EESC 6352: Digital Communication Project

Ishita Pal (IXP180006)

1. Question 1: To Calculate BER and SER of 16-QAM modulation
    - Matlab code
    - Results
    - Plots of Stimulation Results

2. Question 2: To Calculate BER of pi/4 shifted DQPSK modulation
    - Matlab code
    - Results
    - Plots of Stimulation Results

3. Question 3: To Calculate BER of OFDM modulation
    - Matlab code
    - Results
    - Plots of Stimulation Results

Question1:

Consider a digital communication system with 16-QAM in an AWGN channel. By means of Monte Carlo simulations using MATLAB, evaluate the symbol error probability and bit error probability of the coherent maximum likelihood detection for E_b/N_0 values of 6 dB, 8 dB, 10 dB, and 12 dB.

16 QAM:

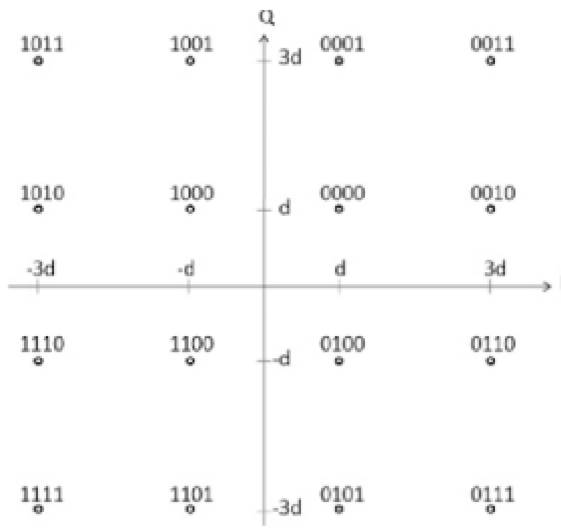

Fig1: Constellation Diagram of QAM

Bit error probability of QAM over AWGN channel is given by

$$P_{b,16QAM} = \frac{3}{2k} erfc\left(\sqrt{\frac{kE_b}{10N_o}}\right)$$

Where M = size of modulation n = number of bits per symbols

N0 = noise power spectral density Eb = energy per bit

The relationship between the Bit error probability and Symbol error probability are:

$$P_b \approx \frac{1}{\log_2 M} P_s$$

Matlab Code:

```
clc;
clear all;
close all;
M = 16;  %modulation index
k = log2(M);
symbols = [-3,-1,1,3];
EbNo = [6:2:12];
EsNo = k*EbNo;
```

```
N = 100000; %number of symbol
S = zeros(1,N);

for i = 1:length(EbNo)

    s = randsrc(1,N,symbols) + j*randsrc(1,N,symbols);
    n = (1/sqrt(10))*s;  %normalizing
    a = (1/sqrt(2))*[randn(1,N) + j*randn(1,N)];
    y = n + 10^(-EbNo(i)/20)*a;  %adding white guassian noise

    %demodulation
    y_x = real(y);
    y_y = imag(y);

    S_re(find(y_x< -2/sqrt(10))) = -3;
    S_re(find(y_x>= -2/sqrt(10) & y_x<0)) = -1;
    S_re(find(y_x >=0 & y_x<2/sqrt(10))) = 1;
    S_re(find(y_x >= 2/sqrt(10))) = 3;

    S_im(find(y_y>= 2/sqrt(10))) = 3;
    S_im(find(y_y<2/sqrt(10) & y_y>=0)) = 1;
    S_im(find(y_y<0 & y_y>=-2/sqrt(10))) = -1;
    S_im(find(y_y<-2/sqrt(10))) = -3;

    S = S_re + j*S_im;
    TotError(i) = size(find(s-S),2); %calculating total error
end

BER = TotError/N; % calculating bit error rate
SER = k*BER; %calculation sysmbol error rate

%displaying the BER and SER
disp('Bit error rate');
disp(BER);
disp('symbol error rate');
disp(SER);

%calualting using theortitical formula
BERtheory = 3/2*erfc(sqrt(0.1*(10.^(EbNo/10))));
SERtheory = k * BERtheory;

%plotting BER graph
figure;
semilogy(EbNo,BERtheory,'-','Linewidth',2);
hold on;
semilogy(EbNo,BER,':*','Linewidth',2);
grid on;
legend('Theoritical', 'Simulation');
xlabel('Eb/No  (dB)')
ylabel('Bit Error Rate')
title('Bit error probability for 16-QAM modulation');

%plotting SER graph
figure;
semilogy(EsNo,SERtheory,'-','Linewidth',2);
hold on;
semilogy(EsNo,SER,':*','Linewidth',2);
grid on;
legend('Theoritical', 'Simulation');
xlabel('Es/No  (dB)')
ylabel('Symbol Error Rate')
```

```
title('Symbol error probability for 16-QAM modulation');
```

Results:

Bit error rate

0.4774 0.3518 0.2239 0.1113

symbol error rate

1.9097 1.4071 0.8956 0.4452

Plots of Stimulation Results:

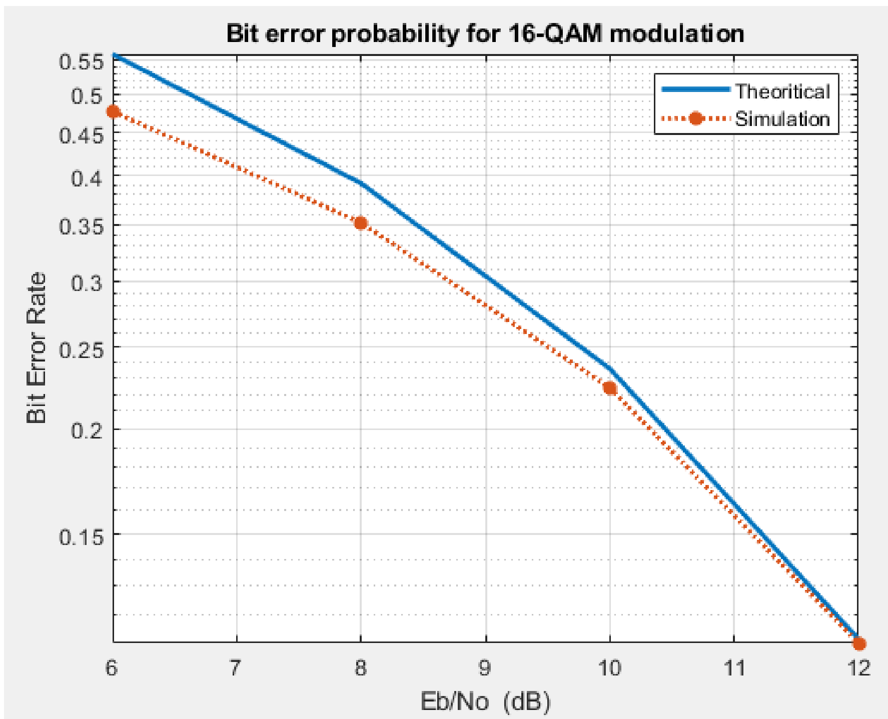

Fig2: Stimulation plot of BER for 16-QAM

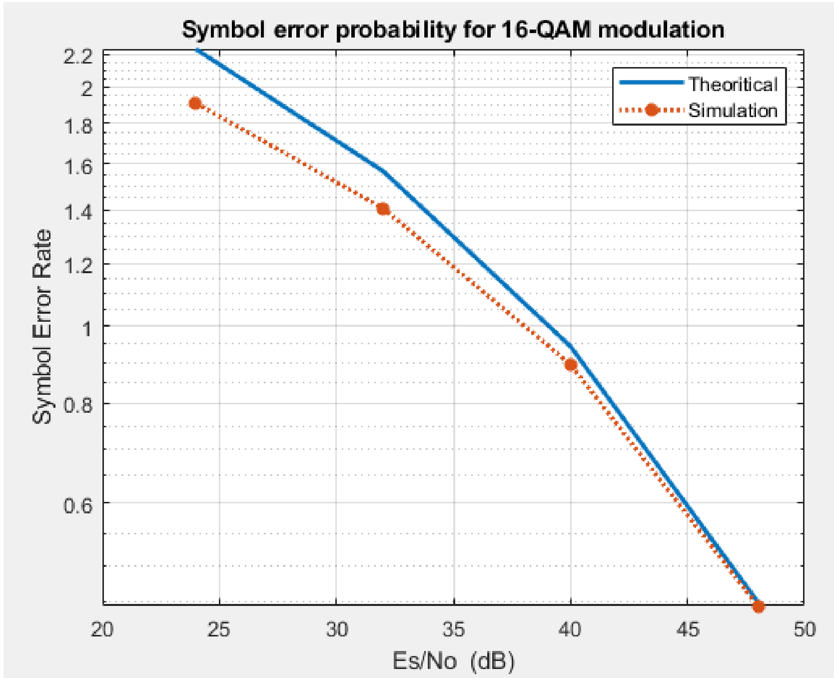

Fig3: Stimulation plot for SER for 16-QAM

Question2:

Consider a digital communication system with π/4-shifted DQPSK in an AWGN channel. By means of Monte Carlo simulations using MATLAB, evaluate the bit error probability of differentially coherent detection for E_b/N_0 values of 4 dB, 6 dB, 8 dB, and 10 dB.

DQPSK:

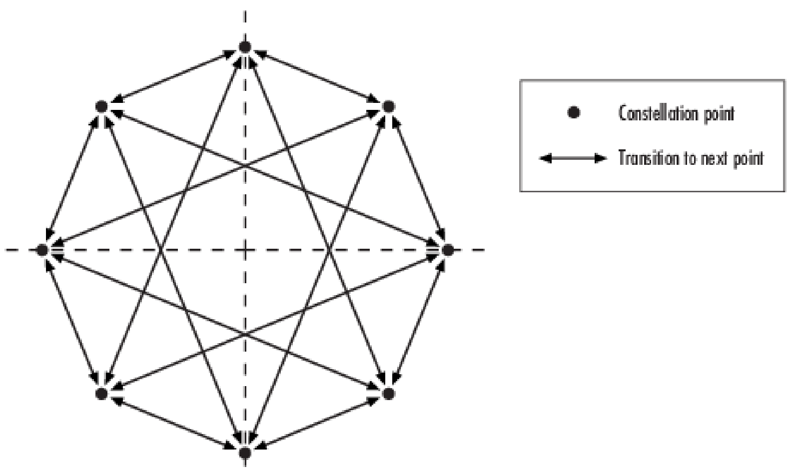

Fig4: Signal Constellation Diagram of pi/4 DQPSK

Bit error probability for DQPSK, when gray coding is employed, is given as :

$$P_b = Q_1(a, b) - \frac{1}{2} I_0(ab) e^{-\frac{a^2+b^2}{2}}$$

where Q1(a, b) is the Marcum Q function I0(x) is the modified Bessel function of order zero, parameters a, b is defined by the following formula:

$$a = \sqrt{\frac{2\mathcal{E}_b}{N_0}\left(1 - \sqrt{\frac{1}{2}}\right)}$$

$$b = \sqrt{\frac{2\mathcal{E}_b}{N_0}\left(1 + \sqrt{\frac{1}{2}}\right)}$$

Matlab Code:

```
clc;
close all;
clear all;
N = 100000; % number of bits or symbols
M = 4;
k = log2(M); % number of bits per symbol
EbNo = [4:2:10];
EsNo = EbNo + 10*log10(k);
for i = 1:length(EbNo)

     s = rand(1,N)>0.5; % generating random binary signals

     grp = reshape(s,2,N/2).';
     bintodec = ones(N/2,1)*2.^[k-1:-1:0];
     dec_s = sum(grp.*bintodec,2);

     % converting to gray coded symbols
     gray_s = bitxor(dec_s,floor(dec_s/2));
     phasegray = 2*gray_s.'+1;

     % generating differential modulated symbols
     diffPhase = filter([ 1 ],[1 -1],phasegray); % start with 0 phase
     dqpsk_s = exp(j*diffPhase*pi/4);

     % white gaussian noise, 0 mean
     a = 1/sqrt(2)*[randn(1,N/2) + j*randn(1,N/2)];
     dqpsk = dqpsk_s + 10^(-(EsNo(i))/20)*a; % additive white gaussian noise

     %demodulation
     estphase = angle(dqpsk);
     estdphase = filter([1 -1],1,estphase)*4/pi;
     quant_dphase = 2*floor(estdphase/2)+1; % quantizing

     % gray to binary
     quant_dphase(find(quant_dphase<0))=quant_dphase(find(quant_dphase<0))+8;
     bin_phase = floor(bitxor(quant_dphase,floor(quant_dphase/2))/2);
     estBit = (dec2bin(bin_phase.')).';
     estBit = str2num(estBit(1:end).').';

     TotError(i) = size(find([s - estBit]),2); %calculating total error

     % theoretical BER computation
     a = sqrt(2*10.^(EbNo(i)/10)*(1-sqrt(1/2)));
     b = sqrt(2*10.^(EbNo(i)/10)*(1+sqrt(1/2)));
     k_b = 0:10;
     temp = exp(-((a.^2+b.^2)/2)).*sum((a/b).^k_b.*besseli(k_b,a*b));
     dqpskTheoritical(i) = temp - 0.5*besseli(0,a*b)*exp(-((a.^2+b.^2)/2));
```

```
end

%calculating BER
Ber = TotError/N;

disp('Bit error rate');
disp(Ber);

%plotting the BER
figure;
semilogy(EbNo,dqpskTheoritical,'-','Linewidth',2);
hold on
semilogy(EbNo,Ber,':*','Linewidth',2);
grid on
legend('theoritical', 'simulation');
xlabel('Eb/No (dB)')
ylabel('Bit Error Rate')
title('Bit error probability curve for pi/4 DQPSK');
```

Results:

Bit error rate

0.0490 0.0175 0.0038 0.0003

Plots for stimulation:

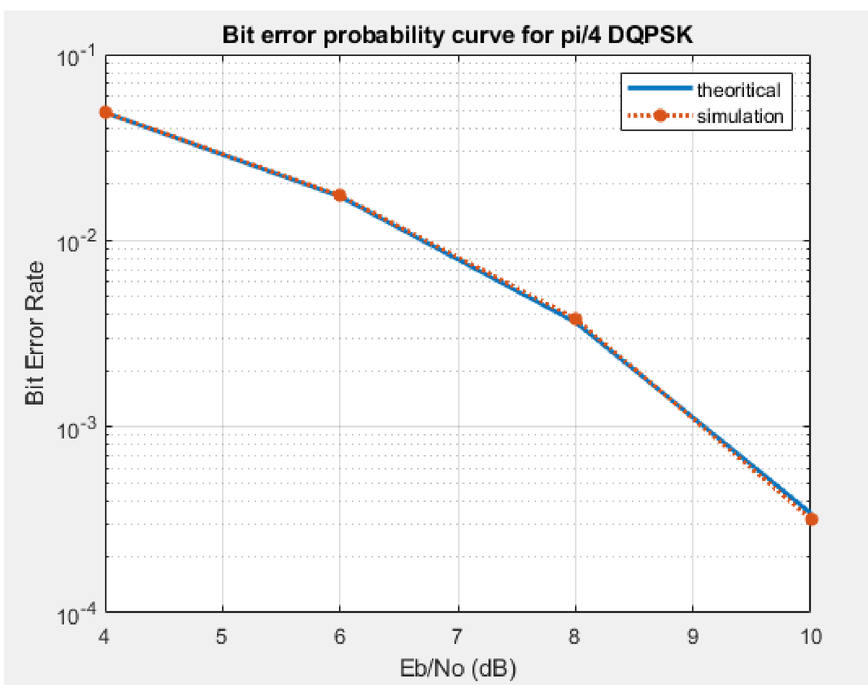

Fig5: Stimulation plot of pi/4 DQPSK

Question3:

Consider an OFDM system in AWGN channel with the DFT size of 64 and 52 used subcarriers on sub-carrier indexes -26, -25, … , -1, 1, 2, … , 26. The cyclic prefix length is ¼ of the useful OFDM symbol duration. Assume QPSK modulation on each used subcarrier. By means of Monte Carlo simulations using MATLAB, evaluate the bit error probability of coherent maximum likelihood detection for E_b/N_0 values of 2 dB, 4 dB, 6 dB, and 8 dB. Include the cyclic prefix in calculating Eb/N0.

Steps to follow:

1. Generate random bits
2. Subjected the random bits to QPSK modulation
3. Assign the symbols to the subcarriers
4. Take FFT and then add the cyclic Prefix
5. Signal is passed though a additive white Gaussian noise channel
6. Signal is then demultiplexed and demodulated and cyclic prefix is removed
7. Total number of error bits are calculated
8. BER is calculated as the total error bits by total number of bits.

Matlab Code:

```
clc;
clear all;
close all;
M = 4;  % Modulation
k = log2(M);  % Bits/symbol
nFFT = 64;
nDSC = 52;  %number of subcarrier
nbitspersym = 104; %number of bits per symbol
N = 100; %number of symbols
l= nbitspersym*N;

EbNo = [2:2:8]; %bit to noise
EsNo = EbNo + 10*log10(nDSC/nFFT) + 10*log10(64/80); %symbol to noise

for i = 1:length(EbNo)

    %generating symbols
    si=2*(round(rand(1,l))-0.5); %In-phase symbol generation
    sq=2*(round(rand(1,l))-0.5); %Quadrature symbol generation
    s=si+j*sq;
    n=(1/sqrt(2*EbNo(i)))*(randn(1,l)+j*randn(1,l)); %Random noise generation
    r=s+n;
    grp = reshape(r,nbitspersym,N).'; %grouping the symbols

    %assigning modulated symbols to subcarrier
    M = [zeros(N,6) grp(:,[1:nbitspersym/2]) zeros(N,1) grp(:,[nbitspersym/2+1:nbitspersym])
         zeros(N,5)];

    F = (nFFT/sqrt(nDSC))*ifft(fftshift(M.')).'; %taking FFT and normalizing the power

    F = [F(:,[49:64]) F]; %adding cyclic prefix

    F = reshape(F.',1,N*132);

    %adding white guassian noise
    A = 1/sqrt(2)*[randn(1,N*132) + j*randn(1,N*132)];
    y = sqrt(132/64)*F + 10^(-EsNo(i)/20)*A;
```

```
    y = reshape(y.',132,N).'; % formatting the received vector into symbols
    y = y(:,[17:80]); %removing cyclic prefix

    Y = (sqrt(nDSC)/nFFT)*fftshift(fft(y.')).';
    yMod = Y(:,[6+[1:nbitspersym/4] 7+[nbitspersym/4+1:nbitspersym/2] ]);
    si = sign(real(yMod)); %In-phase demodulation
    si = reshape(si,1,l/2).';
    sq = sign(imag(yMod)); %quadrature phase demodulation
    sq = reshape(sq,1,l/2).';
    BERi=(l-sum(si==si))/l; %In-phase BER calculation
    BERq=(l-sum(sq==sq))/l; %Quadrature BER calculation
    ber(i)=mean([BERi BERq])
end

%calculating BER
BER = ber/(N*nbitspersym);
%theortical calculation
BERTheory = (1/2)*erfc(sqrt(10.^(EbNo/10)));
%displaying the results
disp('BER:');
disp(BER);

%Plotting the graph
figure;
semilogy(EbNo,BERTheory,'-','Linewidth',2);
hold on
semilogy(EbNo,BER,':*','Linewidth',2);
grid on
legend('theortical', 'simulation');
xlabel('Eb/No  (dB)')
ylabel('Bit Error Rate')
title('Bit error probability curve for QPSK using OFDM')
```

Results:

BER:

0.0375   0.0125   0.0024   0.0002

Plots of Stimulation results:

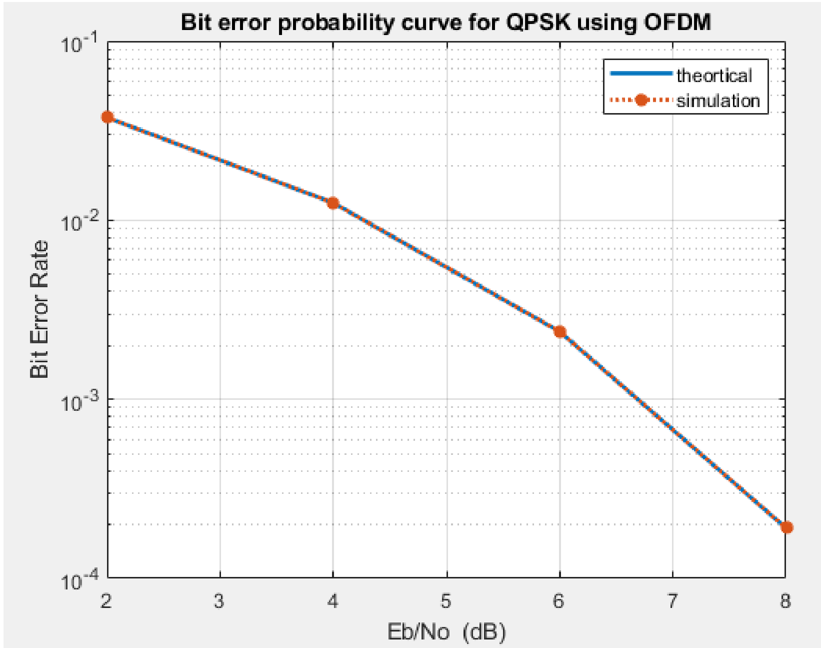

Fig6 : Stimulation plot of OPSK – OFDM

References

1. John G. Proakis and Masoud Salehi, Digital Communication 5th Edition
2. Mathswork.com