

Design and implementation of An OFDM transceiver

Submitted By:

Ahmed Khaled Gamal

Abdelrahman Mohamed Ibrahim

Mazen Ahmed Elaraby

Mohamed Hussein Mohamed

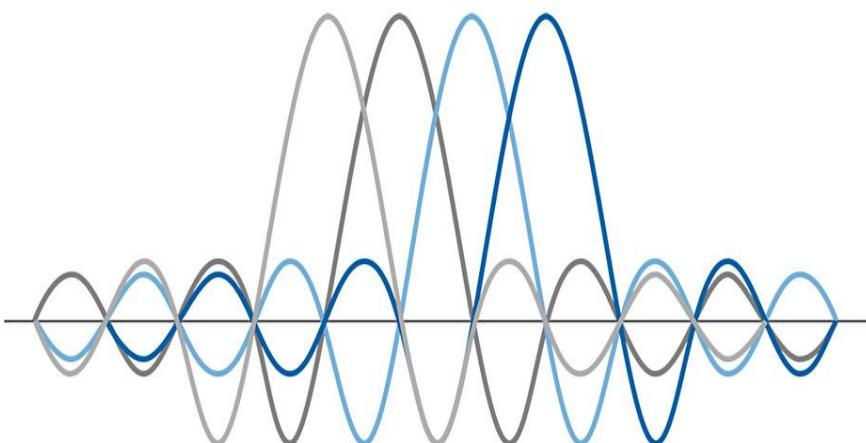


COMMUNICATION
SYSTEMS ENGINEERING
PROGRAM

Graduation Project Report

Supervisor:
Dr. Michael Ibrahim

Date:
30 January 2024



DECLARATION

We/I hereby certify that this material, which we/I now submit for assessment on the programme of study leading to the award of Bachelor of Science in communication Engineering is entirely our/my own work, that we/I have exercised reasonable care to ensure that the work is original, and does not to the best of our/my knowledge breach any law of copyright, and have not been taken from the work of others and to the extent that such work has been cited and acknowledged within the text of our/my work.

Signed: by all students

Name Ahmed Khaled Gamal	Ahmed Khaled
Name Abdelrahman Mohamed Ibrahim	Abdelrahman Bekhet
Name Mazen Ahmed Elaraby	Mazen El-Araby
Name Mohamed Hussein Mohamed	Mohamed Hussein

Date: 30 January 2024.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to Dr. Michael Ibrahim, our esteemed supervisor, whose guidance, expertise, and unwavering support played a pivotal role in the successful completion of this group project. Dr. Michael Ibrahim's insightful feedback, valuable suggestions, and encouragement have been instrumental throughout the research and development phases.

We also extend our appreciation to our fellow group members for their collaborative efforts and dedication. Each team member's unique contributions and commitment significantly enriched the project, creating a synergistic environment for creativity and innovation.

Furthermore, we acknowledge the resources and facilities provided by the faculty, which greatly contributed to the realization of our project goals.

ABSTRACT

Reliable high-speed data transmission in wireless environments necessitates a thorough understanding of channel impairments and their impact on communication systems. This thesis delves into the intricate world of Orthogonal Frequency Division Multiplexing (OFDM) channel modeling, performance analysis, and synchronization techniques, paving the way for robust data transmission amidst the uncertainties of wireless channels.

We establish a theoretical foundation by exploring path-loss models, large-scale and small-scale fading models, and statistical models for characterizing fading channels. Subsequently, we delve into the principles of OFDM, its advantages over single-carrier transmission, modulation techniques, guard intervals, and channel estimation methods. Utilizing Jake's model, we develop and validate accurate channel models, encompassing both flat and frequency-selective scenarios with Rayleigh and Rician processes.

Theoretical and simulated Bit Error Rate (BER) curves reveal the impact of diverse channel conditions and SNR levels on OFDM performance. We evaluate the efficacy of various coding schemes in mitigating these effects and boosting system resilience. The focus then shifts to ensuring reliable data transmission through accurate synchronization. Simulation results assess the effectiveness of different techniques in estimating and mitigating symbol time offset (STO) and carrier frequency offset (CFO) under diverse channel conditions. Packet detection techniques are also explored, facilitating the initiation of communication and decoding of received data.

To validate our findings in a real-world context, we present a hardware model designed to replicate the theoretical and simulated scenarios. Additionally, we identify promising avenues for future research, including investigating advanced communication techniques, adaptive systems capable of dynamically adjusting to changing channel conditions, and incorporating novel hardware implementations.

This thesis contributes to the ongoing pursuit of robust and reliable wireless communication, offering valuable insights into OFDM channel modeling, performance analysis, and synchronization techniques. It paves the way for further advancements in this ever-evolving field, ensuring seamless and reliable data transmission over wireless channels.

TABLE OF CONTENTS

LIST OF FIGURES.....	3
LIST OF TABLES.....	5
1 INTRODUCTION	6
1.1 Overview.....	6
1.2 Problem Statement.....	6
1.3 Project Description.....	6
1.4 Thesis Organization	7
1.4.1 Chapter 2: Literature Review.....	7
1.4.2 Chapter 3: Channel Modeling Results	7
1.4.3 Chapter 4: System Modeling Results.....	7
1.4.4 Chapter 5: Synchronization Results.....	7
1.4.5 Chapter 6: Hardware Model and Future Work.....	8
2 LITERATURE REVIEW	9
2.1 The Wireless Channel	9
2.2 Deterministic Path-loss Models	10
2.2.1 Friis Free-Space Propagation Model	10
2.2.2 Hata-Okumura Model for Outdoor Propagation.....	13
2.3 Large-Scale Fading Model.....	14
2.3.1 Log-Normal Shadowing	14
2.4 Small Scale Fading Models.....	16
2.4.1 The Multipath Channel Model - Clarke's Model	16
2.4.2 Channel Correlations & Power Spectra	17
2.4.3 The Power Delay Profile.....	20
2.4.4 The Doppler Power Spectrum.....	21
2.4.5 Jake's Model	22
2.4.6 Classification of Small-Scale Fading.....	24
2.5 Statistical Models for Fading Channels	24
2.5.1 The Rayleigh Process.....	24
2.5.2 The Rician Process	26
2.6 Orthogonal Frequency Division Multiplexing (OFDM).....	29
2.6.1 Introduction.....	29
2.6.2 limitations of Single-Carrier Transmission.....	29
2.6.3 OFDM vs. FDM.....	30
2.6.4 OFDM Modulation Overview.....	32
2.6.5 Guard Interval and Cyclic Prefix	33
2.7 Channel Estimation in OFDM.....	34
2.7.1 Why do we need channel estimation?.....	34
2.7.2 Pilot based channel estimation.....	34
2.7.3 Channel estimation techniques	36
2.8 Coded OFDM in 802.11 standards	39
2.8.1 Frame Generation in 802.11 PHY layer.....	40
2.9 Synchronization.....	54
2.9.1 Effect of Symbol Time Offset.....	54
2.9.2 Estimation techniques for STO	56
2.9.3 Effect of Carrier frequency Offset	57
2.9.4 Estimation Techniques for CFO	60

2.9.5	Packet detection	62
3	CHANNEL MODELLING RESULTS.....	77
3.1	Generation of Tap-Gain Processes	77
3.2	The Sum-of-Sinusoids Method	78
3.3	Flat Fading Channel with a Jake's Doppler Spectrum.....	80
3.3.1	Channel with a Rayleigh Process.....	80
3.3.2	Channel with a Rician Process.....	83
3.4	Frequency-Selective Channel with a Jake's Doppler Spectrum.....	85
3.4.1	Channel with a Rayleigh Process.....	86
3.4.2	Channel with a Rician Process.....	88
4	SYSTEM MODELLING RESULTS	92
4.1	Theoretical BER curves	92
4.1.1	AWGN channels	92
4.1.2	Rayleigh flat-fading channel.....	92
4.2	Simulated BER curves	93
4.2.1	Symbol to Noise Ratio	93
4.2.2	MODEM Results	93
4.2.3	Uncoded OFDM Results.....	96
4.2.4	Coded OFDM Results.....	98
4.3	Conclusion	100
5	SYNCHRONIZATION RESULTS.....	101
5.1	Simulation Results of STO.....	101
5.2	Analysis of STO Results.....	102
5.3	Simulation Results of CFO	103
5.4	Analysis of CFO Results	103
5.5	Packet Detection Simulation Results.....	104
5.5.1	Decision of Packet Existence.....	106
5.5.2	Estimating the Packet location, STO and CFO of Received Packet	108
5.6	Conclusion	110
6	HARDWARE MODEL AND FUTURE WORK.....	111
6.1	Advanced Testing and Validation	111
6.1.1	Software Defined Radio (SDR) Integration:.....	111
6.1.2	Technological Integration and Deployment:.....	111
6.2	Hardware Implementation and Efficiency:	111
6.2.1	Baseband Signal Processing Hardware Implementation:.....	112
6.2.2	Integration and Functional Verification:.....	112
6.2.3	DSP Blocks and Deep Learning Integration:.....	112
6.3	SDR Integration and Real-Time Operation:	113
6.4	Proposed Architecture	113
REFERENCES		115

LIST OF FIGURES

Figure 2-1: Classification of fading channels	9
Figure 2-2: Friis transmission model 2.4Ghz	12
Figure 2-3: Friis transmission model 5Ghz	12
Figure 2-4: Hata Path Loss Model.....	14
Figure 2-5: Log-Normal Shadowing 2.4Ghz.....	15
Figure 2-6: Log-Normal Shadowing 5Ghz.....	15
Figure 2-7: Scattering function under different constraints	17
Figure 2-8: Scattering function.....	18
Figure 2-9: Power Delay Profile	19
Figure 2-10: Doppler Power Spectrum.....	19
Figure 2-11: Jake's ACF.....	23
Figure 2-12: Jake's PSD	23
Figure 2-13: An instant of a gaussian process	25
Figure 2-14: Envelope Rayleigh Distribution.....	25
Figure 2-15: Frequency-Flat Rayleigh Distributed.....	26
Figure 2-16: Rician process.....	27
Figure 2-17: Rayleigh distribution Envelope.....	28
Figure 2-18: Frequency-Flat Rician Distributed.....	28
Figure 2-19: Single-carrier baseband communication system model	29
Figure 2-20 Subcarriers in FDM and OFDM systems.....	31
Figure 2-21: OFDM cyclic prefix.....	33
Figure 2-22: Block type pilots	35
Figure 2-23: Comb type pilot	35
Figure 2-24: Lattice type pilots.....	36
Figure 2-25: Weight factor in MMSE.....	37
Figure 2-26: Time domain correlation characteristic.....	39
Figure 2-27: Generated OFDM frame	40
Figure 2-28: scrambler diagram.....	43
Figure 2-29: burst error illustration	44
Figure 2-30: Deep fading.....	44
Figure 2-31: Interleaving and bit index	46
Figure 2-32: Block diagram of the convolutional encoder	47
Figure 2-33: Puncturing	48
Figure 2-34 FFT radix 2 butterfly diagrams	52
Figure 2-35 Windowing functions.....	53
Figure 2-36: Four Cases of Symbol Time Offset in OFDM [9]	55
Figure 2-37: Constellation diagram for case 1 and 2 of STO [9].....	55
Figure 2-38: Constellation diagram for case 3 and 4 of STO [9].....	55
Figure 2-39: STO Estimation in Time domain using CP Sliding windows [9]	56
Figure 2-40: STO Estimation using training symbol [9]	57
Figure 2-41: the frequency shift of $-\epsilon$ in the frequency domain [9].....	58
Figure 2-42: The effect of integer CFO on received signal [9].....	59
Figure 2-43: The constellation of received signal with CFO ϵ [9]	60
Figure 2-44: MSE vs CFO for OFDM periodicity 1 and 4 [9]	61
Figure 2-45: Packet Detection Algorithms [12].....	64
Figure 2-46: Sliding Window Received Signal Energy Detection [12]	65
Figure 2-47: Energy Detection using single sliding Window [12]	65
Figure 2-48 Double Sliding Windows Energy Detection [12]	66
Figure 2-49 : Double sliding Window Synchronization [12].....	67
Figure 2-50: Block Diagram of auto-correlation Detection [12].....	68
Figure 2-51 Auto-correlation algorithm applied to packet [12]	68
Figure 2-52 Cross-Correlation Block Diagram [12].....	69
Figure 2-53: Cross-Correlation Algorithm applied to short symbols [12].....	70
Figure 2-54: False Alarm rate at different SNRs and Thresholds [12]	71
Figure 2-55: Miss Detection rate at different SNRs and Thresholds [12].....	71

Figure 2-56: A decision for auto-correlation algorithm [12]	73
Figure 2-57: Comparison between Complex and simple decision [12].....	73
Figure 2-58: Comparison between different detection algorithms accuracy [12].....	74
Figure 2-59: Synchronization error comparison of different algorithms [12].....	75
Figure 3-1: TDL channel model	77
Figure 3-2: Filter methods	78
Figure 3-3: Sum-of-Sinusoid's method	79
Figure 3-4: Spaced time correlation function	79
Figure 3-5: Channel Impulse Response	80
Figure 3-6: Channel Frequency Response.....	81
Figure 3-7: Magnitude of Complex Channel Coefficients.....	81
Figure 3-8: Histogram of Magnitude of Complex Channel Coefficients.....	82
Figure 3-9: Doppler Power Spectrum.....	82
Figure 3-10: Channel Impulse Response:.....	83
Figure 3-11: Channel Frequency Response	83
Figure 3-12: Magnitude of Complex Channel Coefficients.....	84
Figure 3-13: Histogram of Magnitude of Complex Channel Coefficients.....	84
Figure 3-14: Doppler Power Spectrum.....	85
Figure 3-15: Channel Impulse Response	86
Figure 3-16: Channel Frequency Response	86
Figure 3-17: Magnitude of Complex Channel Coefficients.....	87
Figure 3-18: Histogram of Magnitude of Complex Channel Coefficients.....	87
Figure 3-19: Doppler Power Spectrum.....	88
Figure 3-20: Channel Impulse Response	88
Figure 3-21: Channel Frequency Response	89
Figure 3-22: Magnitude of Complex Channel Coefficients.....	89
Figure 3-23: Histogram of Magnitude of Complex Channel Coefficients.....	90
Figure 3-24: Doppler Power Spectrum.....	90
Figure 4-1: M-PSK AWGN.....	93
Figure 4-2: M-PAM AWGN	94
Figure 4-3: M-QAM AWGN.....	94
Figure 4-4: M-PSK Rayleigh.....	95
Figure 4-5: M-PAM Rayleigh	95
Figure 4-6: M-QAM Rayleigh.....	96
Figure 4-7: M-PSK CP-OFDM AWGN SER.....	96
Figure 4-8: M-QAM CP-OFDM AWGN SER.....	97
Figure 4-9: M-PSK CP-OFDM RAYLEIGH SER.....	97
Figure 4-10: M-QAM CP-OFDM RAYLEIGH SER.....	98
Figure 4-11: M-PSK Coded CP-OFDM AWGN SER.....	99
Figure 4-12: M-QAM Coded CP-OFDM AWGN SER.....	99
Figure 4-13: M-PSK Coded CP-OFDM RAYLEIGH SER.....	99
Figure 4-14: M-QAM Coded CP-OFDM RAYLEIGH SER.....	100
Figure 5-1: Simulation results of STO using squared difference method and Correlation method	101
Figure 5-2: IDE console result of squared difference and correlation methods.....	101
Figure 5-3: Squared Difference and Correlation STO estimation [9].....	102
Figure 5-4: Simulation result of CFO different algorithms	103
Figure 5-5: Running correlation detection (Schmidl & Cox) algorithm on 802.11a Short preamble symbols...	104
Figure 5-6: Theoretical vs Simulated results of Schmidl algorithm on Short Training symbols.....	105
Figure 5-7: Theoretical vs Simulated results of Schmidl algorithm on Long training symbol	106
Figure 5-8: Coarse Estimation of Packet location after applying the decision	107
Figure 5-9: Decision of packet existence after applying decision	108
Figure 5-10: Running cross-correlation algorithm on long preamble symbols to find fine STO	109
Figure 5-11: For 1000 packets (CFO=1.9, SNR=10dB), the False alarm, Packets missed, wrong packet location and Number of detections within 32 points	109
Figure 6-1 SDR integration	112
Figure 6-2 SDR & hardware interfaces	113
Figure 6-3: Proposed architecture.....	114

LIST OF TABLES

Table 2-1: Path loss exponent.....	10
Table 2-2: Hata model parameters.....	13
Table 2-3:Coherence bandwidth approximation.....	21
Table 2-4: multipath delay spread	24
Table 2-5: Doppler spread	24
Table 2-6: Modulation scheme parameters.....	41
Table 2-7: Modulation specific parameters	49
Table 2-8: Normalization factors	50
Table 2.9-1: Effect of STO in Time and Frequency domains.....	54
Table 2.9-2: Effect of CFO in Time and Frequency domains	58
Table 2.9-3: Running times of different packet detection algorithms	75
Table 4-1: Equations used to generate the BER curves for the AWGN channel.....	92
Table 4-2: Equations used to generate the BER curves for the Rayleigh channel	92
Table 5.5-1: The performance of packet detection.....	110

1 INTRODUCTION

1.1 OVERVIEW

The ubiquitous nature of wireless communication technologies has ushered in an era of unparalleled connectivity. From smartphones streaming information to connected homes automating daily tasks, the backbone of this revolution lies in the efficient transmission of data over wireless channels. However, the inherent unpredictability of these channels, characterized by fading and other impairments, poses significant challenges to reliable communication. Optimizing performance in such dynamic environments necessitates a thorough understanding of channel modeling and its integration into efficient communication systems.

This thesis delves into the captivating world of channel modeling for Orthogonal Frequency Division Multiplexing (OFDM) systems, a cornerstone technology employed in various high-speed wireless standards like Wi-Fi (802.11). We embark on a journey to comprehend the intricate dance between wireless channels and OFDM transmission, dissecting the effects of channel impairments and exploring techniques to mitigate their impact. Through theoretical analysis and meticulous simulations, we aim to illuminate the path towards robust and reliable data transmission in the face of wireless channel uncertainties.

1.2 PROBLEM STATEMENT

The fundamental hurdle in wireless communication lies in the inherent variability of the transmission medium. The wireless channel, unlike its wired counterpart, suffers from phenomena like path loss, multipath propagation, and fading, significantly impacting signal integrity and ultimately, system performance. OFDM, while inherently resilient to certain channel impairments, still experiences performance degradation in the presence of complex fading channels and synchronization errors. Understanding the intricacies of these impairments and devising effective channel models and communication strategies becomes crucial to guaranteeing reliable data transmission under diverse channel conditions.

1.3 PROJECT DESCRIPTION

This thesis addresses the aforementioned challenges by undertaking a comprehensive investigation into OFDM channel modeling, synchronization, and performance analysis. We delve into the theoretical underpinnings of wireless channels, delving into deterministic and stochastic path-loss models, large-scale and small-scale fading models, and statistical models for characterizing fading channels. Equipped with this theoretical framework, we embark on building robust channel models, particularly focusing on Jake's model for representing realistic multipath fading profiles.

Next, we shift our focus to the intricacies of OFDM and its sensitivity to channel impairments. We explore the advantages of OFDM compared to traditional single-carrier transmission and delve into the intricacies of OFDM modulation, guard intervals, and channel estimation techniques. Moving beyond theoretical analysis, we utilize simulations to evaluate the performance of OFDM systems in various channel scenarios. We explore the impact of channel conditions on Bit Error Rate (BER) and investigate the effectiveness of coding schemes in mitigating these effects.

To ensure successful data transmission, accurate synchronization is paramount. We explore the detrimental effects of symbol time offset (STO) and carrier frequency offset (CFO) on OFDM performance and delve into various synchronization techniques employed in 802.11 standards. Through meticulous simulations, we assess the accuracy and efficiency of these techniques in diverse channel conditions.

The culmination of our theoretical analysis and simulation results manifests in a comprehensive hardware model designed to validate our findings in a real-world environment. This model paves the way for future research endeavors investigating advanced communication techniques and adaptive systems capable of dynamically adapting to changing channel conditions.

1.4 THESIS ORGANIZATION

This thesis adopts a structured approach to delve into the intricacies of OFDM channel modeling, performance analysis, and synchronization techniques. To facilitate a comprehensive understanding, the key themes are laid out in distinct chapters, each building upon the foundation established in the previous ones.

1.4.1 Chapter 2: Literature Review

This chapter lays the intellectual groundwork for the subsequent analysis. It embarks on a journey through the existing body of knowledge in the relevant fields, exploring:

The Wireless Channel: Understanding the nature of the wireless channel, including path-loss models, large-scale and small-scale fading models, and statistical models for characterizing fading.

Orthogonal Frequency Division Multiplexing (OFDM): Delving into the principles of OFDM, its advantages over single-carrier transmission, modulation techniques, guard intervals, and channel estimation methods.

Coded OFDM in 802.11 Standards: Examining the integration of coding schemes within the 802.11 framework and its impact on system performance.

Synchronization in 802.11 Standards: Investigating the detrimental effects of symbol time offset (STO) and carrier frequency offset (CFO) and exploring existing synchronization techniques employed in 802.11.

1.4.2 Chapter 3: Channel Modeling Results

This chapter shifts the focus towards practical implementation, detailing the development and validation of accurate channel models. It delves into:

Generation of Tap-Gain Processes: Exploring various techniques for generating realistic tap-gain processes, including the sum-of-sinusoids method.

Flat Fading Channel with a Jake's Doppler Spectrum: Examining the behavior of OFDM systems in flat fading channels characterized by a Jake's Doppler spectrum, considering both Rayleigh and Rician processes.

Frequency-Selective Channel with a Jake's Doppler Spectrum: Extending the analysis to frequency-selective channels, again comparing the performance under Rayleigh and Rician processes.

1.4.3 Chapter 4: System Modeling Results

This chapter builds upon the established channel models to evaluate the performance of OFDM systems. It explores:

Theoretical BER curves: Deriving theoretical Bit Error Rate (BER) curves for OFDM systems in various channel environments, including AWGN, Rayleigh flat-fading, and frequency-selective channels.

Simulated BER curves: Validating the theoretical findings through detailed simulations across diverse channel conditions and SNR levels.

Impact of Coding: Evaluating the efficacy of various coding schemes in mitigating the impact of channel impairments and boosting system performance.

1.4.4 Chapter 5: Synchronization Results

This chapter focuses on ensuring reliable data transmission through accurate synchronization. It delves into:

Simulation Results of STO and CFO: Simulating the effects of STO and CFO on OFDM performance and evaluating the accuracy of different synchronization techniques in estimating and mitigating these offsets.

Analysis of STO and CFO Results: Analyzing the simulation results, drawing conclusions about the effectiveness of various synchronization techniques under different channel conditions and offering insights for further optimization.

Packet Detection Simulation Results: Exploring techniques for packet detection, which is crucial for initiating communication and decoding received data. This includes both the decision of packet existence and the estimation of packet location, STO, and CFO.

1.4.5 Chapter 6: Hardware Model and Future Work

This concluding chapter synthesizes the theoretical and empirical findings into a practical framework. It encompasses:

Hardware Model: Presenting a hardware model designed to validate the findings of the research in a real-world environment. This model serves as a platform for further exploration and experimentation.

Future Work: Identifying promising avenues for future research, including investigating advanced communication techniques, adaptive systems capable of dynamically adjusting to changing channel conditions, and incorporating novel hardware implementations.

2 LITERATURE REVIEW

2.1 THE WIRELESS CHANNEL

Multipath wireless communication channels are a fundamental aspect of modern wireless networks, playing a pivotal role in shaping the quality and reliability of wireless communications. These channels are characterized by the phenomenon where a transmitted signal traverses multiple paths before reaching the receiver. This occurs as a result of various environmental interactions such as reflection off buildings and other structures, diffraction over and around obstacles, and scattering caused by smaller objects in the environment. The resultant effect is a complex superposition of several versions of the transmitted signal at the receiver, each having traversed a different path with varying path lengths and propagation characteristics.

Within the realm of multipath propagation, two primary types of fading are observed: large-scale and small-scale fading. Large-scale fading, also known as long-term fading, is predominantly influenced by factors such as path loss and shadowing. Path loss refers to the reduction in signal power as the distance between the transmitter and receiver increases, following a certain propagation model. Shadowing, on the other hand, is caused by obstacles obstructing the line-of-sight path, resulting in signal attenuation as it passes through or around these obstructions. These effects vary more slowly with the spatial movement of the transmitter or receiver and are dependent on the larger-scale features of the environment.

Conversely, small-scale fading, or short-term fading, is characterized by rapid fluctuations in signal amplitude and phase over short distances or time intervals. This type of fading is a direct consequence of the multipath propagation where the multiple reflected, diffracted, and scattered signal components interfere with each other. Depending on their relative phases, this interference can be constructive (resulting in signal reinforcement) or destructive (leading to signal cancellation), causing rapid changes in the received signal's strength. Small-scale fading is heavily influenced by factors such as the relative motion between the transmitter and receiver, the distribution of scatterers in the environment, and the signal wavelength.

In the upcoming sections we will cover the main areas of study concerning the wireless communication channel. Namely:

- Deterministic Path-Loss Models
- Large-Scale Fading Model
- Small-Scale Fading Models
- Statistical Models for Fading Channels

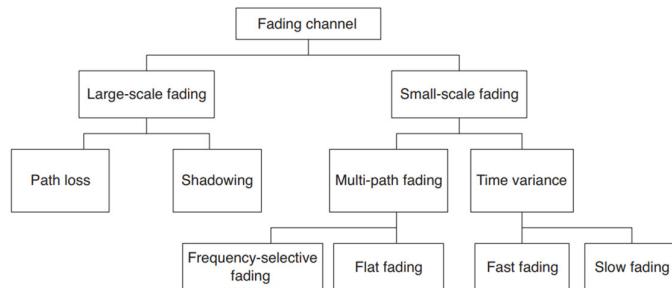


Figure 2-1: Classification of fading channels

2.2 DETERMINISTIC PATH-LOSS MODELS

2.2.1 Friis Free-Space Propagation Model

The Friis Free-Space Propagation Model is a fundamental theorem in the field of wireless communications, providing a critical framework for understanding the transmission of radio waves in an idealized, free-space environment. Developed by Harald Friis in 1946, this model offers a pivotal insight into the relationship between the power received by an antenna and the power transmitted, factoring in the distance between the antennas and the wavelength of the radio waves. Central to this model is the assumption of a clear, unobstructed path between the transmitter and receiver, which makes it particularly applicable to satellite and deep-space communications where line-of-sight propagation is predominant. The Friis equation elegantly encapsulates the inverse square law, indicating that the received power decreases with the square of the distance from the source, and it also incorporates the gain of the antennas involved in the transmission. This model's simplicity and precision make it a foundational tool in wireless communication, especially in the preliminary stages of system design where it provides a baseline for understanding signal attenuation over distance in free-space conditions. Its relevance persists in modern communication system design, serving as a starting point before more complex models that consider atmospheric or environmental factors are applied.

The model is mathematically stated as follows:

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi d)^2 L} \quad (2.2-1)$$

$P_r(d)$ is the received signal power in Watts expressed as a function of separation distance (d meters) between the transmitter and the receiver. P_t is the power of the transmitted signal's Watts. G_t & G_r are the gains of transmitter and receiver antennas when compared to an isotropic radiator with unity gain. λ is the wavelength of carrier in meters. L represents other losses that are not associated with the propagation loss. The parameter L may include system losses like loss at the antenna, transmission line attenuation, loss at various filters etc.[1]

The Friis equation can be modified to accommodate different environments, on the reason that the received signal decreases as the n th power of distance, where the parameter n is the path-loss exponent (PLE) that takes constant values depending on the environment that is modelled

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^n L} \quad (2.2-2)$$

Table 2-1: Path loss exponent

Environment	Path Loss Exponent (n)
Free space	2
Urban area cellular radio	2.7 to 3.5
Shadowed urban cellular radio	3 to 5
Inside a building - line-of-sight	1.6 to 1.8
Obstructed in building	4 to 6
Obstructed in factory	2 to 3

The alternative form of Friis equation in log scale is given by:

$$[P_r(d)]_{dBm} = [P_t]_{dBm} + [G_t]_{dBi} + [G_r]_{dBi} - [P_L]_{dB} \quad (2.2-3)$$

Where $[P_L]_{dB}$ is the propagation path loss in free space:

$$[P_L]_{dB} = 20 \cdot \log_{10} \left(\frac{\lambda}{4\pi} \right) - 10n \cdot \log_{10}(d) - 10 \cdot \log_{10}(L) \quad (2.2-4)$$

The following figures are plots for the free-space path loss for Wi-Fi frequencies for different path loss exponents:

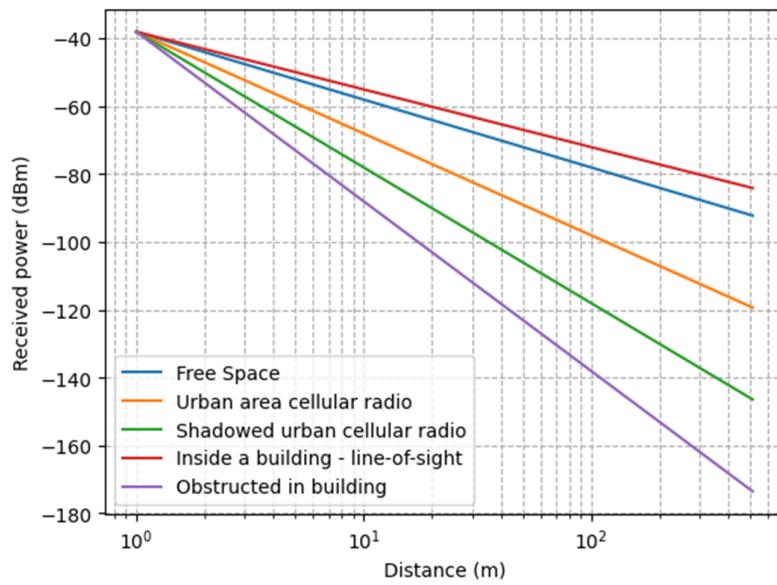


Figure 2-2: Friis transmission model 2.4Ghz

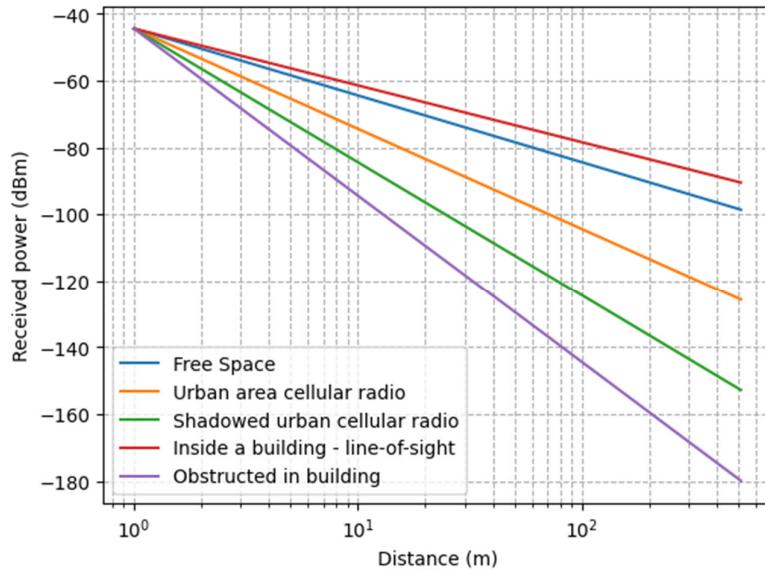


Figure 2-3: Friis transmission model 5Ghz

2.2.2 Hata-Okumura Model for Outdoor Propagation

Outdoor propagation models involve estimation of propagation loss over irregular terrains such as mountainous regions, simple curved earth profile, etc., with obstacles like trees and buildings. All such models predict the received signal strength at a particular distance or on a small sector. These models vary in approach, accuracy and complexity. Hata-Okumura model is one such model.

In 1986, Yoshihisa Okumura travelled around Tokyo city and made measurements for the signal attenuation from base station to mobile station. He came up with a set of curves which gave the median attenuation relative to free space path loss. Okumura came up with three set of data for three scenarios: open area, urban area and sub-urban area. Okumura model can be adopted for computer simulations by digitizing those curves provided by Okumura and using them in the form of look-up-tables. [2]

Hata, in 1980, came up with closed form expressions based on curve fitting of Okumura models. It is the most referred macroscopic propagation model. He extended the Okumura models to include effects due to diffraction, reflection and scattering of transmitted signals by the surrounding structures in a city environment.

The received power level is given by:

$$[P_r(d)]_{dBm} = [P_t]_{dBm} + [G_t]_{dB} - [P_L]_{dB} \quad (2.2-5)$$

The closed form expression for $[P_L]_{dB}$ is given by:

$$[P_L]_{dB} = A + B \log_{10}(d) + C \quad (2.2-6)$$

Here, the Tx-Rx separation distance (d) is specified in kilometres (valid range 1 km to 20 Km). The factors A, B, C depend on the frequency of transmission, antenna heights and the type of environment, as given next

$$A = 69.55 + 26.16 \log_{10}(f_c) - 13.82 \log_{10}(h_b) - a(h_m) \quad (2.2-7)$$

$$B = 44.9 - 6.55 \log_{10}(h_b) \quad (2.2-8)$$

- f_c = frequency of transmission in MHz, valid range – 150 MHz to 1500 MHz
- h_b = effective height of transmitting base station antenna in meters, valid range 30 m to 200 m
- h_m = effective receiving mobile device antenna height in meters, valid range 1m to 10 m
- $a(h_m)$ = mobile antenna height correction factor that depends on the environment
- C = a factor used to correct the formulas for open rural and suburban areas

Below the Hata model parameters ($a(h_m)$ & C) are tabulated:

Table 2-2: Hata model parameters[2]

Environment	$a(h_m)$	C
Open		$-4.78[\log_{10}(f_c)]^2 + 18.33\log_{10}(f_c) - 40.98$
Suburban	$[1.1\log_{10}(f_c) - 0.7]h_m - [1.56\log_{10}(f_c) - 0.8]$	$-2[\log_{10}(f_c/28)]^2 - 5.4$
Small/medium city		0
Metropolitan ($f_c \leq 200 \text{ MHz}$)	$8.29[\log_{10}(1.54h_m)]^2 - 1.1$	0
Metropolitan ($f_c > 200 \text{ MHz}$)	$3.2[\log_{10}(11.75h_m)]^2 - 4.92$	0

The Following figure plots path loss for different environments using the Hata model

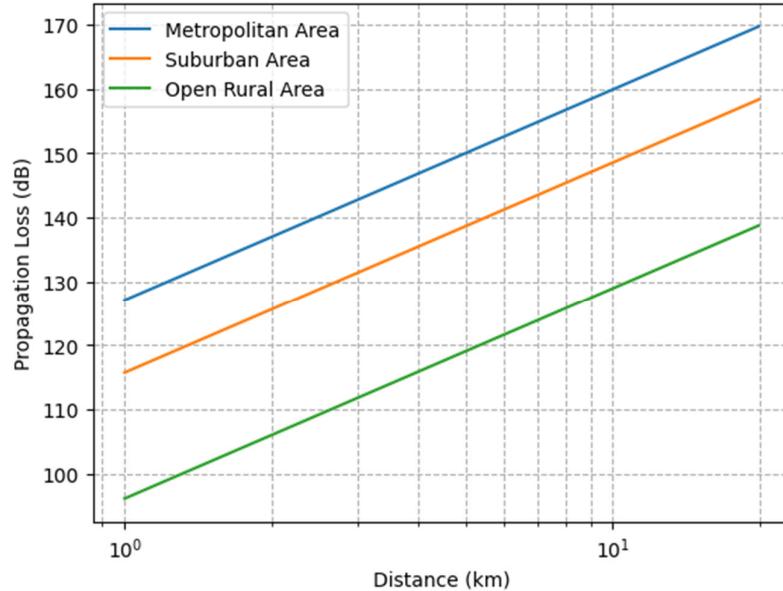


Figure 2-4: Hata Path Loss Model

2.3 LARGE-SCALE FADING MODEL

2.3.1 Log-Normal Shadowing

Log-Normal Shadowing model is an extension to the Friis free space model. It is used to predict the propagation loss for a wide range of environments, whereas, the Friis free space model is restricted to unobstructed clear path between the transmitter and the receiver. The model encompasses random shadowing effects due to signal blockage by hills, trees, buildings etc.

In the far field region of the transmitter, for distances beyond d_f , if $P_L(d_0)$ is the path loss at a distance d_0 meters from the transmitter, then the path loss at an arbitrary distance $d > d_0$ is given by:[2]

$$P_L(d)_{dB} = P_L(d_0)_{dB} + 10n \cdot \log_{10}\left(\frac{d}{d_0}\right) + \chi, \quad d_f \leq d_0 < d \quad (2.3-1)$$

χ is a zero-mean Gaussian distributed random variable with standard deviation σ expressed in dB, used only when there is a shadowing effect.

The following are plots for log-normal shadowing for Wi-Fi frequencies for different path loss exponents:

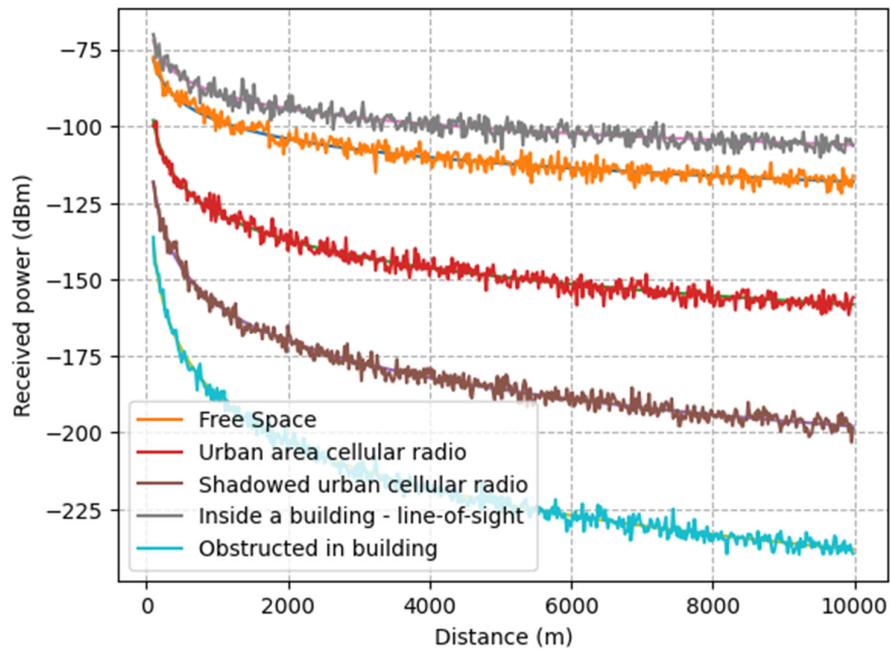


Figure 2-5: Log-Normal Shadowing 2.4Ghz

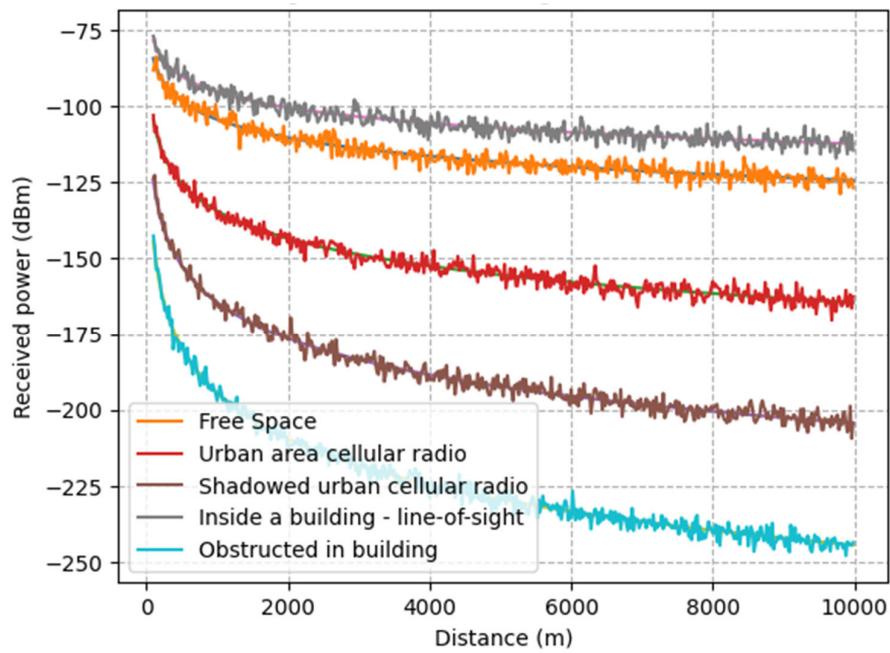


Figure 2-6: Log-Normal Shadowing 5Ghz

2.4 SMALL SCALE FADING MODELS

This section is adapted from chapter 2 in [3]

Propagation models that characterize the rapid fluctuations of the received signal strength over very short travel distances or short time durations are called small-scale models. Small-scale models are helpful in accurate performance evaluation of applications implemented over wireless channels.

2.4.1 The Multipath Channel Model - Clarke's Model

In a multipath channel, multiple copies of a signal travel different paths with different propagation delays τ and are received at the receiver at different phase angles and strengths. These rays add constructively or destructively at the receiver front end, thereby giving rise to rapid fluctuations in the channel. The multipath channel can be viewed as a linear time variant system where the parameters change randomly with respect to time. The channel impulse response is a two dimensional random variable – $h(t, \tau)$ that is a function of two parameters, instantaneous time t and the propagation delay τ . The channel is expressed as a set of random complex gains at a given time t and the propagation delay τ . The output of the channel $y(t)$ can be expressed as the convolution of the complex channel impulse response $h(t, \tau)$ and the input $x(t)$.

$$y(t) = \int_0^{\infty} h(t - \tau, \tau) x(t - \tau) d\tau \quad (2.4-1)$$

The Channel Auto-correlation function is given by:

$$R_{hh}(t_1, t_2; \tau_1, \tau_2) = E[h(t_1, \tau_1) h^*(t_2, \tau_2)] \quad (2.4-2)$$

The following assumptions can be made to restrict the channel model to the following specific set of categories:

- Wide sense stationary (WSS) channel model
- Uncorrelated scattering (US) channel model
- Wide sense stationary uncorrelated scattering (WSSUS) channel model

Wide sense stationary (WSS) channel model:

In this channel model, the impulse response of the channel is considered wide sense stationary (WSS), that is the channel impulse response is independent of time t , but it depends only on the difference between the time instants.

$$R_{hh}(\Delta t; \tau_1, \tau_2) = E[h(t, \tau_1) h^*(t + \Delta t, \tau_2)] \quad (2.4-3)$$

Uncorrelated scattering (US) channel model

Here, the individual scattered components arriving at the receiver front end (at different propagation delays) are assumed to be uncorrelated.

$$R_{hh}(t_1, t_2; \tau_1, \tau_2) = R_{hh}(t_1, t_2; \tau_1) \delta(\tau_1 - \tau_2) \quad (2.4-4)$$

Wide sense stationary uncorrelated scattering (WSSUS) channel model

The WSSUS channel model combines the aspects of WSS and US channel model that are discussed above. it gives the following model:

$$R_{hh}(\Delta t, \tau) = E[h(t, \tau) h^*(t + \Delta t, \tau)] \quad (2.4-5)$$

2.4.2 Channel Correlations & Power Spectra

The Fourier transform on the variable Δt gives specific insight to channel properties in terms of propagation delay τ and the Doppler frequency f simultaneously. The Fourier transform of the two-dimensional autocorrelation function for a WSSUS channel on the variable Δt is called scattering function and is given by:

$$S(f, \tau) = \int_{-\infty}^{\infty} R_{hh}(\Delta t, \tau) e^{-j2\pi f \Delta t} d\Delta t \quad (2.4-6)$$

Two important relationships can be derived from the scattering function - power delay profile (PDP) and Doppler power spectrum. Both of them affect the performance of a given wireless channel

Power delay profile, also called as multipath intensity profile $S(\tau)$ gives the signal intensity received over a multipath channel as a function of propagation delays. It can be obtained from scattering function, by integrating it over the entire frequency range (removing the dependence on Doppler frequency).

$$S(\tau) = \int_{-\infty}^{\infty} S(f, \tau) df \quad (2.4-7)$$

Power delay profile is a function of time which can be transformed to frequency domain by taking Fourier transform. Fourier transform of power delay profile is called spaced-frequency correlation function. Spaced-frequency correlation function describes the spreading of a signal in frequency domain. This gives rise to the derived channel parameter - coherence bandwidth.

Similarly, the Doppler power spectrum $S(f)$ can be obtained by integrating the scattering function over the entire range of propagation delays.

$$S(f) = \int_{-\infty}^{\infty} S(f, \tau) d\tau \quad (2.4-8)$$

The Doppler power spectrum describes the output power of the multipath channel in frequency domain. The Doppler power spectrum when translated to time-domain by means of inverse Fourier transform is called spaced-time correlation function. Spaced-time correlation function describes the correlation between different scattered signals received at two different times as a function of the difference in the arrival times. It gives rise to the concept of coherence time

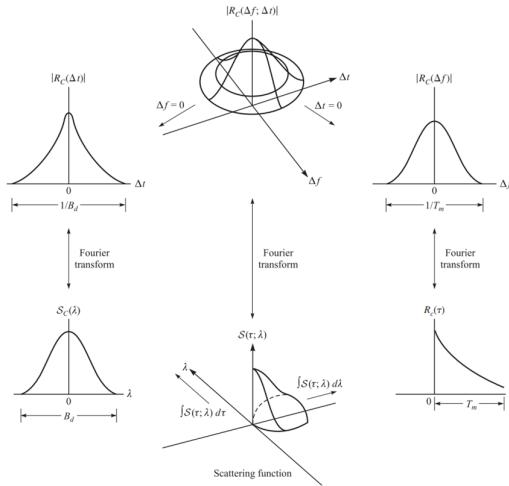


Figure 2-7: Scattering function under different constraints

An Example Scattering Function:

The scattering function combines information about Doppler spread and the path delays. Consider the typical U-shaped Jakes Doppler power spectral density, with the maximum doppler shift and an exponential delay spread with mean τ_{rms} . The scattering function that describes the amount of scatter power per frequency and time bin is given by:

$$S(f, \tau) = \frac{P_{local-mean}}{4\pi f} \cdot \frac{1}{m \sqrt{1 - \left(\frac{f-f_c}{f_m}\right)^2}} \exp\left(-\frac{\tau}{\tau_{rms}}\right) \quad (2.4-9)$$

where, $P_{local-mean}$ is the received power level averaged over an area of tens or hundreds of meters. The following script plots the given scattering function as a 3D plot. It also plots the power delay profile and Doppler power spectrum

The following is a plot of the scattering function:

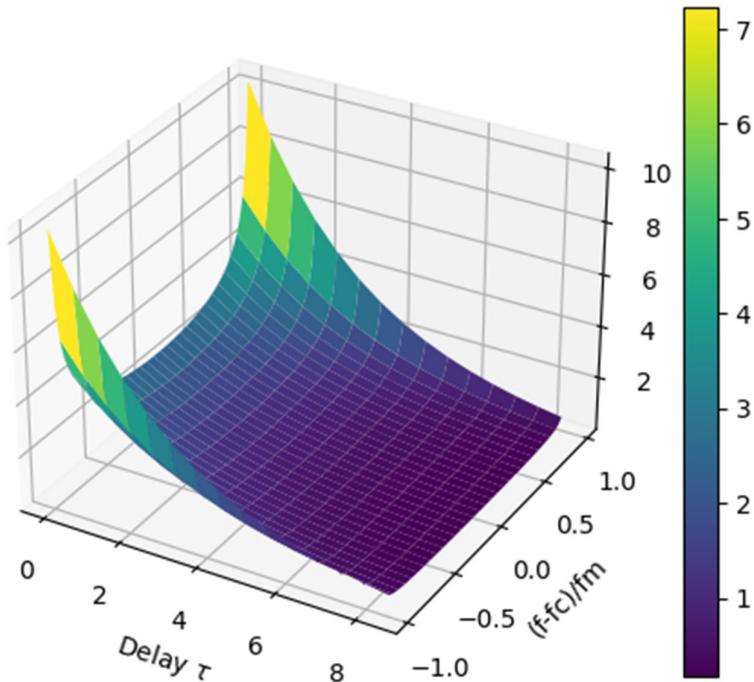


Figure 2-8: Scattering function

The following plot is of the power delay profile:

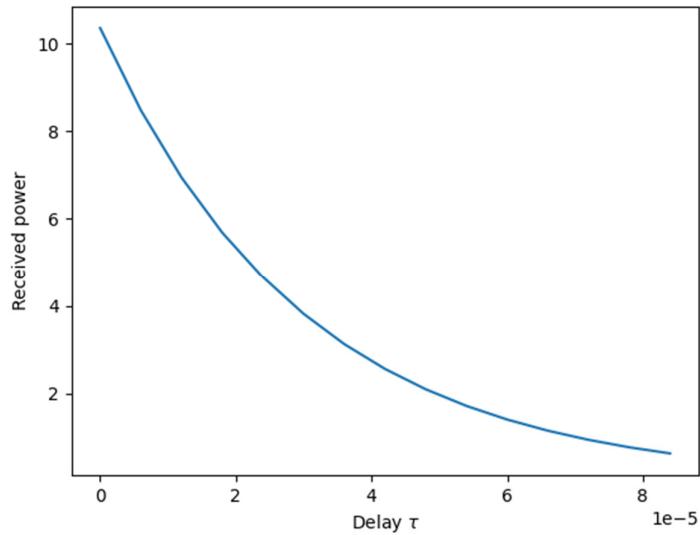


Figure 2-9: Power Delay Profile

The following plot is of the Doppler Power Spectrum:

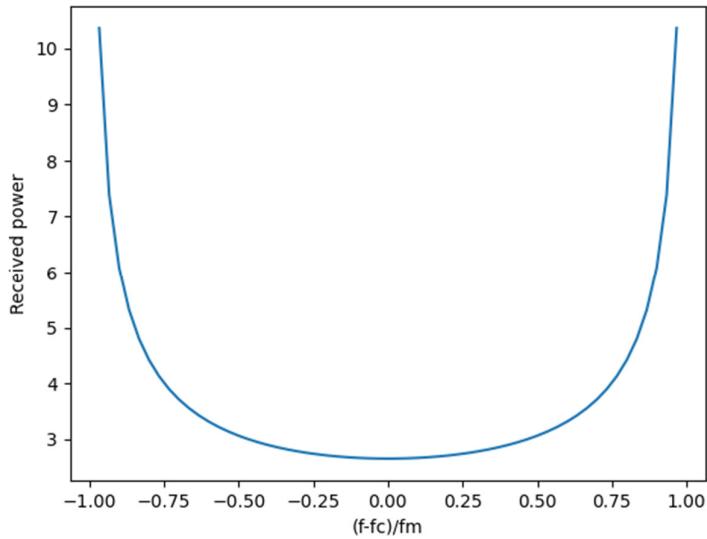


Figure 2-10: Doppler Power Spectrum

2.4.3 The Power Delay Profile

A multipath channel can be characterized in multiple ways for deterministic modelling and power delay profile (PDP) is one such measure. In a typical PDP plot, the signal power on each multipath is plotted against their respective propagation delays.

The PDP, when expressed as an intensity function $S(\tau)$, gives the signal intensity received over a multipath channel as a function of propagation delays. The PDP plots can be obtained as the spatial average of the complex channel impulse response as:

$$S(\tau) = R_{hh}(0, \tau) = E[|h(t, \tau)|^2] \quad (2.4-10)$$

For continuous PDP, the RMS delay spread (τ_{rms}) can be calculated as:

$$\tau_{rms} = \sqrt{\frac{\int_{-\infty}^{\infty} S(\tau)(\tau - \tilde{\tau})^2 d\tau}{\int_{-\infty}^{\infty} S(\tau) d\tau}} \quad (2.4-11)$$

where, the mean delay is given by:

$$\tilde{\tau} = \frac{\int_{-\infty}^{\infty} S(\tau) \tau d\tau}{\int_{-\infty}^{\infty} S(\tau) d\tau} \quad (2.4-12)$$

For discrete PDP:

$$\tau_{rms} = \sqrt{\frac{\sum_i p_i (\tau_i - \tilde{\tau})^2}{\sum_i p_i}} \quad (2.4-13)$$

$$\tilde{\tau} = \frac{\sum_i p_i \tau_i}{\sum_i p_i} \quad (2.4-14)$$

Knowledge of the delay spread is essential in system design for determining the trade-off between the symbol rate of the system and the complexity of the equalizers at the receiver. The ratio of RMS delay spread (τ_{rms}) and symbol time duration (T_{sym}) quantifies the strength of intersymbol interference (ISI). Typically, when the symbol time period is greater than 10 times the RMS delay spread, no ISI equalizer is needed in the receiver.

Maximum delay spread (T_m) and the symbol time period (T_{sym}) can be used to classify a channel into frequency selective or non-selective category. This classification can also be done using coherence bandwidth (a derived parameter from spaced frequency correlation function which in turn is the frequency domain representation of power delay profile).

Spaced frequency correlation function and coherence bandwidth:

The spaced-frequency correlation function, also called frequency correlation function (FCF), is defined as the Fourier transform of power delay profile.

$$R(\Delta f) = F.T[S(\tau)] = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} S(f, \tau) df \right] e^{-j2\pi\Delta f \tau} d\tau \quad (2.4-15)$$

The frequency correlation function is a useful indicator of dependencies of channel responses as a function of difference between two given frequency components ($\Delta f = f_1 - f_2$). It is used to assess the distance between two frequency components for the values to be effectively correlated. Alternatively, it is an indicator of signal spreading in frequency domain often measured using a derived parameter called coherence bandwidth, which is similar to the delay spread parameter derived from power-delay profile.

The normalized correlation between the frequency responses of the channel at two different frequencies is given by:

$$\rho(\Delta f) = \frac{\int_{-\infty}^{\infty} |h(t, \tau)|^2 \exp(-j\Delta f \tau) d\tau}{\int_{-\infty}^{\infty} |h(t, \tau)|^2 d\tau} \quad (2.4-16)$$

The coherence bandwidth is inversely proportional to the delay spread. There is no exact relationship between the two parameters as it is heavily dependent on the shape of the power-delay-profile or the frequency correlation function of the channel

Approximate equations for coherence bandwidth can be found in the following table:

Table 2-3:Coherence bandwidth approximation

Approximate equation	Correlation $\rho(\Delta f)$	Comments
$B_c \approx \frac{1}{\tau_{rms}}$	≈ 0	The channel gain at frequency f is independent of the gain at $f + B_c$. Changing frequency by B_c results in a channel with a completely different channel gain.
$B_c \approx \frac{1}{5\tau_{rms}}$	≈ 0.5	The channel gain at frequency f is similar to the gain at $f + B_c$. Equalizer not required if B_c is greater than symbol rate.
$B_c \approx \frac{1}{50\tau_{rms}}$	> 0.9	The channel gain at frequency f almost same as the gain at $f + B_c$.

2.4.4 The Doppler Power Spectrum

Doppler power spectrum gives the power profile as a function of Doppler frequency. The range of Doppler frequency values for which the Doppler power profile is essentially nonzero is defined as the Doppler spread of the channel. Doppler spread is also referred as fading rate of the channel. A large value of Doppler spread characterizes a rapidly changing channel.

The Doppler power spectrum is obtained by integrating the scattering function over the entire range of propagation delays

$$S(f) = \int_{-\infty}^{\infty} S(f, \tau) d\tau \quad (2.4-17)$$

Spaced time correlation function and coherence time:

The inverse Fourier transform of Doppler power spectrum defines the spaced time correlation function, that quantifies the correlation of two samples of channel response taken at different times.

$$R(\Delta t) = F.T^{-1}[S(f)] = \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} S(f, t) dt \right) e^{j2\pi f \Delta t} df \quad (2.4-18)$$

A derived parameter of this function is the coherence time T_c that indicates the period of time over which the fading process is highly correlated. The reciprocal of Doppler spread B_d is a measure of coherence time T_c . Coherence time provides a strong indication, about the rate at which the channel attenuation and the phase shift change over time.

Therefore, due to inverse relationship between Doppler spread B_d and coherence time T_c , a channel with larger Doppler spread when compared to the signal bandwidth B_s is also characterized by smaller coherence time T_c

with respect to the symbol period T_{sym} . Hence, for the case $Bs < Bd$ or equivalently $T_{sym} > T_c$, the transmitted signal undergoes time-selective or fast fading. Alternatively, a channel with smaller Doppler spread when compared to the signal bandwidth, that is $Bs > Bd$ (equivalently, larger coherence time compared to signalling interval, that is $T_{sym} < T_c$, undergoes time-nonselective or slow fading.

2.4.5 Jake's Model

The Fading Channel is usually statistically described by one of the famous distributions like: Rayleigh, Rician, Nakagami-m. These models give us a statistically description of the fading channel coefficient.

The fading model chosen assumes that all scattered waves arrive, each with their own random angle of arrival that are uniformly distributed within $[0, 2\pi]$. Hence, such scattered waves arrive with different random Doppler frequency shifts. When viewed in frequency domain, the Doppler effect causes frequency dispersion in the received signal that can be visualized in the form of probability density function of frequency, also referred as power spectral density or power spectrum.

The power spectral density $S_{\mu\mu}(f)$ for the scattered components described by The Rayleigh model is given by:

$$S_{\mu\mu}(f) = S_{\mu i \mu i}(f) + S_{\mu q \mu q}(f) \quad (2.4-19)$$

The inverse Fourier transform of power spectral density is the auto-correlation function. The corresponding auto-correlation function for the scattered components in the previous equation is given by:

$$r_{\mu\mu}(\tau) = r_{\mu i \mu i}(\tau) + r_{\mu q \mu q}(\tau) \quad (2.4-20)$$

The auto-correlation function for the Jakes model is given by:

$$r_{\mu i \mu i}(\tau) = r_{\mu q \mu q}(\tau) = \sigma_0^2 J_0(2\pi f_{max} \tau) \quad (2.4-21)$$

where, $J_0(\cdot)$ Denotes the 0-th order Bessel function of the first kind The corresponding PSD:

$$S_{\mu i \mu i}(f) = S_{\mu q \mu q}(f) = \sigma_0^2 \pi f \max \sqrt{1 - \left(\frac{f}{f_{max}}\right)^2}, \& |f| \leq f_{max} \quad (2.4-22)$$

Plotting Jake's ACF:

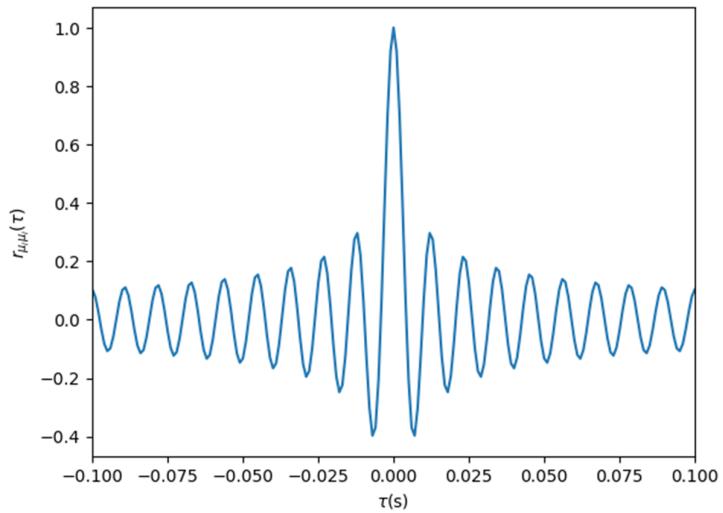


Figure 2-11: Jake's ACF

Plotting Jake's PSD:

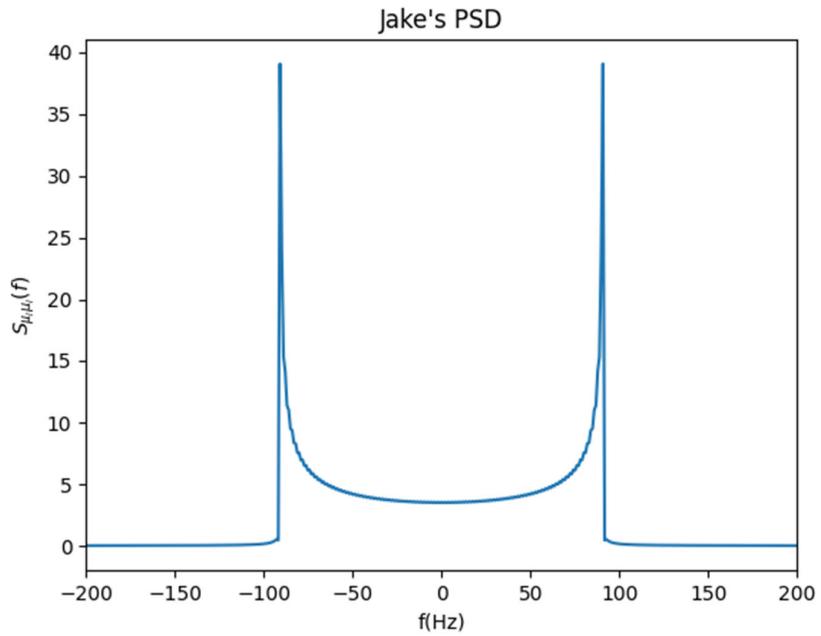


Figure 2-12: Jake's PSD

2.4.6 Classification of Small-Scale Fading

Classification of fading channels based on multipath delay spread:

Table 2-4: multipath delay spread

Frequency non-selective (flat) fading	Coherence BW of channel $B_c >$ BW of signal B_s Delay spread $\tau_{rms} <$ Symbol period T_{sym}
Frequency-selective fading	Coherence BW of channel $B_c <$ BW of signal B_s Delay spread $\tau_{rms} >$ Symbol period T_{sym}

Classification of fading channels based on Doppler spread:

Table 2-5: Doppler spread

Fast fading	High Doppler spread compared to signal BW: $B_d > B_s$ Coherence time $T_c <$ Symbol period T_{sym}
Slow fading	Low Doppler spread: $B_d < B_s$ Coherent time $T_c >$ Symbol period T_{sym}

2.5 STATISTICAL MODELS FOR FADING CHANNELS

In mobile communication system modelling, the knowledge of statistical properties of the received signal's envelope is also important. For a frequency-flat fading channel, the received signal can be simply modelled by multiplying the transmitted signal with a suitably chosen stochastic process. Depending on the absence or presence of a line-of-sight path between the transmitter and the receiver, we can choose Rayleigh process or Rice process for modelling the envelope of the receiver signal.

If a signal is transmitted over a frequency-flat channel, the received signal is expressed as a sum of all incident waves. In complex baseband equivalent model, the received signal is described as the sum of all scattered components and the line-of-signal component.

2.5.1 The Rayleigh Process

2.5.1.1 The Complex Gaussian Model:

The sum of all scattered components gives rise to a zero-mean complex Gaussian random process[2]

$$\mu(t) = \mu_i(t) + j\mu_q(t) \quad (2.5-1)$$

where, $\mu_i(t)$ and $\mu_q(t)$ are statistically uncorrelated real-valued colored Gaussian random processes with variance equal to σ_0^2

The following is a plot of an instant of a complex gaussian process:

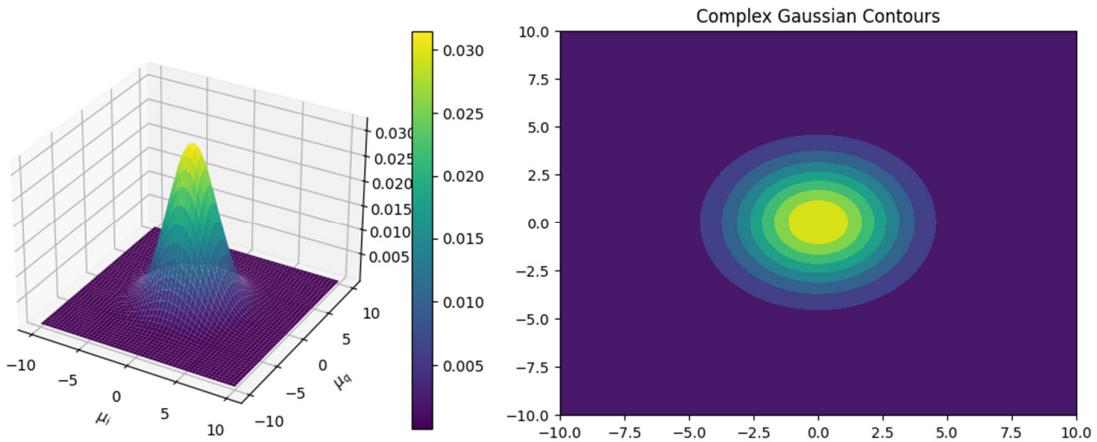


Figure 2-13: An instant of a gaussian process

2.5.1.2 The Rayleigh Distribution

The envelope of a complex Gaussian random process is a Rayleigh process.[2]

$$\zeta(t) = |\mu(t)| = |\mu_i(t) + j\mu_q(t)| \quad (2.5-2)$$

For the generic Rayleigh process $\zeta(t)$, the probability density function of the received signal amplitude is given by the Rayleigh distribution

$$f_R(|\zeta(t)|) = f_R(r) = \frac{r}{\sigma_h^2} \exp\left(-\frac{r^2}{2\sigma_h^2}\right) \quad (2.5-3)$$

The following is plot for the envelope of the Rayleigh distribution:

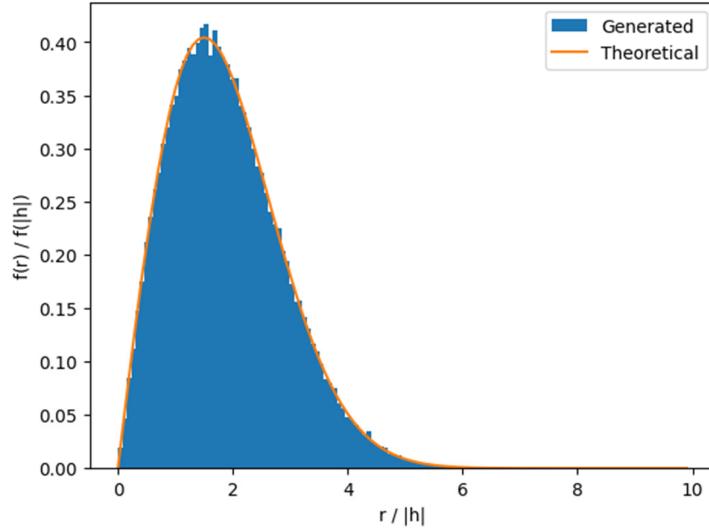


Figure 2-14: Envelope Rayleigh Distribution

The following is a plot of Modelling a Frequency-Flat Rayleigh Distributed Wireless Fading Channel:

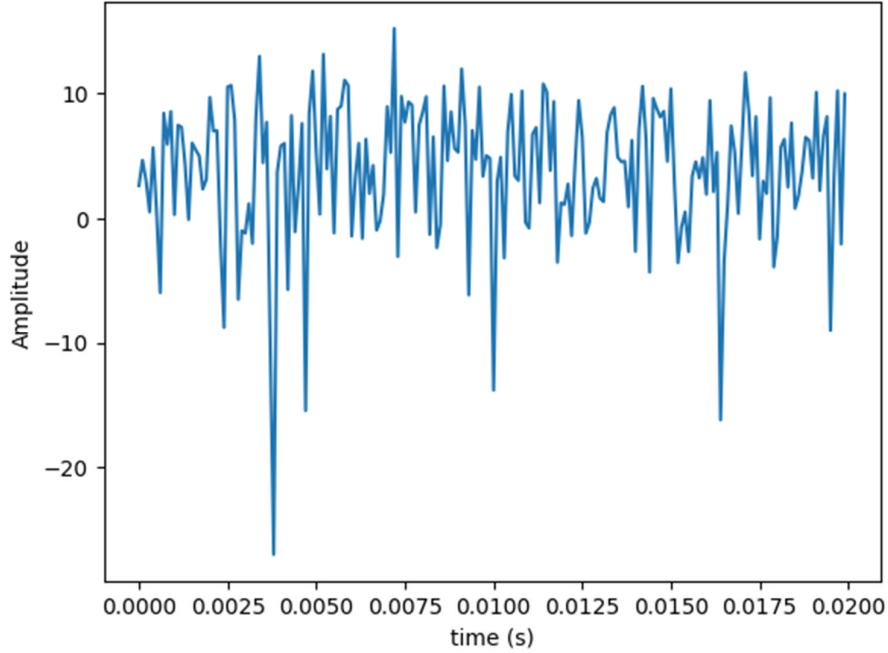


Figure 2-15: Frequency-Flat Rayleigh Distributed

2.5.2 The Rician Process

2.5.2.1 A Mean-Shifted Complex Gaussian

The Rayleigh process describes a multipath fading channel with no dominant Line-of-Sight (LoS) Component. The more generic case would be a superpositions of scattered Non-LoS components plus a dominant LoS component. The line-of-sight component of the received signal is described by a general time-variant complex random process

$$m(t) = m_i(t) + j m_q(t) \quad (2.5-4)$$

Therefore, the superposition of both the scattered and the line-of-sight components is described by adding equation:

$$\mu_p(t) = \mu(t) + m(t) \quad (2.5-5)$$

Which renders an instant of the Rician process a complex gaussian with a shifted mean

The following plot is of an instant of a Rician process:

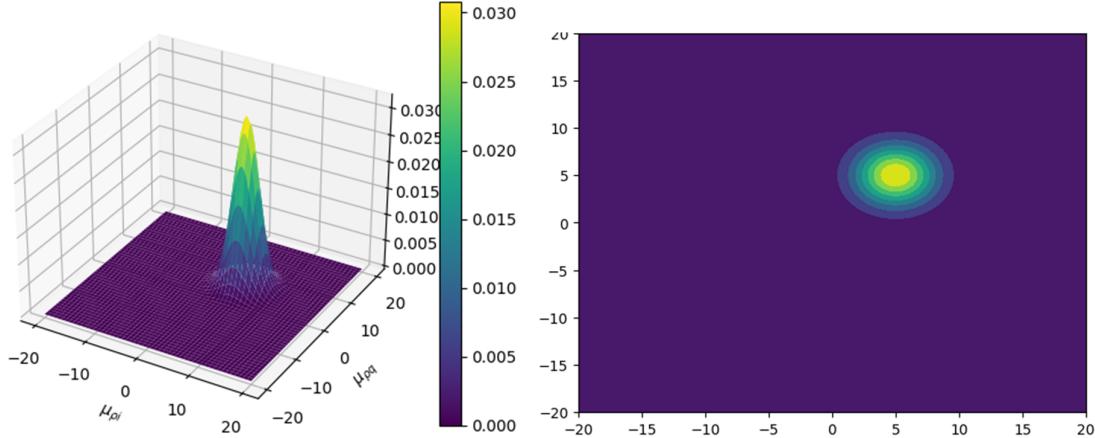


Figure 2-16: Rician process

2.5.2.2 The Rician Distribution

The envelope process for this generic case is a Rice process[2]

$$\zeta(t) = |\mu_p(t)| = |\mu(t)| + m(t) \quad (2.5-6)$$

For the generic Rice process $\xi(t)$, the probability density function of the received signal amplitude is given by the Rice distribution

$$p_\xi(x) = \frac{x}{\sigma_0^2} e^{-\frac{x^2+\rho^2}{2\sigma_0^2}} I_0\left(\frac{x\rho}{\sigma_0^2}\right), \quad x \geq 0 \quad (2.5-7)$$

In the context of Rice distribution, the Rician K factor, denoted by κ , captures the ratio of power of line-of-sight component and the sum of power of all the scattered components.

$$\kappa = \frac{E[m(t)^2]}{E[\mu(t)^2]} = \frac{\rho^2}{2\sigma_0^2} \quad (2.5-8)$$

If $\rho \rightarrow 0$, i.e., $\kappa \rightarrow 0$, the Rice process becomes a Rayleigh process, whose amplitude variations are statistically described by Rayleigh distribution.

And as ρ increases, the distribution approaches a Gaussian.

A Rician statistical channel model for flat-fading can be described by two parameters: Ω and κ . $\Omega = \rho^2 + \sigma_0^2$ is the total power from line-of-sight path and scattered paths, and κ is ratio of powers in the line-of-sight path and the scattered path as described by equation

An alternative parametrization for the Rician distribution could be given in terms of Ω and κ as follows:

$$p_\xi(x) = \frac{2x(K+1)}{\Omega} \exp\left(-\frac{(K+1)x^2}{\Omega} - K\right) I_0\left(2x\sqrt{\frac{K(K+1)}{\Omega}}\right), \quad x \geq 0, \quad K \geq 0, \quad \Omega \geq 0 \quad (2.5-9)$$

A simple Rician fading channel can be simulated by setting the variables g1 and g2, as follows:

$$g_1 = \sqrt{\frac{K}{2(1+K)}} \quad (2.5-10)$$

$$g_2 = \frac{1}{\sqrt{2(1+K)}} \quad (2.5-11)$$

The following is plot for the envelope of the Rayleigh distribution:

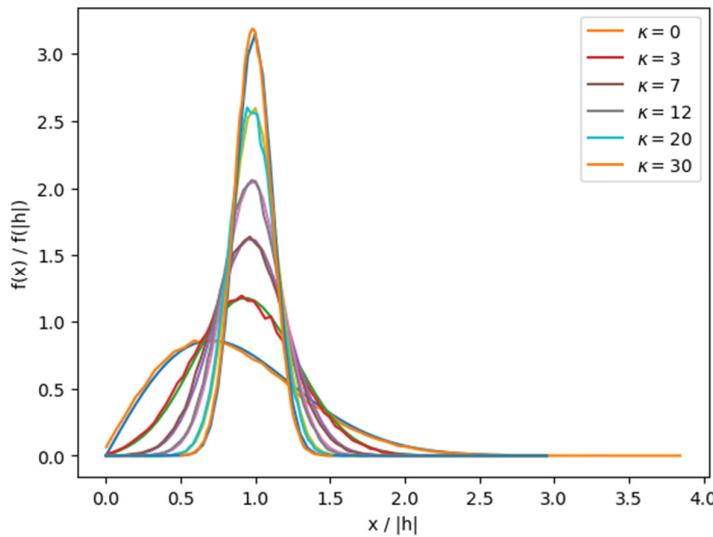


Figure 2-17: Rayleigh distribution Envelope

The following is a plot of Modelling a Frequency-Flat Rician Distributed Wireless Fading Channel:

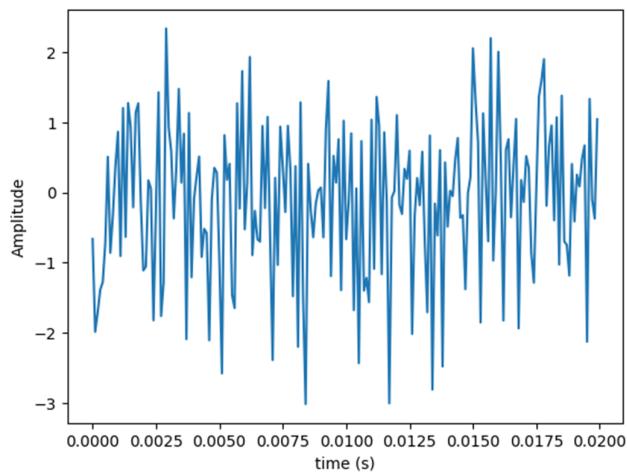


Figure 2-18: Frequency-Flat Rician Distributed

2.6 ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING (OFDM)

This chapter provides a concise overview of the basics of the OFDM modulation scheme and outlines the process of system design. Initially, we delve into the fundamental principles and mathematical model of a baseband OFDM system. Subsequently, we introduce the specifications related to the OFDM modulation technique for the Physical Layer (PHY) of the IEEE 802.11a standard, along with an examination of its architecture.

Building upon this groundwork, we present simulation results to validate the theoretical aspects, providing insights into the performance of the described OFDM system. This chapter serves as a gateway to understanding OFDM, bridging theoretical foundations with practical applications and laying the foundation for a comprehensive grasp of this modulation scheme.

2.6.1 Introduction

Orthogonal Frequency-Division Multiplexing (OFDM) stands as a digital modulation technique where a signal undergoes division into several narrowband channels at varying frequencies. Notably, there has been a global shift towards the adoption of OFDM in wireless communication applications, positioning it as an emerging technology capable of facilitating high data rates.

The realm of wireless communication is currently experiencing unparalleled growth, driven by significant advancements in the field. OFDM, serving as a digital modulation method, strategically divides a signal into multiple narrowband channels at distinct frequencies. This approach has found widespread application in various technologies, including Asymmetric Digital Subscriber Line (ADSL) services, IEEE 802.11a/g, IEEE 802.16a, Digital Audio Broadcast (DAB), and digital terrestrial television broadcasts like DVD in Europe and ISDB in Japan, along with advancements in 4G, IEEE 802.11n, IEEE 802.16, and IEEE 802.20[4].

The operational principle of OFDM lies in its capability to transform a frequency-selective channel into a parallel assembly of frequency-flat sub-channels, rendering it resilient against significant delay spreads. OFDM boasts numerous advantages over conventional techniques, with its subcarrier frequencies strategically chosen to ensure mathematical orthogonality over a single symbol period. This meticulous selection mitigates the impact of inter-channel interference, enhancing the overall reliability and efficiency of the communication system[5].

2.6.2 limitations of Single-Carrier Transmission

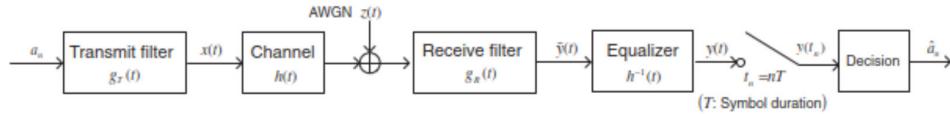


Figure 2-19: Single-carrier baseband communication system model

assuming that the effect of the channel is perfectly compensated by the equalizer and ignoring any added noise, the sampled output signal of the equalizer can be expressed as:

$$y(t_n) = \sum_{m=-\infty}^{\infty} a_m g((n-m)T) \quad (2.6-1)$$

We can observe from the previous equation that for $m=n \Rightarrow g((n-m)T)$ equals 0 inversely for $m \neq n$ this term will remain as inter-symbol interference (ISI) to a_n the ISI is caused by a trail of the overall impulse response, which could degrade the performance of a digital communication system. ISI can be completely eliminated by fulfilling the Nyquist criterion which guarantees an ISI-free communication even with a short symbol period T for high-rate transmission in a single-carrier transmission system.

In order to support the symbol rate of Rs symbols per second, the minimum required bandwidth is the Nyquist bandwidth, which is given by $Rs=2(Hz)$. It implies that wider bandwidth is required to support a higher data rate in a single-carrier transmission.

However, as the symbol rate increases, the signal bandwidth becomes larger. When the signal bandwidth becomes larger than the coherence bandwidth in the wireless channel, the link suffers from multi-path fading, incurring the inter-symbol interference (ISI)[2].

Adaptive equalizers are employed to deal with the ISI incurred by the time-varying multi-path fading channel. However, the complexity of an equalizer increases with the data rate. The optimum detector for the multi-path fading channel is a maximum-likelihood sequence detector (MLSD), which is very complex. When the data rate is too high, the complexity of the equalizer becomes prohibitive.

Therefore, a high data rate single-carrier transmission may not be feasible.

To overcome the frequency selectivity of the wideband channel experienced by single-carrier transmission, multiple carriers can be used for high-rate data transmission. A frequency-selective wideband channel is a channel where different frequencies experience different attenuation and phase shifts. This can lead to inter-symbol interference (ISI), which can corrupt the received signal[5].

In multi-carrier transmission, the high data rate signal is divided into several narrowband signals, which are then transmitted in parallel over the wideband channel. This reduces the frequency selectivity of the channel, which simplifies the equalization problem and reduces ISI.

To approximate a frequency-selective wideband channel with multiple frequency-flat narrowband channels, we can use multiple filters to divide the wideband channel into several subchannels. Each subchannel is a narrowband channel that is centred at a different frequency. At the transmitter, each subchannel is then modulated onto a different carrier frequency. At the receiver, the subchannels are demodulated and recombined to form the original high data rate signal.

2.6.3 OFDM vs. FDM

In the rapidly evolving landscape of wireless communication, the demand for higher data rates has become more pronounced, introducing complexities in dealing with unpredictable wireless communication channels. The challenges escalate at higher data rates, where channel distortion is more pronounced, making the recovery of transmitted data difficult, and sometimes even impossible[2]. This predicament necessitates the implementation of complex and expensive receivers.

Frequency Division Multiplexing (FDM) presents a solution to address frequency-selective fading by transforming it into flat fading, simplifying the receiver's equalizer requirements to a one-tap equalizer for channel estimation. FDM modulates multiple low-rate signals using separate carrier frequencies, requiring efficient spacing to avoid overlap and reduce receiver filter complexity. This method is also applied to wide-band single wireless channels by breaking them into sub-channels to counter inter – symbol interference. By narrowing the bandwidth of sub-channels, only flat fading is tolerated, minimizing inter-symbol interference and eliminating the need for costly and complex equalizers.

While FDM offers advantages, it suffers from low spectrum efficiency due to the considerable bandwidth of separated wireless sub-channels compared to what is needed for a wide-band signal wireless channel.

In response to these challenges, the Orthogonal Frequency Division Multiplexing (OFDM) technique was proposed to enable high data rate communications in selective fading channels. OFDM is a type of multicarrier transmission, where numerous low-rate parallel subcarriers are regularly transmitted over the frequency band to ensure orthogonality. Unlike classical non-overlapping FDM systems, OFDM's sub-channels overlap without interference, resulting in almost 50% bandwidth savings compared to non-overlapping systems.

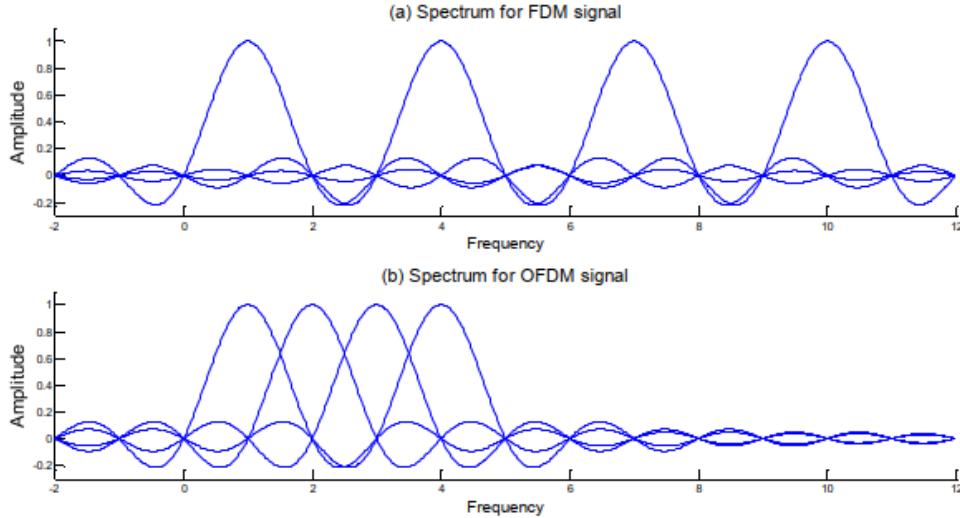


Figure 2-20 Subcarriers in FDM and OFDM systems.

The subcarriers in an OFDM signal are arranged in such a way that the peak of each subcarrier exactly shows up at the point where all the others have zero amplitude. Unlike the FDM, the subcarriers are overlapped. However, they do not interfere with one another due to the inherent orthogonality between these subcarriers. This orthogonality property is satisfied when the spacing between the centre frequencies of two adjacent subcarriers is the reciprocal of the OFDM symbol period. In the frequency domain, the subcarriers should be integer multiples of the spacing frequency, i.e., for an OFDM symbol of period T , any two subcarriers follow the following equation:

$$\int \exp\left(j2\pi \frac{i-m}{T}\right) dt = \begin{cases} 0 & i \neq m \\ T & i = m \end{cases} \quad (2.6-2)$$

OFDM employs a sinusoidal generator for each subcarrier, traditionally requiring a bank of generators for transmission and reception. However, the use of Discrete Fourier Transform (DFT) has significantly simplified the OFDM system's structure by replacing these banks at the transmitter and receiver, leading to a substantial reduction in system complexity[2].

As previously elucidated, the fundamental concept behind Orthogonal Frequency Division Multiplexing (OFDM) revolves around the subdivision and transmission of a singular high-rate data stream across multiple low-rate subcarriers, each possessing orthogonality to its counterparts. If we envision each subcarrier employing the same modulation method, under similar data rate conditions, the OFDM signal with N subcarriers extends the signal duration to N times that of a single-carrier system, while maintaining an identical signal bandwidth between the two systems. Conversely, when considering an equivalent signal duration, the data rate of an OFDM system escalates to N times, achieving this enhancement at a minimal cost of increased bandwidth.

In the pragmatic realm of wireless communication systems, signals must navigate diverse paths to reach the receiver, influenced by the presence of obstacles along their propagation route. As alluded to earlier, one pivotal reason for the widespread adoption of OFDM lies in its adept handling of delay spread induced by multipath fading. It effectively eradicates both inter-symbol interference (ISI) and inter-carrier interference (ICI), offering a robust solution to the challenges posed by signal propagation through varying paths. To accomplish this objective, a cyclically extended prefix is introduced at the transmitter, albeit with a marginal trade-off involving a slight degradation in bandwidth efficiency. This strategic incorporation of a cyclically extended prefix emerges as a cost-effective method to mitigate the adverse effects of multipath fading, ensuring the reliability and coherence of the transmitted signals in OFDM systems.

In an OFDM system, input data undergoes conversion from a single-stream data format to a parallel data stream using a serial-to-parallel converter (SPC)[6]. Subsequently, the data is modulated using an Inverse Fast Fourier

Transformer (IFFT), breaking down the wide-band signal into narrow-band parallel signals. At the receiver, the data streams, demodulated by a Fast Fourier Transformer (FFT), are converted back into a single data stream using a parallel-to-serial converter. This streamlined approach enhances the efficiency and implementation of OFDM systems.

Hence, OFDM has numerous advantages, including:

- Achieving high data rates through the simultaneous transmission on multiple subcarriers.
- Maximizing spectral utilization by maintaining close spacing between subcarriers.
- Enhancing resilience against time dispersion in multipath channels by elongating symbol duration.
- Bolstering resistance to both Inter-Symbol Interference (ISI) and Inter-Carrier Interference (ICI) through the incorporation of a cyclic prefix (CP).
- Mitigating implementation complexity through the streamlined utilization of forward and inverse Fast Fourier Transform (FFT/IFFT) pairs.
- Facilitating simple equalization processes for the removal of channel effects in the frequency domain.

2.6.4 OFDM Modulation Overview

Conventionally, the multi-carrier transmitter consists of a set of modulators, each with different carrier frequencies. The transmitter then combines the modulator outputs and generates the transmitted signal. Suppose that the N data to be transmitted are X_k where $k = 0, 1, \dots, N-1$, where X_k is a complex number in a given constellation, such as QPSK or QAM. Also suppose that the $k^{\{th\}}$ carrier frequency for X_k is f_k . Then, the complex-valued multi-carrier transmitter output is given by[5]:

$$x(t) = \sum_{k=0}^{N-1} X_k e^{j2\pi f_k t} \quad (2.6-3)$$

Contemporary communication systems frequently opt for digital implementation in both transmitters and receivers whenever feasible. In the context of a digital transmitter, the generation of its output adopts a sampled-data approach. This involves defining time as $t = nT_s$, where T_s denotes the sample interval. Consequently, the output of the digital multi-carrier transmitter is now determined within this framework.

$$x(nT_s) = \sum_{k=0}^{N-1} X_k e^{j2\pi f_k nT_s} \quad (2.6-4)$$

In addition, if the carrier frequencies are uniformly spaced in the frequency domain by a frequency spacing of f_s then,

$$x(nT_s) = \sum_{k=0}^{N-1} X_k e^{j2\pi k f_s nT_s} \quad (2.6-5)$$

These individual carriers are commonly referred to as subcarriers, and typically, an additional modulation step is employed to shift these subcarriers to a higher frequency band. The provided formula, apart from a multiplying constant ($1/N$), represents the equation of an N-point Inverse Discrete Fourier Transform (IDFT). When N is a power of two, a plethora of efficient algorithms and architectures are available for implementing this IDFT operation. The presence of such efficient digital realization in the OFDM transmitter is what renders OFDM technology a viable and effective solution for advanced communication systems. The absence of specific time-domain windowing on the OFDM symbol to shape its waveform results in sinc-shaped spectra for the OFDM subcarriers, as depicted before in previous figure.

2.6.5 Guard Interval and Cyclic Prefix

Given that the OFDM symbol comprises multiple narrow-band signals with an extended symbol duration, OFDM modulation demonstrates robustness against multi-path delay spread, especially when the delay is significantly smaller than the symbol duration. To enhance resistance against multipath delay spread in the radio channel, guard intervals can be strategically inserted between long OFDM symbols. These guard intervals should possess sufficient length to surpass the maximum delay spread. At the receiver end, these guard intervals, housing time-domain symbol interference, are removed, ensuring adjacent symbols remain interference-free.

However, introducing a zero signal during the guard time can lead to intercarrier interference (ICI), disrupting the preservation of orthogonality between subcarriers. To address this challenge, the OFDM symbol undergoes cyclic extension within the guard interval. This involves replicating a portion of the OFDM symbol, with the extension being an integer multiple of Inverse Fast Fourier Transform (IFFT) sampling time and surpassing the maximum expected delay spread. This cyclic extension ensures the transmitted OFDM signal retains periodicity, preserving the crucial property of orthogonality[2].

Cyclic extension can be implemented through three different methods: cyclic prefix, cyclic suffix, or timing advance. The widely adopted approach is cyclic prefix due to its simplicity in implementation, avoiding the need for rotation processes at the receiver, as in the case of cyclic suffix. Nonetheless, the addition of guard intervals between OFDM symbols comes at the expense of reduced spectral efficiency.

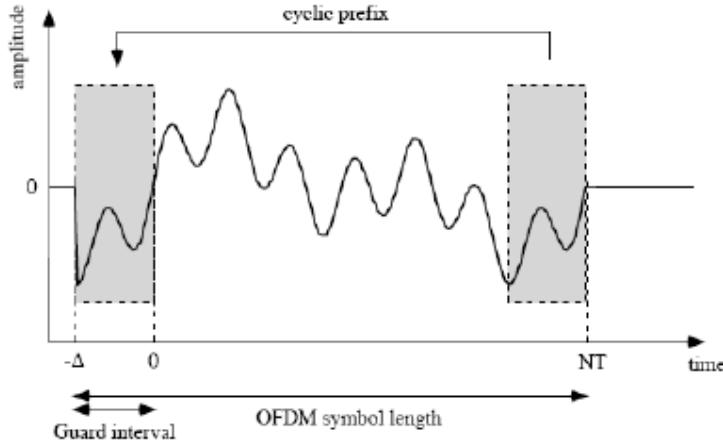


Figure 2-21: OFDM cyclic prefix

2.7 CHANNEL ESTIMATION IN OFDM

This section is adapted from chapter 6 in [2]

2.7.1 Why do we need channel estimation?

In an OFDM system, a wide bandwidth is divided into multiple subcarriers that are closely spaced in frequency. These subcarriers are particularly susceptible to variations in the transmission channel, such as multipath fading and frequency-selective fading. Channel estimation becomes essential to accurately determine the channel response at each subcarrier, allowing for the mitigation of distortions and the compensation of the channel-induced impairments. By acquiring knowledge of the channel conditions, OFDM systems can employ techniques like equalization to combat the effects of multipath propagation, ensuring reliable and efficient communication. Channel estimation is fundamental for achieving robust performance in OFDM systems, enabling them to adapt to changing channel conditions and deliver high-quality communication in diverse and challenging environments.

2.7.2 Pilot based channel estimation

Pilot signals play a pivotal role in the realm of Orthogonal Frequency Division Multiplexing (OFDM) systems, acting as indispensable reference points strategically embedded at predetermined intervals within the transmitted data.

These reference points, specifically allocated as pilot subcarriers in the OFDM scheme, serve as beacons for the system to discern and comprehend the dynamic characteristics of the channel at distinct time or frequency instances. Their strategic placement allows the OFDM system to adeptly navigate and adapt to the ever-changing conditions of the communication channel. By judiciously inserting these pilots, the system gains a heightened ability to gauge and respond to fluctuations in the channel environment, ensuring robust and reliable communication amidst dynamic and challenging scenarios.

The process of channel response extraction is pivotal in unlocking the full potential of pilot-based channel estimation. During the reception phase, the received pilot symbols undergo a meticulous comparison with their predefined transmitted values. This intricate comparative analysis serves as the key to unlocking valuable insights into the channel's response. By discerning the differences between the received and expected pilot symbols, the system can extract the nuanced alterations incurred during transmission, thus offering a snapshot of the prevailing channel conditions specifically at the pilot subcarriers.

To transcend the limited insights gained from these strategically placed pilot subcarriers and garner a comprehensive understanding of the entire communication channel, an interpolation step is often employed. This interpolation methodology effectively extends the acquired knowledge from the pilot subcarriers to estimate the channel characteristics across all subcarriers, contributing to a more thorough, accurate, and holistic channel estimation. In essence, this pilot-based approach not only fortifies the adaptability of OFDM systems but also enhances their reliability in navigating the complexities of real-world communication environments.

2.7.2.1 Block type pilots

All subcarriers send pilots at a specific time instant; this allows for tracking of a frequency selective channel; at a specific time, interval we have a specific subcarrier sending the pilots.

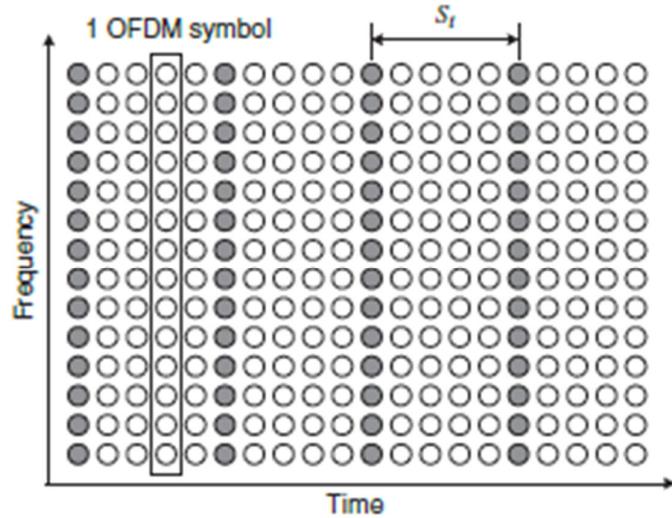


Figure 2-22: Block type pilots

2.7.2.2 Comb type pilots

A subcarrier is always mapped to a pilot to allow tracking of a fast-fading channel, we usually use the comb pilots in the 802.11 standard; some subcarriers are assigned to only send pilots at all time intervals.

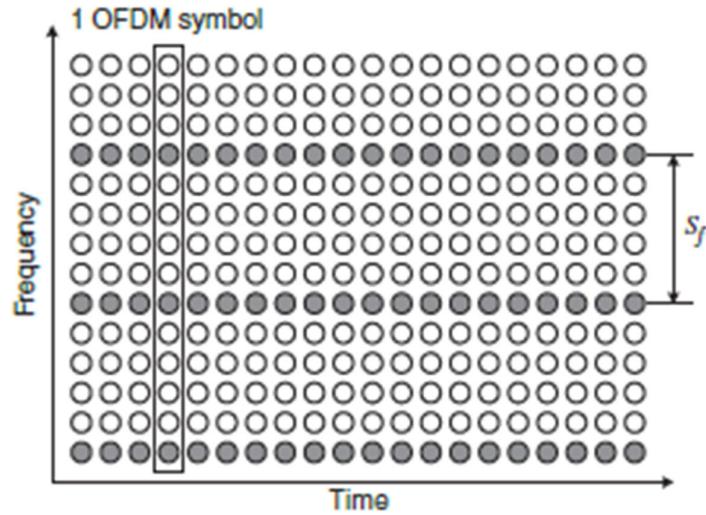


Figure 2-23: Comb type pilot

2.7.2.3 Lattice type pilot

Pilot tones are inserted along both the time and frequency axes with given periods to facilitate time/frequency-domain interpolations for channel estimation.

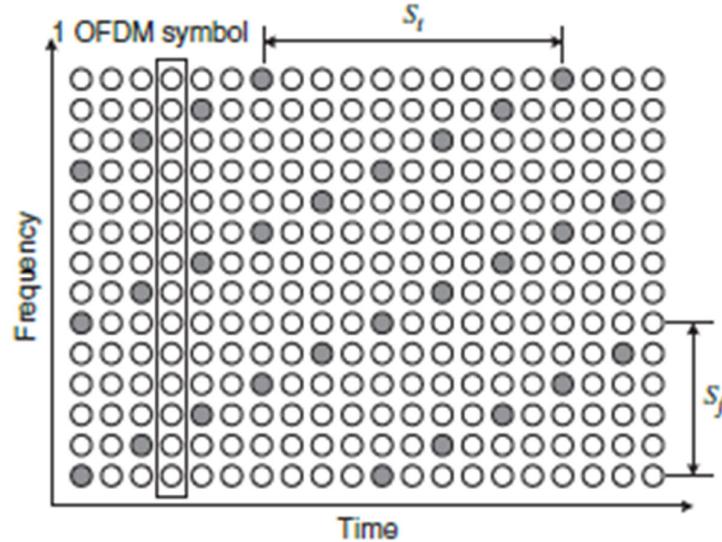


Figure 2-24: Lattice type pilots

2.7.3 Channel estimation techniques

2.7.3.1 Least Squares (LS)

The least-square (LS) channel estimation method finds the channel estimate in such a way that the following cost function is minimized:

$$\begin{aligned}
 J(\hat{\mathbf{H}}) &= \|\mathbf{Y} - \mathbf{X}\hat{\mathbf{H}}\|^2 \\
 &= (\mathbf{Y} - \mathbf{X}\hat{\mathbf{H}})^H(\mathbf{Y} - \mathbf{X}\hat{\mathbf{H}}) \\
 &= \mathbf{Y}^H \mathbf{Y} - \mathbf{Y}^H \mathbf{X}\hat{\mathbf{H}} - \hat{\mathbf{H}}^H \mathbf{X}^H \mathbf{Y} + \hat{\mathbf{H}}^H \mathbf{X}^H \mathbf{X}\hat{\mathbf{H}}
 \end{aligned} \tag{2.7-1}$$

The \mathbf{Y} term represents the received signal, \mathbf{X} is the sent data and \mathbf{H} is the channel response term that we wish to find from the channel estimate.

By setting the derivative of the function with respect to \mathbf{H} to zero:

$$\frac{\partial J(\hat{\mathbf{H}})}{\partial \hat{\mathbf{H}}} = -2\mathbf{X}^T \mathbf{Y} + 2\mathbf{X}^T \mathbf{X}\hat{\mathbf{H}} = 0 \tag{2.7-2}$$

$$\widehat{\mathbf{H}}_{LS} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} = \mathbf{X}^{-1} \mathbf{Y} \tag{2.7-3}$$

So, the least squares estimate is dividing the received signal by the transmitted signal, here we will have pilots at the receiver with predetermined values, we will divide it by the received pilot signals, here we will get the deviation between the reference and received signal, then we use interpolation techniques are employed to infer values between the known pilot locations. This ensures a smooth and continuous estimation of the channel characteristics across the entire frequency spectrum.

2.7.3.2 Minimum Mean Square Error (MMSE)

The Minimum Mean Square Error (MMSE) method is a powerful technique that seeks to minimize the mean square error between the estimated channel response and the true channel response. To adapt the MMSE estimate to a specific system, a regularization factor, often denoted by W , is introduced.

This regularization factor plays a crucial role in controlling the trade-off between fitting the estimate closely to the observed data and maintaining stability in the presence of noise. The MMSE channel estimation involves taking the Least Squares (LS) estimate and multiplying it by the regularization factor. This regularization term is added to the diagonal elements of the channel covariance matrix, providing a controlled amount of regularization to the estimate.

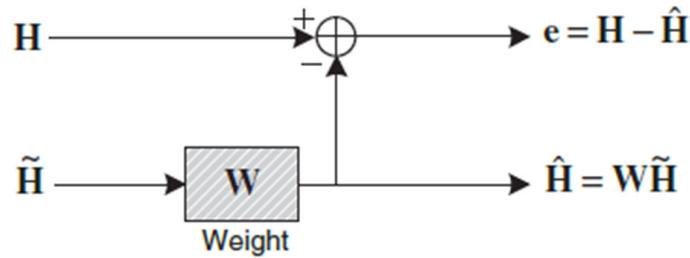


Figure 2-25: Weight factor in MMSE

The equation we want to minimize is:

$$J(\hat{\mathbf{H}}) = E\{\|\mathbf{e}\|^2\} = E\{\|\mathbf{H} - \hat{\mathbf{H}}\|^2\} \quad (2.7-4)$$

Thus, the MMSE channel estimation method finds a better (linear) estimate in terms of W in such a way that the MSE in Equation is minimized. The orthogonality principle states. that the estimation error vector is orthogonal to such that:

$$\begin{aligned}
 E\{\mathbf{e}\tilde{\mathbf{H}}^H\} &= E\{(\mathbf{H} - \hat{\mathbf{H}})\tilde{\mathbf{H}}^H\} \\
 &= E\{(\mathbf{H} - \mathbf{W}\tilde{\mathbf{H}})\tilde{\mathbf{H}}^H\} \\
 &= E\{\mathbf{H}\tilde{\mathbf{H}}^H\} - \mathbf{W}E\{\tilde{\mathbf{H}}\tilde{\mathbf{H}}^H\} \\
 &= \mathbf{R}_{\mathbf{HH}} - \mathbf{WR}_{\tilde{\mathbf{H}}\tilde{\mathbf{H}}} = \mathbf{0}
 \end{aligned} \quad (2.7-5)$$

Solving for W we get:

$$\mathbf{W} = \mathbf{R}_{\mathbf{HH}} \mathbf{R}_{\tilde{\mathbf{H}}\tilde{\mathbf{H}}}^{-1} \quad (2.7-6)$$

Solving for autocorrelation matrix we get:

$$\begin{aligned}
\mathbf{R}_{\tilde{\mathbf{H}}\tilde{\mathbf{H}}} &= E\{\tilde{\mathbf{H}}\tilde{\mathbf{H}}^H\} \\
&= E\{\mathbf{X}^{-1}\mathbf{Y}(\mathbf{X}^{-1}\mathbf{Y})^H\} \\
&= E\{(\mathbf{H} + \mathbf{X}^{-1}\mathbf{Z})(\mathbf{H} + \mathbf{X}^{-1}\mathbf{Z})^H\} \\
&= E\{\mathbf{H}\mathbf{H}^H + \mathbf{X}^{-1}\mathbf{Z}\mathbf{H}^H + \mathbf{H}\mathbf{Z}^H(\mathbf{X}^{-1})^H + \mathbf{X}^{-1}\mathbf{Z}\mathbf{Z}^H(\mathbf{X}^{-1})^H\} \\
&= E\{\mathbf{H}\mathbf{H}^H\} + E\{\mathbf{X}^{-1}\mathbf{Z}\mathbf{Z}^H(\mathbf{X}^{-1})^H\} \\
&= E\{\mathbf{H}\mathbf{H}^H\} + \frac{\sigma_z^2}{\sigma_x^2} \mathbf{I}
\end{aligned} \tag{2.7-7}$$

Now the MMSE estimate is:

$$\begin{aligned}
\hat{\mathbf{H}} &= \mathbf{W}\tilde{\mathbf{H}} = \mathbf{R}_{\mathbf{H}\tilde{\mathbf{H}}}\mathbf{R}_{\tilde{\mathbf{H}}\tilde{\mathbf{H}}}^{-1}\tilde{\mathbf{H}} \\
&= \mathbf{R}_{\mathbf{H}\tilde{\mathbf{H}}}\left(\mathbf{R}_{\mathbf{H}\mathbf{H}} + \frac{\sigma_z^2}{\sigma_x^2} \mathbf{I}\right)^{-1}\tilde{\mathbf{H}}
\end{aligned} \tag{2.7-8}$$

The auto and cross correlation terms are:

$$E\left\{h_{k,l}\tilde{h}_{k',l'}^*\right\} = E\left\{h_{k,l}h_{k',l'}^*\right\} = r_f[k-k']r_t[l-l'] \tag{2.7-9}$$

The frequency domain correlation:

$$r_f[k] = \frac{1}{1+j2\pi\tau_{rms}k\Delta f} \tag{2.7-10}$$

The time domain correlation:

$$r_t[l] = J_0(2\pi f_{\max} l T_{sym}) \tag{2.7-11}$$

The time domain correlation yields unity as it is a Bessel function as it an autocorrelation function between the symbol index and itself, it can be simplified from this graph:

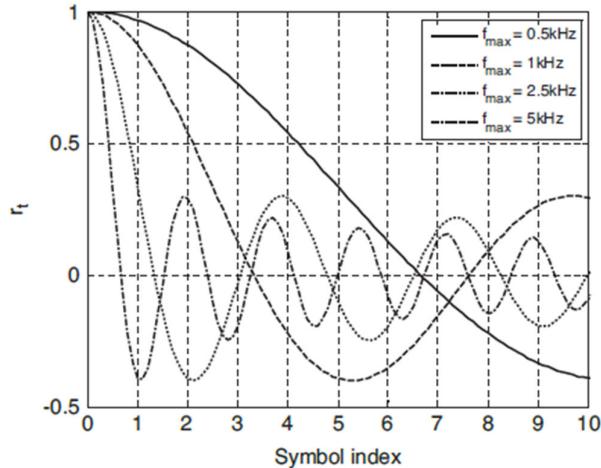


Figure 2-26: Time domain correlation characteristic

2.7.3.3 DFT based estimation

The DFT-based channel estimation method has been developed to enhance the efficacy of both Least Squares (LS) and Minimum Mean Square Error (MMSE) channel estimation by mitigating the influence of noise beyond the maximum channel delay. Leveraging the sparsity inherent in the channel response, the DFT transforms this response into the frequency domain, where only frequencies corresponding to substantial delays exhibit noteworthy energy.

This transformation by the DFT serves to segregate the frequency components associated with significant channel delays from those attributable to noise and interference. While noise outside the maximum channel delay in the time domain may disperse across multiple frequencies, the frequency domain allows for a clearer distinction and isolation of the pertinent components.

The DFT-based approach facilitates the suppression of noise components unrelated to significant channel delays. By concentrating on the pertinent frequency components, the impact of noise beyond the maximum channel delay is mitigated, resulting in a more precise and accurate channel estimation.

2.8 CODED OFDM IN 802.11 STANDARDS

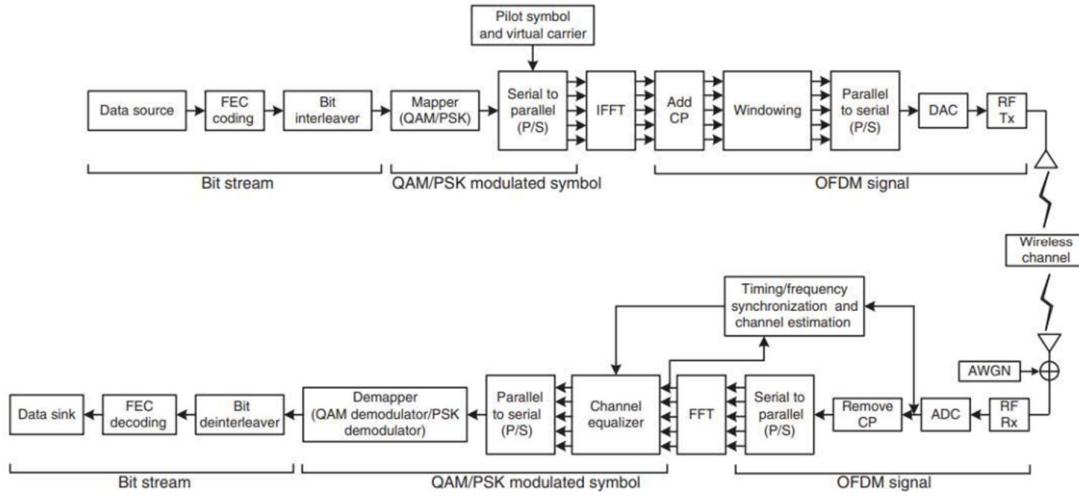
This chapter is adapted from reference [7]

Orthogonal Frequency Division Multiplexing (OFDM) is a key technology used in the physical layer (PHY layer) of various wireless communication standards, including the 802.11 series, which defines the specifications for wireless local area networking (WLAN). OFDM is employed in several 802.11 standards to enable high data rates and improve robustness in wireless communication

The OFDM system implemented is the PHY layer of a communication system it takes a Physical Service Data Unit from the MAC layer adds headers and modulates the packet to be transmitted so our system mainly serves 2 purposes:

1. Mapping the PSDU to PPDU
2. Transmitting and receive through a Wireless Medium (WM) between 2 stations (STAs) using OFDM

The PSDU is a packet received from higher layers in the system, it consists of the data we want to send over the network, this data can be a text message, in the coming sections we will go through the whole flow of how the system maps the text message to an OFDM frame and send it through the network.



2.8-1: Block Diagram of OFDM in WIFI

2.8.1 Frame Generation in 802.11 PHY layer

We first start with a PSDU packet from a higher layer that we tend to send over the OFDM system, we add headers, pilots and preamble to the data then we send it as bursts through the channel to the receiver to demodulate and detect the sent data.

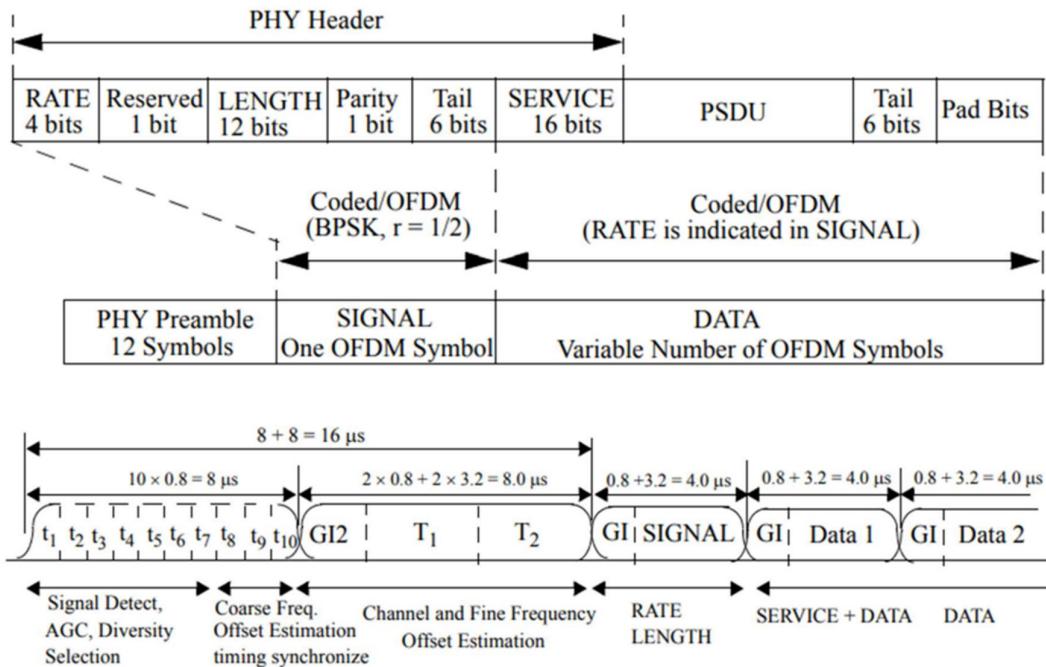


Figure 2-27: Generated OFDM frame

For different modulation schemes there are different parameters regarding the data rate, coding rate and coding bits. This was standardized from the IEEE 802.11 WIFI standard by the table below:

Table 2-6: Modulation scheme parameters

Modulation	Coding rate (R)	Coded bits per subcarrier (N_{BPSC})	Coded bits per OFDM symbol (N_{CBPS})	Data bits per OFDM symbol (N_{DBPS})	Data rate (Mb/s) (20 MHz channel spacing)	Data rate (Mb/s) (10 MHz channel spacing)	Data rate (Mb/s) (5 MHz channel spacing)
BPSK	1/2	1	48	24	6	3	1.5
BPSK	3/4	1	48	36	9	4.5	2.25
QPSK	1/2	2	96	48	12	6	3
QPSK	3/4	2	96	72	18	9	4.5
16-QAM	1/2	4	192	96	24	12	6
16-QAM	3/4	4	192	144	36	18	9
64-QAM	2/3	6	288	192	48	24	12
64-QAM	3/4	6	288	216	54	27	13.5

The Frame Generation is as follows:

- i. Generating the short training sequence section of the preamble
- ii. Generating the long preamble sequence section of the preamble
- iii. Scrambling and zeroing the tail bits
- iv. Encoding the DATA with a convolutional encoder and puncturing
- v. Data interleaving
- vi. Pilot insertion
- vii. Mapping into complex 16-QAM symbols
- viii. Transforming from frequency to time and adding a circular prefix
- ix. Windowing
- x. Concatenating the OFDM symbols into a single, time-domain signal

The preamble generation is mentioned separately in section 2.9 as it is related to synchronization, in this chapter we will only deal with the PSDU only

The system is defined by the coding rate, number of used subcarriers and the modulation order used the data bits to be sent is determined from the upper layers, so the PSDU size is fixed, we need to calculate the number of symbols needed to transmit this frame, this can be done using the calculations below to add PAD bits for an integer number of symbols

So, for an M-order modulation, coding rate and a number of used subcarriers (N_{used}):

```
k = log2(M)
Nbpsc = k
Ncbps = Nused * k
Ndbps = Ncbps * coding_rate
Nsym = int(ceil(data_no_pad_size/Ndbps))           #number of symbols in a frame
Ndata = Nsym * Ndbps                                #data size after padding
Npad = Ndata - data_no_pad_size                      #number of pad bits
```

The system we have will have a coding rate of 0.5, number of used subcarriers of 64, 48 used, 4 pilot, 12 guard bands, for the modulation order, we will run SER simulations for different modulations schemes with modulation orders in order to test the whole system performance.

2.8.1.1 Scrambling and Descrambling

Scrambling and descrambling are fundamental techniques used in various fields like telecommunications, data storage, and security. They are used to modify data streams in a way that makes them unintelligible without a special key or algorithm. This section aims to explain the concepts of scrambling and descrambling, their applications, and the different techniques used to implement them.

What is Scrambling?

Scrambling, also known as randomizing, is the process of manipulating a data stream before transmission or storage. This manipulation can involve various techniques like:

Bit inversion: Flipping individual bits in the data stream.
Byte transposition: Swapping the positions of bytes within the data stream.
Substitution: Replacing specific data patterns with other patterns.
XOR encryption: Applying bitwise XOR operation with a secret key to the data stream.

The main objective of scrambling is to achieve the following:

- Improve synchronization: Scrambling helps to prevent long sequences of identical bits, which can disrupt synchronization in communication systems.
- Reduce DC component: Scrambling helps to remove the DC bias from the data stream, which can be detrimental to certain transmission channels.
- Enhance security: Scrambling can provide a basic level of security by making the data unintelligible without prior knowledge of the scrambling algorithm.

What is Descrambling?

Descrambling is the inverse process of scrambling. It involves applying the same algorithm used for scrambling, but in reverse order, to recover the original data. The descrambler must possess the same key or secret information as the scrambler to perform this operation successfully.

Applications of Scrambling and Descrambling

Scrambling and descrambling find applications in various domains, including:

- Telecommunications: Scrambling is used in satellite, radio, and PSTN communication systems to improve synchronization and reduce DC component.
- Data storage: Scrambling can be used to protect sensitive data stored on magnetic or optical media.

- Digital audio and video: Scrambling is employed in cable and satellite TV to restrict access to premium content.
- Wireless networks: Scrambling is used in Wi-Fi and cellular networks to enhance security and privacy
- Techniques for Scrambling and Descrambling
- Different techniques can be used for scrambling and descrambling data, depending on the specific requirements of the application. Some common techniques include:
- Linear feedback shift registers (LFSRs): These are simple and efficient circuits that generate pseudorandom sequences used for scrambling.
- Block ciphers: These are complex algorithms that encrypt data in blocks using a secret key. Scrambling can be achieved by applying a block cipher in a specific mode.
- Stream ciphers: These are algorithms that encrypt data one bit at a time using a secret key. Stream ciphers are particularly suitable for real-time applications like voice and video communication.

Scrambling and descrambling in 802.11

A simple implementation of the Scrambler consists of 7 shift registers and 2 XORs as shown in Figure 5. The Scrambler is of length-127, meaning it repeatedly generates a 127-bit sequence for a given pseudo-random initial state[5]. Each incoming data bit is XORed with the current bit in the 127-bit sequence. PPDU synchronous scrambler uses the generator polynomial $S(x)$ as follows[8]:

$$S(x) = x^7 + x^4 + 1$$

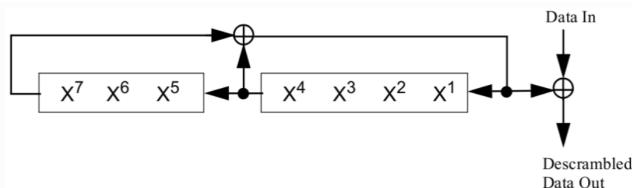


Figure 2-28: scrambler diagram

2.8.1.2 Interleaver & Deinterleaver

In wireless communication systems, errors can occur due to various factors such as channel noise, fading, and interference. These errors can corrupt the transmitted data, making it impossible for the receiver to decode the message correctly. Interleaving and deinterleaving are two techniques that are used to combat these errors and improve the performance of wireless communication systems.

Types of errors

To understand the functions of an interleaver/deinterleaver, understanding of error characteristics is essential. There are two types of errors that are our main concern, they are burst error and random error.

RANDOM ERRORS:

Random errors, also known as independent errors or channel memoryless errors, are unpredictable and uncorrelated occurrences that can corrupt or interfere with the transmission of data in wireless communication systems. These errors occur independently of each other, meaning that the occurrence of an error at one location does not affect the probability of an error occurring at another location. This characteristic distinguishes random errors from burst errors, which tend to occur in clusters within a short time interval[9].

BURST ERRORS:

The term burst error means that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1. The figure shows the effect of a burst error on a data unit. In this case, 0100010001000011 was sent, but 0101110101100011 was received. A burst error does not necessarily mean that the errors occur in consecutive bits. The length of the burst is measured from the first corrupted bit to the last corrupted bit. Some bits in between may not have been corrupted.

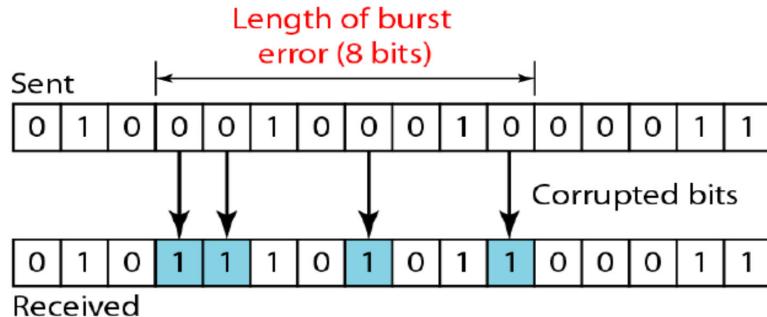


Figure 2-29: burst error illustration

Figure 2 shows the complex spectrum of a fading multipath channel, in a specific communication system, the fading, or weakening of a signal, can be quite severe. This fading can lead to the loss of an entire carrier signal. When a carrier is lost, damaged, or eliminated, the bits of data associated with that carrier are corrupted and considered errors. If these corrupted bits happen to be adjacent to each other, they form a burst error.

The purpose of an interleaver is to spread out adjacent bits across different carriers. This way, if a carrier is lost or corrupted, the affected bits are scattered, and the resulting errors are more likely to be random rather than burst. This makes it easier for the decoder to correct the errors.

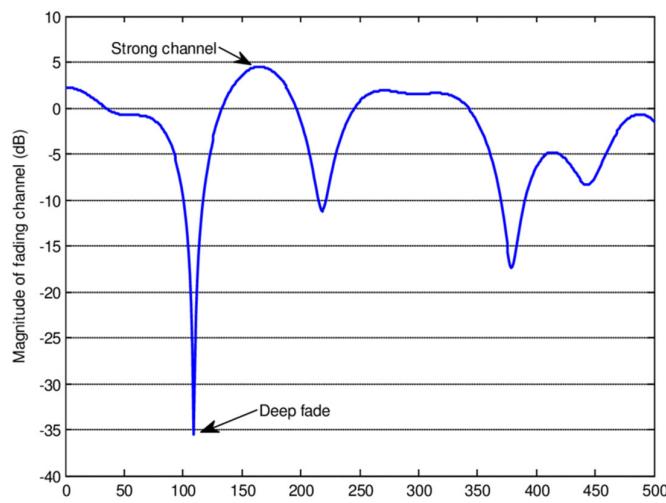


Figure 2-30: Deep fading

Interleaving

In digital communication systems, burst errors occur when a group of consecutive bits are corrupted due to factors such as fading, interference, or noise. These errors can significantly impact the integrity of the transmitted data and hinder the performance of the communication system. To mitigate the effects of burst errors, interleaving and deinterleaving techniques are employed.

Interleaving is a technique that intentionally spreads out adjacent bits across different carriers or time slots. This randomization process disrupts the burst error pattern, making it appear random to the decoder. By converting burst errors into random errors, interleaving allows the decoder to utilize its error correction capabilities more effectively[10].

Types of Interleaver

Two types of interleaves are commonly used:

- block interleaver
- convolutional interleaves.

BLOCK INTERLEAVER:

Block interleaver operate on a block of data, typically consisting of multiple codewords. The interleaver is filled row by row with each codeword, and the data is transmitted column by column. This process ensures that adjacent bits from the same codeword are spread out across different carriers or time slots. At the receiver, the data is interleaved before being decoded.

CONVOLUTIONAL INTERLEAVER:

Convolutional interleaver process data in a continuous manner, eliminating the need for block buffering. They achieve this by maintaining a memory structure that stores the data as it is received. The interleaver output is generated by selecting data from the memory structure based on a specific pattern. Convolutional interleaver offer reduced memory requirements and eliminate the delay associated with block interleaver filling.

Advantages of Interleaving

Effective burst error correction: Interleaving effectively converts burst errors into random errors, improving the decoder's ability to correct errors.

Reduced decoder complexity: Interleaving allows the use of simpler decoders designed for random errors, reducing the complexity and power consumption of the receiver.

Flexibility: Interleaving can be applied to various modulation schemes and channel conditions.

Disadvantages of Interleaving

Delay: Interleaving introduces a delay in the transmission of data, which may be unacceptable for real-time applications.

Reduced performance for burst-error correcting codes: Interleaving can mask the structure of burst errors, potentially reducing the effectiveness of burst-error correcting codes.

Illustration of an Interleaver (block interleaver)

In digital communications, a block interleaver is often employed. Assume that the N is input bits to the interleaver after error correction coding. These N bits are rearranged and supplied in a predetermined order.

Consider a N = 14-bit block for demonstration purposes. The interleaver output bit index is bit inverted from the input bit index. The sequence of the input and output bit indexes is shown in figure 3. It can be seen that the interleaver output bit index is no longer continuous.

Deinterleaving is simply another bit reversal to get back to the original bit stream index.

Decimal Input Bit Index	Binary Input Bit Index	Decimal Output Bit Index	Binary Output Bit Index
0	0000	0	0000
1	0001	8	1000
2	0010	4	0100
3	0011	12	1100
4	0100	2	0010
5	0101	10	1010
6	0110	6	0110
7	0111	14	1110
8	1000	1	0001
9	1001	9	1001
10	1010	5	0101
11	1011	13	1101
12	1100	3	0011
13	1101	11	1011

Figure 2-31: Interleaving and bit index

Interleaver Used in the IEEE 802.11a

In the IEEE 802.11a, the interleaver is defined by a two-step permutation. The first permutation converts the input bit index k to index i according to the following formula:

$$i = \frac{N_{cbps}}{16} (k \bmod 16) + \text{int} \frac{k}{16} \quad (2.8-2)$$

where N_{cbps} is the number of coded bits per symbol, int denotes the integral part, and \bmod is the remainder after the division.

The second permutation converts the index i to index j according to the following formula:

$$J = n \left(\text{int} \left(\frac{i}{n} \right) \right) + \left(i + N_{cbps} - \text{int} \left(\frac{16i}{N_{cbps}} \right) \right) \bmod(n) \quad (2.8-3)$$

where n depends upon the number of bits per subcarrier, N_{bpsc} , and is given here:

$$n = \max \left(\frac{N_{bpsc}}{2}, 1 \right) \quad (2.8-4)$$

For illustration purposes, consider BPSK used in the IEEE 802.11a, each OFDM symbol has 24 data bits and 48 coded bits after 1/2 convolutional coding[11].

Deinterleaver Used in the IEEE 802.11a

The deinterleaver in IEEE 802.11a is the inverse process of the interleaver described in the previous Section. It's done after two permutations[8]. The first permutation, which changes index j to index i , is given by the equation:

$$i = n \left(\text{int} \left(\frac{j}{n} \right) \right) + \left(j + \text{int} \left(\frac{16j}{N_{cbps}} \right) \right) \bmod(n) \quad (2.8-5)$$

The second permutation which converts the index i back to the original index k and is given by the following equation:

$$k = 16i - (N_{cbps} - 1) \text{int} \left(\frac{16i}{N_{cbps}} \right) \quad (2.8-6)$$

For the same example defined above, the interleaver shown converts index $k = 1$ to index $i = 16$. In the deinterleaver, the reverse process converts the index $I = 16$ back to index $k = 1$.

2.8.1.3 Forward Error Encoding and decoding

Forward Error Correction (FEC) is a technique used in communication systems to enhance the reliability of data transmission by adding redundancy to the transmitted data. The primary purpose of forward error coding is to detect and correct errors that may occur during data transmission.

The system employs an encoding technique derived from the IEEE 802.11 standard. This technique utilizes a convolutional encoder with a code rate of 0.5, implying the addition of one redundant bit for each data bit.

The generator polynomials of the encoder are:

- $S_1 = 1 + x^2 + x^3 + x^5 + x^6$
- $S_2 = 1 + x^1 + x^2 + x^3 + x^6$

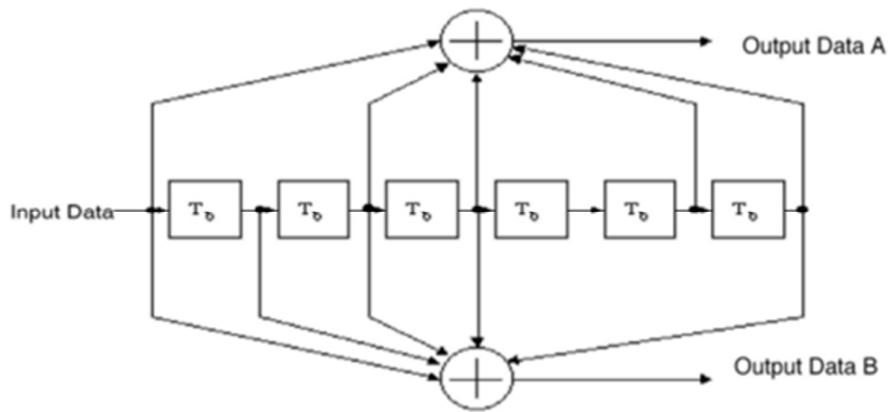


Figure 2-32: Block diagram of the convolutional encoder

The decoding process is intricately tied to the encoder type in use. For convolutional encoding, a Viterbi decoder is employed. The Viterbi decoder operates on a maximum likelihood estimation of the given codeword, mapping it to the most likely transmitted message based on error analysis. This decoder proves effective in correcting random errors. Even when punctuating (omitting) certain bits from the codeword and replacing them with zeros,

the Viterbi decoder can still provide a reliable estimate of the message (details on puncturing will be discussed later). In the presence of random errors, the Viterbi decoder can consistently recover the correct data stream.

However, if the system encounters deep fades resulting in a burst of errors bit by bit, the Viterbi decoder's performance diminishes. In such cases, an alternative encoding and decoding algorithm, like Reed-Solomon, may be necessary. Reed-Solomon is adept at handling burst errors, but it operates as a block-type encoding algorithm, introducing potential overhead to our system due to its non-instantaneous (bit by bit) nature, unlike the convolutional encoder. To address this challenge, the introduction of an interleaver is required. The convolutional encoder serves to rectify random errors induced by channel effects, particularly signal fading in the time domain, including both random and burst fading. While the convolutional encoder and decoder effectively manage random errors, they are less suited to handling burst errors, a gap filled by linear block codes. This limitation underscores the necessity of an interleaver to minimize the impact of burst errors, a topic that will be explored further in subsequent sections.

2.8.1.4 Puncturing

The standard offers a specific rate for each modulation scheme, the rate is defined in the Code Rate column in the table we mentioned before. After the convolutional encoder, we have a coding rate of $\frac{1}{2}$ the standard states that we must have a $\frac{2}{3}$ coding rate for a 64-QAM for example, so we will need to remove some redundancy bits from the codeword randomly (some will be data bits too) which will increase our coding rate, the following visualization explains more what we want to do:

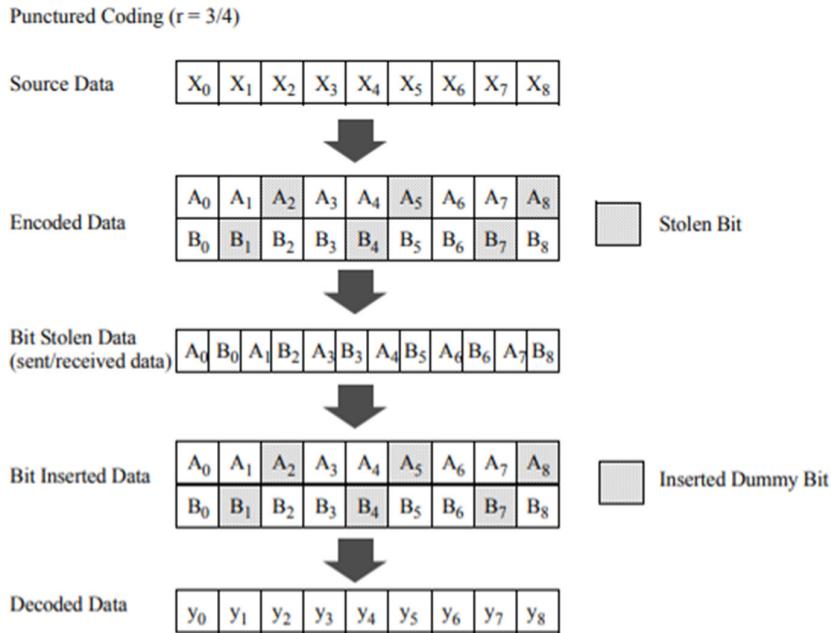


Figure 2-33: Puncturing

The following example explains how the coding rate is increased:

Table 2-7: Modulation specific parameters

Modulation	Coding rate (R)	Coded bits per subcarrier (N_{BPSC})	Coded bits per OFDM symbol (N_{CBPS})	Data bits per OFDM symbol (N_{DBPS})	Data rate (Mb/s) (20 MHz channel spacing)	Data rate (Mb/s) (10 MHz channel spacing)	Data rate (Mb/s) (5 MHz channel spacing)
64-QAM	3/4	6	288	216	54	27	13.5

In a typical OFDM system using 802.11 we have 48 subcarriers carrying the data so in total the coded bit per symbol is: $48 * 6 = 288$

After the encoder we have half of these bits representing the DATA bits, which will be 144 bits

After puncturing a whole data stream, we remove some redundancy bits and some data then we take again 288 bits and map them to an OFDM symbol, this on average gets us to 216 Data bits per OFDM symbol which is an increase to the data rate and the code rate too:

$$r_{old} = \frac{144}{288} = \frac{1}{2} \quad R_{b,old} = \frac{144}{4us} = 36 Mbps$$

$$r_{new} = \frac{216}{288} = \frac{3}{4} \quad R_{b,new} = \frac{216}{4us} = 54 Mbps$$

Where $4us$ is the time of the OFDM symbol

2.8.1.5 Pilot Insertion

We need not to place the same pilots for all the symbols, instead we want to send a variety of pilots to allow for a better estimation, we can achieve this by using the scrambler and a predefined pilot sequence. The pilots are real valued signals inserted to the OFDM symbol, we have 4 pilots

The contribution of the pilot subcarriers for the n^{th} OFDM symbol is produced by inverse Fourier transform of sequence P , given by

Initiating the scrambling with a seed of $(1111111)_2$ gets us the pseudorandom sequence for the pilot arrangement with 0s and 1s, replacing 1s with -1 and 0s with 1 we get the following sequence:

$$p_{0..126v} = \{1,1,1,1, -1,-1,-1,1, -1,-1,-1,-1, 1,1,-1,1, -1,-1,1,1, -1,1,1,-1, 1,1,1,1, 1,1,-1,1, \\ 1,1,-1,1, 1,-1,-1,1, 1,1,-1,1, -1,-1,-1,1, -1,1,-1,-1, 1,-1,-1,1, 1,1,1,1, -1,-1,1,1, \\ -1,-1,1,-1, 1,-1,1,1, -1,-1,-1,1, 1,-1,-1,-1, -1,1,-1,-1, 1,-1,1,1, 1,1,-1,1, -1,1,-1,1, \\ -1,-1,-1,-1, -1,1,-1,1, 1,-1,1,-1, 1,1,1,-1, -1,1,-1,-1, -1,1,1,1, -1,-1,-1,-1, -1,-1,-1\}$$

Where the element P_0 multiplies the SIGNAL symbol, P_1 multiplied the first DATA symbol and so on, the preamble symbols do not need pilots as they act as pilots themselves and are used for CFO and STO

2.8.1.6 Modulation Mapping

Now we are left with a stream of bits, we need to allocate the bits to symbols and each symbol to be transmitted by subcarriers, depending on the modulation scheme used, we group the bits and form constellations (6 bits for 64-QAM, 4 bits for QPSK etc.), the constellations are then multiplied by normalization factors to ensure the same average power to all mappings; maintain a uniform power level across different symbols to ensure efficiency of the comm channel. This normalization makes sure that each signal contributes equally to the total power of the transmitted signal:

Modulation	K_{MOD}
BPSK	1
QPSK	$1/\sqrt{2}$
16-QAM	$1/\sqrt{10}$
64-QAM	$1/\sqrt{42}$

Table 2-8: Normalization factors

We will be using the MODEM described in section 2.8.1.7 and 2.8.1.8 to perform the modulation and demodulation

2.8.1.7 Modulation

We used 3 modulation schemes: M-PAM, M-PSK and M-QAM for each we needed to make a constellation grid to map each received symbol to the respective I-Q components

M-PSK

The In-phase component of the M-PSK

$$I = \frac{1}{\sqrt{2}} \cos\left(2\pi \frac{m}{M}\right) \quad (2.8-7)$$

The Quadrature component of the M-PSK

$$Q = \frac{1}{\sqrt{2}} \sin\left(2\pi \frac{m}{M}\right) \quad (2.8-8)$$

m is symbol value and *M* is the modulation order

M-PAM

The In-phase component of the M-PAM

$$I = 2m - M + 1 \quad (2.8-9)$$

m is symbol value and M is the modulation order

M-QAM

The In-phase component of the M-QAM

$$I = \frac{1}{\sqrt{M}} \cos \left(2\pi \frac{m}{M} \right) \quad (2.8-10)$$

The Quadrature component of the M-QAM

$$Q = \frac{1}{\sqrt{M}} \sin \left(2\pi \frac{m}{M} \right) \quad (2.8-11)$$

m is symbol value and M is the modulation order

2.8.1.8 Demodulation

Demodulator is done using an IQ detector, the detector computes the pair-wise Euclidean distance of each point in the received vector against every point in the reference constellation. It then returns the symbols from the reference constellation that provide the minimum Euclidean distance.

The equation to calculate the Euclidean distance is

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.8-12)$$

2.8.1.9 Frequency Domain to Time Domain (FFT)

The 6-stage radix-2 64-point FFT (Fast Fourier Transform) is a sophisticated algorithm designed to efficiently compute the discrete Fourier transform of a sequence with 64 data points. Employing a radix-2 algorithm means that the transformation process is recursively divided into stages, with each stage handling a factor of 2 points. In the context of a 64-point FFT, six stages are involved.

At each stage, the algorithm performs various operations, including butterfly operations and twiddle factor multiplications. The butterflies, which involve complex number additions and subtractions, are fundamental to the FFT algorithm and contribute to its efficiency. Twiddle factors introduce phase shifts and scale the intermediate results.

The radix-2 64-point FFT is particularly advantageous for power-of-two-sized data sets, offering a balance between computational complexity and efficiency. The algorithm leverages the inherent symmetry and periodicity properties of the FFT to optimize computations and reduce redundancy.

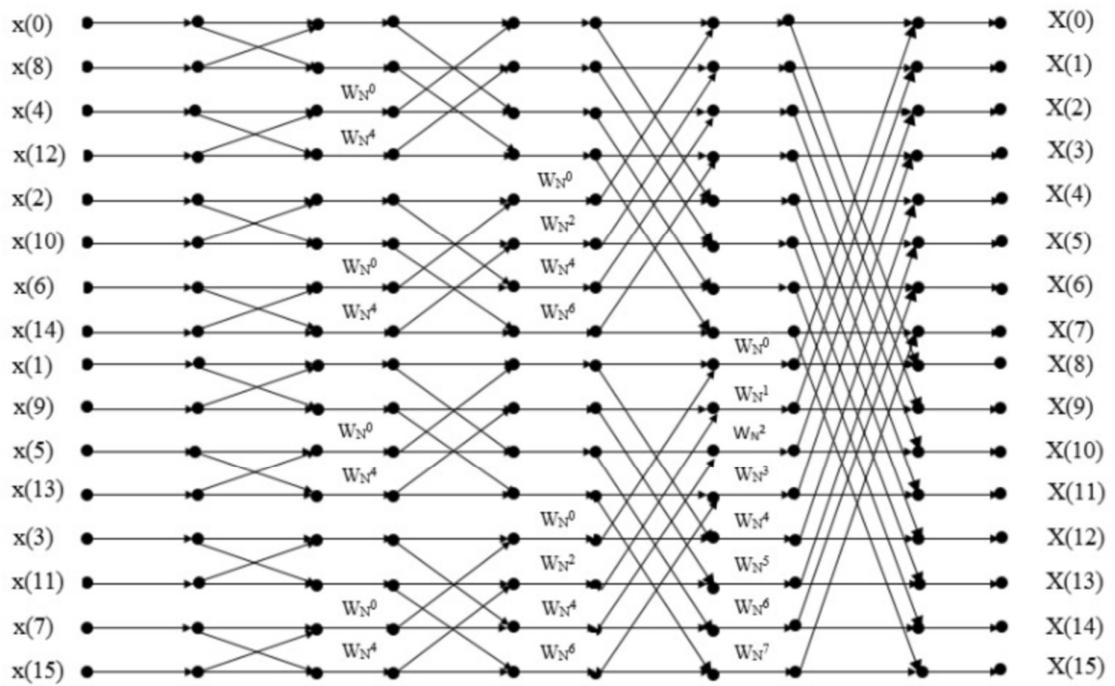


Figure 2-34 FFT radix 2 butterfly diagrams

2.8.1.10 Windowing

The time domain signal can be seen as an inverse Fourier transform multiplied by a window function in the same period, the windowing function is used to smooth the transition order to reduce the spectral sidelobes of the transmitted waveform.

After the parallel to serial conversion, we concatenate all the time domain signals after each other to form an OFDM burst, this burst is then raised on a carrier and sent through the antenna, the sudden shift in phase between each time domain signal causes spectral leakage, this can be encountered by windowing in time domain, where we insert a cyclic suffix of an OFDM symbol, and an overlap it took from the following OFDM signal cyclic prefix multiply each by windowing functions and the corresponding indices, below is a visual representation:

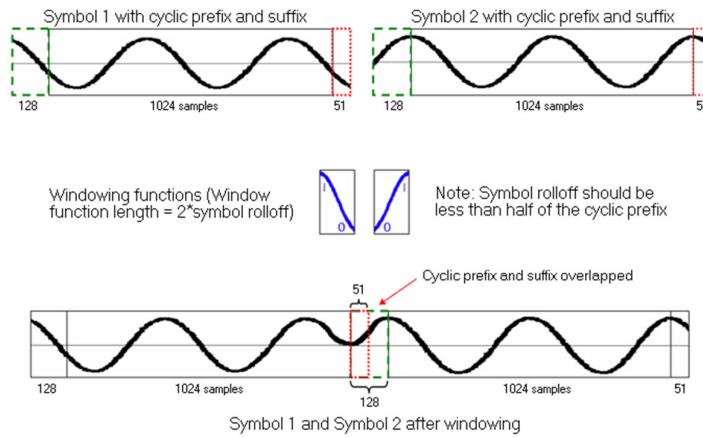


Figure 2-35 Windowing functions

The windowing function used is the Hann Windowing function:

$$w(n) = 0.5 \left(1 + \cos \left(\frac{\pi n}{N-1} \right) \right) \quad (2.8-13)$$

2.9 SYNCHRONIZATION

This Section is adapted from [9] Chapter 5.

2.9.1 Effect of Symbol Time Offset

In order to take the N-point FFT in the receiver, we need the exact samples of the transmitted signal for the OFDM symbol duration. In other words, a symbol-timing synchronization must be performed to detect the starting point of each OFDM symbol (with the CP removed), which facilitates obtaining the exact samples.

The Symbol Time Offset “STO” of δ in time domain incurs the phase offset of $2\pi k \delta / N$ in the frequency domain, which is proportional to subcarrier index k and STO δ .

Addition of linear phase in frequency domain indicates that the symbol still keeps its shape, but it is phase shifted, this is shown in Table 2.9-1.

Table 2.9-1: Effect of STO in Time and Frequency domains

	Received signal	STO (δ)
Time domain	$y[n]$	$x[n + \delta]$
Frequency domain	$Y[k]$	$e^{j2\pi k \delta / N} X[k]$

There are 4 cases for STO:

1. There is no STO and the DFT frame is perfectly synchronized, this is the ideal case.
2. The estimated starting point is before the exact point and after the delayed channel response, the symbol is not overlapped with the previous symbol no ISI occurs here.
But the symbol is considered phase shifted: $Y_l[k] = X_l[k]e^{j2\pi k \delta / N}$
3. The starting point of the OFDM symbol is estimated to exist prior to the end of the (lagged) channel response to the previous OFDM symbol, and thus, the symbol timing is too early to avoid the ISI. In this case, the orthogonality among subcarrier components is destroyed by the ISI (from the previous symbol) and furthermore, ICI (Inter-Channel Interference) occurs.
4. This is the case when the estimated starting point of the OFDM symbol is after the exact point, which means the symbol timing is a little later than the exact one. In this case, the signal within the FFT interval consists of a part of the current OFDM symbol, ISI occurs and also ICI. Orthogonality is not maintained.

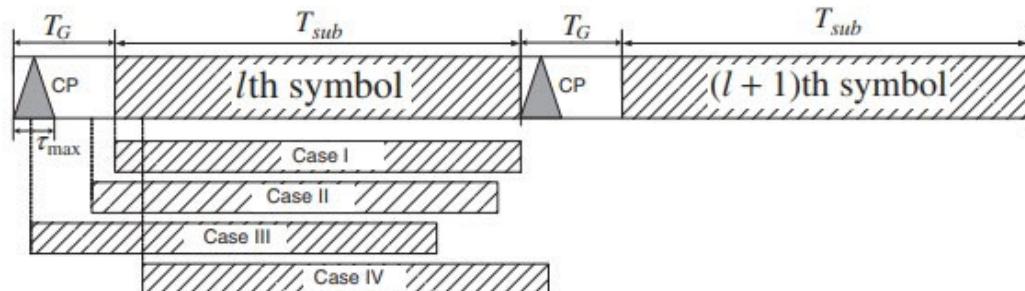


Figure 2-36: Four Cases of Symbol Time Offset in OFDM [9]

The constellation diagram for case 1 and 2 are shown in Figure 2-37

The constellation diagram for case 3 and 4 are shown in Figure 2-38.

Case 4 constellation is severe, this is why synchronization is crucial to prevent such case.

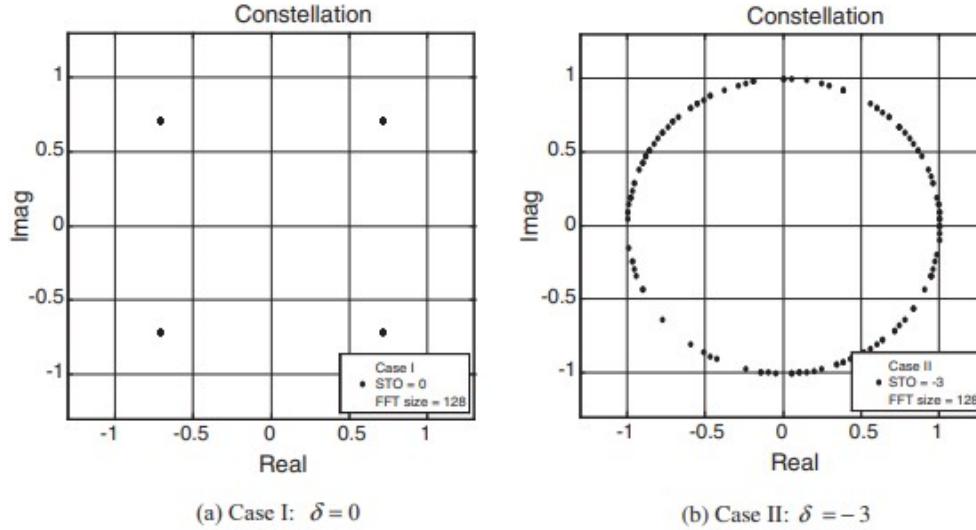


Figure 2-37: Constellation diagram for case 1 and 2 of STO [9]

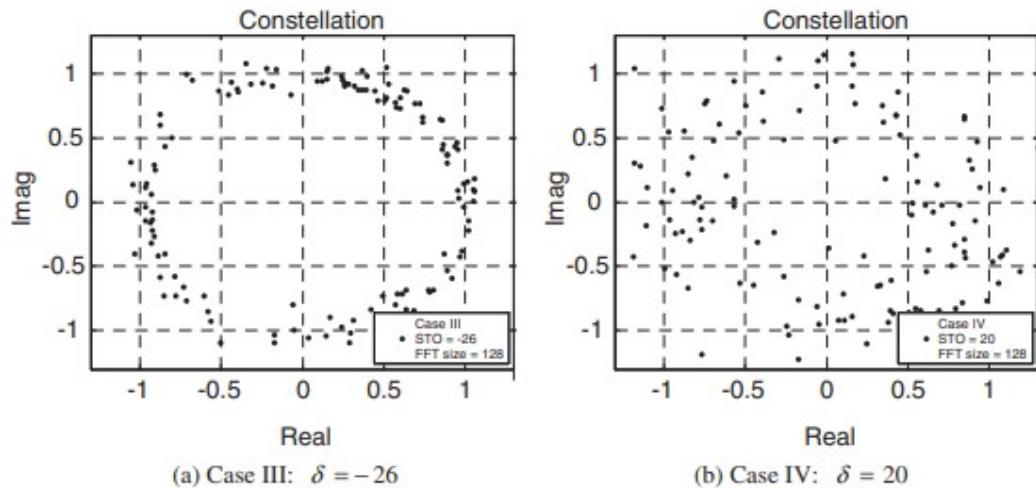


Figure 2-38: Constellation diagram for case 3 and 4 of STO [9]

2.9.2 Estimation techniques for STO

- Not only STO may cause phase distortion (can be compensated by using equalizer), but also ISI and ICI that cannot be corrected if occurred.
- The starting point of OFDM symbol must be accurately determined by estimating the STO with synchronization techniques at the receiver.
- STO estimation can be done in frequency or time domain.
- In time domain, STO can be estimated by using CP or training symbols.

2.9.2.1 Estimation in time domain using CP

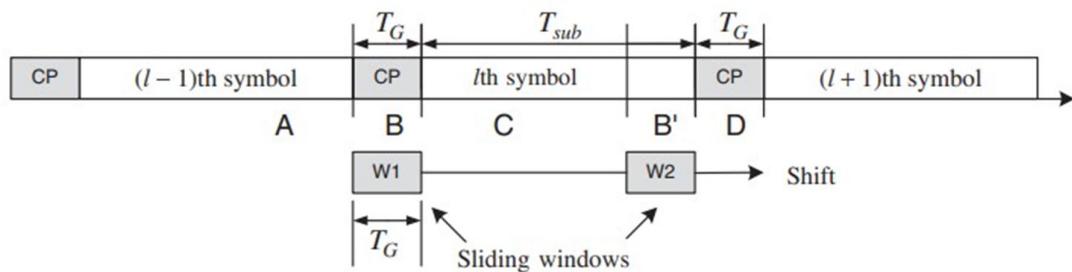


Figure 2-39: STO Estimation in Time domain using CP Sliding windows [9]

Here, there are 2 sliding windows of length N_{guard} and with spacing N (which is the actual length of OFDM symbol without CP) between them. The windows are sliding and picking the values and apply a certain estimation technique to these values.

Estimation with correlation

One of these techniques is the correlation function:

$$\hat{\delta} = \arg \max_{\delta} \left\{ \sum_{i=\delta}^{N_G-1+\delta} |y[n+i] \cdot y^*[n+N+i]| \right\} \quad (2.9-1)$$

The sweep of the variable δ is made and the δ that achieves the highest value of correlation is the estimate.

2.9.2.2 Squared Difference Method

There is the squared difference technique that is more robust than the correlation in case of existence of CFO.

$$\hat{\delta} = \arg \min_{\delta} \left\{ \sum_{i=\delta}^{N_G-1+\delta} (|y[n+i]| - |y^*[n+N+i]|)^2 \right\} \quad (2.9-2)$$

2.9.2.3 Estimation in time domain using Training Symbol

Training symbols can be transmitted to be used for symbol synchronization in the receiver. In contrast with CP, it involves overhead for transmitting training symbols, but it does not suffer from effect of multi-path channel.

Two identical OFDM training symbols, or a single OFDM symbol with a repetitive structure can be used.

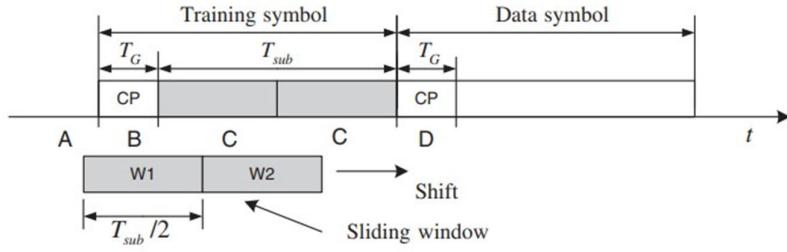


Figure 2-40: STO Estimation using training symbol [9]

2.9.3 Effect of Carrier frequency Offset

Effects on carrier frequency:

The baseband transmit signal is converted up to the passband by a carrier modulation and then, converted down to the baseband by using a local carrier signal of (hopefully) the same carrier frequency at the receiver.
In general, there are three phenomena affecting the carrier frequency:

- Phase noise, which can be modelled as a zero-mean Wiener random process.
- Difference in central frequency of oscillation between transmitter and receiver. (unavoidable).
- Doppler Frequency Shift which depends on difference in velocities of transmitter and receiver.

We won't consider the effect of the Phase noise in the section of CFO.

2.9.3.1 Frequency Offset Definitions

Carrier Frequency Offset

Let f_c and f'_c denote the carrier frequencies at transmitter and receiver respectively.

So the difference is the frequency offset defined as:

$$f_{\text{offset}} = f_c - f'_c \quad (2.9-3)$$

while include the doppler frequency in the f'_c , doppler frequency is:

$$f_d = \frac{v \cdot f_c}{c} \quad (2.9-4)$$

v is the frequency of the receiver, c is speed of light.

Normalized CFO

Let us define the normalized CFO, it the ratio between frequency offset and the subcarrier spacing

$$\epsilon = \frac{f_{\text{offset}}}{\Delta f} \quad (2.9-5)$$

Let epsilon consist of 2 parts, an integer part (ϵ_i) and fractional part (ϵ_f)

$$\epsilon = \epsilon_i + \epsilon_f \quad (2.9-6)$$

For the time-domain signal $x[n]$, a CFO of ϵ causes a phase offset of: $2\pi n\epsilon$

That is, proportional to the CFO ϵ and time index n . It is equivalent to a frequency shift of $-\epsilon$ on the frequency-domain signal $X[k]$.

The effect of CFO on the received signal is shown in Table 2.9-2.

Table 2.9-2: Effect of CFO in Time and Frequency domains [9]

	Received signal	Effect of CFO ϵ on received signal
Time domain	$y[n]$	$e^{(j2\pi\epsilon/N)} x[n]$
Frequency domain	$Y[k]$	$X[k - \epsilon]$

Figure 2-41 shows that the frequency shift of $-\epsilon$ in the frequency domain signal $X[k]$ because of CFO ϵ leads to ICI (Inter-Carrier Interference).

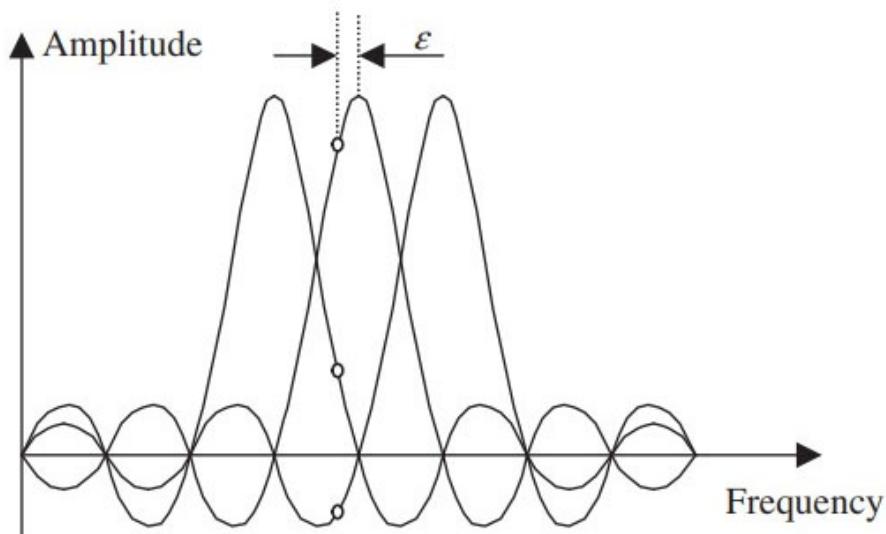


Figure 2-41: the frequency shift of $-\epsilon$ in the frequency domain [9]

2.9.3.2 Integer Carrier Frequency Offset Effect

IFO part leads to:

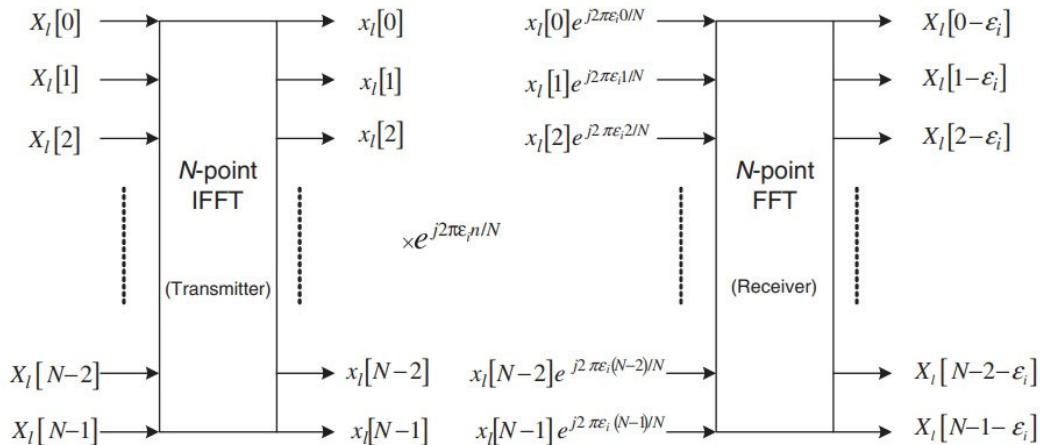


Figure 2-42: The effect of integer CFO on received signal [9]

- Leads to the signal of $e^{j2\pi\epsilon_i n/N}x[n]$ in the receiver.
- The effect of point 1 leads to cyclic shift in k th subcarrier $X[k - \epsilon_i]$.
- Orthogonality of subcarriers is maintained, no ICI.
- Degradation in BER performance if not compensated.

2.9.3.3 Fractional Carrier Frequency Offset Effect

After derivation of the FFT equation, it is summed as:

$$Y_l[k] = \text{FFT}\{y_l[n]\} = \sum_{n=0}^{N-1} y_l[n] e^{-j2\pi kn/N} = \frac{\sin \pi \epsilon_f}{N \sin(\frac{\pi \epsilon_f}{N})} \cdot e^{j\pi \epsilon_f (N-1)/N} H_l[k] X_l[k] + I_l[k] + Z_l[k] \quad (2.9-7)$$

- The coefficient of first term represents amplitude and phase distortion due to FFO.
- I_l represents the ICI from other subcarriers to the k th subcarrier.
- According to point 2, FFO doesn't maintain the orthogonality among subcarriers, ICI exists.
- If $|FFO| > 0.5$, phase difference exceeds π within same OFDM symbol, which leads to phase ambiguity (cannot be determined by a tan inverse function).

Constellation of 3 consecutively received symbols with CFO in the following figure:

Severity increases with increasing FFO due to ICI.

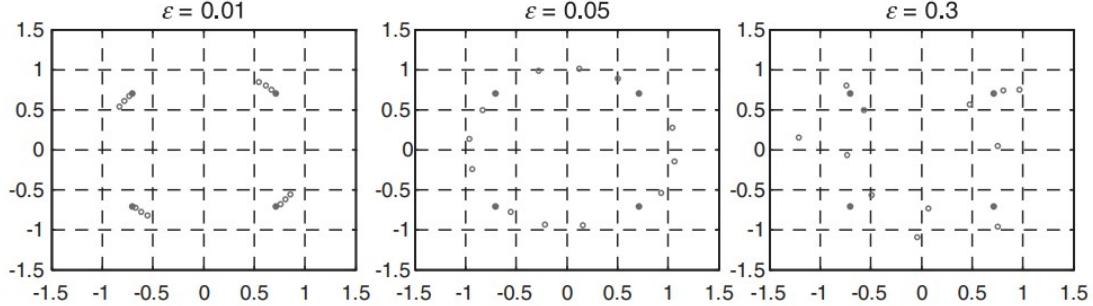


Figure 2-43: The constellation of received signal with CFO ϵ [9]

2.9.4 Estimation Techniques for CFO

CFO estimation can be performed either in the time or frequency domain.

2.9.4.1 Time domain Estimation

For CFO estimation in the time domain, cyclic prefix (CP) or training symbol is used. Each of these techniques is described.

Using CP

For CFO estimation in the time domain, cyclic prefix (CP) or training symbol is used. Each of these techniques is described as below with perfect symbol synchronization, a CFO of ϵ results in a phase rotation of $\frac{2\pi\epsilon n}{N}$ in the received signal. Under the assumption of negligible channel effect, the phase difference between CP and the corresponding rear part of an OFDM symbol (spaced N samples apart) caused by CFO ϵ is $2\pi\epsilon n / N = 2\pi\epsilon$. Then, the CFO can be found from the phase angle of the product of CP and the corresponding rear part of an OFDM symbol, for example,

$$\epsilon = \frac{1}{2\pi} \arg \{ y^*[n]y[n+N] \}, \quad n = -1, -2, \dots, -N_g \quad (2.9-8)$$

In order to reduce the noise effect, its average can be taken over the samples in a CP interval as

$$\hat{\epsilon} = \frac{1}{2\pi} \arg \left\{ \frac{1}{N_g} \sum_{n=-N_g}^{-1} y^*[n]y[n+N] \right\} \quad (2.9-9)$$

Using Time Domain Training Symbol

Estimation Range Problem

The used method above uses tan inverse function, which has the ability to estimate ϵ between -0.5 and 0.5. Since CFO can be large during first synchronization stage, we need other methods to estimate a wider range. The range can be increased by decreasing the distance between the two blocks that are being correlated. This is made possible by using training symbol that has repeated samples inside it. Let D be an integer that represents the ratio of the OFDM symbol length to the length of a repetitive pattern. Let a transmitter send the training symbols with D repetitive patterns in the time domain, which can be generated by taking the IFFT of a comb-type signal in the frequency domain given as:

$$x_I[k] = \begin{cases} A_m, & \text{if } k = D \cdot i, \quad i = 0, 1, \dots, (N/D - 1) \\ 0, & \text{otherwise} \end{cases} \quad (2.9-10)$$

Where A_m represents an M -ary symbol and N/D is an integer. As $x_I[n]$ and $x_I[n+N/D]$ are identical (i.e., $y_I^*[n]y_I[n+N/D] = |y_I[n]|^2 e^{j\pi\epsilon}$), a receiver can make CFO estimation as:

$$\hat{\epsilon} = \frac{D}{2\pi} \arg(\sum_{n=0}^{N/D-1} y_I^*[n]y_I[n+N/D]) \quad (2.9-11)$$

The CFO estimation range covered by this technique is $|\epsilon| \leq \frac{D}{2}$, which becomes wider as D increases. The number of samples for the computation of correlation is reduced by $1/D$, which may degrade the MSE performance. In other words, the increase in estimation range is obtained at the sacrifice of MSE (mean square error) performance.

This is shown in (2.12-1) for $D=1$ and $D=4$

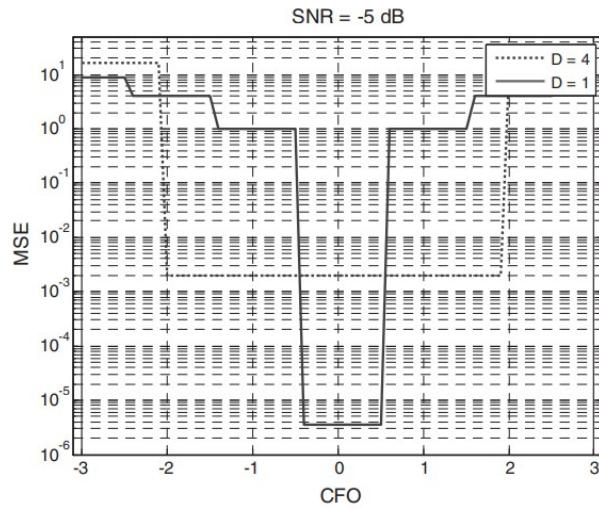


Figure 2-44: MSE vs CFO for OFDM periodicity 1 and 4 [9]

A trade-off relationship between the MSE performance and estimation range of CFO is clearly shown. As the estimation range of CFO increases, the MSE performance becomes worse.

For $D=1$, the MSE is low compared to the MSE of $D=4$ in the range of -0.5 to 0.5. But the opposite happens in the region outside the -0.5 to 0.5 range. So by increasing D we increase the estimation range but with lower precision than D of lower values that can estimate a narrower range with a higher precision. The MSE error can be decreased again by taking the average estimation over more than one symbol.

2.9.4.2 CFO Estimation in Frequency Domain

If two identical training symbols are transmitted consecutively, the corresponding signals with CFO of ϵ are related with each other as follows:

Approach by Moose

$$y_2[n] = y_1[n]e^{j2\pi\epsilon Ne/N} \Leftrightarrow Y_2[k] = Y_1[k]e^{j2\pi\epsilon} \quad (2.9-12)$$

Using the relationship above, the CFO can be estimated as:

$$\hat{\epsilon} = \frac{1}{2\pi} \tan^{-1} \left(\frac{\sum_{k=0}^{N-1} \text{Im}\{Y_1^*[k]Y_2[k]\}}{\sum_{k=0}^{N-1} \text{Re}\{Y_1^*[k]Y_2[k]\}} \right) \quad (2.9-13)$$

which is a well-known approach by Moose. Although the range of CFO estimated by the above equation is $|\hat{\epsilon}| \leq \frac{\pi}{2\pi} = \frac{1}{2}$, it can be increased D times by using a training symbol with D repetitive patterns.

Approach By Classen

The previous approach by Moose is made by a training symbol which requires a preamble before transmitting the actual data. We can think about another technique that allows for transmitting the data symbols while estimating the CFO. As proposed by Classen, pilot tones can be inserted in the frequency domain and transmitted in every OFDM symbol for CFO tracking.

There are two modes for estimation:

Acquisition mode: estimate the IFO and FFO, used at the start of synchronization in order to estimate the IFO.

Tracking mode: estimates the FFO part specifically with higher precision due to the limited range.

After estimating the CFO in the frequency domain, the compensation is then made in the time domain by multiplying the samples by $e^{j\frac{2\pi\hat{\epsilon}n}{N}}$

To sum up Classen approach:

1. Save two consecutive symbols.
2. Estimate the CFO in frequency domain from the pilot tones.
3. Compensate the CFO effect for the next entering samples in time domain.

2.9.5 Packet detection

2.9.5.1 Importance of synchronization and Packet Detection in OFDM [12]

The receiver of an OFDM communication system must perform an accurate synchronization process. If it is not achieved correctly, the received data will not be reliable due to the effect of ICI and ISI, producing degradation of network performance. The two main synchronization processes that must be performed in an OFDM receiver are time and frequency synchronization. A major disadvantage of OFDM signals is the sensitivity to frequency and phase offset. The causes of frequency offset include small differences between the transmitter and receiver carrier frequencies, and the effects of the channel, including Doppler shift. On the other hand, OFDM signals are more robust to time delay, but if this delay is longer than the CP it will yield ICI.

The nature of OFDM signals, allows performing synchronization either in the time or frequency domain. Which domain is chosen, will be determined by a trade-off between performance and computational complexity.

Most of the synchronization algorithms are based on the use of training sequences (TS), which are specified by the 802.11a standard, known at the receiver, and allow detection of packet presence and computation of the frequency and phase of the signal. An important assumption is that because of the fast and short transmission characteristic of a WLAN packet, the channel is considered unchanged during the packet, so the majority of the synchronization is performed in the preamble and used during the whole packet.

#

Timing estimation consists of two main tasks: packet synchronization and symbol synchronization. The IEEE 802.11a MAC protocol is essentially a random access network, so the receiver does not know exactly when a packet starts. The first task of the receiver is to detect the start of an incoming packet. Broadcast systems naturally do not require packet detection algorithms, because transmission is always on. However, for a packet oriented network architecture, finding the packets is obviously of central importance for high network performance.

2.9.5.2 Packet Detection Process [13]

Packet detection is the task of finding an approximate estimate of the start of the preamble of an incoming data packet. As such it is the first synchronization algorithm that is performed, so the rest of the synchronization process is dependent on good packet detection performance. Generally packet detection can be described as a binary hypothesis test. The test consists of two complementary statements about a parameter of interest. These statements are called the null hypothesis, H_0 , and the alternative hypothesis, H_1 . In the packet detection test, the hypotheses assert whether a packet is present or not. The test is set up as shown below.

$$H_0: \text{Packet not present}$$

$$H_1: \text{Packet is present}$$

The actual test is usually of the form that tests whether a decision variable m_n exceeds a predefined threshold W_k . The packet detection case is shown below.

$$H_0: m_n < Th \rightarrow \text{Packet not present} \#$$

$$H_1: m_n \geq Th \rightarrow \text{Packet is present} \#$$

The performance of the packet detection algorithm can be summarized with two probabilities: probability of detection P_D and probability of false alarm P_{FA} . P_D is the probability of detecting a packet when it is truly present, thus high P_D is a desirable quality for the test. P_{FA} is the probability that the test incorrectly decides that a packet is present, when actually there is none, thus P_{FA} should be as small as possible. In general, increasing P_D increases P_{FA} and decreasing P_{FA} decreases P_D , hence the algorithm designer must settle for a some balanced compromise between the two conflicting goals.

Generally it can be said that a false alarm is a less severe error than not detecting a packet at all. The reason is that after a false alarm, the receiver will try to synchronize to non-existent packet and will detect its error at the first received data integrity check. On the another hand, not detecting a packet always results in lost data.

A false alarm can also result in lost data, in case an actual data packet starts during the time the receiver has not yet detected its mistake. In this case, the receiver will not be able to catch that packet. The probability of this occurring depends on several issues like the network load and the time it takes for the receiver to detect its mistake. In conclusion, a little higher P_{FA} can be tolerated to guarantee good P_D .

We introduce several approaches to the packet detection test design problem.

#

2.9.5.3 Types of Packet Detection Algorithms [14]

Four packet detection algorithms are introduced, as shown in figure 2-39. They are divided into two categories according to the signal characteristics used: energy-based and preamble-based. The energy-based algorithms use the change of signal energy. The preamble-based ones take advantage of the preamble structure.

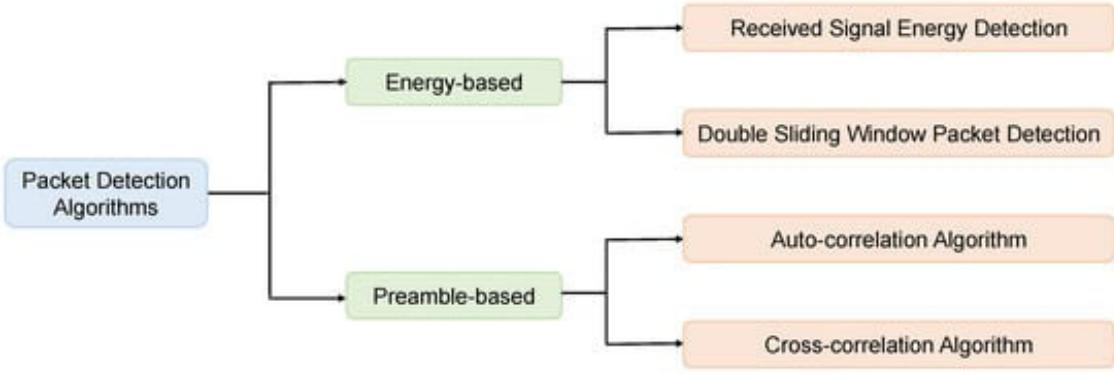


Figure 2-45: Packet Detection Algorithms [12]

Energy-Based Algorithms

Energy-based algorithms measure the received signal energy and do not require additional knowledge. When there is no packet, the received signal r_n only consists of noise. When the packet starts, the received energy increases due to the signal component. Therefore, there will be an obvious energy change when a signal arrives, and that change can be used for packet detection. The following are two energy-based detection algorithms: received signal energy detection and double sliding window packet detection.

RECEIVED SIGNAL ENERGY DETECTION

The received signal energy detection algorithm detects packets by directly identifying the change in the received signal energy level, but using only one sample for energy detection is easily affected by single strong noise. A calculation window containing multiple samples is used. The decision variable m_n is set as the received signal energy accumulated over the window of length L as Equation $m_n = \sum_{k=0}^{L-1} r_{n-k} r_{n-k}^*$

(2.9-14). A sliding window can be used to simplify the computation, as shown in. At every time instant n , one new value enters the window and one old value is discarded, so m_n is calculated by a moving sum of received signal energy by Equation (2.9-17). Therefore, the number of complex multiplications per m_n is reduced to one received sample.

$$m_n = \sum_{k=0}^{L-1} r_{n-k} r_{n-k}^* \quad (2.9-14)$$

$$m_{n+1} = m_n + |r_{n+1}|^2 - |r_{n-L+1}|^2 \quad (2.9-15)$$

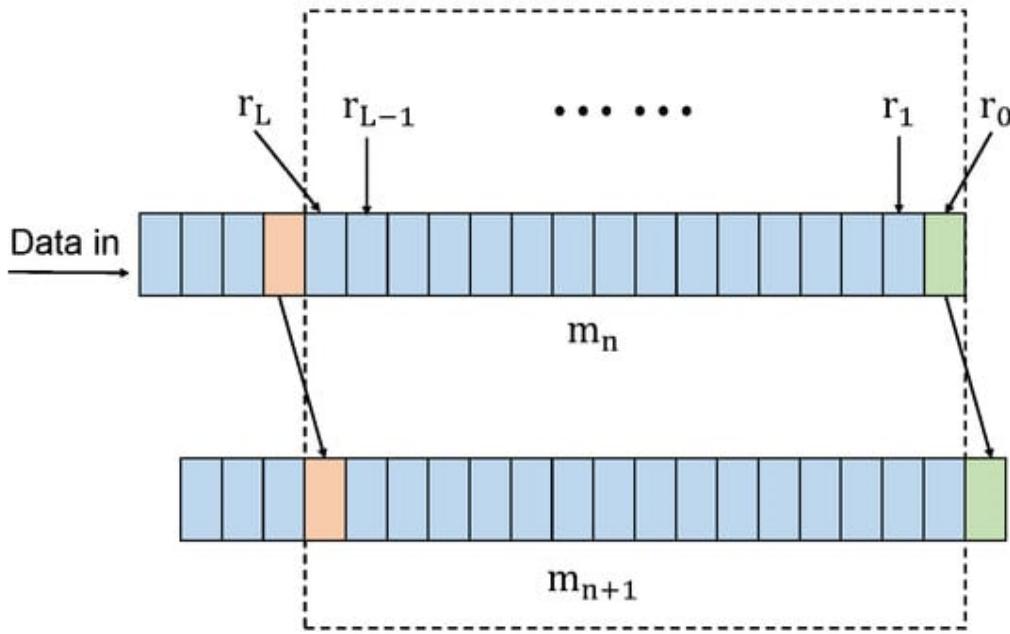


Figure 2-46: Sliding Window Received Signal Energy Detection [12]

Simulation. The response of the simulation experiment is shown in Figure 2-47. The real start location of the packet is at the index of 101. The sliding window length for accumulation is set as 16 ($L = 16$). There is an obvious jump around the index of 101, which is consistent with the real situation. This algorithm is sensitive to noise and is difficult for threshold setting since it directly depends on the received signal energy. The next algorithm reduces reliance on energy power.

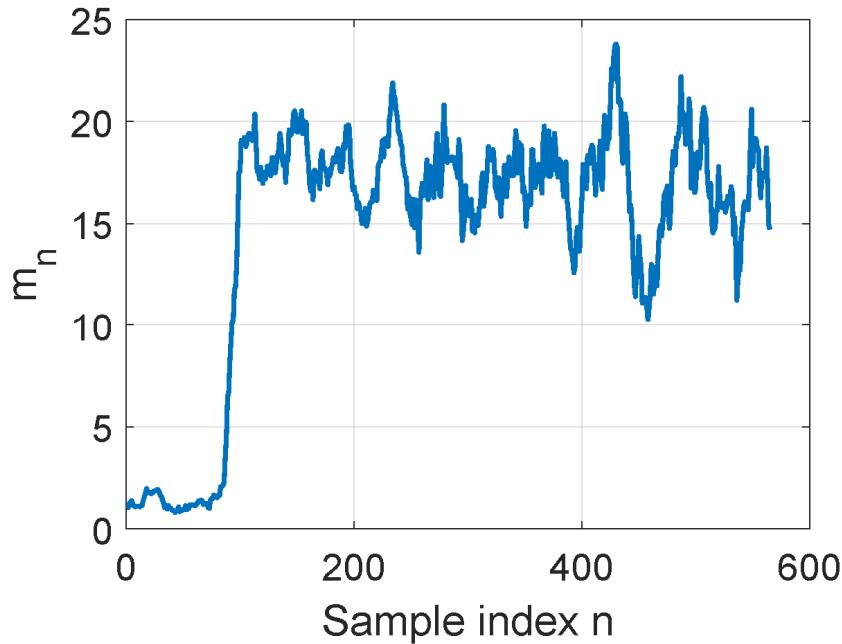


Figure 2-47: Energy Detection using single sliding Window [12]

DOUBLE SLIDING WINDOW PACKET DETECTION

The double sliding window packet detection algorithm is also based on received energy, but it uses the ratio of signal power in two consecutive sliding windows to detect packets , as shown in . Two sliding windows, A and B , are used to calculate the energy inside them by Equations (2.9-16) and (2.9-17). The lengths of two sliding windows are M and L , respectively. m_n is set as a ratio of the values calculated in two windows by Equation (2.9-18). Thus, when the packet is absent, the response is flat since the two windows contain almost the same amount of noise energy. When the packet comes into window A , the energy level in A keeps increasing until A is totally covered by the packet. Later, window B starts to collect signal, so m_n decreases and comes back to flat when B is covered completely. Therefore, m_n is large only when the packet is just coming. The sample index n of the peak point in the triangle-shaped response marks the beginning of the packet.

$$a_n = \sum_{k=0}^{M-1} r_{n-k} r_{n-k}^* \quad (2.9-19)$$

$$b_n = \sum_{k=1}^L r_{n+k} r_{n+k}^* \quad (2.9-20)$$

$$m_n = \frac{a_n}{b_n} \quad (2.9-21)$$

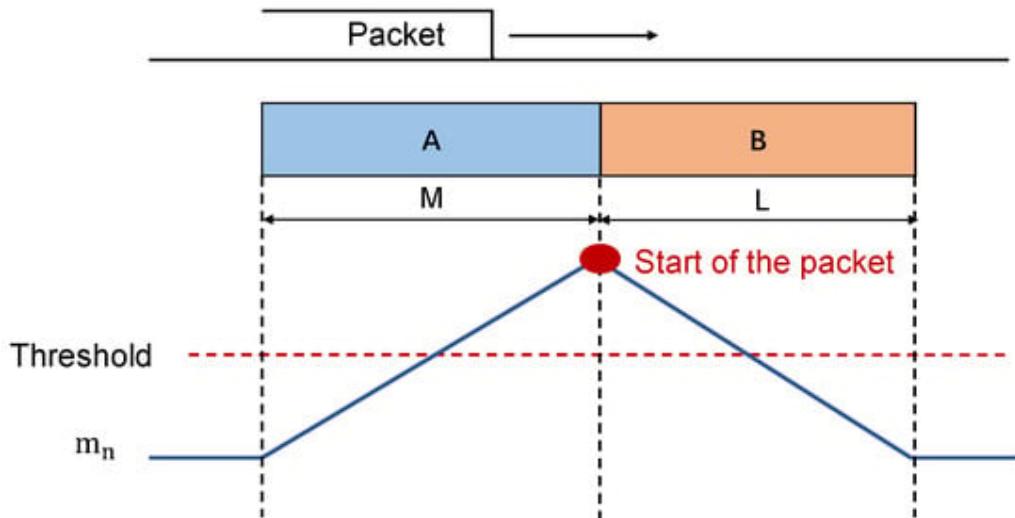


Figure 2-48 Double Sliding Windows Energy Detection [12]

Simulation in Figure 2-49 depicts the simulated response of this algorithm. The lengths of both windows (M, L) are set to 16. There is an obvious triangular peak around the index of 101, which is the real start position of the packet. The other points are very low, which shows that m_n does not depend on the received power. This algorithm is more noise-resistant than the first one, but has twice the amount of computation. Both the above algorithms need no additional knowledge of packets; however, if the receivers know the packet structure, the following algorithms can be considered.

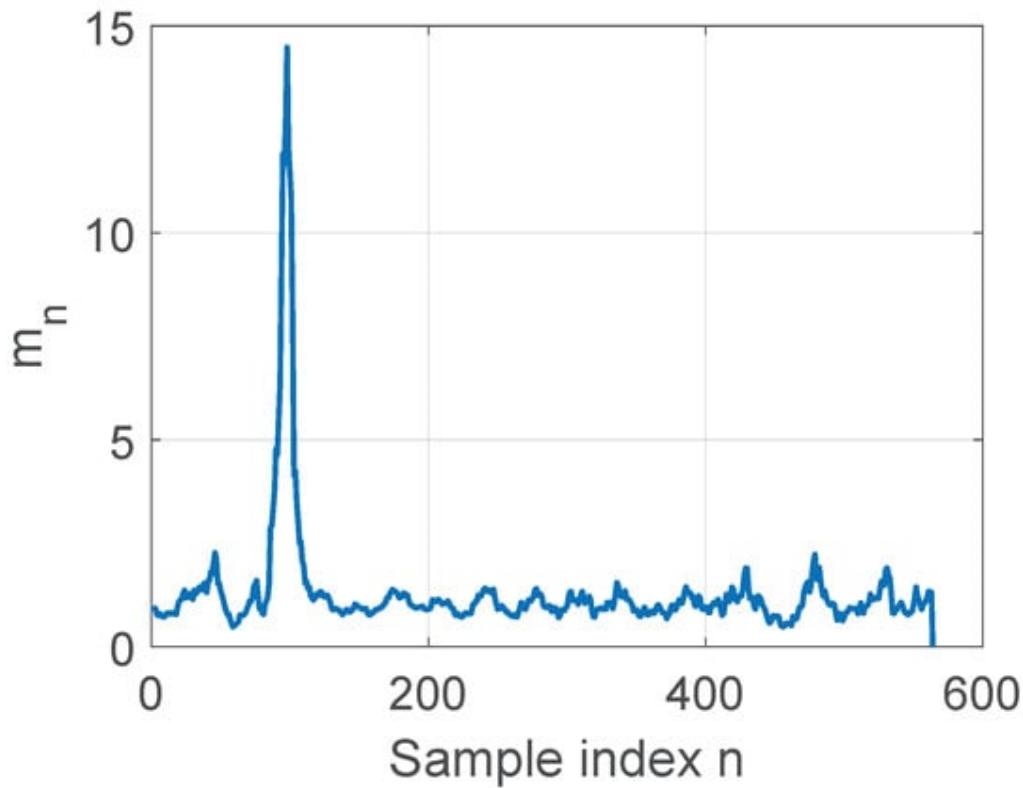


Figure 2-49 : Double sliding Window Synchronization [12]

Preamble-Based Algorithms

A general communication system engineering principle is that the receiver should use all the available information to its advantage. If the structure of the preamble is known, it should be incorporated into the packet detection algorithms to achieve better detection performance. The following two algorithms, auto-correlation and cross-correlation, use the correlation of the preamble to obtain the decision variable m_n

AUTO-CORRELATION ALGORITHM

The auto-correlation algorithm uses the structure of the short preamble and takes advantage of its periodicity. The 10 repetitive short symbols in the short preamble have good auto-correlation, which are not available in noise and other signals. The algorithm framework is shown in Figure 2-44 . Although this method also uses two sliding windows, instead of energy calculation, window C is used to calculate the correlation between the received signal and a delayed version by Equation (2.9-25). When there is only noise, the correlation of noise samples is almost zero since noise does not have auto-correlation characteristics. When the packet is present, the correlation of the identical short preamble is high. The delay Z^{-D} is equal to the period of one symbol length. For IEEE 802.11a, $D = 16$, which is the period of a short training symbol. Window P is used to calculate the signal energy during the correlation window to normalize m_n by Equation (2.9-26). The lengths of both windows are L . Further, both c_n and p_n are squared to ensure that m_n is positive, as shown in Equation (2.9-27).

$$c_n = \sum_{k=0}^{L-1} r_{n+k} r_{n+k+D}^* \quad (2.9-22)$$

$$p_n = \sum_{k=0}^{L-1} r_{n+k+D} r_{n+k+D}^* \quad (2.9-23)$$

$$m_n = \frac{|c_n|^2}{(p_n)^2} \quad (2.9-24)$$

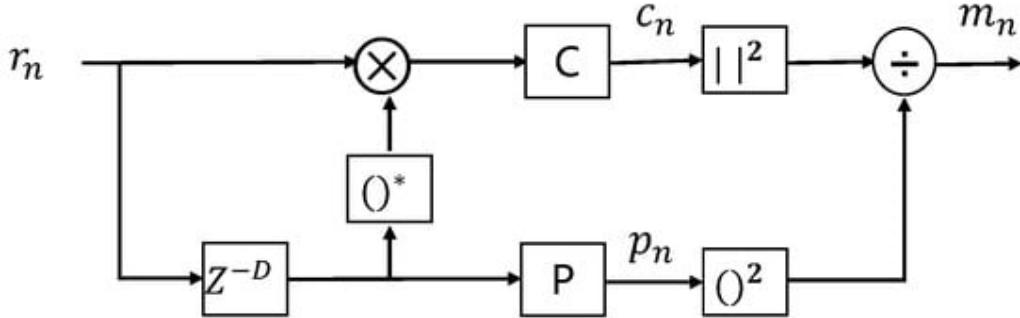


Figure 2-50: Block Diagram of auto-correlation Detection [12]

Simulation. The simulation result is shown in Figure 2.45. The window size L is 16. m_n is high during the whole STF period from index 101 to 260 and remains low before the start of the packet since there is only noise. The difference in response from the double sliding window is caused by the c_n value. When the packet is present, the correlation result c_n of the identical short preamble is high, so m_n jumps to its maximum value. the auto-correlation algorithm only uses the periodicity of the preamble, and there is no need to know the specific content. However, if the structure of the preamble is available, the following method is another good choice.

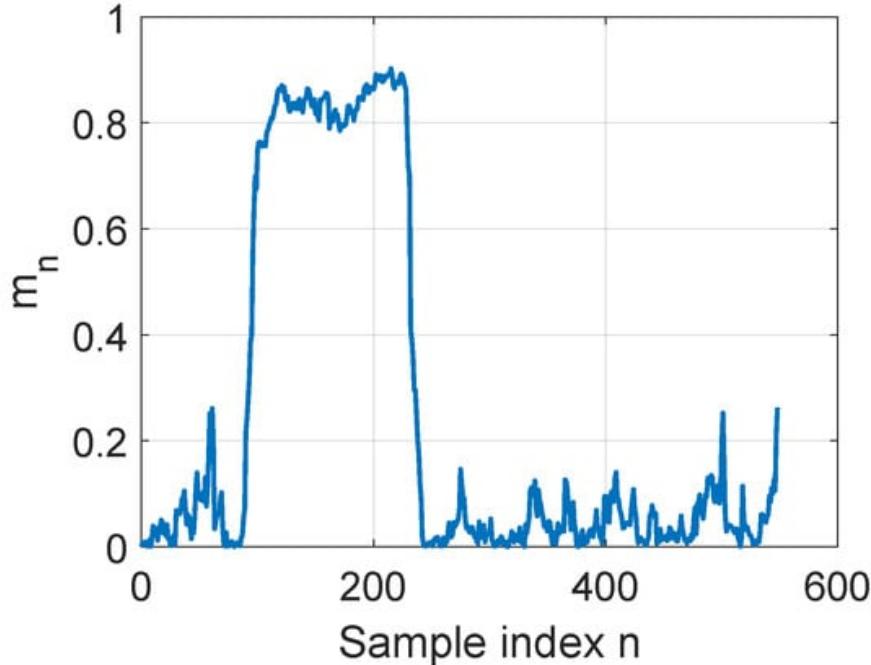


Figure 2-51 Auto-correlation algorithm applied to packet [12]

CROSS-CORRELATION ALGORITHM

The cross-correlation algorithm also uses correlation calculation of the preamble. The difference from the former one is that it uses a template instead of the received signal itself. Therefore, completely knowing the preamble structure is the premise of using this algorithm. The flowchart is shown in Figure 2-46. In this approach, one symbol from the short preamble is used as a template, T_n . The cross-correlation is performed between the received signal r_n and the stored template T_n . Window C is used to calculate the correlation between the two signals by Equation (2.9-29). Only when the two parts match is the calculated result large, and the rest of the results small. There will be 10 peaks in the results since STF contains 10 repeated symbols. These peak points are very easy to identify. The correlation result is also normalized by the template signal power calculated in window P by Equation (2.9-30). m_n is the same as Equation (2-78).

$$c_n = \sum_{k=0}^{15} r_{n+k} T_k^* \quad (2.9-25)$$

$$p_n = \sum_{k=0}^{L-1} T_{n+k} T_{n+k}^* \quad (2.9-26)$$

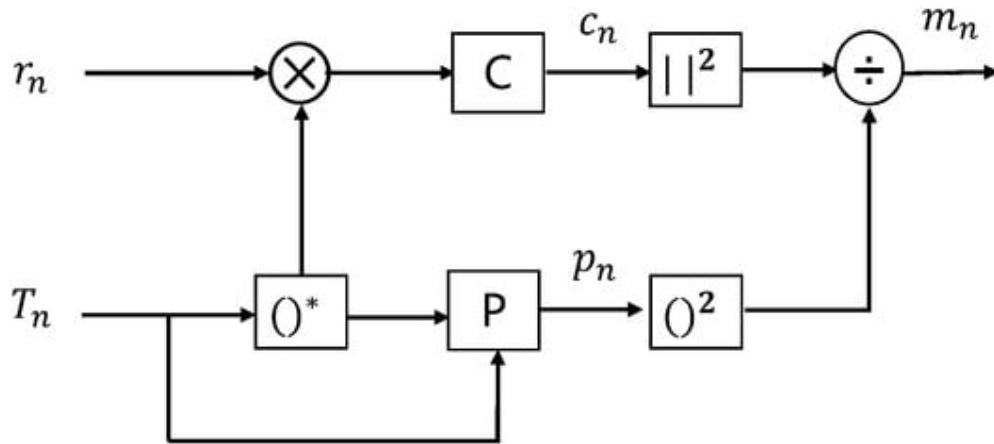


Figure 2-52 Cross-Correlation Block Diagram [12]

Simulation. The experiment result is depicted in Figure 2.47. There are 10 peak points during the index from 101 to 260, which are much higher than others. Especially when there is only ambient noise at the beginning, the matching values are very small. Because the template in this algorithm is fixed, the received signal can only match the template in the specific positions, but for the auto-correlation algorithm, it can be considered that its template is moving along the signal, so m_n is high during the entire STF period.

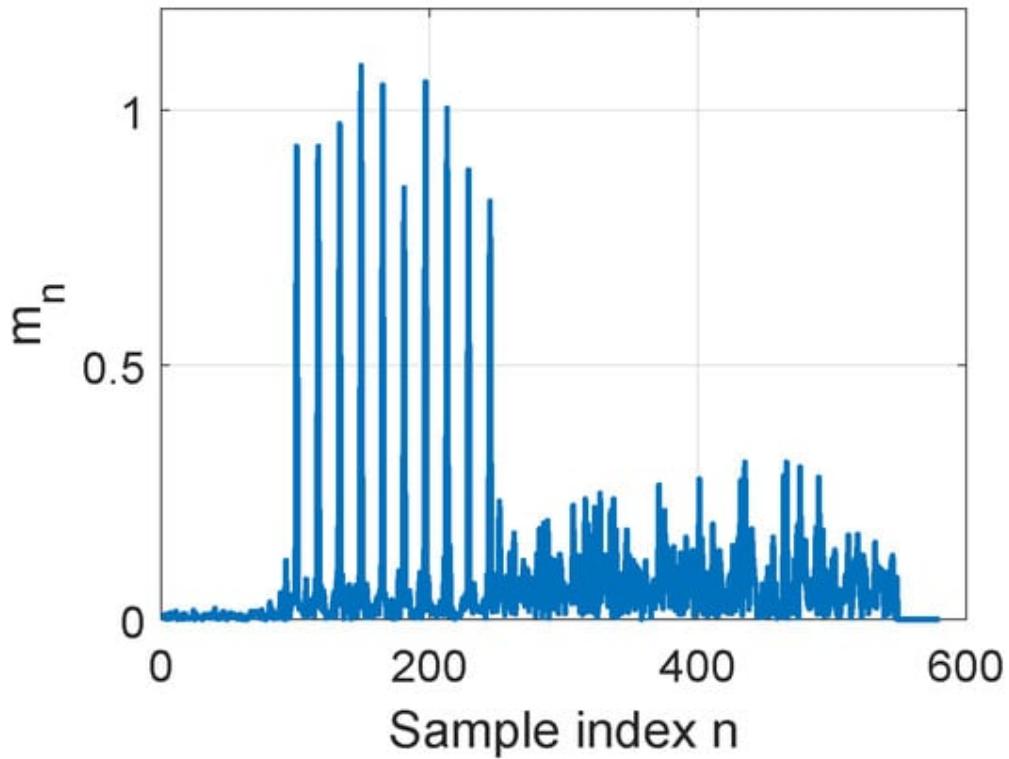


Figure 2-53: Cross-Correlation Algorithm applied to short symbols [12]

2.9.5.4 Measure Metrics

The performance of the packet detection algorithm can be measured by three indicators: the probability of detection P_D , the probability of false alarm P_{FA} , and the probability of misdetection P_m . False alarm is less severe than misdetection, since missing packets result in data loss. A good algorithm should achieve a high P_D , which is related to P_{FA} and P_m . The algorithm designers must balance them properly. There are two main factors that affect these three indicators: threshold setting and decision rule. The detailed discussions are as follows.

2.9.5.5 Threshold Setting

When the threshold is too high, most decision variables m_n are lower than it, so P_m will be high, and P_{FA} and P_D will be low. In contrast, if the threshold is too low, P_D will also be low since the results may be affected by some larger noise points, which will lead to a high P_{FA} . The changes of P_{FA} and P_m with threshold changing are shown in 2.13-19 and figure 2.13-20. The threshold is set from 0 to 1. The simulation experiment uses the auto-correlation algorithm and the SNR is set to -2, 0, and 2, respectively. Overall, as the threshold increases, P_{FA} decreases and P_m increases. Due to uncertain environmental factors, it is hard to set a definite threshold, but the setting range of the threshold can be determined. Here are two examples of how to find the range of threshold settings.

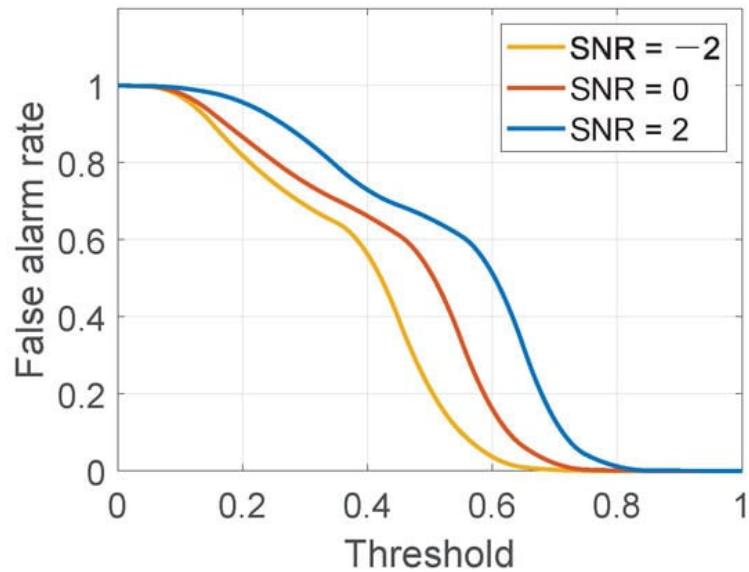


Figure 2-54: False Alarm rate at different SNRs and Thresholds [12]

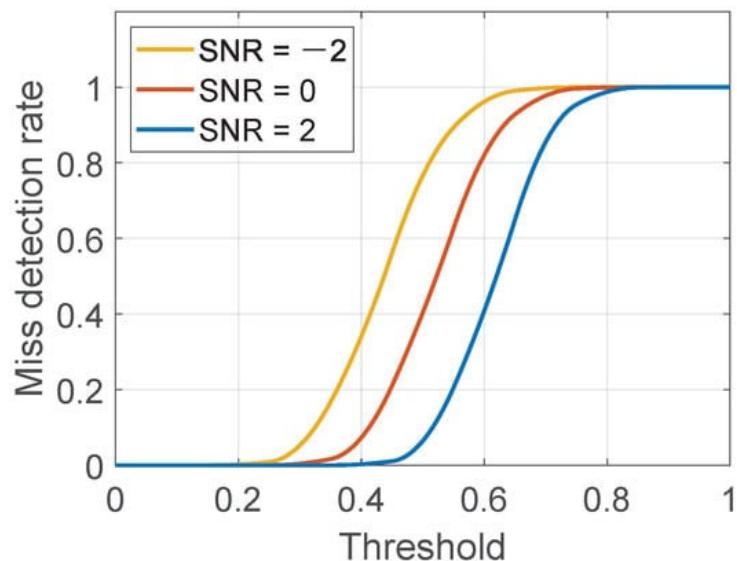


Figure 2-55: Miss Detection rate at different SNRs and Thresholds [12]

Double sliding window. For this algorithm, when the lengths of two windows are the same, the peak point of m_n can be used as a reference for threshold setting. At m_{peak} , the value of a_{peak} contains the sum of signal S and noise N, and the b_{peak} only contains noise N, thus m_{peak} can be estimated by the received SNR as Equation (2.9-31). When knowing the SNR in the environment, the threshold setting has a reference range, which is smaller than SNR+1

$$m_n = \frac{a_n}{b_n} = \frac{S+N}{N} = SNR + 1 \quad (2.9-27)$$

Auto-correlation. Similarly, the results of the auto-correlation algorithm can also be denoted by SNR. When the noise signal is additive white Gaussian noise (AWGN), the auto-correlation result of Gaussian noise signal is 0. Thus, there is only the signal power in the numerator after auto-correlation calculation, and the relationship between m_{peak} and SNR is shown as Equation (2.9-32).

$$m_{peak} = E \left[\left(\frac{c_{peak}}{p_{peak}} \right)^2 \right] = \left(\frac{S}{S+N} \right)^2 = \left(\frac{SNR}{1+SNR} \right)^2 \quad (2.9-28)$$

Then, a moderate value in this range is chosen or a fixed scale factor is set, such as 0.75, multiplied by m_{peak} as the threshold.

2.9.5.6 Decision Rule

Another way to improve P_D is to set up a proper decision rule. The simple method is to compare m_n and T_h . A complex one can be designed based on the characteristics of the response of each algorithm.

The complex decision rule achieves a higher detection rate because it reduces P_{FA} well, and the set conditions can be mostly satisfied when the packet arrives, so it will not cause P_m to increase too much.

A proposed decision rule is showed in figure 2-50, where a packet is decided to be existent when the number of 1's is greater than 1.5 Length of Short training symbol and distance between 1st peak and subsequent peaks is less than or equal to 3 Length of Short training symbols.

Figure 2-51 shows the detection rate of simple auto-correlation decision rule and complex auto-correlation decision rule.

Merely comparing the m_n and the T_h is the simplest rule, which may increase P_{FA} because of the large singularity. However, that does not mean that a more complex decision rule is necessarily good, since it will lead to an increase in P_m . Thus, a suitable decision rule is necessary to achieve high P_D .

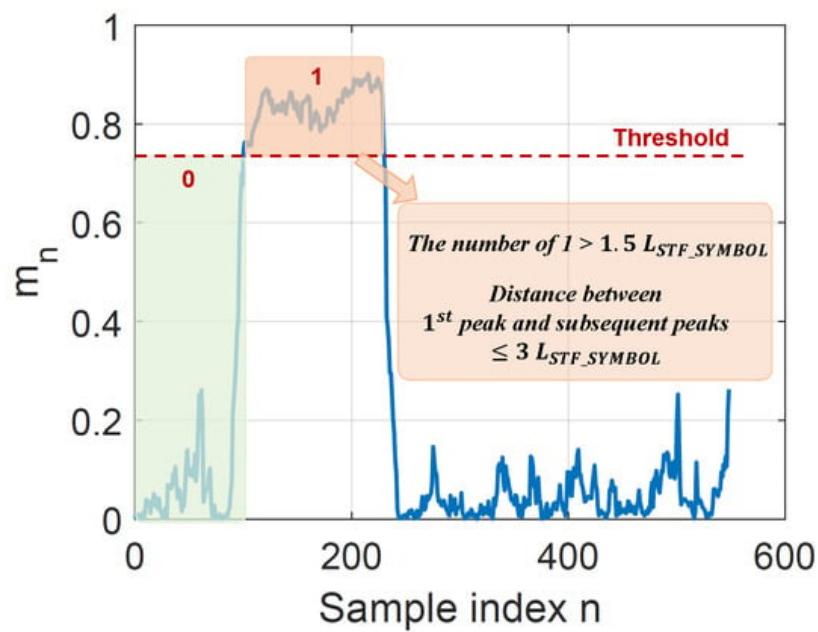


Figure 2-56: A decision for auto-correlation algorithm [12]

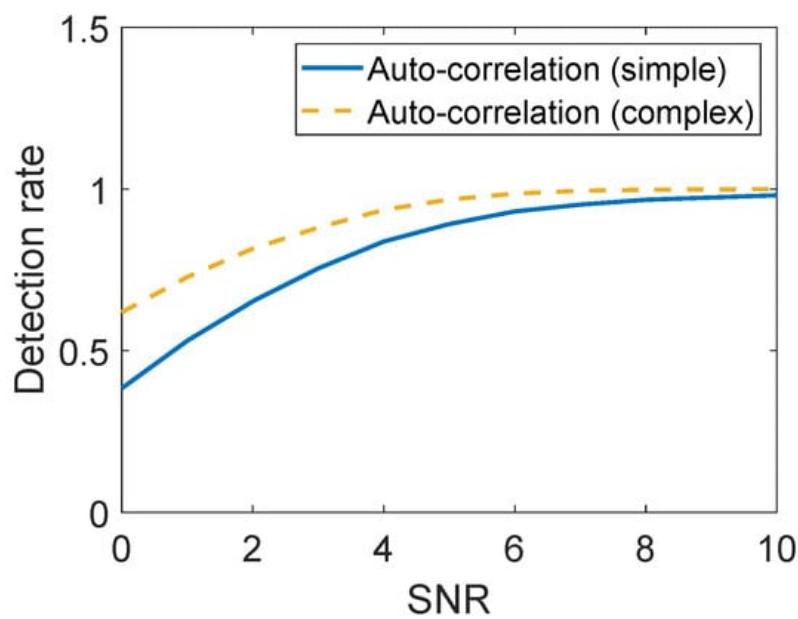


Figure 2-57: Comparison between Complex and simple decision [12]

2.9.5.7 Comparison of Detection Algorithms[14]

we mainly focus on three dimensions for packet detection algorithms: the detection rate, the synchronization accuracy, and computational complexity. To make the comparison reasonable, we use the uniform simple decision rule. The IEEE 802.11ah protocol is used for the simulation experiments. The added noise signal is AGWN and the SNR changes from 0 to 10. To obtain a reliable result, each value is averaged by 10,000 times simulation experiments.

2.9.5.8 Detection Rate

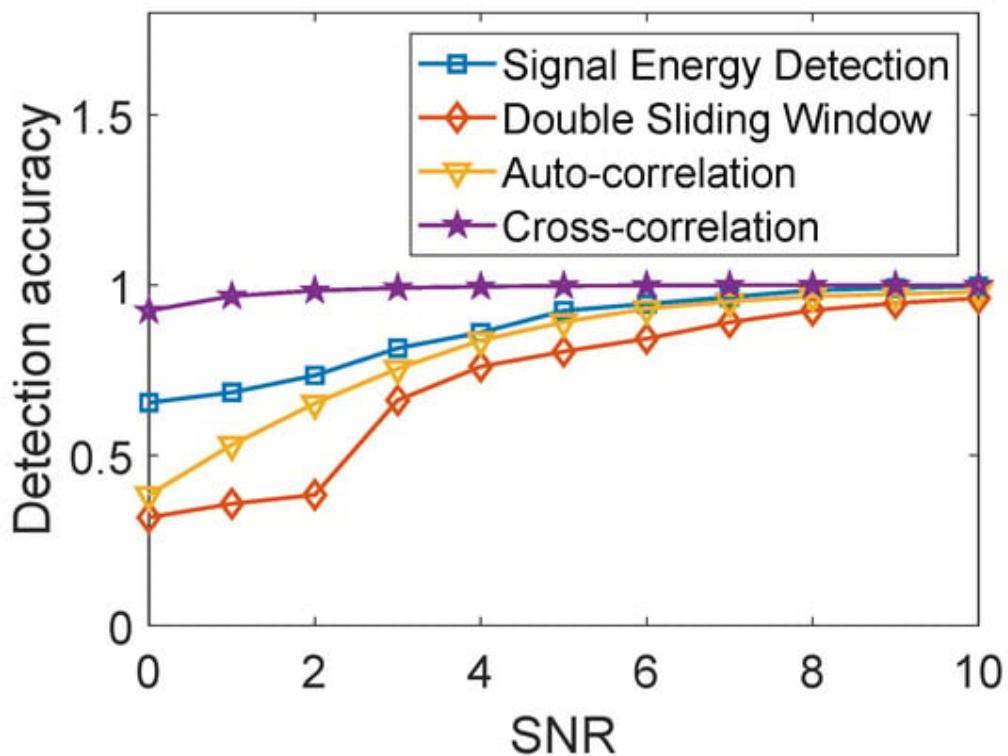


Figure 2-58: Comparison between different detection algorithms accuracy [12]

2.9.5.9 Synchronization Accuracy

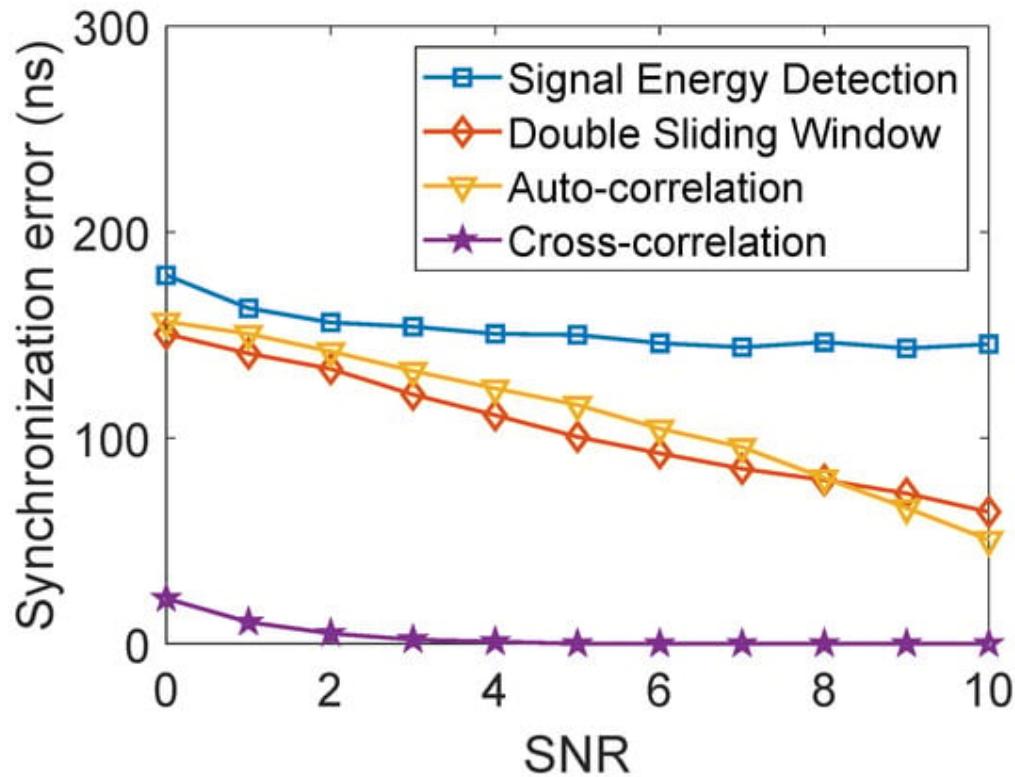


Figure 2-59: Synchronization error comparison of different algorithms [12]

2.9.5.10 Computational Complexity

Computation complexity is very important when applied to a practical application, especially for low-power hardware. In the experiments, we use the program running time of 10,000 times operations to represent the power consumption.

Table 2.9-3: Running times of different packet detection algorithms

Packet Detection Algorithm	Running Time (s)
Signal Energy Detection	0.95
Auto-Correlation	1.97
Double Sliding Window	2.32
Cross-Correlation	3.56

2.9.5.11 Discussion[14]

Signal energy detection

Advantages:

- 1-Simplest Algorithm, requiring no additional information and can work in all systems.
- 2- It has the lowest computational power because it only calculates the energy within a single window.
- 3-It has a decent detection rate when the threshold is set properly, but it is sensitive to noise and cannot set a fixed threshold since m_n directly depends on the received signal energy.

Disadvantages:

The synchronization performance is poor. The reason is that energy accumulation is a continuous process. When the packet arrives, there will be a rising edge for a period of time, and it is difficult to determine the exact start position of the signals.

Double sliding window detection

Advantages:

- 1-uses the ratio of two windows for detection, so m_n does not depend on the signal energy level and computational complexity is twice that of the former one.
- 2-SNR can be regarded as a reference for threshold setting. In this algorithm, there is an obvious marker point to indicate the starting position of the data packet, so the synchronization error is smaller than the previous one.

Disadvantages:

In theory, the detection rate of this algorithm should be close to or better than the first method, but in experiments, we find that its performance is not as good as expected. Compared with the first one, this method has a certain improvement in synchronization, but it costs twice the power consumption, so it has not been widely used in real systems.

Auto-correlation Packet Detection

Description:

- 1-Takes advantage of the 10 repetitive short symbols in the preamble, so it can only be applied to OFDM signals with STF structure.
- 2-By auto-correlation calculation, the value of m_n is high throughout the STF period. When using a simple decision rule, its performance is similar to energy detection because the above structure is not fully utilized.
- 3-When the complex rule is used, the performances of detection rate and synchronization are better, but it will bring some increase in computational complexity.
- 4-Overall, this algorithm has a good trade-off between the three dimensions and is used as an example algorithm for packet detection on MATLAB.

Cross-correlation Packet Detection

Description:

- 1-Takes the known short preamble as a template and calculates the correlation between the template and the received signals.
- 2- m_n will be high only when they are matched exactly. Therefore, a more accurate matching position can be obtained and it can achieve a better synchronization performance.

In this case, the decision rules need to be set properly. If the rules are too strict, the detection rate will decrease.

Disadvantages:

- 1-The biggest problem with this algorithm is the high computational complexity, especially when the decision rules are complicated.
- 2-Compared with the previous method, there is no requirement for the structure of the preamble, but the specific content of the signal needs to be known.

All in all, each of these algorithms have their own advantages and are suitable for different scenarios. The auto-correlation has the best overall performance.

3 CHANNEL MODELLING RESULTS

The simulation method was based on [15]

In this chapter, simulation results of a SISO Channel Simulator will be demonstrated.

The Channel Simulator can simulate a custom channel (ITU models, COST 207, 3GPP models) with the following characteristics:

- Small-scale Fading Characteristics:
 - Frequency Flat/Selective by specifying a time varying or Block faded Power Delay Profile
 - By time varying it is meant: variable tap gains, variable delays and a variable number of taps
 - The Power Delay Profile is sampled at the same rate as the signal. multiple sampling rate with interpolation must be used if TDL is sampled at a different sampling rate
 - Time Variation Slow/Fast with specified Doppler Spectrum:
 - Jake's, Gaussian, (More to be added later like rounded, flat, ...)
 - Channel Envelope Distribution:
 - Rayleigh
 - Rician
- Correlated or Uncorrelated paths (WSSUS not valid) (to be added later)
- Diffuse or Discrete Multipath Channel Model

3.1 GENERATION OF TAP-GAIN PROCESSES

The Modelling approach considered thus far require two main steps for their implementation:

- First, we generate a set of white (discrete-time) Gaussian processes.
- Second, we need to shape the power spectral density of these processes to assume the shape of the Doppler spectrum at each tap location. Common methods include:
 - Autoregressive Moving Average (ARMA) Models
 - The Spectral Factorization

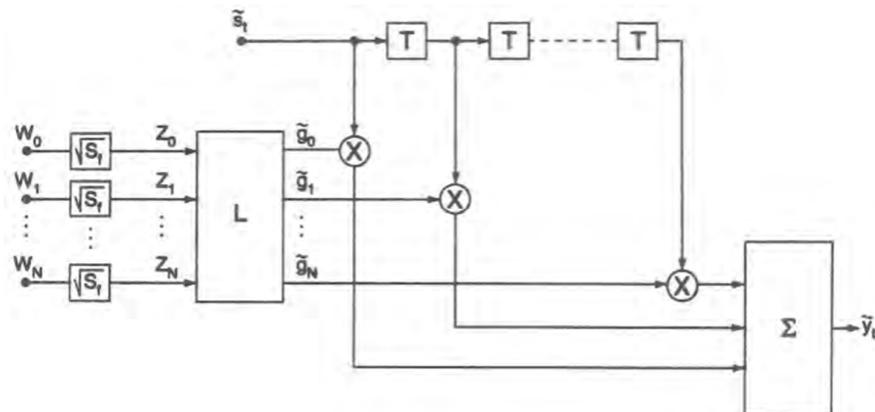


Figure 3-1: TDL channel model

The two fundamental approaches to model colored gaussian random processes according to the required doppler spectrum are:

- The filter method
- The Sum-of-Sinusoid's method

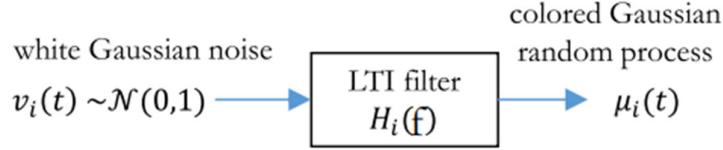


Figure 3-2: Filter methods

The filter method is beyond the scope of this document. The Sum-of-Sinusoids will be given more attention in the upcoming section

3.2 THE SUM-OF-SINUSOIDS METHOD

The Sum-of-Sinusoid's method, also known as the Rice method, is based on weighted summation of an infinite number of harmonic waves with random phases and equidistant frequencies. In this method, the colored Gaussian random process is described as:

$$\mu_i(t) = \lim_{N_i \rightarrow \infty} \sum_{n=1}^{N_i} c_{i,n} \cos(2\pi f_{i,n} t + \theta_{i,n}) \quad (3.2-1)$$

The random phases $\theta_{i,n}$ are random variables uniformly distributed in the interval $(0, 2\pi]$ and the frequency spacing Δf_i is chosen in such a way that the n equidistant frequencies cover the whole frequency range.

The stochastic Rice method can be equivalently converted to a deterministic realizable model by using only a finite number of harmonic functions N_i and by taking the phase terms $\theta_{i,n}$ out of a random generator whose samples are uniformly distributed in the interval $(0, 2\pi)$.

$$\mu_i(t) = \sum_{n=1}^{N_i} c_{i,n} \cos(2\pi f_{i,n} t + \theta_{i,n}) \quad (3.2-2)$$

The following figure shows a deterministic model for generating a generic Rice process. By appropriately choosing the model parameters $c_{i,n}$, $f_{i,n}$ and $\theta_{i,n}$, the time-variant fading behavior caused by the Doppler effect can be simulated. From now, the model parameters are named as follows: $c_{i,n}$ are called Doppler coefficients, $f_{i,n}$ are termed as discrete Doppler frequencies, and $\theta_{i,n}$ are the Doppler phases. For the purpose of computer simulations, the model can be translated to its discrete time equivalent by setting $t = kT_s$, where T_s is the sampling interval and k is an integer. In computer simulations, the model parameters $c_{i,n}$, $f_{i,n}$, and $\theta_{i,n}$ are computed for $n = 1, 2, \dots, N_i$ and kept constant for the whole duration of the simulation.

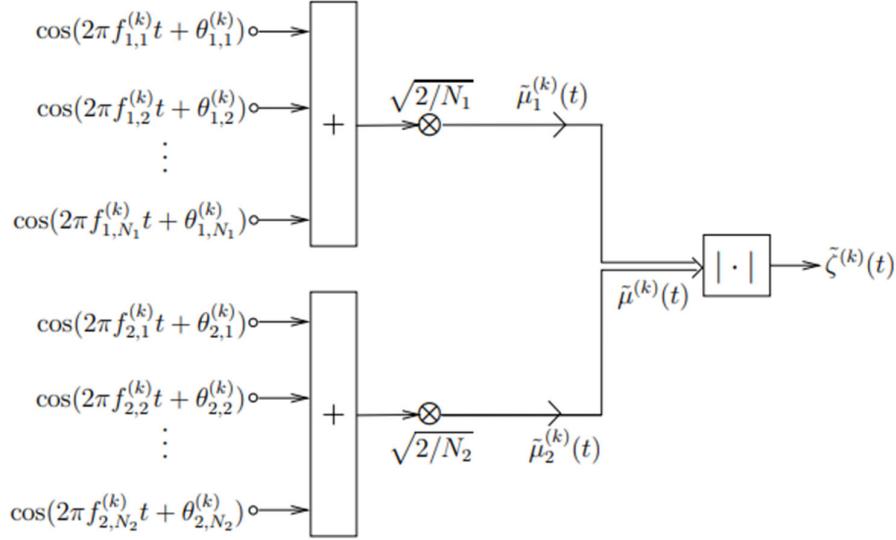


Figure 3-3: Sum-of-Sinusoid's method

Of the several methods that are available to compute the model parameters, the method of exact Doppler spread (MEDS) is chosen here for simulation. This method can be used to simulate a frequency non-selective Rice or Rayleigh fading channel. It is well suited for generating the underlying-colored Gaussian random samples with Jakes power spectral density. Using the MEDS method, the Rice process with Jakes PSD can be generated by choosing samples from an uniform random generator in the interval $(0, 2\pi]$ for $\theta_{i,n}$ and by setting the following values for the other model parameters.

$$c_{i,n} = \sigma_0 \sqrt{\frac{2}{N_i}} \quad (3.2-3)$$

$$f_{i,n} = f_{max} \sin \left(\frac{\pi}{2N_i} \left(n - \frac{1}{2} \right) \right) \quad (3.2-4)$$

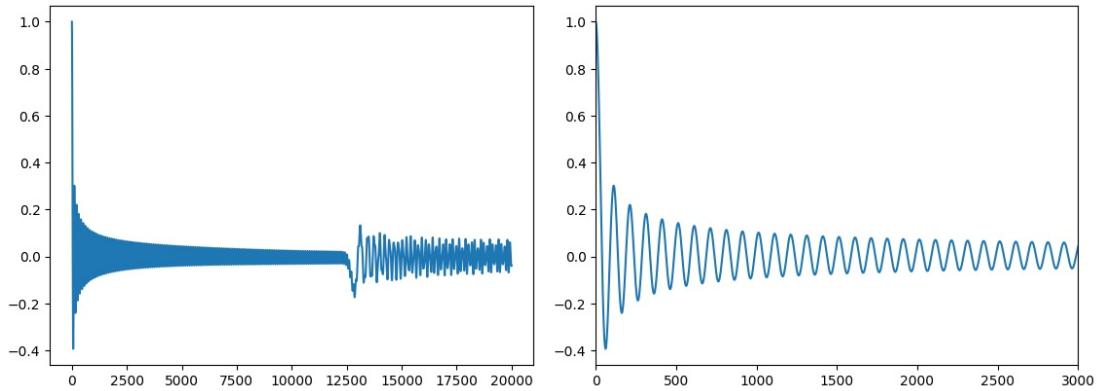


Figure 3-4: Spaced time correlation function

3.3 FLAT FADING CHANNEL WITH A JAKE'S DOPPLER SPECTRUM

The simulator had the following configuration:

- Number of Sinusoids: 200
- Max Doppler Shift: 100 Hz
- For Rician Process:
 - LoS Amplitude: 1 volt
 - LoS Doppler Shift: 5 Hz
 - LoS Phase Shift: 0.5 rad

3.3.1 Channel with a Rayleigh Process

Channel Impulse Response:

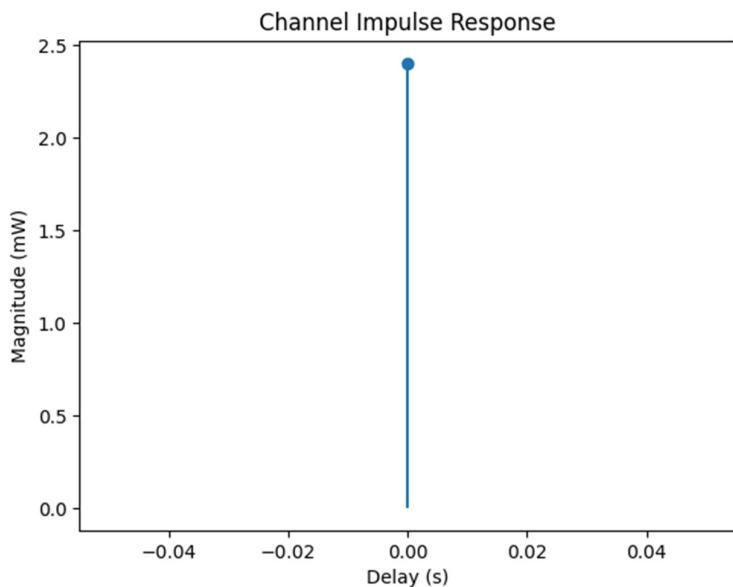


Figure 3-5: Channel Impulse Response

Channel Frequency Response:

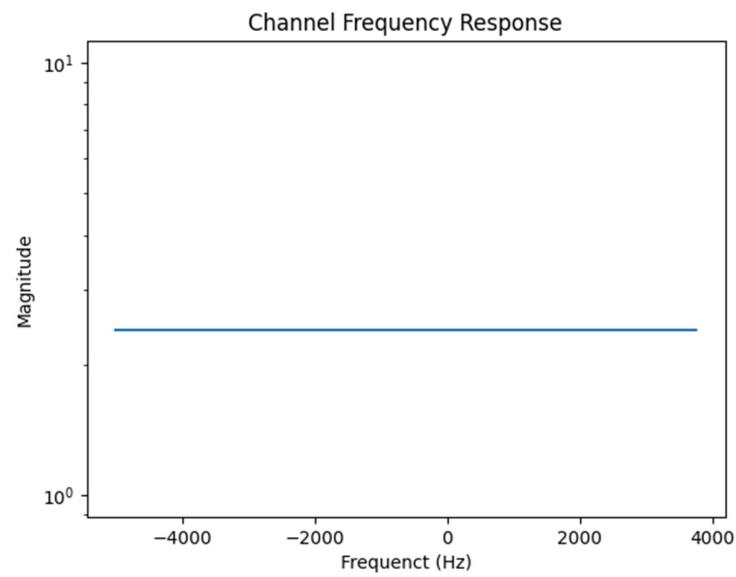


Figure 3-6: Channel Frequency Response

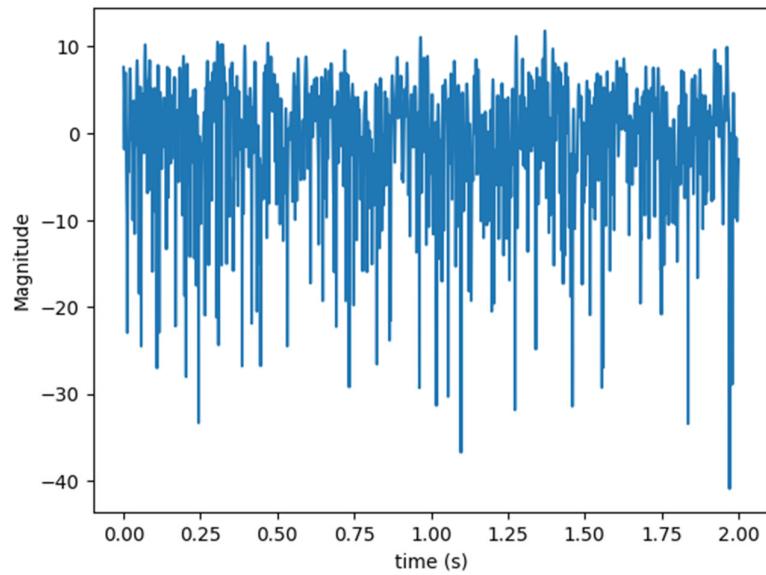


Figure 3-7: Magnitude of Complex Channel Coefficients

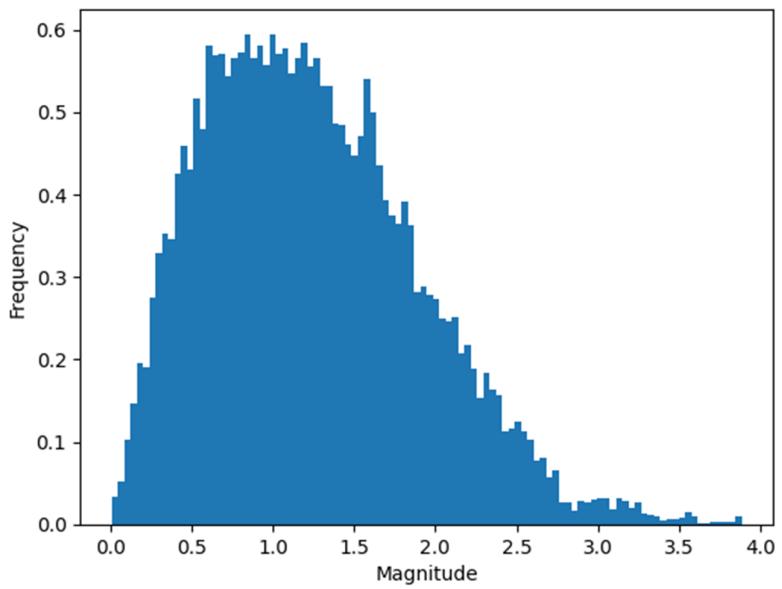


Figure 3-8: Histogram of Magnitude of Complex Channel Coefficients

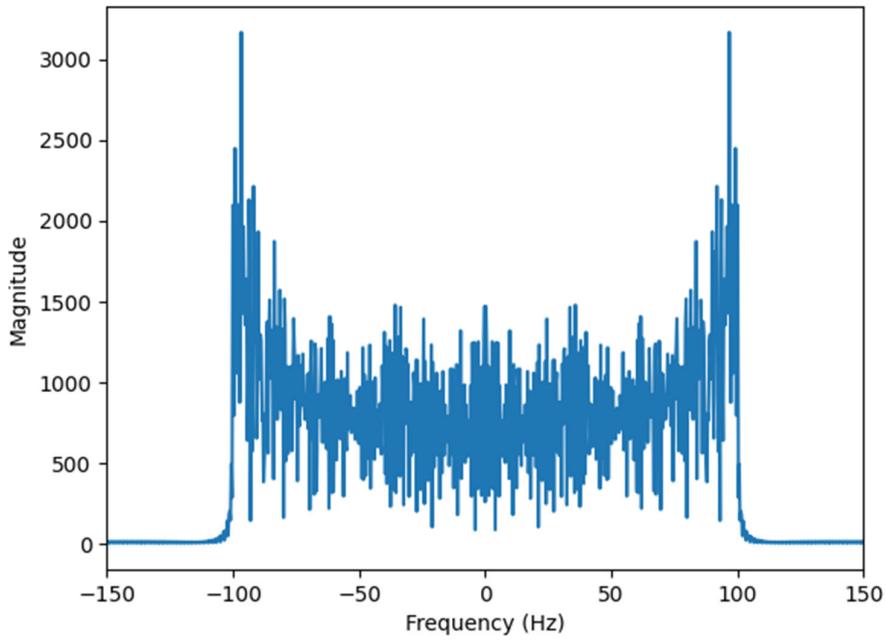


Figure 3-9: Doppler Power Spectrum

3.3.2 Channel with a Rician Process

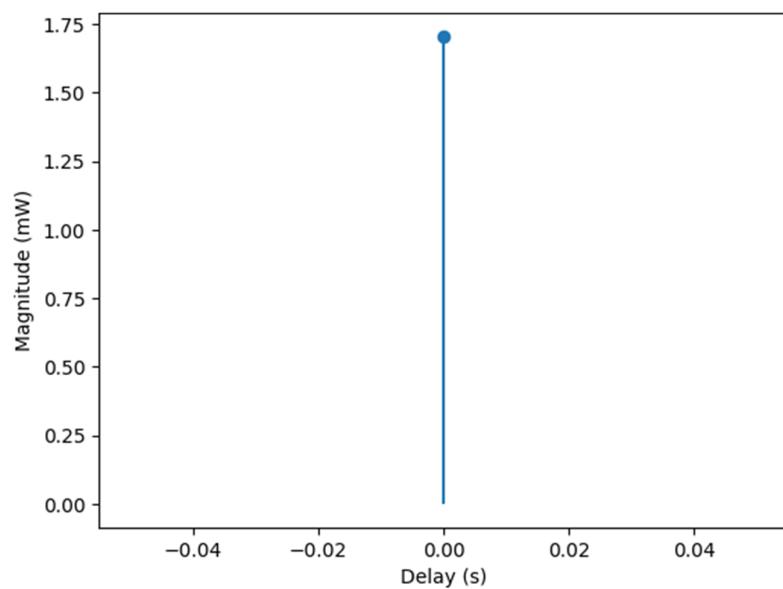


Figure 3-10: Channel Impulse Response:

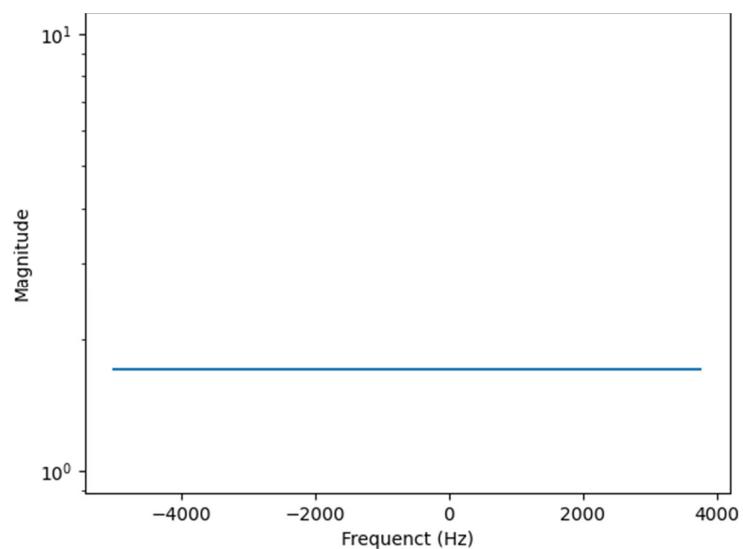


Figure 3-11: Channel Frequency Response

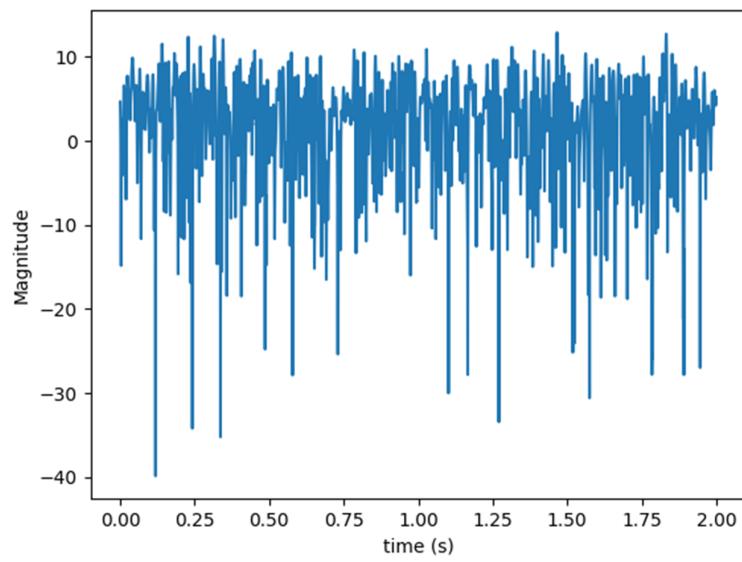


Figure 3-12: Magnitude of Complex Channel Coefficients

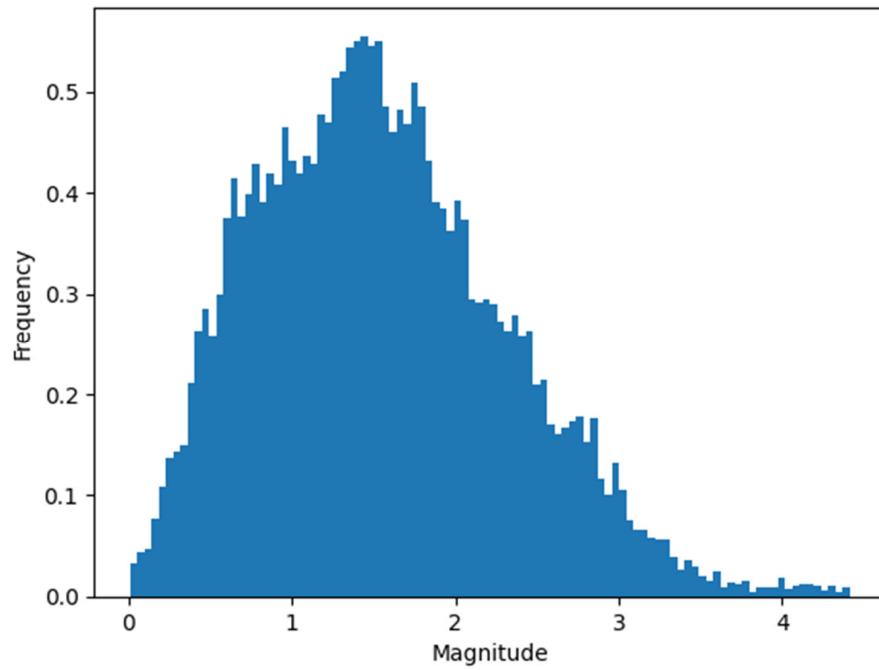


Figure 3-13: Histogram of Magnitude of Complex Channel Coefficients

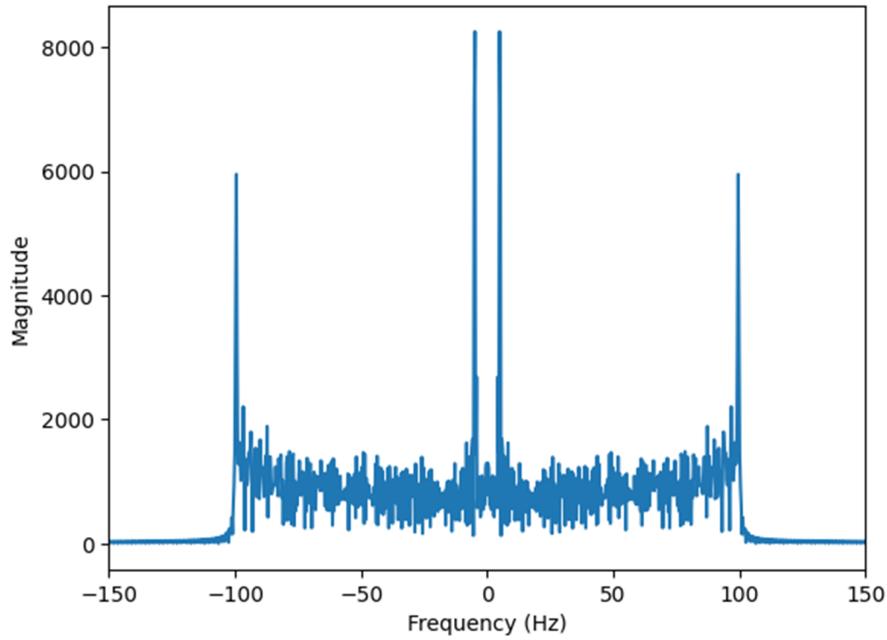


Figure 3-14: Doppler Power Spectrum

3.4 FREQUENCY-SELECTIVE CHANNEL WITH A JAKE'S DOPPLER SPECTRUM

The modeled channel had the following Power Delay Profile:

- Average Path Power in dBm: [0, -5, -10, -15, -20]
- Path Delays in seconds: [0.25, 0.5, 0.75, 1.0, 1.5]

3.4.1 Channel with a Rayleigh Process

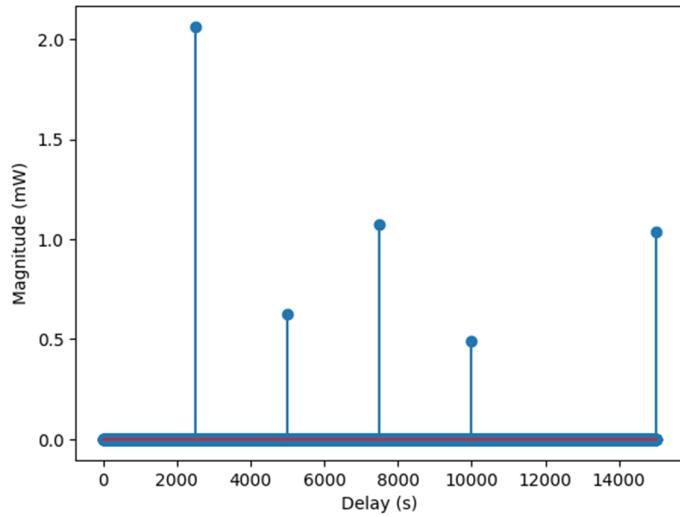


Figure 3-15: Channel Impulse Response

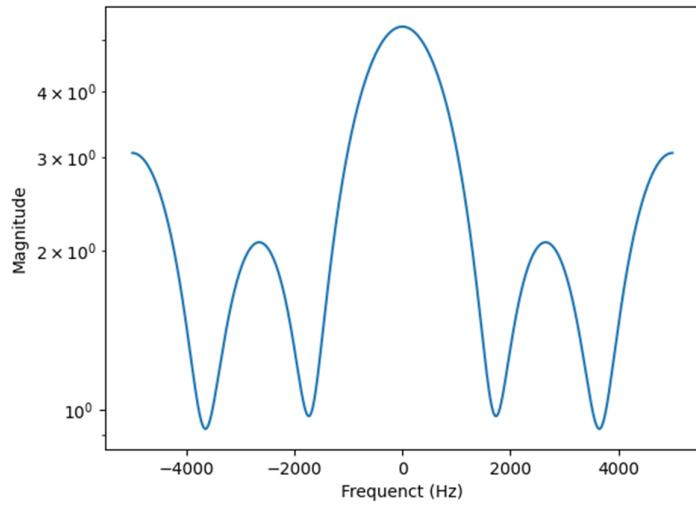


Figure 3-16: Channel Frequency Response

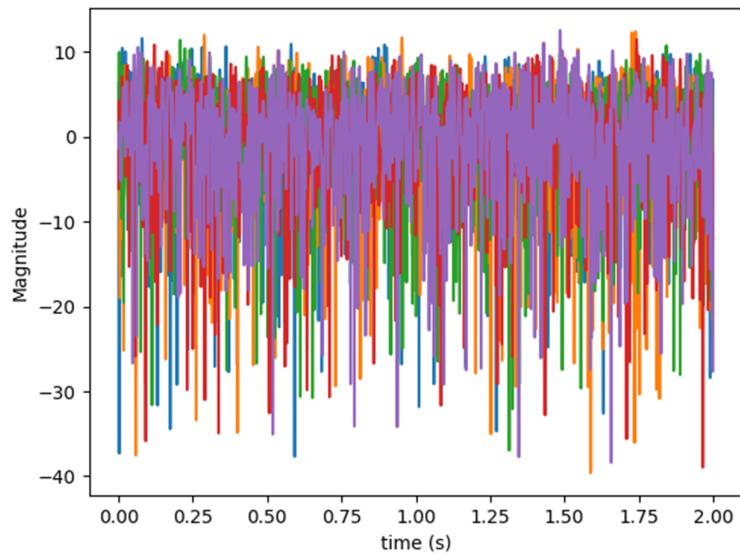


Figure 3-17: Magnitude of Complex Channel Coefficients

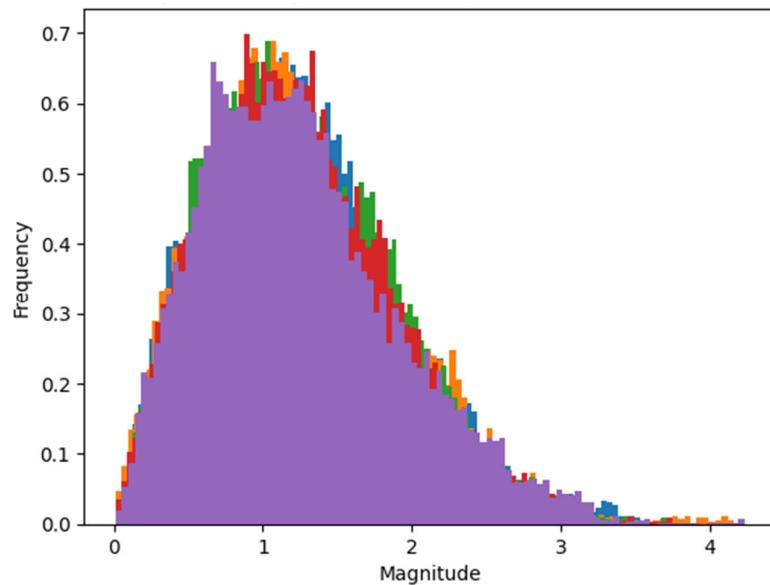


Figure 3-18: Histogram of Magnitude of Complex Channel Coefficients

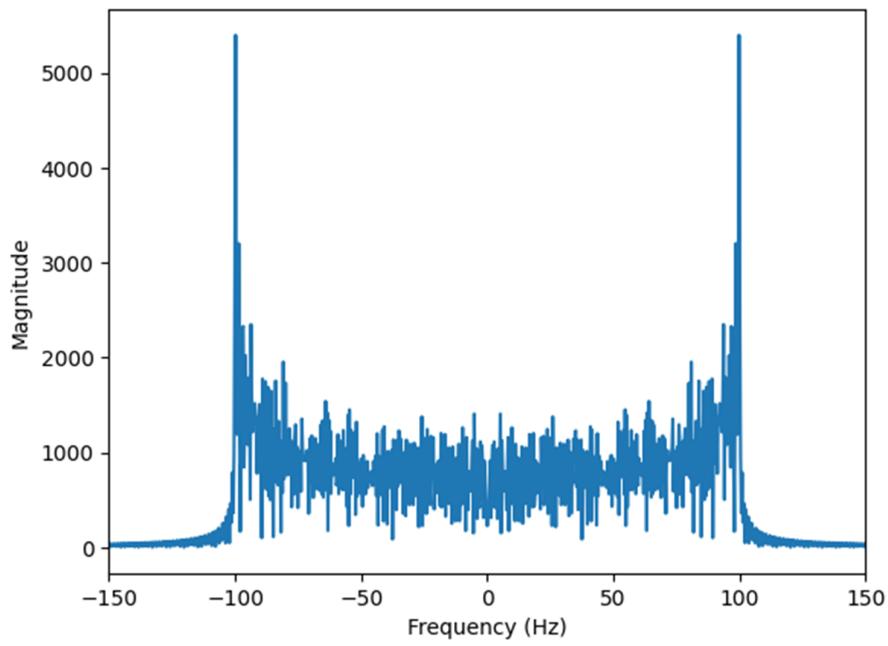


Figure 3-19: Doppler Power Spectrum

3.4.2 Channel with a Rician Process

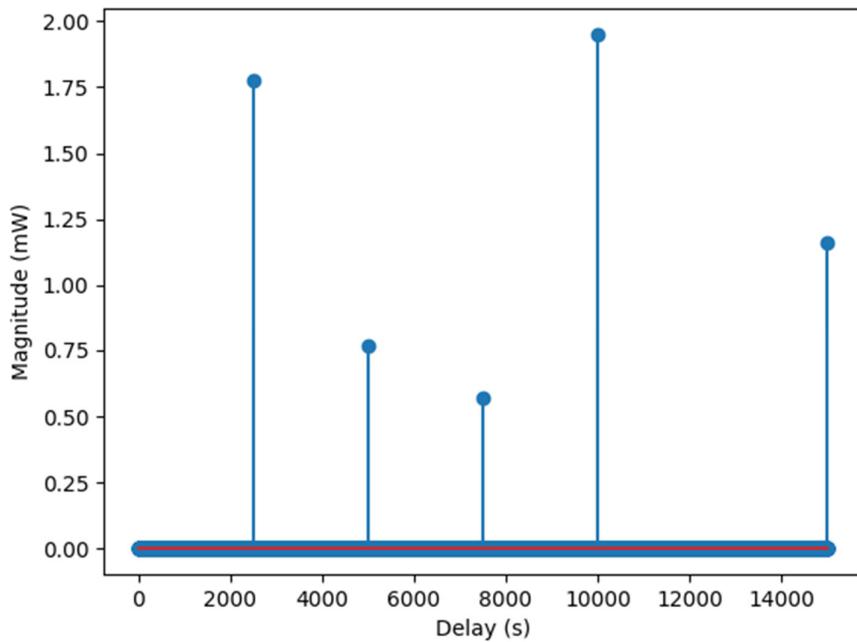


Figure 3-20: Channel Impulse Response

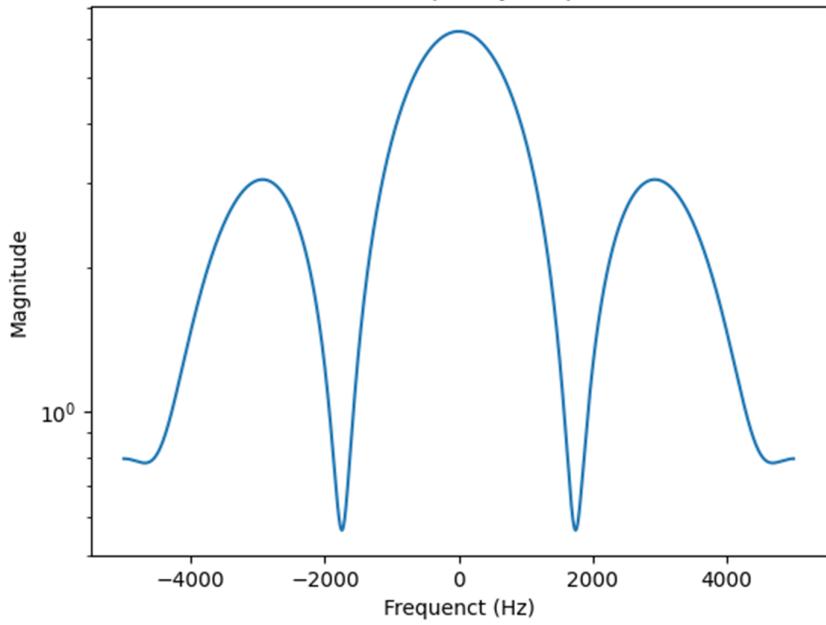


Figure 3-21: Channel Frequency Response

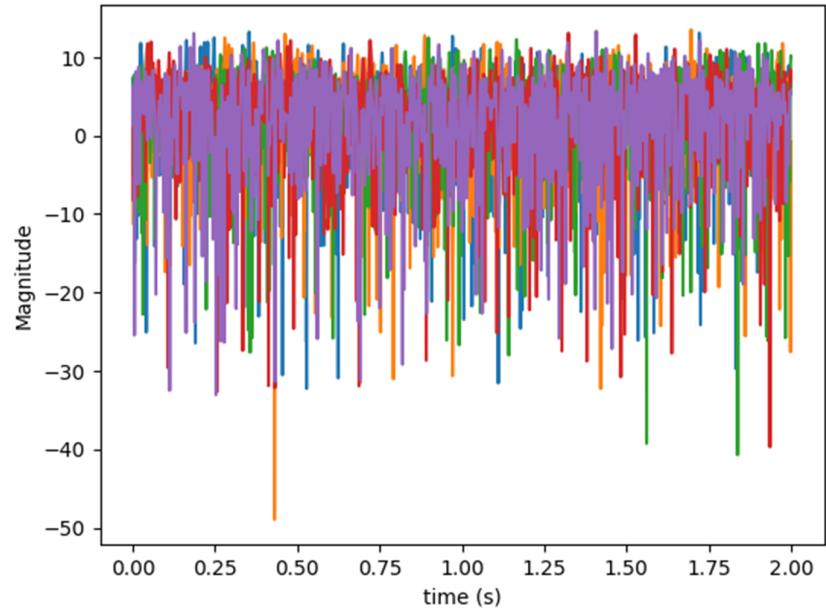


Figure 3-22: Magnitude of Complex Channel Coefficients

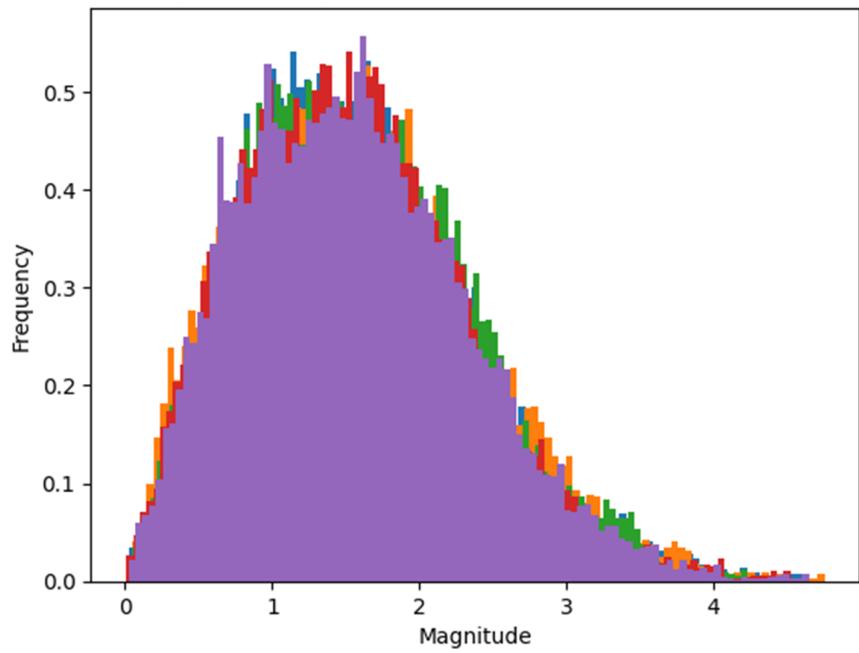


Figure 3-23: Histogram of Magnitude of Complex Channel Coefficients

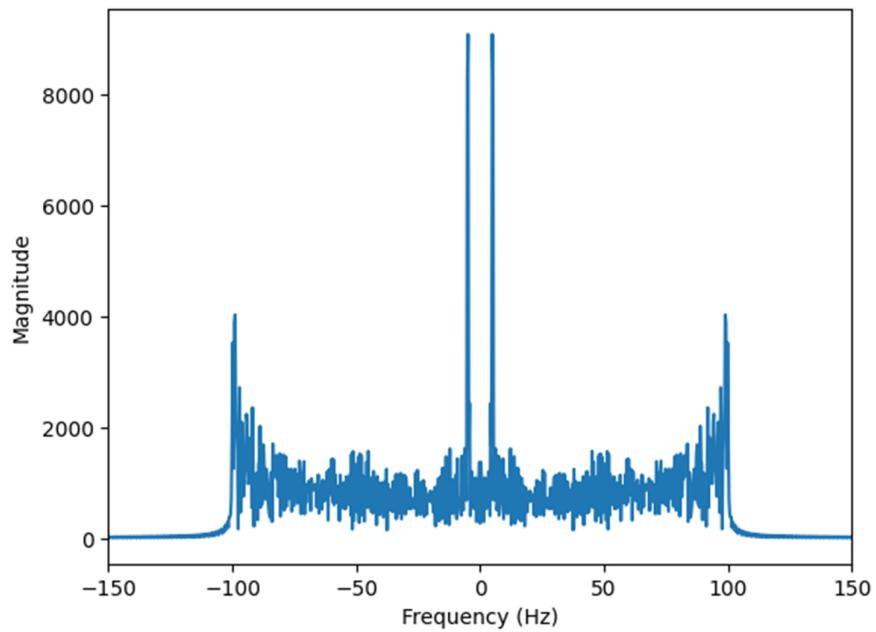


Figure 3-24: Doppler Power Spectrum

Our channel simulator has demonstrated remarkable accuracy in capturing the essential characteristics of wireless communication channels. Notably, it accurately replicates both the channel impulse response and the channel frequency response, crucial parameters for understanding the behaviour of the channel in time and frequency domains respectively. Furthermore, the simulator exhibits a precise spaced time correlation function, an essential aspect for analysing the temporal properties of the channel. The Doppler spectrum, another vital component, is also correctly represented, reflecting the simulator's ability to model the effect of relative motion between the transmitter and receiver. Significantly, for the Rician channel, the simulator effectively highlights the line-of-sight component, with a distinct peak observed at 5 Hz. This peak is a critical feature in Rician channels, indicating the presence of a strong direct path alongside the multipath components. Overall, the results from our simulator provide a comprehensive and accurate representation of real-world channel behaviours, making it an invaluable tool for the analysis and design of wireless communication systems

4 SYSTEM MODELLING RESULTS

4.1 THEORETICAL BER CURVES

The purpose of the system is to produce Bit Error Rate curves, and compare it to the theoretical BER curves of the modulation schemes tested to prove that the overall error rates decrease when we use OFDM based transmission

4.1.1 AWGN channels

Table 4-1: Equations used to generate the BER curves for the AWGN channel[16]

Modulation	Symbol error rate (P_s)
MPAM	$2 \left(1 - \frac{1}{M}\right) Q\left(\sqrt{\frac{6}{M^2-1} \gamma_s}\right)$
BPSK	$Q\left(\sqrt{2\gamma_b}\right)$
QPSK	$2Q\left(\sqrt{2\gamma_b}\right) - Q^2\left(\sqrt{2\gamma_b}\right)$
MPSK ($M > 4$)	$2Q\left[\sin\left(\frac{\pi}{M}\right)\sqrt{2\gamma_s}\right]$
MQAM	$1 - \left[1 - 2\left(1 - \frac{1}{\sqrt{M}}\right) Q\left(\sqrt{\frac{3\gamma_s}{(M-1)}}\right)\right]^2$

4.1.2 Rayleigh flat-fading channel

Table 4-2: Equations used to generate the BER curves for the Rayleigh channel [13]

$$\mathcal{M}_b\left(-\frac{g}{\sin^2\phi}\right) - \left(1 + \frac{g\tilde{\gamma}_s}{\sin^2\phi}\right)^{-1}$$

Modulation	Parameter g	Avg Prob. of Symbol error \bar{P}_s
BPSK	-	$0.5\left(1 - \sqrt{\frac{\gamma_s}{1+\tilde{\gamma}_s}}\right)$
MPSK	$\sin^2\left(\frac{\pi}{M}\right)$	$\frac{1}{\pi} \int_0^{\frac{(M-1)\pi}{M}} \mathcal{M}_b\left(-\frac{g}{\sin^2\phi}\right) d\phi$
MQAM	$\frac{1.5}{(M-1)}$	$\frac{4}{\pi} \left(1 - \frac{1}{\sqrt{M}}\right) \int_0^{\pi/2} \mathcal{M}_b\left(-\frac{g}{\sin^2\phi}\right) d\phi - \frac{4}{\pi} \left(1 - \frac{1}{\sqrt{M}}\right)^2 \int_0^{\pi/4} \mathcal{M}_b\left(-\frac{g}{\sin^2\phi}\right) d\phi$
MPAM	$\frac{3}{(M^2-1)}$	$\frac{2(M-1)}{M\pi} \int_0^{\pi/2} \mathcal{M}_b\left(-\frac{g}{\sin^2\phi}\right) d\phi$

4.2 SIMULATED BER CURVES

To plot the simulated BER and compare it with the theoretical BER we will need to define mainly 3 aspects:

1. Symbol to Noise Ratio
2. Modulation and Demodulation Scheme
3. Channel model

The channel model used is the channel simulator described in Chapter 3, we will use it here to apply noise or fading to the signal with the selected SNR for AWGN channels for example or channel parameters describing LOS and NLOS components for Rayleigh or Rician channels

And for the modulation and demodulation schemes we defined them in section 2.8.1.7 and 2.8.1.6

4.2.1 Symbol to Noise Ratio

First, we will define an array of the different Signal to Noise Ratios that we want to test the system on, to map the respective Signal to noise to Symbol to noise we need to take into account the modulation order used for the test.

Simply we will convert the number of bits used in the modulation order to the Signal to noise ratio we defined to get the Symbol to noise ratio, so for example for a Binary modulation order we add $10 \cdot \log(1)$ to the SNR, for a Quad order (we use 2 bits to represent all symbols) we add $10 \cdot \log(2)$ and so on [13]

4.2.2 MODEM Results

The results are calculated by dividing the total number of symbols with the correctly received symbols and plotting the SER for all modulation schemes under different channel conditions

4.2.2.1 AWGN channel

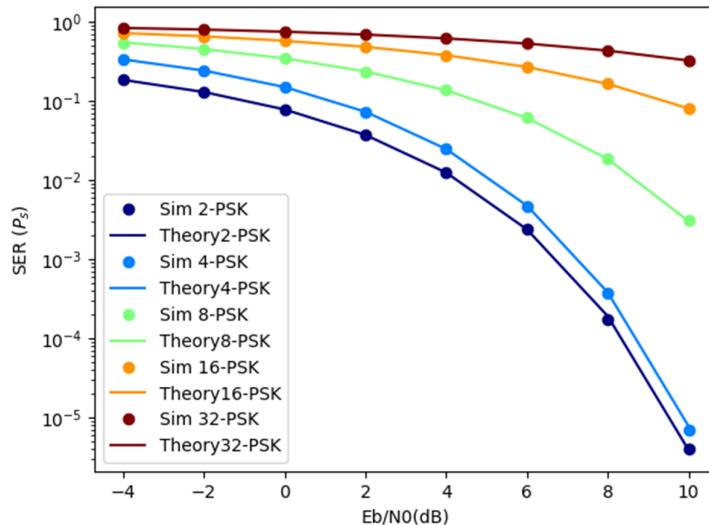


Figure 4-1: M-PSK AWGN

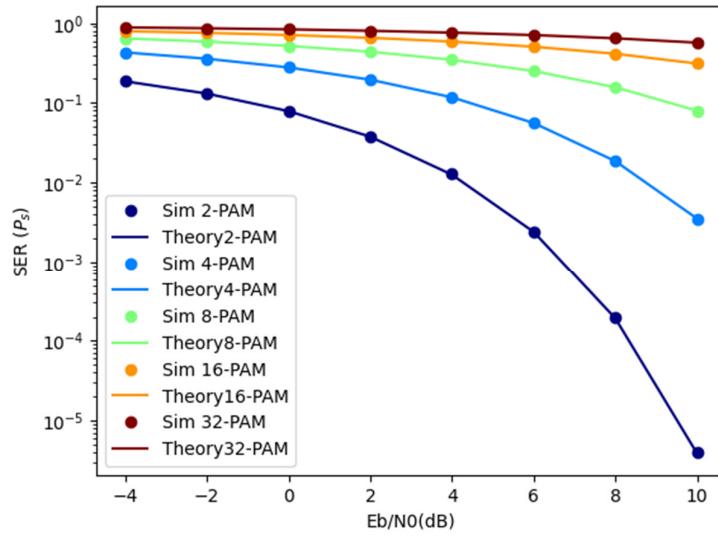


Figure 4-2: M-PAM AWGN

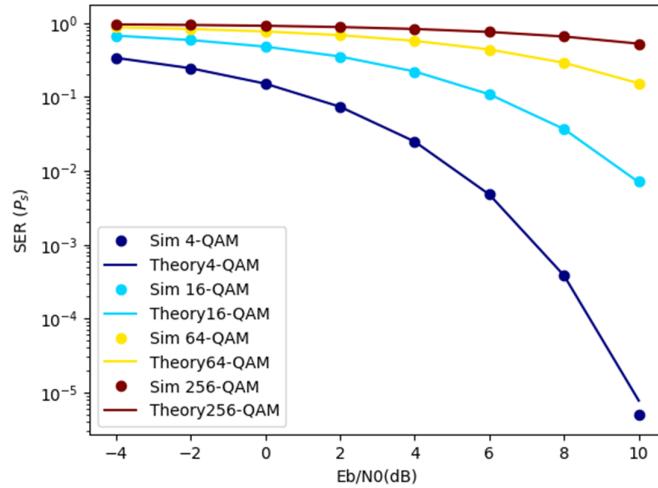


Figure 4-3: M-QAM AWGN

As we can observe here the simulated and theoretical curves align proving that the simulation matches the theoretical for AWGN channel in QAM, PAM and PSK modulation schemes

4.2.2.2 Rayleigh channel

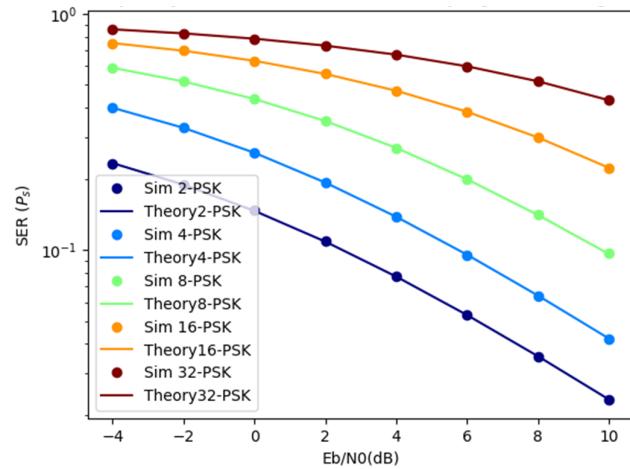


Figure 4-4: M-PSK Rayleigh

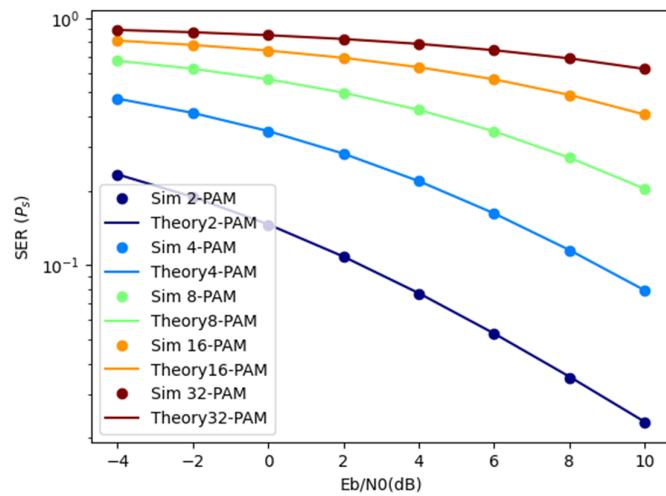


Figure 4-5: M-PAM Rayleigh

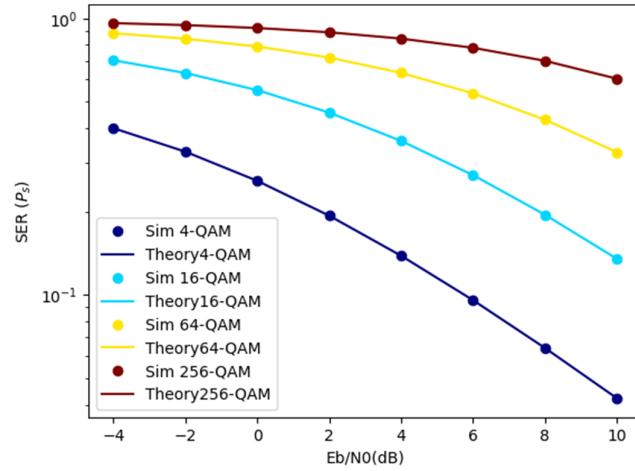


Figure 4-6: M-QAM Rayleigh

As we can observe here the simulated and theoretical curves align proving that the simulation matches the theoretical for Rayleigh Flat fading channel in QAM, PAM and PSK modulation schemes

4.2.3 Uncoded OFDM Results

The results below are done using ideal estimation and synchronization, we are only taking into consideration the channel non-ideality and are trying to overcome it by using an OFDM system with channel coding to correct relative errors.

4.2.3.1 AWGN channel

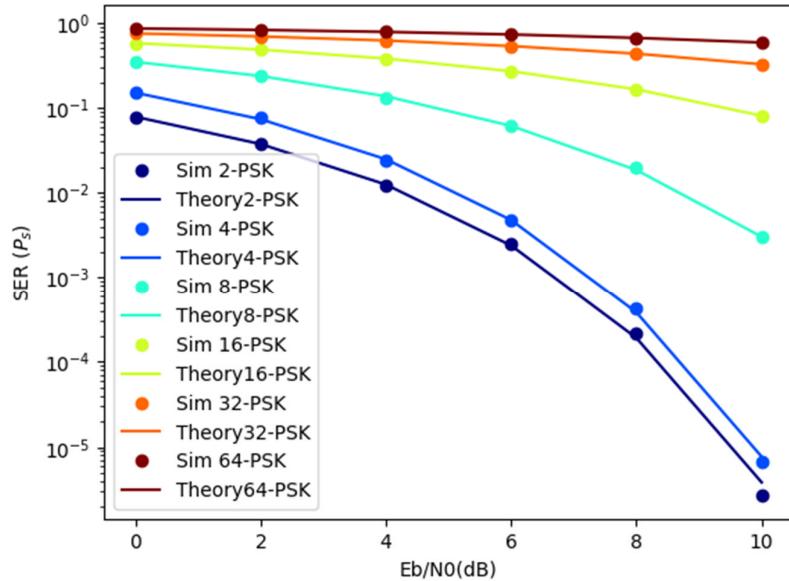


Figure 4-7: M-PSK CP-OFDM AWGN SER

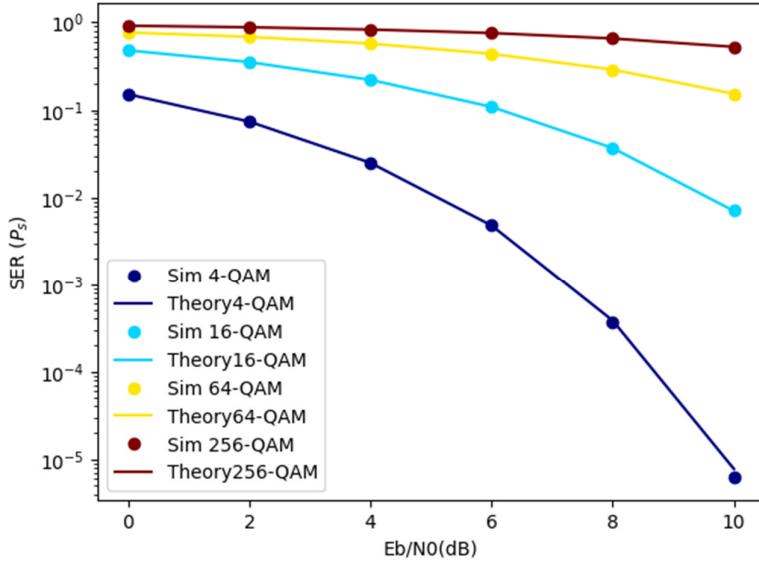


Figure 4-8: M-QAM CP-OFDM AWGN SER

In the scenario of an uncoded Orthogonal Frequency Division Multiplexing (OFDM) system operating in an Additive White Gaussian Noise (AWGN) channel, the results predominantly reflect the expected behaviour. As AWGN channels lack the multipath-induced fading effects present in frequency-selective channels, the inherent advantages of OFDM, particularly its ability to mitigate frequency-selective fading, are less pronounced. Consequently, the simulation outcomes typically align closely with the theoretical curves, revealing minimal performance improvements. Since OFDM is designed to excel in environments characterized by frequency-selective fading, where its subcarriers can adapt to varying channel conditions, the absence of fading in AWGN limits the potential for significant enhancements. In such scenarios, the focus often shifts to optimizing other aspects of the system, such as modulation and coding schemes, to maximize overall performance in the absence of fading-induced challenges.

4.2.3.2 Rayleigh Frequency Selective Channel

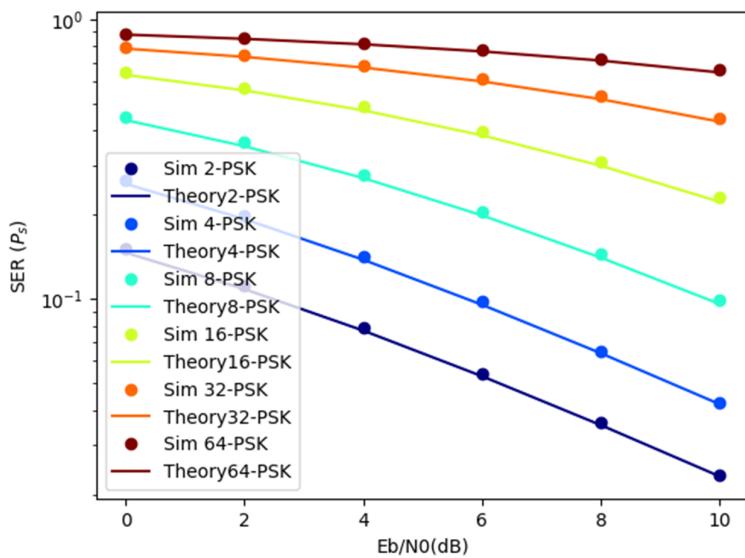


Figure 4-9: M-PSK CP-OFDM RAYLEIGH SER

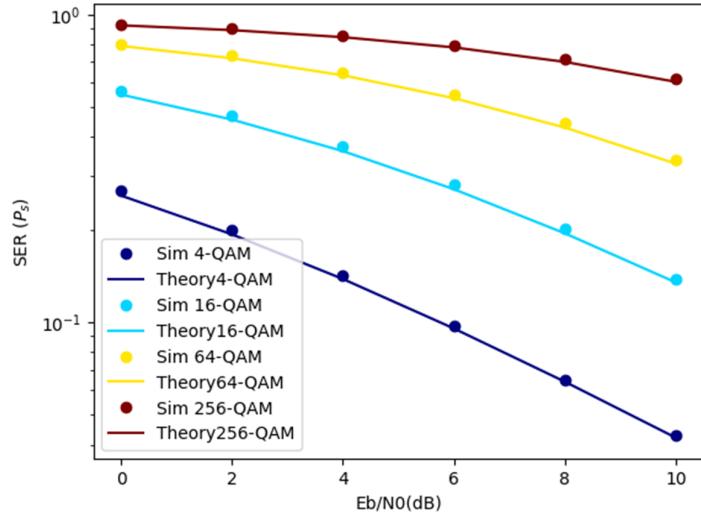


Figure 4-10: M-QAM CP-OFDM RAYLEIGH SER

The dotted curves represent the OFDM simulations under frequency selective channel, the theoretical curves are for a flat fading channel, the 2 curves overlap which means that the OFDM improves the data reception and mitigates the effect of the frequency selective channel as the available spectrum into multiple narrowband subcarriers. Since these subcarriers are spaced closely together, they experience different frequency-selective fading characteristics. Some subcarriers may experience deep fades due to the channel conditions, but others may not be as affected. This frequency diversity helps in combating selective fading.

: The use of cyclic prefix and guard intervals simplifies the equalization process in the receiver. The receiver can perform a simple frequency-domain equalization by dividing each received subcarrier by its channel gain, which helps in compensating for the frequency-selective fading effects.

4.2.4 Coded OFDM Results

The results below are done using ideal estimation and synchronization, we are only taking into consideration the channel non-ideality and are trying to overcome it by using an OFDM system with channel coding to correct relative errors. Here we included some error correction coding like convolutional encoder and Viterbi decoder, bit interleaving, and windowing

4.2.4.1 AWGN channel

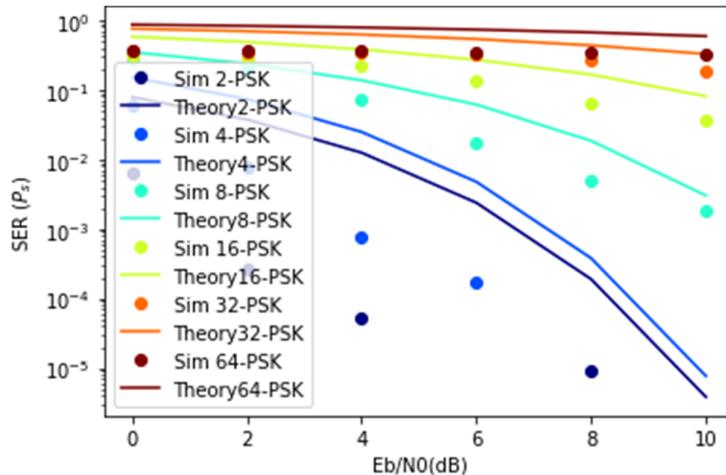


Figure 4-11: M-PSK Coded CP-OFDM AWGN SER

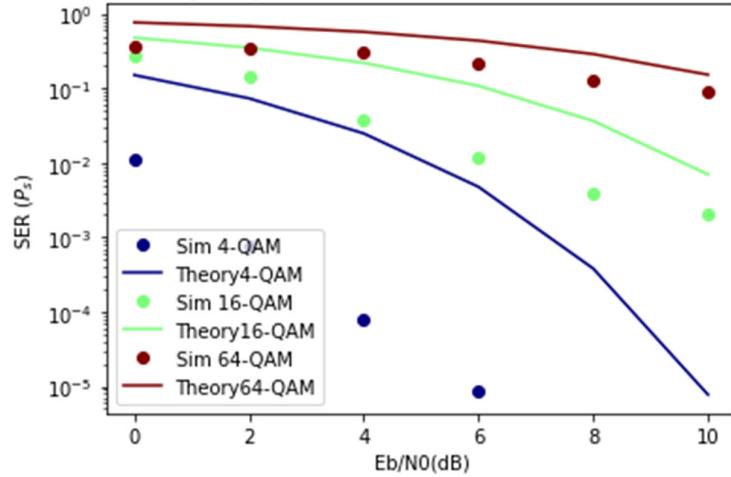


Figure 4-12: M-QAM Coded CP-OFDM AWGN SER

Coded Orthogonal Frequency Division Multiplexing (OFDM) significantly improves communication system performance, particularly in the presence of time-varying channels. By incorporating channel coding and interleaving techniques, the system gains the ability to mitigate burst and random effects induced by the channel's temporal variability. Channel coding introduces redundancy for error correction, making the transmission more robust against noise and interference. Meanwhile, interleaving rearranges the transmitted data, providing temporal diversity that helps combat burst errors by spreading them over time and subcarriers. This combined approach synergizes with OFDM's inherent advantages, such as frequency diversity and efficient spectrum utilization, creating a powerful solution that adapts dynamically to varying channel conditions and enhances overall system reliability.

4.2.4.2 Rayleigh Channel

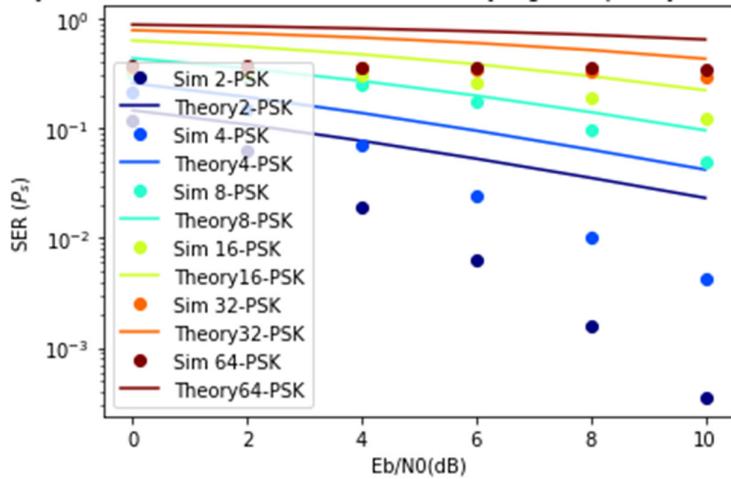


Figure 4-13: M-PSK Coded CP-OFDM RAYLEIGH SER

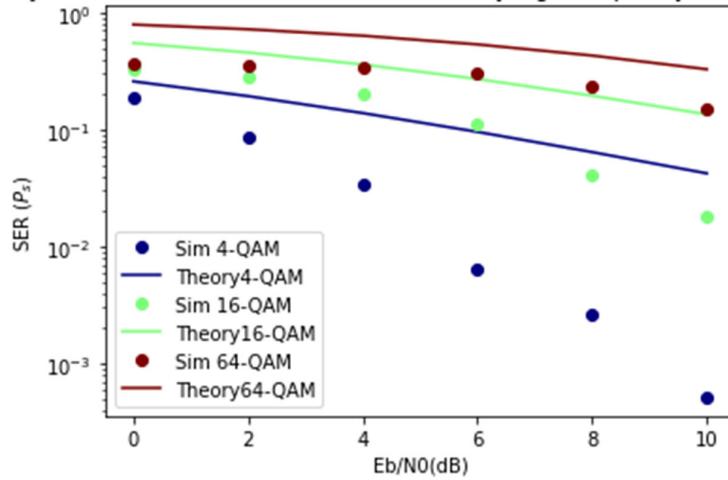


Figure 4-14: M-QAM Coded CP-OFDM RAYLEIGH SER

In the context of Rayleigh fading channels, where the multipath propagation induces fluctuations in signal amplitude and phase, the deployment of Coded Orthogonal Frequency Division Multiplexing (OFDM) is particularly advantageous. By introducing channel coding and interleaving, the system gains resilience against the challenges posed by Rayleigh fading. Additionally, the application of windowing techniques further enhances spectral efficiency. Windowing helps in reducing the interference caused by the abrupt truncation of OFDM symbols, improving the overall performance of the system. Notably, the impact of windowing extends beyond merely aligning theoretical flat fading curves with simulated frequency-selective fading; it also leads to an enhancement in spectral efficiency. The combined effect of windowing and channel coding in Coded OFDM proves instrumental in overcoming the impairments introduced by Rayleigh fading, ensuring robust communication in dynamic wireless environments.

4.3 CONCLUSION

From the findings presented in the preceding section, it is evident that a discernible shift exists between the theoretical and simulated Symbol Error Rates (SER). Remarkably, this shift is in a downward direction, signifying a reduction in error rates upon the implementation of the Orthogonal Frequency Division Multiplexing (OFDM) system. The introduction of OFDM serves as a countermeasure against errors induced by the channel.

In comparing the results obtained in Additive White Gaussian Noise (AWGN) scenarios with those in Rayleigh frequency-selective channels, it becomes apparent that the shift in SER is less pronounced in the former. This disparity can be attributed to the intrinsic suitability of the OFDM system for addressing frequency-selective fading. This adaptability arises from the integration of an interleaver and Forward Error Correction (FEC) encoder, both of which were previously highlighted. Additionally, the OFDM system effectively mitigates the multipath problem.

It is essential to note that while the theoretical curves for Rayleigh fading consider a flat fading channel, the channel utilized in the OFDM system is inherently frequency-selective. Despite this discrepancy, OFDM not only aligns with the theoretical curves but also demonstrates improved SER performance when confronted with frequency selectivity.

5 SYNCHRONIZATION RESULTS

5.1 SIMULATION RESULTS OF STO

Comparing STO estimation using CP of two methods: the squared difference and correlation.

The set of simulation values are:

$$[\text{STO}, \text{CFO}] = \{ [-3, 0], [-3, 0.5], [2, 0], [2, 0.5] \}$$

The results are as shown: The squared difference is shown in red in Figure 5-1 and correlation is shown in blue, the squared difference algorithm is minimum value based and correlation is maximum value based, this is clearly shown in the plots, at the true start position of the OFDM frame, the squared difference is minimum and it is maximum in correlation algorithm.

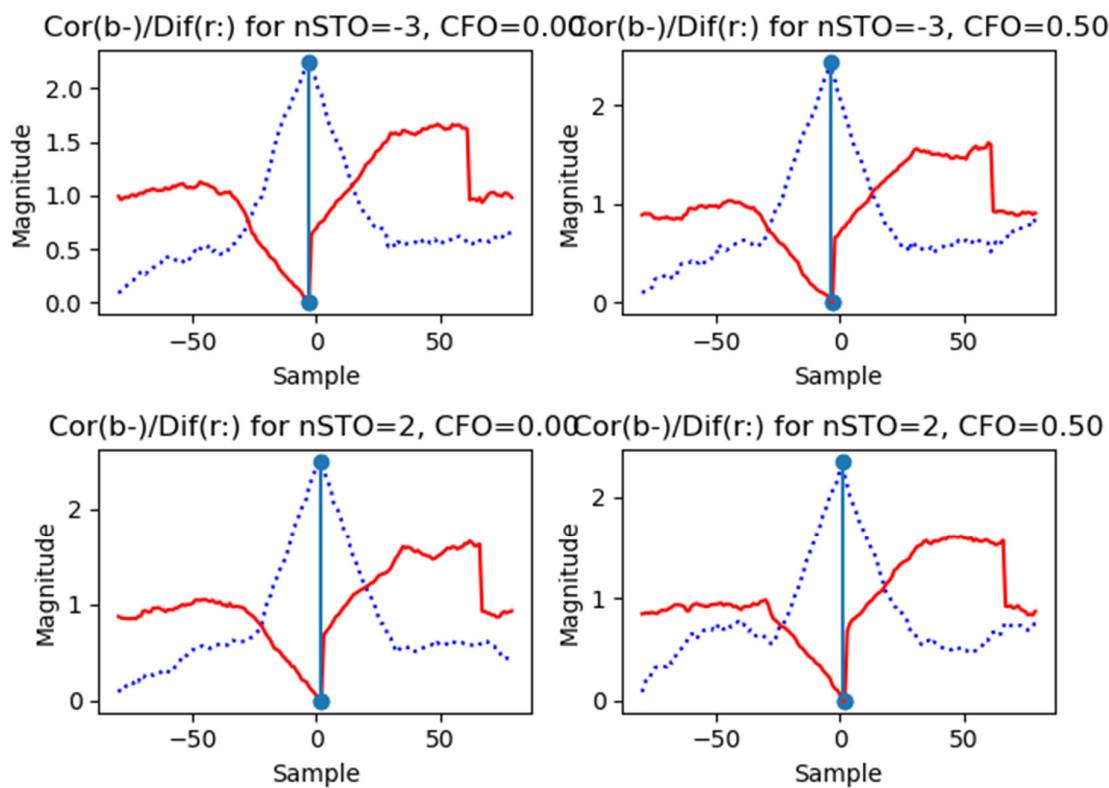


Figure 5-1: Simulation results of STO using squared difference method and Correlation method

The results of the algorithms are printed in the IDE console as shown in 5.1-3:

```
STO: -3 ,CFO: 0 By diff: -3 By Corr: -3
STO: -3 ,CFO: 0.5 By diff: -3 By Corr: -4
STO: 2 ,CFO: 0 By diff: 2 By Corr: 2
STO: 2 ,CFO: 0.5 By diff: 2 By Corr: 1
```

Figure 5-2: IDE console result of squared difference and correlation methods

5.2 ANALYSIS OF STO RESULTS

The results show that at CFO = 0.5, the Correlation algorithm made an error of one sample shift, when STO= -3, the estimated is -4 , when STO=2 , the estimated is 1.

This proves that the squared difference algorithm is more robust than correlation algorithm for CP STO estimation when CFO is found.

The plot in 5.2-1 is consistent with what is given the reference [17]:

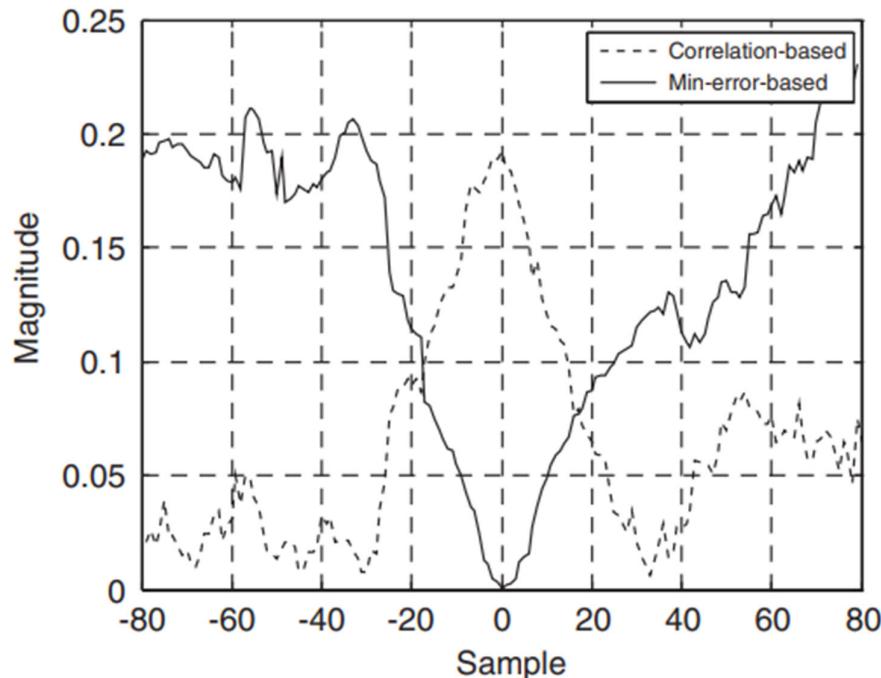


Figure 5-3: Squared Difference and Correlation STO estimation [9]

5.3 SIMULATION RESULTS OF CFO

The simulation compares the MSE of CP-Based technique, Moose (preamble based) and Classen (pilot based) at different SNR conditions, figure 5-4 shows the comparison.

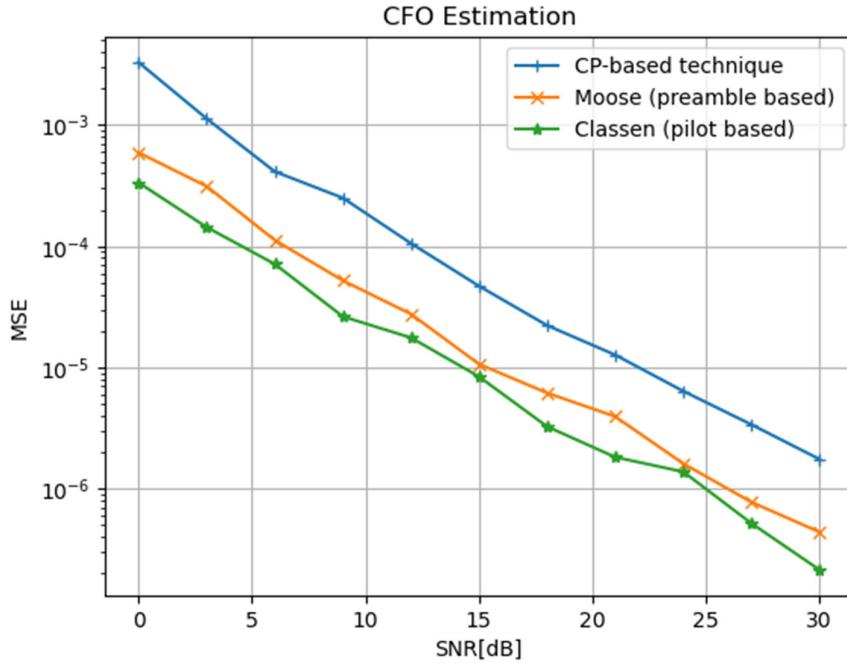


Figure 5-4: Simulation result of CFO different algorithms

5.4 ANALYSIS OF CFO RESULTS

The Classen (pilot-based) method introduces the minimum MSE, then Moose (preamble based), and the least is CP based technique.

Figure (5-7) shows that the simulated results are consistent with the results given in [17].

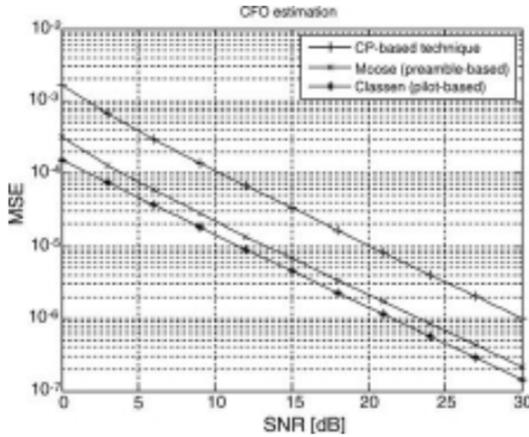


Figure 5-7: CFO estimation of different CP, Moose and Classen [9]

5.5 PACKET DETECTION SIMULATION RESULTS

As discussed, the expected value at the peak of the plateau is found by the formula:

$$\mu = E[(m_{peak})] = E \left[\left(\frac{c_{peak}}{p_{peak}} \right)^2 \right] = \left(\frac{S}{S+N} \right)^2 = \left(\frac{SNR}{1+SNR} \right)^2 \quad (5.5-1)$$

and the variance is found as:

$$\sigma^2 = \frac{2[(1+\mu)SNR + (1+2\mu)SNR^2]}{L(1+SNR)^4} \quad (5.5-2)$$

Where L is the window length.

A function is implemented that creates the short and long preambles of 802.11a connected with 1 data frame for simplicity, this function returns the samples in order to apply the detection and synchronization algorithms.

Figure 5-1 shows an example of running the auto correlation algorithm (also known as Schmidl & Cox synchronization technique) on 800-time sample where the frame is located at sample No. 100, the SNR environment is 10 db.

The window size is adjusted here to be 16 in order to catch the short training symbols.

Note: the function used in figure 5.5-1 is non-causal for simulation only.

A causal version is implemented, the same results are obtained but with a delay of 32 points.

The short training symbols are 160-time samples, the first 32 samples are the duration of filling the double sliding window, so the plateau length is $160-32 = 128$ samples, which are the samples starting at 100 to 228.

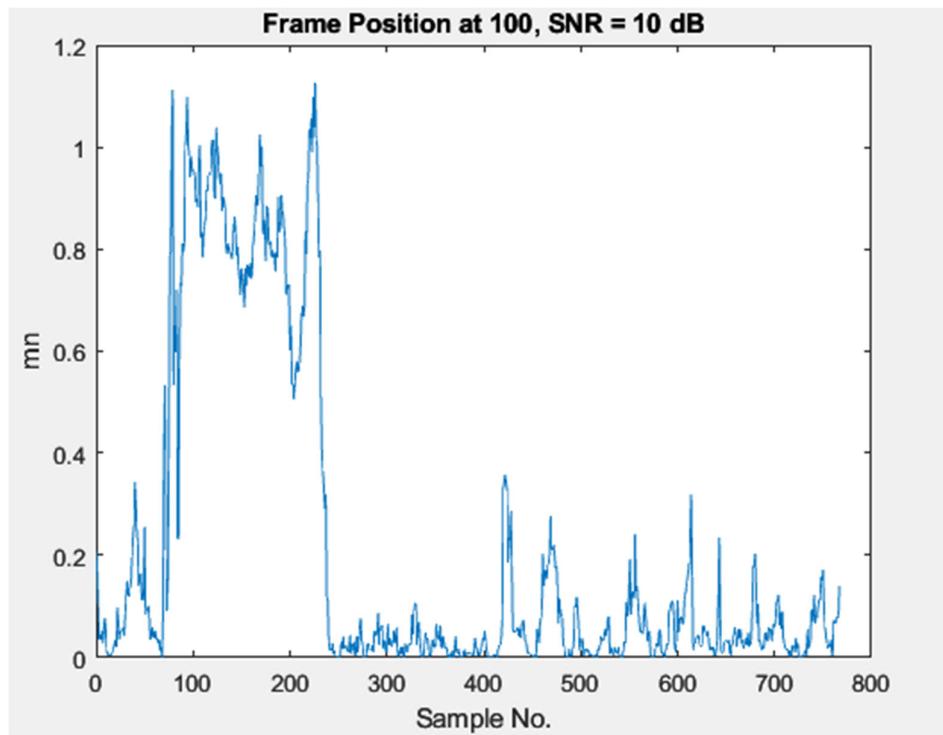


Figure 5-5: Running correlation detection (Schmidl & Cox) algorithm on 802.11a Short preamble symbols

A simulation function is made that calculates the mean and variance values of plateau according to the SNR and window length.

Using the WLAN 802.11a preamble, figure (5.5-2) and figure (5.5-3) show the expected values and standard deviation against different SNRs.

Figure (5.5-2) shows the values when the algorithm is targeting the short training symbols which require a window size of 16.

The plateau length in this case is 128 points.

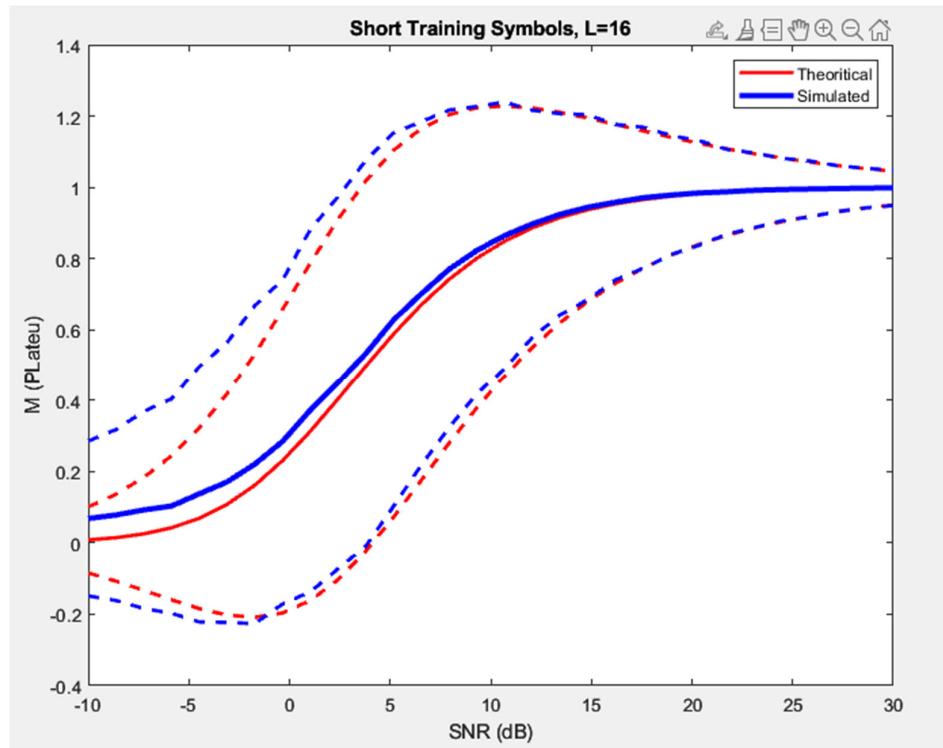


Figure 5-6: Theoretical vs Simulated results of Schmidl algorithm on Short Training symbols

figure (5.5-3) shows the values when the algorithm is targeting the long training symbols which require a window size of 64.

The plateau length in this case is 32 points.

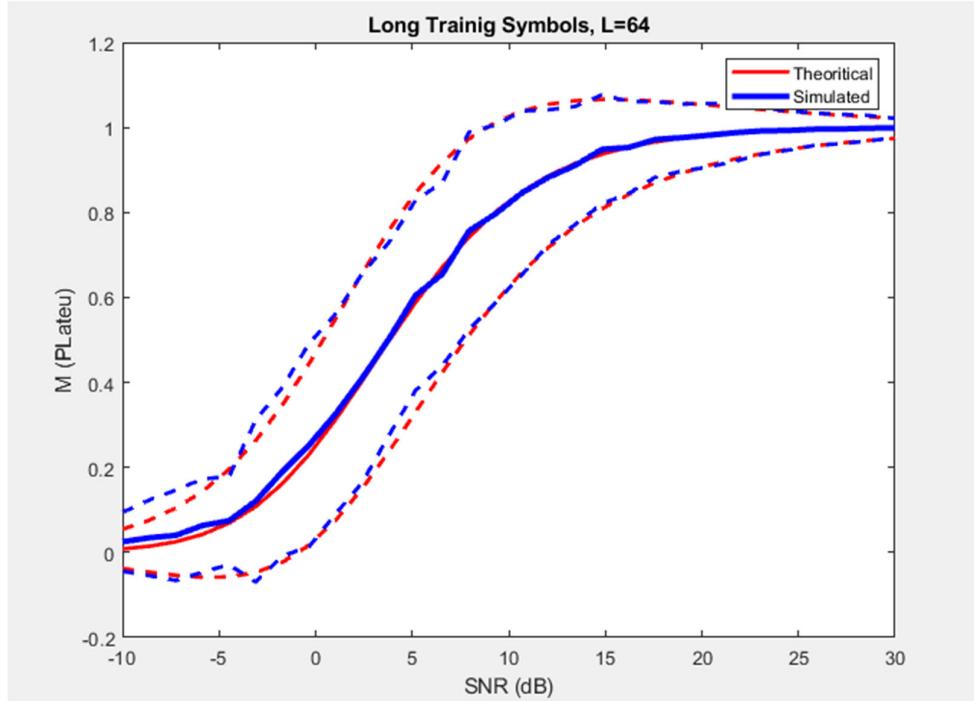


Figure 5-7: Theoretical vs Simulated results of Schmidl algorithm on Long training symbol

The plateau function result shown in Figure (5.5-7) is then passed to a decision function which will decide which a packet exists or not.

5.5.1 Decision of Packet Existence

Choosing a decision and choosing the values of threshold values for it is made by changing these values and deciding if the performance is acceptable.

The threshold I used is to set the threshold according to the SNR value, a low SNR value will lead to increasing the probability of detection due to the lowered threshold, but also the false alarm probability increases, so a low SNR value can be put at the start of operation of receiver then the receiver can adjust this value later after estimating the SNR value of the environment.

Then, 20 points in a row are required to cross the expected value of the value of (mean – 3*standard deviations). If a packet is detected to be found, 200 points will ahead of detection position will be blocked (their values will be zeros even if the decision condition is achieved), this number can be changed according to required performance.

5.5.1.1 Coarse Location Estimation Performance

By running the coarse packet location estimation decision, Figure (5.5-4) shows the estimated packet location after being detected for 100 iterations.

Zero represents that the decision predicted the exact location, negative numbers are before the actual location, positive numbers are after the actual location.

It can be shown that the max frequency is at -13 points with frequency of 16.

The rest of detections are around this value.

It can be also observed that there are no values detected after -5.

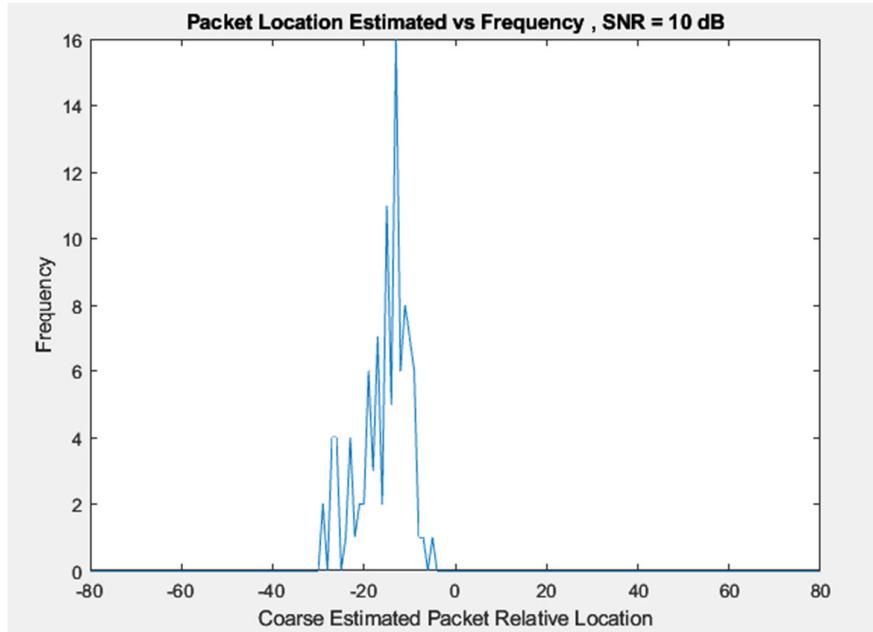


Figure 5-8: Coarse Estimation of Packet location after applying the decision

5.5.2 Estimating the Packet location, STO and CFO of Received Packet

After detecting that a packet exists and estimating its coarse position, it is needed to do a more precise locating, and carrier frequency offset estimation.

The approach we used is to:

1. Estimate the coarse packet location.
2. From the plateau values (values of $c(n)$) can estimate the CFO through the angle value), we can obtain the coarse CFO estimation value, the short preamble symbols can estimate a relative frequency offset between -2 and 2 (because the short symbol consists of 16 samples while OFDM symbol is 64). The values of long symbol samples are known and saved in the receiver.
3. Fix the coarse CFO for next samples.
4. The cross-correlation algorithm can be used against the long preamble symbols and take the position that makes the max correlation.
5. The fine CFO can be estimated from long preamble symbols with higher accuracy than short symbols.
6. Fix the fine CFO for the next coming samples.
7. Pass the value of estimated packet location to the rest blocks of receiver.

A function is implemented that generates 1000-time realizations and the probability of packet existence is 0.5 (it can be changed to another value).

Figure (5.5-5) shows one realization of 1000 that the decision made is that a packet exited at certain position (the packet actually is found at position 101).

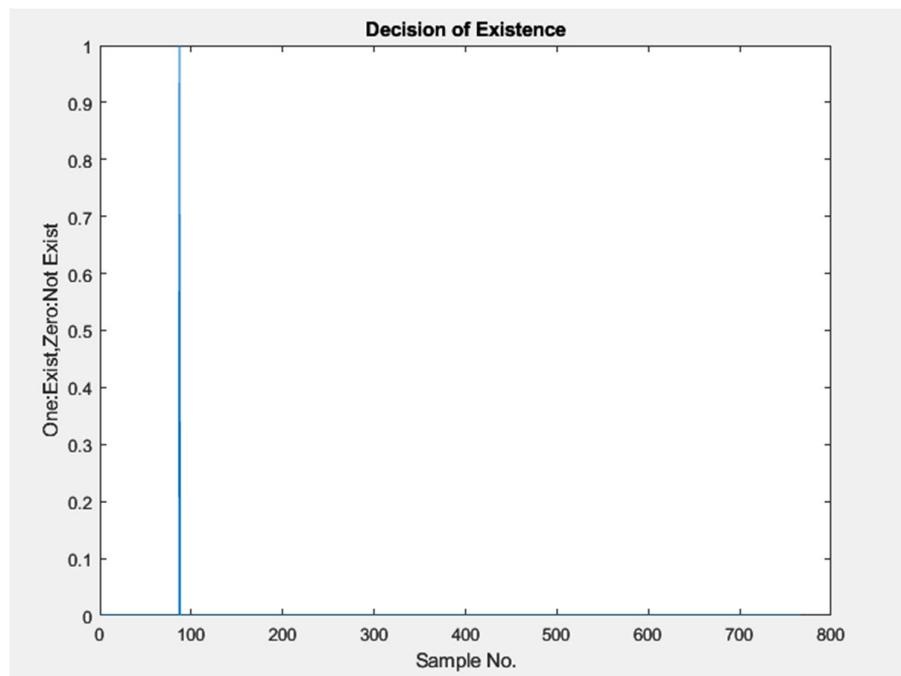


Figure 5-9: Decision of packet existence after applying decision

Figure (5.5-13) shows the cross-correlation mentioned in step 2, where a max value is clearly distinguished.

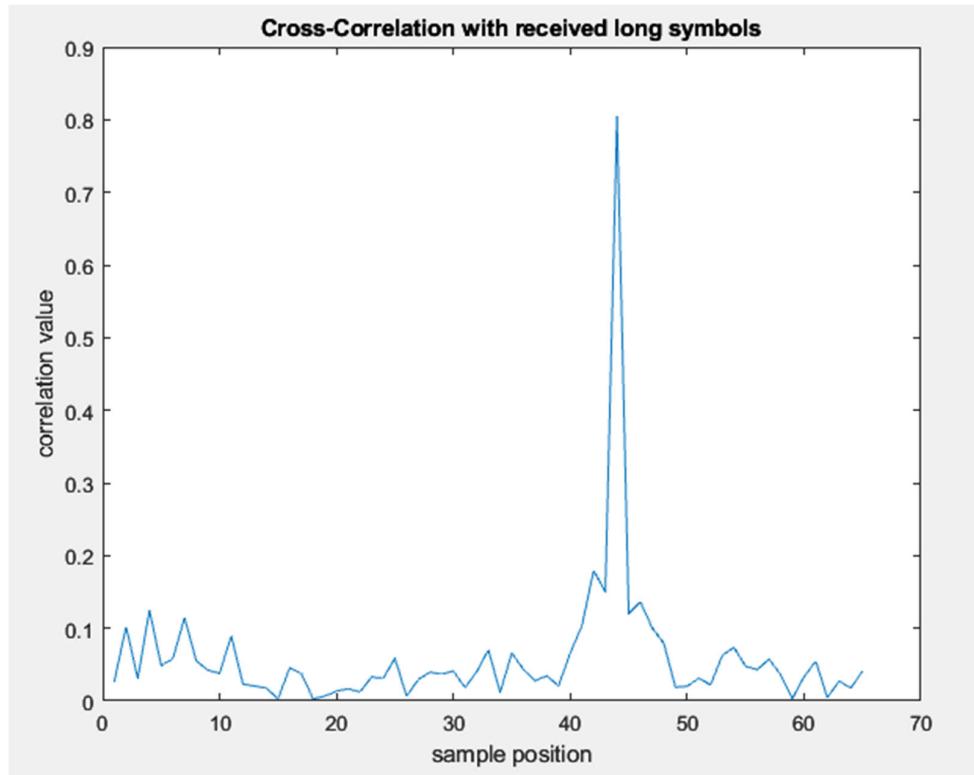


Figure 5-10: Running cross-correlation algorithm on long preamble symbols to find fine STO

Figure (5.5-7) and Table 5.5-1 show the performance criteria of detection algorithm used.

It can be used to know the quality and performance of the chosen decision methods and the threshold values used in it.

```
False_alarms =  
    0  
  
Packets_Missed =  
    0  
  
wrong_count =  
    1  
  
Detection_Within_range =  
    0
```

Figure 5-11: For 1000 packets (CFO=1.9, SNR=10dB), the False alarm, Packets missed, wrong packet location and Number of detections within 32 points

*Table 5.5-1: The performance of packet detection, STO, CFO algorithms used, according to 1000 packets
(CFO=1.9, SNR=10dB)*

	False Alarms	Packets Missed	Unacceptable STO	Acceptable STO
Count	0	0	1	0

5.6 CONCLUSION

OFDM's robustness against multipath fading and its efficient spectrum usage make it ideal for high-speed communication systems. However, accurate synchronization is critical. STO and CFO must be corrected to prevent inter-symbol interference and maintain subcarrier orthogonality. Packet detection is also crucial for demarcating the start of data frames.

The flow we used is to use auto-correlation algorithm to detect the existence of a packet, and use a decision of 20 points above a certain decision Threshold based on SNR as a decision, this also estimates the coarse packet location.

From the calculation of m_n (the parameter calculated by auto-correlation algorithm), the CFO can be calculated by using the values from c_n parameter, we used 16 points to give a better estimate of coarse CFO.

Then to fix the coarse CFO and run cross-correlation algorithm on long symbols to obtain fine STO.

And use the long symbols to have better and finer estimation of fine CFO.

This method gave a satisfying and acceptable performance of 2 mistakes out of 1000 packets at CFO=1.9 and SNR = 10 dB environment.

6 HARDWARE MODEL AND FUTURE WORK

The simulation methodology is adapted from [18]

6.1 ADVANCED TESTING AND VALIDATION

The simulations conducted in this thesis provide valuable insights into the theoretical behaviour of our OFDM system. However, the true test lies in its performance under real-world radio conditions. This chapter outlines two crucial future directions:

6.1.1 Software Defined Radio (SDR) Integration:

To validate our system's practical capabilities, we propose integrating it with a Software Defined Radio (SDR). This allows us to:

Field Testing: Transmit and receive OFDM signals in actual radio environments, capturing the complexities of multipath fading, interference, and channel dynamics.

Performance Evaluation: Measure key metrics like Bit Error Rate (BER) and throughput under diverse channel conditions, validating our theoretical predictions and revealing the system's adaptability.

Optimization and Refinement: Based on real-world data, we can refine synchronization algorithms, channel estimation techniques, and coding schemes for enhanced performance and robustness.

6.1.2 Technological Integration and Deployment:

Moving beyond the SDR integration, we envision deploying the OFDM system on a heterogeneous platform encompassing:

Hardware Acceleration: Utilize dedicated hardware, like Field-Programmable Gate Arrays (FPGAs), to accelerate computationally intensive tasks like FFT/IFFT and channel estimation. This significantly reduces processing delays and enhances real-time performance.

Deep Learning Integration: Harness the power of Deep Learning (DL) for improved channel estimation and modeling. Convolutional Neural Networks (CNNs) can learn complex channel behavior from real-world data, leading to:

Adaptability: CNNs can adapt to diverse channel conditions, overcoming the limitations of traditional estimation methods.

Reduced Complexity: DL-based estimation can achieve similar performance with lower computational requirements, improving efficiency.

Scalability: CNNs can efficiently process large datasets, paving the way for real-time applications and high-throughput communications.

6.2 HARDWARE IMPLEMENTATION AND EFFICIENCY:

Transitioning from software simulation to hardware implementation is a crucial step towards practical deployment. This chapter outlines the key aspects of this endeavour:

6.2.1 Baseband Signal Processing Hardware Implementation:

The initial focus will be on implementing the baseband signal processing blocks in hardware using Verilog and SystemVerilog Hardware Description Languages (HDL). Key blocks include:

- Scrambler
- Interleaver
- Encoder
- QAM Mapper
- OFDM Symbol Assembler
- Pilot Insertion
- FFT/IFFT
- Synchronization
- Decoder

Optimizations will be made at each stage to minimize resource utilization and maximize processing speed while ensuring accuracy and functionality.

6.2.2 Integration and Functional Verification:

The implemented blocks will be integrated and rigorously tested to ensure their functionality and proper interaction within the system. This verification process will involve:

Unit Tests: Individual blocks will be tested in isolation to verify their basic operation.

Integration Tests: The entire system will be tested as a whole to ensure seamless data flow and correct execution of all functionalities.

Performance Tests: The system's performance metrics like processing time, resource utilization, and throughput will be measured and analysed.

6.2.3 DSP Blocks and Deep Learning Integration:

The hardware implementation will be further enhanced by integrating the Deep Learning algorithm for channel estimation. This involves:

Hardware Acceleration: Utilizing Vivado's Deep Processing Unit (DPU) to accelerate the computationally intensive operations of the DL algorithm, significantly reducing processing time and improving real-time performance.

System Integration: Seamlessly integrating the DPU block with the existing hardware architecture to leverage the power of DL for channel estimation while maintaining overall system efficiency.

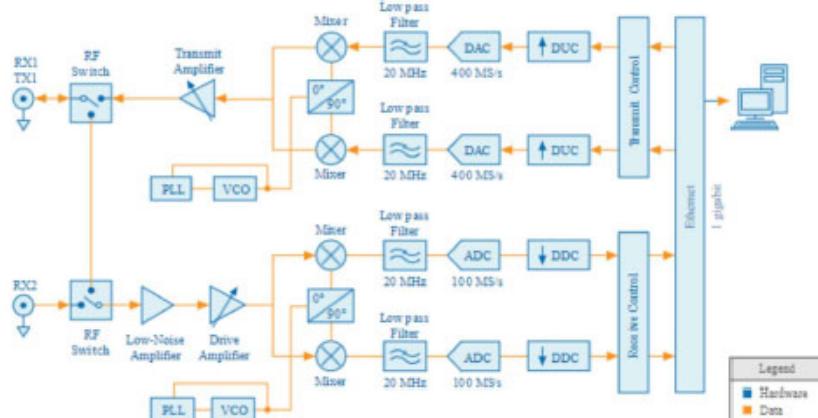


Figure 6-1 SDR integration

6.3 SDR INTEGRATION AND REAL-TIME OPERATION:

The final stage involves integrating the FPGA-based OFDM system with the SDR for real-time operation. This entails:

Interface Development: Establishing communication channels between the FPGA and the SDR using suitable protocols and interfaces like Gigabit Ethernet.

SDR Role: The SDR will act as a tunable RF transceiver, equipped with a high-speed ADC and DAC for real-time signal conversion and processing.

Software Integration: Developing the software framework for communication and control between the FPGA, SDR, and the host computer. This framework will enable data exchange, parameter configuration, and real-time performance monitoring.

Open-Source Software Options:

The SDR can be interfaced using various software options, such as:

LabVIEW and LabVIEW FPGA

C/C++

Python with open-source hardware drivers

Open-source SDR frameworks like USRP Hardware Driver (UHD) or GNU Radio

MathWorks MATLAB® software

Communication Interface:

To efficiently communicate with the SDR, the FPGA will require:

An Ethernet interface, as the SDR communicates using this protocol.

A processor to run the software framework for communication

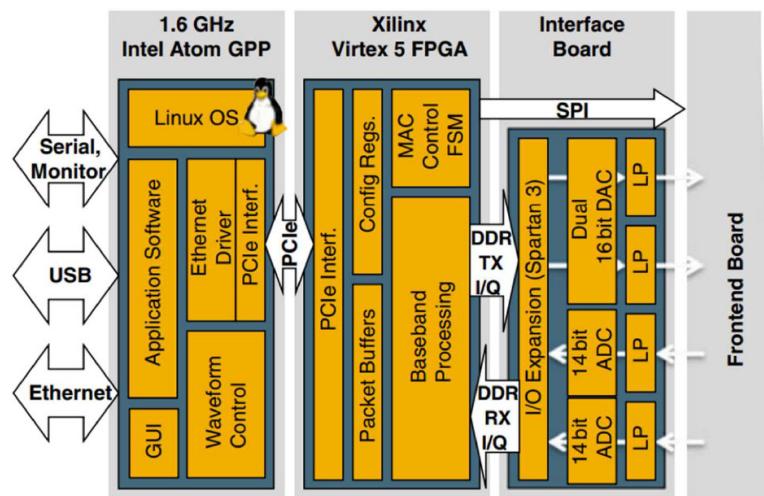


Figure 6-2 SDR & hardware interfaces

6.4 PROPOSED ARCHITECTURE

The central part of the following figure, represented by the FPGA component, operates as the core functionality within the system. The entire process involves a series of self-designed function blocks, with key components such as the IFFT/FFT, Encoder and Decoder, and PCIe Interface implemented as standard Xilinx IP cores.

Control signals, denoted by thin arrows, and data signals, illustrated with thick arrows, orchestrate the communication between various function blocks. The frontend configuration, encompassing the Automatic Gain

Control (AGC) and transceiver settings, is managed by simple Finite State Machines (FSMs) located at the top of the diagram.

The PCIe interface, positioned in the lower left part of the illustration, facilitates the exchange of video and control data through IP frames on the network layer. These IP frames undergo serialization into bitstreams within the High-Level Data Link Control (HDLC) blocks on the transmitter side (TX). Upon reception, similar HDLC blocks on the receiver side (RX) are responsible for de-serialization of the bitstreams.

Moving along the processing chain depicted from the HDLC blocks to the right-hand edge of the figure, a standard OFDM baseband processing chain unfolds. This processing chain, resembling conventional OFDM configurations found in literature, disregards the OFDMA aspect [22]. Within this chain, the bitstream undergoes mapping to I/Q constellation points in the frequency domain and is ultimately transformed to I/Q time-domain signals using an Inverse Fast Fourier Transform (IFFT). The overall structure elegantly encapsulates the intricacies of OFDM signal processing, combining self-designed elements with standard Xilinx IP cores to achieve a robust and efficient waveform processing system.

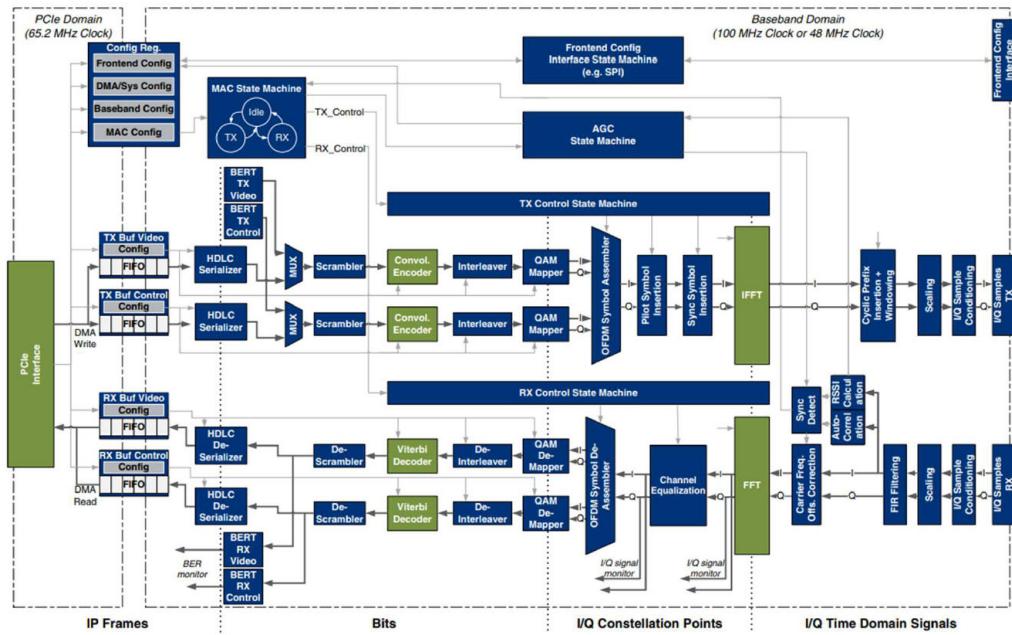


Figure 6-3: Proposed architecture

REFERENCES

1. Molisch, A.F., *Wireless communications*. 2nd ed. 2011, Chichester, West Sussex, U.K.: Wiley : IEEE. lvi, 827 p.
2. Cho, Y.S., et al., *MIMO-OFDM wireless communications with MATLAB*. 2010: John Wiley & Sons.
3. Simon, M.K. and M.-S. Alouini, *Digital communication over fading channels*. 2nd ed. Wiley series in telecommunications and signal processing. 2005, Hoboken, N.J.: Wiley-Interscience. xxxiv, 900 p.
4. Kim, J. and I.J.I.C.M. Lee, *802.11 WLAN: history and new enabling MIMO techniques for next generation standards*. 2015. **53**(3): p. 134-140.
5. Chiueh, T.-D. and P.-Y. Tsai, *OFDM baseband receiver design for wireless communications*. 2008: John Wiley & Sons.
6. Park, J.S., T.J.C. Ogunfunmi, Systems,, and S. Processing, *Efficient FPGA-based implementations of MIMO-OFDM physical layer*. 2012. **31**: p. 1487-1511.
7. *IEEE Standard for Information Technology--Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks--Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016), 2021.
8. Liu, Y., *Introduction to OFDM Receiver Design and Simulation*. 2019: Artech House.
9. Andrews, K., C. Heegard, and D. Kozen, *A theory of interleavers*. 1997, Cornell University.
10. Asghar, R. and D. Liu. *Low complexity hardware interleaver for MIMO-OFDM based wireless LAN*. in *2009 IEEE International Symposium on Circuits and Systems*. 2009. IEEE.
11. Zhang, Z.-d., et al. *Design and implementation of a multi-mode interleaver/deinterleaver for MIMO OFDM systems*. in *2009 IEEE 8th International Conference on ASIC*. 2009. IEEE.
12. Sanfuentes, J.L., *Software defined radio design for synchronization of 802.11 A receiver*. 2007, Monterey, California. Naval Postgraduate School.
13. Terry, J. and J. Heiskala, *OFDM wireless LANs: A theoretical and practical guide*. 2002: Sams publishing.
14. Huang, Y., L. Yuan, and W.J.A.S. Gong, *Research on IEEE 802.11 OFDM Packet Detection Algorithms for Household Wireless Sensor Communication*. 2022. **12**(14): p. 7232.
15. Michel C. Jeruchim , P.B., K. Sam Shanmugan, *Simulation of Communication Systems*. Springer New York, NY, 2000.
16. Goldsmith, A., *Wireless communications*. 2005, Cambridge ; New York: Cambridge University Press. xxviii, 644 p.
17. Cho, Y.S., *MIMO-OFDM wireless communications with MATLAB*. 2010, Singapore ; Hoboken, NJ: IEEE Press : J. Wiley & Sons (Asia). xiv, 439 p.
18. Blümm, C., C. Heller, and R. Weigel, *SDR OFDM Waveform Design for a UGV/UAV Communication Scenario*. Journal of Signal Processing Systems, 2012. **69**(1): p. 11-21.