ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΑΜ: 03120233 Ονοματεπώνυμο: Χρήστος Ηλιακόπουλος

6η Εργαστηριακή Άσκηση

1ο ΜΕΡΟΣ

Παρατίθεται ο κώδικας που υλοποιήθηκε για την ασύμφωνη FSK.

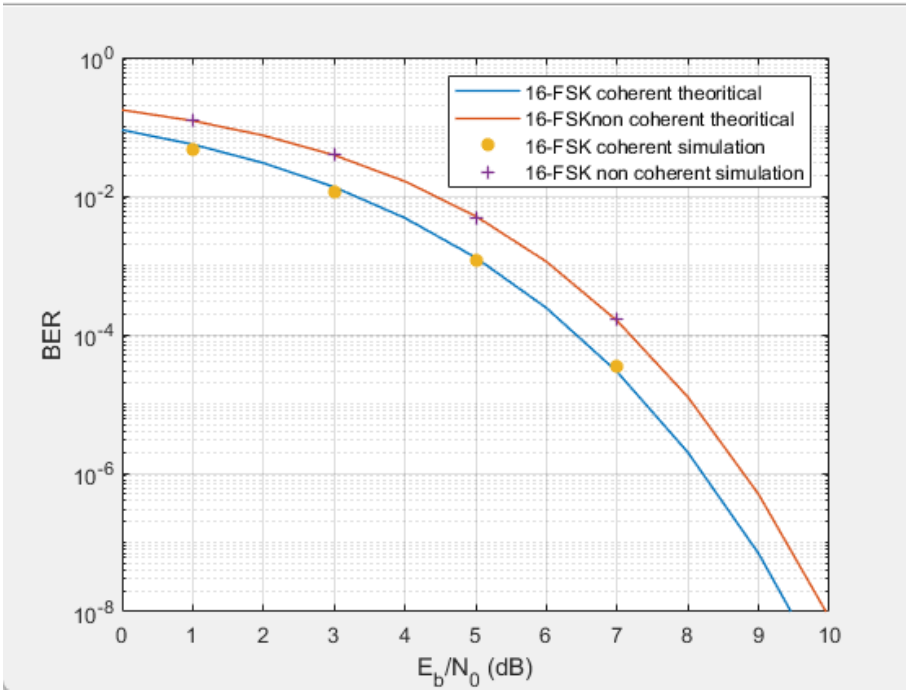
```
function errors=fsk_errors_non_coh(bps,Nsymb,ns,EbNo)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Input parameters
% bps: bits per symbol, Nsymb: numb of simulated symbols
% ns: number of samples per symbol (oversampling)
% EbNo: normalized signal-to-noise ratio, in db
M=2^bps; % number of different symbols
BR=1; % Baud Rate
fc=2*M*BR; % RF frequency
%% Derived parameters
nb=bps*Nsymb; % number of simulated data bits
T=1/BR; % one symbol period
Ts=T/ns; % oversampling period
% M frequencies in "non-coherent" distance (BR)
f=fc+BR*((1:M)-(M+1)/2);
% awgn channel
SNR=EbNo+10*log10(bps)-10*log10(ns/2); % in db
% input data bits
y=randi(2,1,nb)-1; %
x=reshape(y,bps,length(y)/bps)';
t=[0:T:length(x(:,1))*T]'; % time vector on the T grid
tks=[0:Ts:T-Ts]';
%% FSK signal
s=[];
A=sqrt(2/T/ns);
for k=1:length(x(:,1))
fk=f(bi2de(x(k,:))+1);
tk=(k-1)*T+tks;
s=[s; sin(2*pi*fk*tk)];
end
%pwelch(s);
% add noise to the FSK (passband) signal
s=awgn(s,SNR, 'measured');
%% FSK receiver
%non coherent demodulation
th=rand();
xr=[];
for k=1:length(s)/ns
tk=(k-1)*T+tks;
sk=s((k-1)*ns+1:k*ns);
smi=[];
```

```
for i=1:M
si=sin(2*pi*(f(i)*tk+th));
sq=cos(2*pi*(f(i)*tk+th));
smi=sum(sk.*si);
smq=sum(sk.*sq);
sm(i)=sqrt(smi^2+smq^2);
end
[m,j]=max(sm);
xr=[xr;de2bi(j-1,bps)];
end
% count errors
err=not(x==xr);
errors=sum(sum(err));
end
```

με την κύρια αλλαγή να είναι και η προσθήκη Quadrature συνιστώσας(sq), η οποία θα είναι υπεύθυνη για τα πλάτη στον μιγαδικό άξονα και είναι πρακτικά υπεύθυνη για τις αλλαγές στη φάση, καθώς η in-phase συνιστώσα (si) ήταν υπεύθυνη για τα πλάτη του άξονα πραγματικών τιμών, πρακτικά δηλαδή για τη μεταβολή των συχνοτήτων που θα λαμβάνει ο δέκτης .

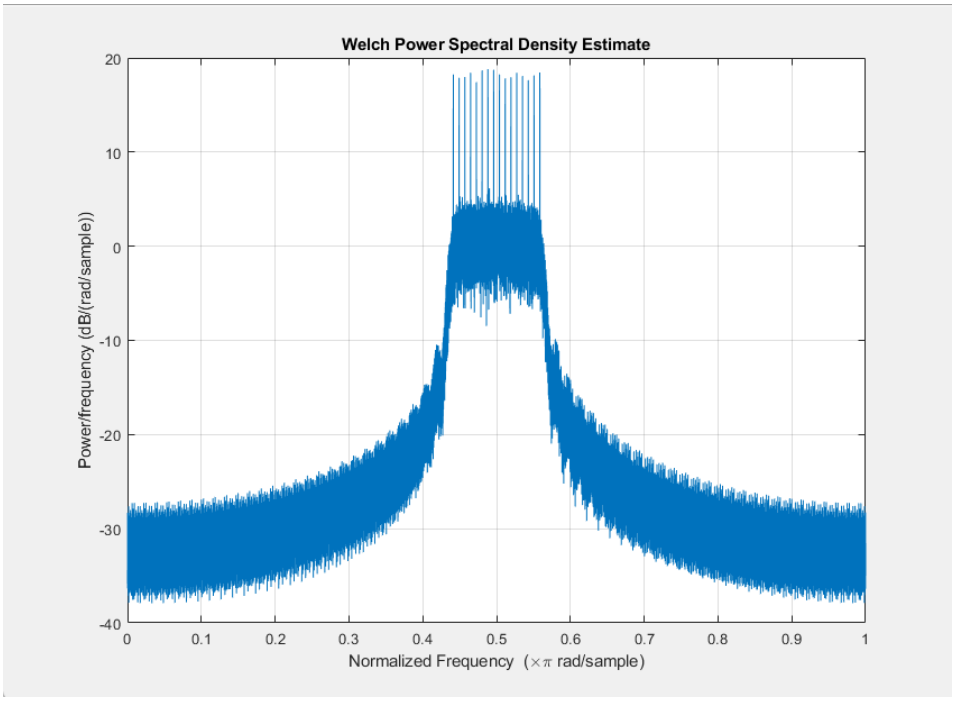
2)

Χρησιμοποιούμε την bertool και σχεδιάζουμε τις θεωρητικές γραφικές για τη 16-FSK σύμφωνη και ασύμφωνη, ενώ στη συνέχεια μέσω της ask_ber_func, χρησιμοποιώντας τις αντίστοιχες συναρτήσεις για σύμφωνη και ασύμφωνη 16-fsk τρέχουμε και τις αντίστοιχες προσομοιώσεις.

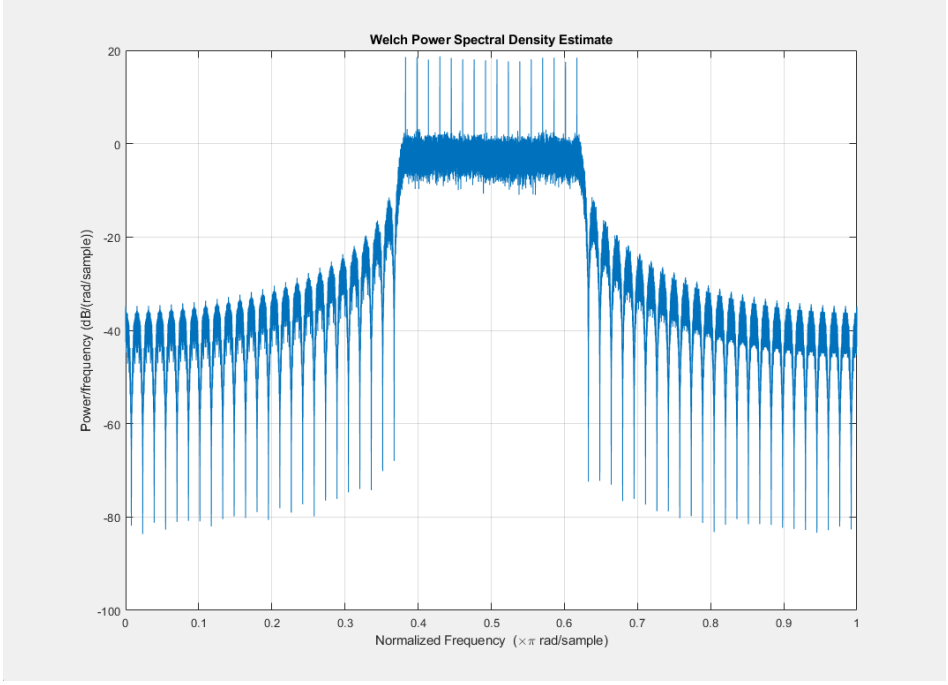


Δεν βάλαμε μεγάλο όριο EbNo, καθώς το ματλαμπ μετά το πέρας της τιμής 8(Db) έκανε υπερβολικά πολλή ώρα για να τρέξει την προσομοίωση.

3) Τα φάσματα των 16-FSK για σύμφωνη και ασύμφωνη FSK.



Σύμφωνη 16-FSK



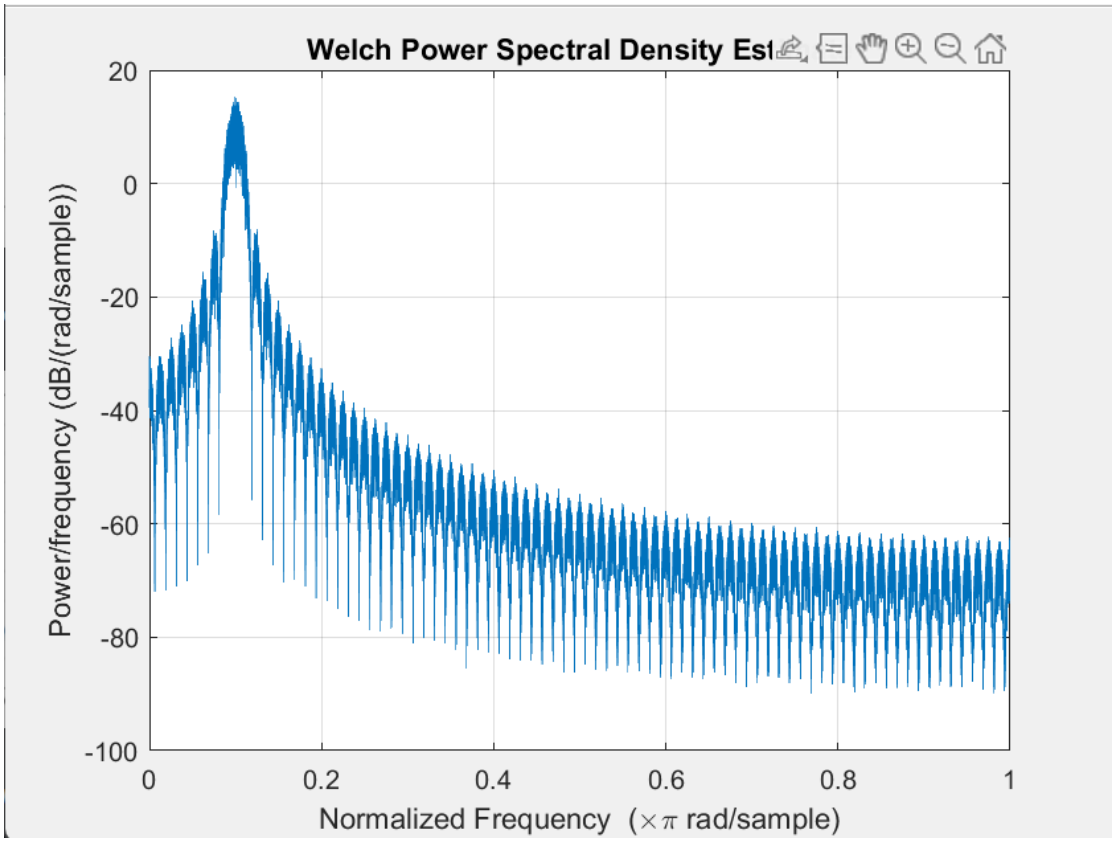
16-FSK ασύμφωνη

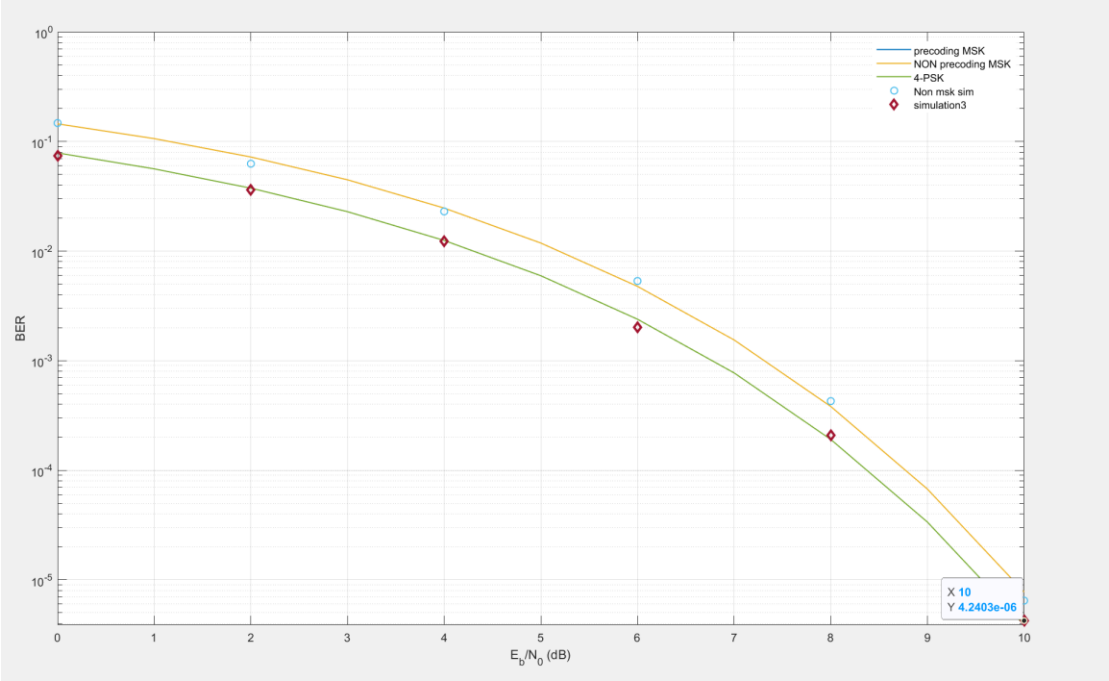
Όπως γνωρίζουμε και από τι θεωρία, η ασύμφωνη θα έχει μεγαλύτερη απόσταση στους λοβούς καθώς η μινιμουμ απόσταση θα είναι 1/T ενώ στη σύμφωνη το ελάχιστο μπιτ ρειτ θα είναι 1/2T .

Τα μπαστουνάκια προκύπτουν από τις κρουστικές που δημιουργούνται λόγω της διαφοράς φάσης που δημιουργείται από το ημίτονο και το συνημίτονο.

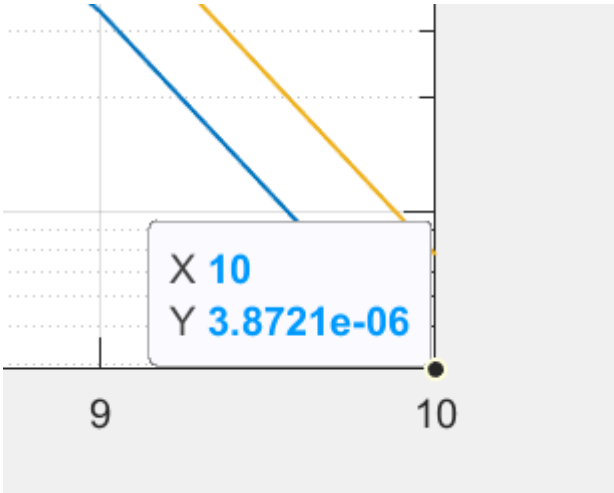
4)

Το σήμα s για τα στοιχεία που μας ζητήθηκαν.





Στο διάγραμμα φαίνεται η τιμή του BER για προ κωδικοποιημένη MSK, η οποία για την προσομοίωση είναι 4.2403*10^(-6)



Η θεωρητική τιμή θα είναι 3.8721*10(-6).

5)

Η QPSK μεταφέρει δύο μπιτ ανά σύμβολο, ενώ η MSK μεταφέρει 1 μπιτ ανά σύμβολο, επομένως το εύρος ζώνης της πρώτης θα είναι μεγαλύτερο από αυτό της δεύτερης. Κάθε σύμβολο στην QPSK θα έχει φάση 0, 90, 180 ή 270 στο μιγαδικό επίπεδο, ενώ η MSK θα έχει φάσεις 0 και 180 συνήθως(και 90 φυσικά)

Η συνάρτηση της msk που χρησιμοποιήθηκε. Περιέχει και προκωδικοποίηση και χωρίς προκωδικοποίηση.

```
function errors=msk_errors(Nbits,nsamp,EbNo)
% Οριζόμενες παράμετροι
%Nbits=2000;nsamp=16;EbNo=6;
n=Nbits; % αριθμός data bits
R=2*10^6; % bit rate
fc=4*R; % φέρουσα συχνότητα
ns=nsamp; % παράγωντας υπερδειγμάτισης
%
% δίαυλος awgn
SNR=EbNo-10*log10(ns/2); % in db
% Παραγώμενες παράμετροι
T=1/R; % περίοδος 1 bit (= βασικη περιοδος)
Ts=T/ns; % Συχνότητα δειγματοληψίας
% ακολουθία εισόδου
y=[1;sign(rand(n-1,1)-0.5)]; % random numbers, -1 Þ 1
%
% προκωδικοποίηση
x(1)=1;
for i=2:length(y)
x(i)=y(i)*y(i-1);
end
x=x';
g=ones(ns,1);
xx=conv(upsample(x,ns),g); % δείγματα παλμοσειράς NRZ polar
% ÷ñïíéêü ðëÝãìá
ts=[0:Ts:length(xx)*Ts]'; % μήκους ns*(n+1)
%
%% ΠΟΜΠΟΣ MSK
xs=xx;
theta=cumsum(xs)*pi/2/ns;
xs_i=cos(theta); % ΣΥΜΦΑΣΙΚΗ ΣΥΝΙΣΤΩΣΑ
xs_i=[xs_i; xs_i(length(xs_i))]; % επέκταση με ενα ακομη δειγμα
xs_q=sin(theta); % εγκαρσια συνιστωσα
xs_q=[xs_q; xs_q(length(xs_q))]; % επέκταση με ενα ακομη δειγμα
% Διαμόρφωση
s=xs_i.*cos(2*pi*fc*ts)-xs_q.*sin(2*pi*fc*ts);
figure(2);
pwelch(s);
%figure;pwelch(s,[],[],fc,1/Ts);
%figure;pwelch(xs,[],[],[],1/Ts);
% προσθηκη θορυβου
s=awgn(s,SNR, 'measured');
%% ΔΕΚΤΗΣ MSK
xs_i=s.*cos(2*pi*fc*ts);
xs_q=-s.*sin(2*pi*fc*ts);
figure(5);
pwelch(xs_i);
figure(6);
pwelch(xs_q);
% Φιλτρο LP (Parks-McClellan)
f1=0.75/ns; f2=4*f1;
order=4*ns;
fpts=[0 f1 f2 1];
mag=[1 1 0 0];
wt=[1 1];
b = firpm(order,fpts,mag,wt);
```

```
a=1;
len=length(xs_i);
dummy=[xs_i;zeros(order,1)];
dummy1=filter(b,a,dummy);
delay=order/2; % δοκιμαστε με delay=0!
xs_i=dummy1(delay+(1:len));
dummy=[xs_q;zeros(order,1)];
dummy1=filter(b,a,dummy);
delay=order/2;
xs_q=dummy1(delay+(1:len));
bi=1; xr_1=1;
for k=1:2:n-1
li=(k*ns+1:(k+2)*ns)';
lq=((k-1)*ns+1:(k+1)*ns)';
xi=xs_i(li);
xq=xs_q(lq);
gmi=cos(pi/2/T*Ts*li); % παλμος matched-filter
gmq=-gmi; % =sin(pi/2/T*Ts*lq);
bi_1=bi;
bi=sign(sum(xi.*gmi));
bq=sign(sum(xq.*gmq));
% χωρις προκωδικοποιηση, αποσχολιαστε τις επομενες δυο γραμμες.
%xr(k)=bi_1*bq;
%xr(k+1)=bi*bq;
% με προκωδικοποιηση, αποσχολιαστε τις επομενες δυο γραμμες.
xr(k)=xr_1*bi_1*bq;
xr(k+1)=xr(k)*bi*bq;
xr_1=xr(k+1);
end
xr=xr';
%err=not(x==xr);
err=not(y==xr); % αποσχολιαστε με προκωδικοποιηση
errors=sum(err);
end
```