Democratic and Popular Republic of Algeria
Ministry of Higher Education and Scientific Research
NOUR el Bachir EL BAYADH University Center
Institute of Technology
Department of Electrical Engineering

Educational Handout

« MENEZLA Fayssal » **University Center Nour El Bachir of El Bayadh**
« DEBBAL Mohammed » **University of Ain Temouchent**

Title

Advanced Digital Communication Lab

Course intended for students of

Master 1 Telecommunication, Specialization: Networks and Systems

Academic Year: 2024 – 2025

Preface

This document is a comprehensive guide to advanced digital communication systems, with a focus on practical learning through simulations in MATLAB and Simulink. It aims to provide students with both theoretical knowledge and hands-on experience in digital modulation techniques, transmission systems, and performance analysis.

Through a series of structured labs, students will explore key concepts such as modulation schemes (BASK, BPSK, BFSK, QAM), advanced transmission methods (OFDM, CDMA, MIMO), and the impact of channel conditions on system performance. Each lab is designed to bridge the gap between theory and practice, equipping students with the analytical and technical skills required for modern telecommunications.

We hope this document serves as a valuable resource for students to develop a deeper understanding of communication systems and inspires further exploration in this dynamic field.

Contents

General Introduction

In the era of rapid technological advancements, digital communication systems have become the backbone of modern connectivity, powering applications ranging from wireless networks and satellite communication to internet services and multimedia streaming. As data demands grow exponentially, understanding the principles, challenges, and performance of digital communication systems is critical for engineers and researchers.

This document explores key aspects of digital communications, focusing on practical implementations and performance analysis through MATLAB and Simulink simulations. It addresses essential topics such as digital modulation techniques (BASK, BPSK, BFSK, and QAM), advanced transmission methods (OFDM, CDMA, and MIMO), and the impact of real-world channel conditions on system reliability and efficiency.

The structured lab sessions aim to equip students with the skills needed to design, simulate, and analyze digital communication systems, bridging theoretical knowledge with real-world applications. By studying modulation schemes, error rates, spectral efficiency, and noise resilience, students will develop a solid foundation to innovate in telecommunications and meet the challenges of modern communication networks.

This introduction sets the stage for a deep dive into the principles of digital communications, offering a blend of theory, hands-on practice, and critical analysis to foster a comprehensive understanding of this essential field.

Lab 1 : Communication Blockset in Simulink

1. Objective of the Lab:

The objective of this lab is to familiarize students with Simulink tools, particularly the Communication Blockset, to simulate digital communication chains.

This lab will allow students to:

- Understand signal terminology, including the concepts of frame and sample.
- Explore source libraries (signal generation).
- Explore sink libraries (signal visualization).
- Explore other libraries.
- Build a complete digital communication chain and analyze its performance.

2. Theoretical Part:

2.1 Introduction to the Communication Blockset:

- **Simulink**: A graphical simulation environment for modeling and simulating dynamic systems.
- **Communication Blockset**: A library of blocks dedicated to modeling and simulating communication systems. It includes blocks for modulation, channels, coding, and more.

2.2 Signal Terminology (Frame or Sample):

- **Sample**: A discrete representation of a signal at a given instant, mainly used for continuous-time signals.
- **Frame**: A collection of samples grouped together, representing a data sequence or transmission block.

2.3 Source and Sink Libraries:

- **Sources**: Signal generation blocks (examples: sinusoidal signals, noise, random sequences). Example: Random Integer Generator, Sine Wave.
- **Sinks**: Signal visualization or storage blocks (examples: scopes, constellation diagrams, files). Example: Scope, Display.

2.4 Simulation of a Digital Communication Chain:

A digital communication chain consists of the following steps:

- **Information Source**: Generates the data to be transmitted.
- **Modulator**: Transforms the data into a transmittable signal.

- **Channel**: The medium through which the signal is transmitted (possibly noisy or altered).
- **Demodulator**: Recovers the data from the received signal.
- **Sink**: Visualizes the received data to analyze performance.

2.5 Creating a SIMULINK Model:

To open Simulink, type simulink in MATLAB's command window or click on its icon in MATLAB's main window.

Steps to Create a Simulink Model:

1. In the Simulink Library Browser menu, select File > New > Model or click on the new model icon. A working window named "Untitled" will open.
2. Open block collections by double-clicking (you can search for any block by typing its name in the search bar of the Simulink Library Browser).
3. Drag the required blocks into the working window to build the diagram (you can copy a block by right-clicking and dragging).
4. Link the blocks using the mouse.
5. Change any block's parameters by double-clicking it, which will open a configuration window.
6. Once the diagram is completed, save the model into a file: File > Save or use the save icon, and give the model a name (*.mdl).

Simulation Steps:

7. Run the simulation from the Simulation > Start menu or by clicking on the start icon.
8. To set the simulation time, enter the value in the menu toolbar.

3. Practical Part:

In this section, simulations will be conducted using the Simulink model.

Two transmission chains will be constructed:

- Digital transmission chain using DIGITAL BPSK modulation.
- Digital transmission chain using baseband modulation (Line coding).

3.1 Digital Transmission Chain using DIGITAL Modulation:

Proposed Steps for Simulation:

- Use a Random Integer Generator as the source.
- Add a modulation block, e.g., BPSK Modulator Baseband.
- Start with an ideal (wired) channel.
- Add a BPSK Demodulator Baseband and connect an Error Rate Calculation.

- Visualize the result using a Scope.

Frame vs. Sample Study:

- Modify block parameters to simulate differences between frame and sample processing.
- Observe the impact on latency and signal visualization.

Extension with a Noisy Channel:

- Replace the ideal channel with an AWGN Channel.
- Study the impact of noise on system performance by varying the Signal-to-Noise Ratio (SNR).

3.2 Digital Transmission Chain using Baseband Modulation (NRZ, RZ, and Manchester Codes):

Simulate the following three codes in Simulink:

- **NRZ, RZ, and Manchester** using appropriate blocks and parameters:
  - Bernoulli Binary Generator: Sample time = 1e-3 (Ts = 1 ms).
  - Unipolar to Bipolar Converter: M-ary number = 2.
  - Pulse Generator: Pulse type = Sample Based, Period = 2, Pulse width = 1, Sample time = 5e-4.
  - Logical Operator: Operator = XOR.
  - Data Type Conversion: Output Data = double.

Simulation Tasks:

- Simulate the diagram for 0.01 seconds.
- Plot the resulting signals.
- Extract and label the binary sequence from the graph.

4. Conclusion

This lab introduced students to Simulink and the Communication Blockset for modeling digital communication systems. It provided both theoretical knowledge of signal concepts, such as frames and samples, and practical experience in building and analyzing digital communication chains. Through hands-on simulations, students explored modulation techniques, noise effects, and line coding, reinforcing their understanding of key concepts while developing essential analytical and technical skills for telecommunications.

Lab 02: Study of the Performance of Digital Modulation Techniques - BASK, BPSK, BFSK, and QAM

1. Lab Objective:

The objective of this lab is to study the performance of digital modulation techniques in a coherent digital communication system. Specifically, this lab focuses on BASK (Binary Amplitude Shift Keying), BPSK (Binary Phase Shift Keying), and BFSK (Binary Frequency Shift Keying) modulation techniques. Subsequently, we will study and simulate a digital communication system using QAM (Quadrature Amplitude Modulation) and evaluate its performance.

The aim is to understand how each technique influences the reliability and efficiency of data transmission by analyzing parameters such as the bit error rate in relation to the signal-to-noise ratio (SNR). This study helps students grasp the trade-offs between robustness and efficiency for different digital modulation techniques in various communication environments.

2. Theory:

2.1 Introduction to Digital Modulation:

Digital modulation is a technique that converts a digital signal, typically consisting of bits (0 and 1), into an analog signal that can be efficiently transmitted over a communication channel. This conversion enables information to be transmitted over longer distances and through various media, such as cables, optical fibers, or radio waves. Digital modulation offers advantages over analog modulation in terms of quality, noise resistance, and bandwidth utilization.

2.2 Main Digital Modulation Techniques:

Digital modulation techniques are diverse, each offering specific advantages depending on the application and channel type. The main techniques used in modern telecommunications systems are:

2.2.1 BASK (Binary Amplitude Shift Keying)

- **Definition:** BASK is a digital modulation method where information is transmitted by varying the amplitude of a carrier wave. For binary data, two amplitude levels are used: a "high" level (typically associated with a bit "1") and a "low" or zero level (associated with a bit "0").

- **Operation:**
   - When the bit to be transmitted is **1**, the carrier's amplitude is at its maximum level.
   - When the bit is **0**, the carrier's amplitude is reduced to zero.
- **Advantages:**
   - Simple to implement.
   - Suitable for systems with limited bandwidth.
- **Disadvantages:**
   - ❖ Sensitive to noise, which can affect the amplitude and thus the accuracy of demodulation.

2.2.2 BPSK (Binary Phase Shift Keying)

- **Definition:** BPSK is a digital modulation method that uses two distinct phases to represent binary data. Each bit is represented by a phase change in the carrier signal.
- **Operation:**
   - ❖ For a bit **1**, the carrier is in-phase (0°).
   - ❖ For a bit **0**, the carrier is phase-shifted by 180° (or inverted).
- **Advantages:**
   - ❖ High robustness against noise compared to BASK.
   - ❖ Simple to implement and efficient in terms of bandwidth usage.
- **Disadvantages:**
   - ❖ Less efficient in terms of transmission capacity compared to more complex modulations (e.g., QPSK).

2.2.3 BFSK (Binary Frequency Shift Keying)

- **Definition:** BFSK is a digital modulation method that uses two distinct frequencies to represent binary data. Each bit is transmitted by alternating between the two frequencies.
- **Operation:**
   - ❖ For a bit **1**, a signal at one frequency (f_1) is transmitted
   - ❖ For a bit **0**, a signal at a different frequency (f_0) is transmitted
- **Advantages:**
   - ❖ Less sensitive to noise compared to BASK, as frequency signals are less affected by amplitude variations.
   - ❖ Suitable for long-distance transmissions

- **Disadvantages:**
  - ❖ Requires more bandwidth than BPSK and BASK

2.3 QAM (Quadrature Amplitude Modulation) :

QAM combines ASK and PSK to encode information using both amplitude and phase changes. This technique is highly efficient for high-speed transmissions and is commonly used in modern telecommunications systems.

Example:

16-QAM uses 16 different combinations of phase and amplitude to encode 4 bits per symbol, offering high data rates but requiring precise synchronization and good channel quality.

2.3.1 Operating Principles:

Symbols:

In a QAM system, each symbol represents multiple bits. For instance, in the case of 16-QAM, each symbol represents 4 bits (since 2^4=16).

The symbols are arranged in a Cartesian plane, where the horizontal axis represents the amplitude (or phase) of one carrier, and the vertical axis represents the amplitude (or phase) of another carrier in quadrature (shifted by 90 degrees).

Constellation Diagram:

The modulated symbols are graphically represented in a constellation diagram. Each point in this diagram corresponds to a unique symbol, and the position of each point is determined by the amplitude and phase values.

For example, a constellation diagram for 16-QAM will have 16 points forming a grid (4x4).

Transmission:

The modulated signal is transmitted over the communication channel. Due to channel imperfections (such as noise and interference), the received signal may be distorted.

Demodulation:

At the receiver, the signal is demodulated to recover the original symbols. This involves determining which point in the constellation diagram is closest to the received signal.

2.3.2 Advantages of QAM:

Spectral Efficiency:

QAM allows multiple bits per symbol to be transmitted, increasing spectral efficiency. This means it uses available bandwidth more effectively.

Flexibility:

By adjusting the modulation order (e.g., switching from 16-QAM to 64-QAM or 256-QAM), it is possible to increase the bit rate without requiring additional bandwidth.

Applications in Modern Systems:

QAM is widely used in modern communication systems, including wireless networks (Wi-Fi), digital television, and satellite communication systems.

2.3.3 Disadvantages of QAM:

Sensitivity to Noise:

As the modulation order increases, QAM becomes more sensitive to noise and interference. For example, 64-QAM is more prone to transmission errors than 16-QAM.

Demodulation Complexity:

The demodulation of QAM signals is more complex than simpler modulations like BPSK or QPSK.

2.4 Performance Criteria for Modulation:

2.4.1 Spectral Efficiency:

Spectral efficiency measures how much data can be transmitted within a certain bandwidth. It is expressed in bits per second per hertz (bps/Hz). A modulation scheme with good spectral efficiency allows more data to be sent without consuming excessive bandwidth, which is crucial in communication networks with limited bandwidth.

2.4.2 Bit Error Rate (BER):

The bit error rate is the percentage of bits received incorrectly compared to the total number of bits transmitted.

A low BER indicates good transmission quality. Modulation techniques are chosen based on their ability to minimize BER, ensuring reliable communication.

2.4.3 Noise Robustness:

Noise robustness measures a modulation's ability to resist interference and noise. Techniques like FSK and PSK are more robust against noise than ASK, making them better choices for noisy or long-distance transmission channels.

2.4.4 Implementation Complexity:

Implementation complexity considers the difficulty of implementing each technique in terms of hardware and computation. For example, advanced techniques like QAM require precise synchronization and sophisticated signal processing, making them more complex than simpler techniques like BPSK.

2.5 Theoretical Conclusion:

Digital modulation techniques enable reliable and efficient data transmission based on specific channel and transmission conditions. The choice of modulation depends on several factors: the need to minimize BER, spectral efficiency, noise robustness, and implementation complexity. Each technique has specific applications, requiring an understanding of the trade-offs involved.

3. Practical:

3.1 Comparison of the Performance of Digital Modulation Techniques BASK, BPSK, and BFSK:

Source Code:

The source code provided simulates the performance of coherent digital communication systems using BASK, BPSK, and BFSK modulation techniques. It evaluates the BER as a function of the signal-to-noise ratio (SNR).

```
% START
% Simulation of the performance of a coherent communication system
% using BASK, BPSK, and BFSK

clear;
clc;

% Simulation parameters
numBits = 1e5;                    % Number of bits
Eb_N0_dB = 0:2:20;                % Range of Eb/N0 values in dB
Eb_N0 = 10.^(Eb_N0_dB/10);        % Convert Eb/N0 to linear scale

% Binary data generation
data = randi([0, 1], 1, numBits);

% Initialize BER vectors for each modulation
ber_BASK = zeros(1, length(Eb_N0_dB));
ber_BPSK = zeros(1, length(Eb_N0_dB));
ber_BFSK = zeros(1, length(Eb_N0_dB));

% Loop over different Eb/N0 values
for i = 1:length(Eb_N0_dB)
    % Noise (variance depends on SNR)
    N0 = 1 ./ Eb_N0(i);  % Noise power

    % BASK modulation
    BASK_signal = data * sqrt(2) - sqrt(2)/2;
    BASK_noise = sqrt(N0/2) * randn(1, numBits);
    BASK_received = BASK_signal + BASK_noise;
    BASK_decoded = BASK_received > 0;
    ber_BASK(i) = sum(BASK_decoded ~= data) / numBits;

    % BPSK modulation
    BPSK_signal = 2*data - 1;
    BPSK_noise = sqrt(N0/2) * randn(1, numBits);
    BPSK_received = BPSK_signal + BPSK_noise;
```

```
    BPSK_decoded = BPSK_received > 0;
    ber_BPSK(i) = sum(BPSK_decoded ~= data) / numBits;

    % BFSK modulation
    freq1 = sqrt(2);   % Frequency of the first tone
    freq2 = -sqrt(2);  % Frequency of the second tone
    BFSK_signal = data * freq1 + (1 - data) * freq2;
    BFSK_noise = sqrt(N0/2) * randn(1, numBits);
    BFSK_received = BFSK_signal + BFSK_noise;
    BFSK_decoded = BFSK_received > 0;
    ber_BFSK(i) = sum(BFSK_decoded ~= data) / numBits;
end

% Display results
figure;
semilogy(Eb_N0_dB, ber_BASK, '-o', 'DisplayName', 'BASK');
hold on;
semilogy(Eb_N0_dB, ber_BPSK, '-s', 'DisplayName', 'BPSK');
semilogy(Eb_N0_dB, ber_BFSK, '-d', 'DisplayName', 'BFSK');
xlabel('Eb/N0 (dB)');
ylabel('Bit Error Rate (BER)');
title('Performance of BASK, BPSK, and BFSK Modulations');
legend;
grid on;
hold off;
% END
```

Required Task:

- ✓ Open the MATLAB text editor.
- ✓ Copy the following source code into the editor.
- ✓ Run this program.
- ✓ Visualize the signals in the figures.
- ✓ Modify the parameters and visualize the signals again.
- ✓ Interpret the simulation results.
- ✓ Compare these results with the theoretical part.
- ✓ Provide a conclusion.

3.2 Study and Comparison of QAM Performance (4-QAM, 16-QAM, and 32-QAM)

Source Code:

The source code for QAM modulation simulates a digital communication system using 4-QAM, 16-QAM, and 32-QAM. It evaluates the BER for each modulation type.

```
% START
% Lab: Performance of a coherent digital communication system with QAM
modulation

clear;
clc;

% Simulation parameters
numBits = 1e5;                % Number of bits
Eb_N0_dB = 0:2:20;            % Range of Eb/N0 values in dB
```

```
Eb_N0 = 10.^(Eb_N0_dB/10);     % Convert Eb/N0 to linear scale

% Initialize BER vectors for each modulation
ber_4QAM = zeros(1, length(Eb_N0_dB));
ber_16QAM = zeros(1, length(Eb_N0_dB));
ber_32QAM = zeros(1, length(Eb_N0_dB));

% Loop over different Eb/N0 values
for i = 1:length(Eb_N0_dB)
    % Noise (variance depends on SNR)
    noisePower = 1 / (2 * Eb_N0(i)); % Noise power

    % 4-QAM modulation
    data_4QAM = randi([0, 3], 1, numBits / 2); % Random data for 4-QAM
    modulatedData_4QAM = qammod(data_4QAM, 4, 'UnitAveragePower', true);
% 4-QAM modulation
    noise_4QAM = sqrt(noisePower) * (randn(size(modulatedData_4QAM)) + 1i
* randn(size(modulatedData_4QAM))); % Complex noise
    receivedSignal_4QAM = modulatedData_4QAM + noise_4QAM; % Received
signal
    demodulatedData_4QAM = qamdemod(receivedSignal_4QAM, 4,
'UnitAveragePower', true); % 4-QAM demodulation
    ber_4QAM(i) = sum(data_4QAM ~= demodulatedData_4QAM) / (numBits / 2);
% Bit Error Rate for 4-QAM

    % 16-QAM modulation
    data_16QAM = randi([0, 15], 1, numBits / 4); % Random data for 16-QAM
    modulatedData_16QAM = qammod(data_16QAM, 16, 'UnitAveragePower',
true); % 16-QAM modulation
    noise_16QAM = sqrt(noisePower) * (randn(size(modulatedData_16QAM)) +
1i * randn(size(modulatedData_16QAM))); % Complex noise
    receivedSignal_16QAM = modulatedData_16QAM + noise_16QAM; % Received
signal
    demodulatedData_16QAM = qamdemod(receivedSignal_16QAM, 16,
'UnitAveragePower', true); % 16-QAM demodulation
    ber_16QAM(i) = sum(data_16QAM ~= demodulatedData_16QAM) / (numBits /
4); % Bit Error Rate for 16-QAM

    % 32-QAM modulation
    data_32QAM = randi([0, 31], 1, numBits / 5); % Random data for 32-QAM
    modulatedData_32QAM = qammod(data_32QAM, 32, 'UnitAveragePower',
true); % 32-QAM modulation
    noise_32QAM = sqrt(noisePower) * (randn(size(modulatedData_32QAM)) +
1i * randn(size(modulatedData_32QAM))); % Complex noise
    receivedSignal_32QAM = modulatedData_32QAM + noise_32QAM; % Received
signal
    demodulatedData_32QAM = qamdemod(receivedSignal_32QAM, 32,
'UnitAveragePower', true); % 32-QAM demodulation
    ber_32QAM(i) = sum(data_32QAM ~= demodulatedData_32QAM) / (numBits /
5); % Bit Error Rate for 32-QAM
end

% Display results
figure;
semilogy(Eb_N0_dB, ber_4QAM, '-o', 'DisplayName', '4-QAM');
hold on;
semilogy(Eb_N0_dB, ber_16QAM, '-s', 'DisplayName', '16-QAM');
semilogy(Eb_N0_dB, ber_32QAM, '-d', 'DisplayName', '32-QAM');
xlabel('Eb/N0 (dB)');
ylabel('Bit Error Rate (BER)');
```

```
title('Performance of QAM Modulations (4-QAM, 16-QAM, 32-QAM)');
grid on;
legend;
hold off;
% END
```

Code Explanation

1. **Simulation Parameters:**
   - numBits: The total number of bits to transmit.
   - M: The modulation order for QAM (e.g., 16 for 16-QAM).
   - Eb_N0_dB: The range of signal-to-noise ratio (SNR) values expressed in dB.
2. **Data Generation:**
   - Random bits are generated and converted into symbols for QAM modulation.
3. **QAM Modulation:**
   - The symbols are modulated using the qammod function.
4. **Loop over Eb/N0E_b/N_0Eb/N0 Values:**
   - For each SNR value, noise is added to the modulated signal, and the received signal is demodulated using qamdemod.
5. **BER Calculation:**
   - The bit error rate (BER) is calculated by comparing the original bits with the demodulated bits.
6. **Results Display:**
   - A graph is plotted to show the performance of QAM modulation as a function of SNR.

Required Task:

- Open the MATLAB text editor.
- Copy the following source code into the editor.
- Run this program.
- Visualize the signals in the figures.
- Modify the parameters and visualize the signals again.
- Interpret the simulation results.
- Compare these results with the theoretical part.
- Provide a conclusion.

4. Conclusion

This lab provided a comprehensive study of the performance of key digital modulation techniques, including BASK, BPSK, BFSK, and QAM.

By analyzing parameters such as bit error rate (BER) and signal-to-noise ratio (SNR), students explored the trade-offs between noise robustness, spectral efficiency, and implementation complexity for each modulation scheme.

Simulations confirmed theoretical expectations, showing that simpler modulation schemes like BPSK offer better noise robustness, while advanced techniques like QAM achieve higher spectral efficiency at the cost of increased sensitivity to noise and implementation complexity. These insights are critical for selecting appropriate modulation techniques for different communication system requirements.

Lab 3 : Simulation of OFDM and CDMA Transmission Using Simulink

Objectives of the TP

- Understand the principle of OFDM (Orthogonal Frequency Division Multiplexing) and CDMA (Code Division Multiple Access) modulation.
- Simulate and analyze the performance of OFDM and CDMA transmission using Simulink.
- Study the effect of multipath channels on these transmission techniques.

1. Theoretical Background on OFDM and CDMA Transmission

1.1 Introduction to OFDM and CDMA Modulations

OFDM (Orthogonal Frequency Division Multiplexing) and **CDMA (Code Division Multiple Access)** are widely used modulation techniques in modern communication systems to enhance transmission reliability and network capacity.

- **OFDM** is primarily used in high-speed wireless communication systems such as WiFi (IEEE 802.11), LTE, and digital broadcasting. It exploits frequency division multiplexing with orthogonal subcarriers to ensure robust transmission against multipath effects.
- **CDMA** is a multiple access technique that allows several users to share the same frequency channel using unique spreading codes. It is well-known for its use in 3G networks and other mobile communication technologies.

1.2 Principles of OFDM Transmission

- **Concept of OFDM**:
  - OFDM is a **multi-carrier modulation technique** that divides a high-speed signal into multiple low-speed subcarriers. These subcarriers are **orthogonal** to each other, which helps avoid **inter-carrier interference (ICI)**.
  - Dividing the signal into subcarriers reduces the symbol rate, making OFDM highly efficient against interferences caused by path dispersion in multipath channels.

- **Orthogonality**:
  - The subcarriers in OFDM are arranged orthogonally, meaning that the spectral peaks of one subcarrier align with the zeros of others. This helps minimize ICI without requiring frequency guards.
  - The orthogonality is achieved through the use of **Fourier transform** (FFT/IFFT).
- **Structure of the OFDM Transmission Chain**:
  - **Random Bit Generation**: A binary bit stream is generated and then mapped into symbols using modulation schemes like **QPSK** or **QAM**.
  - **Modulation of Subcarriers**: The **Inverse Fast Fourier Transform (IFFT)** is used to modulate the subcarriers, converting the frequency sequence into a time sequence.
  - **Addition of Cyclic Prefix (CP)**: To prevent **Inter-Symbol Interference (ISI)** caused by multipath propagation, a cyclic prefix is added to each OFDM symbol.
  - **Channel Transmission**: The OFDM signal is transmitted over a channel, which may be a **multipath fading channel**, simulating a real propagation environment.
  - **Demodulation**: At the receiver, the cyclic prefix is removed, and an **FFT** is applied to retrieve the symbols modulated by the subcarriers.
- **Advantages of OFDM**:
  - **Resilience to Multipath Channels**: The addition of a cyclic prefix helps mitigate the effects of time dispersion.
  - **Spectral Efficiency**: The orthogonality of subcarriers minimizes the need for guard bands, thereby maximizing spectral efficiency.
  - **Reduced Complexity**: Using IFFT/FFT simplifies the modulation and demodulation processes.

1.3 Principles of CDMA Transmission

- **Concept of Spread Spectrum CDMA**:

- **CDMA** is based on **spread spectrum** technology where each user is identified by a unique **orthogonal spreading code** (e.g., Walsh codes or **Pseudo-Noise (PN) codes**).
- The spread spectrum concept involves multiplying each data bit by a longer spreading code, thereby "spreading" the signal across a wider frequency band.

- **Multiple Access**:
  - Thanks to orthogonal codes, multiple users can simultaneously transmit over the same frequency channel without significant interference, as long as the codes are perfectly orthogonal.
  - This technique increases system capacity and makes efficient use of the spectrum.
- **Structure of the CDMA Transmission Chain**:
  - **Spreading Sequence**: Each user is assigned a specific spreading sequence, which is used to modulate the data signal. This process is called **code spreading**.
  - **Modulation and Transmission**: The spread signal is transmitted through a channel, with or without interference.
  - **Reception and Despreading**: Upon reception, the signal is demodulated and correlated with the spreading sequence assigned to the user, allowing recovery of the original signal while reducing interference from other users.
- **Spreading Codes**:
  - **Walsh Codes**: Used for synchronous communication (**synchronous CDMA**), these codes are **orthogonal**, ensuring perfect separation between users when synchronization is maintained.
  - **PN Codes (Pseudo-Noise)**: Used for asynchronous communication, these codes are **quasi-orthogonal** and are more suitable for environments with low synchronization.
- **Advantages of CDMA**:
  - **Interference Resistance**: The spreading technique makes the system robust to intentional or accidental interference.

- **Security**: The signal is spread over a wide bandwidth, making interception and eavesdropping difficult without knowledge of the spreading codes.
- **Multiple Access**: Allows several users to share the same spectrum without frequent channelization, thus increasing network capacity.

1.4 Comparison between OFDM and CDMA

- **OFDM** is particularly suited for high-speed systems where multipath propagation is a major challenge, especially in urban environments with significant reflections.
- **CDMA** is effective for managing multiple access, allowing many users to share the same channel without affecting transmission quality, which made it popular for mobile networks (notably 3G).
- In terms of **resilience to multipath**, OFDM uses a cyclic prefix to mitigate effects, while CDMA relies on correlation-based reception techniques to isolate users despite fading.

1.5 Applications

- **OFDM** is used in systems such as **WiFi**, **LTE**, and **digital television broadcasting (DVB-T)**.
- **CDMA** is used in **mobile networks** (e.g., 3G) and for systems requiring secure and robust communication against noise.

2. Detailed Study of the OFDM System Simulation Blocks

2.1 Overview of the Blocks Used in the OFDM Transmission Chain

To simulate an OFDM transmission chain in Simulink, it is essential to understand the functionalities and configurations of each block. Below is a detailed description of the main blocks used:

a. Random Bit Generation

- **Block Used**: *Random Integer Generator*
- **Function**: This block generates a random sequence of bits representing the information to be transmitted. The number of bits per symbol can be defined according to the chosen modulation scheme (e.g., 2 bits for QPSK).
- **Parameters**:

    - **M-ary Number**: Choose a value according to the modulation scheme (e.g., 4 for QPSK).
    - **Sample Time**: Set the sampling period to define the bit generation frequency.

b. Mapper - Symbol Modulation

- **Block Used**: *Rectangular QAM Modulator Baseband* or *QPSK Modulator*
- **Function**: This block maps the input binary sequence to complex symbols. The modulation type can be chosen according to the compromise between robustness and data rate (e.g., QPSK or 16-QAM).
- **Parameters**:
    - **M-ary Number**: Define the modulation level (e.g., 16 for 16-QAM).
    - **Normalization Method**: Normalization to ensure constant signal power.

c. IFFT Block (Inverse Fast Fourier Transform)

- **Block Used**: *IFFT*
- **Function**: The IFFT block converts the modulated symbols into a time-domain signal. This allows the modulation of OFDM subcarriers, where each subcarrier is assigned a complex symbol.
- **Parameters**:
    - **FFT Size**: Determine the number of subcarriers (e.g., 64). This defines the transmission granularity.
    - **Cyclic Prefix Length**: To add a cyclic prefix for combating inter-symbol interference.

d. Cyclic Prefix Insertion

- **Block Used**: *Cyclic Prefix Insertion*
- **Function**: This block adds a cyclic prefix to each OFDM symbol. This helps reduce **Inter-Symbol Interference (ISI)** caused by multipath propagation.
- **Parameters**:
    - **Cyclic Prefix Length**: Adjusted according to the duration of the multipath components in the channel.

e. Multipath Channel

- **Block Used**: *Multipath Rayleigh Fading Channel* or *Multipath Rician Fading Channel*
- **Function**: Simulates the propagation environment in a real channel, which introduces fading and multiple paths. This block allows configuring several paths, each with delays and attenuation coefficients.
- **Parameters**:
  - **Number of Paths**: Specify the number of paths.
  - **Path Delays and Average Path Gains**: Determine the delay associated with each path and the respective gain.

f. Adding Noise (AWGN Channel)

- **Block Used**: *AWGN Channel*
- **Function**: Adds **Additive White Gaussian Noise (AWGN)** to the OFDM signal to simulate a realistic propagation environment.
- **Parameters**:
  - **SNR (Signal to Noise Ratio)**: Set the signal-to-noise ratio to study system robustness in the presence of noise.

g. Cyclic Prefix Removal

- **Block Used**: *Cyclic Prefix Removal*
- **Function**: This block removes the cyclic prefix before FFT demodulation to recover the original signal from the subcarriers.

h. FFT Block (Fast Fourier Transform)

- **Block Used**: *FFT*
- **Function**: Applies **Fast Fourier Transform** to the received OFDM signal to retrieve the transmitted symbols. Each subcarrier is then demodulated.
- **Parameters**:
  - **FFT Size**: Should match the size used in the IFFT on the transmitter side.

i. Demapper - Symbol Demodulation

- **Block Used**: *Rectangular QAM Demodulator Baseband* or *QPSK Demodulator*

- **Function**: This block performs the inverse of the mapper operation, converting the complex symbols back into bits.
- **Parameters**:
  - **M-ary Number**: The modulation level must match the one used during modulation.

j. Bit Error Rate (BER) Calculation

- **Block Used**: *Error Rate Calculation*
- **Function**: Calculates the **Bit Error Rate (BER)** by comparing the input bit sequence to the output bit sequence.
- **Parameters**:
  - **Receive Delay**: Set the delay due to transmission to ensure correct comparison.

2.2 Experimental Procedure for Simulation

1. **Construction of the Simulation Diagram**:
   - Add and interconnect all the blocks mentioned above.
   - Adjust the parameters to match the desired configuration (number of subcarriers, modulation level, cyclic prefix length, etc.).
2. **Configuration of the Multipath Channel**:
   - Configure the propagation channel to model multiple paths. This simulates the reality of a wireless channel where the signal reaches the receiver through different paths, creating interference.
3. **Simulation and Analysis of Results**:
   - Run the simulation in **Simulink** and observe the OFDM signal waveform.
   - Use the **Scope block** to visualize waveforms and analyze the effects of noise and multipath.
   - Calculate **BER** as a function of **SNR** to evaluate the system's robustness.
4. **Comparison of Performance**:
   - Compare the performance of the transmission with and without a cyclic prefix.

- Study the impact of the number of subcarriers on transmission quality and robustness against multipath.

3. Examples of Multipath Channels

Multipath channels play a crucial role in wireless transmission because signals can reach the receiver through multiple paths due to reflections from buildings, mountains, or other obstacles. This creates **constructive and destructive interferences** that can affect the transmission quality. Below are a few scenarios of multipath channels to study:

3.1 Two-Path Channel (Simple Model)

a. Description:

- This type of channel simulates a simplified case where the signal reaches the receiver through **one direct path** and **one reflected path**.
- The **direct path** represents the **line of sight (LOS)**, while the **reflected path** represents a reflection off an obstacle, causing additional delay and attenuation.

b. Parameters to Configure:

- **Number of Paths**: 2
- **Path Delays**:
  - Path 1 (Direct): 0 ms (no delay).
  - Path 2 (Reflected): e.g., 0.5 μs.
- **Path Gains**:
  - Path 1 (Direct): 0 dB (maximum gain).
  - Path 2 (Reflected): -5 dB (attenuation due to reflection).

c. Analysis of Effects:

- **Interference**: Both signals arrive with a slight phase difference, which can lead to **destructive interference** if the reflected path has an opposite phase to the direct path.
- **Impact on BER**: Observe the effect on **Bit Error Rate (BER)** as the delay difference between the two paths increases.

d. Simulation in Simulink:

- Use the *Multipath Rayleigh Fading Channel* block and define two paths with the parameters mentioned above.

- Observe the system response using an **oscilloscope** or **Scope block** to visualize the interference effects.

3.2 Four-Path Channel (More Realistic Model)

a. Description:

- This model is more realistic, comprising **four paths**: one direct and three reflected. This simulates an urban environment with multiple reflections from obstacles.

b. Parameters to Configure:

- **Number of Paths**: 4
- **Path Delays**:
    - Path 1 (Direct): 0 ms (line of sight).
    - Path 2: 1 µs.
    - Path 3: 2 µs.
    - Path 4: 3.5 µs.
- **Path Gains**:
    - Path 1 (Direct): 0 dB.
    - Path 2: -3 dB.
    - Path 3: -6 dB.
    - Path 4: -9 dB.

c. Analysis of Effects:

- **Path Diversity**: The presence of multiple paths causes the **superposition** of signals with different amplitudes and delays. This generates **constructive or destructive interference**.
- **Time Dispersion**: The higher the number of paths, the greater the **time dispersion**, which causes **signal distortion** and increases **Inter-Symbol Interference (ISI)**.
- **Cyclic Prefix**: Study the effect of the **cyclic prefix** on reducing interference.

d. Simulation in Simulink:

- Configure a *Multipath Rayleigh Fading Channel* block with four paths.

- Use a **Scope block** to observe the modulation constellation (e.g., QPSK) before and after equalization of the channel to see the impact of multipath.
- Compare the performance with and without a cyclic prefix.

3.3 Six-Path Channel (Extreme Scenario)

a. Description:

- A **six-path channel** represents a scenario in very dense environments, such as urban centers with many buildings causing significant signal dispersion.

b. Parameters to Configure:

- **Number of Paths**: 6
- **Path Delays**:
  - Path 1: 0 ms (line of sight).
  - Path 2: 0.8 µs.
  - Path 3: 1.5 µs.
  - Path 4: 3 µs.
  - Path 5: 4 µs.
  - Path 6: 5 µs.
- **Path Gains**:
  - Path 1: 0 dB.
  - Path 2: -2 dB.
  - Path 3: -4 dB.
  - Path 4: -6 dB.
  - Path 5: -8 dB.
  - Path 6: -10 dB.

c. Analysis of Effects:

- **Severe Fading**: With an increasing number of paths, fading becomes more complex. This can be simulated as **Rayleigh fading** or **Rician fading**, depending on whether there is a dominant LOS component.
- **Impact on BER**: Analyze the effect of increased delay and gain dispersion on **BER** performance.

d. Simulation in Simulink:

- Use a *Multipath Rayleigh Fading Channel* with six paths to simulate multipath effects in dense environments.
- Measure the **BER** for different **SNR** levels and compare the results with those obtained for two and four-path channels.

3.4 Comparison of Different Scenarios

- **BER vs SNR Curve Analysis**:
    - For each channel configuration (2, 4, and 6 paths), plot the **BER as a function of SNR**. This comparison will highlight the degradation in performance due to the increased number of paths.
    - Observe the effectiveness of the **cyclic prefix** in reducing **ISI**, especially in scenarios with a large number of paths.
- **Student Conclusions**:
    - Students should conclude on the impact of multiple paths and discuss the need for techniques such as **cyclic prefix** to improve transmission robustness.
    - Compare the performance of **OFDM** and **CDMA** techniques in the presence of multipath, noting how each manages **interference** and **fading**.

4. Instructions for Students

To help students make the most of this TP, here are the detailed instructions to follow:

4.1 Step-by-Step Realization

4.1.1. Creating the Simulation Model in Simulink:

Step 1: Define the Inputs:

- Start by adding the necessary blocks for **random bit generation**, as well as a **mapper** to modulate the generated bits.

Step 2: Construct the OFDM Transmission Chain:

- Add blocks such as **IFFT**, **cyclic prefix insertion**, **multipath channel**, and **FFT**.
- Configure the blocks based on the instructions given in the TP.

Step 3: Add the AWGN Channel:

- Use the *AWGN Channel* block to introduce noise and observe its impact on signal quality.

Step 4: Comparison of Configurations:

- Perform several simulations with different channel configurations (2, 4, and 6 paths) and varying **SNR** levels.

Step 5: Construct the CDMA Transmission Chain:

- Simulate a multi-user system with at least four users, using orthogonal spreading codes.

4.1.2 Using Analysis Blocks

- **Scope (Oscilloscope)**: Add **scope** blocks to visualize the waveforms at different stages of the transmission chain.
- **Constellation Diagram**: Use a **Constellation Diagram** block to observe the constellation of the symbols before and after transmission.

4.1.3 Simulation of Multipath Channels

- For each channel configuration, simulate the **multipath environment** by adjusting the **delays** and **gains** for each path.
- Compare the results with and without the **cyclic prefix** to observe the reduction in interference.

4.1.4 SNR Adjustment

- Vary the **Signal-to-Noise Ratio (SNR)** in the AWGN channel and observe the **BER vs SNR** curves.

- Analyze how different channel configurations (number of paths) influence system performance.

4.1.5 Performance Analysis

- Study the impact of modulation type (**QPSK vs QAM**) on the **Bit Error Rate (BER)**.
- Compare the performance of the **OFDM** and **CDMA** transmission techniques in terms of **robustness to interference** and **BER**.

4.2 Reflection Questions

- What is the importance of the cyclic prefix in OFDM transmission? How does it improve performance in the presence of multipath channels?
- What is the impact of orthogonal codes on the quality of transmission in a multi-user CDMA system?
- Why is OFDM better suited for multipath channels compared to other modulation techniques?
- Compare the BER curves obtained for different SNR levels and analyze the effect of the number of paths on transmission quality.

5. Deliverables

At the end of the TP, students are expected to submit the following:

5.1 TP Report

The report should include the following sections:

- **Introduction**: Theoretical background on **OFDM** and **CDMA**, and the importance of multipath channels.
- **Description of the Simulation Blocks**: Present each block used in the simulation and explain its role.
- **Simulink Diagrams**: Include screenshots of the **OFDM** and **CDMA** simulation models.
- **Results and Analysis**:
    - Graphs of the **constellations** of symbols before and after transmission.
    - **BER vs SNR** graphs for different channel configurations (2, 4, and 6 paths).

    - Analysis of the impact of the **cyclic prefix** and multipath on performance.

- **Conclusion**: Summary of the results obtained, comparisons of the performance of OFDM and CDMA techniques, and recommendations for possible improvements.

5.2 Simulink Files

- Submit the **Simulink files (*.slx)** for both the **OFDM** and **CDMA** simulations for evaluation.
- Each file should be clearly annotated, using text blocks to explain each stage of the transmission chain.

5.3 Presentation

- Prepare a **presentation** (PowerPoint or PDF) summarizing the TP results and the conclusions drawn from the simulations.
- Students should be able to present the structure of the models created, explain the parameter choices for the blocks, and discuss the results obtained.

6. Conclusion

This TP is designed to allow students to apply their theoretical knowledge of digital transmission. By simulating OFDM and CDMA systems in Simulink, they will understand the importance of different techniques to mitigate interference and improve the quality of transmissions in multipath environments.
They will also see the impact of multiple paths on system performance and how concepts such as the cyclic prefix or spread spectrum contribute to enhancing signal robustness.

Appendix: Simulink Diagram for OFDM and CDMA Transmission

1. Simulink Diagram for OFDM Transmission

Steps to Build the OFDM Diagram:

1. **Random Bit Generation**
   - Use a **Random Integer Generator** block.
   - Set the parameters to generate a random sequence of bits.
2. **Map Bits to Symbols (Modulation)**
   - Use a **QPSK Modulator Baseband** or **Rectangular QAM Modulator Baseband** block.
   - Connect the output of the bit generator to the modulator.
3. **Inverse Fast Fourier Transform (IFFT)**
   - Use an **IFFT** block.
   - Connect the output of the modulation block to the input of the IFFT.
   - Set the IFFT size according to the number of subcarriers.
4. **Cyclic Prefix Insertion**
   - Use a **Cyclic Prefix Insertion** block.
   - Connect the output of the IFFT to the cyclic prefix block.
   - Set the length of the cyclic prefix.
5. **Multipath Channel with Noise (Rayleigh or AWGN)**
   - Use a **Multipath Rayleigh Fading Channel** block.
   - Connect the output of the cyclic prefix to the channel.
   - Add an **AWGN Channel** block after the Rayleigh channel to simulate noise.

6. **Cyclic Prefix Removal**
   - Use a **Cyclic Prefix Removal** block.
   - Connect the output of the noisy channel to this block.
7. **Fast Fourier Transform (FFT)**
   - Use an **FFT** block.
   - Connect the output of the cyclic prefix removal block to the input of the FFT.
8. **Symbol Demodulation**
   - Use a **QPSK Demodulator Baseband** or **Rectangular QAM Demodulator Baseband** block.
   - Connect the output of the FFT to the demodulator.
9. **Bit Error Rate (BER) Calculation**
   - Use an **Error Rate Calculation** block.
   - Compare the original input bits with the demodulated bits to measure the **BER**.
   - Add **Display** and **Scope** blocks to visualize the results.
10. **Constellation Visualization**

- Use a **Constellation Diagram** block connected after the modulation block and after the channel to observe the quality of the symbol constellation.

2. Simulink Diagram for CDMA Transmission

Steps to Build the CDMA Diagram:

1. **Random Bit Generation for Multiple Users**
   - Use multiple **Random Integer Generator** blocks to simulate bits from multiple users (e.g., 4 users).
   - Each block will generate a distinct sequence of bits.
2. **Spreading Codes (Orthogonal Codes)**
   - Use a **PN Sequence Generator** or **Walsh Code Generator** block for each user.
   - Generate orthogonal codes for each bit sequence to be transmitted.

3. **Modulation of the Spread Sequence**
   - Spread each bit sequence by multiplying it with the corresponding orthogonal code.
   - Use **Product** blocks to perform the spreading.
4. **Addition of Spread Signals**
   - Use a **Sum** block to add the signals from each user.
   - This simulates multi-user transmission over a single channel.
5. **Transmission through a Noisy Channel**
   - Use an **AWGN Channel** or **Rayleigh Fading Channel** block to add noise to the combined signal.
6. **Despreading at Reception**
   - Use a **Product** block to multiply the received signal by the orthogonal code of each user.
   - This allows the specific data of each user to be recovered.
7. **Demodulation and Decoding**
   - Use a **QPSK Demodulator Baseband** block for each user to demodulate the signal.
   - Compare the demodulated signal with the original signal for each user.
8. **Bit Error Rate (BER) Calculation**
   - Use an **Error Rate Calculation** block for each user to measure the error rate.

Visual Diagram For a visual diagram in **Simulink**, you can:

- Create **subsystems** for each important part of the system (bit generation, modulation, channel, demodulation) to make the diagram clearer and more readable.
- Use **Scope** blocks to observe signals at different points of the chain, allowing analysis of the effects of multiple paths, noise, and the cyclic prefix.

Configuration Example

- **Number of Subcarriers (OFDM)**: 64.

- **Modulation**: QPSK or 16-QAM.
- **Number of Users (CDMA)**: 4.
- **SNR (AWGN)**: Vary the SNR to observe its impact on BER.
- **Multipath Delays**: Set delays and gains based on given examples (2, 4, and 6 paths).

Lab 4: Simulation of a MIMO Transmission Chain

Objectives of the Lab:

- Understand the basic concepts of MIMO (Multiple Input, Multiple Output) systems.
- Simulate a MIMO transmission chain in a software environment (e.g., MATLAB).
- Observe and analyze the performance of MIMO transmission based on different parameters.

1. Theoretical Introduction:

1.1. Basic Concepts of MIMO Systems

1.1.1. Definition of MIMO

MIMO, or Multiple Input Multiple Output, is a transmission technology that uses multiple antennas at both the transmitter and receiver. Unlike SISO (Single Input Single Output) systems, which use a single antenna for transmission and reception, MIMO systems make better use of spatial resources and improve performance.

1.1.2. Operating Principles

In MIMO systems, multiple antennas simultaneously transmit signals that are then received by multiple antennas. These signals undergo multiple paths through the channel, creating diversity. Using advanced signal processing techniques, the MIMO system exploits this diversity to:

- Increase transmission reliability: by reducing the impact of signal fading (antenna diversity).
- Increase transmission rate: by exploiting spatial multiplexing, allowing multiple data streams to be transmitted in parallel.

1.1.3. Spatial Multiplexing and Diversity Gain

- **Spatial Multiplexing**: In this mode, MIMO uses each antenna to send an independent data stream, increasing the system's capacity. For instance, a 2x2 system (2 transmit antennas, 2 receive antennas) can double the rate compared to a SISO system.
- **Diversity Gain**: Diversity gain helps reduce transmission errors. When multiple antennas receive the signal, it is less likely that all copies will be affected by interference or fading, leading to more reliable reception even in challenging propagation conditions.

1.1.4. Propagation Channels

MIMO systems depend on the nature of the propagation channel. The two most common channel models are:

- **Rayleigh Channel**: Used to model urban environments with significant obstacles, suitable for scenarios with many indirect paths (multipath).
- **Rician Channel**: Models environments where there is a dominant line of sight in addition to reflected paths.

1.1.5. Factors Influencing MIMO System Performance

- **Signal-to-Noise Ratio (SNR)**: Higher SNR results in better reception quality, as the noise affects the signals less.
- **Number of Antennas**: Generally, the more antennas, the better the performance, but it also increases system complexity.
- **Antenna Spacing**: Good spacing reduces interference between antennas and improves the quality of received signals.

1.1.6. Advantages and Applications of MIMO Systems

- **Advantages**: MIMO systems increase the capacity and reliability of wireless communications, enabling higher data rates and more stable connections.
- **Applications**: MIMO is used in many modern technologies, such as Wi-Fi (IEEE 802.11n/ac/ax standards), 4G and 5G mobile networks, and high-speed communication systems.

1.1.7. Practical Example

To illustrate the operation of a 2x2 MIMO transmission chain using MATLAB, we can simulate a simple MIMO system using BPSK (Binary Phase Shift Keying) modulation over a Rayleigh channel.

Example Code for This Simulation:

```
% Simulation Parameters
numTx = 2;           % Number of transmit antennas
numRx = 2;           % Number of receive antennas
numSymbols = 1e5;    % Number of symbols transmitted
SNR_dB = 10;         % Signal-to-noise ratio in dB

% Data Generation (BPSK)
data = randi([0 1], numSymbols, numTx); % Binary data
bpskModulated = 2*data - 1;            % BPSK modulation: 0 -> -1, 1 -> 1
```

```
% Rayleigh Channel
rayleighChannel = (randn(numSymbols, numRx, numTx) + 1i*randn(numSymbols,
numRx, numTx)) / sqrt(2);

% Additive Noise (AWGN)
SNR_linear = 10^(SNR_dB / 10);
noiseStdDev = 1 / sqrt(2 * SNR_linear);
noise = noiseStdDev * (randn(numSymbols, numRx) + 1i*randn(numSymbols,
numRx));

% Transmission through Rayleigh Channel
receivedSignal = zeros(numSymbols, numRx);
for i = 1:numRx
    for j = 1:numTx
        receivedSignal(:, i) = receivedSignal(:, i) + rayleighChannel(:,
i, j) .* bpskModulated(:, j);
    end
end

% Add Noise to Received Signal
receivedSignal = receivedSignal + noise;

% Maximum Ratio Combining Detection
estimatedData = real(receivedSignal .* sum(conj(rayleighChannel), 2)) > 0;

% Error Rate Calculation
numErrors = sum(sum(data ~= estimatedData));
BER = numErrors / (numSymbols * numTx);

% Display Results
disp(['Bit Error Rate (BER): ' num2str(BER)]);
```

Code Explanation:

- **Simulation Parameters**: Defines the number of transmit (Tx) and receive (Rx) antennas, the number of symbols, and SNR.
- **Data Generation**: Random binary data is generated for each antenna, then modulated using BPSK.
- **Rayleigh Channel Model**: Simulates a Rayleigh channel for each Tx/Rx antenna combination with random complex numbers.
- **Adding Noise**: Adds white Gaussian noise to simulate realistic transmission conditions.
- **Transmission through Channel**: Signal is multiplied by the channel for each Tx/Rx link, and signals are summed at each receive antenna.
- **Decoding**: The received signal is decoded using Maximal Ratio Combining (MRC), optimizing the signal to minimize noise and interference.

- **Error Rate Calculation**: The bit error rate (BER) is calculated by comparing transmitted and received data.

Expected Results:

You will obtain a bit error rate (BER) that depends on the SNR and propagation conditions. You can adjust parameters, such as SNR_dB, to observe how it affects the MIMO system's performance.

1.2. Presentation of Important Parameters

To understand the advantages of MIMO systems, it is essential to present some fundamental parameters that play a determining role in wireless transmission performance.

1.2.1. Diversity Gain

Diversity gain is a key factor in MIMO technology, aimed at improving transmission reliability. By using multiple transmit and receive antennas, a MIMO system can benefit from multiple independent signal paths, reducing the likelihood that all paths are simultaneously affected by phenomena such as fading. The greater the number of antennas, the higher the diversity gain, which improves the overall reception quality by reducing the error rate.

1.2.2. Throughput

Throughput (or data rate) is another major advantage of MIMO systems. By exploiting spatial multiplexing, multiple data streams can be sent simultaneously by different antennas, thus increasing system capacity in proportion to the number of antennas. For example, a 2x2 MIMO system can double the rate compared to a SISO (Single Input Single Output) system, since two data streams are transmitted simultaneously.

1.2.3. Signal-to-Noise Ratio (SNR)

The signal-to-noise ratio (SNR) is a measure of the quality of a transmission signal. A high SNR indicates that the signal is much stronger than the noise, allowing better demodulation and reducing the error rate. In MIMO systems, SNR is particularly important, as combining the signals from multiple antennas can artificially increase this ratio, especially using techniques such as Maximal Ratio Combining (MRC), which optimizes the weighted sum of the received signals.

In summary, MIMO systems use diversity gain to improve transmission reliability, spatial multiplexing to increase throughput, and signal processing techniques to optimize the signal-to-noise ratio. This allows them to meet the growing needs for data rate and quality of wireless communications, particularly in complex environments where propagation conditions are unfavorable.

2. Setting Up the Simulation:

The objective of this simulation setup is to allow students to configure a MIMO transmission chain, understand the impact of different parameters (number of antennas, channel type, SNR), and observe the system's performance in realistic propagation environments.

2.1. Antenna Configuration

- **Number of Transmit Antennas (Tx)**: Determine the number of transmit antennas, e.g., two antennas (Tx1 and Tx2) for a 2x2 MIMO configuration.
- **Number of Receive Antennas (Rx)**: Set the number of receive antennas, e.g., two antennas (Rx1 and Rx2). This configuration allows simulating diversity and spatial multiplexing.

2.2. Channel Model Selection

The choice of the channel model is crucial, as it determines how the signal is propagated and affected by the environment:

- **Rayleigh Channel**: The Rayleigh channel is used to simulate environments rich in indirect paths, such as urban areas with no direct line of sight. This model is appropriate for capturing the effects of multipath propagation, characteristic of complex propagation environments.
- **Rician Channel (optional)**: This channel can be used to model situations where a dominant line of sight exists in addition to indirect paths. Students can compare performance under Rayleigh and Rician conditions to better understand the impact of the channel on MIMO transmission.

2.3. Modulation Selection

For modulation, **BPSK (Binary Phase Shift Keying)** is a good starting option, as it is simple to understand and implement. Students can also experiment with other modulation techniques, such as **QPSK (Quadrature Phase Shift Keying)** or **16-QAM (16-Quadrature Amplitude Modulation)**, to observe the effects on the bit error rate (BER).

2.4. Noise and SNR Parameters

- **Noise (AWGN - Additive White Gaussian Noise)**: Noise is added to simulate realistic transmission conditions. It is important to configure the noise in such a way as to achieve an SNR representative of real conditions.

- **Signal-to-Noise Ratio (SNR)**: Define the SNR in dB for different values, such as 5 dB, 10 dB, 15 dB, to see how transmission quality varies with noise.

2.5. Data Generation and Transmission

- **Data to Transmit**: Generate random data (bits 0 and 1) for each transmit antenna.
- **Transmission Through the Channel**: Transmit the data through the Rayleigh channel by applying the channel characteristics to the modulated signals. It is important to simulate the interactions between the transmit and receive antennas, taking into account the effects of multipath.

2.6. Adding Noise

Noise is added to the received signals to simulate real propagation conditions. This allows students to observe the impact of noise on the performance of the MIMO system.

2.7. Tools Used

- **MATLAB**: For this simulation, MATLAB will be used. You will configure channel, modulation, and noise parameters in this software environment to analyze MIMO system performance.
- **Detection and Decoding Block**: For processing the received signals, it is necessary to set up a detection block that implements a method such as **Maximal Ratio Combining (MRC)** or **Zero-Forcing** to estimate the transmitted data from the received signals.

3. Simulation Steps

In this section, we will follow a series of steps to perform the simulation of a MIMO transmission chain. These steps will allow us to understand how each system component interacts from the perspective of signal transmission and reception.

3.1. Implementing the MIMO Chain

We will start by implementing the MIMO chain in MATLAB. We will configure the number of transmit (Tx) and receive (Rx) antennas, using a 2x2 system.
We will also define the data to be transmitted. To do this, we will generate random bits (0 and 1), which we will modulate using **BPSK** modulation.

3.2. Simulating Transmission Through the Rayleigh Channel

We will then transmit the modulated data through the Rayleigh channel. For each transmit antenna, the signal will be propagated through the channel, simulating multipath.

The Rayleigh channel will be modeled using random complex coefficients, reflecting the nature of wireless propagation with random fading.

3.3. Adding Noise to the Received Signal

To make our simulation realistic, we will add **Additive White Gaussian Noise (AWGN)** to the received signals. This will allow us to simulate real transmission conditions where signals are always affected by environmental noise.
We will use different signal-to-noise ratio (SNR) values to see how they affect the performance of the transmission chain.

3.4. Receiving and Combining Signals

Each receive antenna will receive a combination of the signals sent by the different transmit antennas. We will then use a method like **Maximal Ratio Combining (MRC)** or **Zero-Forcing** to combine the received signals.
The goal is to recover the original data while minimizing the effect of interference and noise. We will use signal processing algorithms to estimate the transmitted bits.

3.5. Data Decoding and Error Rate Estimation

Once the signals are combined, we will proceed with demodulation to obtain the received binary data.
Then, we will compare the received data with the initial data to calculate the **bit error rate (BER)**. This step will allow us to measure the performance of our MIMO system under different SNR and propagation conditions.

3.6. Visualization and Analysis of Results

We will plot **BER vs. SNR curves** for different configurations (e.g., 2x1, 2x2, etc.) to understand the impact of the number of antennas and the quality of propagation on performance.
At the end, we will discuss the results and draw conclusions about the efficiency of MIMO systems, notably the importance of antenna diversity and the impact of the channel on transmission quality.

4. Result Analysis

In this section, we will analyze the results obtained from simulating the MIMO transmission chain, focusing on performance in terms of **bit error rate (BER)** for different configurations and on the impact of the number of antennas on transmission quality.

4.1 Measuring Performance in Terms of BER for Different Configurations

To measure performance, we need to calculate the **bit error rate (BER)** under different conditions, such as varying the signal-to-noise ratio (SNR), the number of transmit and receive antennas, and the type of modulation used.

- **Varying the SNR**: Run the simulation with different SNR values (e.g., 5 dB, 10 dB, 15 dB) and observe how the BER changes. Generally, increasing the SNR leads to a reduction in BER.

- **Changing Modulation Type**: Compare the performance of QPSK modulation with other types, such as BPSK or 16-QAM. This will allow us to see how system robustness is affected by the chosen modulation.

- **Comparing MIMO Configurations**: Vary the number of transmit and receive antennas (e.g., MIMO 2x1, MIMO 2x2, MIMO 4x4) to understand how it influences BER. A higher number of antennas can reduce the BER by increasing reception diversity.

4.2 Analyzing the Advantages of MIMO Compared to SISO Systems

To better understand the advantages of MIMO, we will compare the performance of our MIMO system with that of a **SISO (Single Input, Single Output)** system.
In a SISO system, there is only one transmit and one receive antenna, which limits system capacity and makes the signal more susceptible to fading effects.
MIMO offers gains in terms of diversity and spatial multiplexing:

- **Diversity Gain**: The MIMO system can combine multiple copies of the received signal, reducing the risk of deep fading and improving reliability.

- **Spatial Multiplexing**: MIMO allows multiple data streams to be transmitted in parallel, increasing channel capacity compared to SISO.

For this comparison, we could run the same simulation in SISO mode, configuring numTx and numRx to 1, and then compare the BER with that of the MIMO system. We should observe that MIMO offers a lower BER, particularly in low SNR environments, and a higher data rate due to spatial multiplexing.

4.3 Observing the Impact of the Number of Tx and Rx Antennas on Performance

- **Number of Transmit Antennas (Tx)**: Increasing the number of transmit antennas diversifies data transmission and reduces interference. In MIMO 4x2 mode (4 transmit antennas and 2 receive antennas), for example, it is possible to send more parallel data streams, thereby increasing the data rate.

- **Number of Receive Antennas (Rx)**: Increasing the number of receive antennas enhances diversity, reducing the effect of fading and improving the quality of the received signal. More receive antennas allow multiple versions of the signal with different paths to be combined, which is particularly beneficial in high multipath environments.

For this analysis, we will simulate different configurations, such as **MIMO 2x2**, **MIMO 2x4**, and **MIMO 4x4**, and compare the corresponding BERs. In general, the higher the number of antennas, the more robust and efficient the system becomes, and the BER decreases.

Conclusion of Result Analysis

By performing this analysis, we should conclude that:

- **MIMO** offers significant improvements in terms of data rate and reliability compared to **SISO** systems.
- Increasing the number of antennas improves performance, reduces BER, and increases data transmission rate.
- The choice of simulation parameters, such as SNR, modulation, and antenna configuration, has a significant impact on transmission quality and overall MIMO system performance.

5. Reflection Questions

To conclude this lab on MIMO system simulation, we will address a few reflection questions that will help you better understand the concepts and results obtained.

5.1 Why Does the MIMO System Offer Better Performance?

MIMO (Multiple Input Multiple Output) uses multiple transmit and receive antennas, allowing it to exploit two main phenomena: **diversity** and **spatial multiplexing**.

- **Diversity Gain**: By using multiple antennas, the MIMO system can receive multiple copies of the same signal. This diversity compensates for fading, as it is unlikely that all copies of the signal will fade simultaneously. This significantly increases transmission reliability.
- **Spatial Multiplexing**: Thanks to this technique, MIMO can transmit multiple independent data streams in parallel, increasing data rate without requiring more bandwidth. This results in a higher channel capacity compared to a SISO system.

In summary, **MIMO** improves both robustness (reducing the error rate) and data rate (increasing capacity), making the system more efficient for modern wireless applications.

5.2 How Does Channel Selection Affect Performance?

The propagation channel has a significant impact on the performance of a wireless communication system. In the case of MIMO, we have mainly considered the **Rayleigh** and **Rician** channels.

- **Rayleigh Channel**: This model is used for environments rich in multipath without a dominant line of sight, such as urban environments. The Rayleigh channel introduces significant fading, meaning that signal strength can vary randomly. In such an environment, MIMO's diversity gain is essential for improving performance and reducing **bit error rate (BER)**.

- **Rician Channel**: This channel is used when the environment has both a direct line of sight and multipath. The presence of a line-of-sight component allows for more stable signal reception compared to the Rayleigh channel. Choosing the Rician channel generally results in better performance in terms of reliability and data rate, as there are fewer random variations in received signal power.

6. Conclusion

The choice of channel model directly affects transmission performance, as it determines the level of diversity and fading conditions the signals are subjected to. **MIMO** is particularly effective in complex channels with significant multipath, as it exploits the available diversity to maintain reliable transmission.

Appendix: MATLAB Code of a MIMO Transmission Chain

```
% Simulation Parameters
numTx = 2;           % Number of transmit antennas
numRx = 2;           % Number of receive antennas
numSymbols = 1e5;    % Number of symbols transmitted
SNR_dB = 10;         % Signal-to-noise ratio in dB

% Data Generation (QPSK)
data = randi([0 3], numSymbols, numTx); % Random data for QPSK
qpskModulated = pskmod(data, 4, pi/4);  % QPSK modulation with phase shift

% Rayleigh Channel
rayleighChannel = (randn(numSymbols, numRx, numTx) + 1i*randn(numSymbols,
numRx, numTx)) / sqrt(2);

% Additive Noise (AWGN)
SNR_linear = 10^(SNR_dB / 10);
noiseStdDev = 1 / sqrt(2 * SNR_linear);
noise = noiseStdDev * (randn(numSymbols, numRx) + 1i*randn(numSymbols,
numRx));

% Transmission via Rayleigh Channel
receivedSignal = zeros(numSymbols, numRx);
for i = 1:numRx
    for j = 1:numTx
        receivedSignal(:, i) = receivedSignal(:, i) + rayleighChannel(:, i,
j) .* qpskModulated(:, j);
    end
end

% Adding Noise to Received Signal
receivedSignal = receivedSignal + noise;

% Zero-Forcing Detection
estimatedSymbols = zeros(numSymbols, numTx);
for k = 1:numSymbols
    H = reshape(rayleighChannel(k, :, :), numRx, numTx); % Reshape channel
for each symbol
    H_inv = pinv(H); % Pseudo-inverse of the channel
    estimatedSymbols(k, :) = H_inv * receivedSignal(k, :).';
end

% QPSK Demodulation
demodulatedData = pskdemod(estimatedSymbols, 4, pi/4);

% Error Rate Calculation
numErrors = sum(sum(data ~= demodulatedData));
BER = numErrors / (numSymbols * numTx);

% Display Results
disp(['Bit Error Rate (BER): ' num2str(BER)]);
```

General Conclusion

This document has provided a comprehensive exploration of digital communication systems, emphasizing both theoretical understanding and practical application through MATLAB and Simulink simulations. By studying key concepts such as digital modulation techniques (BASK, BPSK, BFSK, QAM), advanced transmission methods (OFDM, CDMA, MIMO), and performance metrics like bit error rate and spectral efficiency, students have gained a holistic view of modern telecommunications.

The labs demonstrated the trade-offs inherent in various modulation and transmission schemes, highlighting the balance between noise robustness, spectral efficiency, and implementation complexity. Through simulations, students analyzed the impact of channel conditions, noise, and multipath effects, equipping them with the analytical tools to design and evaluate communication systems.

In conclusion, this document bridges the gap between theory and real-world applications, empowering students to apply their knowledge in addressing the challenges of evolving communication networks. The skills and insights gained from these labs will serve as a foundation for innovation and excellence in the dynamic field of digital communications.

References

[1]. Sklar, B. (2001). *Digital Communications: Fundamentals and Applications*. Prentice Hall.

[2]. Proakis, J. G., & Salehi, M. (2008). *Digital Communications*. McGraw-Hill.

[3]. Rappaport, T. S. (2002). *Wireless Communications: Principles and Practice*. Prentice Hall.

[4]. MathWorks Documentation. *Simulink and MATLAB Documentation*. https://www.mathworks.com

[5]. Haykin, S. (2009). *Communication Systems*. Wiley.

[6]. Goldsmith, A. (2005). *Wireless Communications*. Cambridge University Press.

[7]. Molisch, A. F. (2012). *Wireless Communications*. Wiley-IEEE Press.

[8]. Anderson, J. B. (1995). *Digital Transmission Engineering*. Wiley.

[9]. Tse, D., & Viswanath, P. (2005). *Fundamentals of Wireless Communication*. Cambridge University Press.

[10]. Van Trees, H. L. (2001). *Detection, Estimation, and Modulation Theory*. Wiley.

[11]. Viterbi, A. J. (1995). *CDMA: Principles of Spread Spectrum Communication*. Addison-Wesley.

[12]. Andrews, J. G., & Ghosh, A. (2007). *Fundamentals of WiMAX: Understanding Broadband Wireless Networking*. Prentice Hall.

[13]. Paulraj, A., Nabar, R., & Gore, D. (2003). *Introduction to Space-Time Wireless Communications*. Cambridge University Press.

[14]. Richardson, T. & Urbanke, R. (2008). *Modern Coding Theory*. Cambridge University Press.

[15]. Cover, T. M., & Thomas, J. A. (2006). *Elements of Information Theory*. Wiley.

[16]. Madhow, U. (2008). *Fundamentals of Digital Communication*. Cambridge University Press.