1. SNRvsBER248_easy

```
clc;
clear all;
close all;

% Parameters
num_symbols = 1e5; % Number of symbols
SNR_dB = 0:2:20; % Range of SNR values in dB

% PSK orders to be tested
M_values = [2, 4, 8];

% Initialize BER array
BER = zeros(length(M_values), length(SNR_dB));

% BER calculation for each PSK order and SNR value
for i = 1:length(M_values)
    M = M_values(i);

    for j = 1:length(SNR_dB)
        % Generate random symbols
        data_symbols = randi([0, M-1], 1, num_symbols);

        % Modulate symbols using PSK
        modulated_signal = pskmod(data_symbols, M, pi/M);

        % Add AWGN to the signal
```

```
        received_signal = awgn(modulated_signal, SNR_dB(j), 'measured');

        % Demodulate received signal
        demodulated_symbols = pskdemod(received_signal, M, pi/M);

        % Calculate BER
        BER(i, j) = sum(data_symbols ~= demodulated_symbols) / num_symbols;
    end
end

% Plot BER vs SNR
figure;
semilogy(SNR_dB, BER(1,:), 'bo-', 'LineWidth', 2); % BPSK (M=2)
hold on;
semilogy(SNR_dB, BER(2,:), 'ro-', 'LineWidth', 2); % QPSK (M=4)
semilogy(SNR_dB, BER(3,:), 'go-', 'LineWidth', 2); % 8-PSK (M=8)
grid on;

xlabel('SNR (dB)');
ylabel('Bit Error Rate (BER)');
title('BER vs SNR for Different PSK Modulation Orders');
legend('BPSK (M=2)', 'QPSK (M=4)', '8-PSK (M=8)');
hold off;
```

2. BPSK_mod_demod

```
clc;
```

```
clear all;
close all;

number_bits = 30;
BINSEQ = abs(round(rand(1,number_bits)*2-1));

t = 0:1/100:1;
Eb = 1;
Tb = 1;
nc = 4; %number of carrier cycle in a bit
fc = nc/Tb;

% Carrier Signal
Ac = 1.0;
xc = Ac * cos(2*pi*fc*t);

figure(1)
plot(t, Ac*xc);
title('Carrier Signal')
xlabel(' time s ');
ylabel( ' amplitude ' );

% PSK Modulation
TX=[];
for m=1:number_bits
   if(BINSEQ(m)==1)
      TX = [TX sqrt(2*Eb/Tb)*cos(2*pi*fc*t)];
```

```
    else
        TX = [TX -1*sqrt(2*Eb/Tb)*cos(2*pi*fc*t)];
    end
end



snr_db = -80; % Example SNR
signal_power = mean(TX.^2);
noise_power = signal_power / (10^(snr_db/10))
noise = sqrt(noise_power/2) * randn(size(TX));
RX_AWGN = TX + noise;

figure(2)
subplot(2,1,1)
plot(1:length(TX),TX)
title('PSK signal')
subplot(2,1,2)
plot(1:length(RX_AWGN),RX_AWGN)
title('PSK with noise ')

%Coherent Detection
LO = sqrt(2/Tb) * cos(2*pi*fc*t); % Local oscillator
BINSEQDET = []; % Detected binary sequence
CS = []; % Correlation results

for n = 1:number_bits
    start_idx = (n-1)*length(t) + 1;
```

```
    end_idx = n*length(t);
    temp = RX_AWGN(start_idx:end_idx);

    % Correlate with LO
    S = sum(temp .* LO);
    CS = [CS S]; % Store correlation result

    % Decision: If S > 0, bit = 1; else, bit = 0
    if S > 0
        BINSEQDET = [BINSEQDET 1];
    else
        BINSEQDET = [BINSEQDET 0];
    end
end
figure(3)
subplot(2,2,1)
stem(CS)
title('Output of the correlation receiver')

subplot(2,2,2)
scatter(CS,zeros(1,number_bits))
title('Signal-space diagram for the PSK signal');

subplot(2,2,3)
stem(BINSEQ)
title('Transmitted binary sequence')
```

```
subplot(2,2,4)
stem(BINSEQDET)
title('Detected binary sequence')

Bit_error = sum(abs(BINSEQDET - BINSEQ))
fprintf('Number of error bits: %d\n',Bit_error);
```