

ΤΗΛ301

ΑΣΚΗΣΗ 3, 9/1/2020

Σοφία Καφρίτσα Γεωργαντά
ΑΜ | 2016030136

ΕΡΩΤΗΜΑ Α

Ερώτημα 1°

Ορίζω τα δεδομένα και με $N=200$, δημιουργώ την εξής δυαδική ακολουθία με στοιχεία $4N$ ισοπίθανα bits:

```
%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;

%-----Erwthma 1-----

%dhmiourgia duadikhs akolouthias me 4N isopithana bits
b = (sign(randn(4*N,1))+1)/2;
```

Ερώτημα 2°

Συντάσσω τη συνάρτηση `function X = bits_to_4_PAM(bit_sec, A)`, η οποία χρησιμοποιώντας κωδικοποίηση Grey, απεικονίζει τη δυαδική ακολουθία εισόδου `bit_sec` σε ακολουθία 4-PAM συμβόλων. Παρατίθεται ο κώδικάς της παρακάτω:

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

Ερώτημα 3°

Παρακάτω, απεικόνισα τα πρώτα $2N$ bits της ακολουθίας στα 4-PAM σύμβολα $X_{l,n}$ για $n = 1, \dots, N$ και τα επόμενα $2N$ bits στα 4-PAM σύμβολα $X_{Q,n}$ για $n = 1, \dots, N$.

```
%-----Erwthma 2,3-----
bit_seq1=b(1:2*N); %ta prwta 2N bits
```

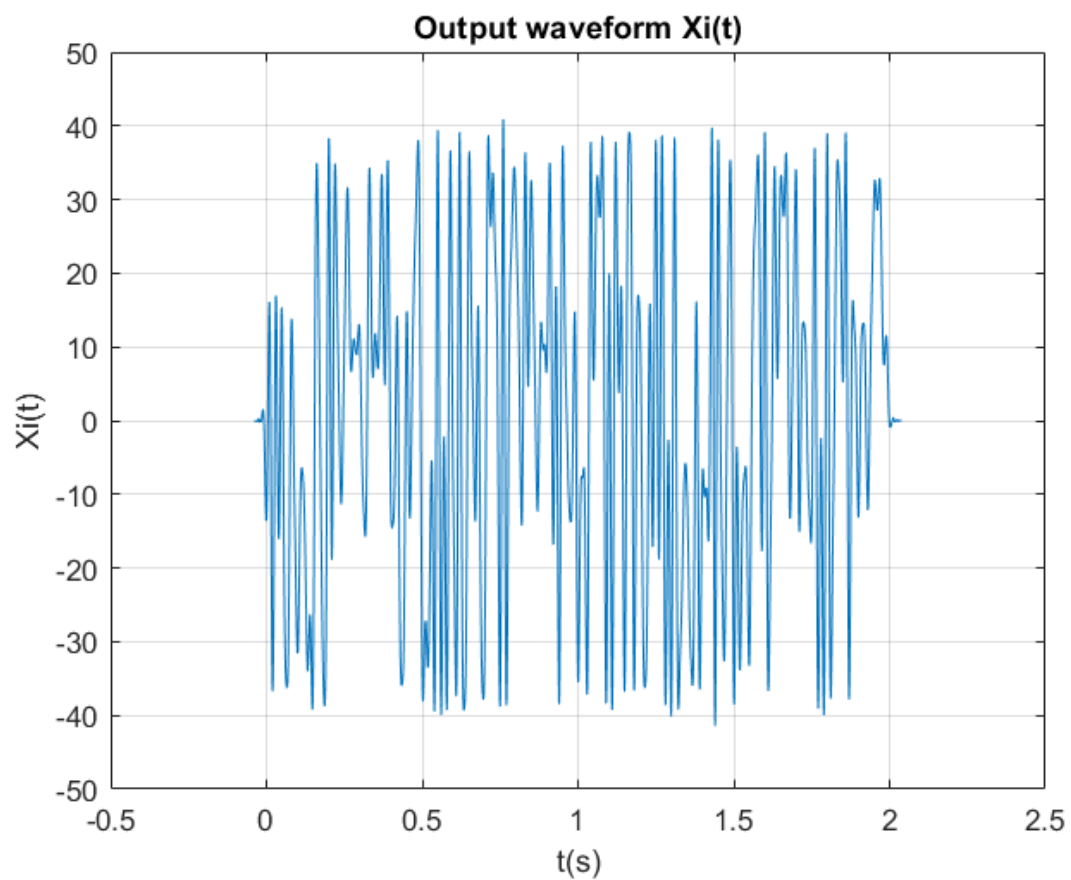
```
bit_seq2=b(2*N+1:4*N); %ta epomena 2N bits
```

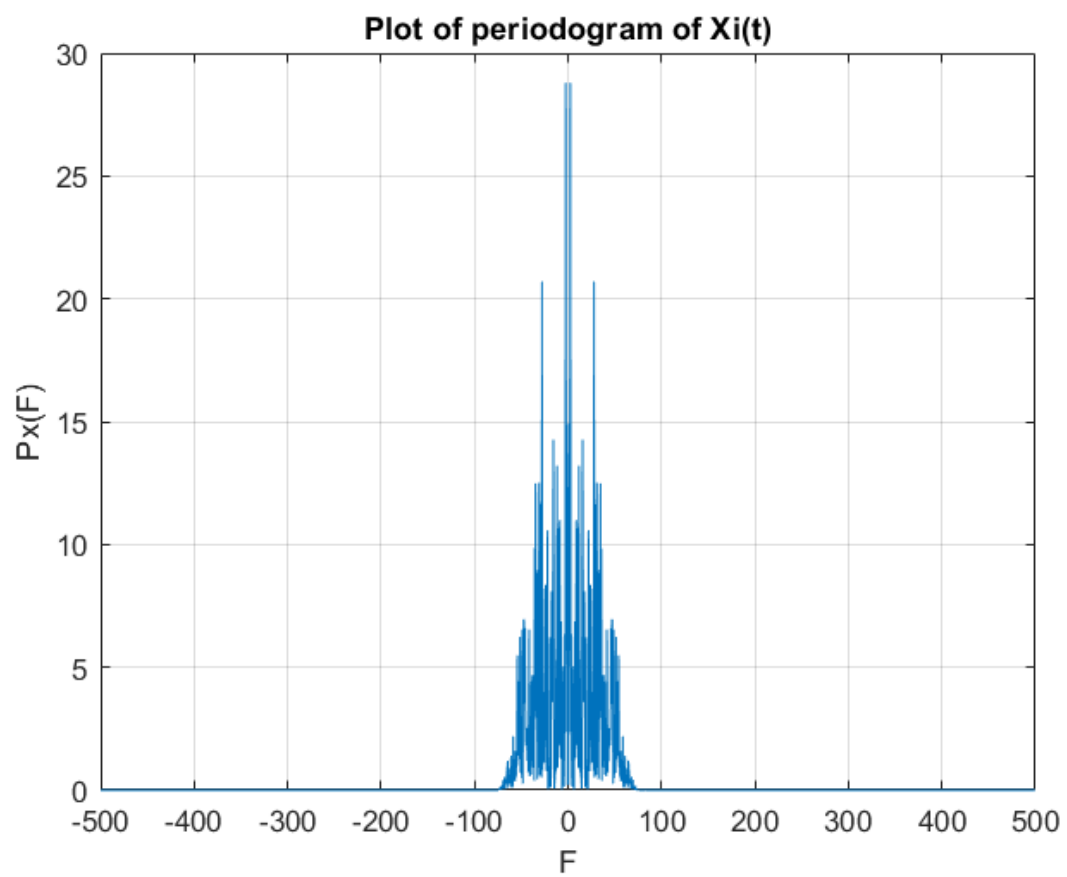
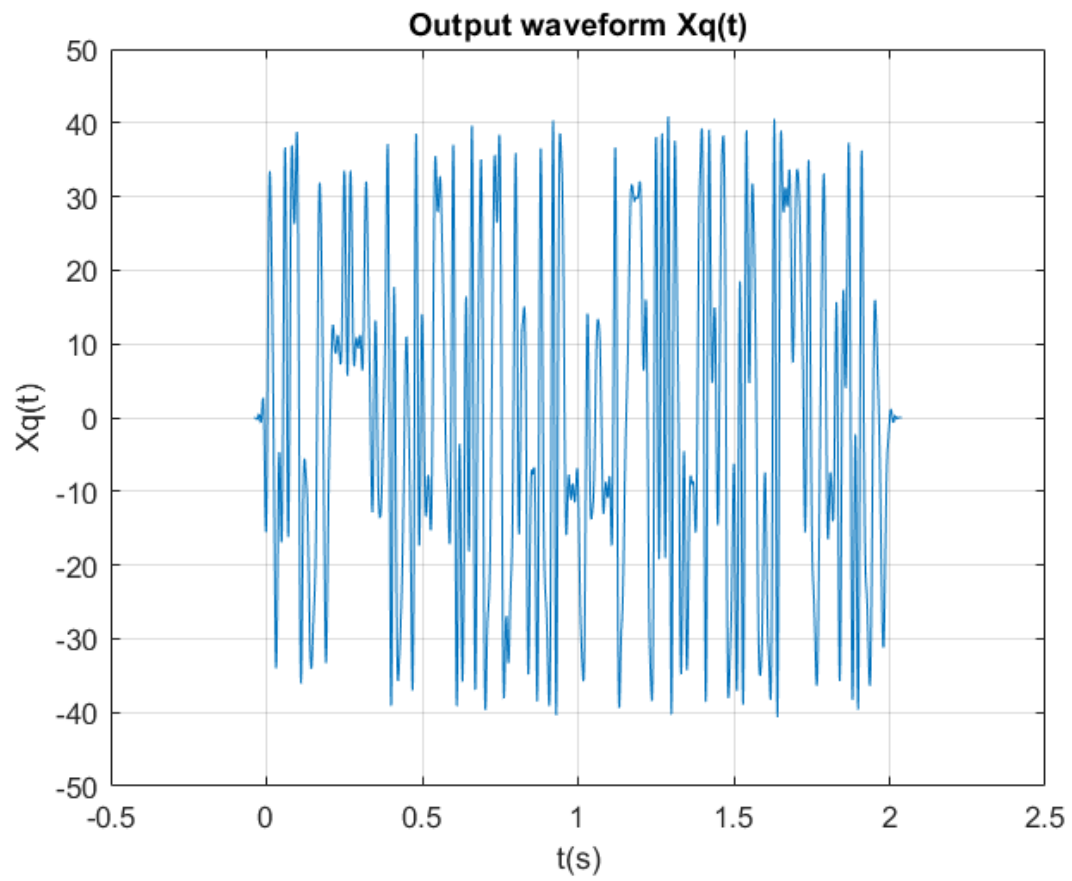
```
Xi_n= bits_to_4_PAM(bit_seq1,A);
```

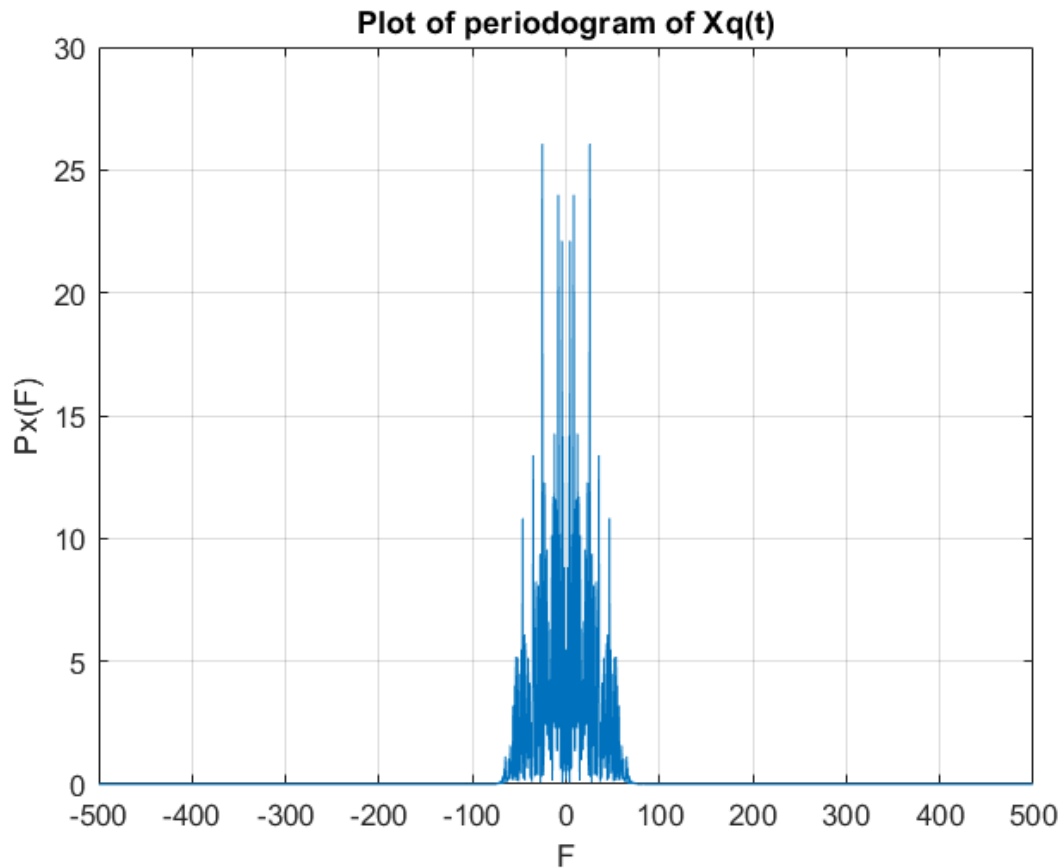
```
Xq_n= bits_to_4_PAM(bit_seq2,A);
```

Ερώτημα 4^ο

Θέτω $T = 0.01\text{sec}$, $\text{over} = 10$, $T_s = \frac{T}{\text{over}}$. Αφού περάσουν οι παραπάνω ακολουθίες από φίλτρα μορφοποίησης SRRC, σχηματίζω και σχεδιάζω τις κυματομορφές εξόδου $X_i(t)$ και $X_q(t)$. Οι κυματομορφές φαίνονται παρακάτω:







Παρατίθεται ο κώδικας:

```
%----Erwthma 4-----
%Dimiourgia apokomenou palmou:
[phi,t]=srrc_pulse(T,Ts,A_srrc,a);

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

Xq_d= 1/Ts*upsample(Xq_n,over);
Xq_t = Ts*conv(Xq_d, 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('Output waveform Xi(t)');
xlabel('t(s)');
ylabel('Xi(t)');

figure();
plot(t_conv,Xq_t);
grid on;
title('Output waveform Xq(t)');
xlabel('t(s)');
ylabel('Xq(t)');
```

```

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

%Periodogrammata:

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

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

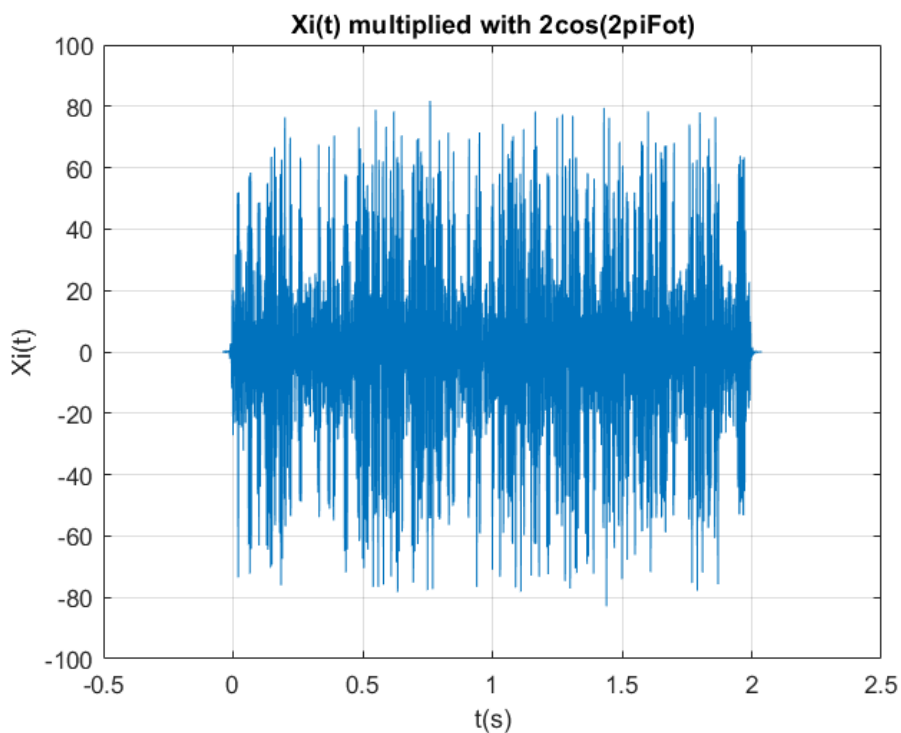
figure();
plot(f, Period_Xi_t);
title('Plot of periodogram of Xi(t)');
xlabel('F');
ylabel('Px(F)');
grid on;

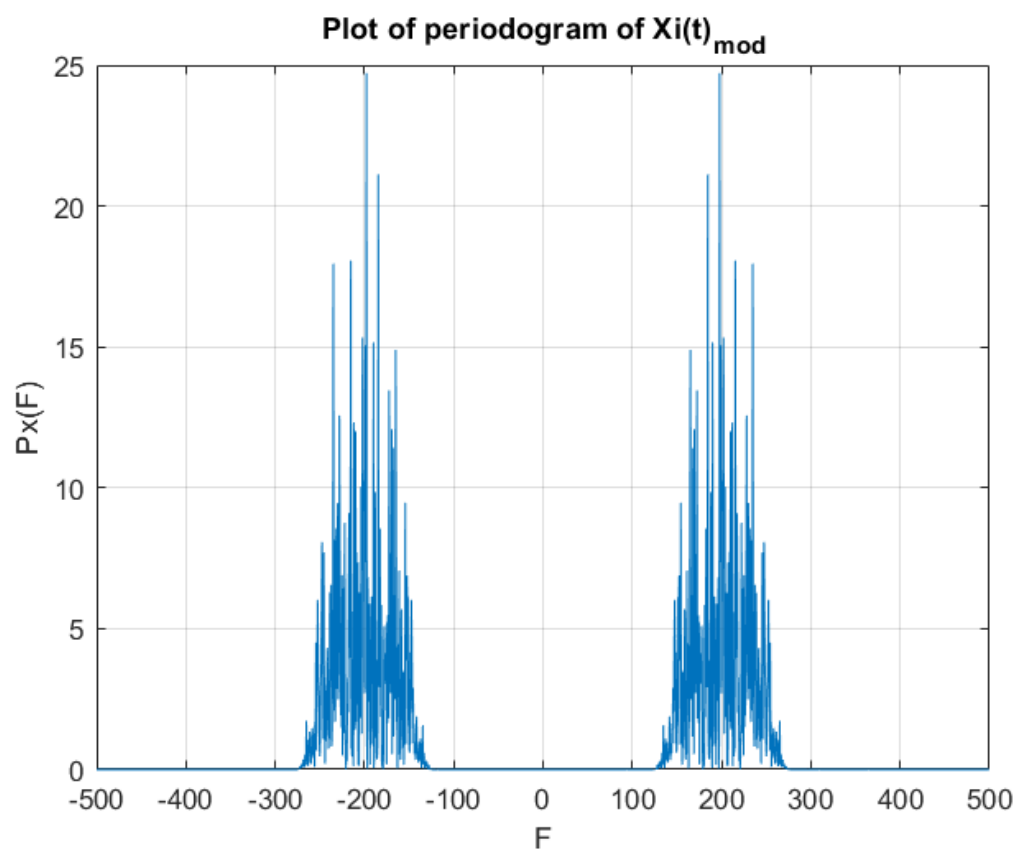
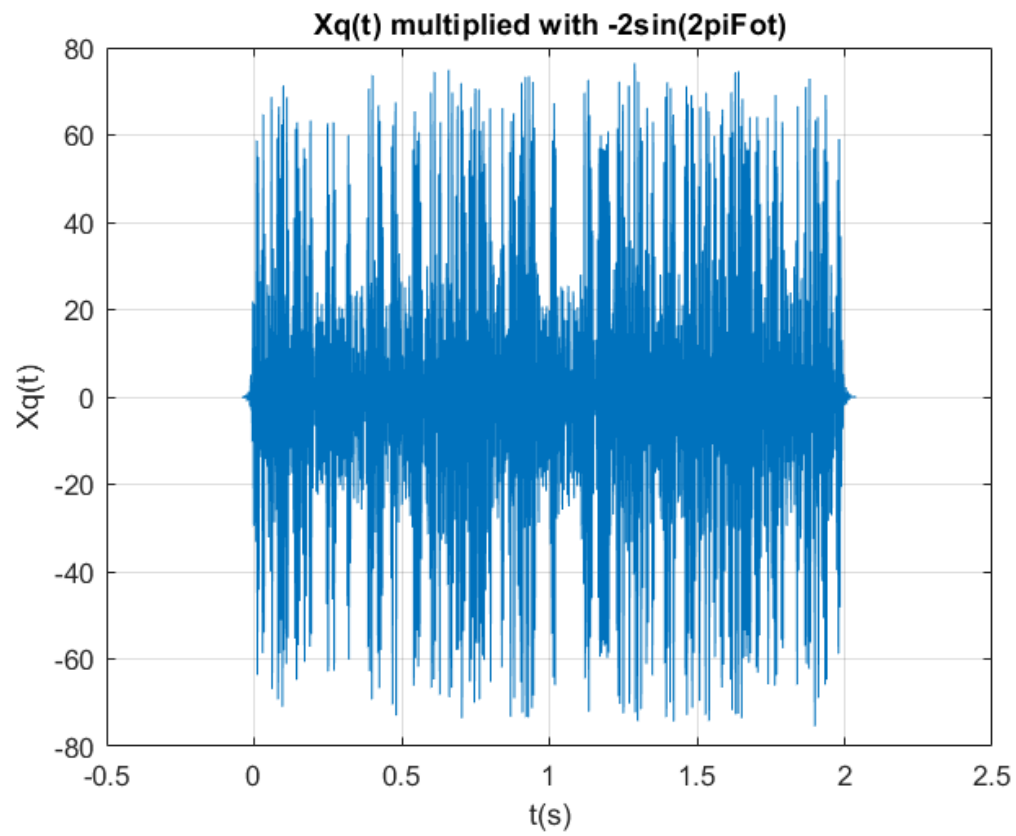
figure();
plot(f, Period_Xq_t);
title('Plot of periodogram of Xq(t)');
xlabel('F');
ylabel('Px(F)');
grid on;

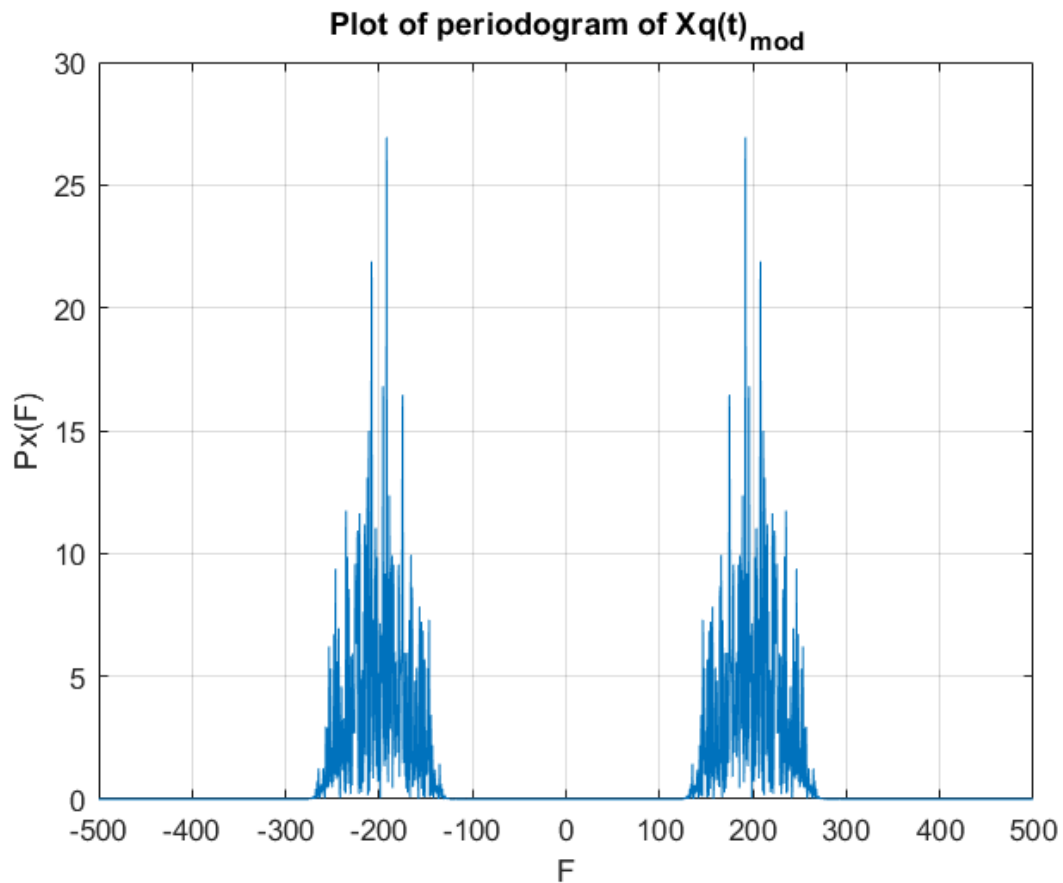
```

Ερώτημα 5°

Πολλαπλασιάζω τις κυματομορφές $X_i(t)$ και $X_q(t)$ με τους φορείς που ορίζονται στην εκφώνηση, ώστε να δημιουργηθούν οι κυματομορφές $X_i^{\text{mod}}(t) = 2 X_i(t)\cos(2\pi F_0 t)$ και $X_q^{\text{mod}}(t) = -2 X_q(t)\sin(2\pi F_0 t)$, με $F_0=200$ Hz. Όπως παρατηρούμε από τα περιοδογράμματα, έχει γίνει διαμόρφωση γύρω από τη συχνότητα $F=200\text{Hz}$ που θέσαμε. Ακολουθούν οι κυματομορφές που προκύπτουν και τα αντίστοιχα περιοδογράμματα:







Παρατίθεται ο κώδικας:

```
%-----Erwthma 5-----

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;
Period_Xi_mod=(abs(Xi_mod_f).^2)./Ttotal3;

Xq_mod_f= Ts*fftshift(fft(Xq_mod,Nf));
Ttotal4 = length(Xq_mod)*Ts;
Period_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) ');

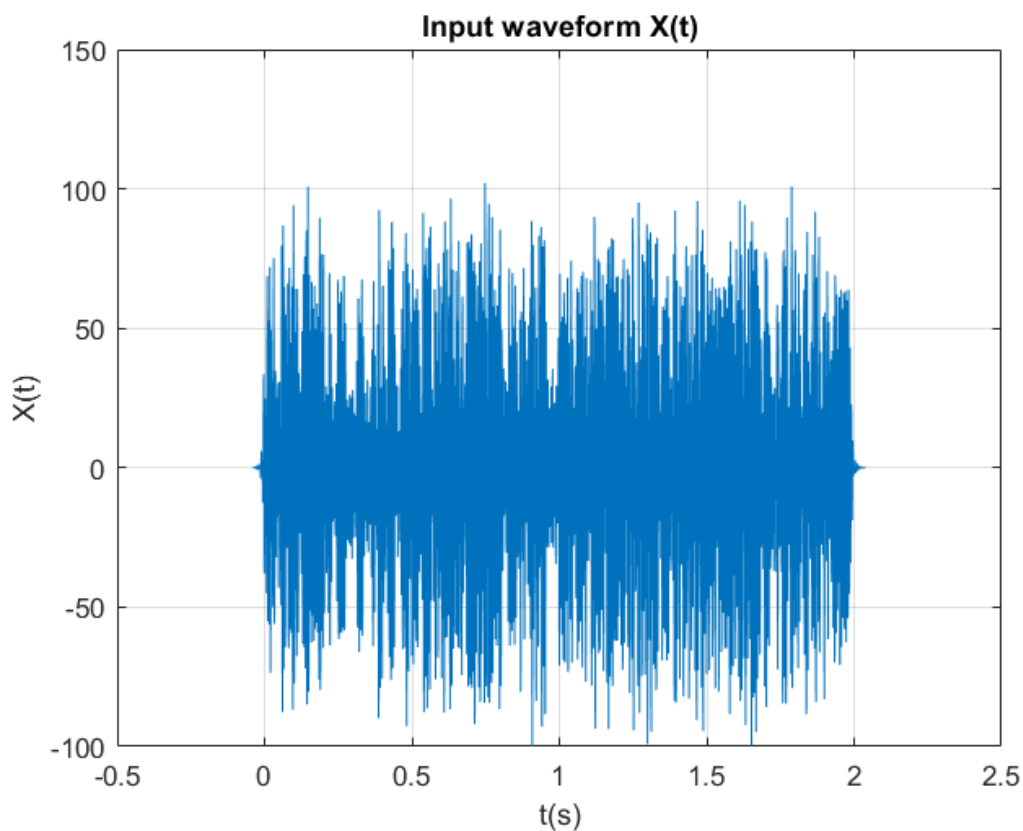
% periodogrammata Xi_mod kai Xq_mod se plot:
figure();
plot(f, Period_Xi_mod);
grid on
title('Plot of periodogram of Xi(t)_m_o_d');
xlabel('F');
ylabel('Px(F) ')

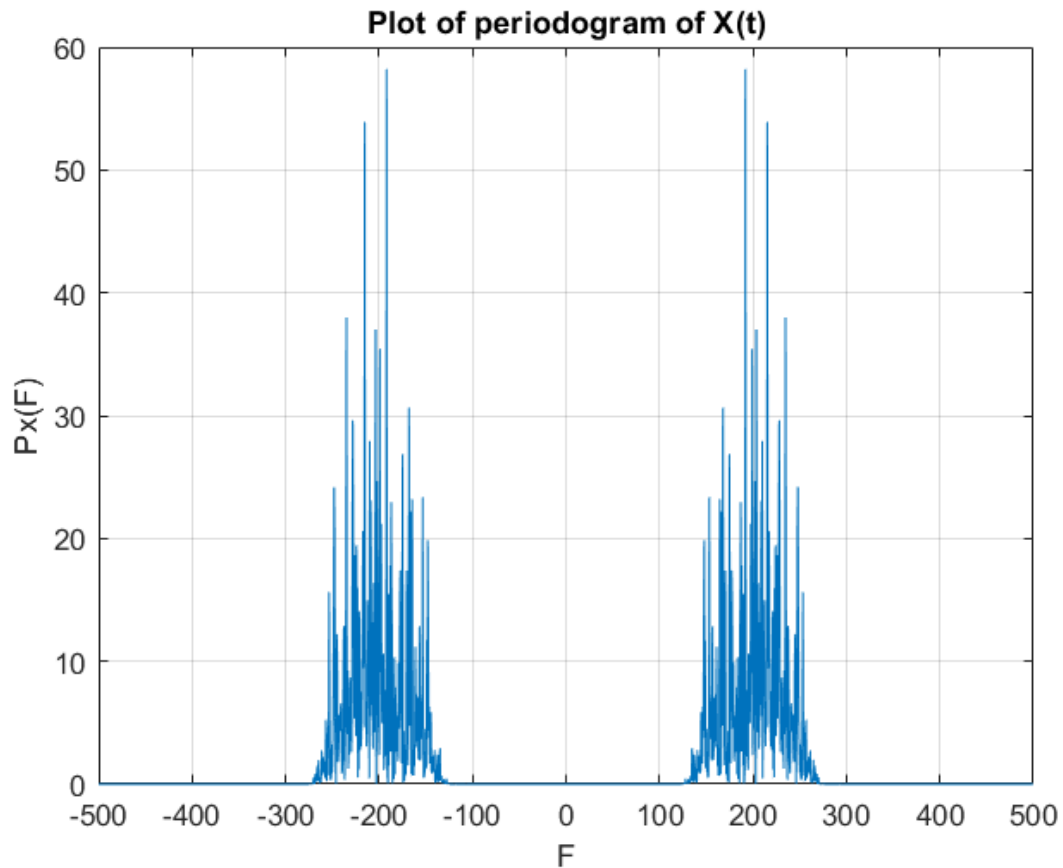
figure();
plot(f,Period_Xq_mod);
title('Plot of periodogram of Xq(t)_m_o_d');
xlabel('F');
ylabel('Px(F) ')
grid on

```

Ερώτημα 6

Παρακάτω, σχεδιάσα την είσοδο $X^{mod}(t) = X_i^{mod}(t) + X_q^{mod}(t)$ και το περιοδόγραμμά της. Από τα διαγράμματα, παρατηρούμε ότι το πλάτος έχει αυξηθεί, καθώς προσθέτουμε τις δύο κυματομορφές. Επίσης, οι $X(t)$ είναι κεντραρισμένη στην συχνότητα $|F_0|$ που θέσαμε προηγουμένως.





Παρατίθεται ο κώδικας:

```
%-----Erwthma 6-----

X_mod= Xi_mod + Xq_mod; %eisodos kanaliou, proshtesi

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

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

figure();
plot(f,Period_Xmod);
title('Plot of periodogram of X(t)');
xlabel('F');
ylabel('Px(F)');
grid on
```

Ερώτημα 7

Θεωρούμε ότι το κανάλι είναι ιδανικό.

Ερώτημα 8

Στην έξοδο του καναλιού προσθέτουμε λευκό Gaussian θόρυβο $W(t)$ με διασπορά

$$\sigma_W^2 = \frac{10A^2}{T_s \cdot 10^{\frac{SNR_{dB}}{10}}}.$$

Με αυτό τον τρόπο, διασφαλίζουμε ότι το SNR στην έξοδο του προσαρμοσμένου φίλτρου, μετρημένο σε dB είναι SNR_{dB} , διότι:

$$10 \log_{10} \frac{P_X}{P_N} = 10 \log_{10} \frac{10A^2}{2\sigma_N^2} = SNR_{dB}.$$

```
%-----Erwthma 8-----
```

```
%Noise
```

```
SNR=20;
```

```
Var_w = (10*(A^2))/(Ts*10^(SNR/10));
```

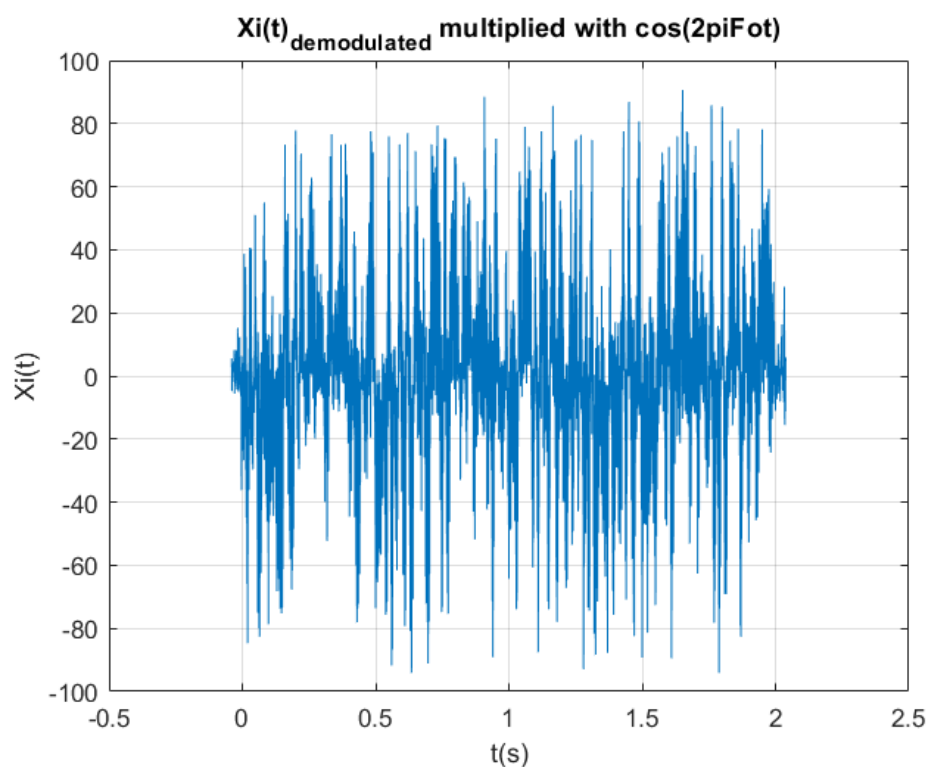
```
Noise= sqrt(Var_w)*randn(1,length(X_mod));
```

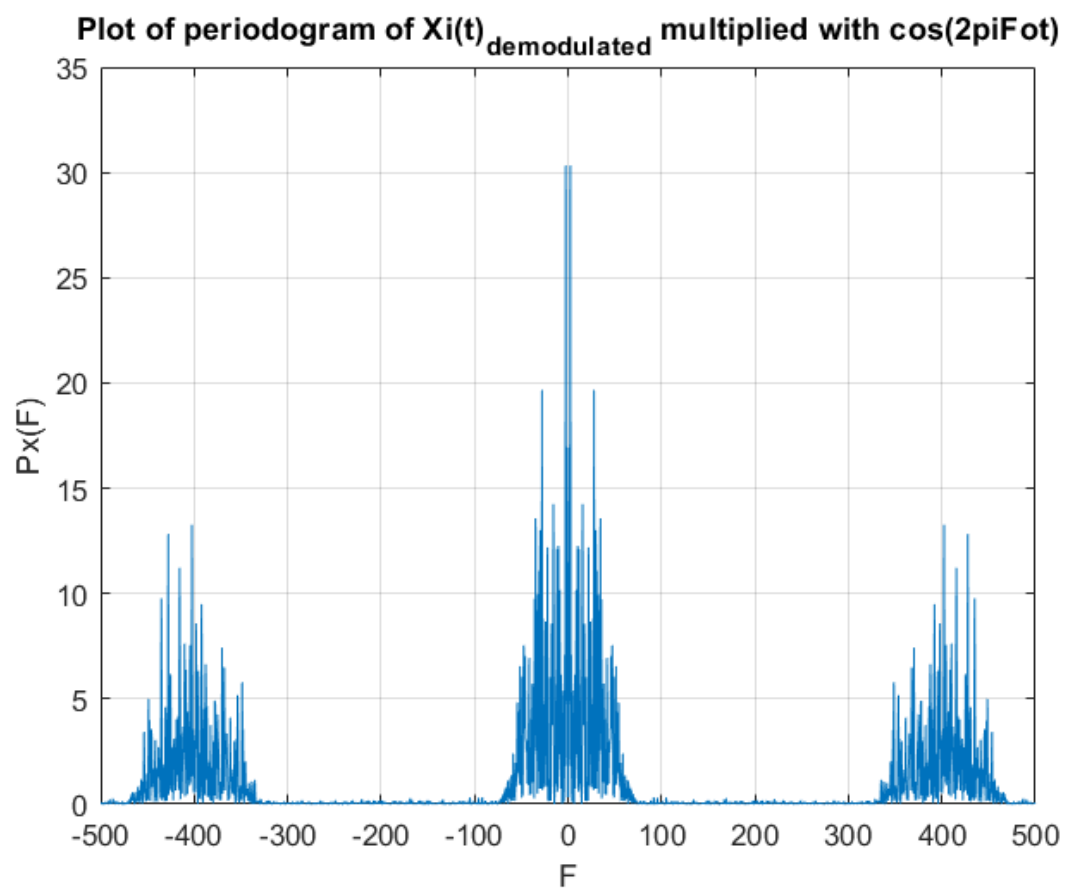
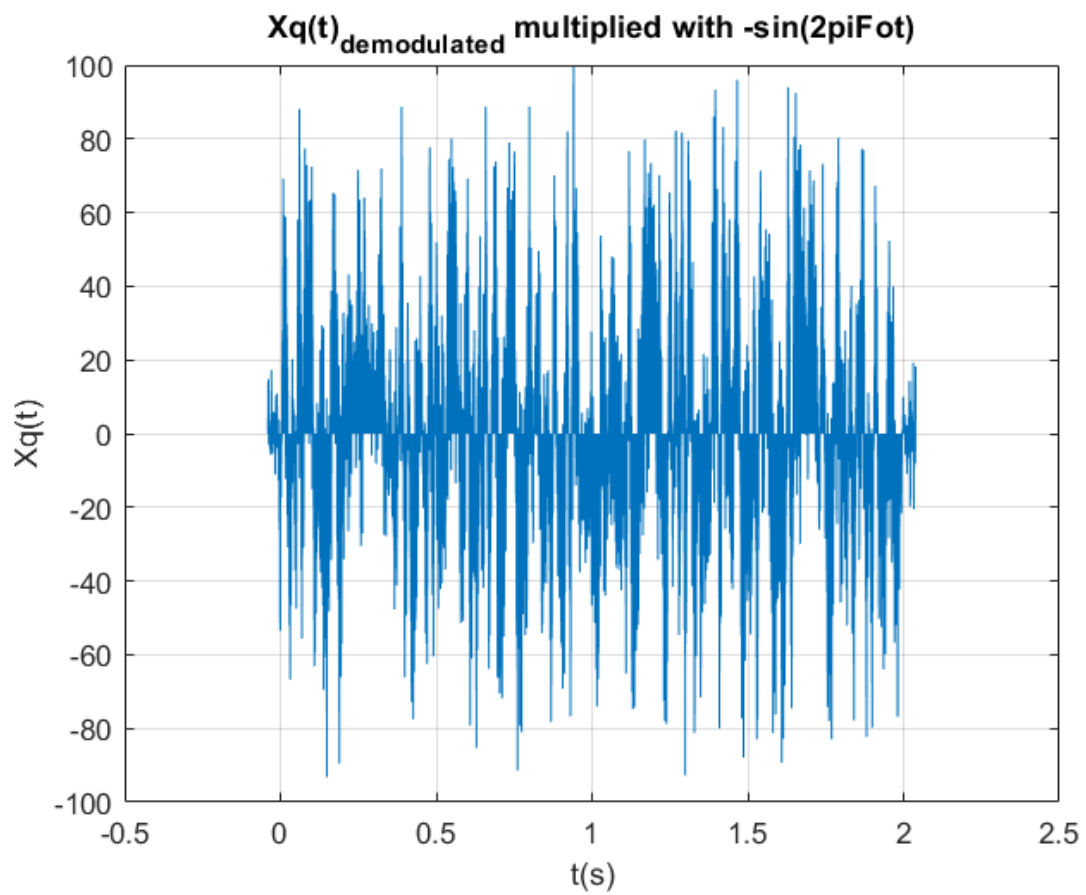
```
%Dhmiourgia enthorybhs kymatomorfhs:
```

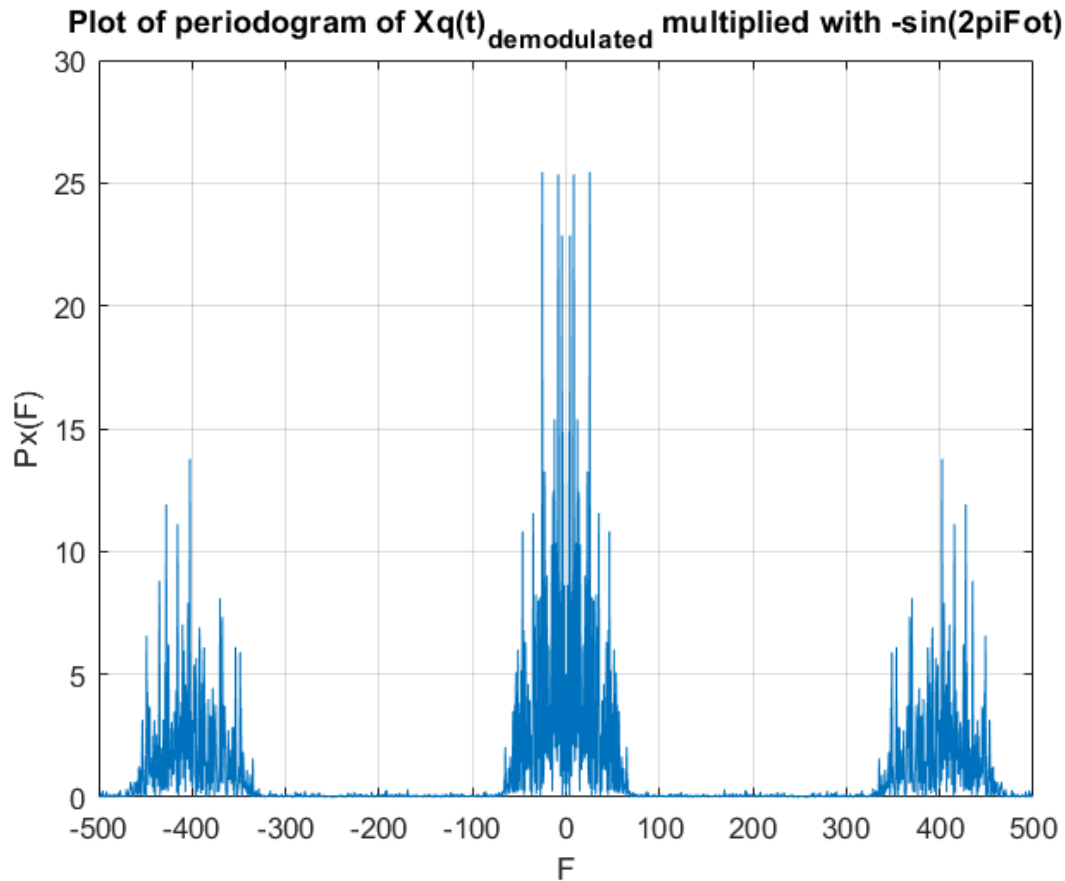
```
X_mod_noise = X_mod + Noise;
```

Ερώτημα 9

Στον δέκτη, μας ζητήθηκε να διακλαδώσουμε την ενθόρυβη κυματομορφή και να την πολλαπλασιάσουμε με φορείς $\cos(2\pi F_0 t)$ και $\sin(2\pi F_0 t)$. Από τις κυματομορφές παρατηρούμε ότι τα σήματα έχουν κέντρο τις συχνότητες $2F_0$, 0 και $-2F_0$.







Παρατίθεται ο κώδικας:

```
%-----Erwthma 9-----

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

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

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

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

figure
plot(t_conv,Xq_demod)
title('Xq(t)_d_e_m_o_d_u_l_a_t_e_d multiplied with -sin(2piFot)');
xlabel('t(s)');
```

```

ylabel('Xq(t)');
grid on

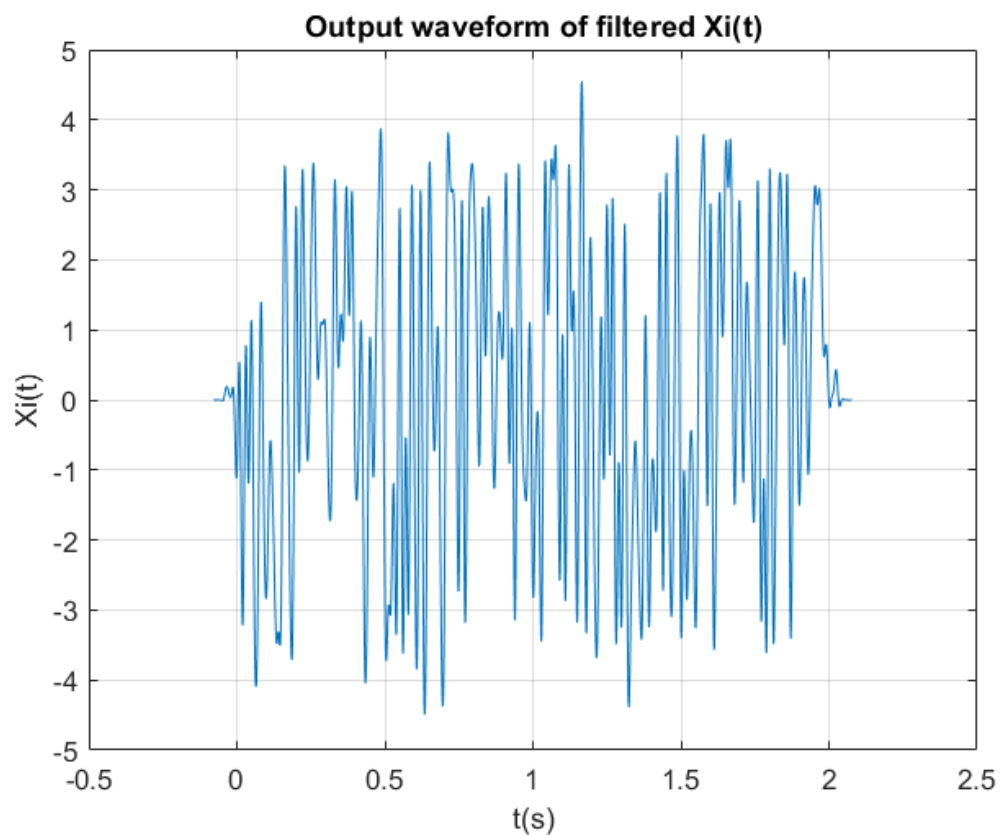
% periodogrammata Xi_demod kai Xq_demod
figure();
plot(f, Period_Xidemod);
title('Plot of periodogram of Xi(t)_d_e_m_o_d_u_l_a_t_e_d multiplied with cos(2piFot)');
xlabel('F');
ylabel('Px(F)');
grid on

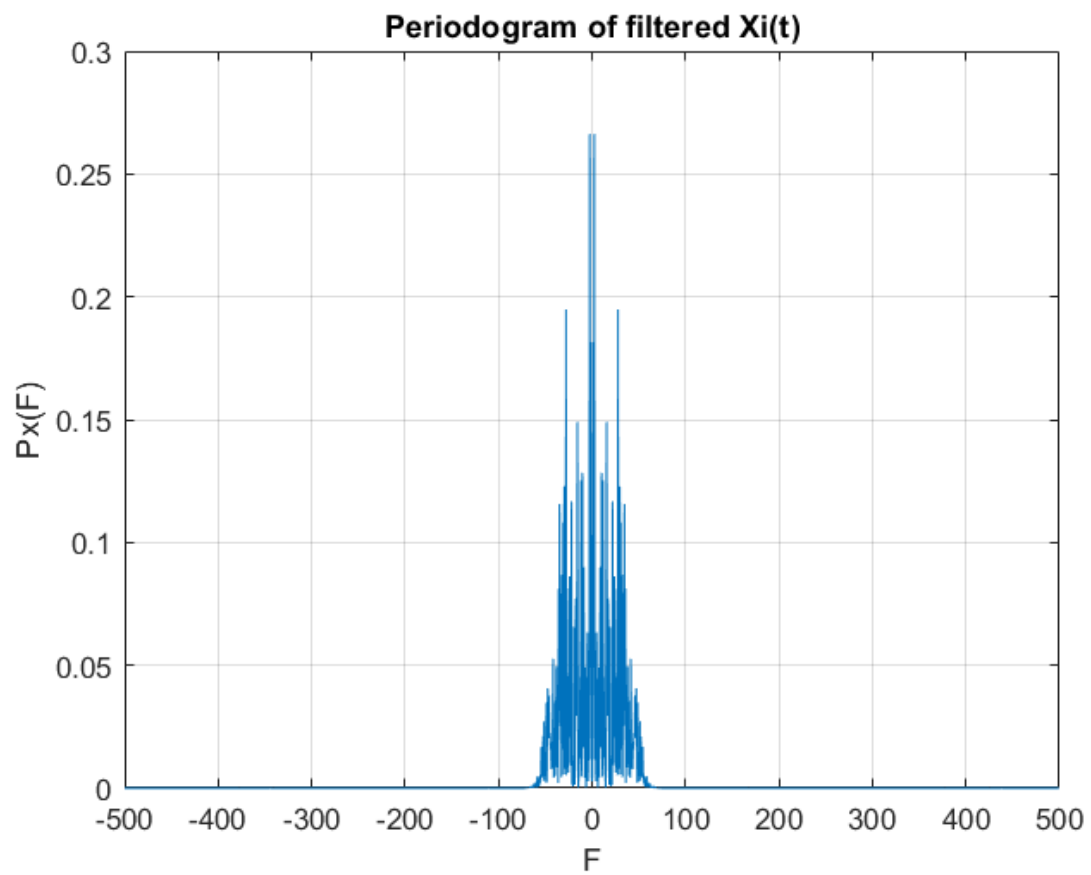
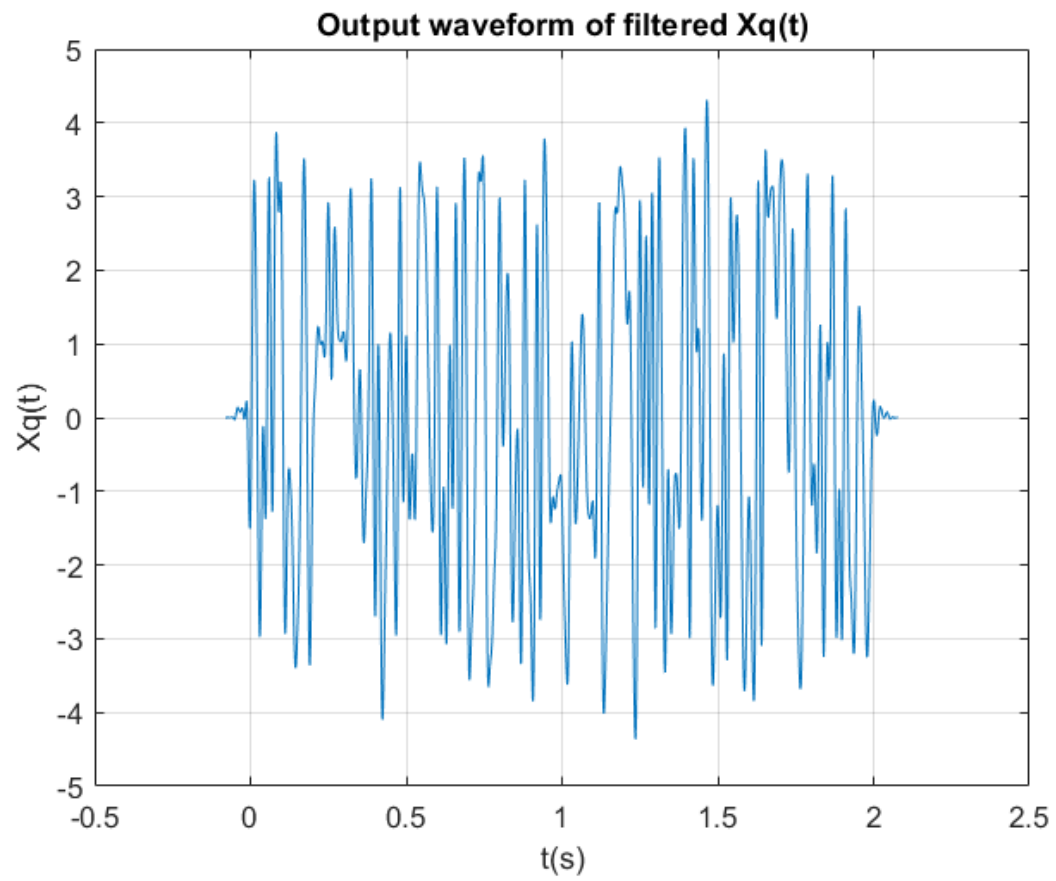
figure();
plot(f, Period_Xqdemod);
title('Plot of periodogram of Xq(t)_d_e_m_o_d_u_l_a_t_e_d multiplied with -sin(2piFot)');
xlabel('F');
ylabel('Px(F)');
grid on;
grid on

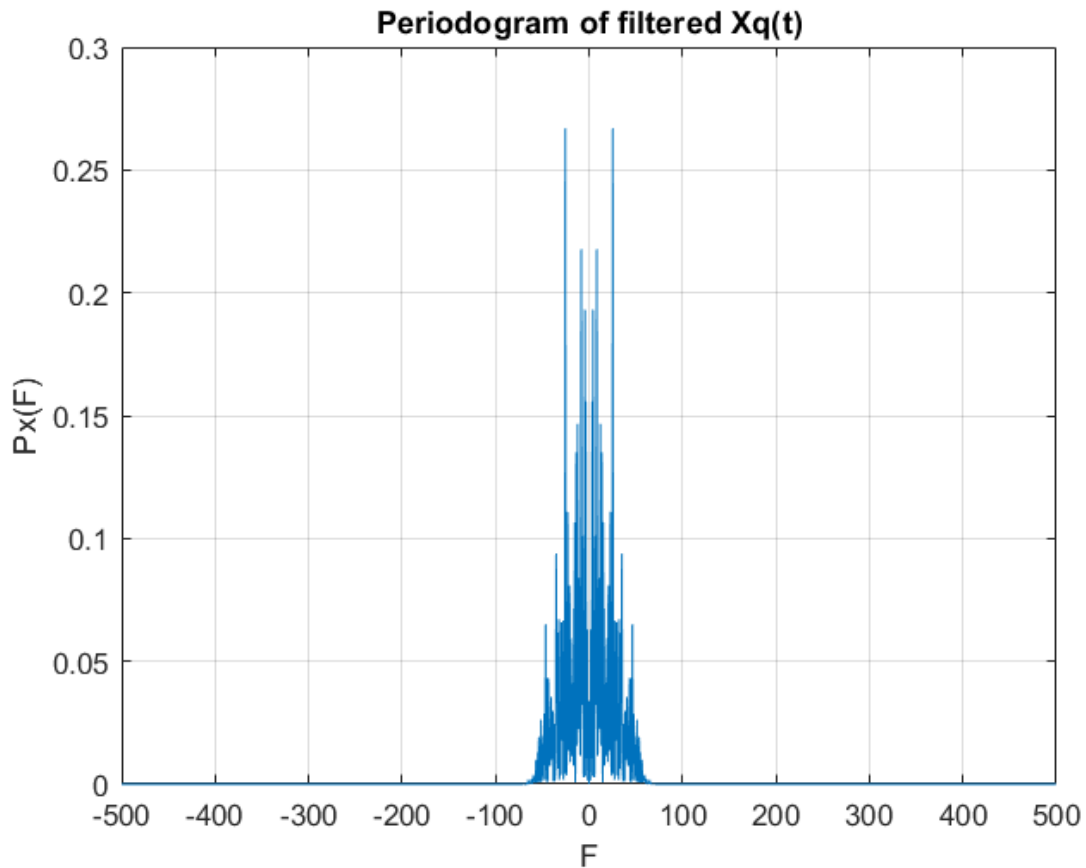
```

Ερώτημα 10

Σε αυτό το βήμα, περνάμε τις κυματομορφές από τα προσαρμοσμένα φίλτρα. Όπως βλέπουμε, έχουν αποκοπεί οι όροι στις μεγάλες συχνότητες. Οι κυματομορφές και τα περιοδογράμματα που προκύπτουν είναι τα εξής:







Παρατίθεται ο κώδικας:

```
%-----Erwthma 10-----

%De xrhsimopoiw upsample giati ta shmata einai synexh

Xi_demodulated_t = Ts*conv(Xi_demod, phi);
Xq_demodulated_t = Ts*conv(Xq_demod, phi);

Xi_demod_t_f= Ts*fftshift(fft(Xi_demodulated_t,Nf));
Ttotal8 = length(Xi_demodulated_t)*Ts;
Period_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;
Period_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('Output waveform of filtered Xi(t)');
xlabel('t(s)');
ylabel('Xi(t)');
```



```

figure
plot(t_conv2,Xq_demodulated_t)
title('Output waveform of filtered Xq(t)');
xlabel('t(s)');
ylabel('Xq(t)');
grid on

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

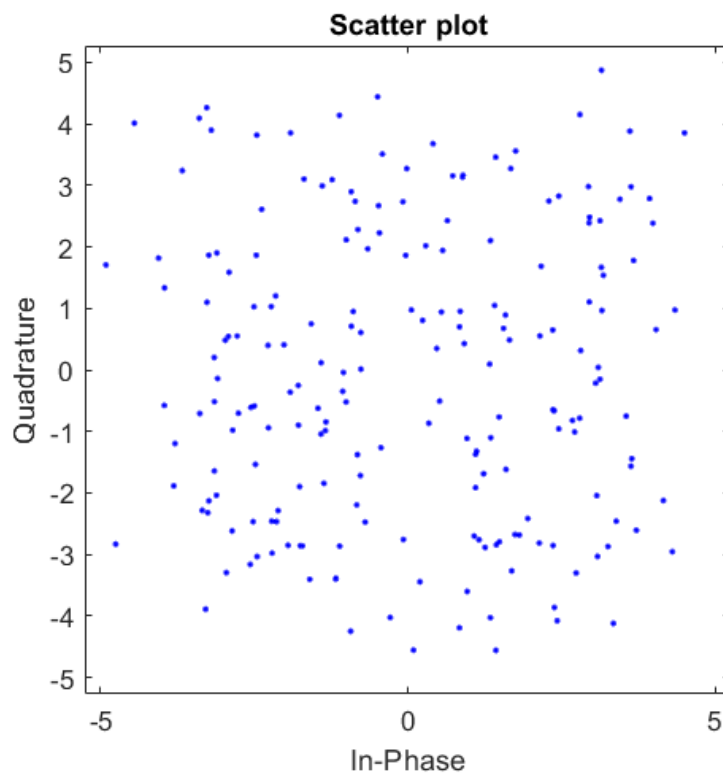
figure();
plot(F,Period_Xq_demod_t);
title('Periodogram of filtered Xq(t)');
xlabel('F');
ylabel('Px(F)');
grid on

```

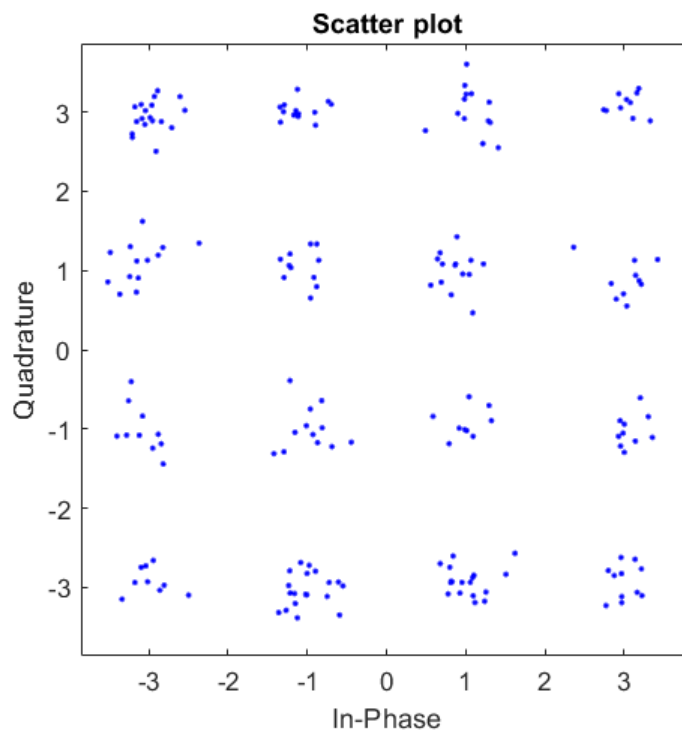
Ερώτημα 11

Στο 11^ο ερώτημα, δειγματολήπτησα την έξοδο των προσαρμοσμένων φίλτρων τις κατάλληλες χρονικές στιγμές και σχεδιάζω την ακολουθία με την εντολή scatterplot, με SNR = 10 και SNR = 20.

SNR=10



SNR=20



Όπως παρατηρούμε και από τα scatterplots, για μεγαλύτερο Signal – to – Noise Ratio υπερτερεί το σήμα και η χρήσιμη πληροφορία, έναντι του θορύβου και της άχρηστης πληροφορίας. Μάλιστα, για SNR = 20, διακρίνεται ευκρινέστερα η 16-QAM διαμόρφωση.

Παρατίθεται ο κώδικας:

```
%-----Erwthma 11-----

Xi_demod_t_sampling= Xi_demodulated_t((2*A_srrc*T/Ts) +1 : over :
length(Xi_demodulated_t)-(2*A_srrc*T/Ts));
Xq_demod_t_sampling= 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_sampling(i); %pinakas me i kai q
Sampling(i,2)=Xq_demod_t_sampling(i);
end
%length(Xi_demod_t_sampling)

scatterplot(Sampling)
```

Ερώτημα 12

Σε αυτό το ερώτημα, χρειαστήκαμε τη συνάρτηση `detect_4_PAM`, η οποία χρησιμοποιεί τον κανόνα του εγγύτερου γείτονα και αποφασίζει για την ακολουθία εισόδου 4-PAM σύμβολο προς σύμβολο. Παρατίθεται ο κώδικάς της.

```
function est_X = detect_4_PAM(Y,A)
Y_size = size(Y,2);
```

```

est_X = zeros(1,Y_size(1));
X = zeros(1,4);

X(1)=3*A;
X(2)=A;
X(3)=-A;
X(4)=-3*A;

distance=zeros(1,4);

for i=1:Y_size(1)
    distance = 0.*distance;
    for j=1:4
        distance(j) = ((X(j)-Y(i)).^2);
    end
    pos = min(distance);
    if (distance(1) == pos)
        est_X(i)= X(1);
    elseif (distance(2) == pos)
        est_X(i)= X(2);
    elseif (distance(3) == pos)
        est_X(i)= X(3);
    else
        est_X(i)= X(4);
    end
end
end

```

Έπειτα, την εφαρμόζω στις δειγματοληπτημένες εξόδους του προηγούμενου ερωτήματος.

Παρατίθεται ο κώδικας:

```

%-----Erwthma 12-----
for i=1:N
    Xi_det(i)=detect_4_PAM(Xi_demod_t_sampling(i),A); %thn efarmozw sto
demodulated
    Xq_det(i)=detect_4_PAM(Xq_demod_t_sampling(i),A);
end

```

Ερώτημα 13

Σε αυτό το ερώτημα, χρησιμοποιώντας τις ακολουθίες εισόδου και τις αποφάσεις, υπολόγισα τον αριθμό σφαλμάτων απόφασης συμβόλου. Αναλυτικότερα, έβαλα σε έναν πίνακα τις αρχικές ακολουθίες εισόδου και σε έναν δεύτερο πίνακα τις αποφάσεις, αφαίρεσα αυτούς τους δύο πίνακες μεταξύ του και έβαλα τη διαφορά σε έναν τρίτο πίνακα, και μετά συνέκρινα τις στήλες μεταξύ τους, ώστε να υπολογίσω τα σφάλματα απόφασης συμβόλου. Παρατίθεται ο κώδικας, για να γίνει κατανοητή η διαδικασία:

```

%-----Erwthma 13-----

%Pinakas sumbolwn pou esteila:
X_sent=[Xi_n ; Xq_n];

%Pinakas sumbolwn pou ekana detect:
X_detect=[Xi_det ; Xq_det];

Diafora = X_sent-X_detect; %ta afairw gia na dw poia den einai isa

```

```

error=0;

for i=1:N
    if (Diafora(1,i)~=0 || (Diafora(2,i)~=0)) %elegxw poia den einai isa me
to 0, dhladh yparxei error
        error=error+1;
    end
end

error

```

Για SNR=10, το σφάλμα υπολογίζεται ίσο με 51 ενώ για SNR=20 το σφάλμα υπολογίζεται ίσο με 0. Επαληθεύεται και σε αυτή την περίπτωση, ότι με μεγαλύτερο SNR λαμβάνουμε περισσότερη χρήσιμη πληροφορία.

Ερώτημα 14

Στο 14^ο ερώτημα, έγραψα τη συνάρτηση PAM_4_to_bits, η οποία χρησιμοποιεί την αντίστροφη απεικόνιση Gray, δηλαδή από σύμβολα σε δυάδες bits, και από τις αποφάσεις για τις ακολουθίες συμβόλων εισόδου υπολογίζει την εκτιμώμενη δυαδική ακολουθία εισόδου, ως εξής:

```

function [est_bit] = PAM_4_to_bits(X,A)
j=1;
for i=1:length(X)
    if (X(i)==3*A)
        est_bit(j)=0;
        est_bit(j+1)=0;
    elseif (X(i)==A)
        est_bit(j)=0;
        est_bit(j+1)=1;
    elseif (X(i)==-A)
        est_bit(j)=1;
        est_bit(j+1)=1;
    else
        est_bit(j)=1;
        est_bit(j+1)=0;
    end
    j=j+2;
end
end

```

Ερώτημα 15

Για τον αριθμό σφαλμάτων bit, υπολόγισα την εκτιμώμενη ακολουθία εισόδου μέσω της συνάρτησης PAM_4_to_bits, έβαλα τα αποτελέσματα σε ένα πίνακα, από τον οποίο αφαίρεσα την αρχική δυαδική ακολουθία, και στη συνέχεια υπολόγισα τα σφάλματα όπως φαίνεται παρακάτω.

```

%-----Erwthma 15-----

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

bit_est=[bit_i bit_q];

Pinakas_bit= bit_est-b; %afairw ta arxika bits
bit_error=0;

```

```

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

bit_error

```

Για SNR=10, το σφάλμα υπολογίζεται ίσο με 388 ενώ για SNR=20 το σφάλμα υπολογίζεται ίσο με 390.

ΕΡΩΤΗΜΑ Β

Στο 2^ο βήμα, υπολογίζουμε την πιθανότητα σφάλματος συμβόλου και bit με τη μέθοδο Monte Carlo. Για SNR από 0 μέχρι 16 με βήμα 2 υπολογίζω πειραματικά την πιθανότητα σφάλματος συμβόλου, επαναλαμβάνοντας όλο το ερώτημα Α για K=200 φορές, για κάθε SNR. Δημιουργώ δύο πίνακες P_symbol και P_bit, όπου εισάγω τις πειραματικές τιμές, τις οποίες υπολογίζω ως εξής:

```

sum_symbol_errors = sum_symbol_errors + error;
sum_bit_errors = sum_bit_errors + bit_error;

P_symbol(s) = sum_symbol_errors / (N*K);
P_bit(s) = sum_bit_errors / (4*N*K);

```

P_symbol	0.0037	0.0069	0.0096	0.0123	0.0140	0.0151	0.015	0.0160	0.0160
P_bit	0.0026	0.0051	0.0076	0.0102	0.0127	0.0153	0.0178	0.0203	0.0288

Για τον θεωρητικό υπολογισμό των SER, BER, έλυσα τη σχέση $10 \log_{10} \frac{P_X}{P_N} = 10 \log_{10} \frac{10A^2}{2\sigma_N^2} = \text{SNR}_{\text{dB}}$ ως προς $\frac{A}{\sigma_N}$ και χρησιμοποίησα τη συνάρτηση Q ως εξής:

```

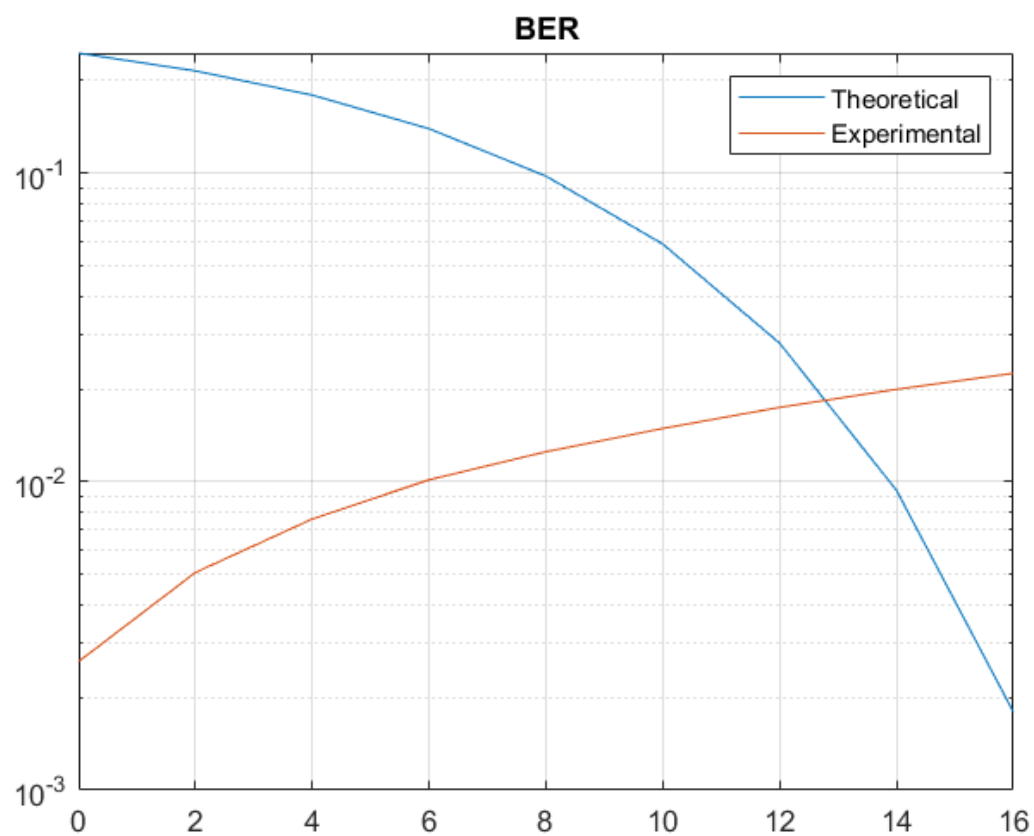
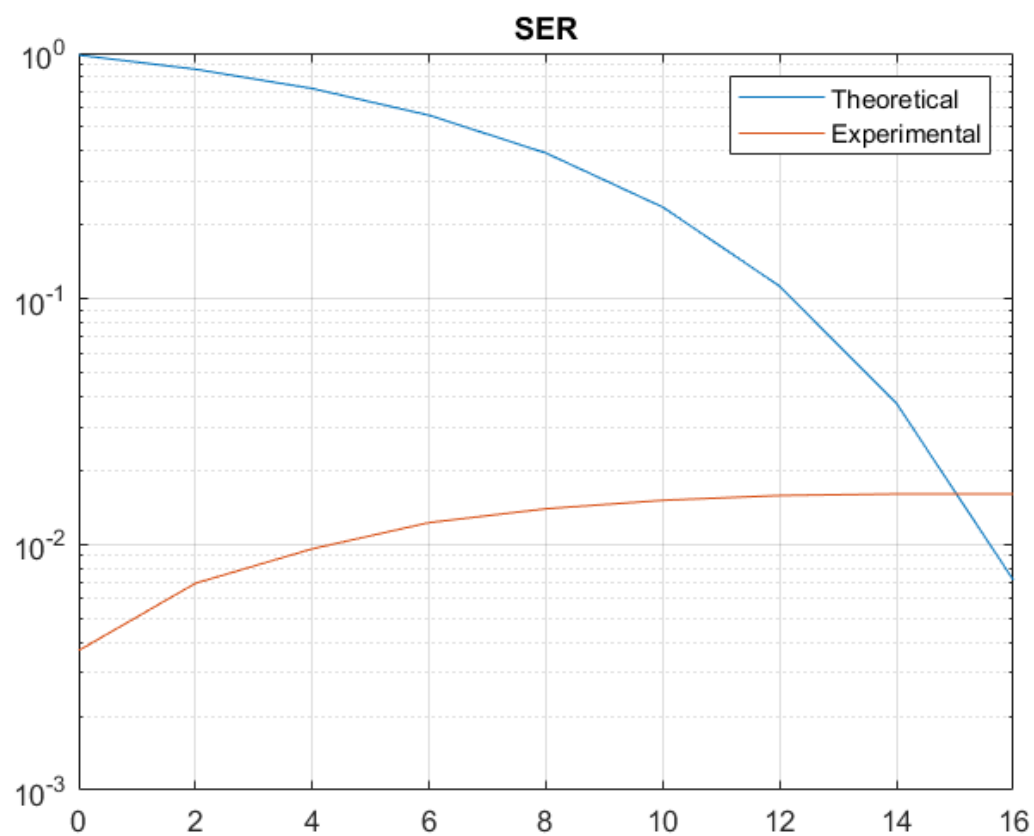
P_theor_symbol(b) = 3*Q(sqrt(0.2.*(10.^(SNR/10)))));
P_theor_bit(b) = P_theor_symbol(b)./4;
b=b+1;

```

Οι τιμές τους είναι:

P_theor_symbol	0.9821	0.8601	0.7177	0.5583	0.3919	0.2359	0.1125	0.0375	0.0072
P_theor_bit	0.2455	0.2150	0.1794	0.1396	0.0980	0.0590	0.0281	0.0094	0.0018

Οι καμπύλες δεν συμπίπτουν, καθώς έχει γίνει κάποιο λάθος στον πειραματικό υπολογισμό, και οι πειραματικές τιμές είναι πολύ μικρότερες.



Παρατίθεται ο κώδικας του Β.1

%-----Erwthma B.1-----

```
K=200;
sum_symbol_errors=0;
sum_bit_errors=0;
s=1;

for SNR=0:2:16
    for i=1:K

        b = (sign(randn(4*N,1))+1)/2;

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

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

        [phi,t]=srrc_pulse(T,Ts,A_srrc,a);

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

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

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

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

        Xi_t_f= Ts*fftshift(fft(Xi_t,Nf));
        Ttotal1 = length(Xi_t)*Ts;
        Period_Xi_t=(abs(Xi_t_f).^2)./Ttotal1;

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

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

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

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

        X_mod= Xi_mod + Xq_mod;

        X_mod_f= Ts*fftshift(fft(X_mod,Nf));
        Ttotal5 = length(X_mod)*Ts;
        Period_Xmod=(abs(X_mod_f).^2)./Ttotal5;
```

```

%Noise
Var_w = (10*(A^2))/(Ts*10^(SNR/10));

Noise= sqrt(Var_w)*randn(1,length(X_mod));

X_mod_noise = X_mod + Noise;

Xi_demod= X_mod_noise.*cos(2*pi*Fo.*t_conv);
Xq_demod= X_mod_noise.*(-sin(2*pi*Fo.*t_conv));

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

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

Xi_demodulated_t = Ts*conv(Xi_demod, phi);
Xq_demodulated_t = Ts*conv(Xq_demod, phi);

Xi_demod_t_f= Ts*fftshift(fft(Xi_demodulated_t,Nf));
Ttotal8 = length(Xi_demodulated_t)*Ts;
Period_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;
Period_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;

Xi_demod_t_sampling= Xi_demodulated_t((2*A_srrc*T/Ts) +1 :over:
length(Xi_demodulated_t)-(2*A_srrc*T/Ts));
Xq_demod_t_sampling= 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_sampling(i);
    Sampling(i,2)=Xq_demod_t_sampling(i);
end
length(Xi_demod_t_sampling)

for i=1:N
    Xi_det(i)=detect_4_PAM(Xi_demod_t_sampling(i),A);
    Xq_det(i)=detect_4_PAM(Xq_demod_t_sampling(i),A);
end

X_sent=[Xi_n ; Xq_n];
X_detect=[Xi_det ; Xq_det];
Diafora = X_sent-X_detect;

error=0;

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

```



```

        bit_i=PAM_4_to_bits(Xi_det,A);
        bit_q=PAM_4_to_bits(Xq_det,A);
        bit_est=[bit_i bit_q];
        Pinakas_bit= bit_est-b;

        bit_error=0;

        for i=1:4*N
            if(Pinakas_bit(i) ~=0 )
                bit_error=bit_error+1;
            end
        end

    end

    sum_symbol_errors = sum_symbol_errors + error;
    sum_bit_errors = sum_bit_errors + bit_error;

    P_symbol(s) = sum_symbol_errors/(N*K);
    P_bit(s) = sum_bit_errors/(4*N*K);

    s=s+1;

end
P_symbol
P_bit
b=1;
for SNR=0:2:16

    P_theor_symbol(b) = 3*Q(sqrt(0.2.*(10.^(SNR/10))));
    P_theor_bit(b) = P_theor_symbol(b)./4;
    b=b+1;

end
P_theor_symbol
P_theor_bit

%-----Erwthma B.2-----

SNR=[0:2:16];
figure();
semilogy(SNR, P_theor_symbol);
hold on;
semilogy(SNR, P_symbol);
legend ('Theoretical','Experimental')
hold off;
grid on;
title('SER');

%-----Erwthma B.3-----


figure();
semilogy(SNR, P_theor_bit);
hold on;
semilogy(SNR, P_bit);

```

```

hold off;
grid on;
legend('Theoretical','Experimental')
title('BER');

```

 Παρατίθεται όλος ο κώδικας

```

clear all;
close 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;

%-----Erwthma 1-----

%dhmiourgia duadikhhs akolouthias me 4N isopithana bits
b = (sign(randn(4*N,1))+1)/2;

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

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

%----Erwthma 4----
%Dimiourgia apokomenou palmou:
[phi,t]=srrc_pulse(T,Ts,A_srrc,a);

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

Xq_d= 1/Ts*upsample(Xq_n,over);
Xq_t = Ts*conv(Xq_d, 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('Output waveform Xi(t)');
xlabel('t(s)');
ylabel('Xi(t)');

figure();

```

```

plot(t_conv,Xq_t);
grid on;
title('Output waveform Xq(t)');
xlabel('t(s)');
ylabel('Xq(t)');

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

%Periodogrammata:

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

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

figure();
plot(f, Period_Xi_t);
title('Plot of periodogram of Xi(t)');
xlabel('F');
ylabel('Px(F)');
grid on;

figure();
plot(f, Period_Xq_t);
title('Plot of periodogram of Xq(t)');
xlabel('F');
ylabel('Px(F)');
grid on;

%-----Erwthma 5-----

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;
Period_Xi_mod=(abs(Xi_mod_f).^2)./Ttotal3;

Xq_mod_f= Ts*fftshift(fft(Xq_mod,Nf));
Ttotal4 = length(Xq_mod)*Ts;
Period_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 kai Xq_mod se plot:
figure();

```

```

plot(f, Period_Xi_mod);
grid on
title('Plot of periodogram of Xi(t)_m_o_d');
xlabel('F');
ylabel('Px(F) ');

figure();
plot(f,Period_Xq_mod);
title('Plot of periodogram of Xq(t)_m_o_d');
xlabel('F');
ylabel('Px(F) ');
grid on

%-----Erwthma 6-----

X_mod= Xi_mod + Xq_mod; %eisodos kanaliou, proshtesi

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

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

figure();
plot(f,Period_Xmod);
title('Plot of periodogram of X(t) ');
xlabel('F');
ylabel('Px(F) ');
grid on

%-----Erwthma 8-----

%Noise
SNR=20;
Var_w = (10*(A^2))/(Ts*10^(SNR/10));

Noise= sqrt(Var_w)*randn(1,length(X_mod));

%Dhmiourgia enthorybhs kymatomorfhs:
X_mod_noise = X_mod + Noise;

%-----Erwthma 9-----

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

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

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

%Kymatomorfes exodou Xi_demod kai Xq_demod

```

```

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

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

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

figure();
plot(f, Period_Xqdemod);
title('Plot of periodogram of Xq(t)_d_e_m_o_d_u_l_a_t_e_d multiplied with -
sin(2piFot)');
xlabel('F');
ylabel('Px(F)');
grid on;
grid on

%-----Erwthma 10-----

%De xrhsimopoiw upsample giati ta shmata einai synexh

Xi_demodulated_t = Ts*conv(Xi_demod, phi);
Xq_demodulated_t = Ts*conv(Xq_demod, phi);

Xi_demod_t_f= Ts*fftshift(fft(Xi_demodulated_t,Nf));
Ttotal8 = length(Xi_demodulated_t)*Ts;
Period_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;
Period_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('Output waveform of filtered Xi(t)');
xlabel('t(s)');
ylabel('Xi(t)');

figure

```

```

plot(t_conv2,Xq_demodulated_t)
title('Output waveform of filtered Xq(t)');
xlabel('t(s)');
ylabel('Xq(t)');
grid on

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

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

%-----Erwthma 11-----

Xi_demod_t_sampling= Xi_demodulated_t((2*A_srrc*T/Ts) +1 : over :
length(Xi_demodulated_t)-(2*A_srrc*T/Ts));
Xq_demod_t_sampling= 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_sampling(i); %pinakas me i kai q
Sampling(i,2)=Xq_demod_t_sampling(i);
end
%length(Xi_demod_t_sampling)

scatterplot(Sampling)

%-----Erwthma 12-----
for i=1:N
Xi_det(i)=detect_4_PAM(Xi_demod_t_sampling(i),A); %thn efarmozw sto
demodulated
Xq_det(i)=detect_4_PAM(Xq_demod_t_sampling(i),A);
end

%-----Erwthma 13-----

%Pinakas sumbolwn pou esteila:
X_sent=[Xi_n ; Xq_n];

%Pinakas sumbolwn pou ekana detect:
X_detect=[Xi_det ; Xq_det];

Diafora = X_sent-X_detect; %ta afairw gia na dw poia den einai isa
error=0;

for i=1:N
if(Diafora(1,i)~=0 || (Diafora(2,i)~=0)) %elegxw poia den einai isa me
to 0, dhladh yparxei error
error=error+1;
end
end
end

```

```

error

%-----Erwthma 15-----

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

bit_est=[bit_i bit_q];

Pinakas_bit= bit_est-b; %afairw ta arxika bits
bit_error=0;

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

bit_error

%-----Erwthma B.1-----

K=200;
sum_symbol_errors=0;
sum_bit_errors=0;
s=1;

for SNR=0:2:16
    for i=1:K

        b = (sign(randn(4*N,1))+1)/2;

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

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

        [phi,t]=srrc_pulse(T,Ts,A_srrc,a);

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

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

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

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

        Xi_t_f= Ts*fftshift(fft(Xi_t,Nf));
        Ttotal1 = length(Xi_t)*Ts;
        Period_Xi_t=(abs(Xi_t_f).^2)./Ttotal1;

        Xq_t_f= Ts*fftshift(fft(Xq_t,Nf));
        Ttotal2 = length(Xq_t)*Ts;
        Period_Xq_t=(abs(Xq_t_f).^2)./Ttotal2;
    end
end

```

```

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

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

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

X_mod= Xi_mod + Xq_mod;

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

%Noise
Var_w = (10*(A^2))/(Ts*10^(SNR/10));

Noise= sqrt(Var_w)*randn(1,length(X_mod));

X_mod_noise = X_mod + Noise;

Xi_demod= X_mod_noise.*cos(2*pi*Fo.*t_conv);
Xq_demod= X_mod_noise.*(-sin(2*pi*Fo.*t_conv));

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

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

Xi_demodulated_t = Ts*conv(Xi_demod, phi);
Xq_demodulated_t = Ts*conv(Xq_demod, phi);

Xi_demod_t_f= Ts*fftshift(fft(Xi_demodulated_t,Nf));
Ttotal8 = length(Xi_demodulated_t)*Ts;
Period_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;
Period_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;

Xi_demod_t_sampling= Xi_demodulated_t((2*A_srrc*T/Ts) +1 :over:
length(Xi_demodulated_t)-(2*A_srrc*T/Ts));
Xq_demod_t_sampling= 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_sampling(i);
    Sampling(i,2)=Xq_demod_t_sampling(i);
end

```



```

length(Xi_demod_t_sampling)

for i=1:N
    Xi_det(i)=detect_4_PAM(Xi_demod_t_sampling(i),A);
    Xq_det(i)=detect_4_PAM(Xq_demod_t_sampling(i),A);
end

X_sent=[Xi_n ; Xq_n];
X_detect=[Xi_det ; Xq_det];
Diafora = X_sent-X_detect;

error=0;

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

bit_i=PAM_4_to_bits(Xi_det,A);
bit_q=PAM_4_to_bits(Xq_det,A);
bit_est=[bit_i bit_q];
Pinakas_bit= bit_est-b;

bit_error=0;

for i=1:4*N
    if(Pinakas_bit(i) ~=0 )
        bit_error=bit_error+1;
    end
end

end

sum_symbol_errors = sum_symbol_errors + error;
sum_bit_errors = sum_bit_errors + bit_error;

P_symbol(s) = sum_symbol_errors/(N*K);
P_bit(s) = sum_bit_errors/(4*N*K);

s=s+1;

end
P_symbol
P_bit
b=1;
for SNR=0:2:16

    P_theor_symbol(b) = 3*Q(sqrt(0.2.*(10.^(SNR/10)))));
    P_theor_bit(b) = P_theor_symbol(b)./4;
    b=b+1;

end
P_theor_symbol
P_theor_bit

%-----Erwthma B.2-----

```

```

SNR=[0:2:16];
figure();
semilogy(SNR, P_theor_symbol);
hold on;
semilogy(SNR, P_symbol);
legend ('Theoretical','Experimental')
hold off;
grid on;
title('SER');

```

```

%-----Erwthma B.3-----

```

```

figure();
semilogy(SNR, P_theor_bit);
hold on;
semilogy(SNR, P_bit);
hold off;
grid on;
legend ('Theoretical','Experimental')
title('BER');

```