TABLE OF CONTENTS

PAGE NO.

Certificate i

Candidate’s declaration ii

Acknowledgement iii

Abstract iv

CHAPTER 1 – INTRODUCTION 1-3

1.1 Introduction

1.2 Project Goals 1

1.2.1 Major Components and Functions of an OFDM System 2

1.2.2 Key Challenges in Wireless Communication Addressed by OFDM 2

1.3 OFDM: Significance and Applications 3

1.3.1 Elaborating on the Significance of OFDM 3

1.3.2 Exploring Various Applications of OFDM 4

CHAPTER 2 – BASICS OF OFDM 4-8

2.1 Introduction 5

2.2 Fundamentals of OFDM 5

2.3 OFDM Transmitter 5

2.3.1 Data Mapping and Subcarrier Assignment 7

2.3.2 Modulation Techniques in OFDM 7

2.3.3 Guard Interval Implementation 7

2.4 OFDM Receiver 8

2.4.1 Signal Demodulation and Symbol Recovery 8

2.4.2 Cyclic Prefix Removal and Channel Equalization 8

CHAPTER 3 – MATLAB IMPLEMENTATION 9-21

3.1 Introduction 9

3.1.1 Basic coding rules for MATLAB 9-11

3.1.2 Basic Operations 11-14

3.1.3 AMPLITUDE SHIFT KEYING 14

3.2 Transmitter Design in MATLAB 16

3.2.1 Data Generation 16

3.2.2 QPSK Modulation 17

3.2.3 Cyclic Prefix and Serial-to-Parallel Conversion 18-19

3.3 Receiver Design in MATLAB 20

3.3.1 Data Re-shaping 20

3.3.2 Extracting FFT 20

CHAPTER 4 – SIMULATION RESULTS 22-32

4.1 Performance Evaluation

CHAPTER 5 – CONCLUSION 33-36

5.1 Summary 33

5.2 Contributions 33

5.3 Future Directions 33

5.4 PAPR Reduction Techniques

5.4.1 Overview of PAPR in OFDM 34

5.4.2 Selected PAPR Reduction Techniques 34

5.4.3 Comparative Analysis of PAPR Reduction Techniques 35

5.4.4 Practical Implementation Challenges 35

5.4.5 Impact on OFDM System Performance 35

CHAPTER 1

INTRODUCTION

* 1. INTRODUCTION:

This project sets out to deeply investigate the intricacies of Orthogonal Frequency Division Multiplexing (OFDM), a key technology pivotal in modern communication systems. Its core principles, elucidating how it allocates and utilizes frequency spectrum resources to enable simultaneous data transmission. The primary objectives are aiming to unravel the underlying principles of OFDM, construct a fully operational OFDM system and evaluate its performance metrics. To achieve these goals, a step-by-step approach is adopted, beginning with a comprehensive study of the theoretical foundations of OFDM. By grasping the theoretical underpinnings, we lay a solid foundation for the subsequent practical implementation.

Moving forward, the project transitions into the realm of practical implementation, where the focus shifts to building both the transmitter and receiver components of the OFDM system using MATLAB. This phase involves not only coding the system but also simulating its behavior under various conditions. Through hands-on experimentation and simulation techniques, participants gain invaluable insights into the nuances of OFDM system development, acquiring skills in system design, coding, and simulation. Subsequently, the project embarks on an extensive performance evaluation journey, meticulously analyzing the efficacy of the implemented OFDM system. Key performance parameters, such as data transmission efficiency, signal integrity, and resilience to channel distortions, are scrutinized to gauge the system's effectiveness in real-world scenarios.

1.2 Project Goals

Orthogonal Frequency Division Multiplexing (OFDM) has become a cornerstone in modern communication systems, revolutionizing data transmission across various applications. Its significance lies in its unique ability to overcome key challenges associated with wireless communication, making it a preferred choice in contemporary scenarios.

OFDM serves as a robust solution to challenges such as multipath fading and frequency-selective channels. In conventional communication systems, these issues often result in signal distortion and data loss. OFDM, however, excels in mitigating these challenges by dividing the communication channel into orthogonal subcarriers. This division allows for simultaneous data transmission, enhancing the system's overall efficiency and reliability.

The transmitter plays a crucial role in the OFDM system by encoding and modulating the data before transmission. It breaks down the information into multiple subcarriers, each operating independently and concurrently. This parallel transmission capability significantly boosts data rates and improves overall spectral efficiency.

At the receiving end, the OFDM system employs a receiver responsible for demodulating the transmitted signal and reconstructing the original data. The receiver's ability to handle multiple subcarriers in parallel enhances its robustness and contributes to the system's resilience against channel impairments.

OFDM's effectiveness in addressing multipath fading and frequency-selective channels stems from its inherent capacity to manage interference. By distributing the data across multiple

subcarriers, it ensures that even if certain frequencies experience fading or interference, others remain unaffected. This diversity enables reliable and high-throughput communication, making OFDM particularly suitable for various wireless communication applications.

In the current landscape of wireless communication, where data rates and reliability are paramount, OFDM stands out as a key technology. Its adaptability to diverse channel conditions and its ability to provide high-data-rate communication make it indispensable for applications ranging from Wi-Fi networks to broadband communication systems.

By understanding the foundational principles and advantages of OFDM, this project aims to contribute to the optimization and advancement of communication systems, addressing the evolving needs of modern wireless communication.

1.2.1 Major Components and Functions of an OFDM System

We introduce the core components and functions that constitute an OFDM system:

Transmitter: The transmitter plays a crucial role in encoding data, modulating signals, and preparing the data for transmission. It encompasses processes such as data mapping, modulation, and the addition of guard intervals to mitigate issues like inter symbol interference.

Receiver: On the receiving end, the receiver demodulates the received signal and reconstructs the original data. This involves removing the cyclic prefix added during transmission and decoding the signal to retrieve the initial data.

Understanding these components and their functions lays the groundwork for a detailed  
exploration of OFDM system implementation and simulation in subsequent chapters.

1.2.2 Key Challenges in Wireless Communication Addressed by OFDM

Highlighting the practical relevance of OFDM, we discuss challenges commonly encountered in wireless communication and how OFDM addresses them:

Multipath Fading: OFDM mitigates the effects of multipath fading by dividing the channel into multiple subcarriers. This approach ensures that even if some subcarriers experience fading, others remain unaffected, contributing to overall signal reliability.

Frequency-Selective Channels: OFDM's ability to divide the channel into orthogonal subcarriers helps combat frequency-selective channels. By distributing data across a range of frequencies, OFDM ensures that even if certain frequencies are attenuated, others can still convey information effectively.

Understanding these challenges and the corresponding solutions provided by OFDM establishes the necessity and relevance of this technology in modern wireless communication scenarios.

1.3 OFDM: Significance and Applications

Orthogonal Frequency Division Multiplexing (OFDM) stands as a cornerstone in modern communication systems, and its significance is deeply rooted in its distinctive features, contributing to its widespread adoption across various domains. This section delves into the multifaceted importance of OFDM, highlighting its unique attributes and diverse applications.

OFDM's prowess in mitigating multipath fading and efficiently handling channel impairments sets it apart. By dividing the communication channel into orthogonal subcarriers, OFDM minimizes the impact of fading, ensuring a more robust and reliable signal transmission. Additionally, its resistance to channel impairments enhances the overall quality of communication.

The efficient utilization of available bandwidth is a key attribute of OFDM. Through the simultaneous transmission of multiple subcarriers, each occupying a narrow frequency band, OFDM maximizes spectral efficiency. This capability is particularly crucial in scenarios where bandwidth is a limited and valuable resource.

OFDM finds application across a spectrum of domains, underscoring its versatility and adaptability. In wireless local area networks (WLANs), OFDM enables high-speed data transmission, providing a seamless and reliable network experience. Furthermore, its utilization in broadband communication over existing infrastructure showcases its capability to enhance communication efficiency on a larger scale.

As technology advances, OFDM continues to play a pivotal role in emerging domains such as the Internet of Things (IoT). Its ability to handle diverse communication requirements makes it well-suited for the complex and dynamic connectivity needs of IoT devices. OFDM's resilience and efficiency contribute to the seamless integration of IoT into the fabric of modern communication systems.

The exploration of various applications serves to emphasize how OFDM addresses real-world communication challenges. Whether it's ensuring high-speed data transfer, improving network reliability, or facilitating connectivity in emerging technologies, OFDM consistently proves its practical relevance.

In essence, this section aims to unravel the layers of OFDM's importance, shedding light on its unique attributes and demonstrating how its practical applications impact and enhance communication across different domains.

1.3.1 Elaborating on the Significance of OFDM

Expanding on the significance of OFDM, we explore its unique attributes that make it a cornerstone in modern communication systems:

Mitigating Multipath Fading: OFDM's ability to handle multipath fading ensures robust communication in environments with reflective surfaces, where signals may take multiple paths before reaching the receiver.

Efficient Bandwidth Utilization: OFDM efficiently utilizes available bandwidth by dividing it into orthogonal subcarriers, allowing for simultaneous transmission and increasing data rates.

Resilience to Channel Impairments: OFDM's resilience to channel impairments, including multipath propagation and frequency-selective fading, positions it as a reliable modulation scheme in the face of real-world communication challenges.

1.3.2 Exploring Various Applications of OFDM

In this section, we delve into the diverse applications of Orthogonal Frequency Division Multiplexing (OFDM) across different domains, elucidating how its unique attributes contribute to elevating communication efficiency and reliability.

OFDM plays a pivotal role in Wireless Local Area Networks (WLANs), addressing the escalating demand for high-speed and reliable data transmission in varied environments. By employing OFDM in WLANs, seamless connectivity is achieved, providing users with a robust and efficient network experience. The ability of OFDM to combat signal distortions and fading ensures consistent and high-quality data transfer in WLAN settings.

OFDM emerges as a key player in the realm of broadband communication, showcasing its proficiency in delivering high data rates over existing infrastructure. The efficient use of available bandwidth through the orthogonal subcarrier scheme allows OFDM to maximize spectral efficiency, making it integral to the success of modern communication networks. Its application in broadband communication is a testament to its adaptability and effectiveness in optimizing data transfer.

The relevance of OFDM extends into emerging technologies, particularly the Internet of Things (IoT). In the IoT landscape, where diverse devices demand reliable communication with an efficient use of resources, OFDM finds application. The ability of OFDM to handle multiple connections, resist interference, and ensure robust communication aligns seamlessly with the dynamic and varied requirements of IoT devices.

By comprehensively understanding the significance and varied applications of OFDM, we gain valuable insights into its versatility and relevance across diverse communication scenarios. This knowledge serves as a foundation for the subsequent chapters, where we embark on the journey of implementing and evaluating an OFDM system.

CHAPTER 2

BASICS OF OFDM

2.1 Introduction

We shall examine in-depth the basic ideas underlying orthogonal frequency division multiplexing (OFDM) in this chapter. The introduction establishes the scene by highlighting OFDM's crucial function in contemporary communication systems. It restates the difficulties in wireless communication and offers justification for the revolutionary solution of adopting OFDM. The goal as we go into the fundamentals is to provide readers with a thorough grasp of the theoretical underpinnings of OFDM.

Readers are given insight into the evolutionary process that led to OFDM's establishment as a cornerstone in modern communication systems by following its development and emphasizing significant turning points. This historical context presents OFDM as a solution in addition to providing depth to our understanding of it.

2.2 Fundamentals of OFDM

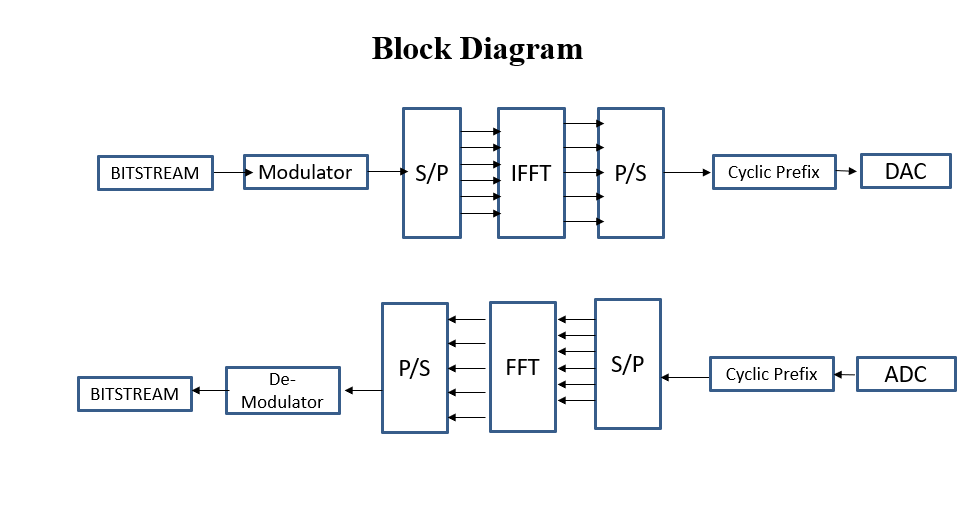
Expanding on the fundamentals, we delve deeper into the theoretical foundations of OFDM. This section provides a nuanced exploration of the orthogonal frequency division and multiplexing principles. By elucidating how OFDM achieves orthogonality among its subcarriers, we demystify its efficiency in data transmission. The discussion includes the mathematical underpinnings that govern the orthogonality of subcarriers, offering readers a more rigorous grasp of the mechanisms at play. This deeper dive into the theoretical intricacies enhances the reader's ability to appreciate the elegance and robustness of OFDM in addressing communication challenges.

2.3 OFDM Transmitter

In this section, we navigate through the intricate workings of the Orthogonal Frequency Division Multiplexing (OFDM) transmitter. The transmitter is the nerve center of the OFDM system, responsible for encoding data, modulating signals, and preparing the information for transmission. We delve into the foundational processes, starting with data mapping, where information is associated with specific subcarriers. The modulation step introduces the data to the carrier frequencies, ensuring efficient transmission. A critical aspect addressed here is the addition of guard intervals, a safeguard mechanism against intersymbol interference. By examining each of these processes meticulously, readers gain a nuanced understanding of how the OFDM transmitter optimally readies data for seamless and robust transmission.

Furthermore, the section explores practical aspects of implementing the OFDM transmitter in MATLAB. Code snippets and simulations provide a hands-on approach, guiding readers through the intricacies of translating theoretical knowledge into practical implementation. Potential challenges and considerations during the coding process are discussed, empowering readers to navigate the complexities of real-world application. This comprehensive exploration equips readers not only with theoretical insights but also with the practical skills necessary to implement an efficient OFDM transmitter.

BLOCK DIAGRAM



*Figure 2.1* Block diagram of OFDM Transmitter and Receiver

2.3.1 Data Mapping and Subcarrier Assignment

This sub-section delves into the intricate process of data mapping, a fundamental aspect where information is intelligently associated with specific subcarriers in the Orthogonal Frequency Division Multiplexing (OFDM) system. The exploration unfolds the principles behind subcarrier assignment, emphasizing the paramount importance of efficiently utilizing the frequency spectrum. Practical examples and scenarios are elucidated, providing a clear understanding of how data is strategically distributed across subcarriers. This strategic allocation maximizes transmission efficiency, ensuring optimal utilization of the available bandwidth.

2.3.2 Modulation Techniques in OFDM

Building upon the foundation laid by data mapping, this sub-section delves into the diverse modulation techniques employed in OFDM. The focus is on understanding how the encoded data is introduced to carrier frequencies, with specific considerations for modulation schemes such as Quadrature Amplitude Modulation (QAM) and Phase Shift Keying (PSK). Practical insights guide the reader in selecting suitable modulation schemes tailored to the requirements of different applications. The section offers a comprehensive exploration of modulation strategies, providing valuable knowledge for effective implementation in OFDM systems.

2.3.3 Guard Interval Implementation

The addition of guard intervals is a critical step in mitigating intersymbol interference within OFDM systems. This sub-section offers an in-depth analysis of guard interval implementation, providing readers with insights into the selection of appropriate guard interval lengths, their impact on system performance, and the associated trade-offs. Practical considerations and simulations further illustrate the significance of guard intervals in enhancing the robustness of OFDM transmission. By understanding guard interval implementation, readers gain valuable knowledge in optimizing the reliability and resilience of the OFDM system against channel distortions and impairments.2.4 OFDM Receiver

Complementing the transmitter, the OFDM receiver is a critical element in the data transmission process. This section takes a deep dive into the receiver's operations, demystifying the intricacies of signal demodulation and the removal of the cyclic prefix. The role of the receiver is to reconstruct the original data accurately, making its operations crucial to the overall success of the OFDM system. Practical considerations, such as decoding techniques and potential challenges in signal recovery, are explored in detail. By comprehending the nuances of the receiver's functionalities, readers gain insights into the entire lifecycle of an OFDM transmission.

The discussion extends to practical considerations when implementing the OFDM receiver in MATLAB. Code snippets and simulations provide a tangible representation of the theoretical concepts discussed earlier. Through this hands-on approach, readers not only grasp the theoretical foundations but also cultivate the skills necessary to design and implement an effective OFDM receiver. Challenges encountered during the implementation process are addressed, fostering a robust understanding of the practical aspects associated with OFDM receivers. This section thus serves as a bridge, connecting theoretical knowledge with the practical skills needed to navigate the intricacies of OFDM system design.

2.4.1 Signal Demodulation and Symbol Recovery

In this detailed sub-section, we intricately dissect the operations involved in signal demodulation within the OFDM receiver. The exploration encompasses the nuanced process of how the received signal is demodulated, unraveling the intricacies of symbol recovery. Decoding techniques, strategies for effective symbol recovery, and the influence of noise on the received signal are thoroughly discussed. Practical examples and simulations are employed to illustrate the challenges encountered during signal demodulation, providing a comprehensive understanding of the associated complexities and the strategies employed to overcome them.

2.4.2 Cyclic Prefix Removal and Channel Equalization

Expanding on the foundation laid by signal demodulation, this sub-section delves into the crucial processes of cyclic prefix removal and channel equalization in the OFDM receiver. The significance of eliminating the cyclic prefix to enhance signal integrity is elucidated. Simultaneously, the importance of channel equalization in mitigating distortions induced by the communication channel is explored. Practical considerations guide the reader in selecting appropriate equalization techniques, and simulations vividly demonstrate the impact of these processes on signal recovery. By comprehensively addressing cyclic prefix removal and channel equalization, this sub-section provides invaluable insights into optimizing the OFDM receiver's performance in the presence of channel-induced distortions and impairments.

CHAPTER 3

MATLAB IMPLEMENTATION

3.1 Introduction

In this introductory section, the importance of simulation in comprehending and implementing Orthogonal Frequency Division Multiplexing (OFDM) is underscored. The chapter aims to provide a bridge between theoretical concepts and practical implementation, emphasizing the role of simulation in gaining insights into OFDM system dynamics. Furthermore, MATLAB is introduced as the preferred platform for implementation due to its versatility, powerful signal processing capabilities, and widespread usage in the scientific and engineering communities.

3.1.1 Basic coding rules for MATLAB

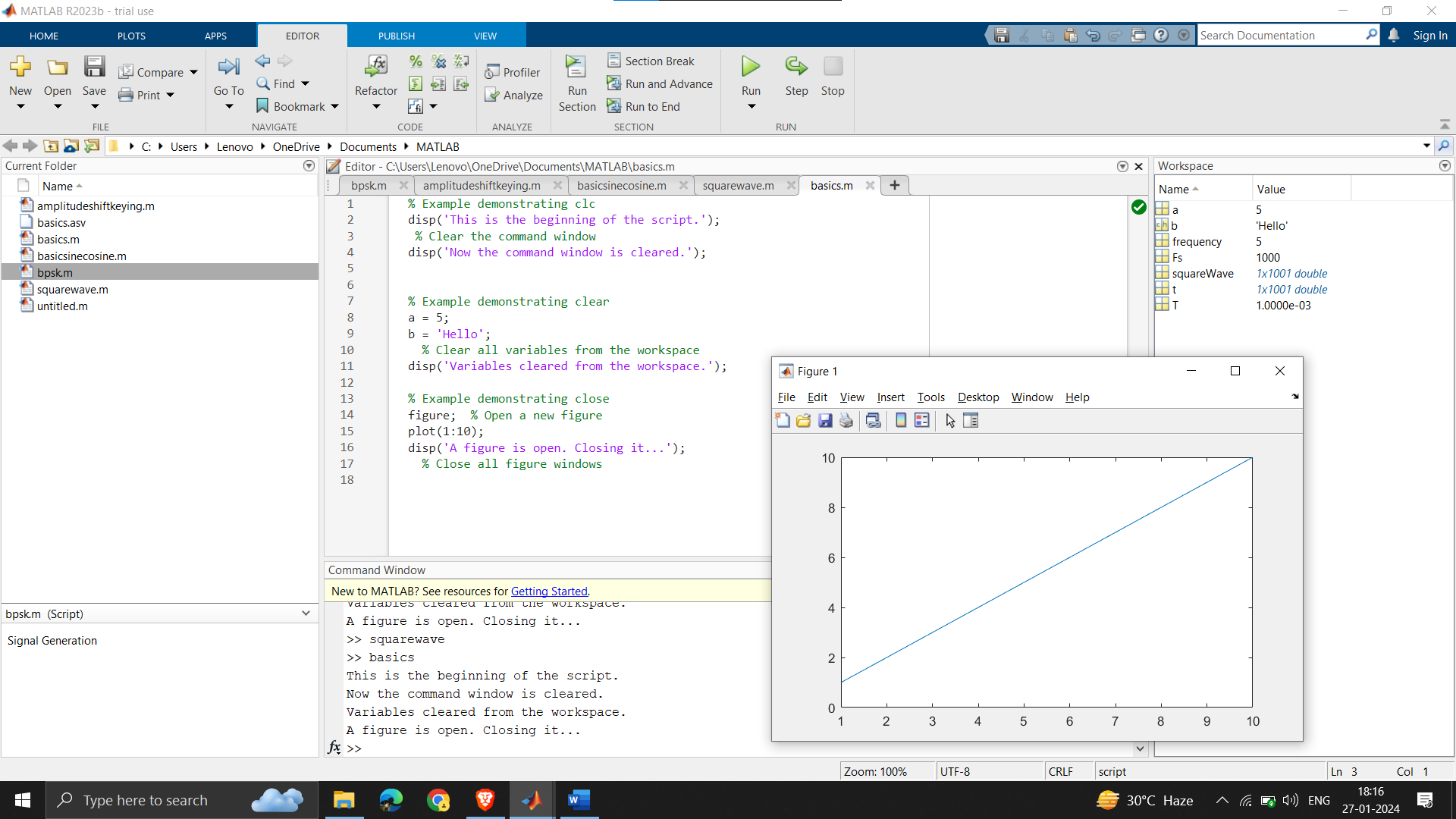
MATLAB (MATrix LABoratory) is a powerful numerical computing environment widely used in engineering, science, and mathematics. It provides a variety of tools for data analysis, visualization, and algorithm development.

Getting Started Commands

clc - Clear Command Window: The clc command is used to clear the command window, enhancing readability. It removes previous outputs and messages, creating a clean environment for new commands.

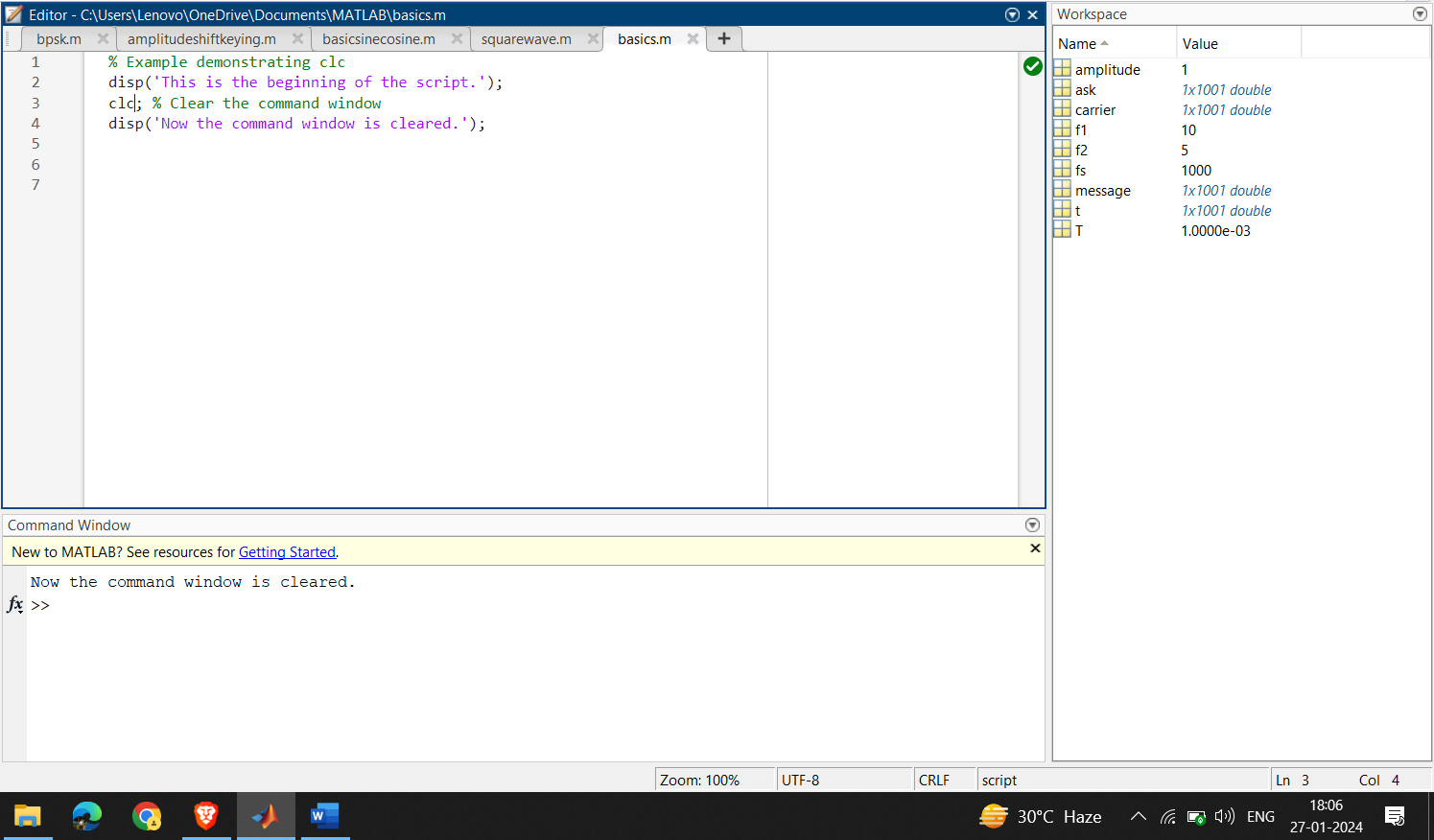
clear-The clear command without arguments removes all variables from the workspace. It helps avoid naming conflicts and ensures a clean slate for variable declarations.

close (Close Figures): The close command without any arguments closes all open figure windows. It is useful to prevent a large number of figure windows from accumulating.



*Figure 3.1* Command window without clc, clear, close

Here we can clearly see that the command window is filled with previous works and also the workspace is occupied with the previous variables and the graph window is open



*Figure 3.2*  Code with clc command

As of now we can see the command window is cleared and is only filled with the required information

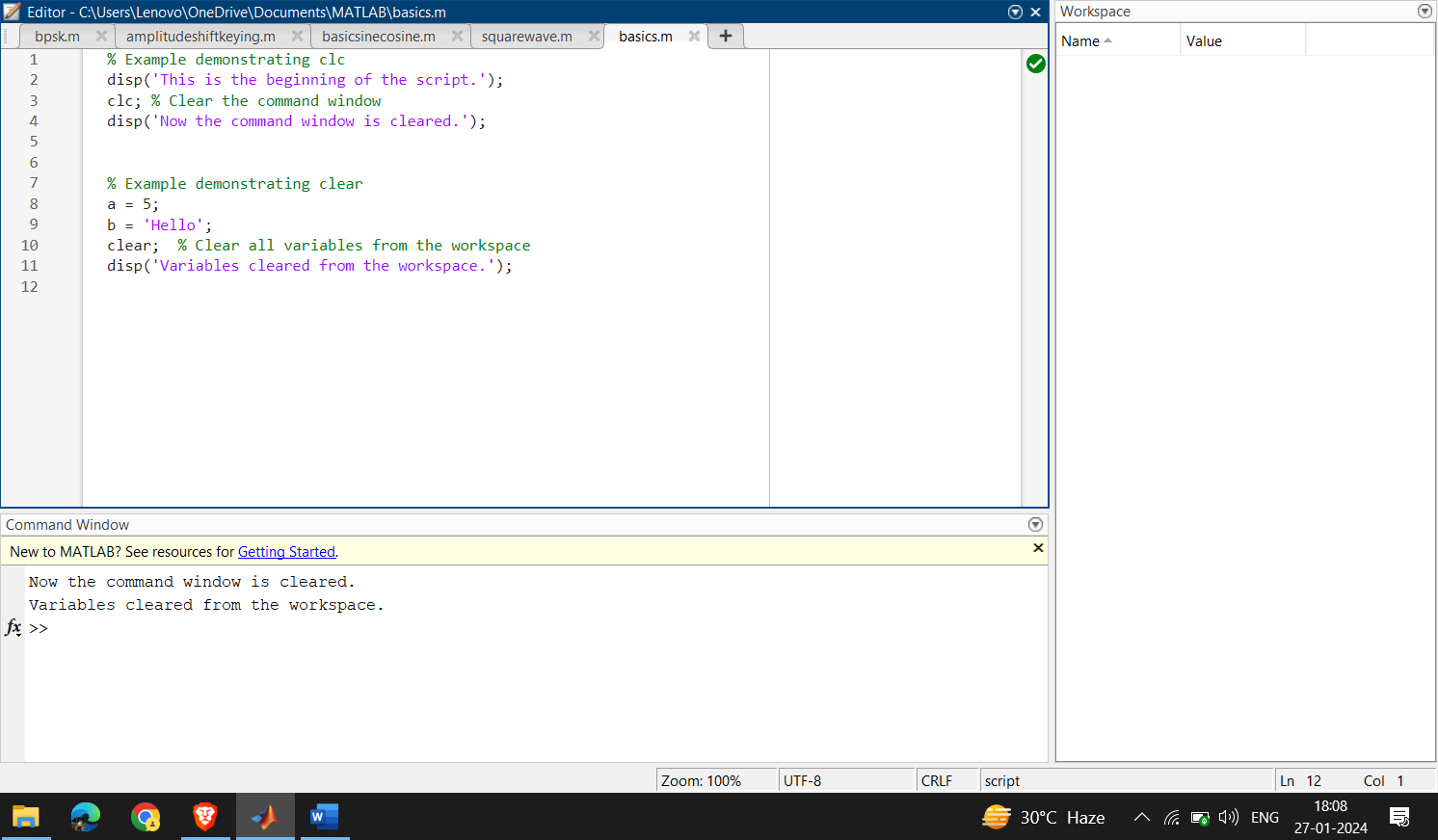


Figure 3.3 Code with clear command

As of now with the use of the clear command the workspace values are cleared and is ready for the next work

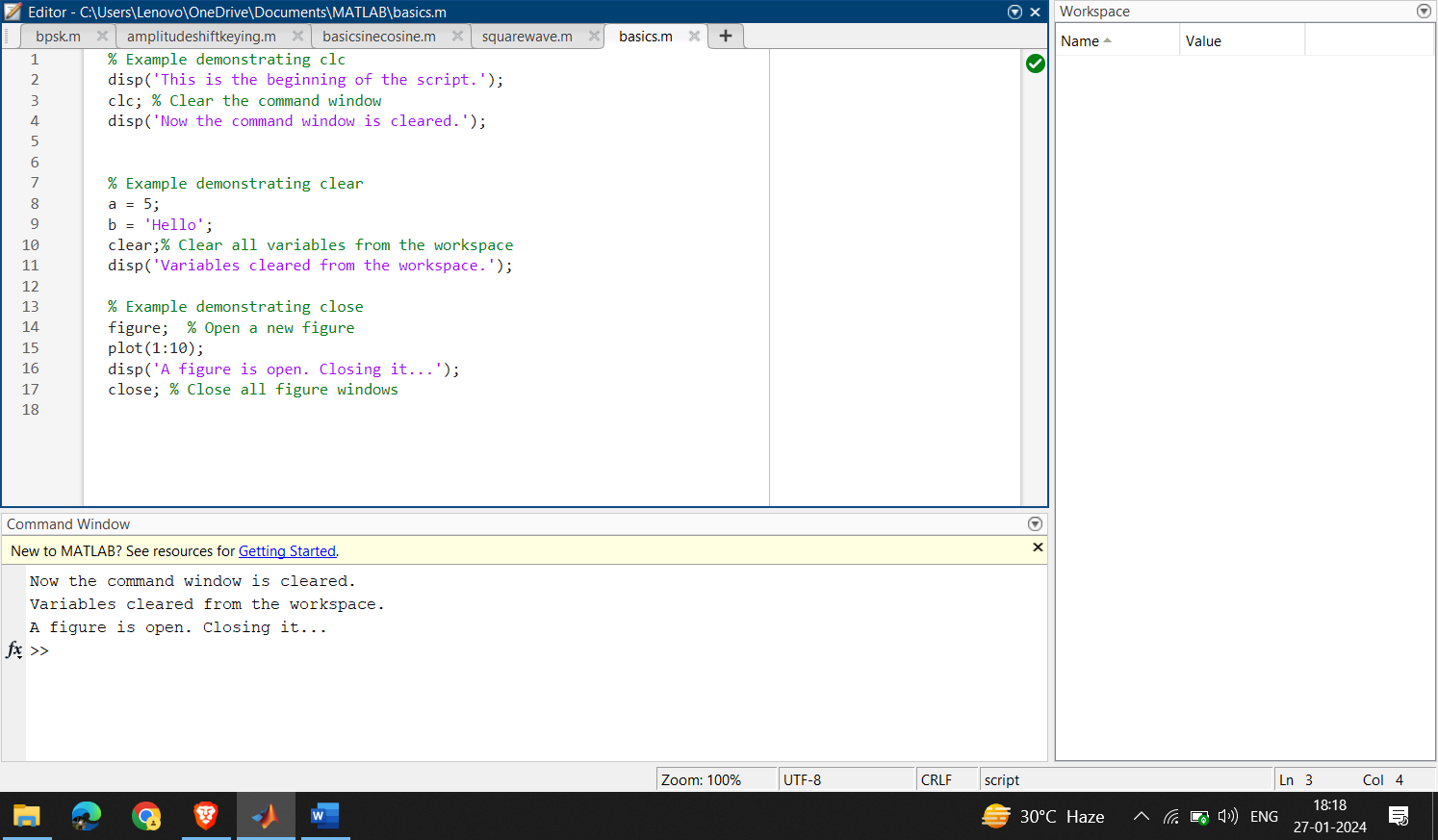


Figure 3.4: Code with close command

As now we can see the plot graph which was previously opened is now closed

It's important to note that these commands are often used for script development, debugging, and maintaining a clean workspace. However, in larger programs or functions, using them might have implications, so it's advisable to use them judiciously in such cases.

MATLAB supports various data types, including numeric, character, and logical types.

Variables are used to store and manipulate data, and their names must follow certain conventions.

3.1.2 Basic Operations:

MATLAB is a powerful programming language and environment commonly used for numerical computing, data analysis, and visualization. Here are some basics of MATLAB code:

1. Comments:

- Use `%` for single-line comments and `%{ ... %}` for multi-line comments.

% Single-line comment

%{

Multi-line

comment

%}

2. Variables:

- Declare variables using `=`. MATLAB is case-sensitive.

- Variables can store numbers, strings, arrays, etc.

CODE:

x = 10; % Numeric variable

name = 'John'; % String variable

A = [1, 2; 3, 4]; % Matrix variable

3. Display Output:

- Use `disp` or simply enter a variable name to display its value.

CODE:

matlab

disp(x);

OUTPUT:

x= 10

4. Basic Operations:

- MATLAB supports standard arithmetic operations.

CODE:

matlab

result = x + 5;

disp(result);

OUTPUT:

Result = 15

5. Vectors and Matrices:

- Create vectors and matrices using square brackets `[]`.

- Access elements using indexing.

CODE:

matlab

vec = [1, 2, 3];

mat = [1, 2; 3, 4];

element = vec(2); % Accessing the second element of vec

6. Functions:

- Define functions using the `function` keyword.

- Save functions in separate files with the same name as the function.

CODE:

function output = myFunction(input)

output = input \* 2;

end

7. Script Files:

- Create script files with a `.m` extension to store a sequence of MATLAB commands.

CODE:

% Example Script

x = 5;

y = x^2;

disp(y);

OUTPUT:

Y=25

8. Control Flow:

- Use `if`, `else`, `elseif`, `for`, `while` for control flow.

CODE:

if x > 0

disp('Positive');

elseif x < 0

disp('Negative');

else

disp('Zero');

end

OUTPUT:

Positive

9. Plotting:

- MATLAB provides powerful plotting functions.

CODE:

x = linspace(0, 2\*pi, 100);

y = sin(x);

plot(x, y);

These basics cover a wide range of MATLAB programming. As you progress, you'll delve into more advanced topics such as advanced data structures, file I/O, and more complex mathematical operations. MATLAB's extensive documentation and online resources are valuable references for learning and mastering the language.

3.1.3 AMPLITUDE SHIFT KEYING

Amplitude Shift Keying (ASK) is a type of modulation where the amplitude of a carrier signal is varied to represent digital data. For ASK with a message signal as a sine wave and the carrier as a square wave, the modulated signal can be expressed mathematically.

Let the message signal be m(t) = Am\* sin(2\pi fm(t) ), where:

- ( Am ) is the amplitude of the message signal,

- ( fm ) is the frequency of the message signal.

The carrier signal is a square wave, which can be represented as c(t) = square(2pi fc(t) ),

where:

- ( fc ) is the frequency of the carrier signal.

The ASK-modulated signal is then given by:

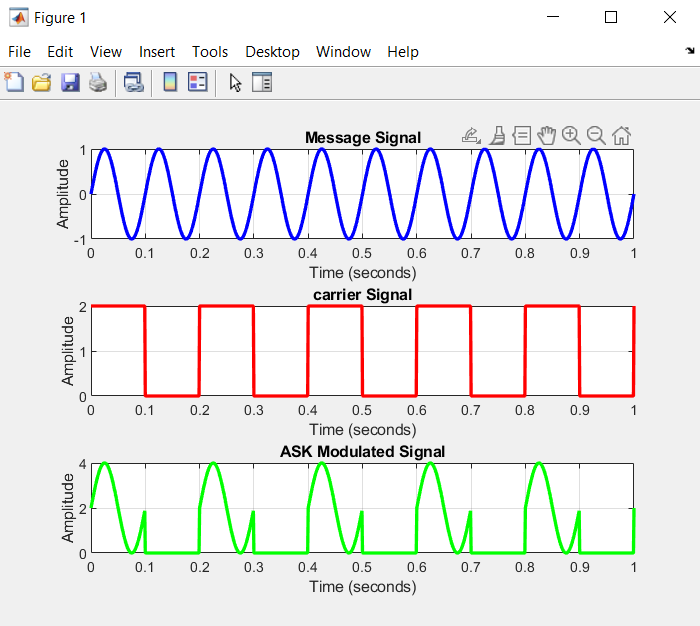
s(t) = (Am + Ac \*m(t)) \* c(t)

Where:

- ( Ac ) is the amplitude of the carrier signal.

In this formula, the amplitude of the carrier signal is varied based on the amplitude of the message signal. If m(t) is positive, the carrier amplitude is increased; if m(t) is negative, the carrier amplitude is decreased.

This is a basic formula for ASK modulation, and it assumes a simple case where there's no additional complexity such as pulse shaping, filtering, or noise. The actual implementation may involve additional considerations based on the specific requirements of your system.



*Figure 3.5*  Output for ask is as follows

Here the message is only seen when the value of the carrier is 1 or other than zero and hence is the output of ASK modulation

3.2 Transmitter Design in MATLAB

This section immerses itself in the practical implementation of the OFDM transmitter using MATLAB, providing a comprehensive step-by-step guide. The reader gains detailed insights into the coding process, supported by illustrative code snippets that elucidate key aspects such as data mapping, modulation techniques, and the inclusion of guard intervals. Accompanying explanations ensure a clear understanding of the transmitter's functionalities, while discussions about encountered challenges offer valuable insights into practical considerations and potential troubleshooting approaches.

3.2.1 Data Generation

To kickstart the transmitter design, a random data source of 64 bits is generated. This source data lays the foundation for the transmission system, serving as the basis for modulation and subsequent analysis. The generation of reliable and diverse data is crucial in ensuring the overall performance of the communication system. This section meticulously focuses on the process of creating a random data source, emphasizing its pivotal role as the input for the transmission system.

Random Data Generation

The 'randsrc' function in MATLAB is employed to generate the source data. This function facilitates the creation of a random matrix with user-specified dimensions and values. In this context, a row vector of 64 bits is generated, with each bit capable of taking values from 0 to M-1. The modulation order M determines the range of values, and for this design, QPSK modulation (M=4) is utilized. The following MATLAB code snippet illustrates this data generation process:

Matlab code

data = randsrc(1, no\_of\_data\_bits, 0:M-1);

This concise code snippet captures the essence of generating diverse and random data, a foundational step in preparing the input for the OFDM transmission system. The subsequent stages of data processing and modulation build upon this crucial initial step, contributing to the robustness and effectiveness of the entire communication system.

The code is as follows

% Generate random data source to be transmitted of length 64

data = randsrc(1, no\_of\_data\_bits, 0:M-1);

Data Visualization

To gain insights into the generated data, we utilize a stem plot for visualization. The stem plot provides a clear representation of each data point's amplitude. This step aids in understanding the characteristics of the data before modulation:

The code is as follows

% Visualize the generated data using a stem plot

figure(1);

stem(data);

grid on; % Enable grid lines for better readability

xlabel('Data Points'); % Label for the x-axis

ylabel('Amplitude'); % Label for the y-axis

title('Original Data'); % Title of the plot

Explanation:

Figure 1: This command creates a new figure window with the specified number in this case Multiple figures can be created in MATLAB, and assigning a number helps in referring to or updating specific figures.

grid on: Activates grid lines on the plot, providing a visual aid for better readability and analysis.

xlabel and ylabel: These commands label the x-axis and y-axis, respectively, providing context and scale information to the reader.

title: Adds a title to the plot, summarizing its content or purpose.

The grid lines enhance the plot's clarity, while labeling and titling contribute to a comprehensive understanding of the visualized data. This step prepares us for subsequent stages in the transmitter design process.

3.2.2 QPSK Modulation

The source data undergoes Quadrature Phase Shift Keying (QPSK) modulation to prepare it for transmission. QPSK is chosen for its efficiency in spectral utilization.

Matlab code

% Perform QPSK modulation on the input source data

qpsk\_modulated\_data = pskmod(data, M);

figure(2), stem(qpsk\_modulated\_data); title('QPSK Modulation')

The QPSK modulation involves mapping each pair of bits to a complex symbol, effectively

doubling the data transmission rate.

This completes the source data generation and modulation process, setting the stage for further stages of the transmitter design.

3.2.3 Cyclic Prefix and Serial-to-Parallel Conversion

In the initial stages of the OFDM transmitter design, the QPSK-modulated data undergoes a crucial transformation from a serial stream to four parallel subcarriers. This process is pivotal for efficient parallel processing and subsequent transmission.

Matlab code

S2P = reshape(qpsk\_modulated\_data, no\_of\_data\_bits/M, M);

sub\_carrier1 = S2P(:, 1);

sub\_carrier2 = S2P(:, 2);

Sub\_carrier3 = S2P(:, 3);

Sub\_carrier4 = S2P(:, 4);

The ‘reshape’ function organizes the QPSK-modulated data into four parallel subcarriers (sub\_carrier1 to sub\_carrier4). This parallel structure facilitates simultaneous processing, a fundamental characteristic of OFDM.

IFFT of Four Subcarriers

Following the parallel organization, the Inverse Fast Fourier Transform (IFFT) is applied to each subcarrier. This transformation is crucial for mapping the data from the frequency domain back to the time domain.

Matlab code

ifft\_subcarrier1 = ifft(sub\_carrier1);

ifft\_subcarrier2 = ifft(sub\_carrier2);

ifft\_subcarrier3 = ifft(sub\_carrier3);

ifft\_subcarrier4 = ifft(sub\_carrier4);

The ‘ifft’ function is employed, ensuring the signal is prepared for modulation and transmission over the channel.

Cyclic Prefix Addition

To mitigate inter-symbol interference and enhance signal robustness, a cyclic prefix is added to each subcarrier. This involves extending the signal by appending a portion of the signal to itself.

IFFT Transformation:- The ‘ifft’ function is applied to each subcarrier (‘S2P(:, i)’) to convert the data from the frequency domain to the time domain. This is a crucial step in preparing the signal for transmission.

Cyclic Prefix Addition:- For each subcarrier, a cyclic prefix is extracted from the end of the time-domain signal (‘ifft\_Subcarrier’) and appended to the beginning. The length of the cyclic prefix (‘cp\_len’) is determined based on the design parameters.

Signal Concatenation:- The ‘Append\_prefix’ matrix is formed by vertically concatenating the cyclic prefix and the original time-domain subcarrier. This results in an extended signal with the cyclic prefix preceding the actual data.

ifft(S2P(:, i), 16):- This part performs the Inverse Fast Fourier Transform (IFFT) on the i-th subcarrier of the matrix S2P. The 16 indicates the length of the IFFT, specifying the number of points in the time domain. It is essential to choose a length based on the design requirements and system specifications.

ifft\_Subcarrier(:, i):- This part assigns the result of the IFFT operation to the i-th column of the matrix ifft\_Subcarrier. The (:, i) notation ensures that the entire column is assigned the values obtained from the IFFT.

‘Vertcat’ is a MATLAB function used for vertical concatenation of arrays. It stands for "vertical concatenation." The function concatenates input arrays along their vertical dimension, meaning it stacks the arrays on top of each other. This is useful when you want to combine arrays vertically to create a larger array.

This line of code uses ‘Vertcat’ to vertically concatenate the arrays cyclic\_prefix(:, i) and ifft\_Subcarrier(:, i). The result is stored in the column i of the matrix Append\_prefix. It effectively appends the cyclic prefix to the corresponding subcarrier in the time-domain signal, creating the final signal with cyclic prefix added.

Matlab code

for i = 1:number\_of\_subcarriers

ifft\_Subcarrier(:, i) = ifft(S2P(:, i), 16); % Applying IFFT to each subcarrier

for j = 1:cp\_len

cyclic\_prefix(j, i) = ifft\_Subcarrier(j + cp\_start, i); % Extracting cyclic prefix

end

Append\_prefix(:, i) = vertcat(cyclic\_prefix(:, i), ifft\_Subcarrier(:, i)); % Appending prefix to each subcarrier

End

Noise and Channel:

Matlab code:

channel = randn(1,2) + sqrt(-1)\*randn(1,2);

after\_channel = filter(channel, 1, ofdm\_signal);

awgn\_noise = awgn(zeros(1, length(after\_channel)), 0);

recvd\_signal = awgn\_noise + after\_channel;

Explanation:

channel = randn(1,2) + sqrt(-1)\*randn(1,2);

This line generates a complex-valued channel response. randn(1,2) creates two random numbers from a normal distribution, and sqrt(-1)\*randn(1,2) creates two random numbers for the imaginary part. The resulting channel represents the frequency response of the communication channel.

after\_channel = filter(channel, 1, ofdm\_signal);

This line simulates the transmission through the channel. The filter function convolves the input signal ofdm\_signal with the channel response. This mimics the effect of the signal passing through a communication channel.

awgn\_noise = awgn(zeros(1, length(after\_channel)), 0);

This line generates additive white Gaussian noise (AWGN). zeros(1, length(after\_channel)) creates a zero-filled vector with the same length as after\_channel. The awgn function then adds Gaussian noise to this vector. The second argument 0 specifies the signal-to-noise ratio (SNR), and in this case, it's set to 0 dB, indicating that the noise power is equal to the signal power.

recvd\_signal = awgn\_noise + after\_channel; % With AWGN noise:

This line combines the original signal after transmission through the channel (after\_channel) with the generated AWGN (awgn\_noise). The result is the received signal (recvd\_signal) with added noise, simulating the effects of real-world communication channels.

3.3 Receiver Design in MATLAB

Building on the transmitter, this section guides readers through the implementation of the OFDM receiver in MATLAB. A comprehensive step-by-step approach is adopted, covering signal demodulation, cyclic prefix removal, and the reconstruction of the original data. Code snippets and explanations are provided to elucidate each phase of the receiver design. Challenges faced during implementation are discussed, enhancing the reader's understanding of potential pitfalls and solutions in implementing an effective OFDM receiver using MATLAB.

3.3.1 Data Re-shaping

recvd\_signal\_paralleled = reshape(recvd\_signal, rows\_Append\_prefix, cols\_Append\_prefix);

reshape is a MATLAB function used to change the shape of an array without changing its data. It takes the input array (recvd\_signal) and reshapes it into a matrix with the specified number of rows (rows\_Append\_prefix) and columns (cols\_Append\_prefix).

In this specific case, the received signal (recvd\_signal), which is typically a one-dimensional array, is being reshaped into a matrix form to match the structure of the original transmitted signal.

rows\_Append\_prefix represents the number of rows in the reshaped matrix, and cols\_Append\_prefix represents the number of columns.

The resulting matrix, stored in recvd\_signal\_paralleled,

3.3.2 Extracting FFT

% Perform FFT on the received signal for each subcarrier

for i = 1:number\_of\_subcarriers

fft\_data(:, i) = fft(recvd\_signal\_paralleled(:, i), 16); % 16 is the FFT size

end

% Extract individual subcarriers after FFT

F1 = fft\_data(:, 1);

F2 = fft\_data(:, 2);

F3 = fft\_data(:, 3);

F4 = fft\_data(:, 4);

The code is performing the Fast Fourier Transform (FFT) on the received signal for each subcarrier in an OFDM system.

The for loop iterates over each subcarrier (indexed by i), and for each subcarrier, it computes the FFT of the corresponding column of the recvd\_signal\_paralleled matrix.

The FFT size is specified as 16 in this case. The FFT size determines the number of points in the FFT output. The results of the FFT for each subcarrier are stored in the fft\_data matrix.

Each column of fft\_data corresponds to the FFT result of a specific subcarrier.After performing the FFT, individual subcarriers are extracted from the fft\_data matrix for further analysis or processing. These individual subcarriers are stored in variables F1, F2, F3, and F4.

Signal Reconstruction:

recvd\_serial\_data = reshape(fft\_data, 1, (16 \* 4));

The received signal, which was processed in the frequency domain using FFT (Fast Fourier Transform), is now reshaped into a serial stream. The fft\_data is reshaped into a 1x64 matrix, converting it from parallel streams for each subcarrier to a serial stream.

Demodulation:

qpsk\_demodulated\_data = pskdemod(recvd\_serial\_data, 4);

The reshaped serial data is then demodulated using PSK (Phase Shift Keying) demodulation. The pskdemod function is used with a modulation order of 4 (QPSK), as indicated by the '4' parameter. This step aims to recover the original data from the modulated signal.

Plotting:

figure(10): Initiates a new figure for plotting.

stem(data): Plots the original data (transmitted data) using a stem plot.

hold on: Keeps the existing plot and allows subsequent plots to be overlaid.

stem(qpsk\_demodulated\_data, 'rx'): Overlays the demodulated data on the same plot, marking

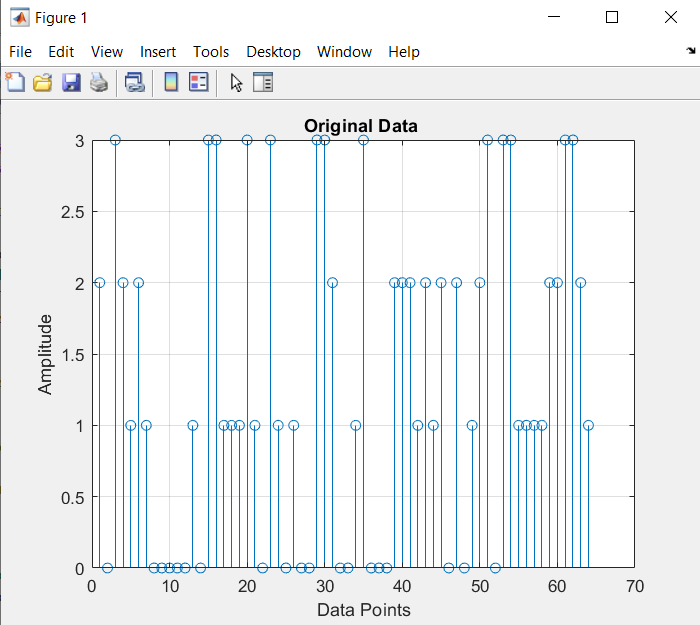
demodulated points with red 'x' symbols.

grid on; xlabel('Data Points'); ylabel('Amplitude'); title('Received Signal with error'): Sets the grid,

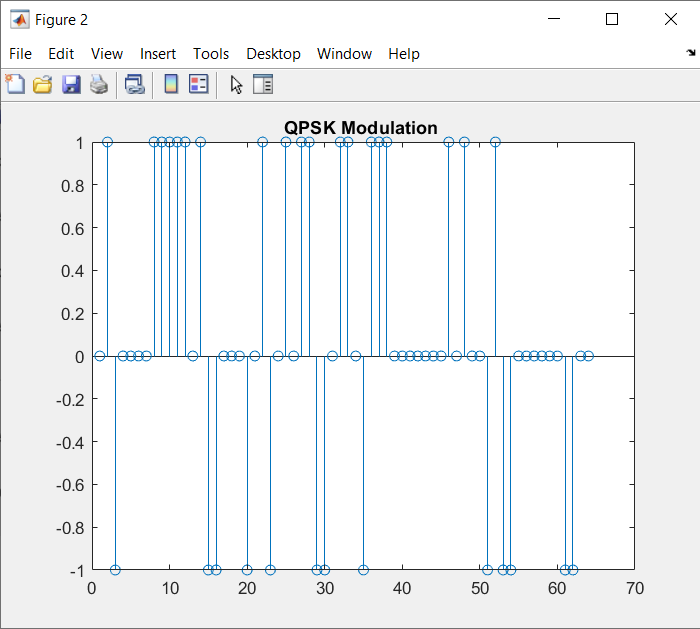
labels, and title for better visualization of the received signal.

CHAPTER 4 – SIMULATION RESULTS

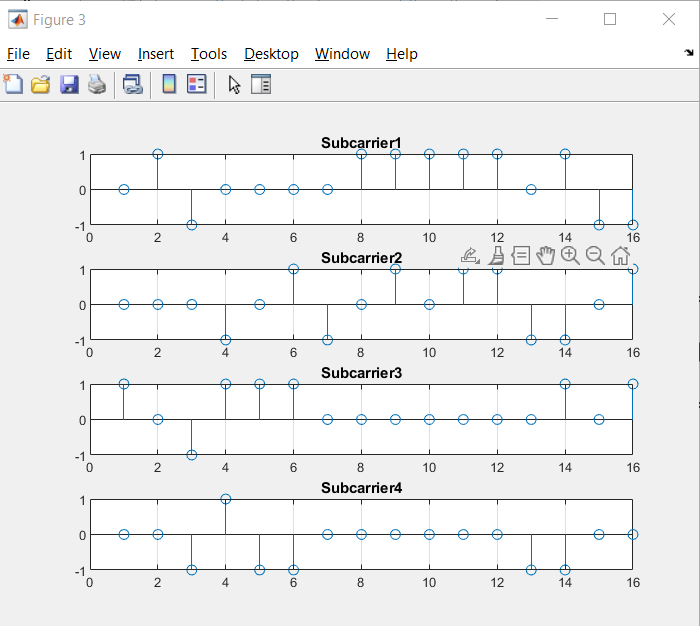
Main results of the OFDM:



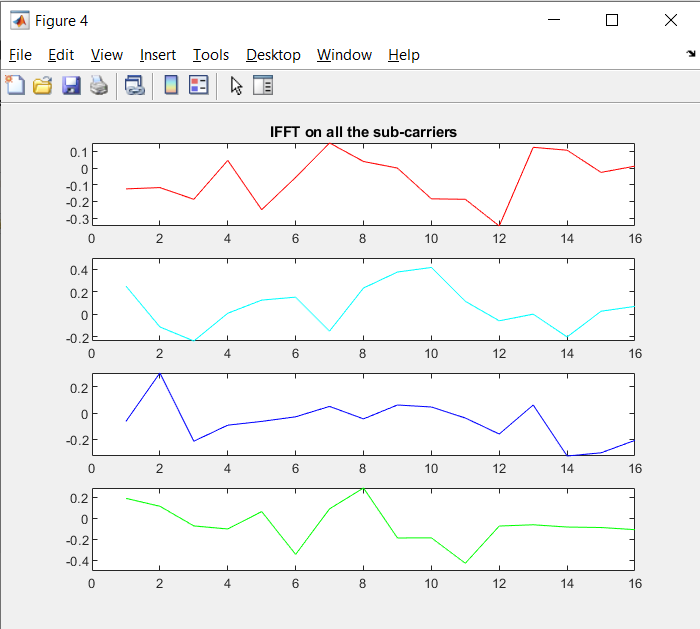
*Figure (4.1)*  Depicts the original 64-bit data generated randomly for transmission. The vertical stem plot visually represents each data point, with the x-axis indicating individual bits and the y-axis representing the amplitude or value of each bit. This dataset serves as the initial source of information before undergoing modulation for transmission.



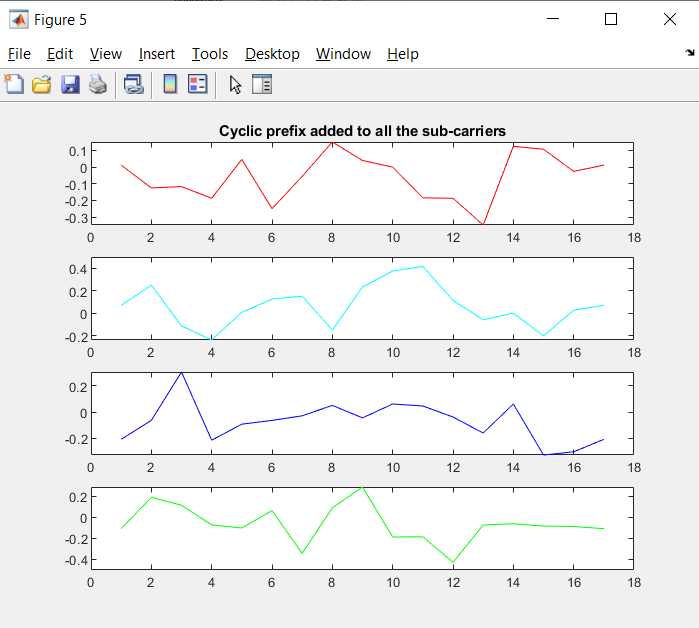
*Figure(4.2)*  Representing the modulated data which is QPSK modulated



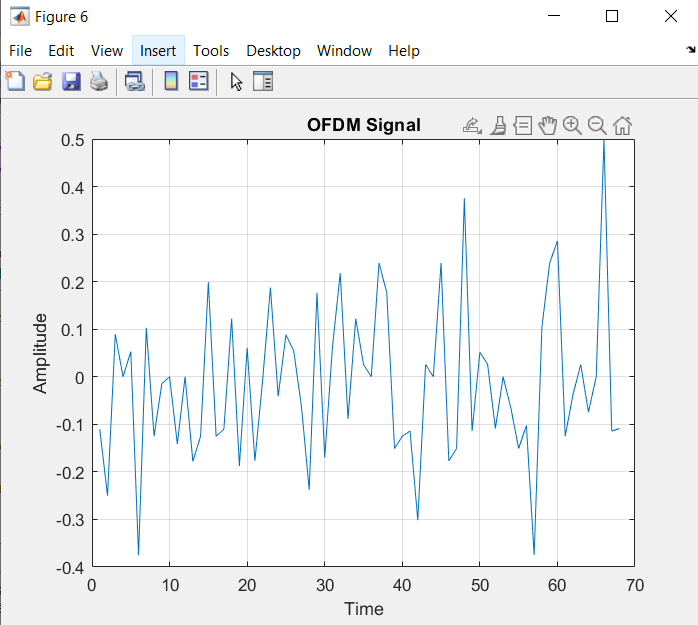
*Figure (4.3)*  Representing the modulated data which is divided among the subcarriers the 64-Bit data is divided among the 4 carriers as 16-Bit



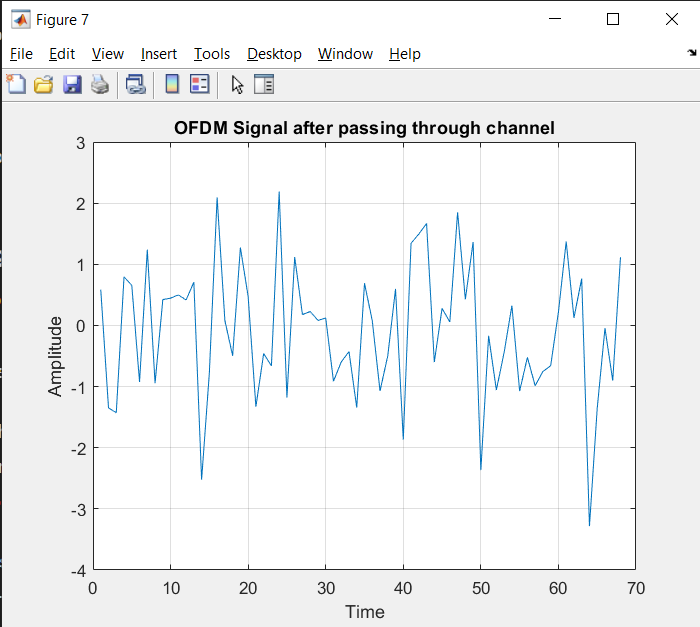
*figure 4.4*  Representing the 16-Bit subcarrier data that is Inverse Fast Fourier Transformed



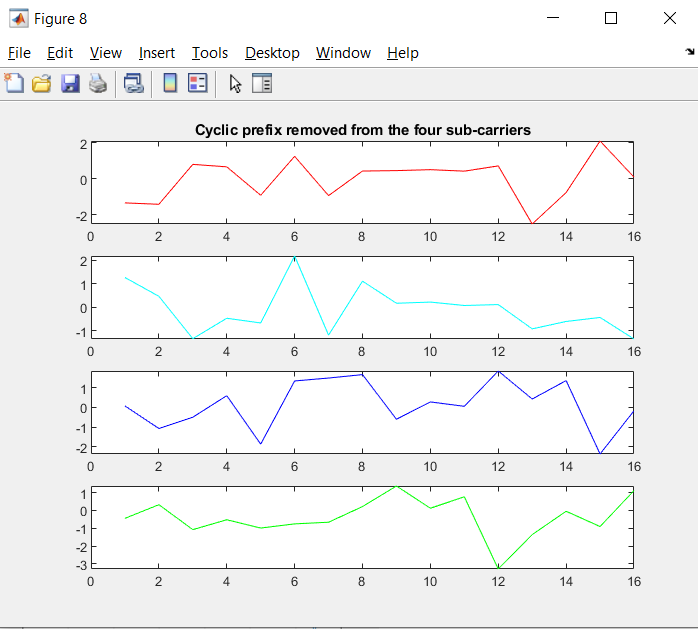
*Figure 4.5* The impact of adding the cyclic prefix to each subcarrier is depicted. The cyclic prefix is added to enhance orthogonality and mitigate inter-symbol interference.



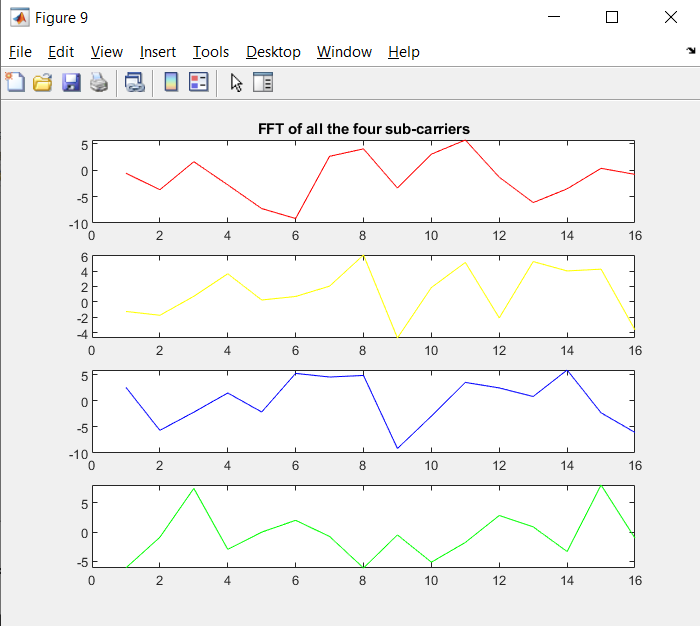
*Figure 4.6*  Showcases the transmitted Orthogonal Frequency Division Multiplexing (OFDM) signal. This plot illustrates the temporal characteristics of the composite signal, which is a result of combining multiple subcarriers with cyclic prefixes.



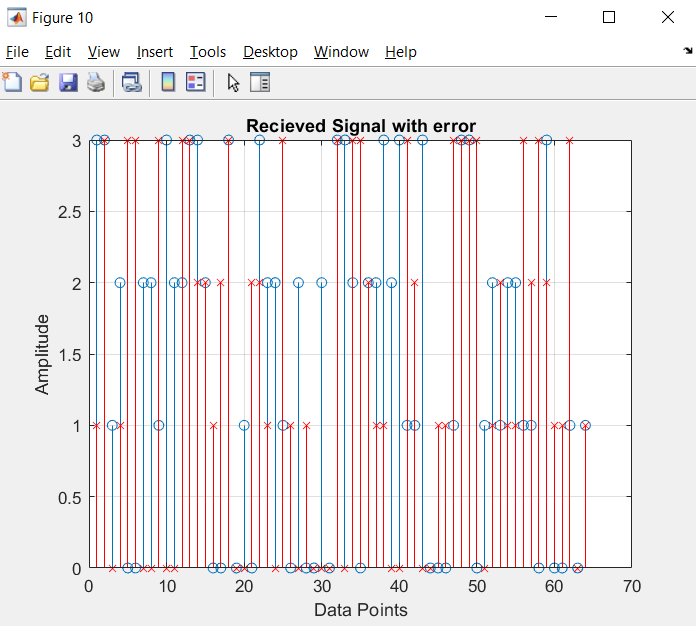
*Figure 4.7* Illustrates the OFDM signal after it has traversed through the communication channel. This plot provides a visual representation of the signal's response to the channel conditions, incorporating potential distortions, noise, and other channel-induced effects.



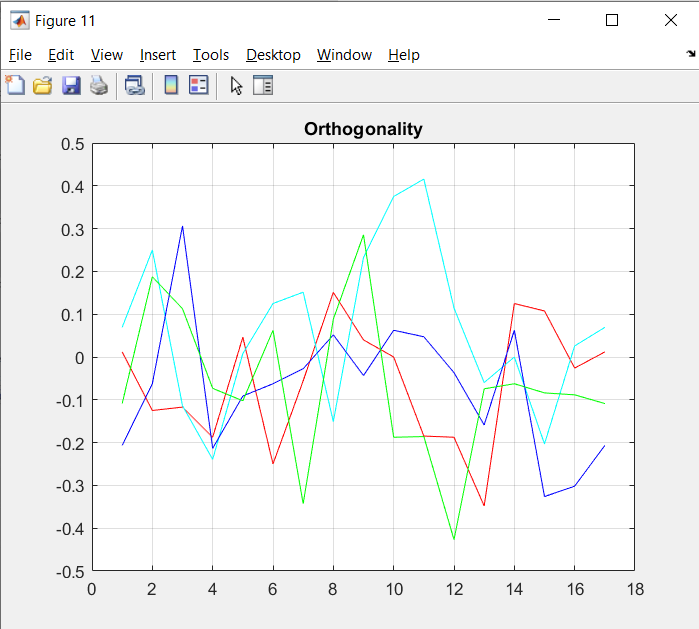
*Figure 4.8*  Portrays the signal processing stages after the reception of the OFDM signal. The plot visually represents the removal of cyclic prefixes from the four sub-carriers and the subsequent separation of individual sub-carriers.



*Figure 4.9*  The Fast Fourier Transform (FFT) is applied to the sub-carriers, showcasing the frequency domain representation of each channel. This crucial step follows the removal of cyclic prefixes and provides insights into the spectral characteristics of the received OFDM signal.



*Figure 4.10* Illustrates the received signal after demodulation, highlighting potential errors or discrepancies introduced during the transmission and reception process. Deviations between the blue stems and red 'X' markers indicate areas where errors may have occurred.



*Figure 4.11* illustrates the orthogonality concept among subcarriers in an Orthogonal Frequency Division Multiplexing (OFDM) system.Line Plots (Different Colours): Individual Subcarriers Each colour represents one of the four subcarriers in the OFDM system.

visually demonstrates the orthogonality property, a fundamental aspect ensuring the effective functioning of OFDM in transmitting multiple data streams simultaneously over a channel.

CHAPTER 5

CONCLUSION

5.1 Summary

The concluding chapter presents a comprehensive summary of the key findings and accomplishments of the thesis. It revisits the initial aim and objectives, providing a concise overview of how each objective was met and discussing the main outcomes of the research. By summarizing the main findings, this section encapsulates the essence of the thesis, paving the way for a deeper exploration of its contributions and implications.

5.2 Contributions

This section outlines the distinctive contributions made by the thesis to the field of Orthogonal Frequency Division Multiplexing (OFDM). It delves into any novel approaches, improvements, or insights gained during the course of the research. Whether it be advancements in the understanding of OFDM principles or innovative solutions to identified challenges, the contributions are outlined to underscore the significance of the work in advancing the current state of OFDM technology.

5.3 Future Directions

While reflecting on the achieved objectives, this section shifts the focus to potential future research directions. It proposes areas for further investigation based on identified limitations and gaps in the current study. Recommendations for extending the scope of the research or applying the findings in practical scenarios are discussed. This section serves as a guide for future researchers interested in exploring new dimensions within the realm of OFDM.

5.4 PAPR Reduction Techniques

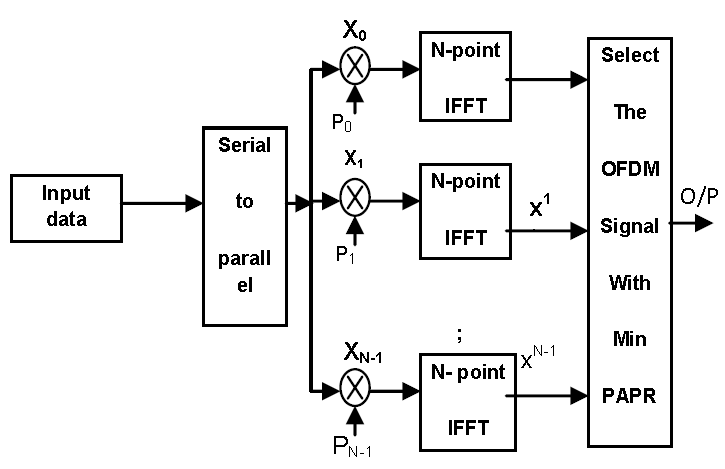
In this section, an in-depth exploration of Peak-to-Average Power Ratio (PAPR) reduction techniques is presented, highlighting their pivotal role in optimizing the overall performance of the Orthogonal Frequency Division Multiplexing (OFDM) system.

5.4.1 Overview of PAPR in OFDM

To set the stage, a comprehensive overview of PAPR within the context of OFDM is provided. The challenges posed by high PAPR in OFDM signals are detailed, elucidating how they can lead to power inefficiency and reduced system reliability. This overview lays the groundwork for understanding the critical need for PAPR reduction techniques.

5.4.2 Selected PAPR Reduction Techniques

This subsection delves into specific techniques employed to mitigate the impact of high PAPR in OFDM signals. Techniques such as selected mapping (SLM), partial transmit sequence (PTS), and tone reservation are discussed in detail. Each technique is dissected to elucidate its underlying principles, advantages, and potential limitations. Real-world applicability and considerations for selecting the most suitable technique based on system requirements are explored.



*Figure 5.1* Representing the PAPR Reduction method for the N-Point IFFT

5.4.3 Comparative Analysis of PAPR Reduction Techniques

A comparative analysis is conducted to evaluate the effectiveness of various PAPR reduction techniques. Comparative metrics include the extent of PAPR reduction achieved, computational complexity, and impact on system performance. By contrasting the strengths and weaknesses of each technique, readers gain valuable insights into selecting the most appropriate strategy based on specific application scenarios.

5.4.4 Practical Implementation Challenges

While these PAPR reduction techniques offer promising solutions, their practical implementation is not without challenges. This subsection addresses the practical considerations and challenges encountered during the implementation of PAPR reduction techniques in real-world scenarios. Factors such as computational complexity, implementation overhead, and compatibility with existing standards are discussed.

5.4.5 Impact on OFDM System Performance

The section concludes with an analysis of how the implemented PAPR reduction techniques influence the overall performance of the OFDM system. Metrics such as bit error rate, signal quality, and spectral efficiency are evaluated in the context of PAPR reduction. This assessment provides a holistic understanding of how these techniques contribute to enhancing the robustness and efficiency of OFDM communication systems.

By thoroughly exploring PAPR reduction techniques, this section contributes not only to the theoretical understanding of these strategies but also provides practical insights into their application, limitations, and impact on the overall performance of OFDM systems. This knowledge serves as a valuable resource for researchers, engineers, and practitioners seeking to optimize the power efficiency and reliability of OFDM communication.