EE5141: INTRODUCTION TO WIRELESS AND CELLULAR COMMUNICATIONS

COMPUTER ASSIGNMENT –7

Submitted by

Akash Pramod – EE17B037

Adithya Swaroop – EE17B115

Date of Submission

18-04-21

Honour Code

We certify that this assignment submission is our own work and not from obtained from any other source

Student 1 Electronic Signature

Akash Pramod Y

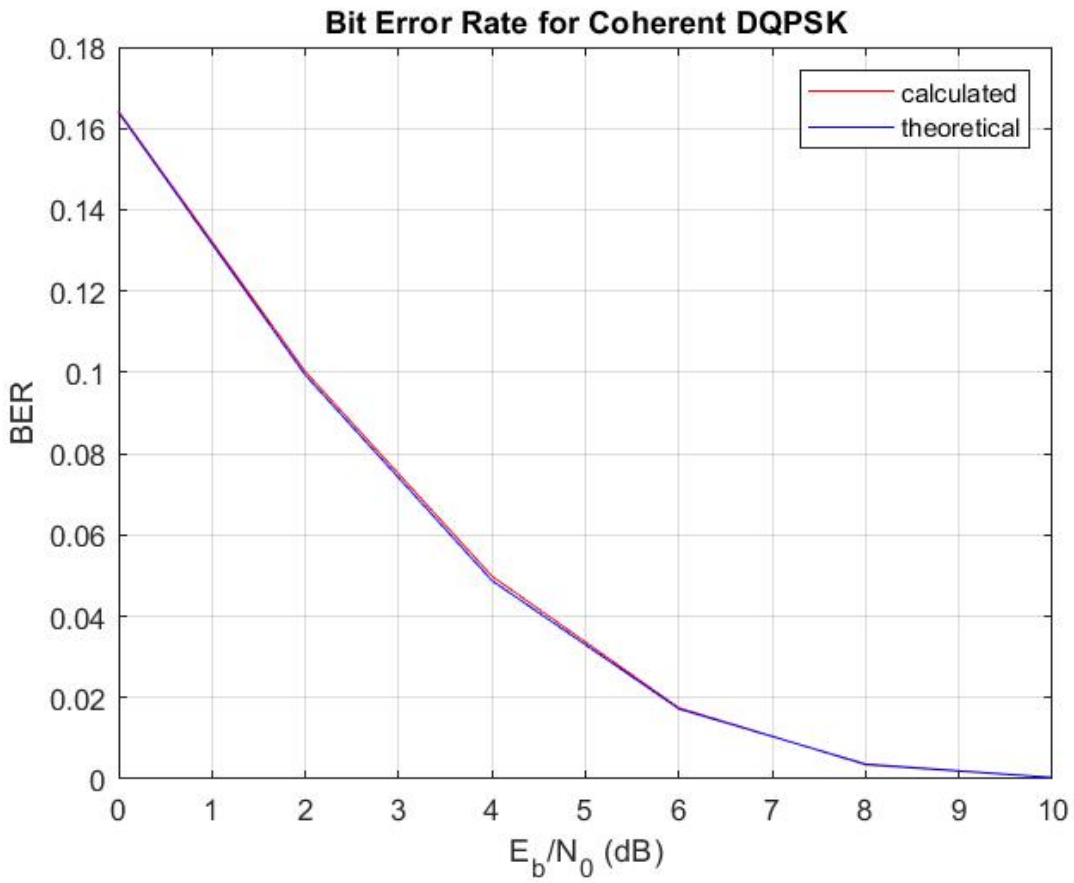
Student 2 Electronic Signature

Adithya Swaroop S

NOTE: FILENAME SHOULD BE **student1rollno_cassign1.pdf**

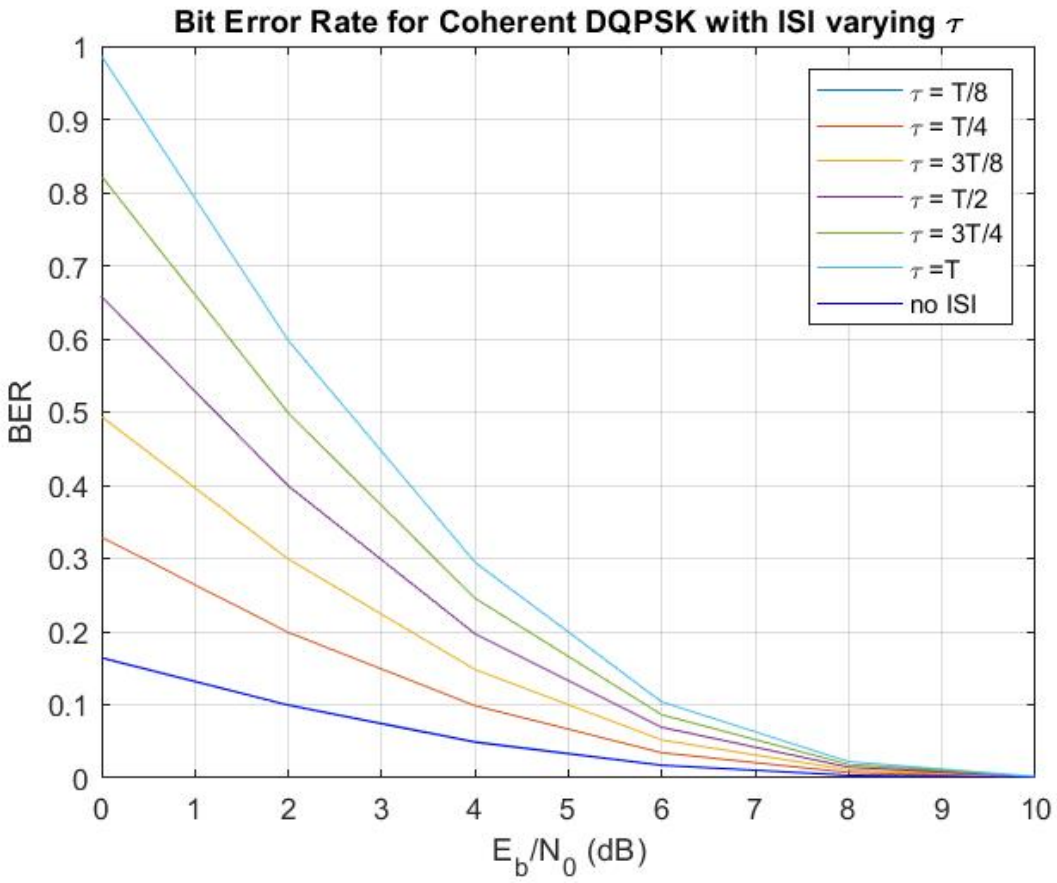
Question 1:

Plot for alpha = 0.1, tau = T/8 BER vs Es/No is like this.



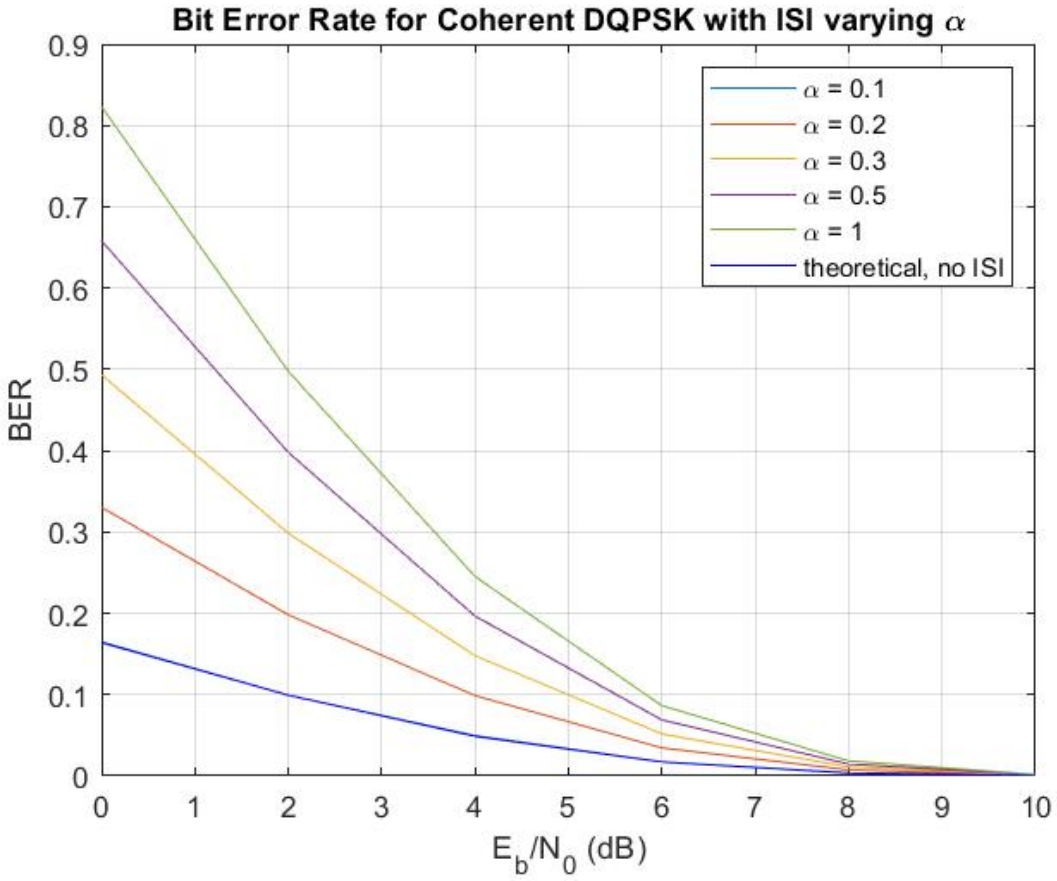
We can observe that there is no much difference for above values of alpha and tau for any value of SNR.

Question 2



We can observe that as the tau in ISI increases, BER increases. But as the SNR increases the effect of more delay is not that much. More delay implies more error because, there is more involvement of previous symbol as delay increases and so error increases.

Question 3



We can observe that as the alpha in ISI increases, BER increases. But as the SNR increases the effect of more alpha is not that much. More delay implies more error because, there is more involvement of ISI magnitude as alpha increases and so error increases.

CODE:

Part 1:

```
clear all;
alpha= 0.35;
A = 5;
T = 1/25000;
Ts = T/8;
t = [-A*T:Ts:A*T] + 10^(-8);
num = sin(pi*(1-alpha)*t/T) +
(4*alpha*t/T).*cos(pi*(1+alpha)*t/T);
den = (1-(4*alpha*t/T).^2).*(pi*t/T);
h = num./den;
DQPSK = [0 1 2 3];
N = 512;
esN0_dB = [0:2:10];
ps = zeros(1, length(esN0_dB));
pn = zeros(1, length(esN0_dB));

noErrs = zeros(1, length(esN0_dB));
noBitErrs = zeros(1, length(esN0_dB));

trails = 500;
beta = 0.1;
tau = 1;%samples delayed after conv

for ii = 1:length(esN0_dB)
    for j = 1:trails
        x = randsrc(1, N, DQPSK);
        bits = zeros(1, 2*N);
        for i = 1:N
            if x(i)==0
                bits(2*i-1)= 0;
                bits(2*i) = 0;
            elseif x(i)==1
                bits(2*i-1)= 0;
                bits(2*i) = 1;
            elseif x(i)==2
                bits(2*i-1)= 1;
                bits(2*i) = 0;
            else
                bits(2*i-1)= 1;
                bits(2*i) = 1;
            end
        end
        prev = exp(1i*pi/4);
        s = zeros(1,N);
```

```
        deltatheta = [pi/4, 3*pi/4, -pi/4, -3*pi/4];
        for i = 1:N
            s(i)= prev*exp(1i*deltatheta(x(i)+1));
            prev = s(i);
        end

        y = upsample(s, 8);
        w = conv(y, h);
        wdelay = beta*delayseq(w, tau);
        wf = w + wdelay;
        W = conv(wf, h);

        ps = norm(W(81:8:end-80))^2;
        w1 = awgn(wf, esN0_dB(ii));
        %noise = 10^(-esN0_dB(i)/20)*noise;
        noise = w1 - wf;
        nw = conv(noise, h);

        pn = norm(nw(81:8:end-80))^2;
        %estimate alpha from ps, pn and dB
        alpha1 = sqrt(ps/(pn*(10^((esN0_dB(ii)+3)/10))));
        final = W + alpha1 * nw;


        rs = final(81:8:end-80);
        rx = zeros(1,2*N);
        prev = exp(1i*pi/4);
        for i = 1:N
            if real(rs(i)*conj(prev))>0
                rx(2*i)= 0;
            else
                rx(2*i) = 1;
            end
            if imag(rs(i)*conj(prev))>0
                rx(2*i-1)= 0;
            else
                rx(2*i-1) = 1;
            end
            prev = rs(i);
        end
        noErrs(ii) = noErrs(ii) + size(find([bits-
rx]),2);
        noBitErrs(ii) = noBitErrs(ii) + size(find([bits-
rx]),2);
    end

end
```

```
ser = noErrs/(2*N*trails);
ber = noBitErrs/ (2*N*trails);
figure(1)
a = (2-sqrt(2))*10.^(esN0_dB/10);
b = (2+sqrt(2))*10.^(esN0_dB/10);

BER = 0.5*(1-
marcumq(sqrt(b),sqrt(a))+marcumq(sqrt(a),sqrt(b)));
plot(esN0_dB, ber,'r')
grid on
hold on
plot(esN0_dB, BER,'b')
ylabel('BER')
xlabel('E_b/N_0 (dB)')
title('Bit Error Rate for Coherent DQPSK')
legend('calculated', 'theoretical')
```

Part 2

```
clear all;
alpha= 0.35;
A = 5;
T = 1/25000;
Ts = T/8;
t = [-A*T:Ts:A*T] + 10^(-8);
num = sin(pi*(1-alpha)*t/T) +
(4*alpha*t/T).*cos(pi*(1+alpha)*t/T);
den = (1-(4*alpha*t/T).^2).*(pi*t/T);
h = num./den;

DQPSK = [0 1 2 3];
N = 512;
esN0_dB = [0:2:10];
ps = zeros(1, length(esN0_dB));
pn = zeros(1, length(esN0_dB));

noErrs = zeros(1, length(esN0_dB));
noBitErrs = zeros(1, length(esN0_dB));

trails = 500;
beta = 0.1;
%tau = 1;%samples delayed after conv
for tau = [1, 2, 3, 4, 6, 8]
    for ii = 1:length(esN0_dB)
        for j = 1:trails
            x = randsrc(1, N, DQPSK);
```

```
                bits = zeros(1, 2*N);
                for i = 1:N
                    if x(i)==0
                        bits(2*i-1)= 0;
                        bits(2*i) = 0;
                    elseif x(i)==1
                        bits(2*i-1)= 0;
                        bits(2*i) = 1;
                    elseif x(i)==2
                        bits(2*i-1)= 1;
                        bits(2*i) = 0;
                    else
                        bits(2*i-1)= 1;
                        bits(2*i) = 1;
                    end
                end
                prev = exp(1i*pi/4);
                s = zeros(1,N);
                deltatheta = [pi/4, 3*pi/4, -pi/4, -3*pi/4];
                for i = 1:N
                    s(i)= prev*exp(1i*deltatheta(x(i)+1));
                    prev = s(i);
                end

                y = upsample(s, 8);
                w = conv(y, h);
                wdelay = beta*delayseq(w, tau);
                wf = w + wdelay;
                W = conv(wf, h);

                ps = norm(W(81:8:end-80))^2;
                w1 = awgn(wf, esN0_dB(ii));
                %noise = 10^(-esN0_dB(i)/20)*noise;
                noise = w1 - wf;
                nw = conv(noise, h);

                pn = norm(nw(81:8:end-80))^2;
                %estimate alpha from ps, pn and dB
                alpha1 =
sqrt(ps/(pn*(10^((esN0_dB(ii)+3)/10))));


                final = W + alpha1 * nw;


                rs = final(81:8:end-80);
                rx = zeros(1,2*N);
```

```
            prev = exp(1i*pi/4);
            for i = 1:N
                if real(rs(i)*conj(prev))>0
                    rx(2*i)= 0;
                else
                    rx(2*i) = 1;
                end
                if imag(rs(i)*conj(prev))>0
                    rx(2*i-1)= 0;
                else
                    rx(2*i-1) = 1;
                end
                prev = rs(i);
            end
            noErrs(ii) = noErrs(ii) + size(find([bits-
rx]),2);
            noBitErrs(ii) = noBitErrs(ii) +
size(find([bits- rx]),2);


        end


    end
    ser = noErrs/(2*N*trails);
    ber = noBitErrs/ (2*N*trails);
    figure(1)
    plot(esN0_dB, ber)
    grid on
    hold on
end

a = (2-sqrt(2))*10.^(esN0_dB/10);
b = (2+sqrt(2))*10.^(esN0_dB/10);

BER = 0.5*(1-
marcumq(sqrt(b),sqrt(a))+marcumq(sqrt(a),sqrt(b)));


plot(esN0_dB, BER,'b')
ylabel('BER')
xlabel('E_b/N_0 (dB)')
title('Bit Error Rate for Coherent DQPSK with ISI varying
\tau')
legend('\tau = T/8','\tau = T/4','\tau = 3T/8','\tau =
T/2','\tau = 3T/4','\tau =T', 'no ISI')
```

Part 3

```
clear all;
alpha= 0.35;
A = 5;
T = 1/25000;
Ts = T/8;
t = [-A*T:Ts:A*T] + 10^(-8);
num = sin(pi*(1-alpha)*t/T) +
(4*alpha*t/T).*cos(pi*(1+alpha)*t/T);
den = (1-(4*alpha*t/T).^2).*(pi*t/T);
h = num./den;

DQPSK = [0 1 2 3];
N = 512;
esN0_dB = [0:2:10];
ps = zeros(1, length(esN0_dB));
pn = zeros(1, length(esN0_dB));

noErrs = zeros(1, length(esN0_dB));
noBitErrs = zeros(1, length(esN0_dB));

trails = 500;
%beta = 0.1;
tau = 1;%samples delayed after conv
for beta = [0.1, 0.2, 0.3, 0.5, 1]
    for ii = 1:length(esN0_dB)
        for j = 1:trails
            x = randsrc(1, N, DQPSK);
            bits = zeros(1, 2*N);
            for i = 1:N
                if x(i)==0
                    bits(2*i-1)= 0;
                    bits(2*i) = 0;
                elseif x(i)==1
                    bits(2*i-1)= 0;
                    bits(2*i) = 1;
                elseif x(i)==2
                    bits(2*i-1)= 1;
                    bits(2*i) = 0;
                else
                    bits(2*i-1)= 1;
                    bits(2*i) = 1;
                end
            end
            prev = exp(1i*pi/4);
            s = zeros(1,N);
            deltatheta = [pi/4, 3*pi/4, -pi/4, -3*pi/4];
```

```
            for i = 1:N
                s(i)= prev*exp(1i*deltatheta(x(i)+1));
                prev = s(i);
            end

            y = upsample(s, 8);
            w = conv(y, h);
            wdelay = beta*delayseq(w, tau);
            wf = w + wdelay;
            W = conv(wf, h);

            ps = norm(W(81:8:end-80))^2;
            w1 = awgn(wf, esN0_dB(ii));
            %noise = 10^(-esN0_dB(i)/20)*noise;
            noise = w1 - wf;
            nw = conv(noise, h);

            pn = norm(nw(81:8:end-80))^2;
            %estimate alpha from ps, pn and dB
            alpha1 =
sqrt(ps/(pn*(10^((esN0_dB(ii)+3)/10))));


            final = W + alpha1 * nw;


            rs = final(81:8:end-80);
            rx = zeros(1,2*N);

            prev = exp(1i*pi/4);
            for i = 1:N
                if real(rs(i)*conj(prev))>0
                    rx(2*i)= 0;
                else
                    rx(2*i) = 1;
                end
                if imag(rs(i)*conj(prev))>0
                    rx(2*i-1)= 0;
                else
                    rx(2*i-1) = 1;
                end
                prev = rs(i);
            end
            noErrs(ii) = noErrs(ii) + size(find([bits-
rx]),2);
            noBitErrs(ii) = noBitErrs(ii) +
size(find([bits- rx]),2);
```

```
        end


    end
    ser = noErrs/(2*N*trails);
    ber = noBitErrs/ (2*N*trails);
    figure(1)
    plot(esN0_dB, ber)
    grid on
    hold on


end

a = (2-sqrt(2))*10.^(esN0_dB/10);
b = (2+sqrt(2))*10.^(esN0_dB/10);

BER = 0.5*(1-
marcumq(sqrt(b),sqrt(a))+marcumq(sqrt(a),sqrt(b)));


plot(esN0_dB, BER,'b')
ylabel('BER')
xlabel('E_b/N_0 (dB)')
title('Bit Error Rate for Coherent DQPSK with ISI varying
\alpha')
legend('\alpha = 0.1','\alpha = 0.2','\alpha =
0.3','\alpha = 0.5','\alpha = 1', 'theoretical, no ISI')
```