ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Αναφορά 3ης Εργαστηριακής Άσκησης στις Ψηφιακές Επικοινωνίες Ι

«Προσαρμοσμένα φίλτρα και L-ASK»

Μπουφίδης Ιωάννης 03120162

Μέρος 1 : Διερεύνηση του κώδικα εξομοίωσης

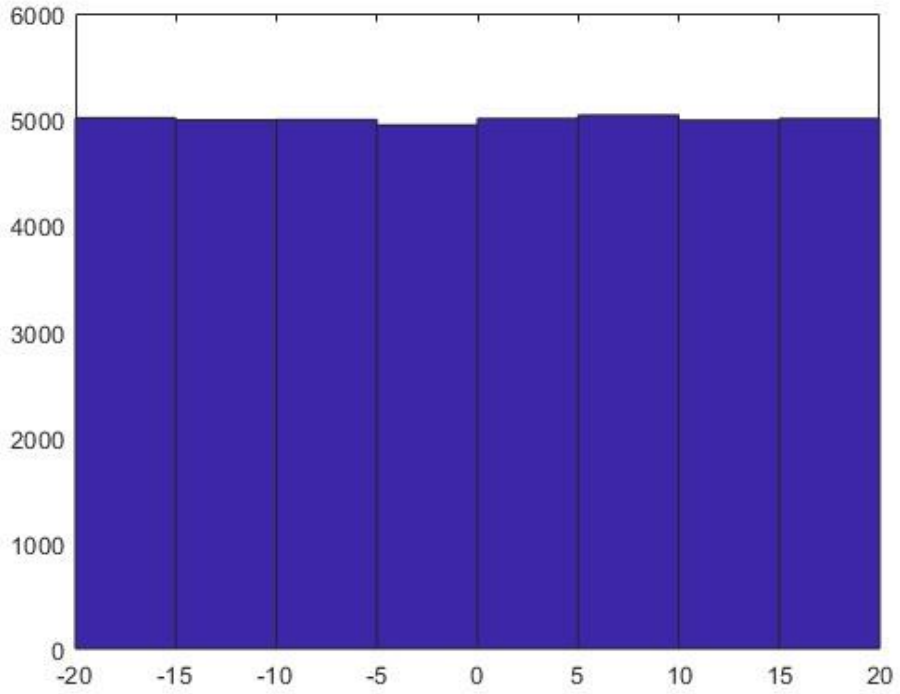
α) Παρακάτω ακολουθεί ο τροποποιήμενος Κώδικας 3.3, έτσι ώστε τα L στοιχεία του διανύσματος x να λαμβάνουν τιμές από το σύνολο {±d/2, ±3d/2, ±5d/2… }, όπου η απόσταση d των σημείων δίνεται ως παράμετρος και ισούται με 5. Τέλος, χρησιμοποιείται η τιμή του k=mod(20162,2)+3=3 και παράγεται ένα διάνυσμα 40000 στοιχείων.

Κατάλληλα, επίσης, τροποποιήθηκε τόσο το διάνυσμα A των L διαφορετικών στάθμεων, όσο και ο τύπος του θεωρητικού υπολογισμού ισχύος (κάτι που επιβεβαιώνει το παρακάτω μήνυμα).

```
Theoritically: 1.312500e+02
Calculated: 1.318925e+02
```

```
close all; clear all; clc;
k=mod(20162,2)+3; Nsymb=40000; EbNo = 16; nsamp = 20;
d=5;

L=2^k;
SNR=EbNo-10*log10(nsamp/2/k); % SNR ανά δείγμα σήματος

% Διάνυσμα τυχαίων ακεραίων {±d/2, ±3(d/2), ... ±(L-1)(d/2)}. Να επαληθευθεί
x=(2*floor(L*rand(1,Nsymb))-L+1)*(d/2); % παραγωγή του L-ASK σήματος
Px=((d^2)/4)*(L^2-1)/3; % θεωρητική ισχύς σήματος
Px2=sum(x.^2)/length(x); % μετρούμενη ισχύς σήματος (για επαλήθευση)
y=rectpulse(x,nsamp); % κρουστική απόκριση φίλτρου
n=wgn(1,length(y),10*log10(Px)-SNR); % παραγωγή θορύβου
ynoisy=y+n; % θορυβώδες σήμα
y=reshape(ynoisy,nsamp,length(ynoisy)/nsamp); % από εδώ
matched=ones(1,nsamp);
z=matched*y/nsamp; % μέχρι εδώ εφαρμογή προσαρμοσμένου φίλτρου
A=(-d*(L-1)/2):d:(d*(L-1)/2); % σύνολο στάθμεων

for i=1:length(z) % ανιχνευτής συμβόλου
    [m,j]=min(abs(A-z(i)));
    z(i)=A(j);
end

err=not(x==z); % αν το ανιχνευμένο δεν ταυτίζεται με το πραγματικό, τότε 1
errors=sum(err);

fprintf('Theoretically: %d\n', Px);
fprintf('Calculated: %d\n', Px2);
```

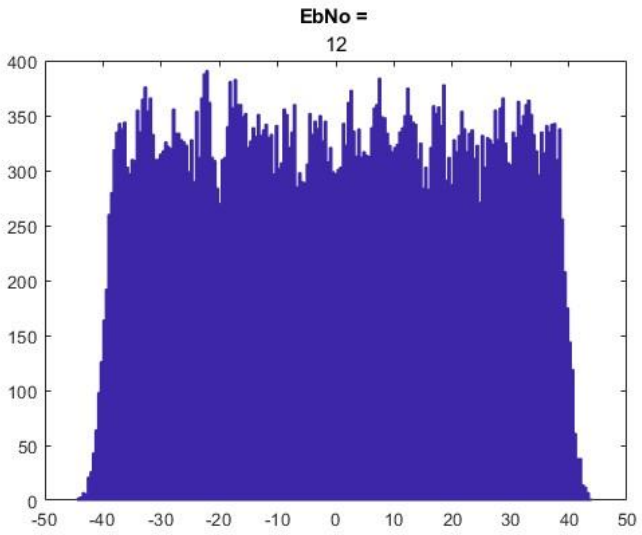
Τυπώνοντας το παρακάτω ιστόγραμμα με τις εντολές «figure(1); hist(x,A); pause»
μπορεί να παρατηρήσει κανείς ότι τα στοιχεία του x ακολουθουν όντως την ομοιόμορφη κατανομή.

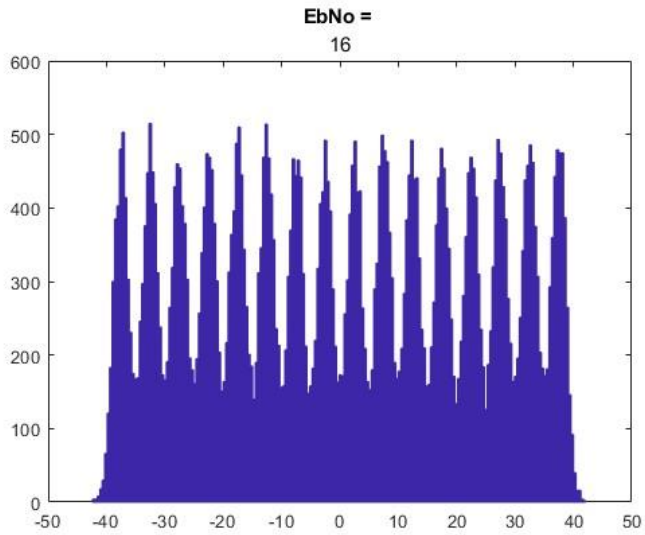


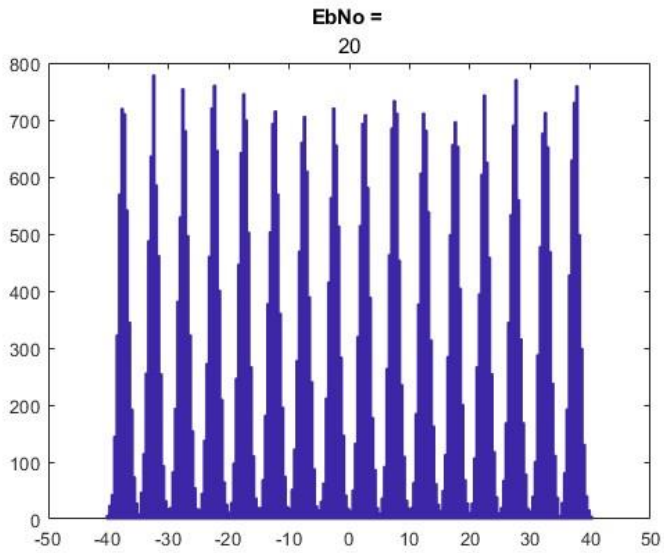
β) Χρησιμοποιώντας τον παρακάτω κώδικα σχεδιάζουμε το ιστόγραμμα του z για 3 διαφορετικές τιμές του EbNo (12, 16, 20) και για k=4, Nsymb=60000, nsamp=20

```
%% Question 1.B
clear all; close all; clc;

d=5; k=4; Nsymb=60000; nsamp=20; EbNo=[12,16,20];
L=2^k;

% Διάνυσμα τυχαίων ακεραίων {±d/2, ±3(d/2), ... ±(L-1)(d/2)}. Να επαληθευθεί
x=(2*floor(L*rand(1,Nsymb))-L+1)*(d/2);

Px=((d^2)/4)*(L^2-1)/3; % θεωρητική ισχύς σήματος
Px2=sum(x.^2)/length(x); % μετρούμενη ισχύς σήματος (για επαλήθευση)
y=rectpulse(x,nsamp); % παραγόμενο από τον πομπό σήμα (τετραγωνικός παλμός)

for i=1:3
    SNR=EbNo(i)-10*log10(nsamp/2/k); % SNR ανά δείγμα σήματος

    n=wgn(1,length(y),10*log10(Px)-SNR);
    ynoisy=y+n; % θορυβώδες σήμα
    y1=reshape(ynoisy,nsamp,length(ynoisy)/nsamp);
    matched=ones(1,nsamp);
    z=matched*y1/nsamp;

    figure(i+1);
    hist(z,200);
    title('EbNo = ', EbNo(i)); pause
end;
```

Τα ιστόγραμματα του z που παράχθηκαν είναι τα εξής:







Παρατηρούμε ότι, όσο αυξάνεται ο σηματοθορυβικός λόγος EbNo, τόσο περισσότερο στενεύουν οι «κορυφές» γύρω από τις L τιμές στάθμεων {±d/2, ±3d/2, ±5d/2... }, όπου d=5. Αυτό συμβαίνει, καθώς από την στιγμή που ο λόγος σήματος προς θόρυβο αυξάνεται, λιγότερα σύμβολα θα βρεθούν ενδιάμεσα σε 2 στάθμες, ανάμεσα σε αυτή που ανήκουν και στην επόμενη ή προηγούμενη της. Με λίγα λόγια, όσο αυξάνεται το EbNo, τόσο περισσότερα σύμβολα θα ανγνωρίζονται σωστά απ' το προσαρμοσμένο φίλτρο του δέκτη.

γ) Οι εντολές 20-22, οι οποίες παρουσιάζονται παρακάτω, αντιπροσωπεύουν την εφαρμογή του προσαρμοσμένου φίλτρου

```
y=reshape(ynoisy,nsamp,length(ynoisy)/nsamp); % από εδώ
matched=ones(1,nsamp);
z=matched*y/nsamp; % μέχρι εδώ εφαρμογή προσαρμοσμένου φίλτρου
```

Επιπλέον, ο πίνακας matched (κρουστική απόκριση του φίλτρου) είναι διαστάσεων **1 x nsamp=20**, ο x (παραγόμενο σήμα L-ASK) **1 x Nsymb = 40000**, ο y (σήμα που εκπέμπει ο πομπός) **nsamp = 20 x Nsymb = 40000**, ενώ το z (σήμα που λαμβάνει ο δέκτης αφού πέρασε από φίλτρο) **1 x Nsymb = 40000**.

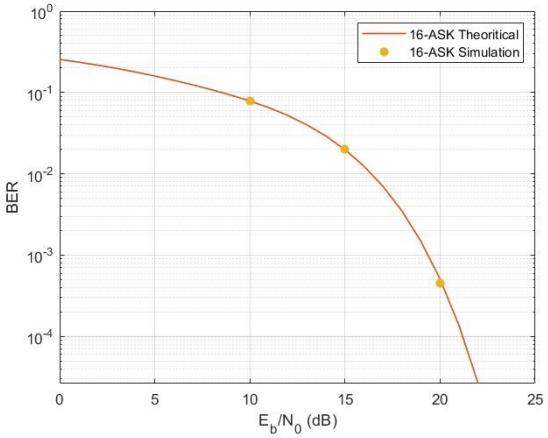
δ) Ο διπλανός βρόχος λειτουργεί ως ανιχνευτής ελάχιστης απόστασης για την L-ASK. Συγκεκριμένα, για καθένα από τα σύμβολα του διανύσματος z, δηλαδή του σήματος που δέχθηκε ο δέκτης, ελέγχουμε σε ποια από τις στάθμες του διανύσματος A βρίσκεται πιο κόντα και θέτουμε την τιμή του ίση με την τιμή της στάθμης.

```
for i=1:length(z) % ανιχνευτής συμβόλου
    [m,j]=min(abs(A-z(i)));
    z(i)=A(j);
end
```

<u>Μέρος 2</u> : Καμπύλες επίδοσης (BER συναρτήσει του σηματοθορυβικού λόγου)

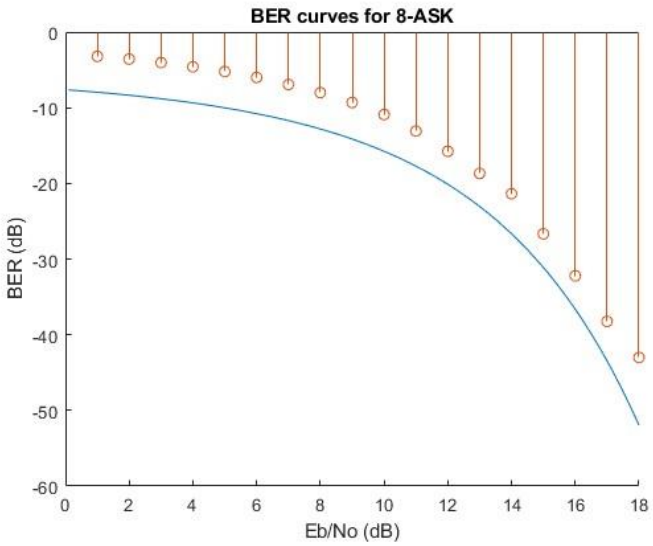
Διαβάζοντας την διπλανή καμπύλη 16-ASK, η οποία προέκυψε από το bertool, οι τιμές BER για Eb/No = {10,15,20} db είναι οι εξής:

- Για τα 10db: BER = 0.078344
- Για τα 15db: BER= 0.020156
- Για τα 20db: BER= 0.0004543

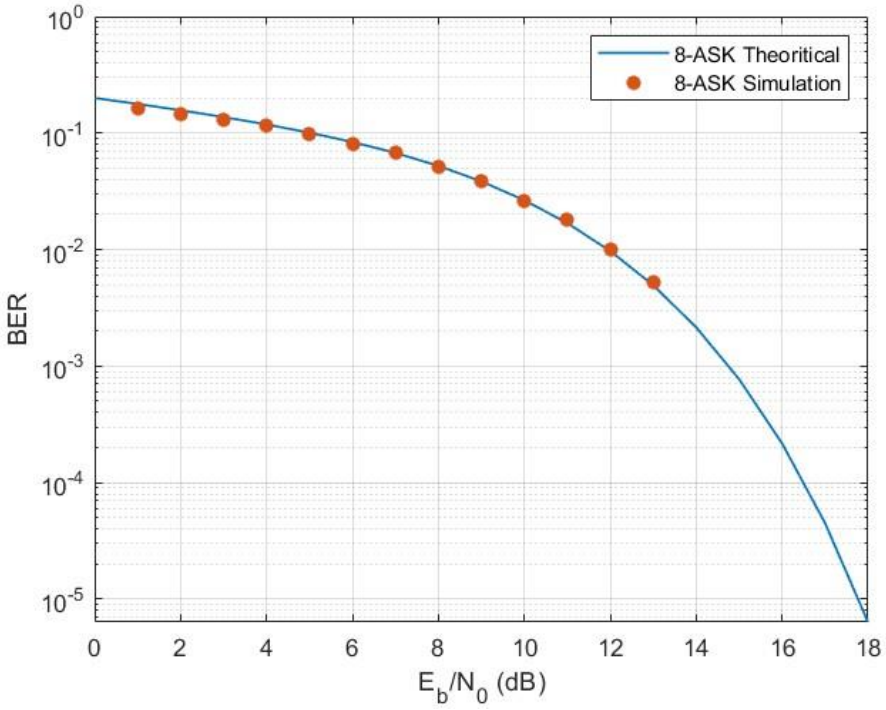


α) Με την εκτέλεση του παρακάτω κώδικα, παράγεται το σχήμα που ακολουθεί για την L-ASK με $L=2^k$, k=mod(20162,2)+3. Το σχήμα αποτελείται από την θεωρητική καμπύλη σε συνδυασμό με τα αποτελέσματα της εξοιμοίωσης για διακριτές τιμές του EbNo.

```
%% Σχεδιασμός καμπύλης
clear all; close all; clc;

k = mod(20162,2)+3;
L = 2^k;
EbNo = 0.1:0.1:18;
EbNo_d = 1:18;
figure; hold on;

Pe = ((L-1)/L)*erfc(sqrt(((3*log2(L))/((L^2)-
1))*10.^(0.1*EbNo)));
BER = Pe/log2(L);

plot(EbNo, 10*log10(BER)); % switch axes to
log scale

for i=1:18
  ber(i) = ask_errors(3,20000,20,i)/20000;
end
stem(EbNo_d, 10*log10(ber));

xlabel('Eb/No (dB)');
ylabel('BER (dB)');
title('BER curves for 8-ASK');
```



β) Από την άλλη, με τη χρήση του εργαλείου BERTOOL του MATLAB, προκύπτει το διάγραμμα BER για 8-ASK που ακολουθεί και το οποίο φαίνεται να επαληθεύει τόσο την δεδομένη καμπύλη, όσο και αυτή του ερωτήματος **(α)**. Συγκεκριμένα, για τον υπολογισμό των διακριτών σημείων, το BERTOOL καλεί την συνάρτηση ask_ber_func (κώδικας 3.4), η οποία χρησιμοποιεί την ask_errors, αφού έχουν τεθεί οι κατάλληλες τιμές (εδώ k=3, Nsymb=20000, nsamp=20), για να υπολογίσει το κάθε αποτέλεσμα.



<u>Μέρος 3</u> : Υλοποίηση με συνέλιξη - Χρήση άλλων παλμών

α) Αφού ακολουθήσαμε τα βήματα που περιγράφονται στην εκφώνηση και αντικαταστήσαμε τις εντολές με τις αντίστοιχες τους, προκύπτει ο παρακάτω κώδικας

```
function errors=ask_errors_conv(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
Px2=sum(x.^2)/length(x); % μετρούμενη ισχύς σήματος (για επαλήθευση)
h=ones(1,nsamp); h=h/sqrt(h*h'); % κρουστική απόκριση φίλτρου
% πομπού (ορθογωνικός παλμός μοναδιαίας ενέργειας)
y=upsample(x,nsamp); % μετατροπή στο πυκνό πλέγμα
y=conv(y,h); % το προς εκπομπή σήμα
y=y(1:Nsymb*nsamp); % περικόπτεται η ουρά που αφήνει η συνέλιξη
ynoisy=awgn(y,SNR,'measured'); % θορυβώδες σήμα
for i=1:nsamp
    matched(i)=h(end-i+1);
end
yrx=conv(ynoisy,matched);
z = yrx(nsamp:nsamp:Nsymb*nsamp); % Υποδειγμάτιση -- στο τέλος κάθε περιόδου T
A=[-L+1:2:L-1];
for i=1:length(z)
```

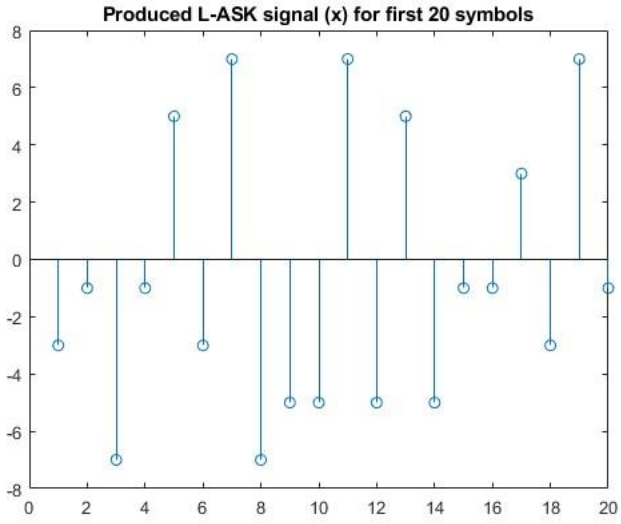
```
    [m,j]=min(abs(A-z(i)));
    z(i)=A(j);
end
err=not(x==z);
errors=sum(err);
end
```

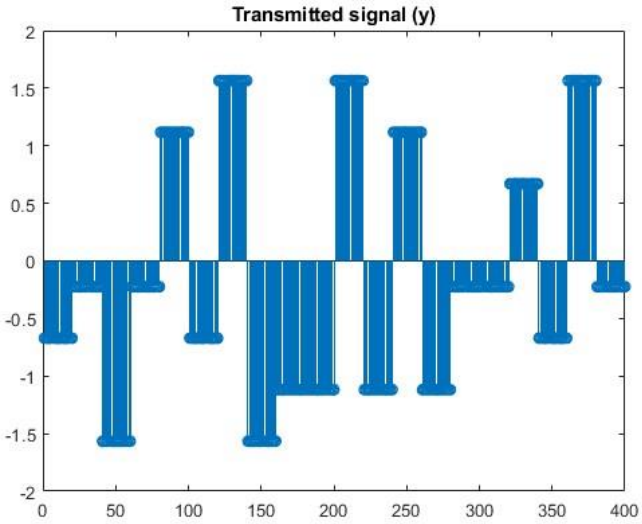
Αν εκτελέσουμε το κομμάτι κώδικα που ακολουθεί, επαληθεύεται μέσω του μηνύματος ότι ο τροποποιημένος κώδικας παράγει τα ίδια αποτελέσματα με τον αρχικό

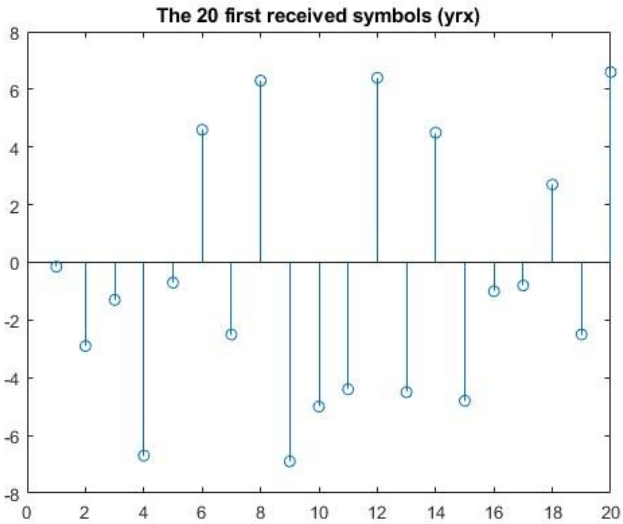
```
k=mod(20162,2)+3; Nsymb=20000; nsamp=20; EbNo=12;
errors1 = ask_errors(k,Nsymb,nsamp,EbNo);
errors2 = ask_errors_conv(k,Nsymb,nsamp,EbNo);
fprintf('Original: %d\n', errors1);
fprintf('Modified: %d\n', errors2);
```

```
Original: 597
Modified: 523
```

β) Εκτελούμε το σώμα της ask_errors_conv αυτή την φορά χωρίς προσθήκη θορύβου (δηλαδή ynoisy=y). Αν σχεδιάσουμε τα σήματα x, y και yrx για τα πρώτα 20 σύμβολα, προκύπτουν τα παρακάτω σχήματα





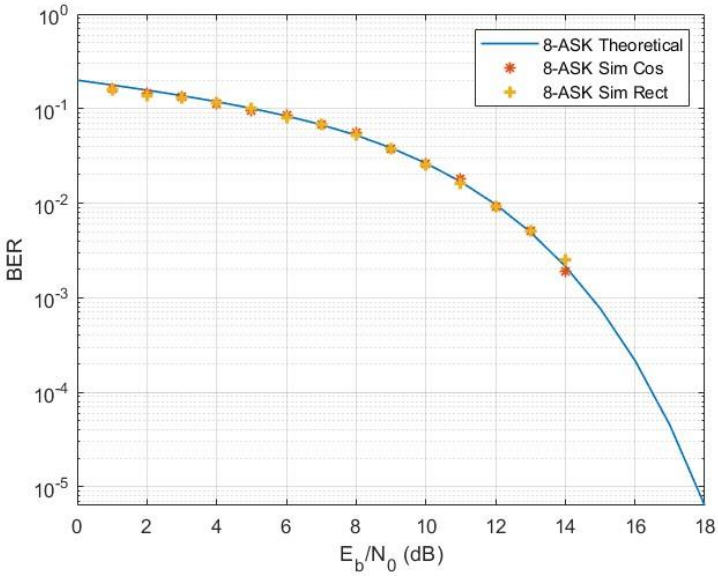


<u>Κώδικας:</u>

```
figure(1); stem(x(1:20));
title('Produced L-ASK signal (x) for first 20
symbols'); pause;
figure(2); stem(y(1:20*nsamp));
title('Transmitted signal (y)'); pause;
figure(3); stem(yrx(1:nsamp:20*nsamp));
title('The 20 first received symbols (yrx)');
pause;
```

Παρατηρούμε ότι, εφόσον στο σήμα που μεταδίδεται δεν εφαρμόστηκε θόρυβος, το αρχικό σήμα x και το σήμα yrx που «διάβασε» ο δέκτης ταυτίζονται. Το σήμα y αποτελεί το σήμα που εκπέμπει ο πομπός και δημιουργήθηκε αφού κάναμε upsample με nsamp=20 το αρχικό σήμα x. Ενώ, το σήμα yrx προκύπτει όταν, από κάθε σύμβολο του y, κρατήσουμε μόνο τον πρώτο μίσχο.

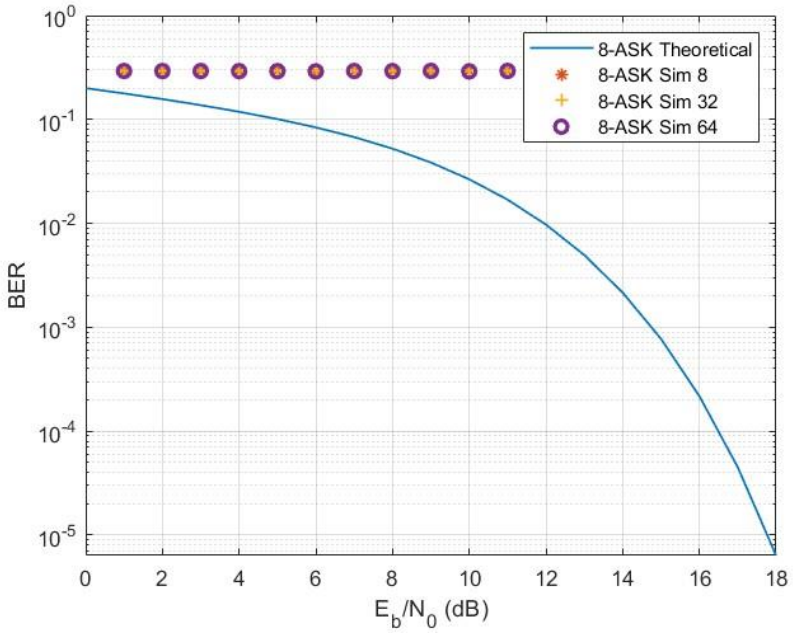
γ) Αντικαθιστούμε τον ορθογωνικό παλμό h με έναν συνημιτονικό μίας περιόδου. Εκτελώντας πάλι την εξομοίωση για 8-ASK με το bertool, παράγεται το διπλανό σχήμα. Μπορεί κανείς να παρατηρήσει, ότι τα αποτελέσματα που παράχθηκαν με το νέο παλμό h είναι εξίσου σωστά με αυτά που παράχθηκαν με το ορθογωνικό. Επομένως, η επίδοση του νέου συστήματος L-ASK παραμένει υψηλή.



Στην συνέχεια, ωστόσο, αν επαναφέρουμε την εντολή `matched=h;` αντί της

```
for i=1:nsamp
    matched(i)=h(end-i+1);
end
```

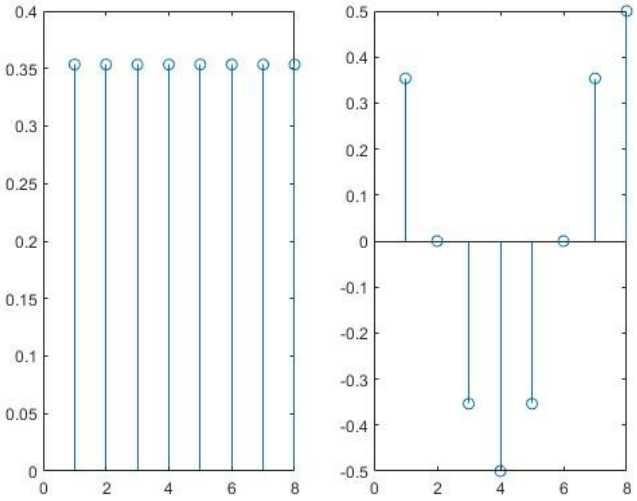
θα παρατηρήσουμε, με την βοήθεια του παρακάτω σχήματος ότι τα αποτελέσματα που παίρνουμε δεν είναι σωστά, ειδικά για μικρές τιμές του nsamp, όπως nsamp = 8 ή 16.



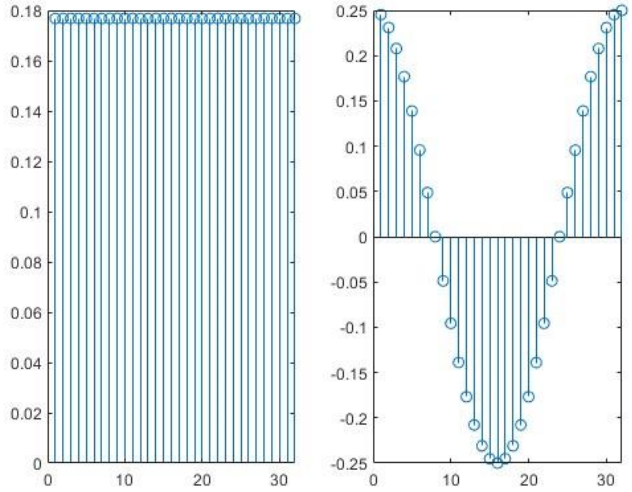
Αυτό συμβαίνει, καθώς, όπως παρατηρούμε από τα παρακάτω σχήματα, όσο μικρότερο είναι το nsamp, τόσο μεγαλύτερη απόκλιση θα έχουν οι διαδοχικές τιμές του καθενός συμβόλου στο σήμα y. Επομένως, η πιθανότητα αναγνώρισης εσφαλμένου συμβόλου είναι σταθερά μεγαλή. Αντίθετα, εφόσον με την χρήση του ορθογωνικού παλμού κάθε σύμβολο παραμένει σε μία στάθμη μετά το upsample, τότε το BER θα εξαρτάται αποκλειστικά απ' τον σηματοθορυβικό λόγο EbNo.

Τα παρακάτω σχήματα προέκυψαν μέσω του εξής κώδικα:

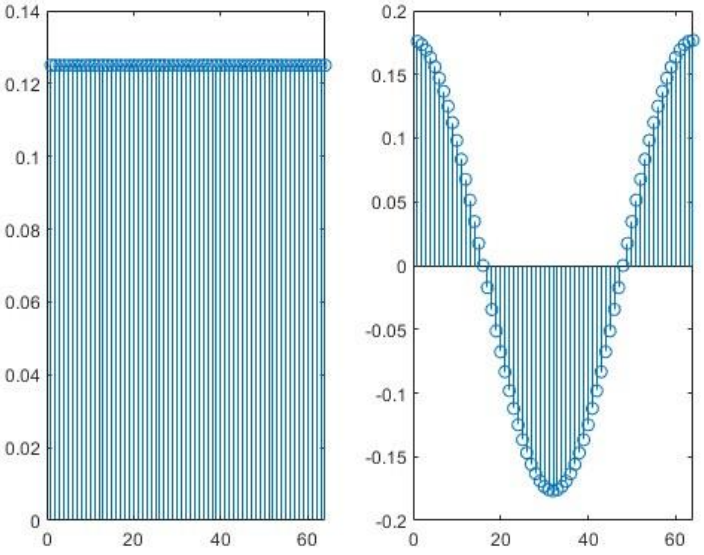
```
%% Question 3.C
clear all;
nsamp = [8,32,64];

for i=1:3
    xaxis = [1:nsamp(i)];
    h1=ones(1,nsamp(i)); h1=h1/sqrt(h1*h1');
    h2=cos(2*pi*(1:nsamp(i))/nsamp(i)); h2=h2/sqrt(h2*h2');

    figure(i+3);
    subplot(1,2,1); stem(xaxis,h1);
    subplot(1,2,2); stem(xaxis,h2);
    pause
end
```



nsamp = 8



nsamp = 32



nsamp = 64