
ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΗΜΜΥ
Τηλεπικοινωνιακά Συστήματα Ι
Παράδοση 3ης εργασίας
Ημερομηνία Παράδοσης: 10/06/2024

Ομάδα 120

Επώνυμο	Δήμας	Λαμπράκης
Όνομα	Χρήστος	Μιχάλης
A.M.	2021030183	2020030077

Πλήθος ωρών ενασχόλησης:15

A. Σε αυτήν την άσκηση, θα προσομοιώσουμε το τηλεπικοινωνιακό σύστημα του Σχήματος, υποθέτοντας ότι χρησιμοποιείται διαμόρφωση 16-QAM, και θα μελετήσουμε την απόδοσή του.

1. Για δεδομένα N (ενδεικτικά $N=200$), να δημιουργήσετε δυαδική ακολουθία με στοιχεία $4N$ ισοπίθανα bits.

Λύση:

Ακολουθεί ο κώδικας MatLab:

```
1 %% Variable decleration
2 N = 200;
3 A = 1;
4 A_SRRC = 4;
5 a = 0.5;
6 T = 0.01;
7 over = 10;
8 Ts = T/over;
9 Fs = 1/Ts;
10 Nf = 2048;
11
12 %% A.1
13 bits = (sign(randn(4*N, 1)) + 1)/2;
```

2. (15) Να συντάξετε συνάρτηση

$$X = \text{bits_to_4PAM}(\text{bit_seq}, A)$$

η οποία, χρησιμοποιώντας κωδικοποίηση Gray, απεικονίζει την δυαδική ακολουθία σε εισόδου bit_seq σε ακολουθία 4-PAM συμβόλων.

Λύση:

Ακολουθεί ο κώδικας MatLab:

```
1 function [x] = bits_to_4_PAM(bit_seq, A)
2
3 for i = 1:2:length(bit_seq)
4     bits = bit_seq(i:i+1);
5     if isequal(bits, [0;0])
6         x((i+1)/2) = 3*A;
7     elseif isequal(bits, [0;1])
8         x((i+1)/2) = 1*A;
9     elseif isequal(bits, [1;1])
10        x((i+1)/2) = -1*A;
11    elseif isequal(bits, [1;0])
12        x((i+1)/2) = -3*A;
```

```

13     end
14 end
15 end

```

3. Να απεικονίσετε τα πρώτα $2N$ bits της ακολουθίας του βήματος 1 στα 4-PAM σύμβολα $X_{1,n}$, για $n = 1, \dots, N$. και τα επόμενα $2N$ bits στα 4-PAM σύμβολα $X_{Q,n}$, για $n = 1, \dots, N$.

Λύση:

Ακολουθεί ο κώδικας MatLab:

```

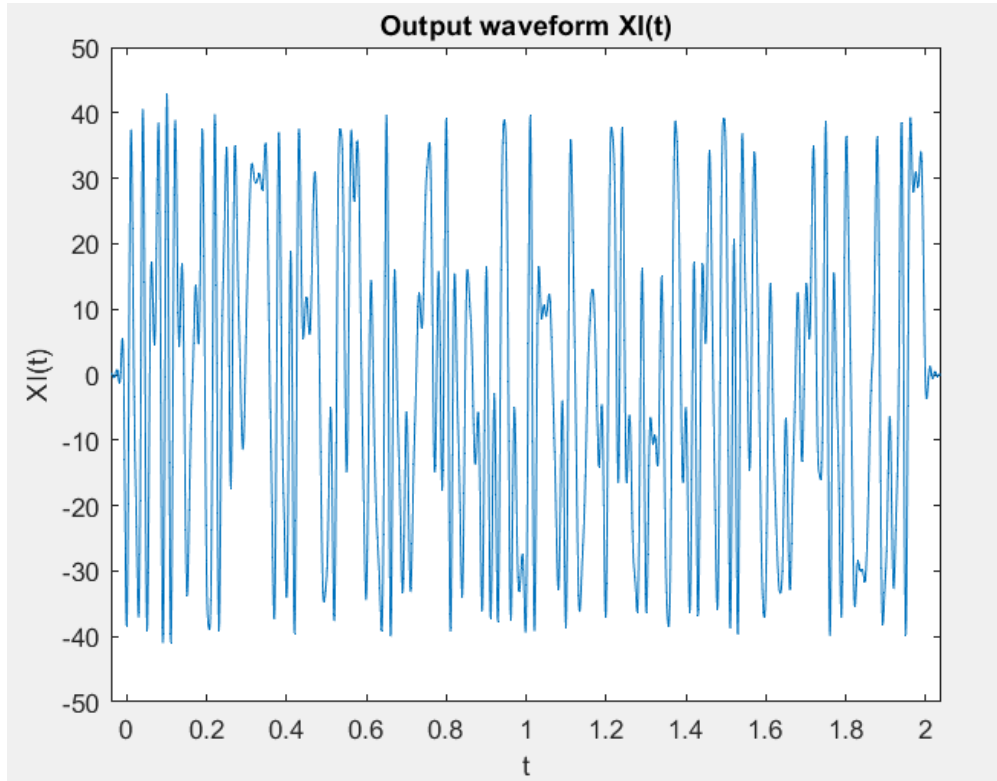
1 %% A.3
2
3 % For bits in the [0,2N]
4 bitXI = bits(1:2*N);
5 XIn = bits_to_4_PAM(bitXI, A);
6
7 % For bits in the [2N+1,4N]
8 bitXQ = bits(2*N+1:4*N);
9 XQn = bits_to_4_PAM(bitXQ, A);

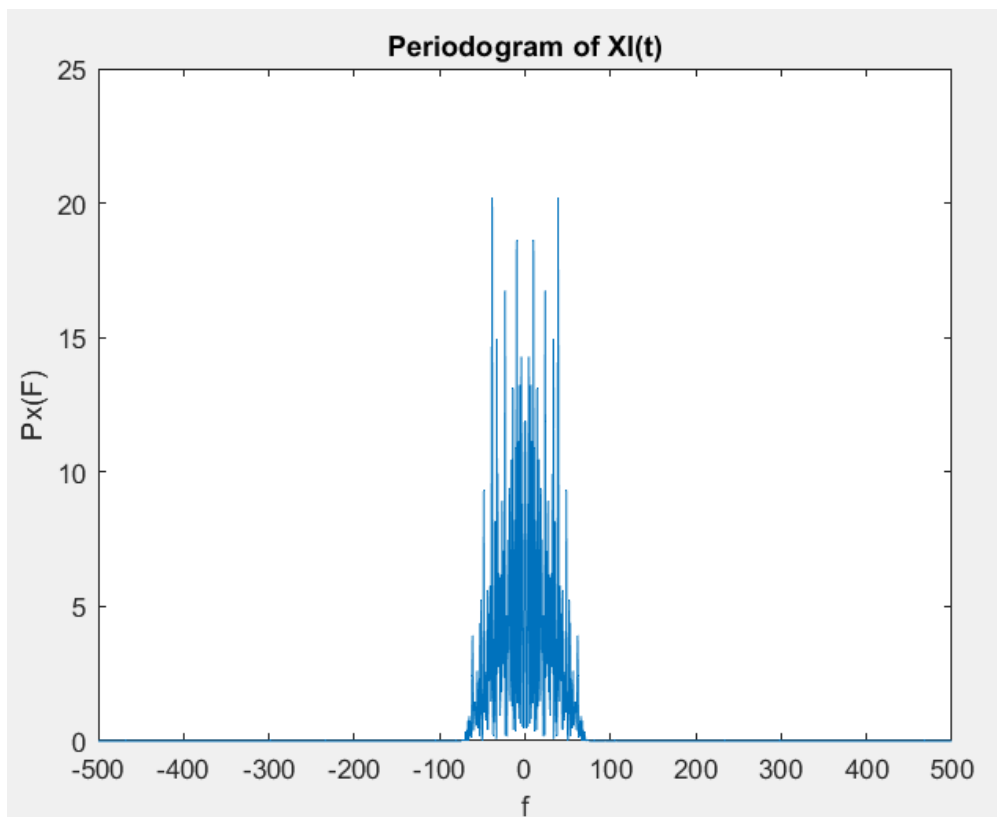
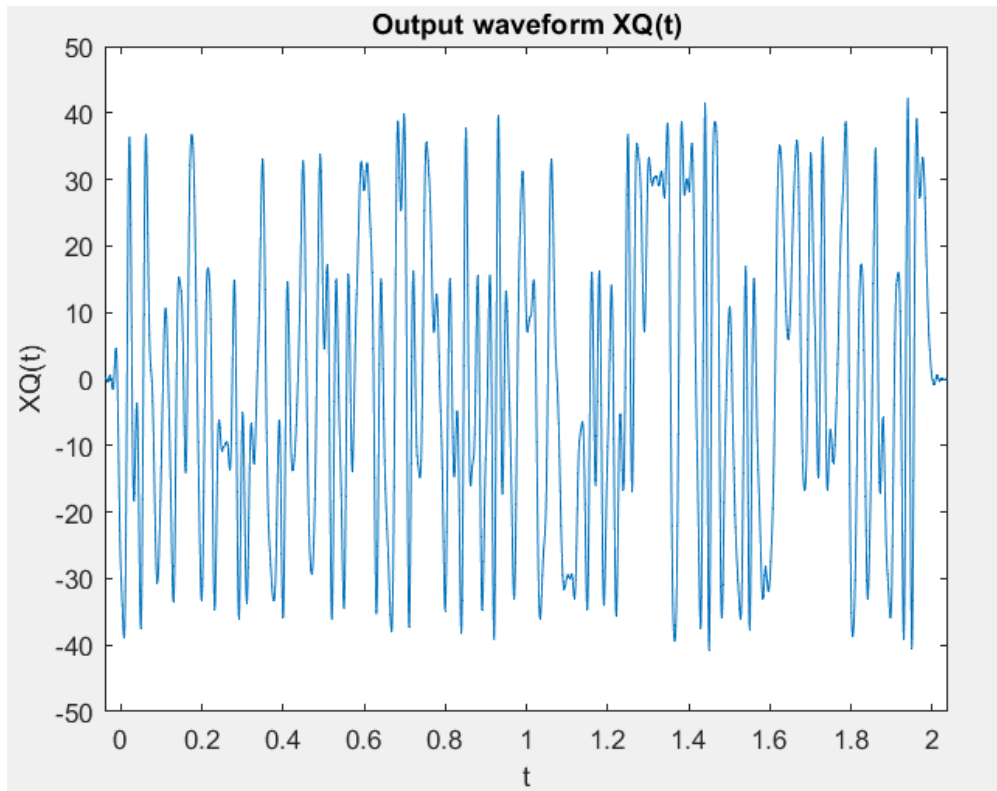
```

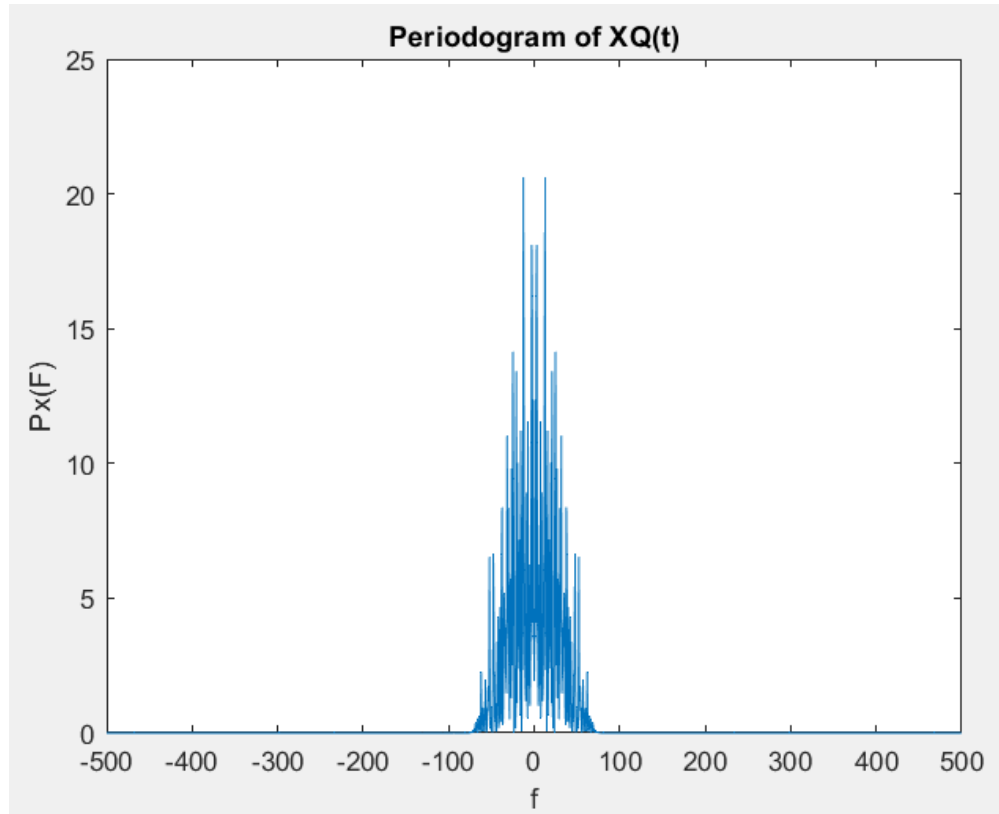
4. (5) Να περάσετε τις ακολουθίες $X_{1,n}$, και $X_{Q,n}$, από τα SRRC φίλτρα μορφοποίησης και υποθέτοντας περίοδο συμβόλου $T = 0.01sec$, $over = 10$, $T_s = \frac{T}{over}$, να σχηματίσετε και να σχεδιάσετε τις κυματομορφές εξόδου $X_I(t)$ και $X_Q(t)$ και τα περιοδογράμμά τους.

Λύση:

Παρακάτω φαίνονται οι κυματομορφές:







Και ο κώδικας MatLab:

```

1 %% A.4
2
3 % Creating the pulse
4 [phi, t] = srrc_pulse(T, over, A_SRRC, a);
5
6 % Upsample the 2 sequencies
7 XIn_up = (1/Ts)*upsample(XIn, over);
8 XI_up_time = 0:Ts:N*T-Ts;
9
10 XQn_up = (1/Ts)*upsample(XQn, over);
11 XQ_up_time = XI_up_time;
12
13 % Convolutions and time
14 XIt = Ts*conv(XIn_up, phi);
15 XQt = Ts*conv(XQn_up, phi);
16 conv_t = t(1) + XQ_up_time(1):Ts:t(end) + XQ_up_time(end);
17
18 % Output waveform plots
19 figure;
20 plot(conv_t, XIt);
21 title("Output waveform XI(t)");

```

```

22 xlabel("t");
23 ylabel("XI(t)");
24 xlim([conv_t(1) conv_t(end)]);
25
26 figure;
27 plot(conv_t, XQt);
28 title("Output waveform XQ(t)");
29 xlabel("t");
30 ylabel("XQ(t)");
31 xlim([conv_t(1) conv_t(end)]);
32
33 % Frequency Range
34 f_axis = linspace(-Fs/2,(Fs/2-Fs/Nf), Nf);
35
36 % Periodograms creation
37 XIF = fftshift(fft(XIt, Nf)*Ts);
38 num = power(abs(XIF),2);
39 T_total = conv_t(end) - conv_t(1);
40 PXIF = num/T_total;
41
42 XQF = fftshift(fft(XQt, Nf)*Ts);
43 num = power(abs(XQF),2);
44 T_total = conv_t(end) - conv_t(1);
45 PXQF = num/T_total;
46
47 % Periodogram plots
48 figure;
49 plot(f_axis, PXIF);
50 title("Periodogram of XI(t)");
51 xlabel("f");
52 ylabel("Px(F)");
53
54 figure;
55 plot(f_axis, PXQF);
56 title("Periodogram of XQ(t)");
57 xlabel("f");
58 ylabel("Px(F)");

```

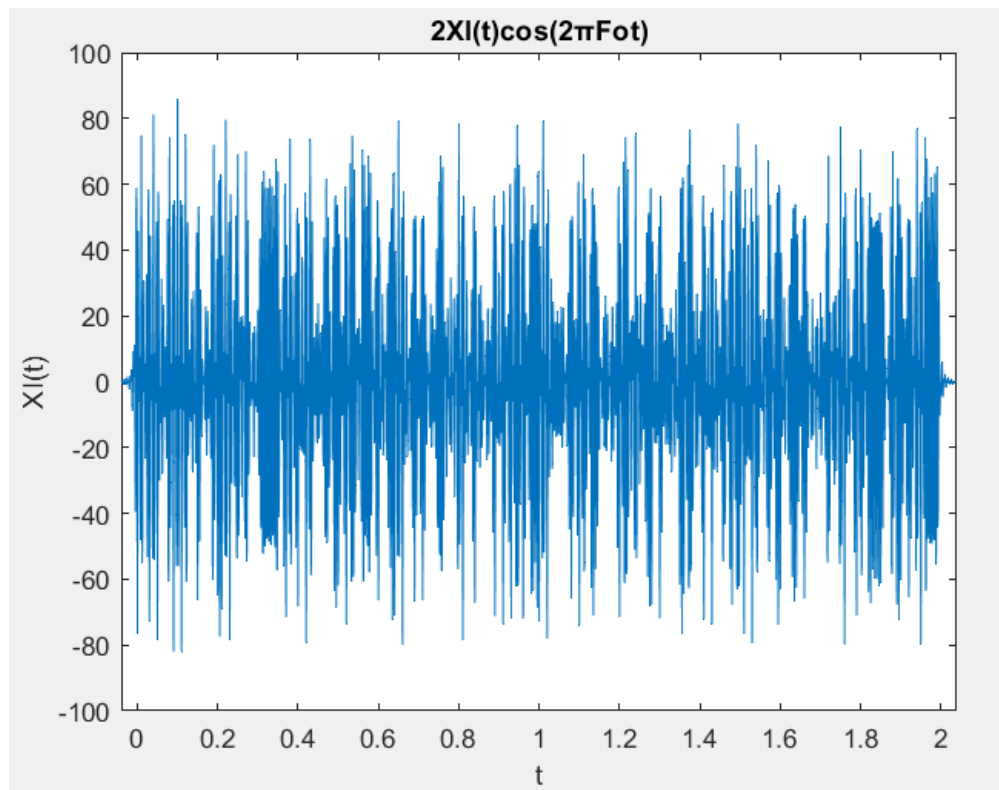
5. (5) Να πολλαπλασιάσετε τις κυματομορφές $X_1(t)$ και $X_Q(t)$ με τους αντίστοιχους φορείς (ενδεικτικά $F_0 = 200Hz$) και να δημιουργήσετε τις κυματομορφές:

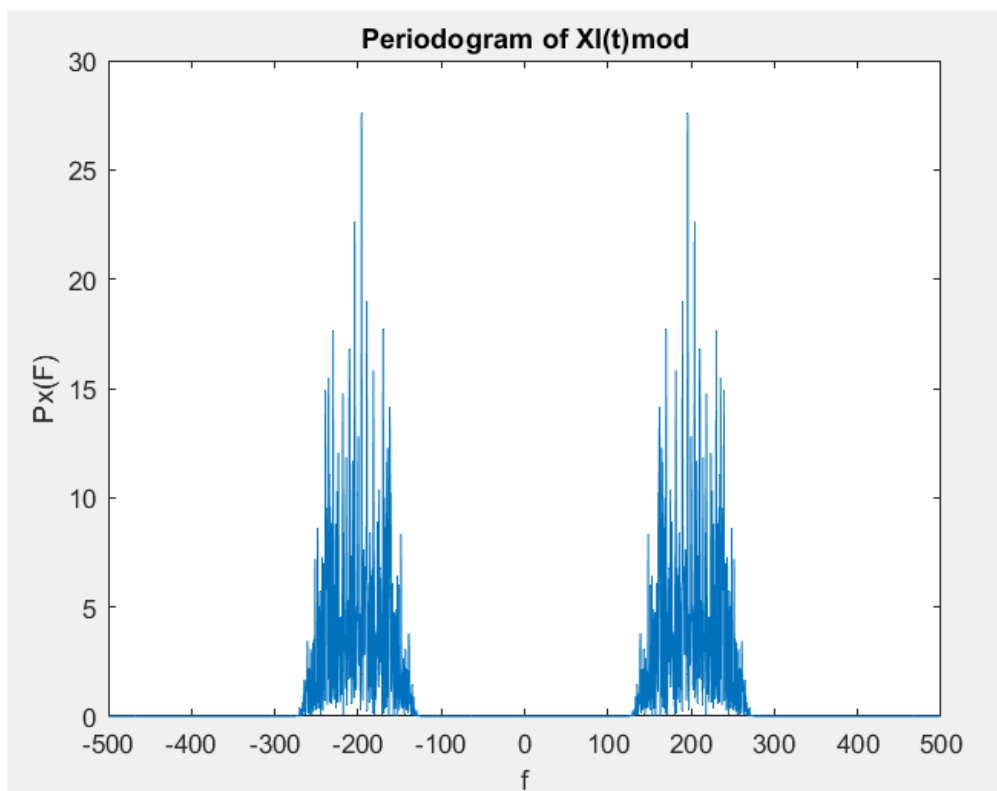
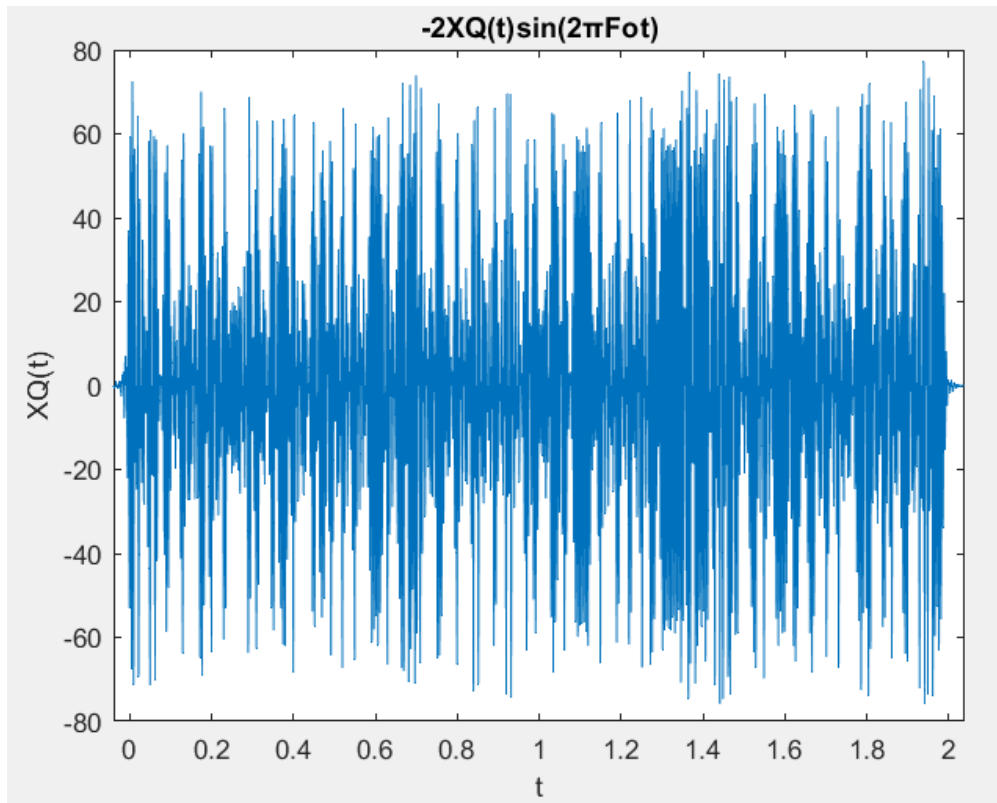
$$X_1^{\text{mod}}(t) = 2X_1(t) \cos(2\pi F_0 t), X_Q^{\text{mod}}(t) = -2X_Q(t) \sin(2\pi F_0 t)$$

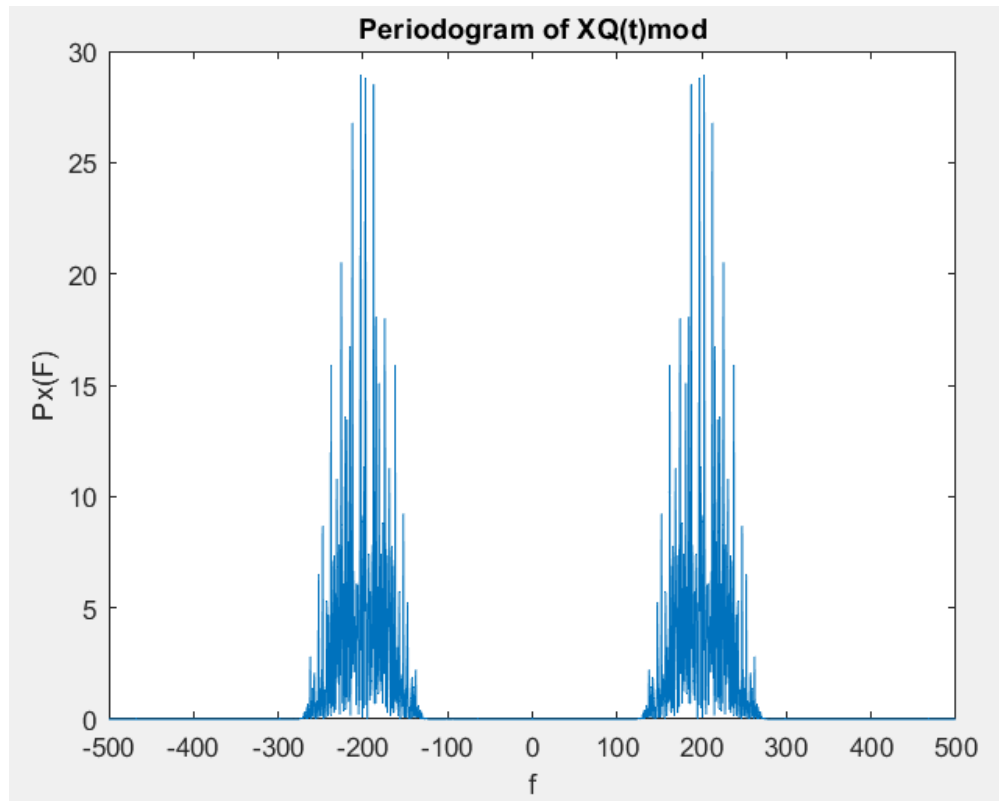
Να σχεδιάσετε τις κυματομορφές που προκύπτουν καθώς και τα αντίστοιχα περιοδογράμματα της. Τι παρατηρείτε;

Λύση:

Κάνοντας τις κατάλληλες ενέργειες προκύπτουν οι κυματομορφές:







Και ο κώδικας MatLab:

```

1 %% A.5
2
3 Fo = 200;
4 XImod_t = 2*XIt.*cos(2*pi*Fo*conv_t);
5 XQmod_t = -2*XQt.*sin(2*pi*Fo*conv_t);
6
7 % Plots of the new waveforms
8 figure;
9 plot(conv_t, XImod_t);
10 title("2XI(t)cos(2piFot)");
11 xlabel("t");
12 ylabel("XI(t)");
13 xlim([conv_t(1) conv_t(end)]);
14
15 figure;
16 plot(conv_t, XQmod_t);
17 title("-2XQ(t)sin(2piFot)");
18 xlabel("t");
19 ylabel("XQ(t)");
20 xlim([conv_t(1) conv_t(end)]);
21

```

```

22 % Periodograms creation
23 XIFmod = fftshift(fft(XImod_t, Nf)*Ts);
24 num = power(abs(XIFmod),2);
25 T_total = conv_t(end) - conv_t(1);
26 PXIFmod = num/T_total;
27
28 XQFmod = fftshift(fft(XQmod_t, Nf)*Ts);
29 num = power(abs(XQFmod),2);
30 T_total = conv_t(end) - conv_t(1);
31 PXQFmod = num/T_total;
32
33 % Periodogram plots
34 figure;
35 plot(f_axis, PXIFmod);
36 title("Periodogram of XI(t)mod");
37 xlabel("f");
38 ylabel("Px(F)");
39
40 figure;
41 plot(f_axis, PXQFmod);
42 title("Periodogram of XQ(t)mod");
43 xlabel("f");
44 ylabel("Px(F)");

```

Όπως παρατηρούμε από τα περιοδογράμματα, η διαμόρφωση έχει γίνει γύρω από την συχνότητα $\pm 200\text{Hz}$.

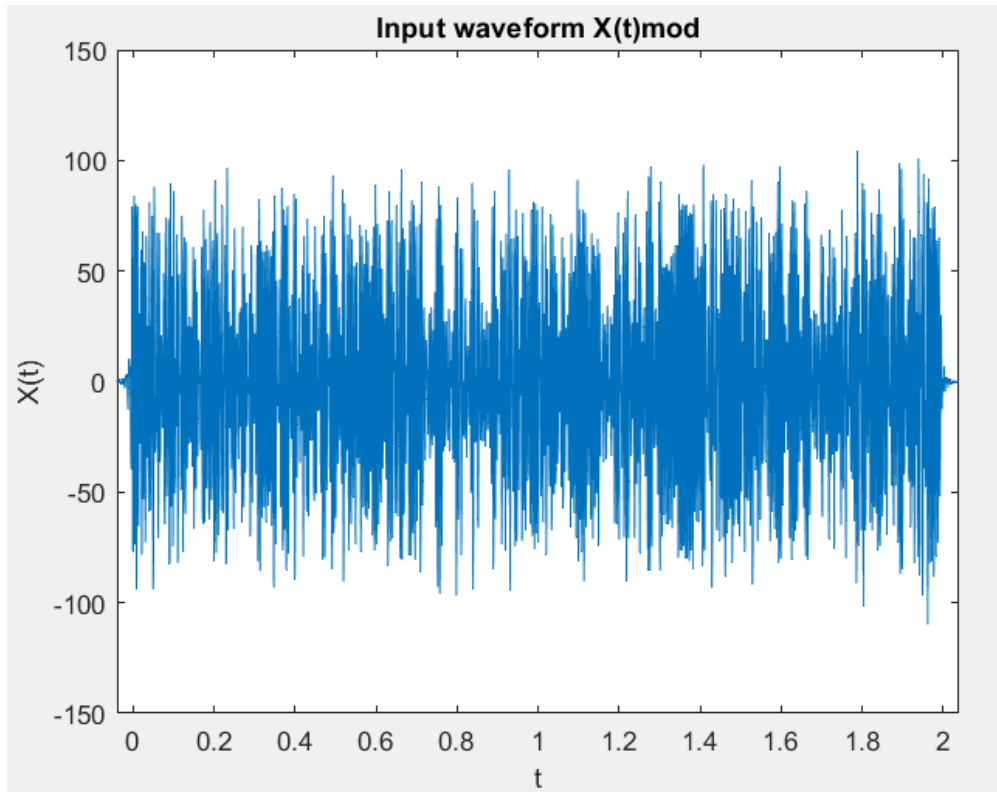
6. (5) Να σχηματίσετε και να σχεδιάσετε την είσοδο του καναλιού,

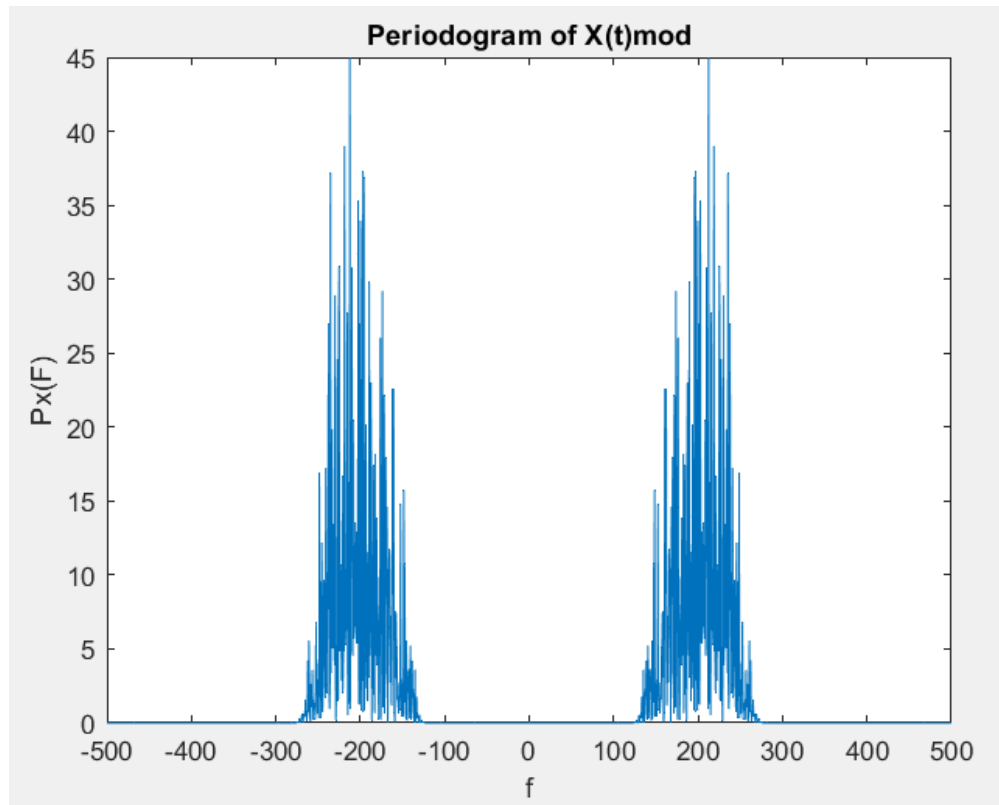
$$X^{\text{mod}}(t) = X_1^{\text{mod}}(t) + X_Q^{\text{mod}}(t)$$

και το περιοδογράμματά της. Τι παρατηρείτε;

Λύση:

Προκύπτουν οι κυματομορφές:





Και ο κώδικας MatLab:

```

1 %% A.6
2
3 % Input signal creation
4 Xmod_t = XImod_t + XQmod_t;
5
6 % Plot of Xmod(t)
7 figure;
8 plot(conv_t, Xmod_t);
9 title("Input waveform X(t)mod");
10 xlabel("t");
11 ylabel("X(t)");
12 xlim([conv_t(1) conv_t(end)]);
13
14 % Periodogram creation
15 Xmod_F = fftshift(fft(Xmod_t, Nf)*Ts);
16 num = power(abs(Xmod_F),2);
17 T_total = conv_t(end) - conv_t(1);
18 PXmod = num/T_total;
19
20 % Periodogram plot
21 figure;

```

```

22 plot(f_axis, PXmod);
23 title("Periodogram of X(t)mod");
24 xlabel("f");
25 ylabel("Px(F)");

```

Παρατηρώντας τα περιοδογράμματα, το πλάτος έχει αυξηθεί, πράγμα λογικό αφού πραγματοποιήσαμε πρόσθεση. Επιπλέον, όμοια με πριν η διαμόρφωση έχει γίνει γύρω από την συχνότητα $\pm 200\text{Hz}$.

7. Να υποθέσετε ότι το κανάλι είναι ιδανικό.

Λύση:

Υποθέτουμε ότι το κανάλι είναι ιδανικό.

8. (5) Στην έξοδο του καναλιού, να προσθέσετε λευκό Gaussian θόρυβο $W(t)$ με διασπορά ίση με

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

Λύση:

Ακολουθεί ο κώδικας MatLab:

```

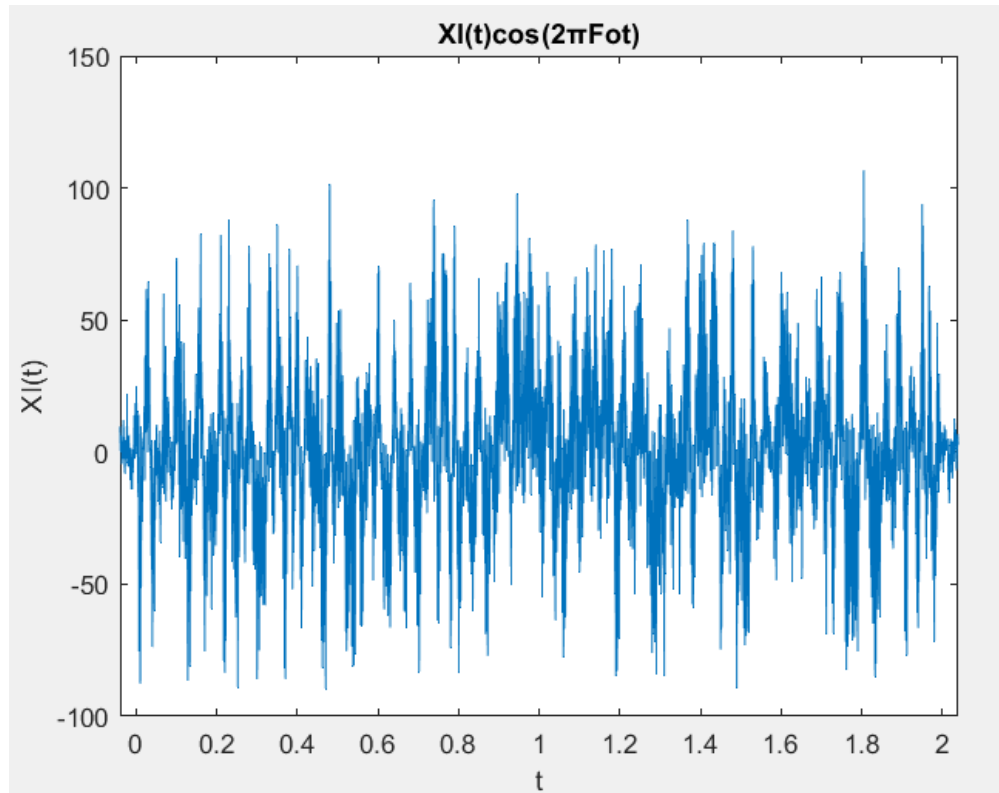
1 %% A.8
2
3 % Noise creation
4 SNR = 10;
5 SNR = 20;
6 variance = (10*(A^2))/(Ts*(10^(SNR/10)));
7 gaussian_noise = sqrt(variance)*randn(1,length(Xmod_t));
8
9 % Adding noise to the waveform

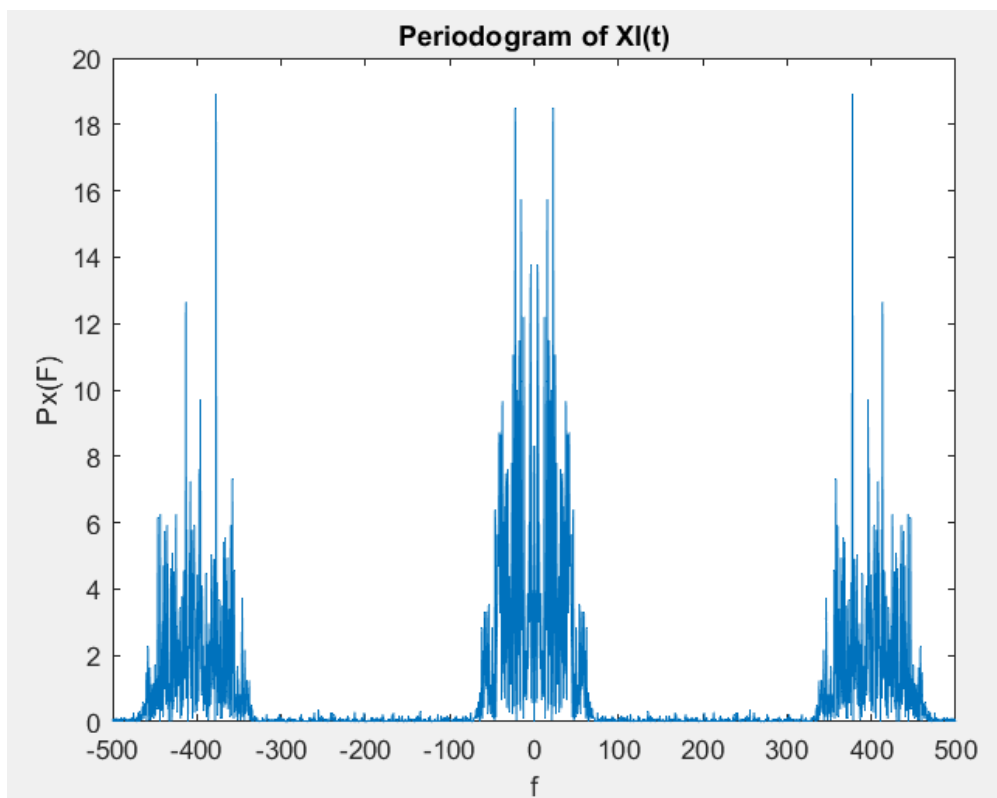
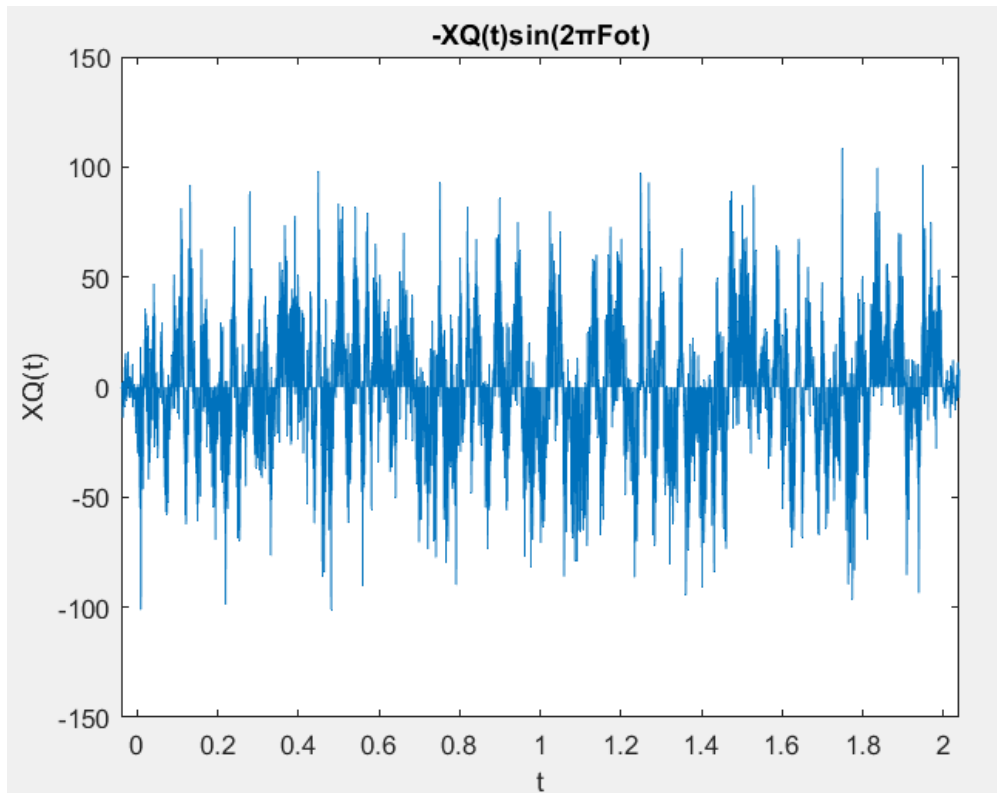
```

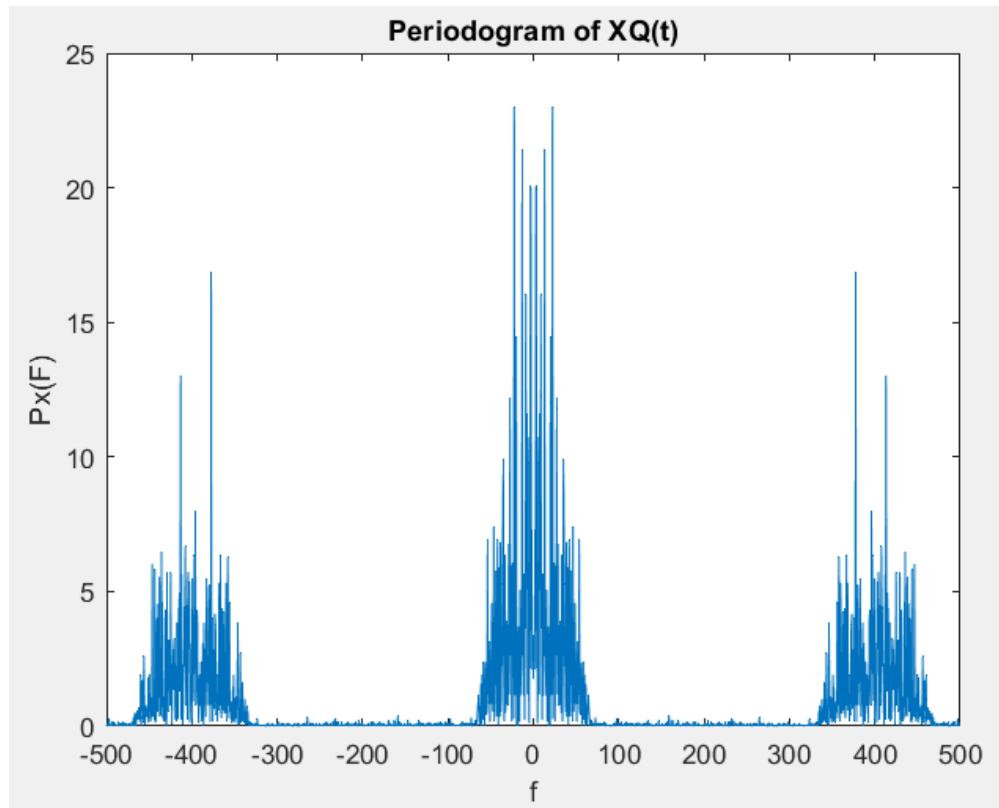
9. (5) Στον δέκτη να διακλαδώσετε την ενθόρυβη κυματομορφή και να την πολλαπλασιάσετε με φορείς $\cos(2\pi F_0 t)$ και $-\sin(2\pi F_0 t)$, αντίστοιχα. Να σχεδιάσετε τις κυματομορφές που προκύπτουν και τα περιοδογράμμά τους. Τι παρατηρείτε;

Λύση:

Προκύπτουν οι κυματομορφές:







Ακολουθεί ο κώδικας MatLab:

```

1 %% A.9
2
3 % Demodulating the signal
4 XImod_noise = Xmod_noise.*cos(2*pi*Fo*conv_t);
5 XQmod_noise = -Xmod_noise.*sin(2*pi*Fo*conv_t);
6
7 % Plot the signals
8 figure;
9 plot(conv_t, XImod_noise);
10 title("XI(t)cos(2piFot)")
11 xlabel("t");
12 ylabel("XI(t)");
13 xlim([conv_t(1) conv_t(end)]);
14
15 figure;
16 plot(conv_t, XQmod_noise);
17 title("-XQ(t)sin(2piFot)")
18 xlabel("t");
19 ylabel("XQ(t)");
20 xlim([conv_t(1) conv_t(end)]);
21

```

```

22 % Periodograms creation
23 XImod_noise_F = fftshift(fft(XImod_noise, Nf)*Ts);
24 num = power(abs(XImod_noise_F),2);
25 T_total = conv_t(end) - conv_t(1);
26 PXImod_noise = num/T_total;
27
28 XQmod_noise_F = fftshift(fft(XQmod_noise, Nf)*Ts);
29 num = power(abs(XQmod_noise_F),2);
30 T_total = conv_t(end) - conv_t(1);
31 PXQmod_noise = num/T_total;
32
33 % Plot the periodogram
34 figure;
35 plot(f_axis, PXImod_noise);
36 title("Periodogram of XI(t)")
37 xlabel("f");
38 ylabel("Px(F)");
39
40 figure;
41 plot(f_axis, PXQmod_noise);
42 title("Periodogram of XQ(t)")
43 xlabel("f");
44 ylabel("Px(F)");

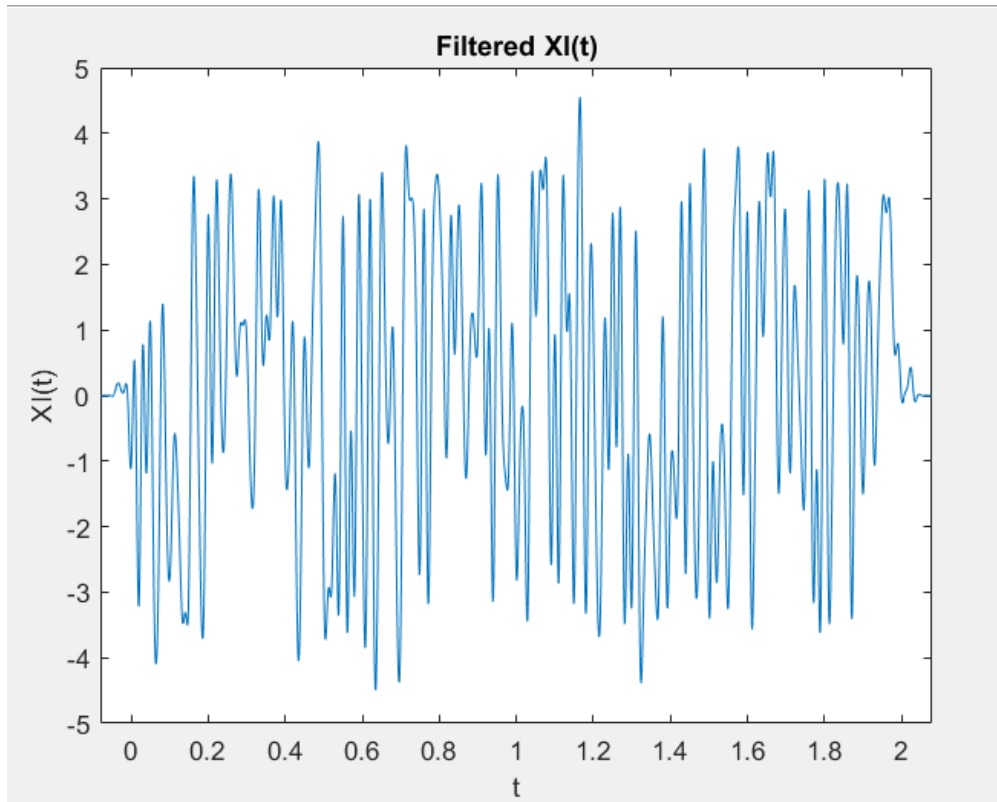
```

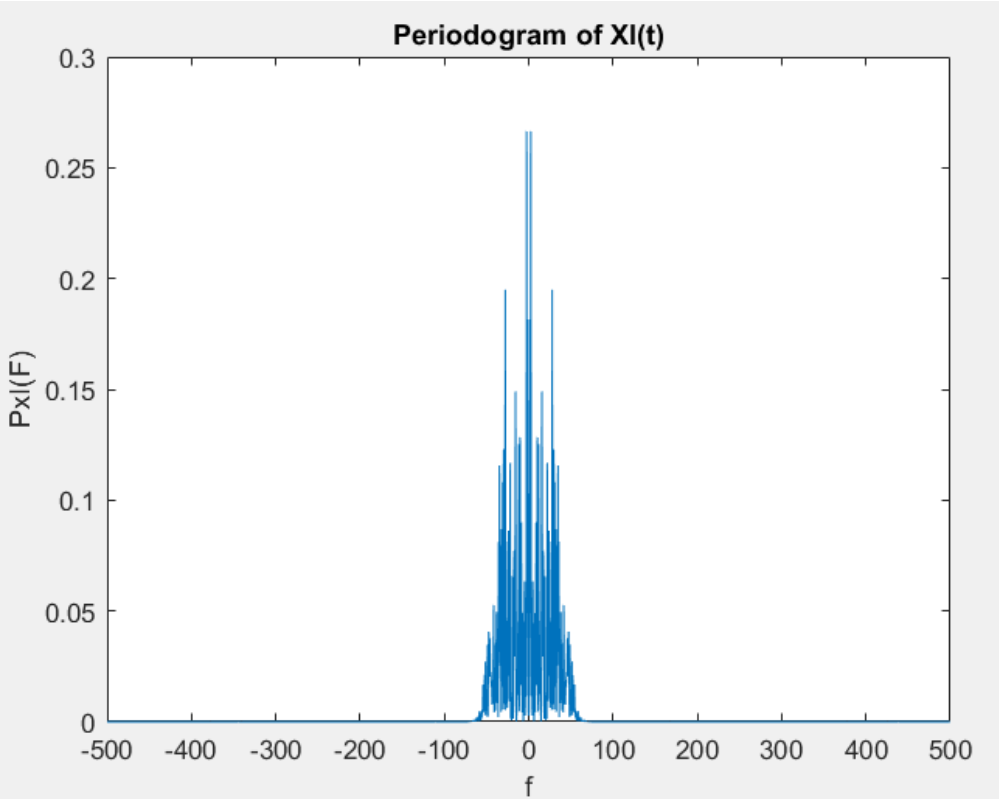
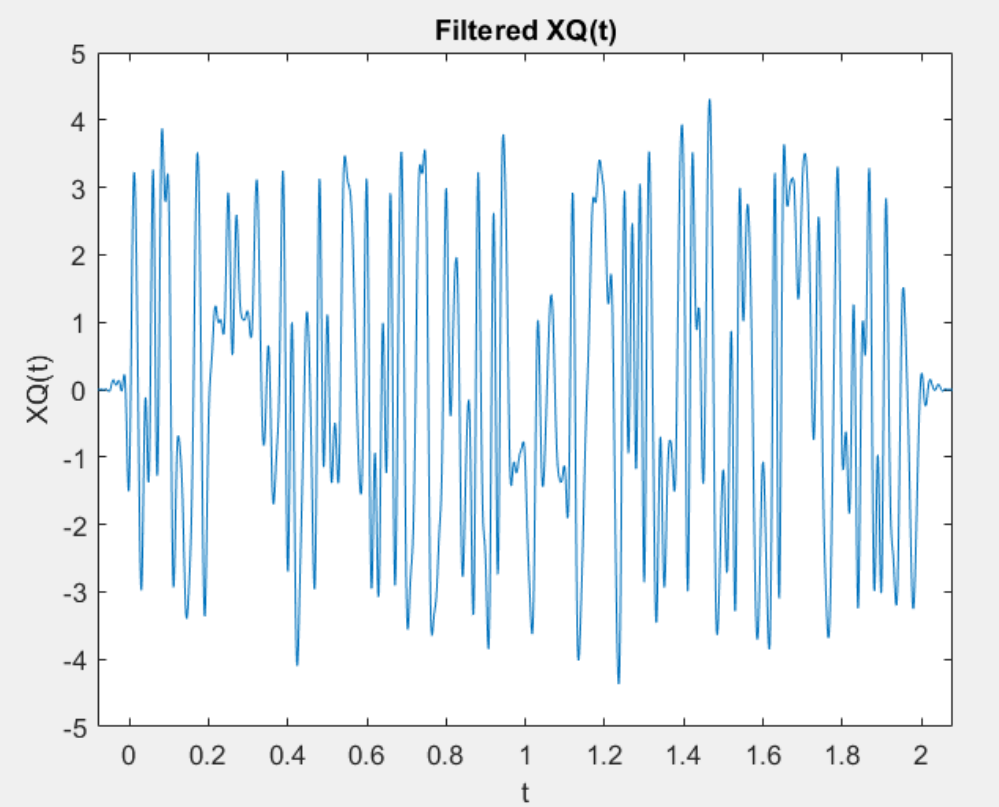
Παρατηρούμε ότι τα περιοδογράμματα έχουν κέντρο τις συχνότητες $-F_0, 0, F_0$.

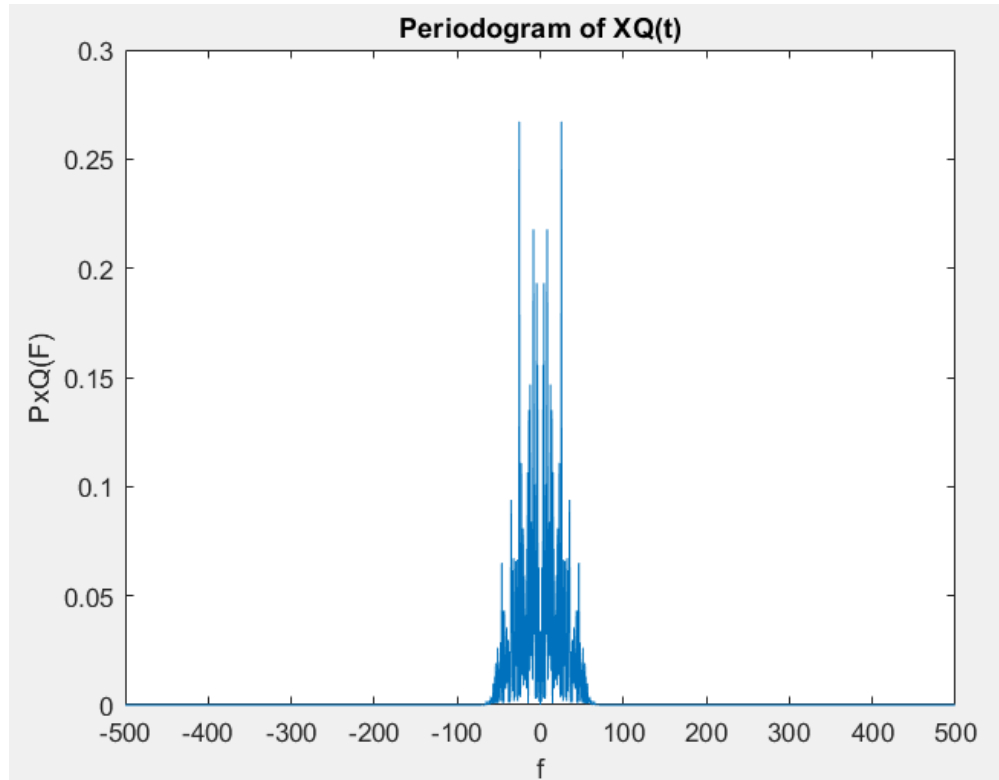
10. (5) Να περάσετε τις κυματομορφές που υπολογίσατε στο προηγούμενο βήμα από τα προσαρμοσμένα φίλτρα. Να σχεδιάσετε τις κυματομορφές που προκύπτουν και τα περιοδογράμματά τους. Τι παρατηρείτε;

Λύση:

Περνάμε τις κυματομορφές από φίλτρα και προκύπτουν:







Και ο κώδικας MatLab:

```

1 %% A.10
2
3 % Filter the waveforms
4 XIt_demodulated = Ts*conv(XImod_noise, phi);
5 XQt_demodulated = Ts*conv(XQmod_noise, phi);
6 conv_t2 = t(1) + conv_t(1):Ts:t(end) + conv_t(end);
7
8 % Plot the filtered waveforms
9 figure;
10 plot(conv_t2, XIt_demodulated);
11 title("Filtered XI(t)")
12 xlabel("t");
13 ylabel("XI(t)");
14 xlim([conv_t2(1) conv_t2(end)]);
15
16 figure;
17 plot(conv_t2, XQt_demodulated);
18 title("Filtered XQ(t)")
19 xlabel("t");
20 ylabel("XQ(t)");
21 xlim([conv_t2(1) conv_t2(end)]);
22

```

```

23 XIF_demodulated = fftshift(fft(XIt_demodulated, Nf)*Ts);
24 num = power(abs(XIF_demodulated),2);
25 T_total = conv_t2(end) - conv_t2(1);
26 PXI_demodulated = num/T_total;
27
28 XQF_demodulated = fftshift(fft(XQt_demodulated, Nf)*Ts);
29 num = power(abs(XQF_demodulated),2);
30 T_total = conv_t2(end) - conv_t2(1);
31 PXQ_demodulated = num/T_total;
32
33 % Plot the periodogram of filtered waveforms
34 figure;
35 plot(f_axis, PXI_demodulated);
36 title("Periodogram of XI(t)")
37 xlabel("f");
38 ylabel("PxI(F)");
39
40 figure;
41 plot(f_axis, PXQ_demodulated);
42 title("Periodogram of XQ(t)")
43 xlabel("f");
44 ylabel("PxQ(F)");

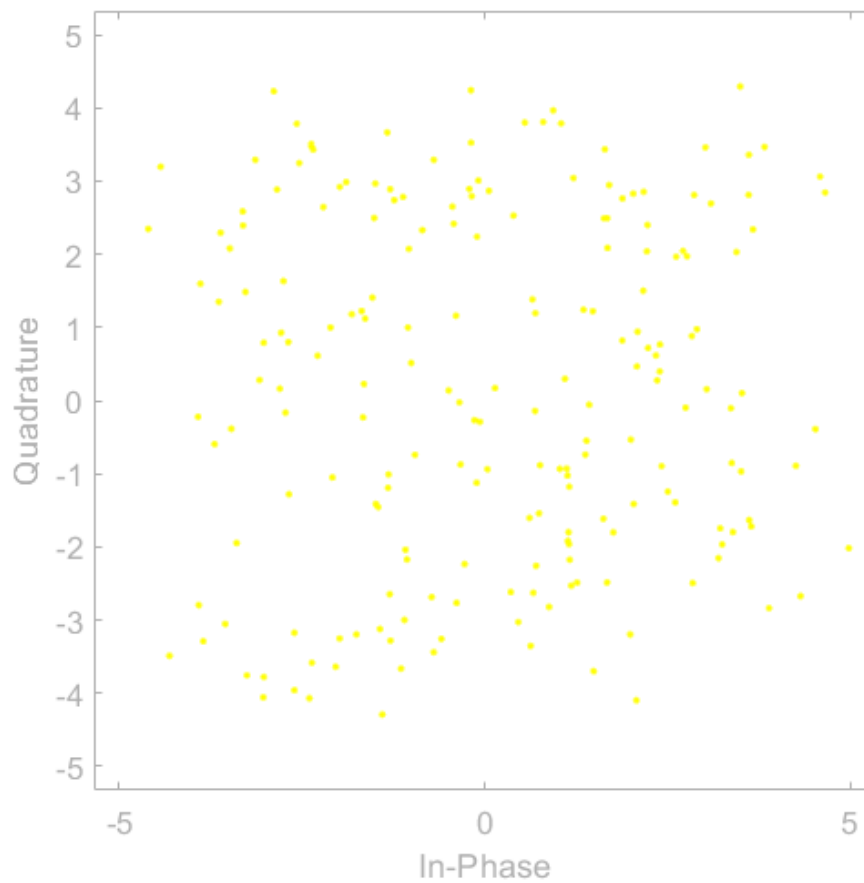
```

Όπως παρατηρούμε, μετά το φιλτράρισμα, έχουν αποκοπεί οι όροι στις μεγάλες συχνότητες.

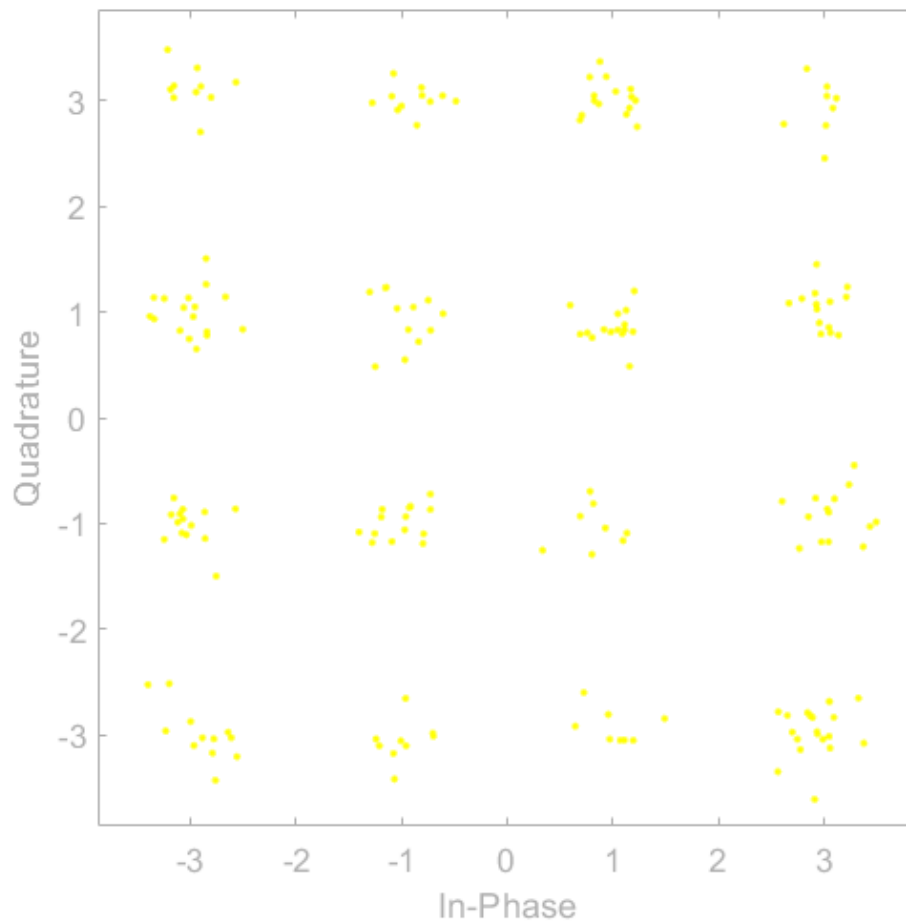
11. (5) Να δειγματοληπτήσετε την έξοδο των προσαρμοσμένων φίλτρων τις κατάλληλες χρονικές στιγμές και να σχεδιάσετε την ακολουθία εξόδου χρησιμοποιώντας την εντολή scatterplot.

Λύση:

Για SNR=10:



Και για SNR=20:



Και ο κώδικας MatLab:

```
1 %% A.11
2
3 XIt_demodulated_sampling= XIt_demodulated((2*A_SRRC*T/Ts)+1:over:
4     length(XIt_demodulated)-(2*A_SRRC*T/Ts));
5
6 XQt_demodulated_sampling= XQt_demodulated((2*A_SRRC*T/Ts)+1:over:
7     length(XQt_demodulated)-(2*A_SRRC*T/Ts));
8
9 for i=1:N
10     sampling(i,1)=XIt_demodulated_sampling(i);
11     sampling(i,2)=XQt_demodulated_sampling(i);
12 end
13 scatterplot(sampling)
```


12. (10) Να γράψετε την συνάρτηση:

$$est_X = detect_4_PAM(Y, A)$$

η οποία χρησιμοποιεί τον κανόνα των εγγύτερων γείτονα και αποφασίζει για την ακολουθία εισόδου 4-PAM σύμβολο προς σύμβολο. Να εφαρμόσετε τη συνάρτηση ξεχωριστά στα δείγματα που λάβατε στην inphase έξοδο και στην quadrature έξοδο και να λάβετε αποφάσεις για τις αντίστοιχες ακολουθίες εισόδου.

Λύση:

Ο κώδικας MatLab για την συνάρτηση detect_4_PAM:

```
1 function [est_x] = detect_4_PAM(Y,A)
2     Y_size = size(Y,2);
3     est_x = zeros(1,Y_size);
4     x = zeros(1,4);
5
6     symbols = [-3*A, -A, A, 3*A];
7     for i = 1:Y_size
8         distances = zeros(1,4);
9         for j = 1:4
10            distances(j) = power((Y(i)-symbols(j)),2);
11        end
12        [~, minIndex] = min(distances);
13        est_x(i) = symbols(minIndex);
14    end
15 end
```

Και έπειτα την εφαρμόζουμε στις δειγματοληπτημένες εξόδους του προηγούμενου ερωτήματος:

```
1 %% A.12
2
3 % Estimating the symbols value
4 for i = 1:N
5     detected_XI(i) = detect_4_PAM(XIt_demodulated_sampling(i), A);
6     detected_XQ(i) = detect_4_PAM(XQt_demodulated_sampling(i), A);
7 end
```

13. (10) Χρησιμοποιώντας τις ακολουθίες εισόδου και τις αποφάσεις, να υπολογίσετε τον αριθμό σφαλμάτων απόφασης συμβόλου για τον αστερισμό 16-QAM.

Λύση:

Για να υπολογιστεί ο αριθμός σφαλμάτων απόφασης συμβόλου, τοποθετήθηκαν σε πίνακες οι αρχικές ακολουθίες εισόδου και οι αποφάσεις. Τέλος συγκρίνεται η διαφορά τους και αν αυτή είναι διαφορετική του μηδενός, υπάρχει σφάλμα απόφασης. Για SNR=10 το σφάλμα είναι **50** ενώ για SNR=20 το σφάλμα είναι **0**. Αυτό είναι λογικό αφού για μεγαλύτερο SNR λαμβάνουμε περισσότερη πληροφορία.

Ο κώδικας MatLab για τον υπολογισμό:

```
1 %% A.13
2
3 x_sent = [XIn; XQn];
4 x_detected = [detected_XI; detected_XQ];
5
6 errors_num = 0;
7 for i = 1:N
8     for j = 1:2
9         difference(j,i) = x_sent(j,i) - x_detected(j,i);
10        if (difference(j,i) ~= 0)
11            errors_num = errors_num + 1;
12        end
13    end
14 end
15 errors_num
```

14. (10) Να γράψετε την συνάρτηση:

$$est_bit = PAM_4_to_bits(X, A)$$

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.

Λύση:

Κατασκευάστηκε η συνάρτηση PAM_4_to_bits:

```
1 function [est_bit] = PAM_4_to_bits(X,A)
2
3 for i = 1:length(X)
4     if X(i) == 3*A
5         est_bit(i:i+1) = [0; 0];
```

```

6         elseif X(i) == A
7             est_bit(i:i+1) = [0; 1];
8         elseif X(i) == -A
9             est_bit(i:i+1) = [1; 1];
10        elseif X(i) == -3*A
11            est_bit(i:i+1) = [1; 0];
12        end
13    end
14 end

```

15. (10) Να υπολογίσετε τον αριθμό σφαλμάτων bit.

Λύση:

Για να υπολογιστεί ο αριθμός σφαλμάτων bit, υπολογίστηκε η εκτιμώμενη ακολουθία εισόδου και συγκρίθηκε με την αρχική δυαδική ακολουθία. Για SNR=10 το σφάλμα είναι **394** ενώ για SNR=20 το σφάλμα είναι **386**.

```

1 %% A.15
2
3 bitXI_received = PAM_4_to_bits(detected_XI, A);
4 bitXQ_received = PAM_4_to_bits(detected_XQ, A);
5
6 estimated_bits = [bitXI_received bitXQ_received];
7 difference = estimated_bits - bits;
8 errors_in_bits = 0;
9 for i = 1:4*N
10     if (difference(i) ~= 0)
11         errors_in_bits = errors_in_bits + 1;
12     end
13 end
14 errors_in_bits

```

Αν έχετε υλοποιήσει τα παραπάνω βήματα, έχετε καταλάβει πως λειτουργεί ένα σύστημα το οποίο χρησιμοποιεί διαμόρφωση 16-QAM (φυσικά, έχουμε υποθέσει ότι το κανάλι είναι ιδανικό και ότι είμαστε τέλεια συγχρονισμένοι).

Στο δεύτερο μέρος, θα εκτιμήσετε την πιθανότητα σφάλματος συμβόλου και bit με χρήση της μεθόδου Monte Carlo.

1. Για $\text{SNR}_{\text{dB}} = [0 : 2 : 16]$, να υπολογίσετε πειραματικά την πιθανότητα σφάλματος απόφασης συμβόλου και bit επαναλαμβάνοντας τα παραπάνω βήματα K φορές (ενδεικτικά, $K = 200, 1000$) για κάθε SNR.

Λύση:

Για τα διάφορα SNR_{dB} και για $K = 200$ επαβαλήψεις, δημιουργήσαμε ακολουθίες από bit και ακολουθώντας τα βήματα του πρώτου μέρους, δημιουργήσαμε το σήμα εξόδου. Ακολουθεί ο κώδικας για τον υπολογισμό της πειραματικής πιθανότητας σφάλματος απόφασης συμβόλου:

```
1 %% Variable decleration
2 N = 200;
3 A = 1;
4 A_SRRC = 4;
5 a = 0.5;
6 T = 0.01;
7 over = 10;
8 Ts = T/over;
9 Fs = 1/Ts;
10 Nf = 2048;
11 %K = 1000;
12 K = 200;
13
14 SNRdb = 0:2:16;
15 %% B.1
16 for i = 1:length(SNRdb)
17     SNR = SNRdb(i);
18     sum_errors_num = 0;
19     sum_errors_in_bits = 0;
20     for k = 1:K
21         bits = (sign(randn(4*N, 1)) + 1)/2;
22
23         % For bits in the [0,2N]
24         bitXI = bits(1:2*N);
25         XIn = bits_to_4_PAM(bitXI, A);
26
27         % For bits in the [2N+1,4N]
28         bitXQ = bits(2*N+1:4*N);
```

```

29     XQn = bits_to_4_PAM(bitXQ, A);
30
31     % Creating the pulse
32     [phi, t] = srrc_pulse(T, over, A_SRRC, a);
33
34     % Upsample the 2 sequencies
35     XIn_up = (1/Ts)*upsample(XIn, over);
36     XI_up_time = 0:Ts:N*T-Ts;
37
38     XQn_up = (1/Ts)*upsample(XQn, over);
39     XQ_up_time = XI_up_time;
40
41     % Convolutions and time
42     XIt = Ts*conv(XIn_up, phi);
43     XQt = Ts*conv(XQn_up, phi);
44     conv_t = t(1) + XQ_up_time(1):Ts:t(end) + XQ_up_time(end);
45
46     % Frequency Range
47     f_axis = linspace(-Fs/2,(Fs/2-Fs/Nf), Nf);
48
49     % Periodograms creation
50     XIF = fftshift(fft(XIt, Nf)*Ts);
51     num = power(abs(XIF),2);
52     T_total = conv_t(end) - conv_t(1);
53     PXIF = num/T_total;
54
55     XQF = fftshift(fft(XQt, Nf)*Ts);
56     num = power(abs(XQF),2);
57     T_total = conv_t(end) - conv_t(1);
58     PXQF = num/T_total;
59
60     Fo = 200;
61     XImod_t = 2*XIt.*cos(2*pi*Fo*conv_t);
62     XQmod_t = -2*XQt.*sin(2*pi*Fo*conv_t);
63
64     % Periodograms creation
65     XIFmod = fftshift(fft(XImod_t, Nf)*Ts);
66     num = power(abs(XIFmod),2);
67     T_total = conv_t(end) - conv_t(1);
68     PXIFmod = num/T_total;
69
70     XQFmod = fftshift(fft(XQmod_t, Nf)*Ts);
71     num = power(abs(XQFmod),2);

```

```

72     T_total = conv_t(end) - conv_t(1);
73     PXQFmod = num/T_total;
74
75     % Input signal creation
76     Xmod_t = XImod_t + XQmod_t;
77
78     % Periodogram creation
79     Xmod_F = fftshift(fft(Xmod_t, Nf)*Ts);
80     num = power(abs(Xmod_F),2);
81     T_total = conv_t(end) - conv_t(1);
82     PXmod = num/T_total;
83
84     % Noise creation
85     variance = (10*(A^2))/(Ts*(10^(SNR/10)));
86     gaussian_noise = sqrt(variance)*randn(1,length(Xmod_t));
87
88     % Adding noise to the waveform
89     Xmod_noise = Xmod_t + gaussian_noise;
90
91     % Demodulating the signal
92     XImod_noise = Xmod_noise.*cos(2*pi*Fo*conv_t);
93     XQmod_noise = -Xmod_noise.*sin(2*pi*Fo*conv_t);
94
95     % Periodograms creation
96     XImod_noise_F = fftshift(fft(XImod_noise, Nf)*Ts);
97     num = power(abs(XImod_noise_F),2);
98     T_total = conv_t(end) - conv_t(1);
99     PXImod_noise = num/T_total;
100
101     XQmod_noise_F = fftshift(fft(XQmod_noise, Nf)*Ts);
102     num = power(abs(XQmod_noise_F),2);
103     T_total = conv_t(end) - conv_t(1);
104     PXQmod_noise = num/T_total;
105
106     % Filter the waveforms
107     XIt_demodulated = Ts*conv(XImod_noise, phi);
108     XQt_demodulated = Ts*conv(XQmod_noise, phi);
109     conv_t2 = t(1) + conv_t(1):Ts:t(end) + conv_t(end);
110
111     XIF_demodulated = fftshift(fft(XIt_demodulated, Nf)*Ts);
112     num = power(abs(XIF_demodulated),2);
113     T_total = conv_t2(end) - conv_t2(1);
114     PXI_demodulated = num/T_total;

```

```

115
116 XQF_demodulated = fftshift(fft(XQt_demodulated, Nf)*Ts);
117 num = power(abs(XQF_demodulated),2);
118 T_total = conv_t2(end) - conv_t2(1);
119 PXQ_demodulated = num/T_total;
120
121 XIt_demodulated_sampling= XIt_demodulated(
122     (2*A_SRRC*T/Ts)+1:over:length(XIt_demodulated)
123     -(2*A_SRRC*T/Ts));
124
125 XQt_demodulated_sampling= XQt_demodulated(
126     (2*A_SRRC*T/Ts)+1:over:length(XQt_demodulated)
127     -(2*A_SRRC*T/Ts));
128
129 for p=1:N
130     sampling(p,1)=XIt_demodulated_sampling(p);
131     sampling(p,2)=XQt_demodulated_sampling(p);
132 end
133
134 % Estimating the symbols value
135 for j = 1:N
136     detected_XI(j) =
137         detect_4_PAM(XIt_demodulated_sampling(j), A);
138     detected_XQ(j) =
139         detect_4_PAM(XQt_demodulated_sampling(j), A);
140 end
141
142 x_sent = [XIn; XQn];
143 x_detected = [detected_XI; detected_XQ];
144
145 errors_num = 0;
146 for n = 1:N
147     for j = 1:2
148         if (x_sent(j,n) ~= x_detected(j,n))
149             errors_num = errors_num + 1;
150         end
151     end
152 end
153
154 bitXI_received = PAM_4_to_bits(detected_XI, A);
155 bitXQ_received = PAM_4_to_bits(detected_XQ, A);
156
157 estimated_bits = [bitXI_received bitXQ_received];

```

```

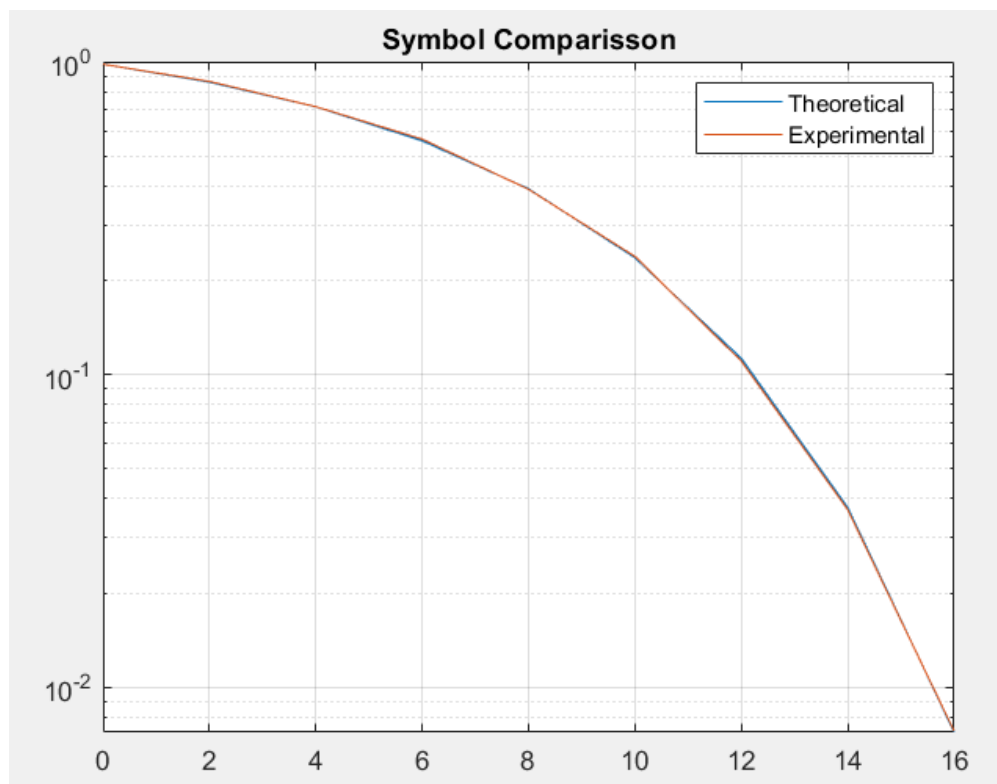
156
157     errors_in_bits = 0;
158     for m = 1:length(estimated_bits)
159         if (estimated_bits(m) ~= bits(m))
160             errors_in_bits = errors_in_bits + 1;
161         end
162     end
163     sum_errors_num = sum_errors_num + errors_num;
164     sum_errors_in_bits = sum_errors_in_bits + errors_in_bits;
165 end
166 P_symbol(i) = sum_errors_num/(N*K);
167 P_bit(i) = sum_errors_in_bits/(4*N*K);
168 P_theor_symbol(i) = 3*Q(sqrt(0.2.*(10.^(SNR/10))));
169 P_theor_bit(i) = P_theor_symbol(i)/4;
170 end

```

2. Να σχεδιάσετε σε semilogy τη θεωρητική και πειραματική πιθανότητα σφάλματος συμβόλου σαν συνάρτηση του SNR_{dB} . Τι παρατηρείτε;

Λύση:

Χρησιμοποιώντας την συνάρτηση Q όπως φαίνεται και από των παραπάνω κώδικα, σχεδιάσαμε σε κοινό semilogy τη θεωρητική και πειραματική πιθανότητα σφάλματος συμβόλου.

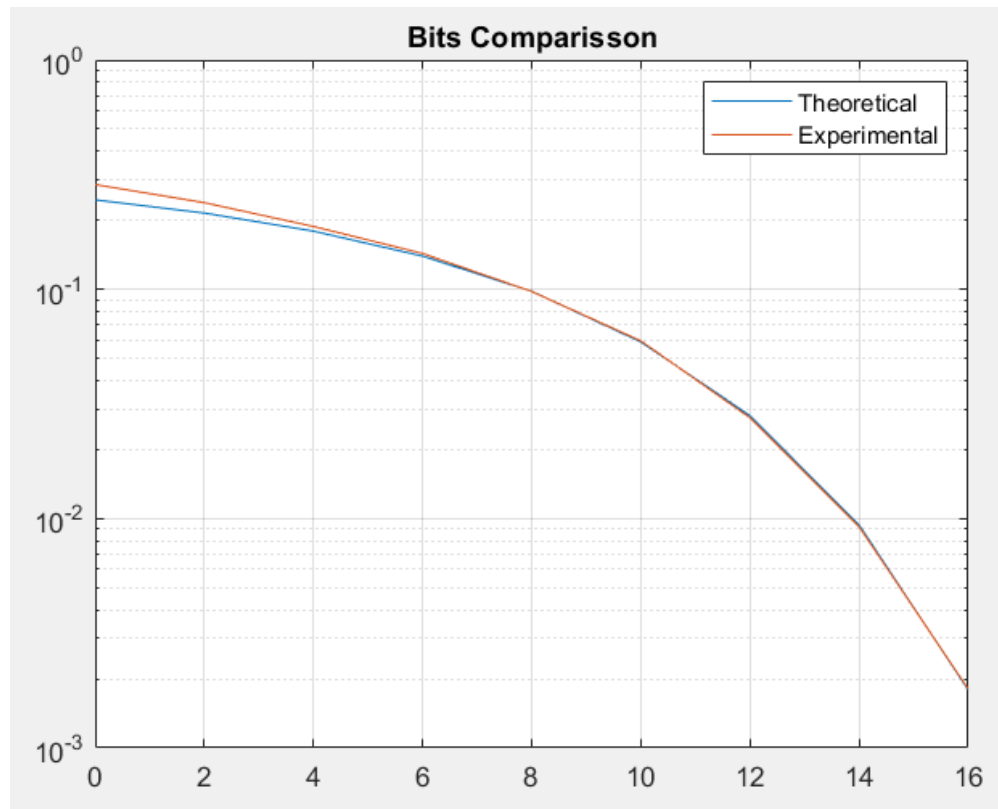


Παρατηρείται ότι οι πειραματικές τιμές προσεγγίζουν τις θεωρητικές με πάρα πολύ καλή ακρίβεια.

3. Να σχεδιάσετε σε semilogy την πειραματική πιθανότητα σφάλματος bit σαν συνάρτηση του SNR_{dB} . Στο ίδιο σχήμα, να σχεδιάσετε και τη θεωρητική προσέγγιση για την πιθανότητα σφάλματος bit. Τι παρατηρείτε; Μπορείτε να εξηγήσετε το φαινόμενο;

Λύση:

Όμοια με τα παραπάνω:



Παρατηρούμε για μικρές τιμές του SNR, το πειραματικό Rate ξεπερνά το θεωρητικό όριο. Αυτό συμβάνει μιας και ο τύπος που χρησιμοποιήσαμε επαληθεύεται για μεγάλες τιμές SNR.

Και ο κώδικας MatLab για τα ερωτήματα **2** και **3**:

```
1 %% B.2 and B.3
2 figure;
3 semilogy(SNRdb, P_theor_symbol);
4 hold on;
5 semilogy(SNRdb, P_symbol);
6 legend("Theoretical","Experimental")
7 hold off;
8 grid on;
9 title("Symbol Comparisson");
```

```
10  
11 figure;  
12 semilogy(SNRdb, P_theor_bit);  
13 hold on;  
14 semilogy(SNRdb, P_bit);  
15 legend ("Theoretical","Experimental")  
16 hold off;  
17 grid on;  
18 title("Bits Comparisson");
```