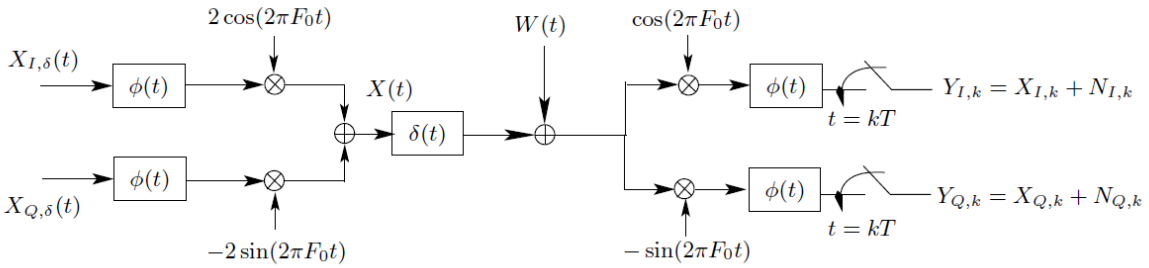
Πολυτεχνείο Κρήτης
Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Τηλεπικοινωνιακά Συστήματα Ι
ΤΗΛ 302
3η Άσκηση
Αναφορά

Ομάδα 112:
Παπαματθαιάκη Ηλέκτρα-Δέσποινα (2018030106)

Σε αυτή την άσκηση, προσομοιώθηκε το τηλεπικοινωνιακό σύστημα του παρακάτω σχήματος, με διαμόρφωση 16-PSK, και θα μελετήθηκε η απόδοσή του.



1.

Έχοντας ως δεδομένο ότι N = 100 δημιουργήθηκε δυαδική ακολουθία με 4N ισοπίθανα bits.

Κώδικας Matlab:

```
% 1.
N = 100;
bit_seq = (sign(randn(4*N, 1)) + 1)/2;
```

2.

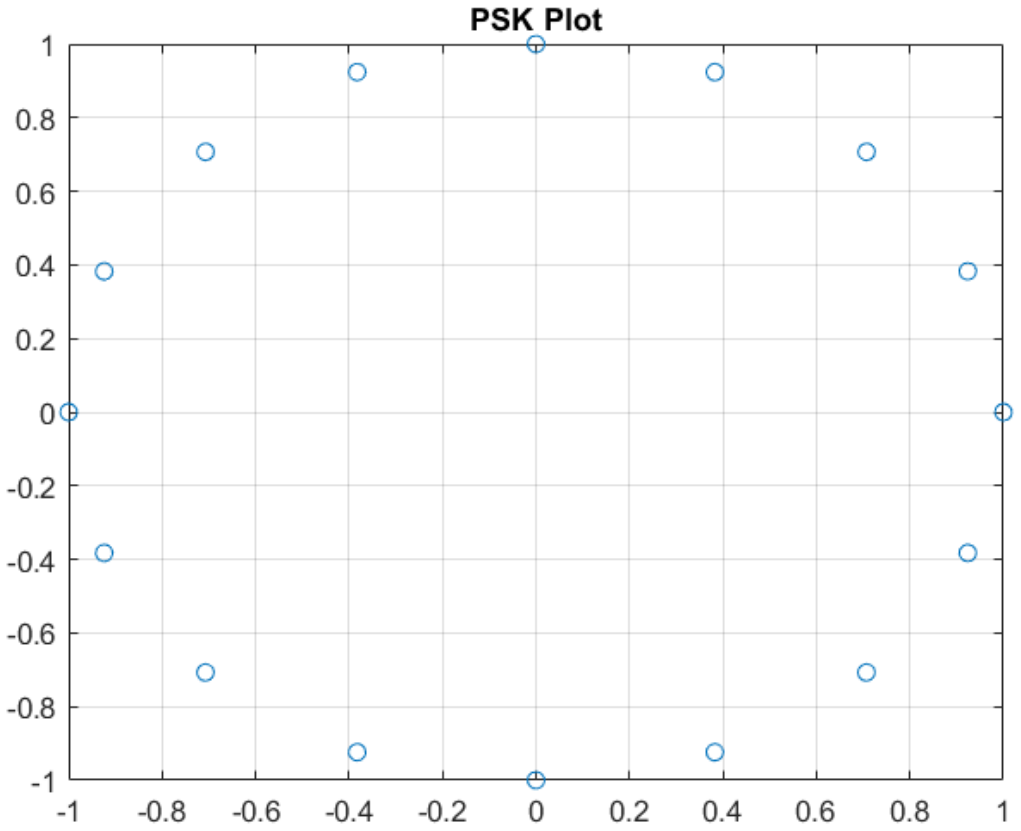
Συντάχθηκε η συνάρτηση bits_to_PSK_16, η οποία παίρνει ως όρισμα μια ακολουθία από bits και την επιστρέφει σύμφωνα με την κωδικοποίηση Gray σε ακολουθία 16-PSK συμβόλων με στοιχεία τα δυσδιάστατα διανύσματα:

$$\mathbf{X}_n = \begin{bmatrix} X_{I,n} \\ X_{Q,n} \end{bmatrix}, \quad \text{για } n = 0, \ldots, N-1.$$

Επιπλέον, λήφθηκε υπόψη ότι κάθε διάνυσμα Xn, για n = 0, . . . ,N − 1, παίρνει τιμές από το αλφάβητο 16-PSK{x0, . . . , x15} με

$$\mathbf{x}_m = \begin{bmatrix} \cos\left(\frac{2\pi m}{16}\right) \\ \sin\left(\frac{2\pi m}{16}\right) \end{bmatrix}, \quad \text{για } m = 0, \ldots, 15.$$

Τέλος, απεικονίστηκε η PSK.



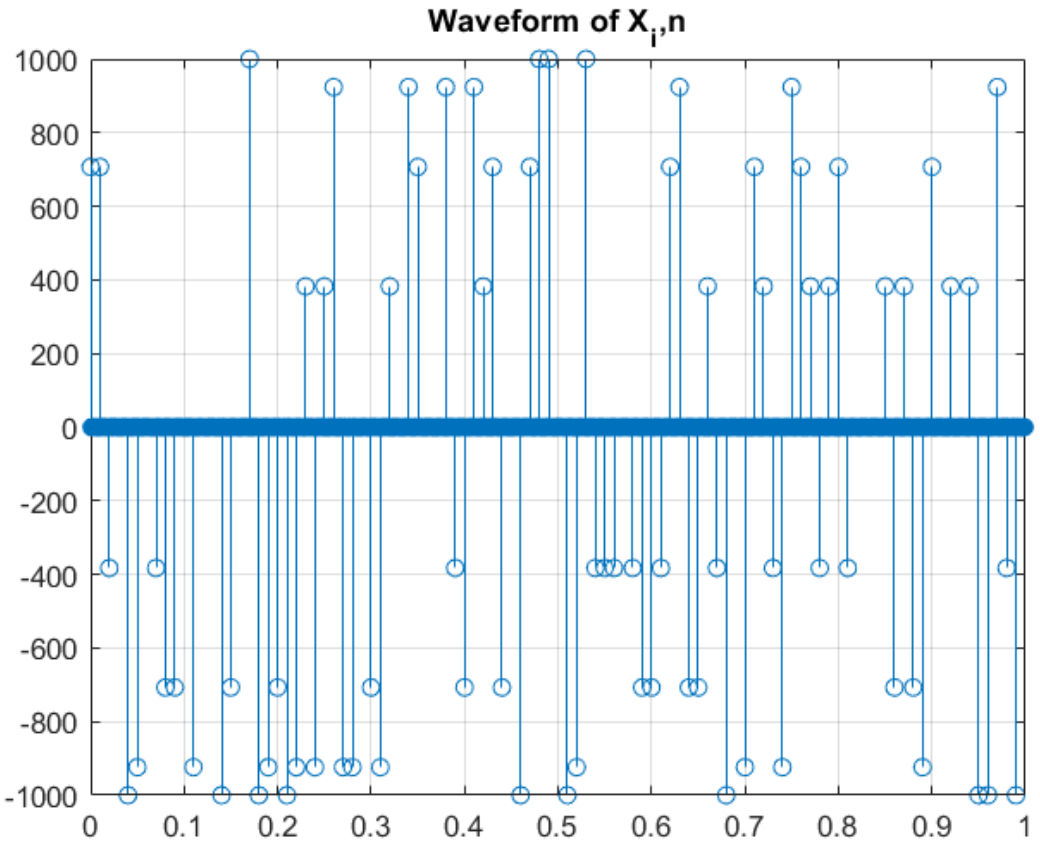
Κώδικας Matlab:

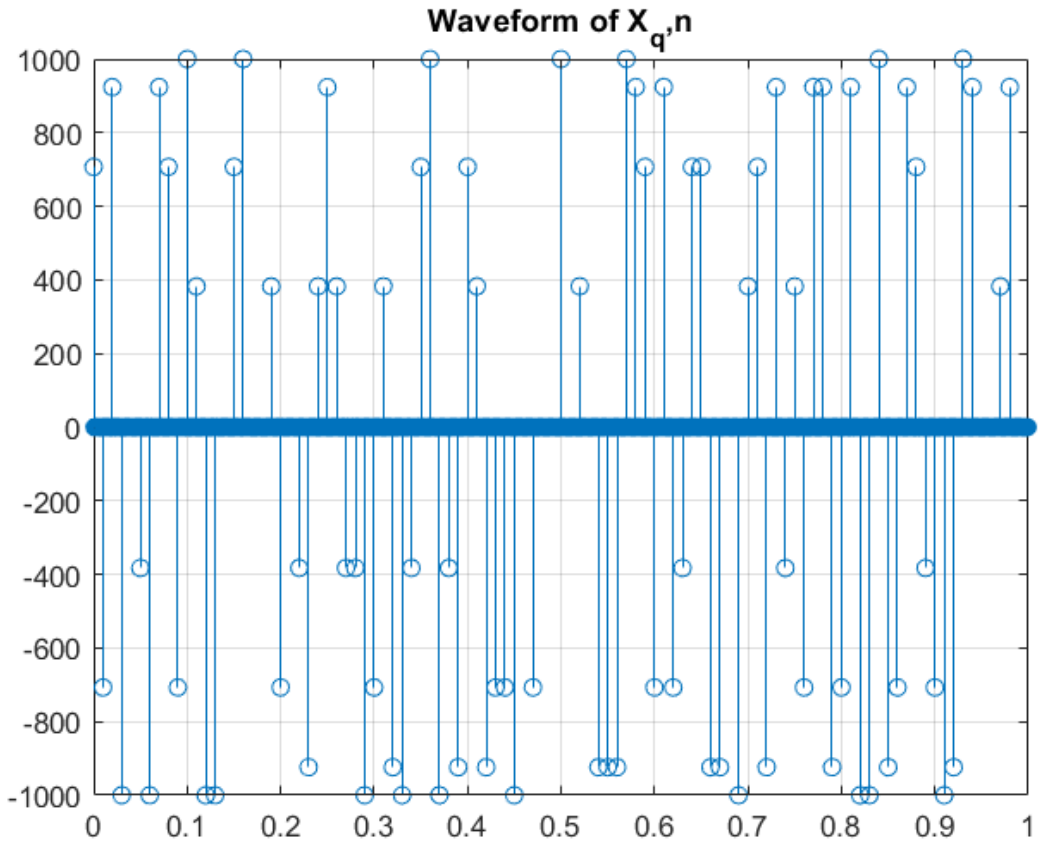
```
function X = bits_to_PSK_16(bit_seq)
b = bit_seq;
len = length(b);
k = 0;
m = 0;
for i=1:4:len
k=k+1;
% Binary to Gray Code to Decimal
% 0 0 0 0 -> 0 0 0 0 -> 0
if (b(i) == 0 && b(i+1) == 0 && b(i+2) == 0 && b(i+3) == 0)
m = 0;
% 0 0 0 1 -> 0 0 0 1 -> 1
elseif (b(i) == 0 && b(i+1) == 0 && b(i+2) == 0 && b(i+3) == 1)
m = 1;
% 0 0 1 0 -> 0 0 1 1 -> 3
elseif (b(i) == 0 && b(i+1) == 0 && b(i+2) == 1 && b(i+3) == 0)
m = 3;
% 0 0 1 1 -> 0 0 1 1 -> 2
elseif (b(i) == 0 && b(i+1) == 0 && b(i+2) == 1 && b(i+3) == 1)
m = 2;
% 0 1 0 0 -> 0 1 1 0 -> 6
```

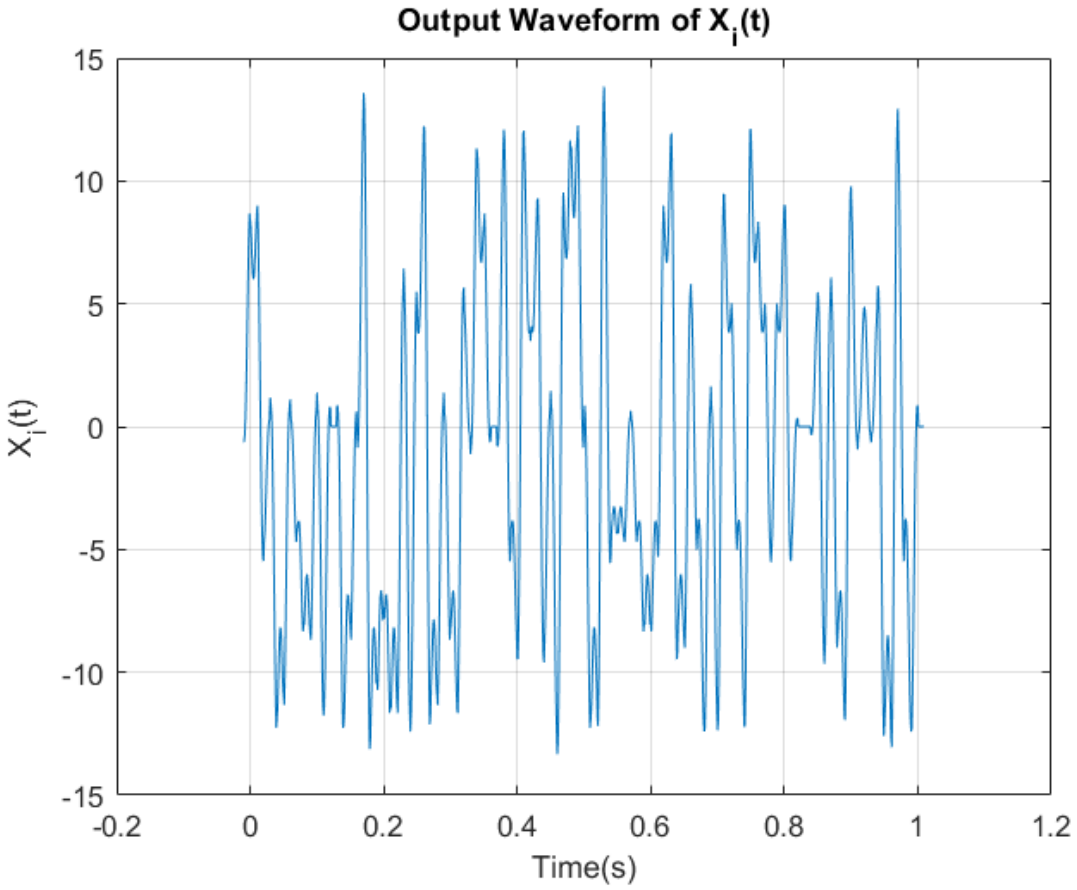
```
elseif (b(i) == 0 && b(i+1) == 1 && b(i+2) == 0 && b(i+3) == 0)
m = 6;
% 0 1 0 1 -> 0 1 1 1 -> 7
elseif (b(i) == 0 && b(i+1) == 1 && b(i+2) == 0 && b(i+3) == 1)
m = 7;
% 0 1 1 0 -> 0 1 0 1 -> 5
elseif (b(i) == 0 && b(i+1) == 1 && b(i+2) == 1 && b(i+3) == 0)
m = 5;
% 0 1 1 1 -> 0 1 0 0 -> 4
elseif (b(i) == 0 && b(i+1) == 1 && b(i+2) == 1 && b(i+3) == 1)
m = 4;
% 1 0 0 0 -> 1 1 0 0 -> 12
elseif (b(i) == 1 && b(i+1) == 0 && b(i+2) == 0 && b(i+3) == 0)
m = 12;
% 1 0 0 1 -> 1 1 0 1 -> 13
elseif (b(i) == 1 && b(i+1) == 0 && b(i+2) == 0 && b(i+3) == 1)
m = 13;
% 1 0 1 0 -> 1 1 1 1 -> 15
elseif (b(i) == 1 && b(i+1) == 0 && b(i+2) == 1 && b(i+3) == 0)
m = 15;
% 1 0 1 1 -> 1 1 1 0 -> 14
elseif (b(i) == 1 && b(i+1) == 0 && b(i+2) == 1 && b(i+3) == 1)
m = 14;
% 1 1 0 0 -> 1 0 1 0 -> 10
elseif (b(i) == 1 && b(i+1) == 1 && b(i+2) == 0 && b(i+3) == 0)
m = 10;
% 1 1 0 1 -> 1 0 1 1 -> 11
elseif (b(i) == 1 && b(i+1) == 1 && b(i+2) == 0 && b(i+3) == 1)
m = 11;
% 1 1 1 0 -> 1 0 0 1 -> 9
elseif (b(i) == 1 && b(i+1) == 1 && b(i+2) == 1 && b(i+3) == 0)
m = 9;
% 1 1 1 1 -> 1 0 0 0 -> 8
else
m = 8;
end
X(1,k) = cos((2*pi*m)/16);
X(2,k) = sin((2*pi*m)/16);
end
end

% 2.
X = bits_to_PSK_16(bit_seq);
Xi = X(1,:);
Xq = X(2,:);
% Plot PSK Symbols
figure()
plot(Xi,Xq,'o')
%plot(Xi,Xq)
set(gcf,'color', 'w');
grid on;
title('PSK Plot')
```

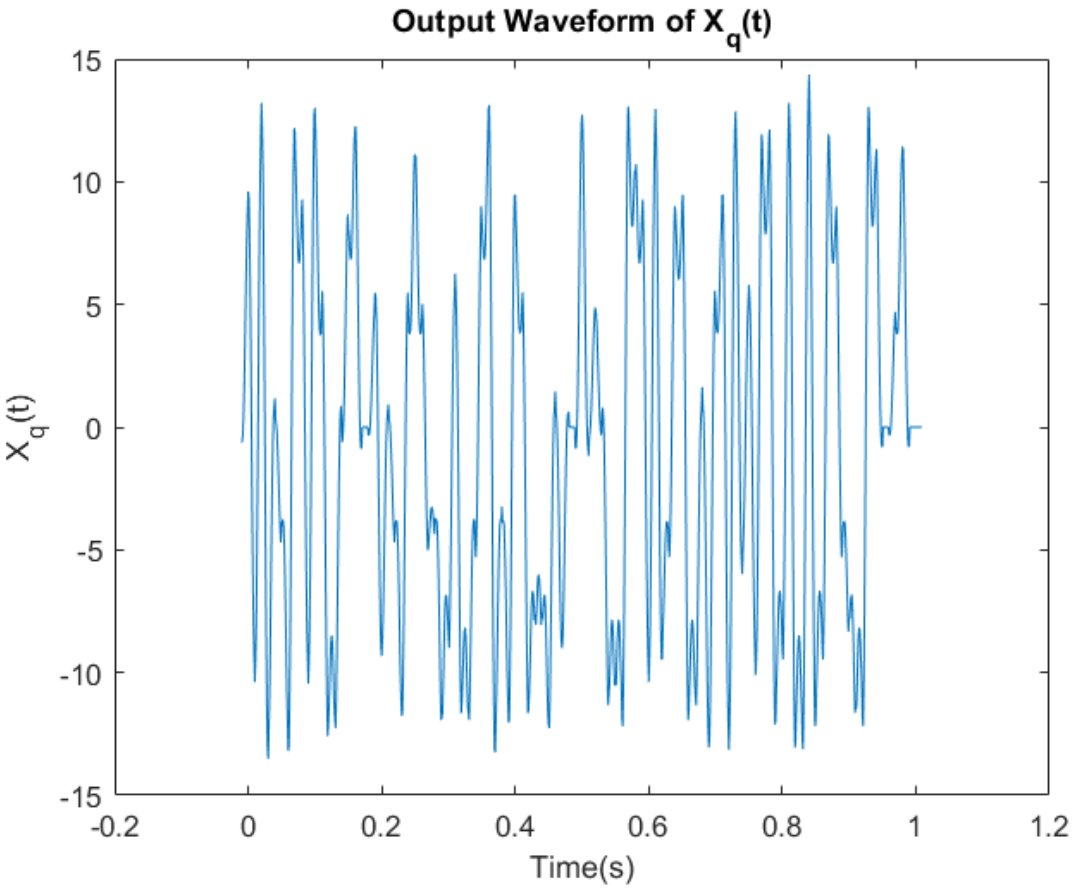
3.

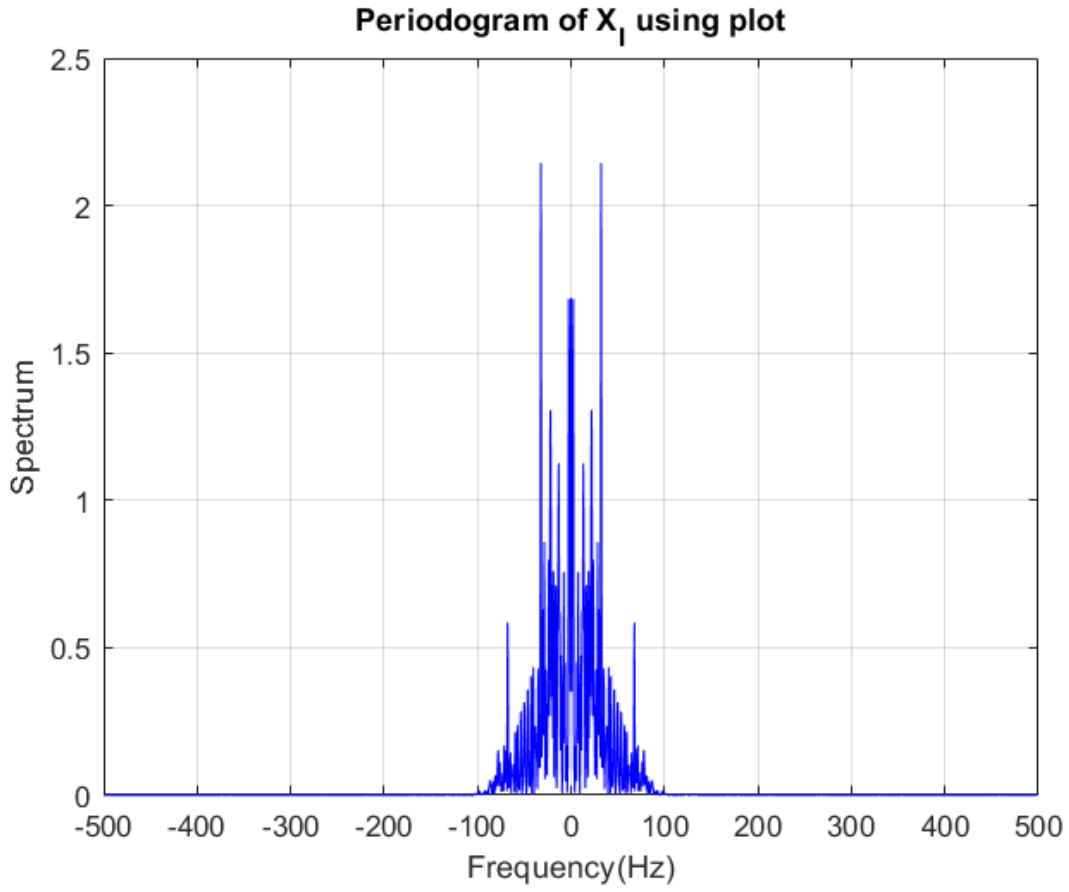
Στη συνέχεια, "περάστηκαν" οι ακολουθίες {Xi,n} και {Xq,n} από τα SRRC φίλτρα μορφοποίησης έχοντας περίοδο συμβόλου $T = 10^{-2}s$, $over = 10$,$T_s = \frac{T}{over}$, και σχεδιάστηκαν οι κυματομορφές εξόδου και τα περιοδογράμματά τους. Τα αποτελέσματα παρουσιάζονται παρακάτω:

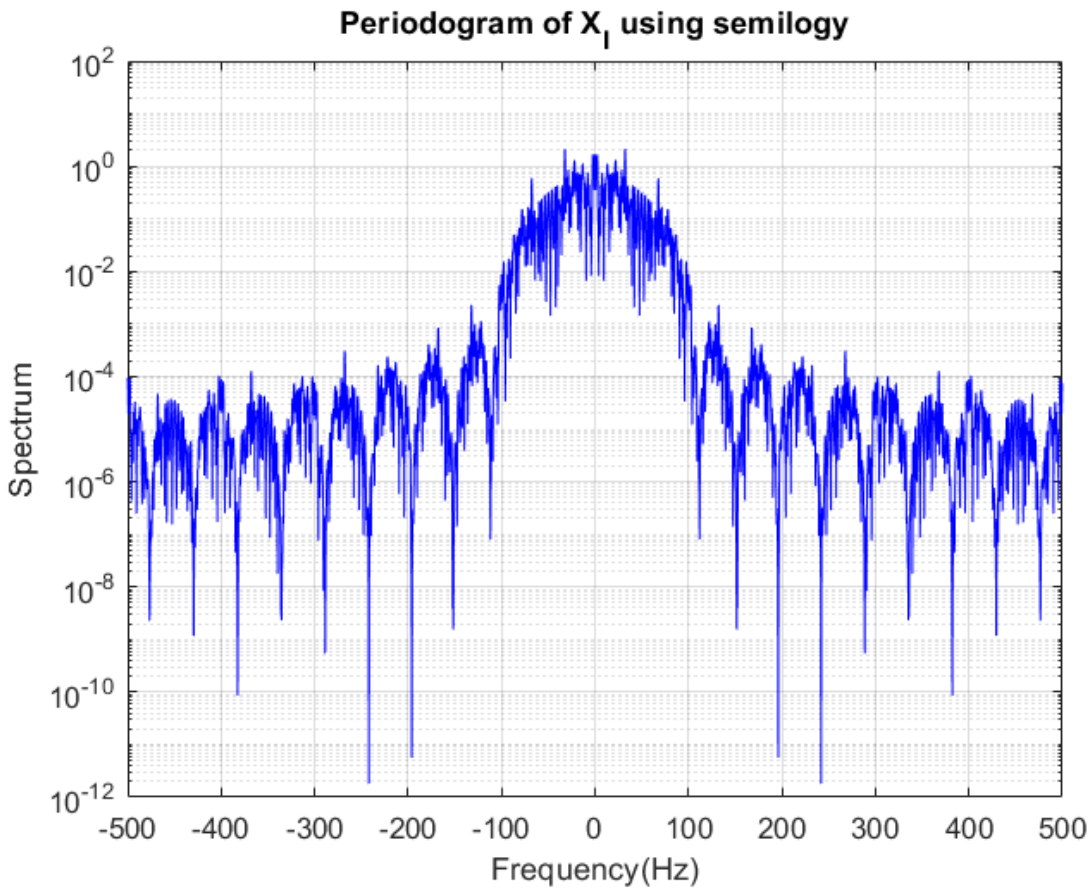


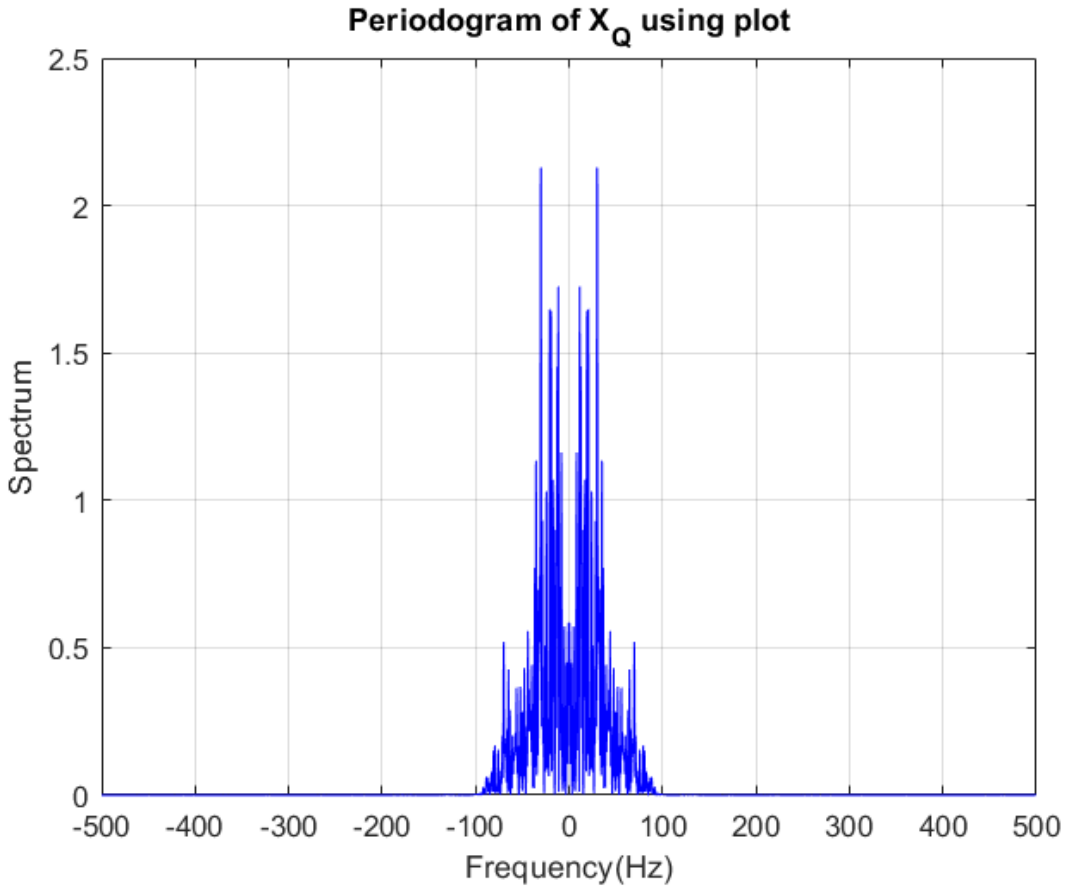

Waveform of X_q,n
1000
800
600
400
200
0
-200
-400
-600
-800
-1000
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

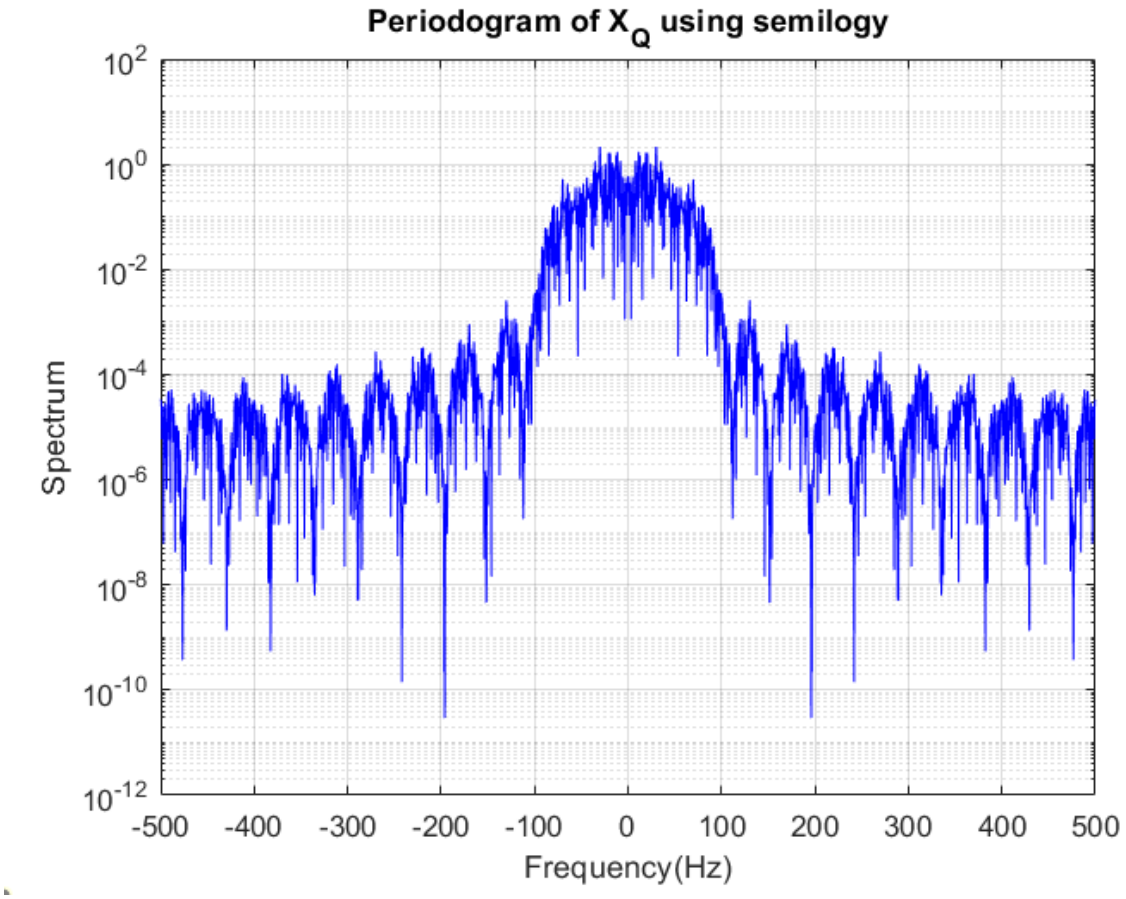

Output Waveform of X_i(t)
X_i(t)
15
10
5
0
-5
-10
-15
-0.2
0
0.2
0.4
0.6
0.8
1
1.2
Time(s)


Output Waveform of X_q(t)
X_q(t)
Time(s)
15
10
5
0
-5
-10
-15
-0.2
0
0.2
0.4
0.6
0.8
1
1.2


Periodogram of X_I using plot
Spectrum
Frequency(Hz)
2.5
2
1.5
1
0.5
0
-500
-400
-300
-200
-100
0
100
200
300
400
500


Periodogram of X_I using semilogy
Spectrum
Frequency(Hz)
10^2
10^0
10^-2
10^-4
10^-6
10^-8
10^-10
10^-12
-500 -400 -300 -200 -100 0 100 200 300 400 500


Periodogram of X_Q using plot
Spectrum
Frequency(Hz)
2.5
2
1.5
1
0.5
0
-500 -400 -300 -200 -100 0 100 200 300 400 500



Κώδικας Matlab:

```
% 3.
% Parameters
T = 0.01;
over = 10;
Ts = T/over;
a = 1;
A = 1;
Nf = 4096;
Fs = 1/Ts;
Fx = -Fs/2:Fs/Nf:Fs/2-Fs/Nf;
% upsampling Xi, Xq
X_delta_i = 1/Ts * upsample(Xi, over);
X_delta_q = 1/Ts * upsample(Xq, over);
t = 0:Ts:N*Ts*over-Ts;
l = t(1):Ts:t(end);
figure();
stem(t,X_delta_i);
title('Waveform of X_i,n');
grid on;
set(gcf,'color', 'w');
figure();
stem(t,X_delta_q);
title('Waveform of X_q,n');
```

```
grid on;
set(gcf,'color', 'w');
% creation of SRRC Pulse
[phi, t1] = srrc_pulse(T, over, A, a);
% Convolution
X_I = conv(X_delta_i,phi)*Ts;
tX_I = t1(1)+t(1):Ts:t1(end)+t(end);
X_Q = conv(X_delta_q,phi)*Ts;
tX_Q = t1(1)+t(1):Ts:t1(end)+t(end);
figure();
plot(tX_I,X_I);
title('Output Waveform of X_i(t)');
ylabel('X_i(t)');
xlabel('Time(s)');
grid on;
set(gcf,'color', 'w');
figure();
plot(tX_Q,X_Q);
title('Output Waveform of X_q(t)');
ylabel('X_q(t)');
xlabel('Time(s)');
set(gcf,'color', 'w');
% Plot the periodograms
Ttotal_I = tX_I(end) - tX_I(1);
S_I = fftshift(fft(X_I,Nf)* Ts);
P_I = ((abs(S_I)) .^ 2) / Ttotal_I;
Ttotal_Q = tX_Q(end) - tX_Q(1);
S_Q = fftshift(fft(X_Q,Nf)* Ts);
P_Q = ((abs(S_Q)) .^ 2) / Ttotal_Q;
figure();
plot(Fx, P_I , 'blue');
title('Periodogram of X_I using plot');
ylabel('Spectrum');
xlabel('Frequency(Hz)');
grid on;
set(gcf,'color', 'w');
figure();
semilogy(Fx, P_I , 'blue');
title('Periodogram of X_I using semilogy');
ylabel('Spectrum');
xlabel('Frequency(Hz)');
grid on;
set(gcf,'color', 'w');
figure();
plot(Fx, P_Q , 'blue');
title('Periodogram of X_Q using plot');
ylabel('Spectrum');
xlabel('Frequency(Hz)');
grid on;
set(gcf,'color', 'w');
figure();
semilogy(Fx, P_Q , 'blue');
title('Periodogram of X_Q using semilogy');
```

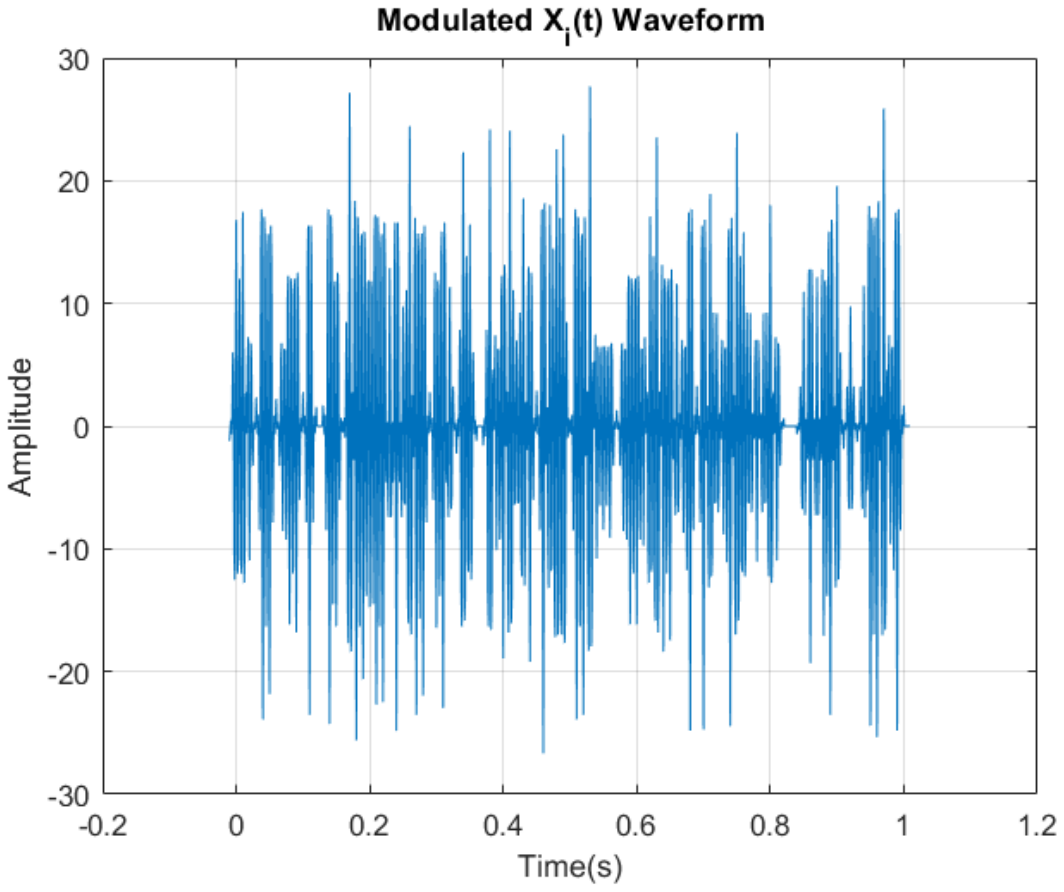
```
ylabel('Spectrum');
xlabel('Frequency(Hz)');
grid on;
set(gcf,'color', 'w');
```

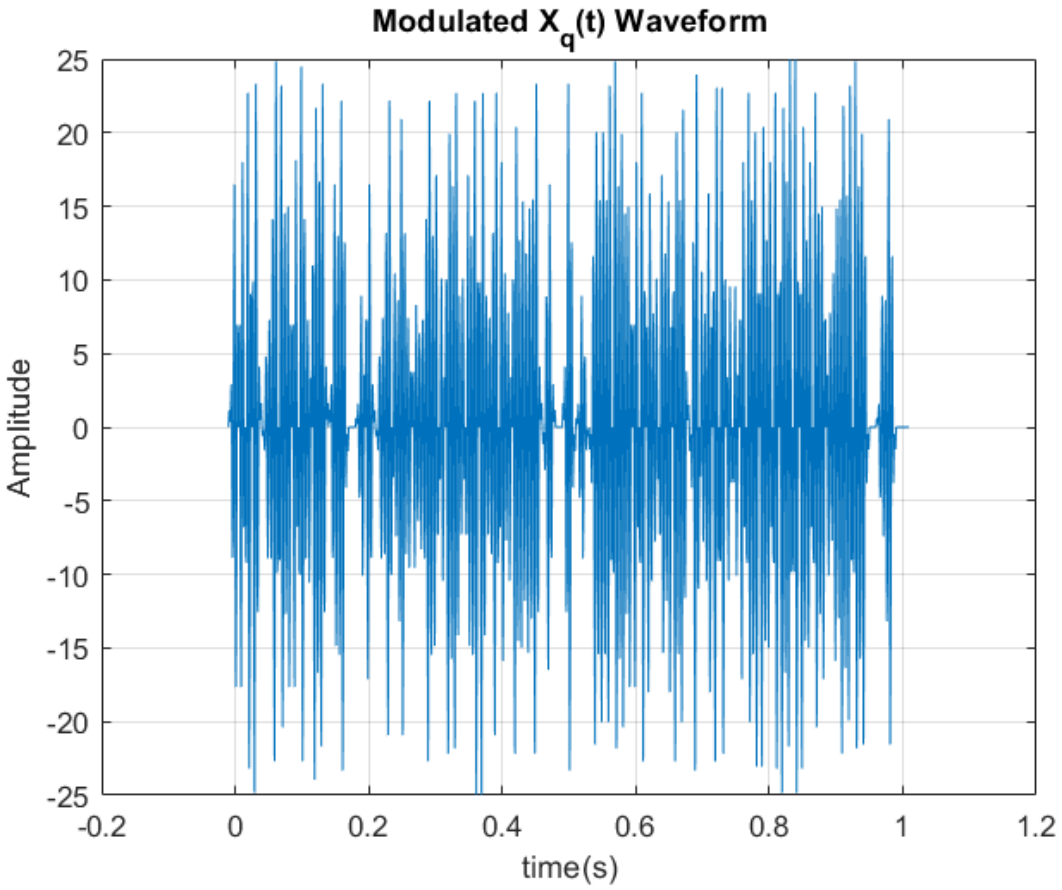
4.

Έγινε διαμόρφωση των ακολουθιών με τον ακόλουθο τρόπο:

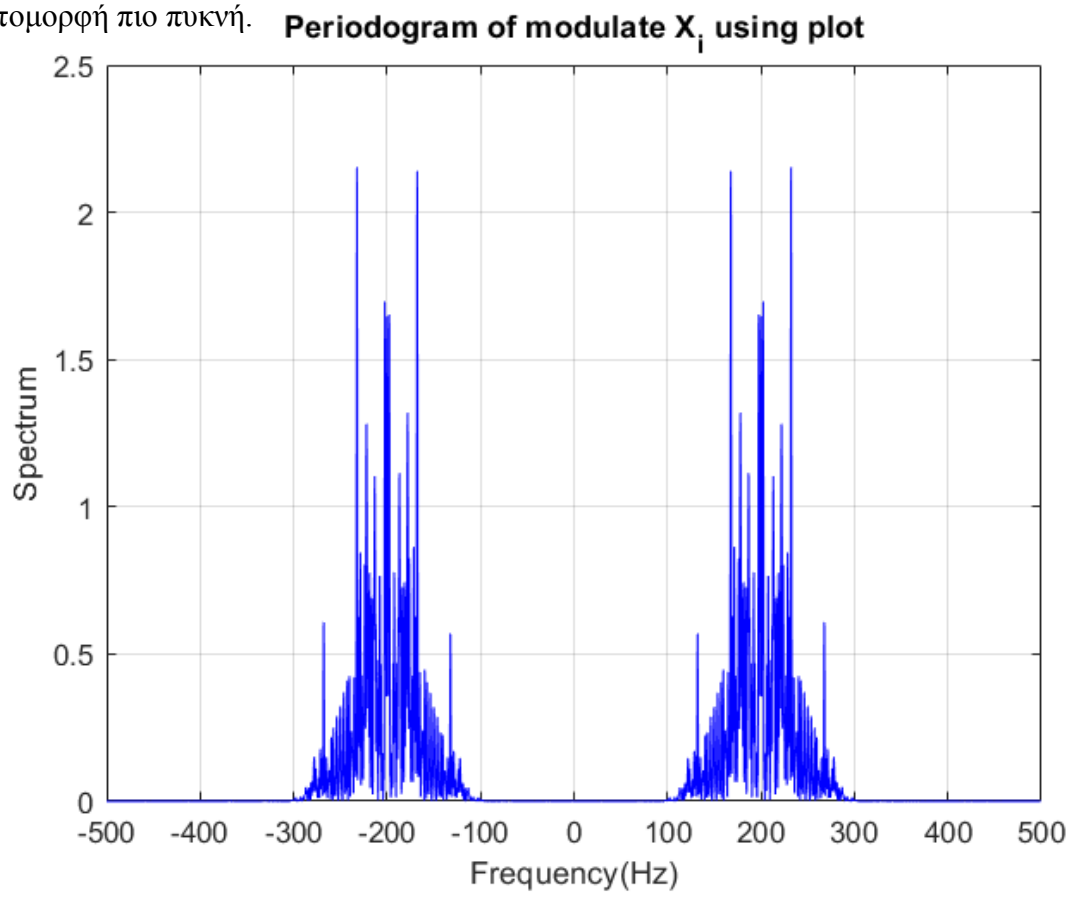
$$X_I, mod(t) = 2X_I(t)cos(2\pi F_0 t)$$

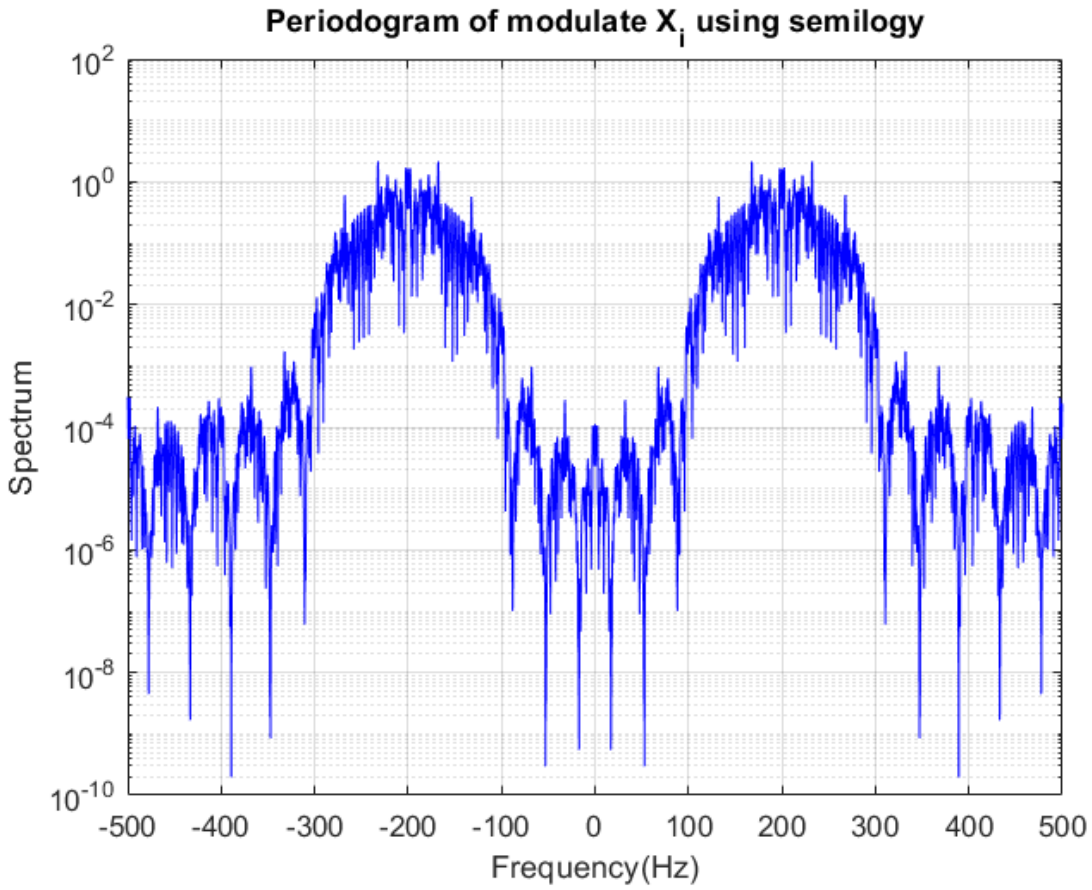
$$X_Q, mod(t) = -2X_Q(t)sin(2\pi F_0 t)$$

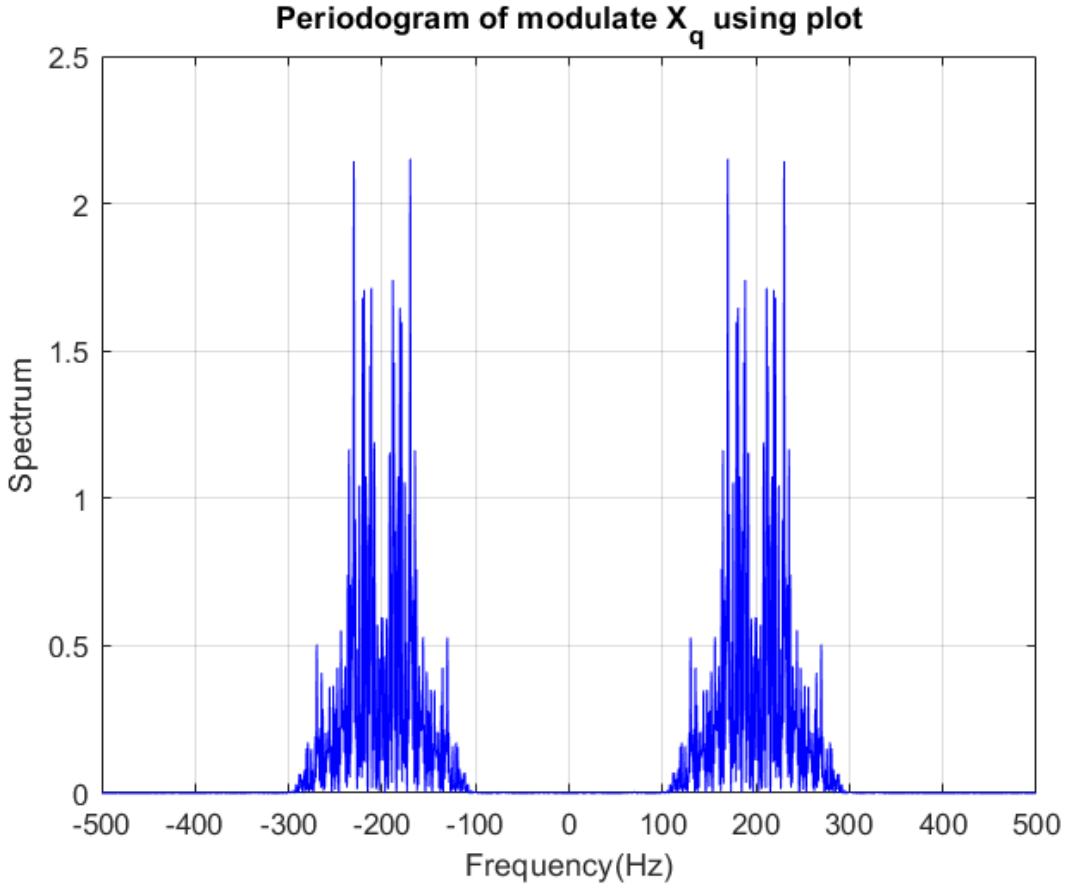


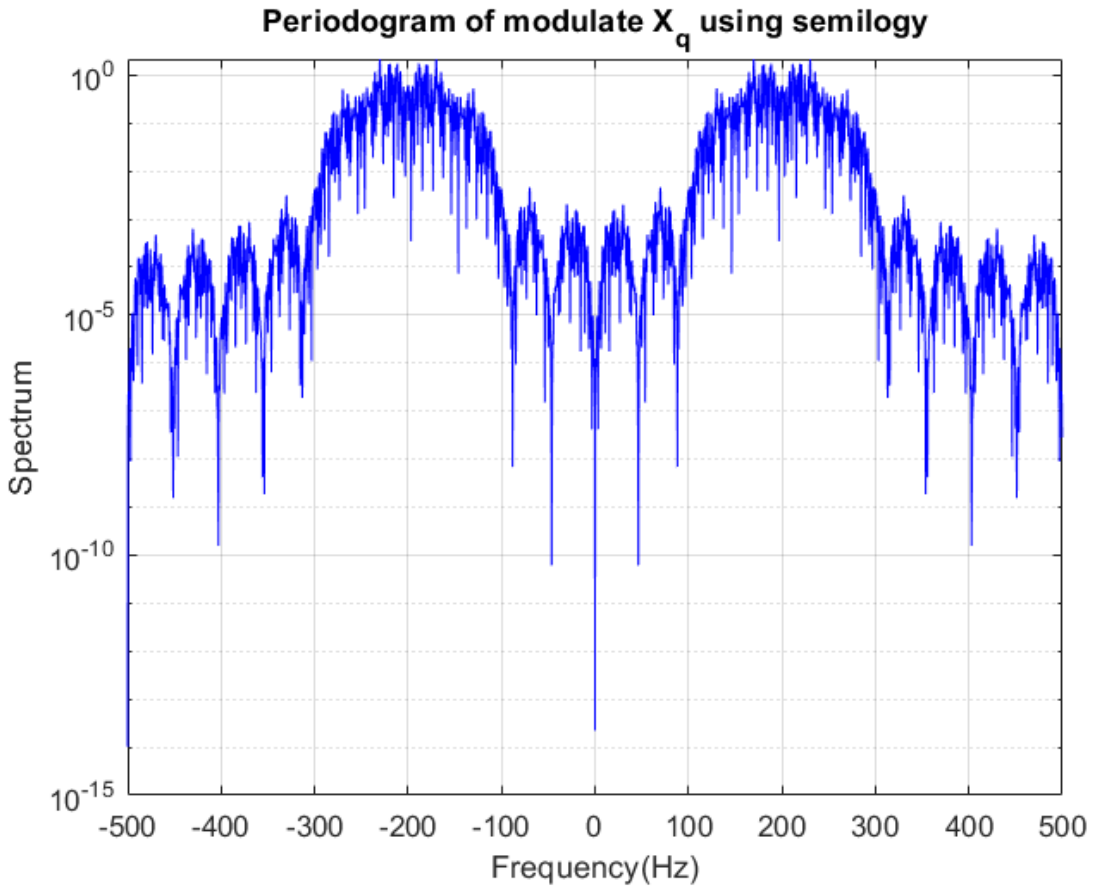


Παρατηρείται ότι δεκαπλασιάζεται το πλάτος, η συχνότητα γίνεται μεγαλύτερη και η κυματομορφή πιο πυκνή.




Periodogram of modulate X_i using semilogy
Spectrum
Frequency(Hz)
10^2
10^0
10^-2
10^-4
10^-6
10^-8
10^-10
-500 -400 -300 -200 -100 0 100 200 300 400 500


Periodogram of modulate X_q using plot
Spectrum
Frequency(Hz)
2.5
2
1.5
1
0.5
0
-500 -400 -300 -200 -100 0 100 200 300 400 500



Παρατηρείται ότι το φάσμα του αρχικού σήματος μετακινήθηκε γύρω από τη συχνότητα του φορέα F0 όπως ήταν αναμενόμενο μετά από διαμόρφωση. Επίσης, φαίνεται και η συμμετρία γύρω από το μηδέν.

Κώδικας Matlab:

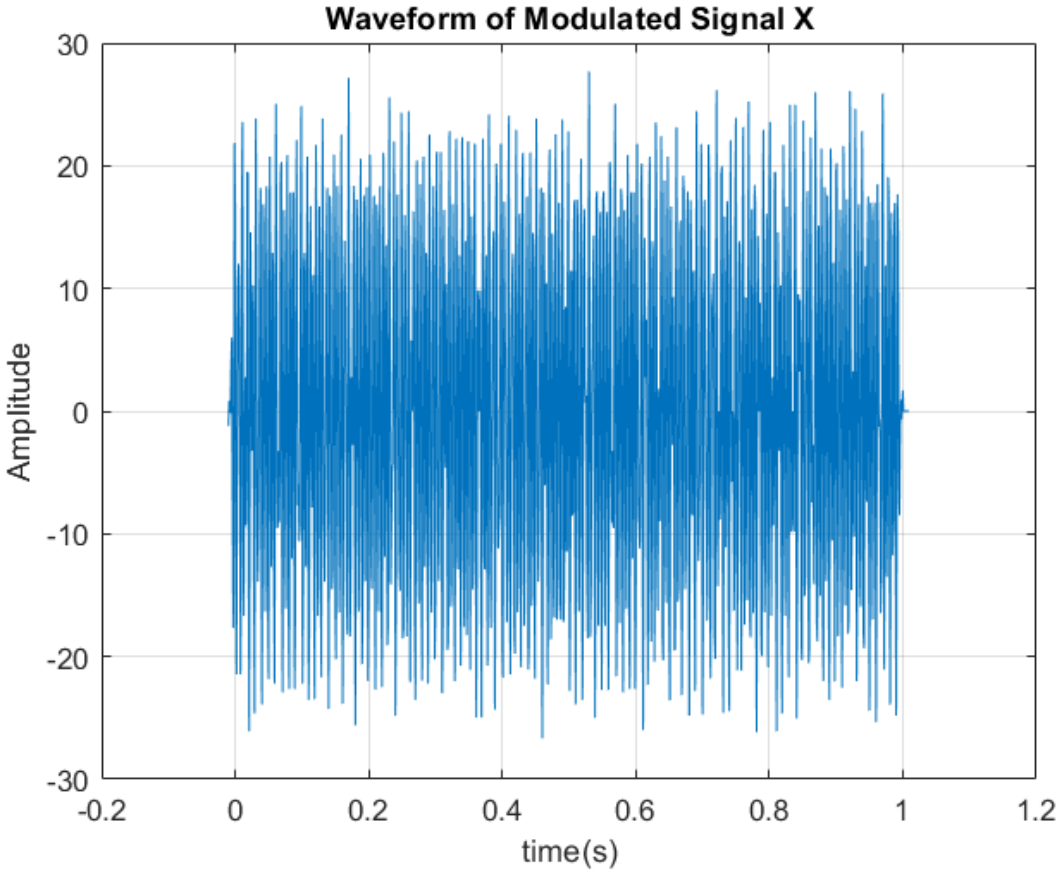
```
% 4.
fo = 200;
X_I_mod = 2 * X_I .* cos(2*pi*(fo.*tX_I));
X_Q_mod = (-2) * X_Q .* sin(2*pi*(fo.*tX_Q));
X_I_2 = X_I.*X_I_mod;
X_Q_2 = X_Q.*X_Q_mod;
figure();
%plot(tX_I,X_I_2);
plot(tX_I,X_I_mod);
title('Modulated X_i(t) Waveform')
xlabel('Time(s)');
ylabel('Amplitude');
grid on;
set(gcf,'color', 'w');
figure()
%plot(tX_Q,X_Q_2);
plot(tX_Q,X_Q_mod);
title('Modulated X_q(t) Waveform')
```

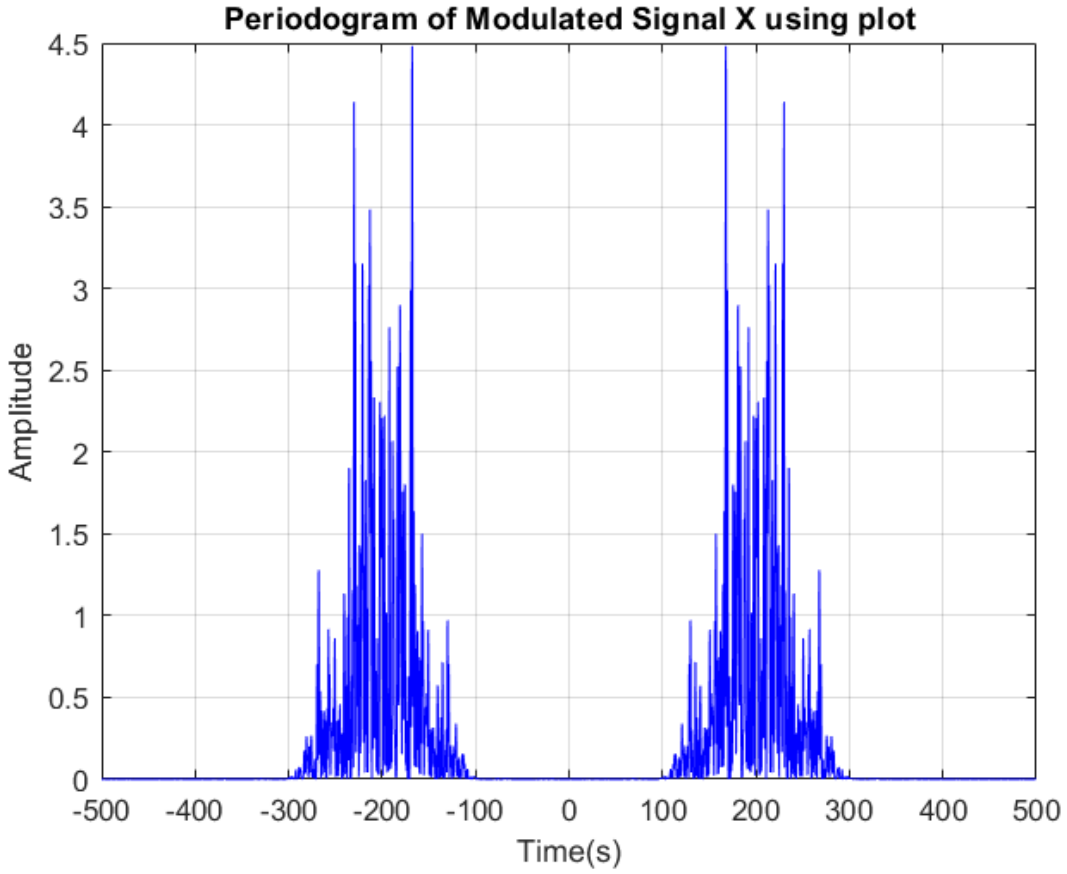
```
xlabel('time(s)');
ylabel('Amplitude');
grid on;
set(gcf,'color', 'w');
% Plot the periodograms
S_I_mod = fftshift(fft(X_I_mod,Nf)* Ts);
P_I_mod = ((abs(S_I_mod)) .^ 2) / Ttotal_I;
S_Q_mod = fftshift(fft(X_Q_mod,Nf)* Ts);
P_Q_mod = ((abs(S_Q_mod)) .^ 2) / Ttotal_Q;
figure();
plot(Fx, P_I_mod , 'blue');
title('Periodogram of modulate X_i using plot');
ylabel('Spectrum');
xlabel('Frequency(Hz)');
grid on;
set(gcf,'color', 'w');
figure();
semilogy(Fx, P_I_mod , 'blue');
title('Periodogram of modulate X_i using semilogy');
ylabel('Spectrum');
xlabel('Frequency(Hz)');
grid on;
set(gcf,'color', 'w');
figure();
plot(Fx, P_Q_mod , 'blue');
title('Periodogram of modulate X_q using plot');
ylabel('Spectrum');
xlabel('Frequency(Hz)');
grid on;
set(gcf,'color', 'w');
figure();
semilogy(Fx, P_Q_mod , 'blue');
title('Periodogram of modulate X_q using semilogy');
ylabel('Spectrum');
xlabel('Frequency(Hz)');
grid on;
set(gcf,'color', 'w');
```

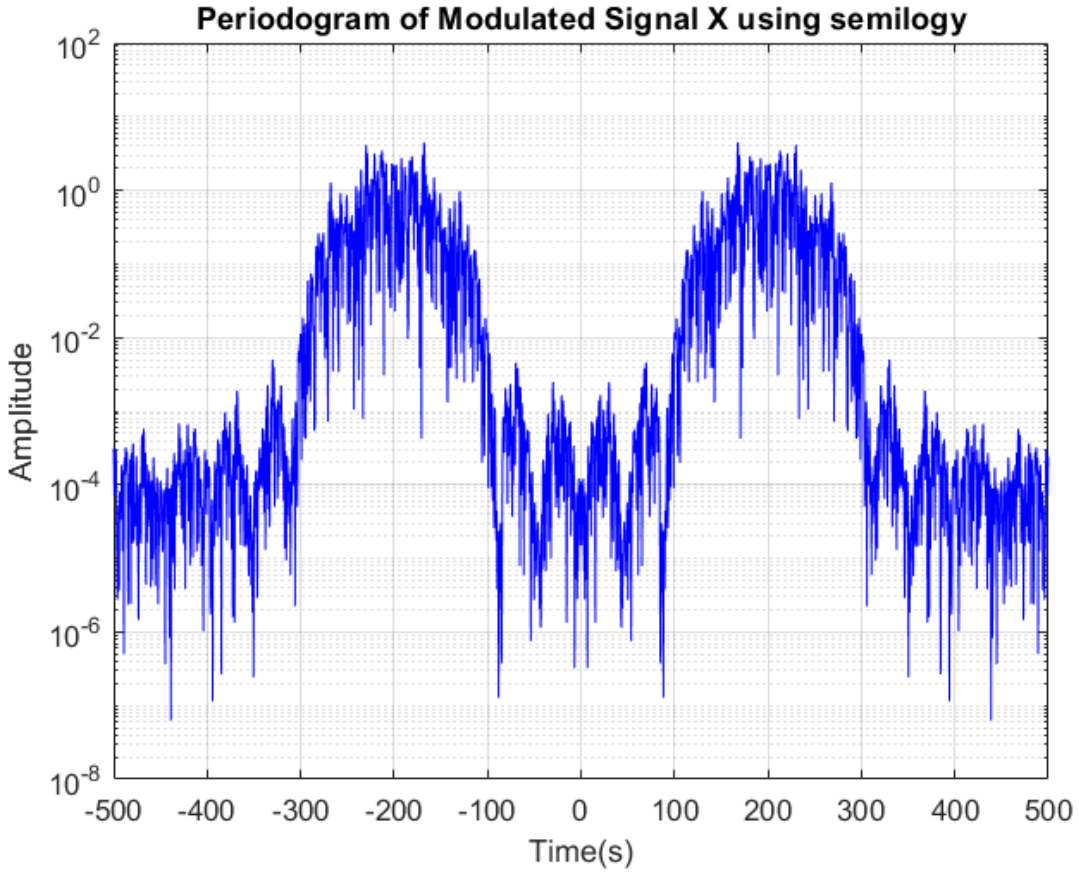
5.

Σχεδιάστηκε η είσοδος του καναλιού και το περιοδόγραμμά της.

$$X_{mod}(t) = X_{I,}\, mod(t) + X_{Q,} mod(t)$$


Waveform of Modulated Signal X
Amplitude
time(s)
30
20
10
0
-10
-20
-30
-0.2
0
0.2
0.4
0.6
0.8
1
1.2


Periodogram of Modulated Signal X using plot
Amplitude
Time(s)
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-500
-400
-300
-200
-100
0
100
200
300
400
500



Παρατηρείται πως το πλάτος και η συχνότητα μεγαλώνουν και το σήμα γίνεται πιο πυκνό.

Κώδικας Matlab:

```
% 5.
X_mod = X_I_mod + X_Q_mod;
figure();
plot(tX_I,X_mod);
title('Waveform of Modulated Signal X');
xlabel('time(s)');
ylabel('Amplitude');
grid on;
set(gcf,'color', 'w');
S_X_mod = fftshift(fft(X_mod,Nf)* Ts);
P_X_mod = ((abs(S_X_mod)) .^ 2) / Ttotal_I;
figure();
plot(Fx, P_X_mod , 'blue');
title('Periodogram of Modulated Signal X using plot');
xlabel('Time(s)');
ylabel('Amplitude');
grid on;
set(gcf,'color', 'w');
figure();
semilogy(Fx, P_X_mod , 'blue');
```

```
title('Periodogram of Modulated Signal X using semilogy');
xlabel('Time(s)');
ylabel('Amplitude');
grid on;
set(gcf,'color', 'w');
```

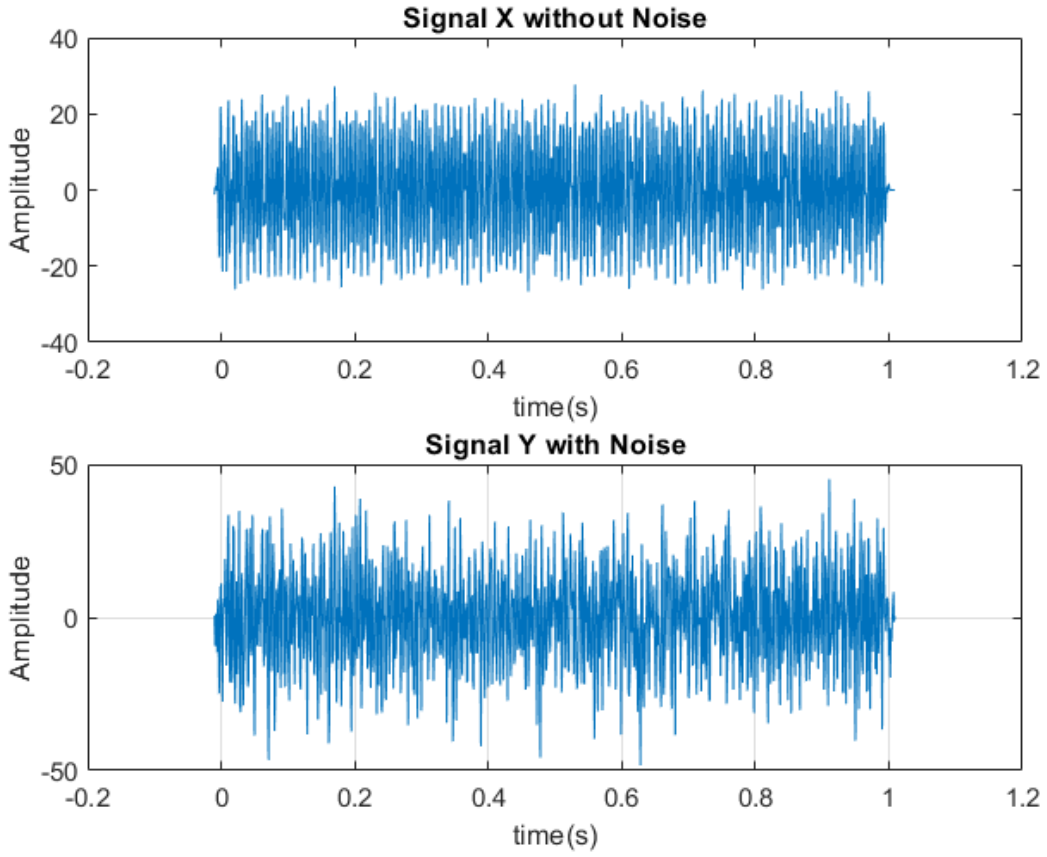
6.

Το κανάλι θεωρήθηκε ιδανικό.

7.

Στην έξοδο του καναλιού, προστέθηκε λευκός Gaussian θόρυβος W(t) με διασπορά ίση με:

$$\sigma_W^2 = \frac{1}{T_s \cdot 10^{\frac{\mathrm{SNR_{dB}}}{10}}},$$



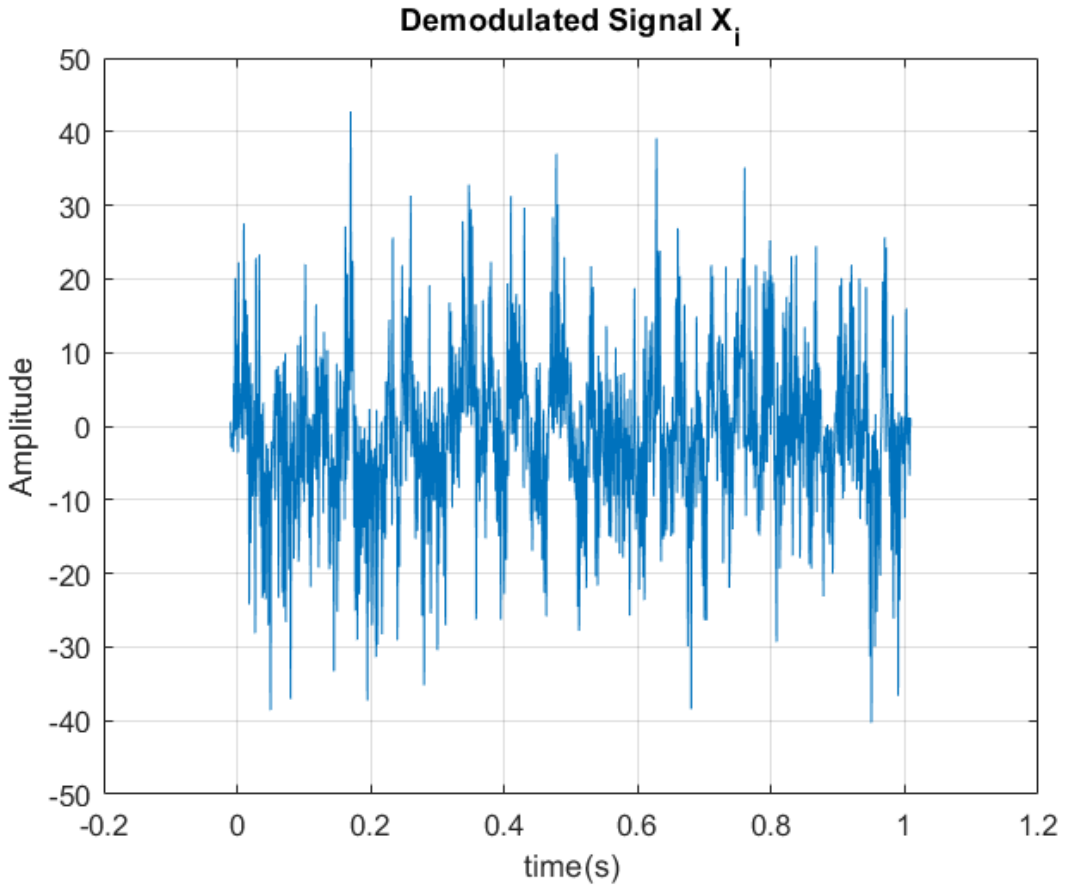
Κώδικας Matlab:

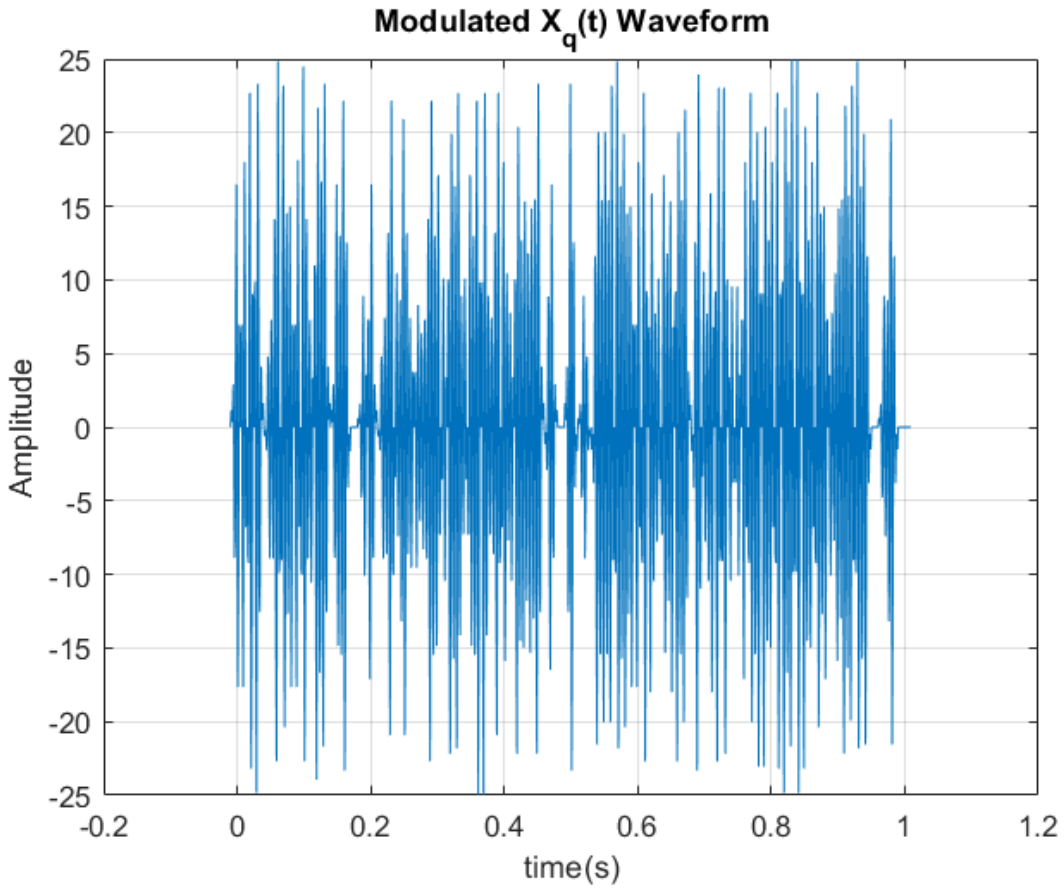
```
% 7.
% Gaussian noise
SNR_db = 20;
var_W = (10*(A^2))/ (Ts* 10^(SNR_db/10));
```

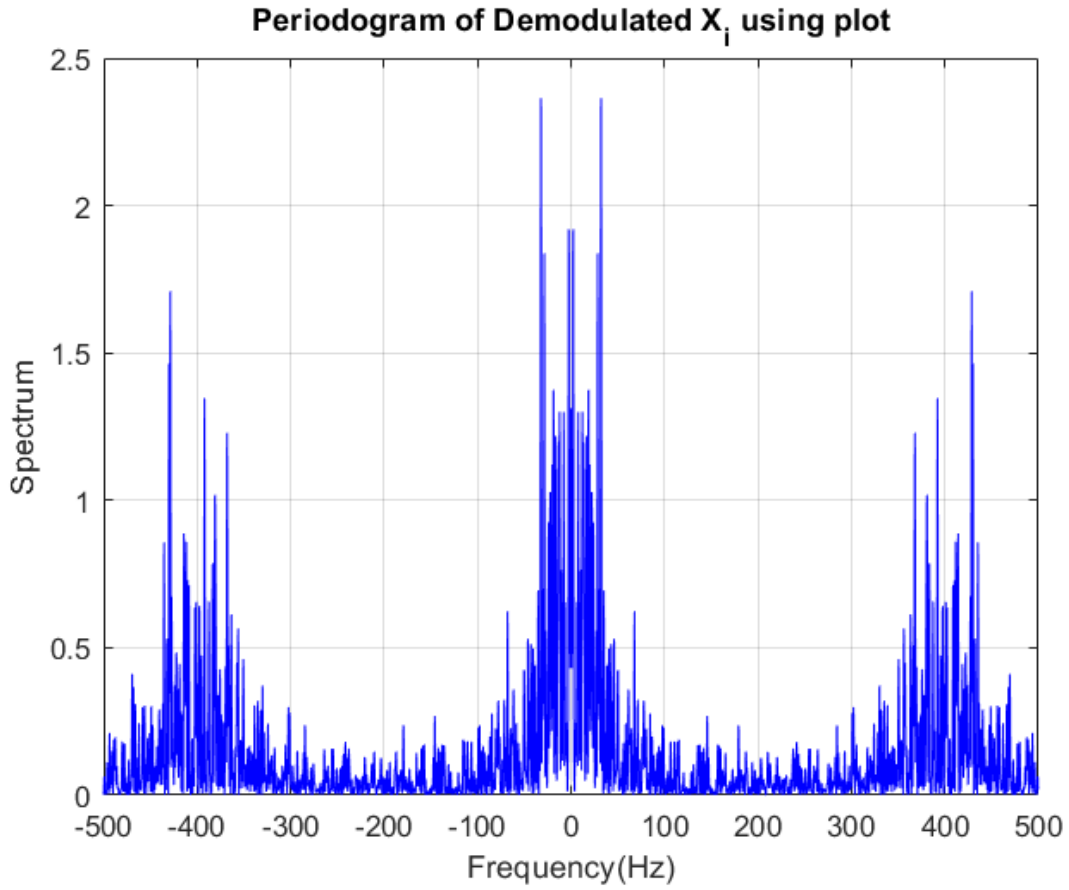
```
Wt = sqrt(var_W).*randn(1,length(X_mod));
% Adding noise
Y = X_mod + Wt;
figure();
subplot(2,1,1)
plot(tX_I,X_mod);
title('Signal X without Noise');
xlabel('time(s)');
ylabel('Amplitude');
subplot(2,1,2)
plot(tX_I,Y);
title('Signal Y with Noise');
xlabel('time(s)');
ylabel('Amplitude');
grid on;
set(gcf,'color', 'w');
```

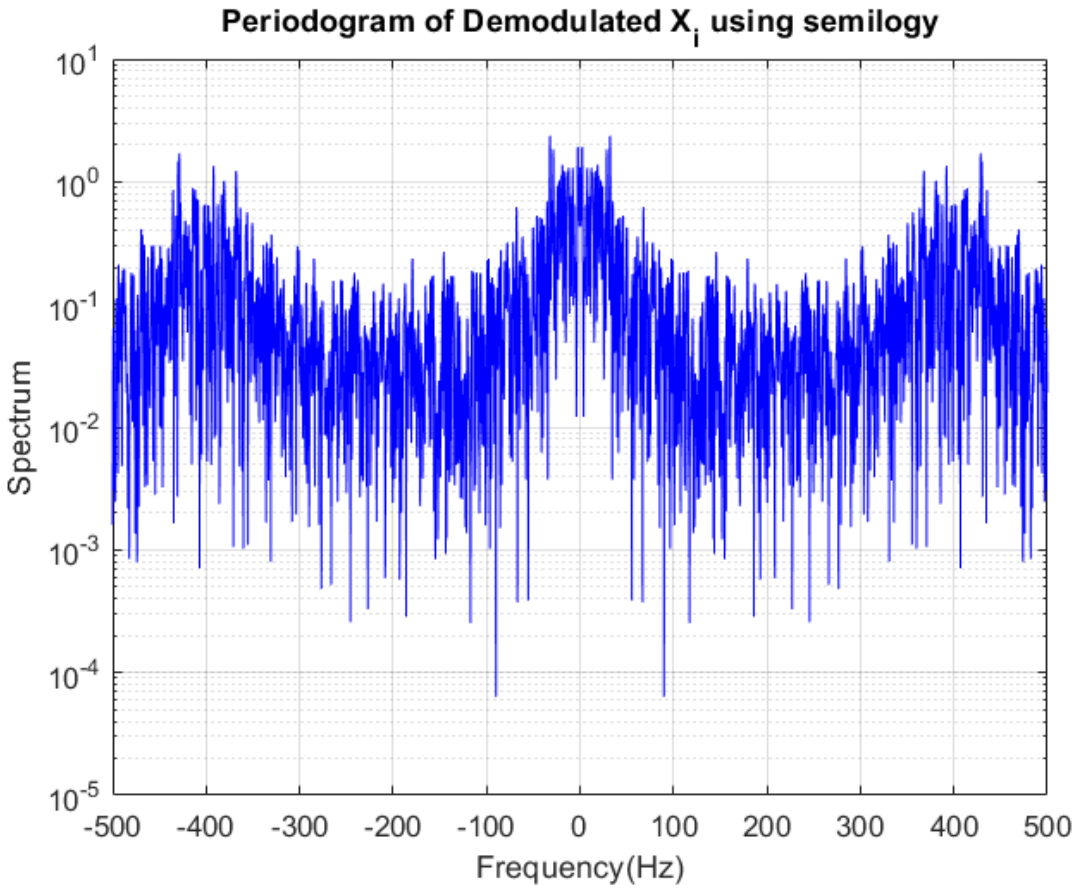
8.

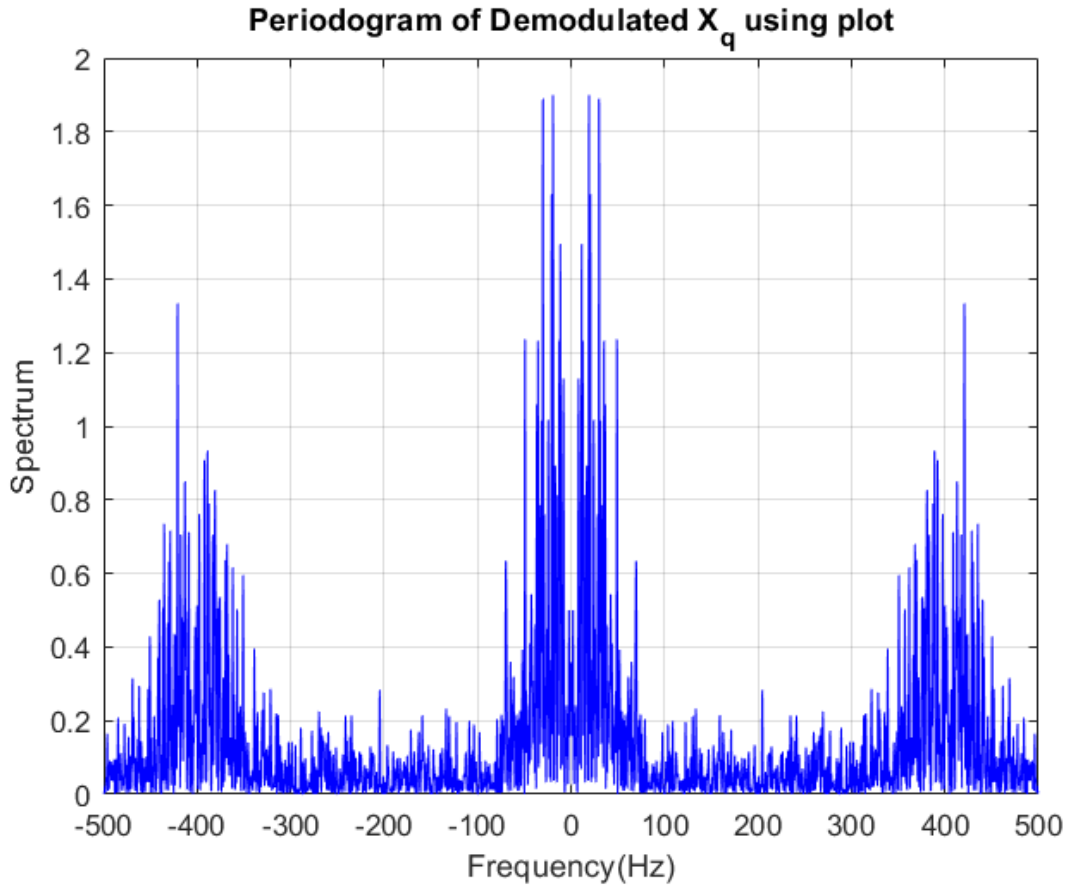
Στον δέκτη, διακλαδώθηκε η ενθόρυβη κυματομορφή και πολλαπλασιάστηκε με φορείς $cos(2\pi F_0 t)$ και $-sin(2\pi F_0 t)$, αντίστοιχα. Οι κυματομορφές και τα περιοδογράμματά που προκύπτουν είναι:

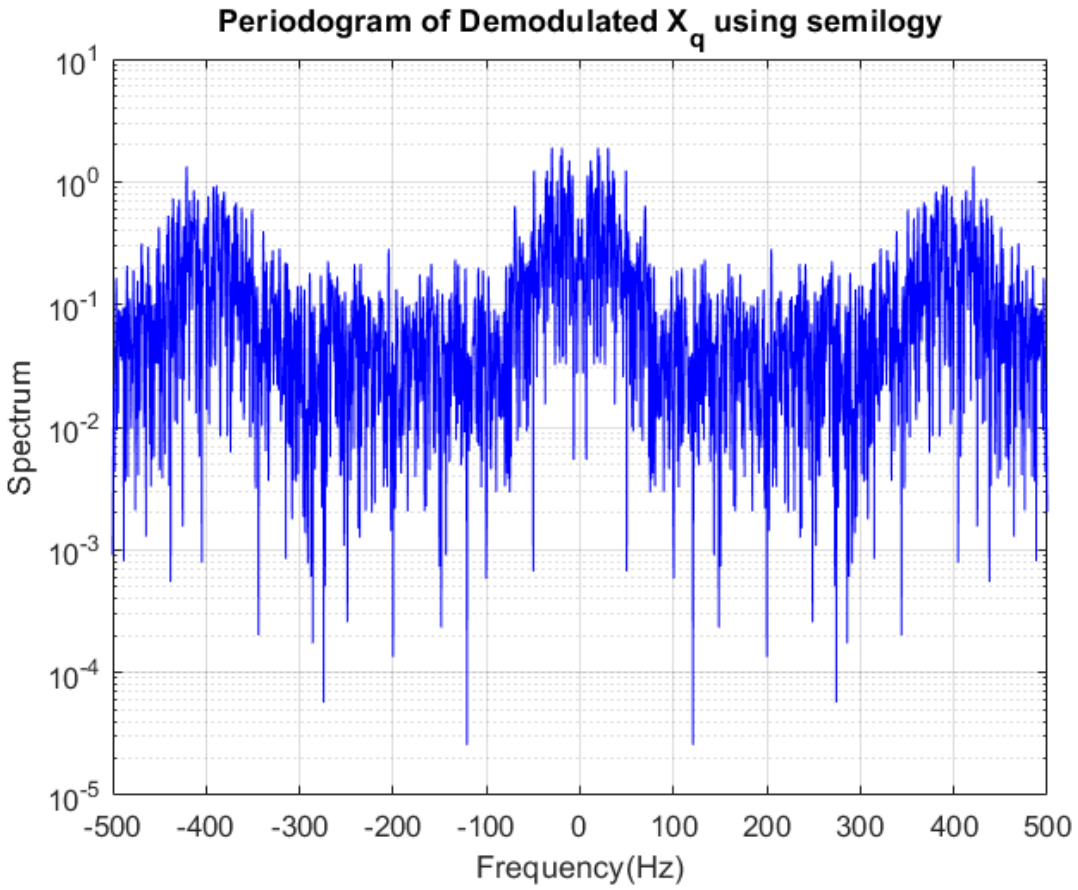



Modulated X_q(t) Waveform
Amplitude
time(s)
25
20
15
10
5
0
-5
-10
-15
-20
-25
-0.2
0
0.2
0.4
0.6
0.8
1
1.2


Periodogram of Demodulated X_i using plot
Spectrum
Frequency(Hz)
2.5
2
1.5
1
0.5
0
-500
-400
-300
-200
-100
0
100
200
300
400
500


Periodogram of Demodulated X_i using semilogy
Spectrum
Frequency(Hz)
10^1
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
-500 -400 -300 -200 -100 0 100 200 300 400 500


Periodogram of Demodulated X_q using plot
Spectrum
Frequency(Hz)
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-500 -400 -300 -200 -100 0 100 200 300 400 500



Παρατηρείται ότι όταν πολλαπλασιαστεί το διαμορφωμένο σήμα πάλι με τους φορείς τότε ανακτάται το αρχικό σήμα στις συχνότητες γύρω από το μηδέν όμως υπάρχουν και δύο ακόμα σήματα σε συχνότητες $2 \cdot F0$.

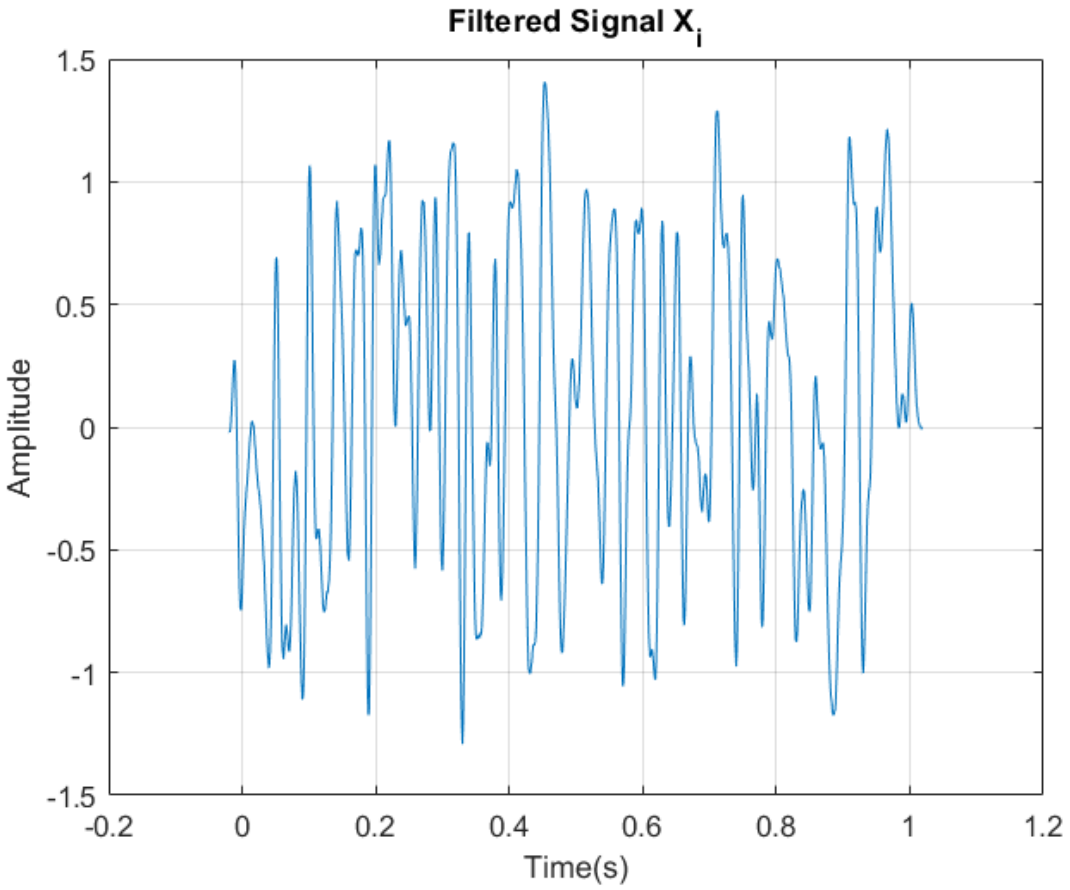
Κώδικας Matlab:

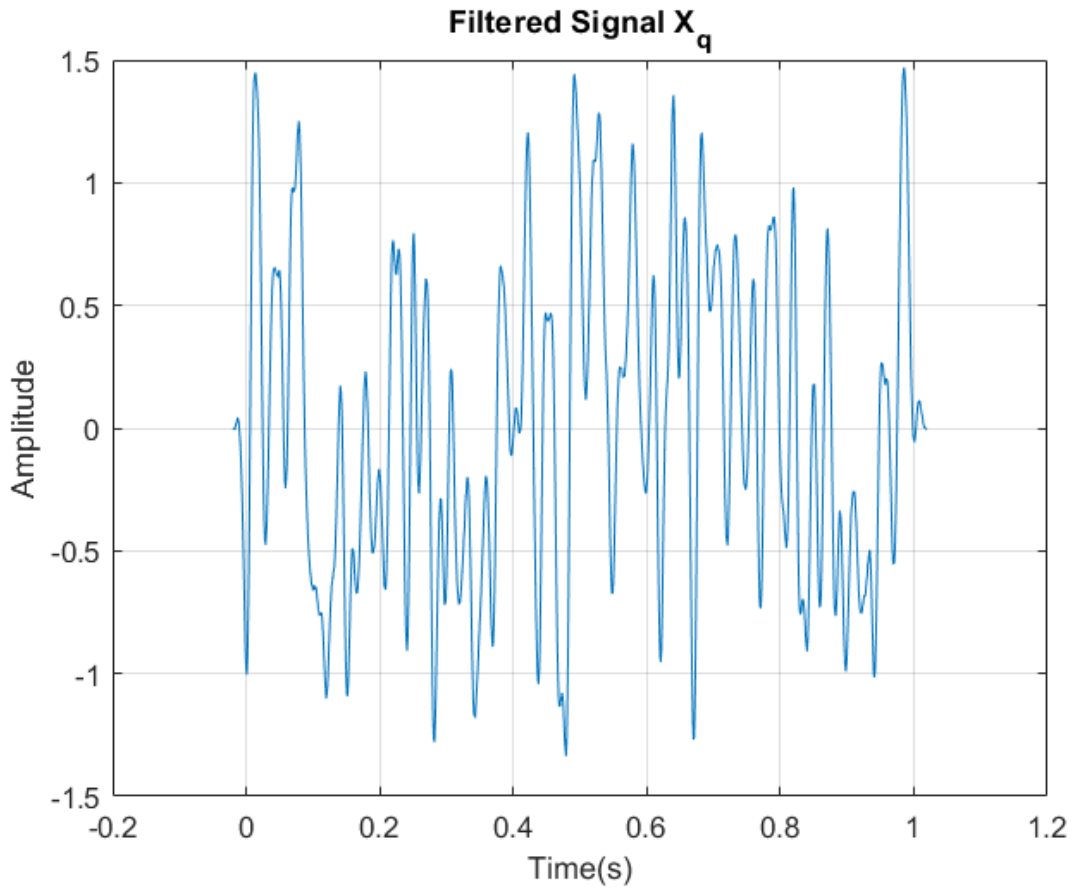
```
% 8.
X_Wi_mod = Y .* cos(2*pi*(fo.*tX_I));
X_Wq_mod = (-1) * Y .* sin(2*pi*(fo.*tX_Q));
figure();
plot(tX_Q,X_Wi_mod);
title('Demodulated Signal X_i');
xlabel('time(s)');
ylabel('Amplitude');
grid on;
set(gcf,'color', 'w');
figure()
plot(tX_Q,X_Wq_mod);
title('Demodulated Signal X_q');
xlabel('time(s)');
ylabel('Amplitude');
set(gcf,'color', 'w');
% Plot the periodograms
```

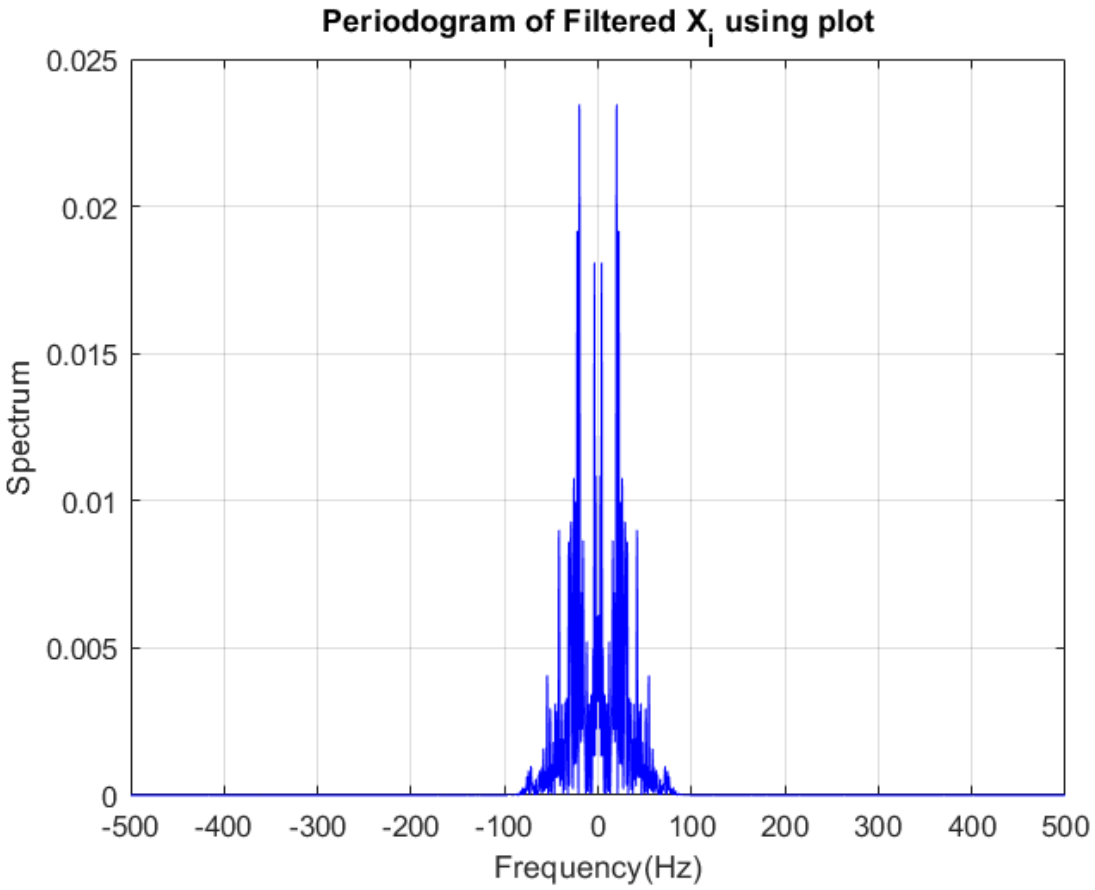
```
Ttotal_Wi = tX_I(end) - tX_I(1);
S_Wi = fftshift(fft(X_Wi_mod,Nf)* Ts);
P_Wi = ((abs(S_Wi)) .^ 2) / Ttotal_Wi;
Ttotal_Wq = tX_Q(end) - tX_Q(1);
S_Wq = fftshift(fft(X_Wq_mod,Nf)* Ts);
P_Wq = ((abs(S_Wq)) .^ 2) / Ttotal_Wq;
figure();
plot(Fx, P_Wi , 'blue');
title('Periodogram of Demodulated X_i using plot');
ylabel('Spectrum');
xlabel('Frequency(Hz)');
grid on;
set(gcf,'color', 'w');
figure();
semilogy(Fx, P_Wi , 'blue');
title('Periodogram of Demodulated X_i using semilogy');
ylabel('Spectrum');
xlabel('Frequency(Hz)');
grid on;
set(gcf,'color', 'w');
figure();
plot(Fx, P_Wq , 'blue');
title('Periodogram of Demodulated X_q using plot');
ylabel('Spectrum');
xlabel('Frequency(Hz)');
grid on;
set(gcf,'color', 'w');
figure();
semilogy(Fx, P_Wq , 'blue');
title('Periodogram of Demodulated X_q using semilogy');
ylabel('Spectrum');
xlabel('Frequency(Hz)');
grid on;
set(gcf,'color', 'w');
```

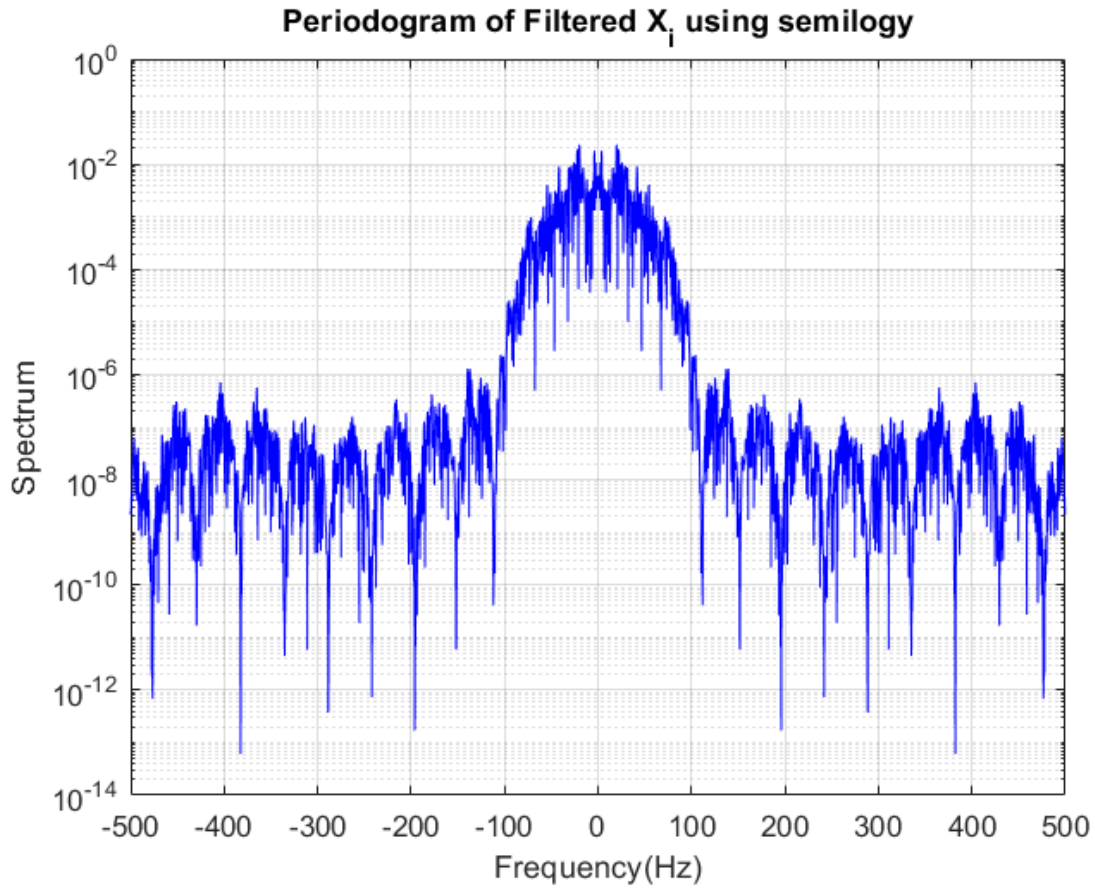
9.

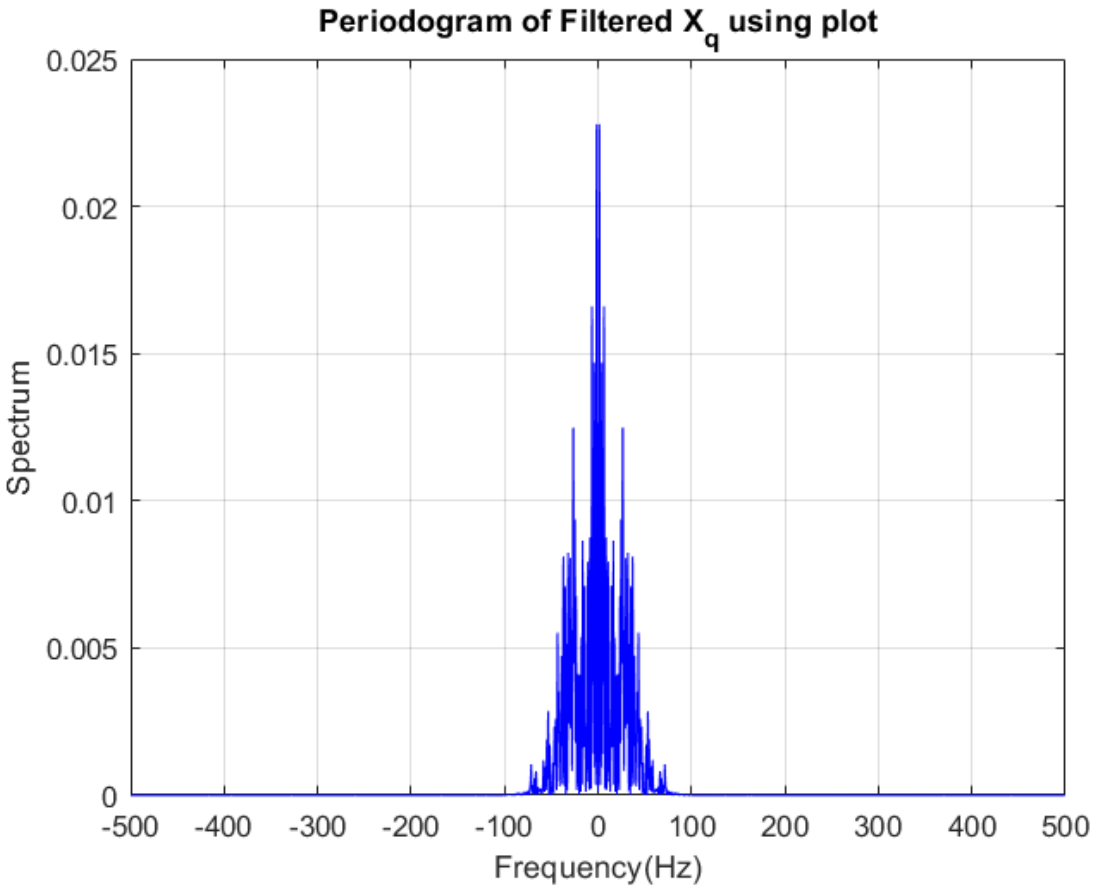
Οι κυματομορφές αυτές λοιπόν, περάστηκαν στα προσαρμοσμένα φίλτρα και σχεδιάστηκαν οι κυματομορφές που προκύπτουν και τα περιοδογράμματά τους.

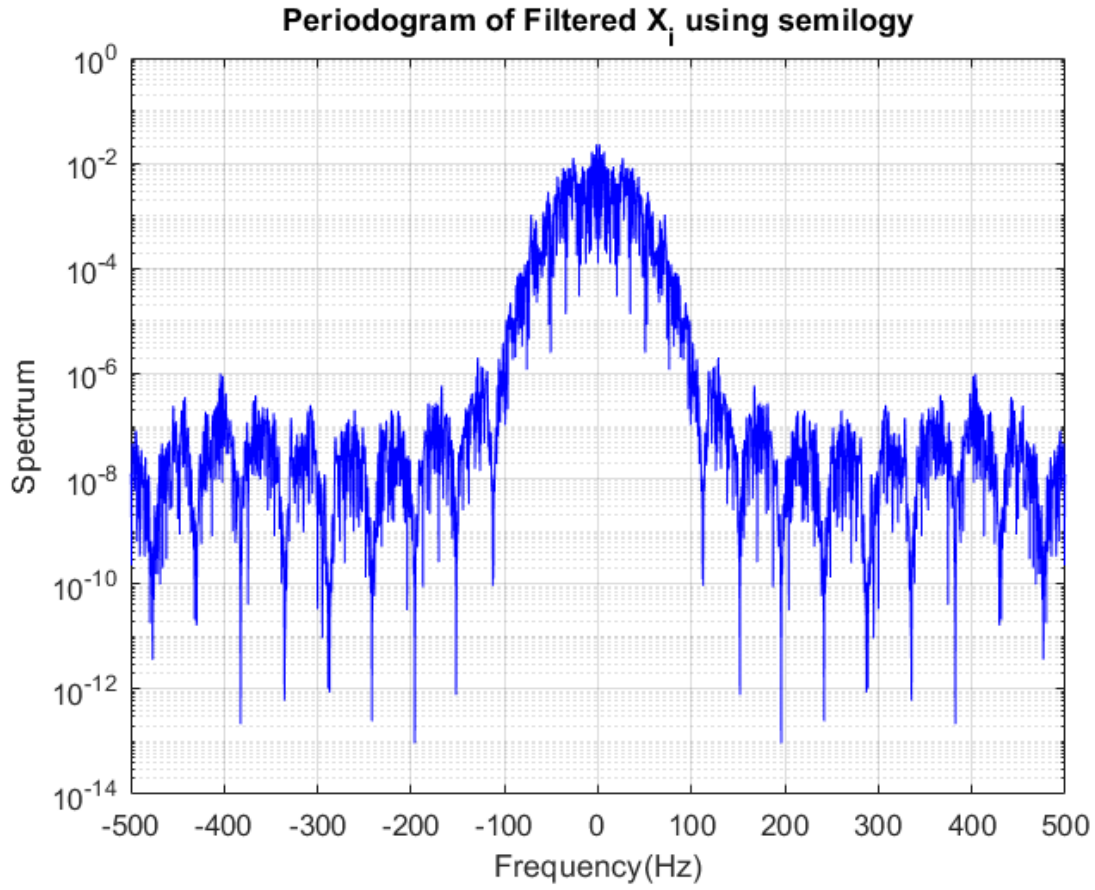

Filtered Signal X_i
Amplitude
Time(s)
1.5
1
0.5
0
-0.5
-1
-1.5
-0.2
0
0.2
0.4
0.6
0.8
1
1.2


Filtered Signal X_q
Amplitude
Time(s)
1.5
1
0.5
0
-0.5
-1
-1.5
-0.2
0
0.2
0.4
0.6
0.8
1
1.2


Periodogram of Filtered X_i using plot
Spectrum
Frequency(Hz)
0.025
0.02
0.015
0.01
0.005
0
-500
-400
-300
-200
-100
0
100
200
300
400
500


Periodogram of Filtered X_i using semilogy
Spectrum
Frequency(Hz)
10^0
10^-2
10^-4
10^-6
10^-8
10^-10
10^-12
10^-14
-500
-400
-300
-200
-100
0
100
200
300
400
500


Periodogram of Filtered X_q using plot
Spectrum
0.025
0.02
0.015
0.01
0.005
0
-500 -400 -300 -200 -100 0 100 200 300 400 500
Frequency(Hz)


Periodogram of Filtered X_i using semilogy
Spectrum
10^0
10^-2
10^-4
10^-6
10^-8
10^-10
10^-12
10^-14
-500 -400 -300 -200 -100 0 100 200 300 400 500
Frequency(Hz)

Παρατηρείται ότι τα δύο φίλτρα λειτουργούν ως χαμηλοπερατά (low pass) φίλτρα και αποκόπτουν τις διπλάσιες συχνότητες $2 \cdot F0$ που δεν χρειάζονται. Έτσι επιτεύχθηκε η ανάκτηση του πρώτου σήματος που βρίσκεται σε συχνότητες γύρω από το μηδέν

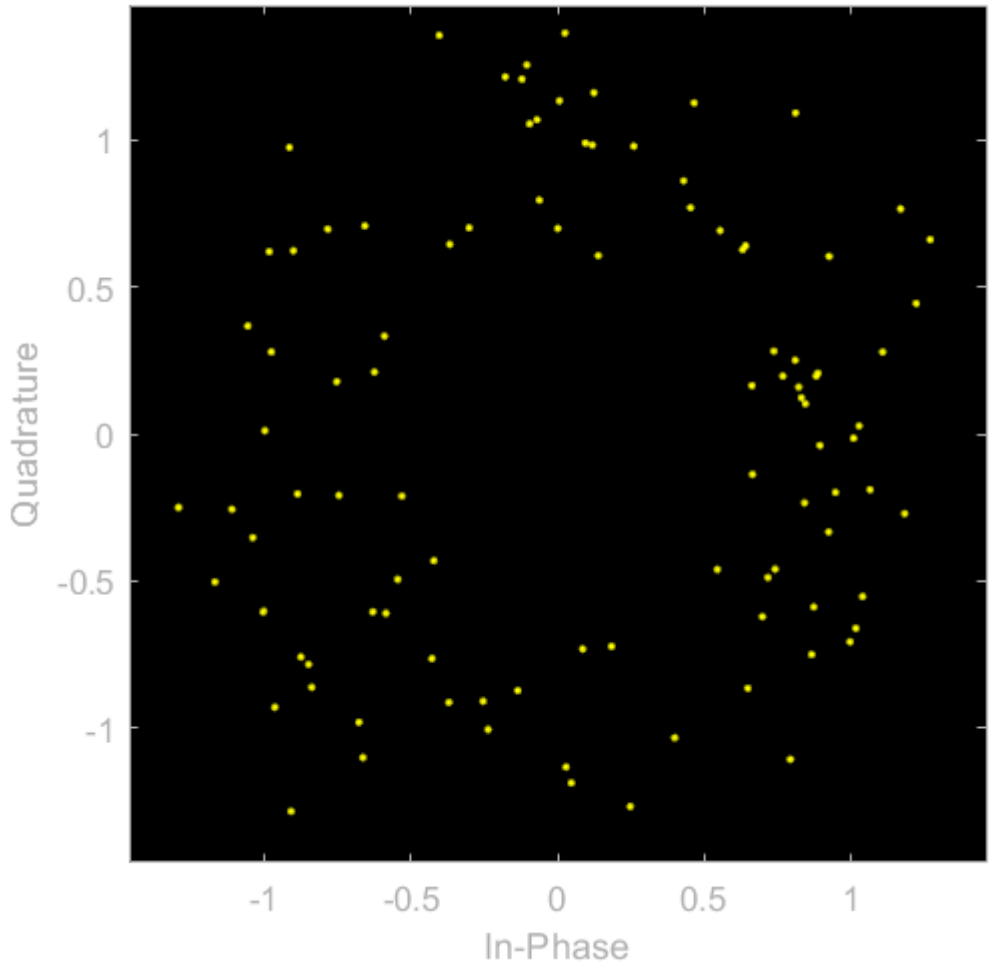
Κώδικας Matlab:

```
X_Wi_conv = conv(X_Wi_mod,phi)*Ts;
tX_I1 = tX_I(1)+t1(1):Ts:tX_I(end)+t1(end);
X_Wq_conv = conv(X_Wq_mod,phi)*Ts;
tX_Q1 = tX_Q(1)+t1(1):Ts:tX_Q(end)+t1(end);
figure();
plot(tX_I1,X_Wi_conv);
title('Filtered Signal X_i');
xlabel('Time(s)');
ylabel('Amplitude');
grid on;
set(gcf,'color', 'w');
figure();
plot(tX_Q1,X_Wq_conv);
title('Filtered Signal X_q');
xlabel('Time(s)');
ylabel('Amplitude');
grid on;
set(gcf,'color', 'w');
Ttotal_Wi_conv = tX_I1(end) - tX_I1(1);
S_Wi_conv = fftshift(fft(X_Wi_conv,Nf)* Ts);
P_Wi_conv = ((abs(S_Wi_conv)) .^ 2) / Ttotal_Wi_conv;
Ttotal_Wq_conv = tX_Q1(end) - tX_Q1(1);
S_Wq_conv = fftshift(fft(X_Wq_conv,Nf)* Ts);
P_Wq_conv = ((abs(S_Wq_conv)) .^ 2) / Ttotal_Wq_conv;
figure();
plot(Fx, P_Wi_conv , 'blue');
title('Periodogram of Filtered X_i using plot');
ylabel('Spectrum');
xlabel('Frequency(Hz)');
grid on;
set(gcf,'color', 'w');
figure();
semilogy(Fx, P_Wi_conv , 'blue');
title('Periodogram of Filtered X_i using semilogy');
ylabel('Spectrum');
xlabel('Frequency(Hz)');
grid on;
set(gcf,'color', 'w');
figure();
plot(Fx, P_Wq_conv , 'blue');
title('Periodogram of Filtered X_q using plot');
ylabel('Spectrum');
xlabel('Frequency(Hz)');
grid on;
set(gcf,'color', 'w');
figure();
```

```
semilogy(Fx, P_Wq_conv , 'blue');
title('Periodogram of Filtered X_i using semilogy');
ylabel('Spectrum');
xlabel('Frequency(Hz)');
grid on;
set(gcf,'color', 'w');
```

10.

Δειγματοληπτήθηκε η έξοδος των προσαρμοσμένων φίλτρων τις κατάλληλες χρονικές στιγμές και σχεδιάστηκε η ακολουθία εξόδου χρησιμοποιώντας την εντολή scatterplot



Κώδικας Matlab:

```
% 10.
YI = X_Wi_conv(2*over+1 :over: (2+N)*over);
YQ = X_Wq_conv(2*over+1 :over: (2+N)*over);
Y_sampled = YI+1i*YQ;
scatterplot(Y_sampled)
set(gcf,'color', 'w');
```

11.

Συντάχθηκε η συνάρτηση detect_PSK_16(Y) η οποία χρησιμοποιεί τον κανόνα εγγύτερου γείτονα και αποφασίζει για την ακολουθία εισόδου 16-PSK σύμβολο-προς-σύμβολο και την αντίστροφη απεικόνιση Gray και υπολογίζει την εκτιμώμενη δυαδική ακολουθία εισόδου.

Κώδικας Matlab:

```
% 11.
[Y_est, est_bit_seq] = detect_PSK_16(Y_sampled);

function [est_X, est_bit_seq] = detect_PSK_16(Y)
% Define the constellation points of 16-PSK
mi = cos((0:15) * 2 * pi / 16);
mq = sin((0:15) * 2 * pi / 16);
% Initialize variables
L = length(Y);
est_X = zeros(1, L);
est_bit_seq = zeros(4 * L, 1);
for i = 1:length(Y)
Yi = real(Y(i));
Yq = imag(Y(i));
% Compute the distances to each constellation point
distances = sqrt((Yi - mi).^2 + (Yq - mq).^2);
% Find the index of the closest constellation point
[~, index] = min(distances);
% Estimate the symbol and corresponding bits
est_X(i) = mi(index) + 1i * mq(index);
est_bit_seq(4*i-3:4*i) = de2bi(index-1, 4, 'left-msb')';
end
end
```

12.

Έπειτα, υλοποιήθηκε η συνάρτηση symbol_errors(est X, X) που υπολογίζει τα λάθη που έχουν συμβεί στα σύμβολα.

Κώδικας Matlab:

```
% 12.
num_of_symbol_errors = symbol_errors(Y_est, X);

function num_of_symbol_errors = symbol_errors(est_X, X)
num_of_symbol_errors = 0;
for i = 1:length(X)
if (est_X(i) ~= X(i))
num_of_symbol_errors = num_of_symbol_errors + 1;
end
end
end
```

13.

Μετά η bit_errors(est_bit_seq, b) που υπολογίζει το πλήθος των σφαλμάτων εκτίμησης bit.

Κώδικας Matlab:

```
% 13.
num_of_bit_errors = bit_errors(est_bit_seq, bit_seq);

function num_of_bit_errors = bit_errors(est_bit_seq, b)
num_of_bit_errors = sum(est_bit_seq ~= b);
end
```

2ο Μέρος

1.

Αρχικά, περάστηκε όλο το πρώτο μέρος χωρίς τα figures σε μια συνάρτηση PSK_16(SNR_db), η οποία εφαρμόζει όλα τα βήματα του πρώτου μέρους με όρισμα SNRdB και επιστρέφει το symbol και το bit error για το συγκεκριμένο SNRdB.

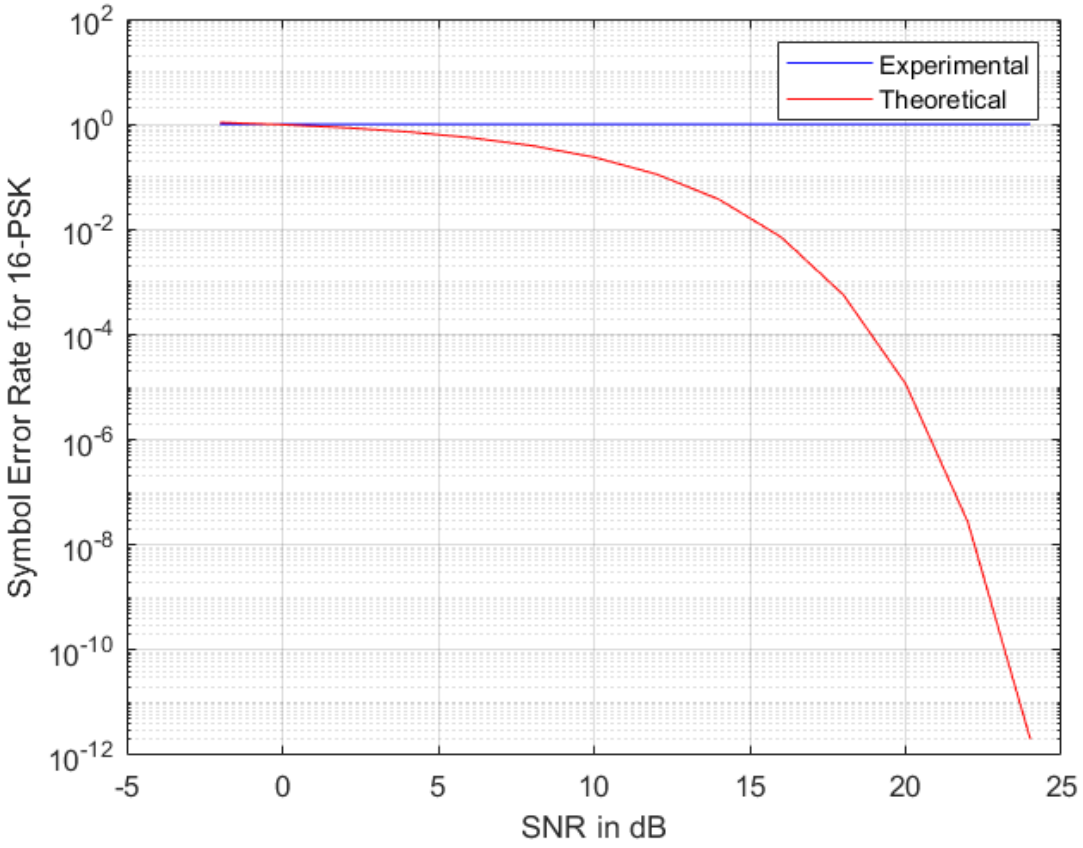
Με την μέθοδο Monte Carlo για κάθε SNRdB = [-2 : 2 : 24] εφαρμόστηκαν 1000 φορές τα βήματα του πρώτου μέρους και βρέθηκαν τα συνολικά symbol και bit errors για κάθε τιμή του SNRdB.

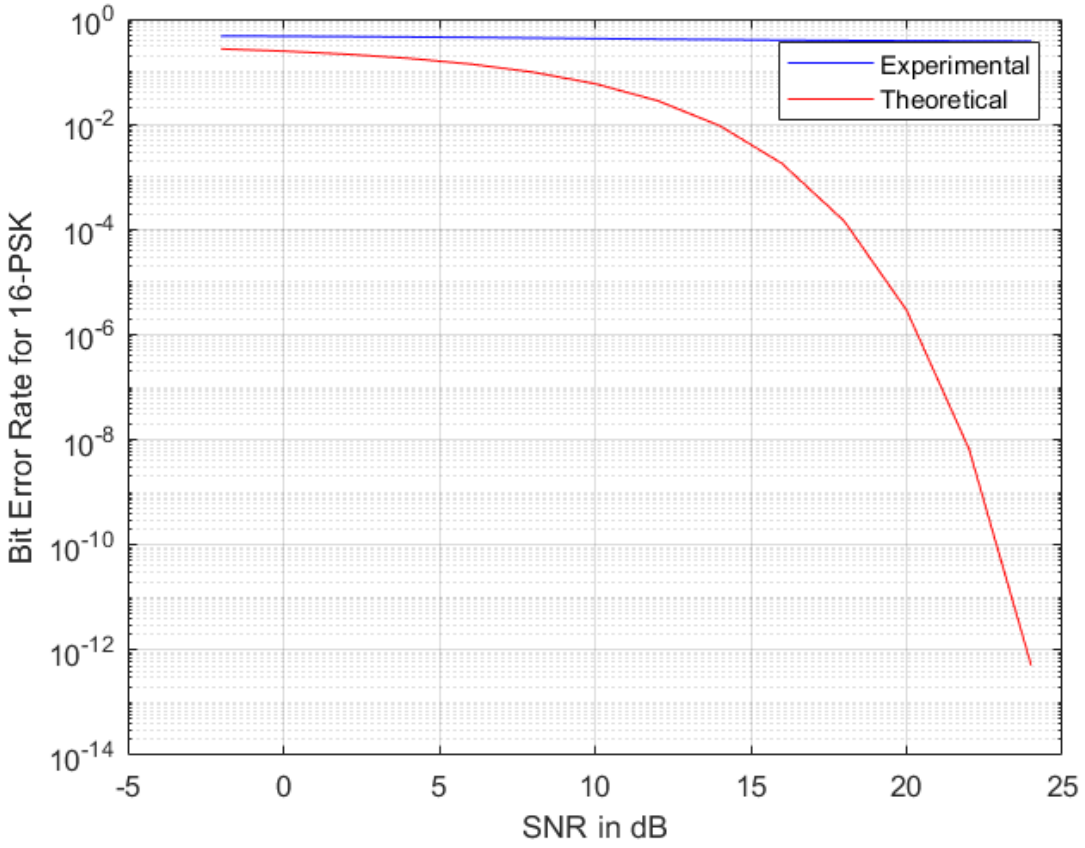
Έπειτα, υπολογίστηκε η πειραματική εκτίμηση της πιθανότητας σφάλματος συμβόλου για κάθε SNRdB διαιρώντας το συνολικό πλήθος σφαλμάτων απόφασης συμβόλου με το συνολικό πλήθος απεσταλμένων συμβόλων και η πειραματική εκτίμηση της πιθανότητας σφάλματος bit διαιρώντας το συνολικό πλήθος σφαλμάτων απόφασης bit με το συνολικό πλήθος απεσταλμένων bits.

$$\hat{P}(\mathrm{E}_{\mathrm{symbol}}) = \frac{\text{συνολικό πλήθος σφαλμάτων απόφασης συμβόλου}}{\text{συνολικό πλήθος απεσταλμένων συμβόλων}},$$

$$\hat{P}(\mathrm{E}_{\mathrm{bit}}) = \frac{\text{συνολικό πλήθος σφαλμάτων απόφασης bit}}{\text{συνολικό πλήθος απεσταλμένων bits}}.$$

Επίσης, δημιουργήθηκαν οι θεωρητικές πιθανότητες σφάλματος συμβόλου και bit με βάση τους τύπους της θεωρίας και της συνάρτησης Q.m που δόθηκε.


Experimental
Theoretical
Symbol Error Rate for 16-PSK
SNR in dB
10^2
10^0
10^-2
10^-4
10^-6
10^-8
10^-10
10^-12
-5
0
5
10
15
20
25



Έχει γίνει κάποιο λάθος στον κώδικα, καθώς και τα πειραματικά θα πρέπει να έχουν ίδια κλίση με τα θεωρητικά.

Κώδικας Matlab:

```
% 1.
K = 1000;
SNR_db_B = -2:2:24;
symbolErrB = 0;
bitsErrB = 0;
symbolErrB_Total = 0;
bitsErrB_Total = 0;
total_symbols = N*K;
total_bits = 4*N*K;
P_error_symbol = ones(1,length(SNR_db_B));
P_error_bit = ones(1,length(SNR_db_B));
for i = 1:length(SNR_db_B)
for j = 1:K
[symbolErrB, bitsErrB] = PSK_16(SNR_db_B(i));
symbolErrB_Total = symbolErrB_Total + symbolErrB;
bitsErrB_Total = bitsErrB_Total + bitsErrB;
end
P_error_symbol(i) = symbolErrB_Total / total_symbols;
```

```
P_error_bit(i) = bitsErrB_Total / total_bits;
symbolErrB_Total = 0;
bitsErrB_Total = 0;
end
% Theoritical calculation
sw = 10.*(A^2)./(Ts .* 10.^(SNR_db_B ./ 10));
sn =(Ts*sw)/2;
Theor_symbolErr = 3*Q(A./ sqrt(sn));
Theor_bitErr = Theor_symbolErr/log2(16);
% 2.
figure()
semilogy(SNR_db_B, P_error_symbol,'blue');
hold on;
semilogy(SNR_db_B, Theor_symbolErr,'red');
legend('Experimental','Theoretical');
xlabel('SNR in dB');
ylabel('Symbol Error Rate for 16-PSK');
legend('Experimental','Theoretical')
grid on;
set(gcf,'color', 'w');
% 3.
figure()
semilogy(SNR_db_B, P_error_bit,'blue');
hold on;
semilogy(SNR_db_B, Theor_bitErr,'red');
xlabel('SNR in dB');
ylabel('Bit Error Rate for 16-PSK');
legend('Experimental','Theoretical')
grid on;
set(gcf,'color', 'w');
```