**Lab Exercise: Simulation of a MIMO Transmission Chain**

**Objectives:**

* Understand the theoretical concepts of MIMO transmission systems.
* Simulate a MIMO transmission chain using MATLAB/Simulink.
* Analyze the performance of different MIMO configurations in terms of bit error rate (BER) and capacity.

**Prerequisites:**

* Basic knowledge of MIMO communication techniques.
* Familiarity with MATLAB and Simulink.

**1. Introduction to MIMO Systems**

**1.1 Theoretical Background**

* **MIMO (Multiple Input Multiple Output):** MIMO is a wireless communication technology that uses multiple transmitters and receivers to send and receive more data simultaneously. It exploits spatial diversity to improve the capacity and reliability of wireless communication systems. MIMO techniques are widely used in modern communication standards, such as LTE and Wi-Fi.

**1.2 Objectives of the Lab**

* Simulate a MIMO transmission system using MATLAB/Simulink.
* Understand the role of each block in the MIMO system.
* Analyze the performance of different MIMO configurations (e.g., 2x2, 4x4) in terms of BER and channel capacity.

**2. Building the MIMO Transmission System in Simulink**

**2.1 Key Components of the MIMO System**

In this section, students will build a MIMO transmission system using Simulink. The key blocks in the system include:

* **Data Source:** Generates the binary data stream to be transmitted.
* **QAM Modulator:** Maps the binary data onto QAM symbols.
* **MIMO Encoder:** Distributes the QAM symbols across multiple transmit antennas.
* **Channel Model:** Simulates the MIMO channel, which may include Rayleigh fading, Rician fading, and noise.
* **MIMO Decoder:** Combines the signals received by multiple antennas to recover the transmitted data.
* **QAM Demodulator:** Demodulates the received symbols to retrieve the binary data.
* **BER Calculation Block:** Compares the transmitted and received data to calculate the bit error rate (BER).

**2.2 Simulation Steps**

1. **Create the Simulink Model:**
   * Open MATLAB and start Simulink.
   * Create a new Simulink model and add the blocks mentioned above.
   * Connect the blocks to form a complete MIMO transmission and reception system.
2. **Configure the Parameters:**
   * Set the QAM modulation order (e.g., 16-QAM).
   * Define the number of transmit and receive antennas (e.g., 2x2 MIMO, 4x4 MIMO).
   * Configure the MIMO channel model with appropriate parameters (e.g., channel fading, noise).
3. **Run the Simulation:**
   * Run the Simulink model and observe the output.
   * Record the BER for different SNR values and MIMO configurations.
4. **Analyze the Results:**
   * Plot the BER versus SNR graph for different MIMO configurations.
   * Discuss the impact of the number of antennas and the channel conditions on the system performance.

**3. Performance Analysis of MIMO Systems**

**3.1 BER Analysis**

Students will analyze the bit error rate (BER) performance of different MIMO configurations (e.g., 2x2, 4x4) under various channel conditions (e.g., Rayleigh fading, Rician fading).

* **Rayleigh Fading:** Simulates a multipath environment with no line-of-sight (NLOS) components.
* **Rician Fading:** Simulates a multipath environment with a dominant line-of-sight (LOS) component.

% Example MATLAB code for BER analysis in a MIMO system

clear;

clc;

N = 10^6; % Number of bits

Eb\_N0\_dB = 0:2:20; % SNR values in dB

% Define MIMO configuration

numTx = 2; % Number of transmit antennas

numRx = 2; % Number of receive antennas

% Generate random binary data

data = randi([0 1], N, 1);

% Modulate using QAM

modData = qammod(data, 16, 'InputType', 'bit', 'UnitAveragePower', true);

% Reshape for MIMO transmission

modData = reshape(modData, N/numTx, numTx);

% Define MIMO channel matrix (Rayleigh fading)

H = 1/sqrt(2) \* (randn(numRx, numTx, N/numTx) + 1i\*randn(numRx, numTx, N/numTx));

% Initialize BER vector

BER = zeros(1, length(Eb\_N0\_dB));

% Loop through different SNR values

for i = 1:length(Eb\_N0\_dB)

% Add noise

noise = (1/sqrt(2))\*(randn(numRx, N/numTx) + 1i\*randn(numRx, N/numTx));

r = squeeze(sum(H .\* reshape(modData, 1, numTx, N/numTx), 2)) + 10^(-Eb\_N0\_dB(i)/20) \* noise;

% MIMO detection (Zero Forcing)

detectedData = qamdemod(H \ r, 16, 'OutputType', 'bit', 'UnitAveragePower', true);

% Calculate BER

BER(i) = sum(sum(data ~= reshape(detectedData, N, 1)))/N;

end

% Plot BER results

figure;

semilogy(Eb\_N0\_dB, BER, 'b-o');

xlabel('Eb/N0 (dB)');

ylabel('Bit Error Rate (BER)');

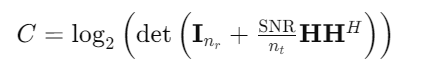
title('BER Performance of 2x2 MIMO System');

grid on;

**3.2 Capacity Analysis**

Students will calculate and analyze the capacity of different MIMO configurations under various channel conditions.

* **Shannon Capacity Formula:**



Where nrn\_rnr​ is the number of receive antennas, ntn\_tnt​ is the number of transmit antennas, H\mathbf{H}H is the MIMO channel matrix, and CCC is the channel capacity.

% Example MATLAB code for Capacity analysis in a MIMO system

capacity = zeros(1, length(Eb\_N0\_dB));

% Loop through different SNR values

for i = 1:length(Eb\_N0\_dB)

snr\_linear = 10^(Eb\_N0\_dB(i)/10);

for j = 1:N/numTx

H\_j = squeeze(H(:,:,j));

capacity(i) = capacity(i) + log2(det(eye(numRx) + (snr\_linear/numTx) \* H\_j \* H\_j'));

end

capacity(i) = capacity(i) / (N/numTx); % Average capacity

end

% Plot capacity results

figure;

plot(Eb\_N0\_dB, capacity, 'r-s');

xlabel('Eb/N0 (dB)');

ylabel('Capacity (bits/s/Hz)');

title('Capacity of 2x2 MIMO System');

grid on;

**4. Comparison of MIMO Configurations**

**4.1 Performance Comparison**

* Compare the BER and capacity of different MIMO configurations (e.g., 2x2, 4x4) under the same channel conditions.
* Discuss the trade-offs between the number of antennas, system complexity, and performance.

**4.2 Impact of Channel Conditions**

* Analyze how different channel conditions (e.g., Rayleigh vs. Rician fading) affect the performance of MIMO systems.
* Discuss strategies to mitigate the effects of fading and improve MIMO system performance.