
Πολυτεχνείο Κρήτης

Σχολή ΗΜΜΥ

Τηλεπικοινωνιακά Συστήματα Ι

Παράδοση 3ης εργασίας

Ημερομηνία Παράδοσης: 10 Ιανουαρίου 2022

Μονάδες 130/1000

Επώνυμο	Παυλόπουλος
Όνομα	Χρήστος
ΑΜ	2018030139

Μέρος Α

1) Δημιουργήθηκε δυαδική ακολουθία με στοιχεία $4N$ ισοπίθανα bits

```
%% A1
%sxediasmos dyadikhs akolouthias me 4*N bits
b = (sign(randn(4*N,1))+1)/2;
```

2) Σχεδιάστηκε η παρακάτω συνάρτηση function $X = \text{bits_to_4_PAM}(\text{bit_seq}, A)$, η οποία χρησιμοποιώντας κωδικοποίηση Gray απεικονίζει την δυαδική ακολουθία 4PAM συμβόλων.

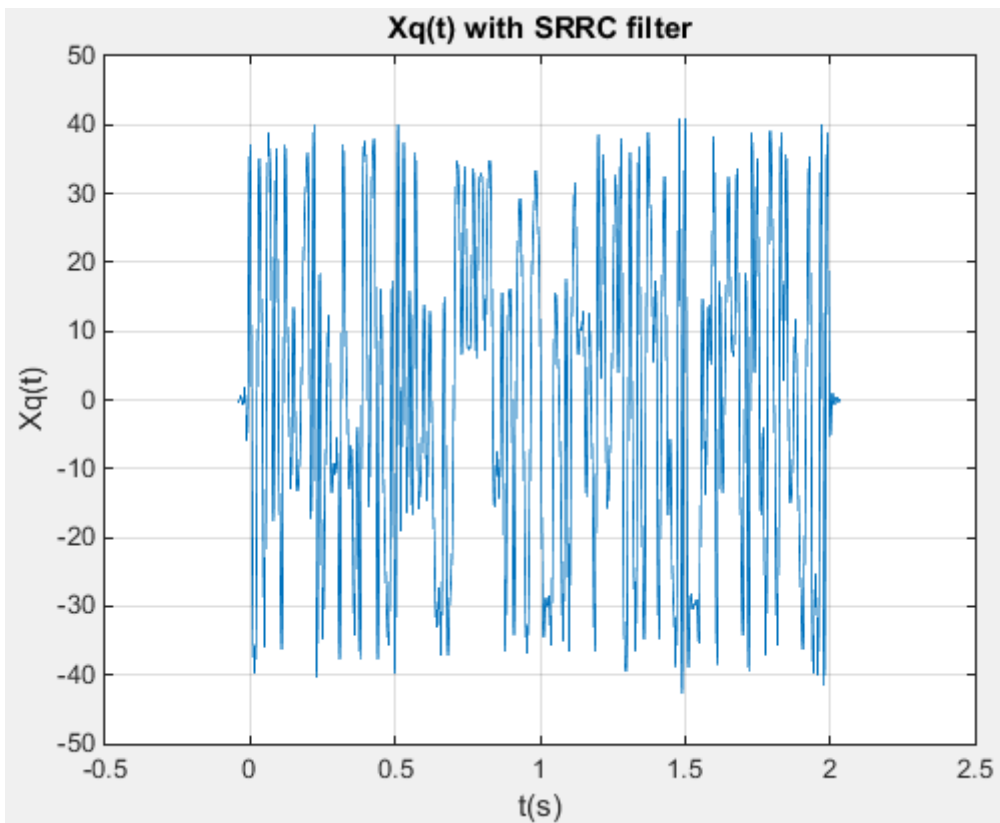
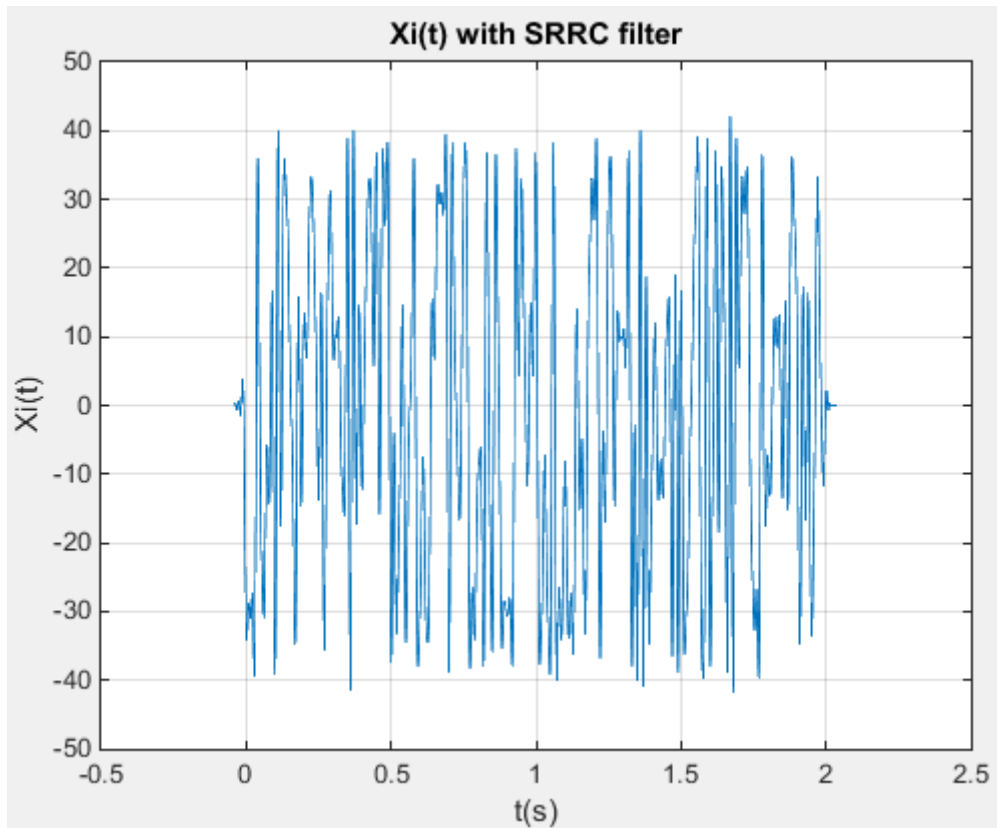
```
function X = bits_to_4_PAM(bit_seq, A)
j=1;
for i=1:2:length(bit_seq)-1
    if (bit_seq(i)==0 && bit_seq(i+1)==0)
        X(j)=3*A;
    elseif (bit_seq(i)==0 && bit_seq(i+1)==1)
        X(j)=A;
    elseif (bit_seq(i)==1 && bit_seq(i+1)==1)
        X(j)=-A;
    else
        X(j)=-3*A;
    end
    j=j+1;
end
end
```

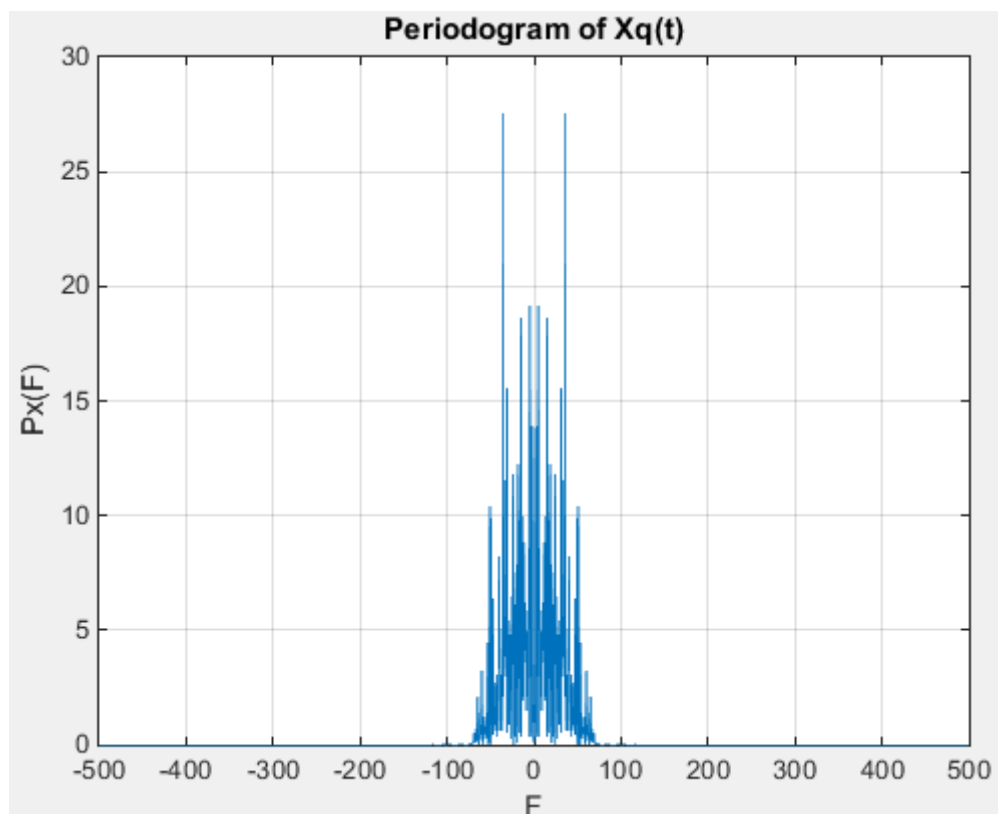
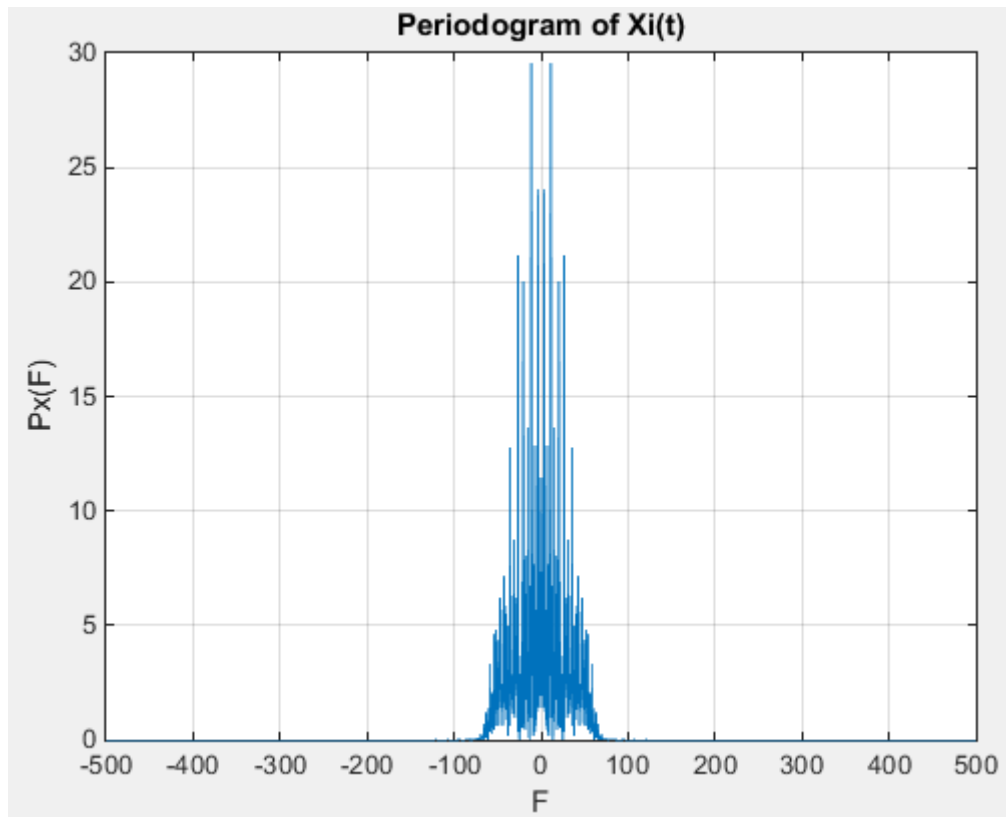
3) Απεικονίστηκαν τα πρώτα $2N$ bits της ακολουθίας στα 4PAM σύμβολα $X_{I,n}$ και τα επόμενα $2N$ bits στα 4PAM σύμβολα $X_{Q,n}$

```
bit_seq1=b(1:2*N); %prwta 2N bits
bit_seq2=b(2*N+1:4*N); %epomena 2N bits

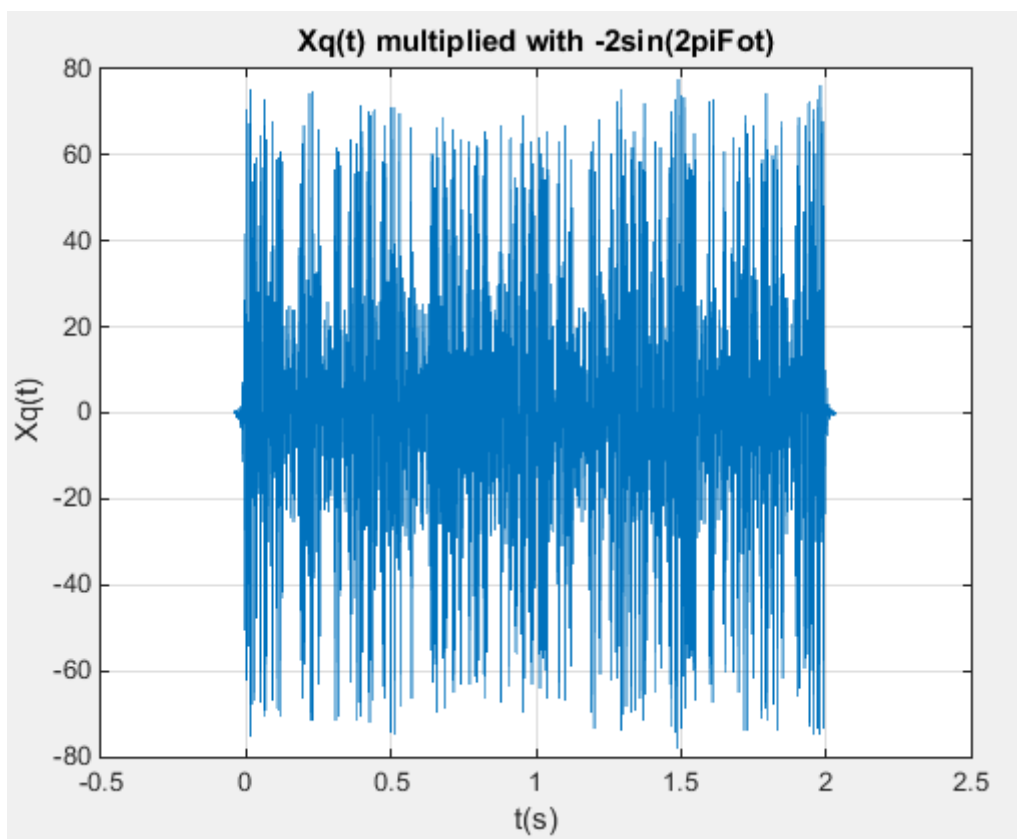
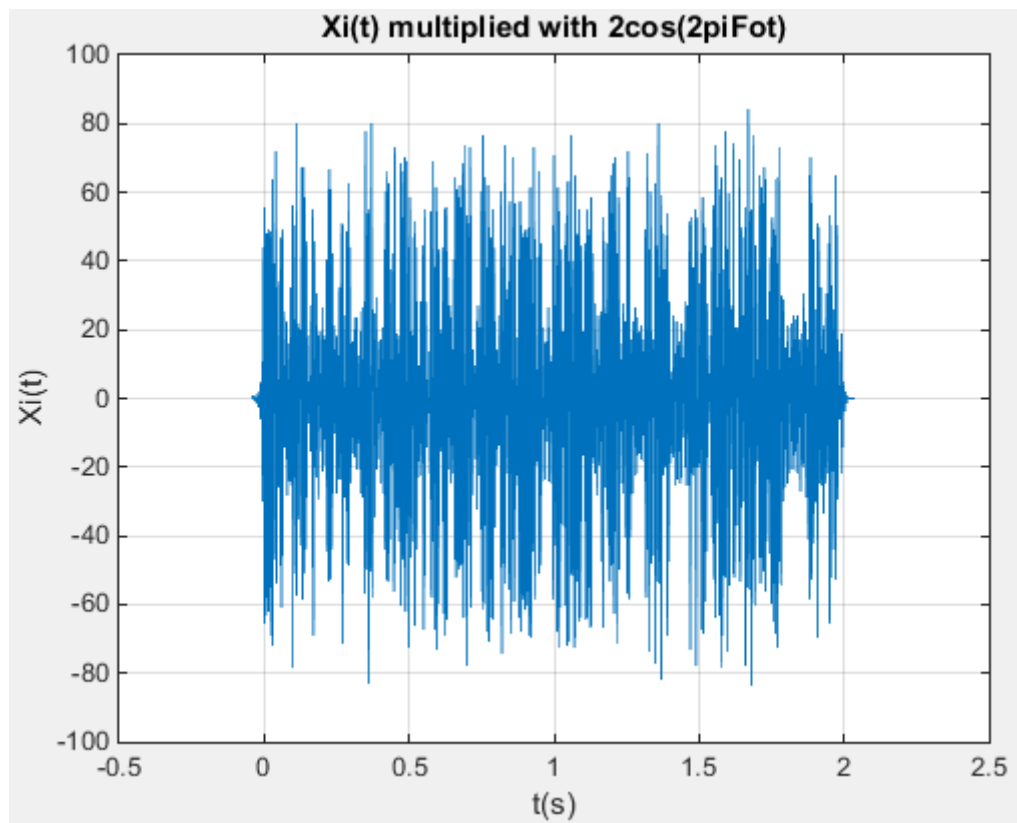
%apeikonish se 4PAM symbola
Xi_n=bits_to_4_PAM(bit_seq1,A);
Xq_n=bits_to_4_PAM(bit_seq2,A);
```

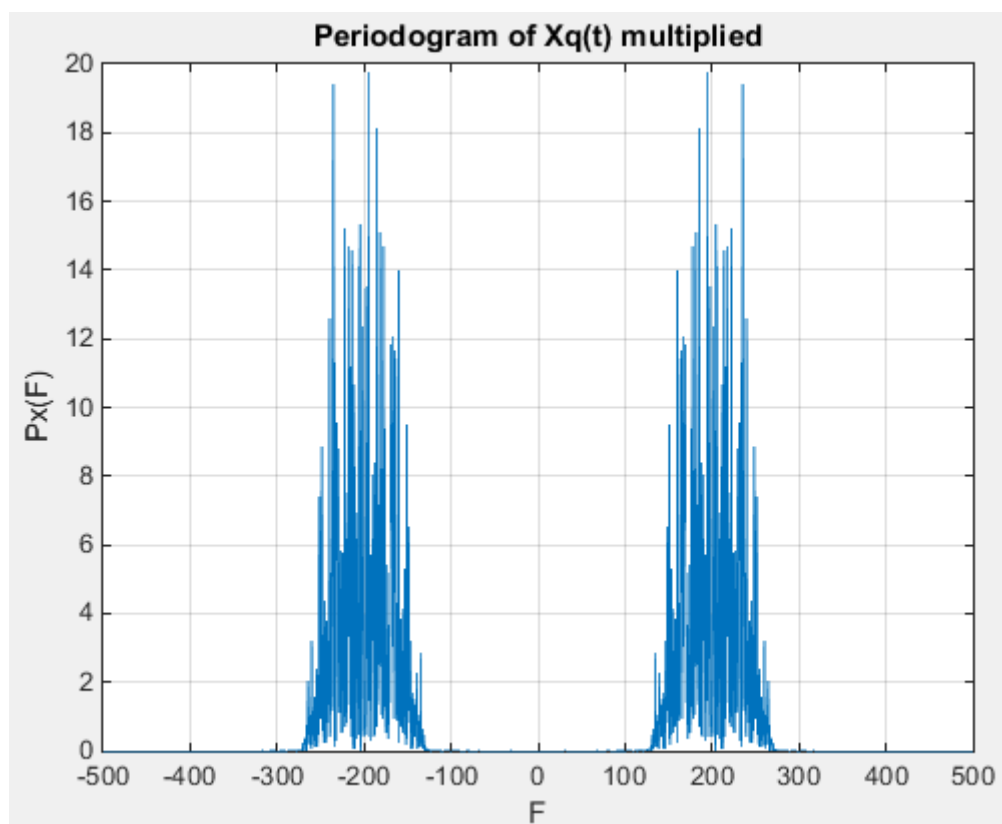
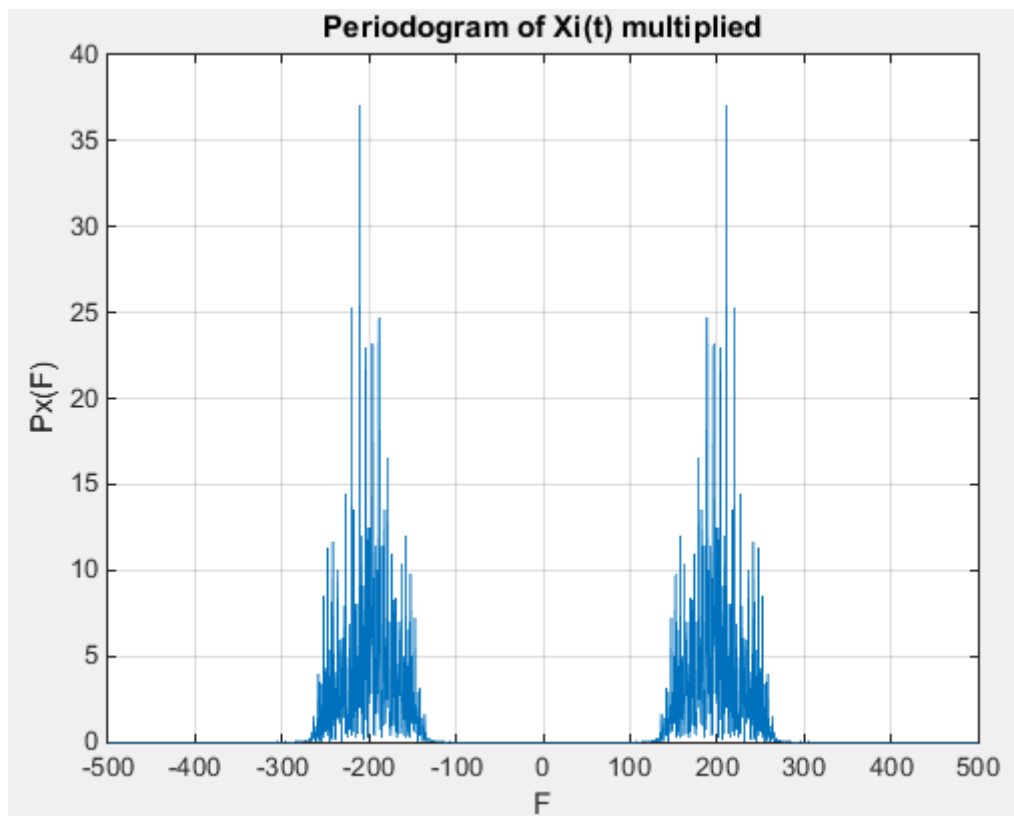
4) Θέτω $T=0.01\text{sec}$, $\text{over} = 10$, $T_s = T/\text{over}$. Περάστηκαν οι ακολουθίες από φίλτρο SRRC, σχηματίστηκαν και απεικονίστηκαν οι κυματομορφές εξόδου και τα περιοδογράμματα $X_i(t)$ και $X_q(t)$.



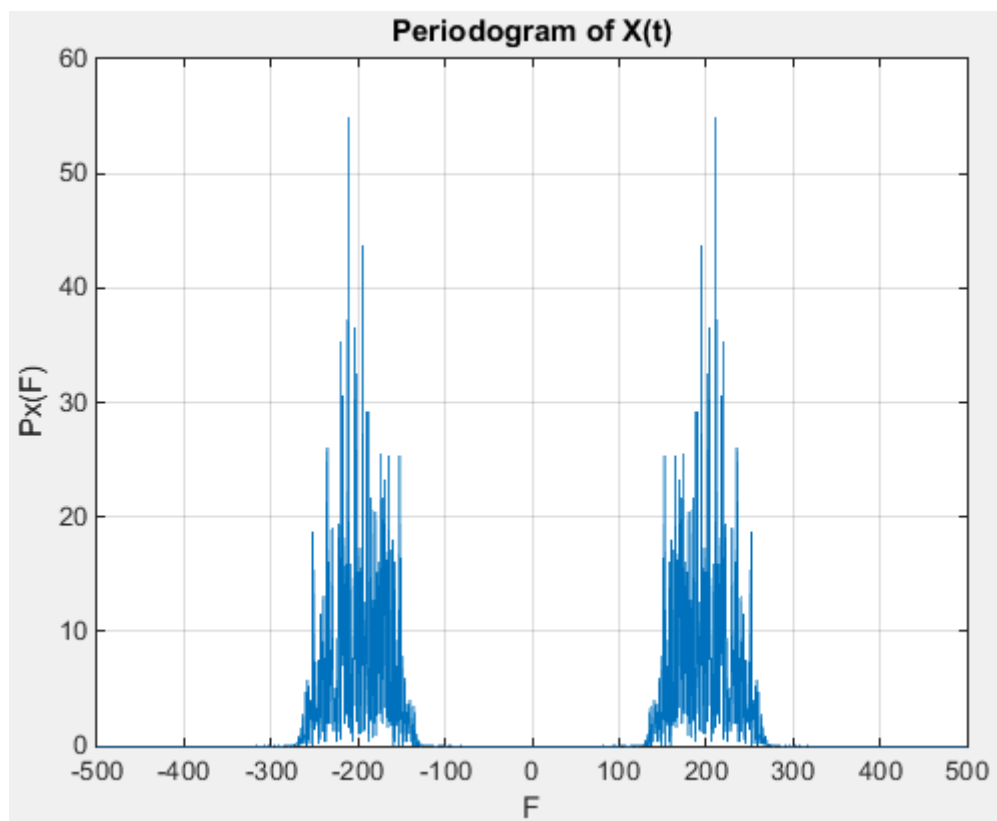
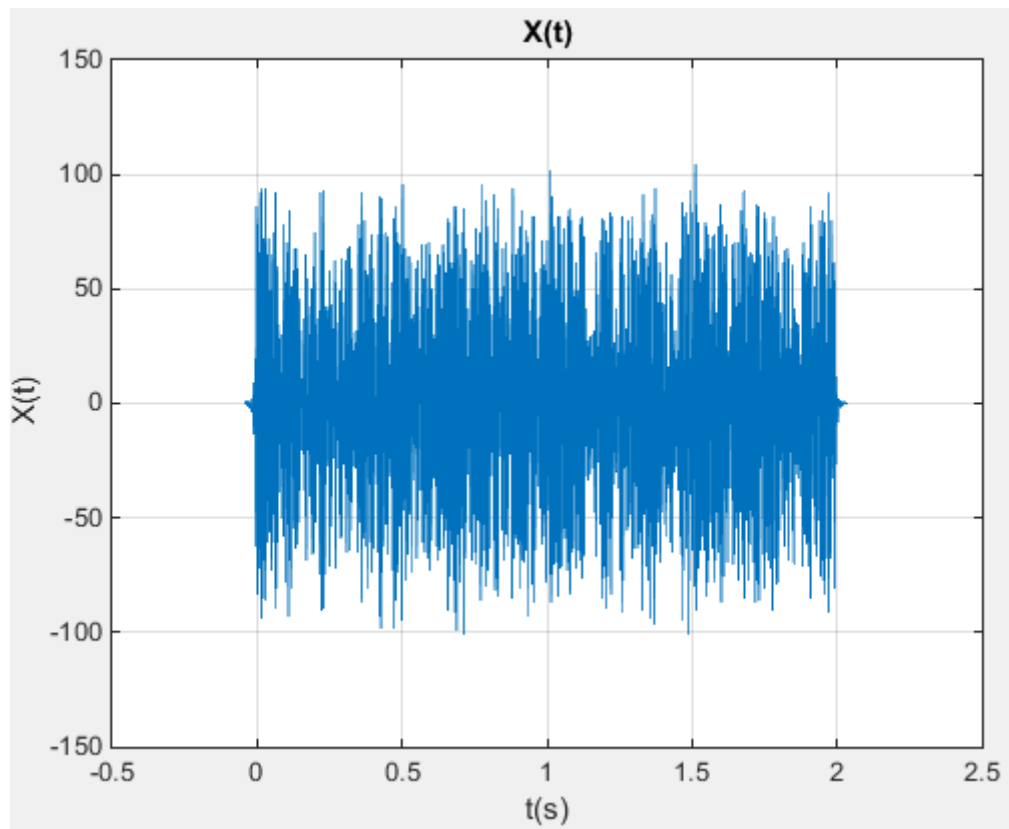


5) Πολλαπλασιάστηκαν οι κυματομορφές $X_i(t)$, $X_q(t)$ με τους φορείς που αναφέρονται στην εκφώνηση και με συχνότητα $F_0 = 200\text{Hz}$. Παρατηρούμε, στα περιοδογράμματα, διαμόρφωση γύρω από την συχνότητα που θέσαμε.





6) Σχεδιάστηκε η συνάρτηση $X^{\text{mod}}(t) = X_i^{\text{mod}}(t) + X_q^{\text{mod}}(t)$. Παρατηρούμε πως το πλάτος έχει αυξηθεί καθώς υπάρχει πρόσθεση δύο κυματομορφών.

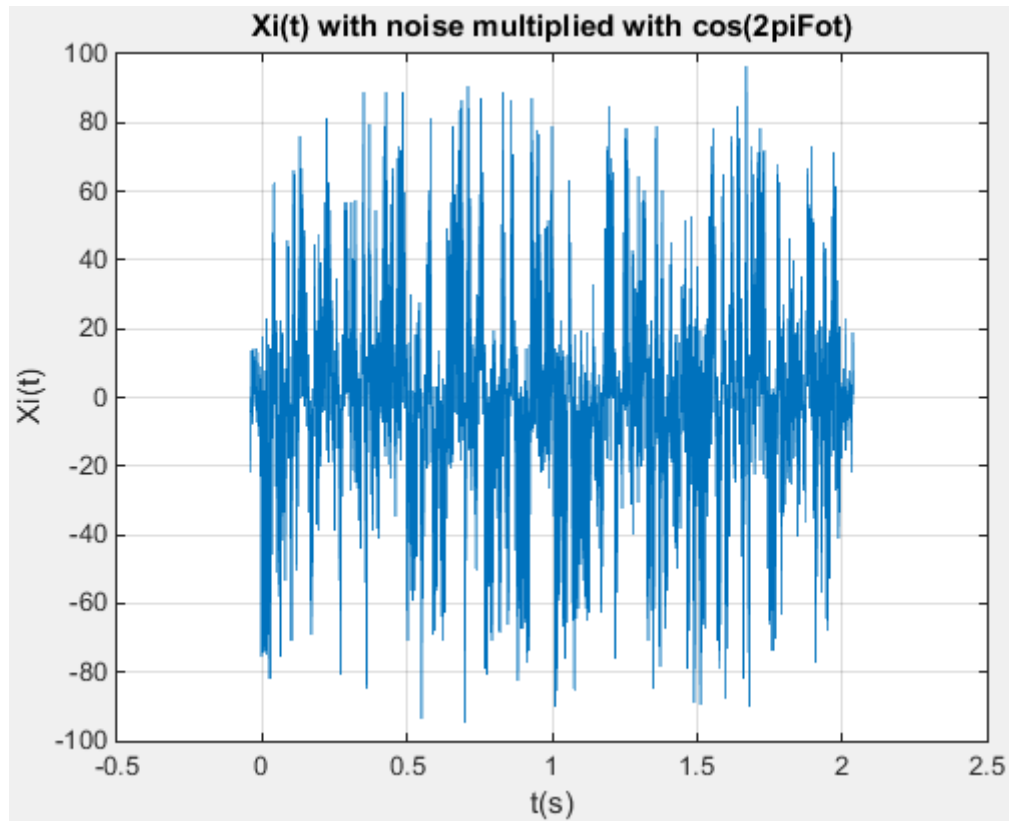


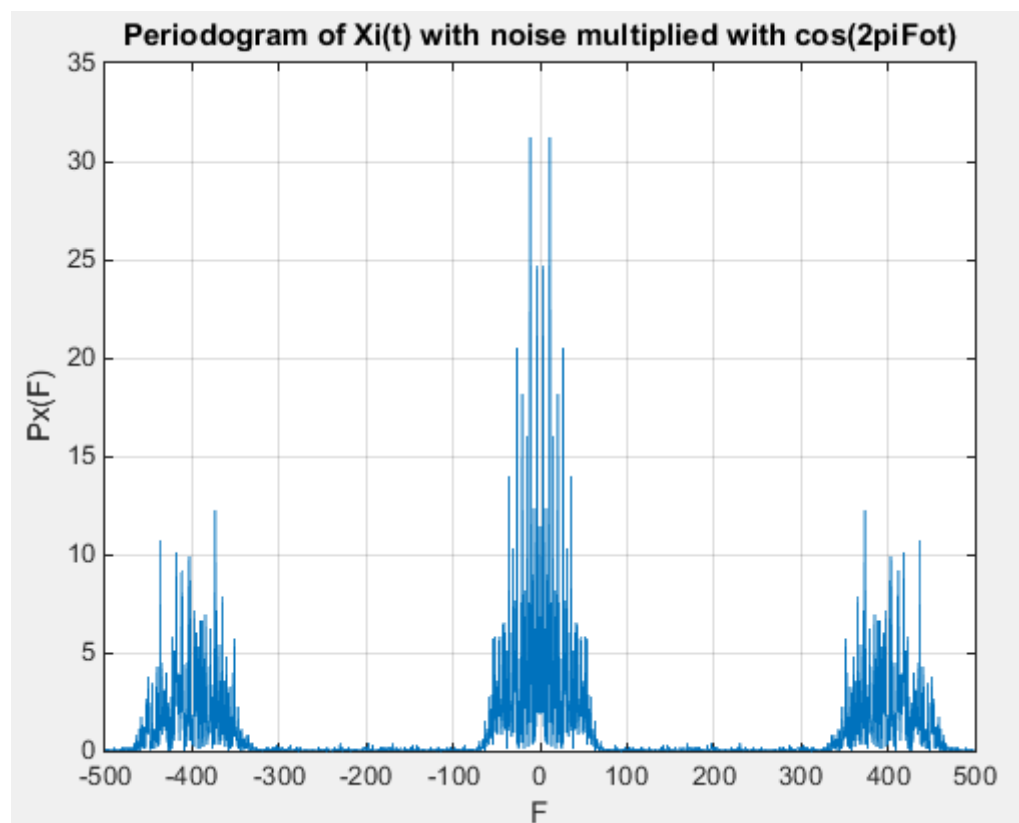
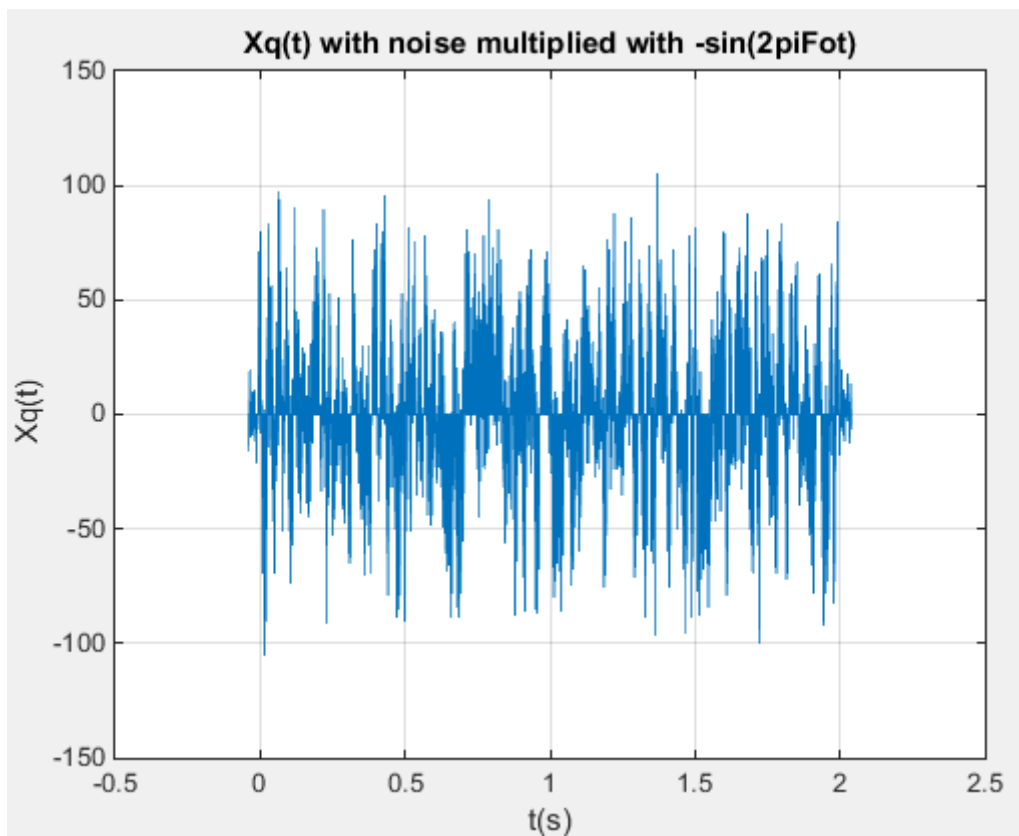
7) Θεωρούμε πως το κανάλι είναι ιδανικό

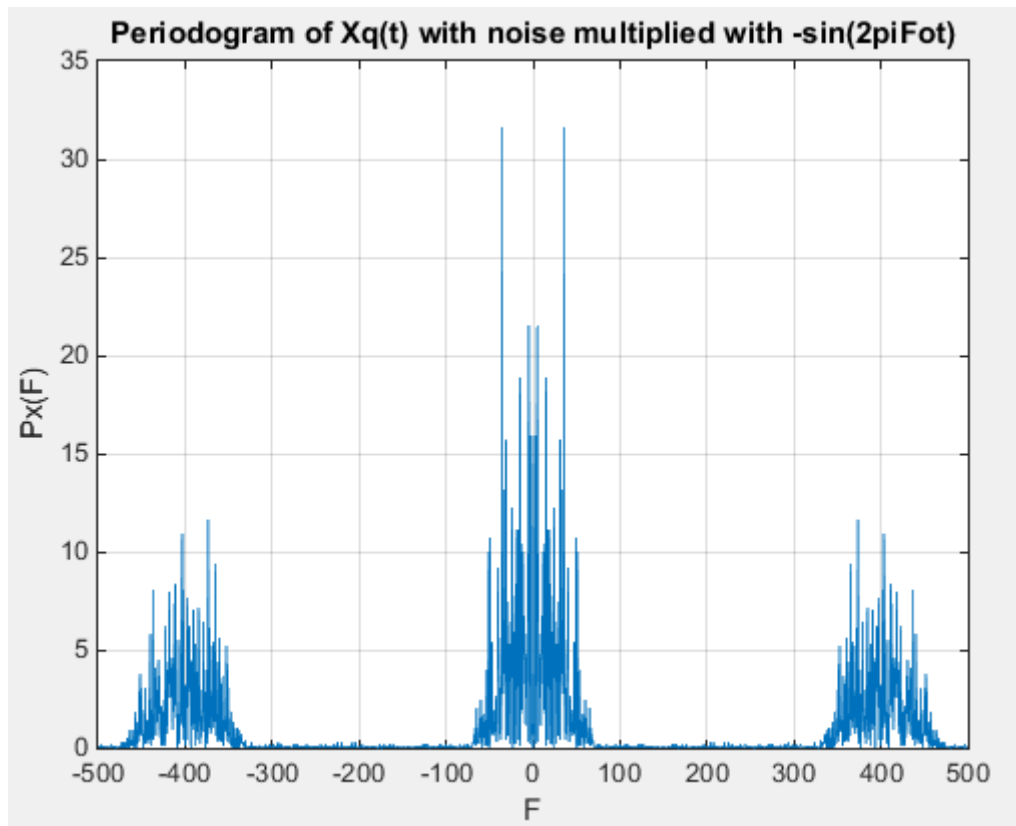
8) Προσθέτουμε λευκό Gaussian θόρυβο στην έξοδο του καναλιού με διασπορά

$$\sigma_W^2 = \frac{10A^2}{T_s \cdot 10^{\frac{\text{SNR}_{\text{dB}}}{10}}}$$

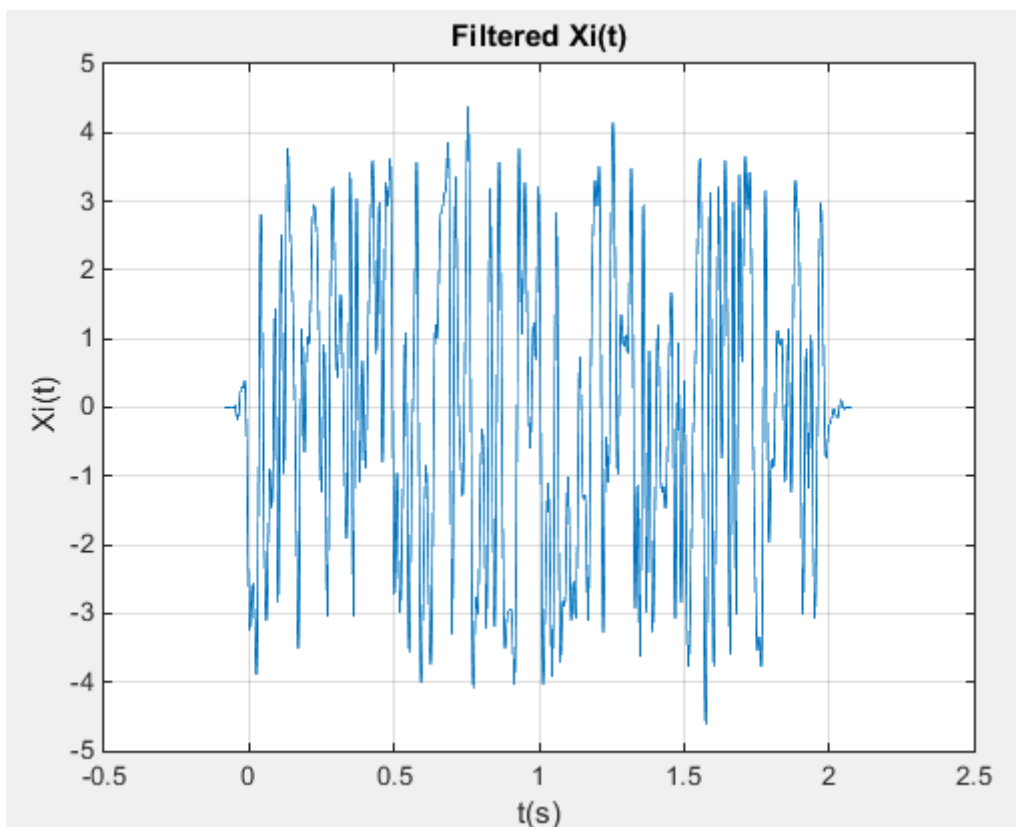
9) Ζητήθηκε να διακλαδώσουμε την ενθόρυβη κυματομορφή και να την πολλαπλασιάσουμε με τους δοσμένους φορείς. Παρατηρούμε ότι τα σήματα έχουν κέντρο τις συχνότητες $2F_0$, 0 και $-2F_0$.

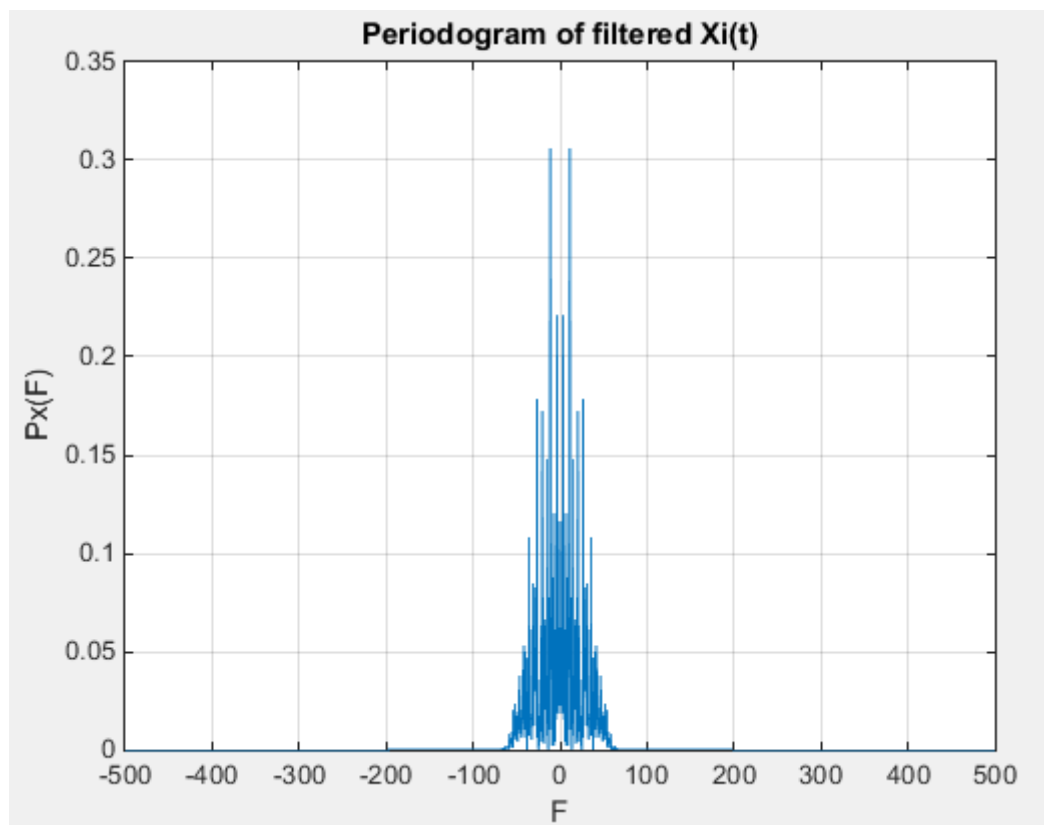
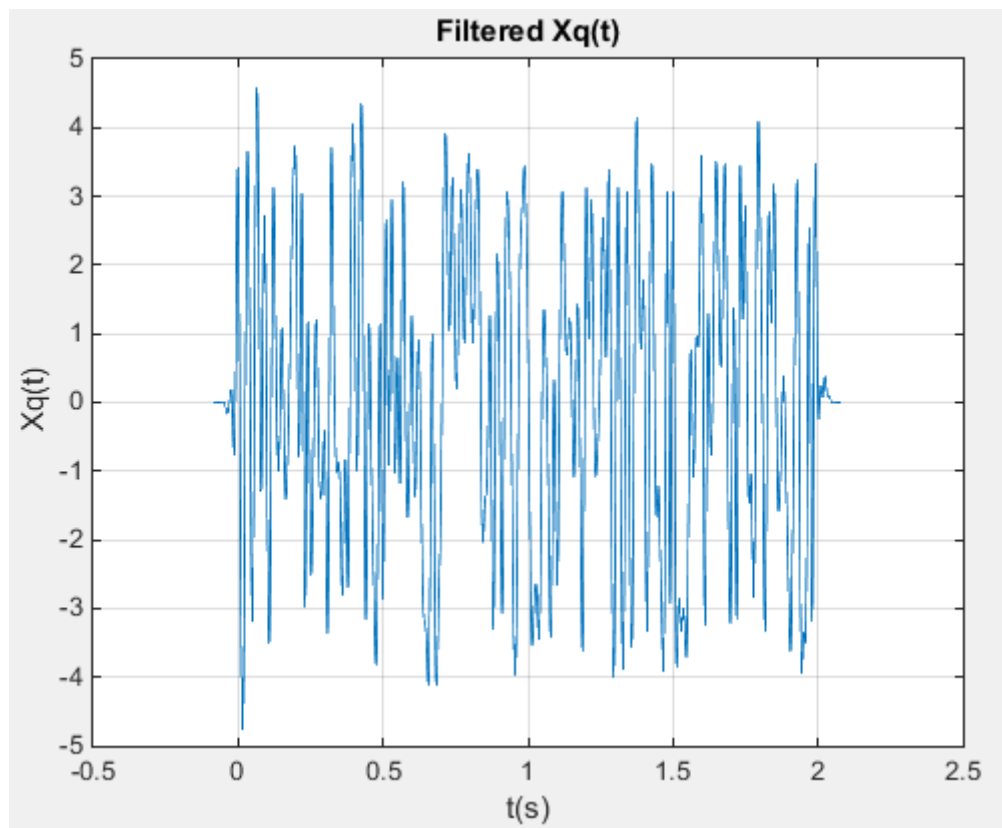


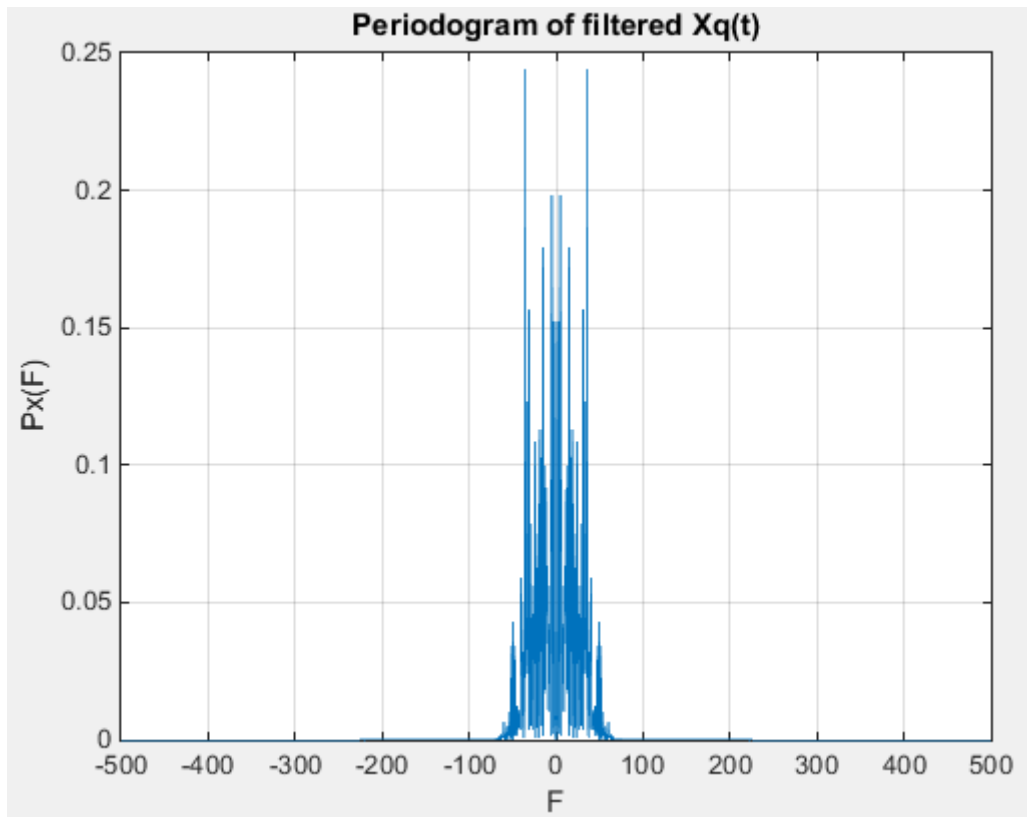




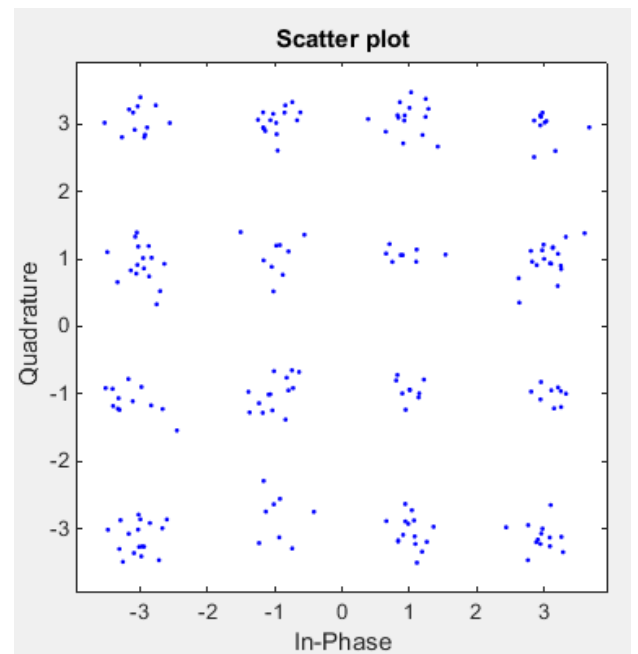
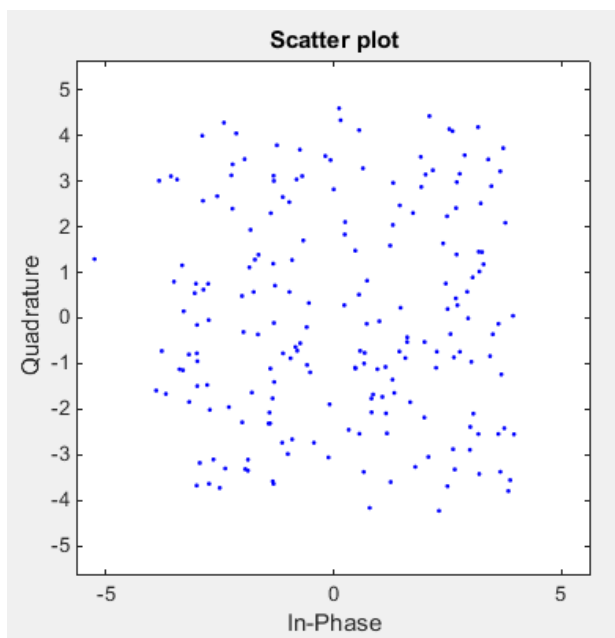
10) Περάστηκαν οι κυματομορφές από τα φίλτρα. Φαίνεται να έχουν αποκοπεί οι όροι στις μεγάλες συχνότητες.







11) Έγινε η δειγματοληψία της εξόδου των προσαρμοσμένων φίλτρων τις σωστές χρονικές στιγμές και σχεδιάστηκε με την εντολή scatterplot για SNR=10 SNR=20



Παρατηρούμε πως για μεγαλύτερο SNR υπερτερεί το σήμα και πως συγκεκριμένα για το SNR = 20 διακρίνεται με εμφανή τρόπο η 16-QAM διαμόρφωση της άσκησής μας.

12) Εδώ χρησιμοποιήθηκε η συνάρτηση detect_4_PAM η οποία χρησιμοποιεί τον κανόνα του εγγύτερου γείτονα για την ακολουθία εισόδου 4PAM. Παρατίθεται ο κώδικάς της. Στην συνέχεια εφαρμόστηκε στις εξόδους του προηγούμενου ερωτήματος.

```
function [ est_X ] = detect_4_PAM( Y, A )
est_X = ones(1,length(Y));
X0=(-3)*A;
X1=(-1)*A;
X2=A;
X3=3*A;
for p=1:length(Y)
    %find the distance for each symbol
    d0=sqrt((Y(:,p)-X0)^2 + (Y(:,p)-X0)^2);
    d1=sqrt((Y(:,p)-X1)^2 + (Y(:,p)-X1)^2);
    d2=sqrt((Y(:,p)-X2)^2 + (Y(:,p)-X2)^2);
    d3=sqrt((Y(:,p)-X3)^2 + (Y(:,p)-X3)^2);
    %find the minimum distance
    if (d0==min([d0,d1,d2,d3]))
        est_X(:,p) = X0;
    elseif (d1==min([d0,d1,d2,d3]))
        est_X(:,p) = X1;
    elseif (d2==min([d0,d1,d2,d3]))
        est_X(:,p) = X2;
    elseif (d3==min([d0,d1,d2,d3]))
        est_X(:,p) = X3;
    end
end
```

13) Υπολογίστηκε ο αριθμός σφαλμάτων απόφασης συμβόλου. Τοποθετήθηκαν σε έναν πίνακα οι ακολουθίες εισόδου και σε έναν άλλο οι αποφάσεις. Αφαιρέθηκαν μεταξύ τους και τα αποτελέσματα τοποθετήθηκαν σε έναν νέο πίνακα. Τέλος, συγκρίθηκαν οι στήλες του νέου πίνακα με σκοπό τον υπολογισμό των σφαλμάτων.

Με δοκιμές για κάποιες τιμές του SNR αποδεικνύεται πως για μεγαλύτερο SNR έχουμε και περισσότερη χρήσιμη πληροφορία.

14) Γράφτηκε η συνάρτηση PAM_4_to_bits, η οποία χρησιμοποιεί την αντίστροφη απεικόνιση Gray, από σύμβολα σε δυάδες bits, και υπολογίζει την εκτιμώμενη δυαδική ακολουθία εισόδου από τις αποφάσεις για τις ακολουθίες.

```
function [ est_bit ] = PAM_4_to_bits( X, A )
est_bit = ones(1,2*length(X));
X0=(-3)*A;
X1=(-1)*A;
X2=A;
X3=3*A;
for p=1:length(X)
    if (X(p)==X0)
        est_bit(2*p-1) = 0;
        est_bit(2*p) = 0;
    elseif (X(p)==X1)
        est_bit(2*p-1) = 0;
        est_bit(2*p) = 1;
    elseif (X(p)==X2)
        est_bit(2*p-1) = 1;
        est_bit(2*p) = 1;
    end
end
```

```

elseif (X(p)==X3)
    est_bit(2*p-1) = 1;
    est_bit(2*p)= 0;
end
end
end

```

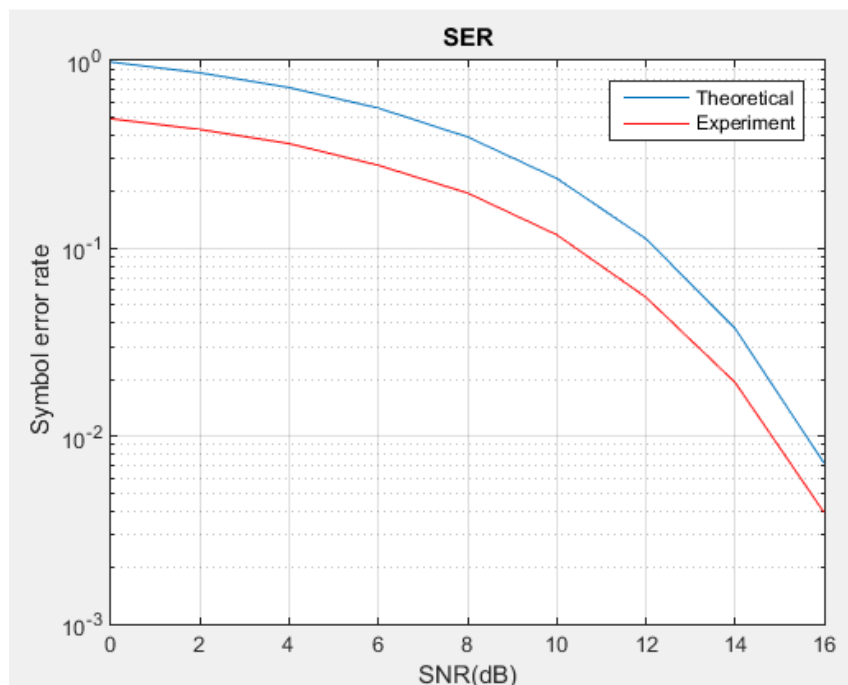
15) Υπολογίστηκε μέσω της συνάρτησης PAM_4_to_bits η εκτιμώμενη ακολουθία εισόδου, τοποθετήθηκαν οι τιμές σε πίνακα και ύστερα αφαιρέθηκε η αρχική ακολουθία με σκοπό να υπολογιστεί το σφάλμα.

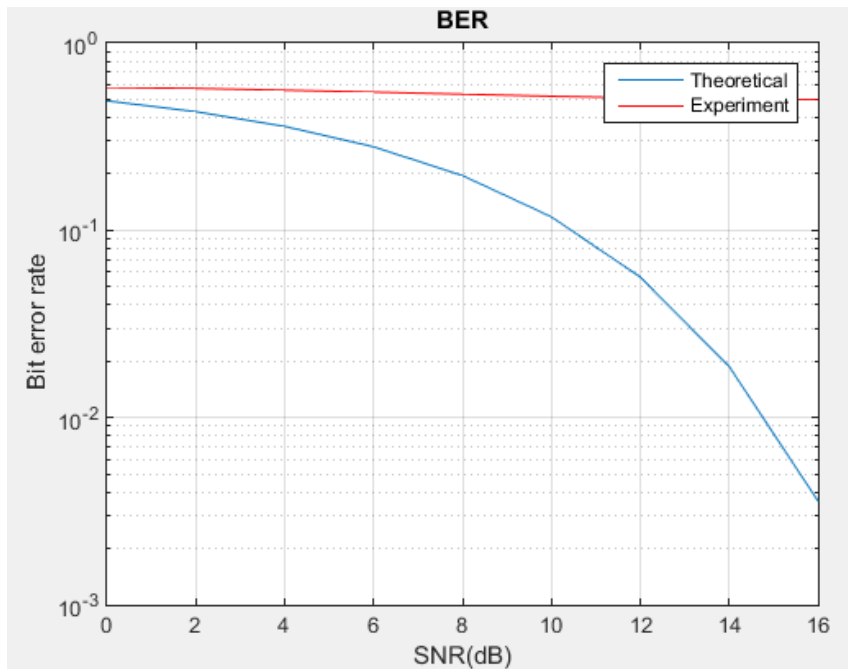
Μέρος Β

Σε αυτό το μέρος υπολογίστηκε η πιθανότητα σφάλματος συμβόλου και bit με τη μέθοδο Monte Carlo. Για SNR από 0 μέχρι 16 με βήμα 2 υπολογίστηκε πειραματικά η πιθανότητα σφάλματος συμβόλου, επαναλαμβάνοντας όλο το μέρος Α για K=200 φορές. Δημιουργώ δύο πίνακες P_symbol και P_bit, όπου εισάγω τις πειραματικές τιμές

Για τον θεωρητικό υπολογισμό των SER, BER χρησιμοποιήθηκε ο τύπος

$$\begin{aligned}
 P_{16-QAM}^E &= 1 - P_{16-QAM}^C \\
 &= 3Q\left(\frac{A}{\sigma_N}\right) - \frac{9}{4}\left(Q\left(\frac{A}{\sigma_N}\right)\right)^2 \\
 &\approx 3Q\left(\frac{A}{\sigma_N}\right).
 \end{aligned}$$





Οι τιμές των σφαλμάτων συμβόλου είναι πανομοιότυπες ενώ ο πειραματικός υπολογισμός του σφάλματος των bit προφανώς κάπου έχει λάθος καθώς μειώνεται ελάχιστα.

Κώδικας:

```
close all;
clear all;
clc;

% dedomena
N=200;
T=0.01;
over=10;
Ts=T/over;
A_srrc=4;
A=1;
a=0.5;
Fs=1/Ts;
Nf=2048;
Fo=200;

%% A1
%sxediasmos dyadikhs akolouthias me 4*N bits
b = (sign(randn(4*N,1))+1)/2;

%% A2,3
bit_seq1=b(1:2*N);      %prwta 2N bits
bit_seq2=b(2*N+1:4*N);  %epomena 2N bits

%apeikonish se 4PAM symbola
Xi_n=bits_to_4_PAM(bit_seq1,A);
Xq_n=bits_to_4_PAM(bit_seq2,A);
```

```

%% A4
%dhmiourgia palmou
[phi, t] = srrc_pulse(T, over, A_srrc, a);

Xi = 1/Ts*upsample(Xi_n,over);
Xi_t = Ts*conv(Xi, phi);

Xq= 1/Ts*upsample(Xq_n,over);
Xq_t = Ts*conv(Xq, phi);

t2 = 0:Ts:N*T-Ts;
t_conv = min(t)+min(t2):Ts:max(t)+max(t2);

figure();
plot(t_conv,Xi_t);
grid on;
title('Xi(t) with SRRC filter');
xlabel('t(s)');
ylabel('Xi(t)');

figure();
plot(t_conv,Xq_t);
grid on;
title('Xq(t) with SRRC filter');
xlabel('t(s)');
ylabel('Xq(t)');

f = (-Fs/2):(Fs/Nf):(Fs/2)-(Fs/Nf);

%Periodogrammata:

Xi_f= Ts*fftshift(fft(Xi_t,Nf)); %fourier
Ttotal1 = length(Xi_t)*Ts; %ttotal
P_Xi=(abs(Xi_f).^2)./Ttotal1; %periodogramma

Xq_f= Ts*fftshift(fft(Xq_t,Nf));
Ttotal2 = length(Xq_t)*Ts;
P_Xq=(abs(Xq_f).^2)./Ttotal2;

figure();
plot(f, P_Xi);
title('Periodogram of Xi(t)');
xlabel('F');
ylabel('Px(F)');
grid on;

figure();
plot(f, P_Xq);
title('Periodogram of Xq(t)');
xlabel('F');
ylabel('Px(F)');
grid on;

%% A5

Xi_mod= 2*Xi_t.*cos(2*pi*Fo.*t_conv); %modulated
Xq_mod= -2*Xq_t.*(sin(2*pi*Fo.*t_conv));%modulated

```

```

Xi_mod_f= Ts*fftshift(fft(Xi_mod,Nf));
Ttotal3 = length(Xi_mod)*Ts;
P_Xi_mod=(abs(Xi_mod_f).^2)./Ttotal3;

Xq_mod_f= Ts*fftshift(fft(Xq_mod,Nf));
Ttotal4 = length(Xq_mod)*Ts;
P_Xq_mod=(abs(Xq_mod_f).^2)./Ttotal4;

figure();
plot(t_conv,Xi_mod);
grid on;
title('Xi(t) multiplied with 2cos(2piFot)');
xlabel('t(s)');
ylabel('Xi(t)');

figure();
plot(t_conv,Xq_mod);
grid on;
title('Xq(t) multiplied with -2sin(2piFot)');
xlabel('t(s)');
ylabel('Xq(t)');

% periodogrammata Xi_mod Xq_mod:
figure();
plot(f, P_Xi_mod);
grid on;
title('Periodogram of Xi(t) multiplied');
xlabel('F');
ylabel('Px(F)');

figure();
plot(f,P_Xq_mod);
title('Periodogram of Xq(t) multiplied');
xlabel('F');
ylabel('Px(F)');
grid on

%% A6
%eisodos kanaliou
X_mod= Xi_mod + Xq_mod;

%periodogramma
X_mod_f= Ts*fftshift(fft(X_mod,Nf));
Ttotal5 = length(X_mod)*Ts;
P_Xmod=(abs(X_mod_f).^2)./Ttotal5;

figure();
plot(t_conv,X_mod)
grid on;
title('X(t)');
xlabel('t(s)');
ylabel('X(t)');

figure();
plot(f,P_Xmod);
title('Periodogram of X(t)');
xlabel('F');
ylabel('Px(F)');

```



```

grid on

%% A8

%Noise
SNR=20;
s_w2 = (10*(A^2))/(Ts*10^(SNR/10));
w = sqrt(s_w2)*randn(1,length(X_mod));
%shma me thoryvo
X_mod_noise = X_mod + w;

%% A9
%pollaplasiasmos me cos kai -sin
Xi_demod=X_mod_noise.*cos(2*pi*Fo.*t_conv);
Xq_demod= X_mod_noise.*(-sin(2*pi*Fo.*t_conv));

%periodogrammata
Xi_demod_f= Ts*fftshift(fft(Xi_demod,Nf));
Ttotal6 = length(Xi_demod)*Ts;
P_Xidemod=(abs(Xi_demod_f).^2)./Ttotal6;

Xq_demod_f= Ts*fftshift(fft(Xq_demod,Nf));
Ttotal7 = length(Xq_demod)*Ts;
P_Xqdemod=(abs(Xq_demod_f).^2)./Ttotal7;

%Kymatomorfes exodou Xi_demod kai Xq_demod
figure
plot(t_conv,Xi_demod)
title('Xi(t) with noise multiplied with cos(2piFot)');
xlabel('t(s)');
ylabel('Xi(t)');
grid on

figure
plot(t_conv,Xq_demod)
title('Xq(t) with noise multiplied with -sin(2piFot)');
xlabel('t(s)');
ylabel('Xq(t)');
grid on

% periodogrammata Xi_demod kai Xq_demod
figure();
plot(f, P_Xidemod);
title('Periodogram of Xi(t) with noise multiplied with cos(2piFot)');
xlabel('F');
ylabel('Px(F)');
grid on

figure();
plot(f, P_Xqdemod);
title('Periodogram of Xq(t) with noise multiplied with -sin(2piFot)');
xlabel('F');
ylabel('Px(F)');
grid on;
grid on

%% A10
%mesa apo to filtro SRRC
Xi_demodulated_t = Ts*conv(Xi_demod, phi);

```

```

Xq_demodulated_t = Ts*conv(Xq_demod, phi);

%periodogrammata
Xi_demod_t_f= Ts*fftshift(fft(Xi_demodulated_t,Nf));
Ttotal8 = length(Xi_demodulated_t)*Ts;
P_Xi_demod_t=(abs(Xi_demod_t_f).^2)./Ttotal8;

Xq_demod_t_f= Ts*fftshift(fft(Xq_demodulated_t,Nf));
Ttotal9 = length(Xq_demodulated_t)*Ts;
P_Xq_demod_t=(abs(Xq_demod_t_f).^2)./Ttotal9;

t_conv2 = min(t_conv)+min(t):Ts:max(t)+max(t_conv);

F = -Fs/2:Fs/Nf:Fs/2-Fs/Nf;

%Kymatomorfes exodou Xi_demod_t kai Xq_demod_t
figure
plot(t_conv2,Xi_demodulated_t)
grid on
title('Filtered Xi(t)');
xlabel('t(s)');
ylabel('Xi(t)');

figure
plot(t_conv2,Xq_demodulated_t)
title('Filtered Xq(t)');
xlabel('t(s)');
ylabel('Xq(t)');
grid on

% periodogrammata Xi_demod_t kai Xq_demod_t se plot:
figure();
plot(F, P_Xi_demod_t);
title('Periodogram of filtered Xi(t)');
xlabel('F');
ylabel('Px(F)');
grid on

figure();
plot(F,P_Xq_demod_t);
title('Periodogram of filtered Xq(t)');
xlabel('F');
ylabel('Px(F)');
grid on

%% A11

Xi_demod_t_samp= Xi_demodulated_t((2*A_srrc*T/Ts) +1 : over : length(Xi_demodulated_t)-(2*A_srrc*T/Ts));
Xq_demod_t_samp= Xq_demodulated_t((2*A_srrc*T/Ts) +1 : over : length(Xq_demodulated_t)-(2*A_srrc*T/Ts));
for i=1:N
    Sampling(i,1)=Xi_demod_t_samp(i);
    Sampling(i,2)=Xq_demod_t_samp(i);
end

scatterplot(Sampling)

%% A12

```

```

for i=1:N
    Xi_det(i)=detect_4_PAM(Xi_demod_t_samp(i),A);
    Xq_det(i)=detect_4_PAM(Xq_demod_t_samp(i),A);
end

```

```

%% A13
%Pinakas pou esteila:
X_sent=[Xi_n ; Xq_n];
%Ekana detect:
X_detect=[Xi_det ; Xq_det];
%diafora tw'n 2
dif = X_sent-X_detect;

```

```

er=0;
for i=1:N
    if(dif(1,i)~=0 || (dif(2,i)~=0))
        er=er+1;
    end
end
er

```

```

%% A15
bit_i=PAM_4_to_bits(Xi_det,A);
bit_q=PAM_4_to_bits(Xq_det,A);

```

```

bit_est=[bit_i bit_q]';

```

```

ber=0;
for i=1:4*N
    if(bit_est(i)~=b(i)) %elegxw gia error
        ber=ber+1;
    end
end
ber

```

```

%% B1

```

```

K = 200;
T = 0.01;
over = 10;
A = 4;
a = 0.5;
Ts = T / over;
fs = 1/Ts;
N = 200;
F0 = 200;
SNR = 0:2:16;

```

```

for k=1:length(SNR)

    total_ber = 0;
    total_ser = 0;

    for p = 1:K
        %% A1
        b = (sign(randn(4*N, 1)) + 1)/2;

        %% A2,3
        bit_seq1=b(1:2*N);
    end
end

```

```

bit_seq2=b(2*N+1:4*N);

Xi_n = bits_to_4_PAM(bit_seq1, A);
Xq_n = bits_to_4_PAM(bit_seq2, A);

%% A4
[phi, t] = srirc_pulse(T, over, A, a);

Xi = (1/Ts) * upsample(Xi_n, over);
Xi_t = conv(phi,Xi) * Ts;

Xq = (1/Ts) * upsample(Xq_n, over);
Xq_t = conv(phi,Xq) * Ts;

t2 = 0:Ts:T*N-Ts;
t_conv = t(1)+t2(1):Ts: t(end)+t2(end);

%% A5
Xi_mod = 2*Xi_t.*cos(2*pi*F0*t_conv);
Xq_mod = -2*Xq_t.*sin(2*pi*F0*t_conv);

%% A6
X_mod = Xi_mod + Xq_mod;

%% A8
s_w2 = (10 * A^2) / (Ts * 10^(SNR(k)/10));
w = sqrt(s_w2)*randn(1, length(X_mod));
X_mod_noise = X_mod + w;

%% A9
Xi_demod = X_mod_noise .* cos(2*pi*F0*t_conv);
Xq_demod = X_mod_noise .* (-sin(2*pi*F0*t_conv));

%% A10
t_conv2 = t(1)+t_conv(1):Ts: t(end)+t_conv(end);
Xi_demodulated = Ts*conv(phi,Xi_demod);
Xq_demodulated = Ts*conv(phi,Xq_demod);

%% A11
tconv2=t_conv2(1+80:end-80);
XI_demodulated=Xi_demodulated(1+80:end-80);
XQ_demodulated=Xq_demodulated(1+80:end-80);

Xi_sampled=downsample(XI_demodulated, over);
Xq_sampled=downsample(XQ_demodulated, over);

%% A12
Xiest=detect_4_PAM(Xi_sampled, A);
Xqgest=detect_4_PAM(Xq_sampled, A);

%% A14
est_bits_i = PAM_4_to_bits(Xiest, A);
est_bits_q = PAM_4_to_bits(Xqgest, A);

%% A15
% esteila
X_sent = [Xi_n Xq_n];
% ekana detect

```

```

X_detect = [Xiest Xqest];
% symbol error
ser = 0;
for u=1:length(X_detect)
    if(X_detect(u) ~= X_sent(u))
        ser = ser + 1;
    end
end

%bits error
ber = 0;
est_bits = [est_bits_i est_bits_q];
for u=1:length(est_bits)
    if(est_bits(u) ~= b(u))
        ber = ber + 1;
    end
end

% Synolika errors
total_ber = total_ber + ber;
total_ser = total_ser + ser;

end

P_bit(k) = total_ber / (K*4*N);
P_sym(k) = total_ser / (K*2*N);

s_w2 = (10 * A^2)/(Ts*10^(SNR(k)/10));
s_n2 = (Ts * s_w2)/2;
P_sym_th(k) = 3*Q(A/sqrt(s_n2));
P_bit_th(k) = P_sym_th(k) / 2;

end

figure
semilogy(SNR, P_sym_th);
hold on;
grid on;
semilogy(SNR, P_sym, 'r');
hold off;
title('SER')
xlabel('SNR(dB)')
ylabel('Symbol error rate')
legend('Theoretical', 'Experiment')

figure
semilogy(SNR, P_bit_th);
hold on;
grid on;
semilogy(SNR, P_bit, 'r');
hold off;
title('BER')
xlabel('SNR(dB)')
ylabel('Bit error rate')
legend('Theoretical', 'Experiment')

```

