QPSK

```
clc;
clear;
close all;
t=0:0.01:1; f=5;
m=randi([0,1],1,10); carrier_sig1=[]; carrier_sig2=[];
qpsk_sig=[];
msg_sig=[];
for i=1:length(m)
if m(i)==0
m(i)=-1;
msg_sig=[msg_sig zeros(1,length(t/2))];
 else
msg_sig=[msg_sig ones(1,length(t/2))];
end
end
for i=1:2:length(m)
q1=m(i)*cos(2*pi*f*t);  q2=m(i+1)*sin(2*pi*f*t); Q=q1+q2;
qpsk_sig=[qpsk_sig Q];
carrier_sig1=[carrier_sig1 cos(2*pi*f*t)]; carrier_sig2=[carrier_sig2 sin(2*pi*f*t)];
end
figure
subplot(6,1,1);
 plot(msg_sig);
xlabel('time');ylabel('amp'); ylim([-0.5 1.5]);
title('message signal'); subplot(6,1,2); plot(qpsk_sig); xlabel('time');
ylabel('amp'); title('qpsk signal');

noise_sig=awgn(qpsk_sig,12)
subplot(6,1,3);
plot(noise_sig);
xlabel('time');ylabel('amp');
title('qpsk signal with noise'); dem_sig1=(noise_sig).*(carrier_sig1); subplot(6,1,4);plot(dem_sig1);
xlabel('time');ylabel('amp');
title('demodulated signal 1');
dem_sig2=(noise_sig).*(carrier_sig2); subplot(6,1,5);plot(dem_sig2);
xlabel('time');ylabel('amp');
title('demodulated signal 2');
 demod_sig=[];
for i=1:length(m)/2
     k1=sum(dem_sig1((1+(i-1)*length(t)):(i*length(t))));
     k2=sum(dem_sig2((1+(i1)*length(t)):(i*length(t)));
     if k1>0
demod_sig=[demod_sig ones(1,length(t))];
     else
demod_sig=[demod_sig zeros(1,length(t))];
     end
                 if k2>0
                   demod_sig=[demod_sigones(1,length(t))];
     else
               demod_sig=[demod_sig zeros(1,length(t))];

     end
end
subplot(6,1,6)plot(demod_sig);
```

```
ylim([-0/.5      1.5]);
xlabel('time');
ylabel('amp');
title('original msg signal');

%QPSK_BER
clcclearall; close all;
 l=10000;
snrdb=1:1:10;    snrlin=10.^(snrdb/10); for snrdb=1:1:10
         si=2*(round(rand(1,l))-0.5);
         sq=2*(round(rand(1,l))-0.5); s=si+1i*sq; w=awgn(s,snrdb,'measured');
         r=w;
                     si_=sign(real(r));
      sq_= sign(imag(r));    ber1=(lsum(si==si_))/l; ber2=(lsum(sq==sq_))/1;
ber(snrdb)=mean([ber1 ber2]);
             end
snrdb=1:1:10; snrlin=10.^(snrdb./10); tber=0.5.*erfc(sqrt(snrlin));
semilogy(snrdb,ber,'-k*',snrdb,tber,'k') title('BER performance of QPSK'); xlabel('SNR in dB');
ylabel('BER'); grid on;

%DPSK
clc;
clear all;
close all;
t=0:0.01:1;
f= 3;
m=10;
msg=randi([0,1],1,m);
msg_sig=[];
dpsk_sig=[];
carrier_sig=[];
%differential encoding
d=1;
dc=[];
for i=1:m
          dc=[dc d];
          d=not(xor(d,msg(i)));
 end
dc=[dc d];
for i=1:m
if msg(i)==0
msg_sig=[msg_sig zeros(1,length(t))];
 else
msg_sig=[msg_sig ones(1,length(t))];
 end
end

%bpsk modulation
for i=1:m+1
if dc(i)==0
dpsk_sig=[dpsk_sig cos((2*pi*f*t)+pi)];
 else
dpsk_sig=[dpsk_sig cos(2*pi*f*t)];
 end
carrier_sig=[carrier_sig cos(2*pi*f*t)];
```

```
end
subplot(5,1,1)
plot(msg_sig) xlabel("time(s)") ylabel("Amplitude") ylim([-1.5 1.5])
title("Message signal")
 subplot(5,1,2) plot(dpsk_sig) xlabel("time(s)") ylabel("Amplitude") ylim([-1.5 1.5])
title("DPSK signal")
% addition of noise
snr=12;
noise_sig=awgn(dpsk_sig,snr);
 subplot(5,1,3) plot(noise_sig) xlabel("time(s)") ylabel("Amplitude")
title("Recieved DPSK signal with noise")

dem_sig=(noise_sig).*(carrier_sig);
subplot(5,1,4)
plot(dem_sig) xlabel("time(s)") ylabel("Amplitude")
title("Demodulated signal with noise")
demod_sig=[];
for i=1:m+1
k=sum(dem_sig((1+(i-1)*length(t)):(i*length(t))));
if k>0
demod_sig=[demod_sig ones(1,length(t))];
 else
demod_sig=[demod_sig zeros(1,length(t))];
 end
end
demodulated_sig=[];
for i=1:m
k1=sum(demod_sig((1+(i-1)*length(t)):((i)*length(t))));
k2=sum(demod_sig((1+(i)*length(t)):((i+1)*length(t))));
if k1==k2
demodulated_sig=[demodulated_sig ones(1,length(t))];
 else
demodulated_sig=[demodulated_sig zeros(1,length(t))];
 end
end
subplot(515)
plot(demodulated_sig)
 ylim([-1.5 1.5])
xlabel("time(s)") ylabel("Amplitude")
title("Original msg signal")
%DPSK_BER
clc;
close all;
 clear all;
 N = 1e6;
b = randi([0,1],N, 1);
dpskmod = comm.DPSKModulator(2,pi/2,'BitInput',true);
dpskdemod = comm.DPSKDemodulator(2,pi/2,'BitOutput',true);
 yx = dpskmod(b);
SNRdB = 1:2:15;
for i=1:length(SNRdB) SNRi = SNRdB(i);
rn = awgn(yx,SNRi,'measured');
 bcat = dpskdemod(rn);
SNR = 10^(SNRi/10);
Ed_by_No = SNR;
```

```
Ed_by_No_dB(i) = 10*log10(Ed_by_No);
 BER_th(i) = (1/2)*exp(-Ed_by_No);
BER(i) = length(find(b~=bcat))/N;
End
figure;
semilogy(Ed_by_No_dB, BER, 'b-', Ed_by_No_dB, BER_th, 'r*');
 axis([0 12 10^-7 1]); grid on;
title('BER PERFOMANCE BY DPSK');
xlabel('SNR in dB'); ylabel('BER');
```

BPSK

```
clc;
clear;
close all;
t=0:0.01:1;
 f=3;
m=randi([0,1],1,10);
 msg_sig=[];
bpsk_sig=[];
carrier_sig=[];
for i=1:length(m)
if m(i)==0
msg_sig=[msg_sig zeros(1,length(t))];
bpsk_sig=[bpsk_sig sin((2*pi*f*t)+pi)];
else
msg_sig=[msg_sig ones(1,length(t))]; bpsk_sig=[bpsk_sig sin(2*pi*f*t)];
            end


carrier_sig=[carrier_sig sin(2*pi*f*t)];
        end
 subplot(6,1,1);
 plot(carrier_sig);
 xlabel('time');ylabel('amp');
 title('carrier signal'); subplot(6,1,2); plot(msg_sig);
 xlabel('time');ylabel('amp');
 title('message signal'); ylim([-0.5 1.5]);
 subplot(6,1,3);
  plot(bpsk_sig);
  xlabel('time');ylabel('amp'); title('bpsk signal');
rec_sig=awgn(bpsk_sig,12);
 subplot(6,1,4); plot(rec_sig);
xlabel('time');ylabel('amp'); title('received signal');
dem_sig=(rec_sig).*(carrier_sig);     subplot(6,1,5);plot(dem_sig);
xlabel('time');ylabel('amp');
title('demodulated signal');
demod_sig=[];
       for i =1:length(m)
         k=sum(dem_sig((1+(i-1)*length(t)):(i*length(t))));
                        if k>0
                        demod_sig=[demod_sig ones(1,length(t))];
                        else
                         demod_sig=[demod_sig  zeros(1,length(t))];
                        end
                        end
                        subplot(6,1,6);
                         plot(demod_sig); ylim([-0.5  1.5]);
                        xlabel('time');ylabel('amp');
                        title('original  message  signal');
                        %BPSK_BER
                        clc;
                        clear all;
                        close all;
                        M = 1e6;
                        b = randi([0,1], M, 1);
                        bpskmod  =  comm.BPSKModulator;
                        bpskdemod  =  comm.BPSKDemodulator;b_mod = bpskmod(b);
                        for SNR = 0:1:12
                        b_rec  =  awgn(b_mod,SNR,"measured");
                        b_demod = bpskdemod(b_rec);
                        BER(SNR+1)  =  length(find(b~=b_demod))/M;
                        SNR_=10.^(SNR/10);
                        BER_th(SNR+1)  =  (1/2)*erfc(sqrt(SNR_));
                        end
                        semilogy(0:1:12,BER,"-k*",0:1:12,BER_th,"-k") grid on;
                        title('BER  PERFOMANCE  BY  BPSK');
                        xlabel('SNR  in  dB');
```

QAM- BER PERFORMANCE

```
% QAM Parameters
M = 16; % QAM modulation order (e.g., 16-QAM)
EbNo_dB = 0:1:20; % Eb/No range in dB
numBits = 1e5; % Number of bits to transmit
% Generate random binary data
data = randi([0, 1], 1, numBits);
% Modulation
modulatedData = qammod(data, M);
% Simulation loop over various Eb/No values
ber = zeros(1, length(EbNo_dB));
for i = 1:length(EbNo_dB)
 % Add AWGN noise
 EbNo = 10^(EbNo_dB(i)/10); % Convert Eb/No from dB to linear scale
 noiseVar = 1 / (2 * EbNo); % Noise variance for each dimension
 noise = sqrt(noiseVar) * (randn(1, length(modulatedData)) + 1i * randn(1,
length(modulatedData)));
```

```
 receivedData = modulatedData + noise;

 % Demodulation
 demodulatedData = qamdemod(receivedData, M);

% Compute and store BER
 ber(i) = sum(demodulatedData ~= data) / numBits;
end
% Plot BER vs. Eb/No
figure;
semilogy(EbNo_dB, ber, '-o');
grid on;
title('BER vs. Eb/No for QAM');
xlabel('Eb/No (dB)');
ylabel('Bit Error Rate (BER)');
```

BER CODES:

```
% BER Performance of Digital Modulation Schemes (BPSK, QPSK, DPSK, 16-QAM)

% Define modulation schemes and SNR range
modulations = {'BPSK', 'QPSK', 'DPSK', '16-QAM'};
SNR_dB = 0:1:20; % SNR range in dB

% Initialize BER arrays
BER_BPSK = zeros(1, length(SNR_dB));
BER_QPSK = zeros(1, length(SNR_dB));
BER_DPSK = zeros(1, length(SNR_dB));
BER_16QAM = zeros(1, length(SNR_dB));

% Number of bits per symbol for each modulation
bits_per_symbol = [1, 2, 1, 4];

% Loop through each modulation scheme and compute BER
for i = 1:length(modulations)
    modulation = modulations{i};
    bits = bits_per_symbol(i);

    for j = 1:length(SNR_dB)
        SNR = 10^(SNR_dB(j) / 10); % Convert SNR from dB to linear scale

        % Generate random binary data
        data = randi([0, 1], 1e6 * bits, 1);

        % Modulate the data
        if strcmp(modulation, 'BPSK')
            modulated_data = pskmod(data, 2);
        elseif strcmp(modulation, 'QPSK')
            modulated_data = pskmod(data, 4);
        elseif strcmp(modulation, 'DPSK')
            modulated_data = dpskmod(data);
        elseif strcmp(modulation, '16-QAM')
            modulated_data = qammod(data, 16);
        else
            error('Unsupported modulation scheme');
        end

        % Add AWGN to the modulated signal
        received_signal = awgn(modulated_data, SNR, 'measured');
        % Demodulate the received signal
        if strcmp(modulation, 'BPSK')
            demodulated_data = pskdemod(received_signal, 2);
        elseif strcmp(modulation, 'QPSK')
            demodulated_data = pskdemod(received_signal, 4);
        elseif strcmp(modulation, 'DPSK')
            demodulated_data = dpskdemod(received_signal);
        elseif strcmp(modulation, '16-QAM')
            demodulated_data = qamdemod(received_signal, 16);
        end

        % Compute the bit error rate
        num_errors = sum(data ~= demodulated_data);
        BER = num_errors / (length(data));

        % Store BER value in the corresponding array
        if strcmp(modulation, 'BPSK')
            BER_BPSK(j) = BER;
        elseif strcmp(modulation, 'QPSK')
            BER_QPSK(j) = BER;
        elseif strcmp(modulation, 'DPSK')
            BER_DPSK(j) = BER;
        elseif strcmp(modulation, '16-QAM')
            BER_16QAM(j) = BER;
        end
    end
end

% Plot BER vs. SNR for each modulation scheme
figure;
semilogy(SNR_dB, BER_BPSK, '-o',  'DisplayName', 'BPSK');
hold on;
semilogy(SNR_dB, BER_QPSK, '-^','DisplayName', 'QPSK');
semilogy(SNR_dB, BER_DPSK, '--','linewidth',2,'DisplayName', 'DPSK');
semilogy(SNR_dB, BER_16QAM,'-*','DisplayName', '16-QAM');
grid on;
xlabel('SNR (dB)');
ylabel('Bit Error Rate (BER)');
title('BER Performance of Digital Modulation Schemes');
legend('Location', 'best');
```