Πολυτεχνείο Κρήτης
Σχολή ΗΜΜΥ
Τηλεπικοινωνιακά Συστήματα Ι

Παράδοση 3ης εργασίας

Ημερομηνία Παράδοσης: 20 Ιουνίου 2024

Μονάδες 130/1000

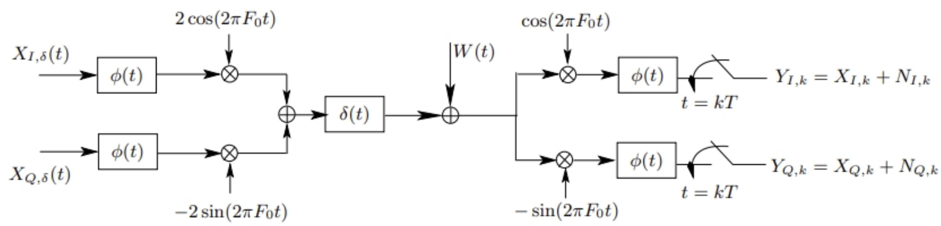
Ομάδα 25

	Φοιτητής 1	Φοιτητής 2
Επώνυμο	Σιώτος	Αγγελόπουλος
Όνομα	Μόδεστος	Γιώργος
Α.Μ.	2016030030	2016030083

Ώρες που απαιτήθηκαν για την υλοποίηση της άσκησης ≈ 15

Περιεχόμενα

A Μέρος Α′



Σχήμα 1: Τηλεπικοινωνιακό σύστημα για προσομοίωση

A.1 Δημιουργία δυαδικής ακολουθίας

Αρχικά δημιουργούμε δυαδική ακολουθία με 4N ισοπίθανα *bits*, για N = 200.

```
%Omada 25

clear all;
close all;

%Data
%Askisi A
N = 200;
A = 1;
A_srrc = 4;
T = 0.01;
over = 10;
Ts = T/over;
a=0.5;

Fs=1/Ts;
Nf=2048;

%1

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

A.2 Συνάρτηση *bits_to_4_PAM*

Στο 2^o βήμα συντάξαμε τη συνάρτηση *bits_to_4_PAM* η οποία, χρησιμοποιώντας κωδικοποίηση *Gray*, απεικονίζει τη δυαδική ακολουθία εισόδου *bit_seq* σε ακολουθία 4-*PAM* συμβόλων.

```
function [X] = bits_to_4_PAM(bit_seq,A)
j=1;
Z = [-3*A, -1*A, A , 3*A];
X=zeros(1,length(b)/2);
for i=1:2:length(b)
if(b(i)==0 && b(i+1)==0)
X(j) = Z(1);
elseif(b(i)==0 && b(i+1)==1)
X(j) = Z(2);
elseif(b(i)==1 && b(i+1)==1)
X(j) = Z(3);
elseif(b(i)==1 && b(i+1)==0)
X(j) = Z(4);
end
j=j+1;
end
end
```

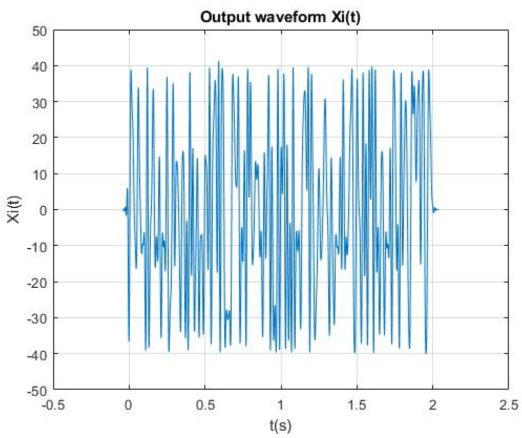
A.3 Δημιουργία X_I, X_Q

Στο 3^o βήμα απεικονίσαμε τα πρώτα $2Nbits$ της ακολουθίας του βήματος 1 στα $4-PAM$ σύμβολα $X_{I,n}$, για $n = 1, ..., N$, και τα επόμενα $2Nbits$ στα $4 - PAM$ σύμβολα $X_{Q,n}$, για $n = 1, ..., N$.

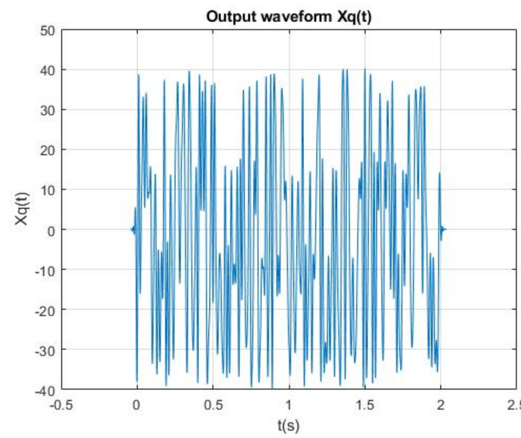
```
%2,3

b1 = bit_seq(1:2*N);
b2 =  bit_seq(2*N + 1:4*N);

X_I = bits_to_4_PAM(b1, A);
X_Q = bits_to_4_PAM(b2, A);
```

A.4 Κυματομορφές εξόδου $X_I(t), X_Q(t)$

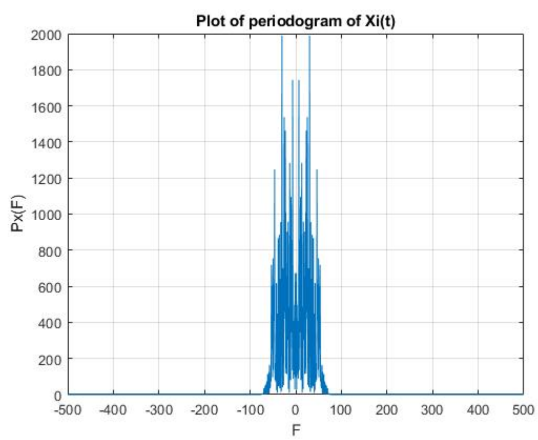
Έπειτα, φιλτράραμε τις ακολουθίες που δημιουργήσαμε στο προηγούμενο βήμα, με τη βοήθεια της γνωστής συνάρτησης *srrc_pulse* και παράχθηκαν στην έξοδο τα σήματα $X_I(t), X_Q(t)$. Χρησιμοποιήσαμε τις δοσμένες παραμέτρους $\mathrm{T} = 0.01$ *sec*, $over = 10$ και θέσαμε $A_{srrc} = 4, a = 0.5$. Επίσης πήραμε $Nf = 2048$.



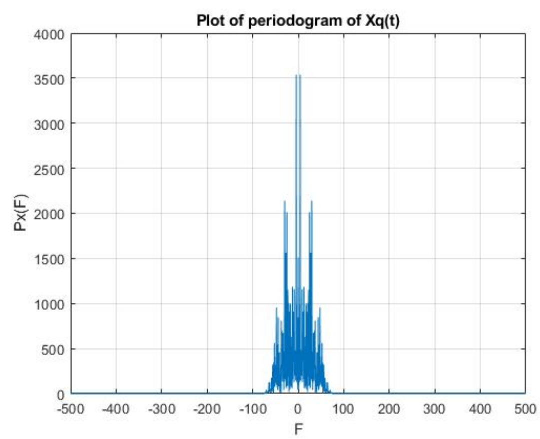
Σχήμα 2: Κυματομορφή $X_I(t)$



Σχήμα 3: Κυματομορφή $X_Q(t)$



Σχήμα 4: Περιοδόγραμμα $X_I(t)$



Σχήμα 5: Περιοδόγραμμα $X_Q(t)$

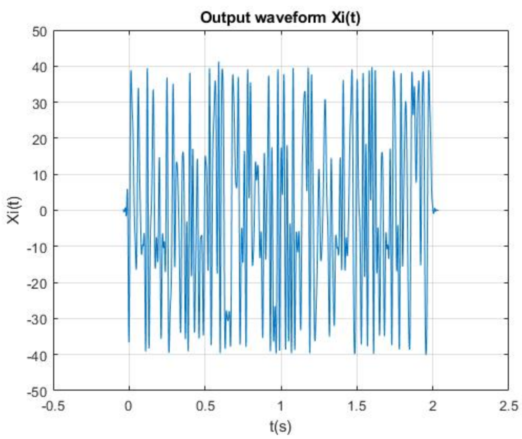
```
%4

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

[phi,t1]=srrc_pulse(T, Ts, A_srrc, a);

%create signals
Xi_n = 1/Ts*upsample(X_I,over);
Xi_t = conv(Xi_n,phi)*Ts;

```

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

%time vector
t2 = 0:Ts:N*T-Ts;
ti_conv = linspace(t1(1)+t2(1), t1(end)+t2(end),length(Xi_t));
tq_conv = linspace(t1(1)+t2(1), t1(end)+t2(end),length(Xq_t));

figure();
plot(ti_conv,Xi_t);
grid on;
title('Output waveform Xi(t)');
xlabel('t(s)');
ylabel('Xi(t)');

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

Ti_total = length(ti_conv)*T;
Tq_total = length(tq_conv)*T;

%PXF (from prev. assignment)

PXFi = ((abs(fftshift(fft(Xi_t,Nf))).^2)*Ts)./Ti_total;
PXFq = ((abs(fftshift(fft(Xq_t,Nf))).^2)*Ts)./Tq_total;

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

figure();
plot(f, PXFq);
title('Plot of periodogram of Xq(t)');
xlabel('F');
ylabel('Px(F)')
```

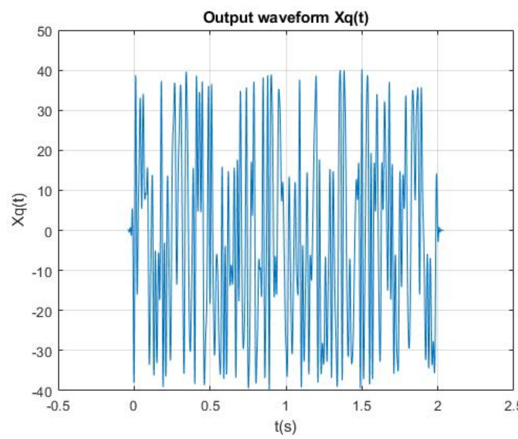
```
grid on;
```

A.5 Δημιουργία $X_I^{mod}(t), X_Q^{mod}(t)$

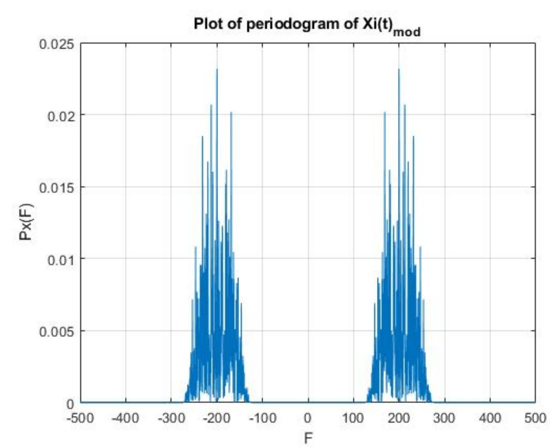
Για το συγκεκριμένο υποερώτημα, πολλαπλασιάζουμε τις κυματομορφές $X_I(t)$ και $X_Q(t)$ με τους φορείς που μας δίνονται στην εκφώνηση και δημιουργούμε τις κυματομορφές $X_I^{mod}(t), X_Q^{mod}(t)$ για $F_0 = 200Hz$. Ακολούθως σχεδιάστηκαν οι κυματομορφές και τα περιοδογράμματα των σημάτων όπως στο προηγούμενο ερώτημα.



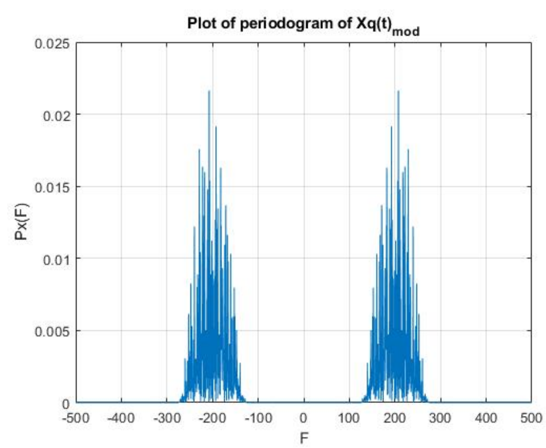
Σχήμα 6: Κυματομορφή X_I^{mod}



Σχήμα 7: Κυματομορφή X_Q^{mod}



Σχήμα 8: Περιοδόγραμμα X_I^{mod}



Σχήμα 9: Περιοδόγραμμα X_Q^{mod}

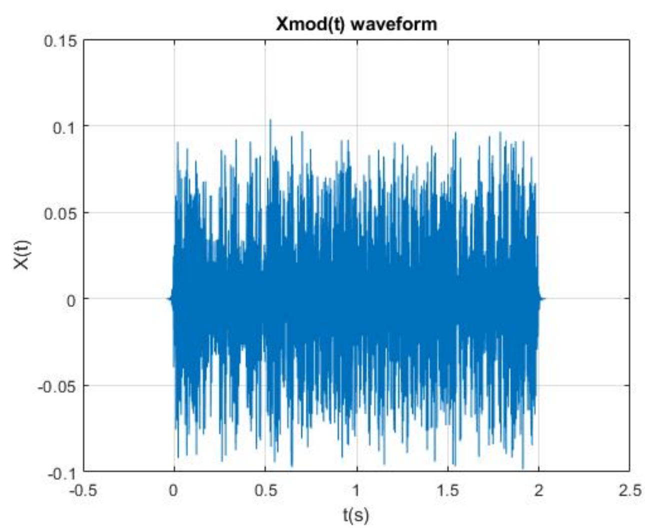
```
%5

F0 = 200;

Xi_mod_t = 2*Xi_t.*cos(2*pi*F0*ti_conv)*Ts;
Xq_mod_t = -2*Xq_t.*sin(2*pi*F0*tq_conv)*Ts;

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

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

T_total_i=length(Xi_mod_t)*Ts; %
T_total_q=length(Xq_mod_t)*Ts; %

PXFimod = ((abs(fftshift(fft(Xi_mod_t,Nf))).^2)*Ts)./T_total_i;
PXFqmod = ((abs(fftshift(fft(Xq_mod_t,Nf))).^2)*Ts)./T_total_q;

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

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

Παρατηρούμε ότι το φάσμα έχει μετακινηθεί γύρω απο τη συχνότητα $F_0 = 200Hz$ του φορέα, όπως ήταν αναμενόμενο.

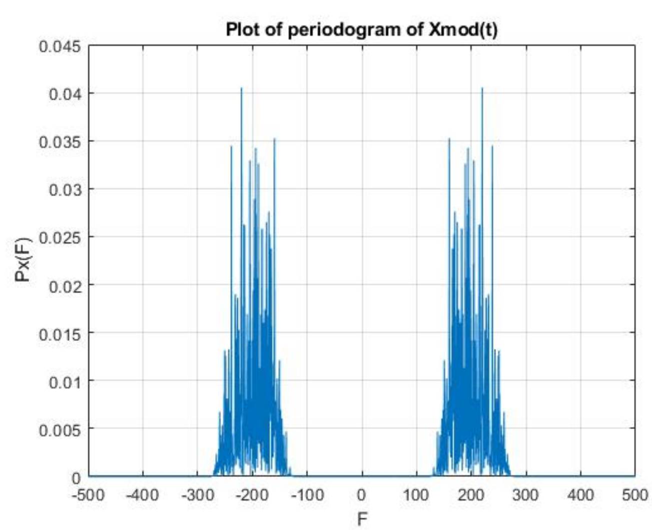
A.6 Άθροισμα κυματομορφών $X_I^{mod}(t), X_Q^{mod}(t)$

Κατόπιν, μας ζητήθηκε να σχηματίσουμε και να σχεδιάσουμε την είσοδο:

$$X^{mod}(t) = X_I^{mod}(t) + X_Q^{mod}(t)$$



Σχήμα 10: Κυματομορφή $X^{mod}(t)$



Σχήμα 11: Περιοδόγραμμα $X^{mod}(t)$

```
%6

X_t_mod = Xi_mod_t + Xq_mod_t;

figure();
%ti_conv=tq_conv
plot(ti_conv,X_t_mod);
grid on;
title('Xmod(t) waveform');
xlabel('t(s)');
ylabel('X(t)');


T_total=length(X_t_mod)*Ts;
PXFmodTotal = ((abs(fftshift(fft(X_t_mod,Nf))).^2)*Ts)./T_total;


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

Και εδώ, το φάσμα είναι μετατοπισμένο γύρω απο τη συχνότητα $F_0 = 200Hz$ και η κυματομορφή είναι πιο πυκνή, όπως είναι λογικό, αφού το $X^{mod}(t)$ αποτελεί το άθροισμα των $X_I^{mod}(t), X_Q^{mod}(t)$

A.7 Ιδανικό κανάλι

Θεωρούμε ότι το κανάλι είναι ιδανικό.

A.8 Προσθήκη Gaussian θορύβου

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

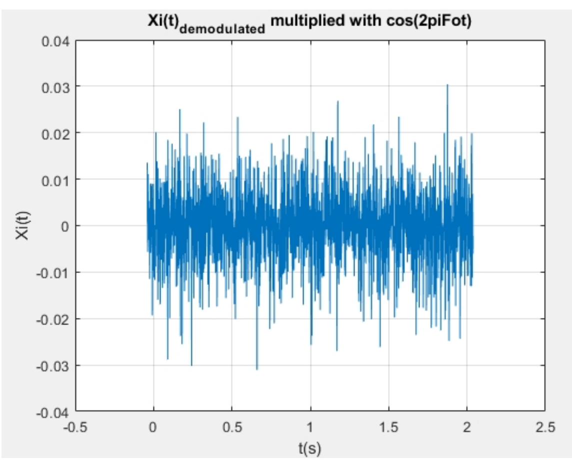
$$\sigma_W^2 = \frac{10A^2}{T_s \cdot 10^{\frac{\mathrm{SNR}_{dB}}{10}}}$$

Πήραμε $SNR_{db} = 20$.

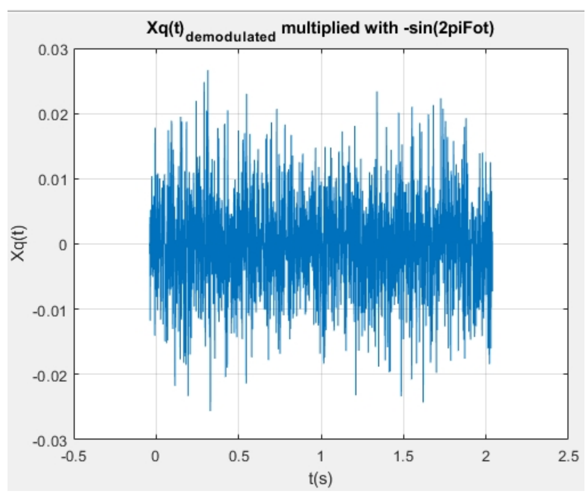
```
%8

SNR = 20;
var_w = 10*(A^2)/(Ts*10^(SNR/10));
noise = sqrt(var_w)*randn(1,length(X_t_mod));
X_mod_noise = X_t_mod + noise;
```

A.9 Ενθόρυβη Κυματομορφή

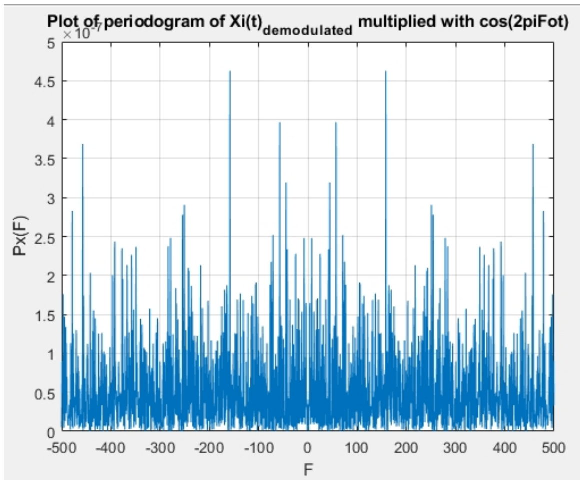
Έπειτα, διακλαδώσαμε την ενθόρυβη κυματομορφή και την πολλαπλασιάσαμε με φορείς $\cos(2\pi F_0 t)$ και -$\sin(2\pi F_0 t)$.



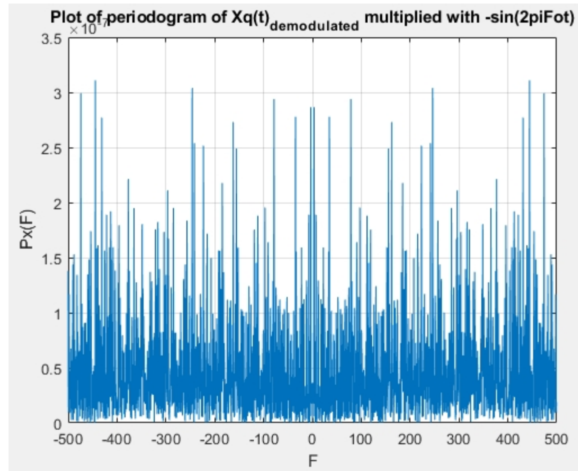
Σχήμα 12: Κυματομορφή X_i_demod



Σχήμα 13: Κυματομορφή X_q_demod



Σχήμα 14: Περιοδόγραμμα PxF_i



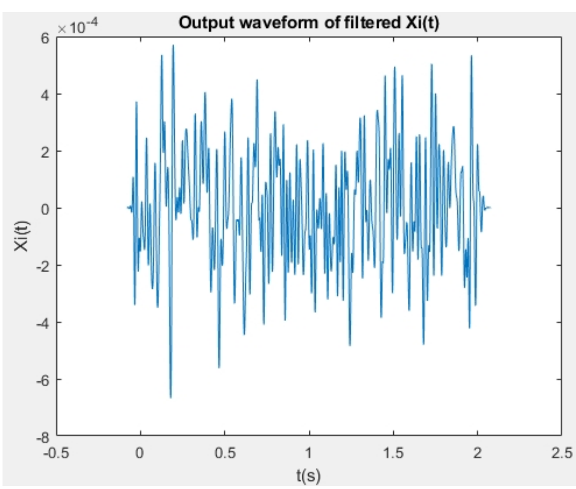
Σχήμα 15: Περιοδόγραμμα PxF_q

```
%9

Xi_demod = X_mod_noise.*cos(2*pi*F0*ti_conv)*Ts;
Xq_demod = X_mod_noise.*(-1*sin(2*pi*F0*tq_conv))*Ts;

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

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

T_total_i=length(Xi_demod)*Ts;
XF_i=fftshift(fft(Xi_demod,Nf))*Ts;
PxF_i=(abs(XF_i).^2)/T_total_i

T_total_q=length(Xq_demod)*Ts;
XF_q=fftshift(fft(Xq_demod,Nf))*Ts;
PxF_q=(abs(XF_q).^2)/T_total_q
```

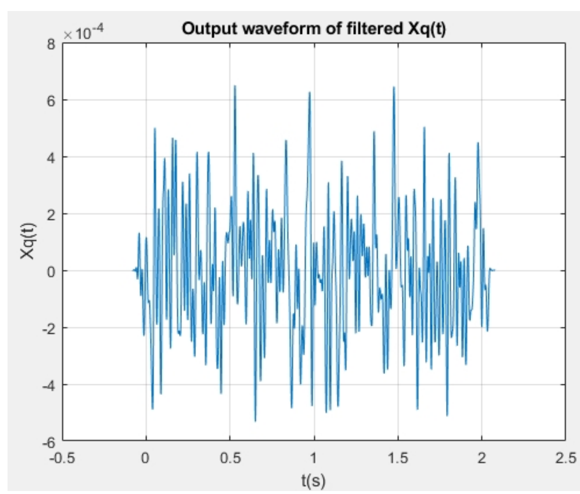
```
figure();
plot(f, PxF_i);
title('Plot of periodogram of Xi(t)_d_e_m_o_d_u_l_a_t_e_d
   multiplied with cos(2piFot)');
xlabel('F');
ylabel('Px(F)')
grid on;

figure();
plot(f, PxF_q);
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;
```

A.10 Πέρασμα κυματομορφών από προσαρμοσμένα φίλτρα

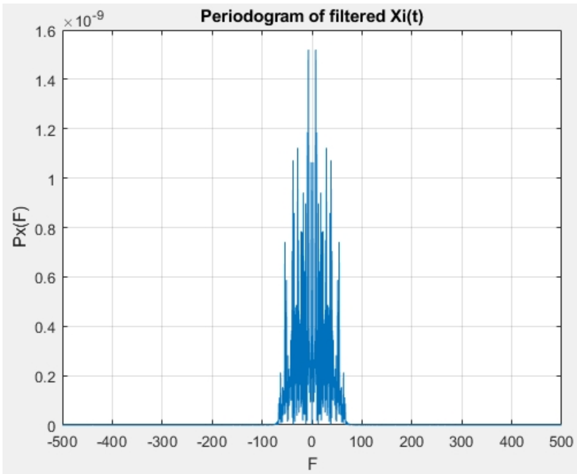
Περάσαμε τις κυματομορφές που υπολογίσαμε στο προηγούμενο βήμα από τα προσαρμοσμένα φίλτρα.



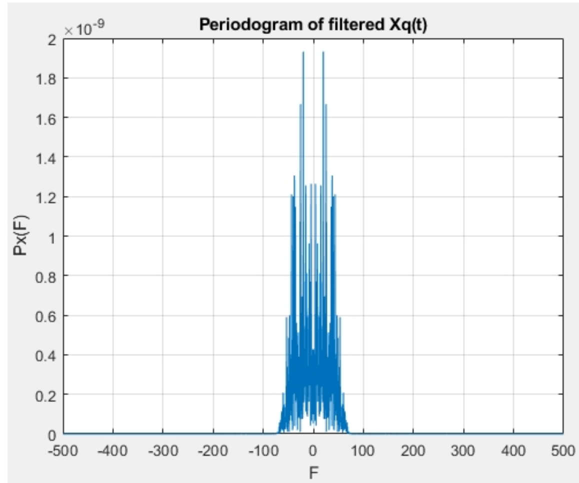
Σχήμα 16: Κυματομορφή φιλτραρισμένου $X_i(t)$



Σχήμα 17: Κυματομορφή φιλτραρισμένου $X_q(t)$



Σχήμα 18: Περιοδόγραμμα φιλτραρισμένου $X_i(t)$



Σχήμα 19: Περιοδόγραμμα φιλτραρισμένου $X_q(t)$

```
%10
Xi_demod = conv(Xi_demod,phi)*Ts;
Xq_demod = conv(Xq_demod,phi)*Ts;

t_conv2 = min(ti_conv)+min(t1):Ts:max(t1)+max(ti_conv)
f = -Fs/2:Fs/Nf:Fs/2-Fs/Nf;

figure()
plot(t_conv2,Xi_demod)
title('Output waveform of filtered Xi(t)');
xlabel('t(s)');
ylabel('Xi(t)');

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

T_total_i=length(Xi_demod)*Ts;
XF_i=fftshift(fft(Xi_demod,Nf))*Ts;
PxF_i=(abs(XF_i).^2)/T_total_i

T_total_q=length(Xq_demod)*Ts;
XF_q=fftshift(fft(Xq_demod,Nf))*Ts;
```

```
PxF_q=(abs(XF_q).^2)/T_total_q


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

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

A.11 Χρήση της scatterplot

```
%11

timesteps_for_sample_i = (2*A*T/Ts)+1 : over : length(Xi_demod)
    -(2*A*T/Ts);
timesteps_for_sample_q = (2*A*T/Ts)+1 : over : length(Xq_demod)
    -(2*A*T/Ts);

W_cos_sampled = Xi_demod(timesteps_for_sample_i);
W_sin_sampled = Xq_demod(timesteps_for_sample_q);

for i=1:N
Samples(i,1)=W_cos_sampled(i);
Samples(i,2)=W_sin_sampled(i);
end

scatterplot(Samples)
```

A.12 Δημιουργία συνάρτησης *Detect_4_PAM*

Για την *Detect_4_PAM*, αρχικοποιήσαμε έναν πίνακα με τα σύμβολα (4 αφού έχουμε $4PAM$), και την τελική *est_X* με μηδενικά, ένα για κάθε δείγμα της Y. Τέλος , για κάθε δείγμα της Y, υπολογίζουμε τις αποστάσεις από κάθε σύμβολο και επιλέγουμε το σύμβολο που έχει την ελάχιστη απόσταση από το δείγμα ως το πιο σωστό σύμβολο με την βοήθεια του *min_distance_symbol*, σύμφωνα με τον κανόνα του εγγύτερου γείτονα.

```
%12

for i=1:N
X_I_possible(i) = detect_4_PAM(W_cos_sampled(i),A);
X_Q_possible(i) = detect_4_PAM(W_sin_sampled(i),A);
end
```

```
function est_X = detect_4_PAM(Y,A)

possible_symbols = [-3*A, -A, A, 3*A];

est_X = zeros(size(Y));
for i = 1:length(Y)
[~, min_distance_symbol] = min(abs(Y(i) - possible_symbols));
est_X(i) = possible_symbols(min_distance_symbol);
end
end
```

A.13 Σφάλματα απόφασης συμβόλου

Χρησιμοποιώντας τις ακολουθίες εισόδου και τις αποφάσεις που πήραμε από το πάνω ερώτημα, τα βάλαμε σε πίνακες και με ένα *forloop* ξεκινήσαμε να συγκρίνουμε ένα προς ένα τις ακολουθίες με τις αποφάσεις. Όπου ήτανε διαφορετικές, αυξάνουμε έναν *errorcounter* κατά 1. Σύνολο βρήκαμε 189 191 λάθη. Παρακάτω είναι ο κώδικας.

```
%13

errors_13 = 0;
actual_symbols = [X_I ; X_Q];
possible_symbols = [X_I_possible ; X_Q_possible];

for i=1:N
if((actual_symbols(1, i) ~= possible_symbols(1, i)) || (
   actual_symbols(2, i) ~= possible_symbols(2, i)))
errors_13 = errors_13 + 1;
end
end

fprintf('Total errors for Task 13: %d\n', errors_13);
```

A.14 Η συνάρτηση *PAM_4_to_bits*

Για την *PAM_4_to_bits*, σε ένα *for loop*, ανάλογα αν το X είναι $[-3A, -A, A, 3A]$, βάζουμε τις σωστές τιμές *bit* (0 ή 1) στα επόμενα δύο *est_bit*, και αυξάνουμε τον *counter* μέχρι να εξαντληθεί το X.

```
function [est_bit] = PAM_4_to_bits(X,A)
i=1;
for i=1:length(X)

if(X(i)== -3*A)
est_bit(i)=0;
est_bit(i+1)=0;

elseif(X(i)== -A)
est_bit(i)=0;
est_bit(i+1)=1;

elseif(X(i)== A)
est_bit(i)=1;
est_bit(i+1)=1;

elseif(X(i)== 3*A)
est_bit(i)=1;
est_bit(i+1)=0;
end
i=i+2;
end
end
```

A.15 Υπολογισμός σφαλμάτων

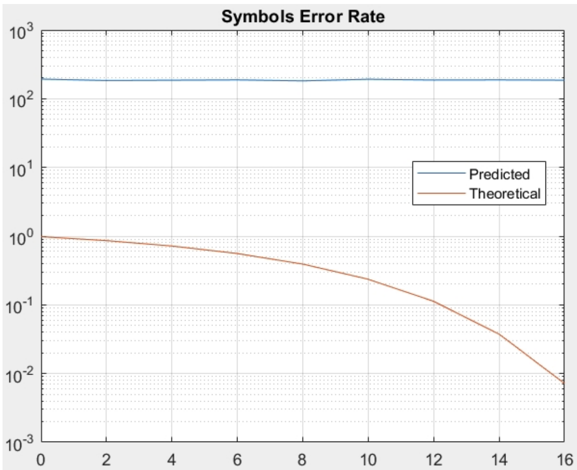
Αφού περάσαμε τα εκτιμόμενα σύμβολα της *inphase* και *qudrature* από την προηγούμενη *function*, βγάλαμε μια εκτιμόμενη *bit sequence*, την οποία συγκρίναμε σημείο προς σημείο με την *bit sequence* στην αρχή της άσκησης. Κάθε φορά που είναι διαφορετικά, αυξάνουμε έναν *error counter* κατά 1. Βρήκαμε περίπου 205 207 *errors*.

```
%15

errors_15 = 0;
iBits=PAM_4_to_bits(X_I_possible,A);
qBits=PAM_4_to_bits(X_Q_possible,A);
bit_est =[iBits qBits];

for i=1:length(bit_est)
if (bit_est(i) ~= bit_seq(i))
errors_15 = errors_15 +1;
end
end

fprintf('Total errors for Task 15: %d\n', errors_15);
```

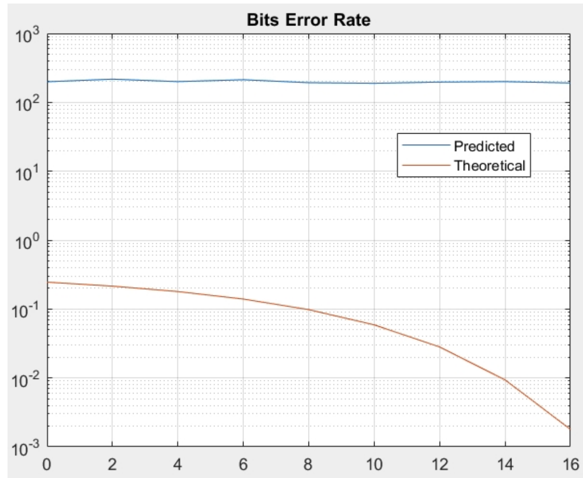
B Πιθανότητα σφάλματος συμβόλου και bit

Στο 2ο βήμα, υπολογίζουμε την πιθανότητα σφάλματος συμβόλου και *bit* με τη μέθοδο *Monte Carlo*. Για *SNR* $[0 : 2 : 16]$ υπολογίσαμε από την αρχή το A μέρος $K = 200$ φορές και σώσαμε τις τιμές των *predicted errors* από τα ερωτήματα 13 και 15 σε πίνακες *Esymbols_per_SNR* και *Ebits_per_SNR* αντίστοιχα.Έπειτα δημιοργήσαμε δύο ίδιους πίνακες αλλα με *theor* στο τέλος για να υπολογίσουμε τα θεωρητικά *error rates* με την βοήθεια μιας μικρής συνάρτησης. Οι θεωρητικές τιμές υπολογίστηκαν ως εξής:

```
function  y = 0.5 * erfc(x / sqrt(2));
Esymbols_per_SNR_theor(th) = 3*Q(sqrt(0.2.*(10.^(SNR/10))));
Ebits_per_SNR_theor(th) = Esymbols_per_SNR_theor(th)./4;
```



Σχήμα 20: Κυματομορφή σφάλματος συμβόλου



Σχήμα 21: Κυματομορφή σφάλματος *bits*