% Simulation Parameters

numUsers = 4;

numSubcarriers = 64;

snrRange = 0:1:10; % Finer SNR steps to capture BER curve shape more precisely

infoBitsLength = 1e6; % Set infoBitsLength to 10^6

repetitionFactor = 3; % Repetition factor for simple error correction

diversityLevels = [1, 2, 3, 4]; % Diversity levels for multiple paths

% Set random seed for repeatability

rng(1);

% Initialize result matrices

berResults = zeros(length(snrRange), numUsers, length(diversityLevels)); % BER per user per diversity level

sumCapacityResults = zeros(length(snrRange), 1); % Sum capacity

totalPowerUsed = zeros(length(snrRange), 1); % Total power used

totalThroughput = zeros(length(snrRange), 1); % Throughput

fprintf('SNR (dB) | User 1 BER | User 2 BER | User 3 BER | User 4 BER | Sum Capacity | Total Power Used | Total Throughput\n');

fprintf('--------------------------------------------------------------------------------------------------------------\n');

for snrIdx = 1:length(snrRange)

snr = snrRange(snrIdx);

noiseVar = 10^(-snr / 10); % Noise variance

% Random channel gains for each user and subcarrier

channelGains = abs(randn(numUsers, numSubcarriers));

%% Water-Filling Power Allocation

waterLevel = 1 / mean(channelGains(:));

allocatedPower = zeros(numUsers, numSubcarriers);

for user = 1:numUsers

for sc = 1:numSubcarriers

if channelGains(user, sc) > 0

allocatedPower(user, sc) = max(0, waterLevel - 1 / channelGains(user, sc));

end

end

end

%% Sum Capacity and Total Power Calculation

subcarrierCapacity = log2(1 + (channelGains .\* allocatedPower) / noiseVar);

sumCapacityResults(snrIdx) = sum(subcarrierCapacity(:));

totalPowerUsed(snrIdx) = sum(allocatedPower(:));

%% BER Calculation with Diversity and Repetition Code

for diversityIdx = 1:length(diversityLevels)

diversityOrder = diversityLevels(diversityIdx);

for user = 1:numUsers

% Generate random binary data for the user

userData = randi([0 1], infoBitsLength, 1);

% BPSK Modulation

txSymbols = 1 - 2 \* userData; % Map 0 -> 1, 1 -> -1

% Add diversity effect by averaging independent noise samples

rxSymbols = zeros(length(txSymbols), diversityOrder);

for diversitySample = 1:diversityOrder

% Add AWGN noise for each diversity path

noise = sqrt(noiseVar / 2) \* (randn(size(txSymbols)) + 1j\*randn(size(txSymbols)));

rxSymbols(:, diversitySample) = txSymbols + noise;

end

% Average over diversity paths for combining

combinedSymbols = mean(rxSymbols, 2);

% Soft-decision decoding with majority voting for repetition code

receivedBits = real(combinedSymbols) < 0;

% Calculate BER with bitwise comparison

numBitErrors = sum(receivedBits ~= userData);

berResults(snrIdx, user, diversityIdx) = numBitErrors / infoBitsLength;

end

end

% Calculate Total Throughput (use last diversity level for throughput)

totalThroughput(snrIdx) = sum((1 - berResults(snrIdx, :, end)) \* infoBitsLength);

% Display results

fprintf('%8.2f | %11.5f | %11.5f | %11.5f | %11.5f | %12.4f | %16.4f | %17.4f\n', ...

snr, berResults(snrIdx, 1, end), berResults(snrIdx, 2, end), berResults(snrIdx, 3, end), ...

berResults(snrIdx, 4, end), sumCapacityResults(snrIdx), totalPowerUsed(snrIdx), ...

totalThroughput(snrIdx));

end

%% Plotting Results - Separate BER Plots for Each User

for user = 1:numUsers

figure;

plot(snrRange, berResults(:, user, end), 'o-', 'LineWidth', 1.5);

set(gca, 'YScale', 'log');

ylim([1e-6, 1]); % Adjust y-axis limits for BER

xlabel('SNR (dB)');

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

title(sprintf('BER vs SNR for User %d', user));

grid on;

end

%% BER for all users in one plot

figure;

hold on;

colors = ['r', 'g', 'b', 'k']; % Different colors for each user

markers = ['o', 's', 'd', '^']; % Different markers for each user

for user = 1:numUsers

plot(snrRange, berResults(:, user, end), ...

[colors(user) '-' markers(user)], 'LineWidth', 1.5, ...

'DisplayName', sprintf('User %d', user)); % Add legend label

end

set(gca, 'YScale', 'log'); % Logarithmic scale for BER

ylim([1e-6, 1]); % Adjust y-axis limits for BER

xlabel('SNR (dB)');

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

title('BER vs SNR for All Users');

legend('show'); % Show legend

grid on;

hold off;

%% Sum Capacity, Total Power Used, and Throughput Plots

figure;

plot(snrRange, sumCapacityResults, 'r-o', 'LineWidth', 2);

xlabel('SNR (dB)');

ylabel('Sum Capacity (bps/Hz)');

title('Sum Capacity vs SNR');

grid on;

figure;

plot(snrRange, totalPowerUsed, 'b-o', 'LineWidth', 2);

xlabel('SNR (dB)');

ylabel('Total Power Used');

title('Total Power Used across SNR Range');

grid on;

figure;

plot(snrRange, totalThroughput, 'k-o', 'LineWidth', 2);

xlabel('SNR (dB)');

ylabel('Total Throughput (bits)');

title('Total Throughput vs SNR');

grid on;

figure;

bar(1:numSubcarriers, mean(allocatedPower, 1), 'FaceColor', [0.2, 0.6, 0.8]);

xlabel('Subcarrier Index');

ylabel('Average Power Allocation');

title('Power Allocation per Subcarrier');

grid on;