04/01/2020

ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΑ ΣΥΣΤΉΜΑΤΑ Ι

<u>ΑΣΚΗΣΗ 3</u>

Περάκης Εμμανουήλ (ΑΜ: 2017030099)
Γεωργακάς Ιωάννης-Ιάσων (ΑΜ: 2017030021)

ΕΡΩΤΗΜΑ Α

Στο πρώτο μέρος αυτής της άσκησης προσομοιώνεται ένα ολοκληρωμένο τηλεπικοινωνιακό σύστημα υποθέτοντας ότι χρησιμοποιείται διαμόρφωση 16-QAM. Επιπλέον, θεωρείται ιδανικό κανάλι και απόλυτος συγχρονισμός μεταξύ πομπού και δέκτη.

1.

Αρχικά, για γνωστό N (στην περίπτωση ατυτή $N=200$) δημιουργείται μία δυαδική ακολουθία (0 ή 1) από 4N ανεξάρτητα και ισοπίθανα bits $\{b_0,\ldots,b_{4N-1}\}$ με χρήση της εντολής

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

Αυτό φαίνεται και στο Παράρτημα Κώδικα 1.1.

```
b = (sign(randn(4*N,1)) + 1)/2;%Generating a sequence of bits
```

Παράρτημα Κώδικα 1.1

2.

Στη συνέχεια δημιουργείται συνάρτηση X = bits_to_4PAM(bit_seq, A) η οποία παίρνει σαν όρισματα την ακολουθία από bits που δημιουργήθηκε παραπάνω και το A το οποίο εχεί αρχικοποιηθεί σε μία πακτωμένη τιμή (ειδικά $A=1$). Σαν έξοδο επιστρέφει μία ακολουθία 4-PAM συμβόλων κωδικοποιημένα κατά Gray με βάση την ακολουθία από bits, τα οποία μπορούν να πάρουν τις τιμές $\pm A, \pm 3A$.

```
function X = bits_to_4_PAM(b,A)
%UNTITLED2 Summary of this function goes here
%   Detailed explanation goes here
for i = 1 : 2: length(b)
    if(b(i) == 0 && b(i+1) == 0)

          X((i + 1)/2) = 3*A;


    elseif(b(i) == 0 && b(i+1) == 1)

          X((i + 1)/2) = A;


    elseif(b(i) == 1 && b(i+1) == 1)

           X((i + 1)/2) = -A;


    elseif(b(i) == 1 && b(i+1) == 0)

          X((i + 1)/2) = -3*A;


  end
end
end
```

Παράρτημα Κώδικα 1.2

3.

Τα πρώτα 2N bits της ακολουθίας απεικονίζονται στην ακολουθία $X_{I,n}$ συμβόλων 4-PAM ενώ τα υπόλοιπα 2N bits απεικονίζονται στην ακολουθία $X_{Q,n}$ συμβόλων 4-PAM. Οι δύο αυτές ακολουθίες υπερδειγματολειπτούνται ώστε να σχηματιστούν τα σήματα συνεχούς χρόνου

και $X_{Q,\delta}(t)=\sum_{n=0}^{N-1} X_{Q,n}\delta(t-nT)$ τα οποία περιέχουν την πληροφορία των αρχικών ακολουθιών. Οι διαδικασίες των δύο προηγούμενων μερών εκτελούνται στον κώδικα του Παραρτήματος Κώδικα 1.2.

```
%%%%%%%%%%%%%%%%%%Part 2 and 3%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Seperating the sequence b of bits in two sequences of bits
b1 = b(1 :1 :2*N);

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

%Creating sequences of symbols from 4-PAM alphabet
X_I = bits_to_4_PAM(b1,A);

X_Q = bits_to_4_PAM(b2,A);

X_I_d = (1/Ts)*upsample(X_I,over);

time_of_X_I_d = [0:Ts:(N*T - Ts)];

X_Q_d = (1/Ts)*upsample(X_Q,over);

time_of_X_Q_d = [0:Ts:(N*T - Ts)];
```

Παράρτημα Κώδικα 1.3

4. ,5.

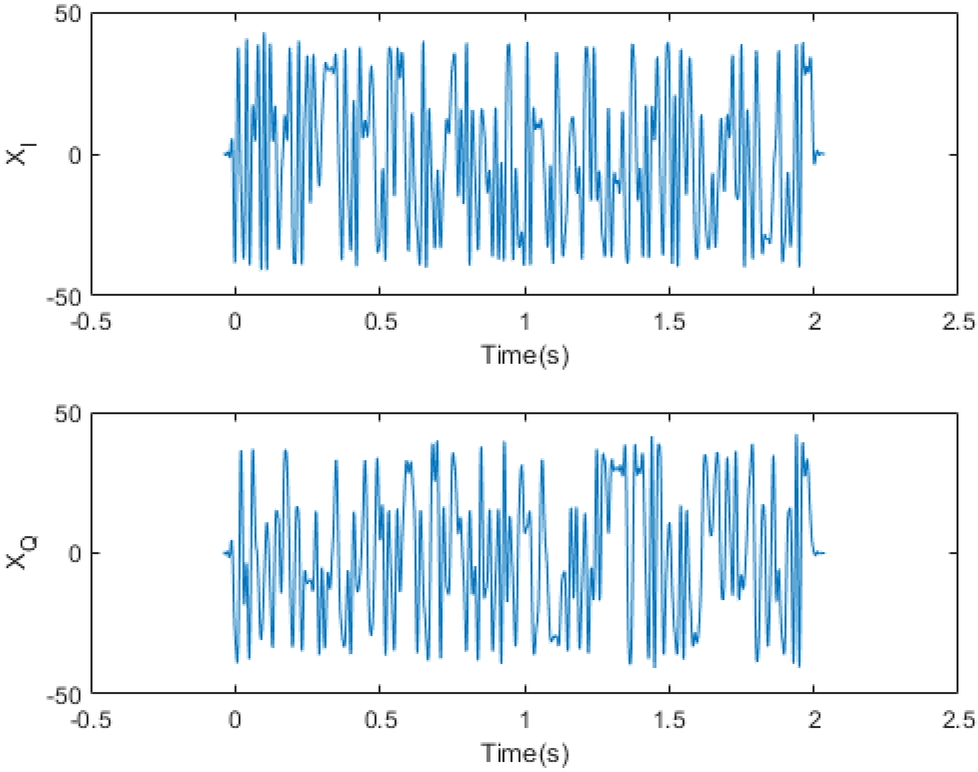
Στο βήμα αυτό δημιουργείται ένα φίλτρο μορφοποίησης με κρουστική απόκριση έναν SRRC παλμό με τα εξής χαρακτηριστικά :

- Περίοδος συμβόλου: T = 10^{-2} s
- Περίοδος δειγματοληψίας: T_s = $\frac{T}{\text{over}}$ με over = 10
- $A=4$ όπου A η μισή διάρκεια των παλμών σε περιόδους συμβόλων
- Συντελεστή roll-off $\alpha=0.5$

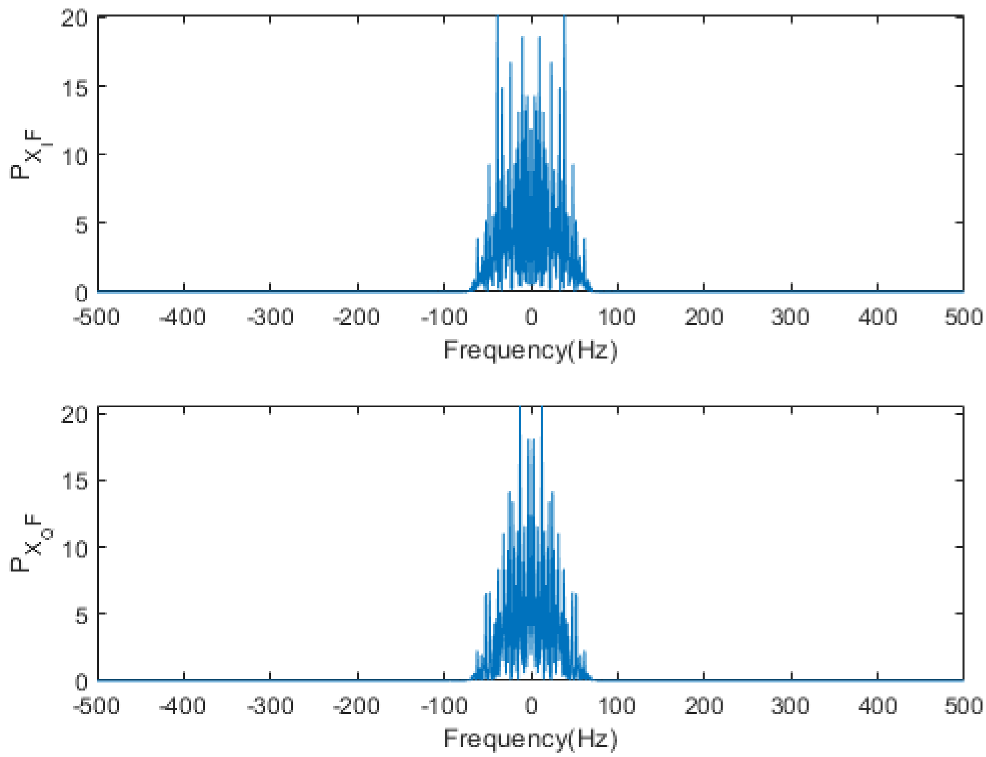
Ύστερα, τα προαναφερθέντα σήματα περνάνε μέσα από το φίλτρο μορφοποίησης παράγοντας στην έξοδο τα σήματα $X_I(t)$ και $X_Q(t)$.Τα περιοδογράμματα των σημάτων απεικονίζονται στο Σχήμα 1.2 . Οι κυματομορφές $X_I(t)$ και $X_Q(t)$ που προκύπτουν πολλαπλασιάζονται με τους φορείς $\cos(2\pi 200t)$ και $-\sin(2\pi 200t)$ αντίστοιχα ώστε να διαμορφωθούν και να μετατοπιστούν τα φάσματα τους στις συχνότητες ± 200 (διαμόρφωση DSB-SC AM) . Οι παράγοντες ± 2 εισάγονται για να αντισταθμιστεί η εξασθένηση του πλάτους που παρατηρείται κατά τη διαμόρφωση. Οι κυματομορφές $X_I^{mod}(t)$, $X_Q^{mod}(t)$ και περιοδογράμματα αυτών φαίνονται στο Σχήμα 1.3. Ο κώδικας φαίνεται στο Παράρτημα Κώδικα 1.3.

```
%%%%%%%%%%%%%%%%%%%Part 4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Creating a SRRC pulse
[phi, t] = srrc_pulse(T, Ts, A_pulse, a);

%Modulating the two sequences X_I_d and X_Q_d
X_i = Ts*conv(phi,X_I_d);

time_conv_X_i = [time_of_X_I_d(1) + t(1): DT : time_of_X_I_d(end) + t(end)];

X_q = Ts*conv(phi,X_Q_d);

time_conv_X_q = [time_of_X_Q_d(1) + t(1): DT : time_of_X_Q_d(end) + t(end)];

%Creating periodograms of X_i and X_q
t_total_X_i = time_conv_X_i(end)-time_conv_X_i(1);

P_X_i_F = (abs(fftshift(fft(X_i,Nf)*Ts)).^2)./(t_total_X_i);

t_total_X_q = time_conv_X_q(end)-time_conv_X_q(1);

P_X_q_F = (abs(fftshift(fft(X_q,Nf)*Ts)).^2)./(t_total_X_q);

X_I_mod = 2*X_i.*cos(2*pi*F0.*time_conv_X_i);

X_Q_mod = -2*X_q.*sin(2*pi*F0.*time_conv_X_q);

P_X_i_mod_F = (abs(fftshift(fft(X_I_mod,Nf)*Ts)).^2)./(t_total_X_i);

P_X_q_mod_F = (abs(fftshift(fft(X_Q_mod,Nf)*Ts)).^2)./(t_total_X_q);
```

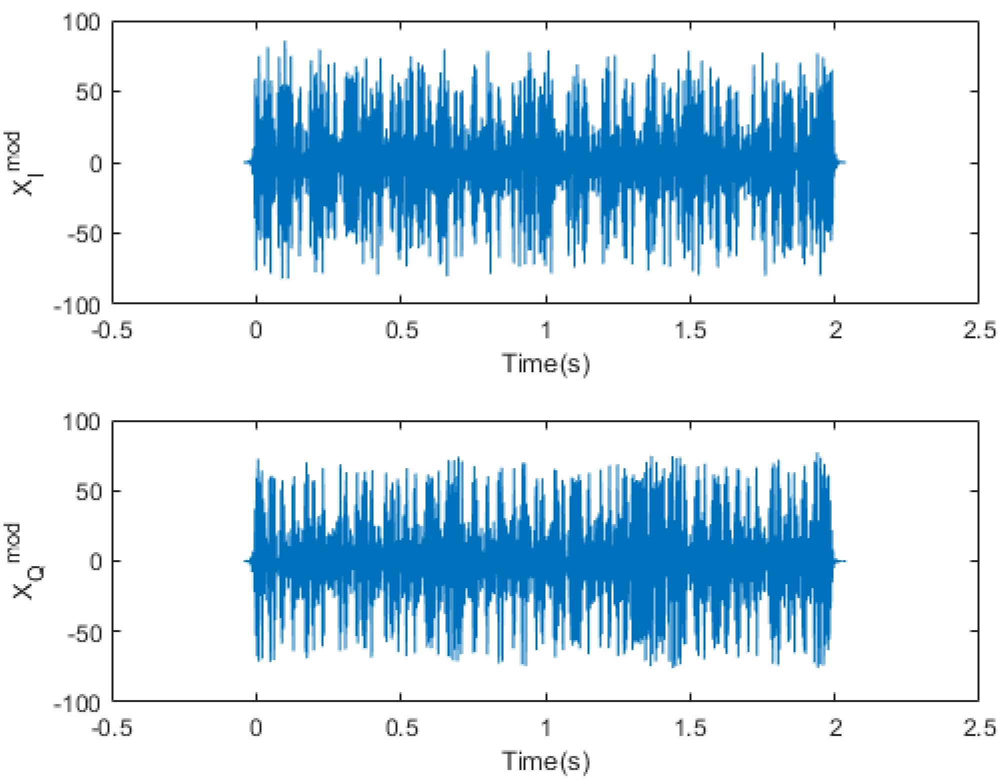
Παράρτημα Κώδικα 1.4



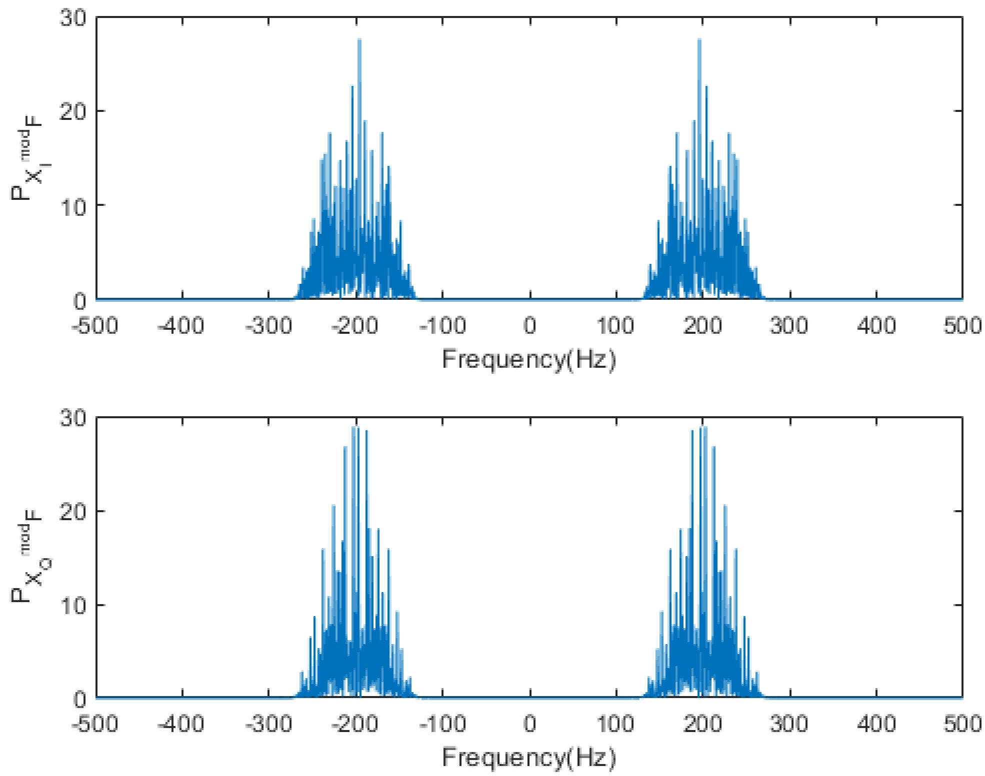
Σχήμα 1.1: Συνέλιξη των X_I , X_Q με τον SRRC παλμό



Σχήμα 1.2: Περιοδόγραμμα των Xi, Xq



Σχήμα 1.3: Τα διαμορφωμένα Xi, Xq



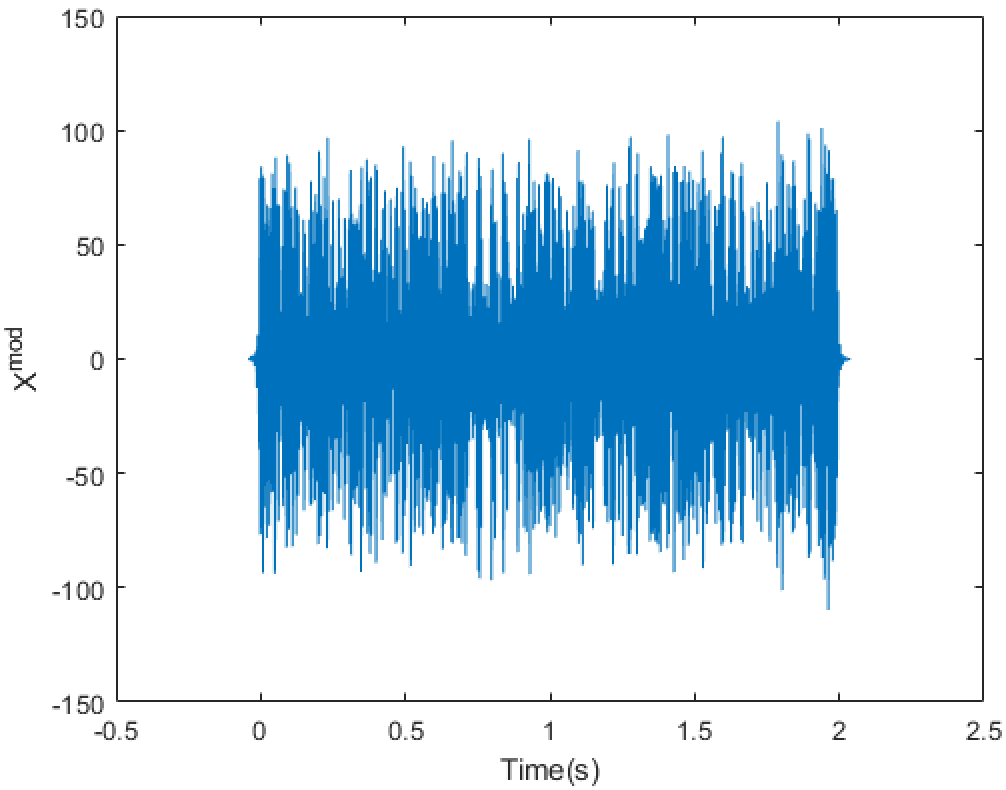
Σχήμα 1.4: Περιοδόγραμμα των διαμορφωμένων Xi, Xq

6. ,7.

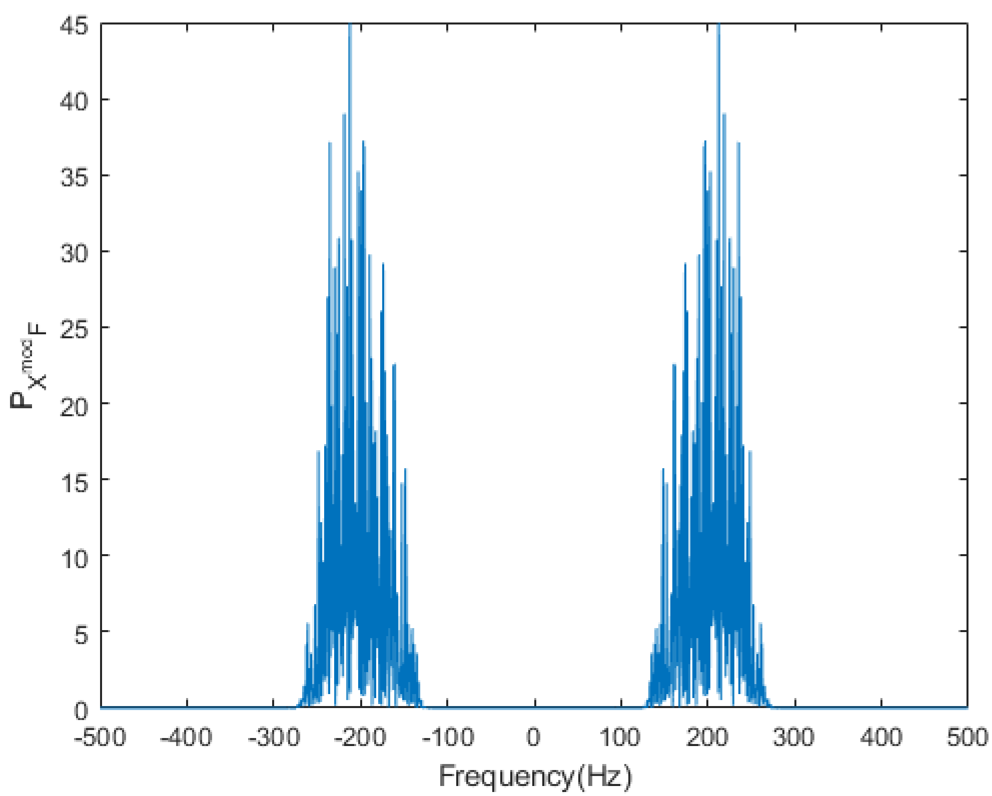
Στο μέρος αυτό προστίθενται τα διαμορφωμένα σήματα για να δημιουργήσουμε το σήμα εισόδου του καναλιού, το οποίο κανάλι θεωρείται ιδανικό για τις ανάγκες της προσομοίωσης. Το σήμα αυτό φτάνει σε πλάτη περίπου διπλάσια του πλάτους των δύο επιμέρους σημάτων-συνιστώσων. Επιπλέον, εφόσον ο μετασχηματισμός Fourier είναι γραμμικός μετασχηματισμός τα δύο επιμέρους φάσματα προστίθενται μεταξύ τους παράγοντας ένα νέο φάσμα με αυξημένη ισχύ στις συχνότητες που βρίσκονται γύρω από τις $\pm F_0$, όπως φαίνεται και στο περιοδόγραμμα της εισόδου. Οι παραπάνω παρατηρήσεις γίνονται ορατές στο Σχήμα 1.4. Ο κώδικας του μέρους αυτού φαίνεται στο Παράρτημα Κώδικα 1.4.

```
%%%%%%%%%%%%%%%%%%%Part 6 %%%%%%%%%%%%%%%%%%%%%%%%%%%
%Creating X_mod which is the sum of X_I_mod and X_Q_mod
 X_mod = X_I_mod + X_Q_mod ;

%Creating periodogram of X_mod
 t_total_X = time_conv_X_i(end)-time_conv_X_i(1);

 P_X_mod_F = (abs(fftshift(fft(X_mod,Nf)*Ts)).^2)./(t_total_X);
```

Παράρτημα Κώδικα 1.5



Σχήμα 1.5: Το άθροισμα των διαμορφωμένων X_I , X_Q



Σχήμα 1.6: Περιοδόγραμμα του αθροίσματος των διαμορφωμένων X_I , X_Q

Παρατήρηση: Το γεγονός ότι το κανάλι διέλευσης του σήματος εισόδου είναι ιδανικό σημαίνει ότι έχει κρουστική απόκριση δ(t), άρα δεν υπάρχει καμία μετατόπιση στο χρόνο στην κυματομορφή εξόδου αλλά και καμία αλλαγή στο μέτρο και τη φάση του φάσματος.

8.

Στην έξοδο του καναλιού προστίθεται λευκός Gaussian θόρυβος με διασπορά

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

Αυτός υλοποιείται με την ακόλουθη εντολή

Noise = sigma*randn(1,length(X_mod));

κατά την οποία το σύνολο των τυποποιημένων κανονικών τυχαίων μεταβλητών, που παράγει η randn, πολλαπλασιάζεται με την τυπική απόκλιση σ_W του θορύβου. Με βάση την ιδιότητα της διασποράς τυχαίων μεταβλητών προκύπτει ότι :

Έστω N_I τυποποιημένη κανονική τυχαία μεταβλητή με $\mu=0, \sigma^2=1$.

Ισχύει ότι $Var(\sigma_W N_I)=\sigma_W^2 Var(N_I)=\sigma_W^2$

```
%%%%%part 8 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sigma = sqrt((10 * (A^2))/(Ts* (10^(SNR_dB(j)/10))));%Variance of Gaussian Noise
%Creating Noise

  Noise = sigma*randn(1,length(X_mod));


%Creating Y which is the sequence of symbols affected by Noise

 Y = X_mod + Noise;
```

Παράρτημα Κώδικα 1.6

9.

Εν συνεχεία, το σήμα εξόδου διακλαδώνεται, στον δέκτη, στα δύο σήματα Y_I^{mod} και Y_Q^{mod} τα οποία πολλαπλασιάζονται με τους φορείς $\cos(2\pi F_0 t)$ και $-\sin(2\pi F_0 t)$ αντίστοιχα. Τα δύο επιμέρους σήματα παρουσιάζουν αύξηση στο πλάτος των κυματομορφών τους λόγω της προσθήκης λευκού Gaussian θορύβου. Επιπροσθέτως στα περιοδογράμματα των δύο σημάτων παρατηρείται εμφάνιση νέων, υψηλότερων συχνοτήτων αλλά και επαναφορά του κεντρικού λοβού λόγω της αποδιαμόρφωσης με τους φορείς. Όλα τα παραπάνω γίνονται εμφανή όταν υπολογιστεί το σήμα $Y_I^{demod}(t)$ (και ανάλογα το $Y_Q^{demod}(t)$) και το φάσμα του ως εξής:

Έχουμε $Y(t)=X_I^{mod}(t)+X_Q^{mod}(t)+N(t)$

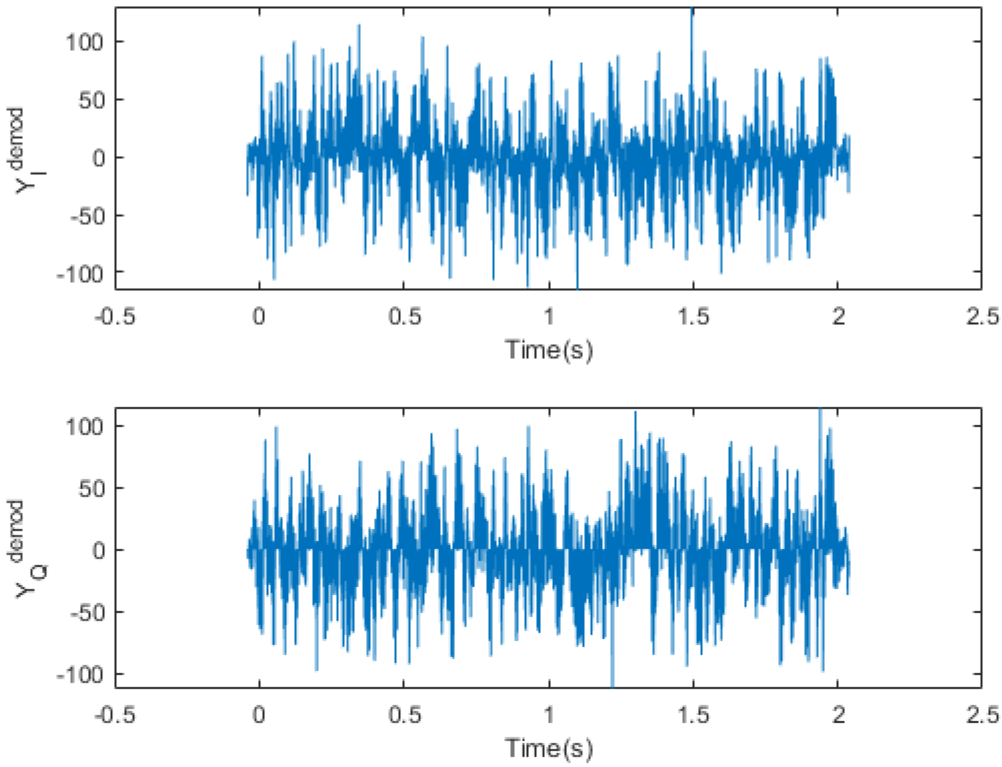
$$Y_I^{demod}(t)=Y(t)\cos(2\pi F_0 t)=2X_I(t)\cos^2(2\pi F_0 t)-2X_Q(t)\sin(2\pi F_0 t)\cos(2\pi F_0 t)+N(t)\cos(2\pi F_0 t)$$
$$Y_I^{demod}(t)=X_I(t)+X_I(t)\cos(2\pi 2F_0 t)-X_Q(t)\sin(2\pi 2F_0 t)+N(t)\cos(2\pi F_0 t)$$

$$F\left(Y_I^{demod}(t)\right)=X_I(F)+\frac{X_I(F-2\mathrm{F}_0)+X_I(F+2\mathrm{F}_0)}{2}-\frac{X_Q(F-2\mathrm{F}_0)+X_Q(F+2\mathrm{F}_0)}{2\mathrm{j}}+N\left(\delta(F-F_0)+\delta(F+F_0)\right)$$

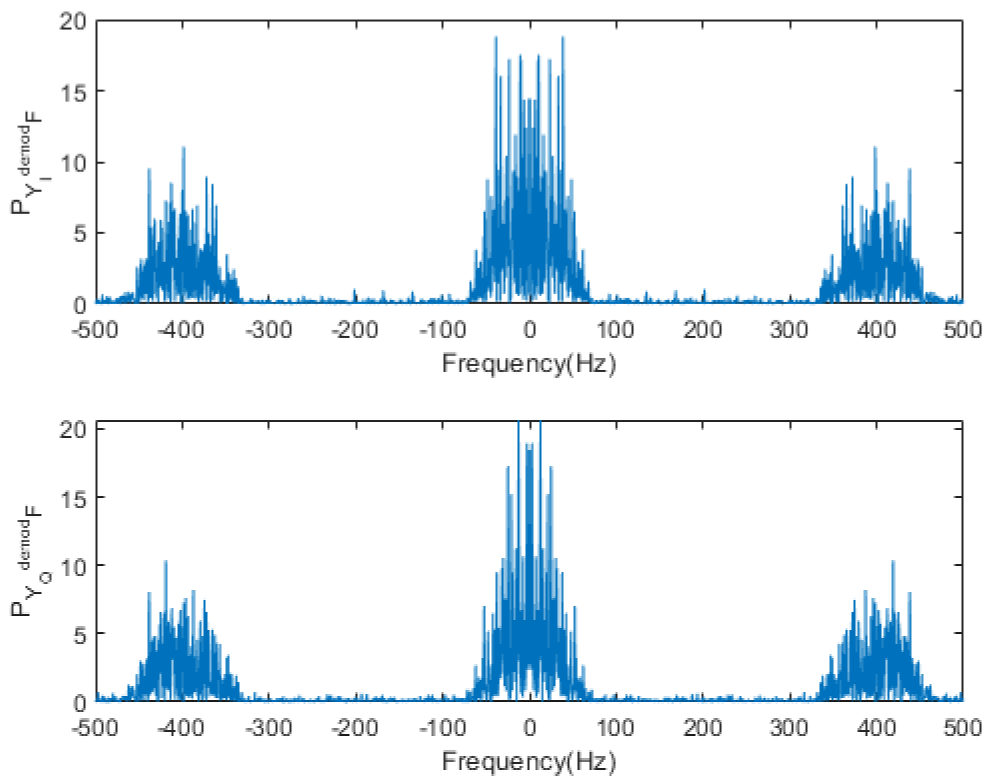
Ο κώδικας του μέρους αυτού καθώς και τα περιοδογράμματα και οι κυματομορφές φαίνονται στο Παράρτημα Κώδικα 1.7 και στα Σχήματα 1.6, 1.7 αντίστοιχα.

```
%%%%%part 9 %%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Multiplying Y with the appropriate carrier, creating Y_I_demod
Y_I_demod = Y.*cos(2*pi*F0.*time_conv_X_i);

%Multiplying Y with the appropriate carrier, creating Y_Q_demod
Y_Q_demod = Y.*(-sin(2*pi*F0.*time_conv_X_q));

t_total_X_i = time_conv_X_i(end) -  time_conv_X_i(1);

P_Y_I_demod_F = (abs(fftshift(fft(Y_I_demod,Nf)*Ts)).^2)./(t_total_X_i);

%Creating periodogram of Y_q_demod
t_total_X_q = time_conv_X_q(end) -  time_conv_X_q(1);

P_Y_Q_demod_F = (abs(fftshift(fft(Y_Q_demod,Nf)*Ts)).^2)./(t_total_X_q);
```

Παράρτημα Κώδικα 1.7



Σχήμα 1.7: Τα αποδιαμορφωμένα Y_I , Y_Q



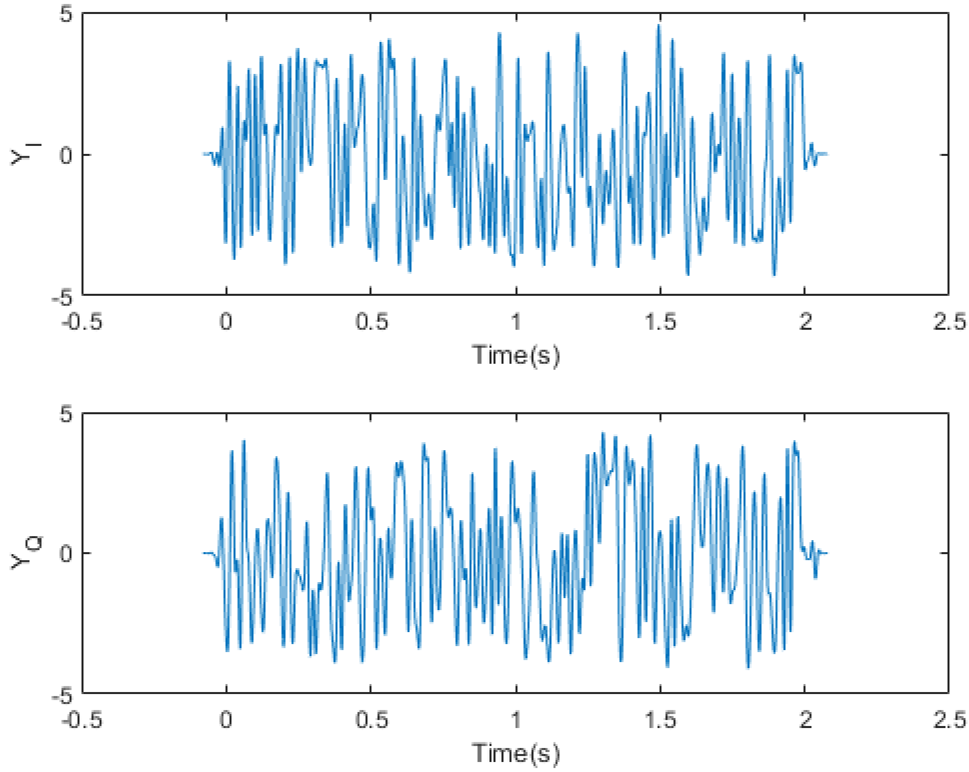
Σχήμα 1.8: Περιοδογράμματα των αποδιαμορφωμένων Y_I , Y_Q

10.

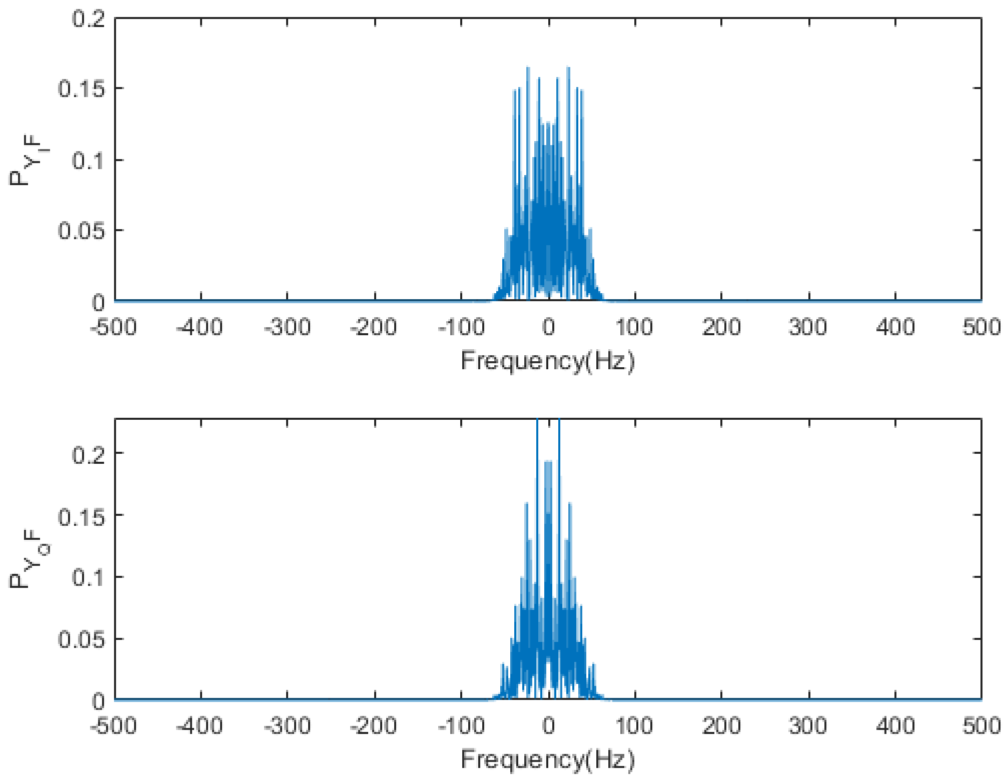
Οι κυματομορφές που προκύπτουν από την παραπάνω αποδιαμόρφωση εισέρχονται στα ίδια προσαρμοσμένα φίλτρα που χρησιμοποιήθηκαν και στο ερώτημα 4. Η απόκριση συχνότητας του SRRC παλμού που χρησιμοποιείται λειτουργεί ως χαμηλοπερατό φίλτρο που αποκόπτει τις συχνότητες μεγαλύτερες από ~75 Hz λόγω και του ότι επιλέχθηκε roll-off factor $\alpha=0.5$. Στην έξοδο του φίλτρου το σήμα που λαμβάνεται δεν περιέχει πια τις υψηλές συχνότητες που του προσδώθηκαν κατά τη διαμόρφωση. Ακόμα, στο φάσμα παρατηρείται η ύπαρξη μόνο του κεντρικού λοβού, πράγμα που επαληθέυει αυτά που αναφέρθηκαν παραπάνω. Τα σήματα στην έξοδο των φίλτρων καθως και τα περιοδογράμματα τους φαίνονται στα Σχήματα 1.11, 1.12, 1.13, 1.14. Ο κώδικας αυτής της διαδικασίας φαίνεται στο Παράρτημα Κώδικα 1.8.

```
%%%%%part 10%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Demodulating signal Y_I_mod
Y_I = Ts*conv(phi,Y_I_demod);

time_of_X_i = [time_conv_X_i(1)+t(1):DT:t(end)+time_conv_X_i(end)];

%Demodulating signal Y_Q_mod
Y_Q = Ts*conv(phi,Y_Q_demod);

time_of_X_q = [time_conv_X_q(1)+t(1):DT:t(end)+time_conv_X_q(end)];

%Creating periodogram of Y__i
t_total_conv_X_i = time_of_X_i(end) - time_of_X_i(1);

P_Y_i_F = (abs(fftshift(fft(Y_I,Nf)*Ts)).^2)./(t_total_conv_X_i);

%Creating periodogram of Y__q
t_total_conv_X_q = time_of_X_q(end) - time_of_X_q(1);

P_Y_q_F = (abs(fftshift(fft(Y_Q,Nf)*Ts)).^2)./(t_total_conv_X_q);
```

Παράρτημα Κώδικα 1.8



Σχήμα 1.9: Τα φιλτραρισμένα αποδιαμορφωμένα Y_I , Y_Q



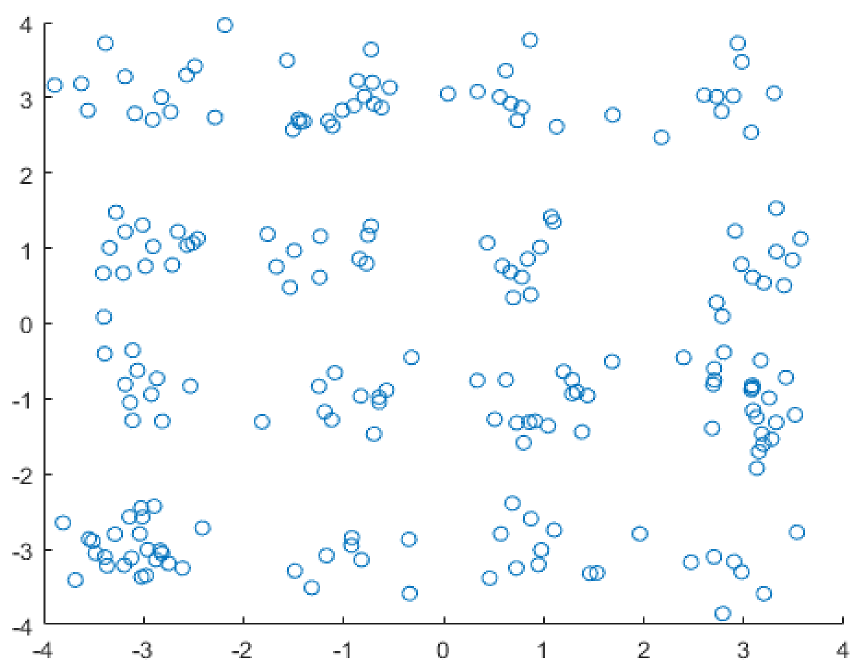
Σχήμα 1.10: Περιοδογράμματα των φιλτραρισμένων αποδιαμορφωμένων Y_I , Y_Q

11.

Στο μέρος αυτό γίνεται η δειγματοληψία των δύο φιλτραρισμένων σημάτων ανά τακτά χρονικά διαστήματα μεγέθους over, δηλαδή κάθε ακέραιο πολλαπλάσιο της περιόδου συμβόλου. Για να ληφθεί ακριβώς ο ίδιος αριθμός συμβόλων που στάλθηκαν πρέπει να "κοπούν" αμφίπλευρα οι δύο ουρές του σήματος οι οποίες περιέχουν μη χρήσιμη πληροφορία. Στο τέλος λαμβάνονται οι δύο ξεχωριστές ακολουθίες δειγμάτων (συνολικά 400 δείγματα) Y_{I,k} και Y_{Q,k} οι οποίες απεικονίζονται με χρήση της συνάρτησης scatter. Το διάγραμμα που προκύπτει φαίνεται στο Σχήμα 1.15 και το κομματι κώδικα στο Παράρτημα Κώδικα 1.9.

```
%%%%%part 11 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

y_i = Y_I(2*A_pulse*over+1:over:2*A_pulse*over +N*over);

y_q =  Y_Q(2*A_pulse*over+1:over:2*A_pulse*over +N*over);


figure(11);
scatter(y_i,y_q);
```

Παράρτημα Κώδικα 1.9



Σχήμα 1.11: Χαρτογράφηση των συμβόλων στο μιγαδικό

12. ,13.

Ύστερα, γράφεται συνάρτηση est_X = detect_4_PAM(Y,A) η οποία έχει σκοπό την εύρεση των συμβόλων που λαμβάνονται στην έξοδο χρησιμοποιώντας τον κανόνα του εγγύτερου γείτονα. Συνοπτικά, η συνάρτηση λαμβάνει σαν εισόδους μία ακολουθία δειγμάτων και το Α και στη συνέχεια βρίσκει τη μικρότερη απόσταση του εκάστοτε δείγματος από τα σύμβολα της 4-PAM απεικόνισης. Τέλος επιστρέφει το εγγύτερο σύμβολο. Η συνάρτηση εφαρμόζεται ξεχωριστά στις δύο ακολουθίες δειγμάτων. Οι ακολουθιές εισόδου του συστήματος συγκρίνονται με τις αποφάσεις που πάρθηκαν για τον εντοπισμό τυχών σφαλμάτων. Κάθε φορά που εντοπίζεται σφάλμα αυξάνεται κατά ένα ο μετρητής σφαλμάτων. Ο κώδικας που υλοποιεί τα παραπάνω βρίσκεται στο Παράρτημα Κώδικα 1.10.

```
%%%%%part 12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Estimating symbol of Sequence x_i
est_X_i = detect_4_PAM(y_i,A);

est_X_q = detect_4_PAM(y_q,A);


%%%%%part 13 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Estimating the propability of symbol error of Sequence X_I and X_Q
E_symbols_X_i = [est_X_i ~= X_I];

E_symbols_X_q = [est_X_q ~= X_Q];

E_symb = [E_symbols_X_i | E_symbols_X_q];

E_symbols = sum(E_symb);
```

Παράρτημα Κώδικα 1.10

```
function est_X = detect_4_PAM(Y,A)
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here
X = [-3*A,-A,A,3*A];

for i = 1 : length(Y)
  temp(1) = abs(Y(i) - X(1));

  temp(2) = abs(Y(i) - X(2));

  temp(3) = abs(Y(i) - X(3));

  temp(4) = abs(Y(i) - X(4));

     tmp = temp(1);

     est_X(i) = -3*A;

    for j = 2:4
         if(tmp > temp(j))
    tmp = temp(j);

   if(j == 2)
        est_X(i) = -A;

   end
   if(j == 3)
```

```
        est_X(i) = A;

   end
   if(j == 4)
        est_X(i) = 3*A;

   end
        end
    end
end
end
```

Παράρτημα Κώδικα 1.11

14., 15.

Το ερωτήματα αυτά επικεντρώνονται στον υπολογισμό σφαλμάτων bit. Για να επιτευχθεί ο έλεγχος για σφάλματα bit δημιουργείται συνάρτηση est_X = PAM_4_to_bits(X,A) η οποία λαμβάνει σαν είσοδο την ακολουθία συμβόλων και παράγει στην έξοδο μία ακολουθία από bits. Η αποκωδικοποίηση γίνεται με βάση την αντίστροφη απεικόνιση Gray, κατά την οποία κάθε σύμβολο μετατρέπεται σε μία δυάδα από bits. Συγκρίνοντας, την αρχική ακολουθία από bits με την εκτιμώμενη υπολογίζεται ο αριθμός σφαλμάτων bit. Λόγω της κωδικοποίησης Gray τα γειτονικά μεταξύ τους σύμβολα έχουν μόνο ένα bit διαφορετικό μειώνοντας έτσι την πιθανότητα να υπάρχει σφάλμα και στα δύο bits ενός συμβόλου. Αυτό επαληθεύεται και από τις μετρήσεις. Ο κώδικας του μέρους αυτού φαίνεται στο Παράρτημα Κώδικα 1.12.

```
%%%%%part 14 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Estimating the propability of bit error of Sequence b_i and b_q
 b_i = PAM_4_to_bits(est_X_i,A);

 b_q = PAM_4_to_bits(est_X_q,A);

%%%%%part 15 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 bits = [b_i ;b_q];

 E_b = [b ~= bits];

 E_bits = sum(E_b);
```

Παράρτημα Κώδικα 1.12

```
function est_X = PAM_4_to_bits(X,A)
%UNTITLED3 Summary of this function goes here
%   Detailed explanation goes here
 temp = zeros(2*length(X),1);

for i=1:1:length(X)
   if(X(i) == -A)
```

```
        temp(2*i-1) = 1;

        temp(2*i) = 1;

   elseif(X(i) == -3*A)
        temp(2*i-1) = 1;

        temp(2*i) = 0;

   elseif(X(i) == 3*A)
        temp(2*i-1) = 0;

        temp(2*i) = 0;

   elseif(X(i) == A)
        temp(2*i-1)= 0;

        temp(2*i) = 1;

   end
end
 est_X = temp;

end
```

Παράρτημα Κώδικα 1.13

Β ΕΡΩΤΗΜΑ

Έχοντας υλοποιήσει ένα ζωνοπερατό τηλεπικοινωνιακό σύστημα με διαμόρφωση 16-QAM στο πρώτο μέρος της άσκησης, υποθέτωντας όμως ότι το κανάλι είναι ιδανικό και ότι ο πομπός και ο δέκτης είναι τέλεια στγχρονισμένοι. Στο δεύτερο μέρος της άσκησης εκτιμάται η πιθανότητα σφάλματος συμβόλου και bit με τη μέθοδο Monte Carlo και συγκρίνονται με τις θεωρητικές στο αντίστοιχο κοινό διάγραμμα . Συγκεκριμένα υπολογίζεται η πιθανότητα σφάλματός συμβόλου και bit για διαφορετικό SNR($SNR_{dB}=[0:2:16]$) κάθε φορά για K = 1000 υλοποιήσεις του τηλεπικοινωνιακού συστήματος το οποίο έχουμε περιγράψει (στο Α ερώτημα). Ο υπολογισμός της πειραματικής εκτίμησης της πιθανότητας σφάλματος απόφασης συμβόλου και bit πραγματοποιήθηκε με τη χρήση των παρακάτω τύπων :

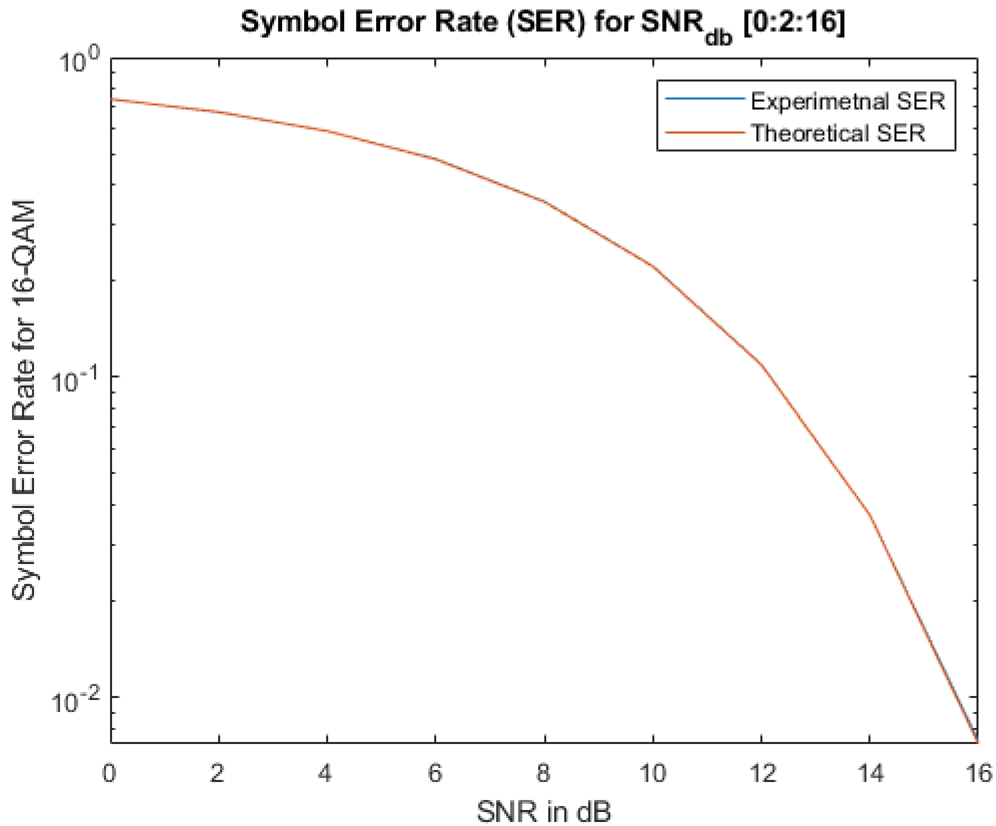
$$P(E_{symbol})=\frac{P(συνολικό\ πλήθος\ σφαλμάτων\ απόφασης\ συμβόλου)}{συνολικό\ πλήθος\ απεσταλμένων\ συμβόλων}$$

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

Για τον θεωρητικό υπολογισμό της πιθανότητας σφάλματος συμβόλου και bit χρησιμοποιήθηκαν οι παρακάτω τύποι :

- $P_{16-QAM}^{E}=3\cdot Q(\frac{A}{\sigma_N})-\frac{9}{4}\cdot Q(\frac{A}{\sigma_N})^2$,όπου $\sigma_N=\sqrt{\frac{Ts\cdot\sigma_W^2}{2}}$
- $P(E_{bit})\simeq\frac{P(E_{symbol})}{\log_2(M)}$, όπου M είναι ίσο με 16 (αφού 16-QAM)

Στις παρακάτω εικόνες απεικονίζονται η θεωρητική και πειραματική πιθανότητα σφάλματος συμβόλου και bit σαν συνάρτηση του SNR_{dB} .



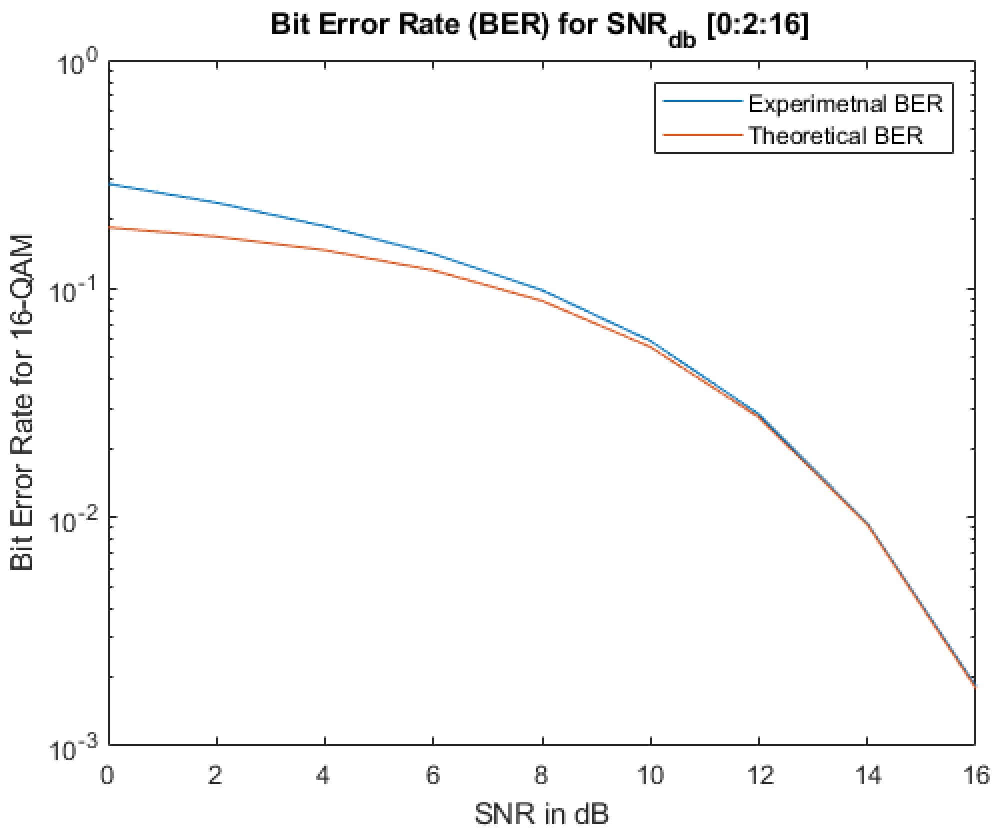
Σχήμα 2.1

Στις απεικονίσεις της θεωρητικής και πειραματικής πιθανότητας σφάλματος συμβόλου παρατηρείται η τάυτιση μεταξύ των δύο κυματομορφών. Ακόμη παρατηρείται ότι με την αύξηση του SNR_{dB} ,μειώνεται η πιθανότητα σφάλματος συμβόλου,το οποίο επιβεβαιώνεται απο τη θεωρία καθώς ο τύπος για την θεωρητική εκτίμηση της πιθανότητας σφάλματος είναι συνάρτηση της συνάρτησης Q, η οποία λαμβάνει σαν όρισμα $\frac{A}{\sigma_N}$.Γνωρίζουμε απο τη θεωρία ότι όσο αυξάνεται η τιμή η οποία δίνεται σαν όρισμα στη συνάρτηση Q και είναι θετική , τότε η τιμή της μειώνεται. Το σ_N είναι αντιστρόφος ανάλογο του SNR_{dB} από τη σχέση (3) , άρα η πιθανότητα σφάλματος θα είναι αντιστρόφος ανάλογη του SNR_{dB} από τη σχέση (3) που αποδείξαμε και απο την αντίστοιχη θεωρία για τη συνάρτηση Q.

$$\sigma_N^2 = \frac{Ts \cdot \sigma_W^2}{2} \quad (1)$$

Από $(1),(2) \Rightarrow \sigma_N^2 = \dfrac{Ts \cdot \dfrac{10 \cdot A^2}{Ts \cdot 10^{\frac{SNR_{dB}}{10}}}}{2} \Rightarrow \sigma_N^2 = \dfrac{10 \cdot A^2}{2 \cdot 10^{\frac{SNR_{dB}}{10}}} \Rightarrow \sigma_N^2 = \dfrac{10^{1-\frac{SNR_{dB}}{10}} \cdot A^2}{2} \Rightarrow$

$$\sigma_N = \frac{A \cdot 10^{\frac{1}{2}(1-\frac{SNR_{dB}}{10})}}{\sqrt{2}} \quad (3)$$



Σχήμα 2.2

Στο δίαγραμμα για την πιθανότητα σφάλματος bit, απεικόνιζονται η πειραματική και θεωρητική εκτίμηση τους. Παρατηρείται ότι με την αύξηση του SNR_{dB} , μειώνεται η πιθανότητα σφάλματος bit, το οποίο επιβεβαιώνεται απο τη θεωρία καθώς η η πιθανότητα σφάλματος bit είναι ανάλογη της πιθανότητας σφάλματος συμβόλου , κατά συνέπεια αντιστρόφος ανάλογη με το SNR_{dB} όπως αποδείξαμε στην προηγούμενη παράγραφο. Ακόμη στις απεικονίσεις της πειραματικής και θεωρητικής εκτίμησης σφάλματος bit παρατηρείται η απόκλιση της θεωρητικής με την πειραματική εκτίμηση μέχρι τη τιμή 12 της κλίμακας SNR_{dB} ,ενώ μετά σχεδόν ταυτίζονται .Αυτό συμβαίνει διότι ο θεωρητικός τύπος υπολογισμού της πιθανότητας σφάλματος bit υπολογίζεται προσεγγιστικά με την θεώρηση ότι όταν συμβεί σφάλμα απόφασης συμβόλου μόνο ένα από τα $\log_2(M)$ bits (M =16) που μεταφέρει το αντίστοιχο σύμβολο είναι εσφαλμένο. Ως εκ τούτου η υπόθεση του ακριβώς ενός λανθασμένου bit και κάθε εσφαλμένη απόφαση συμβόλου είναι προσεγγιστική, ειδικά για χαμηλά SNR_{dB} ,για αυτό όσο αυξάνεται το SNR_{dB} τόσο βελτιώνεται η προσέγγιση και προσομοιώνει καλύτερα την πειραματική πιθανότητα σφάλματος bit.

```
%------------------------PART B----------------------%
 P_E_symbols = E_symbols/N;

 P_E_bits = E_bits/(4*N);

 Average_E_symbols = Average_E_symbols + P_E_symbols;

 Average_E_bits = Average_E_bits + P_E_bits;

end
 Tot_E_symbol(j) = Average_E_symbols/K;

 Tot_E_bits(j) = Average_E_bits/K;

 sigma_N = sqrt((Ts*(sigma^2))/2);

 Theoretical_E_symbol = 3*Q((A/sigma_N)) - (9/4)*(Q((A/sigma_N)))^2;

 Theoretical_E_bit = Theoretical_E_symbol/(log2(M));

 Theoretical_Tot_E_symbol(j) = Theoretical_E_symbol;

 Theoretical_Tot_E_bits(j) = Theoretical_E_bit;

end
```

Παάρτημα Κώδικα 1.14

ΚΩΔΙΚΑΣ ΑΣΚΗΣΗΣ 3

```
clear all;
close all;
M = 16;%16-QAM
N = 200;

A = 1;

A_pulse = 4;

T = 0.01;

over = 10;

Ts = T/over ;

DT = Ts;

Fs = 1/Ts;

a = 0.5;

Nf = 2048;

F0 = 200;

K = 1000;

SNR_dB = [0:2:16];

F = [-Fs/2:Fs/Nf:Fs/2 - Fs/Nf];

Tot_E_symbol = zeros(1,length(SNR_dB));

Tot_E_bits = zeros(1,length(SNR_dB));

Theoretical_Tot_E_symbol = zeros(1,length(SNR_dB));

Theoretical_Tot_E_bits = zeros(1,length(SNR_dB));

for j = 1:1:length(SNR_dB)
Average_E_symbols  = 0;

Average_E_bits = 0;
```

```
for i = 1:1:K
%----------------------------PART A--------------------%
%%%%%%%%%%%%%%%%%%Part 1%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
b = (sign(randn(4*N,1)) + 1)/2;%Generating a sequence of bits
%%%%%%%%%%%%%%%%%%Part 2 and 3%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Seperating the sequence b of bits in two sequences of bits

b1 = b(1 :1 :2*N);

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


%Creating sequences of symbols from 4-PAM alphabet

X_I = bits_to_4_PAM(b1,A);

X_Q = bits_to_4_PAM(b2,A);


%

X_I_d = (1/Ts)*upsample(X_I,over);

time_of_X_I_d = [0:Ts:(N*T - Ts)];

X_Q_d = (1/Ts)*upsample(X_Q,over);

time_of_X_Q_d = [0:Ts:(N*T - Ts)];


%%%%%%%%%%%%%%%%%%%Part 4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Creating a SRRC pulse

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


%Modulating the two sequences X_I_d and X_Q_d

X_i = Ts*conv(phi,X_I_d);

time_conv_X_i = [time_of_X_I_d(1) + t(1): DT : time_of_X_I_d(end) + t(end)];

X_q = Ts*conv(phi,X_Q_d);

time_conv_X_q = [time_of_X_Q_d(1) + t(1): DT : time_of_X_Q_d(end) + t(end)];


%Plotting X_I and X_Q
figure(1);
subplot(2,1,1);
plot(time_conv_X_i,X_i);
xlabel('Time(s)');
ylabel('X_I');
title('Convolution of X_I and srrc pulse \phi');
%
subplot(2,1,2);
plot(time_conv_X_q,X_q);
xlabel('Time(s)');
ylabel('X_Q');
title('Convolution of X_Q and srrc pulse \phi');
%Creating periodograms of X_i and X_q
```

```
t_total_X_i = time_conv_X_i(end)-time_conv_X_i(1);

P_X_i_F = (abs(fftshift(fft(X_i,Nf)*Ts)).^2)./(t_total_X_i);

t_total_X_q = time_conv_X_q(end)-time_conv_X_q(1);

P_X_q_F = (abs(fftshift(fft(X_q,Nf)*Ts)).^2)./(t_total_X_q);
```

```
%Plotting periodograms
figure(2);
subplot(2,1,1);
plot(F,P_X_i_F);
xlabel('Frequency(Hz)');
ylabel('P{_{X_I}_F}');
title('Periodogram of X_I');
%
subplot(2,1,2);
plot(F,P_X_q_F);
xlabel('Frequency(Hz)');
ylabel('P{_{X_Q}_F}');
title('Periodogram of X_Q');
%%%%%%%%%%%%%%%%%%%Part 5 %%%%%%%%%%%%%%%%%%%%%%%%%%%
%Multiplying X_i and X_q with the appropriate carrier
```

```
X_I_mod = 2*X_i.*cos(2*pi*F0.*time_conv_X_i);

X_Q_mod = -2*X_q.*sin(2*pi*F0.*time_conv_X_q);
```

```
%Plotting X_I_mod and X_Q_mod
figure(3);
subplot(2,1,1);
plot(time_conv_X_i,X_I_mod);
xlabel('Time(s)');
ylabel('{X_I}^{mod}');
title('Modulation of X_I');
%
subplot(2,1,2);
plot(time_conv_X_q,X_Q_mod);
xlabel('Time(s)');
ylabel('{X_Q}^{mod}');
title('Modulation of X_Q');
%Creating periodograms of X_I_mod and X_Q_mod
```

```
P_X_i_mod_F = (abs(fftshift(fft(X_I_mod,Nf)*Ts)).^2)./(t_total_X_i);

P_X_q_mod_F = (abs(fftshift(fft(X_Q_mod,Nf)*Ts)).^2)./(t_total_X_q);
```

```
%Plotting periodograms of X_I_mod and X_Q_mod
figure(4);
subplot(2,1,1);
plot(F,P_X_i_mod_F);
xlabel('Frequency(Hz)');
ylabel('P_{{X_I}^{mod}}_F');
title('Periodogram of modulated X_I');
%
subplot(2,1,2);
plot(F,P_X_q_mod_F);
xlabel('Frequency(Hz)');
ylabel('P_{{X_Q}^{mod}}_F');
title('Periodogram of modulated X_Q');
```

```
%%%%%%%%%%%%%%%%%%%Part 6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Creating X_mod which is the sum of X_I_mod and X_Q_mod
 X_mod = X_I_mod + X_Q_mod ;

%Creating periodogram of X_mod
 t_total_X = time_conv_X_i(end)-time_conv_X_i(1);

 P_X_mod_F = (abs(fftshift(fft(X_mod,Nf)*Ts)).^2)./(t_total_X);

%Plotting X_mod and its periodogram
figure(5);
plot(time_conv_X_i,X_mod);
xlabel('Time(s)');
ylabel('X^{mod}');
title('Modulated X');
%
figure(6);
plot(F,P_X_mod_F);
xlabel('Frequency(Hz)');
ylabel('P_{X^{mod}}_F');
title('Periodogram of modulated X');
%%%%%part 8 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sigma = sqrt((10 * (A^2))/(Ts* (10^(SNR_dB(j)/10))));%Variance of Gaussian Noise
%Creating Noise
  Noise = sigma*randn(1,length(X_mod));

%Creating Y which is the sequence of symbols affected by Noise
 Y = X_mod + Noise;

%%%%%part 9 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Multiplying Y with the appropriate carrier, creating Y_I_demod
 Y_I_demod = Y.*cos(2*pi*F0.*time_conv_X_i);

%Multiplying Y with the appropriate carrier, creating Y_Q_demod
 Y_Q_demod = Y.*(-sin(2*pi*F0.*time_conv_X_q));

%Plotting Y_i_demod and its periodogram
figure(7);
subplot(2,1,1);
plot(time_conv_X_i,Y_I_demod);
xlabel('Time(s)');
ylabel('Y{_I}^{demod}');
title('Demodulation of Y_I');
%
subplot(2,1,2);
plot(time_conv_X_q,Y_Q_demod);
xlabel('Time(s)');
ylabel('Y{_Q}^{demod}');
title('Demodulation of Y_Q');
%Creating periodogram of Y_i_demod
 t_total_X_i = time_conv_X_i(end) -  time_conv_X_i(1);
```

```
P_Y_I_demod_F = (abs(fftshift(fft(Y_I_demod,Nf)*Ts)).^2)./(t_total_X_i);

%Creating periodogram of Y_q_demod
t_total_X_q = time_conv_X_q(end) -  time_conv_X_q(1);

P_Y_Q_demod_F = (abs(fftshift(fft(Y_Q_demod,Nf)*Ts)).^2)./(t_total_X_q);

%Plotting Y_q_demod and its periodogram
figure(8);
subplot(2,1,1);
plot(F,P_Y_I_demod_F);
xlabel('Frequency(Hz)');
ylabel('P_{Y{_I}^{demod}}_F');
title('Periodogram of demodulated X_I');
%
subplot(2,1,2);
plot(F,P_Y_Q_demod_F);
xlabel('Frequency(Hz)');
ylabel('P_{Y{_Q}^{demod}}_F');
title('Periodogram of demodulated X_Q');
%%%%%part 10%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Demodulating signal Y_I_mod
Y_I = Ts*conv(phi,Y_I_demod);

time_of_X_i = [time_conv_X_i(1)+t(1):DT:t(end)+time_conv_X_i(end)];

%Demodulating signal Y_Q_mod
Y_Q = Ts*conv(phi,Y_Q_demod);

time_of_X_q = [time_conv_X_q(1)+t(1):DT:t(end)+time_conv_X_q(end)];

figure(9);
subplot(2,1,1);
plot(time_of_X_i,Y_I);
xlabel('Time(s)');
ylabel('Y_I');
title('Convolution of demodulated Y_I and srrc pulse \phi');
%
subplot(2,1,2);
plot(time_of_X_q,Y_Q);
xlabel('Time(s)');
ylabel('Y_Q');
title('Convolution of demodulated Y_Q and srrc pulse \phi');
%Creating periodogram of Y__i
t_total_conv_X_i = time_of_X_i(end) - time_of_X_i(1);

P_Y_i_F = (abs(fftshift(fft(Y_I,Nf)*Ts)).^2)./(t_total_conv_X_i);

%Creating periodogram of Y__q
t_total_conv_X_q = time_of_X_q(end) - time_of_X_q(1);

P_Y_q_F = (abs(fftshift(fft(Y_Q,Nf)*Ts)).^2)./(t_total_conv_X_q);
```

```
figure(10);
subplot(2,1,1);
plot(F,P_Y_i_F);
xlabel('Frequency(Hz)');
ylabel('P_{Y{_I}}_F');
title('Periodogram of Y_I');
%
subplot(2,1,2);
plot(F,P_Y_q_F);
xlabel('Frequency(Hz)');
ylabel('P_{Y{_Q}}_F');
title('Periodogram of Y_Q');
%%%%%part 11 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 y_i = Y_I(2*A_pulse*over+1:over:2*A_pulse*over +N*over);

 y_q =  Y_Q(2*A_pulse*over+1:over:2*A_pulse*over +N*over);


figure(11);
scatter(y_i,y_q);
%%%%%part 12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Estimating symbol of Sequence x_i

 est_X_i = detect_4_PAM(y_i,A);

 est_X_q = detect_4_PAM(y_q,A);


%%%%%part 13 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Estimating the propability of symbol error of Sequence X_I and X_Q

 E_symbols_X_i = [est_X_i ~= X_I];

 E_symbols_X_q = [est_X_q ~= X_Q];

 E_symb = [E_symbols_X_i | E_symbols_X_q];

 E_symbols = sum(E_symb);


%%%%%part 14 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Estimating the propability of bit error of Sequence b_i and b_q

 b_i = PAM_4_to_bits(est_X_i,A);

 b_q = PAM_4_to_bits(est_X_q,A);


%%%%%part 15 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 bits = [b_i ;b_q];

 E_b = [b ~= bits];

 E_bits = sum(E_b);


%------------------------PART B-----------------------%
%%%%%%%%%%%%%%%%%%%%%Part 1%%%%%%%%%%%%%%%%%%%%%%%%%

 P_E_symbols = E_symbols/N;
```

```
P_E_bits = E_bits/(4*N);

Average_E_symbols = Average_E_symbols + P_E_symbols;

Average_E_bits = Average_E_bits + P_E_bits;

end
Tot_E_symbol(j) = Average_E_symbols/K;

Tot_E_bits(j) = Average_E_bits/K;

sigma_N = sqrt((Ts*(sigma^2))/2);

Theoretical_E_symbol = 3*Q((A/sigma_N)) - (9/4)*(Q((A/sigma_N)))^2;

Theoretical_E_bit = Theoretical_E_symbol/(log2(M));

Theoretical_Tot_E_symbol(j) = Theoretical_E_symbol;

Theoretical_Tot_E_bits(j) = Theoretical_E_bit;

end
%%%%%%%%%%%%%%%%%%%%%Part 2%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(12);

semilogy(SNR_dB,Tot_E_symbol);

hold on;
semilogy(SNR_dB,Theoretical_Tot_E_symbol);

legend('Experimetnal SER','Theoretical SER');
xlabel('SNR in dB');
ylabel('Symbol Error Rate for 16-QAM');
title('Symbol Error Rate (SER) for SNR_{db} [0:2:16]');
hold off;
figure(13);

semilogy(SNR_dB,Tot_E_bits);

hold on;
semilogy(SNR_dB,Theoretical_Tot_E_bits);

legend('Experimetnal BER','Theoretical BER');
xlabel('SNR in dB');
ylabel('Bit Error Rate for 16-QAM');
title('Bit Error Rate (BER) for SNR_{db} [0:2:16]');
hold off;
```

BITS TO 4-PAM

```
function X = bits_to_4_PAM(b,A)
%UNTITLED2 Summary of this function goes here
%   Detailed explanation goes here
for i = 1 : 2: length(b)
    if(b(i) == 0 && b(i+1) == 0)

         X((i + 1)/2) = 3*A;


    elseif(b(i) == 0 && b(i+1) == 1)

         X((i + 1)/2) = A;


    elseif(b(i) == 1 && b(i+1) == 1)

          X((i + 1)/2) = -A;


    elseif(b(i) == 1 && b(i+1) == 0)

         X((i + 1)/2) = -3*A;


  end
end
end
```

DETECT 4-PAM

```
function est_X = detect_4_PAM(Y,A)
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here

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


for i = 1 : length(Y)

  temp(1) = abs(Y(i) - X(1));

  temp(2) = abs(Y(i) - X(2));

  temp(3) = abs(Y(i) - X(3));

  temp(4) = abs(Y(i) - X(4));

     tmp = temp(1);

     est_X(i) = -3*A;


    for j = 2:4
        if(tmp > temp(j))

    tmp = temp(j);


   if(j == 2)
```

```
        est_X(i) = -A;

    end
    if(j == 3)
        est_X(i) = A;

    end
    if(j == 4)
        est_X(i) = 3*A;

    end
          end
     end
end
end
```

4-PAM TO BITS

```
function est_X = PAM_4_to_bits(X,A)
 temp = zeros(2*length(X),1);

for i=1:1:length(X)
    if(X(i) == -A)
         temp(2*i-1) = 1;

         temp(2*i) = 1;

    elseif(X(i) == -3*A)
         temp(2*i-1) = 1;

         temp(2*i) = 0;

    elseif(X(i) == 3*A)
         temp(2*i-1) = 0;

         temp(2*i) = 0;

    elseif(X(i) == A)
         temp(2*i-1)= 0;

         temp(2*i) = 1;

   end

end
 est_X = temp;end
```