



# ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

Μάθημα: Ψηφιακές Επικοινωνίες Ι

Ονοματεπώνυμο: Ειρήνη Δόντη

Α.Μ: 03119839

3<sup>η</sup> Σειρά Ασκήσεων

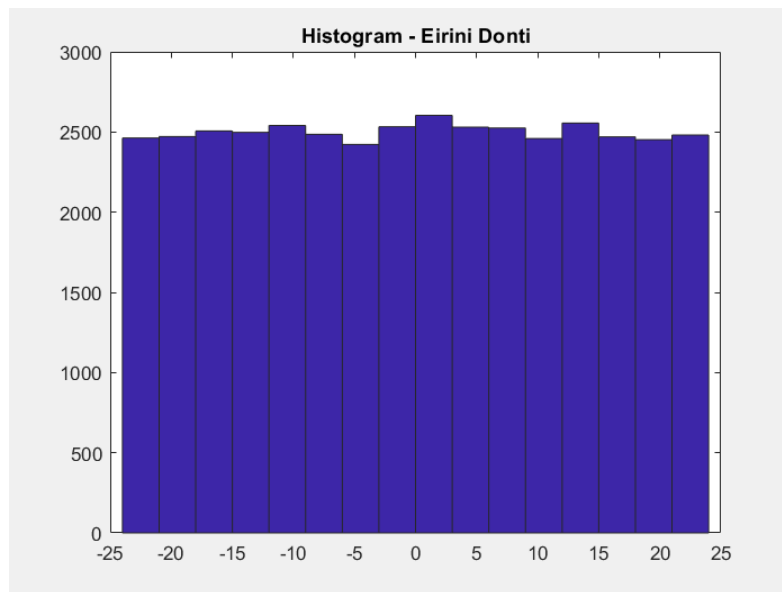
Αθήνα 2022

## Περιεχόμενα

Μέρος 1 <sup>ο</sup> : Διερεύνηση του κώδικα εξομοίωσης.....	1
Μέρος 2 <sup>ο</sup> : Καμπύλες επίδοσης (BER συναρτήσει του σηματοθορυβικού λόγου).....	7
Μέρος 3 <sup>ο</sup> : Υλοποίηση με συνέλιξη – Χρήση άλλων παλμών .....	13

## Μέρος 1<sup>ο</sup>: Διερεύνηση του κώδικα εξομίωσης (α)

Τροποποιούμε τον κώδικα του Παραδείγματος 3.2 των σημειώσεων και το αποθηκεύουμε σε αρχείο με το όνομα `ask_errors.m`. Επαληθεύουμε τον κώδικα με την εντολή `hist(x,A)` για τον υπολογισμό του ιστογράμματος που φαίνεται παρακάτω:



Παρατηρούμε, από το παραπάνω διάγραμμα, ότι τα στοιχεία  $x$  ακολουθούν πράγματι την ομοιόμορφη κατανομή, καθώς έχουμε ισοπίθανα σύμβολα. Το γεγονός αυτό επαληθεύεται από τη μορφή του ιστογράμματος, αφού όλα τα δείγματα βρίσκονται στην ίδια στάθμη. Οπότε, τα στοιχεία του διανύσματος  $x$  ακολουθούν ομοιόμορφη κατανομή.

Επίσης, τροποποιούμε την εντολή 15 για τον θεωρητικό υπολογισμό της ισχύος όπως φαίνεται παρακάτω:

```
Px=(d^2/4)*(L^2-1)/3 % θεωρητική ισχύς σήματος  
sum(x.^2)/length(x) % μετρούμενη ισχύς σήματος (για επαλήθευση)
```

Τροποποιούμε, επιπλέον, την εντολή 23 για τον υπολογισμό του διανύσματος  $A$ :

```
A=[(-L+1)*d/2:2*d/2:(L-1)*d/2];
```

Αφότου εκτελέσουμε το πρόγραμμα, παρατηρούμε ότι ο θεωρητικός υπολογισμός της ισχύος πλησιάζει, σε μεγάλο βαθμό, την πειραματική προσέγγιση της ισχύος, όπως φαίνεται στο Command Window του Matlab:

```
Px =  
  
191.2500  
  
ans =  
  
191.1839
```

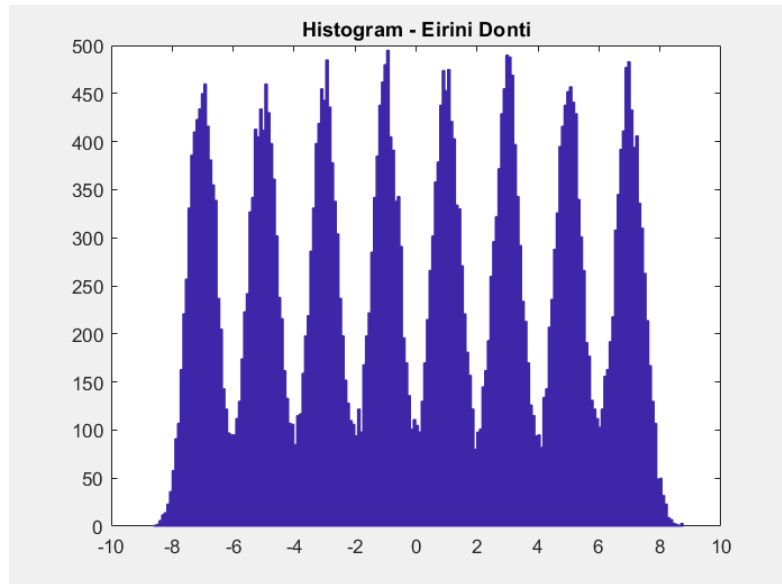
Παρακάτω, παραθέτουμε τον τροποποιημένο κώδικα του ερωτήματος:

```
function errors=ask_errors(k,Nsymb,nsamp,EbNo)  
% Η συνάρτηση αυτή εξομοιώνει την παραγωγή και αποκωδικοποίηση  
% θορυβώδους σήματος L-ASK και μετρά τον αριθμό των εσφαλμένων συμβόλων.  
% Επιτρέπει τον αριθμό των εσφαλμένων συμβόλων (στη μεταβλητή errors).  
% k είναι ο αριθμός των bits/σύμβολο, επομένως L=2^k -- ο αριθμός των  
% διαφορετικών πλατών  
% Nsymb είναι ο αριθμός των παραγόμενων συμβόλων (μήκος ακολουθίας LASK)  
% nsamp ο αριθμός των δειγμάτων ανά σύμβολο (oversampling ratio)  
% EbNo είναι ο ανηγμένος σηματοθορυβικός λόγος Eb/No, σε db  
%  
k=mod(19839,2)+3; L=2^k; d=3; Nsymb=40000; A=[(-L+1)*d/2:2*d/2:(L-1)*d/2];  
%SNR=EbNo-10*log10(nsamp/2/k); % SNR ανά δείγμα σήματος  
% Διάνυσμα τυχαίων ακεραίων {±1*d/2, ±3*d/2, ... ±(L-1)*d/2}. Να επαληθευθεί  
x=[2*floor(L*rand(1,Nsymb))-L+1]*d/2; hist(x,A); title("Histogram - Eirini Donti")  
Px=(d^2/4)*(L^2-1)/3 % θεωρητική ισχύς σήματος  
sum(x.^2)/length(x) % μετρούμενη ισχύς σήματος (για επαλήθευση)
```

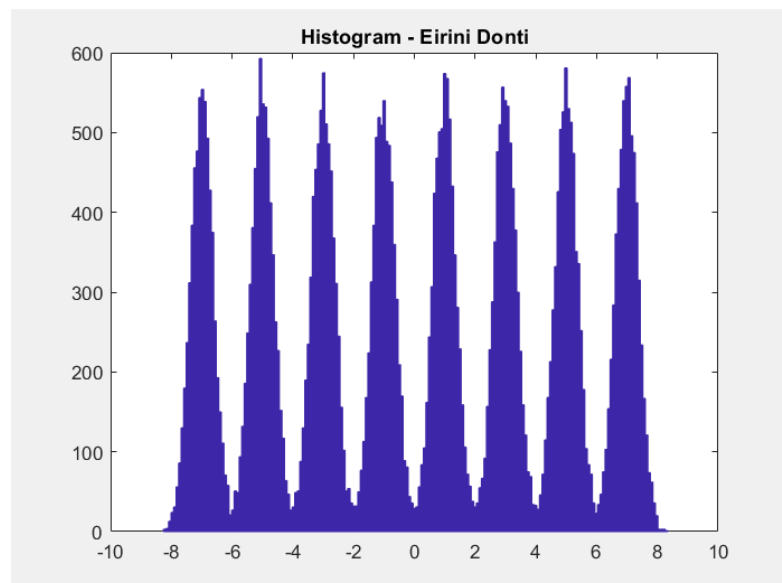
(β)

Με τιμές των παραμέτρων της συνάρτησης  $k=3$ ,  $M=50000$  και  $nsamp=16$  μεταβάλλουμε τις τιμές του σηματοθορυβικού λόγου  $E_bN_0$ :

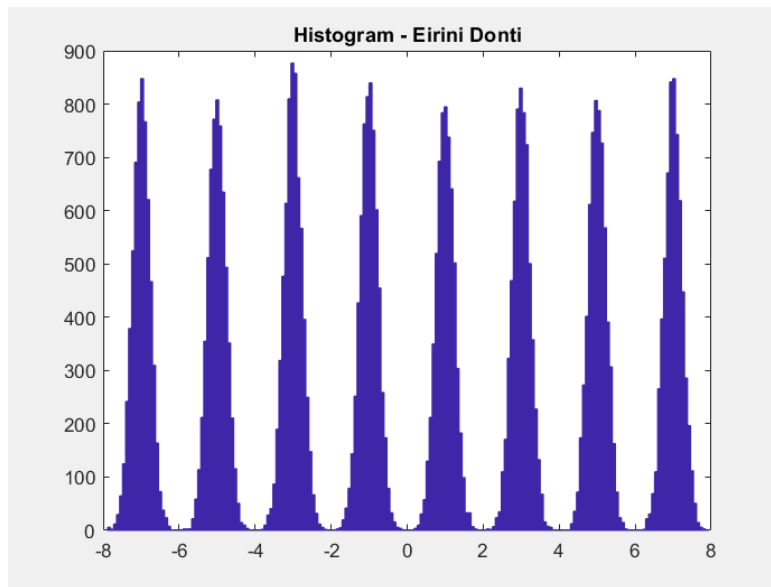
Για τιμή της μεταβλητής  $E_bN_0 = 12$  dB, προκύπτει το παρακάτω ιστόγραμμα:



Για τιμή της μεταβλητής  $E_bN_0 = 14$  dB, προκύπτει το παρακάτω ιστόγραμμα:



Για τιμή της μεταβλητής  $E_b/N_0 = 18$  dB, προκύπτει το παρακάτω ιστόγραμμα:



Παρατηρούμε ότι τα διαγράμματα που προέκυψαν παραπάνω, έχουν αρκετές διαφορές. Καταρχάς παρατηρούμε ότι, καθώς αυξάνεται ο σηματοθορυβικός λόγος  $E_b/N_0$ , τα δείγματα τείνουν να πυκνώσουν σε συγκεκριμένα σημεία και κατ'επέκταση να «αραιώνει» το διάγραμμα. Επίσης, παρατηρούμε ότι, στον κατακόρυφο άξονα, καθώς αυξάνεται ο σηματοθορυβικός λόγος  $E_b/N_0$ , αυξάνεται η στάθμη του διαγράμματος.

Παρακάτω, παραθέτουμε τον τροποποιημένο κώδικα του ερωτήματος:

```
function errors=ask_errors(k,Nsymb,nsamp,EbNo)
% Η συνάρτηση αυτή εξομοιώνει την παραγωγή και αποκωδικοποίηση
% θορυβώδους σήματος L-ASK και μετρά τον αριθμό των εσφαλμένων συμβόλων.
% Επιτρέπει τον αριθμό των εσφαλμένων συμβόλων (στη μεταβλητή errors).
% k είναι ο αριθμός των bits/σύμβολο, επομένως  $L=2^k$  -- ο αριθμός των
% διαφορετικών πλατών
% Nsymb είναι ο αριθμός των παραγόμενων συμβόλων (μήκος ακολουθίας LASK)
% nsamp ο αριθμός των δειγμάτων ανά σύμβολο (oversampling ratio)
% EbNo είναι ο ανηγμένος σηματοθορυβικός λόγος  $E_b/N_0$ , σε db
%
%  $k=\text{mod}(19839,2)+3$ ;  $L=2^k$ ;  $d=3$ ;  $Nsymb=40000$ ;  $A=[(-L+1)*d/2:2:d/2:(L-1)*d/2]$ ; %1)
 $k=3$ ;  $M=50000$ ;  $nsamp=16$ ;  $EbNo = 12$ ;  $L=2^k$ ; %2)
 $SNR=EbNo-10*\log_{10}(nsamp/2/k)$ ; % SNR ανά δείγμα σήματος
% Διάνυσμα τυχαίων ακεραίων  $\{\pm 1*d/2, \pm 3*d/2, \dots, \pm (L-1)*d/2\}$ . Να επαληθευθεί
 $x=[2*\text{floor}(L*\text{rand}(1,Nsymb))-L+1]*d/2$ ;  $\text{hist}(x,A)$ ;  $\text{title}('Histogram - Eirini Donti')$ 
 $x=[2*\text{floor}(L*\text{rand}(1,M))-L+1]$ ;
 $Px=(d^2/4)*(L^2-1)/3$  % θεωρητική ισχύς σήματος %1)
 $\text{sum}(x.^2)/\text{length}(x)$  % μετρούμενη ισχύς σήματος (για επαλήθευση) %1)
 $Px=(L^2-1)/3$ ; % θεωρητική ισχύς σήματος
 $\text{sum}(x.^2)/\text{length}(x)$ ; % μετρούμενη ισχύς σήματος (για επαλήθευση)
 $y=\text{rectpulse}(x,nsamp)$ ;
 $n=\text{wgn}(1,\text{length}(y),10*\log_{10}(Px)-SNR)$ ;
 $y_{noisy}=y+n$ ; % θορυβώδες σήμα
 $y=\text{reshape}(y_{noisy},nsamp,\text{length}(y_{noisy})/nsamp)$ ;
 $\text{matched}=\text{ones}(1,nsamp)$ ;
 $z=\text{matched}*y/nsamp$ ;
 $\text{hist}(z,200)$ ;  $\text{title}('Histogram - Eirini Donti');$  %2)
```

(γ)

Η εντολή 20 υπολογίζει τον ανασχηματισμό του σήματος με το θόρυβο, έτσι ώστε η κάθε στήλη να έχει τα δείγματα μόνο ενός παλμού.

Η εντολή 22 υπολογίζει την απόκριση του προσαρμοσμένου φίλτρου για όλους τους παλμούς.

Η μεταβλητή *y* είναι πίνακας, ο οποίος έχει διάσταση 16x50000. Εκτελώντας την εντολή `size(y)` συμπεραίνουμε το παραπάνω, όπως εμφανίζεται και στο Command Window του Matlab. Το αποτέλεσμα είναι λογικό, καθώς κάθε στήλη είναι τα 16 δείγματα ενός παλμού, όπως αναφέραμε παραπάνω.

```
size_y =  
  
16      50000
```

Η εντολή `matched` είναι η κρουστική απόκριση του προσαρμοσμένου φίλτρου, δηλαδή είναι ένας παλμός με μοναδιαία στοιχεία σε 16 γραμμές 1x16. Εκτελώντας την εντολή `size(matched)` συμπεραίνουμε τα παραπάνω, όπως εμφανίζεται και στο Command Window του Matlab.

```
size_matched =  
  
1      16
```

Η μεταβλητή *z* είναι ουσιαστικά ένα διάνυσμα, το οποίο εκφράζει την απόκριση του προσαρμοσμένου φίλτρου για όλους τους παλμούς. Η μεταβλητή *z* είναι, ουσιαστικά, τα αντίστοιχα πλάτη του *x*, οπότε η μεταβλητή *z* είναι μεγέθους 1x50000.

Εκτελώντας την εντολή `size(z)`, συμπεραίνουμε τα παραπάνω, όπως εμφανίζεται και στο Command Window του Matlab.

```
size_z =  
  
1      50000
```

(δ)

Ο βρόχος στις σειρές κώδικα 24-27 λειτουργεί ως ανιχνευτής ελάχιστης απόστασης για την L-ask. Με άλλα λόγια, υπολογίζεται η απόσταση από το κοντινότερο σημείο, δηλαδή τα πλάτη και επιλέγεται το σημείο που απέχει τη μικρότερη απόσταση από αυτό.

Ο κώδικας που χρησιμοποιήθηκε για τα παραπάνω υποερωτήματα είναι ο παρακάτω:

```
function errors=ask_errors(k,Nsymb,nsamp,EbNo)
% Η συνάρτηση αυτή εξομοιώνει την παραγωγή και αποκωδικοποίηση
% θορυβώδους σήματος L-ASK και μετρά τον αριθμό των εσφαλμένων συμβόλων.
% Επιστρέφει τον αριθμό των εσφαλμένων συμβόλων (στη μεταβλητή errors).
% k είναι ο αριθμός των bits/σύμβολο, επομένως L=2^k -- ο αριθμός των
% διαφορετικών πλατιών
% Nsymb είναι ο αριθμός των παραγόμενων συμβόλων (μήκος ακολουθίας LASK)
% nsamp ο αριθμός των δειγμάτων ανά σύμβολο (oversampling ratio)
% EbNo είναι ο ανηγμένος σηματοθορυβικός λόγος Eb/No, σε db
%
%k=mod(19839,2)+3; L=2^k; d=3; M=50000; nsamp=16; EbNo = 12; A=[(-L+1)*d/2:2*d/2:(L-1)*d/2]; %1)
k=3; M=50000; nsamp=16; EbNo = 12; L=2^k; %2)
L=2^k;
SNR=EbNo-10*log10(nsamp/2/k); % SNR ανά δείγμα σήματος
% Διάνυσμα τυχαίων ακεραίων {±1*d/2, ±3*d/2, ... ±(L-1)*d/2}. Να επαληθευθεί
%x=[2*floor(L*rand(1,M))-L+1]*d/2; hist(x,A); title("Histogram - Eirini Donti")
x=[2*floor(L*rand(1,M))-L+1];
%Px=(d^2/4)*(L^2-1)/3 % θεωρητική ισχύς σήματος %1)
%sum(x.^2)/length(x) % μετρούμενη ισχύς σήματος (για επαλήθευση) %1)
Px=(L^2-1)/3; % θεωρητική ισχύς σήματος
sum(x.^2)/length(x); % μετρούμενη ισχύς σήματος (για επαλήθευση)
y=rectpulse(x,nsamp);
n=wgn(1,length(y),10*log10(Px)-SNR);
ynois=y+n; % θορυβώδες σήμα
y=reshape(ynois,nsamp,length(ynois)/nsamp); %size_y = size(y)
matched=ones(1,nsamp); %size_matched=size(matched)
z=matched*y/nsamp; size_z=size(z)
hist(z,200); title("Histogram - Eirini Donti"); %2)
%A=[-L+1:2:L-1];
for i=1:length(z)
[m,j]=min(abs(A-z(i)));
z(i)=A(j);
end
err=not(x==z);
errors=sum(err);
end
```

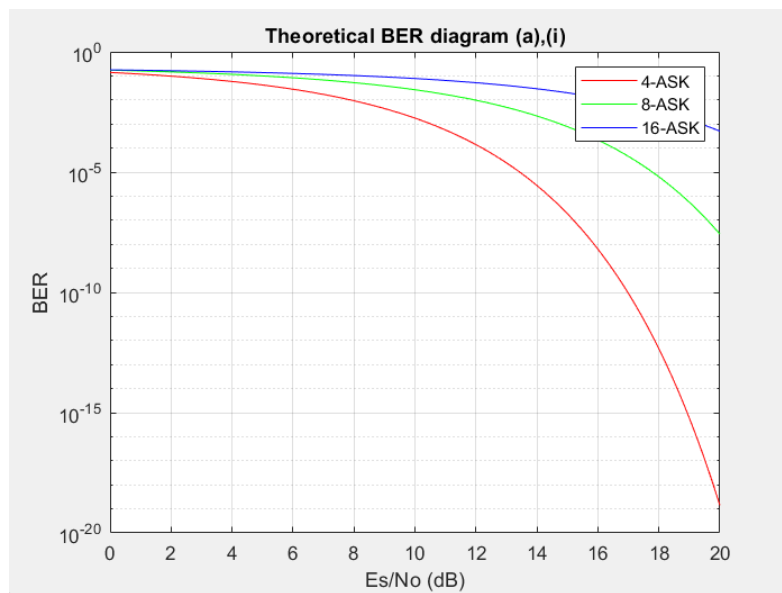


Μέρος 2<sup>ο</sup>: Καμπύλες επίδοσης (BER συναρτήσει του σηματοθορυβικού λόγου)  
 Επαληθεύουμε την καμπύλη του σχήματος 3.10 των σημειώσεων για L-ASK με  $L=2^k$   
 και  $k=\text{mod}(19839,2)+3$ . Σχεδιάζουμε τη θεωρητική καμπύλη και ‘διαβάζουμε’ την  
 καμπύλη 8-ASK, γράφοντας στο αρχείο υποβολής τις τιμές BER για  $E_b/N_0 =$   
 $\{9,11,13,15\}$  dB.

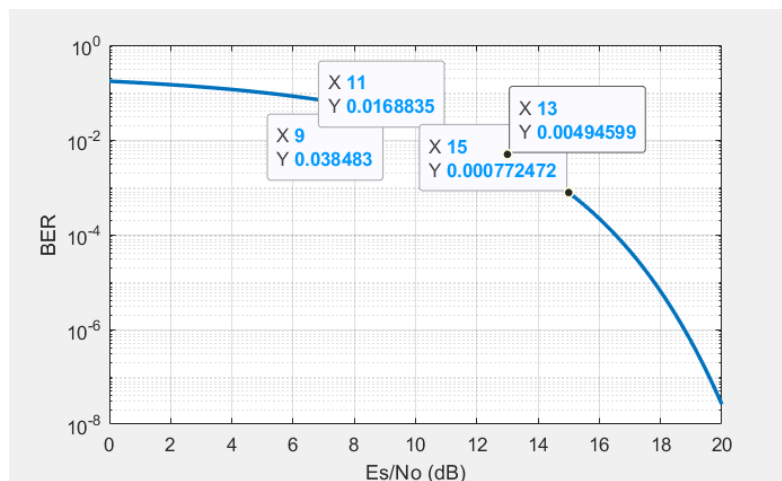
Παρακάτω, παρουσιάζονται οι ζητούμενες καμπύλες που προέκυψαν:

(α)

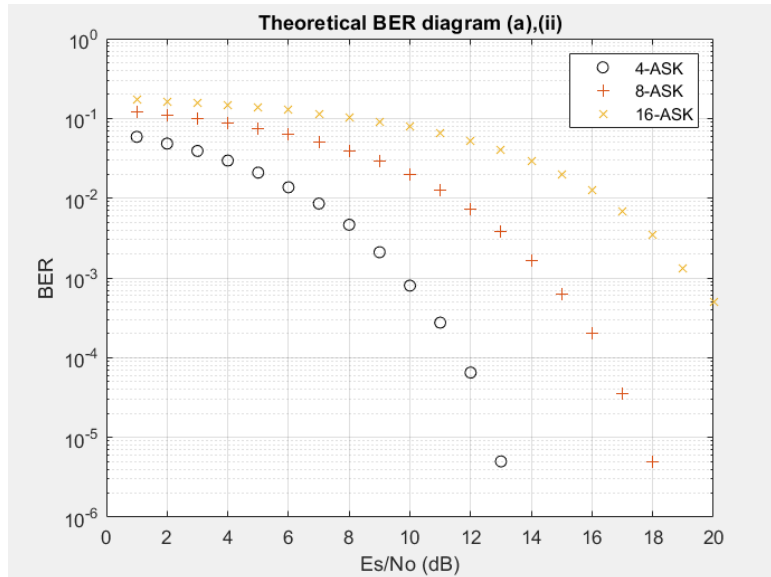
(i) Με δικό μας κύριο πρόγραμμα που βασίζεται στη χρήση σχέσης (3.33) των  
 σημειώσεων και την προσέγγιση  $BER \approx P_e / \log_2 L$ , συμπεραίνουμε το κάτω διάγραμμα:



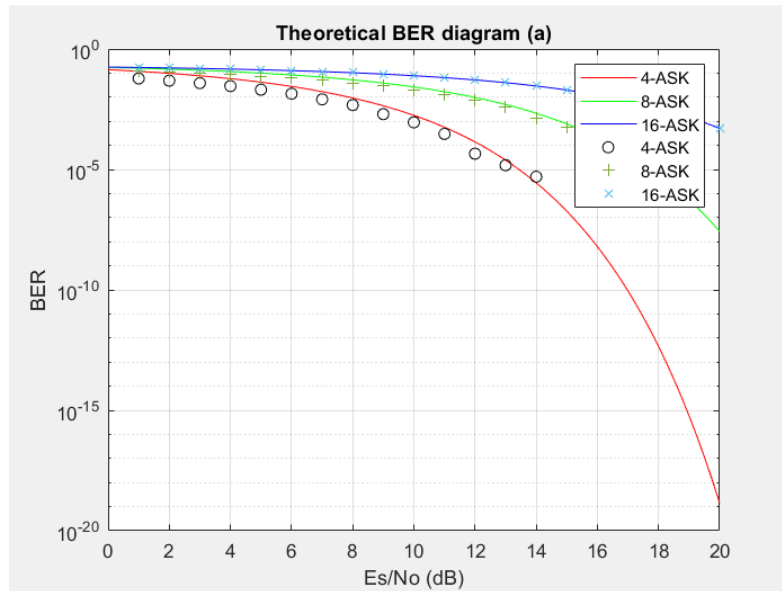
Για  $L = 8$  (8-ASK), διαβάζουμε από την καμπύλη τις τιμές BER, οι οποίες είναι οι  $Y$   
 τιμές που αναγράφονται κάτω από τις τιμές  $X$  της τιμής  $E_b/N_0 = \{9,11,13,15\}$  dB.



(ii) Με δικό μας κύριο πρόγραμμα, το οποίο θα καλεί τη συνάρτηση `ask_errors()`, μία φορά για κάθε διαφορετική τιμή του  $E_b/N_0$  με επαρκώς υψηλή τιμή του  $M$ , για τον υπολογισμό των διακριτών σημείων:



Παρακάτω, απεικονίζουμε μαζί τα διαγράμματα που προέκυψαν στα υποερωτήματα (i) και (ii):



Παρατηρούμε ότι η μορφή του διαγράμματος που μας δόθηκε, επαληθεύεται από τα παραπάνω διαγράμματα.

Ο κώδικας που χρησιμοποιήθηκε για την επαλήθευση του δοσμένου διαγράμματος, απεικονίζεται παρακάτω:

```

clc
close all;
clear all;

%Part 2 - Eirini Donti
%(a)
%(i)
k=2;%4-ASK %k=mod(19839,2)+3;
L=2^k; M=50000.0;
EbNo_dB = 0:0.1:20;
EbNo_linear = 10.^(EbNo_dB/10);
Pe = ((L-1)/L)*erfc(sqrt((3*log2(L)/(L^2-1)).*EbNo_linear));
BER4 = Pe./log2(L);
figure; S1=semilogy(EbNo_dB,BER4,'r'); hold on

k=3;%8-ASK
L=2^k; M=50000.0;
Pe = ((L-1)/L)*erfc(sqrt((3*log2(L)/(L^2-1)).*EbNo_linear));
BER8 = Pe./log2(L);
S2=semilogy(EbNo_dB,BER8,'g'); legend('8-ASK');

k=4;%16-ASK
L=2^k; M=50000.0;
Pe = ((L-1)/L)*erfc(sqrt((3*log2(L)/(L^2-1)).*EbNo_linear));
BER16 = Pe./log2(L);
S3=semilogy(EbNo_dB,BER16,'b'); legend('16-ASK');
title('Theoretical BER diagram (a),(i)');
xlabel('Es/No (dB)'); ylabel('BER'); grid on;
S=[S1;S2;S3];
legend(S,'4-ASK','8-ASK','16-ASK');
hold off;
%(ii)
Pe=[];%4-ASK
for EbNo = 1:20
    errors=ask_errors(2,50000,16,EbNo);
    Pe=[Pe; errors./50000];
end
BER=Pe./log2(L);
EbNo_dB= 1:20;
figure; s1=semilogy(EbNo_dB,BER,'ok'); hold on;

Pe=[];%8-ASK
for EbNo = 1:20
    errors=ask_errors(3,50000,16,EbNo);
    Pe=[Pe; errors./50000];
end
BER=Pe./log2(L);
EbNo_dB= 1:20;
s2=semilogy(EbNo_dB,BER,'+');

Pe=[];%16-ASK
for EbNo = 1:20
    errors=ask_errors(4,50000,16,EbNo);
    Pe=[Pe; errors./50000];
end
BER=Pe./log2(L);
EbNo_dB= 1:20;
s3=semilogy(EbNo_dB,BER,'x');
title('Theoretical BER diagram (a),(ii)');
xlabel('Es/No (dB)'); ylabel('BER'); grid on;
s=[s1;s2;s3];
legend(s,'4-ASK','8-ASK','16-ASK');
hold off;
% S=[S1;S2;S3;s1;s2;s3];
% legend(S,'4-ASK','8-ASK','16-ASK','4-ASK','8-ASK','16-ASK');

```

Επίσης, η συνάρτηση ask\_errors() η οποία χρησιμοποιείται στο παραπάνω πρόγραμμα, είναι η παρακάτω:

```
%3rd-part 2-(a)

function errors=ask_errors(k,Nsymb,nsamp,EbNo)
% Η συνάρτηση αυτή εξομοιώνει την παραγωγή και αποκωδικοποίηση
% θορυβώδους σήματος L-ASK και μετρά τον αριθμό των εσφαλμένων συμβόλων.
% Επιστρέφει τον αριθμό των εσφαλμένων συμβόλων (στη μεταβλητή errors).
% k είναι ο αριθμός των bits/σύμβολο, επομένως L=2^k -- ο αριθμός των
% διαφορετικών πλατών
% Nsymb είναι ο αριθμός των παραγόμενων συμβόλων (μήκος ακολουθίας LASK)
% nsamp ο αριθμός των δειγμάτων ανά σύμβολο (oversampling ratio)
% EbNo είναι ο ανηγμένος σηματοθορυβικός λόγος Eb/No, σε db
%
L=2^k;
SNR=EbNo-10*log10(nsamp/2/k); % SNR ανά δείγμα σήματος
% Διάνυσμα τυχαίων ακεραίων {±1, ±3, ... ±(L-1)}. Να επαληθευθεί
x=2*floor(L*rand(1,Nsymb))-L+1;
Px=(L^2-1)/3; % θεωρητική ισχύς σήματος
sum(x.^2)/length(x); % μετρούμενη ισχύς σήματος (για επαλήθευση)
y=rectpulse(x,nsamp);
n=wgn(1,length(y),10*log10(Px)-SNR);
ynoisyy=y+n; % θορυβώδες σήμα
y=reshape(ynoisyy,nsamp,length(ynoisyy)/nsamp);
matched=ones(1,nsamp);
z=matched*y/nsamp;
A=[-L+1:2:L-1];
for i=1:length(z)
[m,j]=min(abs(A-z(i)));
z(i)=A(j);
end
err=not(x==z);
errors=sum(err);
end
```

(β) Χρήση του εργαλείου BERTOOL του Matlab, πληκτρολογώντας την εντολή bertool στο παράθυρο εντολών του Matlab.

Πληκτρολογούμε την εντολή bertool στο Command Window του Matlab και εμφανίζεται ένα παράθυρο, στο οποίο ρυθμίζουμε τις παραμέτρους της προσομοίωσης. Παρακάτω, απεικονίζονται τα πεδία που συμπληρώσαμε για την εκτέλεση της προσομοίωσης:

Για την πειραματική προσομοίωση, συμπληρώσαμε τα πεδία, όπως φαίνεται παρακάτω:

Plot	BER Data Set	Eb/N0 (dB)	BER	# of Bits	Confidence Level	Fit	Run Time
<input checked="" type="checkbox"/>	16-ASK theoretical	0, 1, 2, 3, 4, 5, 6, 7, ...	0.25461, 0.23609, ...	N/A	N/A	N/A	N/A
<input checked="" type="checkbox"/>	8-ASK theoretical	0, 1, 2, 3, 4, 5, 6, 7, ...	0.19984, 0.17791, ...	N/A	N/A	N/A	N/A
<input checked="" type="checkbox"/>	4-ASK theoretical	0, 1, 2, 3, 4, 5, 6, 7, ...	0.14098, 0.119, 0....	N/A	N/A	N/A	N/A
<input checked="" type="checkbox"/>	8-ASK experimental	0, 2, 4, 6, 8, 10, 12, ...	0.17169, 0.14562, ...	150000, 150000,...	off	<input type="checkbox"/>	00:00:01
<input checked="" type="checkbox"/>	4-ASK experimental	0, 2, 4, 6, 8, 10, 12, ...	0.13865, 0.09782, ...	100000, 100000,...	off	<input type="checkbox"/>	00:01:21
<input checked="" type="checkbox"/>	16-ASK experimen...	0, 2, 4, 6, 8, 10, 12, ...	0.17817, 0.16506, ...	200000, 200000,...	off	<input type="checkbox"/>	00:00:02

Monte Carlo

Theoretical

Eb/N0 range:  dB

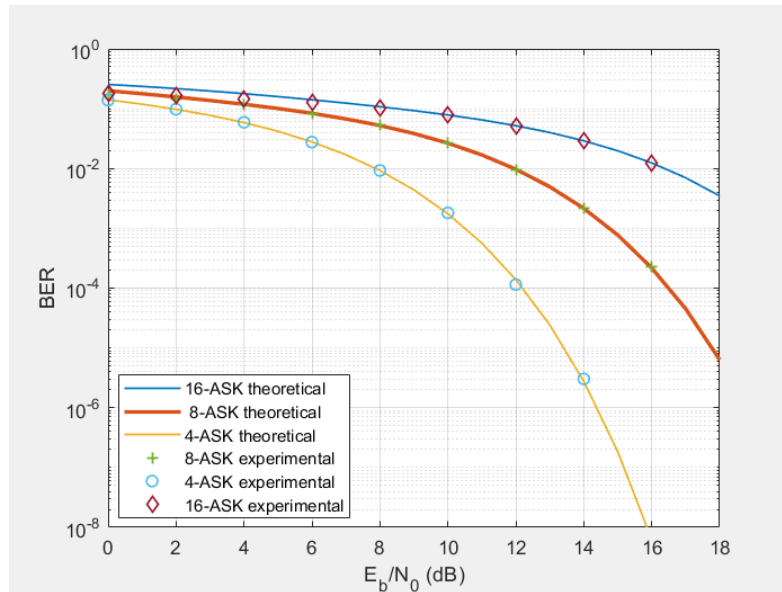
Simulation environment:  
☒ MATLAB  
☐ Simulink

Function name:

Simulation limits:  
Number of errors:   
or  
Number of bits:

Για τη θεωρητική προσομοίωση, μέσω bertool, συμπληρώνουμε τα πεδία με τύπο μορφοποίησης PAM, σηματοθορυβικό λόγο  $E_b/N_0=0:18$  και το εκάστοτε L- ASK.

Συνεπώς, προκύπτει η παρακάτω γραφική παράσταση για τη θεωρητική και την πειραματική προσέγγιση του διαγράμματος BER:



Παρατηρούμε ότι τα σημεία του διαγράμματος που προκύπτουν, επαληθεύουν τη δοσμένη μορφή της γραφικής παράστασης.

Η συνάρτηση `ask_ber_func()` που χρησιμοποιήθηκε για την παραπάνω προσομοίωση είναι η παρακάτω:

```
function [ber,numBits] = ask_ber_func(EbNo, maxNumErrs, maxNumBits,varargin)
% Import Java class for BERTool.
import com.mathworks.toolbox.comm.BERTool.*;
% Initialize variables related to exit criteria.
totErr = 0; % Number of errors observed
numBits = 0; % Number of bits processed
% A. --- Set up parameters. ---
% --- INSERT YOUR CODE HERE.
k=4; % number of bits per symbol
Nsymb=50000; % number of symbols in each run
nsamp=16; % oversampling, i.e. number of samples per T
% Simulate until number of errors exceeds maxNumErrs
% or number of bits processed exceeds maxNumBits.
while((totErr < maxNumErrs) && (numBits < maxNumBits))
% Check if the user clicked the Stop button of BERTool.
if (isBERToolSimulationStopped(varargin{:}))%BERTool.getSimulationStop
break;
end
% B. --- INSERT YOUR CODE HERE.
errors=ask_errors(k,Nsymb,nsamp,EbNo);
% Assume Gray coding: 1 symbol error ==> 1 bit error
totErr=totErr+errors;
numBits=numBits + k*Nsymb;
end % End of loop
% Compute the BER
ber = totErr/numBits;
```

### Μέρος 3<sup>ο</sup>: Υλοποίηση με συνέλιξη – Χρήση άλλων παλμών (α)

Ο τροποποιημένος κώδικας παράγει τα ίδια αποτελέσματα με τον αρχικό Κώδικα 3.3, καθώς:

Η εντολή 17 του αρχικού προγράμματος  $y=\text{rectpulse}(x,\text{nsamp})$ ;, εφαρμόζει ορθογώνια διαμόρφωση παλμού στο  $x$  για την παραγωγή ενός σήματος εξόδου με δείγματα  $\text{nsamp}$  ανά σύμβολο. Ο ορθογώνιος σχηματισμός παλμού σημαίνει ότι, κάθε σύμβολο από το  $x$ , επαναλαμβάνεται  $\text{nsamp}$  για να σχηματίσει την έξοδο  $y$ .

Παρόμοια, οι παρακάτω εντολές που αντικαθιστούν την παραπάνω εντολή:

```
h=ones(1,nsamp); h=h/sqrt(h*h'); y=upsample(x,nsamp); y=conv(y,h);  
y=y(1:M*nsamp);
```

δημιουργούν τον ορθογωνικό παλμό μοναδιαίας ενέργειας ως την κρουστική απόκριση φίλτρου πομπού. Μετατρέπεται το προς εκπομπή σήμα σε πυκνό πλέγμα και περικόπτεται η ουρά που αφήνει η συνέλιξη. Οπότε, εφαρμόζεται ορθογώνια διαμόρφωση παλμού στο  $x$  για την παραγωγή ενός σήματος εξόδου με δείγματα  $\text{nsamp}$  ανά σύμβολο, όπως εκτελείται στην αρχική εντολή 17.

Επίσης, το θορυβώδες σήμα που ορίζεται αρχικά  $\text{ynois}y=y+n$ ;, το οποίο προσθέτει θόρυβο στο αρχικό σήμα, αντικαθίσταται με την εντολή  $\text{ynois}y=\text{awgn}(y,\text{SNR},\text{'measured'})$ ;, η οποία προσθέτει λευκό Gaussian θόρυβο στο σήμα  $y$ . Οπότε, οι δύο παραπάνω εντολές είναι παρόμοιες.

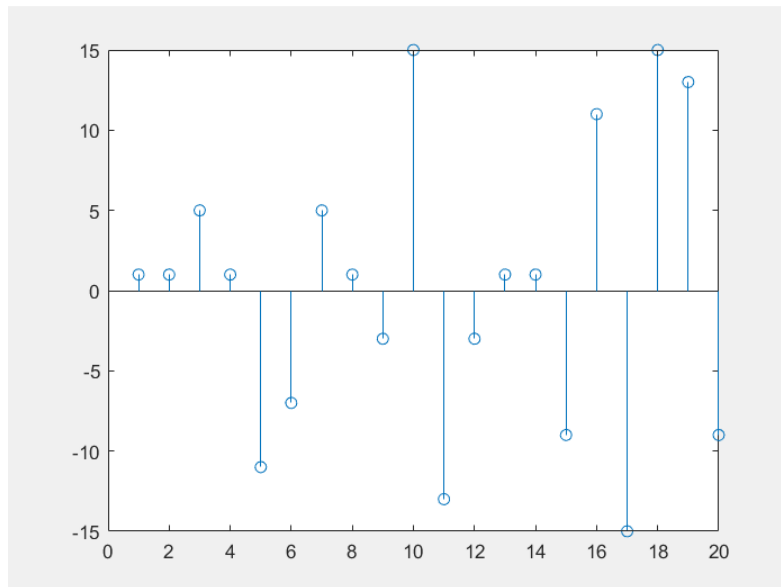
Επίσης, οι αρχικές εντολές  $\text{matched}=\text{ones}(1,\text{nsamp})$ ;  $z=\text{matched}*y/\text{nsamp}$ ; βοηθούν στην έκφραση της απόκρισης του προσαρμοσμένου φίλτρου για όλους τους παλμούς, αφότου γίνει ο ανασχηματισμός του σήματος  $y$ . Οι παραπάνω εντολές αντικαθίστανται με τις εντολές `for i=1:nsamp matched(i)=h(end-i+1); end`  $\text{yrx}=\text{conv}(\text{ynois}y,\text{matched})$ ;  $z = \text{yrx}(\text{nsamp}:\text{nsamp}:M*\text{nsamp})$ ; οι οποίες υποδειγματίζουν το σήμα στο τέλος κάθε περιόδου  $T$ . Οπότε, οι παραπάνω εντολές είναι ισοδύναμες.

Γενικά, οι εντολές που αντικατέστησαν τις αρχικές εντολές είναι ισοδύναμες. Οπότε, ο τροποποιημένος κώδικας παράγει τα ίδια αποτελέσματα με τον Κώδικα 3.3.

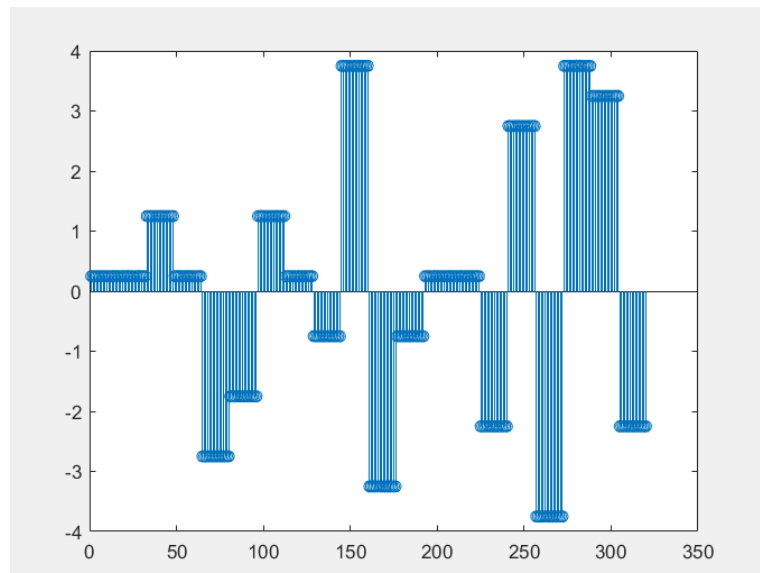
(β)

Εκτελούμε το σώμα της τροποποιημένης συνάρτησης `ask_errors()` χωρίς την προσθήκη θορύβου. Σχεδιάζουμε τα ζητούμενα τμήματα σημάτων, όπως φαίνονται παρακάτω:

Παρακάτω, παρουσιάζεται το τμήμα σημάτων  $x$  όπως ζητήθηκε:



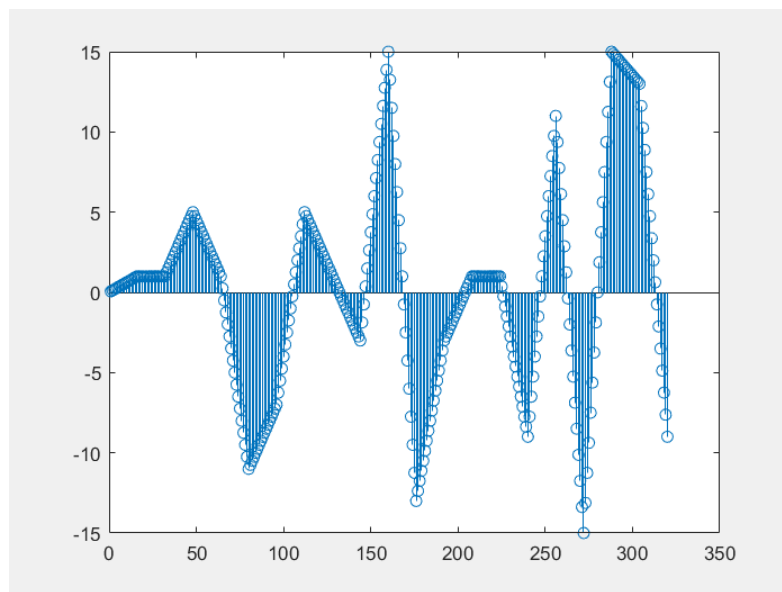
Παρακάτω, παρουσιάζεται το τμήμα σημάτων  $y$  όπως ζητήθηκε:



Παρατηρούμε ότι, για κάθε παλμό, δημιουργούνται `nsamp` σημεία στη γραφική παράσταση της μεταβλητή  $x$  με πλάτος, το πλάτος του εκάστοτε παλμού. Με άλλα λόγια, μεταδίδονται σε βασική ζώνη ορθογωνικοί παλμοί, χωρίς να έχουν υποστεί διαμόρφωση.



Παρακάτω, παρουσιάζεται το τμήμα σημάτων  $y[n]$  όπως ζητήθηκε:



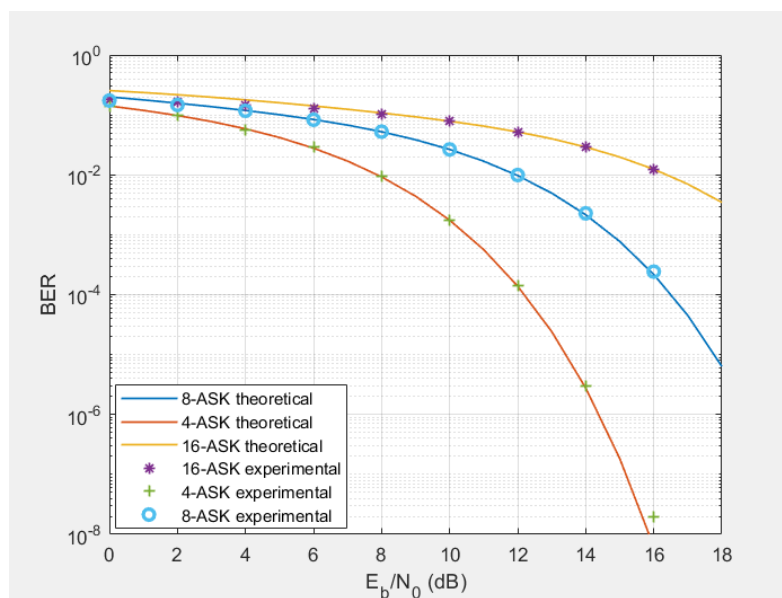
Η παραπάνω γραφική παράσταση, περιέχει θόρυβο προστιθέμενο στο σήμα της προηγούμενης γραφικής παράστασης. Όσο μικρότερος είναι ο σηματοθορυβικός λόγος, τόσο μεγαλύτερη θα είναι η αλλοίωση της γραφικής παράστασης.

(γ)

Αντικαθιστούμε τον ορθογωνικό παλμό  $h$  με τις εξής εντολές:

$h = \cos(2 \cdot \pi \cdot (1:nsamp)/nsamp)$ ;  $h = h/\sqrt{h \cdot h'}$ ; Εκτελούμε πάλι την εξομοίωση με την εντολή `bertool` στο Command Window.

Συμπεραίνουμε ότι, η επίδοση του νέου συστήματος L-ASK, είναι περίπου η ίδια όσο στα προηγούμενα υποερωτήματα. Καταρχάς, η προσομοίωση χρειάστηκε αρκετό χρόνο για να εξάγει αποτέλεσμα. Επίσης, το παραπάνω επαληθεύεται με τη μορφή του διαγράμματος που προκύπτει, όπως φαίνεται παρακάτω:



Ο κώδικας που χρησιμοποιήθηκε για το παραπάνω υποερώτημα, είναι ο παρακάτω:

```
%3rd-part 3
function errors=ask_errors(k,Nsymb,nsamp,EbNo)
% Η συνάρτηση αυτή εξομοιώνει την παραγωγή και αποκωδικοποίηση
% θορυβώδους σήματος L-ASK και μετρά τον αριθμό των εσφαλμένων συμβόλων.
% Υπολογίζει επίσης τη θεωρητική πιθανότητα εσφαλμένου συμβόλου, Pe.
% Επιστρέφει τον αριθμό των εσφαλμένων συμβόλων, καθώς και τον συνολικό
% αριθμό των συμβόλων που παρήχθησαν.
% k είναι ο αριθμός των bits/σύμβολο, ώστε  $L=2^k$ ,
% M ο αριθμός των παραγόμενων συμβόλων (μήκος ακολουθίας L-ASK)
% nsamp ο αριθμός των δειγμάτων ανά σύμβολο (oversampling ratio)
% EbNo είναι ο λόγος Eb/No, σε db
%k = mod(19839,2)+3; L=2^k; EbNo=12; nsamp=16; M=50000;
%k=3; L=2^k; nsamp=32; M=50000;
L=2^k; M=50000;
SNR=EbNo-10*log10(nsamp/2/k); % SNR ανά δείγμα σήματος
% Διάνυσμα τυχαίων ακεραίων {±1, ±3, ... ±(L-1)}. Να επαληθευτεί
x=2*floor(L*rand(1,M))-L+1;
Px=(L^2-1)/3; % θεωρητική ισχύς σήματος
sum(x.^2)/length(x); % μετρούμενη ισχύς σήματος (για επαλήθευση)
%y=rectpulse(x,nsamp);
h=cos(2*pi*(1:nsamp)/nsamp); h=h/sqrt(h*h');%h=ones(1,nsamp); h=h/sqrt(h*h'); % κρουστική απόκριση φίλτρου %Part 3
% πομπού (ορθογωνικός παλμός μοναδιαίας ενέργειας)
y=upsample(x,nsamp); % μετατροπή στο πυκνό πλέγμα
y=conv(y,h); % το προς εκπομπή σήμα
y=y(1:M*nsamp); % περικόπτεται η ουρά που αφήνει η συνέλιξη

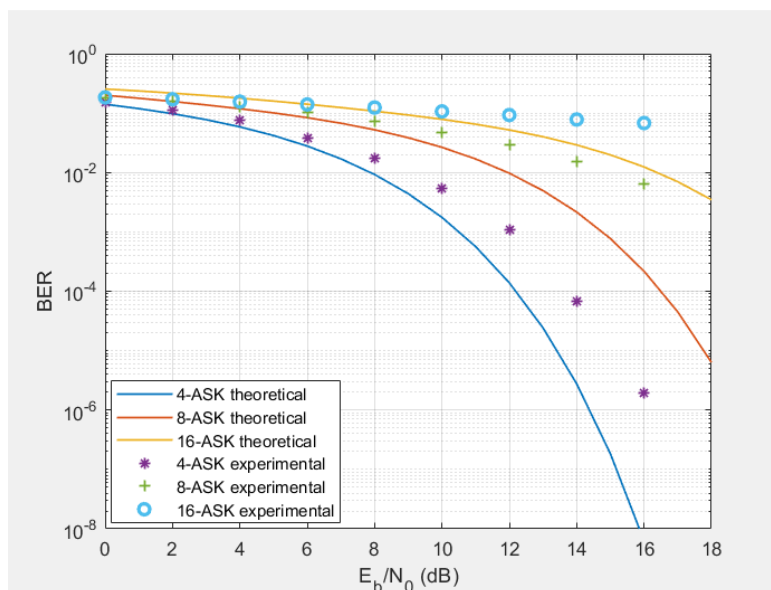
n=wgn(1,length(y),10*log10(Px)-SNR);

ynoisy=awgn(y,SNR,'measured'); % θορυβώδες σήμα %ynois=y+n; % θορυβώδες σήμα %ynois=y;
%matched=h;
%y=reshape(ynois,nsamp,length(ynois)/nsamp);
%matched=ones(1,nsamp);
%z=matched*y/nsamp;
for i=1:nsamp matched(i)=h(end-i+1); end
yrx=conv(ynois,matched);
z = yrx(nsamp:nsamp:M*nsamp); % Υποδειγμάτιση -- στο τέλος

% κάθε περίοδο T
l=[-L+1:2:L-1];
for i=1:length(z)
[m,j]=min(abs(l-z(i)));
z(i)=l(j);
end
err=not(x==z);
errors=sum(err);

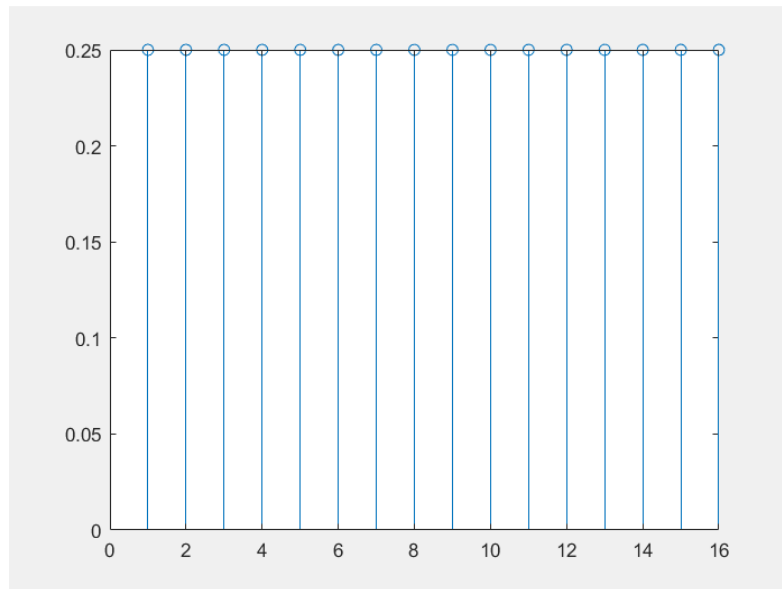
%figure; stem(x(1:20));
%figure; stem(y(1:20*nsamp));
%figure; stem(yrx(1:20*nsamp));
%figure; stem(h);
end
```

Επαναφέρουμε την εντολή `matched=h` και παρατηρούμε ότι δεν παίρνουμε σωστά αποτελέσματα, για τιμή του `nsamp=16`.

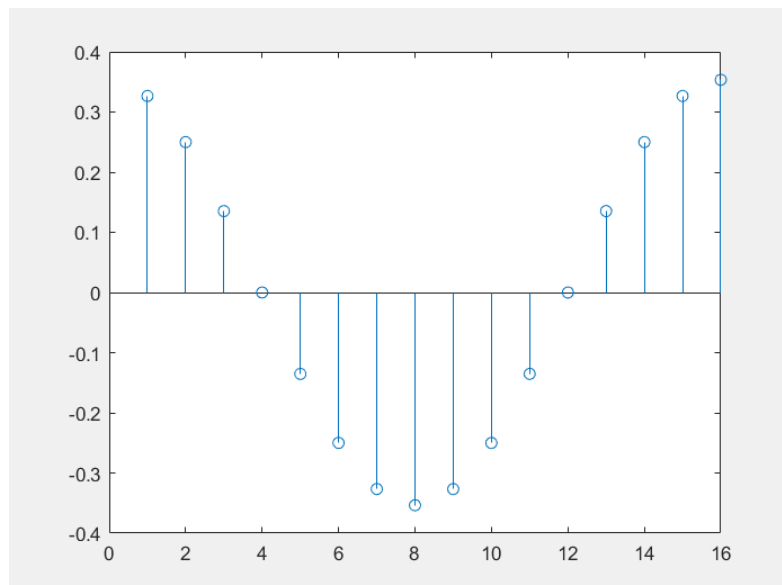


Παρουσιάζουμε τη μορφή του παλμού  $h$  στις δύο περιπτώσεις ορθογωνικού και ημιτονικού παλμού, σχεδιάζοντας με `stem(h)`.

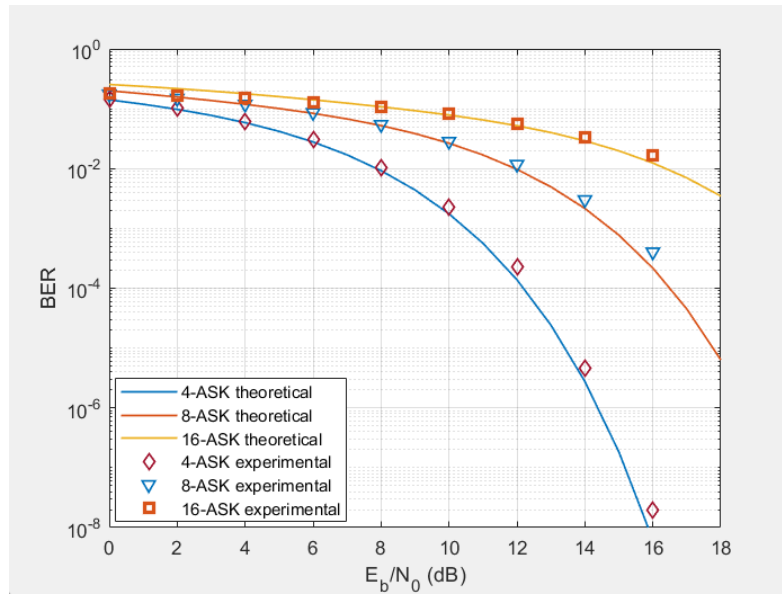
Θεωρούμε ορθογωνικό παλμό  $h$ :



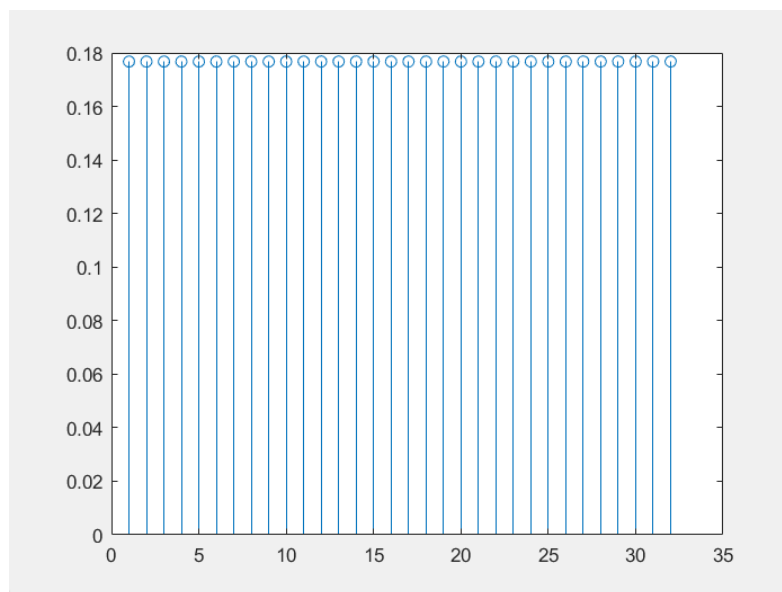
Θεωρούμε ημιτονικό παλμό  $h$ :



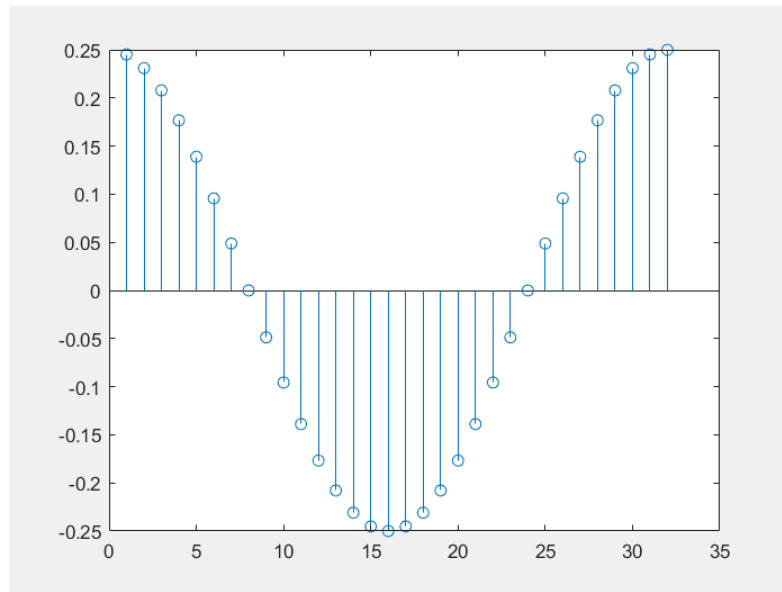
Επαναλαμβάνουμε το πείραμα για  $n_{\text{samp}}=32$ , οπότε προκύπτει το παρακάτω διάγραμμα:



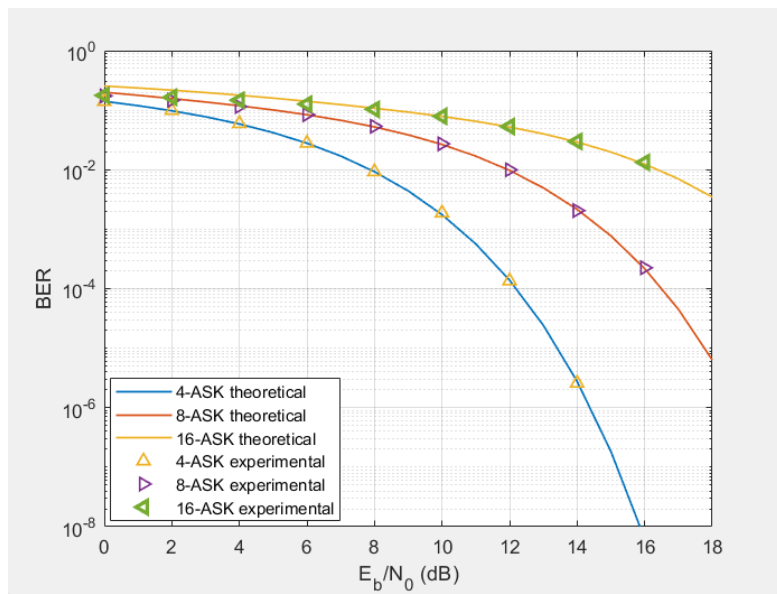
Θεωρούμε ορθογωνικό παλμό  $h$ :



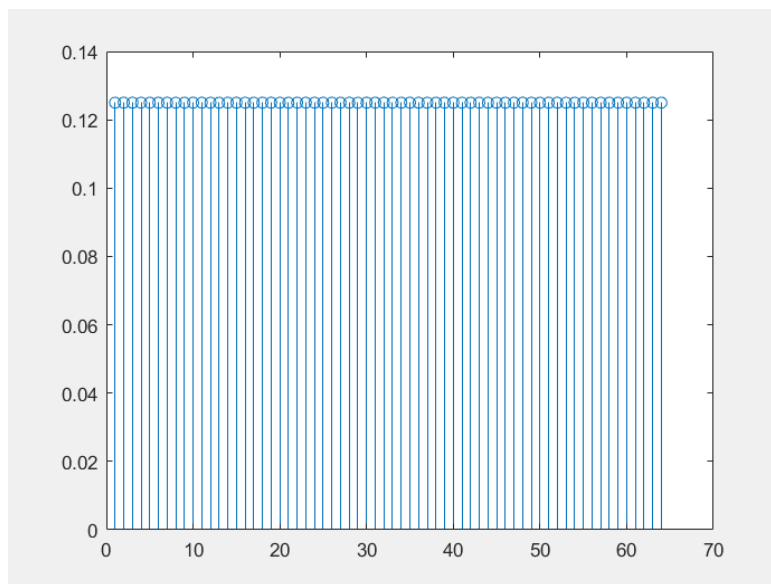
Θεωρούμε ημιτονικό παλμό  $h$ :



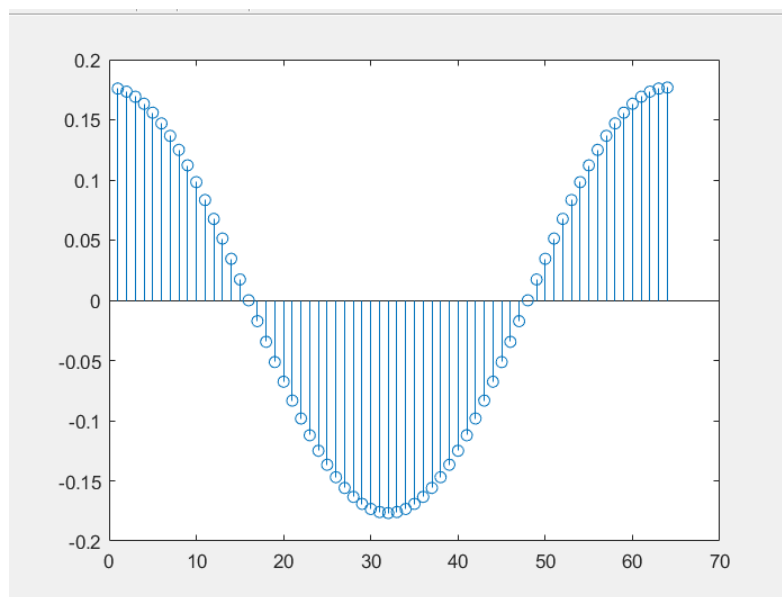
Επαναλαμβάνουμε το πείραμα για  $n_{\text{samp}}=64$ , οπότε προκύπτει το παρακάτω διάγραμμα:



Θεωρούμε ορθογωνικό παλμό  $h$ :



Θεωρούμε ημιτονικό παλμό  $h$ :



Παρατηρούμε ότι, η μορφή του παλμού μεταβάλλεται ανάλογα με τις τιμές της μεταβλητής  $nsamp$ . Αυξάνοντας την τιμή του  $nsamp$ , τα διαστήματα μεταξύ των μίσχων της γραφικής του παλμού  $h$  πλησιάζουν περισσότερο το ένα με το άλλο. Η μορφή, όμως, παραμένει ίδια σε όλες τις περιπτώσεις. Οπότε, τα διαγράμματα BER,

μέσω της προσομοίωσης, πλησιάζουν περισσότερο στις εκάστοτε θεωρητικές γραφικές παραστάσεις, καθώς αυξάνεται η τιμή της μεταβλητής nsamp.

Ο κώδικας που χρησιμοποιήθηκε για το παραπάνω υποερώτημα, είναι ο παρακάτω:

```
%3rd-part 3
function errors=ask_errors(k,Nsymb,nsamp,EbNo)
% Η συνάρτηση αυτή εξομοιώνει την παραγωγή και αποκωδικοποίηση
% θορυβώδους σήματος L-ASK και μετρά τον αριθμό των εσφαλμένων συμβόλων.
% Υπολογίζει επίσης τη θεωρητική πιθανότητα εσφαλμένου συμβόλου, Pe.
% Επιστρέφει τον αριθμό των εσφαλμένων συμβόλων, καθώς και τον συνολικό
% αριθμό των συμβόλων που παρήχθησαν.
% k είναι ο αριθμός των bits/σύμβολο, ώστε L=2^k,
% M ο αριθμός των παραγόμενων συμβόλων (μήκος ακολουθίας L-ASK)
% nsamp ο αριθμός των δειγμάτων ανά σύμβολο (oversampling ratio)
% EbNo είναι ο λόγος Eb/No, σε db
%k = mod(19839,2)+3; L=2^k; EbNo=12; nsamp=16; M=50000;
%k=3; L=2^k; nsamp=32; M=50000;
L=2^k; M=50000;
SNR=EbNo-10*log10(nsamp/2/k); % SNR ανά δείγμα σήματος
% Διάνυσμα τυχαίων ακεραίων {±1, ±3, ... ±(L-1)}. Να επαληθευτεί
x=2*floor(L*rand(1,M))-L+1;
Px=(L^2-1)/3; % θεωρητική ισχύς σήματος
sum(x.^2)/length(x); % μετρούμενη ισχύς σήματος (για επαλήθευση)
%y=rectpulse(x,nsamp);
h=cos(2*pi*(1:nsamp)/nsamp); h=h/sqrt(h*h');%h=ones(1,nsamp); h=h/sqrt(h*h'); % κρουστική απόκριση φίλτρου %Part 3
% πομπού (ορθογωνικός παλμός μοναδιαίας ενέργειας)
y=upsample(x,nsamp); % μετατροπή στο πυκνό πλέγμα
y=conv(y,h); % το προς εκπομπή σήμα
y=y(1:M*nsamp); % περικρίπτεται η ουρά που αφήνει η συνέλιξη

n=wgn(1,length(y),10*log10(Px)-SNR);

ynoisy=awgn(y,SNR,'measured'); % θορυβώδες σήμα %ynoisy=y+n; % θορυβώδες σήμα %ynoisy=y;
matched=h;
%y=reshape(ynoisy,nsamp,length(ynoisy)/nsamp);
%matched=ones(1,nsamp);
%z=matched*y/nsamp;
%for i=1:nsamp matched(i)=h(end-i+1); end
yrx=conv(ynoisy,matched);
z = yrx(nsamp:nsamp:M*nsamp); % Υποδειγμάτιση -- στο τέλος

% κάθε περιόδου T
l=[-L+1:2:L-1];
for i=1:length(z)
[m,j]=min(abs(l-z(i)));
z(i)=l(j);
end
err=not(x==z);
errors=sum(err);

%figure; stem(x(1:20));
%figure; stem(y(1:20*nsamp));
%figure; stem(yrx(1:20*nsamp));
%figure; stem(h);
end
```

Για την υλοποίηση της προσομοίωσης BERTOOL, χρησιμοποιήσαμε την αρχική συνάρτηση ask\_ber\_func().