Memorial University of Newfoundland
Faculty of Engineering and Applied Science

ENGI 9872: Digital Communications

Project Report

Instructor: Dr. Octavia Dobre

Group members:

Ziyang Li #201893331

Tong Jiang #201891214

Shuo Zhang #201892979

Date: Dec 4^{th}, 2019

Problem Analysis

In this project, a communication system consists of Bit source, transmitter, channel, receiver.

In the beginning, an input signal is generated as a sequence of bits by the Bit source. This sequence of bits is then transferred to transmitter from bit source. At the transmitter side, QPSK modulator is used to convert the sequence of bits into symbols. Then, the root raised cosine filter is used for pulse shaping with roll-off factor 0.35. After this, the output signal is sent through the channel.

In this project, we use two channels to investigation: Additive White Gaussian Noise (AWGN) and Rayleigh fading channel. The AWGN channel we use the MATLAB function 'awgn' to add to the channel. Rayleigh fading channel is modulated by multiplying the complex Gaussian random variable. At receiver side, Single and multiple antennas are employed for receiving the transmitted signal. Maximum ratio combining (MRC) which is a method of diversity combining, is used to combine the transmitted signals from single and multiple antennas.

Methodological approach

The prerequisite for this project is we have assumed that perfect synchronization is accomplished in this project. QPSK is the modulation technique used for the process of modulation and demodulation. PSD graph is drawn by using MATLAB function 'pwelch' at various values of roll-off factor. In AWGN channel, signal constellation has been studied at the receiver for different values of the SNR. In both channels, Bit error rate (BER) as a function of SNR is investigated and simulation result. Finally, the phase error is simulated in different numbers of antennas.

Simulations and considerations

Power spectral density about roll-off factor

The figure in shown in Fig.1. The outcomes of pulse shaping are used raised cosine filter, and the PSD is represented for QPSK when roll-off α is changing in $[0,1]$ with the value of $0, 0.35, 0.7$. The side lobes are reduced for the roll-off increases from 0 to 1. On the other hand, the power concentrate in the main lobe and bandwidth also increases. To get more insights, the bandwidth increases as α approaches 1. There is an adjustment between reduce the influence by intersymbol interference (ISI) and more bandwidth consumption.

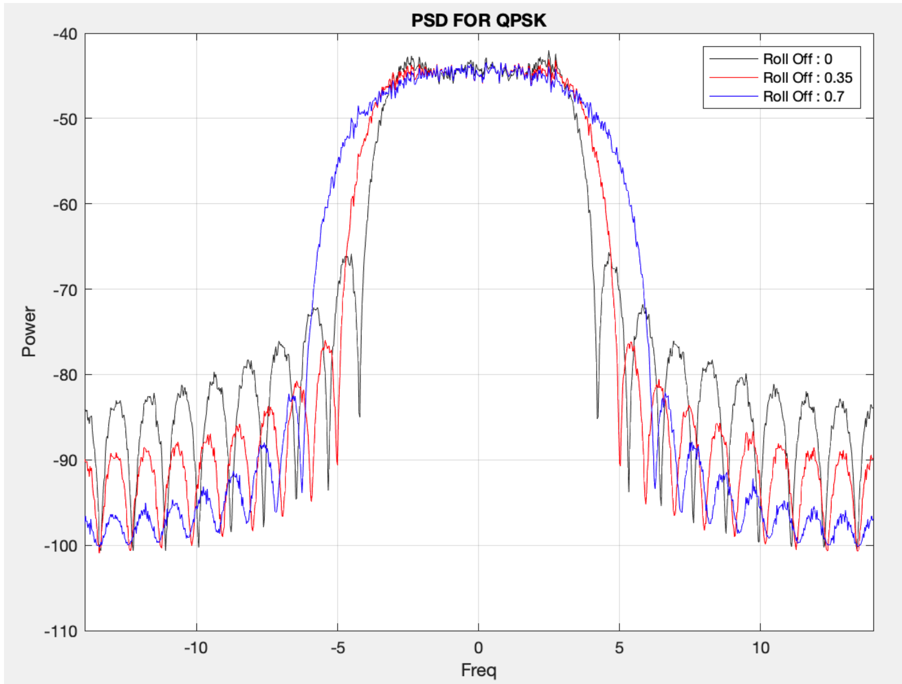

Fig.1 PSD for different values of roll-off factor

Signal constellation in AWGN channel

The following figures (Fig.2 to Fig.5) shows signal constellation at receiver for different values of SNR with the phase equal to $\frac{\pi}{4}$. According to the simulation, the less SNR, the more integrated clouds are mixed (we always ignore the negative values). As the SNR increase, the clouds are separated, and signal becomes concentrated. When $SNR = 20dB$, the clouds are clearly separated.

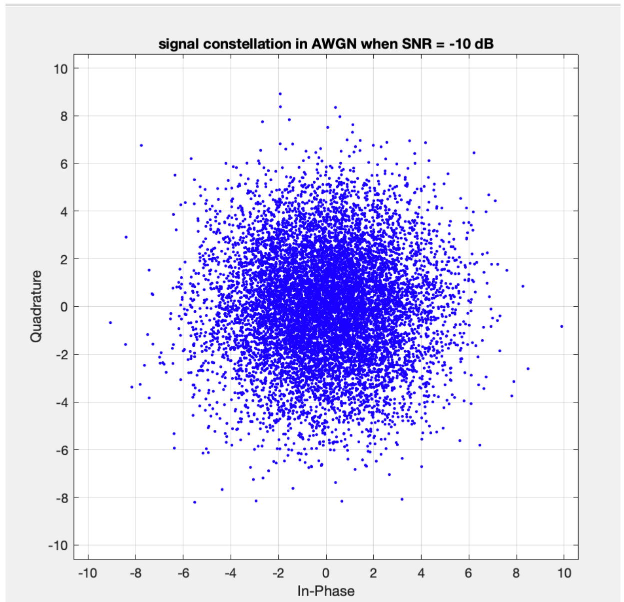

Fig.2 Signal constellation in AWGN channel when $SNR = -10dB$

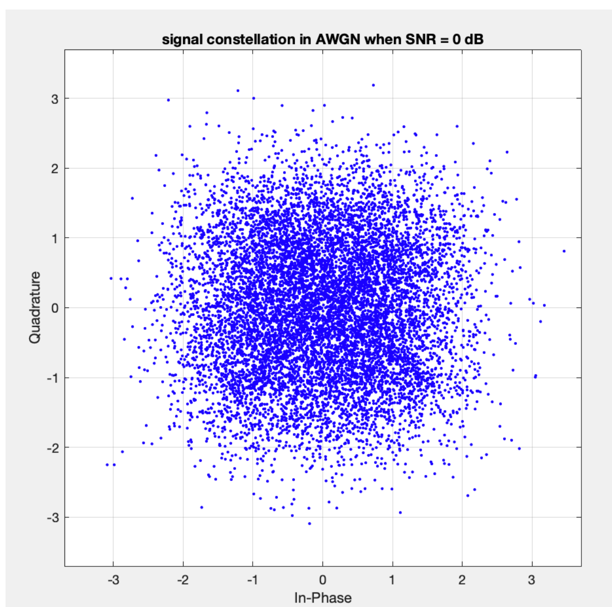

Fig.3 Signal constellation in AWGN channel when $SNR = 0dB$

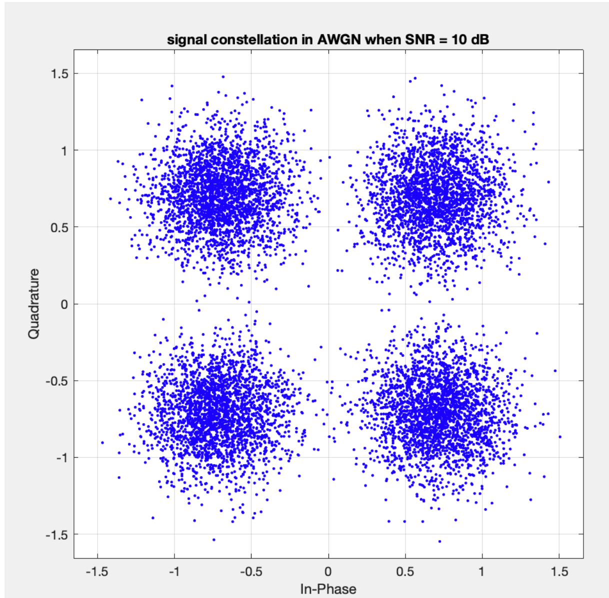

Fig.4 Signal constellation in AWGN channel when $SNR = 10dB$

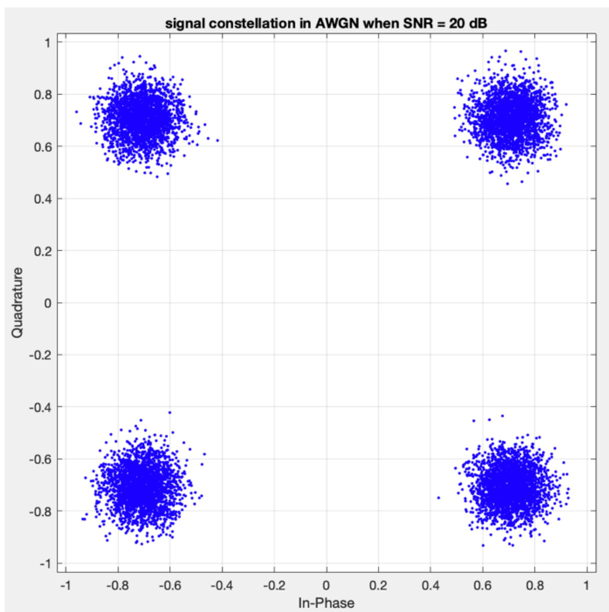

Fig.5 Signal constellation in AWGN channel when $SNR = 20dB$

QPSK modulation with AWGN and Rayleigh fading channels

In the Fig.6, bit error rate (BER) curve for QPSK modulation with AWGN and Rayleigh fading channels are shown. The blue and orange curves are theoretical AWGN and Rayleigh respectively. When the SNR increase, the BER in the both channels are decreasing. The BER for AWGN is exponential and for Rayleigh is nearly linear. The equalization needs to be done for signals over Rayleigh fading channel by dividing the Rayleigh fading factor. It is observed that the performance of AWGN is better because the signals suffer both noise and fading effects through Rayleigh fading channel. During the simulation, our results are almost match theoretical curve perfectly.

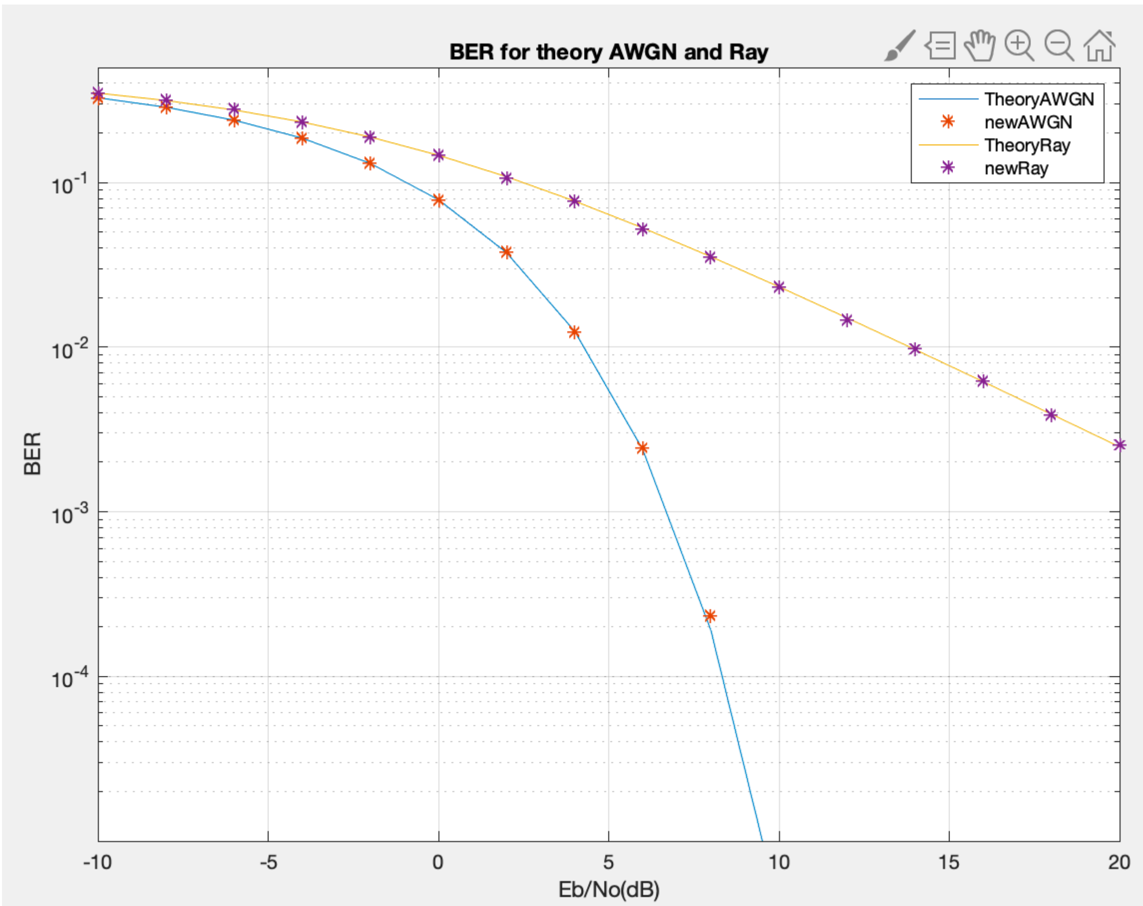

Fig.6 The BER comparison between theoretical and simulation results

Phase estimation for AWGN and Rayleigh fading channels

The phase (phase is $\frac{\pi}{6}$) is estimated on maximum likelihood method in both channels and shows in Fig.9. For AWGN channel, the phase estimation is more accurate and stable than Rayleigh channel. Fig.7 shows the phase estimation through AWGN channel. The blue curve is the signal with no phase and the orange one is signal with phase. The cross symbol is the result after phase estimation. Fig.8 shows the phase estimation through Rayleigh channel, and the legend is the same as AWGN figure. The phase offset is $\left(-\frac{\pi}{4}, \frac{\pi}{4}\right)$. If the offset out of range, the utilized estimator is not able to compute the phase.

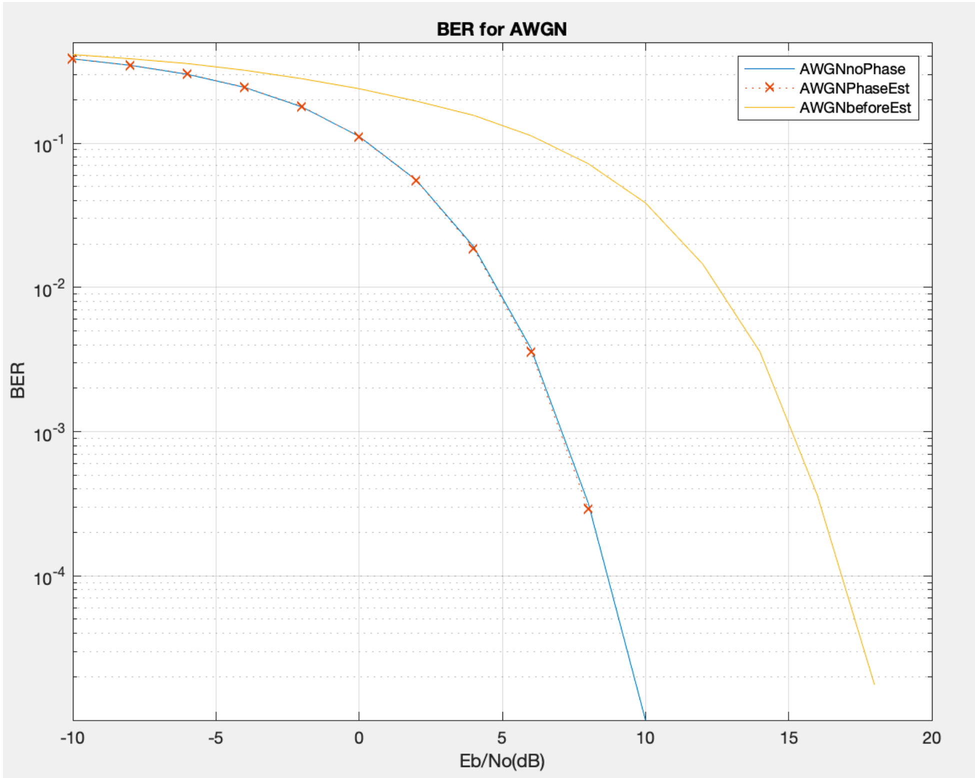

Fig.7 The BER with phase estimation through AWGN channel

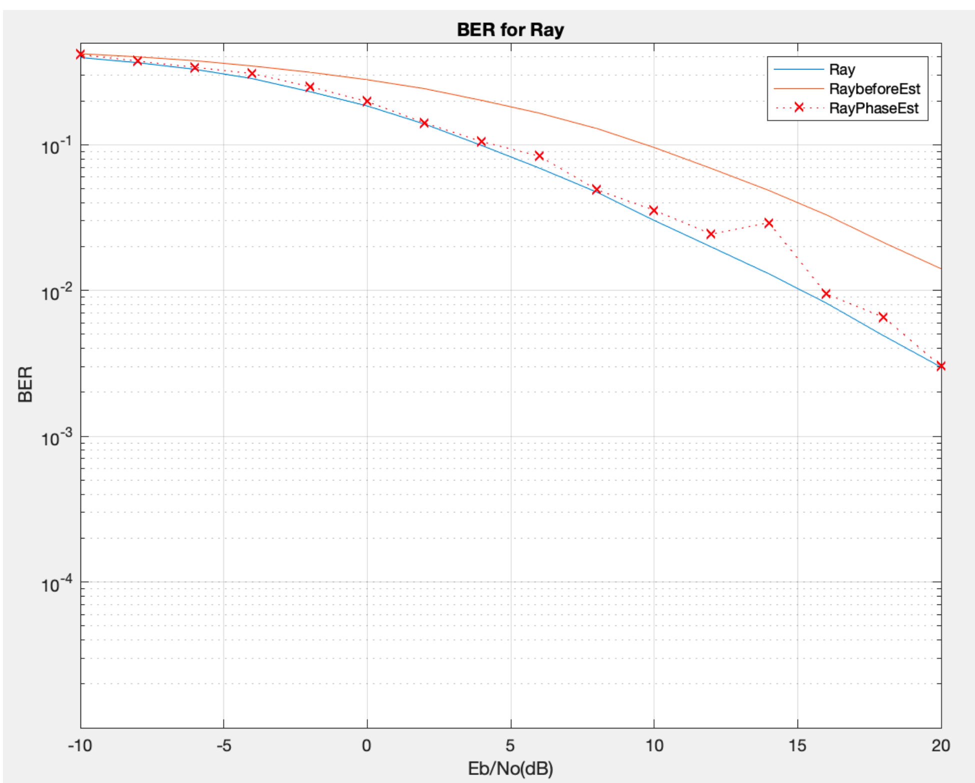

Fig.8 The BER with phase estimation through Rayleigh fading channel

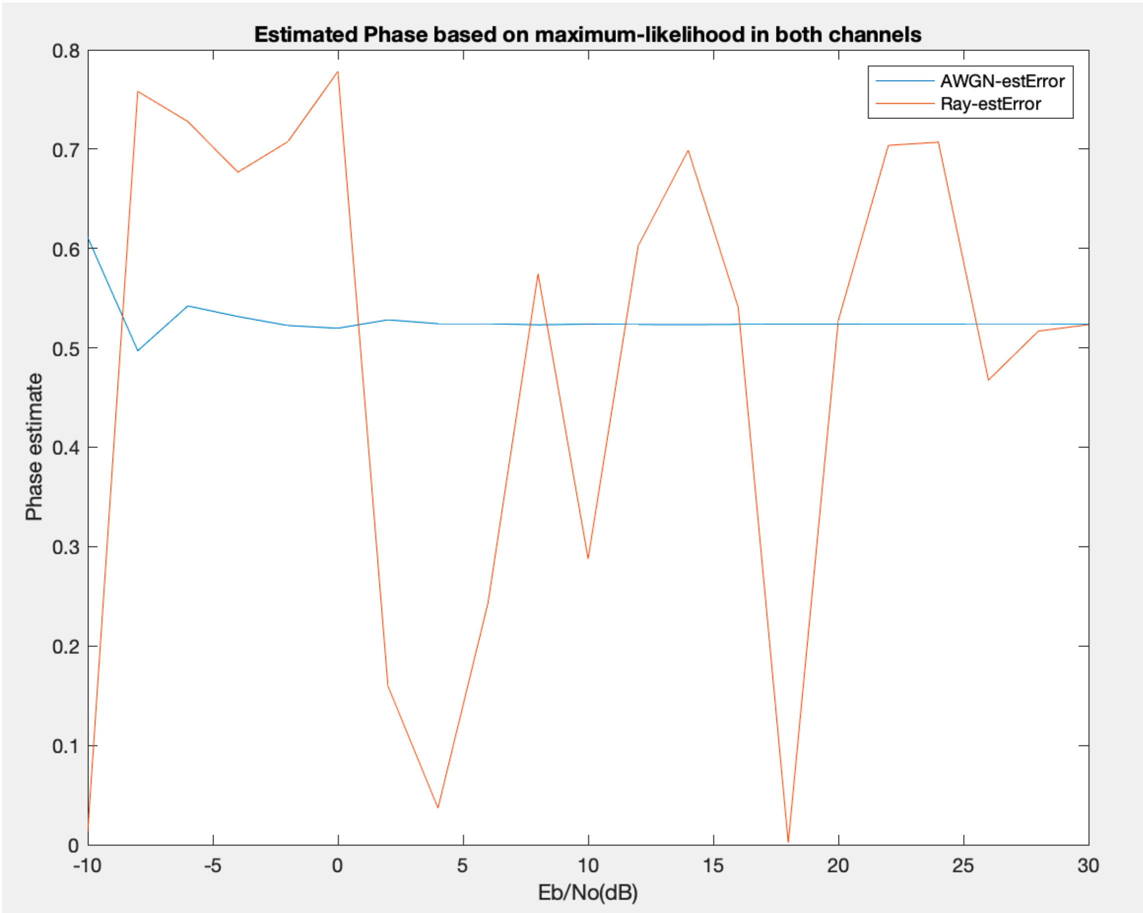

Fig.9 The phase estimation in two channels

BER for AWGN and Rayleigh fading channels with diversity antennas

Fig. shows the BER through AWGN channel with $L = 1, 2, 4$ antennas. While the number of antennas is increasing, the BER at the same SNR is decreasing. In other words, to achieve the same BER, the required SNR at the receiver decreases as the number of antennas increase. In Fig.10, the gap between one antenna and two antennas is $10 \times \log_{10} 2 \approx 3$, and the gap between one antenna and four antennas is $10 \times \log_{10} 4 \approx 6$.

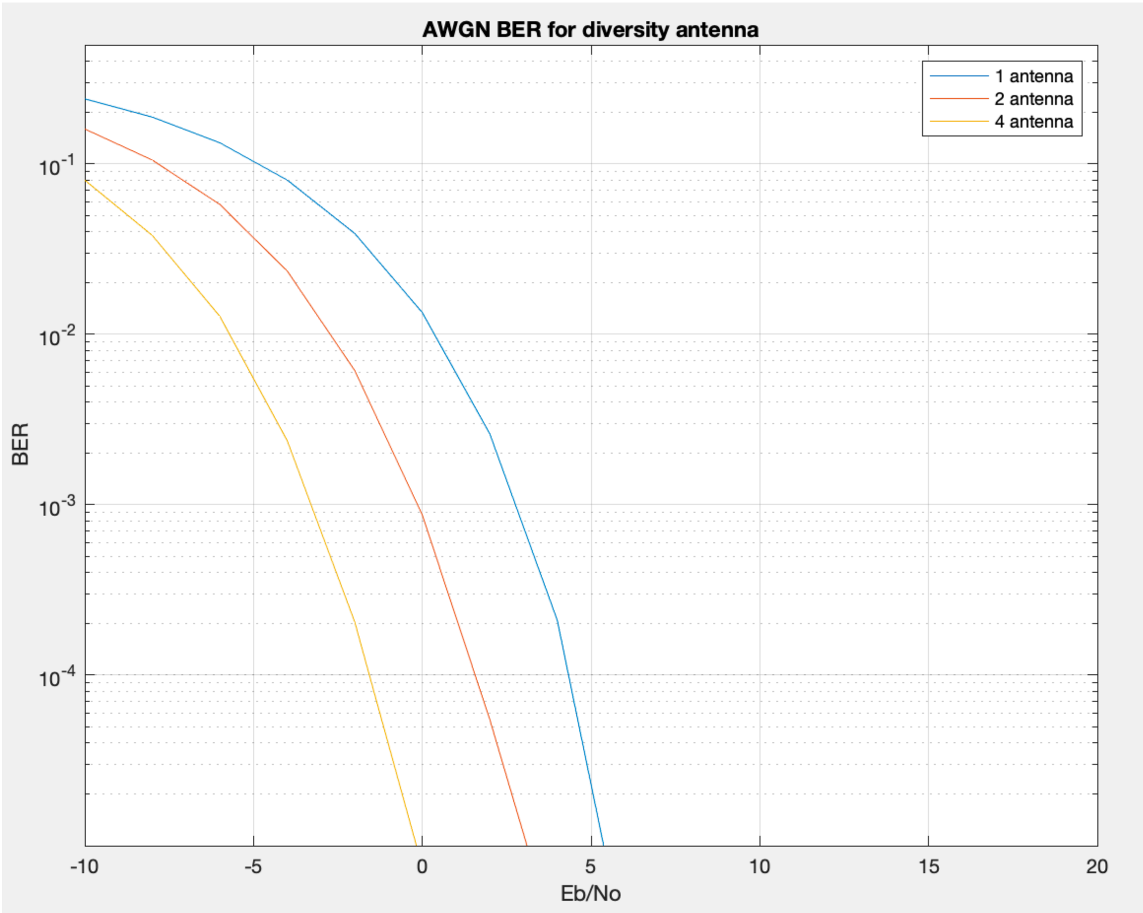

Fig.10 AWGN BER for diversity antennas with $L = 1, 2, 4$

Fig.11 shows the BER through Rayleigh fading channel with $L = 1, 2, 4$ antennas. We also compare the results with the theoretical curve by changing the diversity order. The same as AWGN, the number of antennas is increasing, the BER at the same SNR is decreasing.

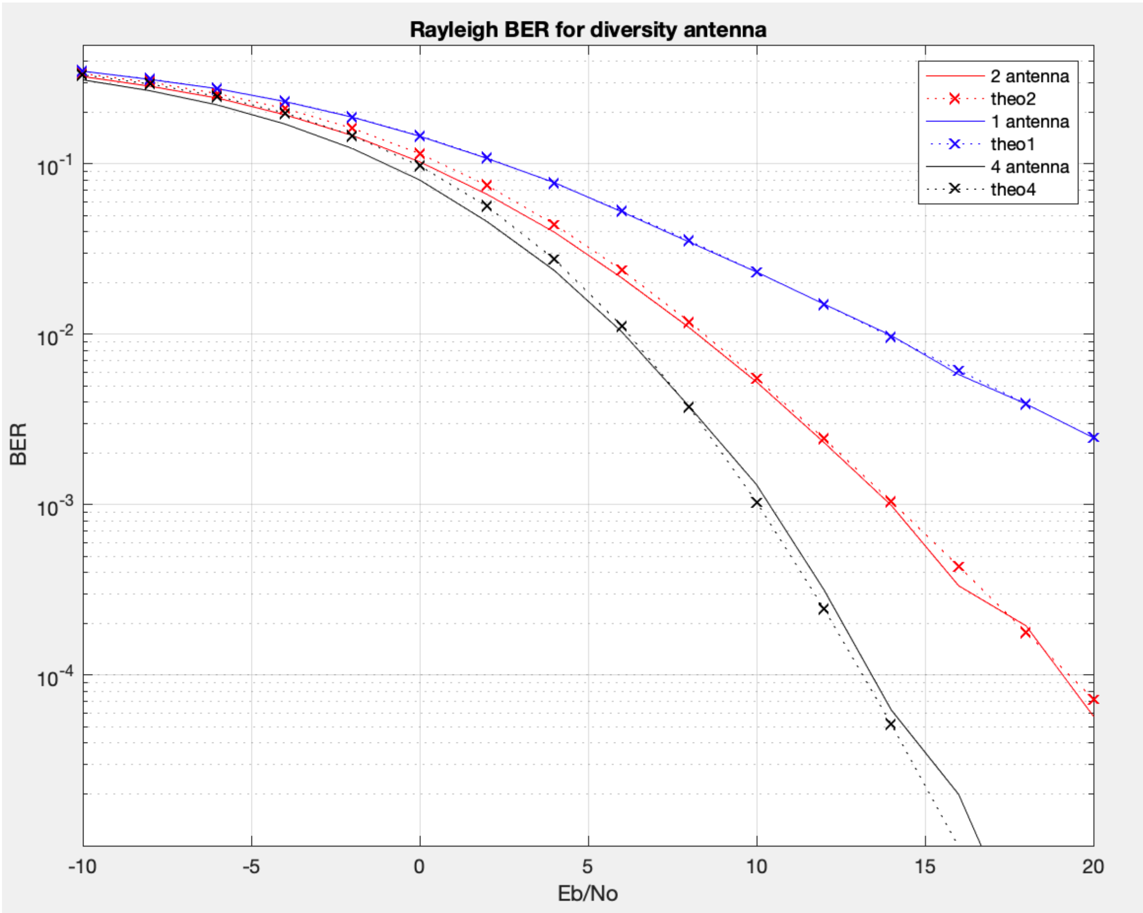

Fig.11 Rayleigh fading BER for diversity antennas with $L = 1, 2, 4$

The phase error through AWGN and Rayleigh fading channel

Phase error is calculated by getting the discrepancy between the value of phase estimation and real phase. The prerequisite of this part is we define SNR as a constant value and change the values of the total data. At the same time, the phase error also compares the different scenario by diversity antennas. Fig12. shows the AWGN channel among $L = 1, 2, 4$ antennas. The phase error decreases when the number of antennas and total data number increasing. However, the phase error through Rayleigh fading can be influenced by the number of antennas, not very clearly influenced by data number by Fig.13. Fig.14 is the same figure with different y-axis range as Fig.13.

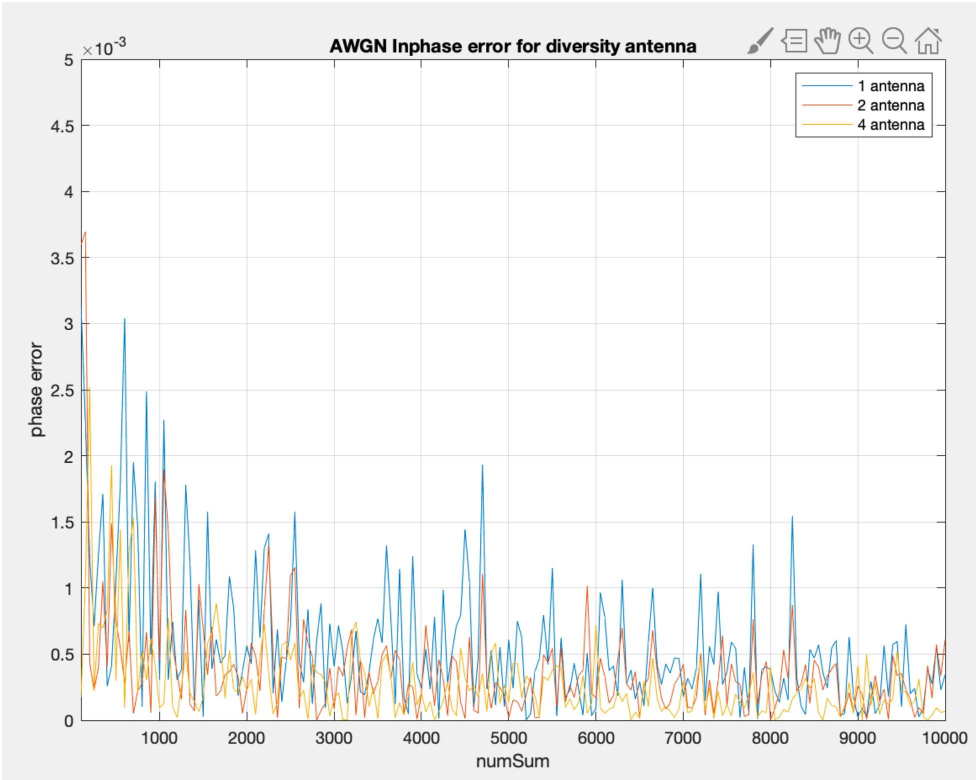

Fig.12 AWGN in phase error for diversity antennas with $L = 1, 2, 4$

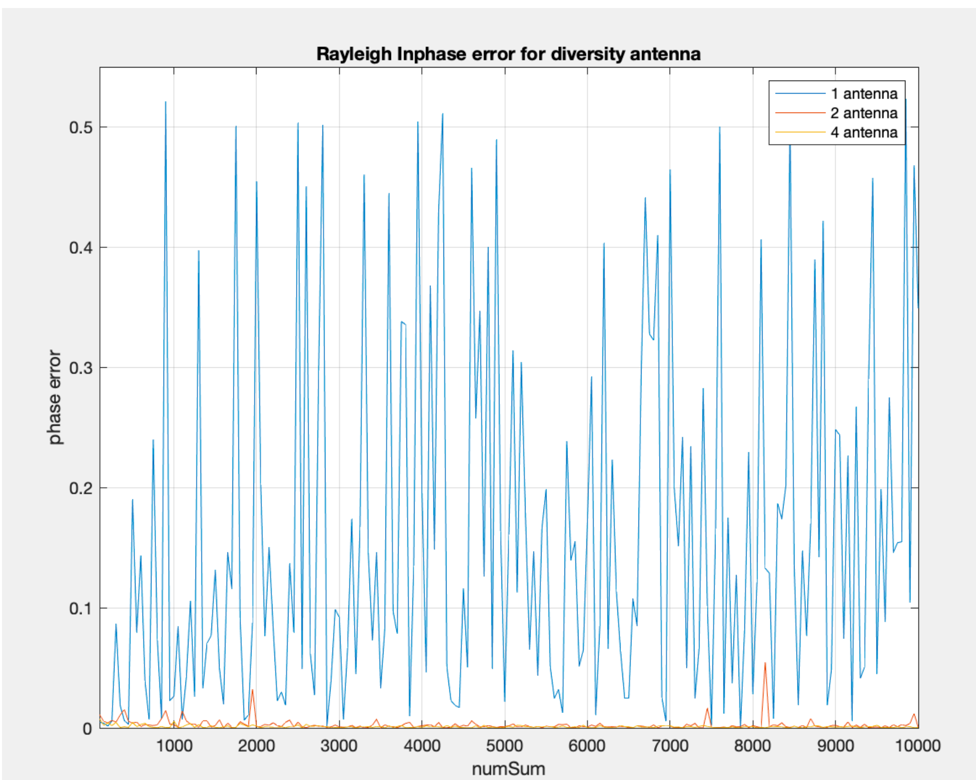

Fig.13 Rayleigh fading in phase error for diversity antennas with $L = 1, 2, 4$

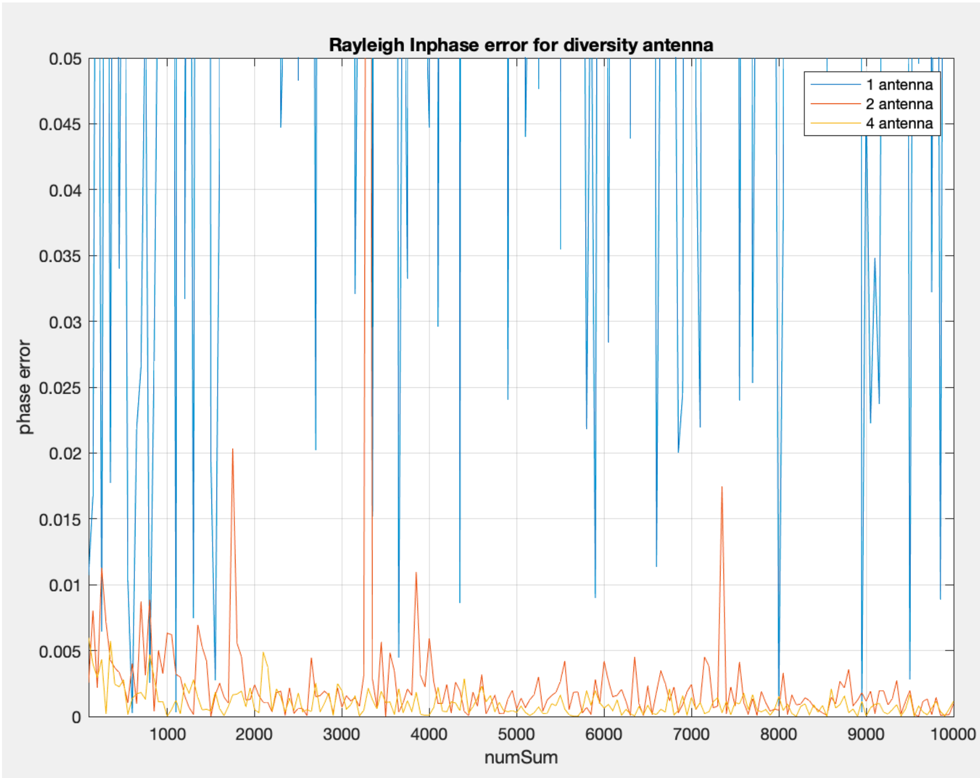

Fig.14 Rayleigh fading in phase error for diversity antennas with $L = 1, 2, 4$ in different y-axis range

Conclusion

In this project, the digital communication system based on QPSK has been simulated by using the MATLAB_R2018b environment We have mainly compared the BER for AWGN and Rayleigh fading channel. During this project, we have already known the basic link which contain transmitter, receiver, and channel.

We have learnt PSD for different roll-off. The effect of square root raised cosine filter is examined for the power spectral density for the transmitted signal where the increase of roll-off value leads to reduce the side lobes where the ISI vanishes. At the same time, we simulate the constellation for QPSK through AWGN channel in several SNR.

The bit error rate is the main part to compare both channel after modulation with the theoretical values. The BER also play an important role in the phase estimation. According above, we learnt that the phase estimation in AWGN is more stable than Rayleigh.

Finally, we focus on the diversity antennas through both two channels. The BER for $L = 1, 2, 4$ antennas indicate that to achieve the same BER, the required SNR at the receiver decreases as the number of antennas increase. The phase error we also plot on the figure, and find the relation between two channels.

Appendix A: MATLAB code for PSD and comparison between calculation and theoretical values.

```
clear
close all


%Tong jiang, ziyang li, shuo zhang
M = 4;
k = log2(M);
Ph_err = pi/4;
symbolrate = 7e3;
rolloff = 0:0.35:1;
data = randi([0 M-1], 10000, 1);
mod = pskmod(data, M, Ph_err);
span = 6;%6 symbols
sps = 4;%4samples
for i= 1:length(rolloff)
    rcosfilter = rcosdesign(rolloff(i),span,sps,'sqrt');
    upsample = upfirdn(mod, rcosfilter, sps);%upsampling

    figure(1);
    pwelch(upsample,hamming(1024),[],[],symbolrate*sps,'centered');
    hold on;
    grid on;
    legend('Roll Off : 0','Roll Off : 0.35','Roll Off : 0.7');
    xlabel('Freq');
    ylabel('Power');
    title('PSD FOR QPSK');
end

%% CONSTELLATION%%%%%%%%%


rxSig = awgn(mod,20);
figure(2)
scatterplot(rxSig)
grid on
title('signal constellation in AWGN when SNR = 20 dB')%constellation in
```

```
20db

rxSig1 = awgn(mod,10);
figure(3)
scatterplot(rxSig1)
grid on
title('signal constellation in AWGN when SNR = 10 dB')%constellation in
-10db

rxSig2 = awgn(mod,0);
figure(4)
scatterplot(rxSig2)
grid on
title('signal constellation in AWGN when SNR = 0 dB')%constellation in
0db

rxSig3 = awgn(mod,-10);
figure(5)
scatterplot(rxSig3)
grid on
title('signal constellation in AWGN when SNR = -10 dB')%constellation
in 10db

%% One antenna compare theo AWGN and Ray%%%%%%%%

EbNoVec = -10:2:30;
berAWGN = zeros(size(EbNoVec));
berRay = zeros(size(EbNoVec));
numSum = 1e5;

%% ber awgn%%%%
for i = 1:length(EbNoVec)

    snrdb = EbNoVec(i) + 10*log10(k);

        dataIn = randi([0,M-1],numSum,k);

        txSig0 = pskmod(dataIn, M, Ph_err,'gray');

        rxSig0 = awgn(txSig0,snrdb,'measured');
```

```
        rxSym0 = pskdemod(rxSig0, M, Ph_err, 'gray');

        [numErr0,berAWGN(i)] = biterr(dataIn, rxSym0);


end


%% ber ray%%%

for i = 1: length(EbNoVec)

    snrdb = 10.^(0.1*EbNoVec(i));

        dataIn0 = randi([0,M-1], numSum, k);

        txSig1 = pskmod(dataIn0, M,Ph_err,'gray');
        noiseRay = 1/sqrt(2*snrdb*k)*(randn(length(txSig1),1) +
1i*randn(length(txSig1),1));% Complex Gaussian niose

        Rayleigh = 1/sqrt(2)*(randn(length(txSig1),1) +
1i*randn(length(txSig1),1));%Rayleigh

        rxSig1 = (Rayleigh.*txSig1+ noiseRay)./Rayleigh;%Rayleigh
fading : output of channel
        rxSym1 = pskdemod(rxSig1, M,Ph_err,'gray');


        [nError1,berRay(i)] = biterr(dataIn0, rxSym1);


end


%% theo%%%%
theoAWGN = berawgn(EbNoVec, 'psk', M,'nondiff');
theoRay = berfading(EbNoVec, 'psk',M , 1);%1 is diversity order
figure(7)
semilogy(EbNoVec,theoAWGN,EbNoVec, berAWGN,'*',EbNoVec,theoRay,EbNoVec,
berRay,'*')
axis([-10,20,1e-5,0.5]);
```

```
grid on;
legend('TheoryAWGN','newAWGN','TheoryRay','newRay');
xlabel('Eb/No(dB)');
ylabel('BER');
title('BER for theory AWGN and Ray');
```

Appendix B: MATLAB for Diversity antennas

```
clear
clc
close all

M = 4;
EbNoVec = -10:2:30;
numSum = 1e5;
ph_err = pi/6;
k = log2(M);


%% 2 antenna%%%%%%%

BerAWGN_2Ant = [];
BerAWGN_1Ant = [];
BerRay_2Ant = [];
BerRay_1Ant = [];

for j = 1 : length(EbNoVec)
    snrdb = EbNoVec(j)+10*log(k);
    dataIn = randi([0,M-1], numSum , k);
    txRig = pskmod(dataIn, M,ph_err,'gray');


    rxSig_a1 = awgn(txRig,snrdb,'measured');
    rxSig_a2 =  awgn(txRig,snrdb,'measured');
    rxSig_tot2 = 1/2*(rxSig_a1 + rxSig_a2);

    rxSym_a1 = pskdemod(rxSig_a1,M,ph_err,'gray');
    rxSym_tot = pskdemod(rxSig_tot2,M,ph_err,'gray');
```

```
    [n1,BerAWGN_1Ant(j)] = biterr(rxSym_a1,dataIn);
    [n2,BerAWGN_2Ant(j)] = biterr(rxSym_tot, dataIn);



end

%% ray_2antenna

for i = 1 : length(EbNoVec)
    snrdb = 10.^(0.1*EbNoVec(i));
    dataInRay2 = randi([0,M-1], numSum , k);
    txRigRay2 = pskmod(dataInRay2, M,ph_err,'gray');

    %%%% #1
    noiseRay_a1 = 1/sqrt(2*snrdb*k)*(randn(length(txRigRay2),1) +
1i*randn(length(txRigRay2),1));% Complex Gaussian niose

    Rayleigh_a1 = 1/sqrt(2)*(randn(length(txRigRay2),1) +
1i*randn(length(txRigRay2),1));%Rayleigh

    Rayleigh_nominator_a1 = Rayleigh_a1.*txRigRay2+ noiseRay_a1;

    rxSigRay_one = Rayleigh_nominator_a1./Rayleigh_a1;%Rayleigh fading :
output of channel

    %%%% #2

    noiseRay_a2 = 1/sqrt(2*snrdb*k)*(randn(length(txRigRay2),1) +
1i*randn(length(txRigRay2),1));% Complex Gaussian niose

    Rayleigh_a2 = 1/2*1/sqrt(2)*(randn(length(txRigRay2),1) +
1i*randn(length(txRigRay2),1));%Rayleigh

    Rayleigh_nominator_a2 = Rayleigh_a2.*txRigRay2+ noiseRay_a2;

    rxSigRay_two = Rayleigh_nominator_a2./Rayleigh_a2;%Rayleigh fading :
output of channel

    %%%% combine
```

```
    nominator = conj(Rayleigh_a1) .* Rayleigh_nominator_a1 +
conj(Rayleigh_a2) .* Rayleigh_nominator_a2;

    denominator = conj(Rayleigh_a1) .* Rayleigh_a1 +
conj(Rayleigh_a2) .* Rayleigh_a2;

    equalizer = nominator./denominator;

    demodDataRay2 = pskdemod(equalizer,M,ph_err,'gray');

    demodDataRay1 = pskdemod(rxSigRay_one,M,ph_err,'gray');

     [nError1,BerRay_2Ant(i)] = biterr(dataInRay2, demodDataRay2);

     [nError2,BerRay_1Ant(i)] = biterr(dataInRay2, demodDataRay1);


end


%% AWGN 4 antenna
BerAWGN_4Ant = [];
for j = 1 : length(EbNoVec)
    snrdb = EbNoVec(j)+10*log(k);
    dataIn = randi([0,M-1], numSum , k);
    txRigAWGN4 = pskmod(dataIn, M,ph_err,'gray');


    rxSigAWGN_a1 = awgn(txRigAWGN4,snrdb,'measured');
    rxSigAWGN_a2 = awgn(txRigAWGN4,snrdb,'measured');
    rxSigAWGN_a3 = awgn(txRigAWGN4,snrdb,'measured');
    rxSigAWGN_a4 = awgn(txRigAWGN4,snrdb,'measured');

    rxSigAWGN_tot4 = 1/4*(rxSigAWGN_a1 + rxSigAWGN_a2 + rxSigAWGN_a3 +
rxSigAWGN_a4);


    rxSymAWGN_tot4 = pskdemod(rxSigAWGN_tot4,M,ph_err,'gray');

    [n2,BerAWGN_4Ant(j)] = biterr(rxSymAWGN_tot4, dataIn);
```

```
end

%% Ray 4 antenna

BerRay_4Ant = [];
for i = 1 : length(EbNoVec)
    snrdb = 10.^(0.1*EbNoVec(i));
    dataInRay4 = randi([0,M-1], numSum , k);
    txRigRay4 = pskmod(dataInRay4, M,ph_err,'gray');

    %%%% #1
    noiseRay_a1 = 1/sqrt(2*snrdb*k)*(randn(length(txRigRay4),1) +
1i*randn(length(txRigRay4),1));% Complex Gaussian niose

    Rayleigh_a1 = 1/sqrt(2)*(randn(length(txRigRay4),1) +
1i*randn(length(txRigRay4),1));%Rayleigh

    Rayleigh_nominator_a1 = Rayleigh_a1.*txRigRay4+ noiseRay_a1;

    rxSigRay_one = Rayleigh_nominator_a1./Rayleigh_a1;%Rayleigh fading :
output of channel

    %%%% #2

    noiseRay_a2 = 1/sqrt(2*snrdb*k)*(randn(length(txRigRay4),1) +
1i*randn(length(txRigRay4),1));% Complex Gaussian niose

    Rayleigh_a2 = 1/2*1/sqrt(2)*(randn(length(txRigRay4),1) +
1i*randn(length(txRigRay4),1));%Rayleigh

    Rayleigh_nominator_a2 = Rayleigh_a2.*txRigRay4+ noiseRay_a2;

    rxSigRay_two = Rayleigh_nominator_a2./Rayleigh_a2;%Rayleigh fading :
output of channel


    %%%% #3

    noiseRay_a3 = 1/sqrt(2*snrdb*k)*(randn(length(txRigRay4),1) +
```

```
1i*randn(length(txRigRay4),1));% Complex Gaussian niose

   Rayleigh_a3 = 1/3*1/sqrt(2)*(randn(length(txRigRay4),1) +
1i*randn(length(txRigRay4),1));%Rayleigh

   Rayleigh_nominator_a3 = Rayleigh_a3.*txRigRay4+ noiseRay_a3;

   rxSigRay_three = Rayleigh_nominator_a3./Rayleigh_a3;%Rayleigh
fading : output of channel


   %%%% #4

   noiseRay_a4 = 1/sqrt(2*snrdb*k)*(randn(length(txRigRay4),1) +
1i*randn(length(txRigRay4),1));% Complex Gaussian niose

   Rayleigh_a4 = 1/4*1/sqrt(2)*(randn(length(txRigRay4),1) +
1i*randn(length(txRigRay4),1));%Rayleigh

   Rayleigh_nominator_a4 = Rayleigh_a4.*txRigRay4+ noiseRay_a4;

   rxSigRay_four = Rayleigh_nominator_a4./Rayleigh_a4;%Rayleigh
fading : output of channel

   %%%% combine

   nominator = conj(Rayleigh_a1) .* Rayleigh_nominator_a1 +
conj(Rayleigh_a2) .* Rayleigh_nominator_a2 + conj(Rayleigh_a3) .*
Rayleigh_nominator_a3 + conj(Rayleigh_a4) .* Rayleigh_nominator_a4;

   denominator = conj(Rayleigh_a1) .* Rayleigh_a1 +
conj(Rayleigh_a2) .* Rayleigh_a2 + conj(Rayleigh_a3) .* Rayleigh_a3 +
conj(Rayleigh_a4) .* Rayleigh_a4;

   equalizer = nominator./denominator;

   demodDataRay4 = pskdemod(equalizer,M,ph_err,'gray');



    [nError1,BerRay_4Ant(i)] = biterr(dataInRay4, demodDataRay4);
```

```
end


theoRay1 = berfading(EbNoVec, 'psk',M , 1);%1 is diversity order
theoRay2 = berfading(EbNoVec, 'psk',M , 2);%2 is diversity order
theoRay4 = berfading(EbNoVec, 'psk',M , 4);%4 is diversity order



figure(1)
semilogy(EbNoVec,BerRay_2Ant,'r',EbNoVec,theoRay2,':rX',EbNoVec,BerRay_
1Ant,'b',EbNoVec,theoRay1,':bX',EbNoVec,BerRay_4Ant,'k',EbNoVec,theoRay
4,':kX')
grid on;
legend('2 antenna','theo2','1 antenna','theo1','4 antenna','theo4');
xlabel('Eb/No');
ylabel('BER');
axis([-10,20,1e-5,0.5]);
title('Rayleigh BER for diversity antenna');

figure(2)

semilogy(EbNoVec,BerAWGN_1Ant,EbNoVec,BerAWGN_2Ant,EbNoVec,BerAWGN_4Ant
);
grid on;
legend('1 antenna','2 antenna','4 antenna');
xlabel('Eb/No');
ylabel('BER');
axis([-10,20,1e-5,0.5]);
title('AWGN BER for diversity antenna');
```

Appendix C: MATLAB for Phase estimation for AWGN and Rayleigh fading channels

```
clear
clc
close all
```

```
M = 4;
Ph_err = pi/6;
k = log2(M);
EbNoVec = -10:2:30;
berAWGN = zeros(size(EbNoVec));
berRay = zeros(size(EbNoVec));
numSum = 1e5;

%% AWGN Phase est

for i = 1:length(EbNoVec)

    snrdb = EbNoVec(i) + 10*log10(k);

        dataIn = randi([0,M-1],numSum,k);

        txSig0 = pskmod(dataIn, M,Ph_err);
        txSig0_noPhase = pskmod(dataIn, M);
        rxSig0 = awgn(txSig0,snrdb,'measured');
        rxSig0_noPhase = awgn(txSig0_noPhase,snrdb,'measured');

        % phase est
        x4 = pskmod([0,M-1],M);
        x4_conj = mean((conj(x4)).^4);
        Rx_4 = mean((rxSig0).^4);
        Ph_err_est = 0.25*mean(angle(x4_conj*Rx_4));

        estArray(i) = Ph_err_est;

        rxSym0 = pskdemod(rxSig0_noPhase, M);

        rxSym0_beforeEst = pskdemod(rxSig0, M);

        rxSym0_est = pskdemod(rxSig0, M,Ph_err_est);

        [numErr0,berAWGN(i)] = biterr(dataIn, rxSym0_beforeEst);

        [numEr,berAWGN_noPhase(i)] = biterr(dataIn,rxSym0);
```

```
        [numE, berAWGN_est(i)] = biterr(dataIn,rxSym0_est);
end


%% Rayleigh Phase est

for i = 1: length(EbNoVec)

    snrdb = 10.^(0.1*EbNoVec(i));

        dataIn0 = randi([0,M-1], numSum, k);

        txSig1 = pskmod(dataIn0, M,Ph_err);
        noiseRay = 1/sqrt(2*snrdb*k)*(randn(length(txSig1),1) +
1i*randn(length(txSig1),1));% Complex Gaussian niose

        Rayleigh = 1/sqrt(2)*(randn(length(txSig1),1) +
1i*randn(length(txSig1),1));%Rayleigh

        rxSig1 = (Rayleigh.*txSig1+ noiseRay)./Rayleigh;%Rayleigh
fading : output of channel

        % phase est
        x4 = pskmod([0,M-1],M);
        x4_conj = mean((conj(x4)).^4);
        Rx_4 = mean((rxSig1).^4);
        Ph_err_est = abs(0.25*mean(angle(x4_conj*Rx_4)));

        estArrayRay(i) = abs(Ph_err_est);



        rxSym1 = pskdemod(rxSig1 , M,Ph_err);

        rxSym1_beforeEst = pskdemod(rxSig1, M);

        rxSym1_est = pskdemod(rxSig1, M,Ph_err_est);

        [nError1,berRay(i)] = biterr(dataIn0, rxSym1);
        [nError2, berRay_beforeEst(i)] = biterr(dataIn0,
rxSym1_beforeEst);
        [nError3, berRay_Est(i)] = biterr(dataIn0, rxSym1_est);
```

```
end
%%
figure(1)
semilogy(EbNoVec,berAWGN_noPhase,
EbNoVec,berAWGN_est,':X',EbNoVec,berAWGN)
axis([-10,20,1e-5,0.5]);
grid on;
legend('AWGNnoPhase','AWGNPhaseEst','AWGNbeforeEst');
xlabel('Eb/No(dB)');
ylabel('BER');
title('BER for AWGN');

figure(2)
semilogy(EbNoVec,berRay,
EbNoVec,berRay_beforeEst,EbNoVec,berRay_Est,':rX')
axis([-10,20,1e-5,0.5]);
grid on;
legend('Ray','RaybeforeEst','RayPhaseEst');
xlabel('Eb/No(dB)');
ylabel('BER');
title('BER for Ray');

figure(3)
plot(EbNoVec,estArray)
hold on;
plot(EbNoVec,estArrayRay)
legend('AWGN-estError','Ray-estError');
xlabel('Eb/No(dB)');
ylabel('Phase estimate');
title('Estimated Phase based on maximum-likelihood in both channels');
```

Appendix D: MATLAB for the phase error through AWGN and Rayleigh fading channel

```
clear
clc
close all
```

```matlab
M = 4;
EbNoVec = 15;
numSum = 100:50:10000;
ph_err = pi/6;
k = log2(M);


%% 2 antenna%%%%%%%


for j = 1 : length(numSum)
    snrdb = EbNoVec+10*log(k);
    dataIn = randi([0,M-1], numSum(j) , k);
    txRig = pskmod(dataIn, M,ph_err,'gray');


    rxSig_a1 = awgn(txRig,snrdb,'measured');
    rxSig_a2 =  awgn(txRig,snrdb,'measured');
    rxSig_tot2 = 1/2*(rxSig_a1 + rxSig_a2);


    % phase est
       x4 = pskmod([0,M-1],M);
       x4_conj = mean((conj(x4)).^4);
       Rx_4 = mean((rxSig_a1).^4);
       Ph_err_est_a1 = 0.25*mean(angle(x4_conj*Rx_4));

     % phase est
       x5 = pskmod([0,M-1],M);
       x5_conj = mean((conj(x5)).^4);
       Rx_5 = mean((rxSig_tot2).^4);
       Ph_err_est_a1a2 = 0.25*mean(angle(x5_conj*Rx_5));


     Sub_01(j) = abs(Ph_err_est_a1 - ph_err);

     Sub_02(j) = abs(Ph_err_est_a1a2 - ph_err);
```

```
end

%% ray_2antenna

for i = 1 : length(numSum)
    snrdb = 10.^(0.1*EbNoVec);
    dataInRay2 = randi([0,M-1], numSum(i) , k);
    txRigRay2 = pskmod(dataInRay2, M,ph_err,'gray');

    %%%% #1
    noiseRay_a1 = 1/sqrt(2*snrdb*k)*(randn(length(txRigRay2),1) +
1i*randn(length(txRigRay2),1));% Complex Gaussian niose

    Rayleigh_a1 = 1/sqrt(2)*(randn(length(txRigRay2),1) +
1i*randn(length(txRigRay2),1));%Rayleigh

    Rayleigh_nominator_a1 = Rayleigh_a1.*txRigRay2+ noiseRay_a1;

    rxSigRay_one = Rayleigh_nominator_a1./Rayleigh_a1;%Rayleigh fading :
output of channel

    %%%% #2

    noiseRay_a2 = 1/sqrt(2*snrdb*k)*(randn(length(txRigRay2),1) +
1i*randn(length(txRigRay2),1));% Complex Gaussian niose

    Rayleigh_a2 = 1/2*1/sqrt(2)*(randn(length(txRigRay2),1) +
1i*randn(length(txRigRay2),1));%Rayleigh

    Rayleigh_nominator_a2 = Rayleigh_a2.*txRigRay2+ noiseRay_a2;

    rxSigRay_two = Rayleigh_nominator_a2./Rayleigh_a2;%Rayleigh fading :
output of channel

    %%%% combine

    nominator = conj(Rayleigh_a1) .* Rayleigh_nominator_a1 +
conj(Rayleigh_a2) .* Rayleigh_nominator_a2;

    denominator = conj(Rayleigh_a1) .* Rayleigh_a1 +
conj(Rayleigh_a2) .* Rayleigh_a2;
```

```
    equalizer = nominator./denominator;

     % phase est
        x4 = pskmod([0,M-1],M);
        x4_conj = mean((conj(x4)).^4);
        Rx_4 = mean((rxSigRay_one).^4);
        Ph_err_RayEst_a1 = abs(0.25*mean(angle(x4_conj*Rx_4)));


     % phase est
        x5 = pskmod([0,M-1],M);
        x5_conj = mean((conj(x5)).^4);
        Rx_5 = mean((equalizer).^4);
        Ph_err_RayEst_a1a2 = 0.25*mean(angle(x5_conj*Rx_5));

     Sub_Ray01(i) = abs(Ph_err_RayEst_a1 - ph_err);

     Sub_Ray02(i) = abs(Ph_err_RayEst_a1a2 - ph_err);


end

%% AWGN 4 antenna

for j = 1 : length(numSum)
    snrdb = EbNoVec + 10*log(k);
    dataIn = randi([0,M-1], numSum(j) , k);
    txRigAWGN4 = pskmod(dataIn, M,ph_err,'gray');


    rxSigAWGN_a1 = awgn(txRigAWGN4,snrdb,'measured');
    rxSigAWGN_a2 = awgn(txRigAWGN4,snrdb,'measured');
    rxSigAWGN_a3 = awgn(txRigAWGN4,snrdb,'measured');
    rxSigAWGN_a4 = awgn(txRigAWGN4,snrdb,'measured');

    rxSigAWGN_tot4 = 1/4*(rxSigAWGN_a1 + rxSigAWGN_a2 + rxSigAWGN_a3 +
rxSigAWGN_a4);

    % phase est
        x5 = pskmod([0,M-1],M);
```

```
        x5_conj = mean((conj(x5)).^4);
        Rx_5 = mean((rxSigAWGN_tot4).^4);
        Ph_err_est4 = 0.25*mean(angle(x5_conj*Rx_5));


        Sub_04(j) = abs(Ph_err_est4 - ph_err);


end

%% Ray 4 antenna

for i = 1 : length(numSum)
    snrdb = 10.^(0.1*EbNoVec);
    dataInRay4 = randi([0,M-1], numSum(i) , k);
    txRigRay4 = pskmod(dataInRay4, M,ph_err,'gray');

    %%%% #1
    noiseRay_a1 = 1/sqrt(2*snrdb*k)*(randn(length(txRigRay4),1) +
1i*randn(length(txRigRay4),1));% Complex Gaussian niose

    Rayleigh_a1 = 1/sqrt(2)*(randn(length(txRigRay4),1) +
1i*randn(length(txRigRay4),1));%Rayleigh

    Rayleigh_nominator_a1 = Rayleigh_a1.*txRigRay4+ noiseRay_a1;

    rxSigRay_one = Rayleigh_nominator_a1./Rayleigh_a1;%Rayleigh fading :
output of channel

    %%%% #2

    noiseRay_a2 = 1/sqrt(2*snrdb*k)*(randn(length(txRigRay4),1) +
1i*randn(length(txRigRay4),1));% Complex Gaussian niose

    Rayleigh_a2 = 1/2*1/sqrt(2)*(randn(length(txRigRay4),1) +
1i*randn(length(txRigRay4),1));%Rayleigh

    Rayleigh_nominator_a2 = Rayleigh_a2.*txRigRay4+ noiseRay_a2;

    rxSigRay_two = Rayleigh_nominator_a2./Rayleigh_a2;%Rayleigh fading :
output of channel
```

```
    %%%% #3

    noiseRay_a3 = 1/sqrt(2*snrdb*k)*(randn(length(txRigRay4),1) +
1i*randn(length(txRigRay4),1));% Complex Gaussian niose

    Rayleigh_a3 = 1/3*1/sqrt(2)*(randn(length(txRigRay4),1) +
1i*randn(length(txRigRay4),1));%Rayleigh

    Rayleigh_nominator_a3 = Rayleigh_a3.*txRigRay4+ noiseRay_a3;

    rxSigRay_three = Rayleigh_nominator_a3./Rayleigh_a3;%Rayleigh
fading : output of channel


    %%%% #4

    noiseRay_a4 = 1/sqrt(2*snrdb*k)*(randn(length(txRigRay4),1) +
1i*randn(length(txRigRay4),1));% Complex Gaussian niose

    Rayleigh_a4 = 1/4*1/sqrt(2)*(randn(length(txRigRay4),1) +
1i*randn(length(txRigRay4),1));%Rayleigh

    Rayleigh_nominator_a4 = Rayleigh_a4.*txRigRay4+ noiseRay_a4;

    rxSigRay_four = Rayleigh_nominator_a4./Rayleigh_a4;%Rayleigh
fading : output of channel

    %%%% combine

    nominator = conj(Rayleigh_a1) .* Rayleigh_nominator_a1 +
conj(Rayleigh_a2) .* Rayleigh_nominator_a2 + conj(Rayleigh_a3) .*
Rayleigh_nominator_a3 + conj(Rayleigh_a4) .* Rayleigh_nominator_a4;

    denominator = conj(Rayleigh_a1) .* Rayleigh_a1 +
conj(Rayleigh_a2) .* Rayleigh_a2 + conj(Rayleigh_a3) .* Rayleigh_a3 +
conj(Rayleigh_a4) .* Rayleigh_a4;

    equalizer = nominator./denominator;

    % phase est
```

```matlab
        x5 = pskmod([0,M-1],M);
        x5_conj = mean((conj(x5)).^4);
        Rx_5 = mean((equalizer).^4);
        Ph_err_RayEst4 = 0.25*mean(angle(x5_conj*Rx_5));

    Sub_Ray04(i) = abs(Ph_err_RayEst4 - ph_err);
end

figure(1)
plot(numSum,Sub_01,numSum,Sub_02,numSum,Sub_04)
grid on;
legend('1 antenna','2 antenna','4 antenna');
xlabel('numSum');
ylabel('phase error');
axis([100,10000,0,0.005]);
title('AWGN Inphase error for diversity antenna');

figure(2)

plot(numSum,Sub_Ray01,numSum,Sub_Ray02,numSum,Sub_Ray04)
grid on;
legend('1 antenna','2 antenna','4 antenna');
xlabel('numSum');
ylabel('phase error');
axis([100,10000,0,0.05]);
title('Rayleigh Inphase error for diversity antenna');
```