ΨΗΦΙΑΚΕΣ ΤΗΛΕΠΙΚΟΙΝΩΝΙΕΣ

2014 – 2015

1η Εργαστηριακή Άσκηση

Γεώργιος-Κάρολος Λύκος 4758

(6ο έτος)

Όλοι κώδικες που αναφέρονται ονομαστικά στα παρακάτω ζητούμενα υλοποιήθηκαν στο περιβάλλον της Matlab και παραθέτονται στο τμήμα ***Κώδικες.***

1. Υλοποιήστε τα παρακάτω σχήματα :

a. PCM με ομοιόμορφο κβαντιστή

Τα ζητούμενα σχήματα υλοποιήθηκαν σε function της *MATLAB* με ονομασία *my_quantizer.m* ,για τον ομοιόμορφο κβαντιστή, και *Lloyd_Max.m* ,για τον μη ομοιόμορφο κβαντιστή, οπου οι κώδικες παραθέτονται στο τμήμα ***Κώδικες*** .
Στην περίπτωση του ομοιόμορφου κβαντιστή ο κώδικας είναι απλός , υπολογίζω τα όρια για όλες τις περιοχές και τα κέντρα τους ως το μέσον των ορίων. Εάν η τιμή του σήματος ανήκει μέσα στην περιοχή θέτω στο διάνυσμα **xq** τον αριθμό της περιοχής αυτής. Αυτό το διάνυσμα σε συνδυασμό με τα κέντρα που επιστρέφονται είναι αρκετά για τον ανακατασκευή του κβαντισμένου σήματος στην έξοδο.

Για την περίπτωση του μη ομοιόμορφου κβαντιστή χρησιμοποιήθηκε όπως υποδεικνύει και η εκφώνηση ο αλγόριθμος των Lloyd και Max.

Ουσιαστικά αρχικοποιούμε τα όρια των περιοχών όπως και στον ομοιόμορφο κβαντιστή όμως τώρα χρησιμοποιούμε μια μετρική για να βρούμε το κέντρο βάρους κάθε περιοχής ώστε να θέσουμε αυτό ως κέντρο της και όχι το μέσο των ορίων. Όταν κάποια περιοχή είναι κενή το οποίο συμβαίνει για μεγάλο N (για 8 bits πχ) θέτω ως κέντρο τον μέσο των ορίων της περιοχής αυτής (όπως και στον ομοιόμορφο). Με τα καινούργια κέντρα υπολογίζουμε πάλι τα όρια ως το μέσον της απόστασης μεταξύ δύο γειτονικών κέντρων για όσες επαναλήψεις χρειάζονται μέχρι η παραμόρφωση να είναι σχεδόν ίδια με την παραμόρφωση του σήματος στην προηγούμενη επανάληψη. Λέγοντας σχεδόν ίδια εννοώ η διαφορά τους να είναι μικρότερη από μια τιμή (κατώφλι), στην περίπτωση μας το βρήκα ότι είναι αρκετά ικανοποιητικό.
Παραπάνω ανέφερα ότι χρησιμοποίησα μια μετρική για τον υπολογισμό του κέντρου βάρους κάθε περιοχής. Μετά από συζήτηση με τους υπεύθυνους του μαθήματος κατέληξα ότι ο μέσος όρος είναι μια αρκετά ικανοποιητική και πρακτικά υλοποιήσιμη τεχνική υποθέτοντας ότι το σήμα μας είναι εργοδικό.

2. Κωδικοποιείστε τα δείγματα της πηγής χρησιμοποιώντας τα παραπάνω σχήματα για N=2,4 και 8 bits. Αξιολογείστε τις παραπάνω μεθόδους βασισμένοι:

a. Στις τιμές του SQNR. Για την περίπτωση του δεύτερου σχήματος να σχεδιάσετε το πως μεταβάλλεται το SQNR σε σχέση με τον αριθμό των επαναλήψεων του αλγορίθμου Lloyd Max.

b. Στο ακουστικό αποτέλεσμα κάθε μεθόδου χρησιμοποιώντας την wavplay() .

c. Στις κυματομορφές εξόδου.

a) Κωδικοποίησα την πηγή για N = 2,4 και 8 bits , με την εκτέλεση της συνάρτησης *PCM.m και* ADM.*m αντιστοιχα* εκτυπώνονται για το SQNR τα παρακάτω αποτελέσματα :

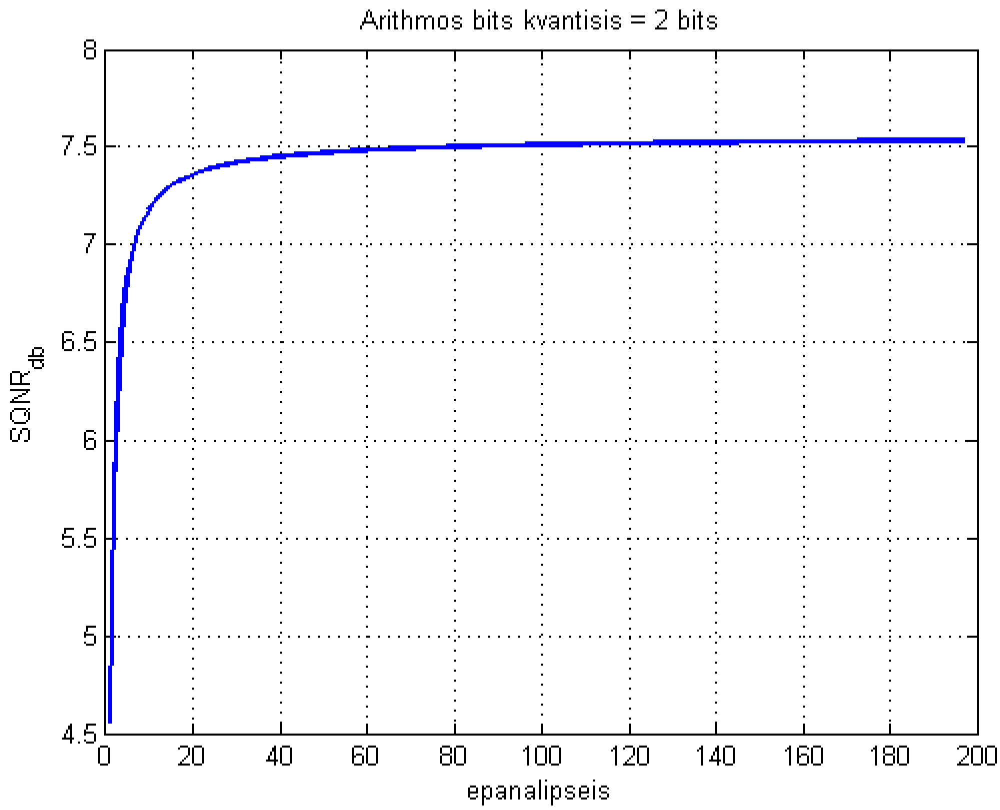
SQNR(decibel)	N = 2 bits	N = 4 bits	N = 8 bits
PCM	7.5328	15.7587	37.2394

Το SQNR το οποίο είναι ο λόγος της ισχύς του σήματος προς την ισχύ του θορύβου, εκφράζει πόσο σημαντικό είναι το σήμα προς τον θόρυβο. Στην περίπτωσή μας ο θόρυβος προκαλείται από την κβάντιση του σήματος εισόδου. Επίσης είναι σε κλίμακα db(decibel).

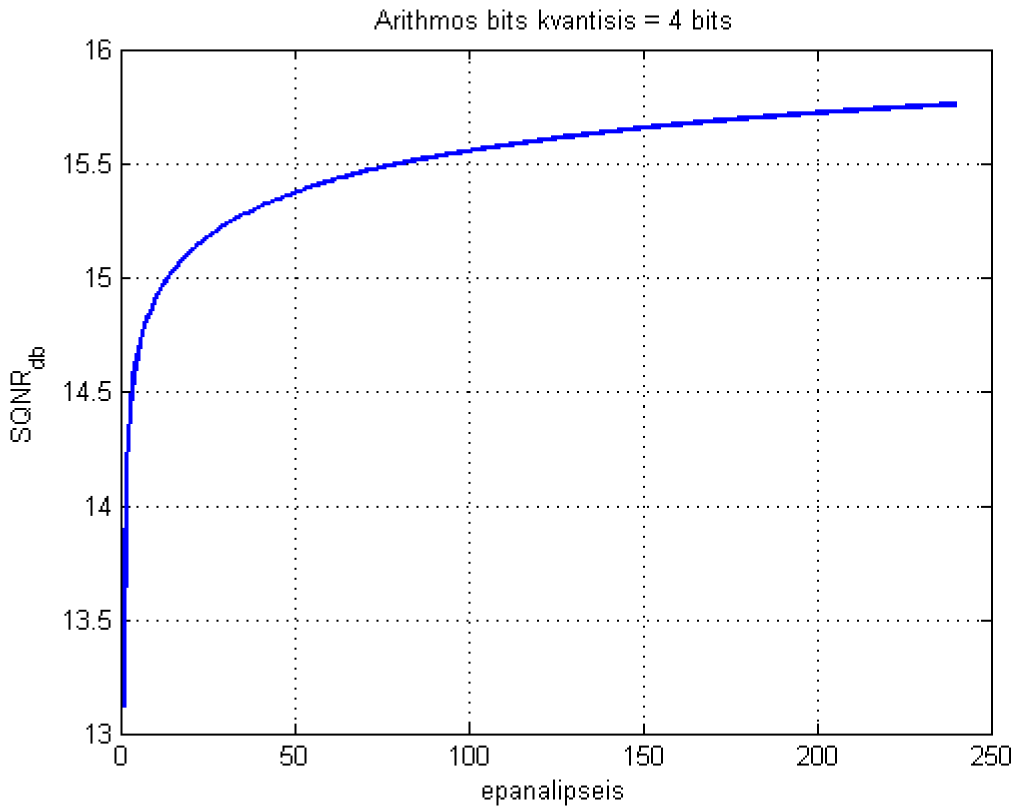
Το πρώτο αποτέλεσμα δηλώνει το SQNR για τον ομοιόμορφο κβαντιστή με 2 bits κβάντισης. Το αποτέλεσμα είναι 7.5328 και καταλαβαίνουμε ότι η ισχύς του θορύβου κβάντισης είναι περίπου το ίδιο σημαντική με το σήμα στην είσοδο, κάτι το οποίο σημαίνει ότι στην έξοδο θα έχουμε ένα αρκετά αλλοιωμένο αποτέλεσμα.

Στα επόμενα αποτελέσματα βλέπουμε ότι με αύξηση των bit κβάντισης σε 4 bits το SQNR αυξάνεται σημαντικά σε 15.7587. Η ισχύς του σήματος μας είναι 15 φορές περίπου πιο σημαντική από αυτήν του θορύβου. Λογικό να έχουμε τέτοια αλλαγή αφού τώρα ο κβαντιστής μας λειτουργεί με   περιοχές άρα και έχει μεγαλύτερη ακρίβεια .Για N = 8 το SQNR είναι 37.2394. .Βλέπουμε ότι η ισχύς του σήματος γίνεται πολύ πιο σημαντική από αυτήν του θορύβου κατά 37 φορές περίπου.

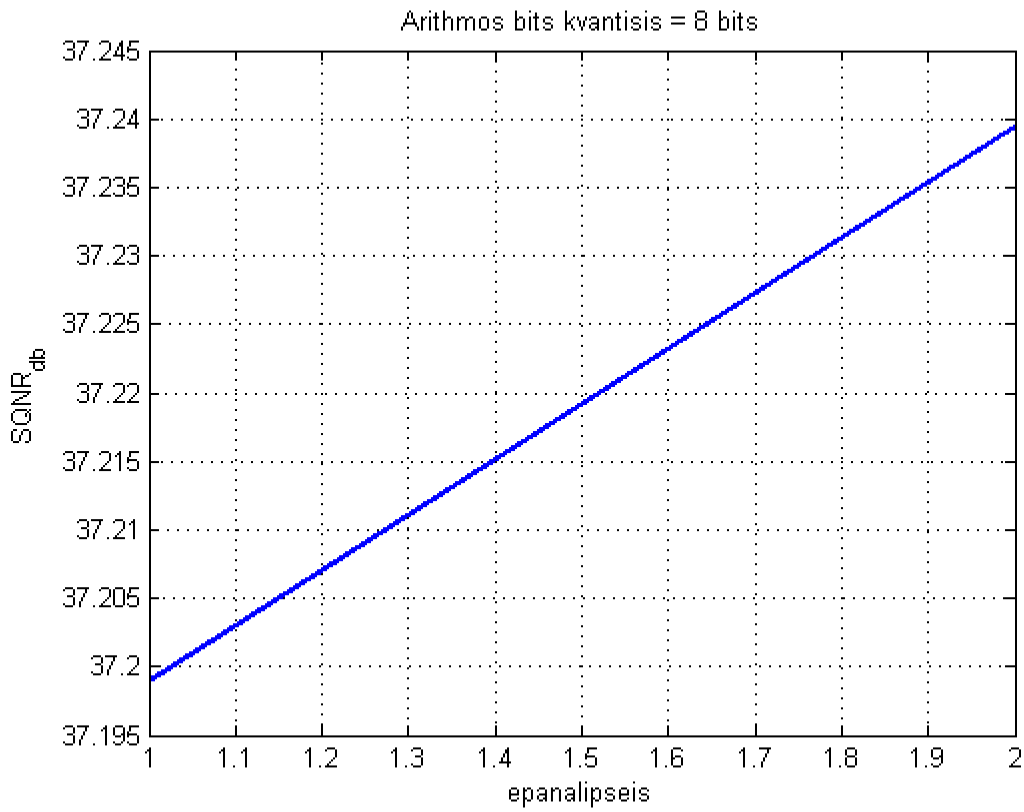
Τώρα ζητείται γραφική παράσταση για το πώς μεταβάλεται το SQNR σε σχέση με τον αριθμό επαναλήψεων του αλγορίθμου *Lloyd Max*.



Εδώ παρατηρούμε πάλι ραγδαία αύξηση του SQNR για τις πρώτες επαναλήψεις και μετά σταδιακή αύξηση μέχρι να φτάσει την μέγιστη τιμή στο 7.5 περίπου μετά από περίπου 200 επαναλήψεις.



Εδώ παρατηρούμε πάλι ραγδαία αύξηση του SQNR για τις πρώτες επαναλήψεις και μετά σταδιακή αύξηση μέχρι να φτάσει την μέγιστη τιμή στο 15.7 περίπου μετά από περίπου 230 επαναλήψεις.



Εδώ παρατηρούμε ότι το SQNR δεν συγκλίνει ακόμα σε κάποια τιμή αλλά αυξάνεται αναλογικά. Δυστυχώς αυτό οφείλεται στο κατώφλι που χρησιμοποιήθηκε αφού βλέπουμε ότι τερμάτισε ο αλγόριθμος μετά από μόλις 2 επαναλήψεις. Ίσως με ένα μικρότερο κατώφλι να βλέπαμε την σύγκλιση .

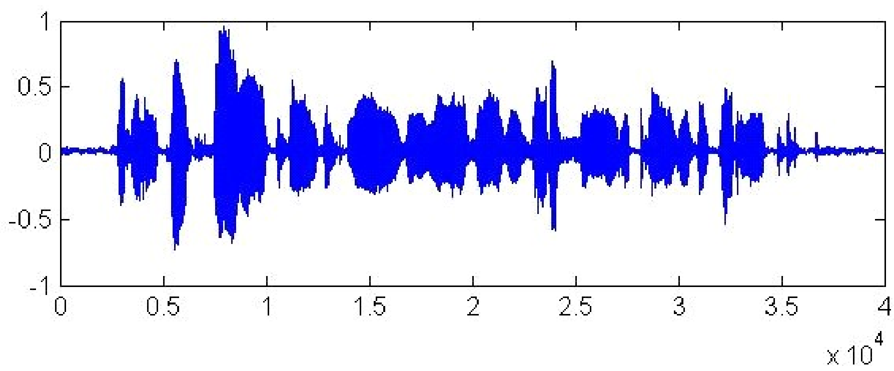
b) Χρησιμοποιώντας την συνάρτηση *wavplay()* της *Matlab* παρατηρούμε τα εξής :
Για **2 bits** κβαντισης το σήμα εξόδου ακούγεται με πολύ θόρυβο.

Για τα **4 bits** κβάντισης ο θόρυβος είναι πάλι αισθητός στο σήμα εξόδου κ ,βέβαια τα πράγματα είναι πολύ καλύτερα από τα 2 bits..

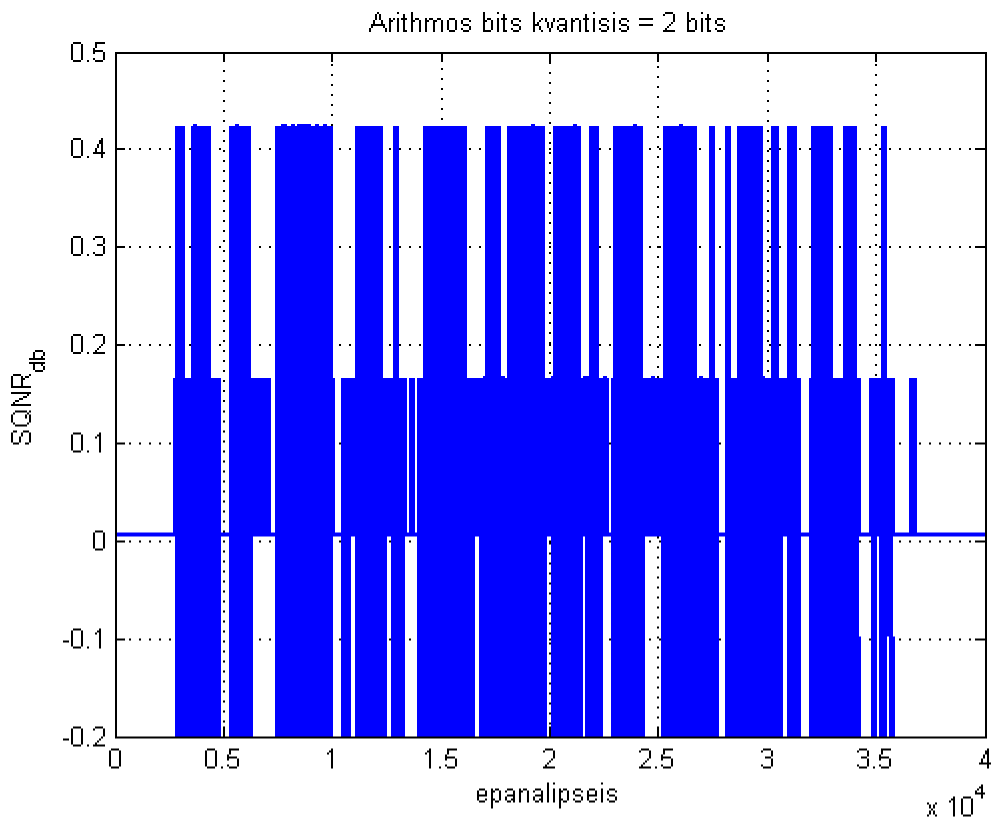
Για τα **8 bits** το σήμα εξόδου ακουστικά είναι ίδιο με το σήμα εισόδου .

c) Τα υπόλοιπα αποτελέσματα που εκτυπώνονται από την εκτέλεση του script pcm.*m που καλεί την συνάρτηση Lloyd_Max.m που με την σειρά της εκτυπώνει τις παρακάτω κυματομορφές* (τμήμα Κώδικες) παρακάτω.

Αρχικά έχουμε το σήμα εισόδου που θέλουμε να κωδικοποιήσουμε.

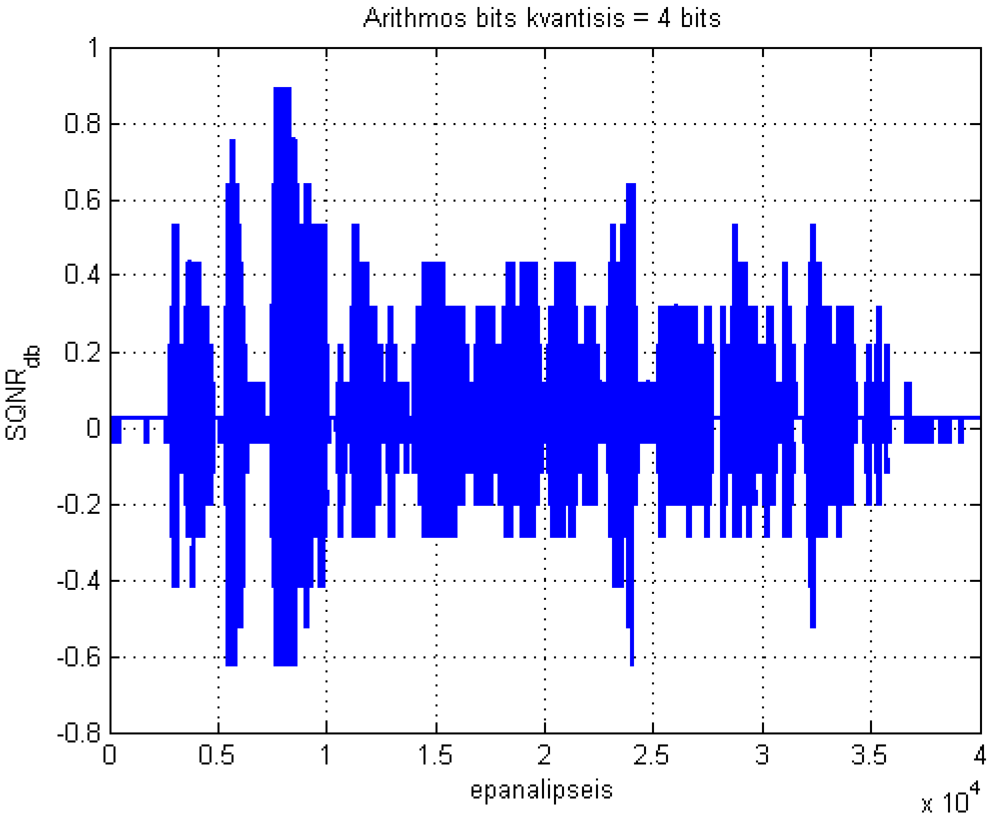


Για N = 2 bits έχουμε τα εξής αποτελέσματα,



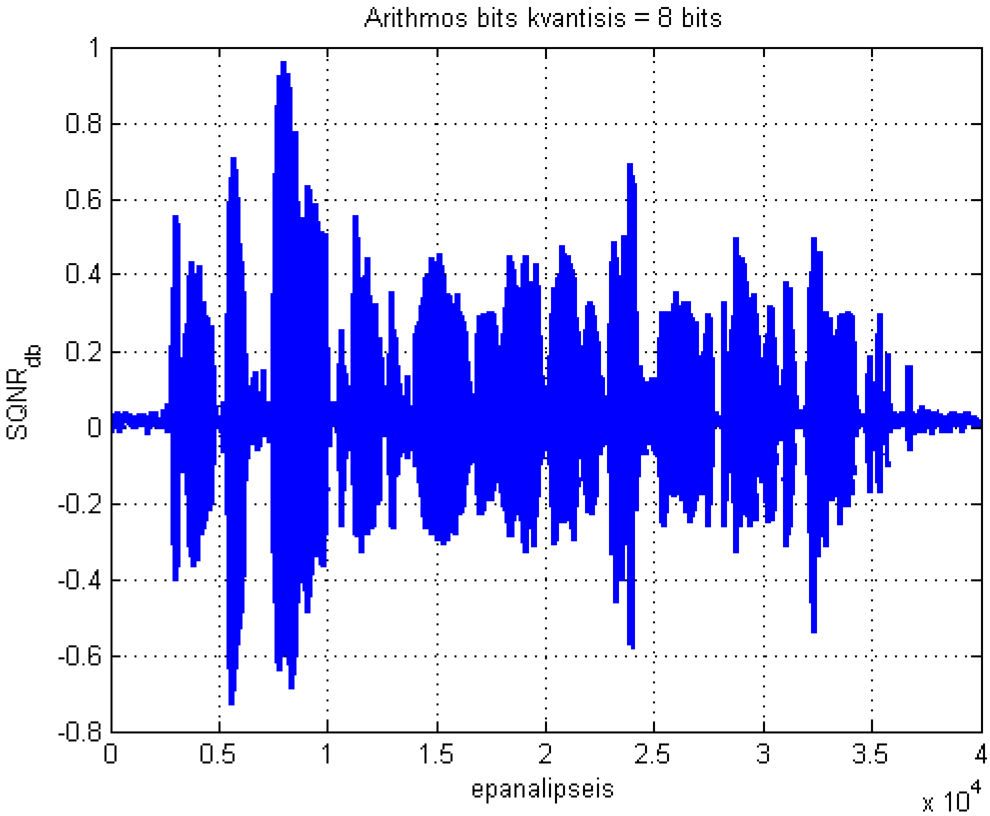
Όπως φαίνεται και στα σχήματα το σήμα εξόδου δεν είναι κοντά στο αρχικό, κάτι το οποίο επιβεβαιώνεται και από τα ακουστικά αποτελέσματα και από τις τιμές του SQNR στα προηγούμενα ερωτήματα.

Για N = 4 bits ,



Εδώ παρατηρούμε ότι το σήμα εξόδου είναι σαφώς πιο κοντά στο αρχικό μας.Αλλά φαίνεται η διαφορά του σήματος λόγω του κβαντισμένου σήματος με 4 bits και οιτι οι περιοχές δεν είναι τόσο ακριβής. Και πάλι επιβεβαιώνονται οι παρατηρήσεις μας στα προηγούμενα ερωτήματα.

Για N = 8 bits ,



Χρησιμοποιώντας 8 bits κβάντισης παρατηρούμε ότι το σήμα που βγαίνει από τους κβαντιστές είναι πανομοιότυπο (οπτικά τουλάχιστον) στο αρχικό μας σήμα.

2.Κωδικοιποιείστε τα δείγματα της πηγής B χρησιμοποιώντας το σχήμα PCM για N=2 και 4 bits .Αξιολογείστε τις παραπάνω μεθόδους βασισμένοι:

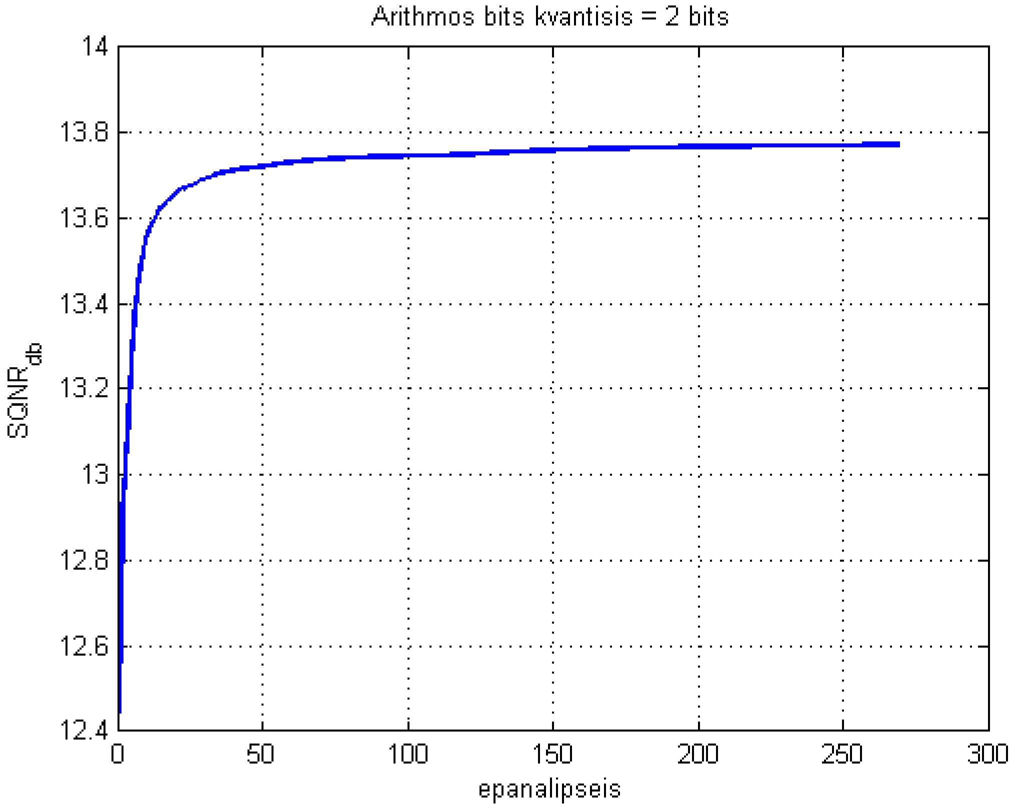
α.Στις τιμές του SQNR .Σχεδιάστε το πως μεταβάλλεται το SQNR σε σχέση με τον αριθμό των επαναλήψεων του αλγορίθμου Lloyd Max.

Χρησιμοποιείται το script ask2.*m* (τμήμα Κώδικες).

a) Κωδικοποίησα την πηγή για N = 2 και 4 bits , με την εκτέλεση της συνάρτησης ask2.*m αντιστοιχα* εκτυπώνονται για το SQNR τα παρακάτω αποτελέσματα :

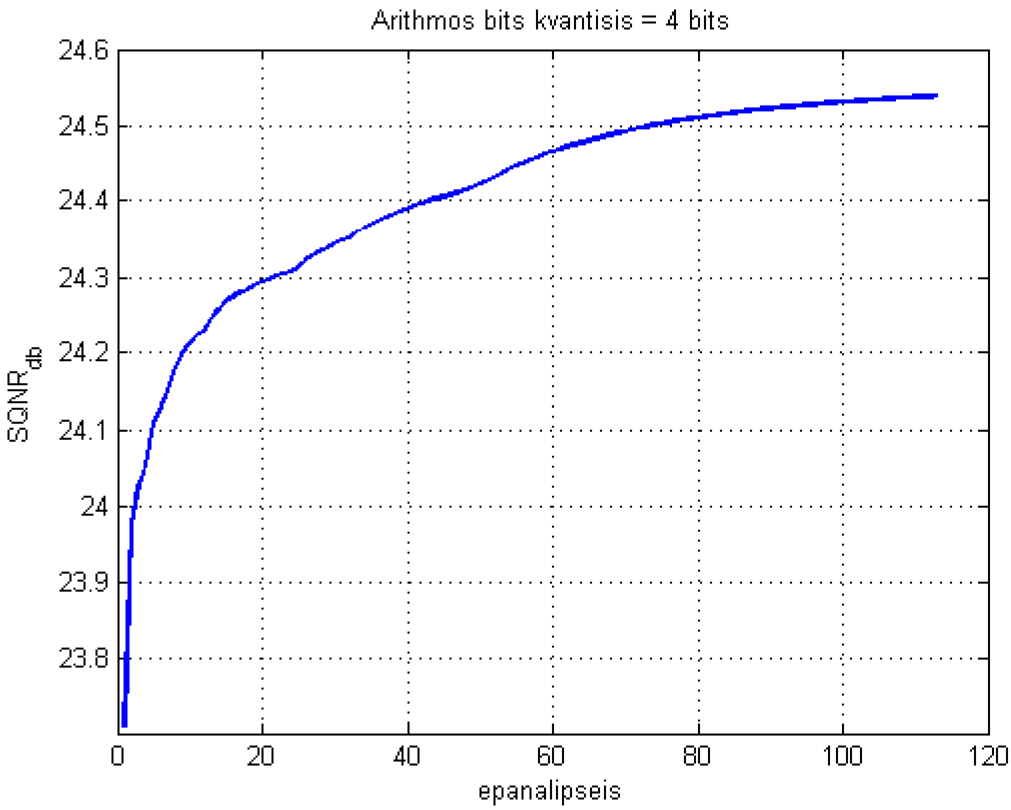
SQNR(decibel)	N = 2 bits	N = 4 bits
PCM	13.7713	24.5385

Για N = 2



Εδώ παρατηρούμε πάλι ραγδαία αύξηση του SQNR για τις πρώτες επαναλήψεις και μετά σταδιακή αύξηση μέχρι να φτάσει την μέγιστη τιμή στο 13.8 περίπου μετά από περίπου 250 επαναλήψεις.

Για N = 4 bits ,



Εδώ παρατηρούμε πάλι ραγδαία αύξηση του SQNR για τις πρώτες επαναλήψεις και μετά σταδιακή αύξηση μέχρι να φτάσει την μέγιστη τιμή στο 24.5περίπου μετά από περίπου 100 επαναλήψεις.

β.Χρησιμοποιώντας την συνάρτηση *imshow()* της *Matlab* παρατηρούμε τα εξής :

Για **2 bits** κβαντισης η εικόνα φαίνεται να έχει πολύ θόρυβο.

Για τα **4 bits** κβάντισης ο θόρυβος είναι πάλι αισθητός στο σήμα εξόδου,βέβαια τα πράγματα είναι πολύ καλύτερα από τα 2 bits..

3.Για την περίπτωση του μη-ομοιόμορφου PCM και για N=2bits, μετρείστε την πιθανότητα εμφάνισης κάθε στάθμης στην έξοδο του κβαντιστή. Θεωρήστε ότι η κατανομή των δειγμάτων ομιλίας μπορεί να προσεγγιστεί από την κανονική κατανομή με μέση τιμή m=-0.04 και διασπορά σ2=0.11. Υπολογίστε σε αυτή τη περίπτωση τις πιθανότητες εμφάνισης κάθε στάθμης και τη μέση παραμόρφωση και συγκρίνετε τις τιμές που προκύπτουν με αυτές που προέκυψαν πειραματικά.

Χρησιμοποιείται το script ask3.*m* (τμήμα Κώδικες).

Αρχικά για να υπολογίσουμε πειραματικά την πιθανότητα εμφάνισης κάθε στάθμης στην έξοδο του κβαντιστή καλέσαμε τον μη-ομοιόμορφο κβαντιστή για N = 2 bits και υπολογίσαμε το ιστόγραμμα του xq διανύσματος και στην συνέχεια το διαιρέσαμε με τον αριθμό των σταθμών για να βρούμε τις ζητούμενες πιθανότητες.

Τα αποτελέσματα :

Περιοχή	Πιθανότητα Εμφάνισης
4	0.03246
3	0.16287
2	0.66213
1	0.14253

Τώρα για να υπολογίσουμε θεωρητικά αυτές τις τιμές λαμβάνουμε υπόψη μας ότι το σήμα εισόδου μας ακολουθεί κανονική κατανομή με μέση τιμή -0.04 και διασπορά 0.11 .

Θα χρησιμοποιήσουμε την συνάρτηση *normcdf* της Matlab επειδή χρησιμοποιείται για τον υπολογισμό της συνάρτησης πυκνότητας πιθανότητας σε μια Gaussian κατανομή με ορίσματα μέση τιμή και την τυπική απόκλιση (σ)που είναι η τετραγωνική ρίζα της διασποράς .

$$p = F(x \mid \mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} e^{\frac{-(t-\mu)^2}{2\sigma^2}}\, dt$$

Για την προτελευταία (τρίτη) περιοχή υπολογίζουμε το όριο στο διάστημα και αφαιρούμε από αυτό την πιθανότητα εμφάνισης της τελευταίας στάθμης για να βρούμε την πιθανότητα εμφάνισης της τρίτης στάθμης.

Συνεχίζουμε αναλόγως για την δεύτερη περιοχή υπολογίζουμε το όριο στο διάστημα και αφαιρούμε τις πιθανότητες εμφάνισης των προηγούμενων δύο περιοχών.

Αντίστοιχα για την πρώτη περιοχή υπολογίζουμε το όριο από και αφαιρούμε όλες τις προηγούμενες πιθανότητες.

Αποτελέσματα :

Περιοχή	Πιθανότητα Εμφάνισης
4	0.21197
3	0.46946
2	0.27789
1	0.04066

4.Υπολογίστε την εντροπία για την έξοδο του κβαντιστή στην περίπτωση του μη-ομοιόμορφου PCM για τα ερωτήματα 1 και 2 .Επιπλέον για την εικόνα υπολογίστε την εντροπία και πριν την κβάντιση.

Χρησιμοποιούνται τα script ask41.*m*, ask42.*m* (τμήμα Κώδικες).

Υπολογίζουμε την Εντροπία πρώτα για το ερώτημα 1.

Εντροπία
1.3814
3.3350
4.5630

Υπολογίζουμε την Εντροπία πρώτα για το ερώτημα 2.

Εντροπία
5.65280
1.95685
3.33500

5.Υλοποιήστε μια συνάρτηση που να επιστρέφει το αλφάβητο του κειμένου στη περίπτωση της πηγής C και να υπολογίζει την πιθανότητα εμφάνισης κάθε συμβόλου του αλφαβήτου, καθώς και την εντροπία της πηγής.

Χρησιμοποιειίται το function ask5() (τμήμα Κώδικες).

Για να βρούμε το αλφάβητο χρησιμοποιούμε τις συναρτήσεις της Matlab, load fearnomore.mat ,A=abs(B) για την φόρτωση του αρχείου και για την μετατροπή του σε μορφή και την συνάρτηση B=unique(str) για να βρούμε το αλφάβητο .

Το αλφάβητο της πηγής
ABCDEFGHIJKLMNOPQRSTUVWXY

Στην συνέχεια για να βρούμε την πιθανότητα εμφάνισης του κάθε συμβόλου χρησιμοποιούμε την συνάρτηση hist(A, max(A) - min(A) + 1) / length(A); της Matlab .

Αποτελέσματα
H pithanotita emfanisis tou sumvolou einai : 0.038462
H pithanotita emfanisis tou sumvolou A einai : 0.038462
H pithanotita emfanisis tou sumvolou B einai : 0.038462
H pithanotita emfanisis tou sumvolou C einai : 0.038462
H pithanotita emfanisis tou sumvolou D einai : 0.038462
H pithanotita emfanisis tou sumvolou E einai : 0.038462
H pithanotita emfanisis tou sumvolou F einai : 0.038462
H pithanotita emfanisis tou sumvolou G einai :

0.038462
H pithanotita emfanisis tou sumvolou H einai : 0.038462
H pithanotita emfanisis tou sumvolou I einai : 0.038462
H pithanotita emfanisis tou sumvolou J einai : 0.038462
H pithanotita emfanisis tou sumvolou K einai : 0.038462
H pithanotita emfanisis tou sumvolou L einai : 0.038462
H pithanotita emfanisis tou sumvolou M einai : 0.038462
H pithanotita emfanisis tou sumvolou N einai : 0.038462
H pithanotita emfanisis tou sumvolou O einai : 0.038462
H pithanotita emfanisis tou sumvolou P einai : 0.038462
H pithanotita emfanisis tou sumvolou Q einai : 0.038462
H pithanotita emfanisis tou sumvolou R einai : 0.038462
H pithanotita emfanisis tou sumvolou S einai : 0.038462
H pithanotita emfanisis tou sumvolou T einai : 0.038462
H pithanotita emfanisis tou sumvolou U einai : 0.038462
H pithanotita emfanisis tou sumvolou V einai : 0.038462
H pithanotita emfanisis tou sumvolou W einai : 0.038462
H pithanotita emfanisis tou sumvolou X einai : 0.038462
H pithanotita emfanisis tou sumvolou Y einai : 0.038462
ABCDEFGHIJKLMNOPQRSTUVWXY

Τέλος βρήκαμε την εντροπία της πηγής:

Εντροπία της πηγής
4.7004

Κώδικες

pcm.m

```
%fortwnoume to hxitiko sima
[y,fs,N]=wavread('speech.wav');
%y to sima hxou pou exoume fortwsei apo to speech.wav
%fs o ruthmos deigmatolipsias
%N ta kvantismena bits



%wavplay(y,fs);

power = mean(y.^2) ;
B=2;
[xq,centers,D] = Lloyd_Max(y,B,min(y),max(y));
% Ypologismos kvantismenou simatos
xq1 = centers(xq) ;
% Ypologismos thoruvou
noise = mean((y-xq1').^2);
% Ypologismos Signal to Noise Ratio
SQNR = power/noise ;
% Ypologismos Signal to Noise Ratio se decibel (dB)
SQNRdb_a_1 = 10*log10(SQNR);
%wavplay(xq1,fs);
ena = centers;





B=4;
[xq,centers,D] = Lloyd_Max(y,B,min(y),max(y));
% Ypologismos kvantismenou simatos
xq1 = centers(xq) ;
% Ypologismos thoruvou
noise = mean((y-xq1').^2);
% Ypologismos Signal to Noise Ratio
SQNR = power/noise ;
% Ypologismos Signal to Noise Ratio se decibel (dB)
SQNRdb_a_2 = 10*log10(SQNR);
%wavplay(xq1,fs);
duo = centers;


B=8;
[xq,centers,D] = Lloyd_Max(y,B,min(y),max(y));
% Ypologismos kvantismenou simatos
xq1 = centers(xq) ;
% Ypologismos thoruvou
noise = mean((y-xq1').^2);
% Ypologismos Signal to Noise Ratio
SQNR = power/noise ;
% Ypologismos Signal to Noise Ratio se decibel (dB)
SQNRdb_a_3 = 10*log10(SQNR);
%wavplay(xq1,fs);

tria= centers;

d = ['SQNRdb ',' unnormalized quantizer ','2 bits  =  ' ,num2str(SQNRdb_a_1)];
disp(d);
d = ['SQNRdb ',' unnormalized quantizer ','4 bits  =  ' ,num2str(SQNRdb_a_2)];
```

```
disp(d);
d = ['SQNRdb ',' unnormalized quantizer ','8 bits  =  ' ,num2str(SQNRdb_a_3)];
disp(d);
```

adm.m

```
N=2;
[y,fs,N]=wavread('speech.wav');
y2=interp(y,N);


[cn]=adm_enc(y2);
[signal]=adm_dec(cn);
figure(1)
subplot(3,1,1)
plot(signal)
title(['Arithmos bits kvantisis = ',num2str(N),' bits'])
ylabel('')
xlabel('')
grid on

N=4;
y2=interp(y,N);
[cn]=adm_enc(y2);
[signal]=adm_dec(cn);

subplot(3,1,2)
plot(signal)
title(['Arithmos bits kvantisis = ',num2str(N),' bits'])
ylabel('')
xlabel('')
grid on



N=8;
y2=interp(y,N);
[cn]=adm_enc(y2);
[signal]=adm_dec(cn);

subplot(3,1,3)
plot(signal)
title(['Arithmos bits kvantisis = ',num2str(N),' bits'])
ylabel('')
xlabel('')
grid on
```

ask2.m

```
N=256;
M=256;

load cameraman.mat
imshow(uint8(i));



x=i(:);
x=(x-128)/128;

power = mean(x.^2) ;

[xq,centers,D]=Lloyd_Max(x,2,min(x),max(x));

a1=centers(xq);
a2=128*a1+128;
y1=reshape(a2,M,N);
imshow(uint8(y1));
```

```
noise = mean((x-a1').^2);
% Ypologismos Signal to Noise Ratio
SQNR = power/noise ;
% Ypologismos Signal to Noise Ratio se decibel (dB)
SQNRdb_a_1 = 10*log10(SQNR);


[xq,centers,D]=Lloyd_Max(x,4,min(x),max(x));

a3=centers(xq);
a4=128*a3+128;
y2=reshape(a4,M,N);
imshow(uint8(y2));

noise = mean((x-a3').^2);
% Ypologismos Signal to Noise Ratio
SQNR = power/noise ;
% Ypologismos Signal to Noise Ratio se decibel (dB)
SQNRdb_a_2 = 10*log10(SQNR);



d = ['SQNRdb ',' unnormalized quantizer ','2 bits  =  ' ,num2str(SQNRdb_a_1)];
disp(d);
d = ['SQNRdb ',' unnormalized quantizer ','4 bits  =  ' ,num2str(SQNRdb_a_2)];
disp(d);
```

ask3.m

```
%fortwnoume to hxitiko sima
[y,fs,N]=wavread('speech.wav');
%y to sima hxou pou exoume fortwsei apo to speech.wav
%fs o ruthmos deigmatolipsias
%N ta kvantismena bits



%wavplay(y,fs);

power = mean(y.^2) ;
B=2;
[xq,centers,D] = Lloyd_Max(y,B,min(y),max(y));

% Ypologismos Delta
Del = (max(y)-min(y))/2^2 ;

% Mesi timi kai apoklisi
m = -0.04;

d = sqrt(0.11);


% Ypologismos oriwn
orio1 = max(y) - Del;
orio2 = max(y) - 2*Del;
orio3 = max(y) - 3*Del;
orio4 = inf;

% Sunarthsh puknotitas pithanotitas
p4 = normcdf(orio3,m,d) ;
p3 = normcdf(orio2,m,d) - p4;
p2 = normcdf(orio1,m,d) -p3-p4;
p1 = normcdf(orio4,m,d) -p2-p3-p4;

% Anathesi timwn se ena dianusma
theoritiki_pithanotita = [p1 p2 p3 p4];

% Ypologismos peiramatikwn timwn
% gia sunarthsh puknotitas pithanotitas
 peiramatiki_pithanotita = hist(xq, max(xq) - min(xq) + 1) / length(xq) ;
```

```
 % dianusma me tis perioxes
s = [1 2 3 4];
```

ask41.m

```
%fortwnoume to hxitiko sima
[y,fs,N]=wavread('speech.wav');
%y to sima hxou pou exoume fortwsei apo to speech.wav
%fs o ruthmos deigmatolipsias
%N ta kvantismena bits



%wavplay(y,fs);

power = mean(y.^2) ;
B=2;
[xq,centers,D] = Lloyd_Max(y,B,min(y),max(y));
% Ypologismos kvantismenou simatos
xq1 = centers(xq) ;
% Ypologismos thoruvou
noise = mean((y-xq1').^2);
% Ypologismos Signal to Noise Ratio
SQNR = power/noise ;
% Ypologismos Signal to Noise Ratio se decibel (dB)
SQNRdb_a_1 = 10*log10(SQNR);
%wavplay(xq1,fs);

p = hist(xq1, 127 - 32 + 1) / length(xq1) ;

  % Ypologismos Entropias
 Entropy1 = 0;
for i = 1:96
   if p(i)~=0
   Entropy1 = Entropy1 + p(i)*log2(1/p(i)) ;
   end
end




B=4;
[xq,centers,D] = Lloyd_Max(y,B,min(y),max(y));
% Ypologismos kvantismenou simatos
xq1 = centers(xq) ;

p = hist(xq1, 127 - 32 + 1) / length(xq1) ;

  % Ypologismos Entropias
 Entropy2 = 0;
for i = 1:96
   if p(i)~=0
   Entropy2 = Entropy2 + p(i)*log2(1/p(i)) ;
   end
end


B=8;
[xq,centers,D] = Lloyd_Max(y,B,min(y),max(y));
% Ypologismos kvantismenou simatos
xq1 = centers(xq) ;



 p = hist(xq1, 127 - 32 + 1) / length(xq1) ;

  % Ypologismos Entropias
 Entropy3 = 0;
for i = 1:96
```

```
    if p(i)~=0
    Entropy3 = Entropy3 + p(i)*log2(1/p(i)) ;
    end
end
```

ask42.m

```
N=256;
M=256;

load cameraman.mat
imshow(uint8(i));



x=i(:);
x=(x-128)/128;

power = mean(x.^2) ;

p = hist(x, 127 - 32 + 1) / length(x) ;

 % Ypologismos Entropias
 Entropy21 = 0;
for i = 1:96
    if p(i)~=0
    Entropy21 = Entropy21 + p(i)*log2(1/p(i)) ;
    end
end

[xq,centers,D]=Lloyd_Max(x,2,min(x),max(x));

a1=centers(xq);
a2=128*a1+128;
y1=reshape(a2,M,N);
imshow(uint8(y1));

p = hist(a2, 127 - 32 + 1) / length(a2) ;

 % Ypologismos Entropias
 Entropy22 = 0;
for i = 1:96
    if p(i)~=0
    Entropy22 = Entropy22 + p(i)*log2(1/p(i)) ;
    end
end


[xq,centers,D]=Lloyd_Max(x,4,min(x),max(x));

a3=centers(xq);
a4=128*a3+128;
y2=reshape(a4,M,N);
imshow(uint8(y2));

p = hist(a3, 127 - 32 + 1) / length(a3) ;

 % Ypologismos Entropias
 Entropy23 = 0;
for i = 1:96
    if p(i)~=0
    Entropy23 = Entropy23 + p(i)*log2(1/p(i)) ;
    end
end
```

ask5()

```
function ask5()
```

```
%fortwnoume to arxeio kai mas epistrefei ena string
load fearnomore.mat

%vriskoume to alfavito tou keimenou m tin entolh unique ths matlab
B=unique(str);

%metatrepoume to string se ASCII morfh gia na vroume thn pithanothta emfanisis
A = abs(B);

% Ypologismos pithanotitwn emfanisis olwn twn sumbolwn xrisimopoioume thn sunarthsh tis matlab hist gia na ftiaxoume ena istogramma
%
p = hist(A, max(A) - min(A) + 1) / length(A);

% Ypologismos Entropias
Entropy = 0;
for i = (1:max(A) - min(A) + 1)
   if p(i)~=0
          Entropy = Entropy + p(i)*log2(1/p(i)) ;
   end
end

 % Vriskoume thn pithanotita emfanisis mono gia to alfavito tou keimenou
 C=p(p~=0);

%ektupwnoume ta apotelesmata
d = ['To alfavito tou keimenou tis phhs C einai :' ,num2str(A)];
disp(d);
d = ['=========================================================='];
disp(d);

d = ['Kai se morfh ASCII' ,num2str(B)];
disp(d);
d = ['=========================================================='];
disp(d);

d = ['H entropia tis phghs     =  ' ,num2str(Entropy)];
disp(d);
d = ['=========================================================='];
disp(d);

for i = (1:length(C))
d = ['H pithanotita emfanisis tou sumvolou ' num2str(B(i)) ' einai : '    num2str(C(i))];
disp(d);

d = ['=========================================================='];
disp(d);

end
```

lloyd_max.m

```
% PCM me mi omoimorfo kvantisti
function [xq,centers,D] = Lloyd_Max(x,N,min_value,max_value)
%x to sima eisodou upo morfh dianismanos
%N o arithmos twn bits pou tha xrisimopoihthoun
%max_value h megisti apodekti timi tou simatos eisodou
%min_value h elaxisti apodekti timi tou simatos eisodou



%xq to kwdikopoihmeno dianusma exodou meta apo Kmax epanalipseis
%centers ta kentra twn perioxwn kvantisis meta apo Kmax epanalipseis
%D dianusma pou periexei tis times [D1:DkmAX] opou Di antstoixei sthn mesh paramorgwsh sthn i-osth epanalhpsh tou algorithmou
```

```
% mhkos tou dianusmatos x
l = length(x) ;

count = 1;

% Arxikopoihsh  twn oriwn kvantismou me ta kentra tou omoiomorfou kvantisti
[xq_tmp,centers_tmp] = my_quantizer(x,N,min(x),max_value) ;

% Ypologismos elaxistis timis simatos eisodou
%upologismos megistis timis simatos eisodou
min_value = min(x);


% Arxikopoihsh dianusmatwn
 s = zeros(1,2^N) ;
 n=zeros(1,2^N) ;
 upper = zeros(1,2^N) ;
 lower = zeros(1,2^N) ;

 % Ypologismos isxus tous simatos eisodou x
 power = mean(x.^2);

 % Megisto kai elaxisto orio tou simatos eisodou

for i = 1:l
  if x(i)> max_value
      x(i) = max_value ;
  elseif x(i) < min_value
     x(i) =min_value ;
  end
end

 loop = 0;
 while loop~=1

for k = 1: 2^N
    %elegxoume se poia perioxi tou sima eimaste wste na vroume tin megisti kai elaxisti timi tou simatos
    %An eimaste stin prwth tote to anw orio einai to max_value tou simatos
     if k == 1
        upper(k) = max_value ;
        lower(k) = (centers_tmp(k) + centers_tmp(k+1) )/2 ;
    %An eimaste stin teleutaia tote to katw orio einai to min_value tou simatos
     elseif k==2^N
         upper(k) = (centers_tmp(k-1) + centers_tmp(k) )/2 ;
         lower(k) = min_value ;
    %Stis upoloipes periptwseis vriskoume to orio upologizontas me thn vohtheia twn duo geitonikwn perioxwn
     else
         upper(k) = (centers_tmp(k-1) + centers_tmp(k) )/2 ;
         lower(k) = (centers_tmp(k) + centers_tmp(k+1) )/2 ;
     end


 %% O algorithmos Lloyd Max
    for i = 1:l

       if x(i) >= lower(k) &&  x(i) <=upper(k)
%upologizoume tin timi tou simatos exodou upologizontas to kvantismeno sima eisodou


             xq_tmp(i) = k ;
 % Voithitika dianusmata gia ton upologismo tou
 % mesou orou twn timwn eisodou se kathe perioxi
            s(k) = s(k) + x(i) ;
            n(k) = n(k)+1 ;
       end

    end

 % Ean uparxei toulaxiston mia timi mesa stin perioxi
 % ypologizoume to neo kentroeides ws ton meso oro
```

```
 % aftwn twn timwn

    if n(k) ~= 0
        centers_tmp(k) = s(k)/n(k) ;
    else
 % alliws to kentroeides paramenei opws itan
        centers_tmp(k) = (upper(k)+lower(k)) /2 ;

    end

end
% Ypologismos kvantismenou simatos
 x_new = centers_tmp(xq_tmp);

 % Ypologismos paramorfwsis gia kathe epanalipsi tou algorithmou
 Dis(count)= mean((x-x_new').^2) ;
 % Ypologismos thoruvou tou kvantismenou simatos
 noise = mean((x-x_new').^2);

 % Ypologismos SQNR gia kathe epanalipsi tou algorithmou
 SQNR(count) = power/noise;
  % Ypologismos SQNR se db
 SQNRdb(count) = 10*log10(SQNR(count));

 % Ean den vriskomaste stin prwti epanalipsi
 % kai h diafora tis paramorfwshs tou neou kvantismenou simatos
 % apo to to kvantismeno sima tis proigoumenis epanalipsis
 % einai mikroteri tou katwfliou 10^-7 termatizei o algorithmos
 if count ~= 1 && Dis(count-1)-Dis(count) <10^-7
     loop =1;
 end

  count = count +1 ;
 end

% Ta dianusmata eksodou pairnoun tis times twn dianusmatwn stin
% teleftaia epanalipsi tou algorithmou
xq = xq_tmp ;
centers = centers_tmp ;

% Ypologismos paramorfwsis stin teleftaia epanalipsi
D = Dis(length(Dis)) ;
F=SQNRdb;
% Grafikh parastash SQNR (db) gia oles tis epanalipseis
figure(N)
plot(SQNRdb,'-b','LineWidth',2);
title(['Arithmos bits kvantisis = ',num2str(N),' bits'])
ylabel('SQNR_d_b')
xlabel('epanalipseis')
grid on

xq1=centers(xq);
figure(N+1)
plot(xq1,'-b','LineWidth',2);
title(['Arithmos bits kvantisis = ',num2str(N),' bits'])
ylabel('SQNR_d_b')
xlabel('epanalipseis')
grid on


end
```

adm_dec.m

```
function [ signal ] = adm_dec( cn )
delta_min = 0.02;
L = length(cn);
k=1.5;
signal = zeros(1,L);  % initializations
delta=ones(1,L);
```

```
for i=1:length(cn)
    if cn(i)==1
        Mq(i)=delta_min;
    else
        Mq(i)=-1.*delta_min;
    end
end
for n=2:L

    if Mq(n)==Mq(n-1)
        delta(n)=k.*delta(n-1);
    else
        delta(n)=delta_min;
    end
 signal(n) = signal(n-1) + delta(n).*Mq(n);
end
end
```

adm_enc.m

```
    function [ cn ] = adm_enc( signal )
delta_min = 0.02; % minimum step size
k=1.5;
m = 4;
% scaling constant
L = length(signal);
Mq = zeros(1,L);
dq = zeros(1,L);
cn = zeros(1,L);
delta=ones(1,L);
for n=2:L

    d = signal(n) - Mq(n-1);
    if d>0
        dq(n)=delta_min;
    else
        dq(n)=-1.*delta_min;
    end

    if dq(n)==dq(n-1) && dq(n-1) == dq(n-2)
        delta(n)=k.*delta(n-1);
    else
        delta(n)=delta_min;
    end
     Mq(n) = Mq(n-1) + delta(n).*dq(n);
end
for i=1:length(cn)
    if dq(i)==delta_min
        cn(i)=1;
    else
        cn(i)=0;
    end
end


end
```