ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΨΗΦΙΑΚΕΣ ΕΠΙΚΟΙΝΩΝΙΕΣ Ι

Εργαστηριακή Άσκηση 4

Ονοματεπώνυμο: Μπέτζελος Χρήστος

Α.Μ. : 031 16 067

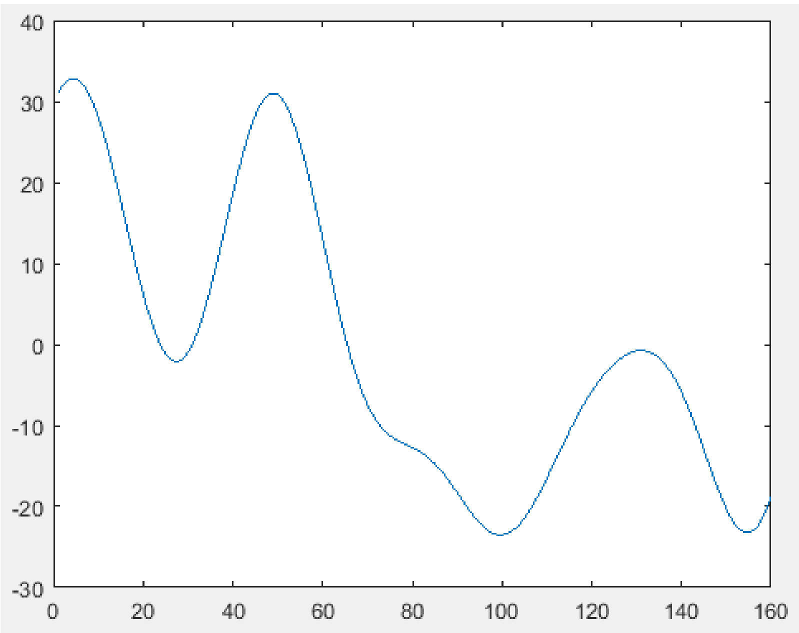
Εξάμηνο: 6°

Μέρος 1°

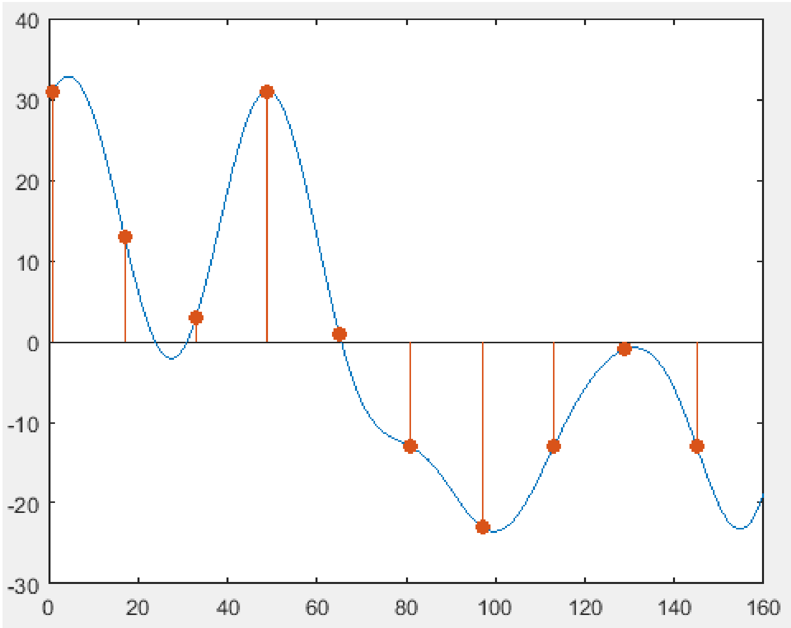
Παράγουμε τυχαία δυαδική ακολουθία 10000 bits και στη συνέχεια αντίστοιχο σήμα 32-ASK βασικής ζώνης, με τα εξής χαρακτηριστικά:

- Κωδικοποίηση Gray
- Σηματοδοσία Nyquist root raised cosine, με roll-off =0.25
- Υπερ-δειγματοληψία με nsamp=16 δείγματα ανά βασική περίοδο T
- Τάξη φίλτρου πομπού: 128 (8 περιόδων, group_delay=4T)

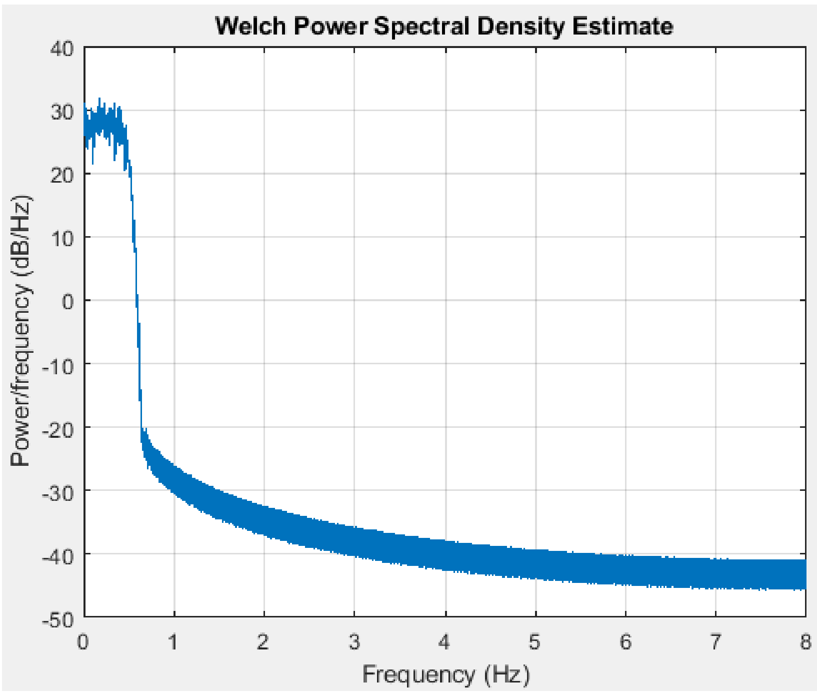
Υπολογίζουμε το σήμα στην έξοδο προσαρμοσμένου φίλτρου στο δέκτη και δείχνουμε τμήμα του σήματος αυτού (με την εντολή plot) διάρκειας 10T.



Υπερθέτουμε (με την εντολή stem) στο τμήμα αυτό τα αντίστοιχα δείγματα του σήματος εισόδου, τα οποία, απουσία θορύβου, συμπίπτουν αντίστοιχα με την γραφική παράσταση εξόδου στον δέκτη.



Σχεδιάζουμε (με την εντολή pwelch) το φάσμα του σήματος στο δέκτη και το διάγραμμα φαίνεται παρακάτω. Παρατηρούμε έτσι ότι έχει την μορφή ενός βαθυπερατού φίλτρου.



Κώδικας 1^{ου} Μέρους και εξήγηση

```
clc;
clear all;
close all;

L=32;
k=log2(L);
Nbits=10000; %πλήθος bits (μήκος ακολουθίας)
Nsymb=Nbits/k; %πλήθος συμβόλων
x=randi([0,1],[1,Nbits]); % τυχαία δυαδική ακολουθία 10000 bits

%κωδικοποίηση Gray
step=2;
mapping=[step/2; -step/2];
if(k>1)
 for j=2:k
 mapping=[mapping+2^(j-1)*step/2; -mapping-2^(j-1)*step/2];
 end
end
xsym=bi2de(reshape(x,k,length(x)/k).','left-msb');
y=[];
for i=1:length(xsym)
 y=[y mapping(xsym(i)+1)];
end

nsamp=16; %συντελεστής υπερδειγμάτισης
delay=4; %group delay
filtorder=delay*nsamp*2; %τάξη φίλτρου
rolloff=0.25; %συντελεστής πτώσης
%κρουστική απόκριση φίλτρου
rNyquist=rcosine(1,nsamp,'fir/sqrt',rolloff,delay);

%υπερδειγμάτιση και εφαρμογή φίλτρου rNyquist
y1=upsample(y,nsamp);
ytx=conv(y1,rNyquist); clear y1;
%φιλτράρισμα σήματος
yrx=conv(ytx,rNyquist);
%περικοπη λόγω καθυστέρησης
yrx=yrx(2*delay*nsamp+1:end-2*delay*nsamp);
%σχεδίαση yrx και υπέρθεση της εισόδου
figure(1);
grid;
plot(yrx(1:10*nsamp)); hold;
stem([1:nsamp:nsamp*10],y(1:10),'filled');
% το φάσμα του σήματος στο δέκτη
figure(2);
pwelch(yrx,[],[],[],nsamp);
```

Για την κωδικοποίηση κατά Gray δυαδικού διανύσματος, x, σε σύμβολα 32-ASK, χρησιμοποιήθηκε το αντίστοιχο τμήμα κώδικα που φαίνεται παραπάνω. Το διάνυσμα mapping περιέχει τα πλάτη των σημείων 32-ASK, όπου τώρα, δεν είναι όπως ήταν στις προηγούμενες εργαστηριακές l=-L+1:2:L-1, αλλά η σειρά τους καθορίζεται από τον κώδικα gray. Το step είναι η απόσταση αυτών των γειτονικών σημείων 32-ASK. Επιπλέον, ο κώδικας κάνει reshape το αρχικό διάνυσμα x ώστε τα 10000 bits να εμφανιστούν σε 2000 5-άδες, δηλαδή στα αντίστοιχα σύμβολα. Στη συνέχεια, μετατρέπει τις 5-άδες στους αντίστοιχους δεκαδικούς και χρησιμοποιώντας το διάνυσμα mapping, τους κατατάσσει στα αντίστοιχα πλάτη.

Μέρος 2ο

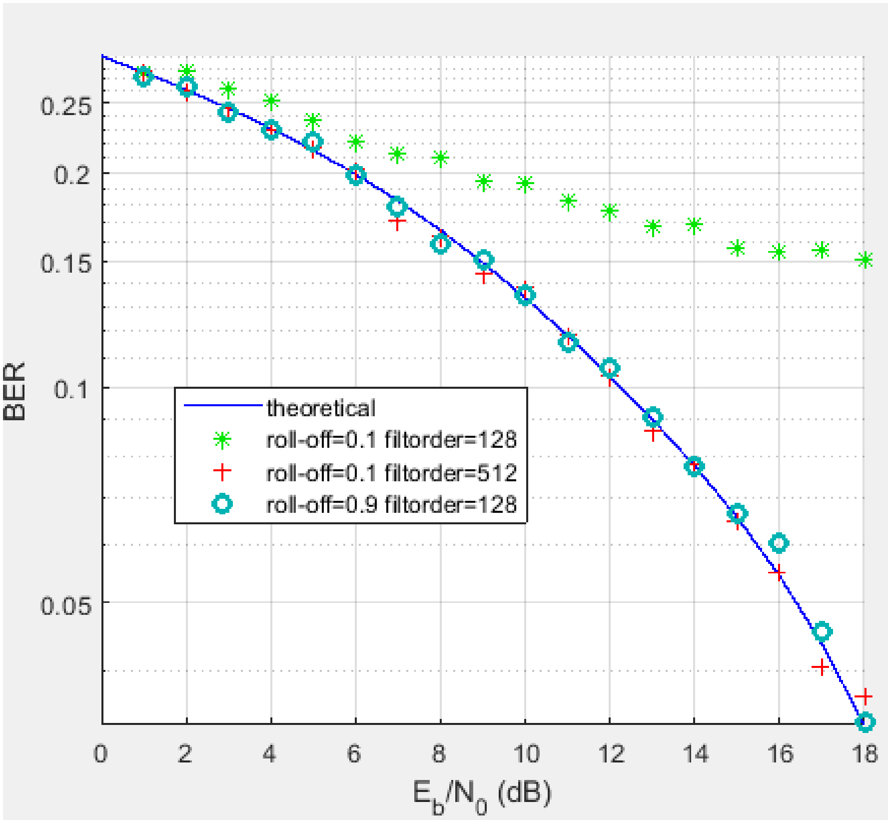
Για την 32-ASK, λαμβάνουμε την καμπύλη BER-Eb/No θεωρητικά και με εξομοίωση

i. με **roll-off=0.1** και **τάξη φίλτρου=128** (8 περιόδων, group_delay =4T).

ii. με **roll-off=0.1** και **τάξη φίλτρου=512** (32 περιόδων, group_delay =16T).

iii. με **roll-off=0.9** και **τάξη φίλτρου=128** (8 περιόδων, group_delay =4T).

Οι καμπύλες με το εργαλείο bertool φαίνονται παρακάτω σε κοινό διάγραμμα.



Παρατηρούμε ότι και οι δύο παράμετροι παίζουν σημαντικό ρόλο στη σηματοδοσία Nyquist. Με χαμηλή τιμή roll-off και τάξης φίλτρου τα αποτελέσματα που παίρνουμε έχουν μεγάλη απόκλιση από τα θεωρητικά. Ενώ, με την αύξηση ενός από τους δύο παράμετρους, είτε του roll-off είτε της τάξης του φίλτρου, η καμπύλη προσομοίωσης είναι πολύ κοντά στη θεωρητική. Άρα, συμπεραίνουμε ότι ο συνδυασμός roll-off και filtorder είναι αυτός που καθορίζει την ποιότητα του φίλτρου.

Παρακάτω παρουσιάζεται ο κώδικας της συνάρτηση ask_Nyq_filter, που μαζί με τη γνωστή ask_ber_func, τρέξαμε το εργαλείο bertool. Τρέξαμε συγκεκριμένα τρεις φορές το Monte Carlo για τις παραπάνω τρεις διαφορετικές περιπτώσεις μεταβάλλοντας κάθε φορά είτε το roll-off, είτε το group delay, που με τη σειρά του μεταβάλλει το filtorder.

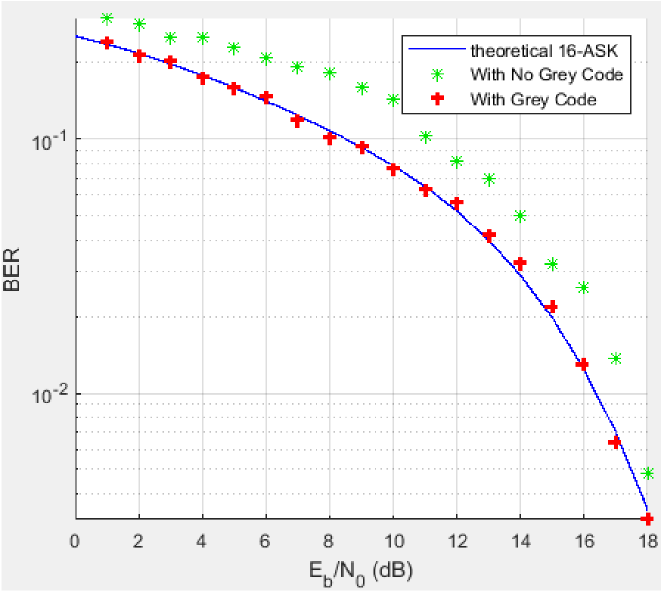
Κώδικας 2ου Μέρους και εξήγηση

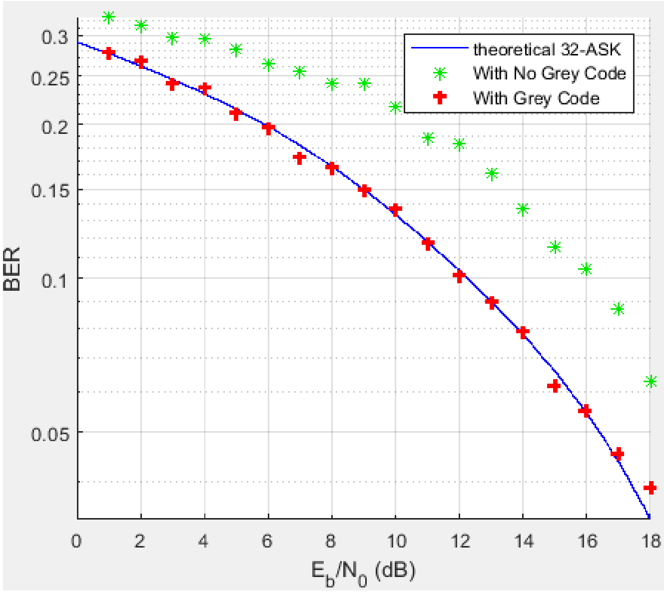
```
function errors=ask_Nyq_filter(k,Nsymb,nsamp,EbNo)
L=2^k;
SNR=EbNo-10*log10(nsamp/2/k);
x=randi([0,1],[1,Nsymb*k]); % τυχαία δυαδική ακολουθία Nsymb*k bits
xreshape=reshape(x,[k,Nsymb])'; %reshape σε σύμβολα

%κωδικοποίηση Gray
step=2;
mapping=[step/2; -step/2];
if(k>1)
 for j=2:k
 mapping=[mapping+2^(j-1)*step/2; -mapping-2^(j-1)*step/2];
 end
end
xsym=bi2de(reshape(x,k,length(x)/k).','left-msb');
y=[];
for i=1:length(xsym)
 y=[y mapping(xsym(i)+1)];
end

delay=4; %group delay
filtorder=delay*nsamp*2; %τάξη φίλτρου
rolloff=0.9; %συντελεστής πτώσης

%κρουστική απόκριση φίλτρου
rNyquist=rcosine(1,nsamp,'fir/sqrt',rolloff,delay);

%υπερδειγμάτιση και εφαρμογή φίλτρου rNyquist
y1=upsample(y,nsamp);
ytx=conv(y1,rNyquist);
ynoisy=awgn(ytx,SNR,'measured');
yrx=conv(ynoisy,rNyquist);
yrx=yrx(2*delay*nsamp+1:end-2*delay*nsamp); %περικοπή λόγω καθυστέρησης
yrx=downsample(yrx,nsamp); %υποδειγμάτιση

%σύγκριση λαμβανόμενου συμβόλου με το διάνυσμα mapping
for i=1:length(yrx)
    [m,j]=min(abs(mapping-yrx(i)));
    xr(i,:)=de2bi(j-1,k,'left-msb');
end
err=not(xr==xreshape);
errors=sum(sum(err)); %βρίσκουμε τα bits errors
```

Για την εκτίμηση Μέγιστης Πιθανοφάνειας, συγκρίνουμε κάθε λαμβανόμενο σύμβολο yrx, (δείγμα στην έξοδο του προσαρμοσμένου φίλτρου) με τα στοιχεία του διανύσματος mapping. Η θέση του πλησιέστερου στοιχείου στο διάνυσμα δίνει και την αντίστοιχη κωδικολέξη. Οπότε τελικά στην περίπτωση μας 32-ASK, προκύπτει στην έξοδο ένας δυσδιάστατος πίνακας xr (2000x5) που περιέχει τις αντίστοιχες κωδικολέξεις. Έτσι, συγκρίνουμε αυτόν τον πίνακα xr με τον αρχικό πίνακα xreshape. Επειδή θέλουμε bit error και όχι symbol error αθροίζουμε όλα τα λάθη και βρίσκουμε έτσι τον συνολικό αριθμό των errors.

Μέρος 3ο

Παράγουμε καμπύλες BER-Eb/No για τις 16-ASK και 32-ASK με κωδικοποίηση άλλη (όχι Gray), με mapping=-(L-1):step:(L-1) και τις συγκρίνουμε με τις θεωρητικές. Έχουμε θεωρήσει συντελεστή πτώσης rolloff=0.5 και τάξη φίλτρου filtorder=128.





Παρατηρούμε ότι με κωδικοποίηση άλλη και όχι gray, τα αποτελέσματα της προσομοίωσης έχουν μεγαλύτερη απόκλιση από τη θεωρητική συγκριτικά με όταν είχαμε την κωδικοποίηση gray. Αυτό ισχύει ειδικά περισσότερο για την 32-ASK. Αυτό συμβαίνει επειδή με την κωδικοποίηση gray τα γειτονικά σύμβολα διαφέρουν κατά ένα μόνο bit οπότε έχουμε χαμηλότερη πιθανότητα εσφαλμένου ψηφίου.

Για τα παραπάνω διαγράμματα χρησιμοποιήσαμε το εργαλείο bertool και αντί της προηγούμενης συνάρτησης ask_Nyq_filter χρησιμοποιήσαμε την ask_Nyq_filter_new της οποίας η μόνη διαφορά είναι στην κωδικοποίηση.

Κώδικας 3ου Μέρους

Η μόνη διαφορά συγκριτικά με τον κώδικα του 2ου μέρους είναι αυτή η κωδικοποίηση στη θέση της gray code.

```
%άλλη κωδικοποίηση
step=2;
mapping=-(L-1):step:(L-1);
xsym=bi2de(reshape(x,k,length(x)/k).','left-msb');
y=[];
for i=1:length(xsym)
 y=[y mapping(xsym(i)+1)];
end
```

Μέρος 4ο

Προσαρμόζουμε τους παραμέτρους του παραπάνω συστήματος μετάδοσης στα εξής πραγματικά δεδομένα/απαιτήσεις:

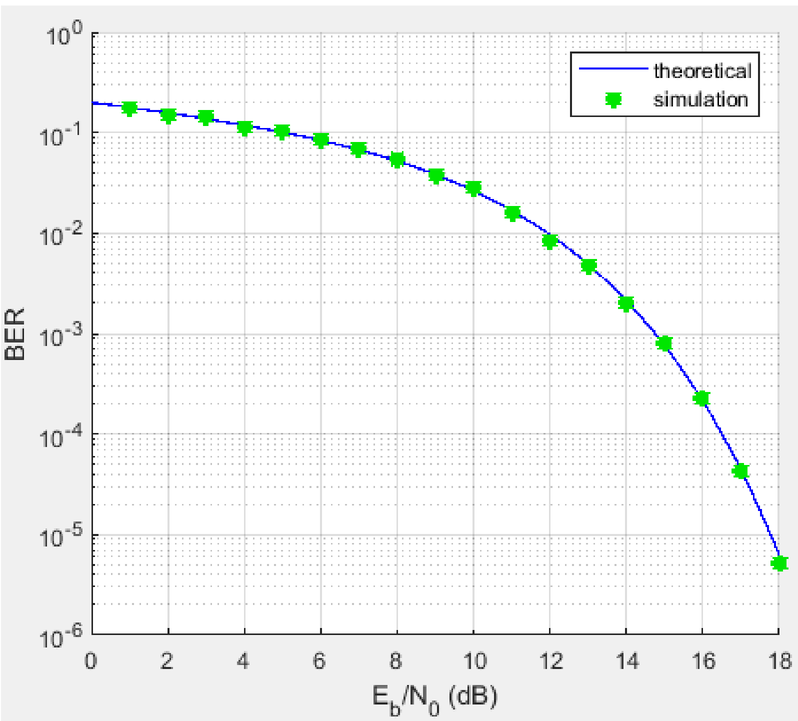
- Εύρος (βασικής) ζώνης διαύλου W=1 MHz
- Πυκνότητα φάσματος θορύβου No=100 picowatt/Hz (μονόπλευρο)
- Ρυθμός μετάδοσης 4 Mbps
- Ανεκτό BER=1 Kbps

Το απαιτούμενο εύρος βασικής ζώνης με σηματοδοσία Nyquist, ισούται με $W = \frac{1}{2T}(1 + a)$, όπου α ο συντελεστής εξάπλωσης (roll-off factor) του φίλτρου Nyquist και 1/T ο ρυθμός μετάδοσης συμβόλων (ονομαζόμενος και Baud Rate).

Ο ρυθμός μετάδοσης, R (bits/s), συνδέεται με το 1/T και το μέγεθος του σχηματικού αστερισμού, L, με τη σχέση $\frac{R}{\log_2 L} = \frac{1}{T}$

Με βάση τα πραγματικά δεδομένα και εκτελώντας τις πράξεις παρατηρούμε ότι το μικρότερο δυνατό μέγεθος σχηματικού αστερισμού L είναι το 8. Δηλαδή L=8. Έτσι προκύπτει ότι ο συντελεστής εξάπλωσης (roll-off factor) ισούται με 0.5.

Προκειμένου να επαληθευθούν οι προδιαγραφές εύρους ζώνης και BER κάναμε πάλι χρήση του bertool προσθέτοντας στη συνάρτηση ask_Nyq_filter τα παραπάνω πραγματικά δεδομένα. Το αποτέλεσμα της προσομοίωσης συγκριτικά με τη θεωρητική καμπύλη φαίνεται στο παρακάτω σχήμα. Ο νέος κώδικας της ask_Nyq_filter είναι ο παρακάτω.



Κώδικας 4ου Μέρους

```
function errors=ask_Nyq_filter(k,Nsymb,nsamp,EbNo)
W=10^6; % Εύρος (βασικής) ζώνης διαύλου
R=4*10^6; % Ρυθμός μετάδοσης
a=(2*W*k/R)-1; % ο συντελεστής εξάπλωσης (roll-off factor)
L=2^k;
SNR=EbNo-10*log10(nsamp/2/k);
x=randi([0,1],[1,Nsymb*k]); % τυχαία δυαδική ακολουθία Nsymb*k bits
xreshape=reshape(x,[k,Nsymb])'; %reshape σε σύμβολα

%κωδικοποίηση Gray
step=2;
mapping=[step/2; -step/2];
if(k>1)
 for j=2:k
 mapping=[mapping+2^(j-1)*step/2; -mapping-2^(j-1)*step/2];
 end
end
xsym=bi2de(reshape(x,k,length(x)/k).','left-msb');
y=[];
for i=1:length(xsym)
 y=[y mapping(xsym(i)+1)];
end

delay=4; %group delay
filtorder=delay*nsamp*2; %τάξη φίλτρου
rolloff=a; %συντελεστής πτώσης

%κρουστική απόκριση φίλτρου
rNyquist=rcosine(1,nsamp,'fir/sqrt',rolloff,delay);

%υπερδειγμάτιση και εφαρμογή φίλτρου rNyquist
y1=upsample(y,nsamp);
ytx=conv(y1,rNyquist);
ynoisy=awgn(ytx,SNR,'measured');
yrx=conv(ynoisy,rNyquist);
yrx=yrx(2*delay*nsamp+1:end-2*delay*nsamp); %περικοπή λόγω καθυστέρησης
yrx=downsample(yrx,nsamp); %υποδειγμάτιση

%σύγκριση λαμβανόμενου συμβόλου με το διάνυσμα mapping
for i=1:length(yrx)
    [m,j]=min(abs(mapping-yrx(i)));
    xr(i,:)=de2bi(j-1,k,'left-msb');
end
err=not(xr==xreshape);
errors=sum(sum(err)); %βρίσκουμε τα bits errors
```