

Πανεπιστήμιο Πατρών
Τμήμα Μηχ. Η/Υ & Πληροφορικής
ΨΗΦΙΑΚΕΣ ΤΗΛΕΠΙΚΟΙΝΩΝΙΕΣ
Ακαδημαϊκό Έτος 2022-2023

Ον/μο : ΣΠΥΡΟ ΣΟΥΛΙ

ΑΜ : 1070263

Έτος : 6^ο

1ο σετ Εργαστηριακών Ασκήσεων

Ζητούμενα για Μέρος 1

A.ΗΧΟΣ

1. Έγραψα κώδικα από τον οποίο υπάρχουν μερικά λάθη τα οποία δεν μπόρεσα να τα διορθώσω οπότε δεν μπορώ να σας δείξω και αποτελέσματα των γραφικών. Το καλύτερο που μπορώ να κάνω είναι να εξηγήσω οτιδήποτε θεωρητικά. Η τιμή του SQNR, υπολογίζεται από τα

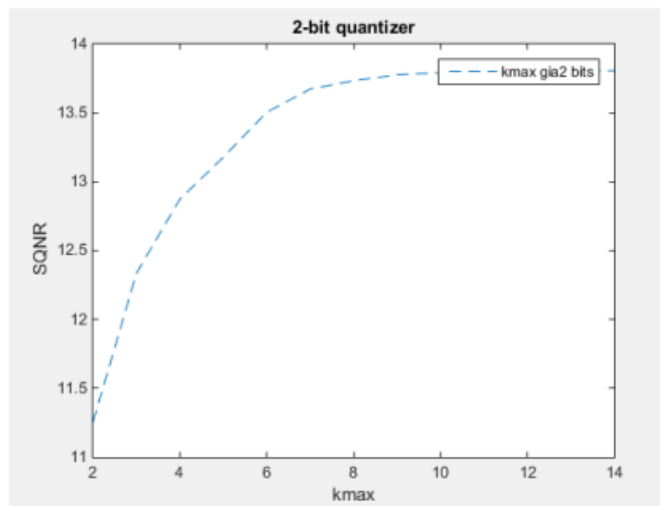
πηλίκο των κέντρων του κωδικοποιημένου διανύσματος εισόδου x_q στο τετράγωνο προς την μέση παραμόρφωση. Εκφράζεται σε dB

2. Κωδικοποίηση ADM

Η κωδικοποίηση ADM είναι πολύ πιο περίπλοκη από την PCM, μιας και η κωδικοποίηση γίνεται δυναμικά και το βήμα κβάντισης είναι εξαρτώμενο από την παρούσα έξοδο του κβαντιστή και το προηγούμενο βήμα.

Β.ΕΙΚΟΝΑ

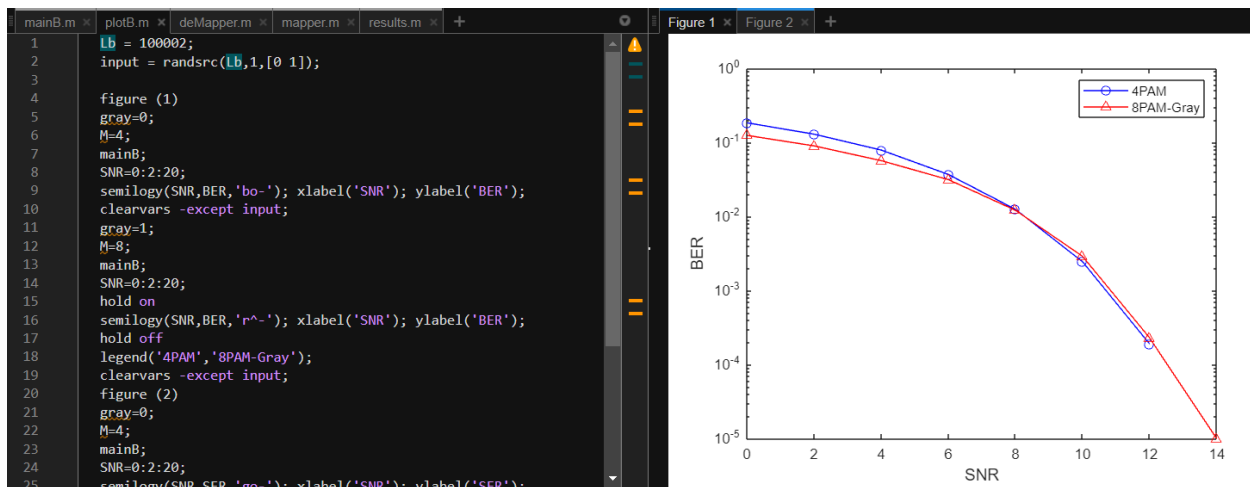
Εκτελούμε τον κώδικα [img.m](#) και έχουμε :



Ζητούμενα για Μέρος 2

1. Στην αρχή για την υλοποίηση της M-PAM φτιάχνουμε τις παραμέτρους και τους δίνουμε τιμές, ύστερα δημιουργούμαστε με την `randsrc` την ακολουθία εισόδου. Στην συνέχεια , δίνουμε την ακολουθία εισόδου σαν είσοδο στον `mapper` που φτιάξαμε ώστε να πολλαπλασιαστεί η έξοδος με τον ορθογώνιο παλμό. Η συνάρτηση `mapper` ελέγχει αν έχουμε επιλέξει κωδικοποίηση Grey η κανονική και αντιστοιχίζει ομάδες $\log_2 M$ bits για $M=4$ και $M=8$ στα αντίστοιχα σύμβολα. Έπειτα για κάθε σύμβολο όπως μας είπατε κρατάμε 40 δείγματα στα οποία προσθέτουμε τον λευκό Γκαουσιανό θόρυβο και ύστερα τα στέλνουμε στον δέκτη , ο δέκτης με την σειρά του συσχετίζει την φέρουσα με το σήμα και έπειτα ο φορατής κάνει ανάκτηση του συμβόλου , αυτό γίνεται με το να κάνουμε αναζήτηση της μικρότερης απόστασης σε σχέση με τον σύμβολο της και τέλος επιστρέφοντας την τελική ακολουθία. Ο `demapper` κάνει ακριβώς το αντίθετο από τον `mapper` δηλαδή αντιστοιχεί σύμβολα σε ομάδες $\log_2 M$.

2.



Βλέπουμε ότι η ανοχή του σφάλματος μεγαλώνει όσο μεγαλώνει το M και αυξάνει ο ρυθμός μετάδοσης

3.

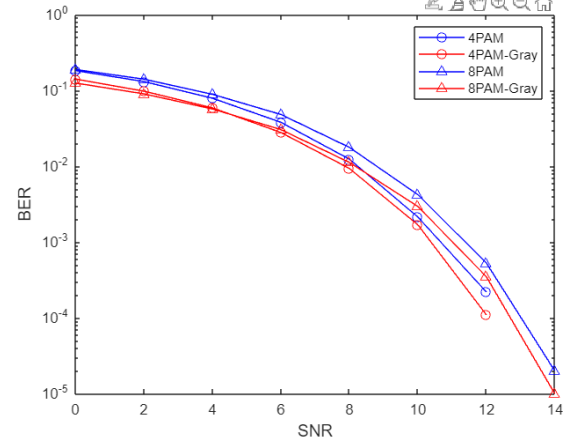
Όταν εφαρμόζουμε Gray στο M-PAM, η γκρι κωδικοποίηση χρησιμοποιείται για την αντιστοίχιση των συμβόλων M-PAM σε δυαδικές τιμές έτσι ώστε το ποσοστό σφάλματος συμβόλων να ελαχιστοποιείται. Αυτό συμβαίνει επειδή η Gray κωδικοποίηση διασφαλίζει ότι τα δύο πλησιέστερα σύμβολα στον αστερισμό σημάτων έχουν τον μικρότερο αριθμό αλλαγών bit μεταξύ τους. Συνεπώς η κωδικοποίηση κατά Gray είναι πιο αποδοτική καθώς δίνει μικρότερα σφάλματα από την αντίστοιχη κανονική κωδικοποίηση και αυτό συμβαίνει στην εργασία.

4.

```

1  Lb = 100002;
2  input = randsrc(Lb,1,[0 1]);
3
4  figure (1)
5  gray=0;
6  M=4;
7  mainB;
8  SNR=0:2:20;
9  semilogy(SNR,BER,'bo-'); xlabel('SNR'); ylabel('BER');
10 clearvars -except input;
11
12 gray=1;
13 M=4;
14 mainB;
15 SNR=0:2:20;
16 hold on
17 semilogy(SNR,BER,'ro-'); xlabel('SNR'); ylabel('BER');
18 hold off
19 clearvars -except input;
20
21 gray=0;
22 M=8;
23 mainB;
24 SNR=0:2:20;
25 hold on

```



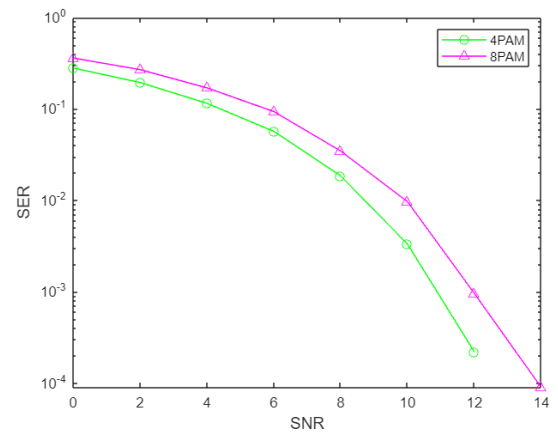
Όσο μεγαλώνει ο αριθμός M , τόσο μειώνεται και η ανοχή στα σφάλματα, αλλά αυξάνεται ο ρυθμός μετάδοσης

5.

```

1  Lb = 100002;
2  input = randsrc(Lb,1,[0 1]);
3
4  figure (1)
5  gray=0;
6  M=4;
7  mainB;
8  SNR=0:2:20;
9  semilogy(SNR,BER,'bo-'); xlabel('SNR'); ylabel('BER');
10 clearvars -except input;
11
12 gray=1;
13 M=4;
14 mainB;
15 SNR=0:2:20;
16 hold on
17 semilogy(SNR,BER,'r^--'); xlabel('SNR'); ylabel('BER');
18 hold off
19 legend('4PAM','8PAM-Gray');
20 clearvars -except input;
21 figure (2)
22 gray=0;
23 M=4;
24 mainB;
25 SNR=0:2:20;
26 semilogy(SNR,SER,'go-'); xlabel('SNR'); ylabel('SER');
27
28 gray=1;
29 M=8;
30 mainB;
31 SNR=0:2:20;
32 hold on
33 semilogy(SNR,SER,'m^--'); xlabel('SNR'); ylabel('SER');
34 hold off
35 legend('4PAM','8PAM');
36 clearvars -except input;

```



Η SER είναι μεγαλύτερη στην 8-PAM από την 4-PAM αυτό ισχύει διότι η 8-PAM έχει περισσότερα σύμβολα από την 4-PAM. Σημαντικό είναι να πούμε ότι για $SNR > 13$ η 8-PAM μειώνει τα σφάλματα περισσότερο από την 4-PAM

ΚΩΔΙΚΕΣ

Ερώτημα 1

adm.m

```
function [SQNR,Sout]=adm(x,M)
```

```
Sin=interp(x,M);
```

```

samples=length(Sin);
ss=1/(5*M);

[ssa,enc_signal]=adm_encoder(Sin,ss);

Sout=adm_decoder(ssa,enc_signal);

err_signal=Sin'-Sout;

Pq=sum(err_signal(1:samples).^2)/samples;
Px=sum(Sin(1:samples).^2)/samples;
SQNR=10*log10(Px/Pq);

```

admDecoder.m

```

function [Sout]=admDecoder(Step,cn)

result(1)=0;

for i=1:length(cn)
    if(cn(i)==1)
        result(i+1)=result(i)+Step(i);
    end
end

```

```

elseif (cn(i)==-1)
    result(i+1)=result(i)-Step(i);
end
end

dec_signal=result(2:length(cn)+1);

Sout=lpf(100,.1,dec_signal);
end

```

admEncoder.m

```

function [Step,cn]=admEncoder(x,StepSize)

Step(1)=StepSize;

if(x(1)>=0)
    result(1)=1;
    accum(2)=result(1)*Step(1);
else

```



```
result(1)=-1;
```

```
accum(2)=result(1)*Step(1); end
```

```
for i=2:length(x)
```

```
    if(x(i)>=accum(i))
```

```
        result(i)=1;
```

```
        if result(i)==result(i-1)
```

```
            Step(i)=1.5*Step(i-1);
```

```
        else
```

```
            Step(i)=(1.5^-1)*Step(i-1); end
```

```
        accum(i+1)=accum(i)+result(i)*Step(i);
```

```
    else
```

```
        result(i)=-1;
```

```
        if result(i)==result(i-1)
```

```
            Step(i)=1.5*Step(i-1);
```

```
        else
```

```
            Step(i)=(1.5^-1)*Step(i-1); end
```

```
        accum(i+1)=accum(i)+result(i)*Step(i); end
```

```
end
```

```
cn=result;
```

AR1.simulation.m

```
%wav  
function [y1, y2] = AR1_simulation(L)  
    L=1000;  
    x = randn(L,1);  
    a1 = 0.9;  
    a2 = 0.01;  
    b = 1;  
    y1 = filter(b, [1 -a1], x);  
    y2 = filter(b, [1 -a2], x);  
    [y1, y2] = AR1_simulation(L);  
end
```

img.m

```
load cameraman.mat  
figure(1)  
imshow(uint8(i));  
title('Input main script')  
  
img_signal_1d=i(:);
```

```
img_signal_1d=(img_signal_1d-128)/128;
```

```
for i=1:2
```

```
    [xq,centers,D]=lloyd_max(img_signal_1d,2*i,-1,1);
```

```
    figure(i*2)
```

```
    p=128*xq+128;
```

```
    y=reshape(p,256,256);
```

```
    imshow(uint8(y));
```

```
    title(strcat(num2str(2^i))
```

```
    SQNR=zeros(length(D),1);
```

```
    for k=1:length(D)
```

```
        SQNR(k)=10*log10(mean(img_signal_1d.^2)/D(k)); end
```

```
    kmax=[1:1:length(D)];
```

```
    figure(i*2+1)
```

```
    plot(kmax,SQNR); xlabel('kmax'); ylabel('SQNR');
```

```
    title(strcat(num2str(2*i)))
```

```
    legend(strcat(num2str(2*i)));
```

```
    tmp=tabulate(xq);
```

```
    prob=tmp(:,3);
```

```
prob=prob./100;  
entropy(i)= -sum(prob.*log2(prob)) end
```

loydMax.m

```
function [xq,centers,D] = lloydMax(x,N,xmin,xmax)  
    levels=2^N;  
    delta=(xmax-xmin)/levels;  
    upperBound=[];  
    lowerBound=[];  
    c=xmin;  
  
    while c<xmax  
        lowerBound=[lowerBound c];  
        upperBound=[upperBound c+delta];  
        c=c+delta;    end  
  
    span=cell(1,length(lowerBound));  
    centers=zeros(1,length(lowerBound));  
    for i=1:length(lowerBound)  
        span{i}=[lowerBound(i) upperBound(i)];
```

```

        centers(i)=mean(span{i});    end
D=[0];
xq=zeros(length(x),1);
for k=1:length(span)
    s=span{k};
    for i=1:length(x)
        if (x(i)>=s(length(s)-1))&&(x(i)<s(length(s)))
            xq(i)=centers(k);
        end
        if x(i)<=xmin
            xq(i)=centers(1);
        end
        if x(i)>=xmax
            xq(i)=centers(length(centers));
        end
    end
end
end
D=[D mean((x-xq).^2)];

count=2;
while abs(D(count)-D(count-1))>=eps
    tmp=[];

```

```

for i=1:length(centers)-1
    tmp=[tmp (centers(i)+centers(i+1))/2];
end
spaces=[xmin,[tmp],xmax];
for i=1:length(spaces)-1
    span{i}=[spaces(i) spaces(i+1)];
end
for i=1:length(spaces)-1
    centers(i)=mean(x(x>=spaces(i) & x<=spaces(i+1)));
end
xq=zeros(length(x),1);
for k=1:length(span)
    s=span{k};
    for i=1:length(x)
        if (x(i)>=s(length(s)-1))&&(x(i)<s(length(s)))
            xq(i)=centers(k);
        end
        if x(i)<=xmin
            xq(i)=centers(1);
        end
        if x(i)>=xmax
            xq(i)=centers(length(centers));
        end
    end
end

```

```

        end
    end
end
D=[D mean((x-xq).^2)];
count=count+1;
end
end

```

Ερώτημα 2

demapper.m

```

function [y] = demapper(x,M,gray)
k = 1;
if gray == 0 %Simple encoding
    if M == 4
        for i = 1:length(x)
            if x(i) == 0

```

```

        y(k) = 0;
        y(k+1) = 0;
    elseif x(i) == 1
        y(k) = 0;
        y(k+1) = 1;
    elseif x(i) == 2
        y(k) = 1;
        y(k+1) = 0;
    elseif x(i) == 3
        y(k) = 1;
        y(k+1) = 1;
    end
    k = k+2 ;
end
elseif M == 8
    for i = 1:length(x)
        if x(i) == 0
            y(k) = 0;
            y(k+1) = 0;
            y(k+2)=0;
        elseif x(i) == 1
            y(k) = 0;

```



```
y(k+1) = 0;  
y(k+2)=1;  
elseif x(i) == 2  
    y(k) = 0;  
    y(k+1) = 1;  
    y(k+2)=0;  
elseif x(i) == 3  
    y(k) = 0;  
    y(k+1) = 1;  
    y(k+2)=1;  
elseif x(i) == 4  
    y(k) = 1;  
    y(k+1) = 0;  
    y(k+2)=0;  
elseif x(i) == 5  
    y(k) = 1;  
    y(k+1) = 0;  
    y(k+2)=1;  
elseif x(i) == 6  
    y(k) = 1;  
    y(k+1) = 1;  
    y(k+2)=0;
```

```

elseif x(i) == 7
    y(k) = 1;
    y(k+1) = 1;
    y(k+2)=1;
end
k = k+3 ;
end
end
elseif gray == 1 %Gray code
if M == 4
for i = 1:length(x)
    if x(i) == 0
        y(k) = 0;
        y(k+1) = 0;
    elseif x(i) == 1
        y(k) = 0;
        y(k+1) = 1;
    elseif x(i) == 2
        y(k) = 1;
        y(k+1) = 1;
    elseif x(i) == 3
        y(k) = 1;

```

```

        y(k+1) = 0;
    end
    k = k+2 ;
end
elseif M == 8
    for i = 1:length(x)
        if x(i) == 0
            y(k) = 0;
            y(k+1) = 0;
            y(k+2)=0;
        elseif x(i) == 1
            y(k) = 0;
            y(k+1) = 0;
            y(k+2)=1;
        elseif x(i) == 2
            y(k) = 0;
            y(k+1) = 1;
            y(k+2)=1;
        elseif x(i) == 3
            y(k) = 0;
            y(k+1) = 1;
            y(k+2)=0;
        end
    end
end

```

```
elseif x(i) == 4
    y(k) = 1;
    y(k+1) = 1;
    y(k+2)=0;
elseif x(i) == 5
    y(k) = 1;
    y(k+1) = 1;
    y(k+2)=1;
elseif x(i) == 6
    y(k) = 1;
    y(k+1) = 0;
    y(k+2)=1;
elseif x(i) == 7
    y(k) = 1;
    y(k+1) = 0;
    y(k+2)=0;
end
k = k+3 ;
end
end
end
```

mainB.m

```
Tsample=1;
Tc=4;
fc=1/Tc;
Tsymbol=40;
Lb = 100002;
z=0;

for SNR=0:2:20
    %Mapper
    in=mapper(input,M,gray);

    %rectangular pulse
    g=sqrt(2/Tsymbol);

    %M-PAM modulation
    A=1/sqrt(M+1);
    for m=0:M-1
        s(m+1)=(2*m-1-M)*A;
    end
end
```

```

for i=1:length(in)
    for t=1:Tsymbol
        signal(i,t)=s(in(i)+1)*g*cos(2*pi*fc*t);
    end
end
end

```

%AWGN

```
sigma_sq=(1/log2(M))/(2*10^(SNR/10)); %σ^2
```

```
noise=sqrt(sigma_sq)*randn(Lb/log2(M),Tsymbol); %Gaussian noise
```

```
send=signal+noise;
```

%M-PAM demodulation

```

for t=1:Tsymbol
    y(t)=g*cos(2*pi*fc*t);
end

```

```
r=send*y';
```

%Envelope Detector

```

for i=1:length(r)
    for j=1:M
        temp(i,j)=norm(r(i)-s(j));
    end
end

```

```

end
s_h(i)=min(temp(i,:));
for j=1:M
    if s_h(i)==temp(i,j)
        s_hat(i)=j-1;
    end
end
end
end

%Demapper
output=demapper(s_hat,M,gray);

z=z+1;
%BER calculation
b_errors=0;
for i=1:length(output)
    if output(i)~=input(i)
        b_errors=b_errors +1 ;
    end
end
end
BER(z)=b_errors/length(input);

```

```

%SER calculation
s_errors=0;
for i=1:length(s_hat)
    if s_hat(i)~=in(i)
        s_errors=s_errors +1 ;
    end
end
SER(z)=s_errors/length(in);
end

```

mapper.m

```

function [y] = mapper(x,M,gray)
k = 1;
if gray == 0 %Simple encoding
    if M == 4
        for i = 1:2:length(x)
            if x(i) == 0 && x(i+1) == 0
                y(k) = 0;
            elseif x(i) == 0 && x(i+1) == 1
                y(k) = 1;
            end
            k = k + 1;
        end
    end
end

```



```

elseif x(i) == 1 && x(i+1) == 0
    y(k) = 2;
elseif x(i) == 1 && x(i+1) == 1
    y(k) = 3;
end
k = k+1 ;
end
elseif M == 8
    for i = 1:3:length(x)
        if x(i) == 0 && x(i+1) == 0 && x(i+2) == 0
            y(k) = 0;
        elseif x(i) == 0 && x(i+1) == 0 && x(i+2) == 1
            y(k) = 1;
        elseif x(i) == 0 && x(i+1) == 1 && x(i+2) == 0
            y(k) = 2;
        elseif x(i) == 0 && x(i+1) == 1 && x(i+2) == 1
            y(k) = 3;
        elseif x(i) == 1 && x(i+1) == 0 && x(i+2) == 0
            y(k) = 4;
        elseif x(i) == 1 && x(i+1) == 0 && x(i+2) == 1
            y(k) = 5;
        elseif x(i) == 1 && x(i+1) == 1 && x(i+2) == 0

```

```

        y(k) = 6;
    elseif x(i) == 1 && x(i+1) == 1 && x(i+2) == 1
        y(k) = 7;
    end
    k = k+1;
end
end
elseif gray == 1 %Gray code
    if M == 4
        for i = 1:2:length(x)
            if x(i) == 0 && x(i+1) == 0
                y(k) = 0;
            elseif x(i) == 0 && x(i+1) == 1
                y(k) = 1;
            elseif x(i) == 1 && x(i+1) == 1
                y(k) = 2;
            elseif x(i) == 1 && x(i+1) == 0
                y(k) = 3;
            end
            k = k+1 ;
        end
    elseif M == 8

```

```

for i = 1:3:length(x)
    if x(i) == 0 && x(i+1) == 0 && x(i+2) == 0
        y(k) = 0;
    elseif x(i) == 0 && x(i+1) == 0 && x(i+2) == 1
        y(k) = 1;
    elseif x(i) == 0 && x(i+1) == 1 && x(i+2) == 1
        y(k) = 2;
    elseif x(i) == 0 && x(i+1) == 1 && x(i+2) == 0
        y(k) = 3;
    elseif x(i) == 1 && x(i+1) == 1 && x(i+2) == 0
        y(k) = 4;
    elseif x(i) == 1 && x(i+1) == 1 && x(i+2) == 1
        y(k) = 5;
    elseif x(i) == 1 && x(i+1) == 0 && x(i+2) == 1
        y(k) = 6;
    elseif x(i) == 1 && x(i+1) == 0 && x(i+2) == 0
        y(k) = 7;
    end
    k = k+1;
end
end
end

```

results.m

```
Lb = 100002;  
input = randsrc(Lb,1,[0 1]);  
  
figure (1) %BER  
gray=0;  
M=4;  
mainB;  
SNR=0:2:20;  
semilogy(SNR,BER,'bo-'); xlabel('SNR'); ylabel('BER');  
clearvars -except input;  
  
gray=1;  
M=4;  
mainB;  
SNR=0:2:20;  
hold on  
semilogy(SNR,BER,'ro-'); xlabel('SNR'); ylabel('BER');  
hold off  
clearvars -except input;
```

```
gray=0;
M=8;
mainB;
SNR=0:2:20;
hold on
semilogy(SNR,BER,'b^-'); xlabel('SNR'); ylabel('BER');
hold off
clearvars -except input;
```

```
gray=1;
M=8;
mainB;
SNR=0:2:20;
hold on
semilogy(SNR,BER,'r^-'); xlabel('SNR'); ylabel('BER');
hold off
legend('4PAM','4PAM-Gray','8PAM','8PAM-Gray');
clearvars -except input;
```

plotB.m

```
Lb = 100002;  
input = randsrc(Lb,1,[0 1]);  
  
figure (1) %BER  
gray=0;  
M=4;  
mainB;  
SNR=0:2:20;  
semilogy(SNR,BER,'bo-'); xlabel('SNR'); ylabel('BER');  
clearvars -except input;  
  
gray=1;  
M=8;  
mainB;  
SNR=0:2:20;  
hold on  
semilogy(SNR,BER,'r^-'); xlabel('SNR'); ylabel('BER');  
hold off  
legend('4PAM','8PAM-Gray');  
clearvars -except input;
```

```
figure (2) %SER
gray=0;
M=4;
mainB;
SNR=0:2:20;
semilogy(SNR,SER,'go-'); xlabel('SNR'); ylabel('SER');
clearvars -except input;

gray=0;
M=8;
mainB;
SNR=0:2:20;
hold on
semilogy(SNR,SER,'m^-'); xlabel('SNR'); ylabel('SER');
hold off
legend('4PAM','8PAM');
clearvars -except input;
```

Τέλος , ευχαριστώ για τον χρόνο σας !