ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

<u>ΑΜ:</u> 03120233 <u>Ονοματεπώνυμο:</u> Χρήστος Ηλιακόπουλος

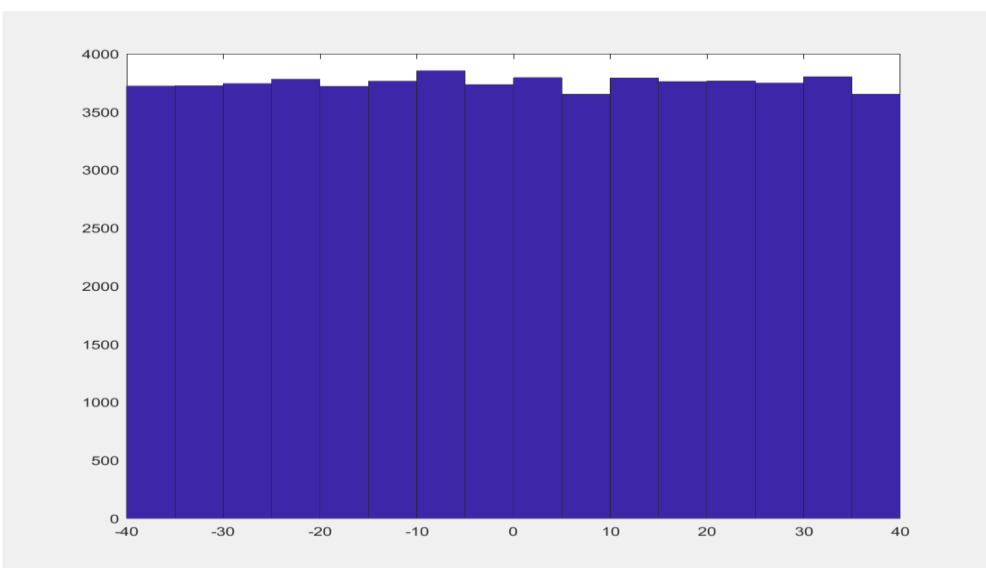
3^{η} Εργαστηριακή Άσκηση

<u>Μέρος 1:</u>

Α) Να τροποποιηθεί ο κώδικας, ώστε τα L στοιχεία του διανύσματος x στην εντολή 14 να λαμβάνουν τιμές από το σύνολο {±d/2, ±3d/2, ±5d/2... }, όπου η απόσταση d των σημείων θα δίνεται ως παράμετρος. Χρησιμοποιώντας για τη συνέχεια την τιμή d=5, να επαληθεύσετε, με υπολογισμό και προβολή σχετικού ιστογράμματος, ότι τα στοιχεία του διανύσματος x ακολουθούν πράγματι την ομοιόμορφη κατανομή. Χρησιμοποιήστε k=mod(nnnnn,2)+3, όπου nnnnn το τελευταίο 5-ψηφίο τμήμα του αριθμού μητρώου σας.

Ο κώδικας που χρησιμοποιήθηκε:

```
%%metablhtes
Nsymb = 40000;
k = 4;
L = 2^k;
d = 5;
D = d/2;
%%metablhtes

x=(2*floor(L*rand(1,Nsymb))-L+1)*(d/2);
Px=(D^2)*(L^2-1)/3; % θεωρητική ισχύς σήματος
sum(x.^2)/length(x); % μετρούμενη ισχύς σήματος (για επαλήθευση)
A = [(-L+1)*D:5:(L-1)*D];
hist(x,A)
```



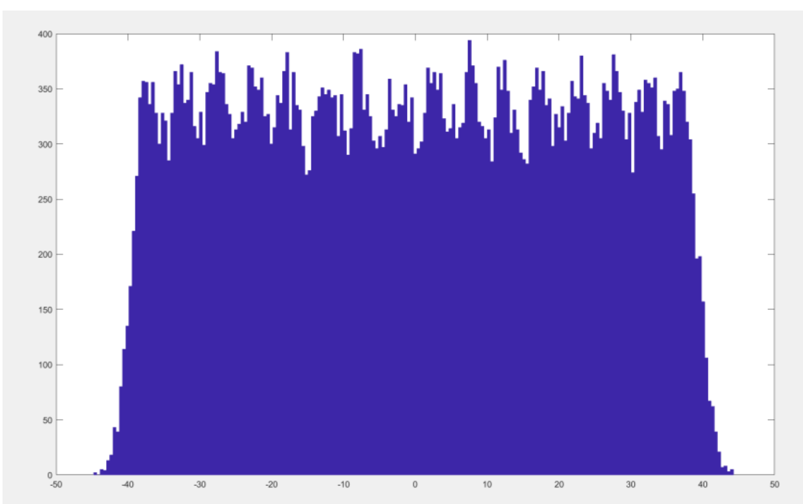
Από το ιστόγραμμα που προκύπτει παρατηρούμε ότι τα στοιχεία του διανύσματος x πράγματι ακολουθούν την ομοιόμορφη κατανομή. Στην συγκεκριμένη περίπτωση επειδή αναγκαζόμαστε να πολλαπλασιάσουμε με d/2 = 2.5 τα αντίστοιχα όρια αυτής θα είναι από -16*2.5 = -40 έως +16*2.5 = 40. Επομένως τα στοιχεία ακολουθούν πράγματι την ομοιόμορφη κατανομή από -37.5 έως 37.5 (±2.5, ±7.5, ±12.5, ±17.5, ±22.5, ±27.5, ±32.5, ±37.5) που είναι και τα τελευταία πλάτη μας. Επιπλέον, αν εκτελέσουμε τις εντολές Px=(D^2)*(L^2-1)/3; και sum(x.^2)/length(x); θα παρατηρήσουμε ότι τα αποτελέσματά της θεωρητικής με την μετρούμενη ισχύς είναι πράγματι αρκετά κοντά.

```
Px =

  531.2500

>> sum(x.^2)/length(x)

ans =

  529.4475
```

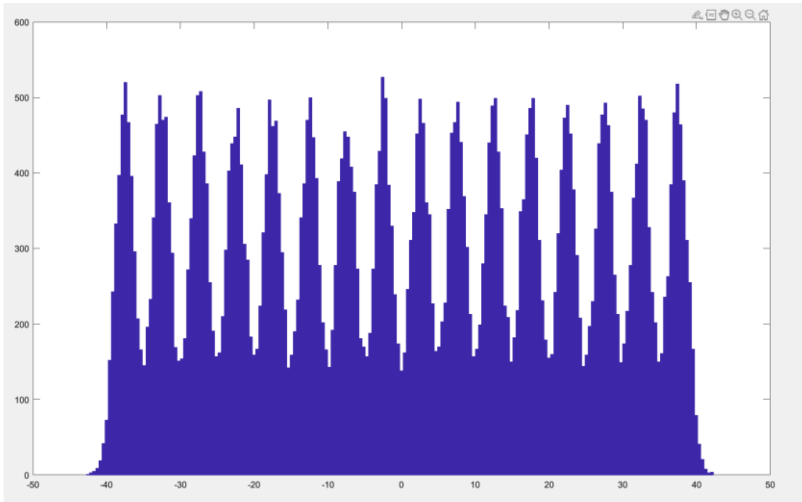
Προφανώς και έγινε η αντίστοιχη τροποποίηση της θεωρητικής τιμής της ισχύος από τον τύπο:

$$S = E_{av} / T = A^2 \frac{L^2 - 1}{3},$$

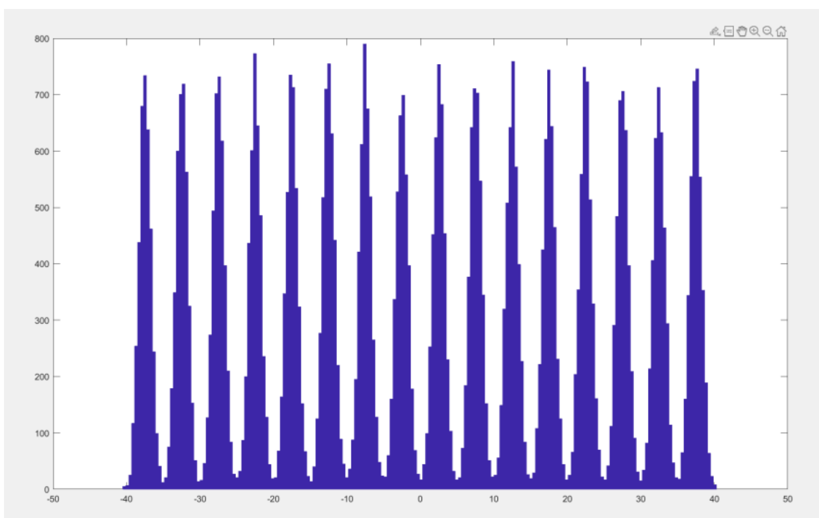
B) Για k = 4; Nsymb(M) = 60000; nsamp = 20; και EbNo=12; Έχουμε το παρακάτω διάγραμμα



Για k = 4; Nsymb(M) = 60000; nsamp = 20; και EbNo=16; Έχουμε το παρακάτω διάγραμμα



Για k = 4; Nsymb(M) = 60000; nsamp = 20; και EbNo=20; Έχουμε το παρακάτω διάγραμμα



Με την αύξηση του λόγου Eb/No, ο οποίος είναι σε dB, παρατηρούμε ότι στο ιστόγραμμα του z υπάρχει αραίωση χωρίς την ύπαρξη επικαλύψεων. Δημιουργούνται κενά στα σύμβολα επιτρέποντας έτσι στις τιμές του z να μαζεύονται αρκετά καθαρά γύρω από τις τιμές των πλατών που έχουμε ορίσει. Γίνεται, λοιπόν, με αυτόν τον τρόπο πιο εύκολη η απόφαση για το ποια είναι η πραγματική τιμή που έλαβε ο δέκτης, δηλαδή, ποια είναι όντως η πραγματική τιμή που στάλθηκε, μειώνοντας με αυτόν τον τρόπο το σφάλμα. Είναι σαφές, πως με την αύξηση της ισχύς του συμβόλου που εκπέμπεται, ο θόρυβος θα επηρεάσει όλο και λιγότερο την αρχική εκπομπή του κάθε συμβόλου, επιτρέποντάς μας να έχουμε καλύτερη λήψη αποφάσεων στο τελικό στάδιο καταγραφής από τον δέκτη.

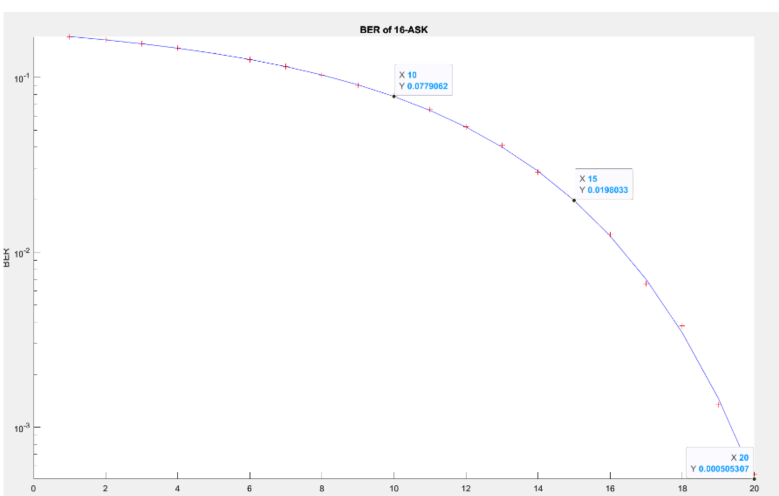
Γ) Η εντολή reshape τοποθετεί σε πίνακα με διαστάσεις “(length(ynoisy)/nsamp) x (nsamp)” το διάνυσμα του ynoisy, δηλαδή δημιουργεί ένα πίνακα διαστάσεων (16,50000). Η εντολή 22 δίνει τη συσχέτιση του y με το προσαρμοσμένο φίλτρο (matched). Η μεταβλητή matched είναι διαστάσεων (1,16) και η y (16 50000).

Η εντολή 20 reshape τοποθετεί σε πίνακα με διαστάσεις (MxN) όπου M = nsamp = 20 και όπου N = length(ynoisy)/nsamp). Δημιουργεί με αυτόν τον τρόπου πίνακα διαστάσεων (20, 60000) στον οποίο θα τοποθετήσει το διάνυσμα του ynoisy, το οποίο είναι το σήμα y με την προσθήκη θορύβου. Η εντολή 22 με την μεταβλητή matched – η οποία είναι ένας πίνακας διαστάσεων 1x20 γεμισμένος με άσσους- μας δίνει την συσχέτιση του y με το προσαρμοσμένο μας φίλτρο. Η μεταβλητή matched είναι μονοδιάστατος πίνακας που περιέχει άσσους με διαστάσεις 1x20. Η μεταβλητή x είναι μονοδιάστατος πίνακας με διαστάσεις 1x60000. Στην εντολή 17 το y είναι μονοδιάστατος πίνακας με διαστάσεις 1x1200000, ενώ στην εντολή 20 είναι δισδιάστατος πίνακας με διαστάσεις 20x60000. Η μεταβλητή z είναι μονοδιάστατος πίνακας με διαστάσεις 1x60000. Όλες οι προαναφερθείσες μεταβλητές είναι τύπου double.

Δ) Ελέγχονται όλα τα σύμβολα στον πίνακα "z" που έχει λάβει ο δέκτης μέσω του βρόχου. Οι τιμές του z που παράγονται συγκρίνονται με μία από τις στάθμες, δηλαδή τις τιμές -37.5, -32.5, … 32.5, 37.5. Για κάθε στοιχείο αφαιρούμε από τον πίνακα με τις τιμές των πλατών των συμβόλων και παίρνουμε την απόλυτη τιμή αυτών και αναζητούμε τη μικρότερη από αυτές προκειμένου να βρούμε τη θέση του συμβόλου που στάλθηκε. Η πιο κοντινή τιμή αποδίδεται στο z και λαμβάνεται απόφαση για το ποια τιμή του x πραγματικά στάλθηκε.

<u>Μερος 2:</u>

Α)Υλοποιήθηκε πρόγραμμα, το οποίο βρίσκεται στο <>, που πραγματοποιεί παρόμοια λειτουργία με τη συνάρτηση bertool σύμφωνα με τις οδηγίες που δόθηκαν στην εκφώνηση. Παρατίθεται το διάγραμμα μαζί με τις τιμές του BER για EbNo = 10, 15, 20.



EbNo = 10 μας δίνει BER = 77.9062×10^{-3}.

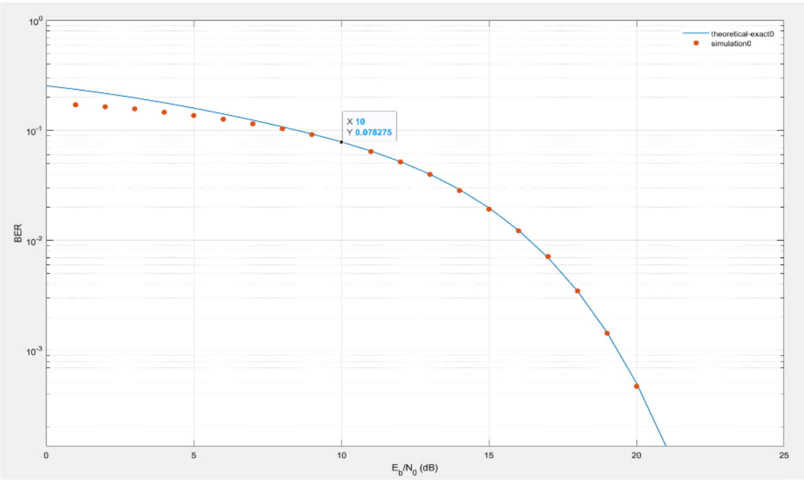
EbNo = 15 μας δίνει BER = 19.8033×10^{-3}.

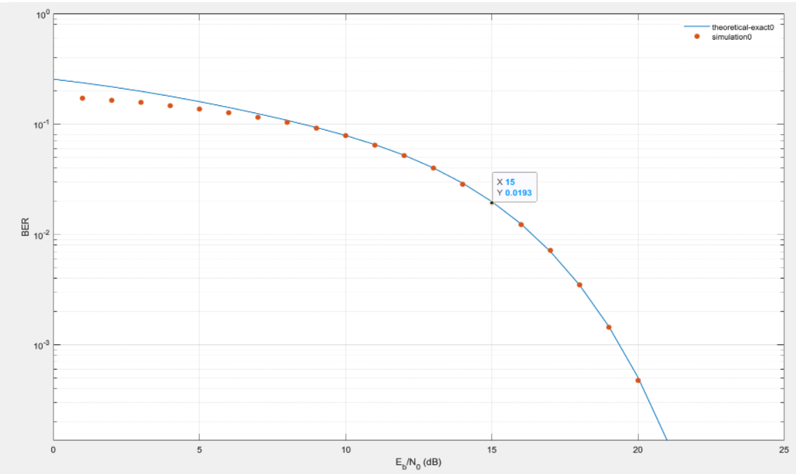
EbNo = 10 μας δίνει BER = 50.5307×10^{-5}.

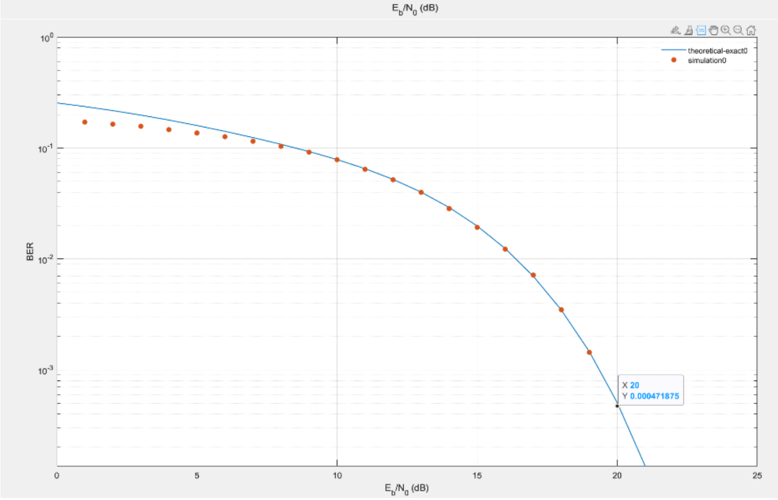
Ο κώδικας του προγράμματος είναι ο παρακάτω:

```
    % Clear all variables, close all open figures, and
clear command window
    clear all;
    close all;
    clc

    % Define system parameters
    k = 4; % Number of bits per symbol, calculated as
20233 mod 2 + 3
    M = 40000; % Number of symbols
    L = 2^k; % Number of amplitude levels
    nsamp = 16; % Number of samples per symbol
    numBits = k * M; % Total number of bits

    % Define Eb/No values and calculate theoretical
bit error rate (BER)
    EbNo = 1:20;
    % also changing EbNo from dB to Watt.
    Pe = ((L-1)/L) * erfc(sqrt(3*k/(L^2-1) *
(10.^(EbNo/10))));
    theoreticalBER = Pe / k;

    % Simulate BER for each Eb/No value
    for i = 1:20
        symbolErrors(i) = ask_errors(k, M, nsamp, i);
        simulatedBER(i) = symbolErrors(i) / numBits;
    end

    % Plot the simulated and theoretical BER
performance
    figure(3);
    hold on;
    set(gca, 'yscale', 'log');
    semilogy(EbNo, simulatedBER, 'r+');
    semilogy(EbNo, theoreticalBER, 'b-');
    title('BER of 16-ASK');
    xlabel('EbNo dB');
    ylabel('BER');
    hold off;
```

B)Στη συνέχεια παρουσιάζεται για τα ίδια δεδομένα με το προηγούμενο πρόγραμμα το αποτέλεσμα της bertool







EbNo = 10 μας δίνει BER = 78.2750×10^{-3}.

EbNo = 15 μας δίνει BER = 19.3000×10^{-3}.

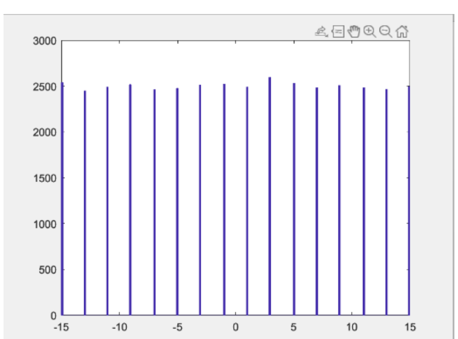
EbNo = 10 μας δίνει BER = 47.1875×10^{-5}.

Παρατηρούμε ότι τα αποτελέσματα είναι αρκετά κοντά μεταξύ τους.

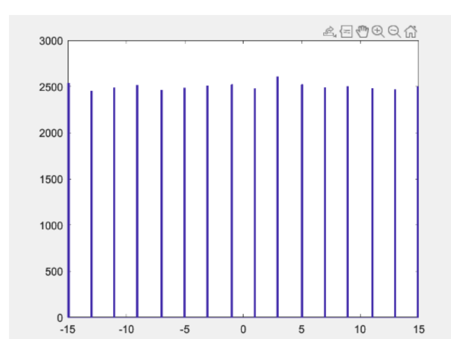
Μέρος 3:

Α)Για το συγκεκριμένο ερώτημα υλοποιήθηκε ένα ενιαίο σκριπτ, το οποίο περιέχει και τους δύο τρόπους κατασκευής του y (με έτοιμη συνάρτηση και με ολόκληρη τη διαδικασία. Παρατίθενται στη συνέχεια τα ιστογράμματα του z και στις δύο περιπτώσεις και γίνεται αντιληπτό ότι είναι αρκετά όμοια μεταξύ τους.

hist(z1,200)



hist(z2, 200)



Στην συνέχεια ελέγχουμε τα σφάλματα και στις δύο περιπτώσεις και παρατηρούμε ότι είναι αρκετά κοντά μεταξύ τους.

errors1	87
errors2	80

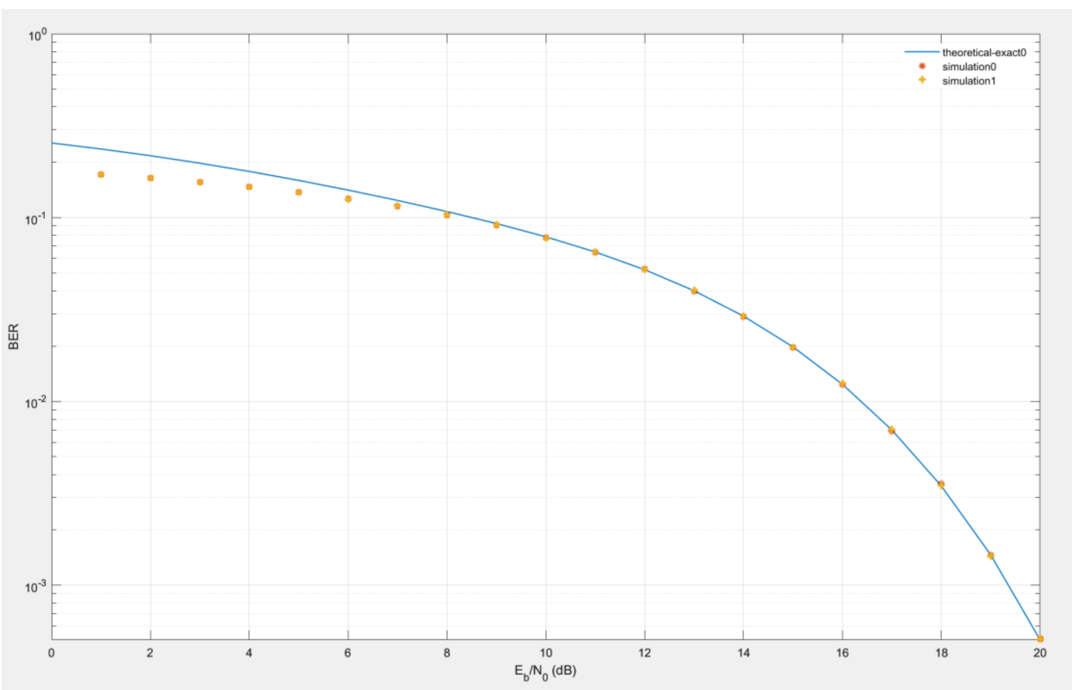
Κώδικας του ενιαίου προγράμματος

```
clc;
clear all;
close all;

k=4;
L=2^k;
M=40000;
nsamp=20;
EbNo=20;
SNR=EbNo-10*log10(nsamp/2/k); % SNR ανά δείγμα σήματος
% Διάνυσμα τυχαίων ακεραίων {±1, ±3, ... ±(L-1)}. Να επαληθευτεί
x=(2*floor(L*rand(1,M))-L+1);
Px=(L^2-1)/3; % θεωρητική ισχύς σήματος
y1=rectpulse(x,nsamp);
n=wgn(1,length(y1),10*log10(Px)-SNR);
y1noisy=y1+n; % θορυβώδες σήμα
y1=reshape(y1noisy,nsamp,length(y1noisy)/nsamp);
matched=ones(1,nsamp);
z1=matched*y1/nsamp;

hist(z1, 200)

l=[-L+1:2:L-1];
 for i=1:length(z1)
 [m,j]=min(abs(l-z1(i)));
 z1(i)=l(j);
 end
 err1=not(x==z1);
 errors1=sum(err1);

 h=ones(1,nsamp); h=h/sqrt(h*h'); % κρουστική απόκριση φίλτρου
 % πομπού (ορθογωνικός παλμός μοναδιαίας ενέργειας)
 y2=upsample(x,nsamp); % μετατροπή στο πυκνό πλέγμα
 y2=conv(y2,h); % το προς εκπομπή σήμα
 y2=y2(1:M*nsamp); % περικόπτεται η ουρά που αφήνει η συνέλιξη
 y2noisy=awgn(y2,SNR,'measured'); % θορυβώδες σήμα
 for i=1:nsamp matched(i)=h(end-i+1); end
 yrx=conv(y2noisy,matched);
 z2 = yrx(nsamp:nsamp:M*nsamp); % Υποδειγμάτιση -- στο τέλος
 % κάθε περιόδου T
 hist(z2,200)

 l=[-L+1:2:L-1];
 for i=1:length(z2)
 [m,j]=min(abs(l-z2(i)));
 z2(i)=l(j);
 end
 err2=not(x==z2);
 errors2=sum(err2);
```

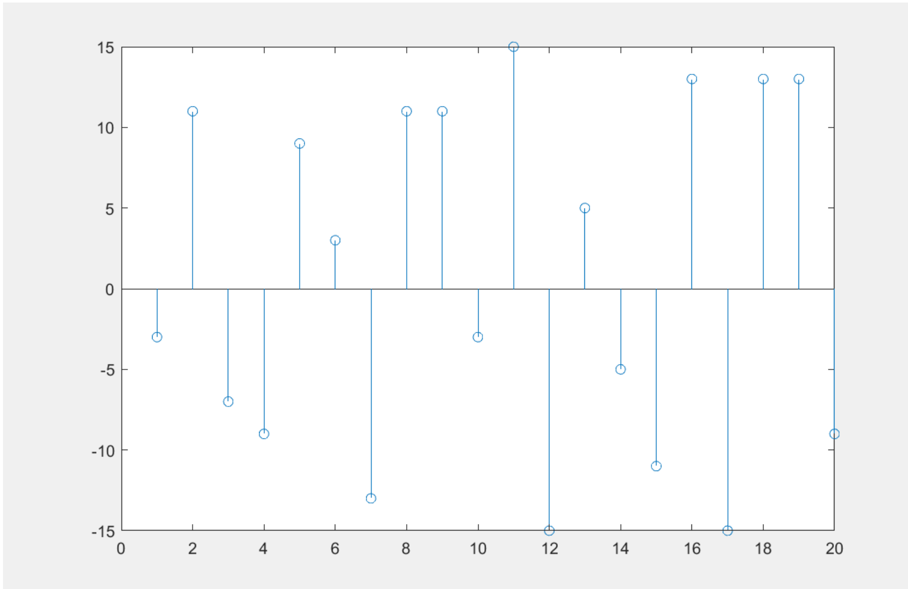
Ενώ αν κρατήσουμε και τις δύο συναρτήσεις σαν προγράμματα θα λάβουμε παραπλήσιες απαντήσεις στα σφάλματα για τιμές k=4, M=40000, nsamp=20, EbNo=20.

```
>> errors1=sum(err1)

errors1 =

    73

>> errors2=sum(err2)

errors2 =

    67

>>
```

Τέλος για να επαληθεύσουμε ότι ο παραπάνω κώδικας παράγει τα ίδια αποτελέσματα μπορούμε να συγκρίνουμε και τις δύο συναρτήσεις στο bertool. Δημιουργώντας την ask_ber_func_new.m η οποία κάνει χρήση της ask_errors_new.m που είναι ουσιαστικά η νέα συνάρτηση παρατηρούμε ότι ο νέος κώδικας παράγει ίδια αποτελέσματα

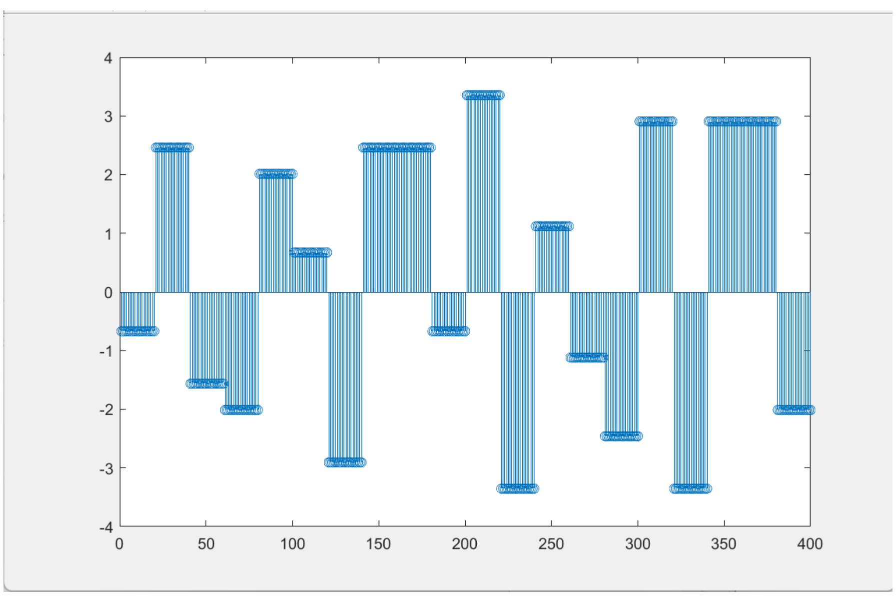


B) Το x για τα πρώτα 20 πλάτη θα είναι:



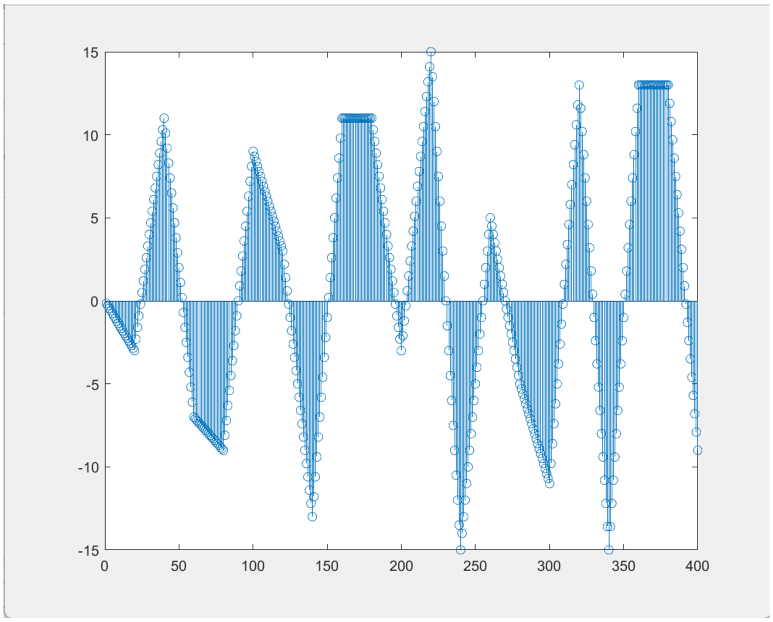
Χρησιμοποιώντας την εντολή stem(x(1:20)) βλέπουμε τα πρώτα 20 δείγματα του διανύσματος χ στα αντίστοιχα πλάτη τους (τα οποία είναι από ±1, ±3,… ±15 για k = 4). Τα νούμερα που παίρνουμε είναι τυχαία από την συνάρτηση rand και στρογγυλοποιημένα από την συνάρτηση floor.

Οι τετραγωνικοί παλμοί(που ο κάθε ένας περιέχει nsamp στοιχεία μέσα) που θα δημιουργηθούν στο y για τα πρώτα 20 του δείγματα:



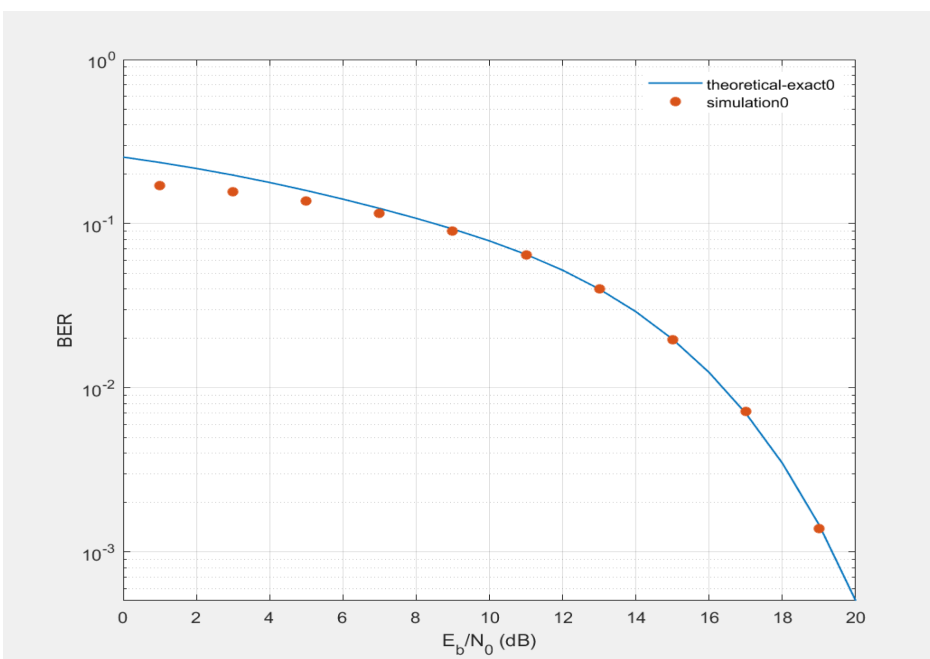
Για το σήμα y, στο συγκεκριμένο διάγραμμα, φαίνεται η εξέλιξή του στο χρόνο στα πρώτα 20*nsamp δείγματά του. Το y το δημιουργήσαμε από την συνέλιξη του σήματος x με τον τετραγωνικό παλμό h, που αποτελείται από nsamp δείγματα. Ουσιαστικά, είναι η προσπάθεια να αναπαραστήσουμε το χ σαν παλμό.

Τα πρώτα 20 δείγματα του yrx:



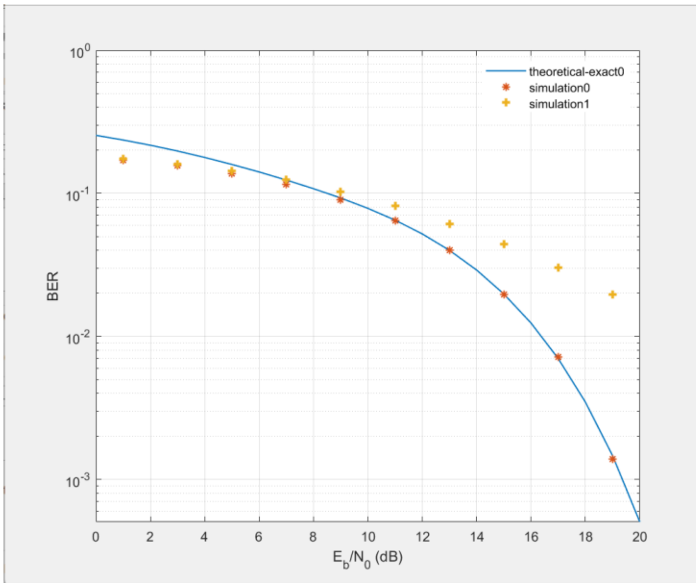
Στο τελευταίο σήμα έχουμε απεικόνιση τα 20*nsamp δείγματα του, που προκύπτουν από τη συνέλιξη με το προσαρμοσμένο φίλτρο h(T-t), με T το μήκος του διανύσματος h. Στο διάγραμμα έχουμε τις τιμές που θα έχει ο παλμός που θα σταλθεί για κάθε χρονική στιγμή.

γ)



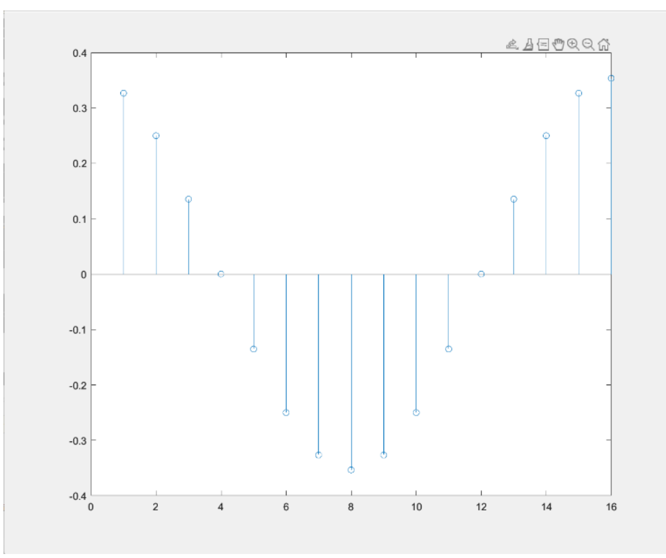
Παρόλη την αλλαγή του παλμού δεν φαίνεται να υπάρχει κάποια σημαντική μεταβολή στα αποτελέσματα μας.

Με την αλλαγή της συνθήκης matched = h παρατηρούμε ότι στην προσομοίωση αρχίζουν και δημιουργούνται σημαντικές αποκλίσεις, λόγω της αλλαγής στο είδος του σήματος που χρησιμοποιούμε πλέον στη συνέλιξη.

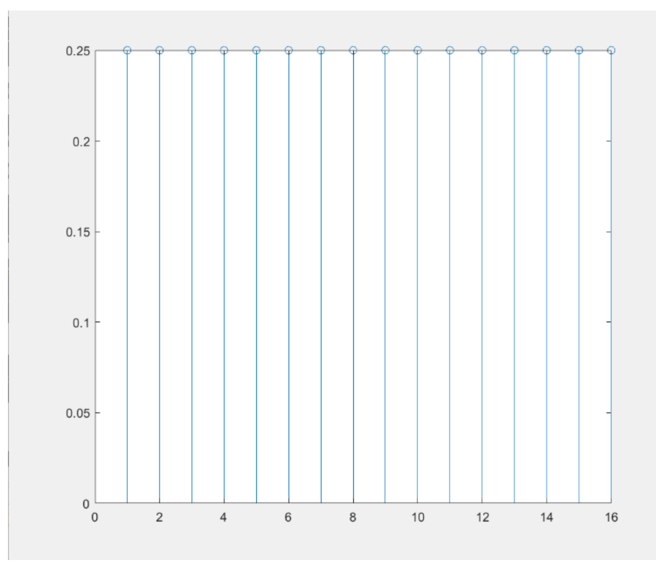


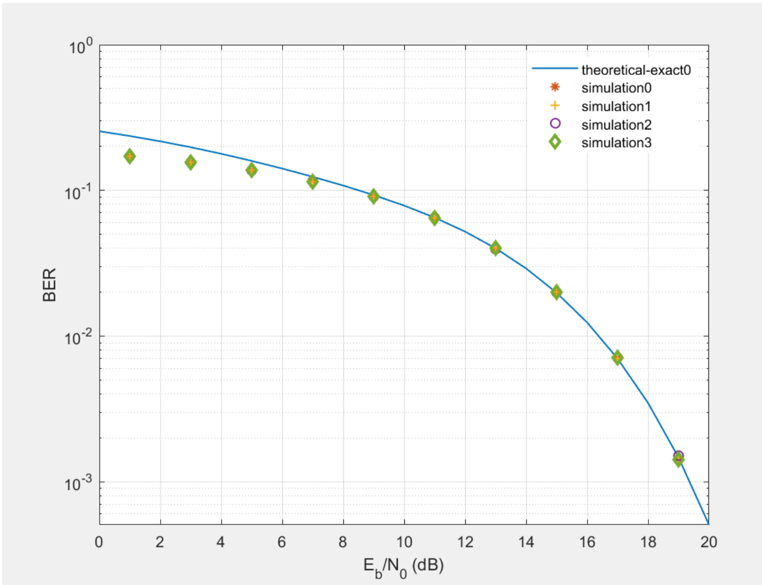
Αφού έχουμε επαναφέρει την εντολή matched(i) = h(i) έχουμε τα εξείς ιστογράμματα για nsamp = 16:

stem(h(1:16)) για h συνημιτονοειδές



stem(h(1:16))για h τετραγωνικό παλμό





Στη συνέχεια για τον τετραγωνικό παλμό σε κάθε τιμή nsamp θα έχουμε το ίδιο αποτέλεσμα στην προσομοίωση. Παρατίθεται προσομοίωση για nsamp = 8, 16, 32, 64 με τετραγωνικό παλμό

Στην συνημιτονοειδή περίπτωση δεν έχουμε τα αντίστοιχα αποτελέσματα. Ξεκινάμε την προσωμοίωση από nsamp = 8 και βλέπουμε ότι έχουμε πολύ μεγάλη απόκλιση από το θεωρητικό κομμάτι, λόγω του μικρού αριθμού δειγμάτων. Όσο μεγαλώνει ο αριθμός των δειγμάτων και αυξάνεται το nsamp, τόσο πιο κοντά πλησιάζει η προσομοίωση στο θεωρητικό κομμάτι.