**EXPERIMENT 5:** Write a code to analyze the performance of Quadrature Amplitude Modulation (QAM) and M-ary Phase Shift Keying (PSK) scheme in AWGN channel, and compare the results with theoretical results

**Matlab Code:**

clc;close all;clear all;

% Number of information bits

m= 10^5;

%Range of SNR values

snr\_dB = [0:1:10];

for j=1:1:length(snr\_dB)

n\_err = 0;

n\_bits = 0;

while n\_err < 100

% Generate sequence of binary bits

inf\_bits=round(rand(1,m));

% BPSK Constellation symbols

x=-2\*(inf\_bits-0.5);

% Noise variance

N0=1/10^(snr\_dB(j)/10);

% Send over Gaussian Link to the receiver

y=x + sqrt(N0/2)\*(randn(1,length(x))+i\*randn(1,length(x)));

% Decision making at the Receiver

est\_bits=y < 0;

% Calculate Bit Errors

diff=inf\_bits-est\_bits;

n\_err=n\_err+sum(abs(diff));

n\_bits=n\_bits+length(inf\_bits);

end

% Calculate Bit Error Rate

BER(j)=n\_err/n\_bits;

end

% AWGN Theoretical BER

theoryBerAWGN=0.5\*erfc(sqrt(10.^(snr\_dB/10)));

semilogy(snr\_dB,theoryBerAWGN,'-','LineWidth',2);

hold on;

semilogy(snr\_dB,BER,'or','LineWidth',2);

hold on;

semilogy(snr\_dB,theoryBerAWGN,'blad-','LineWidth',2);

legend('AWGN Simulated', 'AWGN Theoretical');

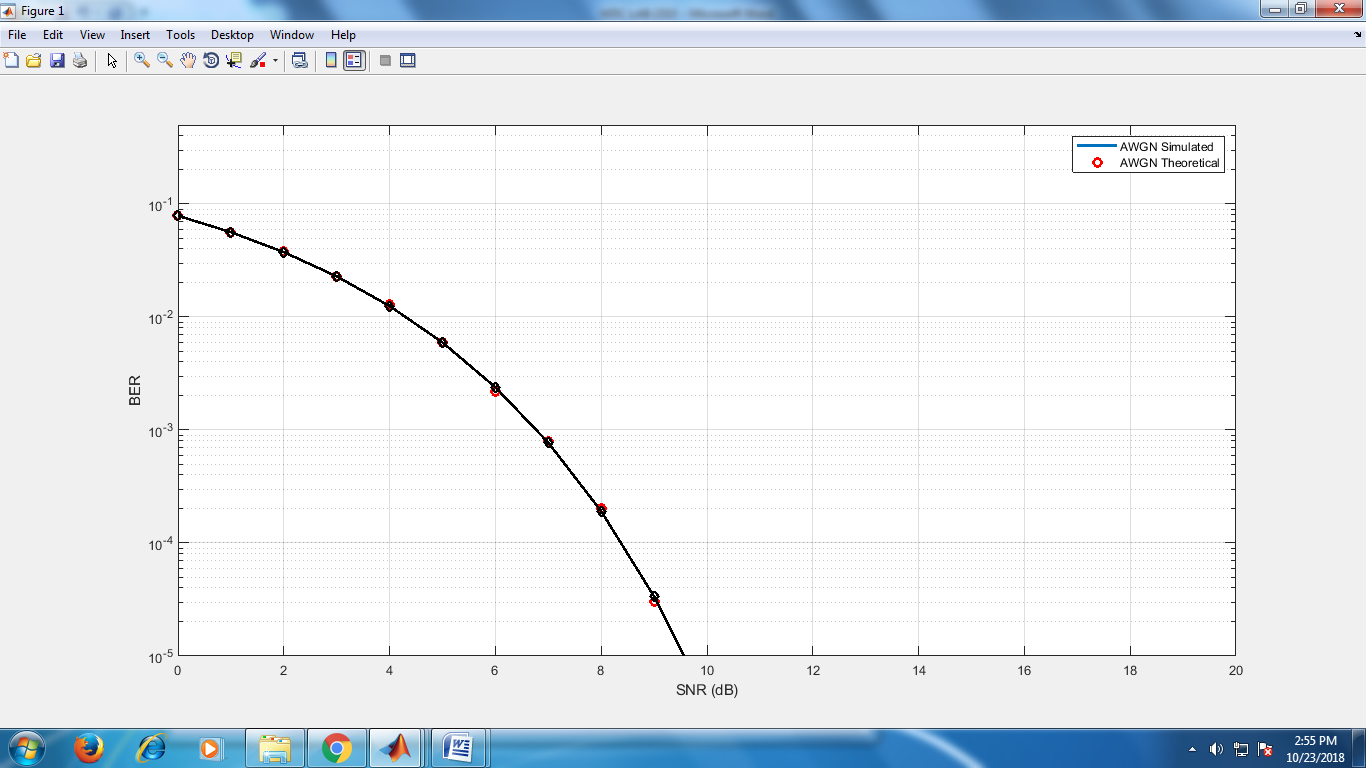
axis([0 20 10^-5 0.5]);

xlabel('SNR (dB)');

ylabel('BER');

grid on;

**RESULT**



**EXPERIMENT 6:** Write a code to compute Bit Error Rate (BER) for different digital modulation scheme in

frequency-flat and slowly varying fading channel.

**Matlab Code:**

clc;close all;clear all;

% Number of information bits

m= 10^5;

%Range of SNR values

snr\_dB = [0:1:20];

for j=1:1:length(snr\_dB)

n\_err = 0;

n\_bits = 0;

while n\_err < 100

% Generate sequence of binary bits

inf\_bits=round(rand(1,m));

% BPSK modulator

x=-2\*(inf\_bits-0.5);

% Noise variance

N0=1/10^(snr\_dB(j)/10);

% Rayleigh channel fading

h=1/sqrt(2)\*[randn(1,length(x)) + i\*randn(1,length(x))];

% Send over Gaussian Link to the receiver

y=h.\*x + sqrt(N0/2)\*(randn(1,length(x))+i\*randn(1,length(x)));

% decision metric

y=y./h;

% Decision making at the Receiver

est\_bits=y < 0;

% Calculate Bit Errors

diff=inf\_bits-est\_bits;

n\_err=n\_err+sum(abs(diff));

n\_bits=n\_bits+length(inf\_bits);

end

% Calculate Bit Error Rate

BER(j)=n\_err/n\_bits;

end

% Rayleigh Theoretical BER

snr = 10.^(snr\_dB/10);

theoryBer=0.5.\*(1-sqrt(snr./(snr+1)));

% AWGN Theoretical BER

theoryBerAWGN=0.5\*erfc(sqrt(10.^(snr\_dB/10)));

semilogy(snr\_dB,theoryBer,'-','LineWidth',2);

hold on;

semilogy(snr\_dB,BER,'or','LineWidth',2);

hold on;

semilogy(snr\_dB,theoryBerAWGN,'blad-','LineWidth',2);

legend('Rayleigh Theoretical','Rayleigh Simulated', 'AWGN Theoretical');

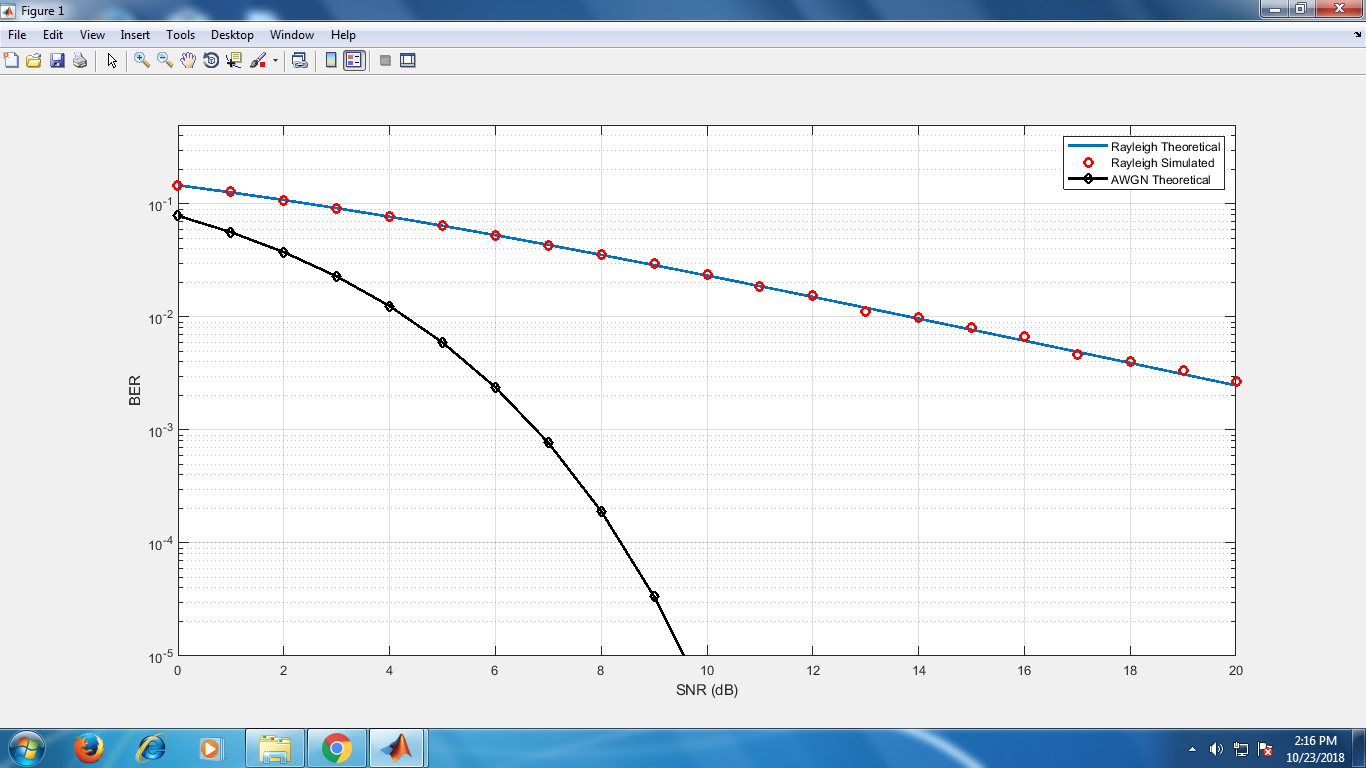
axis([0 20 10^-5 0.5]);

xlabel('SNR (dB)');

ylabel('BER');

grid on;

**RESULT**



**EXPERIMENT 7:** Bit error rate analysis of digital communication receivers with Maximal Ratio Combining (MRC) receive diversity in frequency-flat and slowly varying fading channel.

**Matlab Code:**

% BER performance of BPSK receiver in Rayleigh fading channel, with 2 branch MRC receive

diversity

clc;

clear all;

close all;

%%%%%%%%%%%%%%% Initialization %%%%%%%%%%%%%%%%%%%%%%%%

N=5; % Number of trials

m = 10^6; %Number of bits in each trial

ip = rand(1,m)>0.5; % Generated bits

BPSK = 2\*ip-1; % Generated BPSK symbols

snr\_dB = 0:1:15; % range of snr values

snr = 10.^(snr\_dB/10); % snr value in the normal scale

L=2; % Number of diversity branches

% theoretical BER value for MRC combiner with 2 diversity branches

p\_R\_MRC = 1/2 - 1/2\*(1+1./snr).^(-1/2);

ber\_MRC\_ana = p\_R\_MRC.^2.\*(1+2\*(1-p\_R\_MRC));

%%%%%%%%%% Receive MRC one by Two System %%%%%%%%%%%%%%%%%%%

n\_err=zeros(1,length(snr\_dB)); % Initialize the bit error counter

for p = 1:N

for q = 1:length(snr\_dB)

% Generate white noise samples

No = 1/sqrt(2)\*[randn(L,m) + 1j\*randn(L,m)];

% Generate channel coefficient

h = 1/sqrt(2)\*[randn(L,m) + 1j\*randn(L,m)];

symbol = kron(ones(L,1),BPSK); % array of symbols

rec\_vector = h.\*symbol + 10^(-snr\_dB(q)/20)\*No;% received symbol

% Decision metric

dec\_metric = sum(conj(h).\*rec\_vector,1)./sum(h.\*conj(h),1);

ip\_hat = real(dec\_metric)>0; % Estimated symbol

n\_err(q) = n\_err(q)+size(find([ip- ip\_hat]),2); % compare input and estimated symbols

end

end

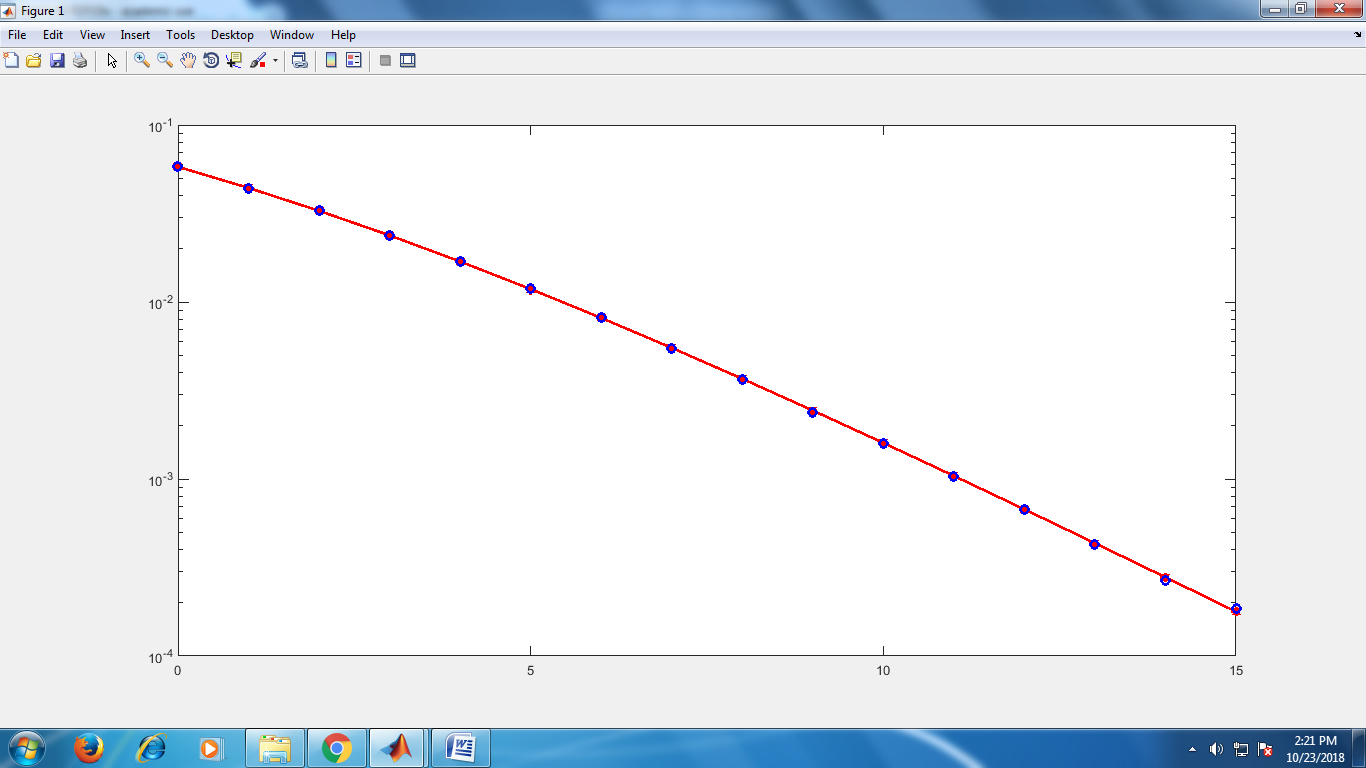
ber\_MRC\_sim = n\_err/(N\*m);

semilogy(snr\_dB,ber\_MRC\_ana,'-r\*','LineWidth',2)

hold on;

semilogy(snr\_dB,ber\_MRC\_sim,'ob','LineWidth',2)

**RESULT**

****

**EXPERIMENT 8:** Bit error rate analysis of digital communication receivers with Equal Gain Combining (EGC) receive diversity in frequency-flat and slowly varying fading channel.

**Matlab Code:**

% BER performance of BPSK receiver in Rayleigh fading channel, with 2 branch EGC receive

diversity

clc;close all;clear all;

% Number of information bits

m= 10^3;

%Range of SNR values

snr\_dB = [0:1:20];

for j=1:1:length(snr\_dB)

n\_err = 0;

n\_bits = 0;

while n\_err < 100

inf\_bits=round(rand(1,m));

% BPSK modulator

x=-2\*(inf\_bits-0.5);

% Noise variance

N0=1/10^(snr\_dB(j)/10);

n1 = sqrt(N0/2)\*abs((randn(1,length(x)) + i\*randn(1,length(x)))); %noise for the first

n2 = sqrt(N0/2)\*abs((randn(1,length(x)) + i\*randn(1,length(x)))); %noise for the first

h1 = sqrt(0.5)\*abs((randn(1,length(x)) + i\*randn(1,length(x)))); %rayleigh amplitude 1

h2 = sqrt(0.5)\*abs((randn(1,length(x)) + i\*randn(1,length(x)))); %rayleigh amplitude 1

%Equal Gain combining

y1 = h1.\*x+n1; % Signal 1

y2 = h2.\*x+n2; % Signal 2

y\_equal = 0.5\*(y1+y2);

% dec\_metric=(norm(y\_equal- h1\*x-h2\*x))^2;

% Decision making at the Receiver

est\_bits=y\_equal < 0;

% Calculate Bit Errors

diff=inf\_bits-est\_bits;

n\_err=n\_err+sum(abs(diff));

n\_bits=n\_bits+length(inf\_bits);

end

% Calculate Bit Error Rate

BER(j)=n\_err/n\_bits;

end

semilogy(snr\_dB,BER,'or-','LineWidth',2);

legend('Rayleigh EGC Simulated', 'Rayleigh Theoretical');

axis([0 20 10^-5 1]);

xlabel('SNR (dB)');

ylabel('BER');grid on;

**RESULT**

****

**EXPERIMENT 8:**To study the performance of the BPSK direct sequence spread spectrum system in AWGN channel.

**Matlab Code:**

clc;close all;clear all;

% Number of information bits

m= 10;

f\_data=1; % DATA FREQUENCY

f\_chip=7; % LENGTH OF CHIP SEQUENCE

fc=210; % RELATIVE CARRIER FREQUENCY

fs=fc\*3; % SAMPLING FREQUENCY

N=fs/f\_chip; % CODING RATE

%Range of SNR values

snr\_dB = [0:2:26];

for jj=1:1:length(snr\_dB)

n\_err = 0;

n\_bits = 0;

while n\_err < 100

% Generate sequence of binary bits

inf\_bits=round(rand(1,m));

PN\_sequence=round(rand(11,1)); % GENERATION OF PN SEQUENCE

%Spread the information bits with PN sequence

j=1;

for i = 1:m

for k = j:j+f\_chip-1

msg\_spread(k)=inf\_bits(i);

end;

msg\_spread(j:(j+f\_chip-1)) = xor(msg\_spread(j:(j+f\_chip-1))',PN\_sequence(1:f\_chip));

j = f\_chip\*i+1;

end;

len\_msg\_spr=length(msg\_spread);

% MODULATING THE SPREAD MESSAGE

% BPSK Constellation symbols

x=-2\*(msg\_spread-0.5);

% Noise variance

N0=1/10^(snr\_dB(jj)/10);

% Send over Gaussian Link to the receiver

y=x + sqrt(N0/2)\*(randn(1,length(x))+i\*randn(1,length(x)));

% Decision making at the Receiver

msg\_demod=y < 0;

% CORRELATING WITH THE PN SEQUENCE

j=1;

for i = 1:m

msg\_demod(j:(j+f\_chip-1)) = xor(msg\_demod(j:(j+f\_chip-1))',PN\_sequence(1:f\_chip));

j = f\_chip\*i+1;

end;

% DESPREADING THE RECEIVED SIGNAL

j=1;

for i = 1:m

s1=0;

for k = j:j+f\_chip-1

s1=s1+msg\_demod(k);

end;

if (s1>=6)

msg\_demod\_rec(i)=1;

else

msg\_demod\_rec(i)=0;

end;

j = f\_chip\*i+1;

end;

% Calculate Bit Errors

diff=inf\_bits-msg\_demod\_rec;

n\_err=n\_err+sum(abs(diff));

n\_bits=n\_bits+length(inf\_bits);

end

% Calculate Bit Error Rate

BER(jj)=n\_err/n\_bits

end

semilogy(snr\_dB,BER,'or-','LineWidth',2);

legend('AWGN Simulated');

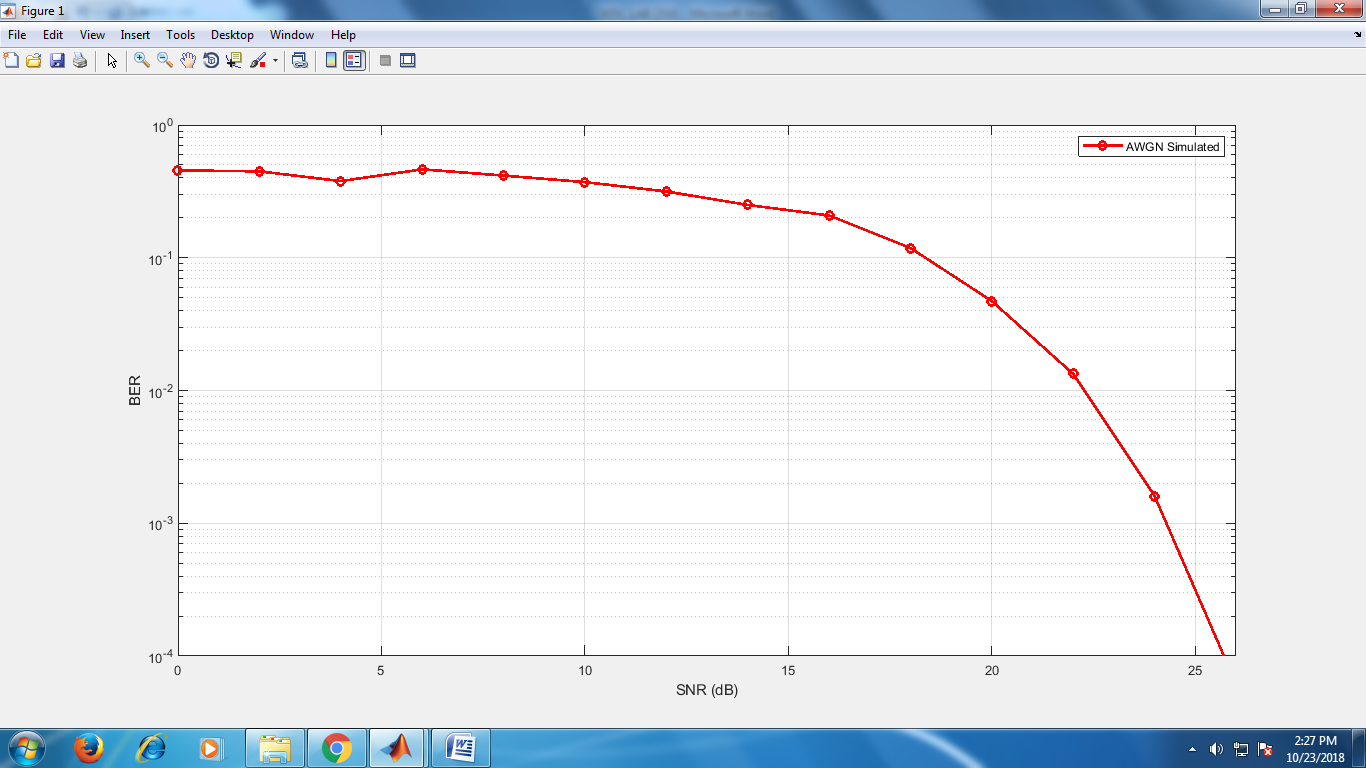
axis([0 26 10^-4 1]);

xlabel('SNR (dB)');

ylabel('BER');

grid on;

**RESULT**

****