ECE471: Testing & Electronic Measurements (2)
Faculty of Engineering
Ain Shams University
4th Year ECE
Spring 2020

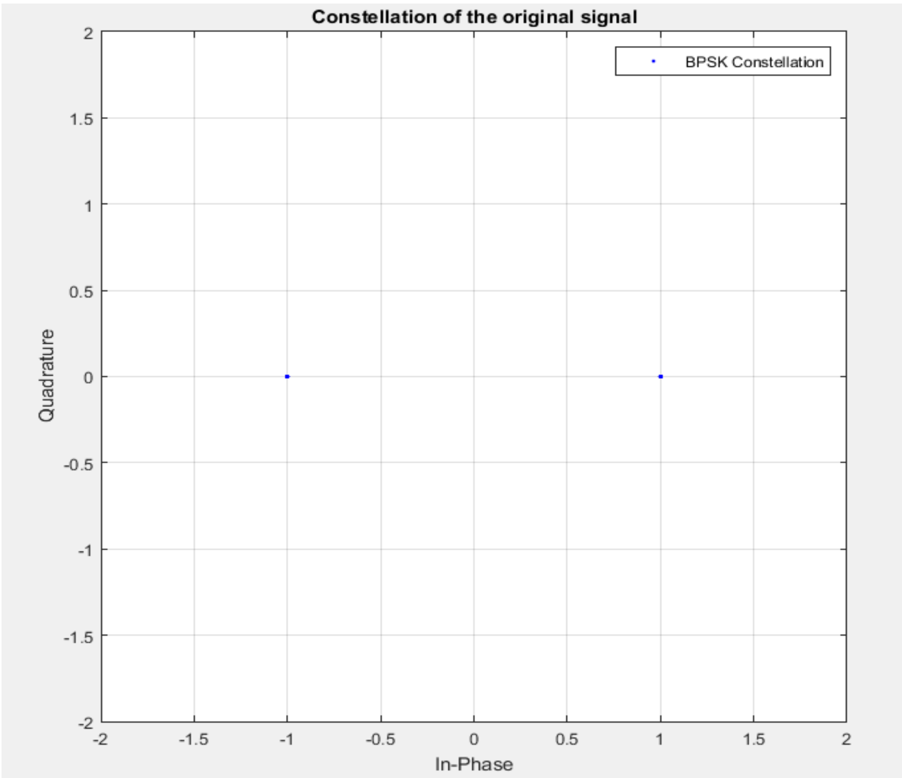
ECE471: TESTING & ELECTRONIC MEASUREMENTS (2) LAB REPORT (4C)	
Experiment: C9	
Section 2	
Group No 4	
Student Name:	**Report Mark**
محمود رمزي محمد السعيد.	

Generating Input Bits And Shaping it:

```
close all;
clc;
```

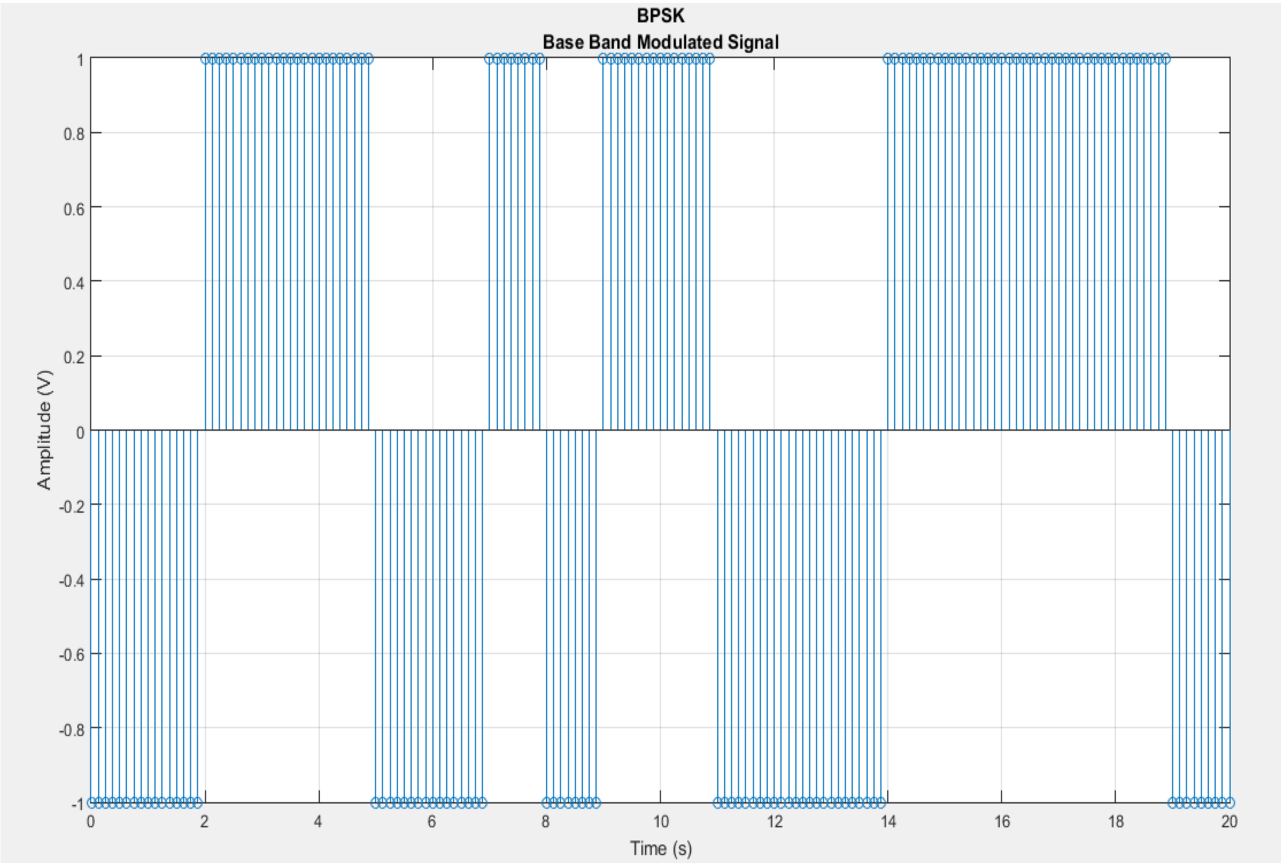
```
N_symbol = 2; % Number of Symbols in MPSK
N_bits = log2(N_symbol); % Number of Bits/Symbol
% generating streams of 1's and 0's by rounding random numbers to nearest 1 or 0 ,
%Mtx size = 2000 x 1
x_stream = randi([0 1],[2000 1]);
% Converting Binary symbols to decimal values so we can map it later
x_decimal = bi2de(x_stream,'left-msb');
```

```
%% Mapping the symbols and drawing constellation

y=BPSK_MOD(x_decimal); % mapping symbols to constellation

%% Drawing of constellation of Original signal after mapping without effect of
channel noise
scatterplot(y,1,0);
%produces a scatter plot for the signal y every 1 value of the signal starting from
the 0 offset
hold on
grid on
figure (1)
axis([-2 2 -2 2]);
title('Constellation of the original signal')
legend('BPSK Constellation')
```



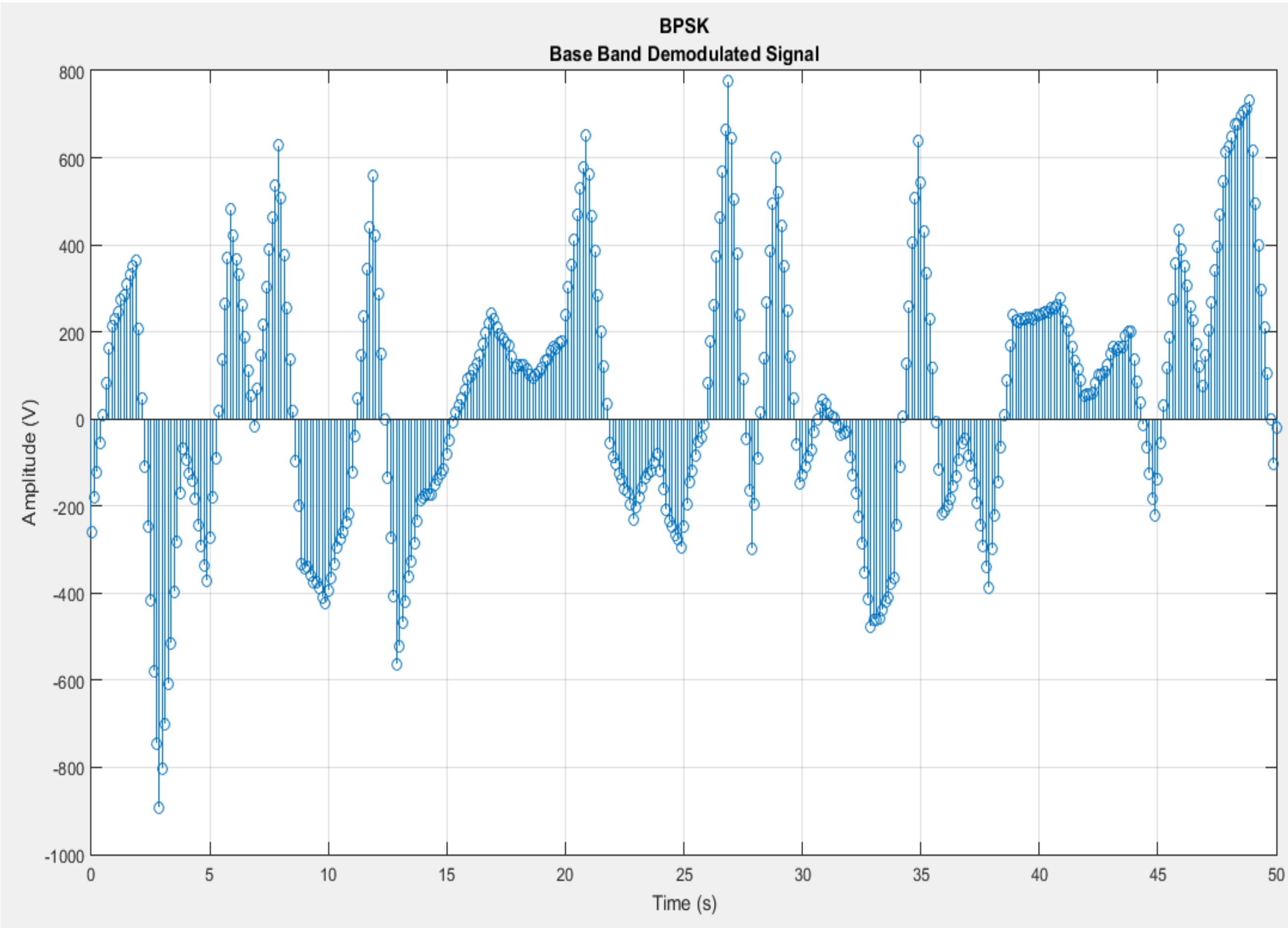
2) Figure for the modulated baseband signal after pulse shaping filter.

```
%% UPsampling &rectangular shaping
L=8; %# of samples
%Z = upsample(y,n);
M= rectpulse(y,L);
t= linspace(0,2000,length(M));
figure(2)
stem(t,real(M));
grid on
xlim([0 20]);
xlabel('Time (s)');
ylabel('Amplitude (V)');
title({'BPSK','Base Band Modulated Signal'});
```



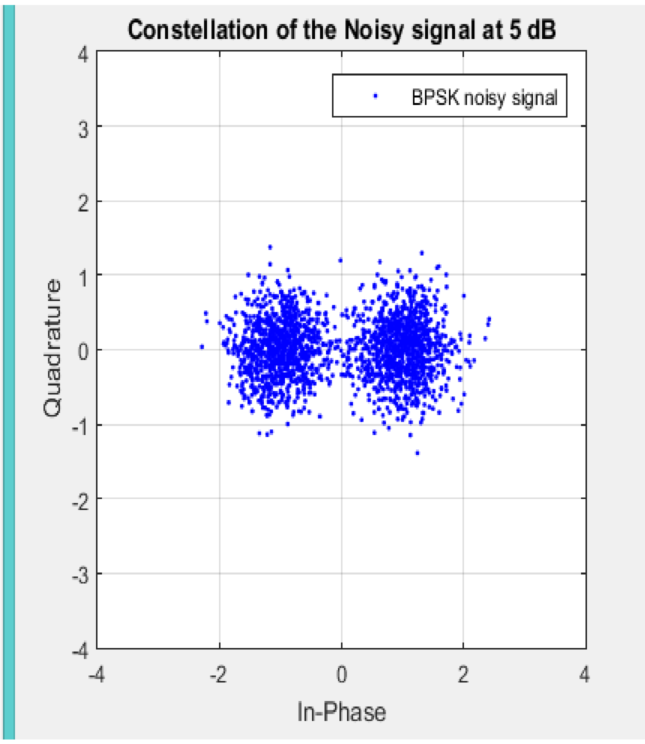
3) Figure for the received signal after matched filter at SNR equals to 10db

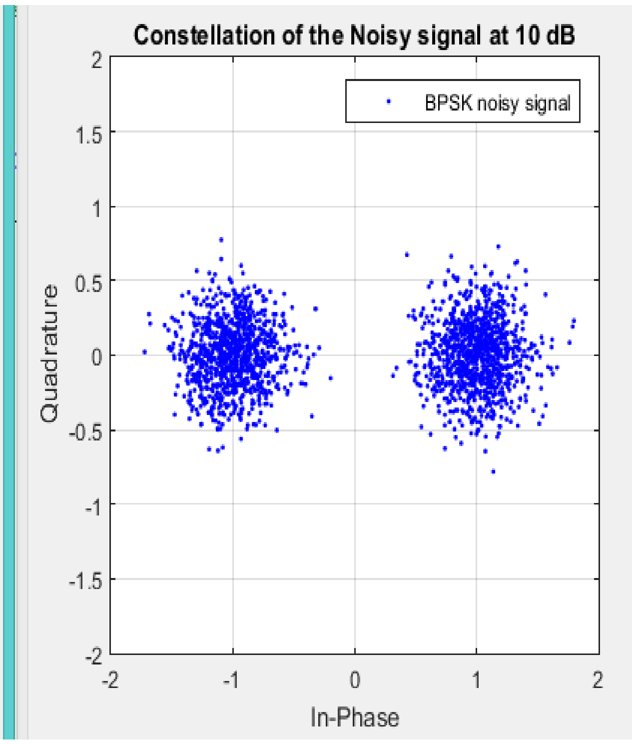
```
%% matched filter:
b1 = M(end:-1:length(M));%matched filter's coefficients is given by time reverse of M
b2= filter(b1,1,M);
p=(White_noise(b2,10));  %adding 10db white noise
MF=conv(M,p,'same');
figure (3)
stem(t,real(MF));
grid on;
xlim([0 50]);
xlabel('Time (s)');
ylabel('Amplitude (V)');
title({'BPSK','Base Band Demodulated Signal'});
```

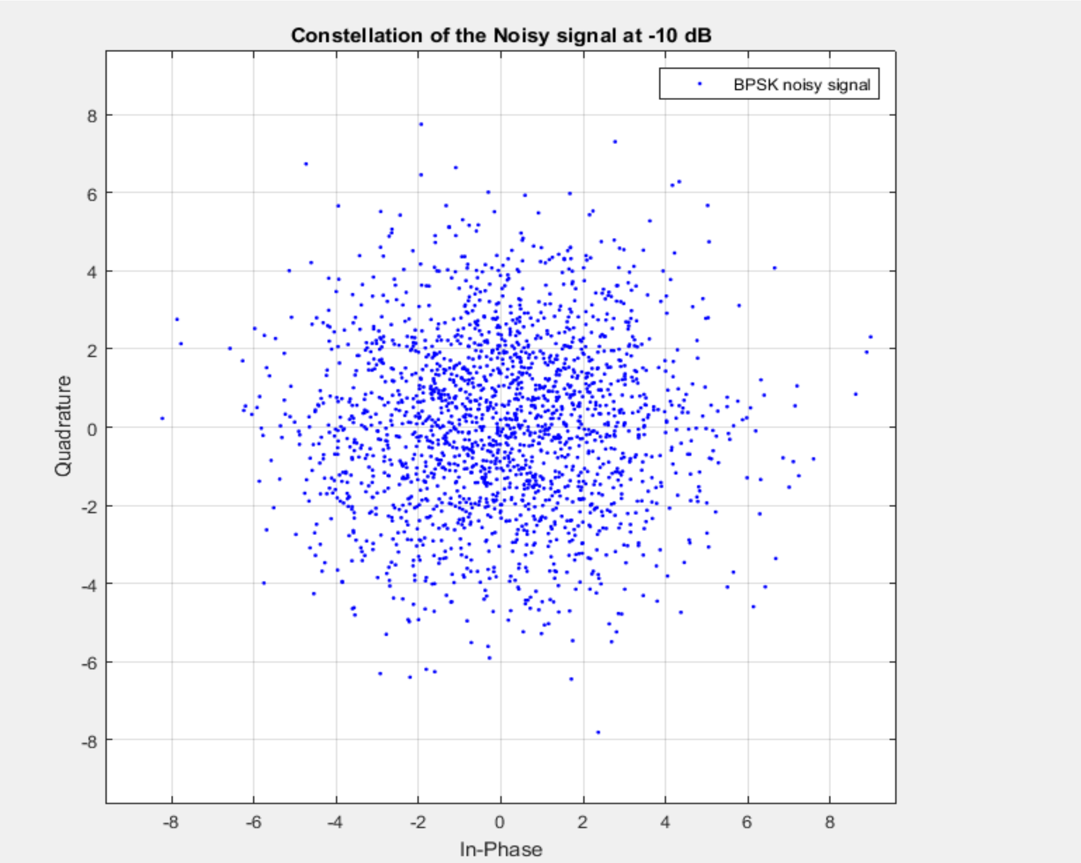


4)Figures for the constellation diagram after downsampling at SNR equals to -10, 5 and 10 dB

```
%% AWGN channel effect
b3= downsample(b2,L); %downsampling
figure (3)
scatterplot(White_noise(b3,10)); %produces a scatter plot for the signal y after
making its SNR dwn to 30db
grid on
axis([-2 2 -2 2]);
legend('BPSK noisy signal');
title('Constellation of the Noisy signal at 10 dB')

figure (4)
scatterplot(White_noise(b3,5)); %produces a scatter plot for the signal y after
making its SNR dwn to 5db
grid on
axis([-4 4 -4 4]);
legend('BPSK noisy signal');
title('Constellation of the Noisy signal at 5 dB')

figure (5)
scatterplot(White_noise(b3,-10)); %produces a scatter plot for the signal y after
making its SNR dwn to -10db
grid on
%axis([-5 5 -5 5]);
legend('BPSK noisy signal');
title('Constellation of the Noisy signal at -10 dB')
```

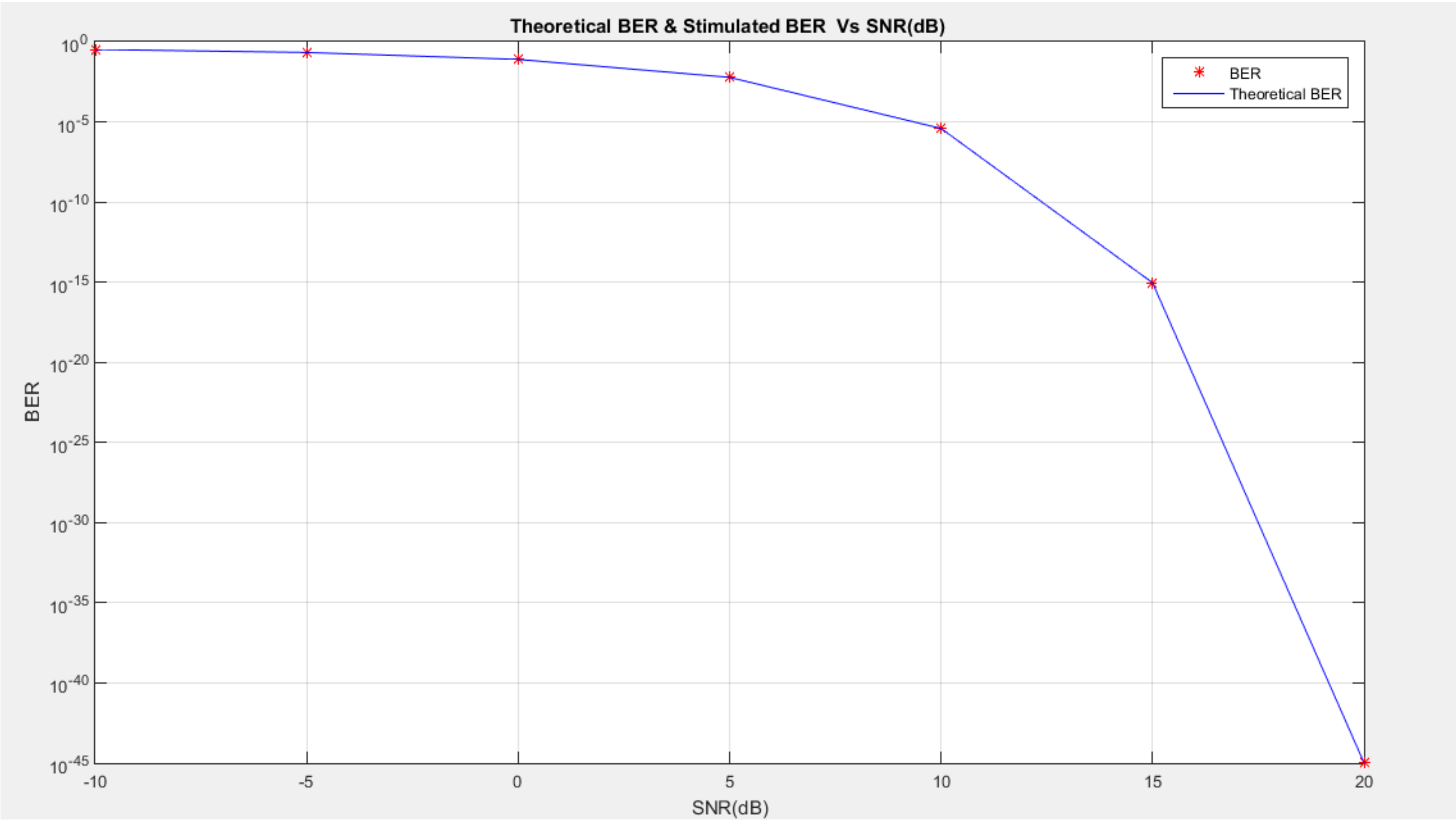





Constellation of the Noisy signal at -10 dB
BPSK noisy signal
Quadrature
In-Phase
8
6
4
2
0
-2
-4
-6
-8

6) Figure for the BER versus SNR for the simulated and therortical systems.

```
%% Detection & BER
Y_Rx= BPSK_DMOD(b3);     %Demodulation of the original signal
Y_Rx_SNR10 = BPSK_DMOD(White_noise(b3,10));  %Demodulation of the sig after adding
noise from channel
Y_Rx_SNR5  = BPSK_DMOD(White_noise(b3,5));    % to it with different values that makes
its SNR dwn
Y_Rx_SNRn10    = BPSK_DMOD(White_noise(b3,-10));     % to different values too
SNR.dB = -10:5:20;  %preparing the SNR axis to be used in plots
BER = [SER(x_decimal,Y_Rx_SNRn10) SER(x_decimal,Y_Rx_SNR5) ...  %form a vector for
the the symbol error rate
    SER(x_decimal,Y_Rx_SNR10)]/N_bits  ; %of the RX demodulation signals in snr
levels
SNR.lin= 10.^(SNR.dB/10);     %convert the SNR  from db to decimal
BER_T = (1/2)*erfc(sqrt(SNR.lin));
figure (7)
semilogy(SNR.dB , BER,'red*')
title('BER Vs SNR(dB)')
legend('BER')
xlabel('SNR(dB)')
ylabel('BER')
grid on
hold on;
semilogy(SNR.dB , BER_T,'blue' )
title('Theoretical BER & Stimulated BER  Vs SNR(dB) ')
legend('BER','Theoretical BER')
xlabel('SNR(dB)')
ylabel('BER')
grid on
```



Functions:

BPSK_MOD:

```
%% This is a function that perform mapping of symbols on constellation for BPSK
Modulation techinque

function mappedSymbol=BPSK_MOD(symbol)

       mappedSymbol=cos((2*pi/2)*symbol);
    %Symbole eqn for the BPSK
end
```

BPSK_DMOD:

```
%% This is a function that performs demapping of symbols on constellation for BPSK
Modulation techinque

function [D]= BPSK_DMOD(y)

D=mod(angle(y),2*pi);   %returns the remainder after division the phase of Rx signal
by 2pi
D(and(D > 0.0*pi, D < 0.5*pi)) = 1;
D(and(D > 0.5*pi, D < 1.0*pi)) = 0;
end
```

White_noise:

```
%% This is a function that simulating the channel where awgn noise is added
function N = White_noise(Signal,SNR_dB)
rng('default'); % to geneate the same random values of noise in every time
L = length(Signal); %length of Signal
SNR = 10^(SNR_dB/10); %SNR to linear scale
Esym = sum(abs(Signal).^2)/(L); %Calculate actual symbol energy
N0 = Esym/SNR; %Find the noise spectral density
noiseSigma=sqrt(N0/2); %Standard deviation for AWGN Noise
n = noiseSigma*(randn(L,1)+1i*randn(L,1));%computed noise
N = Signal + n; %received signal
end
```

SER:

```
%% This function gets the symbol error rate
function [H]=SER(x,y)
n=0;  %just a counter
for i=1:length(x)
    if x(i) == y(i)  %checks if the original sig matched the RX deomodulated sig
        n=n+1;
    end
end
H=(length(x)-n)/length(x);
end
```