_freq_offset_correction;

    final_offset_percentage(no_of_trials)=100*(f_delta+final_freq_offset_correction)/freq_s
    pacing; %CHANGE 4 COMMENT LINE 145, ADD LINE 146
    no_of_trials=no_of_trials+1;
end

figure(2)
bins=[-20:.5:20];
hist(final_offset_percentage,bins)
grid
xlabel('Frequency offset percentage relative to sub-carrier spacing')
ylabel('Trials')

```

lg_sym.m – this is a function used to create the long training symbols.

```

function[mapped_lg_trng_symbols]=lg_sym

% this function returns the row vector of mapped long trng symbols per the
% 802.11a standard

%long training symbols:(-26:26) ref IEEE 802.11a appendix G.3.2 pg 59
lg_trng_symbols_initial=[1 1 -1 -1 1 1 -1 1 -1 1 1 1 1 1 -1 -1 1 1 -1 1 1 1 1 0 1 -1 -
1 1 1 -1 1 -1 1 -1 -1 -1 -1 1 1 -1 -1 1 -1 1 -1 1 1 1];

% mapped short training symbols per figure 109, 802.11 standard (inputs of the idft)

```

```
mapped_lg_trng_symbols=[0 lg_trng_symbols_initial(28:53) 0 0 0 0 0 0 0 0 0 0
lg_trng_symbols_initial(1:26)];
```

lg_sym_time_domain.m – this function creates the time domain representation of the long training samples. It also calls the function `cyclic_prefix_creator.m`

```
function[long_training_samples]=lg_sym_time_domain
% Vik Sardana MSEE thesis
% this function used to generate the long training samples in the time domain
%generate long training samples:
long_training_symbols=lg_sym;
post_ifft_long_trng_symbols=ifft(long_training_symbols);
long_training_samples=cyclic_prefix_creator(post_ifft_long_trng_symbols);
cyclic_prefix_creator.m – this function creates the cyclic prefix for the long training
symbols and also windows the first and last sample in accordance with the standard.
```

```
function[output_samples]=cyclic_prefix_creator(input)
% this function cyclically extends the 64 post ifft inputs to 161
% samples per the 802.11 standard (REF:table G.3)
```

```
output_samples=[input(33:64) input input input(1)];
```

```
% 802.11 windowing function is also applied here
output_samples(1)=output_samples(1)*.5;
output_samples(161)=output_samples(161)*.5;
```

LIST OF REFERENCES

- [1] Rappaport, T. S., *Wireless Communications: Principles and Practice*, 2nd ed, Prentice Hall, 2002.
- [2] Weinstein, S. B., and Ebert, P. M., "Data Transmission by Frequency-Division Multiplexing Using the Discrete Fourier Transform," *IEEE Transactions on Communication Technology*, Vol. Com-19, No. 5, pp. 628-634, October 1971.
- [3] Institute of Electrical and Electronics Engineers, 802.11, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 18 March 1999.
- [4] Institute of Electrical and Electronics Engineers, 802.11a, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer Extension in the 5 GHz Band*, 16 September 1999.
- [5] Institute of Electrical and Electronics Engineers, 802.11g, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band*, 12 June 2003.
- [6] Pollet, T. and Moeneclaey, M., "Synchronizability of OFDM Signals." *Global Telecommunications Conference. GLOBECOM 1995*, IEEE, pp 2054-2058.
- [7] Yang, B., Letaief, K. B., Cheng, R. S., Cao, Z., "Timing Recovery for OFDM Transmission." *IEEE Journal on Selected Areas in Communications*. Vol. 18, No. 11, November 2000. pp. 2278-2291.
- [8] Heiskala, J. and Terry, J., *OFDM Wireless LANs: A Theoretical and Practical Guide*, Sams Publishing, Indianapolis, Indiana, 2002.
- [9] Pollet, T., Van Bladel, M., Moeneclaey, M., "BER Sensitivity of OFDM Systems to Carrier Frequency Offset and Weiner Phase Noise." *IEEE Transactions on Communications*. Vol. 43, Nos. 2/3/4, February/March/April 1995. pp 191-193.
- [10] Sklar, B., *Digital Communications: Fundamentals and Applications*, 2nd ed. Prentice Hall, Upper Saddle River, NJ, 2001.
- [11] Proakis, J.G., *Digital Communications*, 4th ed., McGraw Hill, New York, NY, 2001.
- [12] Lowham, K. D., "Synchronization Analysis and Simulation of a Standard IEEE 802.11g OFDM Signal," Master's Thesis, Naval Postgraduate School, Monterey, California, 2004.
- [13] Van Nee, R. and Prasad, R. *OFDM for Wireless Multimedia Communications*, Artech House, Boston, Massachusetts, 2000.
- [14] Tallarida, R.J., *Pocket Book of Mathematical Integrals and Mathematical Formulas*, 3rd Ed., CRC Press LLC, Boca Raton, FL, 1999.

- [15] Chang, R. W., "Synthesis of Band-Limited Orthogonal Signals for Multichannel Data Transmission." *The Bell System Technical Journal*, Vol. XLV, No. 10, December 1966. pp. 1775-1796.
- [16] Moose, P. H., "A Technique for Orthogonal Frequency Division Multiplexing Frequency Offset Correction," *IEEE Transactions on Communications*, Vol. 42, No. 10, pp. 2908-2914, October 1994.
- [17] Chiu, Y., Markovic, D., Tang, H., Zhang, N., "OFDM Receiver Design," *Technical Report for EE225C*, College of Engineering, University of California – Berkeley, 2000.
- [18] Count, P. A., "Performance Analysis of OFDM in Frequency-Selective, Slowly Fading Nakagami channels," Master's Thesis, Naval Postgraduate School, Monterey, California, 2001.
- [19] Greenberg, I., "Retraining WLAN Receivers for OFDM Operation", *Comms Design Newsletter*, 15 January 2002, http://www.commsdesign.com/design_corner/showArticle.jhtml?articleID=16504117 last accessed 15 February 2006.
- [20] Karaoglu, B., "A Comparison of Frequency Offset Estimation Methods in Orthogonal Frequency Division Multiplexing (OFDM) Systems," Master's Thesis, Naval Postgraduate School, Monterey, California, 2004.
- [21] Litwin, L., and Pugel, M., "The principles of OFDM," *RF Signal Processing*, http://rfdesign.com/mag/radio_principles_ofdm/index2.html, January 2001, last accessed 22 December 2006.
- [22] Robertson, R. C. and Beltz, N. E., "Digital Communications over Fading Channels," Technical Report, Naval Postgraduate School, Monterey, California, November 2004.
- [23] Tubbax, J., Côme, B., Van der Perre, L., Deneire, L., Donnay, S., Engels, M., "OFDM versus Single Carrier with Cyclic Prefix: a system based comparison," Vehicular Technology Conference, 2001. VTC 2001 Fall. IEEE VTS 54th Vol. 2, 7-11 October 2001 pp. 1115 - 1119 vol.2
- [24] Petrovic, D., Rave, W., Fettweis, G., "Common phase error due to phase noise in OFDM-estimation and suppression," *Personal, Indoor and Mobile Radio Communications*, 2004. PIMRC 2004. 15th IEEE International Symposium on Volume 3, 5-8 September 2004, pp. 1901 - 1905 Vol.3
- [25] Armada, A. G., "Understanding the Effects of Phase Noise in Orthogonal Frequency Division Multiplexing (OFDM)," *IEEE Transactions on Broadcasting*, Vol. 47, No. 2, June 2001.
- [26] Armada, A. G. and Calvo, M., "Phase Noise and Sub-Carrier Spacing Effects on the Performance of an OFDM Communication System," *IEEE Communications Letters*, Vol. 2, No. 1, January 1998.

- [27] Abhayawardhana, V. S. and Wassell, I. J., "Residual Frequency Offset Correction for Coherently Modulated OFDM Systems in Wireless Communication," http://www.cl.cam.ac.uk/Research/DTG/publications/public/vsa23/VTC_vsa23.pdf last accessed 16 June 2006.
- [28] Xiong, F. and Andro, M., "The Effect of Doppler Frequency Shift, Frequency Offset of the Local Oscillators, and Phase Noise on the Performance of Coherent OFDM Receivers," NASA Technical Report, Glenn Research Center, Cleveland, OH, March 2001.
- [29] Li, Y., Cimini, L.J., Jr., "Interchannel interference of OFDM in mobile radio channels," Global Telecommunications Conference, 2000. GLOBECOM '00. IEEE Volume 2, 27 November-1 December 2000. pp. 706 - 710 vol.2
- [30] Gertou, M., Karachalios, G., Triantis, D., Papantoni, K., and Dallas, P.I., "Synchronization Approach for OFDM based Fixed Broadband Wireless Access Systems," Intracom Telecommunications and Electronics SA report, Athens, Greece.
- [31] Bahai, A. R. S., Burton, R. S., Ergen, M., *Multi-carrier Digital Communications: Theory and Applications of OFDM*, Springer Science+Business Media Inc, New York, NY, 2004.
- [32] Tang, H., Lau, K. Y., Brodersen, R. W., "Synchronization Schemes for packet OFDM system," *Communications*, 2003, Vol. 5. ICC '03. IEEE International Conference on 11-15 May 2003. pp. 3346 – 3350.
- [33] Makundi, M., Hjørungnes, A., Laakso, T.I., "An interpolation-based frequency-synchronization scheme for OFDM systems," *Signal Processing Advances in Wireless Communications*, 2005 IEEE 6th Workshop on 5-8 June 2005. pp. 151 – 155.
- [34] Kragh, F., E., Notes on Sardana's MSEE thesis (unpublished), 2006-2007.
- [35] "Atheros eXtended Range XR Technology," White Paper, Atheros Communications, Sunnyvale, CA, 2004.
- [36] Technical specifications for D-Link Wireless Cardbus Adapter, model # DWL-G650, http://www.dlinkshop.com/product.asp?sku=2275506§ion_id=7#MyGearSection, last accessed 23 December 2006.
- [37] Hippenstiel, R. D., *Detection Theory: Applications and Digital Signal Processing*, CRC Press, Boca Raton, Florida, 2002.
- [38] Schmidl, T. M., Cox, D. C., "Robust Frequency and Timing Synchronization for OFDM," *IEEE Transactions on Communications*, Vol. 45, No. 12, December 1997.

- [39] Keller, T., Piazzo, L., Mandarini, P., Hanzo, L., "Orthogonal Frequency Division Multiplex Synchronization Techniques for Frequency-Selective Fading Channels," *IEEE Journal on Selected Areas in Communications*, Vol. 19, No. 6, June 2001.
- [40] Li, J., Liu, G., Giannakis, G., "Carrier Frequency Offset Estimation for OFDM-based WLANs," *IEEE Signal Processing Letters*, Vol. 8, No. 3, March 2001.
- [41] Daffara, F. and Chouly, A., "Maximum likelihood frequency detectors for orthogonal multicarrier systems," Communications, 1993. ICC 93. Geneva. Technical Program, Conference Record, IEEE International Conference on Vol. 2, 23-26 May 1993. pp. 766 - 771
- [42] Van de Beek, J., Sandell, M., Börjesson, P., "ML Estimation of Time and Frequency Offset in OFDM Systems," *IEEE Transactions on Signal Processing*, Vol. 45, No. 7, July 1997.
- [43] Weisstein, E. W., "Least Squares Fitting" Mathworld—A Wolfram Web Resource. <http://mathworld.wolfram.com/LeastSquaresFitting.html> last accessed on 03 July 2006.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, VA
2. Dudley Knox Library
Naval Postgraduate School
Monterey, CA
3. Chairman, Code EC
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA
4. Assistant Professor Frank Kragh, Code EC/Kh
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA
5. Professor R. Clark Robertson, Code EC/Rc
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA
6. Professor Tri T. Ha, Code EC/Ha
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA
7. Research Associate Donna L. Miller, Code EC
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA
8. LCDR Vikram Sardana
COMCARSTKGRU 7, Code N39
Coronado, CA