



**Εθνικό Μετσόβιο
Πολυτεχνείο**

**Σχολή Ηλεκτρολόγων
Μηχανικών και Μηχανικών
Υπολογιστών**

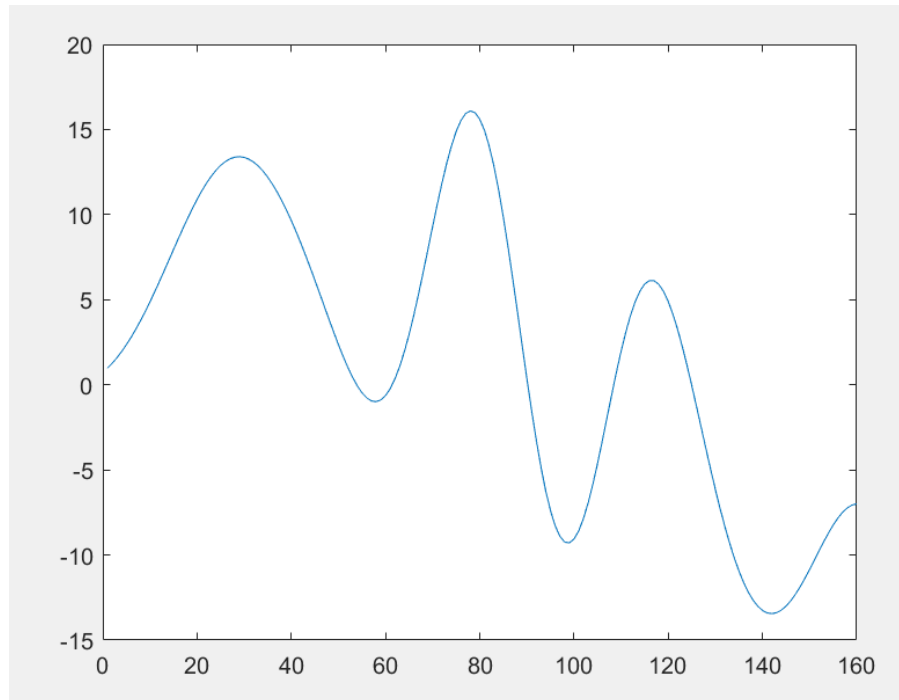
Ψηφιακές Επικοινωνίες Ι
Εργαστηριακή Άσκηση 4

Ελευθερία Αρκαδοπούλου

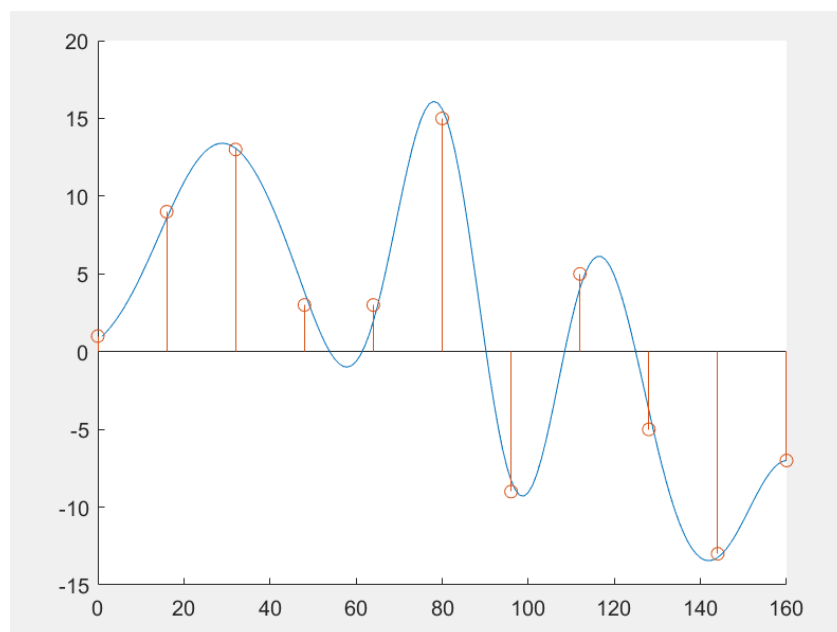
el19442

Ερώτημα 1^ο: Παραγωγή σήματος με φίλτρα Nyquist – Διαγράμματα χρόνου και συχνότητας

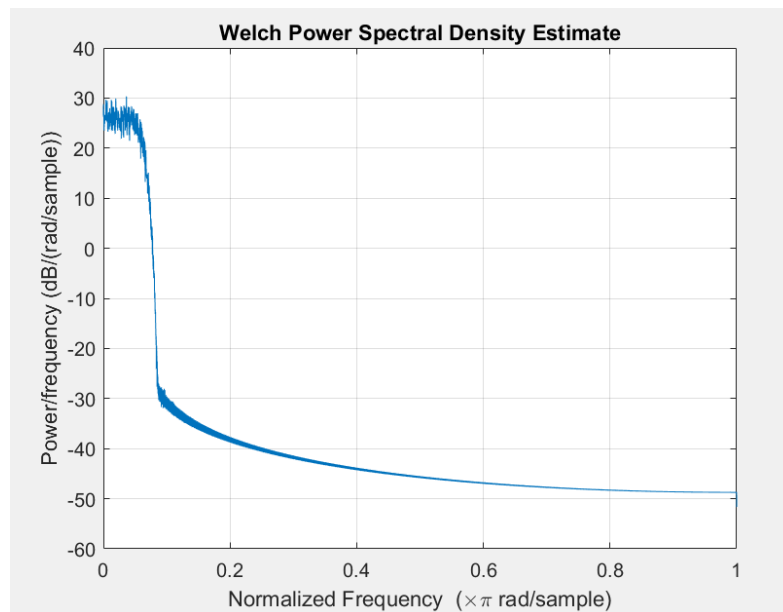
a) Το σήμα στην έξοδο προσαρμοσμένου φίλτρου του δέκτη, διάρκειας 10T



b) Τα αντίστοιχα δείγματα του σήματος εισόδου σε υπέρθεση:



c) Με χρήση της εντολής pwelch, το φάσμα του σήματος στο δέκτη:



Το σήμα μας προσομοιάζει βαθυπερατό φίλτρο, καθώς φαίνονται μόνο χαμηλές συχνότητες σε αυτό. Αυτό οφείλεται στη χαμηλή τιμή του παράγοντα roll-off.

Ο κώδικας για το ερώτημα 1:

```
function errors=ask_Nyq_filter(k,Nsymb,nsamp,EbNo)
%gia 16-ASK theloume k=4
Nsymb=2500
nsamp=16
k=4;
L=2^k;
nsamp=16;
Nsymb=2500;
step=2;
SNR=EbNo-10*log10(nsamp/2/k); % SNR ana deigma simatos
% dianisma tixaion akeraion apo to sinolo {+1, +3, ... +(L-1)}
x=round(rand(1,Nsymb*k));
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
%% orismos parametron filtrou
delay = 4; % Group delay (# of input symbols). gia taksi filtrou 128 delay=4
filtorder = delay*nsamp*2; % taksi filtrou
rolloff = 0.35; % sintelestis ptosis -- rolloff factor
% kroustiki apokrisi filtrou tetr. rizas anips. cos
rNyquist= rcosine(1,nsamp,'fir/sqrt',rolloff,delay);
% -----
% I epomeni entoli gia filtro grammikis ptosis
% (rtrapezium tou kodika 4.1 sto current directory)
% delay>5
% rNyquist=rtrapezium(nsamp,rolloff,delay);
% -----

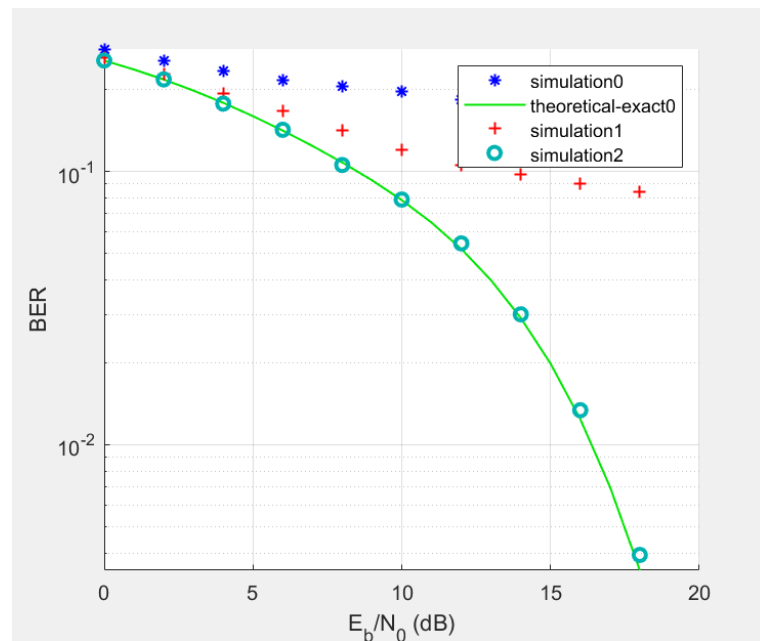
%% ekspempomeno sima
% iperdeigmatisi kai efarmogi filtrou rNyquist
y=upsample(y,nsamp);
ytx = conv(y,rNyquist);
% eyediagram gia meros filtrarismenou simatos
% eyediagram(ytx(1:2000),nsamp*2);
%noisy=awgn(ytx,SNR,'measured'); % ????????? ????
% -----

%% lamvanomeno sima
% filtarisma simatos me filtro tetr. rizas anips. cos
yrx=conv(ytx,rNyquist);
yrx = yrx(2*delay*nsamp+1:end-2*delay*nsamp); % perikopi, logo kathisterisis
yr = downsample(yrx,nsamp);
% anixneutis elaxistis apostasis L platon
l=[-L+1:2:L-1];
for i=1:length(yrx)
    [m,i]=min(abs(l-yrx(i)));
end
figure();
plot(yrx(1:10*nsamp));
figure;
hold on;
plot(yrx(1:10*nsamp));
stem(0:nsamp:10*nsamp,y(1:nsamp:10*nsamp+1));
hold off;
figure();
pwelch(yrx);

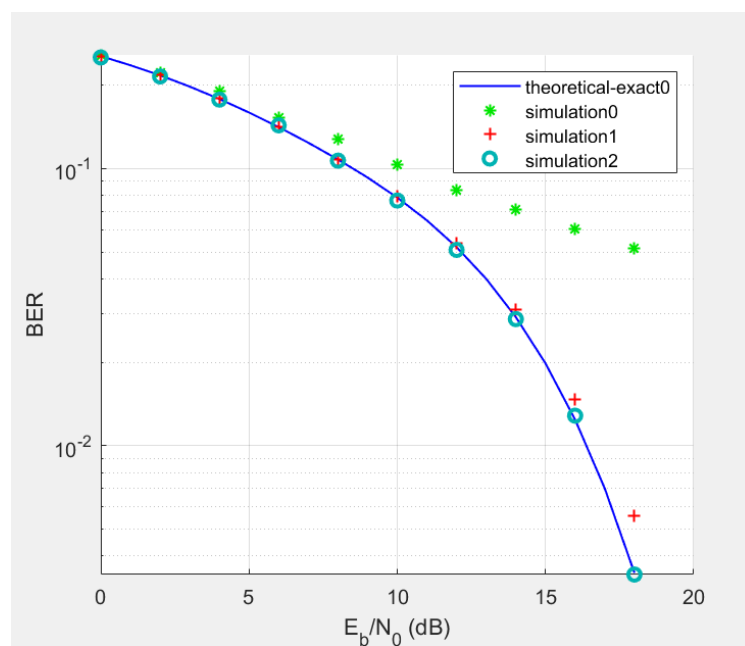
```

Ερώτημα 2º: Υπολογισμός επίδοσης BER vs E_b/N_0

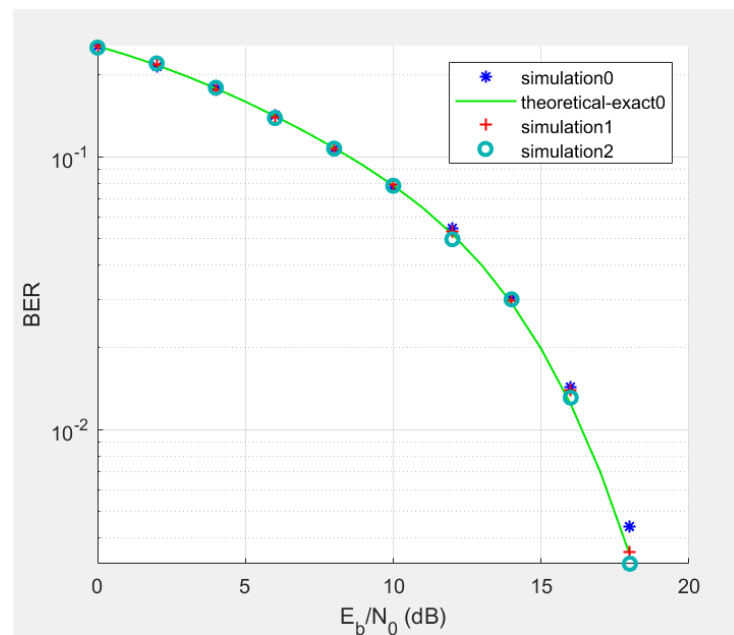
- a) Ως συνάρτηση bertool χρησιμοποιήθηκε η ask_ber_func, η οποία παίρνει τα ορίσματα errors από την lab4_2_19442, η οποία παρατίθεται στο τέλος του ερωτήματος. Η ask_ber_func περιλαμβάνεται στον συμπιεσμένο φάκελο. Με roll-off = 0.1 και τάξη φίλτρου 64, 128 και 512 (simulation 0, 1 και 2 αντίστοιχα) και με χρήση bertool έχουμε τα εξής αποτελέσματα για τη θεωρητική καμπύλη και την καμπύλη εξομοίωσης:



- b) Με roll-off = 0.35:



c) Με roll-off = 0.9:



Παρατηρούμε ότι, όσο ο παράγοντας roll-off γίνεται μεγαλύτερος, τόσο τα αποτελέσματα τείνουν να ταυτιστούν με τη θεωρητική καμπύλη για όλες τις τάξεις φίλτρου. Για μικρό παράγοντα roll-off, όσο μεγαλύτερη η τάξη φίλτρου τόσο πλησιάζουν τα αποτελέσματα τη θεωρητική καμπύλη.

Ο κώδικας:

```
function errors=lab4_2_19442(k,Nsymb,nsamp,EbNo)
%parametroi
delay = 16;
filtorder = delay*nsamp*2;
rolloff = 0.9;

L=2^k;
step=2;
SNR=EbNo-10*log10(nsamp/2/k);
x=round(rand(1,Nsymb*k));
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
```

```

rNyquist= rcosine(1,nsamp,'fir/sqrt',rolloff,delay);
y=upsample(y,nsamp);
ytx = conv(y,rNyquist);
ynoisyy=awgn(ytx,SNR,'measured');
ytx=conv(ynoisyy,rNyquist);
ytx = ytx(2*delay*nsamp+1:end-2*delay*nsamp);
yr = downsample(ytx,nsamp);
xr=[];
for i=1:length(yr)
    [m,j]=min(abs(mapping-yr(i)));
    xr=[xr de2bi(j-1,k,'left-msb')];
end
err=not(x==xr);
errors=sum(err);
end

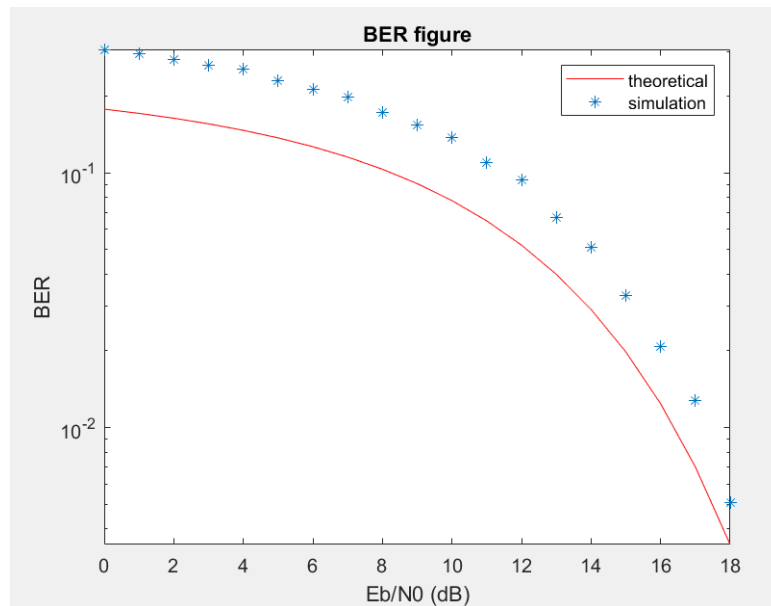
```

Ερώτημα 3^ο: Επίδραση του τρόπου κωδικοποίησης: Gray ή άλλη

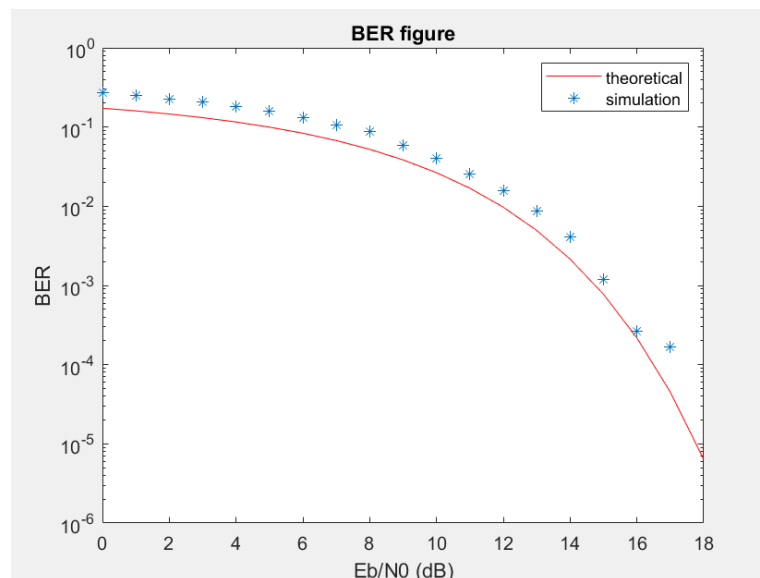
Για την υλοποίηση του ερωτήματος, χρησιμοποιήσαμε τη βοηθητική συνάρτηση `el19442_4_3help`, η οποία κάνει το mapping και χρησιμοποιείται από την `el19442_4_3` για να γεμίσει τον πίνακα errors.

Με κωδικοποίηση `mapping=-(L-1):step:(L-1)`, οι καμπύλες που παράγονται είναι:

16-ASK



8-ASK



Παρατηρούμε ότι, σε αντίθεση με τις αντίστοιχες καμπύλες κωδικοποίησης Gray, τα σφάλματα που προκύπτουν από την εξομοίωση είναι μεγαλύτερα των

αναμενόμενων από την θεωρητική καμπύλη. Παρατηρούμε επίσης ότι για την 16-ASK είναι μεγαλύτερα αυτών της 8-ASK. Το γεγονός αυτό το αποδίδουμε στο ότι, σε αντίθεση με άλλες κωδικοποιήσεις, στην κωδικοποίηση Gray κάθε σφάλμα απόφασης αντιστοιχεί σε ένα μόνο λανθασμένο bit, ενώ όλα τα άλλα θα παραμείνουν ίδια, και άρα θα έχουμε συνολικά λιγότερα σφάλματα.

Ο κώδικας:

```
k=3;
%k=4;
L=2^k;
EbNo=0:18;
EBNO=10.^(EbNo/10);
Petheoretical=(L-1)/L*erfc(sqrt(3*k/(L^2-1)*EBNO));
BERtheoretical=Petheoretical/k;
nsamp=16;
delay = 10;
filtorder = delay*nsamp*2;
rolloff = 0.9;

%disp(EBNO);
errors = zeros(1,18);
for i=1:length(EbNo)
    errors(i)=lab4_3_19442help(k,10000,16,EbNo(i));
end
Penew = errors/10000;
BERnew=Penew/k;

figure(1);
semilogy(EbNo,BERtheoretical,"r",EbNo,BERnew,"*");
title("BER figure");
ylabel("BER");
xlabel("Eb/N0 (dB)");
legend("theoretical","simulation");
```

Ερώτημα 4^ο: Υπολογισμός παραμέτρων συστήματος

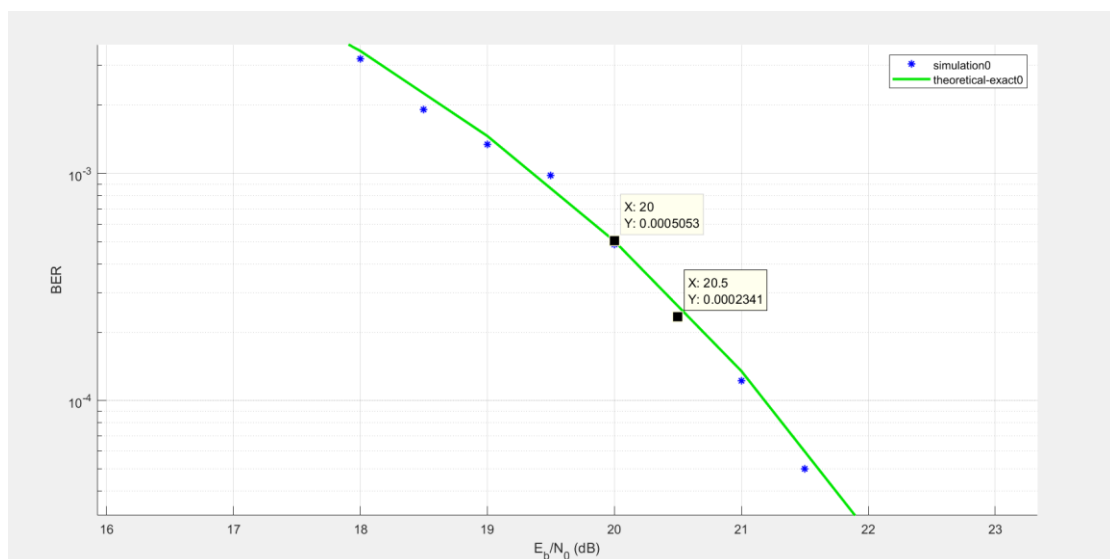
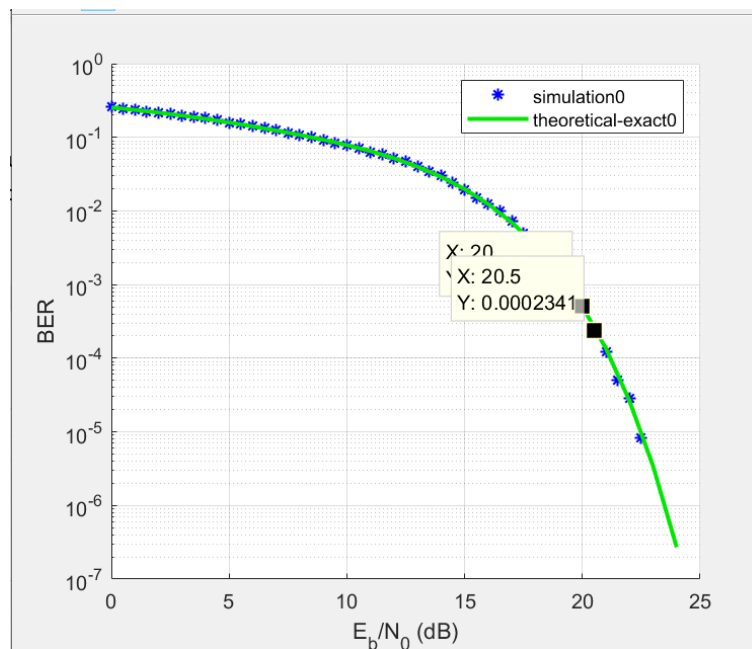
Οι παράμετροι: 16-ASK, W=1 MHz, No=100 picowatt/Hz, R=6Mbps, BER μέχρι 2 Kbps.

$$\frac{1}{T} = \frac{R}{\log 2L} \Rightarrow \frac{1}{T} = 6 * \frac{10^6}{4} \Rightarrow T = 0.66 \mu sec$$

$$W = \frac{1 + roll-off}{2T} \Rightarrow roll-off = W * 2T - 1 = \\ = 2 * 10^6 * 0.66 * 10^{-6} - 1 = 0.32$$

$$Pb = \frac{2 * 10^3}{6 * 10^6} = 0.33 * 10^{-3}$$

Δημιουργούμε τη συνάρτηση lab4_4_el19442, όμοια με αυτή του ερωτήματος 2 αλλά με τις νέες παραμέτρους. Από την προσομοίωση σε bertool:



Άρα, συμπεραίνουμε ότι για τη δικιά μας ζητούμενη τιμή 0.00033, το ζητούμενο Eb/No είναι ανάμεσα στο 20 και στο 20.5.

Ο κώδικας:

```
function errors=lab4_4_19442(k,Nsymb,nsamp,EbNo)
%parametroi
delay = 16;
filtorder = delay*nsamp*2;
rolloff = 0.32;

L=2^k;
step=2;
SNR=EbNo-10*log10(nsamp/2/k);
x=round(rand(1,Nsymb*k));
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

rNyquist= rcosine(1,nsamp,'fir/sqrt',rolloff,delay);
y=upsample(y,nsamp);
ytx = conv(y,rNyquist);
ynoisyy=awgn(ytx,SNR,'measured');
yrx=conv(ynoisyy,rNyquist);
yrx = yrx(2*delay*nsamp+1:end-2*delay*nsamp);
yr = downsample(yrx,nsamp);
xr=[];
for i=1:length(yr)
[m,j]=min(abs(mapping-yr(i)));
xr=[xr de2bi(j-1,k,'left-msb')];
end
err=not(x==xr);
errors=sum(err);
end
```