

# Ψηφιακές Τηλεπικοινωνίες

## **1ο Σετ Εργαστηριακών Ασκήσεων - Ακαδημαϊκό έτος 2022-2023**

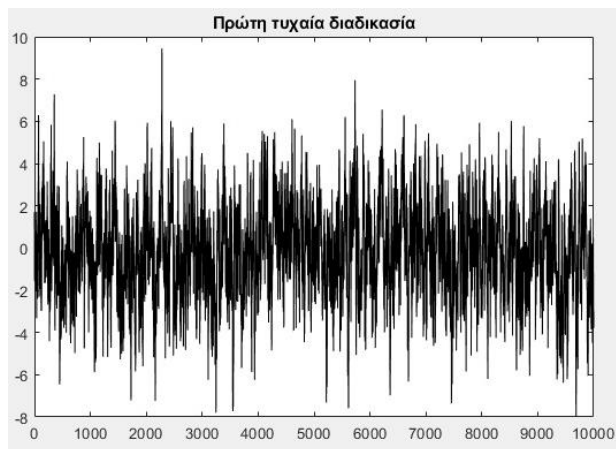
Ονοματεπώνυμο: Χρήστος Μεραντζής

Αριθμός Μητρώου: 1070936

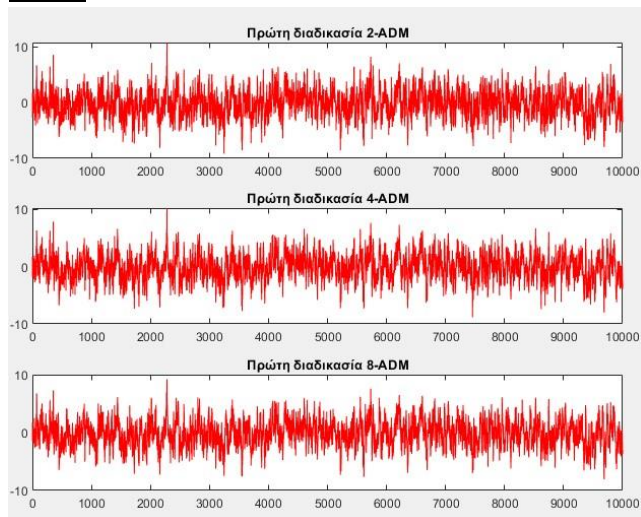
## Μέρος Α

### Ερώτημα 1.1

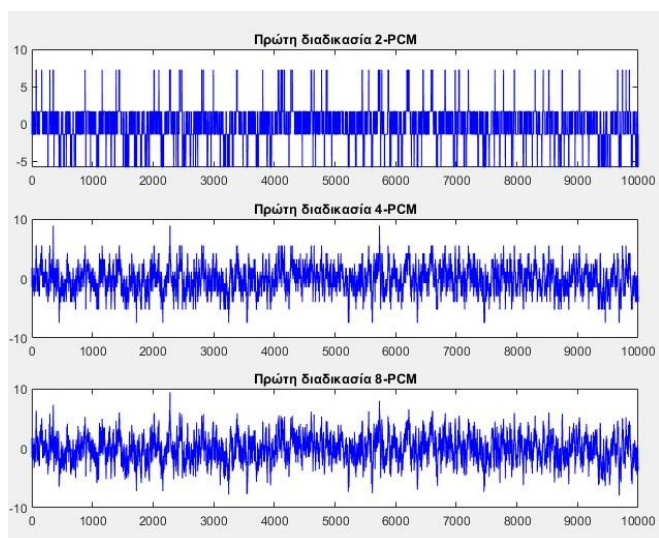
#### 1<sup>η</sup> τυχαία AR :



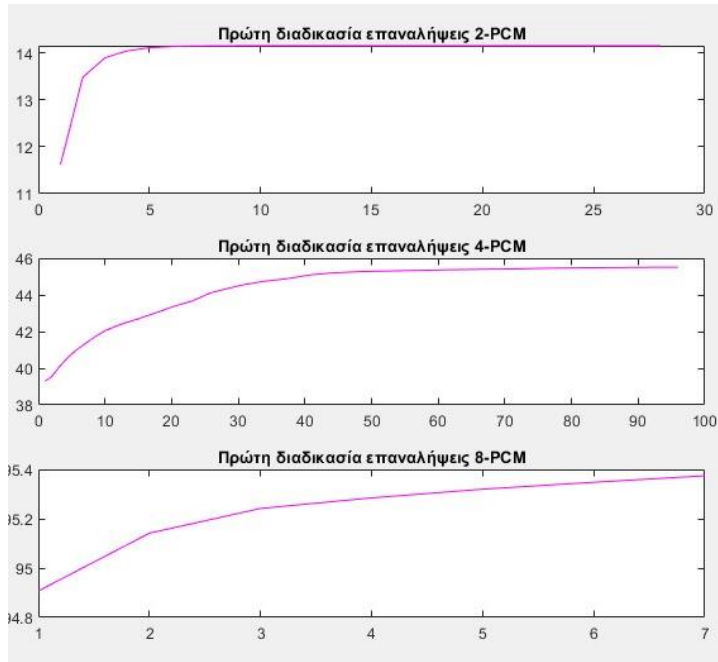
#### ADM:



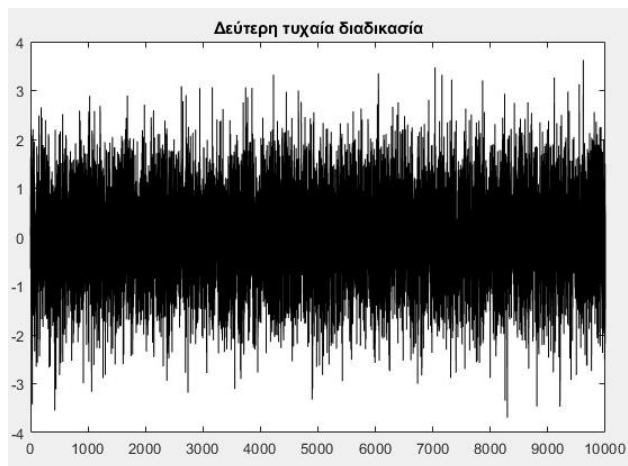
#### PCM:



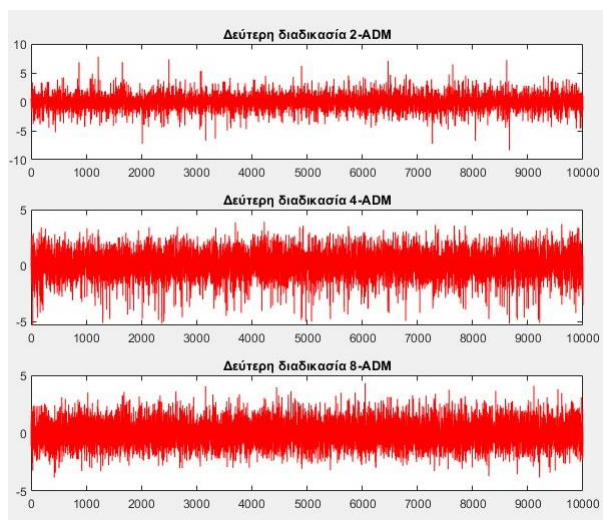
## Επαναλήψεις Lloyd-Max για τη πρώτη τυχαία διαδικασία:



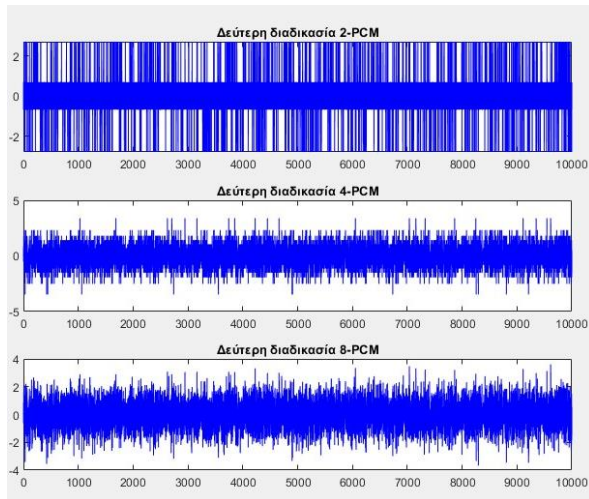
## 2<sup>η</sup> τυχαία AR :



## ADM :



## PCM :



## Οι τιμές SQNR που προέκυψαν είναι:

Πρώτη SQNR-2-ADM: 22.01 decibel	Δεύτερη SQNR-2-ADM: 0.93 decibel
Πρώτη SQNR-4-ADM: 34.11 decibel	Δεύτερη SQNR-4-ADM: 11.42 decibel
Πρώτη SQNR-8-ADM: 48.00 decibel	Δεύτερη SQNR-8-ADM: 26.10 decibel

Πρώτη SQNR-2-PCM: 14.16 decibel	Δεύτερη SQNR-2-PCM: 14.47 decibel
Πρώτη SQNR-4-PCM: 45.52 decibel	Δεύτερη SQNR-4-PCM: 45.52 decibel
Πρώτη SQNR-8-PCM: 95.38 decibel	Δεύτερη SQNR-8-PCM: 96.43 decibel

Με την αύξηση του  $N$  παίρνουμε καλύτερη τιμή στην μέτρηση του SQNR. Η πληροφορία μεταφέρεται καλύτερα και έτσι αυξάνεται η ισχύς του σε αντίθεση με αυτή του θορύβου. Από την θεωρία προκύπτει ότι με την αύξηση των  $N$ -bits κβάντισης βελτιώνεται η ανάλυση και αυτό επαληθεύεται και στις δύο τυχαίες πηγές.

### Ερώτημα 1.2

Οι εντροπίες για το σύστημα PCM του πρώτου και δεύτερου σχήματος υπολογίστηκαν:

Πρώτη Εντροπία 2-PCM = 1.382	Δεύτερη Εντροπία 2-PCM = 1.430
Πρώτη Εντροπία 4-PCM = 3.633	Δεύτερη Εντροπία 4-PCM = 3.614
Πρώτη Εντροπία 8-PCM = 7.069	Δεύτερη Εντροπία 8-PCM = 7.166

Όσο αυξάνεται ο αριθμός των bits κβάντισης ( $N=2 < N=4 < N=8$ ) η τιμή της εντροπίας αυξάνεται.

Η εντροπία εκφράζει το πλήθος bits που κάθε σύμβολο μεταφέρει. Όσο μεγαλύτερη τιμή έχει τόσο καλύτερη πληροφορία παίρνουμε.

Όσο το  $N$  αυξάνεται, δημιουργούνται  $2^N$  στάθμες κβάντισης. Στην περίπτωση όπου  $N=8$  προκύπτουν 256 στάθμες κβάντισης. Για παράδειγμα εάν τα σύμβολα από τις 256 στάθμες είναι μεταξύ τους ισοπίθανα, τότε η μέγιστη εντροπία στην περίπτωση 8-PCM είναι 8bit/symbol. Βλέπουμε ότι και οι δύο τιμές των 8-PCM, στην πρώτη(7,069) και στη δεύτερη(7,166), είναι όμοιες και προσεγγίζουν το 8-bits/symbol που όπως αναφέραμε είναι η μέγιστη εντροπία για 256 διαφορετικά σύμβολα.

Άρα για μεγαλύτερη τιμή του  $N$  λαμβάνουμε καλύτερη αναπαράσταση της πληροφορίας, όπως προκύπτει και από τις τιμές που λάβαμε.

### Ερώτημα 1.3

Όταν το  $N$  παραμένει μικρό, μέσω την τεχνικής ADM παίρνουμε γρηγορότερα αποτελέσματα. Αντίθετα, όσο αυξάνεται το  $N$ , λαμβάνουμε καλύτερα αποτελέσματα μέσω της τεχνικής PCM και το SQNR βελτιώνεται. Έτσι από την υπερδειγματοληψία της ADM, λαμβάνουμε αμεσότερα αποτελέσματα ενώ από την PCM προκύπτουν καλύτερες τιμές SQNR.

## Ερώτημα 2.1

### Χωρίς PCM:

Εικόνα χωρίς διαδικασία PCM

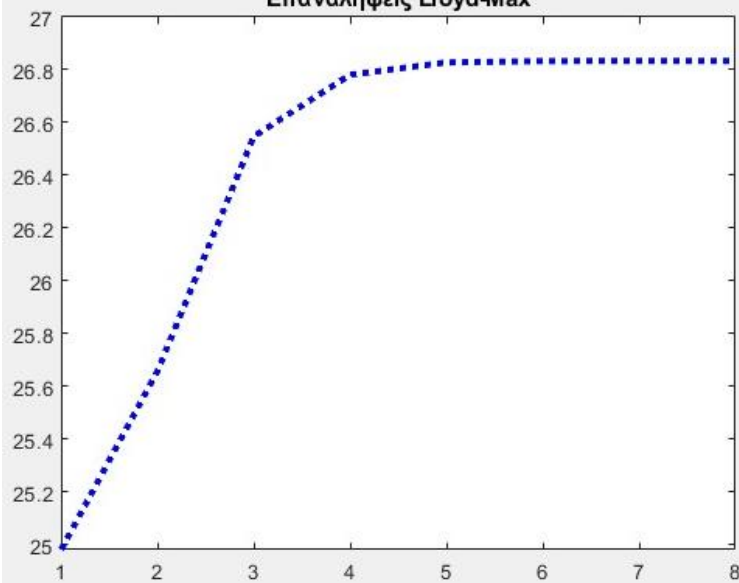


### 2-PCM:

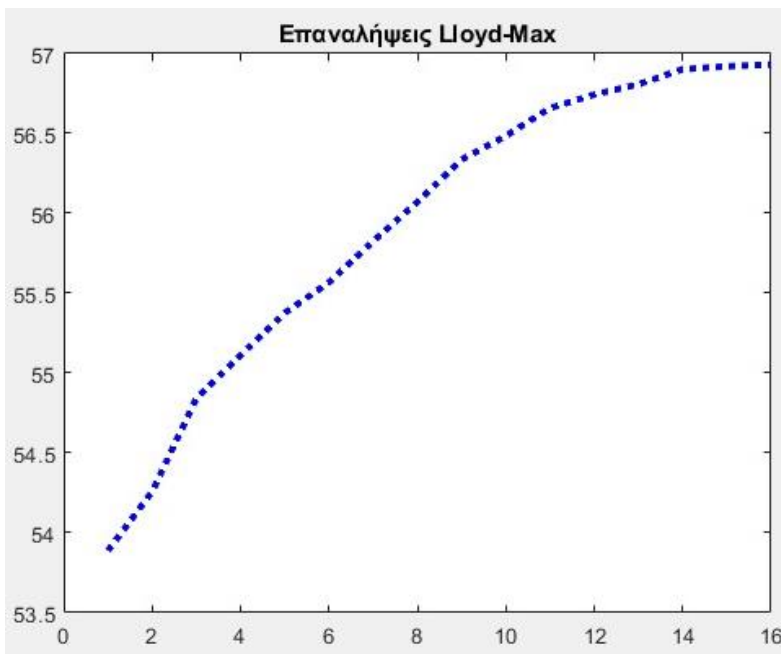
Εικόνα με διαδικασία 2-PCM



Επαναλήψεις Lloyd-Max



#### 4-PCM:



Υπολογίσαμε τις εξής τιμές SQNR:

SQNR-2-PCM: 26.87 decibel	SQNR-4-PCM: 56.96 decibel
---------------------------	---------------------------

Από τις μετρήσεις και το αποτέλεσμα που βλέπουμε, συμπεραίνουμε ότι με την αύξηση του  $N$ , το PCM βελτιώνει την ανάλυση του σήματος και έτσι έχουμε καλύτερη τιμή SQNR. Οδηγούμαστε στο ίδιο συμπέρασμα με το πρώτο ερώτημα.

## Ερώτημα 2.2

Υπολογίσαμε τις εξής Εντροπίες:

Εντροπία 2-PCM = 1.62	Εντροπία 4-PCM = 3.42
-----------------------	-----------------------

Με την αύξηση των  $N$  bits κβάντισης, έχουμε περισσότερες στάθμες, συνεπώς περισσότερα σύμβολα μεταφέρουν καλύτερη πληροφορία. Κι αυτό διότι έχουμε περισσότερα bits για κάθε σύμβολο, άρα αύξηση της εντροπίας όπως προκύπτει και από τις μετρήσεις 2-PCM και 4-PCM.

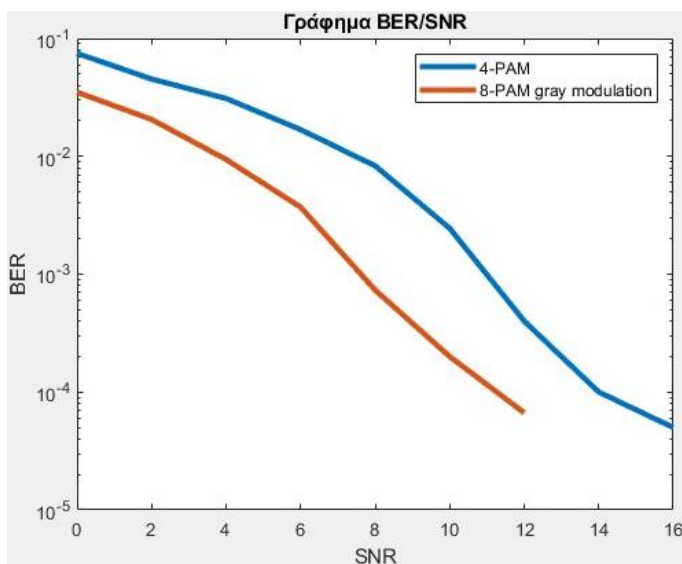
## Μέρος Β

### Ερώτημα 3.1

Μέσω συστημάτων PAM μεταδίδουμε διακριτά σήματα ως ακολουθία παλμών, όπου το πλάτος κάθε παλμού είναι ανάλογο της τιμής του δείγματος του σήματος. Η μεταφορά γίνεται από ένα κανάλι στο οποίο εισέρχεται και λευκός θόρυβος. Στη συνέχεια φωρατής μαζί με τη βοήθεια του demapper προσπαθούν να ταυτοποιήσουν την πληροφορία.

### Ερώτημα 3.2 / 3.3 / 3.4

Γράφημα BER για 4-PAM απλής κωδικοποίησης / 8-PAM κωδικοποίησης Gray:

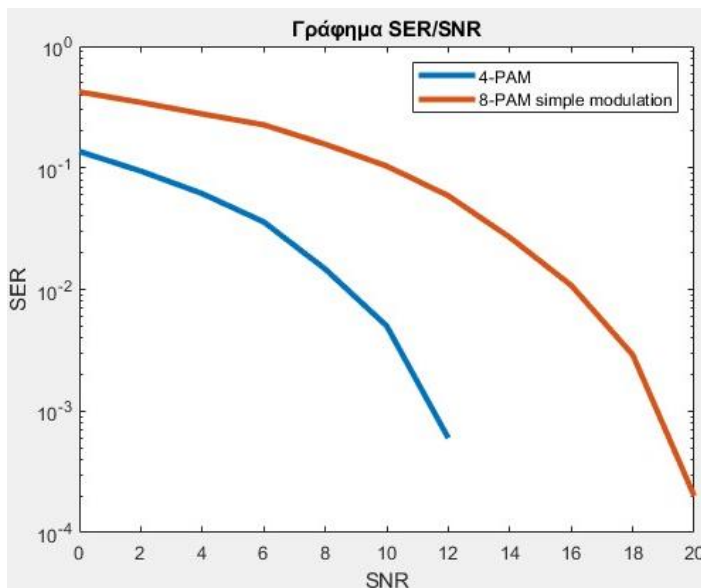


Όπως βλέπουμε το γράφημα για 8-PAM κωδικοποίησης Gray έχει μικρότερη τιμή BER σε σύγκριση με το 4-PAM απλής κωδικοποίησης. Δηλαδή λαμβάνουμε καλύτερα αποτελέσματα με κωδικοποίηση Gray, κι αυτό συμβαίνει διότι η κωδικοποίηση των συμβόλων αναπαρίστανται με μόνο ένα bit διαφορά από το προηγούμενο στο επόμενο.



### Ερώτημα 3.5

Γράφημα SER για 4-PAM / 8-PAM απλής κωδικοποίησης:



Είναι προφανές τόσο από το γράφημα όσο και από την θεωρία ότι το 4-PAM απλής κωδικοποίησης θα παρουσιάζει καλύτερα αποτελέσματα έναντι του 8-PAM απλής κωδικοποίησης, διότι παίρνουμε μικρότερες τιμές SER. Για λιγότερα σύμβολα έχουμε μικρότερη πιθανότητα σφάλματος.

### Κώδικας

#### pt\_meros1.m

```
Editor - C:\Users\xmara\Documents\MATLAB\pt_meros1.m
pt_meros1.m  x  +
1  clc;
2  fprintf("Επιλογή 1 - Υπολογισμός SQNR, οπτικών αποτελεσμάτων, επαναλήψεις Lloyd-Max και εντροπίας για πρώτη τυχαία διαδικασία με al=0.9 για N=2,4,8 bits \n");
3  fprintf("Επιλογή 2 - Υπολογισμός SQNR, οπτικών αποτελεσμάτων, επαναλήψεις Lloyd-Max και εντροπίας για πρώτη τυχαία διαδικασία με al=0.01 για N=2,4,8 bits \n");
4  fprintf("Επιλογή 3 - Υπολογισμός SQNR, οπτικών αποτελεσμάτων, εντροπίας για αρχείο εικόνας με χρήση 2-PCM \n");
5  fprintf("Επιλογή 4 - Υπολογισμός SQNR, οπτικών αποτελεσμάτων, εντροπίας για αρχείο εικόνας με χρήση 4-PCM \n\n");
6  user_input = input("Δώστε την επιλογή σας: ");
7
8  if user_input == 1
9      fprintf("\n");
10     % Δημιουργία Πηγής A με συντελεστή al=0.9
11     a=[1 -0.9];
12     y=filter(1,a,randn(10000,1));
13
14     % Σχήμα - Εμφάνιση πρώτης τυχαίας διαδικασίας
15     figure(1)
16     plot(y,'k');
17     title('Πρώτη τυχαία διαδικασία');
18     % Εύρεση ελάχιστης και μέγιστης τιμής του τυχαίου σήματος
19     minval=min(y);
20     maxval=max(y);
21     pcm=[2,4,8];
```

```

Editor - C:\Users\xmera\Documents\MATLAB\pt_meros1.m
pt_meros2.m  pt_meros1.m  adaptive_modulation.m  Lloyd_Max.m  +

21 - pcm=[2,4,8];
22 - %% Διαδικασία ADM
23 - for i=1:3
24 -     [new_y]=adaptive_modulation(y,2^i);
25 -     figure(2)
26 -     % Εμφάνιση οπτικών αποτελεσμάτων
27 -     subplot(3,1,i);
28 -     plot(new_y,'r');
29 -     title(sprintf('Πρώτη διαδικασία %d-ADM',pcm(i)));
30 -     % Εύρεση SQNR
31 -     SQNR = 10*log(mean(y.^2)/mean((y-new_y').^2));
32 -     fprintf('Πρώτη διαδικασία [SQNR-%d-ADM]: %.2f decibel \n', 2^i, SQNR);
33 -     fprintf('\n');
34 - end
35 - %% Διαδικασία PCM
36 - for i=1:3
37 -     %Κβάντιση PCM για N=2,4,8 bits χρησιμοποιώντας τη Lloyd_Max
38 -     [xq, centers, D]=Lloyd_Max(y,2^i,minval,maxval);
39 -     % Εμφάνιση οπτικών αποτελεσμάτων
40 -     figure(3)
41 -     subplot(3,1,i);
42 -     plot(centers(xq),'b');
43 -     title(sprintf('Πρώτη διαδικασία %d-PCM',pcm(i)));
44 -     % Εύρεση SQNR
45 -     SQNR = 10*log(mean(y.^2)/mean((y-centers(xq)).^2));
46 -     fprintf('Πρώτη διαδικασία [SQNR-%d-PCM]: %.2f decibel - ',2^i,SQNR);
47 -
48 -     figure(4)
49 -     % Επανάληψεις
50 -     kmax = 10*log(mean(y.^2)./D);
51 -     subplot(3,1,i);
52 -     plot(kmax,'m');
53 -     title(sprintf('Πρώτη διαδικασία επανάληψεις %d-PCM',pcm(i)));
54 -
55 -     %Εύρεση εντροπίας
56 -     entropy=0;
57 -     syms=length(y);
58 -     %Πιθανότητα συμβόλων κβάντισης
59 -     for j=1:2^(2^i)
60 -         p_level=sum((xq==j))/syms;
61 -         if (p_level~=0)
62 -             % Υπολογισμός με βάση τον τύπο
63 -             entropy=entropy-p_level*log2(p_level);
64 -         end
65 -     end
66 -     %Εμφάνιση της εντροπίας
67 -     fprintf('Εντροπία %d-PCM = %.3f\n',2^i,entropy);

```

```

Editor - C:\Users\xmara\Documents\MATLAB\pt_meros1.m
pt_meros2.m x pt_meros1.m x adaptive_modulation.m x Lloyd_Max.m x +
65 -         end
66 -         %Εμφάνιση της εντροπίας
67 -         fprintf('Εντροπία %d-PCM = %.3f\n',2^i,entropy);
68 -         fprintf('\n');
69 -     end
70 -
71 - elseif user_input==2
72 -     fprintf("\n");
73 -     % % Δημιουργία Πηγής A με συντελεστή α1=0.01
74 -     a=[1 -0.01];
75 -     y=filter(1,a,randn(10000,1));
76 -
77 -     % Σχήμα - Εμφάνιση δεύτερης τυχαίας διαδικασίας
78 -     figure(1)
79 -     plot(y,'k');
80 -     title('Δεύτερη τυχαία διαδικασία');
81 -     % Εύρεση ελάχιστης και μέγιστης τιμής του τυχαίου σήματος
82 -     minval=min(y);
83 -     maxval=max(y);
84 -     pcm=[2,4,8];
85 -     %% Διαδικασία ADM
86 -     for i=1:3
87 -         [new_y]=adaptive_modulation(y,2^i);
88 -         figure(2)
89 -         % Εμφάνιση οπτικών αποτελεσμάτων
90 -         subplot(3,1,i);
91 -         plot(new_y,'r');
92 -         title(sprintf('Δεύτερη διαδικασία %d-ADM',pcm(i)));
93 -
94 -
95 -         % Εύρεση SQNR
96 -         SQNR = 10*log(mean(y.^2)/mean((y-new_y').^2));
97 -         fprintf('Δεύτερη διαδικασία [SQNR-%d-ADM]: %.2f decibel \n', 2^i, SQNR);
98 -         fprintf('\n');
99 -     end
100 -     %% Διαδικασία PCM
101 -     for i=1:3
102 -         %Κβάντιση PCM για N=2,4,8 bits χρησιμοποιώντας τη Lloyd_Max
103 -         [xq, centers, D]=Lloyd_Max(y,2^i,minval,maxval);
104 -         % Εμφάνιση οπτικών αποτελεσμάτων
105 -         figure(3)
106 -         subplot(3,1,i);
107 -         plot(centers(xq),'b');
108 -         title(sprintf('Δεύτερη διαδικασία %d-PCM',pcm(i)));
109 -
110 -         % Εύρεση SQNR
111 -         SORN = 10*log(mean(v.^2)/mean((v-centers(xq')).^2));

```

```

Editor - C:\Users\xmera\Documents\MATLAB\pt_meros1.m
pt_meros2.m x pt_meros1.m x adaptive_modulation.m x Lloyd_Max.m x +
105 - figure(3)
106 - subplot(3,1,i);
107 - plot(centers(xq),'b');
108 - title(sprintf('Δεύτερη διαδικασία %d-PCM',pcm(i)));
109 -
110 - % Εύρεση SQNR
111 - SQNR = 10*log(mean(y.^2)/mean((y-centers(xq')).^2));
112 - fprintf('Δεύτερη διαδικασία [SQNR-%d-PCM]: %.2f decibel - ',2^i,SQNR);
113 -
114 - figure(4)
115 - % Επαναλήψεις
116 - kmax = 10*log(mean(y.^2)./D);
117 - subplot(3,1,i);
118 - plot(kmax,'m');
119 - title(sprintf('Δεύτερη διαδικασία επαναλήψεις για %d-PCM',pcm(i)));
120 -
121 - %Εύρεση εντροπίας
122 - entropy=0;
123 - syms=length(y);
124 - %Πιθανότητα συμβόλων κβάντισης
125 - for j=1:2^(2^i)
126 -     p_level=sum((xq==j))/syms;
127 -     if (p_level~=0)
128 -         % Υπολογισμός με βάση τον τύπο
129 -         entropy=entropy-p_level*log2(p_level);
130 -     end
131 - end
132 - %Εμφάνιση της εντροπίας
133 - fprintf('Εντροπία %d-PCM = %.3f\n',2^i,entropy);
134 - fprintf('\n');
135 - end
136 -
137 - elseif user_input==3
138 -     fprintf("\n");
139 -     %Ανάγνωση αρχείου cameraman.mat
140 -     load cameraman.mat
141 -
142 -     %Εμφάνιση αρχικής εικόνας
143 -     figure(1)
144 -     y=i(:);
145 -     y=(y-128)/128;
146 -     imshow(uint8(i))
147 -     title('Εικόνα χωρίς διαδικασία PCM');
148 -     %Μέγιστη και ελάχιστη τιμή του σήματος
149 -     minval = min(y);
150 -     maxval = max(y);
151 -

```

```

Editor - C:\Users\xmera\Documents\MATLAB\pt_meros1.m
pt_meros2.m  pt_meros1.m  adaptive_modulation.m  Lloyd_Max.m  +
148      %Μέγιστη και ελάχιστη τιμή του σήματος
149      minval = min(y);
150      maxval = max(y);
151
152      %2-PCM cameraman.mat
153      [xq, centers, D]=Lloyd_Max(y,2,minval,maxval);
154      out=centers(xq);
155      out=out*128+128;
156      final=reshape(out,256,256);
157
158      %figures 2-PCM
159      figure(2)
160      imshow(uint8(final));
161      title('Εικόνα με διαδικασία 2-PCM');
162
163      %Εύρεση SQRN
164      SQRN = 10*log(mean(y.^2)/mean((y-centers(xq')).^2));
165      fprintf('SQNR-2-PCM: %.2f decibel - ',SQRN);
166
167      %Επανάληψεις
168      repeats = 10*log(mean(y.^2)./D);
169      figure(3)
170      plot(repeats,':b','LineWidth',3);
171      title('Επανάληψεις Lloyd-Max');
172
173      % Εύρεση εντροπίας
174      entropy=0;
175      syms=length(y);
176      % Πιθανότητα συμβόλων κβάντισης
177      for j=1:2^2
178          p_level=sum((xq==j))/syms;
179          if (p_level~=0)
180              %Υπολογισμός Εντροπίας
181              entropy=entropy-p_level*log2(p_level);
182          end
183      end
184      %Εμφάνιση εντροπίας
185      fprintf('Εντροπία 2-PCM = %.2f\n',entropy);
186
187      else
188          fprintf("\n");
189          %Ανάγνωση αρχείου cameraman.mat
190          load cameraman.mat
191          %Εμφάνιση αρχικής εικόνας
192          figure(1)
193          y=i(:);
194          v=(v-128)/128;

```

```

Editor - C:\Users\xmara\Documents\MATLAB\pt_meros1.m
pt_meros2.m  pt_meros1.m  adaptive_modulation.m  Lloyd_Max.m  +
190 - load cameraman.mat
191 - %Εμφάνιση αρχικής εικόνας
192 - figure(1)
193 - y=i(:);
194 - y=(y-128)/128;
195 - imshow(uint8(i))
196 - title('Εικόνα χωρίς διαδικασία PCM');
197 - %Μέγιστη και ελάχιστη τιμή του σήματος
198 - minval = min(y);
199 - maxval = max(y);
200
201 %4-PCM cameraman.mat
202 - [xq, centers, D]=Lloyd_Max(y,4,minval,maxval);
203 - out=centers(xq);
204 - out=out*128+128;
205 - final=reshape(out,256,256);
206
207 %figures 4-PCM
208 - figure(2)
209 - imshow(uint8(final));
210 - title('Εικόνα με διαδικασία 4-PCM');
211
212 %Εύρεση SQRN
213 - SQRN = 10*log(mean(y.^2)/mean((y-centers(xq)).^2));
214 - fprintf('SQNR-4-PCM: %.2f decibel - ',SQRN);
215
216 %Επανάληψεις
217 - repeats = 10*log(mean(y.^2)./D);
218 - figure(3)
219 - plot(repeats,':b','LineWidth',3);
220 - title('Επανάληψεις Lloyd-Max');
221
222 %Εύρεση εντροπίας
223 - entropy=0;
224 - syms=length(y);
225 - % Πιθανότητα συμβόλων κβάντισης
226 - for j=1:2^4
227 -     p_level=sum((xq==j))/syms;
228 -     if (p_level~=0)
229 -         %Υπολογισμός Εντροπίας
230 -         entropy=entropy-p_level*log2(p_level);
231 -     end
232 - end
233 - %Εμφάνιση εντροπίας
234 - fprintf('Εντροπία 4-PCM = %.2f\n',entropy);
235 - end
236

```

## adaptive\_modulation.m

```
Editor - C:\Users\xmiera\Documents\MATLAB\adaptive_modulation.m
pt_meros2.m  pt_meros1.m  adaptive_modulation.m  Lloyd_Max.m  +

1 function [out]=adaptive_modulation(sig,n)
2     kapa=1.5;
3     size=length(sig);
4     delta=mean(abs(sig(2:size)-sig(1:size-1)));
5     source=interp(sig,n);
6     length2=length(source);
7     integer=source(1);
8     d(1)=1;
9     del=delta;
10    for i=2:length2
11        error=source(i)-integer;
12        d(i)=(error>0)-(error<0);
13        del=kapa*del*(d(i)==d(i-1))+(kapa^(-1))*del*(d(i)~=d(i-1));
14        integer=integer+d(i)*del;
15    end
16    src(1)=source(1);
17    integer=src(1);
18    del=delta;
19    for i=2:length2
20        del=kapa*del*(d(i)==d(i-1))+(kapa^(-1))*del*(d(i)~=d(i-1));
21        integer=integer+del*d(i);
22        src(i)=integer;
23    end
24    out =src(1:n:length2);
25 end
26
```

## Lloyd\_max.m

```
Editor - C:\Users\xmiera\Documents\MATLAB\Lloyd_Max.m
pt_meros2.m  pt_meros1.m  adaptive_modulation.m  Lloyd_Max.m  +

1 function [xq,centers,D]=Lloyd_Max(y,N,min_value,max_value)
2     steps = 2^N;
3     error=10^(-6);
4     qsteps = (max_value-min_value)/steps;
5     centers = max_value - qsteps/2-qsteps*((1:steps)-1);
6     src=(y<min_value)*(min_value+qsteps/2)+(y>=max_value)*(max_value-qsteps/2);
7     src=src+(y>min_value).*(y<max_value).*y;
8     xq=floor((max_value-src)/qsteps)+1;
9     D(1)=mean((y-centers(xq')).^2);
10    for i=2:1000
11        T=(centers(1:steps-1)+centers(2:steps))/2;
12        for j=2:steps-1
13            n=sum((y<T(j-1)).*(y>=T(j)));
14            if (n==0)
15                centers(j)=(T(j-1)+T(j))/2;
16            else
17                s=sum((y<T(j-1)).*(y>=T(j)).*y);
18                centers(j)=s/n;
19            end
20        end
21        xq=zeros(length(y),1);
22        for j=2:steps-1
23            xq=xq+(y<T(j-1)).*(y>=T(j)).*j;
24        end
25        xq=xq+(y>=T(1))+(y<=T(steps-1))*steps;
26        D(i)=mean((y-centers(xq')).^2);
27        if (abs(D(i)-D(i-1))<error)
28            break;
29        end
30    end
31 end
32
```

## pt\_meros2.m

```
Editor - C:\Users\xmera\Documents\MATLAB\pt_meros2.m
pt_meros2.m  pt_meros1.m  adaptive_modulation.m  Lloyd_Max.m  +

1  c1c;
2
3  % Μεταβλητές προς χρήση
4  Tsym=4*10^(-6);
5  Tc=0.4*10^(-6);
6  Ts=0.1*10^(-6);
7  g=sqrt(2/Tsym);
8  fc=1/Tc;
9  fprintf("Επιλογή 1 - Υπολογισμός γραφήματος BER για διαμορφώσεις 4-PAM απλής κωδικοποίησης και 8-PAM κωδικοποίησης Gray \n");
10 fprintf("Επιλογή 2 - Υπολογισμός γραφήματος SER για διαμορφώσεις 4-PAM και 8-PAM απλής κωδικοποίησης \n");
11 user_input = input("Δώστε την επιλογή σας: ");
12
13 %% 4-PAM
14 for k=1:length(4)
15     if user_input==1
16         fprintf('4-PAM BER \n');
17     else
18         fprintf('4-PAM SER \n');
19     end
20     bits_per_symbol=log2(4);
21     bits=10000*bits_per_symbol;
22     bin=randsrc(bits,1,[0 1]);
23     % Ανάγνωση συμβόλων σε μορφή δυαδική
24     for i=1:bits_per_symbol
25         pig1(:,i)=bin(i:bits_per_symbol:bits);
26     end
27     metritis=1;
28     t=0:Ts:Tsym;
29     % Μετατροπή σε δεκαδικό σύστημα
30     dekadiki=bi2de(pig1);
31     % Ανά 2 τιμές snr στο εύρος [0,20]
32     for snr=0:2:20
33
34         % Ανά 2 τιμές snr στο εύρος [0,20]
35         for snr=0:2:20
36             % Γίνεται ανάγνωση της δεκαδικής πηγής
37             input=dekadiki;
38             % Υπολογισμός ισχύος σήματος/θορύβου
39             for i=1:length(input)
40                 platos=(2*(input(i)+1)-(5));
41                 % Συνάρτηση σήματος
42                 sima=platos*g*cos(2*pi*fc*t);
43                 % Θόρυβος
44                 isxus=sum(sima.^2)/(length(sima));
45                 s=10*isxus/(10^(snr/10));
46                 N=sqrt(s)*randn(1,length(sima));
47                 % Τελικό σήμα: αρχική πληροφορία με θόρυβο
48                 sn = N + sima;
49                 func=sn.*g.*cos(2*pi*fc*t)*Ts;
50                 last=sum(func);
51                 find=5;
52                 for j=1:4
53                     d=abs((2*j-(5))-last);
54                     if(d<find)
55                         find=d;
56                         r=j;
57                     end
58                 end
59                 out(i)=r-1;
60             end
61
62             % Μετατροπή από δεκαδική σε δυαδική μορφή
63             rb=de2bi(out)';
64             % Εύρεση του Bit Error bits (BER) αλλά και
65             % (SER) Symbol Error bits
66             final=rb(:);
67             ber(metritis)=sum(final~=bin)/length(bin);
68             ser(metritis)=sum(dekadiki~=out)/length(input);
69             metritis=metritis+1;
70         end
71     end
72
73 %% BER 8-GRAY-PAM;
74 if (user_input==1)
75     figure(1)
76     semilogy(0:2:20,ber,'LineWidth',3);
77     title('Γράφημα BER/SNR')
78     xlabel('SNR');
79     ylabel('BER');
80     hold on
81     fprintf('8-PAM GRAY BER \n');
```



```

Editor - C:\Users\xmera\Documents\MATLAB\pt_meros2.m
pt_meros2.m  pt_meros1.m  adaptive_modulation.m  Lloyd_Max.m  +

75 - ylabel('BER');
76 - hold on
77 - fprintf('8-PAM GRAY BER \n');
78 - bits_per_symbol=log2(8);
79 - bits=10000*bits_per_symbol;
80 - bin=round(rand(bits,1));
81 - metritis=1;
82 - % Ανά 2 τιμή του snr στο εύρος [0,20]
83 - for snr=0:2:20
84 -     % Ανάγνωση των συμβόλων
85 -     for i=1:bits_per_symbol
86 -         symbols(:,i)=bin(i:bits_per_symbol:bits);
87 -     end
88 -     % Μετατροπή από δυαδική σε δεκαδική μορφή
89 -     dekadiki=bi2de(symbols);
90 -     input=dekadiki;
91 -     % Μετατροπή από δυαδική σε κώδικα gray
92 -     input=bin2gray(input,'pam',8);
93 -     g=sqrt(2/Tsym);
94 -     % Εύρεση ύψους, ισχύος και θορύβου
95 -     for i=1:length(input)
96 -         m=input(i)+1;
97 -         platos=(2*m-(9));
98 -         k=1;
99 -         % Βασική συνάρτηση σήματος
100 -        for t=0:Ts:Tsym
101 -            sima(k)=platos*g*cos(2*pi*fc*t);
102 -            k=k+1;
103 -        end
104 -        % Θόρυβος
105 -        isxus = sum(sima.^2)/(length(sima));
106 -        s = isxus/(10^(snr/10));
107 -        N = sqrt(s)*randn(1,length(sima));
108 -        sn= N + sima;
109 -        k=1;
110 -        for t=0:Ts:Tsym
111 -            func(k)=sn(k)*g*cos(2*pi*fc*t)*Ts;
112 -            k=k+1;
113 -        end
114 -        last=sum(func);
115 -        find=100000;
116 -
117 -        %Εύρεση συμβόλου
118 -        for j=1:8
119 -            d=abs((2*j-(9))-last);
120 -            if(d<find)
121 -                find=d;

```

```

Editor - C:\Users\xmera\Documents\MATLAB\pt_meros2.m
pt_meros2.m x pt_meros1.m x adaptive_modulation.m x Lloyd_Max.m x +
120 -         if(d<find)
121 -             find=d;
122 -             r=j;
123 -         end
124 -     end
125 -     out(i)=r-1;
126 - end
127
128 % Μετατροπή από κώδικα gray σε δυαδική μορφή
129 out=gray2bin(out,'pam',8);
130 % Μετατροπή από δεκαδική μορφή σε δυαδική
131 rb=de2bi(out)';
132 % Εύρεση BER
133 final=rb(:);
134 ber(metritis)=sum(final~=bin)/length(bin);
135 metritis=metritis+1;
136 end
137 % Γράφημα BER 4-PAM / 8-PAM gray modulation
138 figure(1)
139 semilogy(0:2:20,ber,'LineWidth',3);
140 legend('4-PAM','8-PAM gray modulation')
141 hold off;
142
143 else
144 %% SER 8-PAM simple modulation
145 figure(1)
146 semilogy(0:2:20,ser,'LineWidth',3);
147 title('Γράφημα SER/SNR')
148 xlabel('SNR');
149 ylabel('SER');
150 hold on;
151 fprintf('8-PAM SER \n');
152 bits_per_symbol=log2(8);
153 bits=10000*bits_per_symbol;
154 bin=randsrc(bits,1,[0 1]);
155 % Ανάγνωση συμβόλων σε δυαδική μορφή
156 for i=1:bits_per_symbol
157     pigi(:,i)=bin(i:bits_per_symbol:bits);
158 end
159 metritis=1;
160 t=0:Ts:Tsym;
161 % Μετατροπή από δυαδική μορφή σε δεκαδική
162 dekadiki=bi2de(pigi);
163
164 % Ανά 2 τιμή του snr στο εύρος [0,20]
165 for snr=0:2:20
166     input=dekadiki.*

```

```

Editor - C:\Users\xmpera\Documents\MATLAB\pt_meros2.m
pt_meros2.m  pt_meros1.m  adaptive_modulation.m  Lloyd_Max.m  +
160 -         t=0:Ts:Tsym;
161 -         % Μετατροπή από δυαδική μορφή σε δεκαδική
162 -         dekadiki=bi2de(pigi);
163 -
164 -         % Ανά 2 τιμή του snr στο εύρος [0,20]
165 -         for snr=0:2:20
166 -             input=dekadiki;
167 -             % Υπολογισμός ύψους, ισχύος και θορύβου
168 -             for i=1:length(input)
169 -                 platos=(2*(input(i)+1)-(9));
170 -                 % Βασική συνάρτηση σήματος
171 -                 sima=platos*g*cos(2*pi*fc*t);
172 -                 %Θόρυβος
173 -                 isxus=sum(sima.^2)/(length(sima));
174 -                 s=10*isxus/(10^(snr/10));
175 -                 N=sqrt(s)*randn(1,length(sima));
176 -                 sn = N + sima;
177 -                 func=sn.*g.*cos(2*pi*fc*t)*Ts;
178 -                 last=sum(func);
179 -                 %Εύρεση συμβόλου
180 -                 find=9;
181 -                 for j=1:8
182 -                     d=abs((2*j)-(9))-last);
183 -                     if(d<find)
184 -                         find=d;
185 -                         r=j;
186 -                     end
187 -                 end
188 -                 out(i)=r-1;
189 -             end
190 -             % Μετατροπή από δεκαδική μορφή σε δυαδική
191 -             rb=de2bi(out)';
192 -             % Εύρεση του SER
193 -             final=rb(:);
194 -             ser(metritis)=sum(dekadiki~=out')/length(input);
195 -             metritis=metritis+1;
196 -         end
197 -         % Γράφημα 4-PAM, 8-PAM simple modulation
198 -         figure(1)
199 -         semilogy(0:2:20,ser,'LineWidth',3);
200 -         title('Γράφημα SER/SNR')
201 -         xlabel('SNR');
202 -         ylabel('SER');
203 -         legend('4-PAM','8-PAM simple modulation')
204 -         hold off;
205 -     end
206 -

```