

Πολυτεχνείο Κρήτης

Σχολή ΗΜΜΥ

Τηλεπικοινωνιακά Συστήματα Ι

ΑΣΚΗΣΗ 3

	Φοιτητής 1	Φοιτητής 2
Επώνυμο	Καρόγιαννης	Λιάκος
Όνομα	Ανδρέας	Βασίλειος
A.M.	2019030064	2019030024

**A1.** Για το πρώτο ερώτημα του Α' μέρους της άσκησης (A1), δημιουργήθηκε μια δυαδική ακολουθία με  $4N(N=200)$  ισοπίθανα bits.

```
clear all;  
%%A1  
N=200;  
seq=(sign(randn(4*N,1))+1)/2;
```

Η υλοποίηση του κώδικα

**A.2**

Στην συνέχεια συντάξαμε την ζητούμενη συνάρτηση(που χρησιμοποιεί κωδικοποίηση Gray)

```
function [ Xk ] = bits_to_4PAM( b )  
  
for k=1:2:length(b)/2  
if(b(k)==0 && b(k+1)==0)  
Xk(k)=3;  
elseif(b(k)==0 && b(k+1)==1)  
Xk(k)=1;  
elseif(b(k)==1 && b(k+1)==1)  
Xk(k)=-1;  
elseif(b(k)==1 && b(k+1)==0)  
Xk(k)=-3;  
else  
disp('Error')  
return  
end  
end
```

Η ζητούμενη συνάρτηση

**A3.**

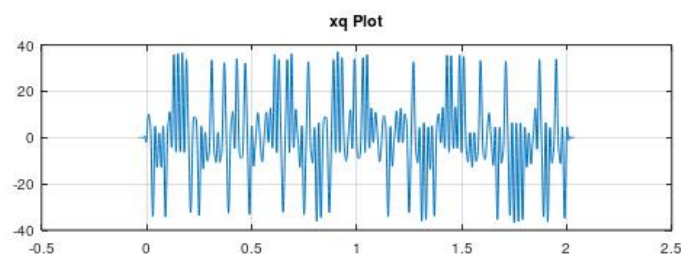
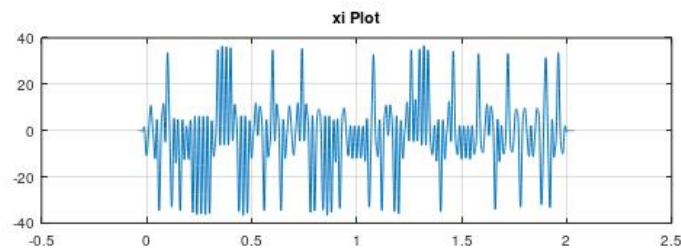
Στην συνέχεια αποθηκεύσαμε τα πρώτα  $2N$  σε μια μεταβλητή και τα επόμενα  $2N$  σε μία άλλη μεταβλητή

```
%%A3
xi=x(1:N);
xq=x(N:2*N-1);
```

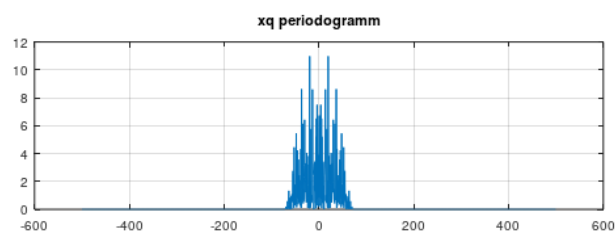
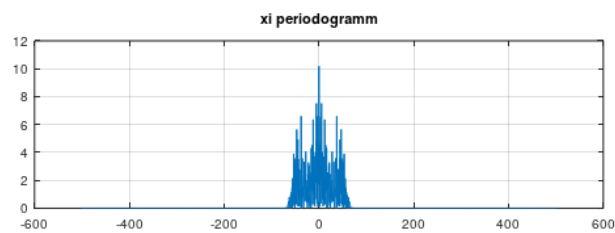
Υλοποίηση κώδικα

**A4.**

Στην συνέχεια περνάω τις παραπάνω μεταβλητές από τα φίλτρα και σχεδιάζω τα αντίστοιχα περιοδογράμματα .



Σχεδίαση του  $x_i$  και  $x_q$



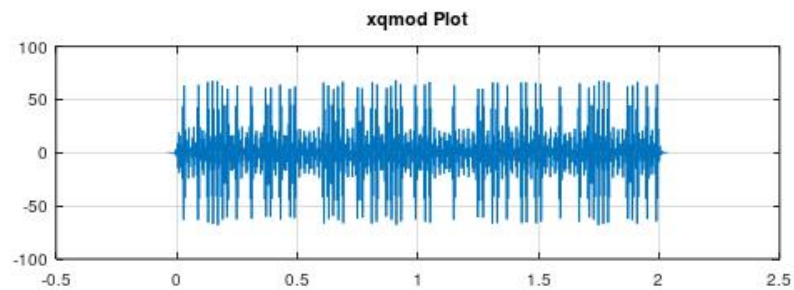
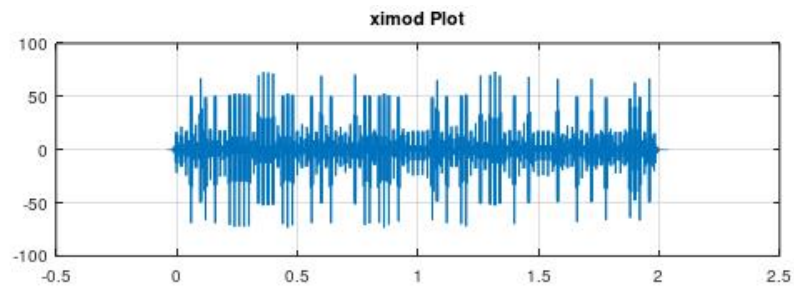
Περιοδόγραμμα  $x_i$  &  $x_q$

### Κώδικας

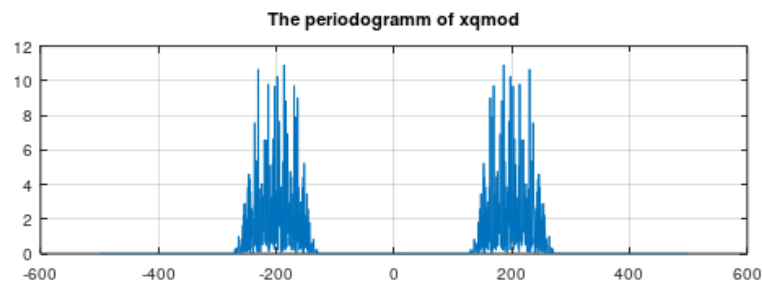
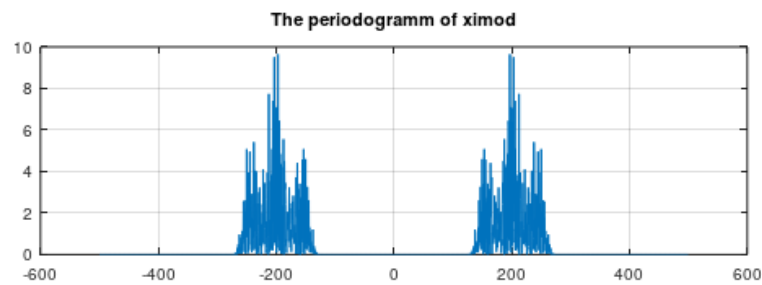
```
[fl,tl]=srrc_pulse(T,over,A,0.5);
xidelta=1/Ts*upsample(xi,over);
xqdelta=1/Ts*upsample(xq,over);
tx=0:Ts:N*T-Ts;
tconv=min(tl)+min(tx):Ts:max(tl)+max(tx)
xifinal=conv(fl,xidelta)*Ts;
xqfinal=conv(fl,xqdelta)*Ts;
figure;
subplot(211);
plot(tconv,xifinal);
grid on;
title('xi Plot');
subplot(212);
plot(tconv,xqfinal);
grid on;
title('xq Plot');
f=[-Fs/2:Fs/Nf:Fs/2-Fs/Nf];
%%xi periodogrammm
fourier_Xi=fft(xifinal,Nf)*Ts;
final_Xi=abs(fftshift(fourier_Xi));
phasmatic_energyXi=final_Xi.^2;
Ttotal=length(tconv)*Ts;
Pi=(phasmatic_energyXi./Ttotal);
%%xq periodogrammm
fourier_Xq=fft(xqfinal,Nf)*Ts;
final_Xq=abs(fftshift(fourier_Xq));
phasmatic_energyXq=final_Xq.^2;
Ttotal=length(tconv)*Ts;
Pq=(phasmatic_energyXq./Ttotal);
figure;
subplot(211);
plot(f,Pi);
grid on;
title('xi periodogrammm');
subplot(212);
plot(f,Pq);
grid on;
title('xq periodogrammm');
```

**A5.**

Στο συγκεκριμένο ερώτημα πολλαπλασιάσαμε τις προηγούμενες συναρτήσεις με τους φορείς και παρατηρήσαμε τις διαφορές.



Σχεδιασμός του ximod και του xqmod



Σχεδιασμός αντίστοιχων περιοδογραμμάτων

Είναι εμφανές ότι στην σχεδίαση το πλάτος είναι μεγαλύτερο καθώς έχει πολλαπλασιαστεί με τον φορέα ενώ στο περιοδόγραμμα βρίσκεται γύρω από την συχνότητα  $F_0$  μας όπως ξέρουμε από την θεωρία και υπάρχει συμμετρία στο 0.

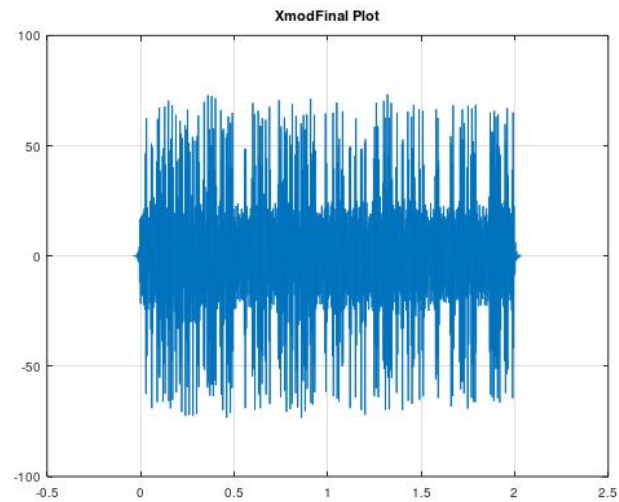
## Κώδικας

```
%%A5
fo=200;
ximod=2*xifinal.*cos(2*pi*fo*tconv);
xqmod=-2*xqfinal.*sin(2*pi*fo*tconv);
figure;
subplot(211);
plot(tconv,ximod);
grid on;
title('ximod Plot');
subplot(212);
plot(tconv,xqmod);
grid on;
title('xqmod Plot');
%%ximod periodogramm
fourier_Ximod=fft(ximod,Nf)*Ts;
final_Ximod=abs(fftshift(fourier_Ximod));
phasmatic_energyXimod=final_Ximod.^2;
Ttotal=length(tconv)*Ts;
Pimod=(phasmatic_energyXimod./Ttotal);

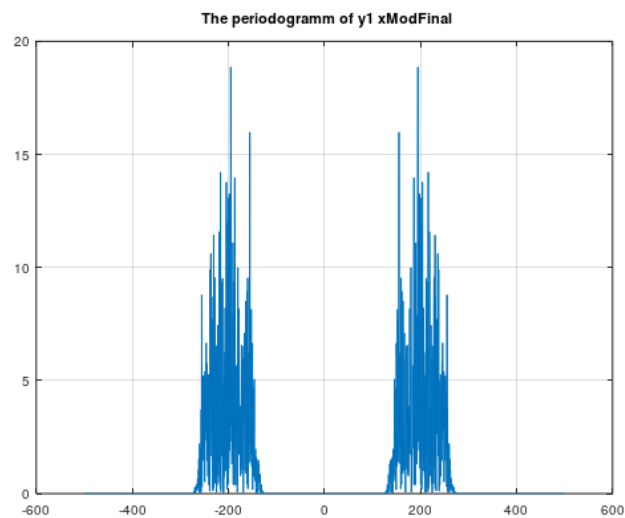
%%xqmod periodogramm
fourier_Xqmod=fft(xqmod,Nf)*Ts;
final_Xqmod=abs(fftshift(fourier_Xqmod));
phasmatic_energyXqmod=final_Xqmod.^2;
Ttotal=length(tconv)*Ts;
Pqmod=(phasmatic_energyXqmod./Ttotal);
figure;
subplot(211);
plot(f,Pimod);
grid on;
title(' The periodogramm of ximod ');
subplot(212);
plot(f,Pqmod);
grid on;
title('The periodogramm of xqmod');
```

**A6.**

Στην συνέχεια σχεδιάσαμε την είσοδο του καναλιού και πάλι βρίσκουμε το περιοδόγραμμα



Σχεδίαση της τελικής εισόδου του καναλιού



Σχεδιασμός του περιοδογράμματος

Παρατηρώ και πάλι ότι το περιοδόγραμμα έχει συμμετρία ως προς το 0 και βρίσκεται και πάλι γύρω από την συχνότητα 200.

### Κώδικας

```
%%A6
xmodfinal=ximod+xqmod;
figure;
plot(tconv,xmodfinal);
grid on;
title('XmodFinal Plot');
%%xqmod periodogramm
fourier_Xmodfinal=fft(xmodfinal,Nf)*Ts;
final_Xmodfinal=abs(fftshift(fourier_Xmodfinal));
phasmatic_energyXmodfinal=final_Xmodfinal.^2;
Ttotal=length(tconv)*Ts;
Pmodfinal=(phasmatic_energyXmodfinal./Ttotal);
figure;
plot(f,Pmodfinal);
grid on;
title(' The periodogramm of y1 xModFinal');
```

**A7.**

Όταν το κανάλι είναι ιδανικό σημαίνει πως η έξοδος είναι αναλλοίωτη οπότε μένει το ίδιο

**A8.**

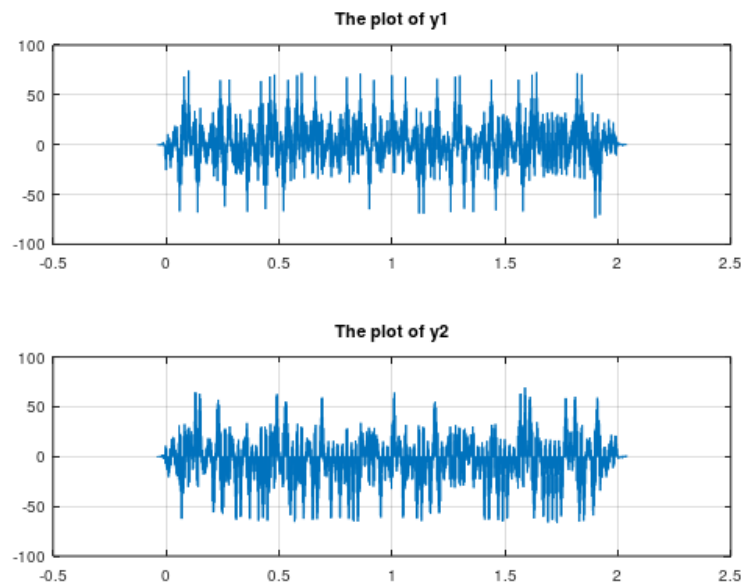
Στο συγκεκριμένο ερώτημα προσθέσαμε θόρυβο στην έξοδο του καναλιού

#### Κώδικας

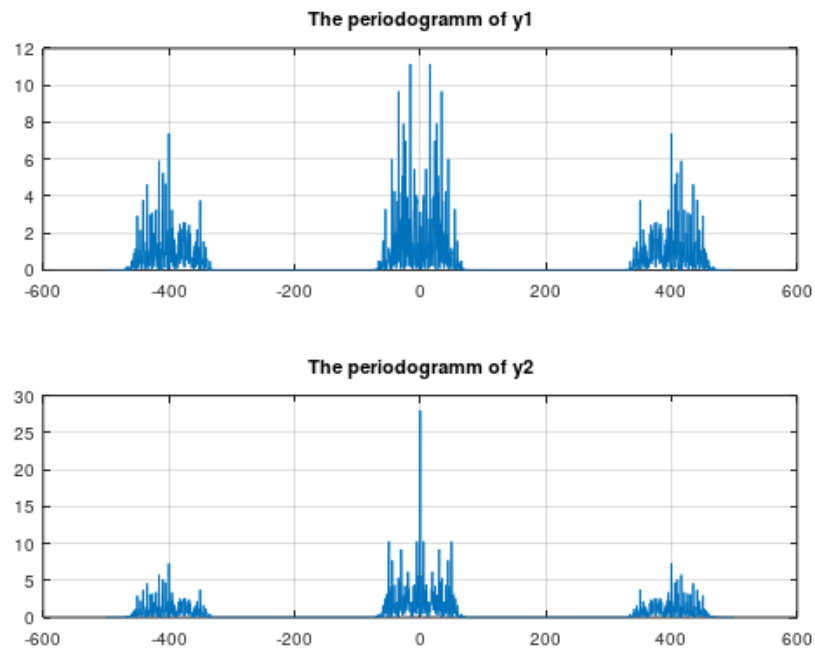
```
%%A7 xmodfinal plot is gonna stay the same
%%A8
snr=20;
varw=((10*A^2)./(Ts*10^(snr/10)));
varn=Ts*varw/2;
%%Creating the wt from theory
wt=sqrt(varn)*randn(1,length(xmodfinal));
y=xmodfinal+wt;
```

**A9.**

Στην συνέχεια με το σήμα που δημιουργήσαμε από την πρόσθεση του θορύβου το διακλαδώνουμε και σχεδιάζουμε το περιοδόγραμμά του.



Σχεδίαση των δύο διακλαδώσεων



Σχεδιασμός των περιοδογραμμάτων

Παρατηρούμε ότι το περιοδόγραμμα βρίσκεται κοντά στο 0 που αυτό περιμέναμε ουσιαστικά αλλά υπάρχουν γραφικές και στις συχνότητες -400 και 400 που ουσιαστικά μας είναι ανεπιθύμητες.

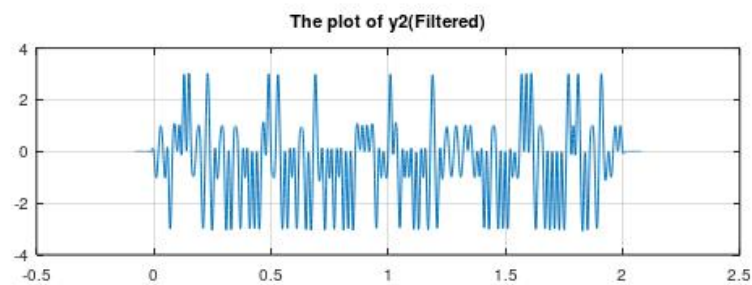
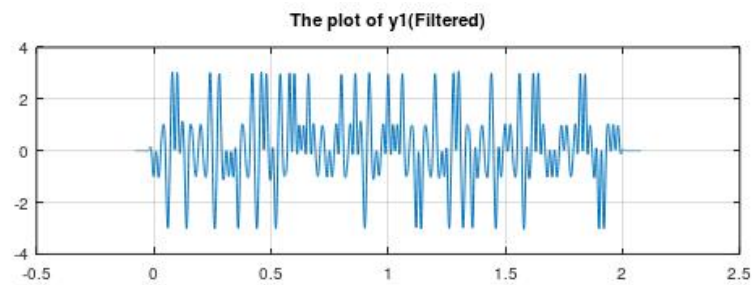
### Κώδικας

```
%%A9
%%making 2 different functions(diakladosi)
y1=y.*cos(2*pi*fo.*tconv);
y2=y.*(-sin(2*pi*fo.*tconv));
figure;
subplot(211);
plot(tconv,y1);
grid on;
title('The plot of y1');
subplot(212);
plot(tconv,y2);
grid on;
title('The plot of y2');
%%y1 periodogramm
fourier_y1=fft(y1,Nf)*Ts;
final_y1=abs(fftshift(fourier_y1));
phasmatic_energyyl=final_y1.^2;
Ttotal=length(tconv)*Ts;
Pyl=(phasmatic_energyyl./Ttotal);
%%y2 periodogramm
fourier_y2=fft(y2,Nf)*Ts;
final_y2=abs(fftshift(fourier_y2));
phasmatic_energyy2=final_y2.^2;
Ttotal=length(tconv)*Ts;
Py2=(phasmatic_energyy2./Ttotal);
figure;
subplot(211);
plot(f,Pyl);
grid on;
title('The periodogramm of y1');
subplot(212);
plot(f,Py2);
grid on;
title('The periodogramm of y2');
```

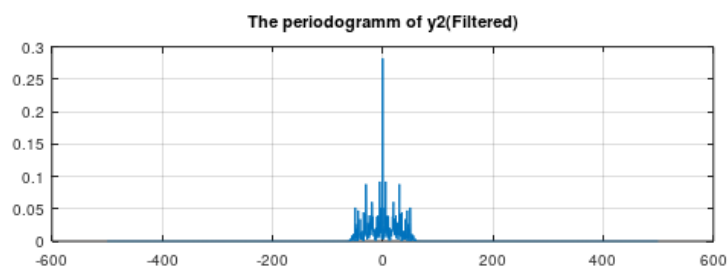
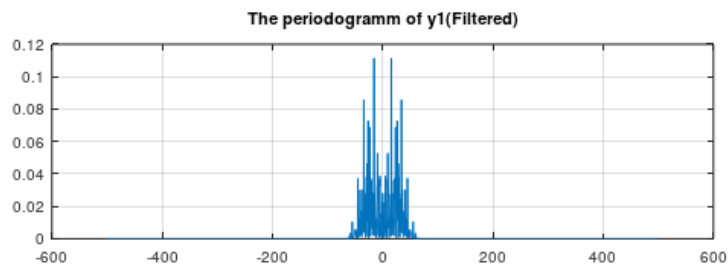


**A10.**

Στην συνέχεια τα περνάμε από τα φίλτρα



Σχεδιασμός των φιλτραρισμένων σημάτων



Περιοδογράμματα των φιλτραρισμένων σημάτων

Παρατηρούμε ότι τώρα εξαφανίστηκαν οι όροι στα -400 και 400 Hz κρατώντας μόνο αυτό που χρειαζόμαστε που είναι κοντά στο 0 .

### Κώδικας

```
%%A10
filteryl=conv(y1,f1)*Ts;
filtery2=conv(y2,f1)*Ts;
tconvnew=tconv(1)+t1(1):Ts:tconv(end)+t1(end);
figure;
subplot(211);
plot(tconvnew,filteryl);
grid on;
title('The plot of y1(Filtered)');
subplot(212);
plot(tconvnew,filtery2);
grid on;
title('The plot of y2(Filtered)');
%%filteryl periodogramm
fourier_filteryl=fft(filteryl,Nf)*Ts;
final_filteryl=abs(fftshift(fourier_filteryl));
phasmatic_energyfilteryl=final_filteryl.^2;
Ttotal=length(tconvnew)*Ts;
Pfilteryl=(phasmatic_energyfilteryl./Ttotal);
%%filtery2 periodogramm
fourier_filtery2=fft(filtery2,Nf)*Ts;
final_filtery2=abs(fftshift(fourier_filtery2));
phasmatic_energyfiltery2=final_filtery2.^2;
Ttotal=length(tconvnew)*Ts;
Pfiltery2=(phasmatic_energyfiltery2./Ttotal);
figure;
subplot(211);
plot(f,Pfilteryl);
grid on;
title('The periodogramm of y1(Filtered)');
subplot(212);
plot(f,Pfiltery2);
grid on;
title('The periodogramm of y2(Filtered)');
```

### A11.

Στην συνέχεια δειγματοληπτούμε κατάλληλα ...

### Κώδικας

```
%%A11
Y=zeros(2,round((length(filteryl)-4*A*Fs)/over));
i=1;
for p=2*A*Fs:Fs:(length(filteryl)-1)-2*A*Fs
Y(1,i)=filteryl(p);
Y(2,i)=filtery2(p);
i=i+1;
end
scatterplot(Y);
```

A12.

### Κώδικας

```
%%setting variables
X0=(-3)*A;
X1=(-1)*A;
X2=A;
X3=3*A;
%%FOR LOOP
for p=1:length(Y)
d1=sqrt((Y(:,p)-X0)^2 + (Y(:,p)-X0)^2);
d2=sqrt((Y(:,p)-X1)^2 + (Y(:,p)-X1)^2);
d3=sqrt((Y(:,p)-X2)^2 + (Y(:,p)-X2)^2);
d4=sqrt((Y(:,p)-X3)^2 + (Y(:,p)-X3)^2);
%find the minimum distance
if (d1==min([d1,d2,d3,d4]))
est_X(:,p) = X0;
elseif (d2==min([d1,d2,d3,d4]))
est_X(:,p) = X1;
elseif (d3==min([d1,d2,d3,d4]))
est_X(:,p) = X2;
elseif (d4==min([d1,d2,d3,d4]))
est_X(:,p) = X3;
end
```

A13.

errornum	double	1x1	200
%%A13 ERRORS			
errornum=0;			
for p=1:length(xi)			
if (est_XI(p)~=xi(p)    est_XQ(p)~=xq(p))			
errornum =errornum+1;			
end			
end			

A14.

```
%%A14(function) prospathisa na ta valo xorista kai na prostheso
%%alla petage out of bounds exception
%%est_bit1=PAM_4_to_bits(est_XI,A);
%%est_bit2=PAM_4_to_bits(est_XQ,A);
%%est_bit=est_bit1+est_bit2;
est_bit=PAM_4_to_bits([est_XI,est_XQ],A);
```

A15.

biterrors	double	1x1	385
<pre>%%A15 biterrors=0; for p=1:length(seq) if(est_bit(p)~=seq(p)) biterrors = biterrors + 1; end end</pre>			

## Β ΜΕΡΟΣ

Στο δεύτερο κομμάτι έπρεπε να κάνουμε ένα μεγάλο for loop(200 φορές) στο οποίο μέσα θα είχαμε βάλει όλες τις συναρτήσεις που είχαμε υπολογίσει στο πρώτο μέρος. Έπειτα δημιουργήσαμε μια μεταβλητή στην οποία τοποθετούσαμε τα συνολικά λάθη τόσο συμβόλου αλλά και bit κάθε φορά που έτρεχε το πρόγραμμά μας. Στο τέλος αφού διαιρέσαμε το συνολικό αριθμό των σφαλμάτων με το κατάλληλο νούμερο(αντίστοιχα για τα σύμβολα και τα bits)υπολογίσαμε τα θεωρητικά νούμερα(με την βοήθεια της q,το πήρα μέσα από την συνάρτηση γιατί μου έβγαζε ένα πρόβλημα) και τα βάζω σε semilogy.

### Κώδικας

```
total_errors=0;
total_errors=0;
c=1;
for SNRdb=0:2:16
for p=1:200
seq=(sign(randn(4*N,1))+1)/2;
x=bits_to_4PAM(seq, 4);
xi=x(1:N);
xq=x(N:2*N-1);
A=4;
T=0.01;
over=10;
Ts=T/over;
Fs=1/Ts;
Nf=2048;
[f1,t1]=srrc_pulse(T,over,A,0.5);
xidelta=1/Ts*upsample(xi,over);
xqdelta=1/Ts*upsample(xq,over);
tx=0:Ts:N*T-Ts;
tconv=min(t1)+min(tx):Ts:max(t1)+max(tx);
xifinal=conv(f1,xidelta)*Ts;
xqfinal=conv(f1,xqdelta)*Ts;
fo=200;
ximod=2*xifinal.*cos(2*pi*fo*tconv);
xqmod=-2*xqfinal.*sin(2*pi*fo*tconv);
xmodfinal=ximod+xqmod;
snr=20;
varw=((10*A^2)./(Ts*10^(snr/10)));
varn=Ts*varw/2;
```

```

%%Creating the wt from theory
wt=sqrt(varn)*randn(1,length(xmodfinal));
y=xmodfinal+wt;
y1=y.*cos(2*pi*fo.*tconv);
y2=y.*(-sin(2*pi*fo.*tconv));
filteryl=conv(y1,f1)*Ts;
filtery2=conv(y2,f1)*Ts;
Y=zeros(2,round((length(filteryl)-4*A*Fs)/over));
i=1;
for p=2*A*Fs:Fs:(length(filteryl)-1)-2*A*Fs
Y(1,i)=filteryl(p);
Y(2,i)=filtery2(p);
i=i+1;
end
%%A12
est_XI=detect_4_PAM(xi,A);
est_XQ=detect_4_PAM(xq,A);
%%A13 ERRORS
errornum=0;
for p=1:length(xi)
if(est_XI(p)~=xi(p) || est_XQ(p)~=xq(p))
errornum =errornum+1;
end
end
end

```

---

```

%%A14(function) prospathisa na ta valo xorista kai na prost
%%alla petage out of bounds exception
%%est_bit1=PAM_4_to_bits(est_XI,A);
%%est_bit2=PAM_4_to_bits(est_XQ,A);
%%est_bit=est_bit1+est_bit2;
est_bit=PAM_4_to_bits([est_XI,est_XQ],A);
%%A15
biterrors=0;
for p=1:length(seq)
if(est_bit(p)~=seq(p))
biterrors = biterrors + 1;
end
end
total_serrors=total_serrors+errornum;
total_berrors=total_berrors+biterrors;
end
Exp_serrors(c)=total_serrors/(N*K);
Exp_berrors(c)=total_berrors/(4*N*K);
theorySer(c)= 0.5 * erfc( x / sqrt(2) );
theoryBer(c)=(1/bps)*3/2 .*theorySer(c);
total_serrors=0;
total_berrors=0;
c=c+1;
end

```

---

```

figure;
semilogy(SNRdb_vector,Exp_serrors);
hold on;
semilogy(SNRdb_vector,The_serrors);
hold off;
figure;
semilogy(SNRdb_vector,Exp_berrors);
hold on;
semilogy(SNRdb_vector,The_berrors);
hold off;

```

---

## ΓΡΑΦΙΚΕΣ

