ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

***ΑΜ:*** ***03120233*** ***Ονοματεπώνυμο:*** ***Χρήστος Ηλιακόπουλος***

4η Εργαστηριακή Άσκηση

***ΜΕΡΟΣ 1:***

Σημείωση: Επειδή, η συνάρτηση reshape χρειάζεται πραγματικό αριθμό και η εκφώνηση ζητάει 10000 bits και σήμα 8-ASK δηλαδή κ=3(αποτέλεσμα length(x)/k = 10000/3=3333.3333 όχι integer) αλλά και αφού με την στρογγυλοποίηση του αποτελέσματος length(x)/k θα χαθούν δεδομένα και παράγεται σφάλμα στη reshape αντικαταστάθηκε το αρχικό μήκος με 9999 για να είναι δυνατοί οι υπολογισμοί.

Παραγωγή τυχαίας δυαδικής ακολουθίας 9999 bits και στη συνέχεια σήμα 8-ASK βασικής ζώνης με τα εξής χαρακτηριστικά:

* Κωδικοποίηση gray
* Σηματοδοσία Nyquist root raised cosine, με roll-off=0.4
* Υπερ-δειγματοληψία με nsamp=32 δείγματα ανά βασική περίοδο Τ
* Τάξη φίλτρου πομπού: 256 (8 περιόδων, group\_delay=4T)

Α)Υπολογισμός σήματος εξόδου προσαρμοσμένου φίλτρου στο δέκτη και προβολή τμήματος αυτού διάρκειας 10Τ.

Εικόνα που περιέχει κείμενο, διάγραμμα, γράφημα, γραμμή

Περιγραφή που δημιουργήθηκε αυτόματα

Β) Υπερθέτουμε (με την εντολή stem) στη συνέχεια, πάνω στην προηγούμενη γραφική παράσταση του σήματος που λαμβάνει ο δέκτης, τα αντίστοιχα σήματα εισόδου, τα οποία, λόγω της απουσίας θορύβου, συμπίπτουν αντίστοιχα με τη γραφική.

Εικόνα που περιέχει γραμμή, διάγραμμα, γράφημα, κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει κείμενο, γράφημα, γραμμή, στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματαΓ) Με την εντολή pwelch σχεδιάζουμε το φάσμα του σήματος στο δέκτη. Παρατηρούμε ότι το αποτέλεσμα θυμίζει αρκετά το αποτέλεσμα που θα έδινε ένα βαθυπερατό φίλτρο. Όσο πιο μικρό είναι το roll-off τόσο πιο απότομο το φίλτρο μας και τόσο πιο πολύ θα θυμίζει low pass filter(αποκόπτει ιδανικά στη συχνότητα αποκοπής που επιθυμούμε). Το εύρος ζώνης είναι περίπου στο 0.08.

clc

clear all;

close all;

%variables

k=3;

L=2^k; %filter 8-ASK

num\_bits=9999;

Nsymb = num\_bits/k; %mhkos ths eksomioumenhs akolouthias

nsamp = 32; %syntelesths uperdigmatishs

delay = 4 %group delay

filterorder = delay\*nsamp\*2; %filter's order = 256, nsamp =32 and delay =4T

rolloff=0.4; %rolloff factor, suntelesths ptwshs

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

%DHMIOURGIA 9999 bits anamesa se 0 kai 1

%x = round(rand(1, num\_bits));

x= randi([0,1],[1,num\_bits]);

%kwdikopoihsh gray

step = 2; %apostash geitwnikwn shmeiwn L-ASK

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

%shma poy ekpempoume

y1 = upsample(y, nsamp);

ytx = conv(y1, rNyquist);

%eyediagram(ytx(1:2000), nsamp\*2);

%an theloume thorubo xreiazomaste snr

%clear y1;

%shma pou lambanoume

yrx = conv(ytx, rNyquist);%filtrarisma shmatos

yrx = yrx(2\*delay\*nsamp+1:end - 2\*delay\*nsamp); %perikoph logw kathisterhshs

figure(1)

grid; %enwsh twn duo grafikwn

plot(yrx(1:10\*nsamp)); hold;

stem([1:nsamp:nsamp\*10], y(1:10), 'filled');

figure(2)

pwelch(yrx, [],[],[], nsamp);

figure(3)

stem(yrx(1:100));

figure(4)

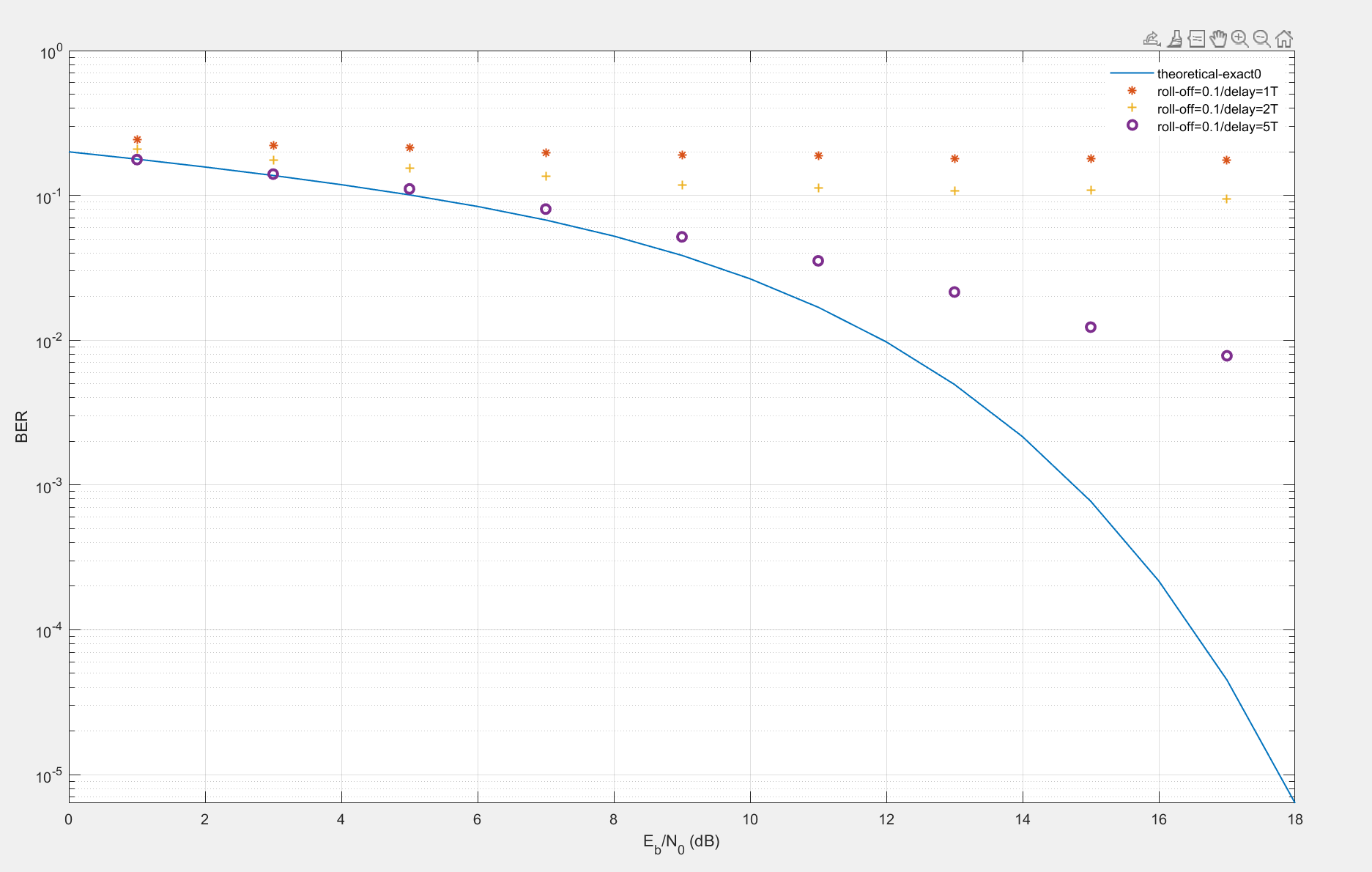
plot(yrx(1:10\*nsamp));

Ο κώδικας είναι ενιαίος και χωρίστηκε σε 2 μέρη για να μην καταναλώσει χώρο στην αναφορά. Βρίσκεται στο αρχείο Lab4\_20233\_1omeros.m

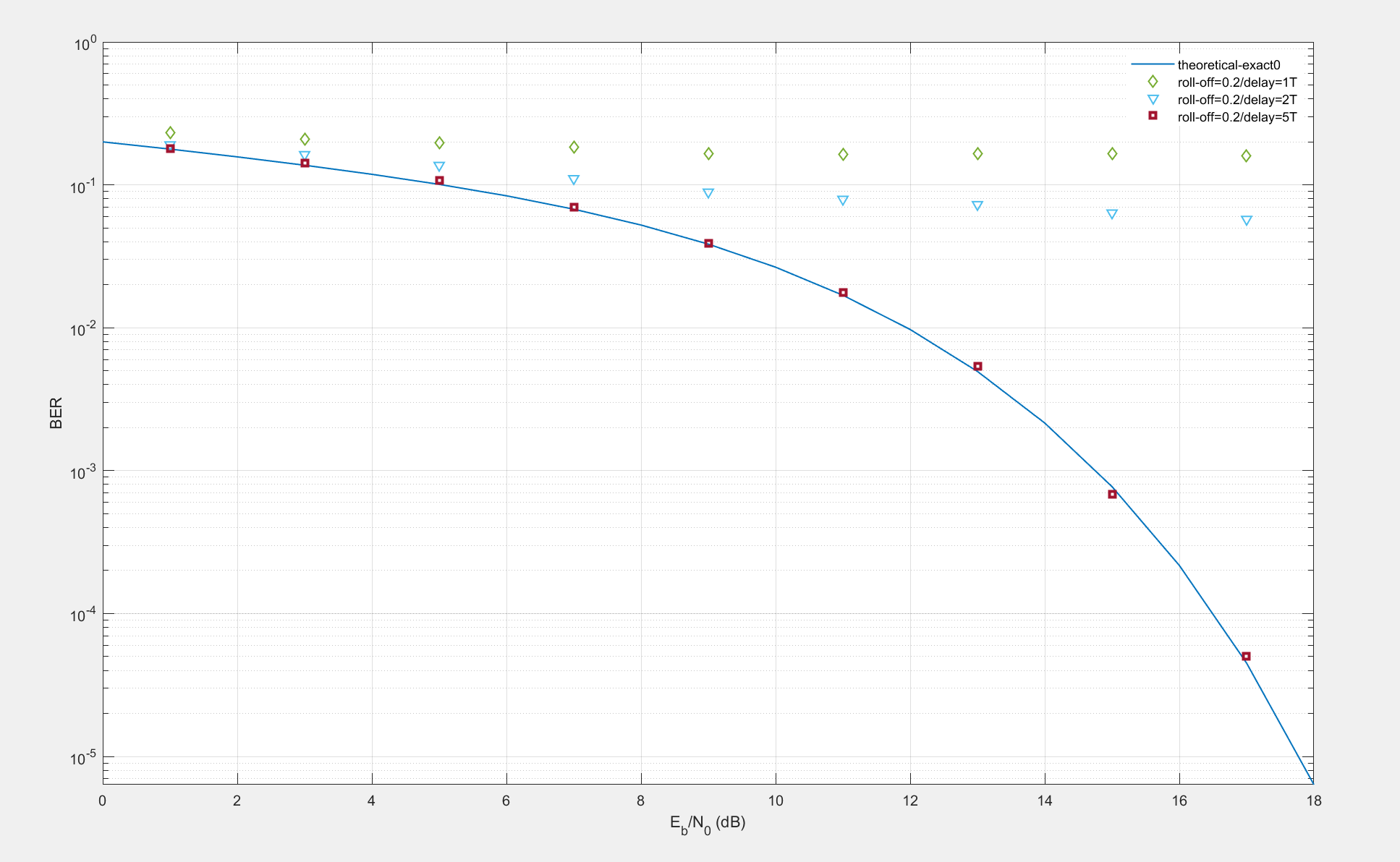
Για την κωδικοποίηση Gray δυαδικού διανύσματος x χρησιμοποιήθηκε ο κώδικας που μας δόθηκε στην άσκηση. Το διάνυσμα mapping περιέχει τα πλάτη των 8-ASK σημείων, όπου καθορίζονται από τον κώδικα gray. Το step είναι η απόσταση μεταξύ των γειτονικών σημείων 8-ASK, όπου στη συγκεκριμένη άσκηση ορίστηκε 2. Ο κώδικας με την εντολή reshape επιτρέπει στο αρχικό διάνυσμα x, των 9999 bits, να εμφανιστεί ως 3333(length(x)/k) 3άδες(k=3), δηλαδή στα αντίστοιχα σύμβολα. Αυτές με τη σειρά τους μετατρέπονται σε αντίστοιχους δεκαδικούς αριθμούς με τη χρήση της bi2de(παίρνει τους binary αριθμούς του πίνακα και τους μετατρέπει σε δεκαδικούς), οι οποίοι στη συνέχεια με το διάνυσμα mapping κατατάσσονται στα αντίστοιχα πλάτη.

***ΜΕΡΟΣ 2:***

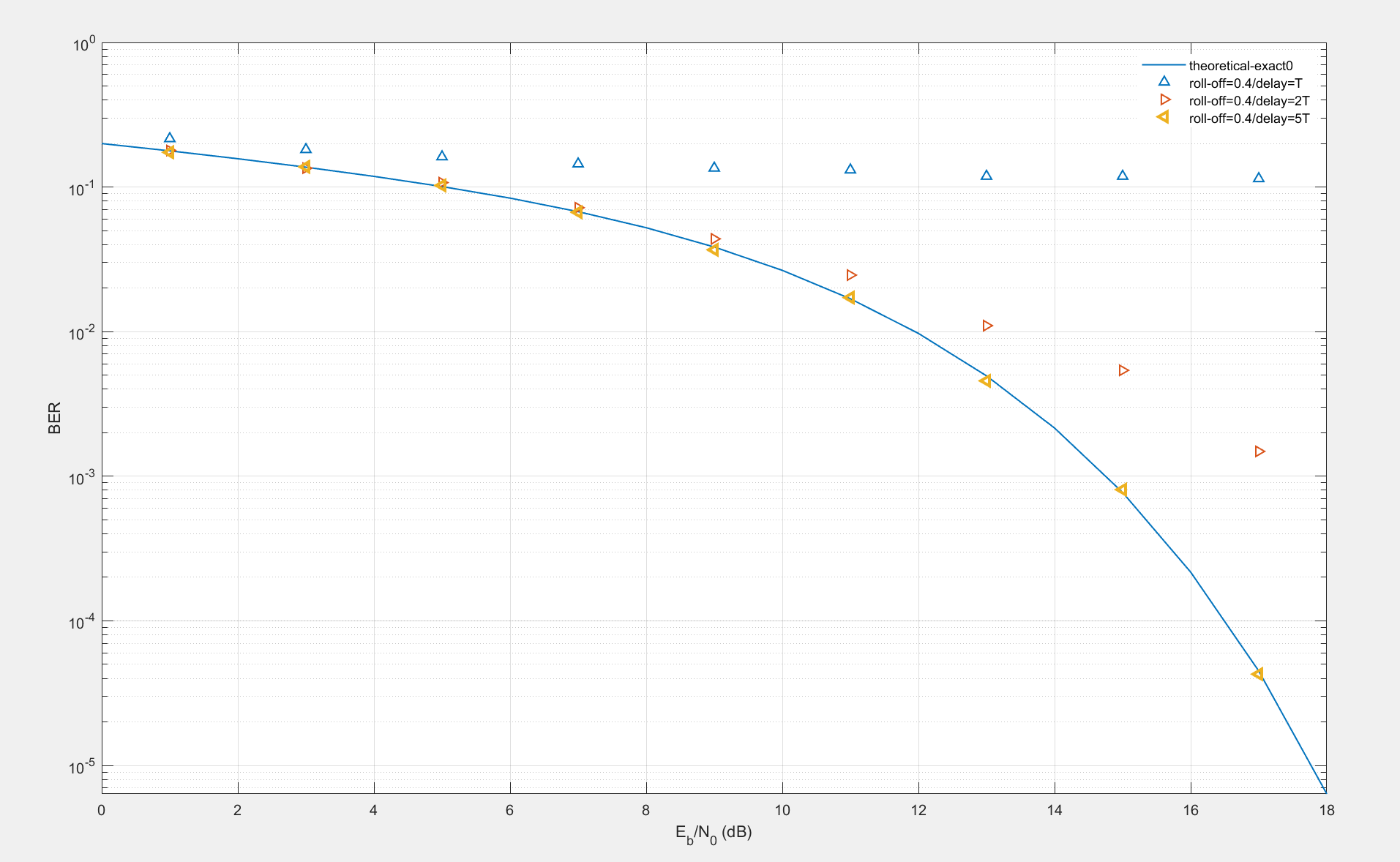
Για roll-off=0.1 και αντίστοιχα group-delay=1T,2T,5T



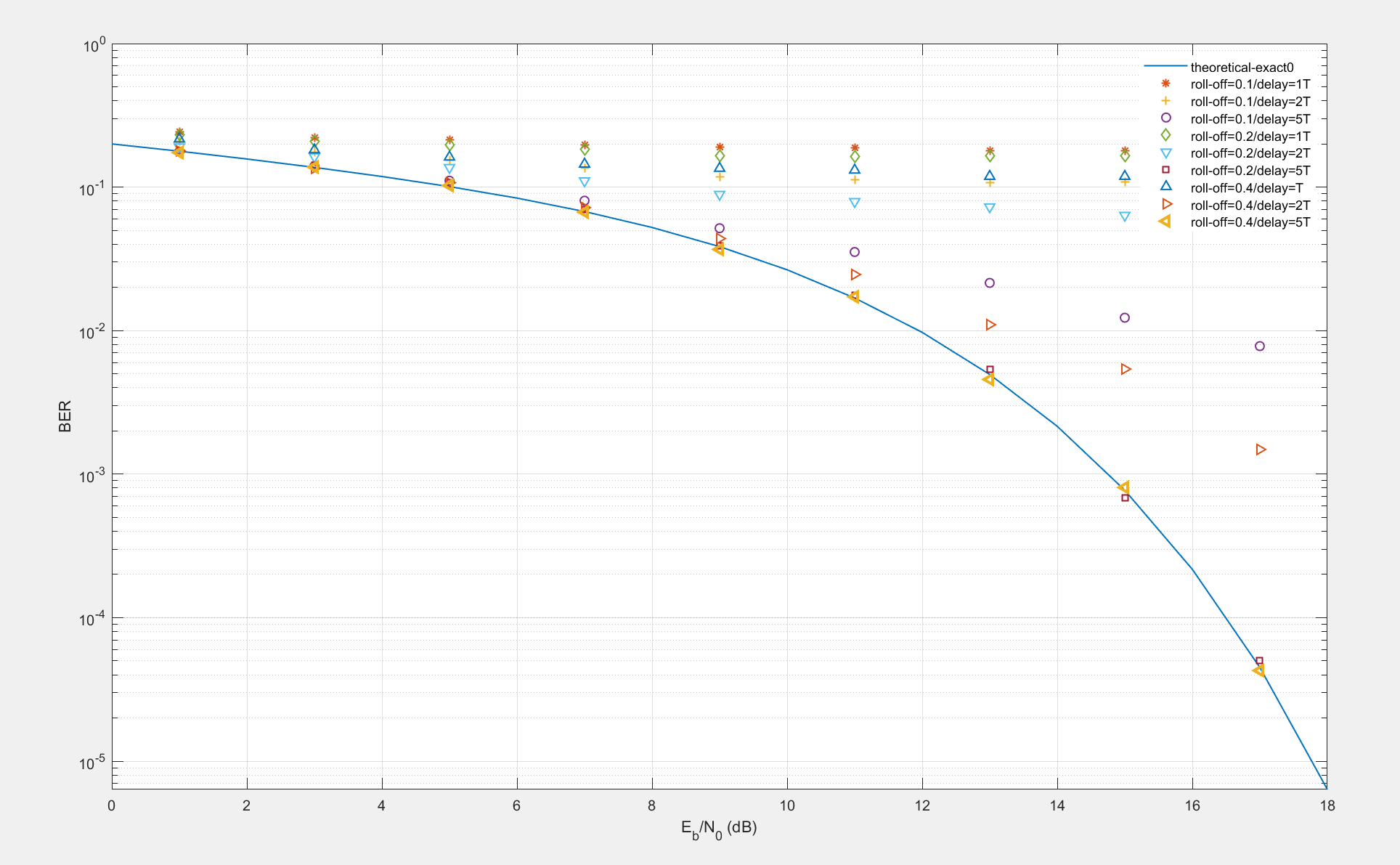
Για roll-off=0.2 και αντίστοιχα group-delay=1T,2T,5T



Για roll-off=0.4 και αντίστοιχα group-delay=1T,2T,5T



Συνολική εικόνα με όλες τις παραμέτρους



Ουσιαστικά χρησιμοποιήσαμε την ask\_ber\_func από το προηγούμενο εργαστήριο και αντικαταστήσαμε τη συνάρτηση που χρησιμοποιεί με τη νέα ask\_Nyq\_filter.

Παρατηρούμε πως και ο παράγοντας roll-off αλλά και το group-delay (το οποίο επηρεάζει την τάξη του φίλτρου μας) έχουν άμεση επίδραση στο να αποκτήσουμε τιμές όσο πιο κοντά γίνεται στις θεωρητικές. Με χαμηλή τιμή roll-off και χαμηλή τιμή καθυστέρησης θα έχουμε το χείριστο αποτέλεσμα καθώς θα έχουμε μικρού τάξης φίλτρου και το φιλτράρισμα θα είναι απότομο(χαμηλό roll-ff). Αυξάνοντας οποιονδήποτε από τους δύο παράγοντας το αποτέλεσμά μας ξεκινάει να πλησιάζει προς τις θεωρητικές τιμές. Γενικότερα, όσο μικρότερος είναι ο συντελεστής roll-off, το φίλτρο παρουσιάζει συμπεριφορά βαθυπερατού φίλτρου, το οποίο όμως έχει ως αποτέλεσμα να είναι πιο ψηλοί οι πλευρικοί λοβοί, άρα και να χρειάζεται μεγαλύτερη καθυστέρηση για την αποφυγή επικάλυψης των εκπεμπόμενων συμβόλων και την αποφυγή αύξησης του BER, δηλαδή να του αυξήσουμε τον βαθμό του φίλτρου(64, 128, 320). Επιπλέον, όσο μεγαλύτερο είναι το delay τόσο μικρότερη θα είναι η πιθανότητα να υπάρχουν επικαλύψεις άρα και σφάλματα στη λήψη.

Κώδικας Ask\_Nyq\_filter

function errors=ask\_Nyq\_filter(k,Nsymb,nsamp,EbNo)

num\_bits=9999;

rolloff=0.4;

L=2^k;

SNR=EbNo-10\*log10(nsamp/2/k);

x=round(rand(1,num\_bits)); % 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=5; %group delay

filtorder=delay\*nsamp\*2; %τάξη φίλτρου

%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

Ο κώδικας της ber\_func είναι προφανώς αυτός που μας έχει δοθεί από το εργαστήριο.

***ΜΕΡΟΣ 3:***

Παραγωγή καμπυλών με άλλη κωδικοποίηση από αυτή τύπου Gray.

Στο συγκεκριμένο παράδειγμα επιλέξαμε mapping=-(L-1):step:(L-1)

Φίλτρο 8-ASK τάξης 320(delay=5T) με roll-off=0.5 με κωδικοποίηση gray και χωρίς

Εικόνα που περιέχει κείμενο, γραμμή, γράφημα, διάγραμμα

Περιγραφή που δημιουργήθηκε αυτόματα

Φίλτρο 16-ASK τάξης 320(delay=5T) με roll-off=0.5 με κωδικοποίηση gray και χωρίς

Εικόνα που περιέχει κείμενο, γραμμή, διάγραμμα, γράφημα

Περιγραφή που δημιουργήθηκε αυτόματα

Προφανώς στο φιλτρο 16-ASK έχουμε num\_bits=10000 και Nsym=2500

Παρατηρείται γενικά πως η κωδικοποίηση Gray θα δώσει καλύτερο αποτέλεσμα από την δεύτερη που επελέγη. Στη δεύτερη κωδικοποίηση τα αποτελέσματα αποκλίνουν περισσότερο από τα θεωρητικά και στις δύο περιπτώσεις (8-ASK και 16-ASK). Αυτό συμβαίνει γιατί στην κωδικοποίηση Gray τα γειτονικά σύμβολα απέχουν μεταξύ τους κατά ένα μόνο bit, ενώ σε διαφορετική κωδικοποίηση τα γειτονικά bit μπορεί να διαφέρουν μεταξύ τους για παραπάνω από ένα bit. Το αποτέλεσμα είναι ότι στην απόφαση του δέκτη για την τοποθέτησή του συμβόλου, πραγματοποιείται σφάλμα στην πρόβλεψη, το οποίο οδηγεί σε μεγαλύτερο αριθμό λανθασμένων bits, άρα και αύξηση του BER.

function errors=ask\_Nyq\_filter\_new(k,Nsymb,nsamp,EbNo)

num\_bits=10000 %9999;

rolloff=0.4;

L=2^k;

SNR=EbNo-10\*log10(nsamp/2/k);

x=round(rand(1,num\_bits)); % τυχαία δυαδική ακολουθία Nsymb\*k bits

xreshape=reshape(x,[k,Nsymb])'; %reshape σε σύμβολα

%κωδικοποίηση Gray

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

delay=2; %group delay

filtorder=delay\*nsamp\*2; %τάξη φίλτρου

%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

Χρησιμοποιήθηκε ξανά η bertool, μόνο που αυτή τη φορά η συνάρτηση που καλούσε ήταν η ask\_Nyq\_filter\_new με την μόνη διαφορά με την απλή ask\_Nyq\_filter να είναι ότι δεν είχα το κομμάτι κωδικοποίησης Gray.

***ΜΕΡΟΣ 4:***

Τα δεδομένα μας είναι τα εξής:

Κ=3, W=10^6 Hz, No=1\*10^(-1)watt/Hz, R=4Mbps, BER≤2\*10^3 bps

Από τις σχέσεις που δίνονται από την εκφώνηση, έχουμε ότι:

W = (1+a)/2T και 1/T = R/log2(L) άρα W=R\*(1+a)/2k αφού L=2^k. Με αντικατάσταση προκύπτει ότι k=(1+a)\*2 (όπου a ο συντελεστής εξάπλωσης roll-off factor). Εφόσον το σύστημα μετάδοσης είναι 8-ASK, k=3, άρα a=0.5 και είναι γνωστό από θεωρία ότι οι τιμές του roll-off factor είναι ανάμεσα στο 0 και το 1, οπότε η παραπάνω τιμή είναι αποδεκτή.

Το BER = 2kbps/R = 2kpbs/4Mbps = 0.0005

Το συγκεκριμένο BER επιτυγχάνεται όταν ο λόγος του Eb/No=15.37

Εικόνα που περιέχει κείμενο, γραμμή, γράφημα, στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

Άρα αφού No=10\*10^(-10)watt/Hz θα έχουμε