Communication Assignment

Report

Name: Sarah Mohamed Ahmed Lotfy

Sec. 1 - Bn.24

Part 2

4) BER an decreasing function of E/No.

Because BER decreases as ratio E/No increases, so that eventually a very "small increase" in transmitted signal energy will make the reception of binary pulses almost error free.

As in the figures in 1st Case and 2nd Case.

And also because of this function

$$P_e < \frac{\exp(-E_b/N_0)}{2\sqrt{\pi E_b/N_0}}$$

5) First Case has lowest Bit error rate.

Because in 1st case the matched filter h(t) = g(T-t) which decrease error and makes SNR maximum.

Because When h(t) = g(T-t) is condition to get maximum SNR.

But in other cases matched filter h(t) = 1 in 2nd case

in 3rd case h(t) = $\sqrt{3}$ t.

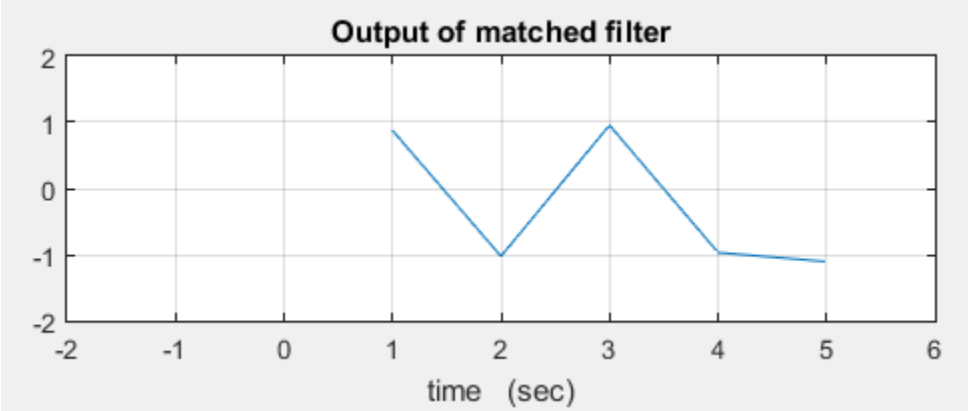
Figures

1st Case :

[1] I plotted the output of received filter for

5 random input bits.

snr_db = 20 db



With Snr_db = 0

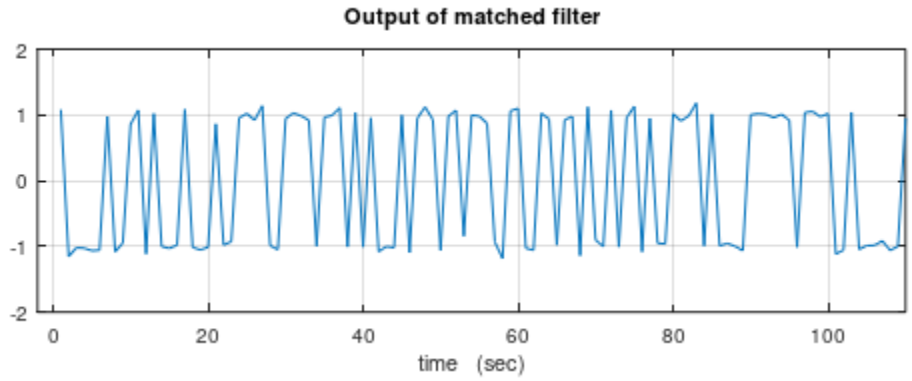
Probability of error = 0.06

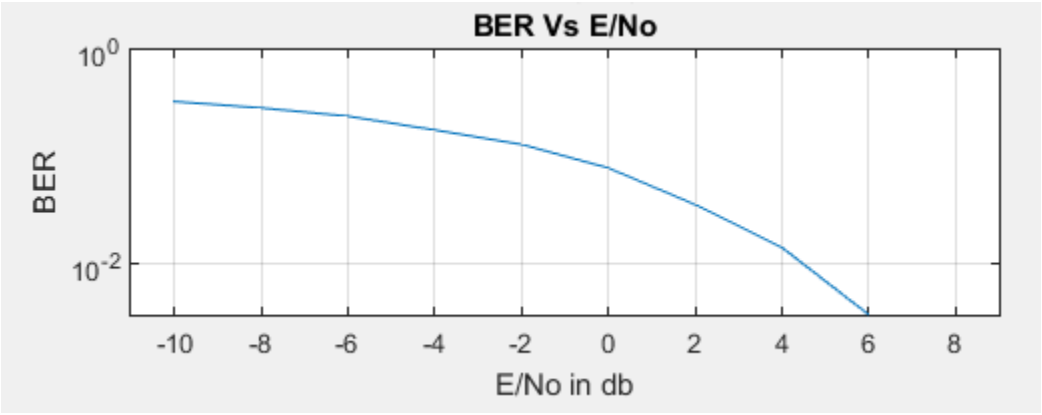
With Snr_db = 20

Probability of error = 0

[2] Output of received filte and BER vs E/No for 10000 bits

(1st case)



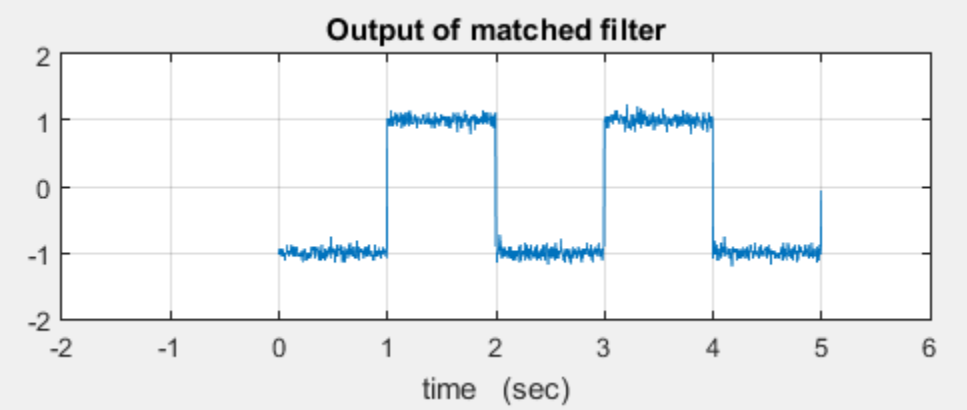


2nd Case

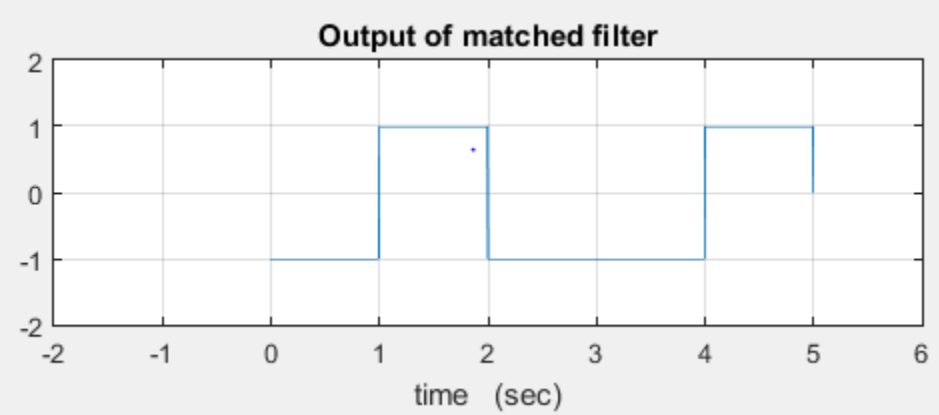
[1] I plotted the output of received filter for

5 random input bits.

snr_db = 20 db



With no noise

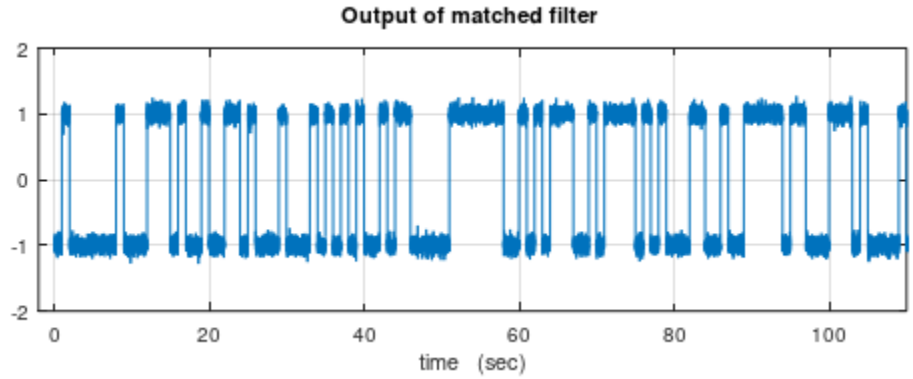


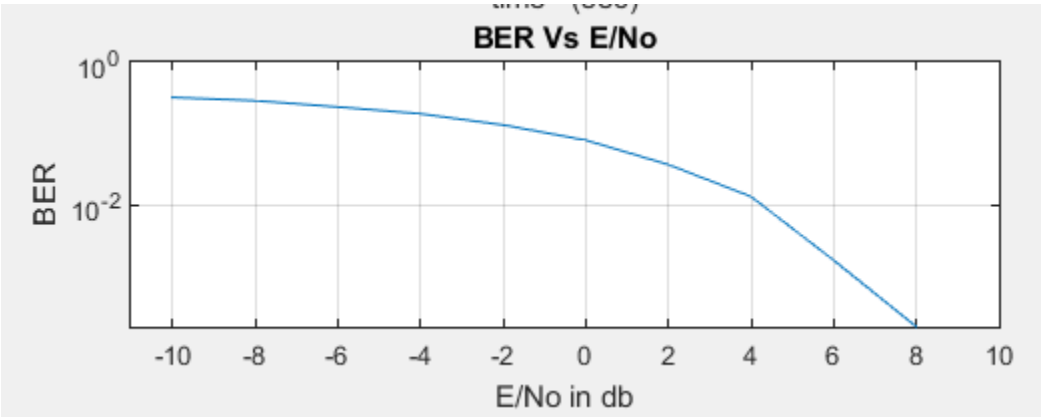
With Snr_db = 0 -----> Probability of error = 0.2

With Snr_db = 20 -----> Probability of error = 0

[2] Output of received filter and BER vs E/No for **10000** bits

(2nd case)



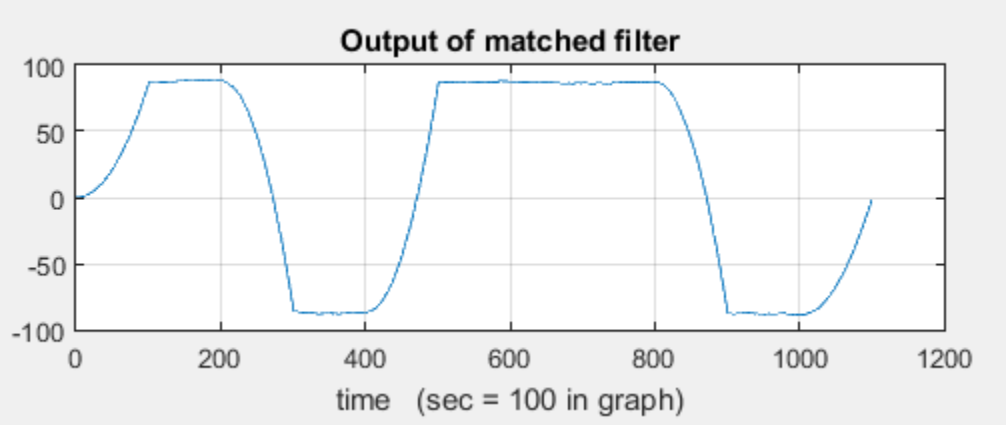


3rd Case

[1] I plotted the output of received filter for

5 random input bits.

snr_db = 20 db

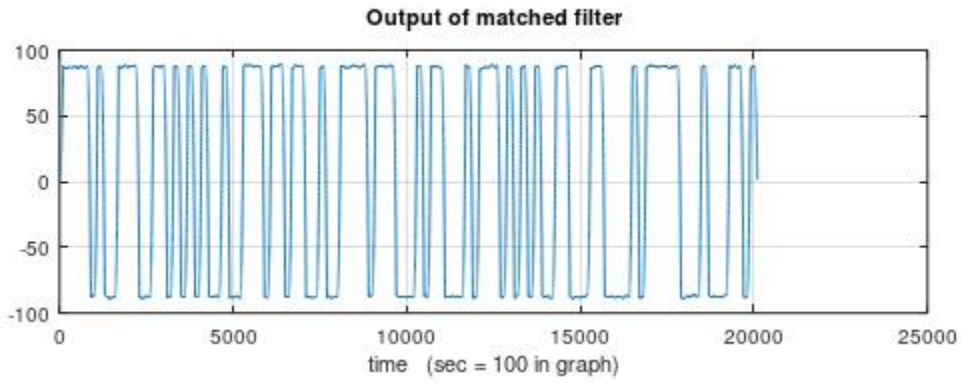


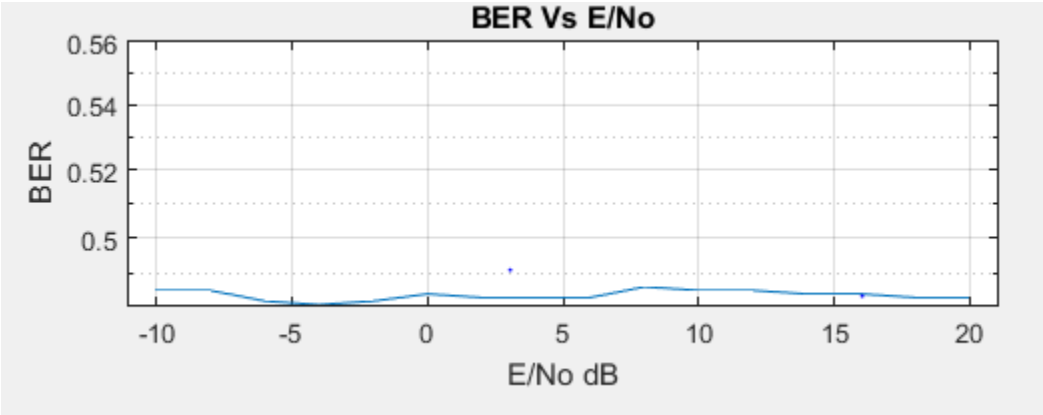
With Snr_db = 0 -----> Probability of error = 0.49

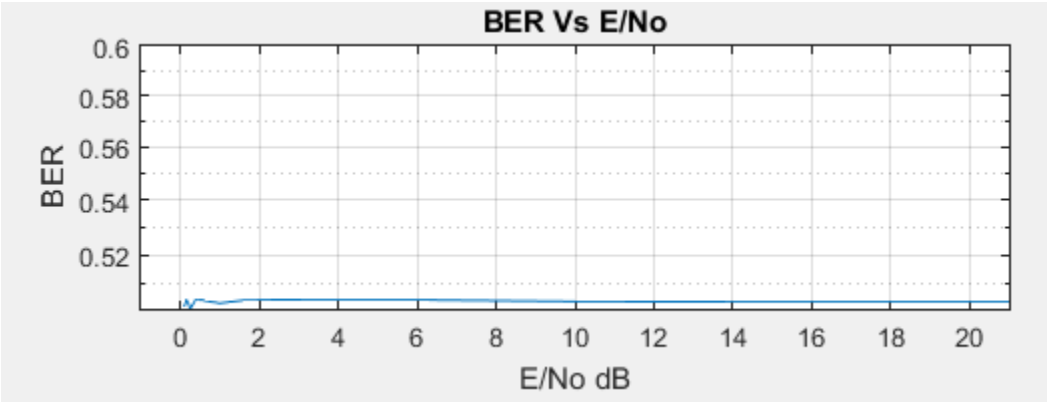
With Snr_db = 20 -----> Probability of error = 0.5

[2] Output of received filter and BER vs E/No for 10000 bits

(3rd case)







Matlab Code

1st Case

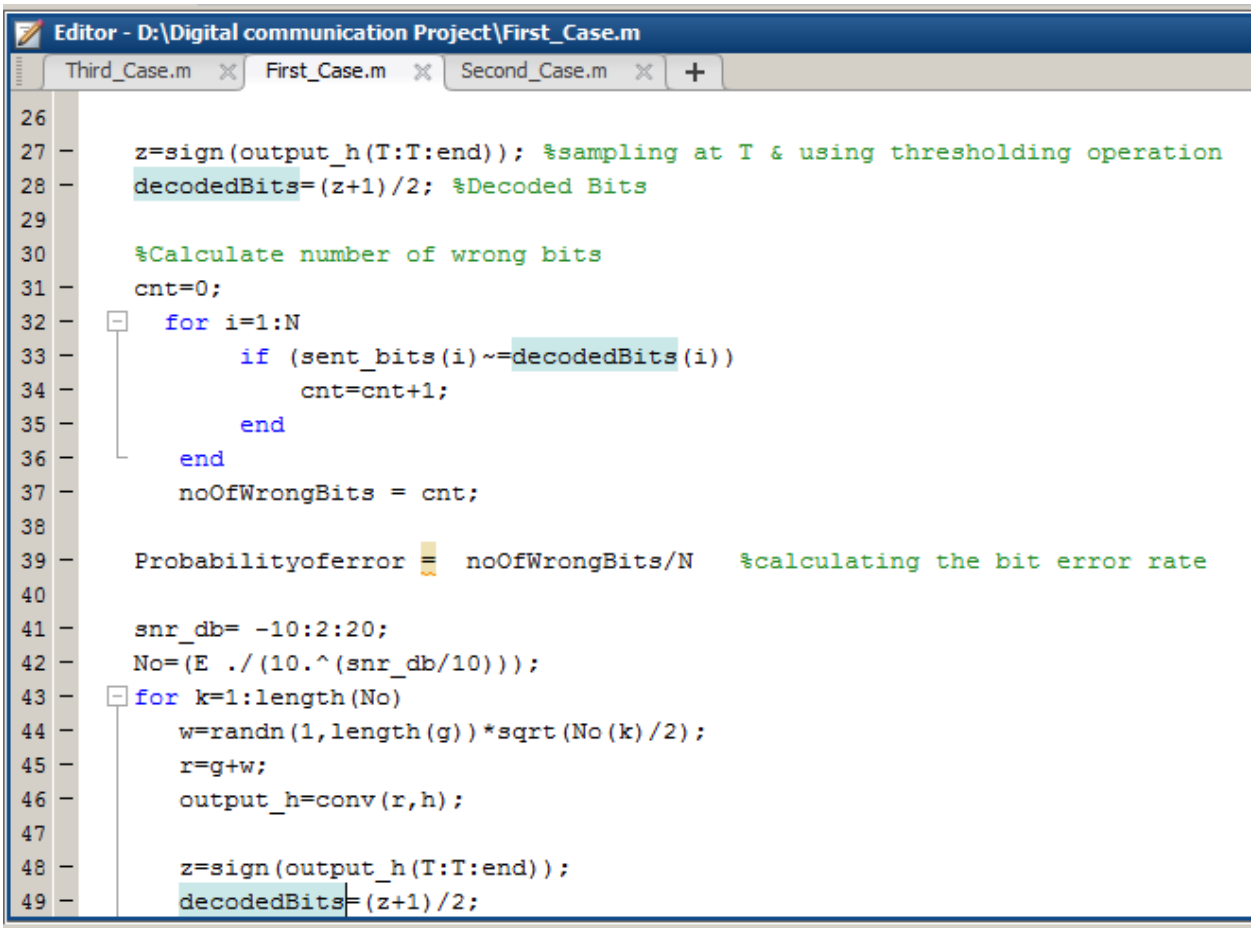
Editor - D:\Digital communication Project\First_Case.m

Third_Case.m | First_Case.m | Second_Case.m

```
N=10000; % Number of input bits
sent_bits=randi([0,1],1,N); %Input Bit Stream
A=1; %amplitude of S(t)
T=1; %duration of S(t)
S = ones(1,T)*A; %rectangular pulse
E = norm(S)^2; %S(t) Energy
h=fliplr(S); %matched filter
bits=(2*sent_bits-1);
g=kron(bits,S);  %Signal that represents bit stream


snr_db= 20;
No=(E ./(10.^(snr_db/10)));
w=randn(1,length(g))*sqrt(No/2); %add noise
r=g+w; %received signal
output_h=conv(r,h);   %matched filter for r by convolution

%Plot output of the receive filter at snr_db = 20  for 5 bits
subplot(2,1,1);
plot(output_h);
axis([-2 110 -2 2]);
grid on;
title('Output of matched filter');
xlabel('time    (sec)')
```


Editor - D:\Digital communication Project\First_Case.m
Third_Case.m
First_Case.m
Second_Case.m
z=sign(output_h(T:T:end)); %sampling at T & using thresholding operation
decodedBits=(z+1)/2; %Decoded Bits
%Calculate number of wrong bits
cnt=0;
for i=1:N
if (sent_bits(i)~=decodedBits(i))
cnt=cnt+1;
end
end
noOfWrongBits = cnt;
Probabilityoferror = noOfWrongBits/N %calculating the bit error rate
snr_db= -10:2:20;
No=(E ./(10.^(snr_db/10)));
for k=1:length(No)
w=randn(1,length(g))*sqrt(No(k)/2);
r=g+w;
output_h=conv(r,h);
z=sign(output_h(T:T:end));
decodedBits=(z+1)/2;

Editor - D:\Digital communication Project\First_Case.m

Third_Case.m | First_Case.m | Second_Case.m

```
43 -  for k=1:length(No)
44 -      w=randn(1,length(g))*sqrt(No(k)/2);
45 -      r=g+w;
46 -      output_h=conv(r,h);
47 -
48 -      z=sign(output_h(T:T:end));
49 -      decodedBits=(z+1)/2;
50 -      cnt=0;
51 -      for i=1:N
52 -          if (sent_bits(i)~=decodedBits(i))
53 -              cnt=cnt+1;
54 -          end
55 -      end
56 -      noOfWrongBits = cnt;
57 -      BER(k)= (noOfWrongBits/N);  %Bit error rate
58 -  end
59
60      %plotting BER Vs E/No
61 -    subplot(2,1,2)
62 -    semilogy(snr_db,BER)
63 -    title ('BER Vs E/No');
64 -    xlabel ('E/No in db');
65 -    ylabel ('BER');
66 -    grid on;
```

2nd Case

Editor - D:\Digital communication Project\Second_Case.m

Third_Case.m | First_Case.m | Second_Case.m

```
%%%   Second Case  %%%
clc;
clear;
numberOfBits=100;  % Number of input bits
inputBits=randi([0,1],1,numberOfBits); %Input Bit Stream

%Rectangular pulse g
len2=numberOfBits ;
n2=200;
N2=n2*len2;
dt=len2/N2;
t2=0:dt:len2;
g_org=zeros(1,length(t2));
for i=0:len2-1;
     g_org(i*n2+1 : (i+1)*n2)=1;
end;

%Generate g which is pulse shaping represents input bitstream
len=numberOfBits;
n=200;
N=n*len;
dt=len/N;
t1=0:dt:len;
g=zeros(1,length(t1));
for i=0:len-1;
    if inputBits(i+1)==1
```

Editor - D:\Digital communication Project\Second_Case.m

Third_Case.m | First_Case.m | Second_Case.m

```
25 -    for i=0:len-1;
26 -        if inputBits(i+1)==1
27 -            g(i*n+1 : (i+1)*n)=1;
28 -        else
29 -            g(i*n+1 : (i+1)*n)=-1;
30 -        end;
31 -    end;
32
33      %Addingg noise at x =20
34 -    E = 1;
35 -    x = 20;
36 -    No=(E ./(10.^(x/10)));
37 -    r = g + sqrt(No/2)* randn(1, length(g));
38
39      %Matched filter
40 -    h=1;
41
42 -    output_h = conv(g,h);%matched filter for r by convolution
43
44      %Plot output of the receive filter
45 -    len4=numberOfBits;
46 -    n4=200;
47 -    N4=n4*len4;
48 -    dt=len4/N4;
49 -    t4=0:dt:len4;
```

Editor - D:\Digital communication Project\Second_Case.m

Third_Case.m | First_Case.m | Second_Case.m

```
49 -    t4=0:dt:len4;
50 -    subplot(2,1,1);
51 -    plot(t4,output_h);
52 -    axis([-2 110 -2 2]);
53 -    grid on;
54 -    title('Output of matched filter');
55 -    xlabel('time   (sec)')
56
57 -    output_h = conv(r,h);
58      %sampling at T & using thresholding operation
59 -    T = 1;
60 -    z=sign(output_h(T:end/numberOfBits:end));
61 -    decoded_bits=(z+1)/2; %Decoding the sent bits
62      %Calculate probability of error
63 -    cnt=0;
64 -      for i=1:numberOfBits
65 -           if (inputBits(i)~=decoded_bits(i))
66 -               cnt=cnt+1;
67 -           end
68 -       end
69 -       noOfWrongBits = cnt;
70 -    Probabilityoferror =  noOfWrongBits/numberOfBits  %calculating the bit error rate
71
72
73      %%%   Graph between BER nand  E/No   %%%
74 -    x= -10:2:20;
```

Editor - D:\Digital communication Project\Second_Case.m

Third_Case.m | First_Case.m | Second_Case.m

```
72
73      %%%   Graph between BER nand  E/No   %%%
74 -    snr_db= -10:2:20;
75 -    E = 1;
76 -    No=(E ./(10.^(snr_db/10)));
77 -    nError = zeros(1,numberOfBits());
78 -  for k=1:length(No)
79 -       r = g + sqrt(No(k)/2)* randn(1, length(g));
80 -       output_h = conv( r, h);
81        %sampling at T & using thresholding operation
82 -       T = 1;
83 -       z=sign(output_h(T:end/numberOfBits:end));
84 -       decoded_bits=(z+1)/2; %Decoding the sent bits
85        %Calculate probability of error
86 -       cnt=0;
87 -       for i=1:numberOfBits
88 -           if (inputBits(i)~=decoded_bits(i))
89 -               cnt=cnt+1;
90 -           end
91 -       end
92 -       noOfWrongBits = cnt;
93 -        BER(k)= noOfWrongBits/numberOfBits; %calculating the bit error
94 -   end
```

```
84
85      %plotting BER Vs E/No
86 -    subplot(2,1,2)
87 -    semilogy(snr_db,BER)
88 -    title ('BER Vs E/No');
89 -    xlabel ('E/No in db');
90 -    ylabel ('BER');
91 -    grid on;
92
```

3rd Case

Editor - D:\Digital communication Project\Third_Case.m

Third_Case.m | First_Case.m | Second_Case.m

```
N=1000;    % Number of input bits
sent_bits=randi([0,1],1,N); %Input Bit Stream
A=1; %amplitude of S(t)
T=1; %duration of S(t)
S=ones(1,T)*A; %rectangular pulse
E=norm(S)^2; %S(t) Energy

% g signal represent bit stream
len=N;
n=200;
dt=(n*len)/N;
t1=0:dt:len;
g=zeros(1,length(t1));
for i=0:len-1;
    if sent_bits(i+1)==1
        g(i*n+1 : (i+1)*n)=1;
    else
        g(i*n+1 : (i+1)*n)=-1;
    end;
end;

t = 0:0.01:1;
h = (sqrt(3))* t; %matched filter

```

Editor - D:\Digital communication Project\Third_Case.m

Third_Case.m | First_Case.m | Second_Case.m

```
25 -    snr_db=20 ;
26 -    E = 1;
27 -    No=(E ./(10.^(snr_db/10)));
28 -    w=randn(1,length(g))*sqrt(No/2); %add noise
29 -    r = g+w; %received signal
30 -    output_h=conv(r,h); %matched filter for r
31
32      %plotting output of matched filter
33 -    subplot(2,1,1);
34 -    plot(output_h);
35 -    grid on;
36 -    title('Output of matched filter');
37 -    xlabel('time   (sec = 100 in graph)') % 2*10^4 = 200
38
39 -    z=sign(output_h(T:T:end)); %sampling at T & using thresholding operation
40 -    decodedBits=(z+1)/2; %decoding the sent bits
41 -    length(decodedBits)
42 -     cnt=0;
43 -      for i=1:N
44 -           if (sent_bits(i)~=decodedBits(i))
45 -               cnt=cnt+1;
46 -           end
47 -       end
48 -       noOfWrongBits = cnt;
```

Editor - D:\Digital communication Project\Third_Case.m

Third_Case.m | First_Case.m | Second_Case.m

```
48 -        noOfWrongBits = cnt;
49 -     Probabilityoferror =  noOfWrongBits/N  %calculating the bit error
50 -
51 -
52 -     snr_db= -10:2:20;
53 -     E = 1;
54 -     No=(E ./(10.^(snr_db/10)));
55 -     for k=1:length(No)
56 -        w=randn(1,length(g))*sqrt(No(k)/2);
57 -        r=g+w;
58 -        output_h=conv(r,h);
59 -
60 -       z=sign(output_h(T:T:end));
61 -       decodedBits=(z+1)/2;
62 -       cnt=0;
63 -       for i=1:N
64 -            if (sent_bits(i)~=decodedBits(i))
65 -                cnt=cnt+1;
66 -            end
67 -        end
68 -        noOfWrongBits = cnt;
69 -        BER(k)= noOfWrongBits/N;
70 -     end
71 -
```

```
65 -                cnt=cnt+1;
66 -            end
67 -        end
68 -        noOfWrongBits = cnt;
69 -        BER(k)= noOfWrongBits/N;
70 -    end
71 -
72 -    %plotting BER Vs E/No
73 -    subplot(2,1,2)
74 -    semilogy(snr_db,BER)
75 -    axis([-11 21 -3 1]);
76 -    title ('BER Vs E/No');
77 -    xlabel ('E/No dB');
78 -    ylabel ('BER');
79 -    grid on;
```