Computer Exercise IV: Digital Modulation

I- Prelab Assignment:

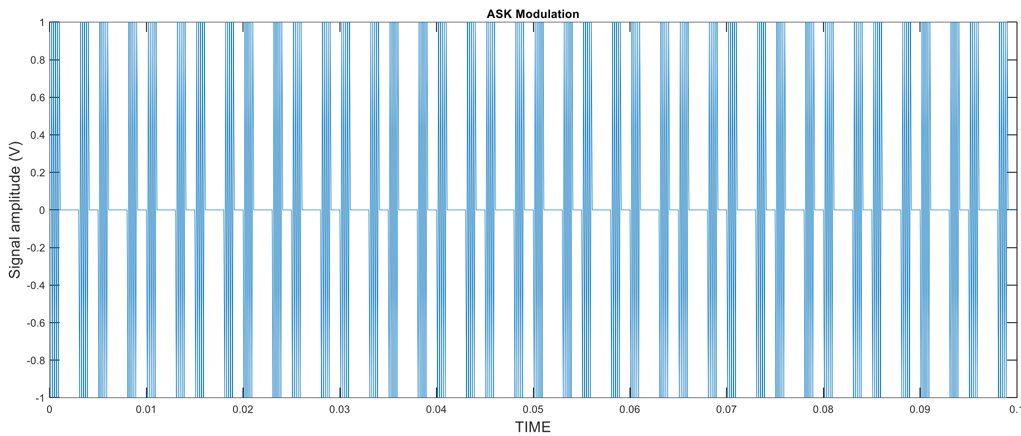
1.

$$b_i = 1\,,0\,,0,1\,,0 \quad \forall i = 1\,,\ldots 5$$

$$b(t) = b_i \qquad 0 \leq t \leq 1/R_b$$

a)

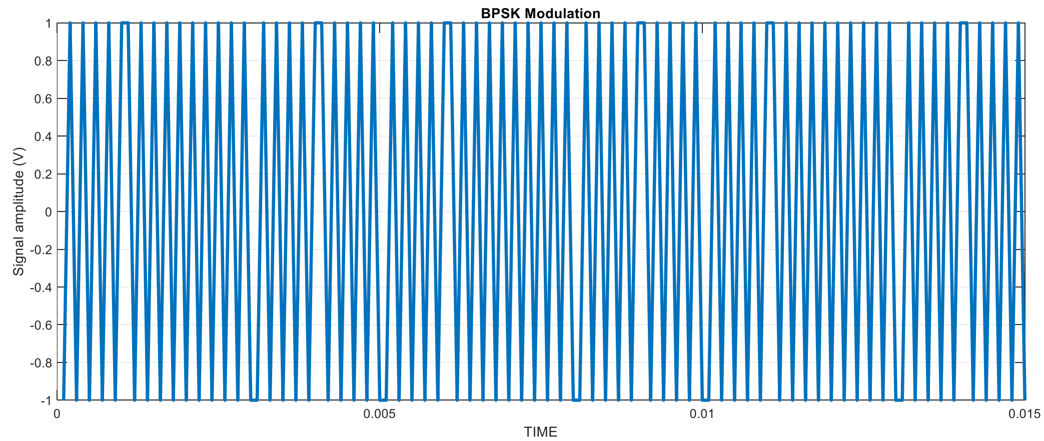
```
clear all
clc
b = [1 0 0 1 0];Rb = 1000;
simulation = 100;e = 1;
time = 0.0001 : 0.0001 : simulation*(1/Rb);
for i = 1:simulation
    if e > length(b)
        e = 1;
    end
Ask(1+(i-1)*10:i*10) = b(e)*cos(2*pi*(5e+3)*time(1+(i-1)*10:i*10));
e = e + 1;% index of binary sequence
end
plot(time,Ask)
```



b)

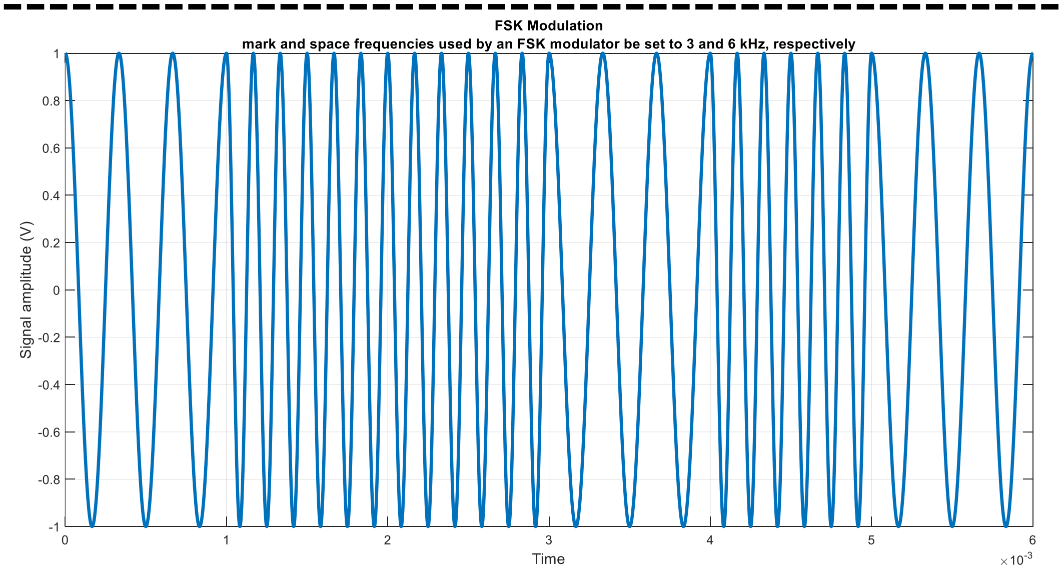
```
clear all
clc
b = [1 0 0 1 0];Rb = 1000;
simulation = 15;e = 1;
time = 0.0001 : 0.0001 : simulation*(1/Rb);
for i = 1:simulation
    if e > length(b)
        e = 1;
    end
Psk(1+(i-1)*10:i*10) = cos(2*pi*(5e+3)*time(1+(i-1)*10:i*10)+pi+pi*b(e));
```

```
e = e + 1;% index of binary sequence
end
plot(time,Psk)
```



c)

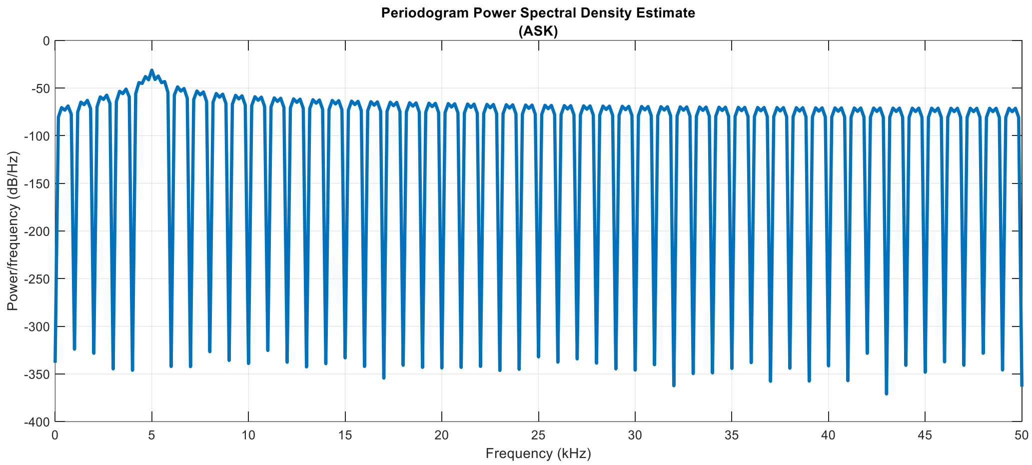
```
clear all
clc
b = [1 0 0 1 0];Rb = 1000;
simulation = 20;e = 1;
time = 0.000001 : 0.000001 : simulation*(1/Rb);
for i = 1:simulation
    if e > length(b)
        e = 1;
    end
Fsk(1+(i-1)*1000:i*1000) = cos(2*pi*((6e+3)-b(e)*(3e+3))*...
time(1+(I 1)*1000:i*1000));
e = e + 1;% index of binary sequence
end
plot(time,Fsk)
```



2.

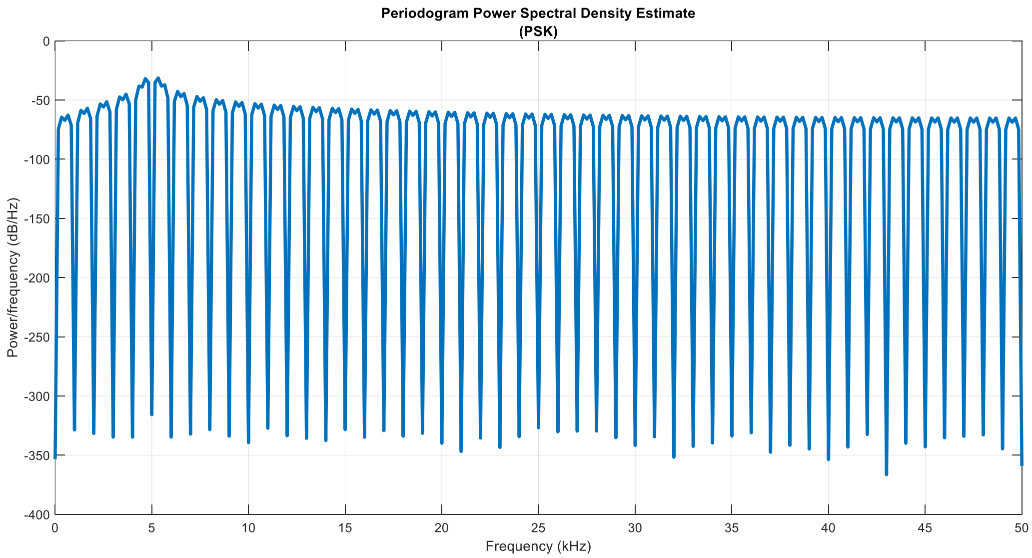
For ASK :

```
plot(psd(spectrum.periodogram,Ask,'Fs',100000,'NFFT',length(Ask)));
```



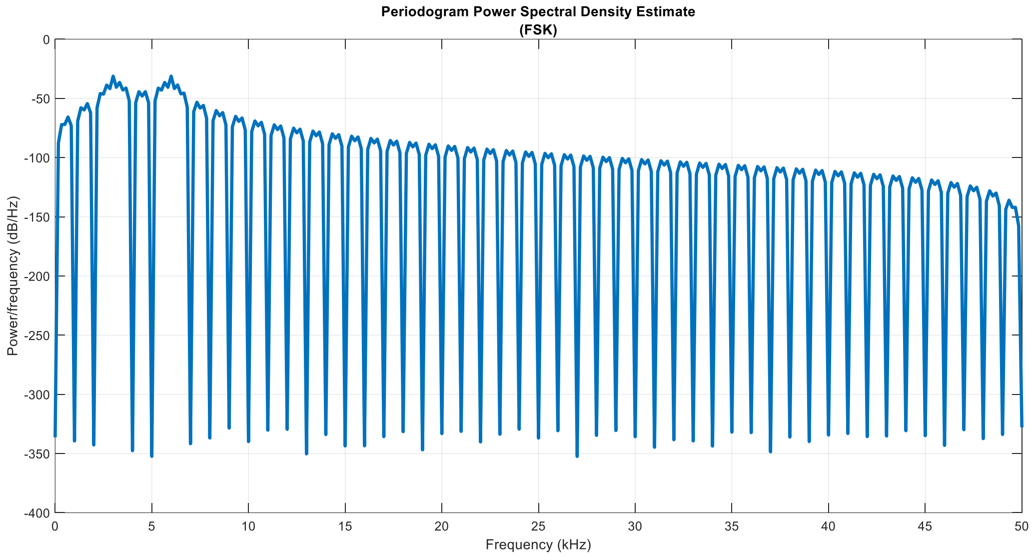
For psk :

```
plot(psd(spectrum.periodogram,Psk,'Fs',100000,'NFFT',length(Psk)));
```



For FSK :

```
plot(psd(spectrum.periodogram,Fsk,'Fs',100000,'NFFT',length(Fsk)));
```



3.

```
v = Ask .* cos(2*pi*(5e+3)*time);
Matchfilter = ones(1,10);
W = conv(Matchfilter,v);
% sampeling at 10th member because it's at the center of convolved signal
S = [W(10),W(20),W(30),W(40),W(50)];
bb = S> 0.5
```

bb =

  1   0   0   1   0

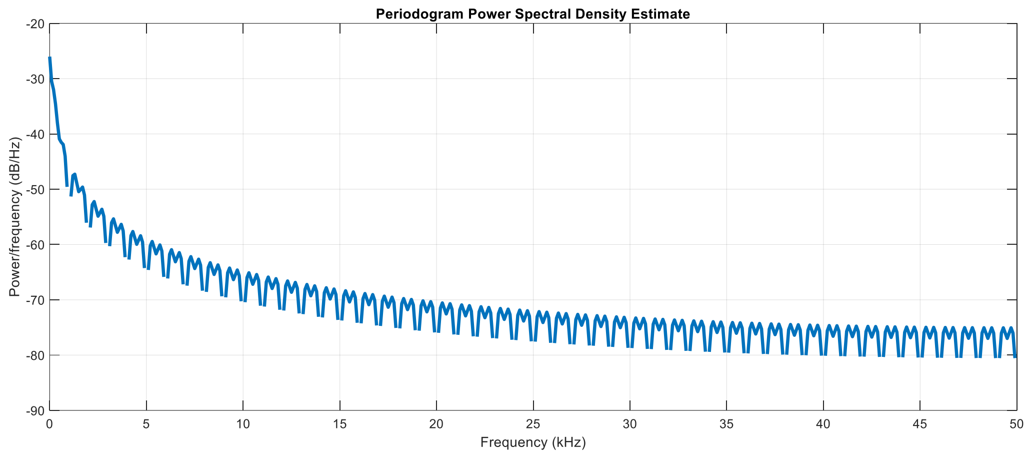
A. Generation of Modulated Signals

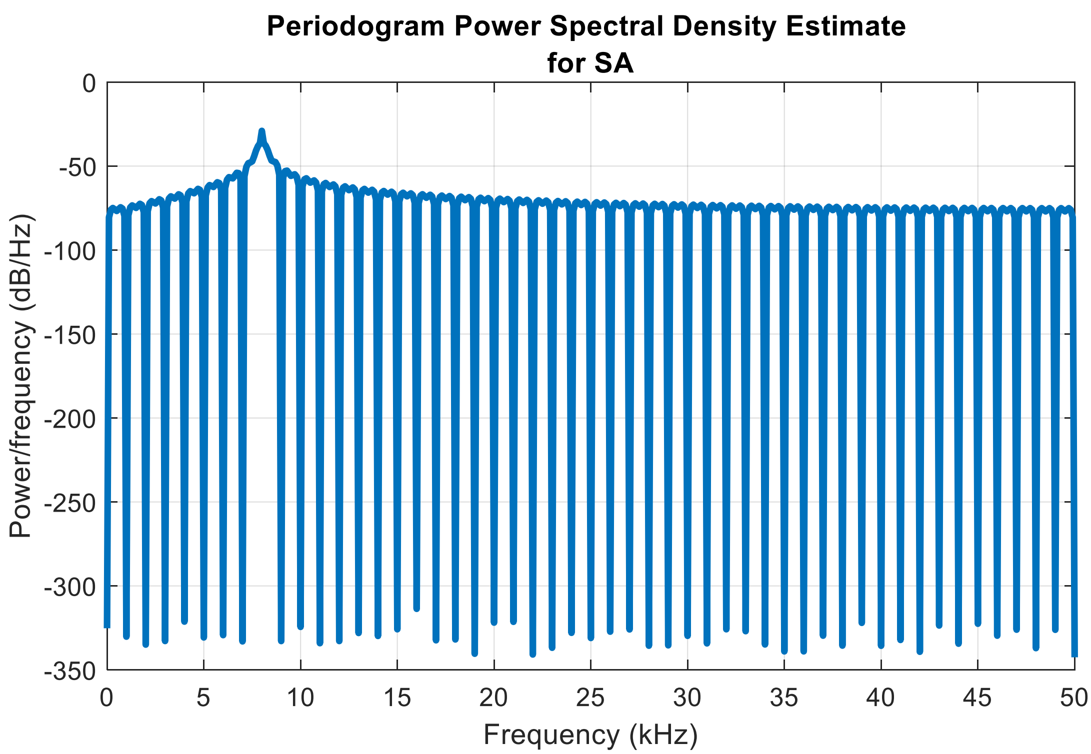
*Amplitude-Shift Keying (ASK)

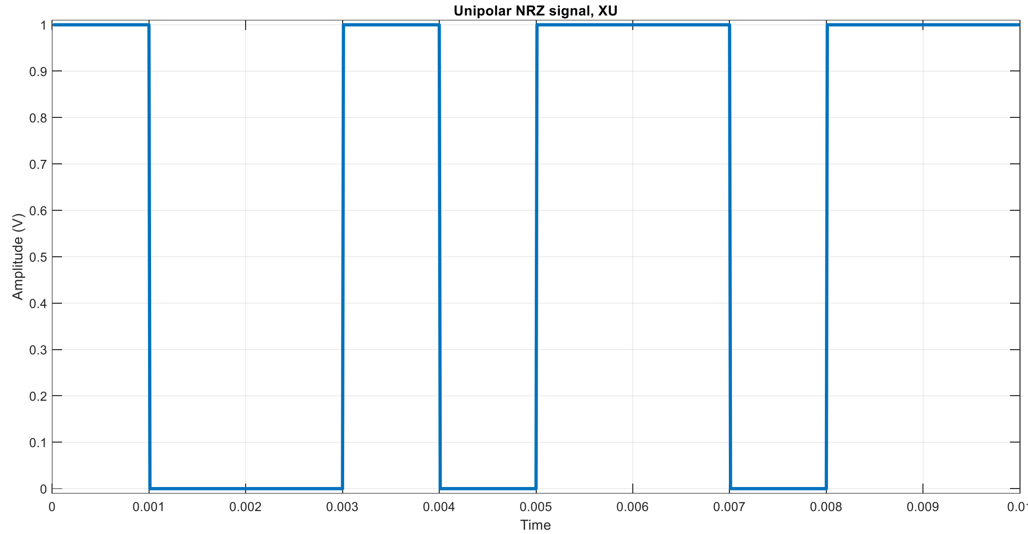
A.1 , 2 , 3

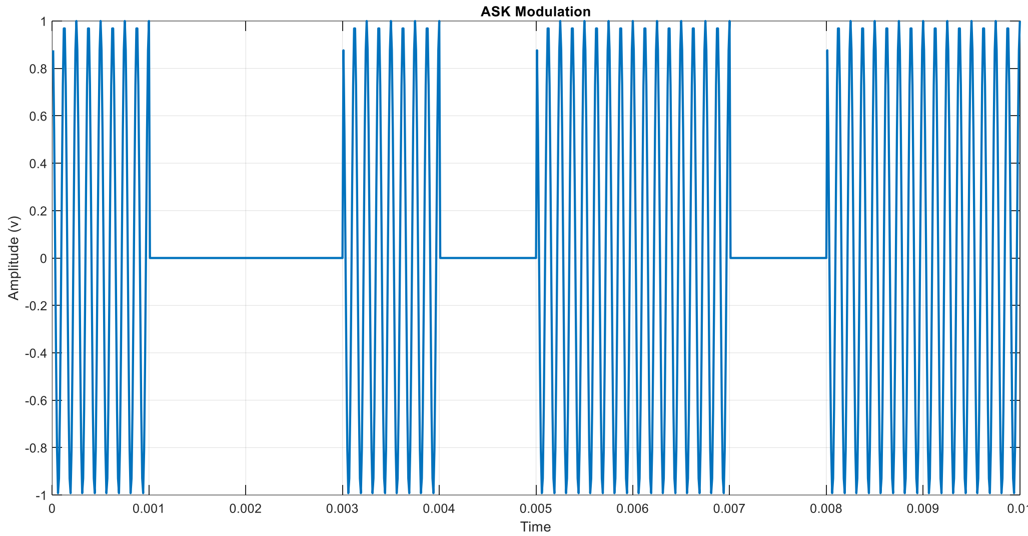
```
clear all
clc
b = [1 0 0 1 0];Rb = 1000;
b(6:10)= randi([0 1],[1 5]);
simulation = 10;
time = 0.00001 : 0.00001 : simulation*(1/Rb);
XU = zeros(1,length(time));
for i = 1:simulation
XU(1+(i-1)*100:i*100)= b(i);
end
SA = XU.*cos(2*pi*((8e+3))*time);
plot(time,XU)
```

```
axis([0 max(time) -0.01 1.01])
figure
plot(time,SA)
figure
plot(psd(spectrum.periodogram,XU,'Fs',100000,'NFFT',length(XU)));
figure
plot(psd(spectrum.periodogram,SA,'Fs',100000,'NFFT',length(SA)));
```









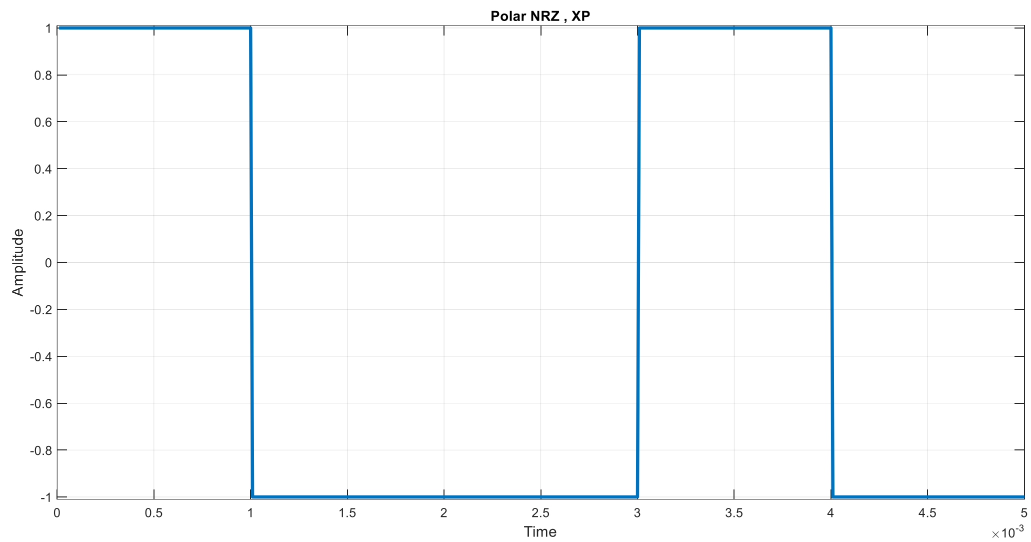
*Phase-Shift Keying (PSK)

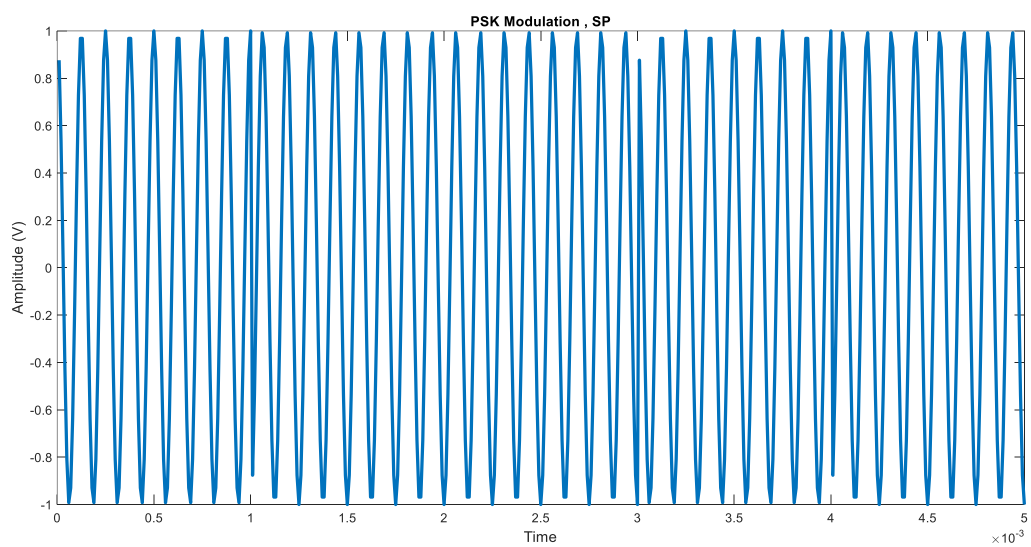
A.4 , 5 , 6

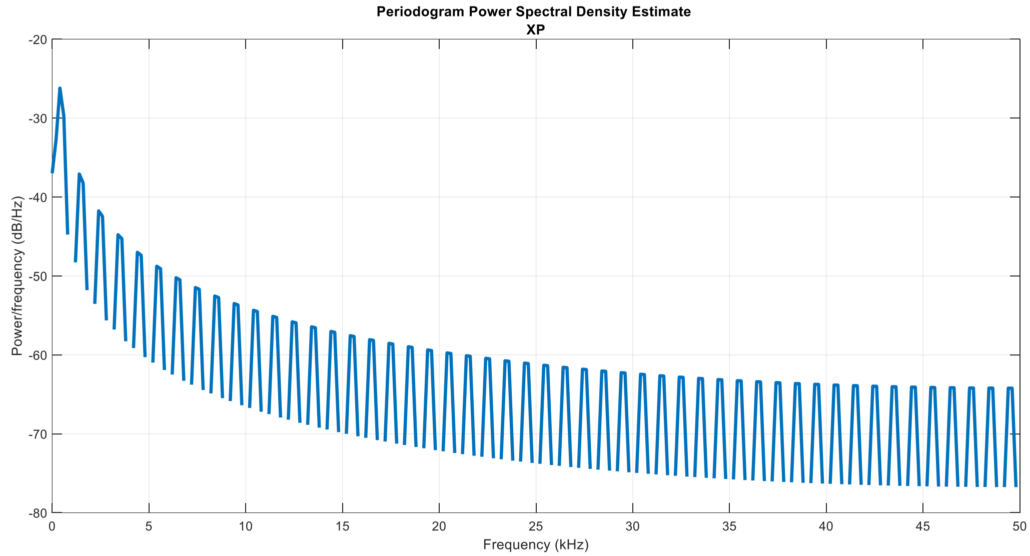
phase difference between sp and the carrier cos(2πfct) during the first = 0° *and second* = 180°

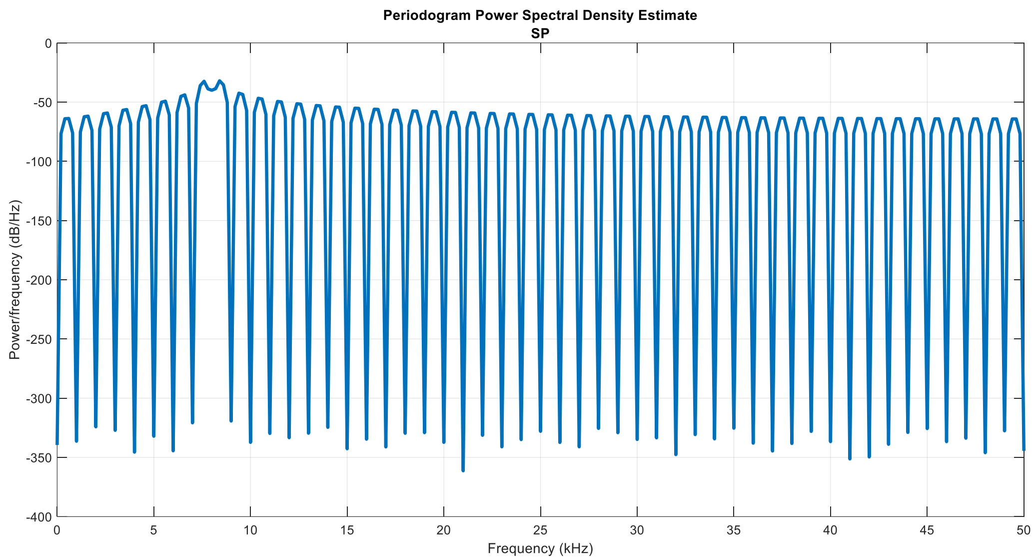
```
clear all
clc
b = [1 0 0 1 0];Rb = 1000;
b(6:10)= randi([0 1],[1 5]);
simulation = 5;
time = 0.00001 : 0.00001 : simulation*(1/Rb);
XP = zeros(1,length(time));
for i = 1:simulation
XP(1+(i-1)*100:i*100)= 2*b(i)-1;
end
```

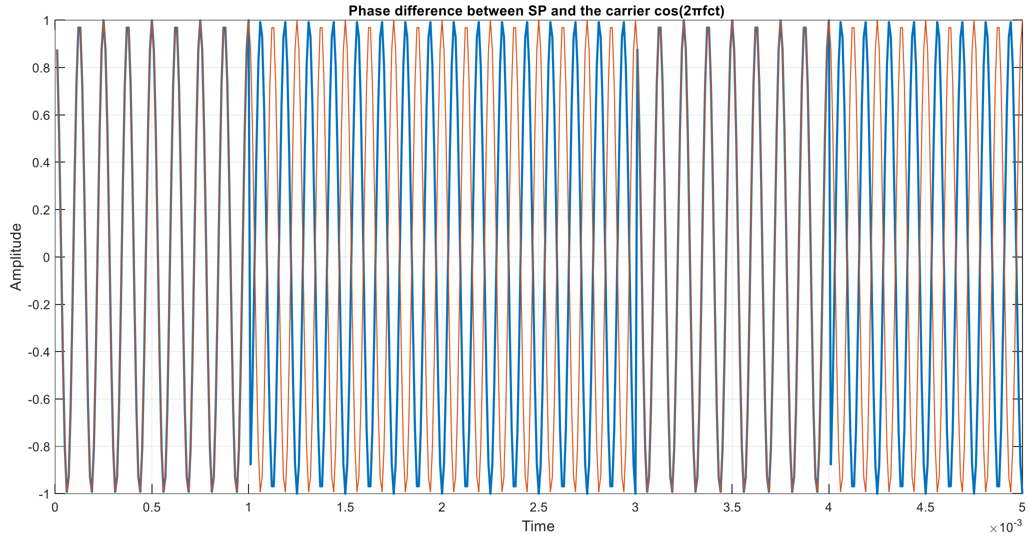
```
SP = XP.*cos(2*pi*((8e+3))*time);
plot(time,XP)
axis([0 max(time) -1.01 1.01])
figure
plot(time,SP)
figure
plot(time,SP)
hold on
plot(time,cos(2*pi*((8e+3))*time))
figure
plot(psd(spectrum.periodogram,XP,'Fs',100000,'NFFT',length(XP)));
figure
plot(psd(spectrum.periodogram,SP,'Fs',100000,'NFFT',length(SP)));
```






Periodogram Power Spectral Density Estimate
XP
Power/frequency (dB/Hz)
Frequency (kHz)

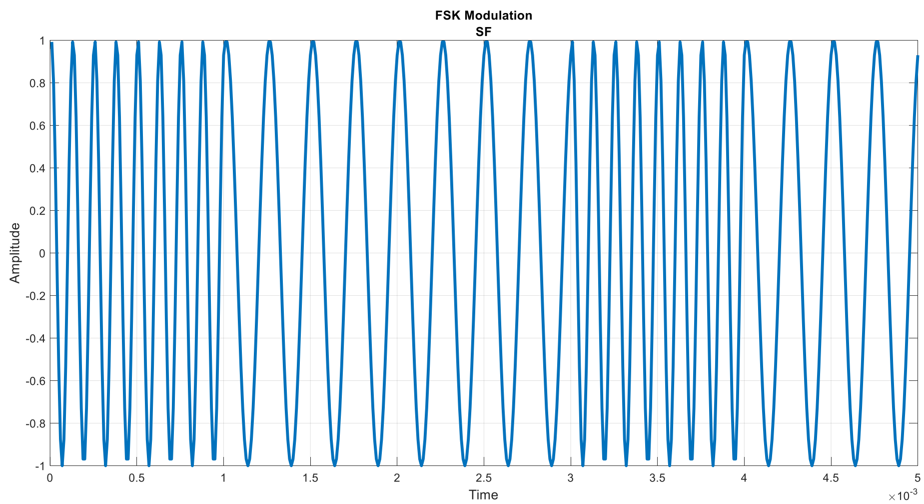

Periodogram Power Spectral Density Estimate
SP
Power/frequency (dB/Hz)
Frequency (kHz)

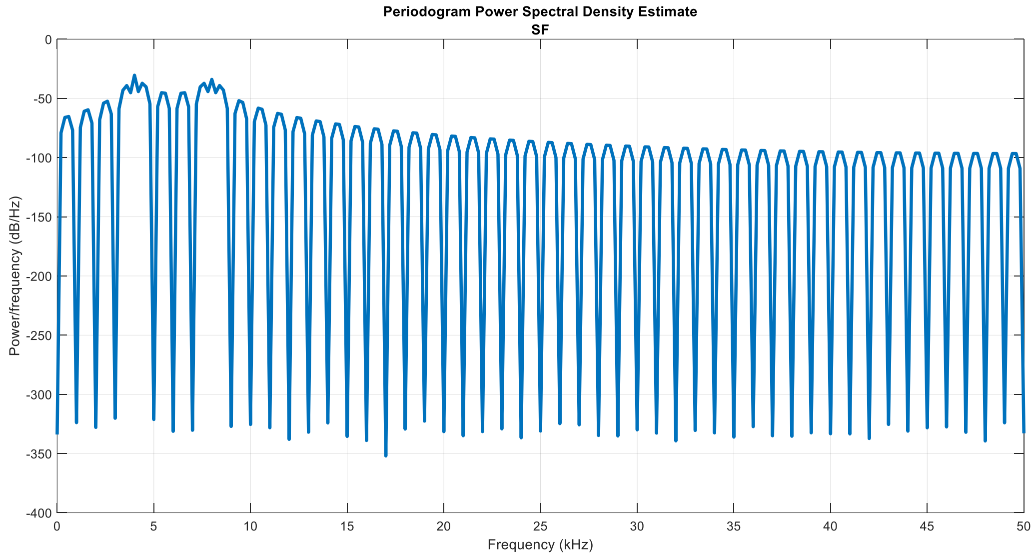


*Frequency-Shift Keying (FSK)

A.6 , 7

```
clear all
clc
b = [1 0 0 1 0];Rb = 1000;
b(6:10)= randi([0 1],[1 5]);
simulation = 5;
time = 0.00001 : 0.00001 : simulation*(1/Rb);
XP = zeros(1,length(time));
for i = 1:simulation
XP(1+(i-1)*100:i*100)= 2*b(i)-1;
end
SF = vco(XP,[4e+3 8e+3],1e+5);
plot(time,SF)
figure
plot(psd(spectrum.periodogram,SF,'Fs',100000,'NFFT',length(SF)));
```





One way to implement an FSK signal is by using to ASK signals with different carrier frequencies:

Where : f1 = mark frequency(4 kHz) and f2 = space frequency(8 kHz)

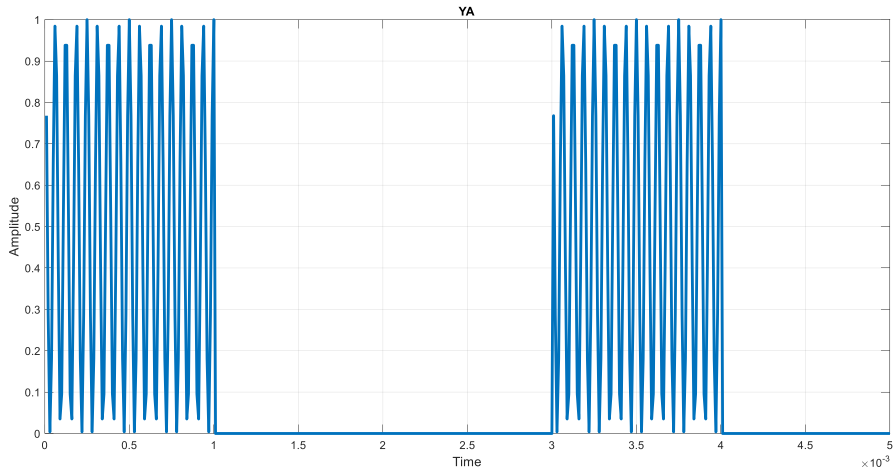
Surely FSK is not efficient bandwidth . ASK is desire as this point.

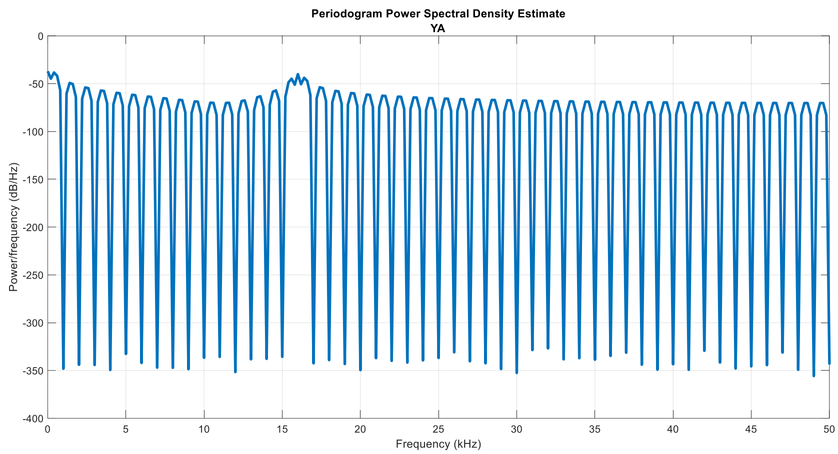
B. Coherent and Noncoherent Detection

***Coherent Detection**

B.1

```
YA = SA.*cos(2*pi*((8e+3))*time);
plot(time,YA)
figure
plot(psd(spectrum.periodogram,YA,'Fs',100000,'NFFT',length(YA)));
```



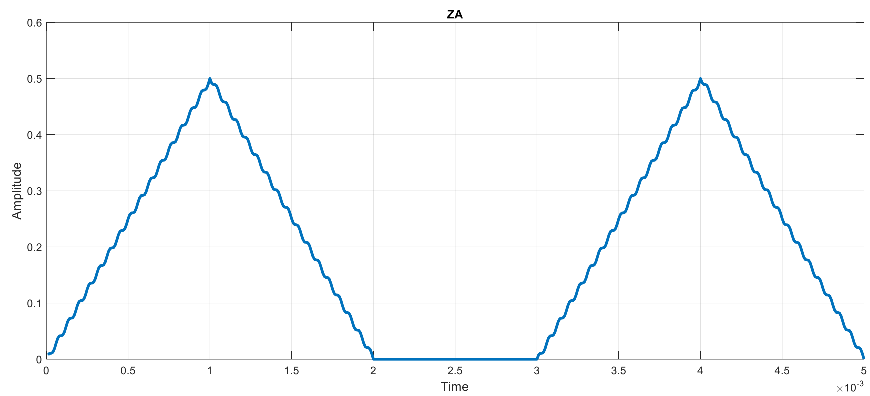


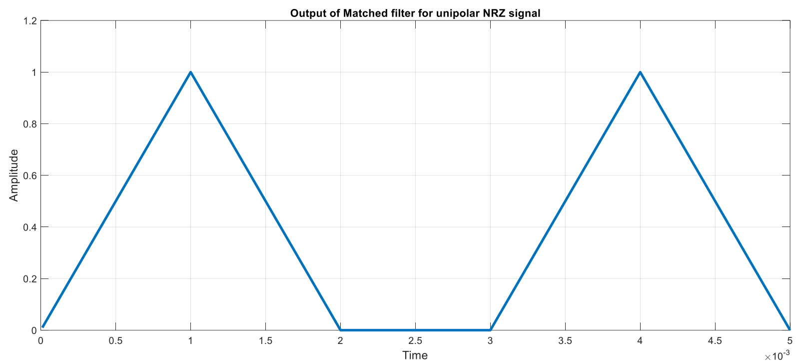
B.2

By assuming matched filter is normalized to it's energy.

```
YA = SA.*cos(2*pi*((8e+3))*time);
ZA = conv(0.01*ones(1,100),YA);
Output_of_unipolar_NRZ_signal = conv(0.01*ones(1,100),XA);
plot(time(1:500),ZA(1:500))
figure
plot(time(1:500),Output_of_unipolar_NRZ_signal(1:500))
```

Because two mentioned output are equivalent to each other.





B.3

By assuming matched filter is normalized to it's energy.

Phase Error	Peak Amplitude [V]
0°	0.5
20°	0.47
60°	0.25
80°	0.09
120°	0

Surely Phase Error = $0°$ *is desirable for BER .*

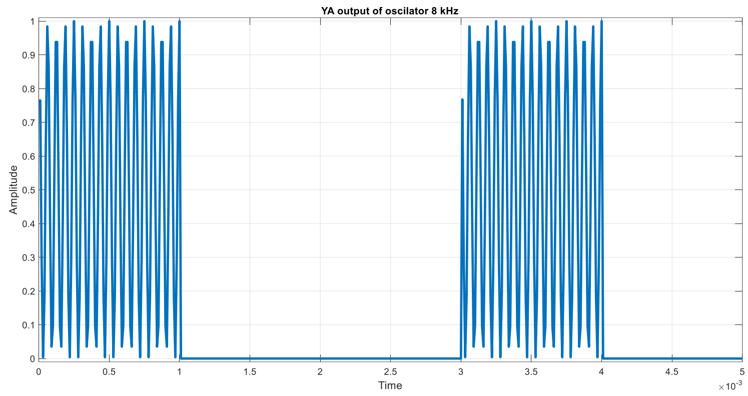
B.4

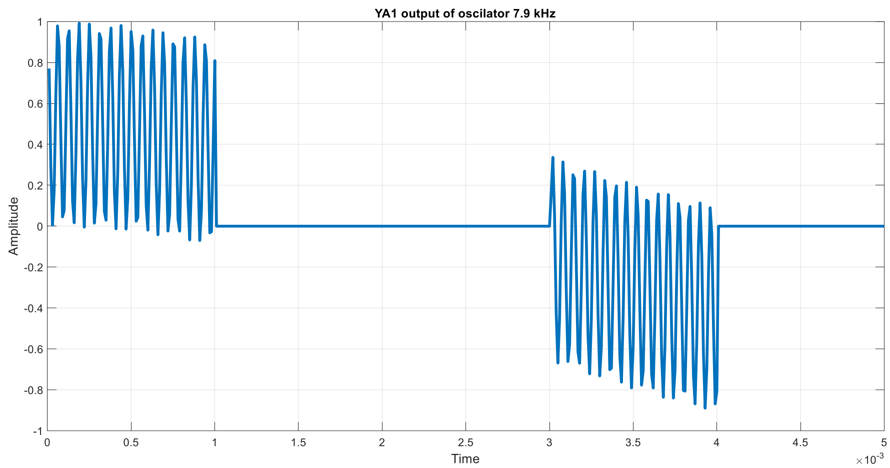
Assuming sampler samples at $1/R_b$ *and Vth = 0.5 :*

At Phase Error = $60° \Longrightarrow \hat{b} = 0\ 0\ 0\ 0\ 0$

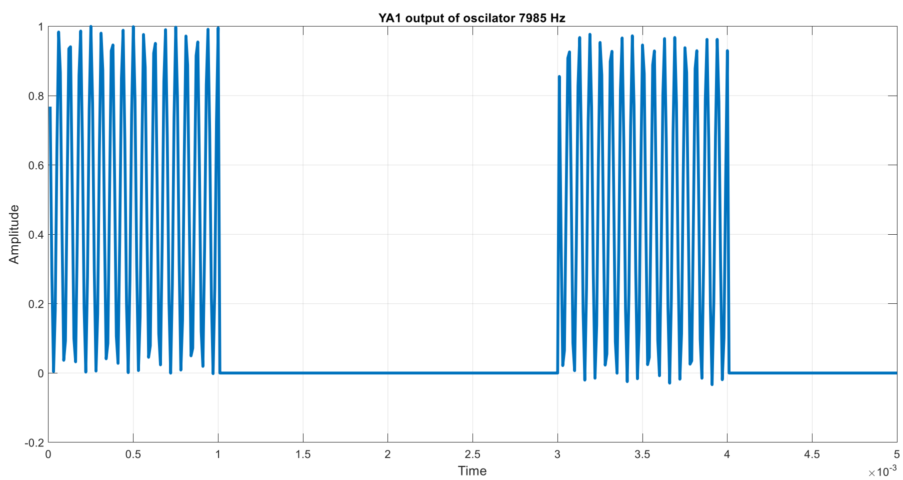
At Phase Error = $120° \Longrightarrow \hat{b} = 0\ 0\ 0\ 0\ 0$

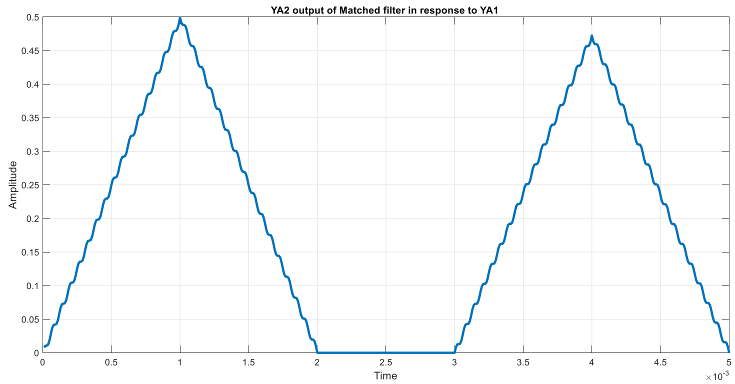
B.5

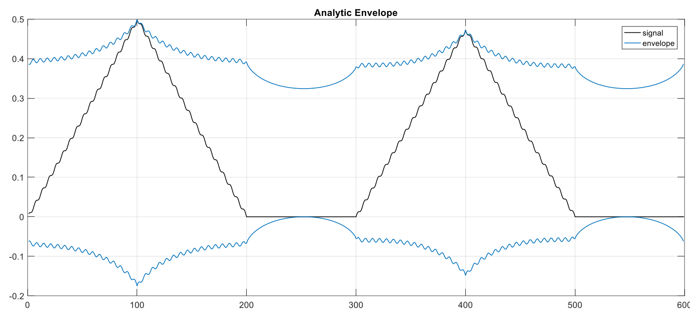




So if we consider Vth = 0.5 all the sequence cannot be decoded correctly.







$frequency\ of\ the\ envelope = f_c - f_0 = 8\ kHz - 7985\ Hz = 15\ Hz$

C. System Performance Under Noise

Coherent Detection

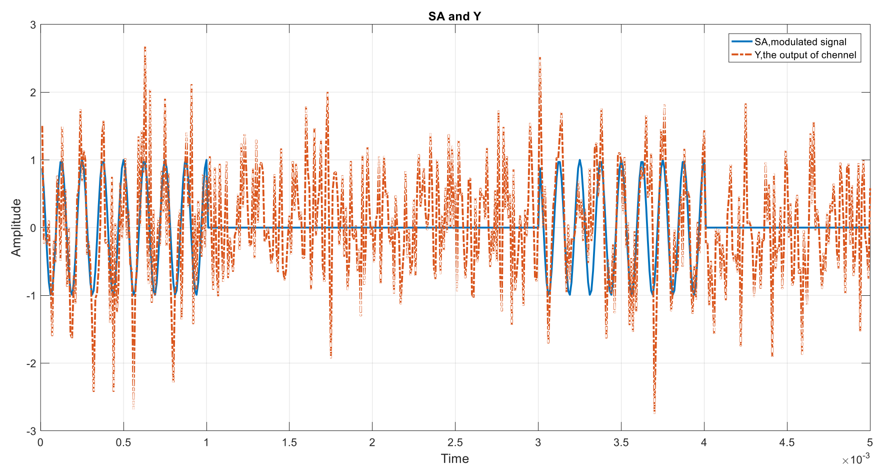
C.1,2,3

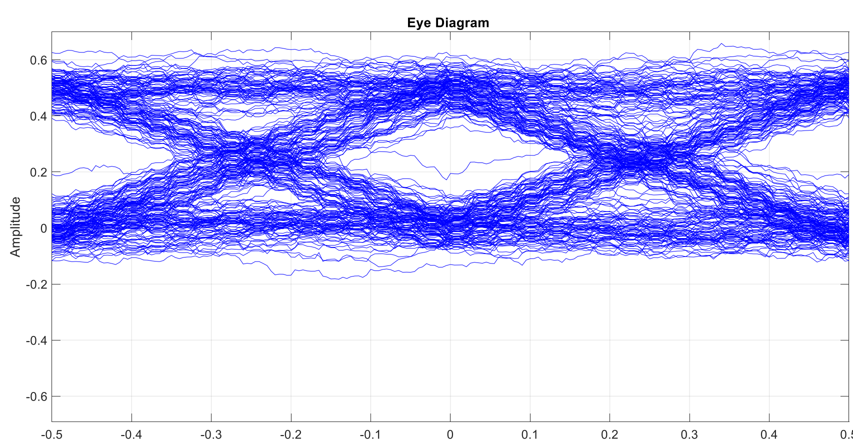
```
clear all
clc
b = [1 0 0 1 0];Rb = 1000;
b(6:500)= randi([0 1],[1 495]);
simulation = 500;
time = 0.00001 : 0.00001 : simulation*(1/Rb);
XA = zeros(1,length(time));
for i = 1:simulation
XA(1+(i-1)*100:i*100)= b(i);
end
SA = XA.*cos(2*pi*((8e+3))*time);
Y = SA + sqrt(0.5)*randn(1,length(SA));
YA = Y.*cos(2*pi*((8e+3))*time);
ZA = conv(0.01*ones(1,100),YA);
eyediagram(ZA,200)
figure
plot(time(1:500),SA(1:500), time(1:500),Y(1:500))
legend('SA,modulated signal','Y,the output of channel')
% sampeling at 1/Rb or at center of eyediagram and Vth = 1/2*(top to down of
open eye) = 0.25
```

```
Z = ZA(100:100:end);% every bit interval includes 100 samples
bb = Z> 0.25;
noe = sum(abs(bb-b));% number of error
BER = noe/simulation
% BER theory
N0 = 2e-5;
EBN0 = (sum(SA.*SA)/(Rb*length(SA)))/N0;
BER_teory = qfunc(sqrt(EBN0))
```

BER = 0

BER_teory = 2.2049e-04





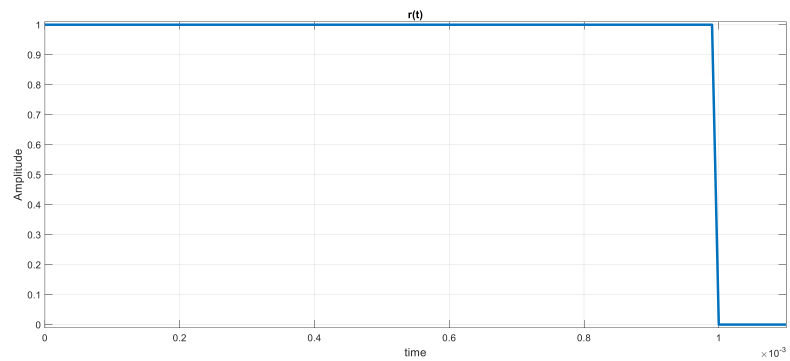
Computer Exercise V: Matched Filter and Bit Error Rate (BER)

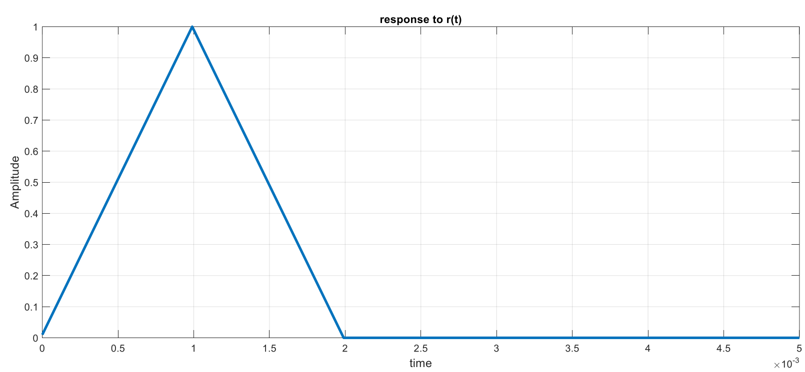
I- Prelab Assignment:

A:

a,b)

```
clear all
clc
simulation = 5;T = 1e-3;
rt = zeros(1,simulation*100);
t = 0.00001:0.00001:simulation*T;
x = 1;% impulse
nsamp = 100;
rt(1:100) = rectpulse(x,nsamp);
plot(t,rt(1:length(t)))
axis([0 0.0011 -0.01 1.01])
figure
out = 0.01*conv(rt,rt);
plot(t,out(1:length(t)))
```

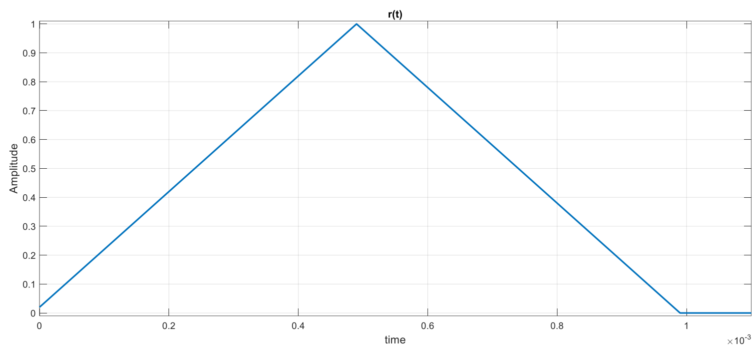


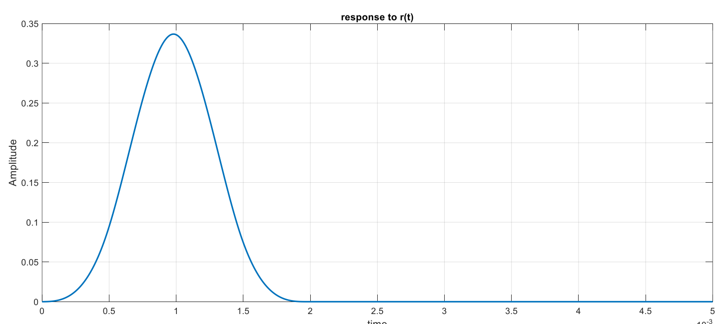


c)

```
clear all
clc
```

```
simulation = 5;T = 1e-3;
rt = zeros(1,1+simulation*100);
t = 0:0.00001:simulation*T;
x = 1;% impulse
nsamp = 100;
rt1 = rectpulse(x,nsamp/2);
plot(t(1:50),rt1)
figure
rt(1:99) = 0.02*conv(rt1,rt1);
plot(t,rt(1:length(t)))
axis([0 0.0011 -0.01 1.01])
figure
out = 1/99*conv(rt,rt);
plot(t,out(1:length(t)))
```





B:

a,b,c)

Assuming ; W = 10 kHz is channel bandwidth

```
clear all
clc
b = [1 0 0 1 0];Rb = 1000;W = 1e+4;
b(6:500)= randi([0 1],[1 495]);
simulation = 500;
time = 0.00001 : 0.00001 : simulation*(1/Rb);
X = zeros(1,length(time));
for i = 1:simulation
```

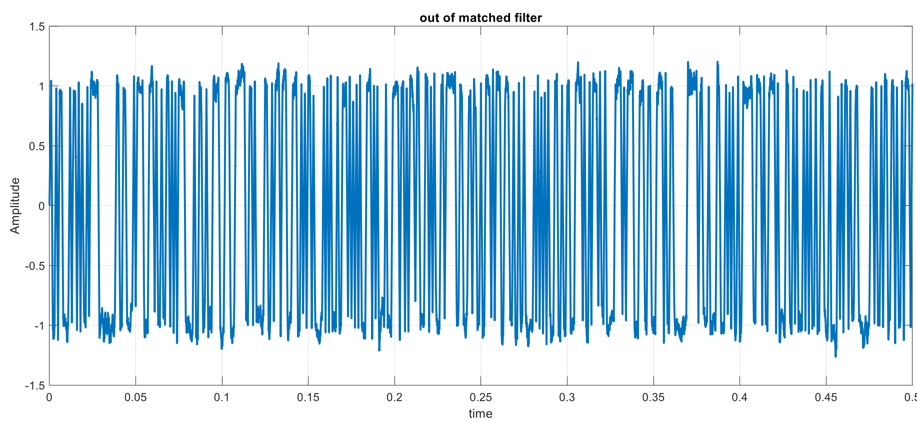
```
X(1+(i-1)*100:i*100)= 2*b(i)-1;
end
T = 1e-3;
Y = X + sqrt(0.5)*randn(1,length(X));
rt = zeros(1,simulation*100);
x = 1;% impulse
nsamp = 100;
rt(1:100) = rectpulse(x,nsamp);
out = 0.01*conv(Y,rt);
plot(time,out(1:length(time)))
rms_noise = sqrt(var(out))
Peak_out_matched_filter = max(abs(out))
the_average_energy_X = sum(X.*X)/(Rb*length(X))
N0 = 1e-4;
EBN0 = the_average_energy_X/N0;
Pe = qfunc(sqrt(2*EBN0))
```

rms_noise = 0.5777

Peak_out_matched_filter = 1.2629

the_average_energy_X = 1.0000e-03

Pe = 3.8721e-06



C :

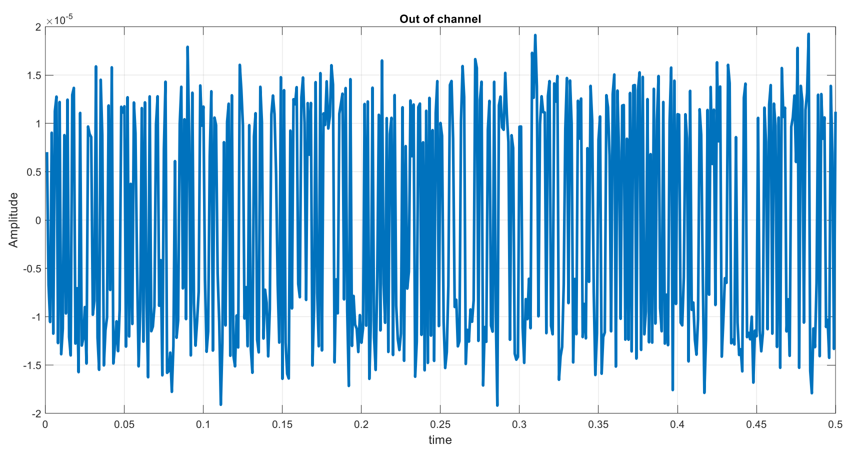
a,b)

$$\frac{1}{1 + j2\pi f RC} \leftrightarrow \frac{1}{RC} e^{-\frac{t}{RC}} u(t) \quad , \qquad \Delta f = 1\ kHz$$

```
clear all
clc
% assuming sampel per symbol = 1
simulation = 1000000;
b = [1 0 0 1 0];Rb = 1000;W = 1e+3;
b(6:simulation)= randi([0 1],[1 simulation-5]);
time = 0.001 : 0.001 : simulation*(1/Rb);
X = zeros(1,length(time));
```

```
for i = 1:simulation
X(1+(i-1)*1:i*1)= 2*b(i)-1;
end
T = 1e-3;
Y = X + sqrt(0.05)*randn(1,length(X));
channel = (2000*pi)*exp(-(2000*pi)*time);
out_channel = (1/length(time))*conv(Y,channel);
receive_signal = (1/length(time))*conv(out_channel,channel);
detected_sig = receive_signal(1:1:simulation*1);
bb = detected_sig > 0 ;
noe = sum(abs(bb-b));
Pe = noe/simulation
plot(time,out_channel(1:length(time)))
rms_noise = sqrt(var(out_channel))
Peak_out_channel = max(abs(out_channel))
```

```
Pe =

   6.0000e-06

rms_noise =

   8.5023e-06

Peak_out_channel =

   2.4881e-05
```

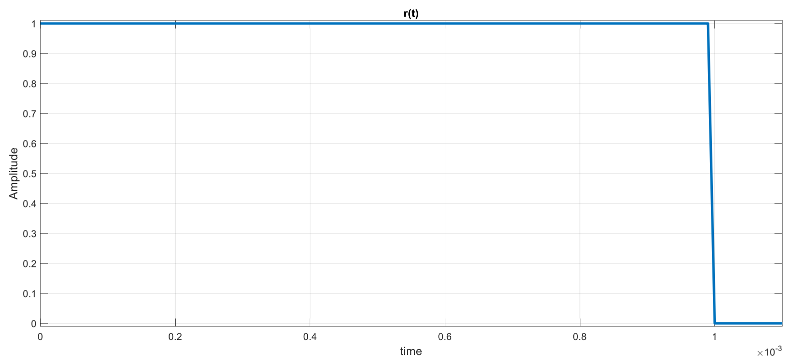


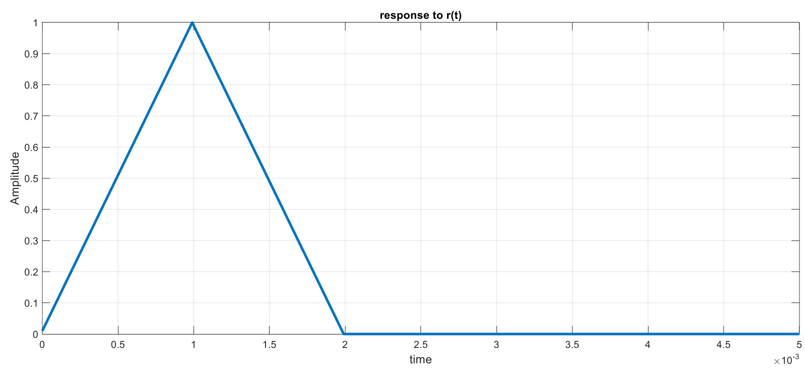
II- Procedure:

A. Characteristics of Matched Filters

A.1 ,2 ,3

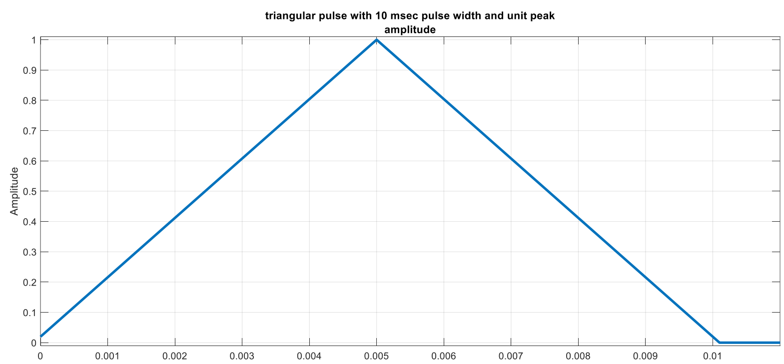
$$r = matched\ filter\ based\ on\ r = rect\left(\frac{t-\frac{T}{2}}{T}\right)\ , T = 1ms$$

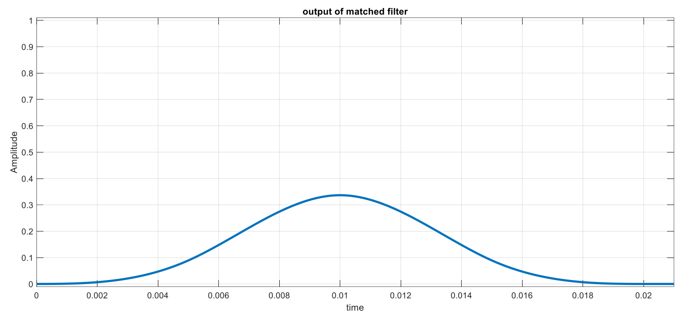




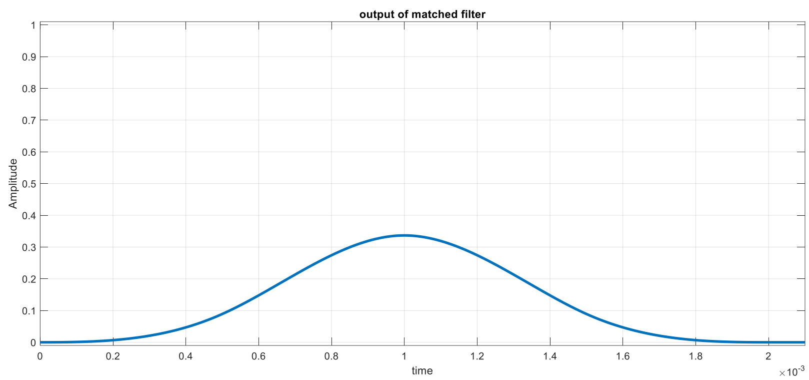
The time when the filter output reaches its maximum value $= T = 1ms$

A.4



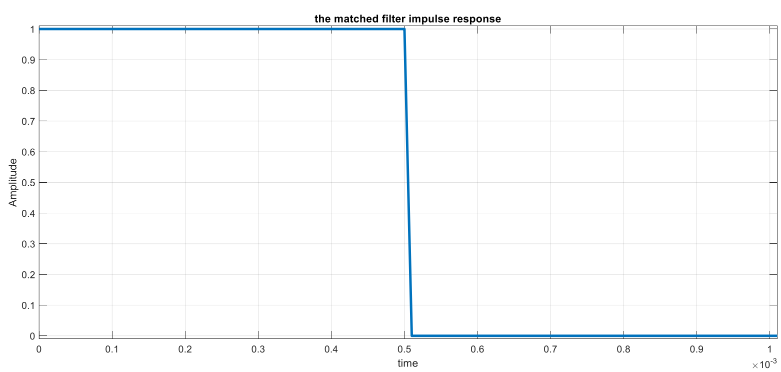


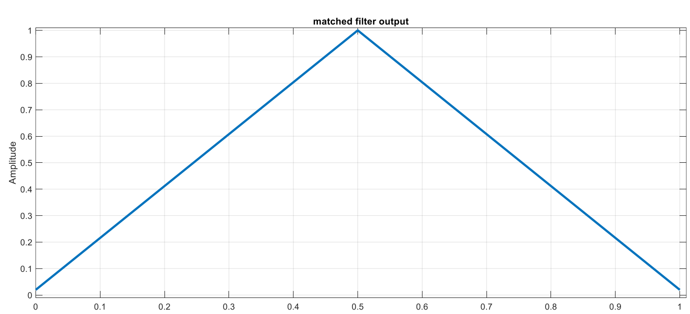
$The\ time\ when\ the\ filter\ output\ reaches\ its\ maximum\ value\ =\ T = 10ms$



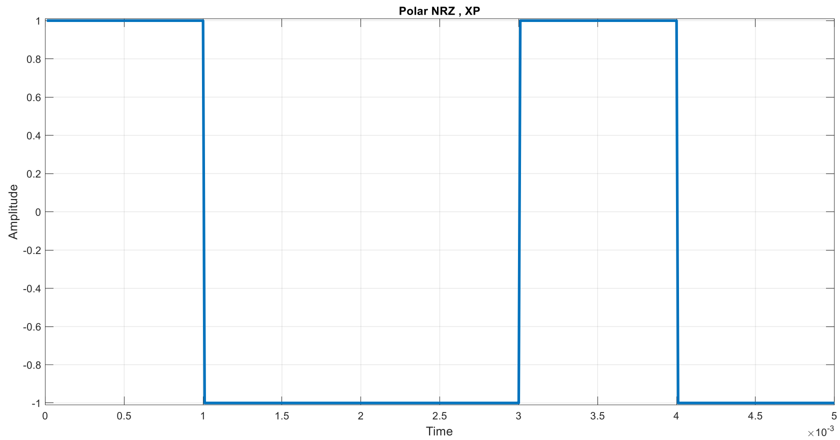
$The\ time\ when\ the\ filter\ output\ reaches\ its\ maximum\ value\ =\ T = 1ms$

A.5



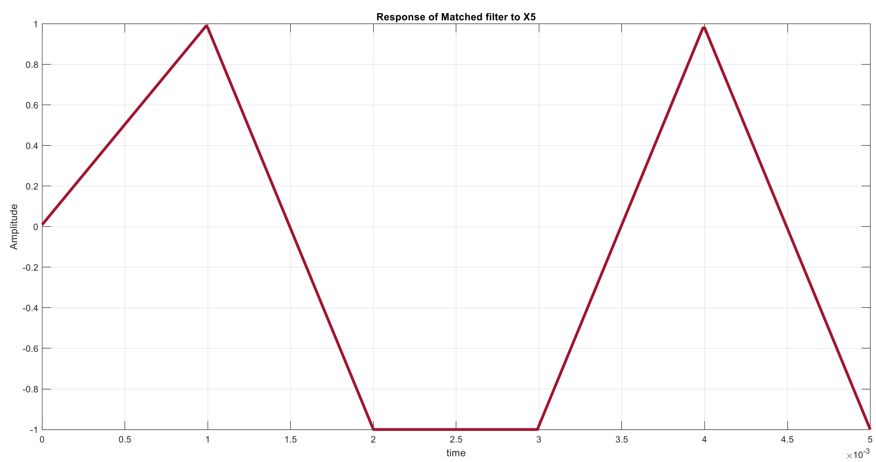


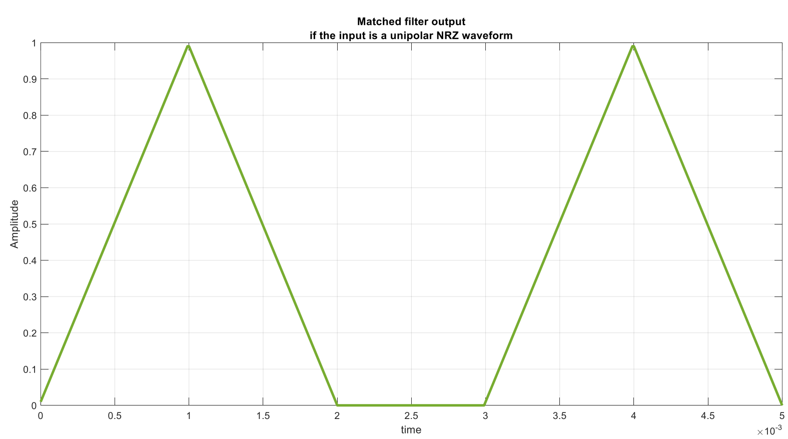
A.6



A.7

Assuming selected matched filter is rect.





B. Signal Detection

B.1 , 2 , 3, 4

$$\frac{1}{1+j2\pi fRC} \leftrightarrow \frac{1}{RC}e^{-\frac{t}{RC}}u(t) \quad , \qquad RC = \frac{1}{2\pi(4900)}, \Delta f = 4.9\ kHz$$

```
clear all
clc
simulation = 10;
b10 = [1 0 0 1 0];Rb = 1000;W = 1e+3;
b10(6:simulation)= randi([0 1],[1 simulation-5]);
time = 0 : 0.00001 : simulation*(1/Rb);
X10 = zeros(1,length(time));
for i = 1:simulation
X10(1+(i-1)*100:i*101)= 2*b10(i)-1;
end
channel = (2*pi*(4900))*exp(-(2*pi*(4900))*time);
out_channel = (1/length(time))*conv(X10,channel);
Y10 = out_channel + sqrt(2/2)*randn(1,length(out_channel));
y10 = Y10(1+100:100:1+simulation*100);
bar_y10 = y10 > 0 ;
noe1 = sum(abs(bar_y10-b10));
Pe1 = noe1/simulation
rt1 = zeros(1,1+simulation*100);
x = 1;% impulse
nsamp = 101;
rt1(1:101) = rectpulse(x,101);
Z10 = (1/length(time))*conv(Y10,rt1);
z10 = Z10(1+100:100:1+simulation*100);
bb = z10 > 0 ;
noe2 = sum(abs(bb-b10));
Pe2 = noe2/simulation
% if sampling instants other than KT
zz10 = Z10(1+100+90:100:1+simulation*100+90);
bb1 = zz10 > 0 ;
noe3 = sum(abs(bb1-b10));
Pe3 = noe3/simulation
plot(time,Y10(1:length(time)))
figure
plot(time,Z10(1:length(time)))
```

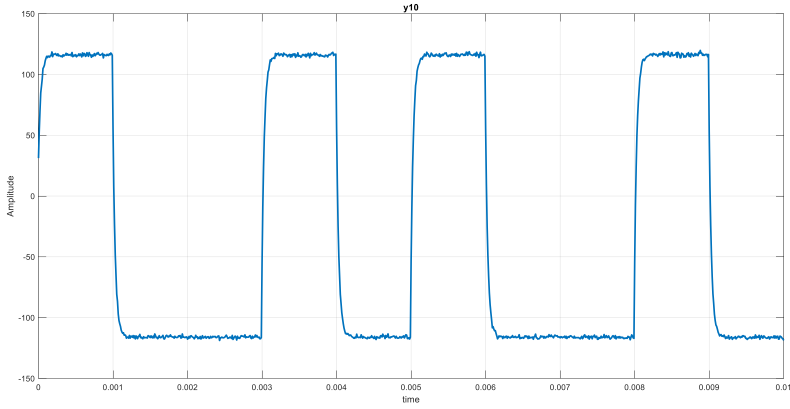
Pe1 = 0

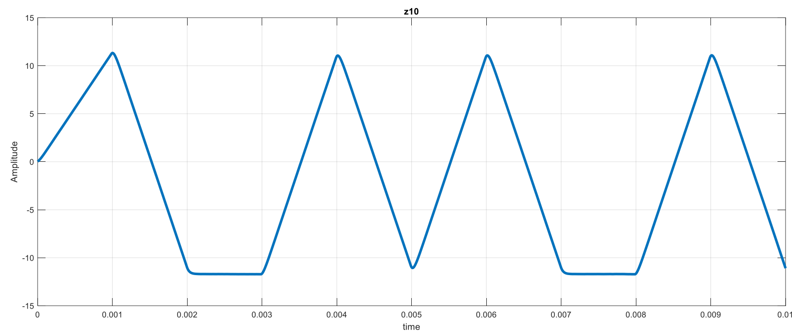
Pe2 = 0

Pe3 = 0.6000

→ *Decoding after Matched filter is easier because Matched filter can degrade the noise effect.*

→ *If sampling instants other than those specified (KT) the probability of error will be larger* (Pe3) *because we are sampeling wrong sampels(next sampels) or correct sampels with amplitude lower than noise.*





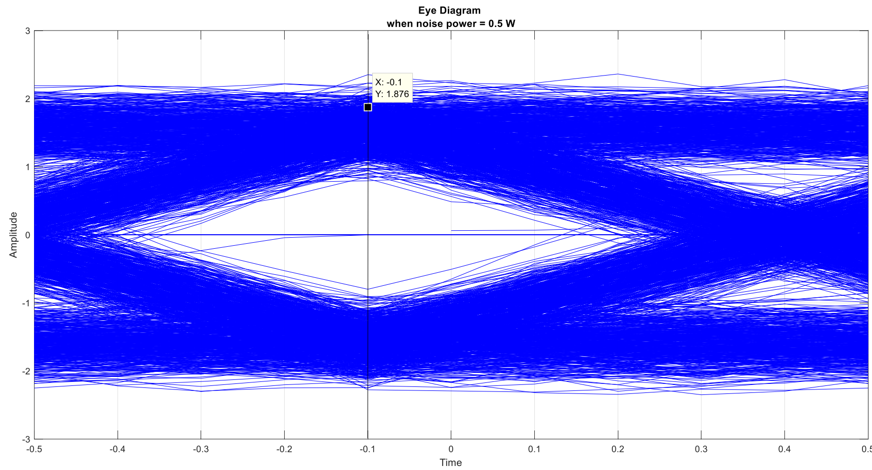
C. Matched-Filter Receiver

C.1 ,2

An example of code for this part just for the case $\sigma^2 = 0.5\ W$ and assumes the sample number = 10 .

```
clear all
clc
simulation = 2000;
nsamp = 10;
b = [1 0 0 1 0];Rb = 1000;W = 1e+3;
b(6:simulation)= randi([0 1],[1 simulation-5]);
time = 0.000001 : 0.0001 : simulation*(1/Rb);
X = zeros(1,length(time));
for i = 1:simulation
X(1+(i-1)*nsamp:i*nsamp)= 2*b(i)-1;
end
channel = (2*pi*(4900))*exp(-(2*pi*(4900))*time);
y = X + sqrt(0.5/2)*randn(1,length(X));
out_channel = (1/length(time))*conv(y,channel);
rt1 = zeros(1,simulation*nsamp);
x = 1;% impulse
rt1(1:nsamp) = rectpulse(x,nsamp);
z = (1/nsamp)*conv(out_channel,rt1);
eyediagram(z,nsamp)
% with help of eyediagram : sampling instant = -0.1*Tb + KTb , V_th = 0
z10 = z(nsamp-1:nsamp:simulation*nsamp-1);
```

```
bb = z10 > 0 ;
noe = sum(abs(bb-b));
Pe_empirical = noe/simulation
% Theory Pe
N0 = 0.5/(4900);
Eb = sum(X.*X)/(Rb*length(X));
EbN0 = Eb/N0;
Pe_theoretical = qfunc(sqrt(2*EbN0))
```



$\boldsymbol{\sigma^2}$ $[\boldsymbol{W}]$	*Pe empirical*	*Pe theoretical*
0.5	0 $(or\ O(10^{-6})$ $when\ at\ least\ 10^6\ bits\ generated\ )$	4.7735e-06
1	0 $(or\ O(10^{-4})$ $when\ at\ least\ 10^4\ bits\ generated\ )$	8.7256e-04
1.5	0.0015	0.0053
2	0.004	0.0134

C.3

Sampling at KTb :

$\boldsymbol{\sigma^2}$ $[\boldsymbol{W}]$	*Pe empirical*
1	0 $(or\ O(10^{-4})$ $when\ at\ least\ 10^4\ bits\ generated\ )$
1.5	0.0015
2	0.0045

C.4

$$\alpha = \frac{Out.Matched\ filter(t = Tb - 0.5(0.1)Tb\ or - 0.9(0.1)Tb)}{Out.Matched\ filter(t = Tb)}$$

$$P_{theoritical} = Q(\sqrt{\frac{2\alpha Eb}{N0}})$$

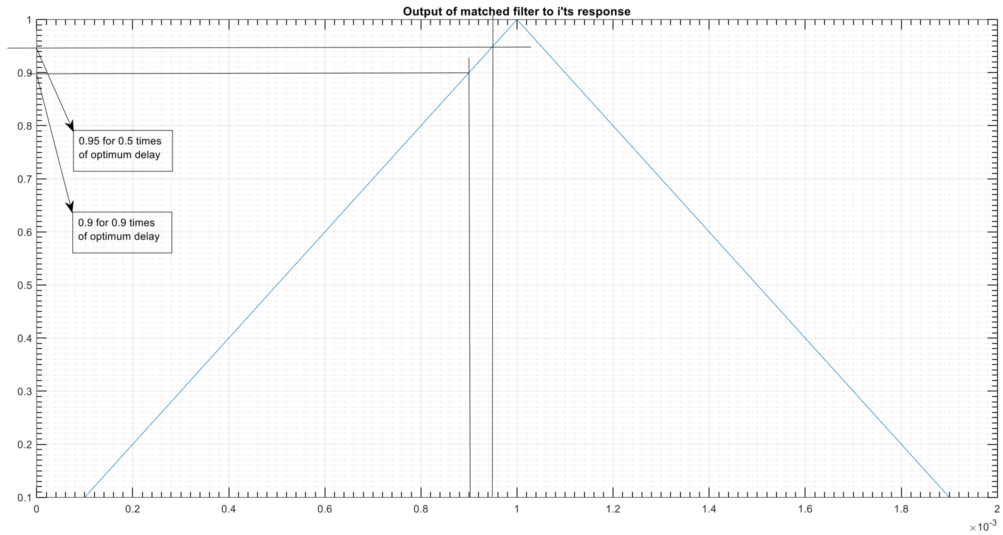
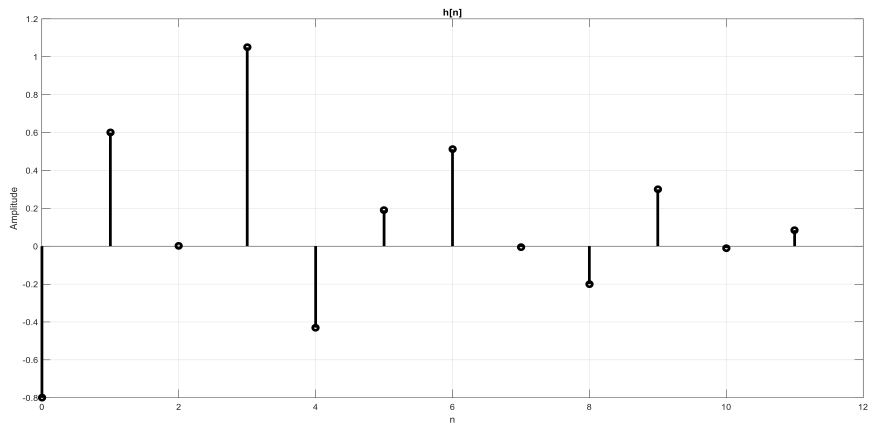


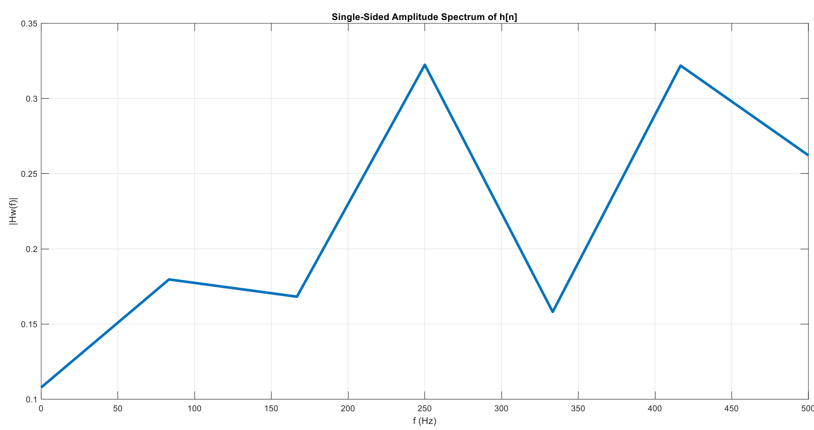
Table 1 : sample instant: $KTb - 0.5(0.1)Tb$

σ^2 $[W]$	*Pe empirical*	*Pe theoretical*
0.5	0 $(or\ O(10^{-6})$ $when\ at\ least\ 10^6\ bits\ generated$)	7.9769e-06
1	0 $(or\ O(10^{-4})$ $when\ at\ least\ 10^4\ bits\ generated$)	0.0011
1.5	0.002	0.0064
2	0.045	0.0155

Computer Exercise VI : Equalization

a)






```
clear all
clc
Fs = 1000;L =11;
h = [-0.8 0.6 0.002 1.05 -0.43 0.19 0.512 -0.005 -0.2 0.3 -0.01 0.085];
n = 0:L;
H = fft(h);
f = Fs*(0:(length(h)/2))/length(h);
P2 = abs(H/(L+1));
Hw = P2(1:(L+1)/2+1);
Hw(2:end-1) = 2*Hw(2:end-1);
plot(n,h)
title('h[n]')
xlabel('n')
ylabel('Amplitude')
figure
```

```
plot(f,Hw)
title('Single-Sided Amplitude Spectrum of h[n]')
xlabel('f (Hz)')
ylabel('|Hw(f)|')
```

b)

Assuming : W = Fs , roll-of-factor = 0 :

$$SNR = \frac{2Eav}{N0}\left(\int_{-W}^{W}\frac{1}{|H(f)|^2}df\right)^{-1}, \qquad N0 = 0.2\,W/Hz$$

$$SNR = 0.0720$$

```
clear all
clc
Fs = 1000;L =11;
h = [-0.8 0.6 0.002 1.05 -0.43 0.19 0.512 -0.005 -0.2 0.3 -0.01 0.085];
n = 0:L;
H = fft(h);
f = Fs*(0:0.01:(length(h)/2))/length(h);
P2 = abs(H/(L+1));
Hw = P2(1:0.01:(L+1)/2+1);
Hw(2:end-1) = 2*Hw(2:end-1);
plot(n,h)
title('h[n]')
xlabel('n')
ylabel('Amplitude')
figure
plot(f,Hw)
title('Single-Sided Amplitude Spectrum of h[n]')
xlabel('f (Hz)')
ylabel('|Hw(f)|')
SNR = 2/0.2*(sum(1./(Hw).^2)*0.01)^-1
```

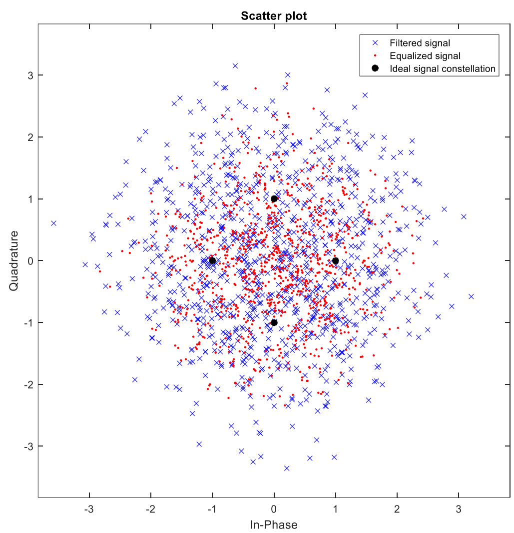
c)

i)

$$\mathrm{Nf} + 1 = 2\mathrm{K} + 1, \boldsymbol{C}_{opt} = \boldsymbol{\Gamma}^{-1}\boldsymbol{\xi}:$$

```
clear all
clc
Fs = 1000;L =11;ntab = 25;
h = [-0.8 0.6 0.002 1.05 -0.43 0.19 0.512 -0.005 -0.2 0.3 -0.01 0.085];
n = 0:L;
H = fft(h);
f = Fs*(0:0.01:(length(h)/2))/length(h);
P2 = abs(H/(L+1));
Hw = P2(1:0.01:(L+1)/2+1);
```

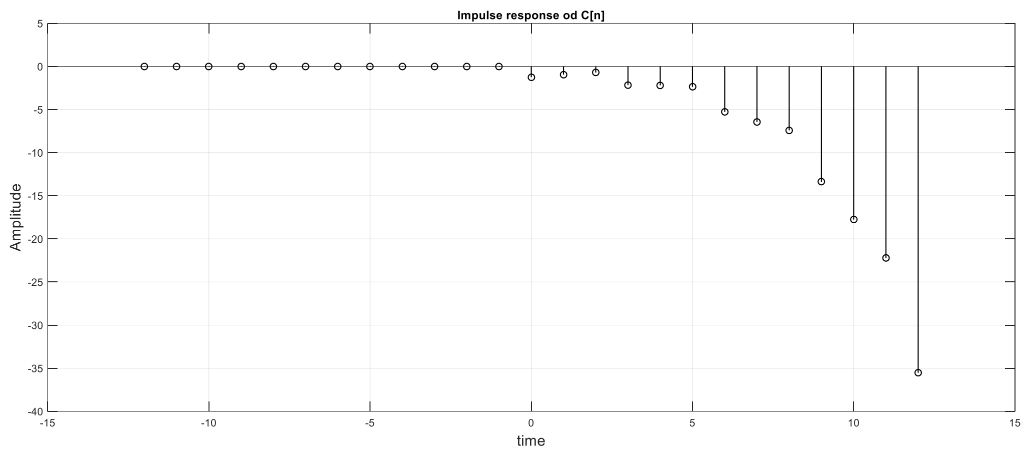
```
Hw(2:end-1) = 2*Hw(2:end-1);
% peak distortion criterion for ZFE EQ.
c = [ h zeros(1,13)];
g = [h(1) zeros(1,24)];
Gama = toeplitz(c,g);
zeta = [zeros(1,12) 1 zeros(1,12)]';
Copt = inv(Gama)*zeta;% ZFE EQ. coefficients
% Set up parameters and signals.
M = 4; % Alphabet size for modulation
msg = randint(1500,1,M); % Random message
modmsg = pskmod(msg,M); % Modulate using QPSK.
trainlen = 500; % Length of training sequence
filtmsg = filter(h,1,modmsg); % Introduce channel distortion
constellation = pskmod([0:M-1],M); % Set signal constellation.
symbolest = 1/norm(Copt)*conv(Copt,filtmsg);% Equalize.
% Plot signals.
hh = scatterplot(filtmsg,1,trainlen,'bx'); hold on;
scatterplot(symbolest,1,trainlen,'r.',hh);
scatterplot(constellation,1,0,'k*',hh);
legend('Filtered signal','Equalized signal',...
'Ideal signal constellation');
hold off;
```

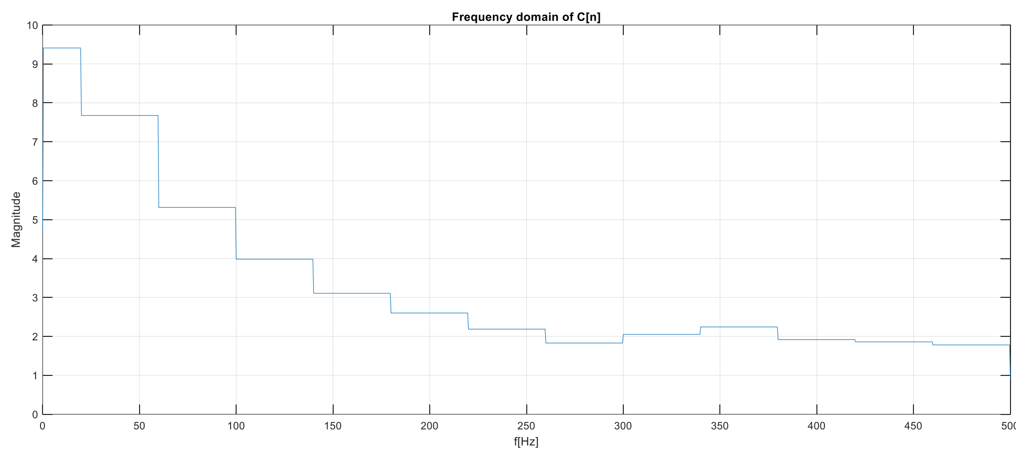



```
nn = -(ntab-1)/2:(ntab-1)/2;
plot(nn,Copt)
figure
FCopt = fft(Copt);
ff = Fs*(0:0.01:(length(Copt)/2))/length(Copt);
PP2 = abs(FCopt/(ntab));
HwCopt = PP2(1:0.01:(ntab)/2+1);
HwCopt(2:end-1) = 2*HwCopt(2:end-1);
plot(ff,HwCopt)
```

By attending to q = conv(Copt,h)

$$SNR = \frac{2Eb}{N0(\sum q[m \neq 0])}(1 - \sum q[m \neq 0])^2 = 10 \ \times 153$$

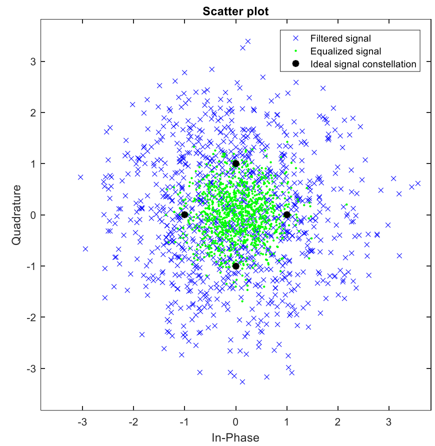


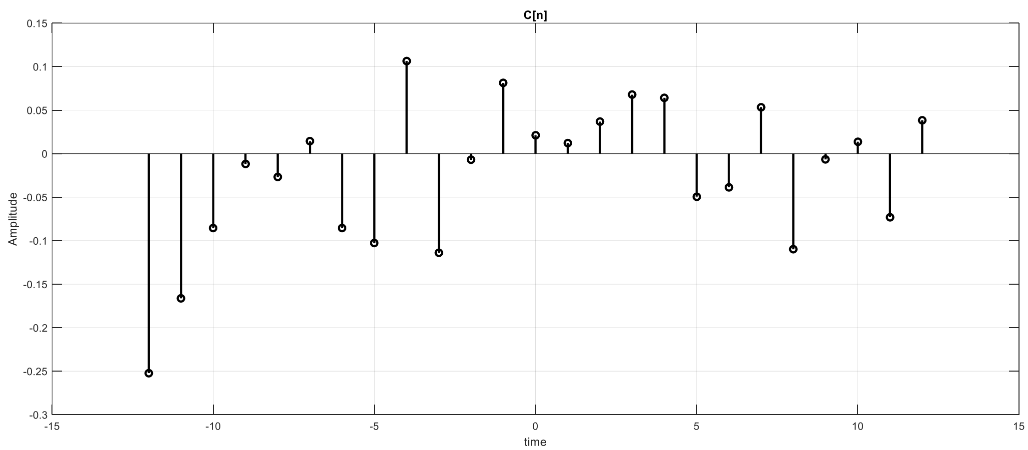


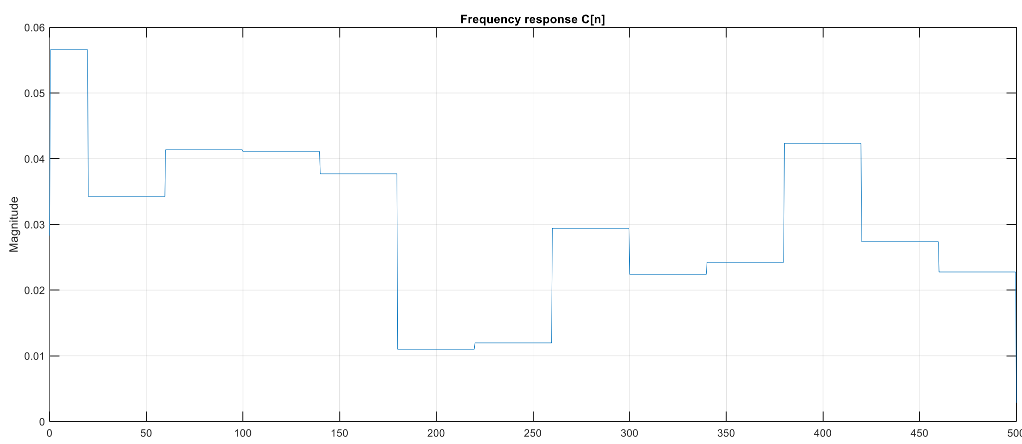
ii)

```
clear all
clc
Fs = 1000;L =11;ntab = 25;
h = [-0.8 0.6 0.002 1.05 -0.43 0.19 0.512 -0.005 -0.2 0.3 -0.01 0.085];
% Set up parameters and signals.
M = 4; % Alphabet size for modulation
msg = randint(1500,1,M); % Random message
modmsg = pskmod(msg,M); % Modulate using QPSK.
trainlen = 500; % Length of training sequence
filtmsg = filter(h,1,modmsg); % Introduce channel distortion.
% Equalize the received signal.
[e,W] = lms1(0.01,ntab,filtmsg,modmsg);% Create an equalizer object.
constellation = pskmod([0:M-1],M); % Set signal constellation.
[symbolest] = conv(W,filtmsg); % Equalize.
```

```
% Plot signals.
h = scatterplot(filtmsg,1,trainlen,'bx'); hold on;
scatterplot(symbolest,1,trainlen,'g.',h);
scatterplot(constellation,1,0,'k*',h);
legend('Filtered signal','Equalized signal',...
'Ideal signal constellation');
hold off
nn = -(ntab-1)/2:(ntab-1)/2;
figure
plot(nn,W)
figure
Feq1 = fft(W);
ff = Fs*(0:0.01:(length(W)/2))/length(W);
PP2 = abs(Feq1/(ntab));
Hweq1 = PP2(1:0.01:(ntab)/2+1);
Hweq1(2:end-1) = 2*Hweq1(2:end-1);
plot(ff,Hweq1)

function [e,w]=lms1(mu,M,u,d)
% Call:
% [e,w]=lms(mu,M,u,d);
%
% Input arguments:
% mu = step size, dim 1x1
% M = filter length, dim 1x1
% u = input signal, dim Nx1
% d = desired signal, dim Nx1
%
% Output arguments:
% e = estimation error, dim Nx1
% w = final filter coefficients, dim Mx1
%initial values: 0
w=zeros(M,1);
%number of samples of the input signal
N=length(u);
%Make sure that u and d are column vectors
u=u(:);
d=d(:);
%LMS
for n=M:N
uvec=u(n:-1:n-M+1);
e(n)=d(n)-w'*uvec;
w=w+mu*uvec*conj(e(n));
end
e=e(:);
```







iii)

```
clear all
clc
```

```
Fs = 1000;L =11;ntabfor = 25;ntabback = 3;
h = [-0.8 0.6 0.002 1.05 -0.43 0.19 0.512 -0.005 -0.2 0.3 -0.01 0.085];
% Set up parameters and signals.
M = 4; % Alphabet size for modulation
msg = randint(1500,1,M); % Random message
modmsg = pskmod(msg,M); % Modulate using QPSK.
trainlen = 500; % Length of training sequence
filtmsg = filter(h,1,modmsg); % Introduce channel distortion.
% Equalize the received signal.
eq1 = dfe(ntabfor,ntabback,rls(0.3)); % DFE
eq1.SigConst = pskmod([0:M-1],M); % Set signal constellation.
[symbolest,yd] = equalize(eq1,filtmsg,modmsg(1:trainlen)); % Equalize.
% Plot signals.
h = scatterplot(filtmsg,1,trainlen,'bx'); hold on;
scatterplot(symbolest,1,trainlen,'g.',h);
scatterplot(eq1.SigConst,1,0,'k*',h);
legend('Filtered signal','Equalized signal',...
'Ideal signal constellation');
hold off;
```