

Τηλεπικοινωνιακά Συστήματα Ι

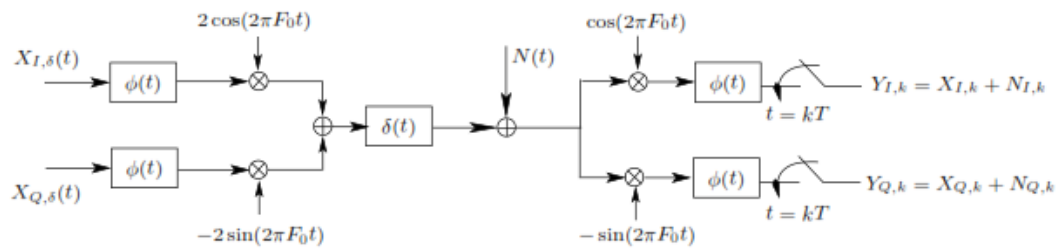
Άσκηση 3 : Σύστημα 16 - QAM

Διδάσκων: Αθανάσιος Π. Λιάβας

Όνομα : Χρήστος Ζαχαριουδάκης

Αριθμός Μητρώου : 2014030056

A



Σε αυτή την άσκηση, θα προσομοιώσουμε το τηλεπικοινωνιακό σύστημα του Σχήματος, υποθέτοντας ότι χρησιμοποιείται διαμόρφωση 16-QAM, και θα μελετήσουμε την απόδοσή του. Πιο συγκεκριμένα, θα υλοποιήσουμε το 16-QAM σύστημα σαν δύο ανεξάρτητα 4-PAM συστήματα, με αστερισμό $\{\pm A, \pm 3A\}$.

A.1.

Για δεδομένο N (ενδεικτικά, $N = 100, 500$), να δημιουργήσετε δυαδική ακολουθία με στοιχεία $4N$ ισοπίθανα bits.

Κώδικας :

```
N = 500 ;  
bit_seq = (sign(randn(4*N,1))+1)/2;
```

Σχόλια :

Αρχικά , για $N = 500$ δημιουργούμε δυαδική ακολουθία $4*N$ (2000 bits) ισοπίθανων bits .

A.2.

(15) Να γράψετε συνάρτηση

```
function X = bits_to_4_PAM(bit_seq, A)
```

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

Κώδικας :

```
A = 1;  
X = bits_to_4_PAM(bit_seq, A);
```

Σχόλια :

Στην συνέχεια , επιλέγοντας $A = 1$, επομένως αστερισμό $\{\pm 1, \pm 3\}$, υλοποιήθηκε η συνάρτηση `bits_to_4_PAM` , η οποία λαμβάνει ως όρισμα το διάνυσμα της δυαδικής ακολουθίας και το A , και επιστρέφει χρησιμοποιώντας την παρακάτω κωδικοποίηση Gray (τα κοντινά σύμβολα έχουν διαφορά 1 bit) :

00 -> - 3

01 -> - 1

11 -> +1

10 -> +3

μια ακολουθία 4-PAM συμβόλων μήκους $2*N$ (1000 σύμβολα) .

Η συνάρτηση λειτουργεί ως εξής :

- Με χρήση ενός βρόγχου `for` , διαβάζονται ανά δύο τα bits της ακολουθίας και ανάλογα με τις τιμές τους , αποθηκεύεται σε ένα νέο διάνυσμα μήκους $2*N$, το αντίστοιχο σύμβολο σύμφωνα με την παραπάνω αναπαράσταση .

Ο κώδικας της συνάρτησης παρατίθεται στο τέλος της αναφοράς .

A.3.

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

Κώδικας :

```
XI = X(1:N);  
XQ = X(N+1:end);
```

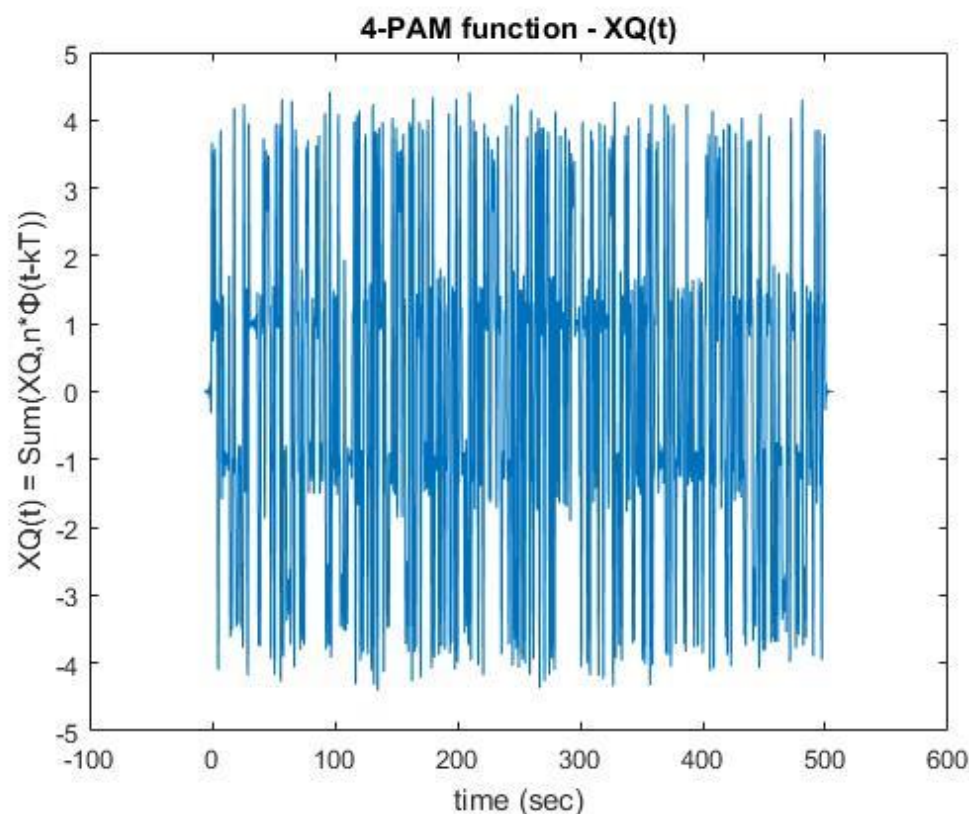
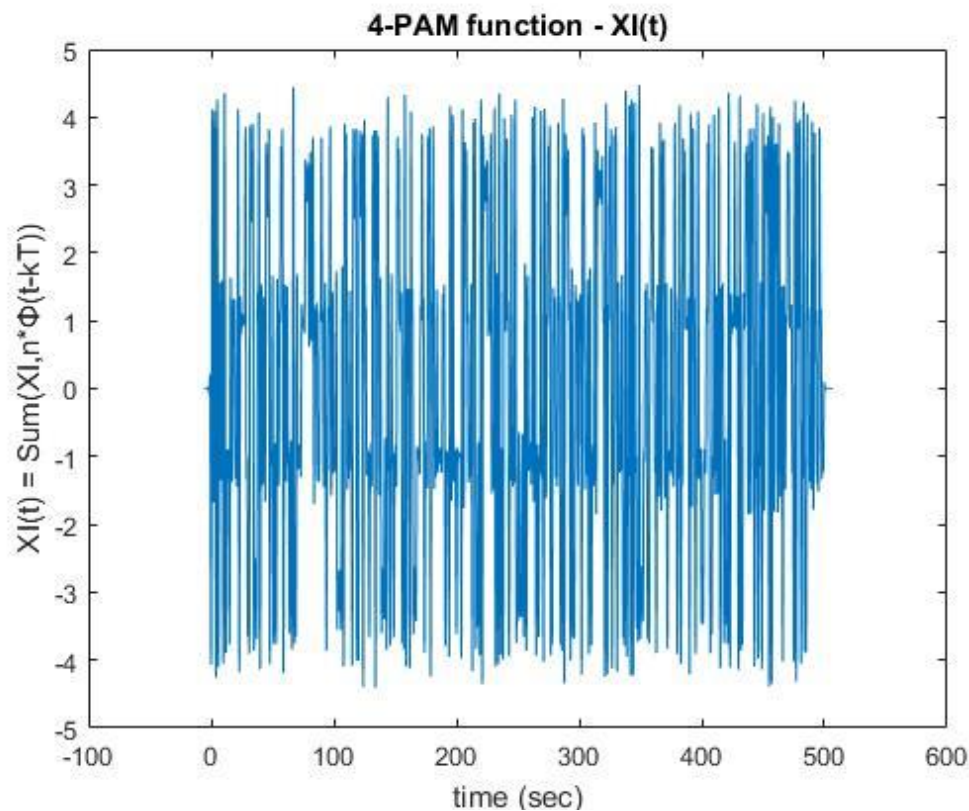
Σχόλια :

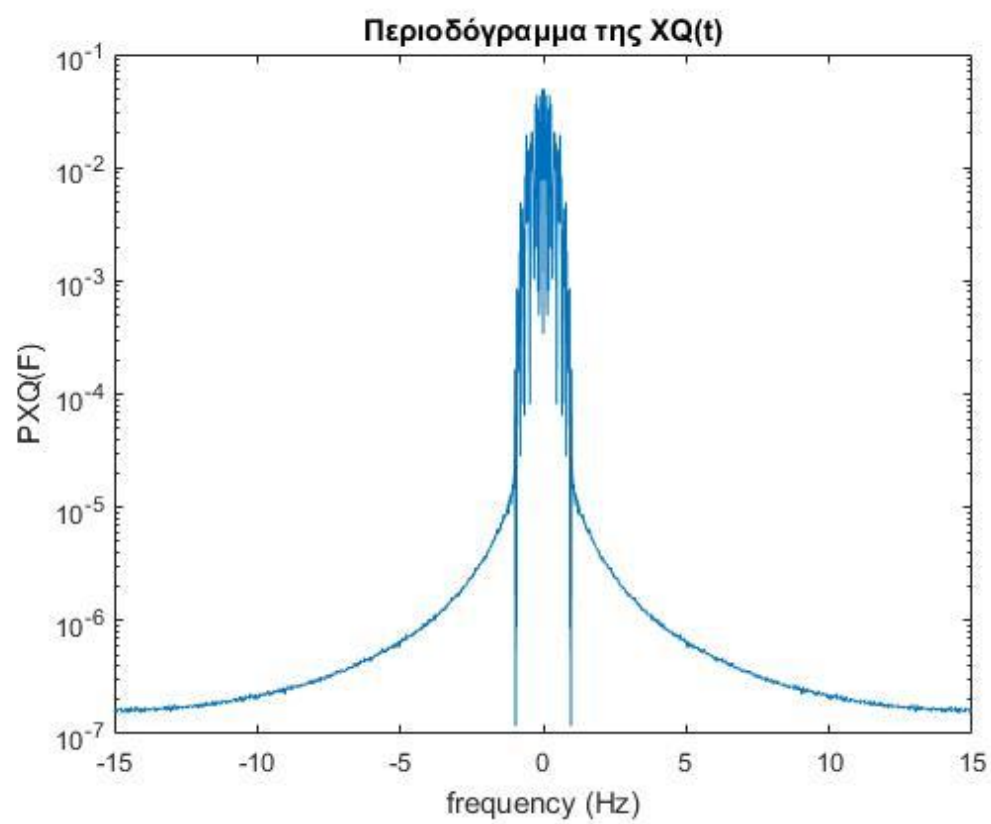
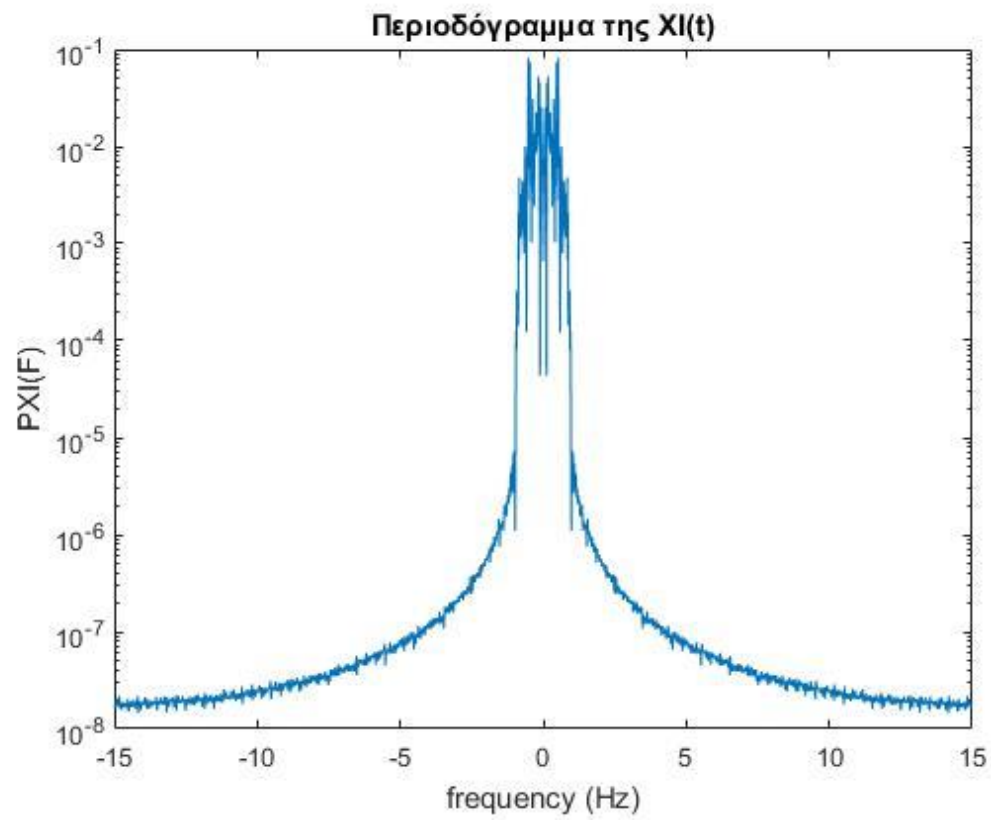
Στην συνέχεια , χωρίζουμε το διάνυσμα συμβόλων X σε δύο επιμέρους διανύσματα μήκους N , τα οποία αντιπροσωπεύουν τις ακολουθίες $\{X_{I,n}\}$ και $\{X_{Q,n}\}$ του συστήματός μας , για $n = 1, \dots, N$

A.4.

(5) Να περάσετε τις ακολουθίες $\{X_{I,n}\}$ και $\{X_{Q,n}\}$ από τα SRRC φίλτρα μορφοποίησης και υποθέτοντας, ενδεικτικά, περίοδο συμβόλου $T = 1$ sec, $\text{over} = 10$, $T_s = \frac{T}{\text{over}}$, να σχηματίσετε και να σχεδιάσετε τις κυματομορφές εξόδου $X_I(t)$ και $X_Q(t)$ (να θέσετε το σωστό άξονα χρόνου), και τα περιοδογράμματά τους.

Γραφικές Παραστάσεις :





Κώδικας :

```
T = 1 ;
over = 30 ;
Ts = T/over ; % Περίοδος δειγματοληψίας
Fs = 1/Ts ; % Συχνότητα δειγματοληψίας

A_phi = 6 ;
roll_off = 1 ;

[phi,t_phi] = srrc_pulse(T,Ts,A_phi,roll_off);

[tx,XIt] = PAM4_function (XI,Ts,over,phi,t_phi) ;
[tx,XQt] = PAM4_function (XQ,Ts,over,phi,t_phi) ;

figure(1);
plot (tx,XIt) ;
title('4-PAM function - XI(t)');
xlabel('time (sec)');
ylabel('XI(t) = Sum(XI,n*Φ(t-kT))');

figure(2)
plot (tx,XQt) ;
title('4-PAM function - XQ(t)');
xlabel('time (sec)');
ylabel('XQ(t) = Sum(XQ,n*Φ(t-kT))');

Nf = 2048 ; % Αριθμός δειγμάτων στο πεδίο της συχνότητας
f_axis = [-1/2 : 1/Nf : 1/2 - (1/Nf)]; % Ανοικτό διάστημα δεξιά ,
κλειστό αριστερά
F_axis = f_axis*Fs;

Ttotal = length(tx);
XIF = fftshift(fft(XIt,Nf)*Ts);
PXIF = power(abs(XIF),2)/Ttotal ;
figure(3)
semilogy(F_axis,PXIF)
title(['Περιοδόγραμμα της XI(t)'])
xlabel('frequency (Hz)');
ylabel('PXI(F)');

XQF = fftshift(fft(XQt,Nf)*Ts);
PXQF = power(abs(XQF),2)/Ttotal ;
figure(4)
semilogy(F_axis,PXQF)
title(['Περιοδόγραμμα της XQ(t)'])
xlabel('frequency (Hz)');
ylabel('PXQ(F)');
```

Σχόλια :

Στο ερώτημα αυτό , περνάμε τις δύο ακολουθίες συμβόλων $\{X_{I,n}\}$ και $\{X_{Q,n}\}$ από SRRC φίλτρα μορφοποίησης .

Για την δημιουργία της συνάρτησης $\Phi(t)$, χρησιμοποιούμε την συνάρτηση **srrec_pulse** που μας έχει δοθεί , με παραμέτρους την περίοδο T ίση με 1 , $over = 30$, περίοδο δειγματοληψίας $T_s = T/over = 0.1$, $A = 6$ και roll off factor = 1 . Η συνάρτηση επιστρέφει το διάνυσμα της συνάρτησης $\Phi(t)$ και το αντίστοιχο διάνυσμα χρόνου .

Για να πραγματοποιηθεί το φιλτράρισμα , πρέπει η κάθε ακολουθία να μετατραπεί σε ένα τρένο παλμών Dirac ($X_{\delta(t)}$) , με το οποίο θα γίνει συνέλιξη με την $\Phi(t)$. Επιπλέον, πρέπει να παραχθεί το νέο διάνυσμα χρόνου , το οποίο λόγω της συνέλιξης θα έχει ως ελάχιστη τιμή το άθροισμα των ελαχίστων τιμών του χρόνου της $\Phi(t)$ και της $X_{\delta(t)}$ και μέγιστη τιμή το άθροισμα των μέγιστων τιμών του χρόνου της $\Phi(t)$ και της $X_{\delta(t)}$. Τα παραπάνω επιτυγχάνονται με χρήση της συνάρτησης PAM4_function με ορίσματα την ακολουθία συμβόλων X , T_s , $over$ και τα διανύσματα τιμών και χρόνου της $\Phi(t)$ και επιστρέφει την διαμορφωμένη 4-PAM συνάρτηση και το αντίστοιχο διάνυσμα χρόνου . Ο κώδικας περιλαμβάνεται στο τέλος της αναφοράς.

Τέλος , ζητείται να σχεδιαστούν οι δύο συναρτήσεις 4-PAM και τα αντίστοιχα περιοδογραμμάτα τους . Οι συναρτήσεις σχεδιάζονται με την συνάρτηση **plot** . Για τον σχεδιασμό των περιοδογραμμάτων πρέπει πρώτα να βρεθεί το φάσμα των συναρτήσεων. Βρέθηκε ο μετασχηματισμός Fourier κάθε παλμού με την χρήση των συναρτήσεων **fft** και **fftshift** της MATLAB , σε N_f ισαπέχοντα διαστήματα στο διάστημα $[-F_s/2, F_s/2)$. Επιλέχθηκε η τιμή $N_f = 2048$ δείγματα , ώστε να αποφευχθούν παραμορφώσεις για τις τιμές $over$ και N που επιλέξαμε . Στην συνέχεια υπολογίστηκαν τα περιοδογράμματα με χρήση του ορισμού :

$$P_X(F) = \frac{|\mathcal{F}[X(t)]|^2}{T_{total}}$$

όπου T_{total} ο συνολικός χρόνος διάρκειας της $X(t)$. Τα δύο περιοδογράμματα σχεδιάστηκαν σε ημιλογαριθμική κλίμακα με τη συνάρτηση **semilogy** .

A.5.

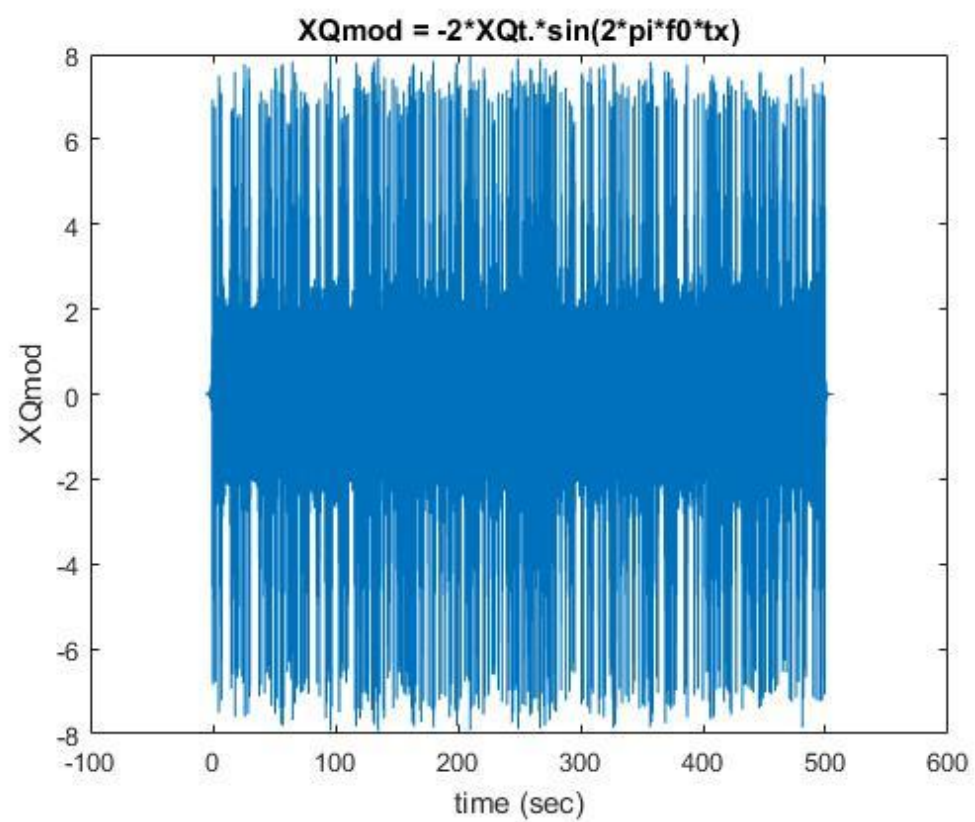
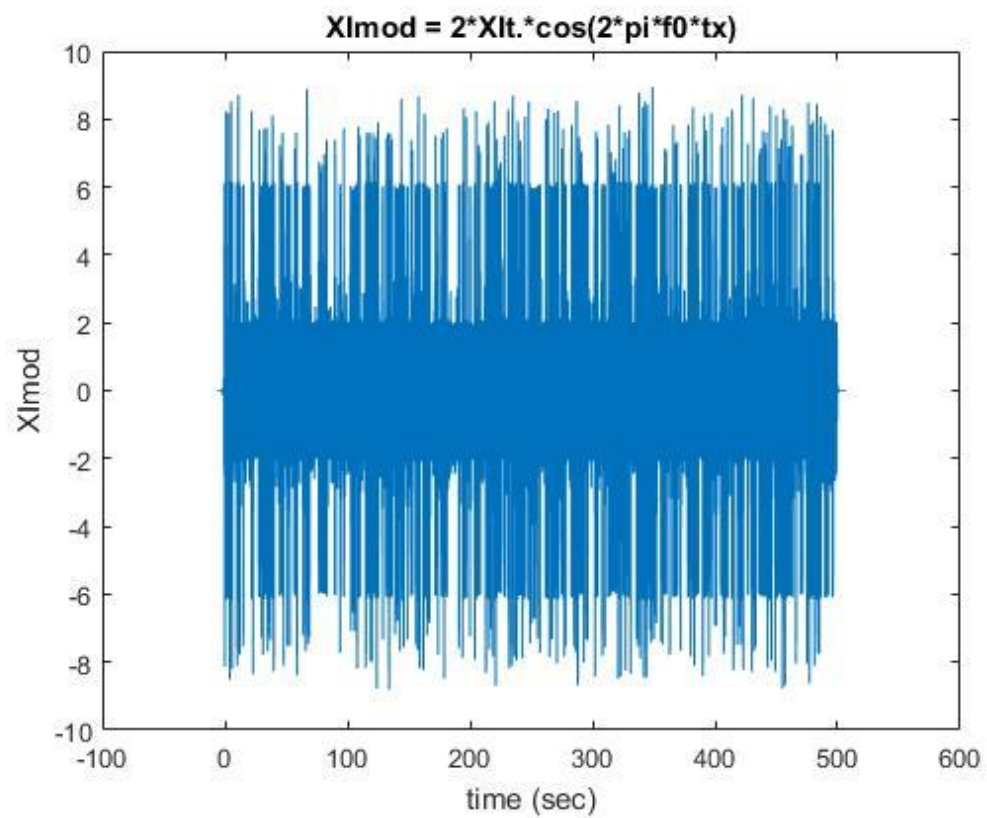
(5) Να πολλαπλασιάσετε τις κυματομορφές $X_I(t)$ και $X_Q(t)$ με τους αντίστοιχους φορείς (ενδεικτικά, $F_0 = 2 \text{ Hz}$) και να δημιουργήσετε τις κυματομορφές

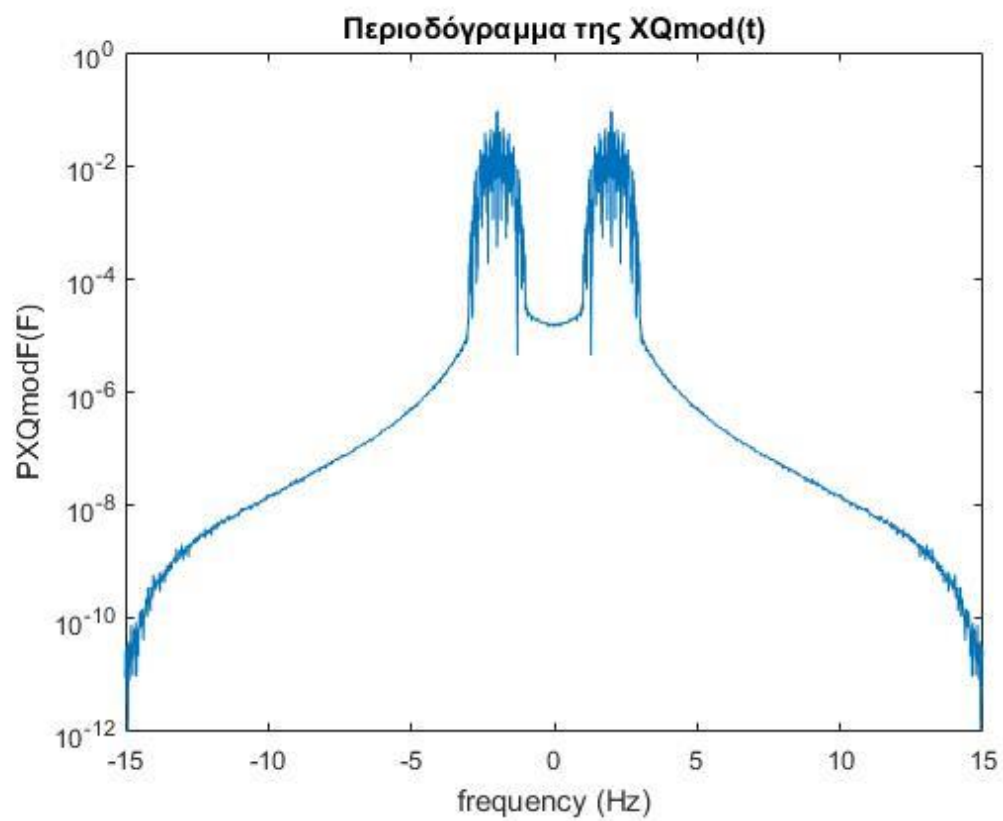
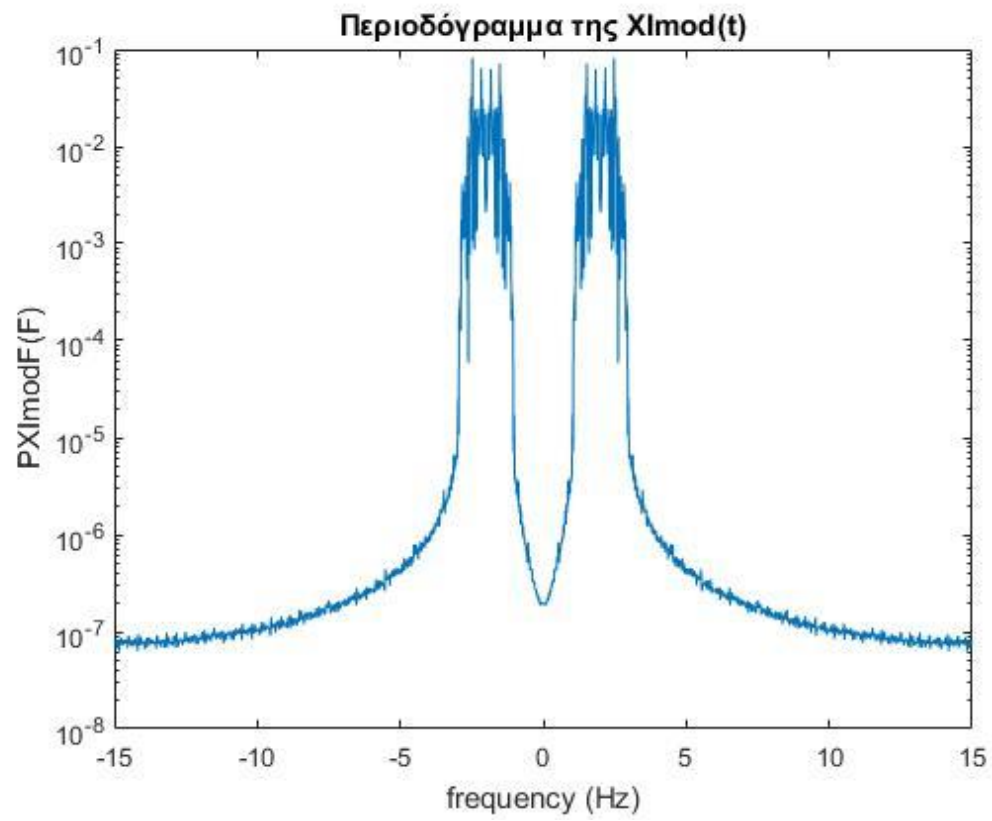
$$X_I^{mod}(t) = 2 X_I(t) \cos(2\pi F_0 t), \quad X_Q^{mod}(t) = -2 X_Q(t) \sin(2\pi F_0 t).$$

Να σχεδιάσετε τις κυματομορφές που προκύπτουν καθώς και τα αντίστοιχα περιοδογράμματα.

Τι παρατηρείτε;

Γραφικές Παραστάσεις :





Κώδικας :

```
f0 = 2;

XImod = XIt.*(2*cos(2*pi*f0*tx)); % Διαμόρφωση
XQmod = XQt.*((-2)*sin(2*pi*f0*tx));

figure(5);
plot (tx,XImod) ;
title('XImod = 2*XIt.*cos(2*pi*f0*tx)');
xlabel('time (sec)');
ylabel('XImod');

figure(6)
plot (tx,XQmod) ;
title('XQmod = -2*XQt.*sin(2*pi*f0*tx)');
xlabel('time (sec)');
ylabel('XQmod');

XImodF = fftshift(fft(XImod,Nf)*Ts);
PXImodF = power(abs(XImodF),2)/Ttotal ;
figure(7)
semilogy(F_axis,PXImodF)
title(['Περιοδογράμμα της XImod(t)'])
xlabel('frequency (Hz)');
ylabel('PXImodF (F)');

XQmodF = fftshift(fft(XQmod,Nf)*Ts);
PXQmodF = power(abs(XQmodF),2)/Ttotal ;
figure(8)
semilogy(F_axis,PXQmodF)
title(['Περιοδογράμμα της XQmod(t)'])
xlabel('frequency (Hz)');
ylabel('PXQmodF (F)');
```

Σχόλια :

Πολλαπλασιάζουμε τις 4PAM συναρτήσεις που προέκυψαν από το προηγούμενο ερώτημα με τους φορείς $2\cos(2\pi F_0 t)$ και $-2\sin(2\pi F_0 t)$ και σχεδιάζουμε τις συναρτήσεις που προκύπτουν και τα αντίστοιχα περιοδογραμμάτα τους .

Αυτό που παρατηρούμε είναι ότι οι γραφικές παραστάσεις των συναρτήσεων είναι σχεδόν ίδιες με την προηγούμενες , άλλα με μεγαλύτερο πλάτος και συχνότητα . Μεγαλύτερο ενδιαφέρον παρουσιάζουν τα περιοδογράμματα , τα οποία είναι ίδια με πριν την διαμόρφωση με τους φορείς , με την σημαντική διαφορά ότι έχουμε δύο αντίγραφα των λοβών , μετατοπισμένα κατά f_0 και στο $-f_0$ αντίστοιχα . Το αποτέλεσμα ήταν αναμενόμενο , καθώς ο μετασχηματισμός Fourier ενός διαμορφωμένου με ημιτονοειδή συνάρτηση σήματος παρουσιάζει συμμετρικό φάσμα γύρω από την αρχή των αξόνων, με αντίγραφα του φάσματος του μη διαμορφωμένου σήματος στο f_0 και στο $-f_0$. Επομένως , είναι λογικό το περιοδογράμμα που είναι συνάρτηση του φάσματος του σήματος , να παρουσιάζει ανάλογες αλλαγές .

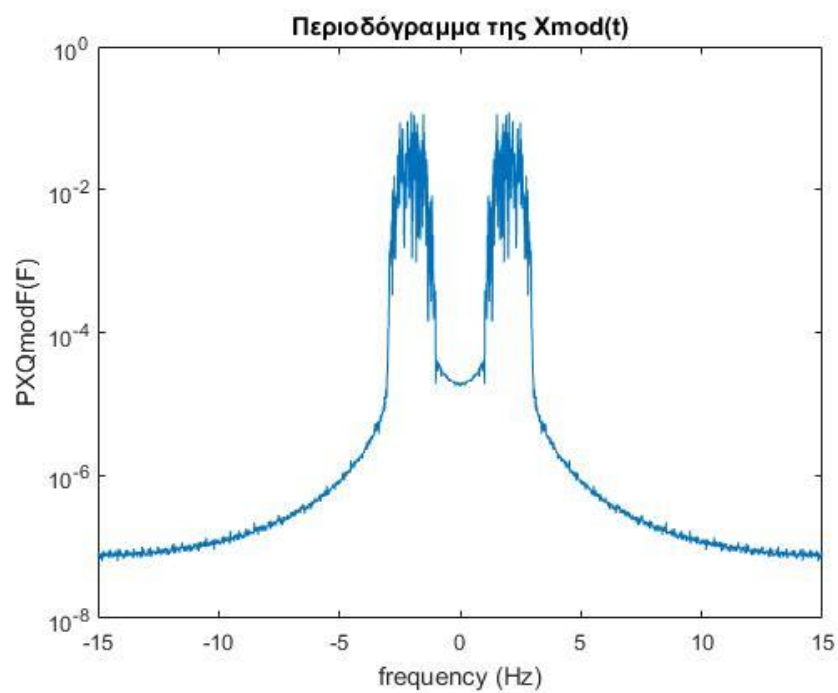
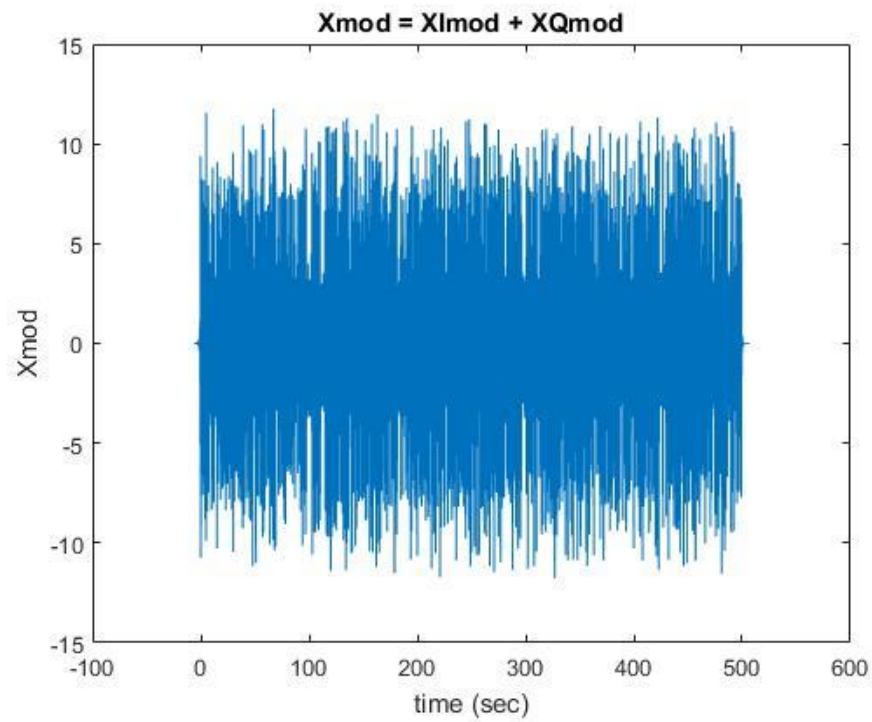
A.6.

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

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

και το περιοδόγραμμά της. Τι παρατηρείτε;

Γραφικές Παραστάσεις :



Κώδικας :

```
Xmod = XImod + XQmod;

figure(9)
plot (tx,Xmod) ;
title('Xmod = XImod + XQmod');
xlabel('time (sec)');
ylabel('Xmod');

XmodF = fftshift(fft(Xmod,Nf)*Ts);
PXmodF = power(abs(XmodF),2)/Ttotal ;
figure(10)
semilogy(F_axis,PXmodF)
title(['Περιοδόγραμμα της Xmod(t)'])
xlabel('frequency (Hz)');
ylabel('PXQmodF (F)');
```

Σχόλια :

Στο ερώτημα αυτό προσθέτουμε τα δύο διαμορφωμένα σήματα και σχεδιάζουμε την κυματομορφή στον χρόνο και το περιοδόγραμμα του σήματος που προκύπτει . Οι κυματομορφή και το περιοδόγραμμα του αθροίσματος έχουν την ίδια μορφή με τα επιμέρους σήματα , αλλά με μεγαλύτερο πλάτος λόγω του αθροίσματος .

A.7.

Να υποθέσετε ότι το κανάλι είναι ιδανικό.

Σχόλια :

Υποθέτουμε ότι το κανάλι είναι ιδανικό , οπότε η έξοδος και η είσοδος ταυτίζονται (η είσοδος συνελίσσεται με $\delta(t)$) , Επομένως δεν υπάρχει καμία αλλαγή , ούτε απαιτείται επιπρόσθετος κώδικας για αυτό το ερώτημα .

A.8.

(5) Στην έξοδο του καναλιού (δείτε το Σχήμα), να προσθέσετε λευκό Gaussian θόρυβο

$W(t)$ με διασπορά ίση με

$$\sigma_W^2 = \frac{10A^2}{T_s \cdot 10^{\frac{\text{SNR}_{4\text{dB}}}{10}}}.$$

Κώδικας :

```
mu = 0 ;  
SNRdb = 20 ;  
  
% Δημιουργία του θορύβου  
  
varianceW = (10*A)/(2*Ts*power(10,SNRdb/10)); sigmaW =  
sqrt(varianceW) ;  
Noise = normrnd(mu,sigmaW,1,length(tx)) ;  
  
XmodNoise = Xmod + Noise ;
```

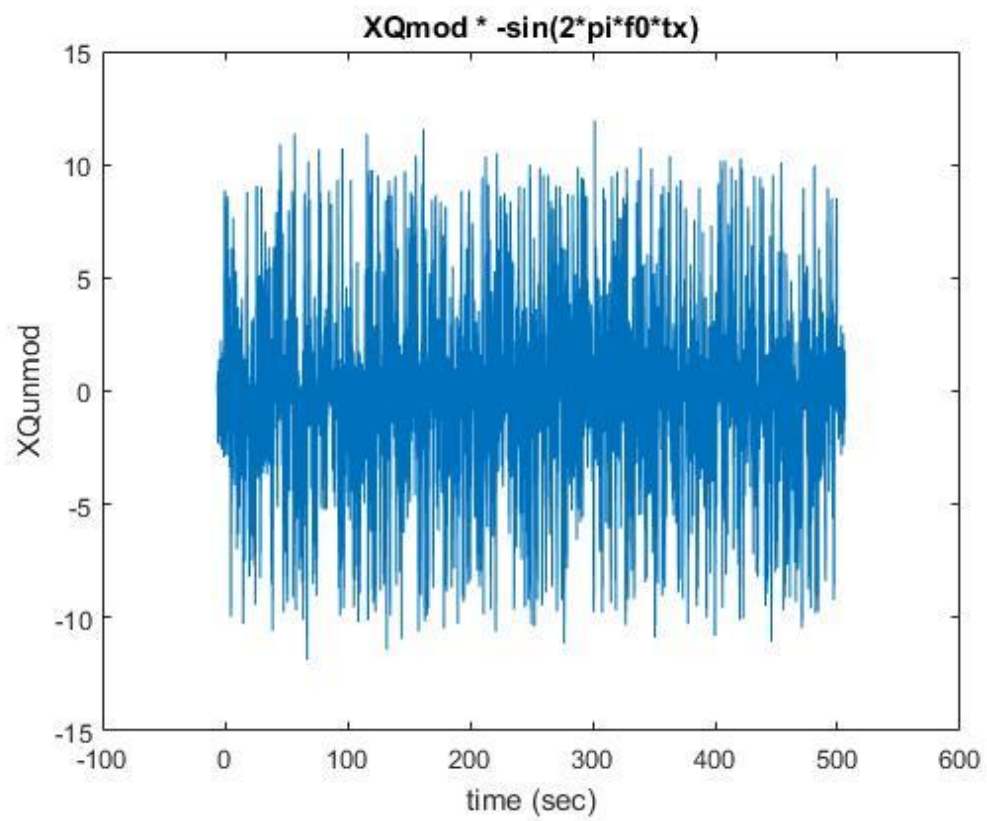
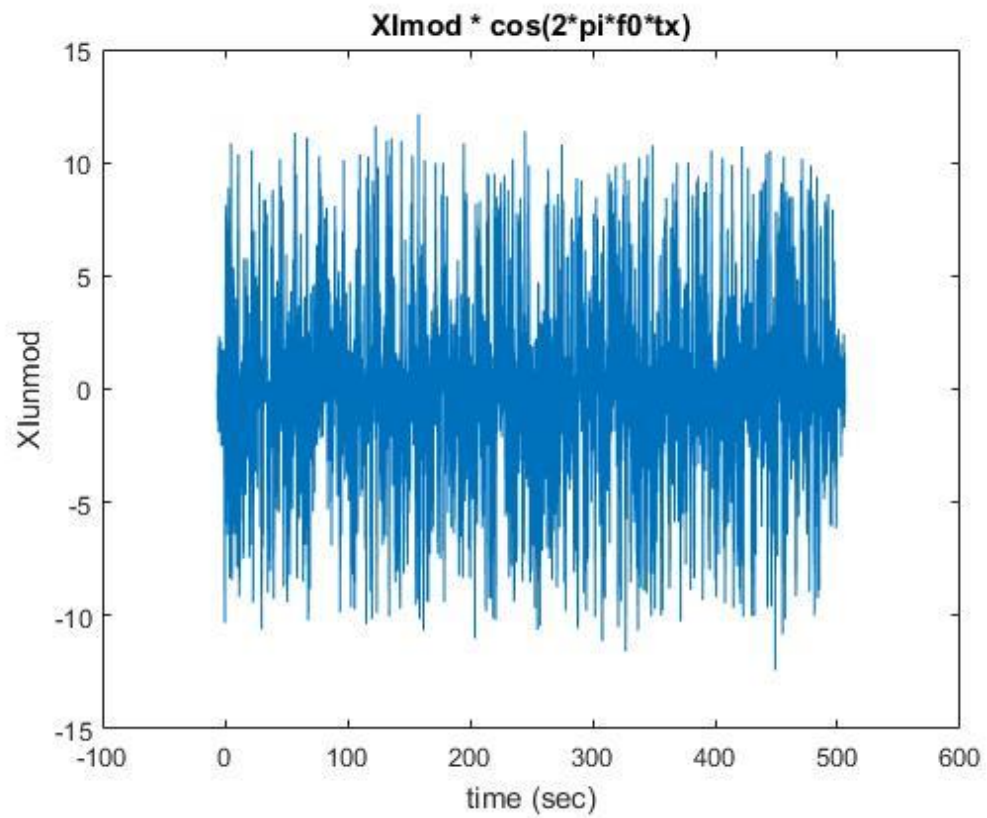
Σχόλια :

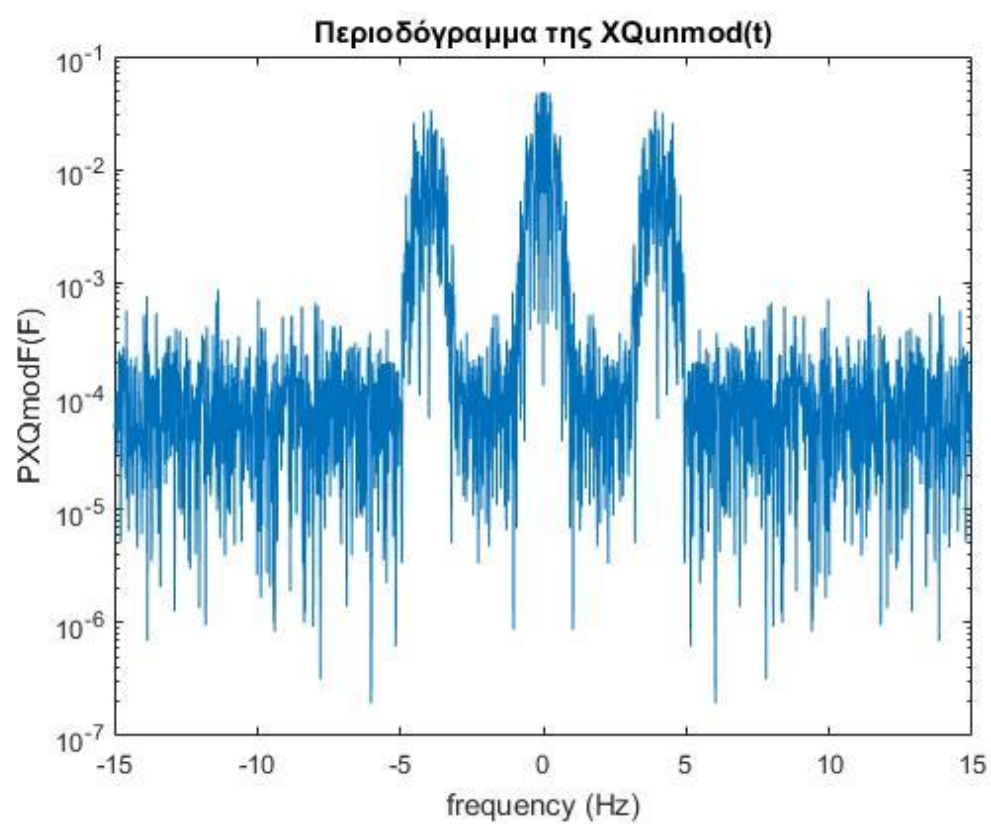
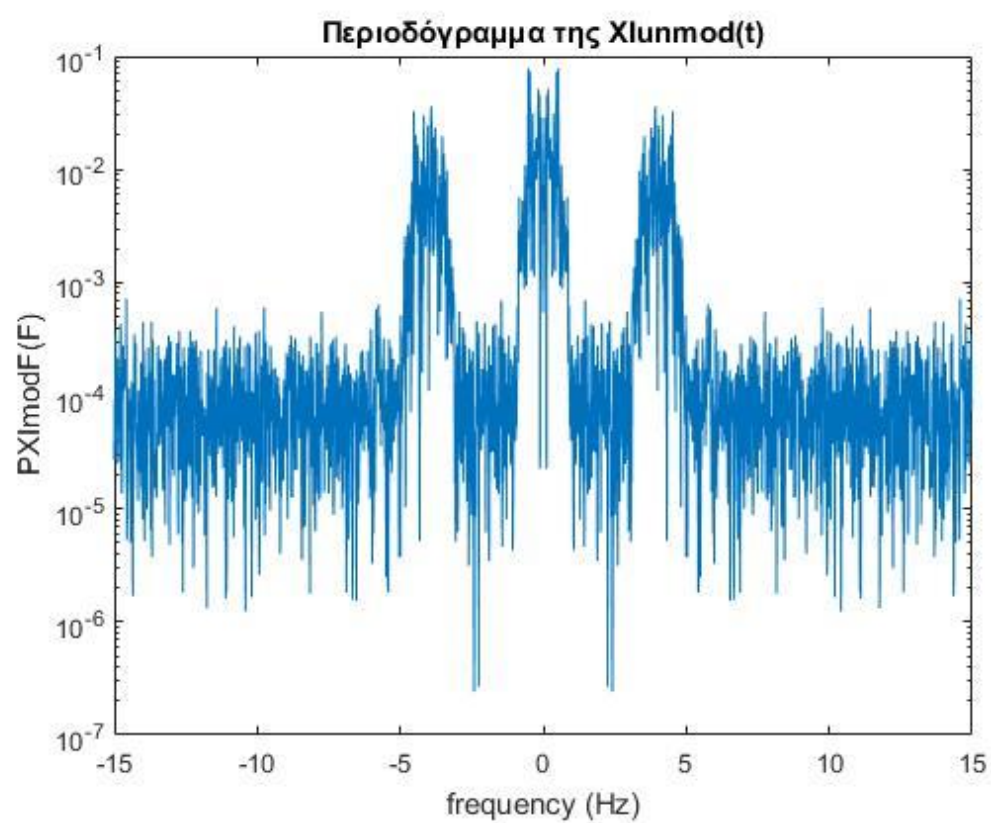
Στα τηλεπικοινωνιακά συστήματα , παρόλο που θα επιθυμούσαμε η έξοδος του πομπού και η είσοδος του δέκτη να ταυτίζονται , αυτό στην πράξη δεν συμβαίνει . Αντίθετα στην είσοδο του δέκτη παρατηρούμε ότι έχει προστεθεί και κάτι άλλο , ένας 'θόρυβος' σε σχέση με την έξοδο του πομπού . Για να προσομοιώσουμε το φαινόμενο αυτό , παράγουμε λευκό Gaussian θόρυβο με μέση τιμή 0 και διασπορά σ_w^2 , όπως δίνεται στην εκφώνηση , με SNR_{db} (signal-to-noise ratio) = 20 . Έπειτα τον προσθέτουμε στην έξοδο του πομπού και προκύπτει το σήμα που θα λάβει ο δέκτης . Η παραγωγή του θορύβου γίνεται την συνάρτηση **normrnd** , η οποία παράγει τιμές Gaussian τυχαίας μεταβλητής με τα παραπάνω χαρακτηριστικά .

A.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 περιοδογράμματά τους. Τι παρατηρείτε;

Γραφικές Παραστάσεις :





Κώδικας :

```
XIunmod = XmodNoise .* cos(2*pi*f0*tx) ;
XQunmod = -XmodNoise .* sin(2*pi*f0*tx) ;

figure(11);
plot (tx,XIunmod) ;
title('XIunmod * cos(2*pi*f0*tx)');
xlabel('time (sec)');
ylabel('XIunmod');

figure(12)
plot (tx,XQunmod) ;
title('XQunmod * -sin(2*pi*f0*tx)');
xlabel('time (sec)');
ylabel('XQunmod');

XIunmodF = fftshift(fft(XIunmod,Nf)*Ts);
PXIunmodF = power(abs(XIunmodF),2)/Ttotal ;
figure(13)
semilogy(F_axis,PXIunmodF)
title(['Περιοδόγραμμα της XIunmod(t)'])
xlabel('frequency (Hz)');
ylabel('PXIunmodF (F)');

XQunmodF = fftshift(fft(XQunmod,Nf)*Ts);
PXQunmodF = power(abs(XQunmodF),2)/Ttotal ;
figure(14)
semilogy(F_axis,PXQunmodF)
title(['Περιοδόγραμμα της XQunmod(t)'])
xlabel('frequency (Hz)');
ylabel('PXQunmodF (F)');
```

Σχόλια :

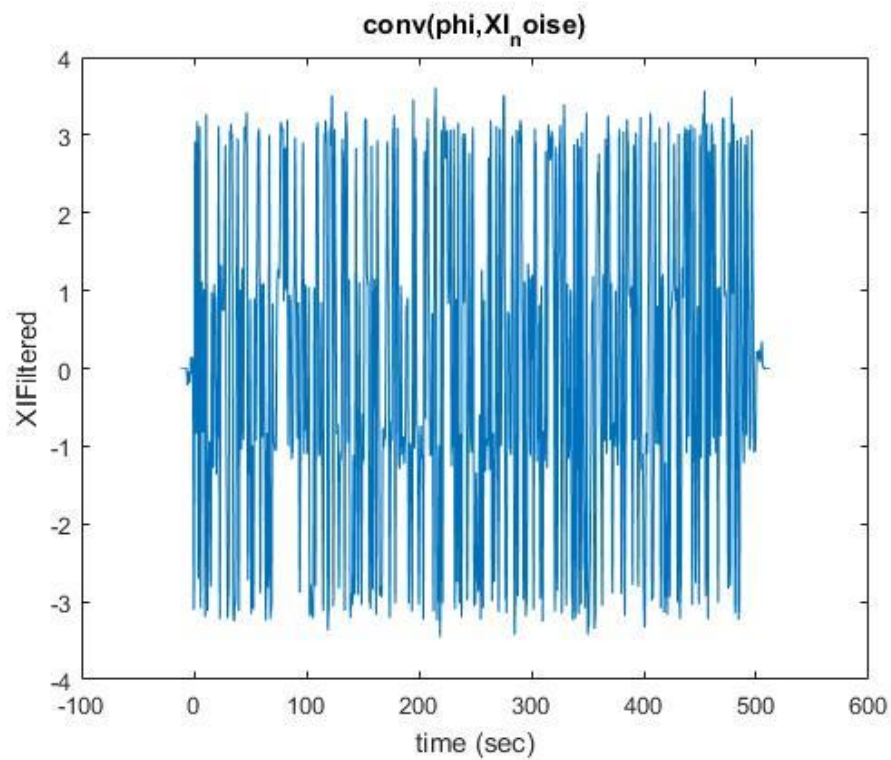
Στο βήμα αυτό , αποδιαμορφώνουμε τα δύο σήματα πολλαπλασιάζοντας με τους αντίστοιχους φορείς και σχεδιάζουμε την κυματομορφή στον χρόνο και το περιοδόγραμμα των δύο συναρτήσεων . Η κυματομορφή είναι ίδια με προηγουμένως , αλλά πιο ασαφώς καθορισμένη λόγω του θορύβου και με μικρότερη συχνότητα λόγω της αποδιαμόρφωσης , ενώ το πλάτος παραμένει περίπου το ίδιο . Τα ίδια ισχύουν για το περιοδόγραμμα , με μια όμως σημαντική διαφορά : Ο λοβός το κέντρο επέστρεψε , όπως το επιθυμούσαμε , όμως παραμένουν δύο λοβοί στο $\pm 2f_0$.

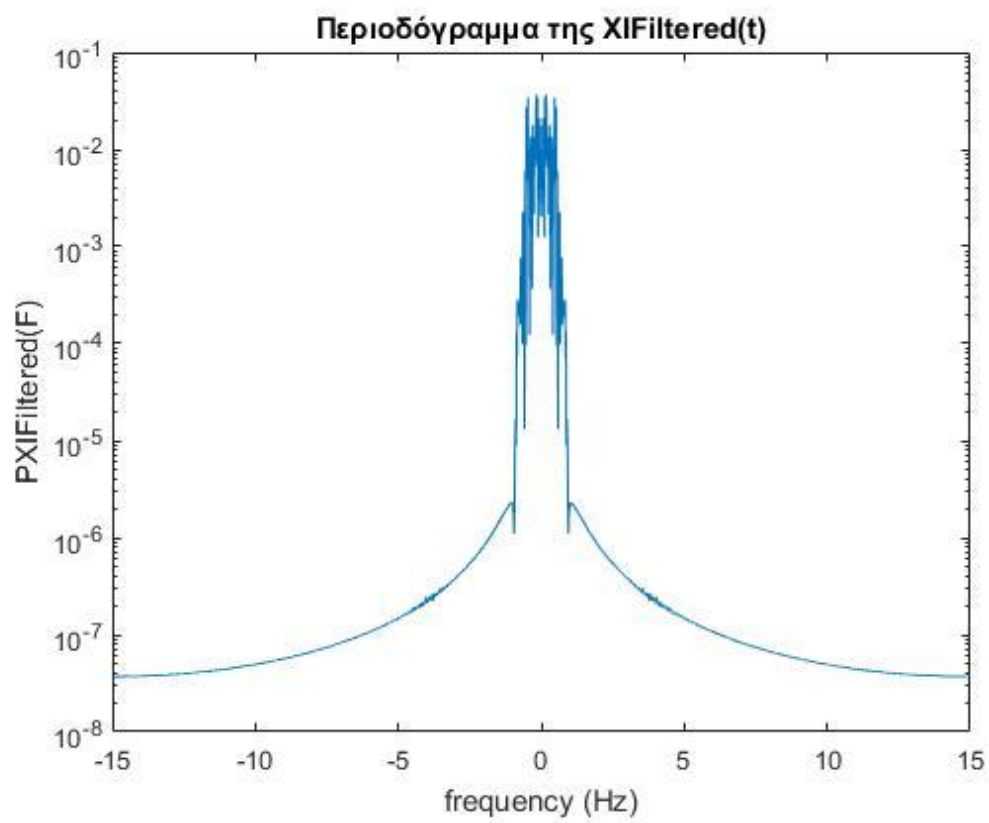
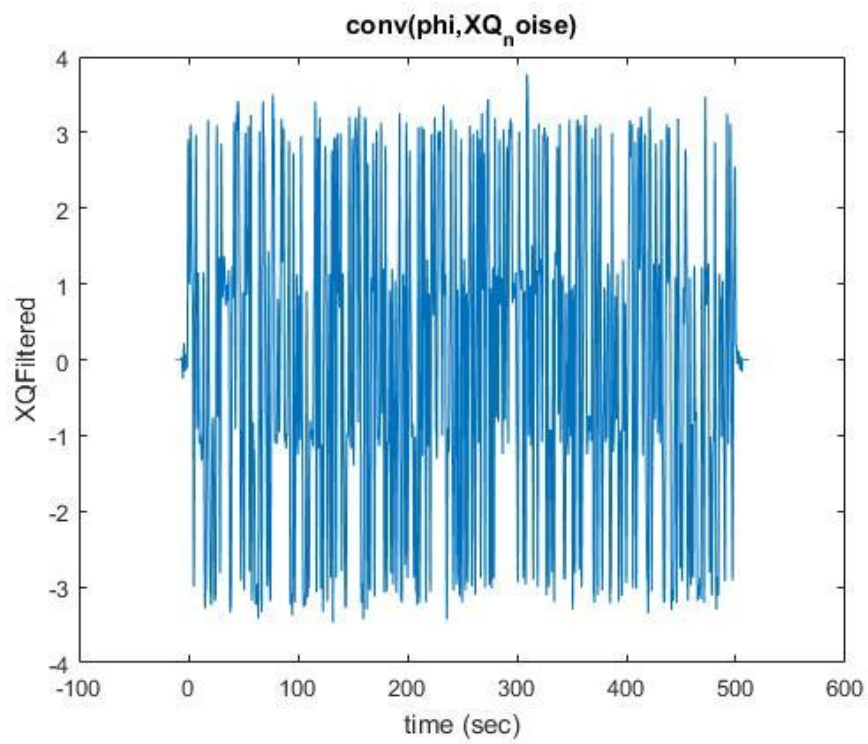
Σε αυτό το σημείο αξίζει να αναφερθεί ότι η διαδικασία της διαμόρφωσης και της αποδιαμόρφωσης γίνεται με ιδανικό τρόπο στην εργασία αυτή . Σε πραγματικά συστήματα , το f_0 του φορέα του διαμορφωτή και του φορέα του αποδιαμορφωτή έχουν μια μικρή διαφορά στις τιμές του και συχνά οι φορείς έχουν διαφορά φάσης . Εμείς θεωρούμε ότι οι συχνότητες διαμόρφωσης και αποδιαμόρφωσης ταυτίζονται απόλυτα και μηδενική διαφορά φάσης .

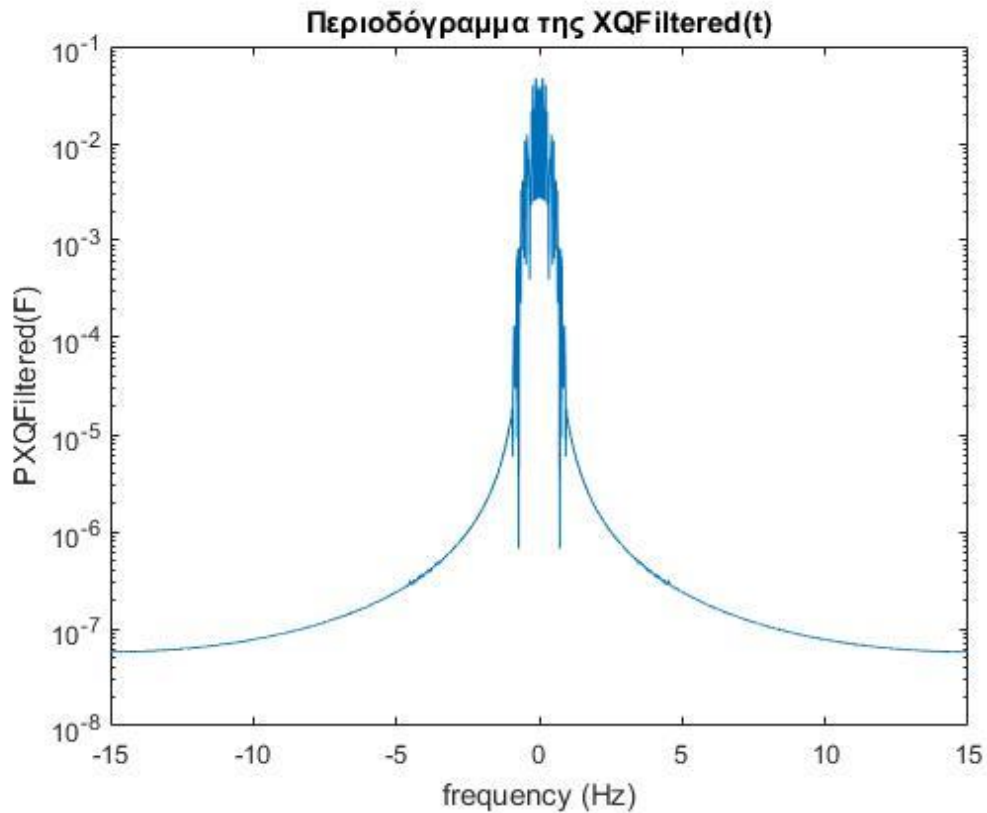
A.10.

(5) Να υπολογίσετε την έξοδο των προσαρμοσμένων φίλτρων. Να σχεδιάσετε τις κυματομορφές που προκύπτουν και τα περιοδογράμματά τους (να θέσετε το σωστό άξονα χρόνου). Τι παρατηρείτε;

Γραφικές Παραστάσεις :







Κώδικας :

```
XIFiltered = conv(phi,XIunmod)*Ts ;
XQFiltered = conv(phi,XQunmod)*Ts ;

tnew = linspace(t_phi(1) + tx(1),t_phi(end) +
tx(end),length(XIFiltered)) ; % λόγω συνέλιξης
Ttotal = length(tnew) ;

figure(15);
plot (tnew,XIFiltered) ;
title('conv(phi,XI_noise)');
xlabel('time (sec)');
ylabel('XIFiltered');

figure(16)
plot (tnew,XQFiltered) ;
title('conv(phi,XQ_noise)');
xlabel('time (sec)');
ylabel('XQFiltered');

XIFilteredF = fftshift(fft(XIFiltered,Nf)*Ts);
PXIFilteredF = power(abs(XIFilteredF),2)/Ttotal ;
figure(17)
semilogy(F_axis,PXIFilteredF)
title(['Περιοδόγραμμα της XIFiltered(t)'])
xlabel('frequency (Hz)');
```

```

ylabel('PXIFiltered(F)');

XQFilteredF = fftshift(fft(XQFiltered,Nf)*Ts);
PXQFilteredF = power(abs(XQFilteredF),2)/Ttotal ;
figure(18)
semilogy(F_axis,PXQFilteredF)
title(['Περιοδογράμμα της XQFiltered(t)'])
xlabel('frequency (Hz)');
ylabel('PXQFiltered(F)');

```

Σχόλια :

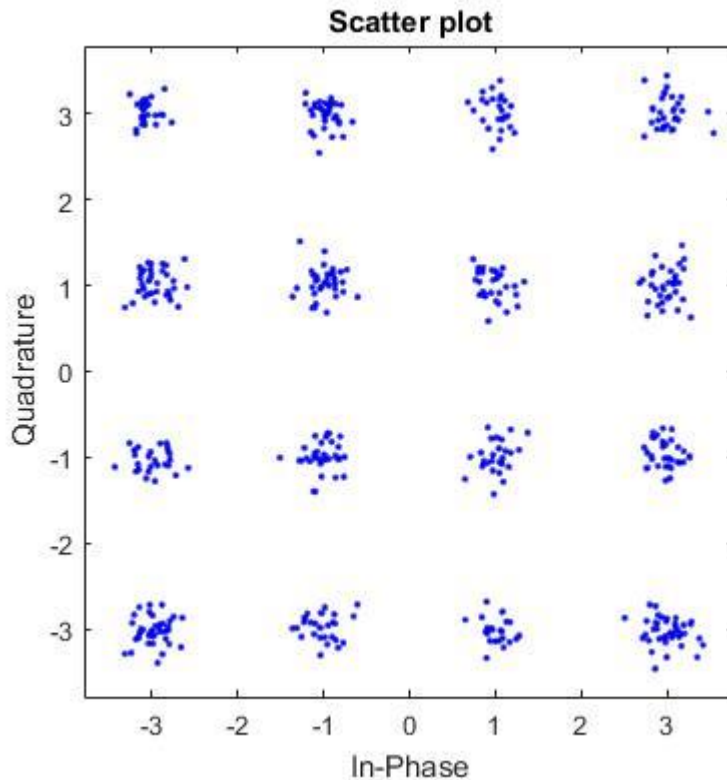
Στην συνέχεια περνάμε τα δύο σήματα μέσα από το προσαρμοσμένο φίλτρο , δηλαδή τα συνελίσσουμε τα $\Phi(t)$ και σχεδιάζουμε τις κυματομορφές στο χρόνο και τα περιοδογράμματα . Παρατηρούμε ότι οι μορφές των κυματομορφών και των περιοδογραμμάτων μοιάζουν σε μεγάλο βαθμό με τις 4-PAM συναρτήσεις που σχεδιάσαμε στην αρχή , πριν το φιλτράρισμα με $\Phi(t)$ και την διαμόρφωση με μια ελαφρά απόκλιση στο πλάτος .

Το αποτέλεσμα αυτό είναι το επιθυμητό καθώς δείχνει ότι έγινε επιτυχής μετάδοση της πληροφορίας από το πομπό στον δεκτή .Το αποτέλεσμα αυτό οφείλεται στην ορθοκανονικότητα της SRRC ως προς τις μετατοπίσεις της , μια ιδιότητα που έχει αναλυθεί εκτενώς στις προηγούμενες ασκήσεις . Τέλος αξίζει να σημειωθεί ότι οι δυο μη επιθυμητοί λοβοί στο $\pm 2f_0$, εξαλείφθηκαν , επομένως το SRRC λειτούργησε ως χαμηλοπερατό φίλτρο , το οποίο κράτησε ανέπαφη μόνο την χρήσιμη πληροφορία του σήματος .

A.11.

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

Γραφικές Παραστάσεις :



Κώδικας :

```
Y = zeros(N,2);
i=1;

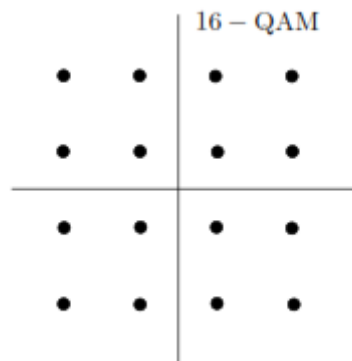
for k = 2*A_phi*(T/Ts) : (T/Ts) : (length(XIFiltered)-1) -
2*A_phi*(T/Ts) % N δειγματα ανα Fs άρα 2N δειγματα
    Y(i,1)=XIFiltered(k);
    Y(i,2)=XQFiltered(k);
    i=i+1;
end

scatterplot (Y)
```

Σχόλια :

Στο βήμα αυτό , δειγματοληπτούμε την έξοδο του δέκτη με βήμα $t = kT/T_s$, παραλείποντας τις πρώτες και τις τελευταίες $2 \cdot A \cdot T/T_s$ μη επιθυμητές τιμές, που προέκυψαν από τις συνελίξεις με την συνάρτηση $\Phi(t)$. Έχουμε λάβει N (500) δείγματα από κάθε έξοδο . Το βήμα αυτό κρίθηκε απαραίτητο , γιατί αν περιλαμβάναμε αυτές τις τιμές διαπιστώθηκε ότι δεν προκύπτει χρήσιμο αποτέλεσμα στα παρακάτω ερωτήματα . Έτσι προκύπτει ένας πίνακας δύο διαστάσεων Y με πρώτη διάσταση τα δείγματα των inphase συμβόλων και δεύτερη διάσταση τα δείγματα των quadrature συμβόλων. Στην συνέχεια , περνάμε ως όρισμα τον πίνακα αυτόν , στην συνάρτηση scatterplot , ώστε να σχεδιάσουμε την ακολουθία εξόδου . Παρατηρούμε ότι η έξοδος είναι αρκετά κοντά στην ιδανική απεικόνιση της 16-

QAM. Η απόκλιση οφείλεται στον θόρυβο, που προσθέσαμε στην είσοδο του δέκτη.
Η ιδανική αναπαράσταση είναι :

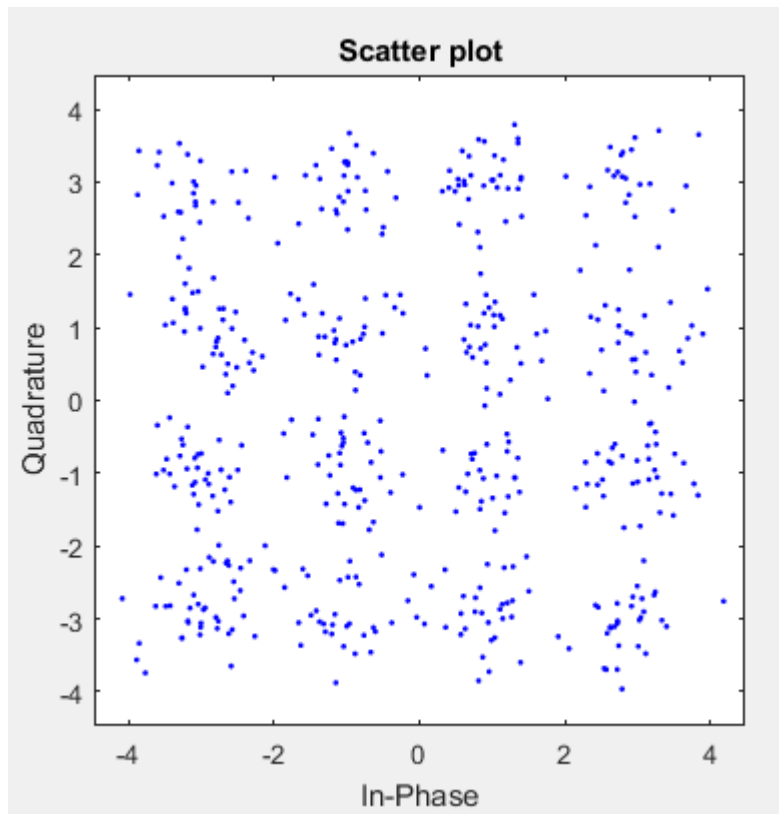


Επίσης στο σχεδιασμό των δειγμάτων της εξόδου , παίζει καίριο ρόλο , η επιλογή του SNR_{db} , δηλαδή του πόσο μεγαλύτερη είναι η ισχύς του σήματος σε σχέση με την ισχύ του θορύβου , καθώς

$$SNR = P_X / P_N \Leftrightarrow$$

$$SNR_{db} = 10 \cdot \log (P_X / P_N)$$

όπου P_X και P_N η ισχύς του σήματος και του θορύβου αντίστοιχα . Για μικρότερο SNR_{db} η αναπαράσταση των δειγμάτων της εξόδου θα αποκλίνει περισσότερο του ιδανικού . Για παράδειγμα για $SNR_{db} = 12$:



η απόκλιση από το ιδανικό 16-QAM , είναι μεγαλύτερη λόγω της αυξημένης επίδρασης του θορύβου.

A.12.

(10) Να γράψετε συνάρτηση

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

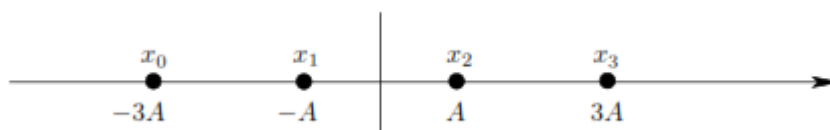
η οποία χρησιμοποιεί τον κανόνα εγγύτερου γείτονα και αποφασίζει για την ακολουθία εισόδου 4-PAM σύμβολο-προς-σύμβολο. Να εφαρμόσετε τη συνάρτηση ξεχωριστά στα δείγματα που λάβατε στην inphase έξοδο και στην quadrature έξοδο και να λάβετε αποφάσεις για τις αντίστοιχες ακολουθίες εισόδου.

Κώδικας :

```
est_XI_symb = detect_4_PAM((Y(:,1)), A); % inphase
est_XQ_symb = detect_4_PAM((Y(:,2)), A); % quadrature
```


Σχόλια :

Στην άσκηση αυτή υλοποιήσαμε ένα σύστημα 16-QAM , με χρήση δυο επιμέρους ανεξάρτητων συστημάτων 4-PAM . Ο αστερισμός του 4-PAM είναι ο εξής :



Στο βήμα αυτό καλούμαστε να υλοποιήσουμε την συνάρτηση **detect_4_PAM** , η οποία για κάθε ακολουθία 4-PAM θα αποφασίζει με τον κανόνα του εγγύτερου γείτονα από τα δείγματα που λάβαμε ποια είναι η ακολουθία εισόδου που αντιστοιχεί σε κάθε είσοδο. Ο κανόνας αυτός λειτουργεί ως εξής :

- Έχουμε 4 περιοχές (σύνολα τιμών) για τις τιμές του Y (της δειγματοληπτημένης εξόδου) , τις $\{D_0, D_1, D_2, D_3\}$. Ανάλογα από ποια από αυτές τις περιοχές ανήκει η έξοδος , τότε η αντίστοιχη είσοδος είναι $\{X_1, X_2, X_3, X_4\}$. Οι περιοχές , όπως φαίνεται και από το παραπάνω σχήμα είναι οι εξής :

1. $D_0 = \{ Y \leq -2A \}$ με $X_0 = -3A = -3$
2. $D_1 = \{ -2A < Y < 0 \}$ με $X_1 = -A = -1$
3. $D_2 = \{ 0 \leq Y < 2A \}$ με $X_2 = +A = +1$
4. $D_3 = \{ Y \geq 2A \}$ με $X_3 = +3A = +3$

Η **detect_4_PAM** λαμβάνει ως ορίσματα την ακολουθία συμβόλων Y που δειγματοληπτήσαμε από την έξοδο και το A και αφού εκτιμήσει σε ποια περιοχή ανήκει κάθε σύμβολο , αποθηκεύει σε ένα νέο διάνυσμα το σύμβολο εισόδου που αντιστοιχεί στην περιοχή αυτή . Τέλος επιστέφει αυτό το διάνυσμα . Ο κώδικας της συνάρτησης περιλαμβάνεται στο τέλος της αναφοράς .

Εφαρμόζουμε λοιπόν την συνάρτηση αυτή και στις δύο ακολουθίες εξόδου και λαμβάνουμε μια εκτίμηση για τις δύο εισόδους $\{X_{I,n}\}$ και $\{X_{Q,n}\}$.

A.13.

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

Κώδικας :

```
est_X_symb = [est_XI_symb est_XQ_symb] ;  
Symbol_Errors = symerr(X,est_X_symb);
```

Σχόλια :

Στο βήμα αυτό ενοποιούμε τις δύο ακολουθίες συμβόλων σε ένα διάνυσμα μήκους $2*N$ (1000 σύμβολα) και τις συγκρίνουμε με τις τιμές του αρχικού διανύσματος συμβόλων X . Η σύγκριση γίνεται με την συνάρτηση **symerr** της MATLAB, η οποία κάνει ένα προς ένα σύγκριση των συμβόλων των δύο διανυσμάτων και επιστρέφει τον αριθμό των σφαλμάτων των συμβόλων. Ο λόγος που δεν χρησιμοποιήθηκε for loop σε αυτό το ερώτημα είναι για πιο σύντομο και γρήγορο κώδικα. Τα αποτελέσματα και με τους δύο τρόπους είναι τα ίδια.

A.14.

(10) Να γράψετε συνάρτηση

```
function 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.

Κώδικας :

```
est_XI_bits = PAM_4_to_bits(est_XI_symb,A) ;  
est_XQ_bits = PAM_4_to_bits(est_XQ_symb,A) ;  
est_X_bits = [est_XI_bits est_XQ_bits] ;
```

Σχόλια :

Στο ερώτημα αυτό υλοποιούμε συνάρτηση **PAM_4_to_bits**, η οποία δέχεται ως όρισμα ένα διάνυσμα συμβόλων και με αντίστροφη απεικόνιση Gray, επιστρέφει ένα διάνυσμα bits, το οποίο αντιστοιχεί στην εκτιμώμενη ακολουθία bits της εισόδου. Συγκεκριμένα με χρήση for loop, ανάλογα με το σύμβολο, δίνεται σε ένα νέο διάνυσμα, η αντίστοιχη δυάδα bits. Έπειτα η συνάρτηση επιστρέφει αυτό το διάνυσμα. Ο κώδικας της συνάρτησης περιλαμβάνεται στο τέλος της αναφοράς.

A.15.

(10) Να υπολογίσετε τον αριθμό σφαλμάτων bit.

Κώδικας :

```
est_X_bits = [est_XI_bits est_XQ_bits] ;  
Bit_Errors = biterr (bit_seq',est_X_bits) ;
```

Σχόλια :

Στο βήμα αυτό ενοποιούμε τις δύο ακολουθίες bits σε ένα διάνυσμα μήκους $4*N$ (2000 bits) και τις συγκρίνουμε με τις τιμές του αρχικού διανύσματος bits **bit_seq**. Η σύγκριση γίνεται με την συνάρτηση **biterr** της MATLAB, η οποία κάνει ένα προς ένα σύγκριση των bits των δύο διανυσμάτων και επιστρέφει τον αριθμό των σφαλμάτων των bits. Ο λόγος που δεν χρησιμοποιήθηκε for loop σε αυτό το ερώτημα

είναι για πιο σύντομο και γρήγορο κώδικα . Τα αποτελέσματα και με τους δύο τρόπους είναι τα ίδια .

Αξίζει να σημειωθεί ότι όσο αυξάνεται το SNR_{db} , τόσο ο αριθμός σφαλμάτων συμβόλων και ο αριθμός σφαλμάτων bits τείνουν στο 0 . Αυτό φαίνεται περισσότερο στο B μέρος της άσκησης , που βρίσκουμε τον αριθμό σφαλμάτων για διαφορετικά SNR_{db} με αύξουσα σειρά . Επομένως διαπιστώνουμε ότι όσο αυξάνουμε την ισχύ του σήματος σε σχέση με τον θόρυβο , τόσο μειώνεται ο αριθμός των σφαλμάτων . Για το τωρινό $SNR_{db} = 20$ δεν έχουμε σχεδόν καθόλου σφάλματα . Για μικρότερο SNR_{db} ο αριθμός σφαλμάτων αυξάνεται σημαντικά .

B

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.

B.1.

1. Για $SNR_{db} = [0 : 2 : 16]$, να υπολογίσετε πειραματικά την πιθανότητα σφάλματος απόφασης συμβόλου και bit επαναλαμβάνοντας τα παραπάνω βήματα K φορές (ενδεικτικά, $K = 200, 1000$) για κάθε SNR. Η πειραματική εκτίμηση της πιθανότητας σφάλματος συμβόλου για κάθε SNR υπολογίζεται ως εξής:

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

ενώ η πιθανότητα σφάλματος bit υπολογίζεται ως εξής:

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

Κώδικας :

```
K = 1000 ;
SNRdb = 0:2:16 ;

SentSymbols = N ;
SentBits = 4*N ;

Pesymb_exp = zeros (length(SNRdb),1) ;
Pebit_exp = zeros (length(SNRdb),1) ;
Pesymb_th = zeros (length(SNRdb),1) ;
Pebit_th = zeros (length(SNRdb),1) ;

j=1 ;
SumOfSymbErr = 0 ;
SumOfBitErr = 0 ;

for SNR = 0:2:16      % Experiment
```

```

SumOfSymbErr = 0 ;
SumOfBitErr = 0 ;

for i = 1:K
    [Symbol_Errors, Bit_Errors] = Error_Possibility
(N,A,T,A_phi,roll_off,over,f0,SNR) ;
    SumOfSymbErr = Symbol_Errors + SumOfSymbErr ;
    SumOfBitErr = Bit_Errors + SumOfBitErr ;
end

% Η διαδικασία επαναλαμβάνεται K φορές
Pesymb_exp(j) = SumOfSymbErr / (SentSymbols*K) ;
Pebit_exp(j) = SumOfBitErr / (SentBits*K) ;

j = j + 1 ;
end

```

Σχόλια :

Στο ερώτημα αυτό μας ζητείται να υπολογίσουμε για 9 διαφορετικές τιμές του SNR_{db} ($\{0,2,4,6,8,10,12,14,16\}$) , την πιθανότητα σφάλματος συμβόλου και πιθανότητα σφάλματος bit με την μέθοδο του Monte Carlo .Η μέθοδος αυτή απαιτεί την επανάληψη του μέρους A της άσκησης (16-QAM σύστημα) K φορές (επιλέχθηκε $K = 1000$ για μεγαλύτερη ακρίβεια) για κάθε τιμή του SNR_{db} , εύρεση του συνολικού αριθμού σφαλμάτων συμβόλων και bits σε κάθε περίπτωση και διαίρεση τους με τον συνολικό αριθμό συμβόλων ($2*N*K$) και συνολικό αριθμό bits ($4*N*K$) για το υπολογισμό της πειραματικής πιθανότητας σφάλματος συμβόλου και bit αντίστοιχα.

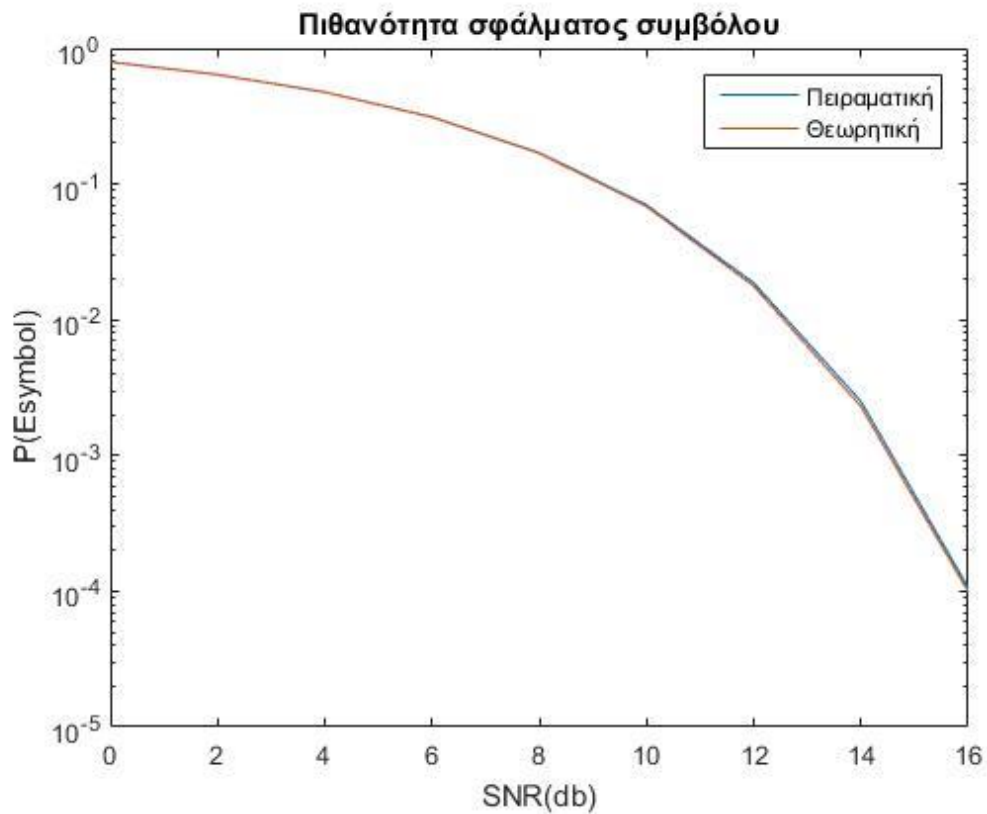
Για τον σκοπό κρίθηκε απαραίτητο , να υλοποιηθεί μια ξεχωριστή συνάρτηση με όνομα Error_Possibility με ορίσματα όλες τις παραμέτρους που ορίσαμε στο A μέρος , δηλαδή το μήκος ακολουθίας N , τον αστερισμό A , την περίοδο , το over , τις παραμέτρους A και roll-off της $\Phi(t)$, την συχνότητα διαμόρφωσης f_0 και το πιο σημαντικό , το SNR_{db} . Ο κώδικας της Error_Possibility είναι ίδιος με τον κώδικα του A μέρους , αλλά δεν περιέχει υπολογισμούς περιοδογραμμάτων , ούτε εκτύπωση οποιασδήποτε κυματομορφής ή scatterplot . Ωστόσο , ο κώδικας περιλαμβάνεται στο τέλος της αναφοράς για λόγους πληρότητας .

Ο υπολογισμός της πειραματικής πιθανότητας σφάλματος επιτυγχάνεται με χρήση εμφωλευμένου loop , όπου για κάθε τιμή του SNR_{db} , εκτελείται η Error_Possibility K φορές και προστίθενται σε δυο ξεχωριστές μεταβλητές τα σφάλματα συμβόλων και bits . Μόλις τελειώσουν οι K επαναλήψεις , υπολογίζονται οι αντίστοιχες πιθανότητες. Έτσι όταν τελειώσει ο διπλός βρόγχος έχουμε δύο διανύσματα 9 τιμών με τις πιθανότητες σφάλματος συμβόλου και bit .

B.2.

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

Γραφικές Παραστάσεις :



Κώδικας :

```
i = 1 ; % Theoretical Possibilities
for SNR = 0:2:16
    varianceW = (10*A)/(2*Ts*power(10,SNR/10));
    varianceN = varianceW * Ts * 0.5 ;
    sigmaN = sqrt(varianceN) ;
    Pesymb_th(i) = 3*Q(A/sigmaN) ; % Πιθανότητα σφάλματος 16-QAM
    i = i+1 ;
end

figure(20)
semilogy (SNRdb,Pesymb_exp) ;
hold on ;
semilogy (SNRdb,Pesymb_th) ;
title (['Πιθανότητα σφάλματος συμβόλου'])
legend ('Πειραματική','Θεωρητική')
xlabel('SNR (db) ');
ylabel('P (Esymbol) ');
```

Σχόλια :

Στο προηγούμενο ερώτημα υπολογίσαμε τις πειραματικές πιθανότητες σφάλματος με την μέθοδο του Monte Carlo για διαφορετικές τιμές του SNR_{db} . Η θεωρητική πιθανότητα σφάλματος υπολογίζεται με την σχέση

$$P_{\text{symbol}}(E) = 3 * Q(A/\sigma_N)$$

με $Q(x) = 1 - \Phi(x)$, όπου Φ η αθροιστική συνάρτηση κατανομής της τυποποιημένης κανονικής μεταβλητής . Για τον υπολογισμό της πιθανότητας αυτής χρησιμοποιείται η συνάρτηση **Q** , της οποίας ο κώδικας δίνεται και πρέπει να υπολογιστεί το σ_N , το οποίο δίνεται ότι είναι :

$$\sigma_N^2 = \sigma_W^2 * T_s * 0.5$$

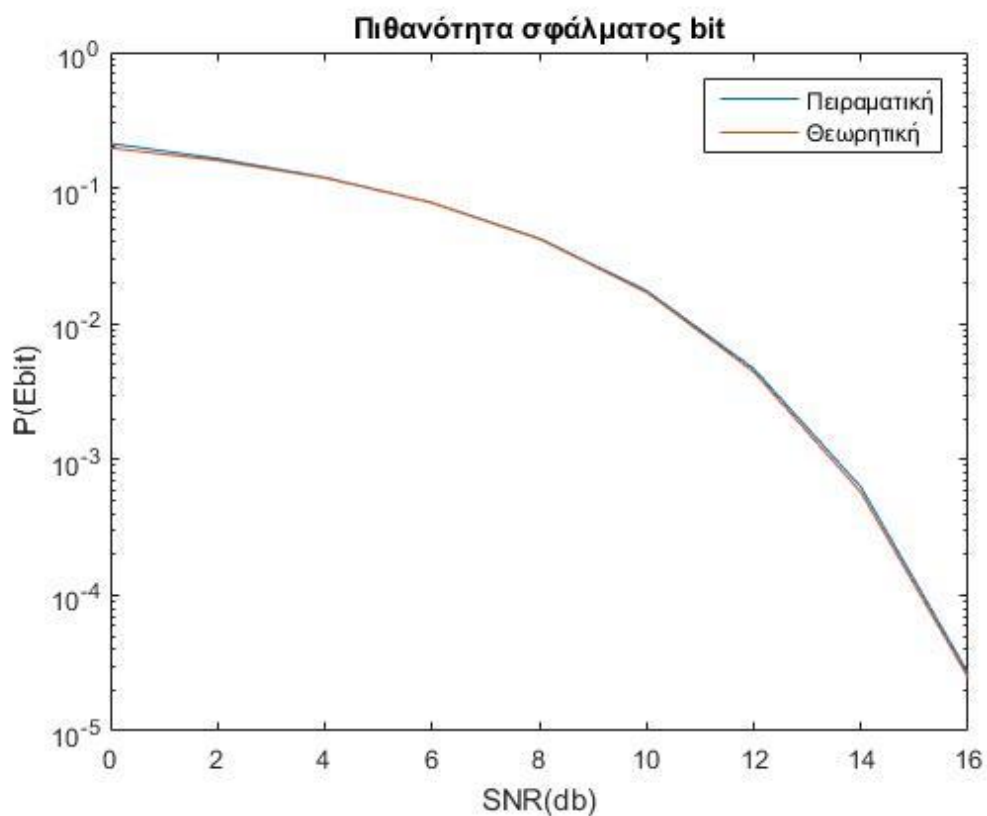
Με χρήση ενός for loop, υπολογίζουμε μια θεωρητική πιθανότητα για κάθε τιμή του SNR_{db}, επομένως έχουμε ένα διάγραμμα θεωρητικών πιθανοτήτων συμβόλου με 9 τιμές .

Έπειτα , σχεδιάζουμε την θεωρητική και την πειραματική πιθανότητα συμβόλου σε κοινό semilogy . Επιπλέον παρατηρούμε ότι οι η πειραματική πιθανότητα σφάλματος συμβόλου με την θεωρητική ταυτίζονται απόλυτα και ότι με την αύξηση του SNR_{db}, μειώνεται η πιθανότητα σφάλματος . Αυτό όπως προαναφέρθηκε συμβαίνει επειδή λόγω της αύξησης της ισχύς του σήματος εισόδου σε σχέση με τον θόρυβο , μειώνεται η επίδραση του θορύβου στον είσοδο του δέκτη , άρα έχουμε καλύτερη λήψη αποφάσεων με βάση την δειγματοληψία της εξόδου του δέκτη .

B.3.

(20) 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;

Γραφικές Παραστάσεις :



Κώδικας :

```
for i = 1:length(Pesymb_th)
    Pebit_th(i) = Pesymb_th(i) / log2(16) ; % log2(16) = bps 16 - QAM
end
```

```
figure(21)
semilogy (SNRdb,Pebit_exp) ;
hold on ;
semilogy (SNRdb,Pebit_th) ;
title (['Πιθανότητα σφάλματος bit'])
legend ('Πειραματική', 'Θεωρητική')
xlabel('SNR(db)');
ylabel('P(Ebit)');
```


Σχόλια :

Στο ερώτημα αυτό καλούμαστε να υπολογίσουμε την θεωρητική πιθανότητα σφάλματος bit . Αυτή προκύπτει από την θεωρητική πιθανότητα σφάλματος συμβόλου , διαιρεμένη με $\log_2(16)$. Το $\log_2(16)$ είναι ο αριθμός των bits ανά σύμβολο (bps) , καθώς έχουμε διαμόρφωση 16-QAM . Έπειτα σχεδιάζουμε την θεωρητική και την πειραματική πιθανότητα σε κοινό semilogy και βλέπουμε ότι πάλι με την αύξηση του SNR_{db} μειώνεται η πιθανότητα σφάλματος για τους λόγους που αναφέρθηκαν στο προηγούμενο ερώτημα . Επιπλέον παρατηρούμε ότι οι η πειραματική πιθανότητα σφάλματος bit με την θεωρητική ταυτίζονται απόλυτα .

ΣΥΝΟΛΙΚΟΣ ΚΩΔΙΚΑΣ

ΣΥΝΑΡΤΗΣΕΙΣ

```
function [Symbol_Errors, Bit_Errors] =  
Error_Possibility(N,A,T,A_phi,roll_off,over,f0,SNRdb)  
  
bit_seq = (sign(randn(4*N,1))+1)/2;  
  
X = bits_to_4_PAM(bit_seq, A);  
  
XI = X(1:N);  
XQ = X(N+1:end);  
  
Ts = T/over ; % Περίοδος δειγματοληψίας  
Fs = 1/Ts ; % Συχνότητα δειγματοληψίας  
  
[phi,t_phi] = srrc_pulse(T,Ts,A_phi,roll_off);  
  
[tx,XIt] = PAM4_function (XI,Ts,over,phi,t_phi) ;  
[tx,XQt] = PAM4_function (XQ,Ts,over,phi,t_phi) ;  
  
XImod = XIt.*(2*cos(2*pi*f0*tx));  
XQmod = XQt.*((-2)*sin(2*pi*f0*tx));  
  
Xmod = XImod + XQmod;  
  
mu = 0 ;  
varianceW = (10*A)/(2*Ts*power(10,SNRdb/10));  
sigmaW = sqrt(varianceW) ;  
Noise = normrnd(mu,sigmaW,1,length(tx)) ;  
  
XmodNoise = Xmod + Noise ;  
  
XIunmod = XmodNoise .* cos(2*pi*f0*tx) ;  
XQunmod = -XmodNoise .* sin(2*pi*f0*tx) ;  
  
XIFiltered = conv(phi,XIunmod)*Ts ;  
XQFiltered = conv(phi,XQunmod)*Ts ;
```

```

Y = zeros(N,2);
i=1;

for k = 2*A_phi*(T/Ts) : (T/Ts) : (length(XIFiltered)-1) -
2*A_phi*(T/Ts) % N δείγματα ανα Fs άρα 2N δείγματα
    Y(i,1)=XIFiltered(k);
    Y(i,2)=XQFiltered(k);
    i=i+1;
end

est_XI_symb = detect_4_PAM((Y(:,1)), A); % inphase
est_XQ_symb = detect_4_PAM((Y(:,2)), A); % quadrature
est_X_symb = [est_XI_symb est_XQ_symb] ;

Symbol_Errors = symerr(X,est_X_symb) ;

est_XI_bits = PAM_4_to_bits(est_XI_symb,A) ;
est_XQ_bits = PAM_4_to_bits(est_XQ_symb,A) ;
est_X_bits = [est_XI_bits est_XQ_bits] ;

Bit_Errors = biterr (bit_seq',est_X_bits) ;



---


function [X] = bits_to_4_PAM(bit_seq,A )

X = zeros(1,length(bit_seq)/2);
j=1;

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



---


function [tx,Xt]= PAM4_function (X,Ts,over,phi,t_phi)

X_delta= 1/Ts * upsample(X,over);
t = 0:Ts:length(X) - Ts ;

Xt = conv(phi,X_delta) * Ts ;
tx = linspace(t(1)+ t_phi(1),t(end)+ t_phi(end),length(Xt)) ;
end



---


function [est_X] = detect_4_PAM(Y,A )

est_X = zeros (1,length(Y)) ;

```

```

for i=1:length(Y)
    if (Y(i)>= 2*A) % D3
        est_X(i) = +3*A;
    elseif ((Y(i)< 2*A) && (Y(i) >= 0)) % D2
        est_X(i) = +A ;
    elseif ((Y(i)> -2*A) && (Y(i) < 0)) % D1
        est_X(i) = -A ;
    elseif (Y(i)<= -2*A) % D0
        est_X(i) = -3*A ;
    end
end

```

```

function [est_X] = PAM_4_to_bits(X,A)

```

```

est_X = ones(1,2*length(X));

```

```

for i=1:length(X)
    if(X(i)==-3*A)
        est_X(2*i-1) = 0;
        est_X(2*i) =0;
    elseif(X(i)==-A)
        est_X(2*i-1) = 0;
        est_X(2*i)= 1;
    elseif(X(i)==A)
        est_X(2*i-1) = 1;
        est_X(2*i)= 1;
    elseif(X(i)==3*A)
        est_X(2*i-1) = 1;
        est_X(2*i)= 0;
    end
end
end

```

ΚΥΡΙΟΣ ΚΩΔΙΚΑΣ

```

clear all;
close all;
clc;

% A1

N = 500 ;
bit_seq = (sign(randn(4*N,1))+1)/2;

% A2

A = 1;
X = bits_to_4_PAM(bit_seq, A);

% A3

XI = X(1:N);
XQ = X(N+1:end);

% A4

```

```

T = 1 ;
over = 30 ;
Ts = T/over ; % Περίοδος δειγματοληψίας
Fs = 1/Ts ; % Συχνότητα δειγματοληψίας

A_phi = 6 ;
roll_off = 1 ;

[phi,t_phi] = srsrc_pulse(T,Ts,A_phi,roll_off);

[tx,XIt] = PAM4_function (XI,Ts,over,phi,t_phi) ;
[tx,XQt] = PAM4_function (XQ,Ts,over,phi,t_phi) ;

figure(1);
plot (tx,XIt) ;
title('4-PAM function - XI(t)');
xlabel('time (sec)');
ylabel('XI(t) = Sum(XI,n*Φ(t-kT))');

figure(2)
plot (tx,XQt) ;
title('4-PAM function - XQ(t)');
xlabel('time (sec)');
ylabel('XQ(t) = Sum(XQ,n*Φ(t-kT))');

Nf = 2048 ; % Αριθμός δειγμάτων στο πεδίο της συχνότητας
f_axis = [-1/2 : 1/Nf : 1/2 - (1/Nf)]; % Ανοικτό διάστημα δεξιά ,
κλειστό αριστερά
F_axis = f_axis*Fs;

Ttotal = length(tx);
XIF = fftshift(fft(XIt,Nf)*Ts);
PXIF = power(abs(XIF),2)/Ttotal ;
figure(3)
semilogy(F_axis,PXIF)
title(['Περιοδόγραμμα της XI(t)'])
xlabel('frequency (Hz)');
ylabel('PXI(F)');

XQF = fftshift(fft(XQt,Nf)*Ts);
PXQF = power(abs(XQF),2)/Ttotal ;
figure(4)
semilogy(F_axis,PXQF)
title(['Περιοδόγραμμα της XQ(t)'])
xlabel('frequency (Hz)');
ylabel('PXQ(F)');

% A5

f0 = 2;

XImod = XIt.*(2*cos(2*pi*f0*tx)); % Διαμόρφωση
XQmod = XQt.*((-2)*sin(2*pi*f0*tx));

figure(5);
plot (tx,XImod) ;
title('XImod = 2*XIt.*cos(2*pi*f0*tx)');
xlabel('time (sec)');

```

```

ylabel('XImod');

figure(6)
plot (tx,XQmod) ;
title('XQmod = -2*XQt.*sin(2*pi*f0*tx)');
xlabel('time (sec)');
ylabel('XQmod');

XImodF = fftshift(fft(XImod,Nf)*Ts);
PXImodF = power(abs(XImodF),2)/Ttotal ;
figure(7)
semilogy(F_axis,PXImodF)
title(['Περιοδόγραμμα της XImod(t)'])
xlabel('frequency (Hz)');
ylabel('PXImodF(F)');

XQmodF = fftshift(fft(XQmod,Nf)*Ts);
PXQmodF = power(abs(XQmodF),2)/Ttotal ;
figure(8)
semilogy(F_axis,PXQmodF)
title(['Περιοδόγραμμα της XQmod(t)'])
xlabel('frequency (Hz)');
ylabel('PXQmodF(F)');

% A6

Xmod = XImod + XQmod;

figure(9)
plot (tx,Xmod) ;
title('Xmod = XImod + XQmod');
xlabel('time (sec)');
ylabel('Xmod');

XmodF = fftshift(fft(Xmod,Nf)*Ts);
PXmodF = power(abs(XmodF),2)/Ttotal ;
figure(10)
semilogy(F_axis,PXmodF)
title(['Περιοδόγραμμα της Xmod(t)'])
xlabel('frequency (Hz)');
ylabel('PXQmodF(F)');

% A7 , A8

mu = 0 ;
SNRdb = 20 ;
% Δημιουργία του θορύβου
varianceW = (10*A)/(2*Ts*power(10,SNRdb/10));
sigmaW = sqrt(varianceW) ;
Noise = normrnd(mu,sigmaW,1,length(tx)) ;

XmodNoise = Xmod + Noise ;

% A9

XIunmod = XmodNoise .* cos(2*pi*f0*tx) ;
XQunmod = -XmodNoise .* sin(2*pi*f0*tx) ;

figure(11);

```

```

plot (tx,XIunmod) ;
title('XImod * cos(2*pi*f0*tx)');
xlabel('time (sec)');
ylabel('XIunmod');

figure(12)
plot (tx,XQunmod) ;
title('XQmod * -sin(2*pi*f0*tx)');
xlabel('time (sec)');
ylabel('XQunmod');

XIunmodF = fftshift(fft(XIunmod,Nf)*Ts);
PXIunmodF = power(abs(XIunmodF),2)/Ttotal ;
figure(13)
semilogy(F_axis,PXIunmodF)
title(['Περιοδόγραμμα της XIunmod(t)'])
xlabel('frequency (Hz)');
ylabel('PXImodF(F)');

XQunmodF = fftshift(fft(XQunmod,Nf)*Ts);
PXQunmodF = power(abs(XQunmodF),2)/Ttotal ;
figure(14)
semilogy(F_axis,PXQunmodF)
title(['Περιοδόγραμμα της XQunmod(t)'])
xlabel('frequency (Hz)');
ylabel('PXQmodF(F)');

% A10

XIFiltered = conv(phi,XIunmod)*Ts ;
XQFiltered = conv(phi,XQunmod)*Ts ;

tnew = linspace(t_phi(1) + tx(1),t_phi(end) +
tx(end),length(XIFiltered)) ; % λόγω συνέλιξης
Ttotal = length(tnew) ;

figure(15);
plot (tnew,XIFiltered) ;
title('conv(phi,XI_noise)');
xlabel('time (sec)');
ylabel('XIFiltered');

figure(16)
plot (tnew,XQFiltered) ;
title('conv(phi,XQ_noise)');
xlabel('time (sec)');
ylabel('XQFiltered');

XIFilteredF = fftshift(fft(XIFiltered,Nf)*Ts);
PXIFilteredF = power(abs(XIFilteredF),2)/Ttotal ;
figure(17)
semilogy(F_axis,PXIFilteredF)
title(['Περιοδόγραμμα της XIFiltered(t)'])
xlabel('frequency (Hz)');
ylabel('PXIFiltered(F)');

XQFilteredF = fftshift(fft(XQFiltered,Nf)*Ts);
PXQFilteredF = power(abs(XQFilteredF),2)/Ttotal ;
figure(18)

```

```

semilogy(F_axis,PXQFilteredF)
title (['Περιοδόγραμμα της XQFiltered(t)'])
xlabel('frequency (Hz)');
ylabel('PXQFiltered(F)');

% A11

Y = zeros(N,2);
i=1;

for k = 2*A_phi*(T/Ts) : (T/Ts) : (length(XIFiltered)-1) -
2*A_phi*(T/Ts) % N δειγματα ανα Fs άρα 2N δειγματα συνολικά
    Y(i,1) = XIFiltered(k);
    Y(i,2) = XQFiltered(k);
    i=i+1;
end

scatterplot (Y)

% A12

est_XI_symb = detect_4_PAM((Y(:,1)), A); % inphase
est_XQ_symb = detect_4_PAM((Y(:,2)), A); % quadrature

% A13

est_X_symb = [est_XI_symb est_XQ_symb] ;
Symbol_Errors = symerr(X,est_X_symb);

% A14

est_XI_bits = PAM_4_to_bits(est_XI_symb,A) ;
est_XQ_bits = PAM_4_to_bits(est_XQ_symb,A) ;

% A15

est_X_bits = [est_XI_bits est_XQ_bits] ;
Bit_Errors= biterr (bit_seq',est_X_bits) ;

% B1

K = 1000 ;
SNRdb = 0:2:16 ;

SentSymbols = N ;
SentBits = 4*N ;

Pesymb_exp = zeros (length(SNRdb),1) ;
Pebit_exp = zeros (length(SNRdb),1) ;
Pesymb_th = zeros (length(SNRdb),1) ;
Pebit_th = zeros (length(SNRdb),1) ;

j=1 ;
SumOfSymbErr = 0 ;
SumOfBitErr = 0 ;

for SNR = 0:2:16 % Experiment

```



```

SumOfSymbErr = 0 ;
SumOfBitErr = 0 ;

for i = 1:K
    [Symbol_Errors, Bit_Errors] = Error_Possibility
(N,A,T,A_phi,roll_off,over,f0,SNR) ;
    SumOfSymbErr = Symbol_Errors + SumOfSymbErr ;
    SumOfBitErr = Bit_Errors + SumOfBitErr ;
end

% Η διαδικασία επαναλαμβάνεται K φορές
Pesymb_exp(j) = SumOfSymbErr / (SentSymbols*K) ;
Pebit_exp(j) = SumOfBitErr / (SentBits*K) ;

j = j + 1 ;
end

% B2

i = 1 ; % Theoretical Possibilities
for SNR = 0:2:16
    varianceW = (10*A)/(2*Ts*power(10,SNR/10));
    varianceN = varianceW * Ts * 0.5 ;
    sigmaN = sqrt(varianceN) ;
    Pesymb_th(i) = 3*Q(A/sigmaN) ; % Πιθανότητα σφάλματος συμβόλου
16-QAM
    i = i+1 ;
end

figure(20)
semilogy (SNRdb, Pesymb_exp) ;
hold on ;
semilogy (SNRdb, Pesymb_th) ;
title (['Πιθανότητα σφάλματος συμβόλου'])
legend ('Πειραματική', 'Θεωρητική')
xlabel('SNR(db)');
ylabel('P(Esymbol)');

% B3

for i = 1:length(Pesymb_th)
    Pebit_th(i) = Pesymb_th(i) / log2(16) ; % log2(16) = bps 16 - QAM
end

figure(21)
semilogy (SNRdb, Pebit_exp) ;
hold on ;
semilogy (SNRdb, Pebit_th) ;
title (['Πιθανότητα σφάλματος bit'])
legend ('Πειραματική', 'Θεωρητική')
xlabel('SNR(db)');
ylabel('P(Ebit)');

```