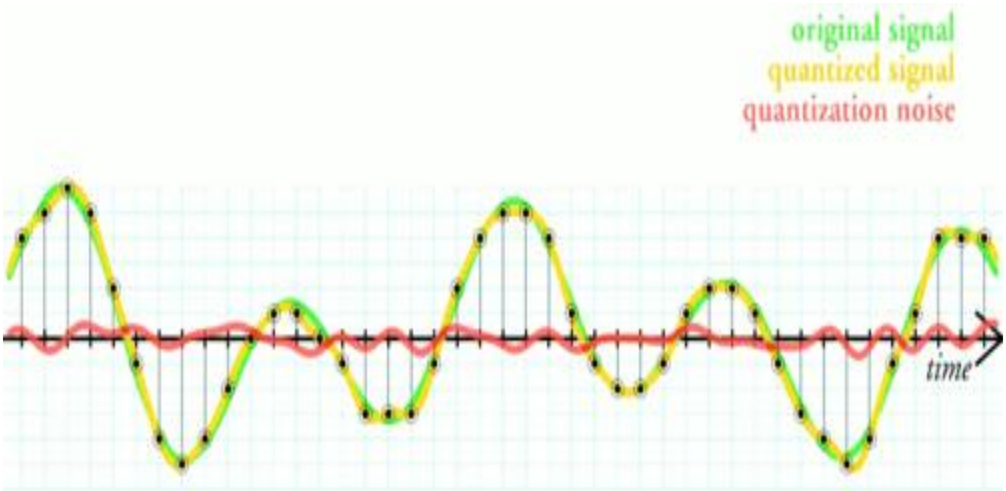
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ψηφιακές Τηλεπικοινωνίες



2ο Ερώτημα

ΑΚΑΔΗΜΑΙΚΟ ΈΤΟΣ 2015-2016

ΜΑΛΛΙΟΣ ΧΑΡΑΛΑΜΠΟΣ 5343

Λίγη Θεωρία ....

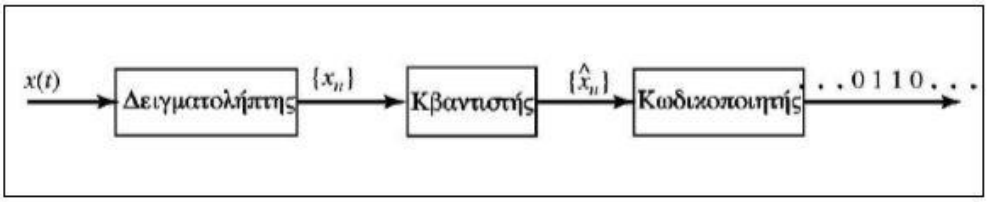
Η άσκηση ζητάει να υλοποιήσουμε έναν ομοιόμορφο κβαντιστή και ένα μη ομοιόμορφο κβαντιστή N bits δηλαδή 2^N επιπέδων χρησιμοποιώντας τον αλγόριθμο Lloyd-Max.

Η κωδικοποίηση PCM είναι το απλούστερο και παλιότερο σχήμα κωδικοποίησης κυματομορφής και αποτελείται από τα εξής υποσυστήματα

->Δειγματολήπτης

->Κβαντιστής

-> Κωδικοποιητής



Κβαντισμός (Quantization) ειναι η διαδικασία της προσέγγισης ενος αναλογικού (συνεχούς) δείγματος με ενα πεπερασμένο αριθμό bits.

Ο κβαντισμός εισάγει πάντοτε **παραμάρφωση**. Μπορούμε να την μειώσουμε αν αυξήσουμε τον αριθμό των bits με τα οποία παριστάνουμε ένα δείγμα.

Παραμορφωση

- Ο κβαντισμος εισαγει παραμορφωση στο σημα

$$X \longrightarrow \boxed{\text{Κβαντιστης}} \longrightarrow \widetilde{X} = f_Q(X)$$

- Θελουμε να ελαχιστοποιησουμε την μεση παραμορφωση D, οπου:

$$D = E[(X - \tilde{X})^2] = \int_{-\infty}^{\infty} (x - \tilde{x})^2 f(x)dx = \sum_{k=1}^{M} \int_{x_{k-1}}^{x_k} (x - \tilde{x}_k)^2 f(x)dx$$

οπου f(x) η pdf του σηματος.

- Αυτο το μετρο παραμορφωσης ονομαζεται και μεσο τετραγωνικο σφαλμα (MSE – Mean Square Error)
- Το MSE «μεγεθύνει» τα μεγαλα σφαλματα περισσοτερο απο τα μικρα.

Θόρυβος υπερφόρτωσης (Overload Noise):

- Εμφανίζεται όταν το σήμα εισόδου ειναι μεγαλύτερο απο το μεγαλύτερο επίεδο κβαντισμού με αποτέλεσμα τον «ψαλλιδισμό» του.

SQNR στην έξοδο του κβαντιστή :

- SNR= $(S/N)_{avg}$ = ισχυς σηματος x / ισχυς θορυβου n_Q

$$\left(\frac{S}{N}\right)_{avg} = \frac{E\left[X^2\right]}{E\left[n_Q^2\right]} = \frac{E\left[X^2\right]}{D} = \frac{\int_{-\infty}^{\infty} x^2 f_X(x)dx}{\int_{-\infty}^{\infty} (x - \widetilde{x})^2 f_X(x)dx}$$

Ζητούμενο 1

α)Υλοποιήσαμε τον ομοιόμορφο κβαντιστή σύμφωνα με το υπόμνημα.

```
function [Ax,kentra] = my_quantizer( x,N,minvalue,maxvalue )

%κάνω κανονικοποίηση στην είσοδο στην μέγιστη και ελάχιστη τιμή
input=x;
input(input>maxvalue)=maxvalue;
input(input<minvalue)=minvalue;





D =(maxvalue-minvalue)/2^(N);%το βήμα Δ


akra(1)=minvalue;
for j=2:2^(N)+1
    akra(j)=akra(j-1)+D; % υπολογίζω τα άκρα
end


for i=1:2^N
    kentra(i)=(akra(i)+akra(i+1))/2;% τα κέντρα
    kat = normcdf([kentra(i)-D/2 , kentra(i)+ D/2], 0, sqrt(1));

end

mikoseis=max(size(x,1),size(x,2));%μήκος εισόδου

Ax=zeros(mikoseis,1);%init
for j=1:mikoseis
    flag=true;
    k=1;
    while(flag)
        if(input(j)<=akra(k+1))
            Ax(j)=k;
            flag=false;
        else
            k=k+1;
        end
    end
end

kentra=kentra';
end
```

Αφού δημιουργήσαμε τη πηγή A

```
%erwtima a%

clear;
clc;

%δείγματα
M = 10000;
t = (randn(M,1)+j*randn(M,1))./sqrt(2);

%σήμα
Ax = abs(t).^2;

%το N
N = 4;
%N = 6;

%ελάχιστη και μέγιστη τιμή
elaxisti = 0;
megisti = 4;
```

Καλούμε τον κβαντιστή και υπολογίζουμε το SQNR

```
%καλώ τον ομοιόμορφο κβαντιστή
[xout,kentra] = my_quantizer( Ax,N,elaxisti,megisti )

%ισχύς σήματος
Psig=mean(Ax.^2);
%το κβαντισμένο σήμα
xquan=kentra(xout);
%ισχύς θορύβου
Pnois=mean((Ax-xquan).^2);

%sqnr υπολογισμένο σε db
sqnr=10*log10(Psig/Pnois)
```

β) και τέλος υπολογίζουμε τη παραμόρφωση θεωρητική και πειραματική και τη πιθανότητα υπερφόρτωσης

```
%πειραματική παραμόρφωση

paramorfwsi_peiramatiki = mean((Ax-kentra(xout)).^2);

%θεωρητική παραμόρφωση απο το ολοκλήρωμα
syms x;
```

```
paramorfwsi_theoritiki=0;


for i=1:2^N-1
apotel=int((x-kentra(i))^2*exp(-x),+(i-1)*((megisti-
elaxisti)/2^(N)),elaxisti+(i)*((megisti-elaxisti)/2^(N)));
paramorfwsi_theoritiki=paramorfwsi_theoritiki+eval(apotel);
end
ypol=eval(int((x-kentra(length(kentra)))^2*exp(-x),x,megisti,inf));
paramorfwsi_theoritiki=paramorfwsi_theoritiki+ypol;

%b πιθανότητα να ξεπεράσει τα όρια
distortion_overload = numel(Ax(Ax>megisti | Ax<elaxisti))/numel(Ax)
```

αποτελέσματα

Για N = 4

```
sqnr =

   16.8434


distortion_overload =

    0.0183

>>
>>
>>
>>
>> paramorfwsi_peiramatiki

paramorfwsi_peiramatiki =

    0.0420

>> paramorfwsi_theoritiki

paramorfwsi_theoritiki =

    0.0466
```

Για N = 6

```
sqnr =

   17.6114


distortion_overload =

    0.0176

>> paramorfwsi_peiramatiki

paramorfwsi_peiramatiki =

    0.0340

>> paramorfwsi_theoritiki

paramorfwsi_theoritiki =

    0.0381
```

Παρατηρούμε ότι όσο αυξάνεται το N αυξάνεται και η τιμή του SQNR και έτσι μεγαλώνει και η ισχύς του σήματος σε σχέση με την ισχύ του θορύβου.Αυτό σημαίνει πως κάθε φορά προσεγγίζουμε καλύτερα το αρχικό σήμα από το κβαντισμένο.

Επιπλέον παρατηρούμε ότι όσο αυξάνεται το N μειώνεται και η υπολογιζόμενη παραμόρφωση.Λογικό αφού είναι αντιστρόφως ανάλογη του sqnr.

Τέλος η θεωρητική παραμόρφωση που υπολόγισα προσομοιώνει αρκετά καλά τη πειραματική καθώς το σήμα ακολουθεί συμμετρική μιγαδική Gaussian κατανομή μέσης τιμής 0 και διασποράς 1

Ζητούμενο 2

α) Υλοποιήσαμε τον κβαντιστή βασισμένοι στο Loyd Max αλγόριθμο σύμφωνα με το υπόμνημα.

```
function [xq,centers,D,ranges]=lloyd_max_simple
(x,N,min_value,max_value)


%elegw oti brsketai se morfi vector
megethosx=size(x);
if megethosx(2)~=1 && megethosx(1)==1
    x=x';
elseif megethosx(2)~=1 && megethosx(1)>1
    error('not vector')
end


%kanonikopoisi  sth dynamikh perioxh
if min_value<min(x)
    min_value=min(x);
end
if max_value>max(x)
    max_value=max(x);
end


deiktis=x;%anafora
xq=zeros(size(x));%eksodos

centers_of_x=x;%ypologismos distortion


D=(max_value-min_value)/2^N; %bima


ranges=[min_value:D:max_value-D;min_value+D:D:max_value]';%epipeda
kbantisis

centers=zeros(size(ranges,1),1);%ypologismos kentrwn
deiktis(deiktis<min_value)=min_value;
deiktis(deiktis>max_value)=max_value;
for i=1:numel(centers)
```

```
    centers(i)=mean(deiktis(deiktis>=ranges(i,1) &
deiktis<=ranges(i,2)));

    centers_of_x(deiktis>=ranges(i,1) &
deiktis<=ranges(i,2))=centers(i);
end




D=[0 mean((x-centers_of_x).^2)]; %mesi paramorfwsi
j=2; % th prwth fora einai 0

while abs(D(j)-D(j-1))>=eps



    %aristero kai deksi akro
    ranges(:,1)=[min_value;(centers(1:end-1)+centers(2:end))/2];
    ranges(:,2)=[(centers(1:end-1)+centers(2:end))/2;max_value];
    for i=1:numel(centers)



        %kenra kathe perioxis
        centers(i)=mean(deiktis(deiktis>=ranges(i,1) &
deiktis<=ranges(i,2)));


        xq(deiktis>=ranges(i,1) & deiktis<=ranges(i,2))=i;
%kbantismos simatos

        centers_of_x(deiktis>=ranges(i,1) &
deiktis<=ranges(i,2))=centers(i);
    end


    D=[D mean((x-centers_of_x).^2)]; %nea epipeda paramorfwsis
    j=j+1; %epomeni epanalipsi
end
D=D(2:end);
xq(xq>=0 & xq<=0)=1;%afairw midenika
```

Υλοποιήσαμε τον κβαντιστή βασισμένοι στο Loyd Max αλγόριθμο σύμφωνα με το υπόμνημα με τη λογική του kmax επαναλήψεων .

```
function [xq,centers,D] = Lloyd_Max(x,N,minv,maxv,Kmax)

%epipeda kbantisis
```

```
epipeda = 2 ^ N;



%arxikopoihseis
new_x=zeros(size(x));
xq=zeros(size(x));

centers=zeros(epipeda,1);
new_centers=zeros(epipeda,1);

oria_zwnis = zeros(epipeda-1,1);
%paramorfwsi
D = zeros(Kmax,1);



%kanonikopoihsh eisodou
for j=1:size(x),
    if (x(j) > maxv)
        new_x(j) = maxv;
    elseif (x(j) < minv)
        new_x(j) = minv;
    else
        new_x(j) = x(j);
    end
end



%arxikopoioume ta oria zwnwn
for j = 1 : (epipeda-1)
    oria_zwnis(j) = j*(maxv - minv) / epipeda;
end
oria_zwnis=flipud(oria_zwnis);



if numel(new_x(new_x<=maxv & new_x>oria_zwnis(1))) > 0
new_centers(1) = mean(new_x(new_x<=maxv & new_x>oria_zwnis(1)));% an
to x anhkei sto max vres to kentro

else
    new_centers(1) = (oria_zwnis(1)+maxv)/2;%an oxi valto sto kentro
end


if numel(new_x(new_x<=oria_zwnis(epipeda-1) & new_x>=minv))
    new_centers(epipeda) = mean(new_x(new_x<=oria_zwnis(epipeda-1) &
new_x>=minv));% an to x anhkei sto min vres to kentro

else
    new_centers(epipeda) = (oria_zwnis(epipeda-1)+minv)/2;%an oxi
valto sto kentro
end

for j = 2 : (epipeda-1) %gia oles tis alles zwnes >>
```

```
    if numel(new_x(new_x<=oria_zwnis(j-1) & new_x>oria_zwnis(j))) > 0
        new_centers(j) = mean(new_x(new_x<=oria_zwnis(j-1) &
new_x>oria_zwnis(j)));
    else
        new_centers(j) = (oria_zwnis(j-1)+oria_zwnis(j))/2;
    end
end

k=2;
D(1) = 0;
D(2) = 1;



while k < Kmax && norm( D(k) - D(k-1) ) >= 10^-16      %oso h diafora
de peftei katw apo to eps(10.^-16)
    k = k + 1;
    centers=new_centers;
    for j = 1 : (epipeda-1),
        oria_zwnis(j) = (centers(j) + centers(j+1)) / 2;
%epanaypologizw ta akra
    end


    xq(new_x<=maxv & new_x>oria_zwnis(1)) = 1; %to kbantismeno
    xq(new_x<=oria_zwnis(epipeda-1) & new_x>=minv) = epipeda;

    for j = 2 : (epipeda-1),
        xq(new_x<=oria_zwnis(j-1) & new_x>oria_zwnis(j)) = j;
    end

    D(k) = mean( (new_x - centers(xq)) .^ 2 );%paramorfwsi



    if numel(new_x(new_x<=maxv & new_x>oria_zwnis(1))) > 0
        new_centers(1) = mean(new_x(new_x<=maxv &
new_x>oria_zwnis(1)));%ananewnw to akrotato center
    else
        new_centers(1) = centers(1);
    end
    if numel(new_x(new_x<=oria_zwnis(epipeda-1) & new_x>=minv))
        new_centers(epipeda) = mean(new_x(new_x<=oria_zwnis(epipeda-
1) & new_x>=minv));%ananewnw to katwtato center
    else
        new_centers(epipeda) = centers(epipeda);
    end

    for j = 2 : (epipeda-1),
        if numel(new_x(new_x<=oria_zwnis(j-1) & new_x>oria_zwnis(j)))
> 0
            new_centers(j) = mean(new_x(new_x<=oria_zwnis(j-1) &
new_x>oria_zwnis(j))); %gia ola ta alla
       else

        end
    end
end
D = D';
```

α) Αφού διαβάσαμε το σήμα φωνής υπολογίζουμε για καθε n = 2,4,6 τα αντίστοιχα sqnr

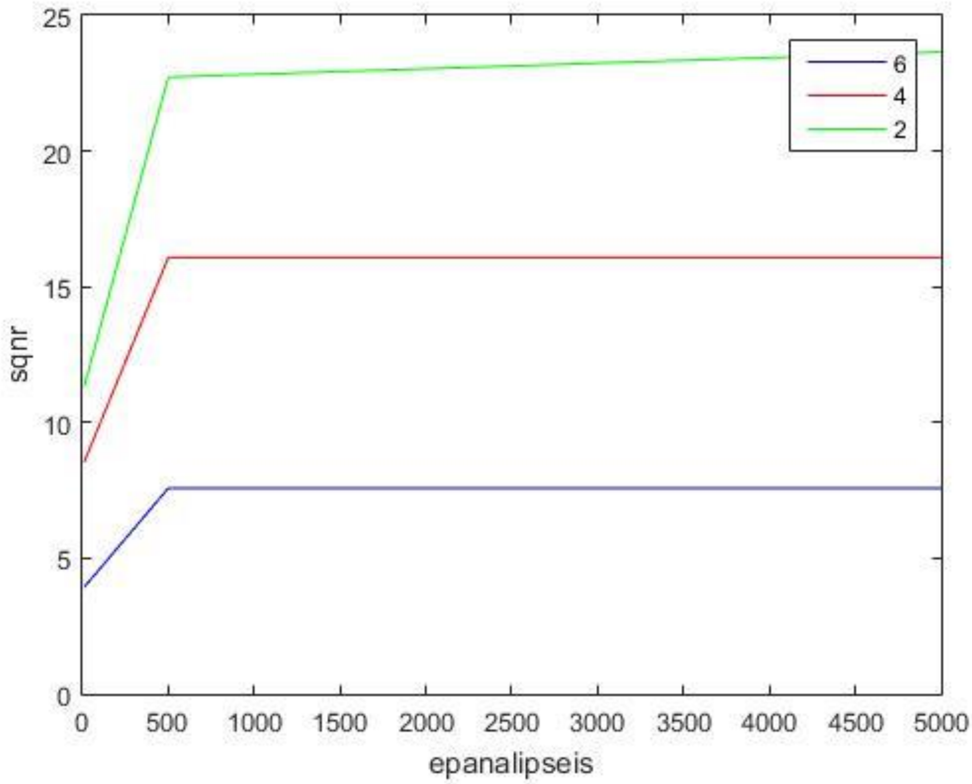
```
clear
clc

         %---------------loyd --------------------------
bn = [10 500 5000]
[y,Fs] = audioread('speech.wav');

%------------ n =2 ---------------------
[Aq1,centers1,D]=Lloyd_Max(y,2,min(y),max(y),10);

 Psig1=mean(y.^2);
 ykbant1=centers1(Aq1); %to kvantismeno shma
 Pnois1 = mean((y-ykbant1).^2);

 %upologismos tou sqnr se db
 loydsqnr1=10*log10(Psig1/Pnois1);

 [Aq2,centers2,D]=Lloyd_Max(y,2,min(y),max(y),500);

 Psig2=mean(y.^2);
 ykbant2=centers2(Aq2); %to kvantismeno shma
 Pnois2 = mean((y-ykbant2).^2);

 %upologismos tou sqnr se db
 loydsqnr2=10*log10(Psig2/Pnois2);

 [Aq3,centers3,D]=Lloyd_Max(y,2,min(y),max(y),5000);

 Psig3=mean(y.^2);
 ykbant3=centers3(Aq3); %to kvantismeno shma
 Pnois3 = mean((y-ykbant3).^2);

 %upologismos tou sqnr se db
 loydsqnr3=10*log10(Psig3/Pnois3);



 loydsqnr_2 = [loydsqnr1 loydsqnr2 loydsqnr3 ];
```

```
%------------ n =4 --------------------

[y,Fs] = audioread('speech.wav');


[Aq1,centers1,D]=Lloyd_Max(y,4,min(y),max(y),10);

 Psig1=mean(y.^2);
 ykbant1=centers1(Aq1); %to kvantismeno shma
 Pnois1 = mean((y-ykbant1).^2);

 %upologismos tou sqnr se db
 loydsqnr1=10*log10(Psig1/Pnois1);

 [Aq2,centers2,D]=Lloyd_Max(y,4,min(y),max(y),500);

 Psig2=mean(y.^2);
 ykbant2=centers2(Aq2); %to kvantismeno shma
 Pnois2 = mean((y-ykbant2).^2);

 %upologismos tou sqnr se db
 loydsqnr2=10*log10(Psig2/Pnois2);

 [Aq3,centers3,D]=Lloyd_Max(y,4,min(y),max(y),5000);

 Psig3=mean(y.^2);
 ykbant3=centers3(Aq3); %to kvantismeno shma
 Pnois3 = mean((y-ykbant3).^2);

 %upologismos tou sqnr se db
 loydsqnr3=10*log10(Psig3/Pnois3);

 loydsqnr_4 = [loydsqnr1 loydsqnr2 loydsqnr3 ];



 %------------ n = 6 --------------------

[y,Fs] = audioread('speech.wav');



[Aq1,centers1,D]=Lloyd_Max(y,6,min(y),max(y),10);

 Psig1=mean(y.^2);
 ykbant1=centers1(Aq1); %to kvantismeno shma
 Pnois1 = mean((y-ykbant1).^2);

 %upologismos tou sqnr se db
 loydsqnr1=10*log10(Psig1/Pnois1);

 [Aq2,centers2,D]=Lloyd_Max(y,6,min(y),max(y),500);

 Psig2=mean(y.^2);
 ykbant2=centers2(Aq2); %to kvantismeno shma
```

```
Pnois2 = mean((y-ykbant2).^2);

%upologismos tou sqnr se db
loydsqnr2=10*log10(Psig2/Pnois2);

[Aq3,centers3,D]=Lloyd_Max(y,6,min(y),max(y),5000);

Psig3=mean(y.^2);
ykbant3=centers3(Aq3); %to kvantismeno shma
Pnois3 = mean((y-ykbant3).^2);

%upologismos tou sqnr se db
loydsqnr3=10*log10(Psig3/Pnois3);


loydsqnr_6 = [loydsqnr1 loydsqnr2 loydsqnr3 ];

 loydsqnr = [loydsqnr_2 loydsqnr_4 loydsqnr_6] ;




plot(bn,loydsqnr_2,'b',bn,loydsqnr_4,'r',bn,loydsqnr_6,'g');
         legend('6','4','2')
         xlabel('epanalipseis')
        ylabel('sqnr')
```

τα κάνουμε plot για να δούμε τι συμβαίνει γραφικά

```
plot(bn,loydsqnr_2,'r',bn,loydsqnr_4,'g',bn,loydsqnr_6,'b');
         legend('2','4','6')
         xlabel('epanalipseis')
        ylabel('sqnr')
```



Καθώς αυξάνεται ο αριθμός των bits(N) αυξάνεται και το sqnr καθώς μεγαλώνει και η ισχύς του σήματος σε σχέση με την ισχύ του θορύβου.

β) Για τη πηγή Β χρησιμοποιώ τώρα και τον ομοιόμορφο προκειμένου να υπολογίσω τα sqnr και να τα συγκρίνω με τον μη ομοιόμορφο

```
%----------------------------------------------
%omoiomorfo
%---------------------------------------------------------

%diabazw to sima
[y,fs]=audioread('speech.wav');

  %-----------N =2 ---------------%
%gia ton omoiomorfo me min = -1 max = 1
[xq1,centers1]=my_quantizer(y,2,min(y),max(y));
%ypologismos toy sqnr

 Psig=mean(y.^2);
 ykbant=centers1(xq1); %to kvantismeno shma
 Pnois = mean((y-ykbant).^2);

%upologismos tou sqnr se db
omoiomsqnr1=10*log10(Psig/Pnois);


  %-----------N = 4  ---------------%


[xq2,centers2]=my_quantizer(y,4,min(y),max(y));
%ypologismos toy sqnr

 Psig=mean(y.^2);
 ykbant=centers2(xq2); %to kvantismeno shma
 Pnois = mean((y-ykbant).^2);

%upologismos tou sqnr se db
omoiomsqnr2=10*log10(Psig/Pnois);


  %-----------N = 6 ---------------%

[xq3,centers3]=my_quantizer(y,6,min(y),max(y));
%ypologismos toy sqnr

 Psig=mean(y.^2);
 ykbant=centers3(xq3); %to kvantismeno shma
 Pnois = mean((y-ykbant).^2);
```

```
%upologismos tou sqnr se db
omoiomsqnr3=10*log10(Psig/Pnois);

omoiomsqnr = [omoiomsqnr1, omoiomsqnr2 ,omoiomsqnr3];
bn = [10,15,25];
```

Αποτελέσματα

```
omoiomsqnr =

    1.3280   11.9372   25.0258

>> loydsqnr

loydsqnr =

    3.9338    7.5779    7.5779    8.5443   16.0772   16.0772   11.3215   22.7231   23.6484

>> |
```

Βλέπουμε ότι ο μη ομοιόμορφος αποδίδει καλύτερα απο τον ομοιόμορφο Το φαινόμενο αυτό είναι αναμενόμενο γιατί η αύξηση του αριθμού διαστημάτων όσο το εύρος τιμών παραμένει σταθερό έχει σαν αποτέλεσμα το μήκος τους να είναι μικρότερο και άρα το σφάλμα κβάντισης(που στη χειρότερη περίπτωση είναι ίση με το μισό του μήκους του διαστήματος) να είναι μικρότερο. Επίσης, το γεγονός ότι ο μη ομοιόμορφος κβαντιστής πετύχαινει μεγαλύτερο SQNR σημαινει ότι πετυχαίνει και καλύτερα αποτελέσματα.

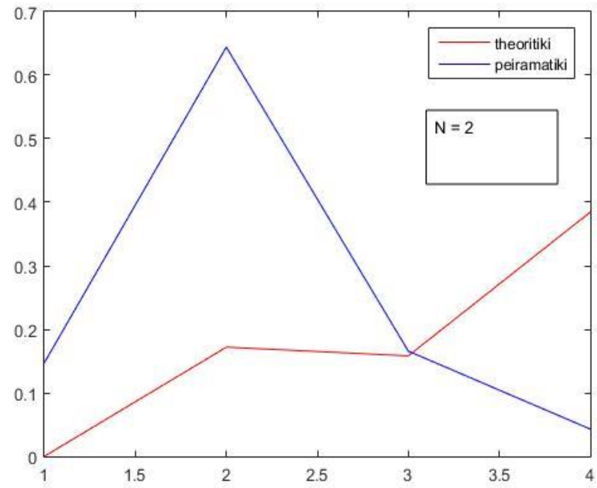
γ) Δημιουργώ τη συνάρτηση pithanotita_stathmis η οποία υπολογίζει θεωρητικά τη πιθανότητα εμφάνισης κάθε στάθμης .

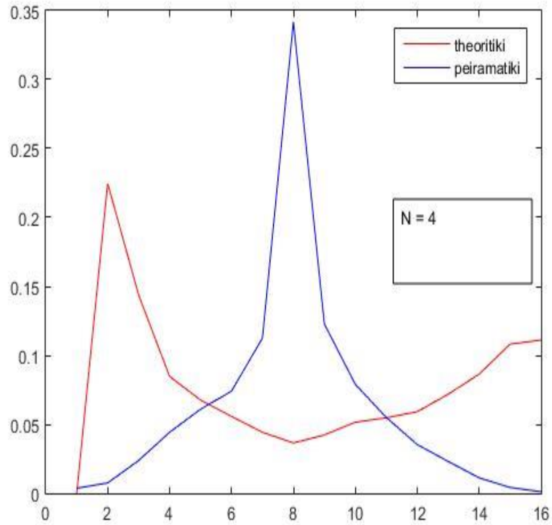
```
function[pithanotita,entropia]=pithanotita_stathmis(x,N,min_value,max
_value)



%αρχικοποίηση
syms s;
pithanotita = zeros(2^N-1,1);



[~, ~, ~, oria] = lloyd_max_simple(x, N, min_value, max_value);

    %ολοκλήρωμα απο 1 μέχρι +άπειρο
    pithanotita(end) = double(int(exp(-s),1,Inf));


        j = 1 ;
        while j <= size(oria,1)
        pithanotita(j) = double(int(exp(-s),oria(j,1),oria(j,2)));
%υπόλοιπες πιθανότητες
        j = j+1 ;
        end

   pithanotita(1) = 0;%μέχρι το 0 η πιθανότητα είναι 0



end
```

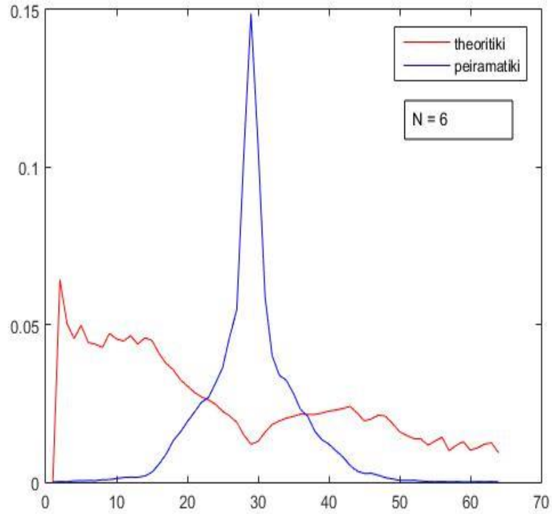
Υπολογίζω πειραματικά τη πιθανότητα εμφάνισης κάθε στάθμης όπως ζητείται στην εκφώνηση.

```
clear;
clc ;


%διαβάζω το σήμα
y = audioread('speech.wav');
%τα N
N = 4; %4,6


%--------------πειραματικός υπολογισμός------------------------------
-----------------------------%

[y1,kentra1,p1]=lloyd_max_simple(y,N,min(y),max(y));
%εύρεση στοιχείων
stoixeia1 = unique(kentra1(y1));

%από το ιστόγραμμα βρίσκω εμφανίσεις
[emfaniseis1,stoixeiwn1] = hist(kentra1(y1),stoixeia1);
%πιθανότητα = εμφανίσεις / μήκος του σήματος
pithanotita_peiramatiki1 =(emfaniseis1/length(y1))
```

τα συγκρίνω με plot για να δώ πόσο διαφέρουν


theoritiki
peiramatiki
N = 2


theoritiki
peiramatiki
N = 4


theoritiki
peiramatiki
N = 6

Παρατηρούμε ότι οι θεωρητικές με τις πειραματικές πιθανότητες έχουν αισθητή απόκλιση, γεγονός που οφείλεται στην Laplace κατανομή του σήματος και όχι γκαουσιανή. Τέλος υπολογίζω την εντροπία :

```
%υπολογισμός εντροπίας
Entropia_peiramatiki = -
pithanotita_peiramatiki1*log2(pithanotita_peiramatiki1)';
entropia_theoritiki= -
pithanotita_theoritiki*log2(pithanotita_theoritiki)';% εντροπία
```

N = 2

```
Entropia_peiramatiki =

    1.4401
```

N = 4

```
>> Entropia_peiramatiki

Entropia_peiramatiki =

    3.1206
```

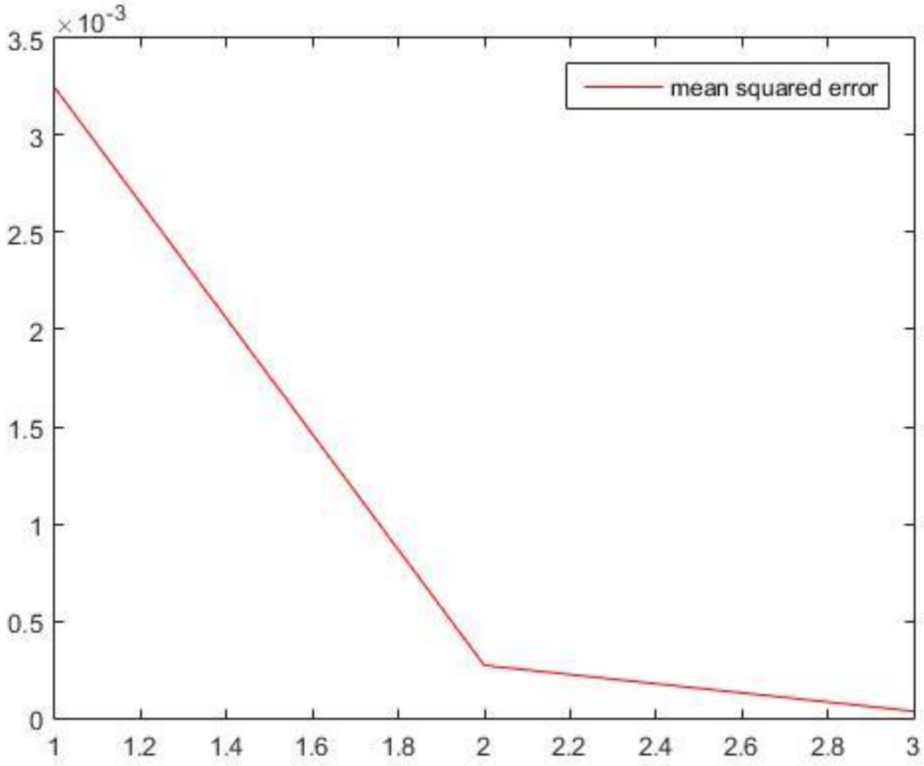
N = 6

```
Entropia_peiramatiki =

    4.4997

>>
```

Όσο λοιπόν αυξάνεται το n αυξάνεται και η εντροπία, πράγμα λογικό αφού αυξάνεται η απόδοση του κβαντιστή.

δ)υπολογίζω το Mean square error για κάθε N από το κβαντισμένο και τα επίπεδα κβάντισης

```
[y,Fs] = audioread('speech.wav');

%------------ n =2 --------------------
[Aq1,centers1,D]=lloyd_max_simple(y,2,-1,1);


 ykbant1=centers1(Aq1); %to kvantismeno shma
 mse2 = mean((y-ykbant1).^2);

 %------------ n =4 --------------------

[Aq2,centers2,D2]=lloyd_max_simple(y,4,-1,1);


 ykbant2=centers2(Aq2); %to kvantismeno shma
 mse4 = mean((y-ykbant2).^2);



 %------------ n = 6 --------------------

[Aq3,centers3,D3]=lloyd_max_simple(y,6,-1,1);


 ykbant3=centers3(Aq3); %to kvantismeno shma
 mse6 = mean((y-ykbant3).^2);

 mse = [mse2 mse4 mse6]




plot(mse','r')
hold on
legend('mean squared error')
```




```
>> mse2

mse2 =

    0.0032

>> mse4

mse4 =

   2.7027e-04

>> mse6

mse6 =

   3.4355e-05
```

Παρατηρώ ότι το MSE μειώνεται όσο αυξάνονται τα επίπεδα κβάντισης . Άρα η PCM είναι πιο αποδοτική καθώς αυξάνεται το n .Αυτό είναι λογικό καθώς μειώνεται η διαφορά μεταξύ του σήματος και του κβαντισμένου.