Name: Cemal Güven Adal

ID:21703986

EE 431 Project 2 Report

Part A

In this part Tb is chosen as 1 second , sampling period is chosen as 1/20 sec and sample size is chosen as 100000. Message signal is created by randomly creating a binary array consisting of 1’s and 0’s. Code for this can be found in appendix. S1 and s0 is created with respect to my foundings in the analytical part which can be seen in the handwritten section of this part. ML decision rule is applied with the specifications I have made in the hand written part too. I have calculated the error by looking at the difference size with the original and message signal. After this process SNR dB graphs are plotted in order to evaluate the results. When SNR rate is high noise is low and message consists of mostly the transferred signal which is what is wanted. Resulting signal can be seen below:

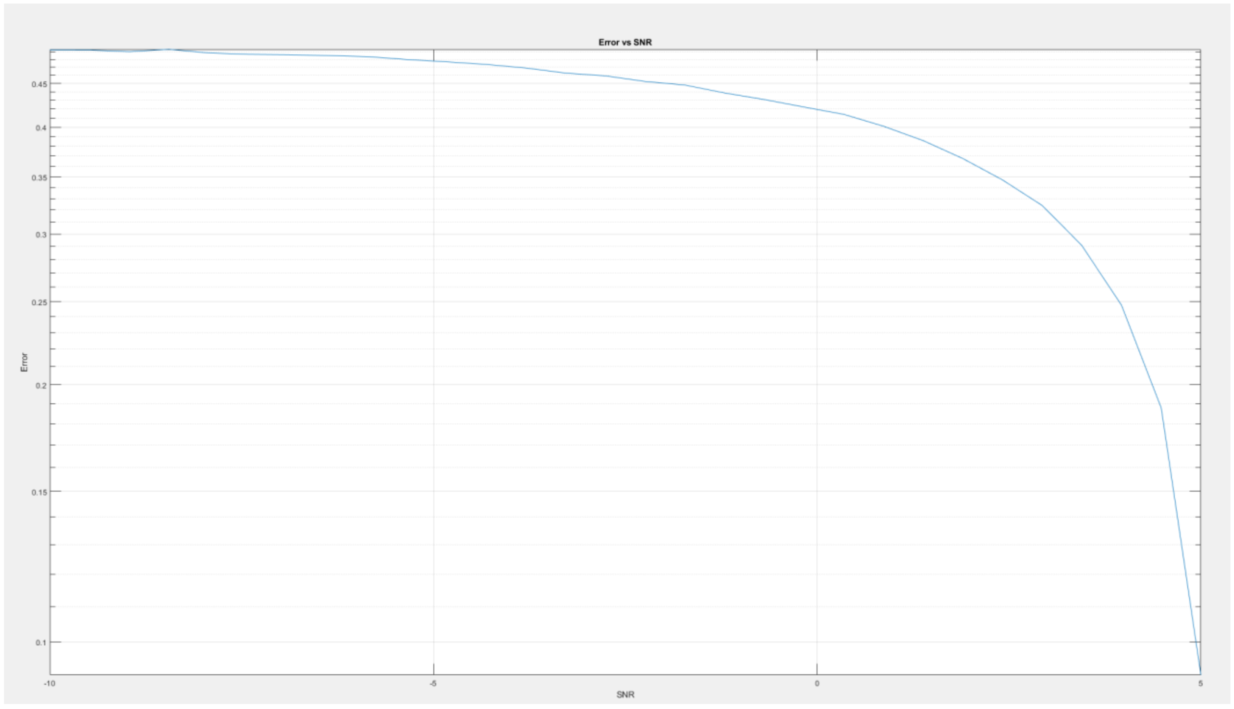


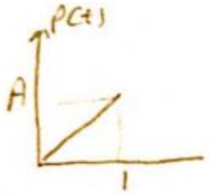
Figure 1: Error vs SNR Plot for Part a

Name: Cemal Güven Adal

ID:21703986

Part A

p(t) is chosen as rectangular pulse as it is shown in the tutorial.

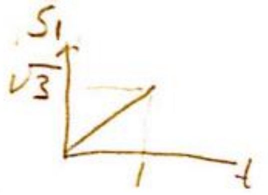


$\Rightarrow p(t) = At$ for $0 < t < 1$
0 ow

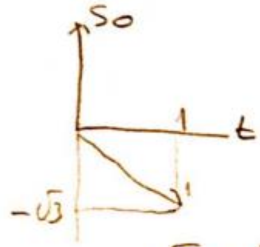
To find A for normalization;

$E_p = \int_0^1 (At)^2 = \int_0^1 A^2 t^2 = \Big|_0^1 \frac{A^2 t^3}{3} = \frac{A^2}{3} \Rightarrow$ should be 1

$\frac{A^2}{3} = 1 \Rightarrow A = \sqrt{3}$. Then p(t) becomes;



$S_1 = \sqrt{3}t$ for $0 < t < 1$
$S_0 = -S_1$



For ML Decision rule

$$M = \begin{cases} 1, & r \geq 0 \\ 0, & r \leq 0 \end{cases}$$

$\frac{\sqrt{3} - \sqrt{3}}{2} = 0$ so threshold is chosen at 0.

$\frac{E_b}{N_0} = \frac{1}{N_0} = \gamma_b$

$P(\text{Error}) = \frac{1}{2} \cdot P(n > 1) + \frac{1}{2} P(n < -1)$

$= Q\left(\sqrt{\frac{2}{N_0}}\right) = \boxed{Q\sqrt{2\gamma_b}} \rightarrow P(\text{Error})$

Name: Cemal Güven Adal

ID:21703986

Part B

In this part as it is asked in the homework sampling period has been changed from Ts/20 to Ts/10 so sampling rate has been reduced to half compared to the sampling rate that has been used in part a. The resulting Error vs SNR plot can be seen below:

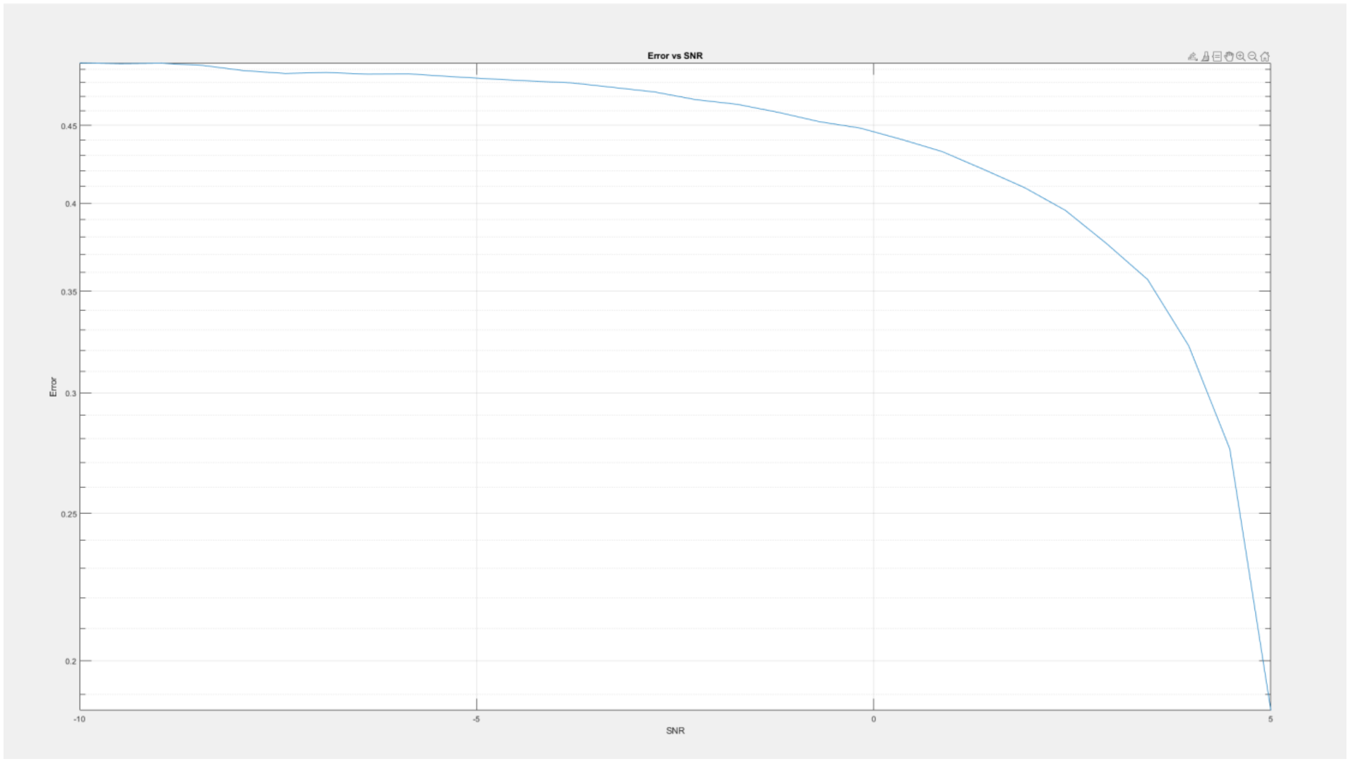


Figure 2: Error vs SNR Plot For Part b

When two graphs are compared it is clear that results are more disturbed in this part. This is because when sampling rate is decreased the result has more error because of under sampling.

Code for this section can be seen in appendix

Name: Cemal Güven Adal

ID:21703986

Part C

In this part rectangular pulse is used for transmission as it is asked in the homework. Basis function is found in the analytical hand written section for this part and can be seen below. When probability of errors are compared it has been calculated that part c has more error compared to part a. S1 and S0 are chosen accordingly to the results found in the analytical part for matlab. Sampling rate at part a is used. Resulting Error vs SNR plot can be seen below

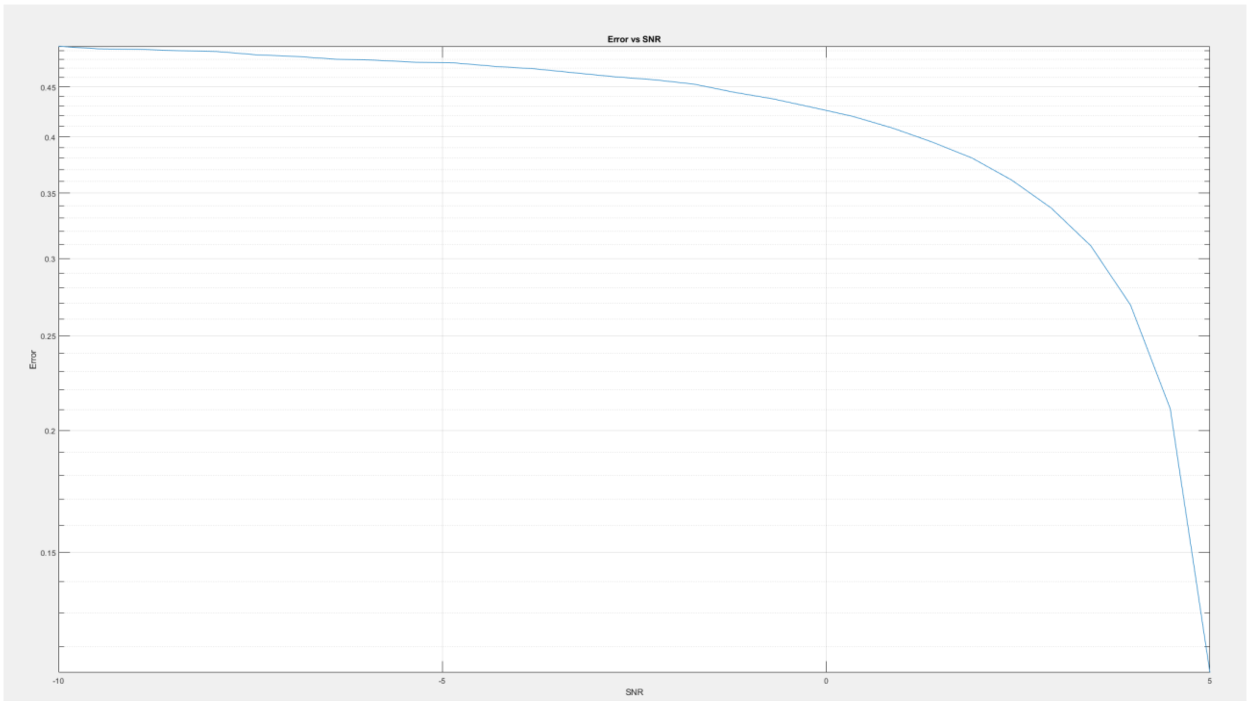


Figure 3: Error vs SNR Plot for Part C

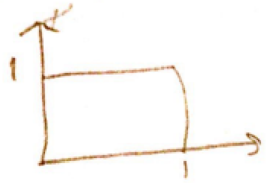
When this is compared to part a as it can be seen from the graph and analytical hand written section for this part it is clear that rectangular pulse has more error compared to triangular pulse when it is used.

Name: Cemal Güven Adal

ID:21703986

Part C

Rectangular pulse is choosen.



$$S_1 = \begin{cases} \sqrt{3}\,t & 0<t<1 \\ 0 & ow \end{cases}$$

$$S_1 = \int_0^1 \sqrt{3}\,t = \left.\frac{\sqrt{3}\,t^2}{2}\right|_0^1 = \frac{\sqrt{3}}{2}$$

$$S_0 = -\int_0^1 \sqrt{3}\,t = -\left.\frac{\sqrt{3}\,t^2}{2}\right|_0^1 = \frac{-\sqrt{3}}{2}$$

$$\frac{\frac{\sqrt{3}}{2} - \frac{\sqrt{3}}{2}}{2} = 0$$

$$m = \begin{cases} 1, & r>0 \\ 0, & r<0 \end{cases}$$

$$P(\text{error}) = \frac{1}{2} P\left(n > \frac{\sqrt{3}}{2}\right) + \frac{1}{2} P\left(n < \frac{\sqrt{3}}{2}\right)$$

$$= Q\left(\sqrt{\frac{3}{2N_0}}\right), \quad 10\log(2) - 10\log\left(\frac{3}{2}\right) = 1.25\ \text{dB}$$

As it carbe seen probability error is larger in c compared to part a. This is becuze basis function was a triangler pulse.

Name: Cemal Güven Adal

ID:21703986

Part D

In this part it has been asked to create a delayed version of the sginal at receiver due to delay. I have done this by shifting the array by 6 in matlab. Code for this can be seen in the appendix. I have tried with different delays and as the delay increases the error also increases. Plot can be seen below:

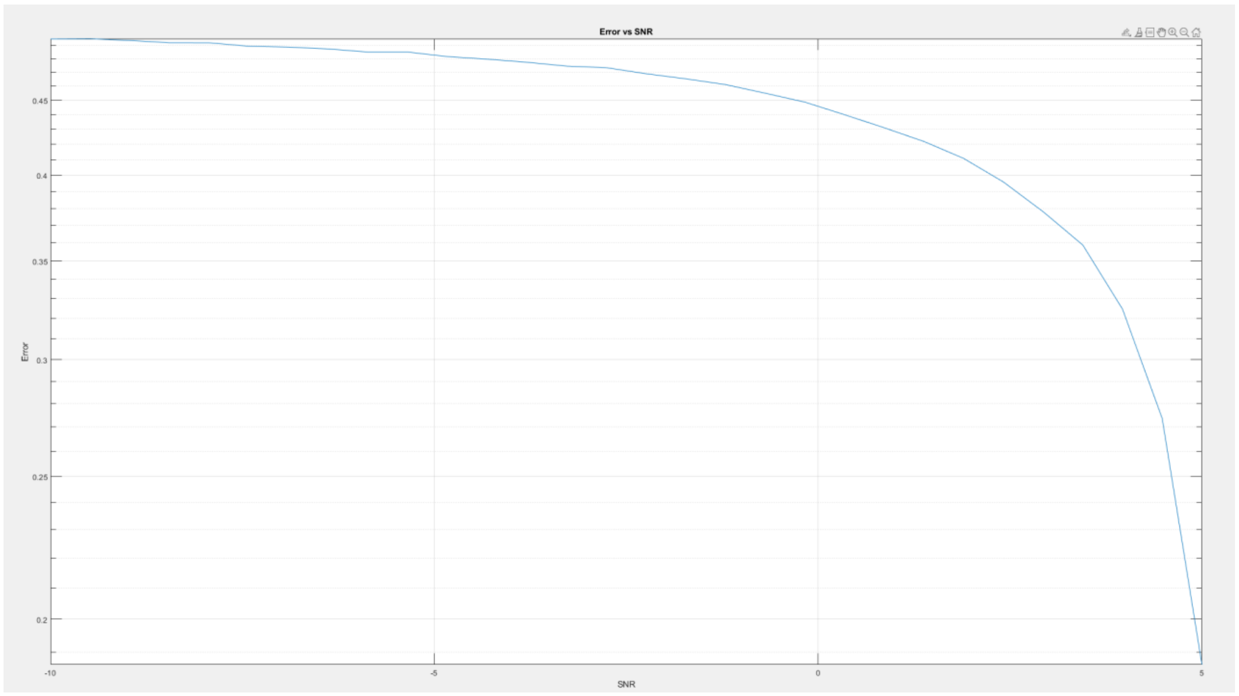


Figure 3: Error vs SNR Plot for Part D

Part E

In this part it has been asked that to simulate M PAM for M=2,4,8,16,32. Arrays are created in matlab and signal is created by taking random elements of this array to create message. 100000 elements are selected. ML rule is applied by checking the distance between each decision region. Explanation can be seen in the handwritten analytic section of this part. Code can be seen in the appendix. Resulting plots can be seen below:

Name: Cemal Güven Adal

ID:21703986

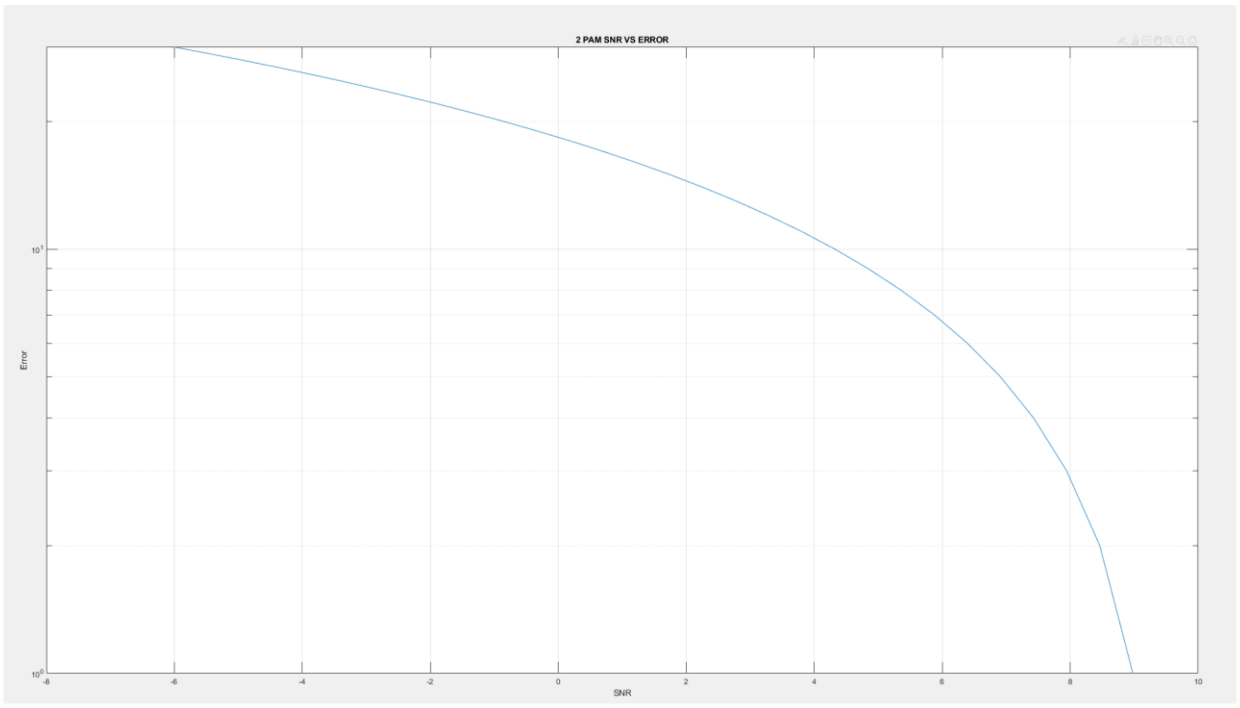


Figure 4: Error vs SNR Plot for 2 Pam

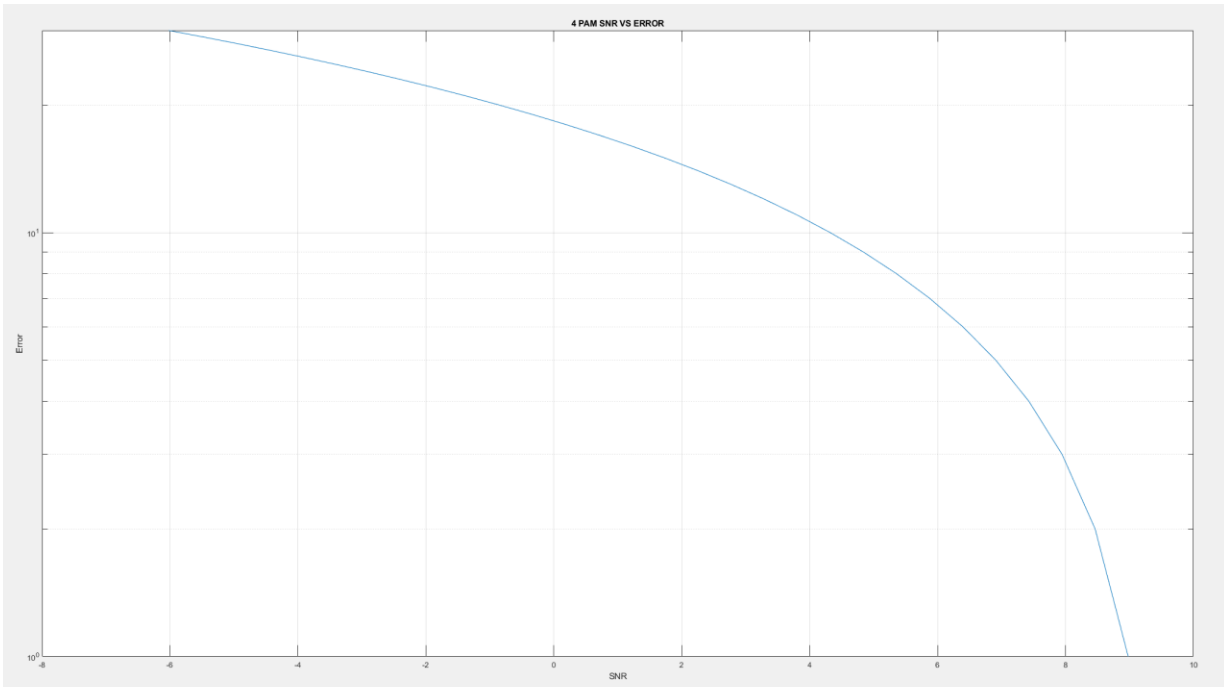


Figure 5: Error vs SNR Plot for 4 Pam

Name: Cemal Güven Adal

ID:21703986

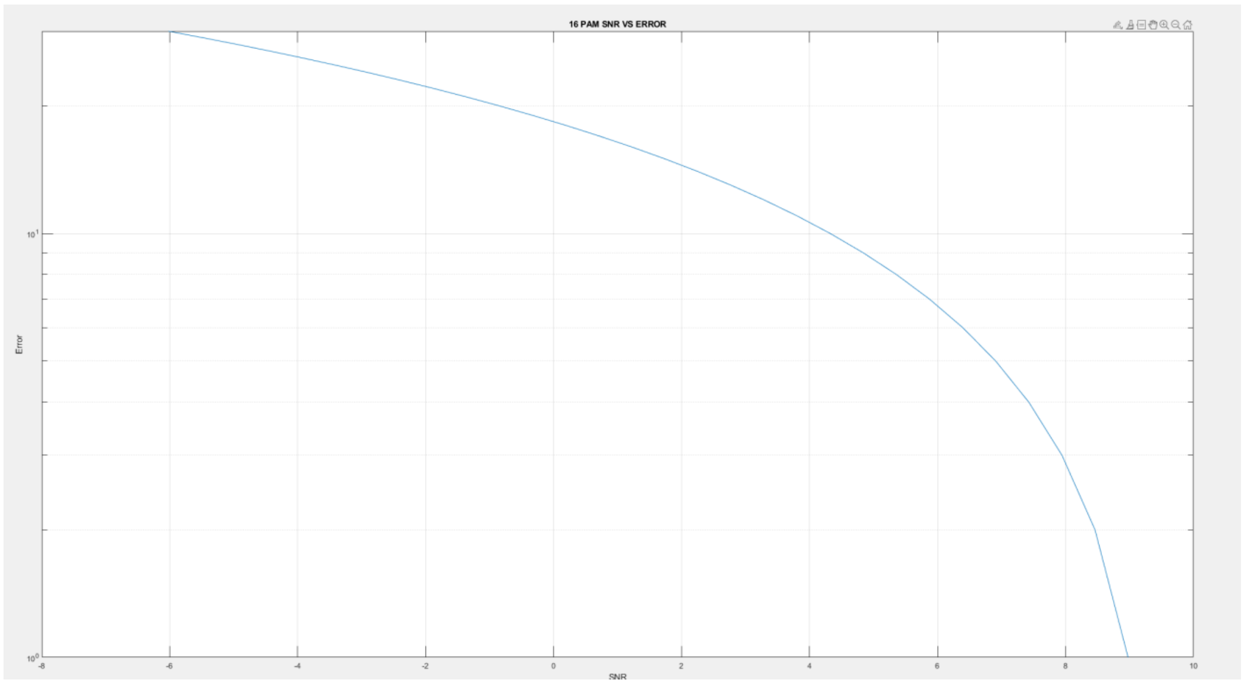


Figure 6: Error vs SNR Plot for 16 Pam

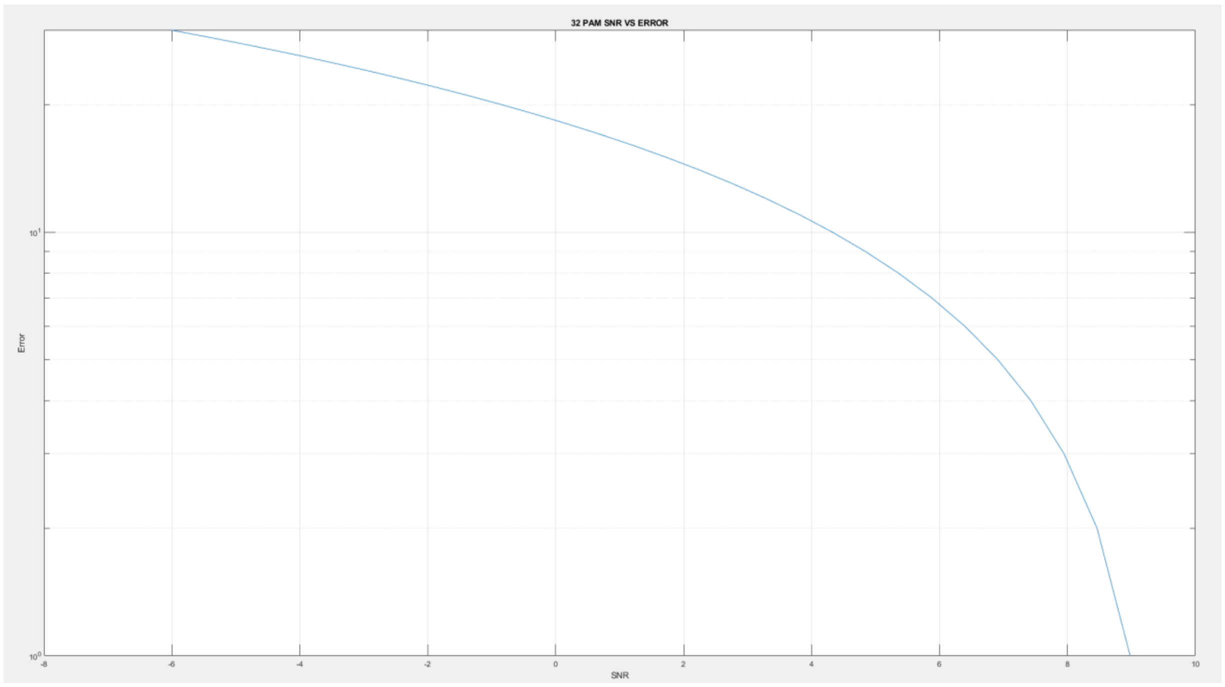


Figure 7: Error vs SNR Plot for 32 Pam

Name: Cemal Güven Adal

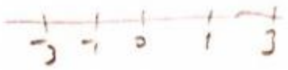
ID:21703986

[illegible]-E)

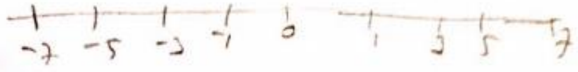
$S_i = (2i - 1 - M)$

for $M=2$ $\Rightarrow$

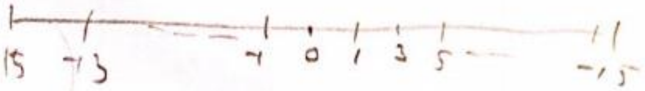
$M = 4$



for $M=8$



for $M=16$



ML

$$m = \begin{cases} 1, & r < -M+2 \\ i, & (2i-2-M) < r < (2i-M)d \\ M, & r > (M-2) \end{cases}$$

$P_e = \frac{1}{M}\sum_{1}^{M} P_{e,m} \Rightarrow$

$m=1 \Rightarrow Q\left(\frac{1}{\sqrt{\frac{N_0}{2}}}\right) \Rightarrow E_s = \frac{1}{2}\left((-1)^2 + 1^2\right) = 1 \Rightarrow$ for $M=2$

$m=2 \Rightarrow P(n>1) + P(n<-1)$ $= 1$

$P_e = 2Q\left(\frac{1}{\sqrt{\frac{N_0}{2}}}\right)$

Name: Cemal Güven Adal

ID:21703986

Appendix

```
% EE431 Project 2
%1
%Creating random binary data
N=100000;
x=randi([0,1],1,N);
T=1
Ts=T/20;
t=[0:Ts:T-Ts]
p=sqrt(3)*t;
eksip=-1*p;
a=zeros(20*length(x),1);
for j=1:length(x)
   if x(j)==1
      a(20*(j-1)+1:j*20,1)=p;
   else
       a(20*(j-1)+1:j*20,1)=-p;
   end
end
s1=p;
b=zeros(N*20,1);
N0=logspace(-0.5,1,30);
errorarray=zeros(1,length(N0));
for k =1:length(N0)
% Noise adding
a=a+(N0(k)/2)*20*randn(length(a),1);
for j=1:N
   b(20*(j-1)+1:20*j,1)=(s1.').*(a(20*(j-1)+1:20*j,1));
end

for j=1:length(b)/20
   y(j)=trapz(1/N,b(20*(j-1)+1:20*j));
end
error=0;
y1=zeros(1,length(y));
for i=1:length(y)
   if y(1,i)>=0
       y1(1,i)=1;
   else
       y1(1,i)=0;
   end
end

var=sum(abs(y1-x))/length(x);
errorarray(1,k)=var;
```

```
end
figure();
semilogy(10*log10(1./N0),(errorarray))
grid on
title('Error vs SNR')
xlabel('SNR')
ylabel('Error')


%
% % % part 2
N=100000;
x=randi([0,1],1,N);
T=1
Ts=T/10;
t=[0:Ts:T-Ts]
p=sqrt(3)*t;
eksip=-1*p;
a=zeros(20*length(x),1);
for j=1:length(x)
    if x(j)==1
       a(10*(j-1)+1:j*10,1)=p;
    else
        a(10*(j-1)+1:j*10,1)=-p;
    end
end
s1=p;
b=zeros(N*20,1);
N0=logspace(-0.5,1,30);
errorarray=zeros(1,length(N0));
for k =1:length(N0)
% Noise adding
a=a+(N0(k)/2)*20*randn(length(a),1);
for j=1:N
    b(10*(j-1)+1:10*j,1)=(s1.').*(a(10*(j-1)+1:10*j,1));
end

for j=1:length(b)/20
    y(j)=trapz(1/N,b(10*(j-1)+1:10*j));
end
error=0;
y1=zeros(1,length(y));
for i=1:length(y)
    if y(1,i)>=0
        y1(1,i)=1;
    else
        y1(1,i)=0;
    end
```

```
end

var=sum(abs(y1-x))/length(x);
errorarray(1,k)=var;

end
figure();
semilogy(10*log10(1./N0),(errorarray))
grid on
title('Error vs SNR')
xlabel('SNR')
ylabel('Error')
% part 3
N=100000;
x=randi([0,1],1,N);
T=1
Ts=T/20;
t=[0:Ts:T-Ts]
p=ones(1,20);
s1=sqrt(3)/2;
s0=-s1;
a=zeros(20*length(x),1);
for j=1:length(x)
    if x(j)==1
       a(20*(j-1)+1:j*20,1)=s1;
    else
        a(20*(j-1)+1:j*20,1)=s0;
    end
end

b=zeros(N*20,1);
N0=logspace(-0.5,1,30);
errorarray=zeros(1,length(N0));
for k =1:length(N0)
% Noise adding
a=a+(N0(k)/2)*20*randn(length(a),1);
for j=1:N
    b(20*(j-1)+1:20*j,1)=(p.').*(a(20*(j-1)+1:20*j,1));
end

for j=1:length(b)/20
    y(j)=trapz(1/N,b(20*(j-1)+1:20*j));
end
error=0;
y1=zeros(1,length(y));
for i=1:length(y)
    if y(1,i)>=0
        y1(1,i)=1;
    else
        y1(1,i)=0;
    end
end

var=sum(abs(y1-x))/length(x);
errorarray(1,k)=var;

end
figure();
semilogy(10*log10(1./N0),(errorarray))
grid on
title('Error vs SNR')
xlabel('SNR')
ylabel('Error')

% part 4
N=100000;
x=randi([0,1],1,N);
T=1
Ts=T/20;
t=[0:Ts:T-Ts]
p=sqrt(3)*t;
eksip=-1*p;
a=zeros(20*length(x),1);
for j=1:length(x)
    if x(j)==1
       a(20*(j-1)+1:j*20,1)=p;
    else
        a(20*(j-1)+1:j*20,1)=-p;
    end
end
s1=p;
b=zeros(N*20,1);
N0=logspace(-0.5,1,30);
errorarray=zeros(1,length(N0));
for k =1:length(N0)
% Noise adding
a=a+(N0(k)/2)*20*randn(length(a),1);
for j=1:N-1
    b(20*(j-1)+1:20*j,1)=(s1.').*(a(20*(j-1)+6:20*j+5,1));
end

for j=1:length(b)/20
    y(j)=trapz(1/N,b(20*(j-1)+1:20*j));
end
error=0;
y1=zeros(1,length(y));
```

```
for i=1:length(y)
    if y(1,i)>=0
        y1(1,i)=1;
    else
        y1(1,i)=0;
    end
end

var=sum(abs(y1-x))/length(x);
errorarray(1,k)=var;

end
figure();
semilogy(10*log10(1./N0),(errorarray))
grid on
title('Error vs SNR')
xlabel('SNR')
ylabel('Error')


%
% % part 5
% 2
N0=logspace(-0.5,1,30);
S11=[-1,1];
signal=zeros(1,100000);
errore=0;
varer=zeros(1,length(N0));
for  i= 1:100000
    a=randi([1,2]);
    signal(1,i)=S11(a);
end
for h =1:length(N0)


signal=signal+sqrt((N0(h)/2))*randn(1,100000);
signal2=zeros(1,100000);
for j= 1:100000
    [x,I]=min(abs(signal(1,j)-S11));
    signal2(1,j)=S11(I);
end
for i=1:100000
    if signal2(i)==signal(i)
        errore=errore;
    else
        errore=errore+1;
    end
end
```

```
varer(h)=errore/100000;
end
figure()
semilogy(10*log10(2.5./N0),(varer))
title('2 PAM SNR VS ERROR');
xlabel('SNR')
ylabel('Error')

grid on
% 4
N0=logspace(-0.5,1,30);
S11=[-3,-1,1,3];
signal=zeros(1,100000);
errore=0;
varer=zeros(1,length(N0));
for  i= 1:100000
    a=randi([1,4]);
    signal(1,i)=S11(a);
end
for h =1:length(N0)


signal=signal+sqrt((N0(h)/2))*randn(1,100000);
signal2=zeros(1,100000);
for j= 1:100000
    [x,I]=min(abs(signal(1,j)-S11));
    signal2(1,j)=S11(I);
end
for i=1:100000
    if signal2(i)==signal(i)
        errore=errore;
    else
        errore=errore+1;
    end
end

varer(h)=errore/100000;
end
figure()
semilogy(10*log10(2.5./N0),(varer))
grid on
title('4 PAM SNR VS ERROR');
xlabel('SNR')
ylabel('Error')
% 8
N0=logspace(-0.5,1,30);
S11=[-7,-5,-3,-1,1,3,5,7];
```

Name: Cemal Güven Adal

ID:21703986

```
signal=zeros(1,100000);
errore=0;
varer=zeros(1,length(N0));
for  i= 1:100000
    a=randi([1,8]);
    signal(1,i)=S11(a);
end
for h =1:length(N0)


signal=signal+sqrt((N0(h)/2))*randn(1,100000);
signal2=zeros(1,100000);
for j= 1:100000
    [x,I]=min(abs(signal(1,j)-S11));
    signal2(1,j)=S11(I);
end
for i=1:100000
    if signal2(i)==signal(i)
        errore=errore;
    else
        errore=errore+1;
    end
end

varer(h)=errore/100000;
end
figure()
semilogy(10*log10(2.5./N0),(varer))
title('8 PAM SNR VS ERROR');
xlabel('SNR')
ylabel('Error')

grid on
% 16
N0=logspace(-0.5,1,30);
S11=[-15,-13,-11,-9,-7,-5,-3,-1,1,3,5,7,9,11,13,15];
signal=zeros(1,100000);
errore=0;
varer=zeros(1,length(N0));
for  i= 1:100000
    a=randi([1,16]);
    signal(1,i)=S11(a);
end
for h =1:length(N0)


signal=signal+sqrt((N0(h)/2))*randn(1,100000);
signal2=zeros(1,100000);
```

Name: Cemal Güven Adal

ID:21703986

```
for j= 1:100000
    [x,I]=min(abs(signal(1,j)-S11));
    signal2(1,j)=S11(I);
end
for i=1:100000
    if signal2(i)==signal(i)
        errore=errore;
    else
        errore=errore+1;
    end
end

varer(h)=errore/100000;
end
figure()
semilogy(10*log10(2.5./N0),(varer))
title('16 PAM SNR VS ERROR');
xlabel('SNR')
ylabel('Error')

grid on
% 32
N0=logspace(-0.5,1,30);
S11=[-31,-29,-27,-25,-23,-21,-19,-17,-15,-13,-11,-9,-7,-5,-3,-
1,1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31];
signal=zeros(1,100000);
errore=0;
varer=zeros(1,length(N0));
for  i= 1:100000
    a=randi([1,32]);
    signal(1,i)=S11(a);
end
for h =1:length(N0)


signal=signal+sqrt((N0(h)/2))*randn(1,100000);
signal2=zeros(1,100000);
for j= 1:100000
    [x,I]=min(abs(signal(1,j)-S11));
    signal2(1,j)=S11(I);
end
for i=1:100000
    if signal2(i)==signal(i)
        errore=errore;
    else
        errore=errore+1;
    end
end
```

Name: Cemal Güven Adal

ID:21703986

```
varer(h)=errore/100000;
end
figure()
semilogy(10*log10(2.5./N0),(varer))
title('32 PAM SNR VS ERROR');
xlabel('SNR')
ylabel('Error')

grid on
```