

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΨΗΦΙΑΚΕΣ ΕΠΙΚΟΙΝΩΝΙΕΣ Ι

Εργαστηριακή Άσκηση 6

Ονοματεπώνυμο: Μπέτζελος Χρήστος

Α.Μ. : 031 16 067

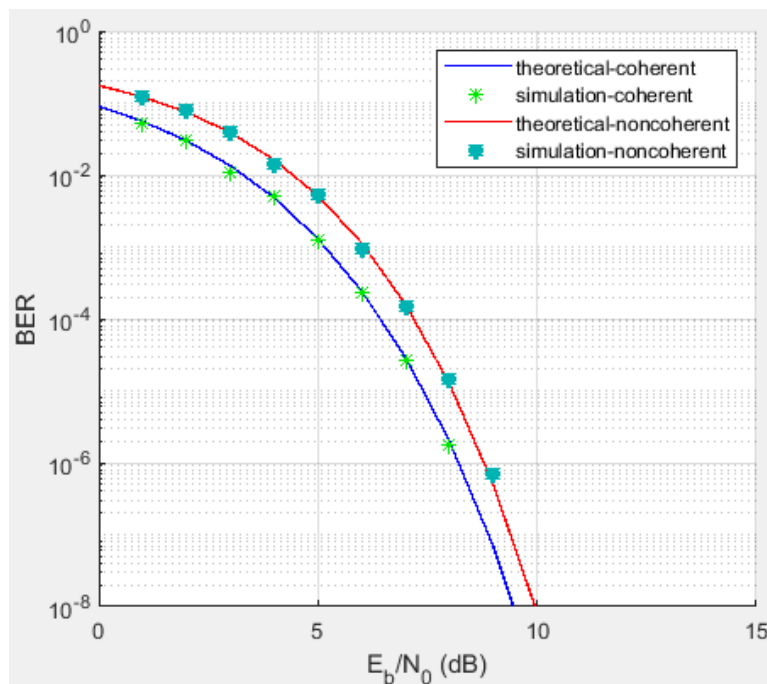
Εξάμηνο: 6^ο

Μέρος 1^ο

Συμπληρώνουμε τον κώδικα 6.1. των σημειώσεων ώστε να εξομοιώνει και την ασύμφωνη FSK. Προσθέσαμε δηλαδή τυχαία φάση στο λαμβανόμενο σήμα, πριν την αποδιαμόρφωση-φώραση. Οι δύο κώδικες των συναρτήσεων σύμφωνης και ασύμφωνης FSK φαίνονται στο παράρτημα στο τέλος της αναφοράς.

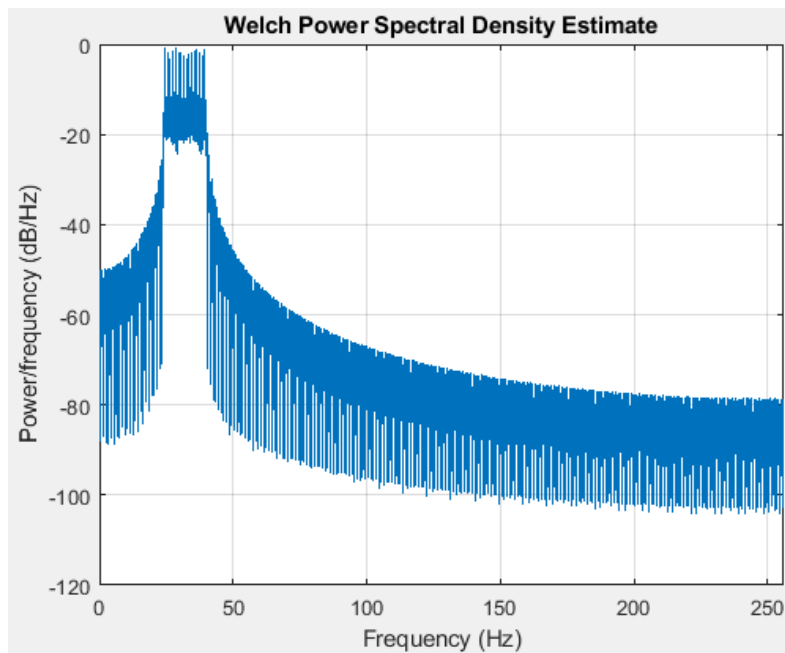
Μέρος 2^ο

Χρησιμοποιώντας τη νέα μας συνάρτηση για εξομοίωση συστήματος 16-FSK σχεδιάζουμε τις καμπύλες $P_b \leftrightarrow E_b/N_0$ για σύμφωνη και ασύμφωνη αποδιαμόρφωση (θεωρητικές και από εξομοίωση). Χρησιμοποιήσαμε το εργαλείο bertool με την γνωστή συνάρτηση `ask_ber_func`, και το διάγραμμα που προκύπτει φαίνεται παρακάτω.



Μέρος 3°

Σχεδιάζουμε το φάσμα του ζωνοπερατού σήματος του ερωτήματος 2 για εξομίωση συστήματος 16-FSK. Το διάγραμμα που προκύπτει φαίνεται παρακάτω.



Παράρτημα-Κώδικας Συναρτήσεων

Σύμφωνη αποδιαμόρφωση

```
function errors=fsk_errors_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([0,1],1,nb);
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,[],[],[],fc);
% add noise to the FSK (passband) signal
s=awgn(s,SNR, 'measured');
% FSK receiver
% coherent demodulation
th=0;
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);
        smi(i)=sum(sk.*si);
    end
    [m,j]=max(smi);
    xr=[xr;de2bi(j-1,bps)];
end
% count errors
err=not(x==xr);
errors=sum(sum(err));
end
```

Ασύμφωνη αποδιαμόρφωση

```
function errors=fsk_errors_noncoh(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; % y=randint(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
% add noise to the FSK (passband) signal
s=awgn(s,SNR, 'measured');
% FSK receiver
% noncoherent demodulation
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
        th=rand();
        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
```