ΜΙΡΑ ΙΣΛΑΜΑΪ
ΑΜ 1070736

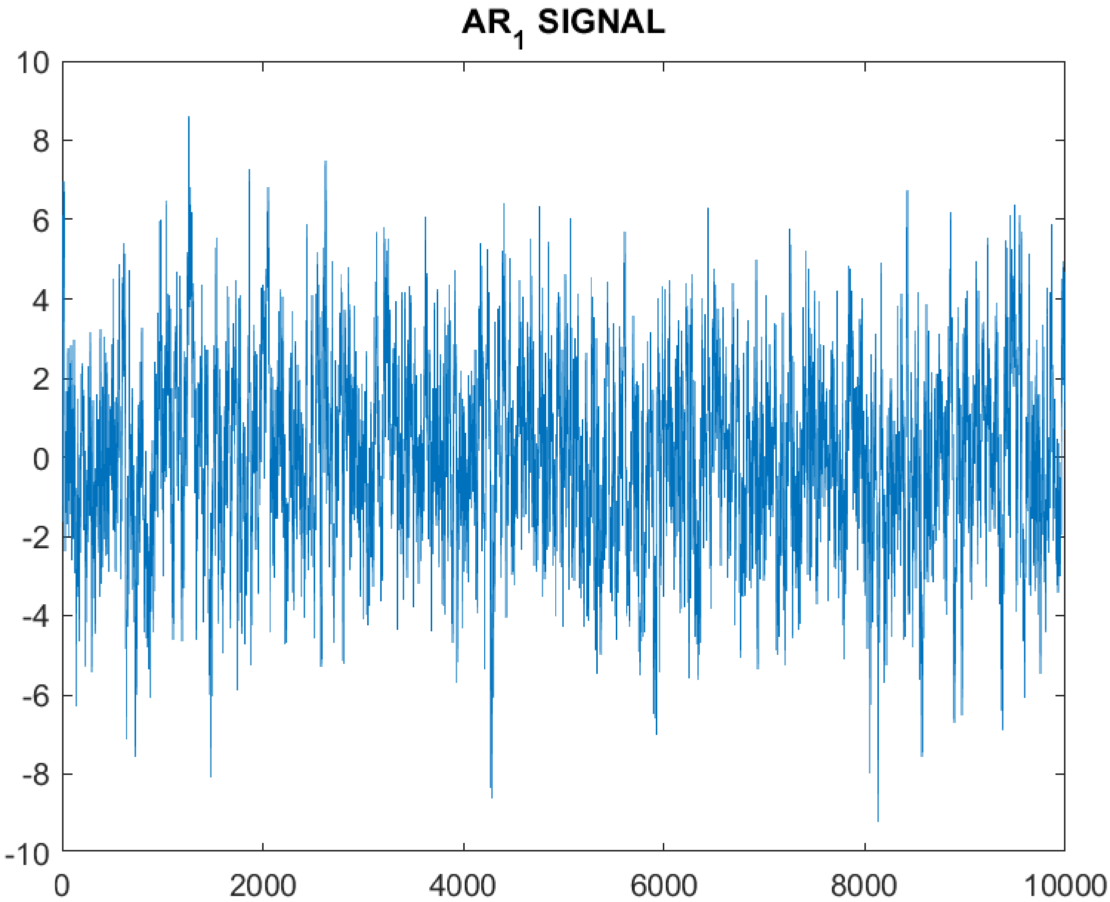


# 1ο ΣΕΤ ΑΣΚΗΣΕΩΝ

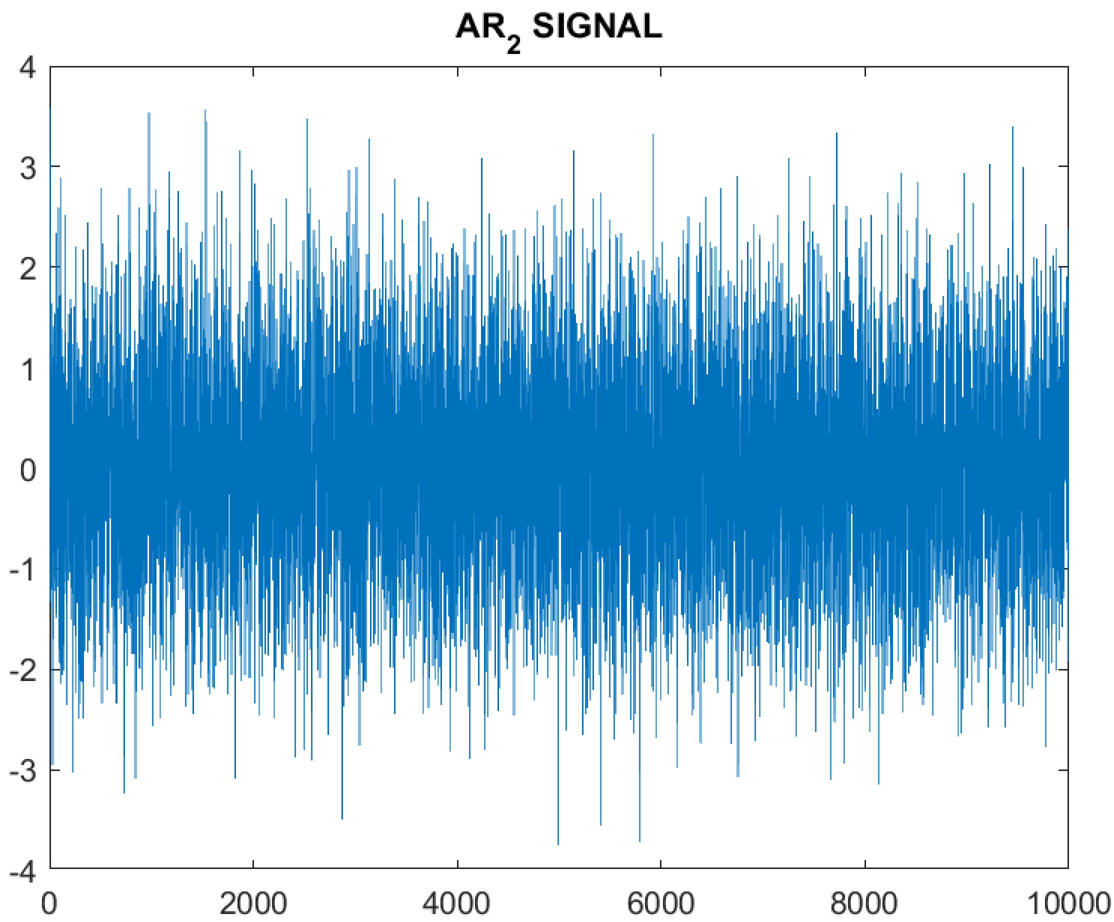
## ΜΕΡΟΣ 1

### ΕΡΩΤΗΜΑ 1ο

Για την επίλυση του ερωτήματος αυτού τρέχουμε τον κώδικα που υπάρχει στο αρχείο source_A.m. Αυτό το αρχείο δημιουργεί τα σήματα AR1 και AR2 και στην συνέχεια καλεί της συναρτήσεις για την κωδικοποίηση τους και διάφορες άλλες συναρτήσεις για την επίλυση κάποιων υπό-ερωτημάτων.




*Εικόνα 1-Αρχικό σήμα AR1*




*Εικόνα 2-Αρχικό σήμα AR2*

Κωδικοποιώντας τα σήματα αυτά με τον Lloyd Max αλγόριθμο για N=2,4 και 8 bits και με το ADM παίρνουμε 4 διαφορετικά κωδικοποιημένα σήματα για το κάθε αρχικό σήμα τα οποία θα τα συγκρίνουμε αρχικά με το SQNR

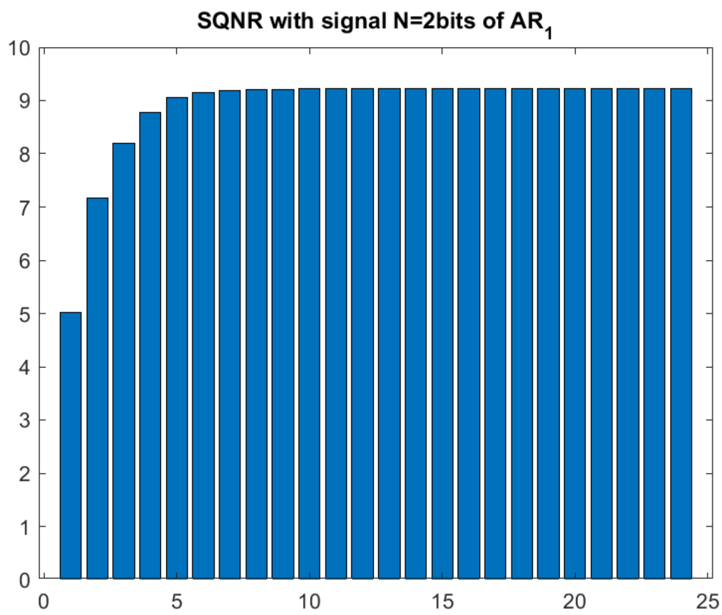
## 1.1.a)

Για τον υπολογισμό του SQNR , έχει οριστεί η συνάρτηση με το όνομα αρχείου SQNR.m όπως βλέπουμε και στην παρακάτω εικόνα.

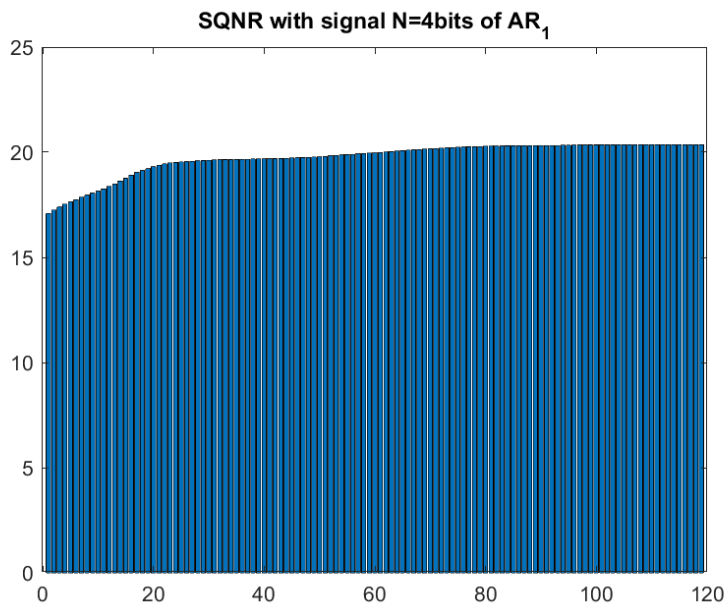
```
function sqnr = SQNR(x,xq)

%Parameters
%x: signal
%xq: quantized signal of x

Sound=mean(x.^2);
noise= abs(x-xq);
Noise=mean(noise.^2);

sqnr=10*log10(Sound/Noise);
%in decibels

fprintf('The SQNR is equal to : %.4fdB\n',sqnr);

end
```

*Εικόνα 3-Κώδικας για το SQNR*

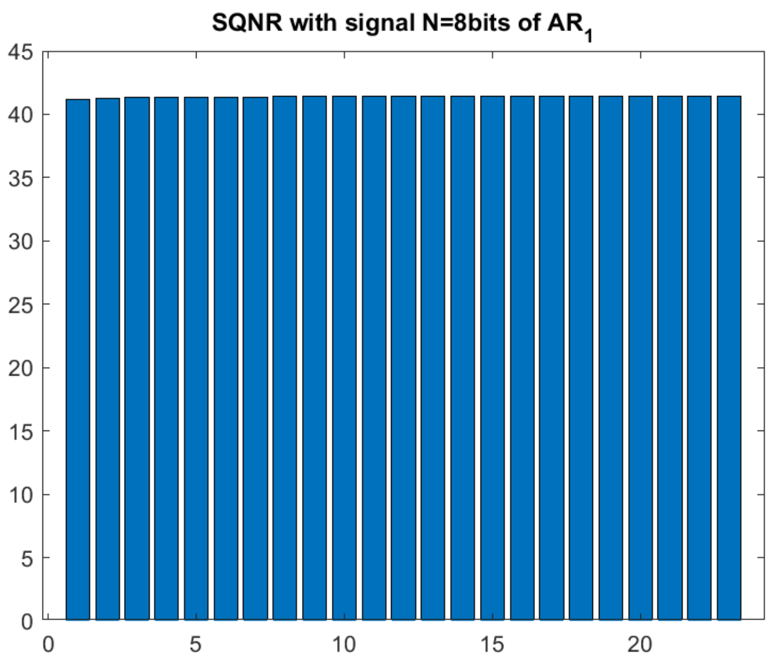
Τις τιμές του SQNR για κάθε περίπτωση τις παίρνουμε μόλις εκτελέσουμε το αρχείο source_A.m και εμφανίζονται στο Command Window αλλά και σε figures (ενσωματώνοντας την εντολή bar() της MATLAB στον κώδικα) που θα παραθέσουμε και παρακάτω. Εκτυπώθηκαν οι τιμές για να έχουμε μεγαλύτερη λεπτομέρεια, γιατί από τα γραφήματα πολλές φορές δεν φαίνεται ξεκάθαρα η διαφορά στην τιμή του SQNR. Παίρνουμε τα παρακάτω αποτελέσματα ,λοιπόν, που φαίνονται στην επόμενη σελίδα.




*Εικόνα 4-Τελικό SQNR 9.2199dB*

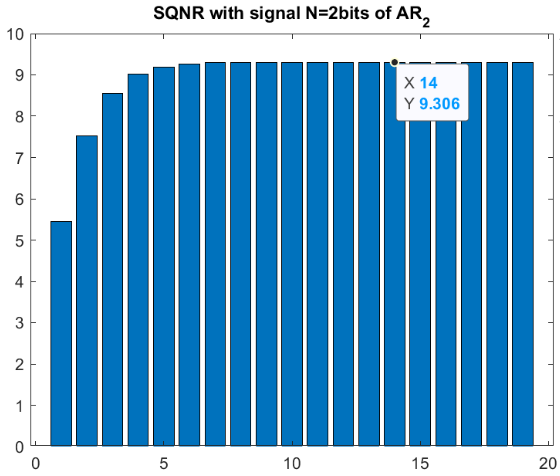



*Εικόνα 5-Τελικό SQNR 20.3632dB*

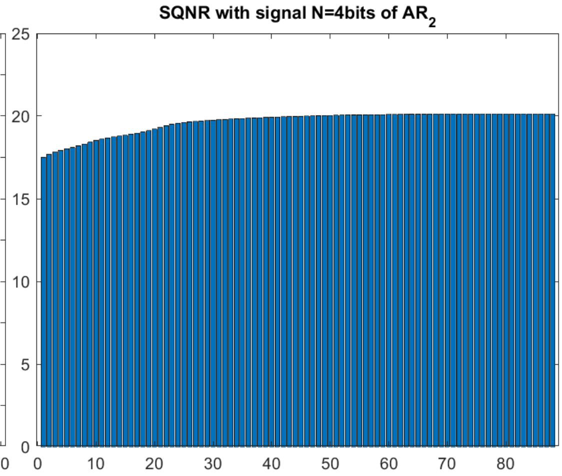



```
This is the AR1 signal quantized with 8 bits
The SQNR is equal to : 41.1410dB
The SQNR is equal to : 41.2712dB
The SQNR is equal to : 41.3082dB
The SQNR is equal to : 41.3360dB
The SQNR is equal to : 41.3588dB
The SQNR is equal to : 41.3723dB
The SQNR is equal to : 41.3824dB
The SQNR is equal to : 41.3917dB
The SQNR is equal to : 41.4002dB
The SQNR is equal to : 41.4078dB
The SQNR is equal to : 41.4128dB
The SQNR is equal to : 41.4175dB
The SQNR is equal to : 41.4217dB
The SQNR is equal to : 41.4270dB
The SQNR is equal to : 41.4298dB
The SQNR is equal to : 41.4330dB
The SQNR is equal to : 41.4353dB
The SQNR is equal to : 41.4373dB
The SQNR is equal to : 41.4381dB
The SQNR is equal to : 41.4384dB
The SQNR is equal to : 41.4388dB
The SQNR is equal to : 41.4391dB
The SQNR is equal to : 41.4391dB
```

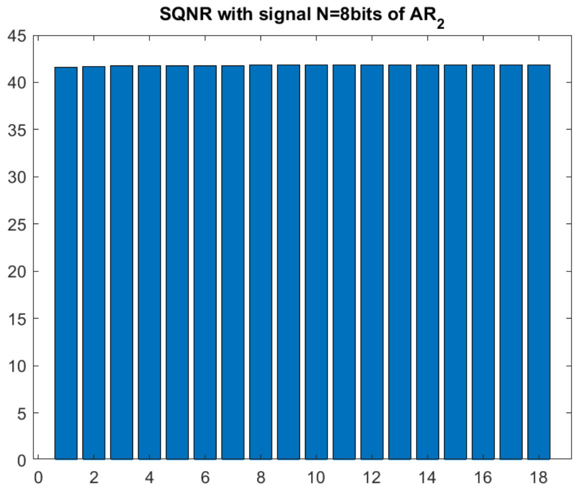
*Εικόνα 6 – Τελικό αποτέλεσμα SQNR =41.4391dB όπως φαίνεται και δίπλα λεπτομερώς*




Εικόνα 7- Τελικό SQNR 9.3057dB




Εικόνα 8- Τελικό SQNR 20.1383dB




```
This is the AR2 signal quantized with 8 bits
The SQNR is equal to : 41.5591dB
The SQNR is equal to : 41.6921dB
The SQNR is equal to : 41.7309dB
The SQNR is equal to : 41.7514dB
The SQNR is equal to : 41.7670dB
The SQNR is equal to : 41.7810dB
The SQNR is equal to : 41.7969dB
The SQNR is equal to : 41.8075dB
The SQNR is equal to : 41.8154dB
The SQNR is equal to : 41.8220dB
The SQNR is equal to : 41.8268dB
The SQNR is equal to : 41.8293dB
The SQNR is equal to : 41.8317dB
The SQNR is equal to : 41.8333dB
The SQNR is equal to : 41.8354dB
The SQNR is equal to : 41.8368dB
The SQNR is equal to : 41.8377dB
The SQNR is equal to : 41.8380dB
```

Εικόνα 9 – Τελικό SQNR 41.8380dB

Σύμφωνα με τα αποτελέσματα που παίρνουμε σε καθένα από τα δύο σήματα σε κάθε επανάληψη του Lloyd Max , το SQNR μεγαλώνει (όπως φαίνεται και σε όλες τις παραπάνω εικόνες). Στις εικόνες 6 και 9, η τιμή του SQNR όλο και αυξάνεται, αλλά είναι πολύ μικρή διαφορά κάθε φορά και κατά συνέπεια δεν φαίνεται στο γράφημα, γιαυτό τον λόγο παραθέτουμε και δίπλα τις τιμές που εκτυπώνονται στο command window. Θεωρητικά είναι επίσης λογικό να αυξάνεται η τιμή του SQNR σε κάθε επανάληψη, αφού σε κάθε επανάληψη μειώνεται και η παραμόρφωση , δηλαδή ο θόρυβος όπου είναι αντιστρόφως ανάλογο του SQNR , αφού $SQNR = \frac{E[\sigma\eta\mu\alpha^2]}{E[\theta\acute{o}\rho\upsilon\beta o\varsigma^2]}$ .

Ακόμα ,άμα συγκρίνουμε τις τιμές του SQNR στην τελευταία επανάληψη του αλγορίθμου , θα παρατηρήσουμε ότι για μεγαλύτερο N , επίπεδο κβάντισης θα έχουμε και μεγαλύτερο SQNR , άρα και μικρότερο σφάλμα κβαντισμού.

Επιπλέον με την κωδικοποίηση των αρχικών σημάτων με το σχήμα ADM έχουμε τα παρακάτω αποτελέσματα.

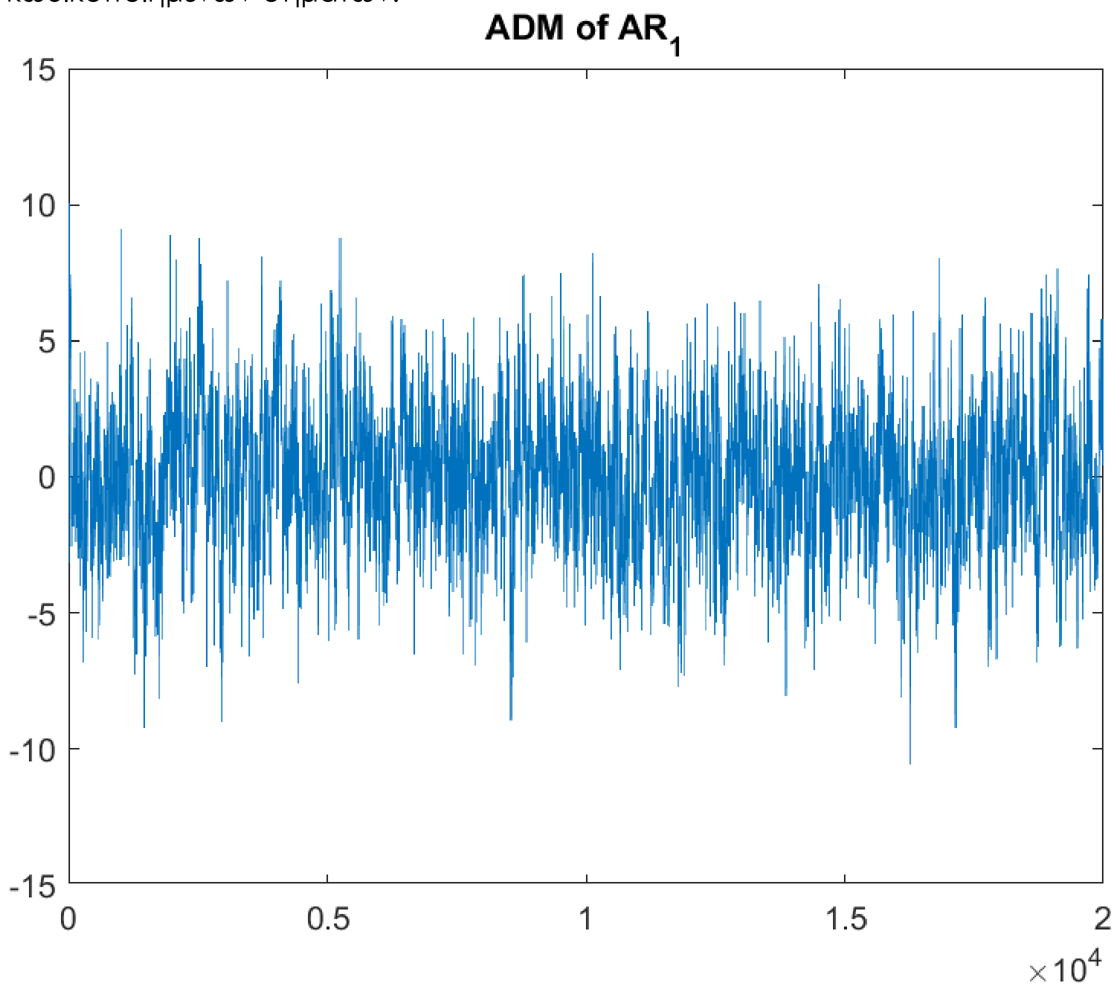
This is the AR1 signal quantized with ADM
The SQNR is equal to : 9.6270dB

This is the AR2 signal quantized with ADM
The SQNR is equal to : 0.1504dB

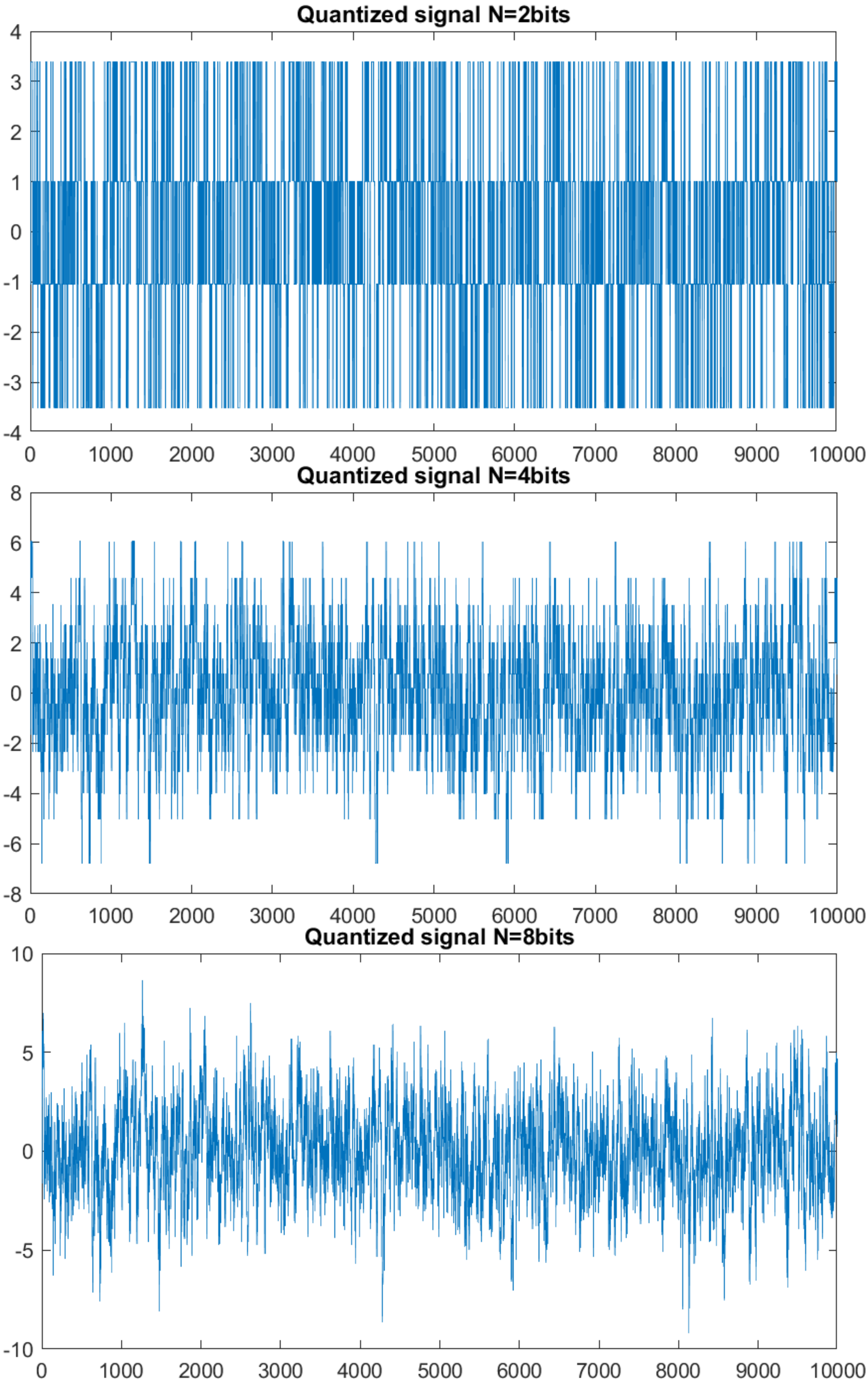
Παρατηρούμε ,λοιπόν, ότι το σχήμα ADM δεν δημιουργεί λιγότερο θόρυβο από το σχήμα PCM, απεναντίας μάλιστα έχει μεγαλύτερο σφάλμα κβαντισμού σε όλες τις περιπτώσεις εκτός από το N=2bits για το σήμα AR1 όπου το SQNR εδώ είναι ελάχιστα μεγαλύτερο. Μάλιστα παρατηρούμε ότι στην περίπτωση του σήματος AR2 το SQNR είναι πολύ κοντά στο μηδέν , δηλαδή έχει μεγάλο σφάλμα κβάντισης. Το σφάλμα αυτό θα μπορούσε να μειωθεί αρκετά με ένα lowpass filter στην έξοδο του αποκωδικοποιητή.

## 1.1.β)

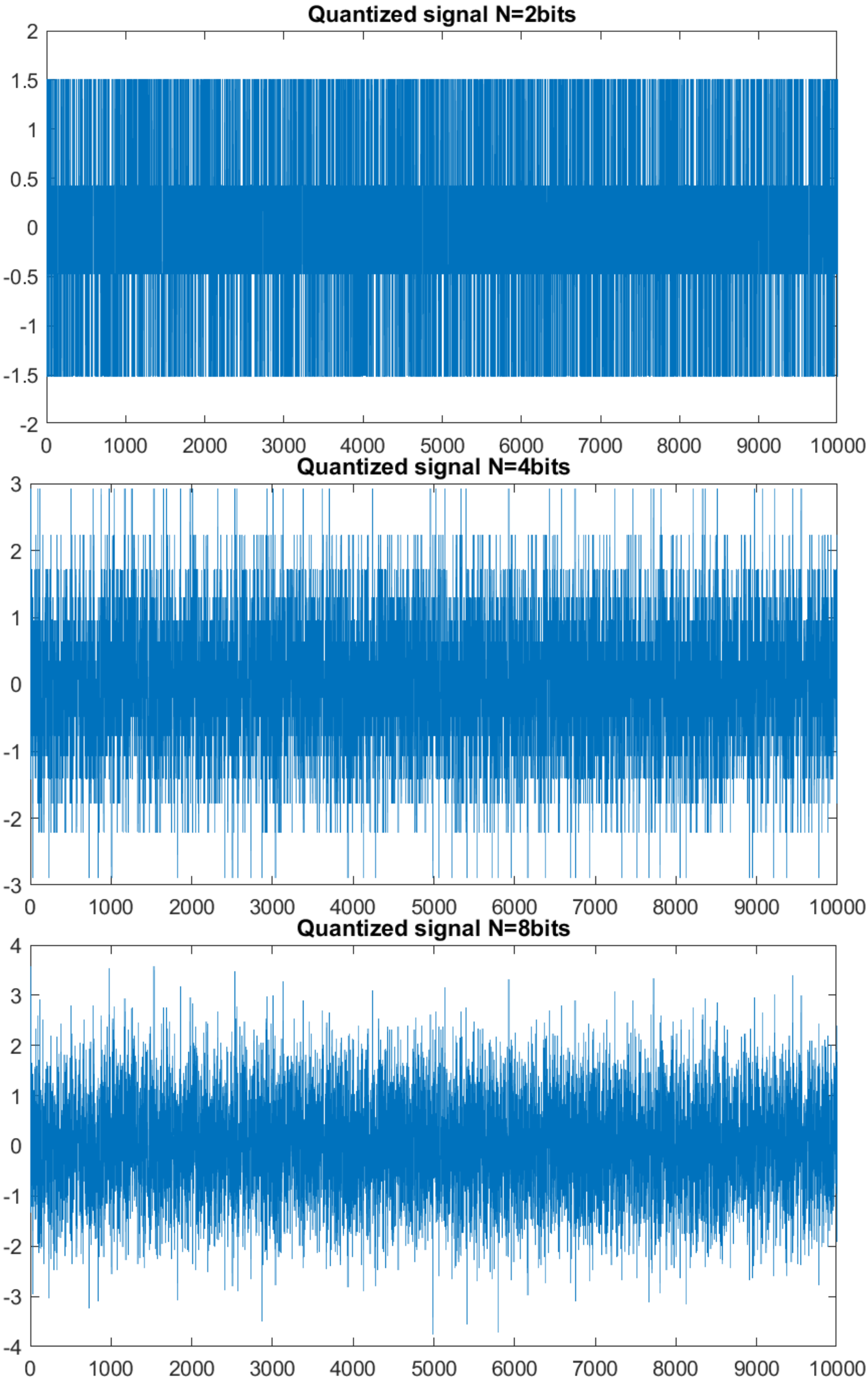
Στην συνέχεια θα συγκρίνουμε τα δύο σχήματα ADM και PCM με την έξοδο των κυματομορφών που μας δίνουν. Παρακάτω βλέπουμε τις κυματομορφές των κωδικοποιημένων σημάτων.



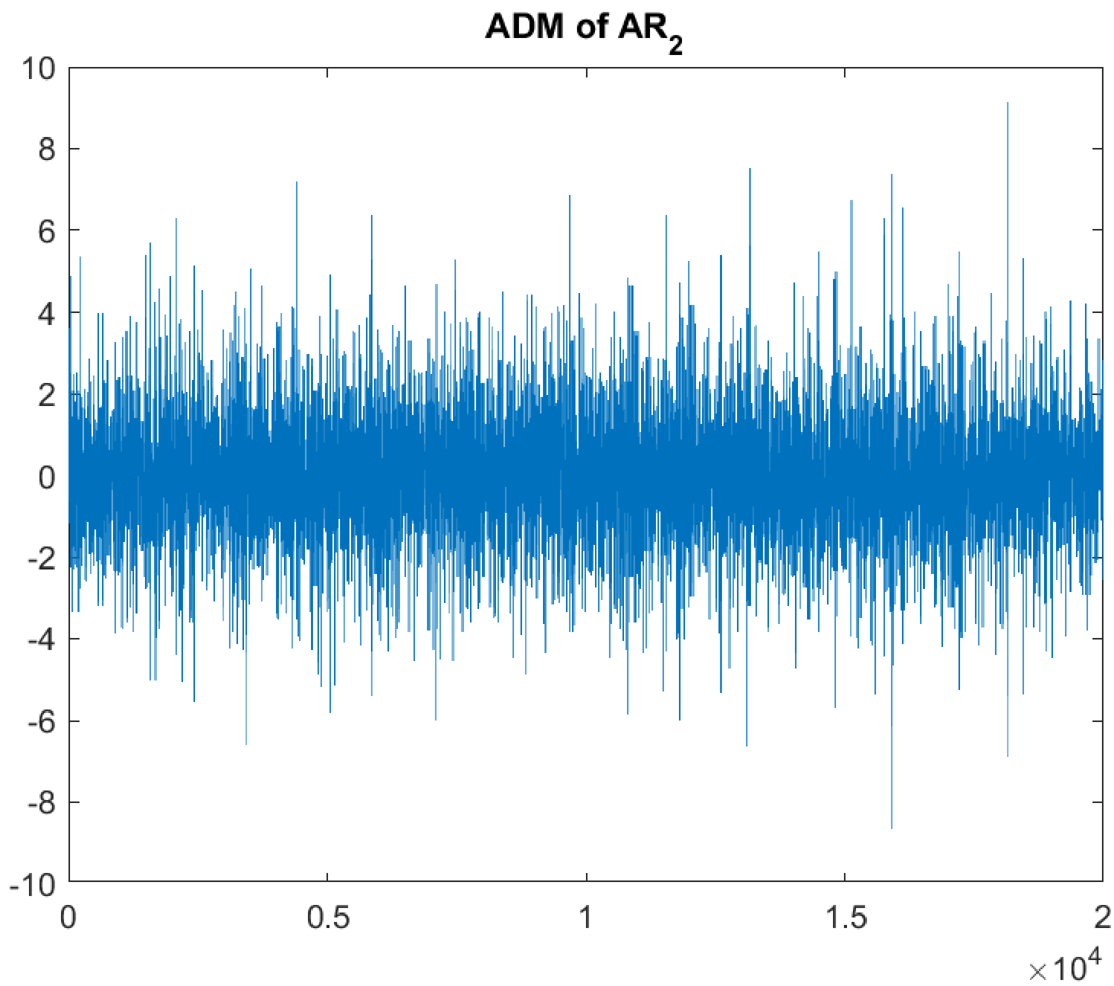

*Εικόνα 10-Κβαντισμένο σήμα AR1 με το ADM*




*Εικόνα 11 -Κβαντισμένα σήματα του αρχικού AR1 με Lloyd Max με N=2,4,8 bits*




*Εικόνα 12- Κβαντισμένα σήματα του αρχικού AR2 με Lloyd Max με N=2,4,8 bits*




*Εικόνα 13- Κωδικοποιημένο σήμα απο ADM*

Από της εικόνες παρατηρούμε ότι οι κυματομορφές που πλησιάζουν παραπάνω στο αρχικό είναι αυτές με την χρήση του σχήματος pcm για N=8bits. Το ίδιο σχήμα για N=4bits και το ADM είναι επίσης αρκετά κοντά στο αρχικό σήμα με μία πρώτη ματιά (αν και το ADM έχει μεγαλύτερο σφάλμα κβαντισμού).

Συγκρίνοντας αυτές τις τιμές της μέτρησης SQNR αλλά και τις κυματομορφές εξόδου συμπεραίνουμε πως η χρήση του ανομοιόμορφου PCM για μεγάλο N έχει τα πιο επιθυμητά αποτελέσματα.

## 1.2

Για την περίπτωση του PCM υπολογίσαμε και την εντροπία της πηγής με τον τρόπο που φαίνεται στις γραμμές 100-109 του κώδικα LloydMax.m (εικόνα 22). Συγκεκριμένα, μετρήσαμε την πιθανότητα να εμφανιστεί η τιμή κάποιου από τα κέντρα κβάντισης του σήματος , έχοντας έναν μετρητή για κάθε ζώνη και διαιρώντας αυτόν με το μήκος του σήματος.

Από την θεωρία γνωρίζουμε ότι η εντροπία μιας διακριτής τυχαίας μεταβλητής X με K πιθανές τιμές ικανοποιεί τα πάνω και κάτω όρια της παρακάτω εξίσωσης:

$$0 \leq H(X) \leq \log K$$

*Άρα για κβάντιση με N bits θα έχουμε 2^N πιθανές τιμές του κωδικοποιημένου σήματος οπότε :* $0 \leq H(X) \leq N$

Τα αποτελέσματα μας για την εντροπία είναι :

| H(X) → | ΣΗΜΑ AR1 | ΣΗΜΑ AR2 |
|---|---|---|
| **N=2bits** $0 \leq H(X) \leq 2$ | 1.9156 | 1.9047 |
| **N=4bits** $0 \leq H(X) \leq 4$ | 3.7556 | 3.6834 |
| **N=8bits** $0 \leq H(X) \leq 8$ | 7.0698 | 7.1422 |

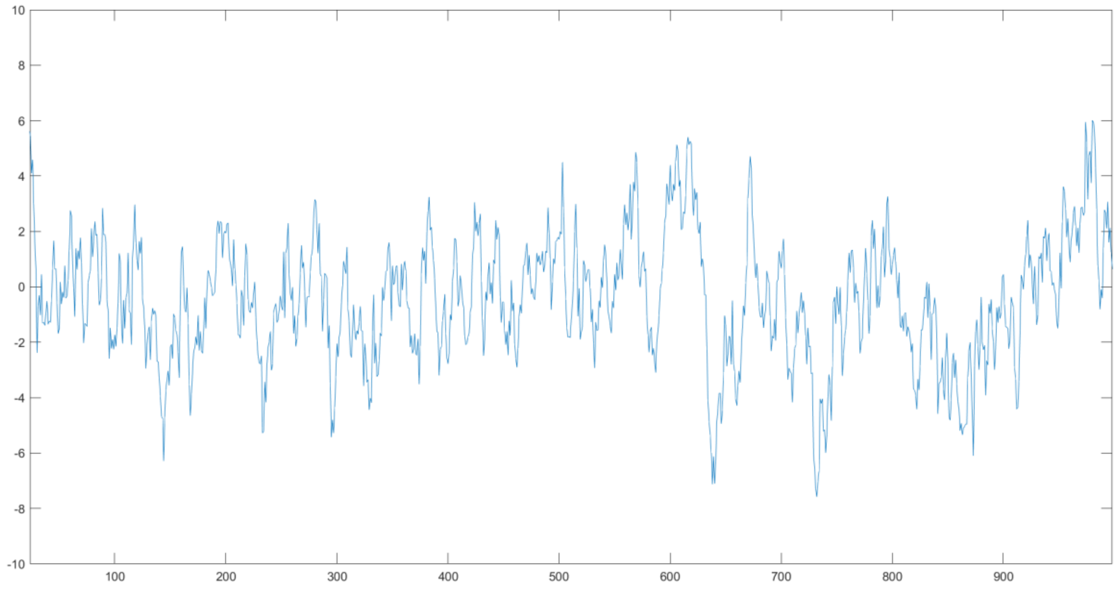
Από τις τιμές που λαμβάνουμε και φαίνονται στον παραπάνω πίνακα μπορούμε να συμπεράνουμε διάφορα πράγματα. Παρατηρούμε ότι σε όλες τις περιπτώσεις η τιμή της μέτρησης φτάνει πολύ κοντά στην μέγιστη. Η εντροπία μεγιστοποιείται όταν η έξοδος είναι ομοιόμορφα κατανεμημένη και είναι το λιγότερο προβλέψιμη. Οι τιμές μας δεν φτάνουν ποτέ ακριβώς την μέγιστη τιμή, αλλά είναι αρκετά κοντά οπότε μπορούμε να συμπεράνουμε ότι τα σήματά μας δεν είναι προβλέψιμα και έχουν μία κατανομή αρκετά κοντά στην ομοιόμορφη.

Επιπλέον παρατηρούμε ότι καθώς αυξάνουμε το N , μεγαλώνει η διαφορά του H(X) με την μέγιστη τιμή που θα μπορούσε να πάρει. Για παράδειγμα στο σήμα AR1 για N=2 bits έχουμε 2-H(X) = 0.0844 ≈ 0.1 και για N=8 bits θα έχουμε 8-H(X) = 0.9302 ≈ 1. Αυτό υποθέτουμε ότι συμβαίνει διότι με λιγότερα bits/sample που θα έχει το κβαντισμένο σήμα δίνει την εντύπωση ότι είναι πιο ομοιόμορφο.

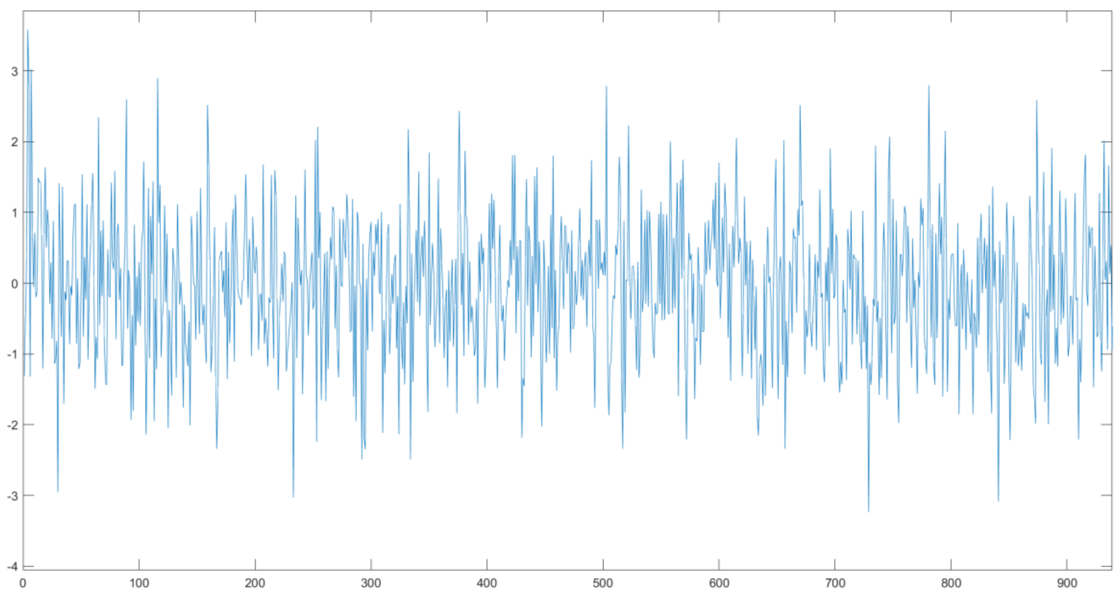
## 1.3

Σύμφωνα με τα αποτελέσματα τον παραπάνω μετρήσεων μας μπορούμε να συγκρίνουμε τις διαδικασίες AR1 και AR2.

Δεν έχουν πολύ διαφορετική συμπεριφορά τα σήματα αυτά, αλλά αυτό που παρατηρούμε είναι ότι στο AR1 η χρήση του ADM έχει πολύ καλύτερα αποτελέσματα από αυτά για το AR2. Αυτό μάλλον επειδή το AR2 είναι πιο πυκνό με πιο απότομες κλίσης όπως βλέπουμε και στις παρακάτω εικόνες.



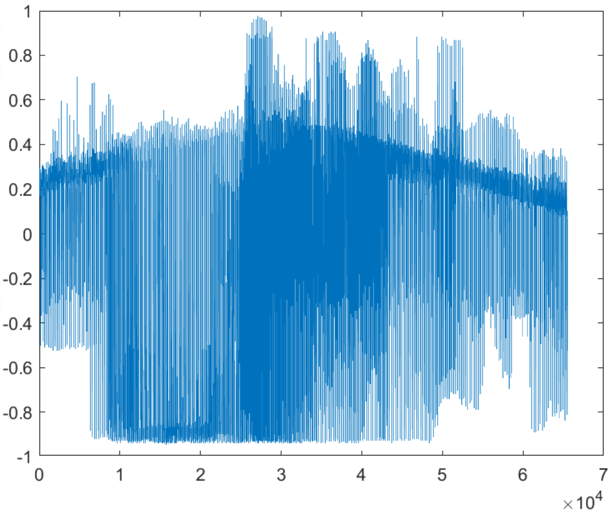

*Εικόνα 14 - Τα πρώτα 1000 δείγματα AR1*




*Εικόνα 15- Τα πρώτα 1000 δείγματα AR2*

## ΕΡΩΤΗΜΑ 2ο

Σε αυτό το σημείο κωδικοποιούνται τα σήματα της πηγής B με το σχήμα του PCM με N=2 και 4bits. Σύμφωνα με την εκφώνηση η πηγή B είναι το διάνυσμα που προκύπτει από τον πίνακα που περιέχει το σήμα της εικόνας cameraman.mat αν αντιστοιχίσουμε τις τιμές που ανήκουν στο [0:255] στο δυναμικό εύρος [-1:1]. Το ερώτημα 2 εκτελείται από τον κώδικα με όνομα source_B.m .

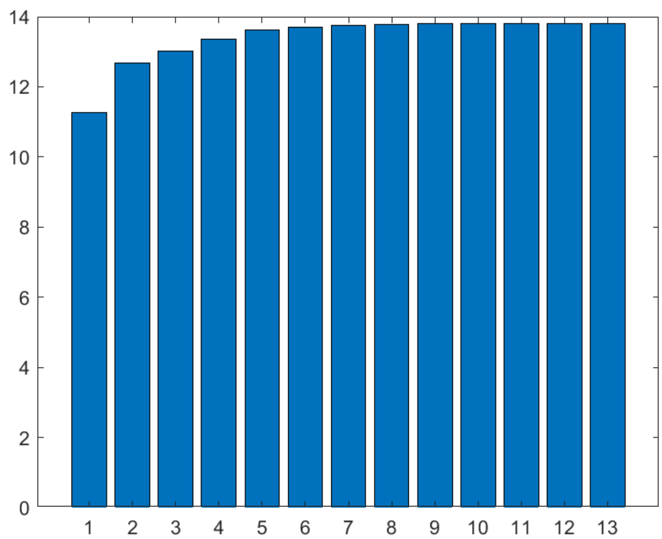




*Εικόνα 16-Αρχικό σήμα εικόνας και δίπλα η πηγή B με δυναμικό εύρος [-1:1]*

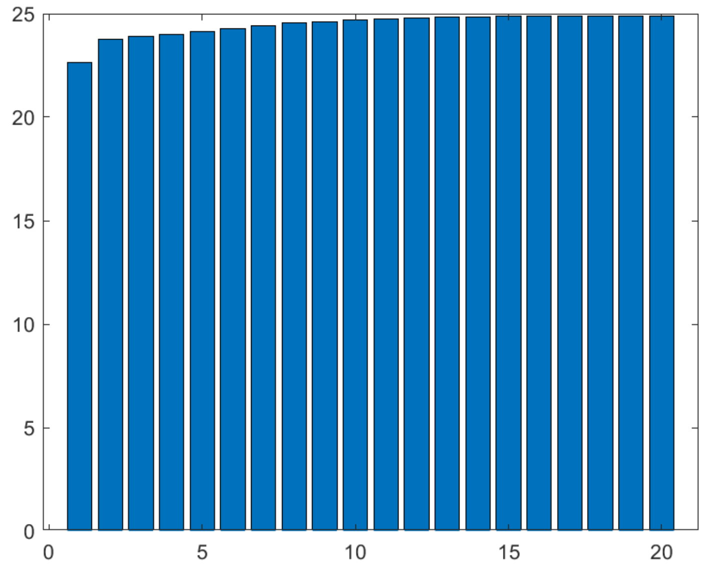
Θα συγκρίνουμε τώρα τα κωδικοποιημένα σήματα για κάθε περίπτωση.

### 2.1.a

Πρώτα βλέπουμε τις μετρήσεις μας για το SQNR, που εκτυπώνονται στο command window , αλλά και σε σχεδιαγράμματα.




*Εικόνα 17- SQNR σε κάθε επανάληψη για N=2bits*

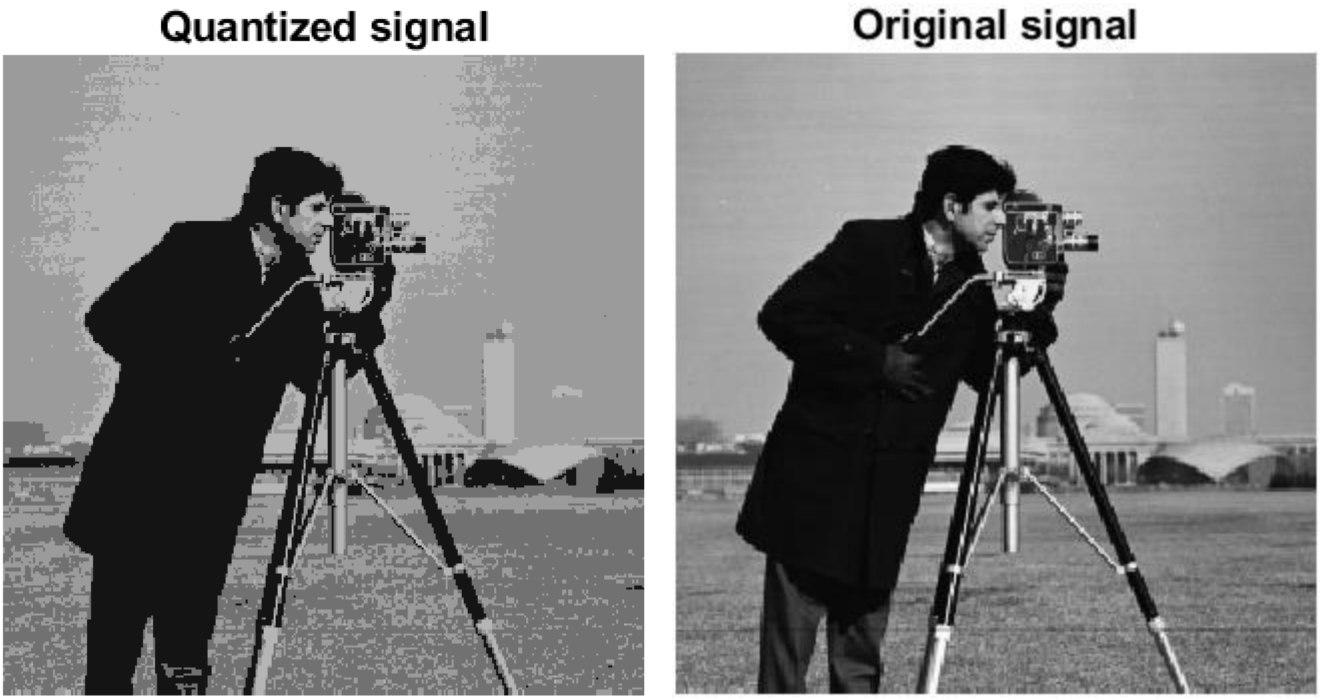



*Εικόνα 18-SQNR σε κάθε επανάληψη για N=4bits*

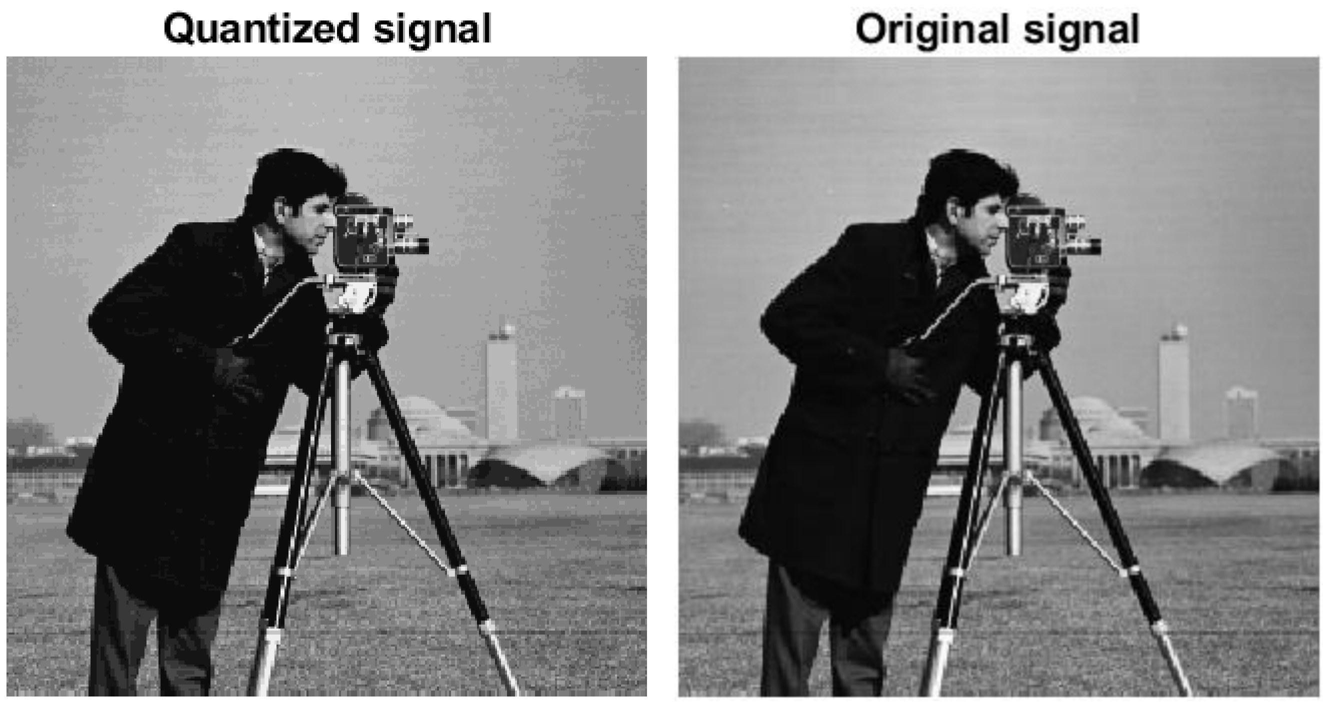
Για πιο ακριβείς τιμές , η τελική τιμή του SQNR για την κωδικοποίηση με N=2 bits θα είναι 13.8065dB και την κωδικοποίηση με 4 bits είναι 24.8949dB. Προφανώς για μεγαλύτερο N, έχουμε και καλύτερη τιμή για το SQNR (δηλαδή μεγαλύτερη τιμή) αφού μπορούμε να προσεγγίσουμε καλύτερα το αρχικό σήμα με τις περισσότερες ζώνες κβάντισης. Γενικότερα ισχύει για το SQNR ότι εξηγήθηκε προηγουμένως στο ερώτημα 1.1.α

## 2.1.β

Παρακάτω βλέπουμε το οπτικό αποτέλεσμα που είχε η κβάντιση , δηλαδή την εικόνα μετά την επεξεργασία του σήματος του.




*Εικόνα 20- Εικόνα με κβάντιση N=2bits και δεξιά η αρχική εικόνα*




*Εικόνα 21-Αριστερά εικόνα με κβάντιση N=4bits και δεξιά αρχική εικόνα*

Όπως φαίνεται και στις εικόνες 20-21, και οι δύο επεξεργασμένες εικόνες δεν είναι το ίδιο «καθαρές» με την αρχική, δεν έχουν την ίδια ομαλότητα, ομοιογένεια των τόνων του μαύρου και άσπρου . Μάλιστα για N=2bits υπάρχει μία πολύ έντονη αντίθεση των χρωμάτων, φαίνεται περισσότερο αλλοιωμένη. Στην εικόνα 21 δεν είναι τόσο αντιληπτή η διαφορά. Βέβαια και στις δύο περιπτώσεις μπορούμε να καταλάβουμε τι απεικονίζεται , απλά δεν έχουμε την ιδία ευκρίνεια

Το συμπέρασμα και για αυτό το ερώτημα όπως και στο 1.1.β είναι ότι για μεγάλο N έχουμε καλύτερα αποτελέσματα με την χρήση του PCM.

## 2.2

Υπολογίσαμε την εντροπία του για την κάθε περίπτωση κωδικοποίησης του σήματος. Έχοντας υπόψιν και όσα ειπώθηκαν στο ερώτημα 1.2 έχουμε τα παρακάτω αποτελέσματα στον πίνακα

| | **H(X)** |
|---|---|
| **N=2bits** $0 \leq H(X) \leq 2$ | 1.9688 |
| **N=4bits** $0 \leq H(X) \leq 4$ | 3.4512 |

Το σήμα της εικόνας είναι μη προβλέψιμο (παρόλο που δεν φτάνουμε την μέγιστη τιμή ακριβώς) και έχει ομοιόμορφη κατανομή, αν και καθώς αυξάνουμε τα N αυξάνεται ελάχιστα η προβλεψιμότητα και δεν φαίνεται να έχει το ίδιο ομοιόμορφη κατανομή όπως για N=2 bits.

Γενικά τα συμπεράσματά δεν είναι καθόλου διαφορετικά με αυτά που βγάλαμε στο 1ο ερώτημα. Λογικό αφού εξετάζουμε το ίδιο σχήμα pcm.

# ΜΕΡΟΣ 2

Δεν πρόλαβα να γράψω πράγματα για αυτό το κομμάτι της εργασίας, οπότε παραθέτω μόνο το κομμάτι του κώδικα που μπόρεσα να τελειώσω στις τελευταίες σελίδες του εγγράφου. Αφορά μόνο το κομμάτι του mapper και demapper.

## ΕΡΩΤΗΜΑ 3

Στις σύγχρονες ψηφιακές επικοινωνίες, ο κώδικας gray παίζει σημαντικό ρόλο στη διόρθωση σφαλμάτων. Για παράδειγμα, σε ένα σχήμα ψηφιακής διαμόρφωσης όπως το PAM όπου τα δεδομένα μεταδίδονται τυπικά σε σύμβολα 4 bit ή περισσότερων, το διάγραμμα αστερισμού του σήματος είναι διατεταγμένο έτσι ώστε τα μοτίβα bit που μεταφέρονται από γειτονικά σημεία αστερισμού να διαφέρουν μόνο κατά ένα bit. Συνδυάζοντας αυτό με τη διόρθωση σφαλμάτων προς τα εμπρός που μπορεί να διορθώσει σφάλματα ενός bit, είναι δυνατό για έναν δέκτη να διορθώσει τυχόν σφάλματα μετάδοσης που προκαλούν την απόκλιση ενός σημείου αστερισμού στην περιοχή ενός παρακείμενου σημείου. Αυτό καθιστά το σύστημα μετάδοσης λιγότερο ευαίσθητο στο θόρυβο.

# ΚΩΔΙΚΕΣ

## Μη – Ομοιόμορφος Κβαντιστής (Lloyd Max algorithm) -- Κώδικας

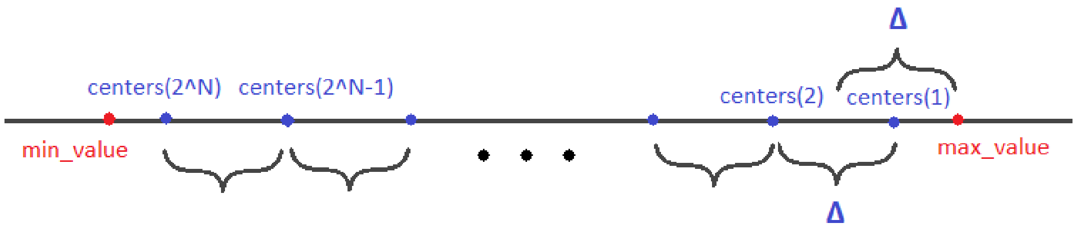
```
function [xq, centers, D, H] = LloydMax(x, N, min_value, max_value)

% Parameters
% x: input signal
% N: number of bit/sample that are gonna be used
% min_value: minimum accepted value of input signal
% max_value: maximum accepted value of input signal


% The Lloyd Max algorithm will start with the centers of the uniform quantizer
levels = 2^N;
delta= abs(max_value-min_value)/levels;

%find the centers

centers = zeros(levels,1);
centers(1) = max_value - delta/2;
for i=2:levels
    centers(i) = centers(i-1) - delta;
end

%THIS IS THE LLOYD MAX
D=[0 1]; %Distortion initialization
k=2;
e=10^-6;

sqnr=[];

while abs(D(k)-D(k-1))>=e
    %some initializations
    xq=zeros(size(x));
    d=zeros(size(x)); %to calculate the distortion in every point of the signal
    %variables needed for the calculations of new centers
    sum_x=zeros(levels,1);
    numOf_x=zeros(levels,1);

    %Calculating quantization zones limits
    t=zeros(levels+1,1);
    t(1)= max_value;
    for i=2:length(centers)
        t(i)=(centers(i-1)+centers(i))/2;
    end
    t(i+1)=min_value;


    %Now we will calculate the quantized signal
    for i=1:length(x)
        for j=1:length(t)-1
            if t(j)>x(i) && x(i)>=t(j+1)
                xq(i)=centers(j);
```

```

                %calculations for the new centers
                sum_x(j)=sum_x(j)+x(i);%sum of values in this zone
                %number of values in this zone
                numOf_x(j)=numOf_x(j)+1;
                break;
            elseif x(i)==t(1)
                %same as above
                xq(i)=centers(1);
                %calculations for the new centers
                sum_x(1)=sum_x(1)+x(i);
                numOf_x(1)=numOf_x(1)+1;
                break;
            end
        end


        %distortion array
        %we calculate the difference the quantized signal has
        %for every sample of the signal
        d(i)=abs(xq(i)- x(i));
    end

    %DISTORTION
    d_avg = sum(d)/length(d);
    D = [D d_avg];

    %New centers , they are equal to E[x] for T_k<x<T_k+1
    %so basically, equal to the average value of the signal in each zone
    for j=1:levels
        if numOf_x(j)~=0
            centers(j)= sum_x(j)/numOf_x(j);
        end
    end
    sqnr(k-1)=SQNR(x,xq);%just to see how SQNR changes in every loop
    %it's not needed for the Lloyd max algorithm

    k = k + 1;
end

%deleting the initialization values
D(1)=[];
D(1)=[];

%unrelated to lloyd max algorithm, only for analysis
figure
bar(sqnr);

%Calculation of Entropy
%not needed for Lloyd max algorithm
p=zeros(levels,1);%we have as many probabilities as centers
H=0;
for i=1:levels
    if numOf_x(i)~=0 %to avoid zones with no samples
        p(i)=numOf_x(i)/length(x); %probability of signals in a certain zone
        H=H+(p(i)*log2(1/p(i)));
    end
end

```

*Εικόνα 22- αρχείο LloydMax.m*

Για την δημιουργία του κώδικα , βασίστηκα σε διαφορές πηγές στο διαδίκτυο και σε ότι μας παρέχεται στο eclass μαζί με την εκφώνηση της άσκησης. Βρήκα και διάφορα παραδείγματα υλοποίησης του αλγορίθμου Lloyd Max τα οποία με βοήθησαν για το τελικό αποτέλεσμα. Ας γίνει όμως μια περιγραφή του συγκεκριμένου κώδικα.

Στις γραμμές 10-20 , ορίζουμε ένα τυχαίο σύνολο επιπέδων κβάντισης, όπου τα οποία κέντρα αντιστοιχούν σε ένα αντίστοιχο ομοιόμορφο κβαντιστή. Ουσιαστικά το εύρος κβάντισης είναι ένας σταθερός αριθμός Δ και κάθε κέντρο απέχει απόσταση Δ από το επόμενο κέντρο. Η νοοτροπία που ακολουθήθηκε γίνεται πιο κατανοητή με την παρακάτω εικόνα.



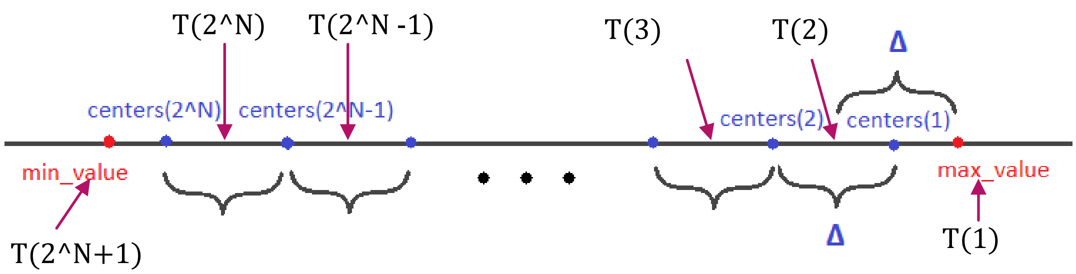

*Εικόνα 13- Κέντρα ομοιόμορφου κβαντιστή*

Στις επόμενες γραμμές του κώδικα , βλέπουμε τις επαναλήψεις του αλγορίθμου Lloyd Max.

Αρχικά ορίστηκαν κάποιες μεταβλητές οι οποίες θα θέσουν πόσες επαναλήψεις του κώδικα θα γίνουν. Συγκεκριμένα έχει γίνει η παραδοχή ότι το «ε» από την συνθήκη $|D_i - D_{i-1}| < \varepsilon$ θα είναι $\varepsilon = 10^{-6}$, αυθαίρετα επειδή θεώρησα ότι είναι ένα αρκετά μικρό νούμερο και μας καλύπτει. Με άλλα λόγια πιστεύω ότι αν η διαφορά της μέσης παραμόρφωσης αυτής της επανάληψης με την μέση παραμόρφωση στη προηγούμενη επανάληψη είναι μικρότερη από $10^{-6}$ το κβαντισμένο σήμα θα έχει συγκλίνει αρκετά στο αρχικό σήμα.

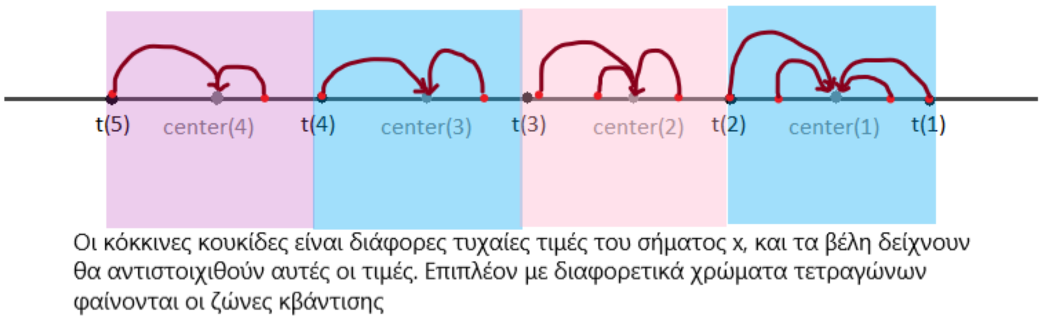
Στις γραμμές 29-90 υπάρχει η βασική επανάληψη του κώδικα Lloyd max και πιο ειδικά:

Ο υπολογισμός τον ζωνών κβάντισης στις γραμμές 37-43. Εδώ υπολογίζονται με τον τύπο που δίνεται στην εκφώνηση και ουσιαστικά γίνεται αυτό που φαίνεται στη παρακάτω εικόνα.




*Εικόνα 24 -Ζώνες κβάντισης*
*(τα arrays είναι σε φθίνουσα σειρά)*

Αντιστοίχιση κάθε δείγματος του σήματος x στον αντίστοιχο εκπρόσωπο της ζώνης στις γραμμές 46-65. Αν το x(ι) βρίσκεται ανάμεσα στις περιοχές T(j) και T(j+1) και πάνω στο T(j+1) τότε το κβαντισμένο σήμα θα έχει τιμή ίση με το κέντρο αυτής της περιοχής. (Το διάνυσμα t , είναι σε φθίνουσα σειρά γιαυτό γίνεται σύγκριση t(j)>x(i) && x(i)>=t(j+1). Μέσα σε αυτούς τους βρόγχους επανάληψης υπολογίζουμε επιπλέον το σύνολο των τιμών (sum_x) και το πλήθος τιμών (numOf_x) της ζώνης κβάντισης , για να μπορέσουμε στην πορεία να ορίσουμε τα καινούργια κέντρα με βάση την μέση τιμή της περιοχής. Η αντιστοίχιση γίνεται με τον τρόπο που φαίνεται στο παράδειγμα τις εικόνας 25.




*Εικόνα 25-Κβάντιση με N=2bits , 4ις ζώνες κβάντισης*

Έπειτα , στις επόμενες γραμμές μετράτε η παραμόρφωση του σήματος, αφαιρώντας από το αρχικό σήμα το κβαντισμένο σε κάθε επανάληψη, και τον μέσο όρο της παραμόρφωσης τον αποθηκεύουμε στο διάνυσμα D.

Στις γραμμές 78-84 υπολογίζονται τα νέα κέντρα με τον τρόπο που περιγράψαμε παραπάνω.

Στην γραμμή 86, καλείται η συνάρτηση SQNR , για την οποία θα αναφερθούμε αργότερα. Ο λόγος που υπάρχει στον κώδικα του Lloyd max είναι για την επίλυση του ερωτήματος 1.1.a

Το τελευταίο που κάνουμε είναι να διαγράψουμε τις τιμές αρχικοποίησης του διανύσματος D, ουσιαστικά εδώ έχουμε τελειώσει με τον αλγόριθμο του Lloyd Max.

Τέλος όμως υπολογίζουμε την εντροπία , πράγμα που θα αναλύσουμε αργότερα. Αυτές οι τελευταίες γραμμές κώδικα υπολογισμού της εντροπίας υπάρχουν για την επίλυση του ερωτήματος 1.2 (Το αποτέλεσμα επιστρέφεται στην έξοδο)

## Προσαρμοστική Διαμόρφωσή Δέλτα (ADM) – Κώδικας

Ο κώδικας για τον ADM, είναι πολύ πιο απλός όπως φαίνεται και στην εικόνα 26.

Το ADM , θα δέχεται ως είσοδο μόνο το σήμα το οποίο θέλουμε να κβαντίσουμε. Το πρώτο πράγμα που θα γίνεται στο σήμα είναι η υπερδειγματοληψία. Επιλέχθηκε ο δείκτης M με τον οποίο θα γίνεται η υπερδειγματοληψία να ισούται με 2. Έτσι δεν θα προκύπτει ένα πολύ μεγαλύτερο σήμα από το αρχικό μας.

Επιπλέον έχει επιλεχθεί K=1.5 , εφόσον έχει βρεθεί πειραματικά ότι είναι η βέλτιστη τιμή, και το αρχικό δέλτα το έχω θέσει να είναι ίσο με 1.

```
function [xq,b]=ADM(x)

%xq is the output of Decoder ADM
%b is the output of encoder ADM
x=interp(x,2);
b=zeros(size(x));
eq=zeros(size(x));
xq=zeros(size(x));
K=1.5;
delta=zeros(size(x));
delta(1)=1;

%ADM Encoder (or transmitter)
for i=2:length(x)
    %1bit quantizer
    if x(i)-xq(i-1)>=0
        b(i)=1;
    else
        b(i)=-1;
    end

    %Step control logic
    if b(i)==b(i-1)
        delta(i)=delta(i-1)*K;
    else
        delta(i)=delta(i-1)/K;
    end

    %multiplication of b(i) with delta gives as
    %the quantization error
    eq(i)=b(i)*delta(i);

    %delay
    xq(i)=eq(i)+xq(i-1);
end

end
```

*Εικόνα 26- αρχείο ADM.m*

Για τη συγγραφή αυτού του κώδικα , μελέτησα το σχήμα του ADM που μας δίνεται από το σχήμα στην εκφώνηση.

Το πρώτο πράγμα που γίνεται είναι η σύγκριση του κβαντισμένου με το αρχικό σήμα. Βρίσκουμε την διαφορά του δείγματος i του σήματός μας από το προηγούμενο δείγμα του αρχικού μας σήματος (λόγω του delay). Στην αρχή, αφού δεν έχουμε υπολογίσει ακόμη το κβαντισμένο σήμα , το έχουμε αρχικοποιήσει στο 0, στις επόμενες επαναλήψεις όμως του κώδικα θα έχουμε κάποια υπολογισμένη τιμή.

Η διαφορά τους θα μας δώσει το σφάλμα το οποίο θα κβαντίσουμε με ένα bit και το αποτέλεσμα θα είναι ένα διάνυσμα b(n) που θα έχει τιμές {+1,-1}. Ειδικότερα όταν το σφάλμα θα είναι αρνητικό θα θέτουμε να είναι ίσο με -1 ενώ αν είναι μεγαλύτερο ή ίσο του μηδενός ίσο με το +1 (γραμμές 16-20 ). Πολλαπλασιάζουμε τώρα κάθε τιμή του b(n) με το αντίστοιχο βήμα Δ (γραμμή 31) , που θα έχει υπολογιστεί με τον τρόπο που μας δίνεται από την εκφώνηση (γραμμή 23-27). Το αποτέλεσμα θα είναι το κβαντισμένο σφάλμα , όπου άμα προσθέσουμε σε αυτό την προηγούμενη τιμή το xq αυτό θα μας δώσει την τωρινή τιμή του xq (γραμμή 34). Η τωρινή τιμή του xq χρησιμοποιείται στην επόμενη επανάληψη. Θα γίνουν όσες επαναλήψεις είναι το μέγεθος του αρχικού σήματος, δηλαδή για κάθε δείγμα.

## PAM – Κώδικας

```
function [sm]=pam(x,M)

len=length(x);

%MAPPER
block = log2(M);
symbols=string(zeros(M,1));
%mapping with gray coding
for i=0:M-1
    symbols(i+1)=bin2gray(dec2bin(i));
end

%if length of array exceeds the number of elements add zeros
z=mod(len,block);
if z~=0
    for i=1:z
        x=[x 0];
    end
end

x=x.';
b=sprintf('%d',x); %one line string with no spaces

sm=zeros(b/block,1); %signal after mapping

for i=1:block:length(b)
    for j=1:M
        if b(i:i+block-1)==symbols(j)
            sm(mod(i,block))=symbols(j);
            %?? ??? ???? ??????????????? ?? ???? ?? ???????
        end
    end
end


%DEMAPPER
b=''; %empty char array
for i=1:length(sm)
    for j=1:M
        if sm(i)==symbols(j)
             b=append(b,symbols(j));%reconstruct the b array
        end
    end
end

xm=zeros(len,1);
for i=1:len
    xm(i)=str2num(b(i)); %reconstructed signal
end

end
```

```
function g = bin2gray(b)
g(1) = b(1);
for i = 2 : length(b);
    x = xor(str2num(b(i-1)), str2num(b(i)));
    g(i) = num2str(x);
end
```

Το σήμα εισόδου

```
%random binary signal
x = randi([0 1],10000,1); % Lb = 10000 - 100000 bits.
```