EEEE-802 - WIRELESS COMMUNICATIONS

Instructor: Dr. Gill R Tsouri

Rahul Gulia

Course Project – Phase I

- Use Matlab to perform the following assignment. Do not use Simulink.
- All simulations are for QPSK in slow flat fading channels.
- Submit all Matlab codes and printed figures.
- For Phase I of the project you may work and submit in pairs, Phase II is individual.

Rayleigh fading channel

1. Find (don't derive) theoretical expressions for the BER of QPSK over AWGN and Rayleigh channels.

Sol: Theoretical expressions for the BER of QPSK over AWGN

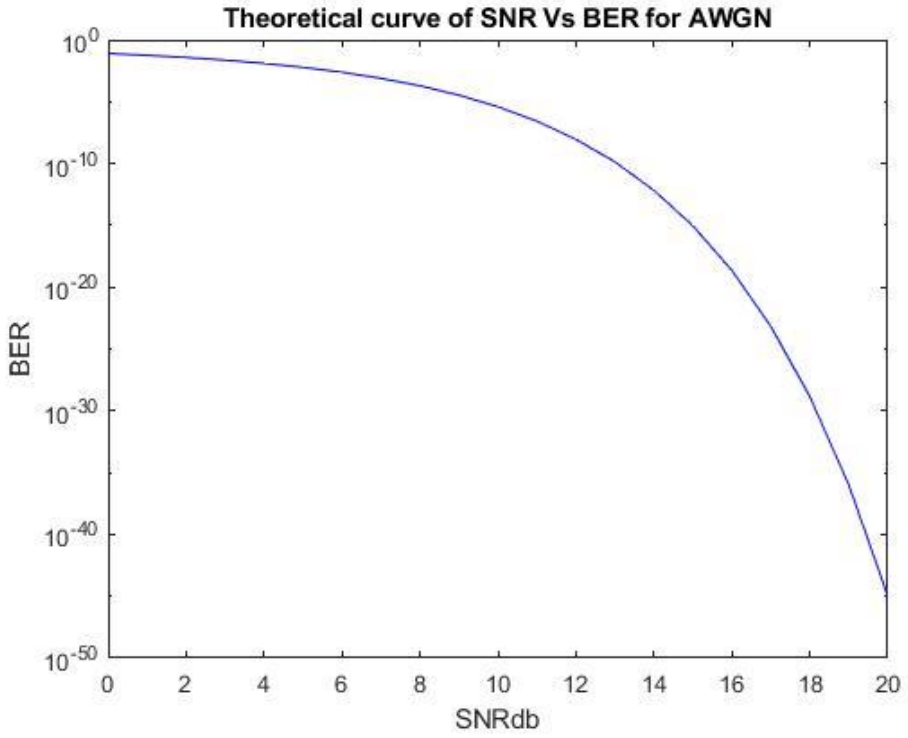
$$\text{BER} = \frac{1}{2}.\text{erfc}(\sqrt{SNR})$$

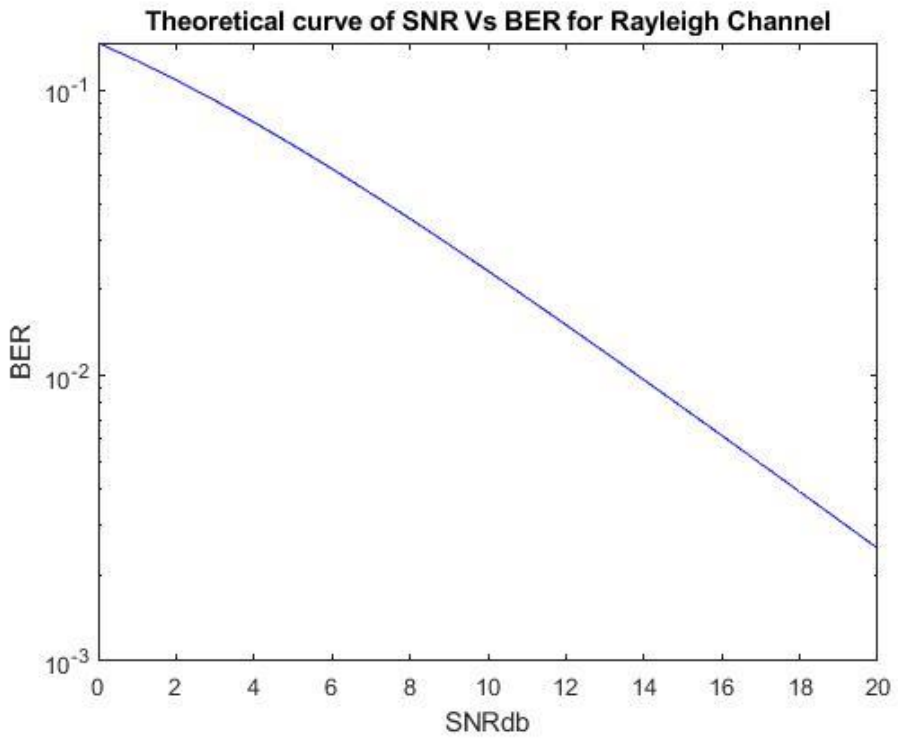
Theoretical expressions for the BER of QPSK over AWGN

$$\text{BER} = \frac{1}{2}.(1 - \sqrt{\frac{SNR}{1+SNR}})$$

2. Draw the curves BER (SNRdB); SNRdB=0, 1, 2, … 20dB for AWGN and Rayleigh channels based on their theoretical expressions.

Sol:





3. Write and run a simulation to match the theoretical curves: simulate BER(SNRdB) ; SNRdB=0,1,2,…,20dB and draw the curves for AWGN and Rayleigh channels.

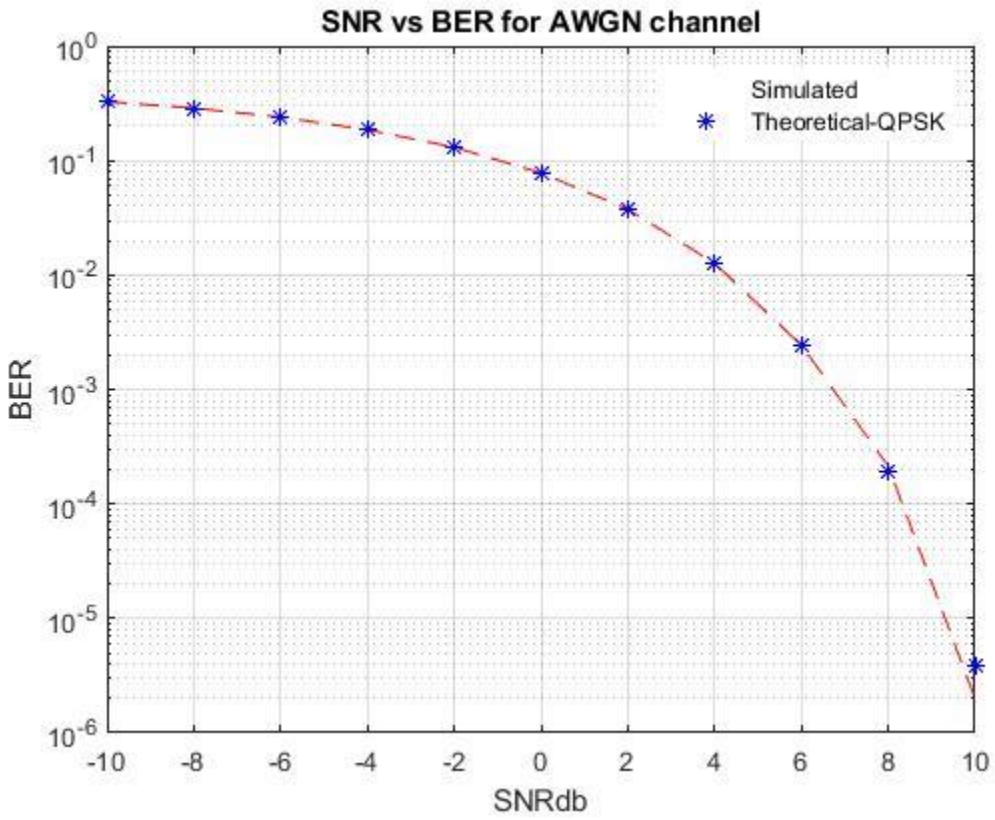
Sol:

Code:

```
% Final code
% Simulation of AWGN channel
clc;
clear all;
close all;
warning off;

% Parameters
nobpc = 10^6;
Es = 1;
SNRdb = -10:2:10;
M = 4;
Rm = log2(M);
```

```
% Transmitting data generation
Tx_bits = round(rand(1,nobpc));
oddData = Tx_bits(1:2:end);
evenData = Tx_bits(2:2:end);
Tx_symb = sqrt(1/2)*(1i*(2*oddData-1)+(2*evenData-1));
%QPSK Mapping
figure, polar(angle(Tx_symb),abs(Tx_symb),'*');

BER = zeros(1,length(SNRdb));
index = 1;

for i =SNRdb
    SNR = 10.^(i/10);
    noiseSigma = sqrt(1./(2*Rm*SNR));
    errors = 0;

    %Creating a complex noise for adding with QPSK
modulated signal
    noise =
noiseSigma*(randn(1,length(Tx_symb))+1i*randn(1,length(
Tx_symb)));

    % Transmitted signal
    Rx_symb = Tx_symb + noise;

    %ML Detector
    detected_real = real(Rx_symb)>=0;
    detected_img = imag(Rx_symb)>=0;

Rx_bits=reshape([detected_img;detected_real],1,nobpc);

    %Bit Error rate Calculation
    BER(index) = sum(xor(Tx_bits,Rx_bits))/nobpc;
    index=index+1;
end

% Theoretical BER
theoreticalBER =0.5*erfc(sqrt(10.^(SNRdb/10)));

figure,semilogy(SNRdb,BER,'r--'),title('SNR vs BER for
AWGN channel');
grid on;
```

```
hold on;
semilogy(SNRdb,theoreticalBER,'b*');
xlabel("SNRdb");
ylabel("BER");
legend('Simulated','Theoretical-QPSK','Theoretical-
QPSK');
grid on;

figure, semilogy(SNRdb,theoreticalBER,'b'),
title("Theoretical curve of SNR Vs BER for AWGN"),
xlabel("SNRdb"),ylabel("BER");
```

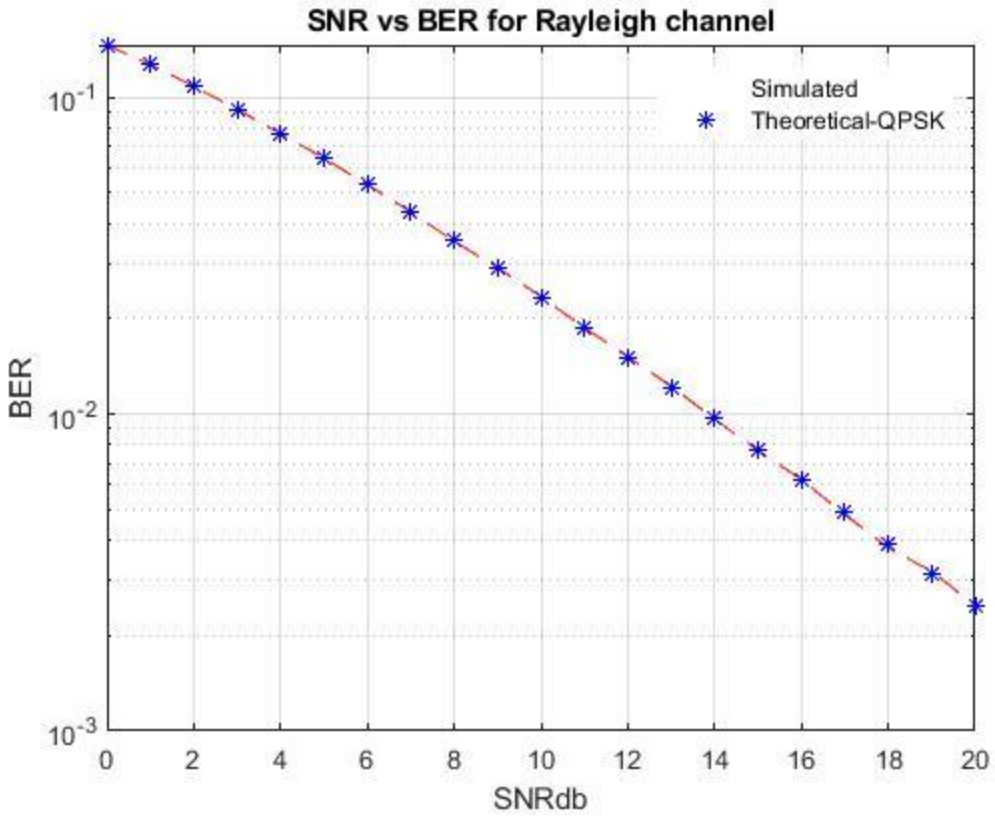


Code:

```
% Final code
% Simulation of Rayleigh channel
clc;
```

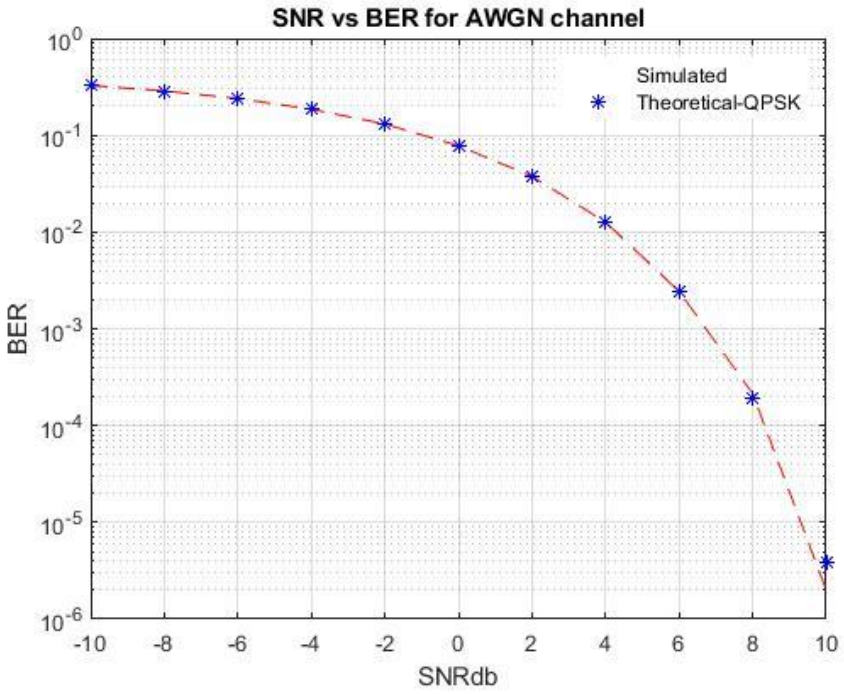
```
clear all;
close all;
warning off;

% Parameters
nobpc = 10^6;
Es = 1;
SNRdb = 0:20;
M = 4;
Rm = log2(M);

% Transmitting data generation
Tx_bits = round(rand(1,nobpc));
oddData = Tx_bits(1:2:end);
evenData = Tx_bits(2:2:end);
Tx_symb = sqrt(1/2)*(1i*(2*oddData-1)+(2*evenData-1));
%QPSK Mapping
figure, polar(angle(Tx_symb),abs(Tx_symb),'*');

BER = zeros(1,length(SNRdb));
index = 1;

for i =SNRdb
    SNR = 10.^(i/10);
    noiseSigma = sqrt(1./(2*Rm*SNR));
    errors = 0;

    %Creating a complex noise for adding with QPSK
modulated signal
    noise =
noiseSigma*(randn(1,length(Tx_symb))+1i*randn(1,length(
Tx_symb)));

    % Creating a Rayleigh channel with variance 0.5
    h =
sqrt(0.5*((randn(1,length(Tx_symb))).^2+(randn(1,length
(Tx_symb))).^2));

    % Transmitted signal
    Rx_symb = Tx_symb.*h + noise;

    % Equalizer
```

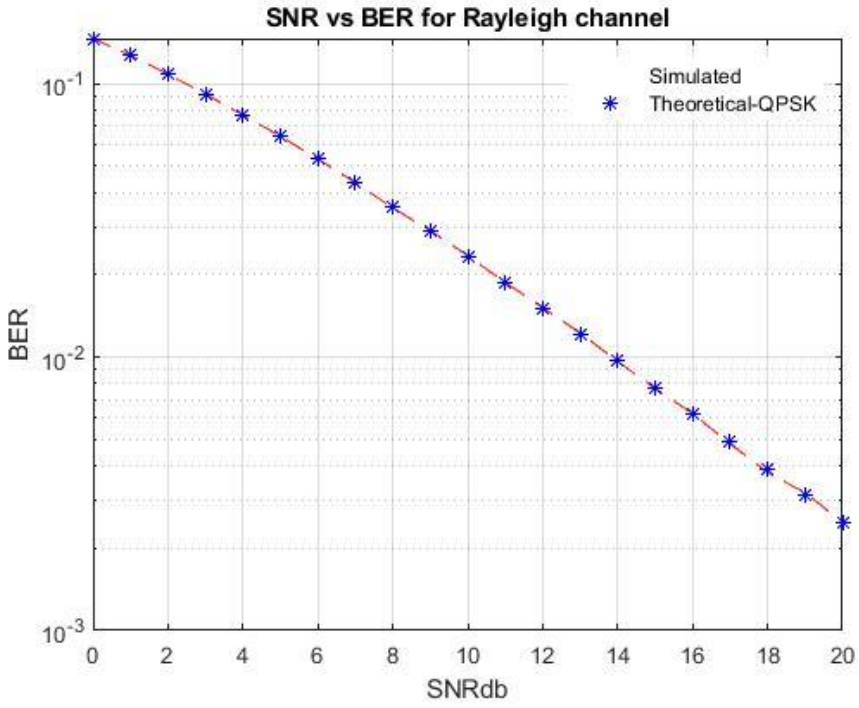
```
    Rx_symb = Rx_symb./h;

    %ML Detector
    detected_real = real(Rx_symb)>=0;
    detected_img = imag(Rx_symb)>=0;

Rx_bits=reshape([detected_img;detected_real],1,nobpc);

    %Bit Error rate Calculation
    BER(index) = sum(xor(Tx_bits,Rx_bits))/nobpc;
    index=index+1;
end

% Theoretical BER for Rayleigh channel
SNR1 = 10.^(SNRdb/10);
theoreticalBER =0.5*(1 - sqrt(SNR1./(SNR1+1)));

figure,semilogy(SNRdb,BER,'r--'),title('SNR vs BER for
Rayleigh channel');
grid on;
hold on;
semilogy(SNRdb,theoreticalBER,'b*');
xlabel("SNRdb");
ylabel("BER");
legend('Simulated','Theoretical-QPSK');
grid on;

figure, semilogy(SNRdb,theoreticalBER,'b'),
title("Theoretical curve of SNR Vs BER for Rayleigh
Channel"), xlabel("SNRdb"),ylabel("BER");
```



4. Print and attach a single Figure presenting theoretical and simulation curves superimposed one on top of the other for AWGN and Rayleigh channels.

Sol:





Rician fading channel

Read the paper posted on my-courses:

C. Xiao, Y.R. Zheng, N.C. Beaulieu, "Novel Sum-of-Sinusoids Simulation Models for Rayleigh and Rician Fading Channels," IEEE Transactions on Wireless Communications, vol. 5, pp. 3667-3679, Dec. 2006.

5. Write a code to implement the "Novel Rician Fading Simulator" presented in the paper.

Sol:

Code:

```
% Rician channel
% Final Code
clc;
clear all;
close all;

% parameters
N = 100;
K0 = 0;
K1 = 1;
K2 = 3;
theta0 = pi/4;

phin = 2*pi*rand(1,N)-pi;   % angle of incidence (w.r.t the x-aaxis)
thetan = 2*pi*rand(1,N)-pi;     % angle of arrival

n = 1 : length(N);
alphan = (2*pi*n + thetan)/N;

t = 1 :0.05 : 1000;
W = 2*pi.*t;

Yc = zeros(1,length(W));
Ys = zeros(1,length(W));

for p = 1 : N
   Yc = Yc + cos(W*cos(alphan(p)+phin(p)));
```

```
   Ys = Ys + sin(W*cos(alphan(p)+phin(p)));
end

Yc = Yc/sqrt(N);
Ys = Ys/sqrt(N);

phio = 2*pi*rand(1)-pi;

% -------- for K0  ----------------------------------------------
Zc0 = (Yc + sqrt(K0)*cos(W*cos(theta0)+phio))/sqrt(1+K0);
Zs0 = (Ys + sqrt(K0)*sin(W*cos(theta0)+phio))/sqrt(1+K0);

Z0 = Zc0 + 1j*Zs0;


% Simulation - Autocorrelation of the complex envelope Z(t) --> Eq. (12c)
bess = besselj(0,W);
Rzz0 = (bess + K0*cos(W*cos(theta0)) + 1i*K0*sin(W*cos(theta0)))/(1+K0);
% Rzz0abs = sum(abs(Rzz0(1:200)).^2);
Rzz0 = Rzz0/Rzz0(3);
Rzz0 = Rzz0(1:250);

% Theoretical - Autocorrelation of the complex envelope Z(t) --> Eq. (12c)
Rz0 = xcorr(Z0,Z0);
Rz0 = Rz0(ceil((length(Rz0))/2):end);
% Rz0abs = sum(abs(Rz0).^2);
Rz0 = Rz0/Rz0(7);
Rz0 = Rz0(1:250);


% -------- for K1    ----------------------------------------------
Zc1 = (Yc + sqrt(K1)*cos(W*cos(theta0)+phio))/sqrt(1+K1);
Zs1 = (Ys + sqrt(K1)*sin(W*cos(theta0)+phio))/sqrt(1+K1);

Z1 = Zc1 + 1j*Zs1;

% Simulation - Autocorrelation of the complex envelope Z(t) --> Eq. (12c)
```

```
J1 = besselj(0,W);
Rzz1 = (J1 + K1*cos(W*cos(theta0)) + 1j*K1*sin(W*cos(theta0)))/(1+K1);
Rzz1 = Rzz1/Rzz1(8);
Rzz1 = Rzz1(1:250);

% Theoretical - Autocorrelation of the complex envelope Z(t) --> Eq. (12c)
Rz1 = xcorr(Z1,Z1);
Rz1 = Rz1(ceil((length(Rz1))/2):end);
Rz1 = Rz1/Rz1(8);
Rz1 = Rz1(1:250);


% -------- for K2  ----------------------------------------------------
Zc2 = (Yc + sqrt(K2)*cos(W*cos(theta0)+phio))/sqrt(1+K2);
Zs2 = (Ys + sqrt(K2)*sin(W*cos(theta0)+phio))/sqrt(1+K2);

Z2 = Zc2 + 1j*Zs2;

% Simulation - Autocorrelation of the complex envelope Z(t) --> Eq. (12c)
J2 = besselj(0,W);
Rzz2 = (J1 + K2*cos(W*cos(theta0)) + 1j*K2*sin(W*cos(theta0)))/(1+K2);
Rzz2 = Rzz2/Rzz2(4);
Rzz2 = Rzz2(1:250);

% Theoretical - Autocorrelation of the complex envelope Z(t) --> Eq. (12c)
Rz2 = xcorr(Z2,Z2);
Rz2 = Rz2(ceil((length(Rz2))/2):end);
Rz2 = Rz2/(Rz2(5));
Rz2 = Rz2(1:250);

% -------- Plotting - Real part of the Autocorrelation of the complex
envelope  ------------
figure(1),plot(Rzz0,'r'),title("Real part of the autocorrelation of the complex
envelope for K=0");
hold on
plot(real(Rz0),'b--');
xlabel("Normalized time");
```

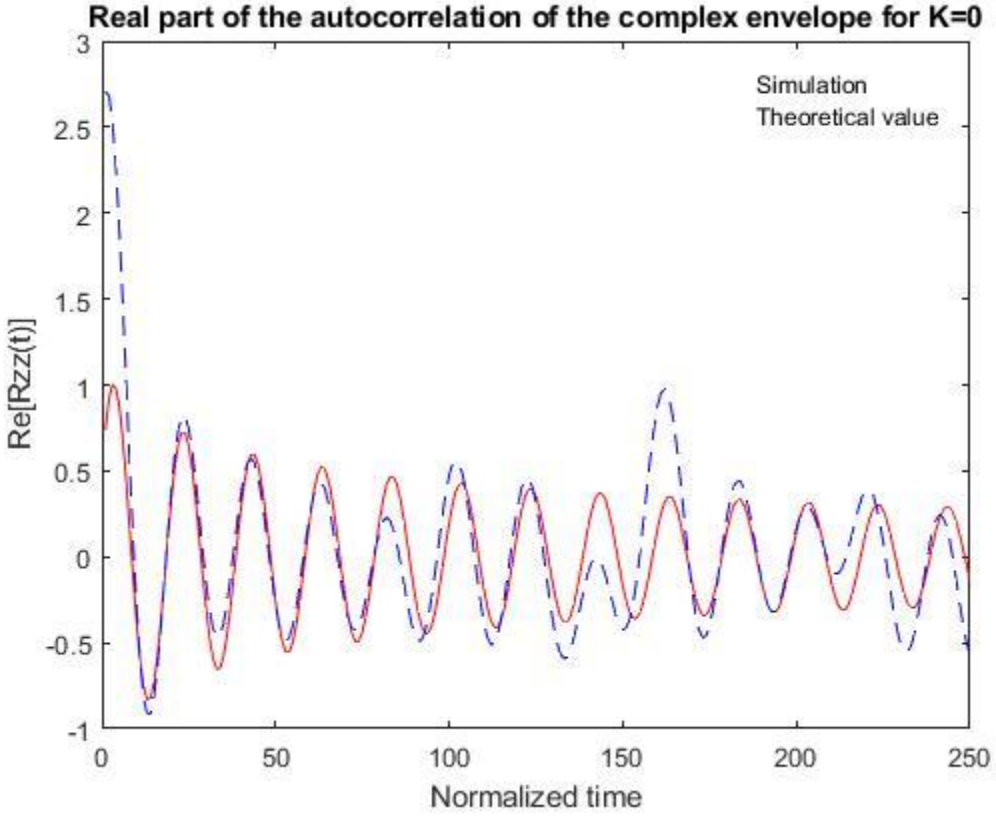
```
ylabel("Re[Rzz(t)]");
legend('Simulation','Theoretical value');
hold off

figure(2),plot(real(Rzz1),'r'),title("Real part of the autocorrelation of the
complex envelope for K=1");
hold on
plot(real(Rz1),'b--');
xlabel("Normalized time");
ylabel("Re[Rzz(t)]");
legend('Simulation','Theoretical value');
hold off

figure(3),plot(real(Rzz2),'r'),title("Real part of the autocorrelation of the
complex envelope for K=3");
hold on
plot(real(Rz2),'b--');
xlabel("Normalized time");
ylabel("Re[Rzz(t)]");
legend('Simulation','Theoretical value');
hold off

% -------- Plotting - Imaginary part of the Autocorrelation of the complex
envelope  ------------
figure(4),plot(Rzz0,'r'),title("Imaginary part of the autocorrelation of the
complex envelope for K=0");
hold on
plot(imag(Rz0),'b--');
xlabel("Normalized time");
ylabel("Re[Rzz(t)]");
legend('Simulation','Theoretical value');
hold off

figure(5),plot(imag(Rzz1),'r'),title("Imaginary part of the autocorrelation of
the complex envelope for K=1");
hold on
plot(imag(Rz1),'b--');
```

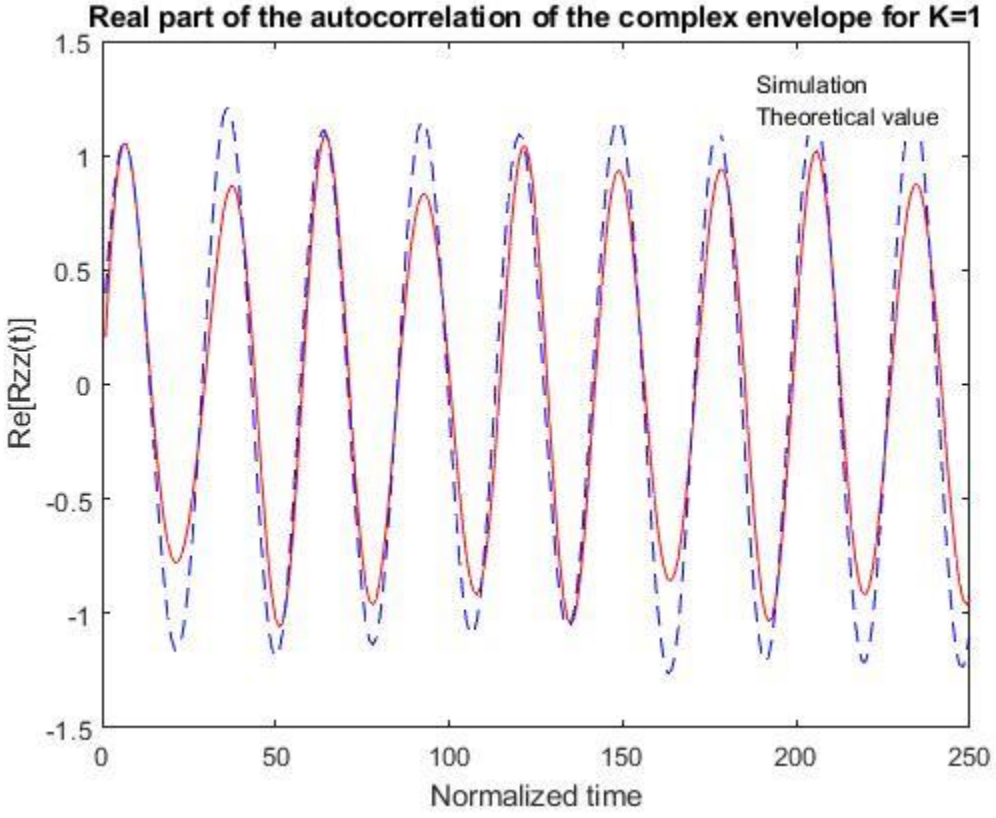
```
xlabel("Normalized time");
ylabel("Re[Rzz(t)]");
legend('Simulation','Theoretical value');
hold off

figure(6),plot(imag(Rzz2),'r'),title("Imaginary part of the autocorrelation of
the complex envelope for K=3");
hold on
plot(imag(Rz2),'b--');
xlabel("Normalized time");
ylabel("Re[Rzz(t)]");
legend('Simulation','Theoretical value');
hold off

% ------------------------------------------
% Autocorrelation of the squared envelope

% Simulated Values
Z0_Th = (abs(Z0)).^2;
Z1_Th = (abs(Z1)).^2;
Z2_Th = (abs(Z2)).^2;

% Theoretical - Autocorrelation of the squared envelope Z(t) --> Eq. (12d)
Rz0_Th = xcorr(Z0_Th,Z0_Th);
Rz0_Th = Rz0_Th(ceil((length(Rz0_Th))/2):end);
Rz0_Th = Rz0_Th(1:250)/(Rz0_Th(1));

Rz1_Th = xcorr(Z1_Th,Z1_Th);
Rz1_Th = Rz1_Th(ceil((length(Rz1_Th))/2):end);
Rz1_Th = Rz1_Th(1:250)/(Rz1_Th(1));

Rz2_Th = xcorr(Z2_Th,Z2_Th);
Rz2_Th = Rz2_Th(ceil((length(Rz2_Th))/2):end);
Rz2_Th = Rz2_Th(1:250)/(Rz2_Th(1));

% Plotting Squared envelope autocorrelation
figure(7), plot(Rz0_Th,'b'),title("Squared envelope autocorrelation");
```

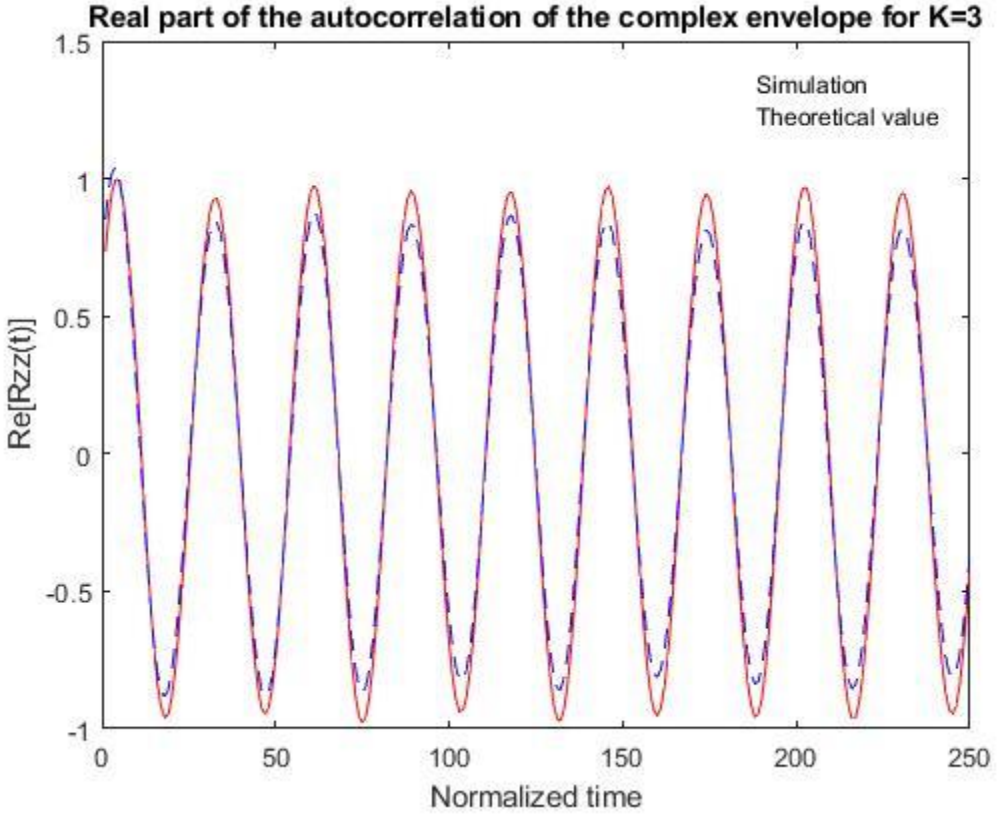
```
hold on
plot(Rz1_Th,'r');
plot(Rz2_Th,'g');
xlabel("Normalized time");
ylabel("Normalized R|z||z|(t)]");
legend('K=0','K=1','K=3');
plot(Rz2_Th)


% PDF of the fading envelope
Z0_pdf = (abs(Z0)).^2;
Z1_pdf = (abs(Z1)).^2;
Z2_pdf = (abs(Z2)).^2;

figure(8),histfit(Z0_pdf),title("PDF of the fading envelope");
hold on
histfit(Z1_pdf)
xlabel("z");
ylabel("fz(z)");
histfit(Z2_pdf)

figure(9), histfit(angle(Z0)),title("PDF of the fading phase");
hold on;
histfit(angle(Z1));
xlabel("fading phase");
ylabel("pdf of fading phase");
histfit(angle(Z2));
```

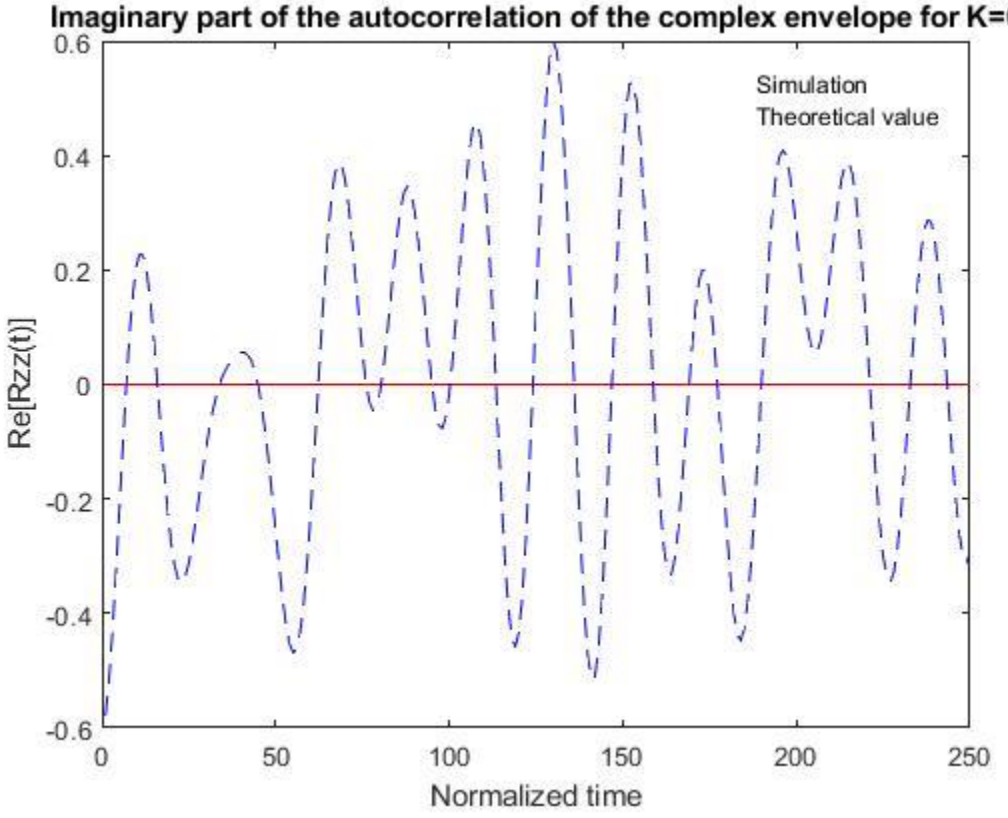
6. Recreate, print and attach Figures 3-7 in the paper.

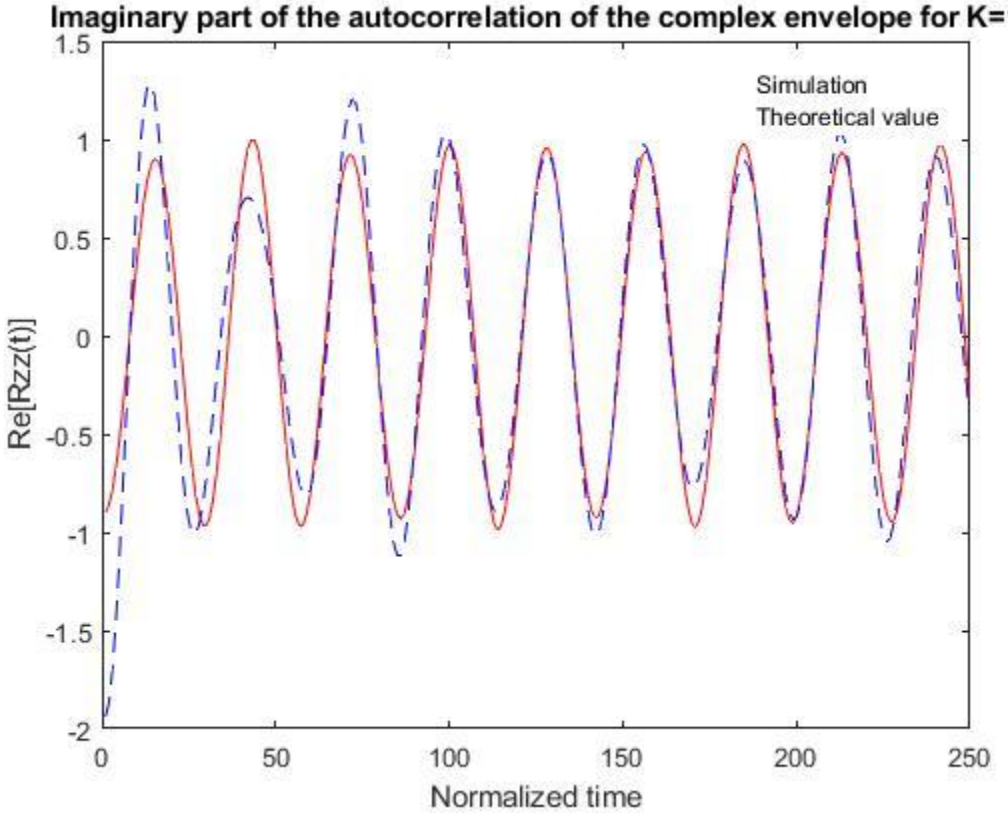
Sol:

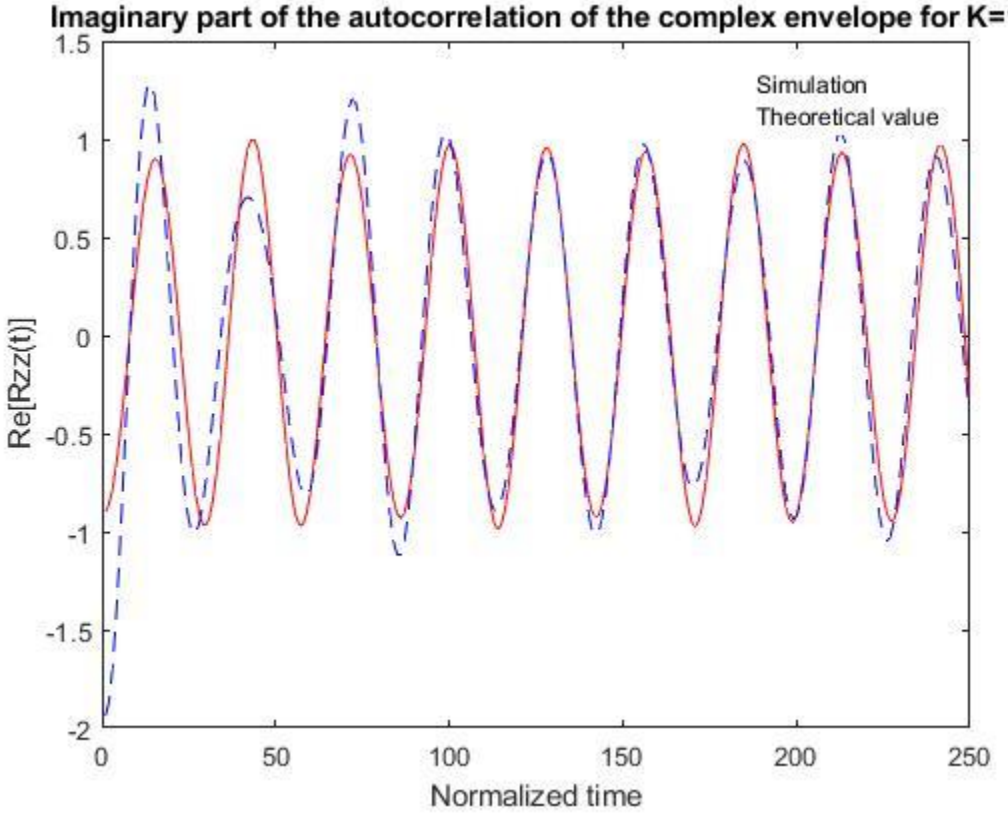

Real part of the autocorrelation of the complex envelope for K=0
Simulation
Theoretical value
Re[Rzz(t)]
Normalized time
3
2.5
2
1.5
1
0.5
0
-0.5
-1
0
50
100
150
200
250

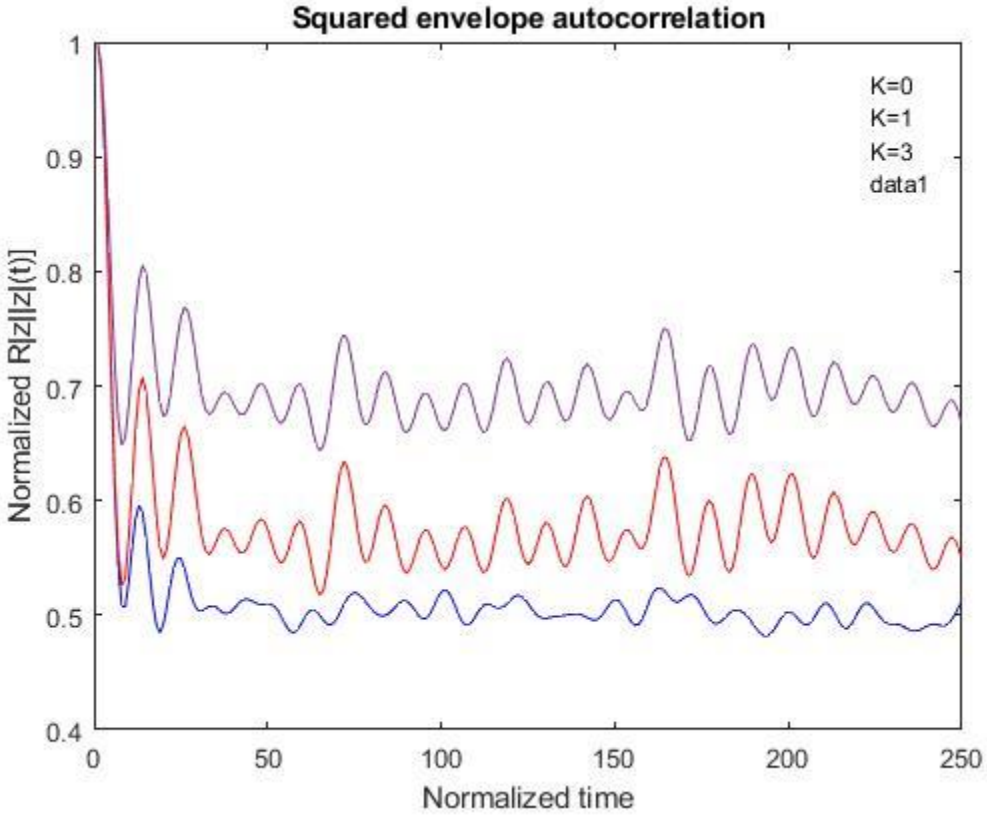

Real part of the autocorrelation of the complex envelope for K=1
Simulation
Theoretical value
Re[Rzz(t)]
Normalized time
1.5
1
0.5
0
-0.5
-1
-1.5
0
50
100
150
200
250

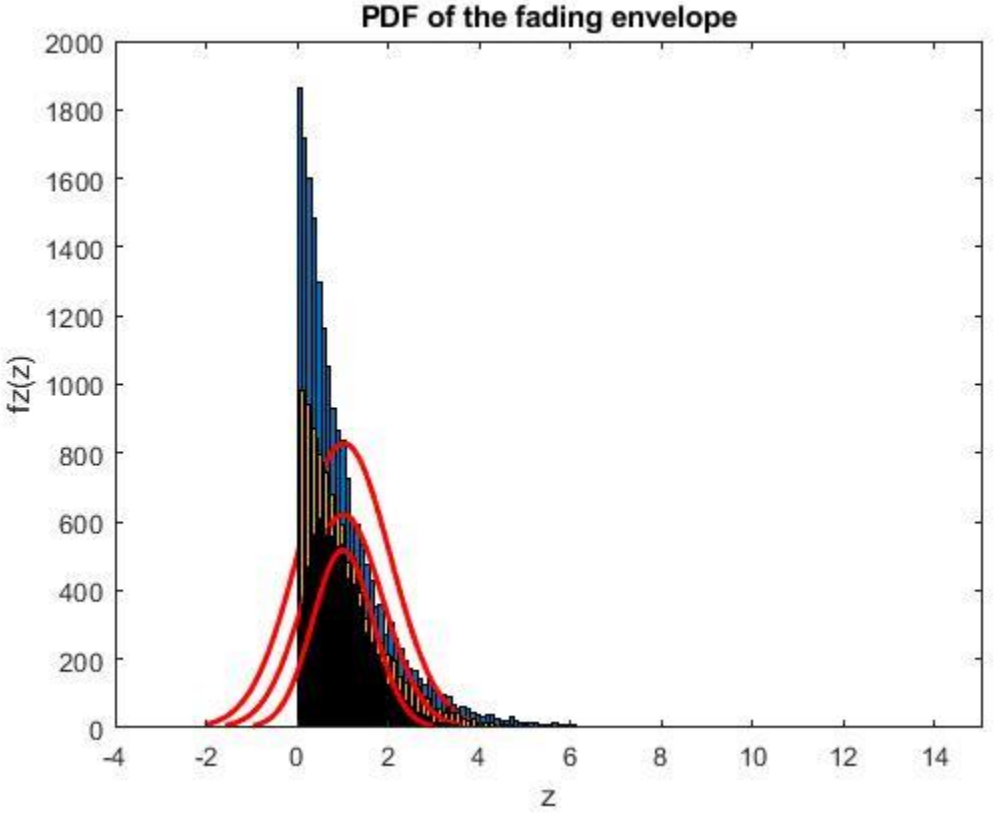

Real part of the autocorrelation of the complex envelope for K=3
Simulation
Theoretical value
Re[Rzz(t)]
Normalized time
1.5
1
0.5
0
-0.5
-1
0
50
100
150
200
250

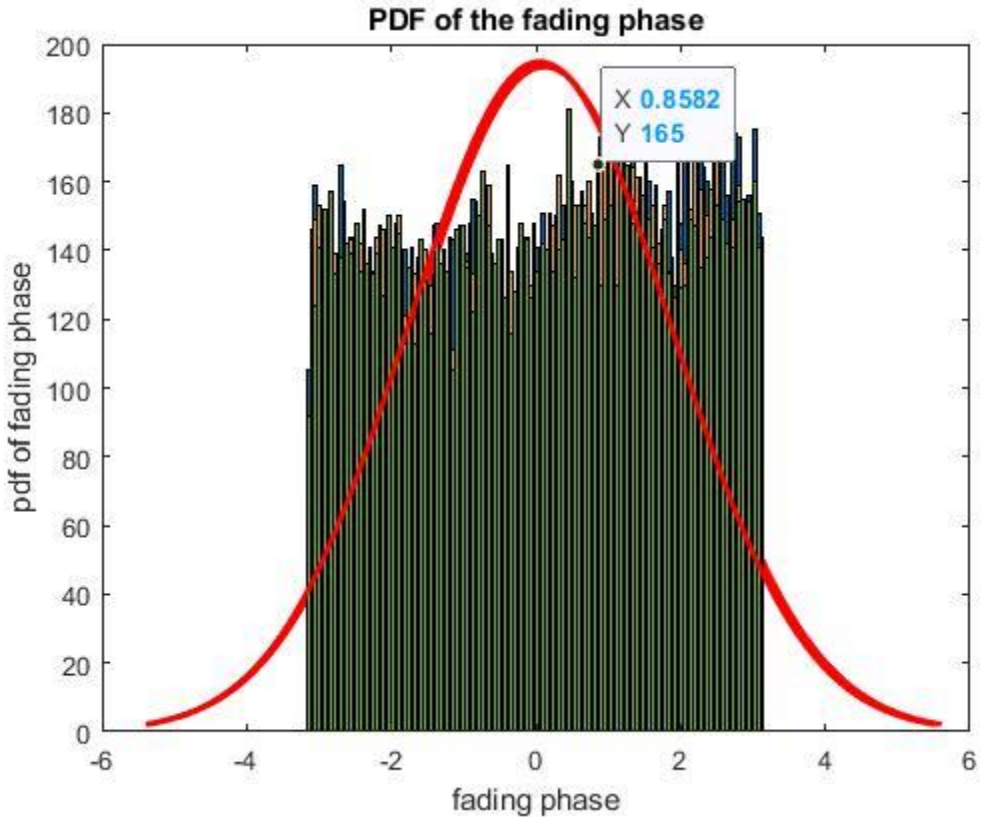

Imaginary part of the autocorrelation of the complex envelope for K=0
Simulation
Theoretical value
Re[Rzz(t)]
Normalized time
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
0
50
100
150
200
250


Imaginary part of the autocorrelation of the complex envelope for K=1
Simulation
Theoretical value
Re[Rzz(t)]
Normalized time
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
50
100
150
200
250


Imaginary part of the autocorrelation of the complex envelope for K=1
Simulation
Theoretical value
Re[Rzz(t)]
Normalized time
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
50
100
150
200
250


Squared envelope autocorrelation
K=0
K=1
K=3
data1
Normalized R|z||z|(t)]
Normalized time
1
0.9
0.8
0.7
0.6
0.5
0.4
0
50
100
150
200
250


PDF of the fading envelope
fz(z)
z
2000
1800
1600
1400
1200
1000
800
600
400
200
0
-4
-2
0
2
4
6
8
10
12
14



Performance in Rician Fading:

7. Find theoretical expressions for the BER of QPSK over Rician channels.

Sol:

Theoretical BER for Rician channel for antipodal signals.

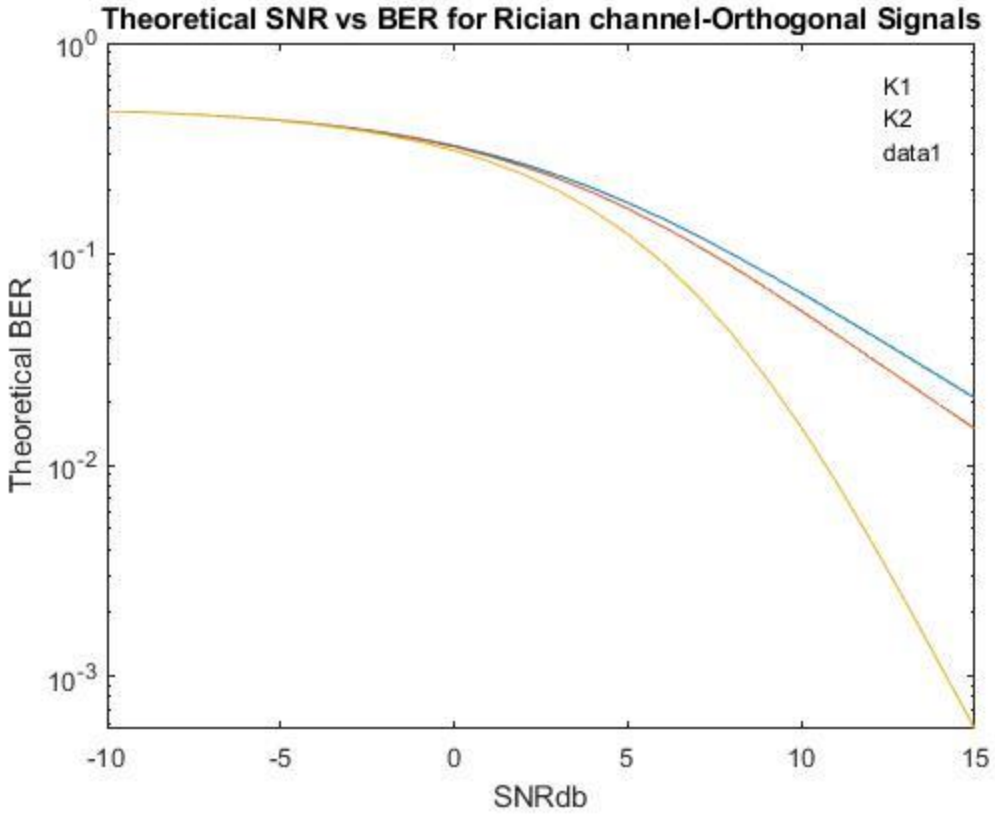
$$\text{BER} = \frac{1+K}{2(1+K+SNR)} \cdot \exp\left(\frac{-K.SNR}{1+K+SNR}\right)$$

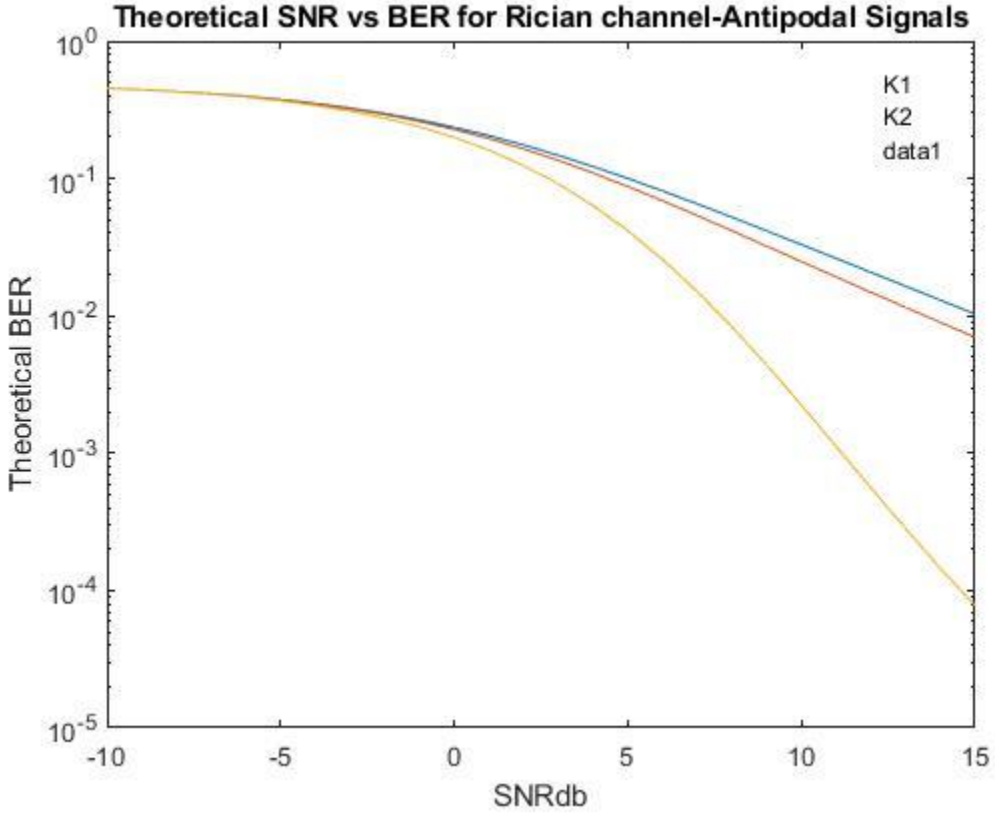
Theoretical BER for Rician channel for orthogonal signals.

$$\text{BER} = \frac{1+K}{2+2K+SNR)} \cdot \exp\left(\frac{-K.SNR}{2+2K+SNR}\right)$$

8. Draw the curves BER(SNRdB) ; SNRdB=0,1,2,…,20dB for Rician factors of K=0,1,3,10 based on the theoretical expressions.

Sol:


Theoretical SNR vs BER for Rician channel-Orthogonal Signals
Theoretical BER
SNRdb
K1
K2
data1
10^0
10^-1
10^-2
10^-3
-10
-5
0
5
10
15


Theoretical SNR vs BER for Rician channel-Antipodal Signals
Theoretical BER
SNRdb
K1
K2
data1
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
-10
-5
0
5
10
15

9. Write and run a simulation for validating the theoretical curves: simulate BER(SNRdB) ; SNRdB=0,1,2,…,20dB and draw curves for K=1,3,10.

Sol:

Code:

```
% Final code
% Simulation of Rayleigh channel
clc;
clear all;
close all;
warning off;

% Parameters
nobpc = 10^6;
Es = 1;
SNRdb = -10:1:15;
M = 4;
Rm = log2(M);
K1 = 1;
K1 = 10^(K1/10);
K2 = 3;
K2 = 10^(K2/10);
K3 = 10;
K3 = 10^(K3/10);

% Transmitting data generation
Tx_bits = round(rand(1,nobpc));
oddData = Tx_bits(1:2:end);
evenData = Tx_bits(2:2:end);
Tx_symb = sqrt(1/2)*(1i*(2*oddData-1)+(2*evenData-1));
%QPSK Mapping

% Rician channel generation ----------------------------
------------
% Parameters
N = 10;
t = 1:10^6;
W = 2*pi*t;
```

```
theta0 = pi/4;
thetan = 2*pi*rand(1,N)-pi;
phin = 2*pi*rand(1,N)-pi;
n = 1 : length(N);
alphan = (2*pi*n + thetan)/N;

Yc = zeros(1,length(W));
Ys = zeros(1,length(W));

for q = 1:N
        Yc = Yc + cos(W*cos(alphan(q)+phin(q)));
        Ys = Ys + sin(W*cos(alphan(q)+phin(q)));
end

Yc = Yc/sqrt(N);
Ys = Ys/sqrt(N);

phio = 2*pi*rand(1)-pi;

% -------- for K0, K1 and K2  -------------------------
----------------------
Zc1 = (Yc +
sqrt(K1)*cos(W*cos(theta0)+phio))/sqrt(1+K1);
Zs1 = (Ys +
sqrt(K1)*sin(W*cos(theta0)+phio))/sqrt(1+K1);

Z1 = Zc1 + 1j*Zs1;
Z1 = Z1(1:length(Tx_symb));

Zc2 = (Yc +
sqrt(K2)*cos(W*cos(theta0)+phio))/sqrt(1+K2);
Zs2 = (Ys +
sqrt(K2)*sin(W*cos(theta0)+phio))/sqrt(1+K2);

Z2 = Zc2 + 1j*Zs2;
Z2 = Z2(1:length(Tx_symb));

Zc3 = (Yc +
sqrt(K1)*cos(W*cos(theta0)+phio))/sqrt(1+K3);
```

```
Zs3 = (Ys +
sqrt(K1)*sin(W*cos(theta0)+phio))/sqrt(1+K3);

Z3 = Zc3 + 1j*Zs3;
Z3 = Z3(1:length(Tx_symb));

BER1 = zeros(1,length(SNRdb));
index = 1;
for i =SNRdb
    SNR = 10.^(i/10);
    noiseSigma = sqrt(1./(2*Rm*SNR));
    errors = 0;

    %Creating a complex noise for adding with QPSK
modulated signal
    noise =
noiseSigma*(randn(1,length(Tx_symb))+1i*randn(1,length(
Tx_symb)));

    % Transmitted signal
    Rx_symb = Tx_symb.*Z1 + noise;

    % Equalizer
    Rx_symb = Rx_symb./Z1;

    %ML Detector
    detected_real = real(Rx_symb)>=0;
    detected_img = imag(Rx_symb)>=0;

Rx_bits=reshape([detected_img;detected_real],1,nobpc);

    %Bit Error rate Calculation
    BER1(index) = sum(xor(Tx_bits,Rx_bits))/nobpc;
    index=index+1;
end

BER2 = zeros(1,length(SNRdb));
index = 1;
for i =SNRdb
    SNR = 10.^(i/10);
```

```
    noiseSigma = sqrt(1./(2*Rm*SNR));
    errors = 0;

    %Creating a complex noise for adding with QPSK
modulated signal
    noise =
noiseSigma*(randn(1,length(Tx_symb))+1i*randn(1,length(
Tx_symb)));

    % Transmitted signal
    Rx_symb = Tx_symb.*Z2 + noise;

    % Equalizer
    Rx_symb = Rx_symb./Z2;

    %ML Detector
    detected_real = real(Rx_symb)>=0;
    detected_img = imag(Rx_symb)>=0;

Rx_bits=reshape([detected_img;detected_real],1,nobpc);

    %Bit Error rate Calculation
    BER2(index) = sum(xor(Tx_bits,Rx_bits))/nobpc;
    index=index+1;
end

BER3 = zeros(1,length(SNRdb));
index = 1;
for i =SNRdb
    SNR = 10.^(i/10);
    noiseSigma = sqrt(1./(2*Rm*SNR));
    errors = 0;

    %Creating a complex noise for adding with QPSK
modulated signal
    noise =
noiseSigma*(randn(1,length(Tx_symb))+1i*randn(1,length(
Tx_symb)));

    % Transmitted signal
```

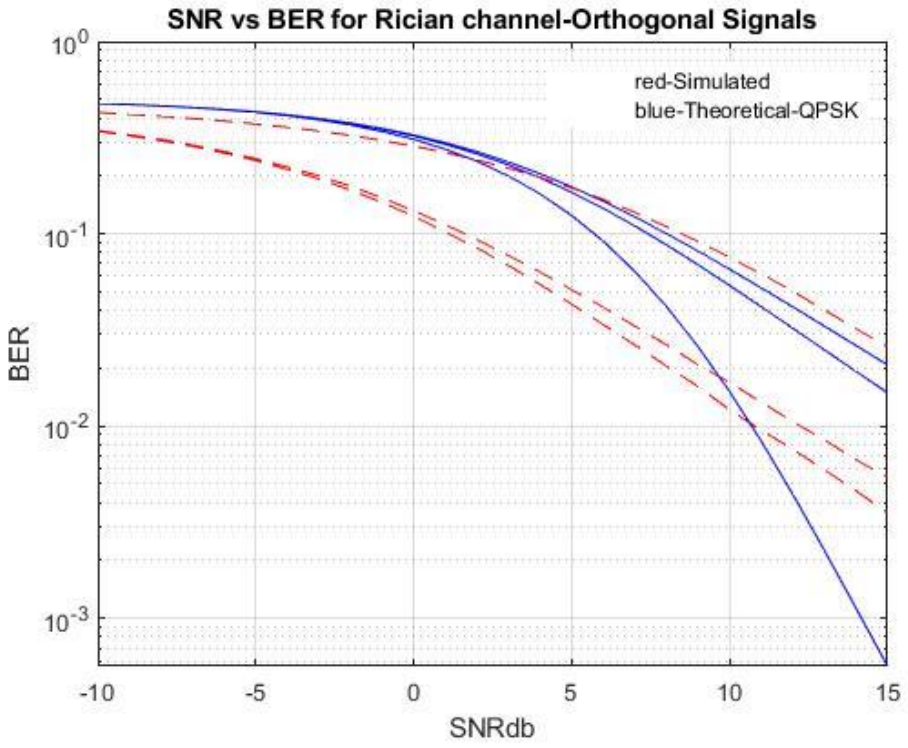
```
    Rx_symb = Tx_symb.*Z3 + noise;

    % Equalizer
    Rx_symb = Rx_symb./Z3;

    %ML Detector
    detected_real = real(Rx_symb)>=0;
    detected_img = imag(Rx_symb)>=0;

Rx_bits=reshape([detected_img;detected_real],1,nobpc);

    %Bit Error rate Calculation
    BER3(index) = sum(xor(Tx_bits,Rx_bits))/nobpc;
    index=index+1;
end

SNR1 = 10.^(SNRdb/10);

% Theoretical BER for Rician channel (Orthogonal
signals)
Th_BER4 = ((1+K1)./(2+2*K1+SNR1)).*exp((-
1*K1*SNR1)./(2+2*K1+SNR1));
Th_BER5 = ((1+K2)./(2+2*K2+SNR1)).*exp((-
1*K2*SNR1)./(2+2*K2+SNR1));
Th_BER6 = ((1+K3)./(2+2*K3+SNR1)).*exp((-
1*K3*SNR1)./(2+2*K3+SNR1));

figure(1), semilogy(SNRdb,Th_BER4),title('Theoretical
SNR vs BER for Rician channel-Orthogonal Signals');
hold on
semilogy(SNRdb,Th_BER5);
xlabel("SNRdb");
ylabel("Theoretical BER");
legend('K1','K2','K3');
semilogy(SNRdb,Th_BER6);

figure(2),semilogy(SNRdb,BER1,'r--'),title('SNR vs BER
for Rician channel-Orthogonal Signals');
grid on;
hold on;
```

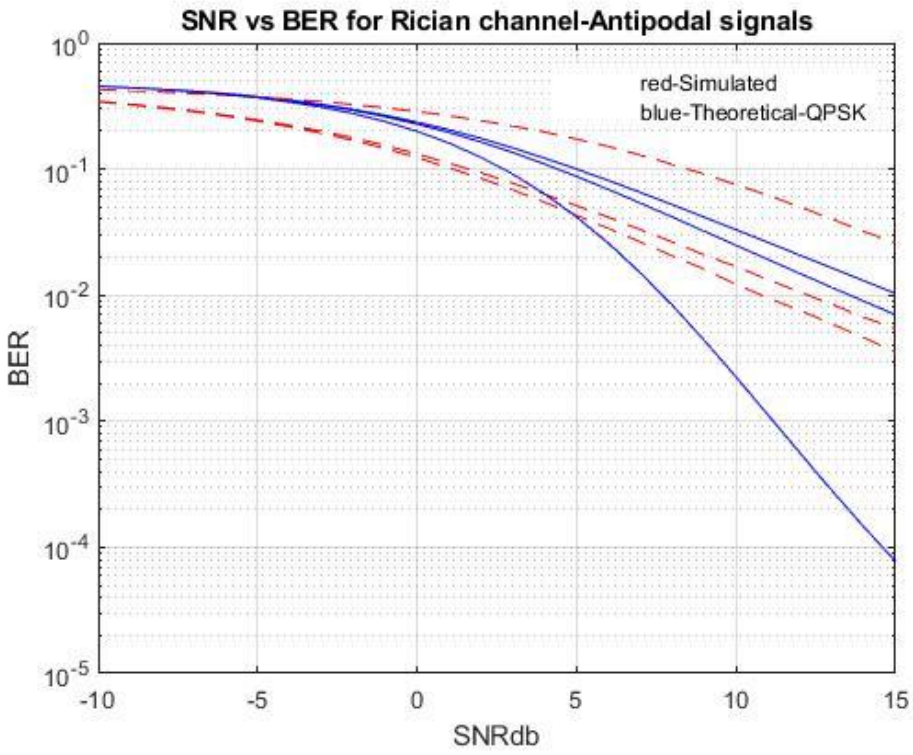
```
semilogy(SNRdb,Th_BER4,'b');
semilogy(SNRdb,BER2,'r--');
semilogy(SNRdb,Th_BER5,'b');
semilogy(SNRdb,BER3,'r--');
semilogy(SNRdb,Th_BER6,'b');
xlabel("SNRdb");
ylabel("BER");
legend('red-Simulated','blue-Theoretical-QPSK');
grid on;

% Theoretical BER for Rician channel (Antipodal
signals)
Th_BER1 = ((1+K1)./(2+2*K1+2*SNR1)).*exp((-
1*K1*SNR1)./(1+K1+SNR1));
Th_BER2 = ((1+K2)./(2+2*K2+2*SNR1)).*exp((-
1*K2*SNR1)./(1+K2+SNR1));
Th_BER3 = ((1+K3)./(2+2*K3+2*SNR1)).*exp((-
1*K3*SNR1)./(1+K3+SNR1));

figure(3), semilogy(SNRdb,Th_BER1),title('Theoretical
SNR vs BER for Rician channel-Antipodal Signals');
hold on
semilogy(SNRdb,Th_BER2);
xlabel("SNRdb");
ylabel("Theoretical BER");
legend('K1','K2','K3');
semilogy(SNRdb,Th_BER3);

figure(4),semilogy(SNRdb,BER1,'r--'),title('SNR vs BER
for Rician channel-Antipodal signals');
grid on;
hold on;
semilogy(SNRdb,Th_BER1,'b');
semilogy(SNRdb,BER2,'r--');
semilogy(SNRdb,Th_BER2,'b');
semilogy(SNRdb,BER3,'r--');
semilogy(SNRdb,Th_BER3,'b');
xlabel("SNRdb");
ylabel("BER");
legend('red-Simulated','blue-Theoretical-QPSK');
```

```
grid on;
```

10. Print and attach a single Figure presenting the theoretical and simulation curves superimposed one on top of the other for K=1,3,10.

Sol:





Discussion:

11. Discuss the results for Rician, Rayleigh and AWGN channels. Address trends and compare BER performance.

Sol: AWGN Channel

- It is considered as a perfect channel, without any scattering and reflections.
- The receivers receives everything without any distortion.
- BER decreases to zero at a very low signal power.
- This channel is used as a standard for other channels to compare their performance in real time scenarios.

Rayleigh Channel

- This channel considers those scenarios from real-time world, in which the signals experience maximum amount of scattering, reflections and attenuation.
- There is no Line-of-Sight in these cases.
- This channel acts as a worst case scenario.
- Even at very high signal power, there will be some attenuation in the bits received, i.e., there will be BER.

Rician Channel

- This channel reflects a more realistic scenario that our transmitted signals experience in our environment.
- It considers those cases too, when there is some Line-of-Sight between the transmitter and receiver.
- The signals experience less attenuation, scattering and reflection.
- And the SNR Vs BER is much lower, when compared to Rayleigh channel.
- This channel is much complex when compared to Rayleigh channel, but it gives us a better understanding of the experience that our transmitted signals get while travelling in the real time environment.