

ΨΗΦΙΑΚΕΣ ΕΠΙΚΟΙΝΩΝΙΕΣ Ι Ακ. Έτος 2022-23

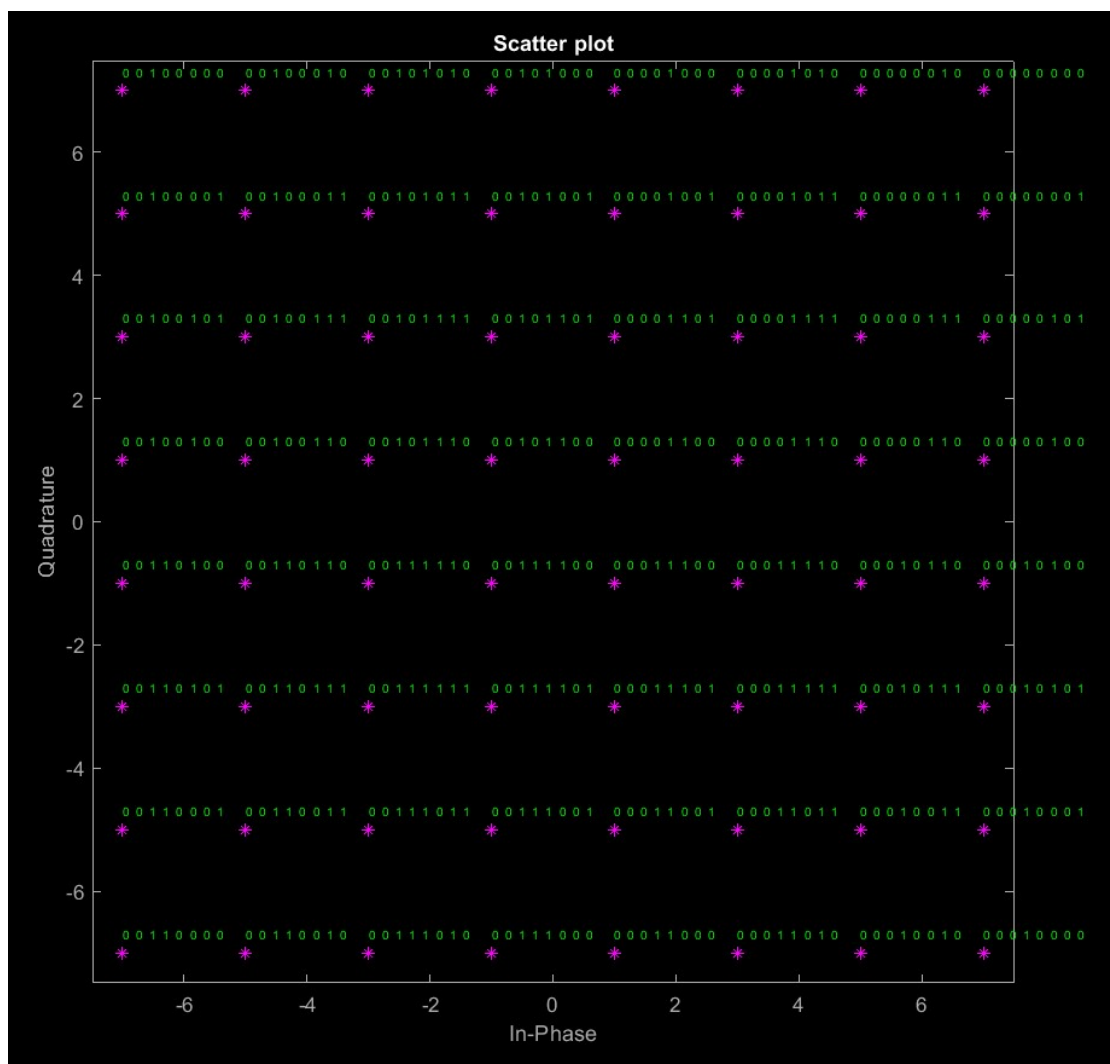
Κουράκου Σοφία 03120869

Εργαστηριακή Άσκηση 5 QAM-PSK

(1) Σχεδιασμός αστερισμού 64-QAM πλήρους ορθογωνικού πλέγματος, με κωδικοποίηση Gray

Δημιουργούμε τη συνάρτηση `mapping=qam_scatterplot_gray(k)` , με την οποία θα σχεδιάσουμε σηματικό αστερισμό 64-QAM πλήρους ορθογωνικού πλέγματος, με σημειωμένες τις δυαδικές λέξεις δίπλα σε κάθε σημείο του, με κωδικοποίηση Gray.

Προκύπτει το εξής :



- Προέκυψε ο αστερισμός 64QAM σαν ορθογωνικό πλέγμα. Αυτό που παρατηρούμε είναι ότι πράγματι στην κωδικοποίηση Gray δυο διαδοχικές τιμές διαφέρουν κατά 1 bit

Ο κώδικας που χρησιμοποιήθηκε είναι ο ακόλουθος :

```

1 % ergasthrio 5
2 function mapping=qam_scatterplot_gray(k)
3 % k is the number of bits per symbol
4 k=6; M=2^k; L=sqrt(M);
5 l=log2(L) % αριθμός bit ανά συνιστώσα
6
7 % Διάνυσμα mapping για την κωδικοποίηση Gray M-QAM
8 core=[1+i;1-i;-1+i;-1-i];
9 mapping=core;
10 if(l>1)
11 for j=1:l-1
12 mapping=mapping+j*2*core(1);
13 mapping=[mapping;conj(mapping)];
14 mapping=[mapping;-conj(mapping)];
15 end
16 end
17 scatterplot(mapping,1,0,'m*');
18 for n=1:length(mapping)
19 text(real(mapping(n))-0.0,imag(mapping(n))+0.3,num2str(de2bi(n-1,k,'left-msb')), ...
20 'FontSize', 6,'Color','g');
21 % δυαδική αναπαράσταση
22 end

```

- (2) Με ζωνοπερατό διάυλο 8.75-11.25 MHz, για να εκπέμψουμε με ρυθμό 12 Mbps , θα επιλέξουμε σύστημα 64-QAM πλήρους ορθογωνικού πλέγματος και σηματοδοσίας Nyquist. Για να εκμεταλλευτούμε το εύρος ζώνης επιλέγουμε roll-off factor $\alpha=0.25$. Το baud rate θα είναι στα 2 Mhz.

Οι υπολογισμοί των παραπάνω :

Έχουμε : εύρος ζώνης $W = 11.25 - 8.75 = 2.5 \text{ MHz}$
 και ρυθμός μετάδοσης $R = 12 \text{ Mbps}$

Ισχύουν οι σχέσεις : $\frac{R}{\log_2 M} = \frac{1}{T}$ ① και $W = \frac{1}{T}(1+\alpha)$ ② (Εύρος ζώνης W για W με Nyquist)

Επομένως $\log_2 M \geq \frac{R}{W}(1+\alpha) \Rightarrow \log_2 M \geq \frac{12}{2.5}(1+\alpha) \Rightarrow \log_2 M \geq 4.8(1+\alpha)$

Το μικρότερο M για το οποίο ισχύει η τελευταία σχέση είναι $M=64$

② $\Rightarrow \frac{1}{T} = \frac{W}{1+\alpha}$

① $\xrightarrow{\text{②}} \frac{R}{\log_2 M} = \frac{W}{1+\alpha} \Rightarrow \alpha = 0.25$

Στη συνέχεια, αναπαριστούμε έναν πομπό και έναν δέκτη και σχεδιάζουμε τη θεωρητική και πειραματική καμπύλη $P_b \rightarrow E_b/N_0$.

Η συχνότητα δειγματοληψίας είναι αρκετά υψηλή, με $nsamp=32$ δείγματα ανά T , ώστε να αποφεύγουμε το φαινόμενο της αλίευσης (aliasing), που προκαλεί σφάλματα στην αναπαράσταση των σημάτων διαμόρφωσης-αποδιαμόρφωσης.

Τροποποιούμε τον κώδικα 5.2 του βιβλίου και ο φτιάχνουμε τη συνάρτηση `ask_qam` :

```
13 function errors=ask_qam(M,Nsymb,nsamp,EbNo,rolloff)
14 % M=64; Nsymb=30000; nsamp=32; rolloff=0.25; %EbNo=18;
15 L=sqrt(M); l=log2(L); k=log2(M);
16 % Gray κωδικοποίηση
17 core=[1+1j;1-1j;-1+1j;-1-1j];
18 mapping=core;
19 if(l>1)
20 for j=1:l-1
21 mapping=mapping+j*2*core(1);
22 mapping=[mapping;conj(mapping)];
23 mapping=[mapping;-conj(mapping)];
24 end
25 end;
26 % Random bits--- symbols
27 x=floor(2*rand(k*Nsymb,1)); %τυχαία δυαδική ακολουθία
28 xsymbols=bi2de(reshape(x,k,length(x)/k).','left-msb');
29 y=[];
30 for i=1:length(xsymbols)
31 y=[y mapping(xsymbols(i)+1)];
32 end
33 % Παραμετροι του φίλτρου
34 delay=10;
35 filterorder = delay*nsamp*2;
36 rNyquist= rcosine(1,nsamp,'fir/sqrt',rolloff,delay);
37 % Transmitter (εκπεμπόμενο σήμα)
38 ytx=upsample(y,nsamp);
39 ytx = conv(ytx,rNyquist);
40 % Υπολογισμός & σχεδιασμός φάσματος
41 figure(1); pwelch(real(ytx),[],[],[],nsamp);
42 R=6000000;
43 Fs=Nsymb/k*nsamp;
44 fc=2; %carrier frequency baud rate
45 m=(1:length(ytx));
46 s=real(ytx.*exp(1j*2*pi*fc*m/nsamp)); % shift to desired frequency band
47 %Υπολογισμός & σχεδιασμός φάσματος
48 figure(2); pwelch(s,[],[],[],Fs); % COMMENT FOR BERTOOL
49 % Noise: Προσθήκη λευκού γκαουσιανού θορύβου
50 SNR=EbNo-10*log10(nsamp/2/k);
51 Ps=10*log10(s*s'/length(s)); %signal power (db)
52 Pn=Ps-SNR; %noise power (db)
53 n=sqrt(10^(Pn/10))*randn(1,length(ytx));
54 snoisy=s+n;
55
56 clear ytx xsymbols s n; % για εξοικονόμηση μνήμης
```

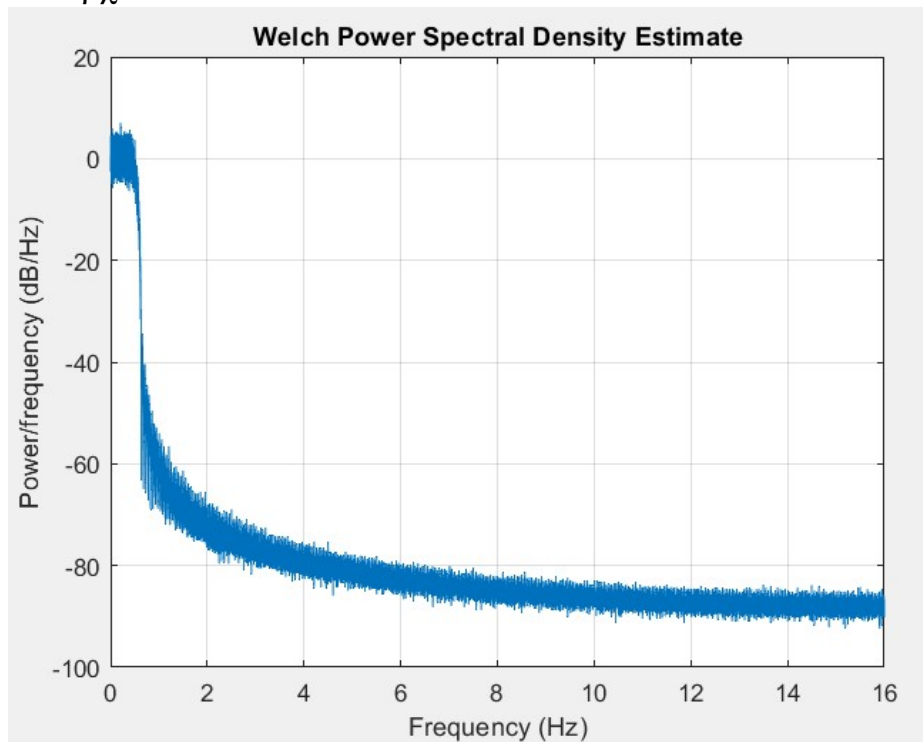
```

57 % Receiver (ΔΕΚΤΗΣ)
58 yrx=2*snoisy.*exp(-1j*2*pi*fc*m/nsamp); clear s; %shift to 0 frequency
59 % Κανονικά ακολουθεί βαθυπερατό φίλτρο.
60 % Όμως στην περίπτωση μορφοποίησης Nyquist δεν χρειάζεται,
61 % αφού το προσαρμοσμένο φίλτρο (Nyquist) είναι ταυτόχρονα
62 % και ένα καλό βαθυπερατό φίλτρο.
63 yrx = conv(yrx,rNyquist); %filter
64 yrx = yrx(2*nsamp*delay+1:end-2*nsamp*delay);
65 figure(3); pwelch(real(yrx),[],[],[],Fs); % COMMENT FOR BERTOOL
66 yrx = downsample(yrx,nsamp); % Υποδειγμάτιση στο πλέγμα nT.
67 % yrx = yrx(2*delay+(1:length(y))); % περικοπή άκρων συνέλιξης.
68 % Error counting
69 yi=real(yrx); yq=imag(yrx);
70 xrx=[];
71 q=[-L+1:2:L-1];
72 for n=1:length(yrx) % επιλογή πλησιέστερου σημείου
73 [m,j]=min(abs(q-yi(n)));
74 yi(n)=q(j);
75 [m,j]=min(abs(q-yq(n)));
76 yq(n)=q(j);
77 end
78 errors=sum(not(y==(yi+1j*yq)));

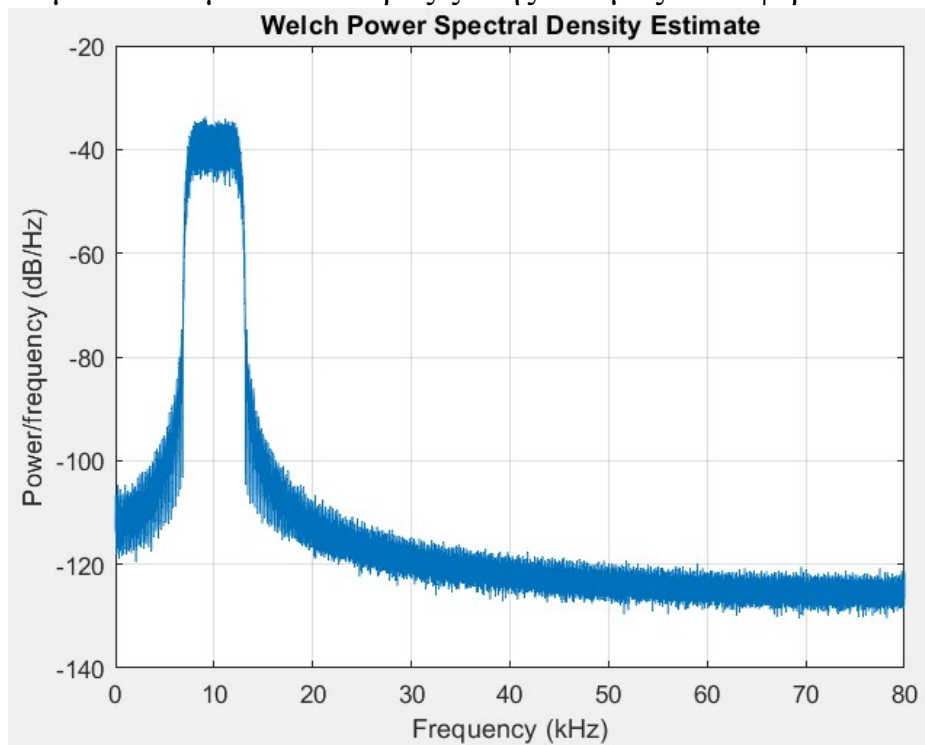
```

Από τις **pwelch** προκύπτουν τα ακόλουθα διαγράμματα για το φάσμα του σήματος σε διάφορα στάδια:

A. Αρχικό

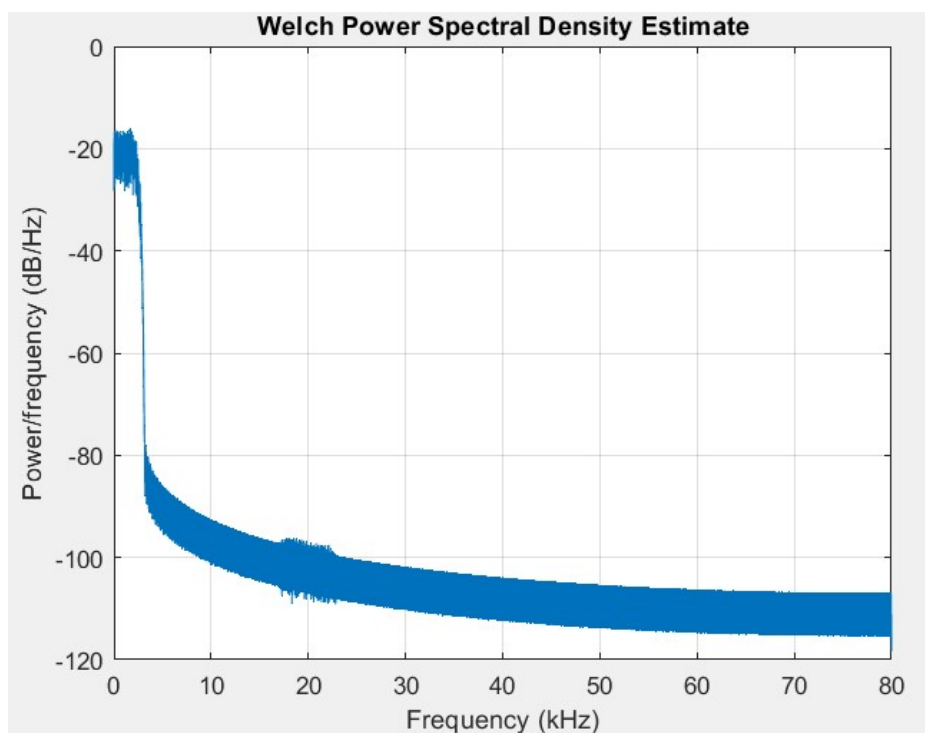


Β. μετατοπισμένο στο εύρος ζώνης που μας ενδιαφέρει



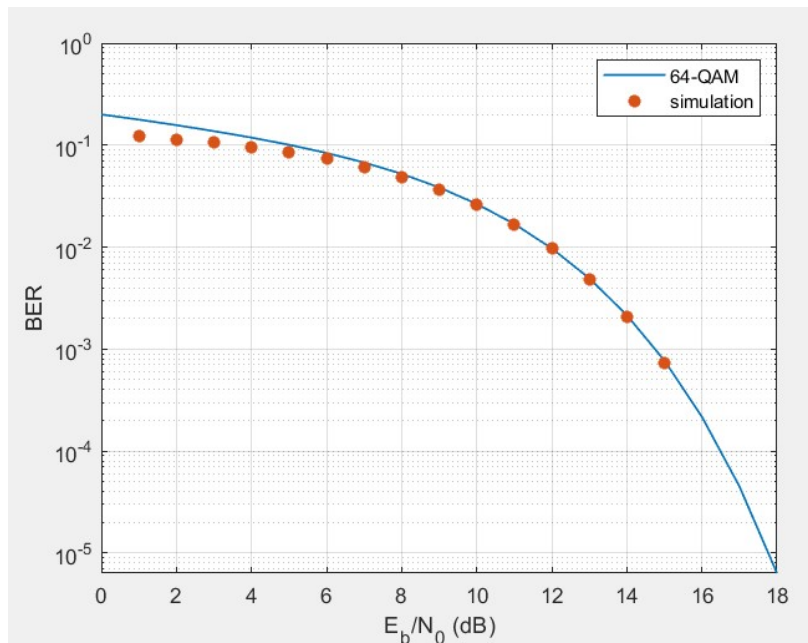
- Στη διαμόρφωση, παρατηρούμε ότι το εξαγόμενο σήμα μετασχηματίζεται από τη βασική ζώνη σε ζωνοπερατό σήμα, το οποίο είναι κεντραρισμένο στη συχνότητα f_c του φέροντος. Η κεντρική συχνότητα είναι 10 και το εύρος ζώνης είναι 2.5

Γ. μετά από προσθήκη θορύβου



- Στην αποδιαμόρφωση, το σήμα μετατρέπεται ξανά σε βασικής ζώνης, επιτρέποντας με αυτό τον τρόπο να φιλτραριστεί από το προσαρμοσμένο φίλτρο του δέκτη. Στη συνέχεια, το σήμα υποβάλλεται σε δειγματοληψία.

Έπειτα, δημιουργούμε την συνάρτηση `ask_ber_func_qam_nyq` και με τη χρήση του εργαλείου `bertool` παίρνουμε το ακόλουθο αποτέλεσμα :



- Παρατηρώ πως πράγματι, η θεωρητική και η πειραματική καμπύλη ταυτίζονται.

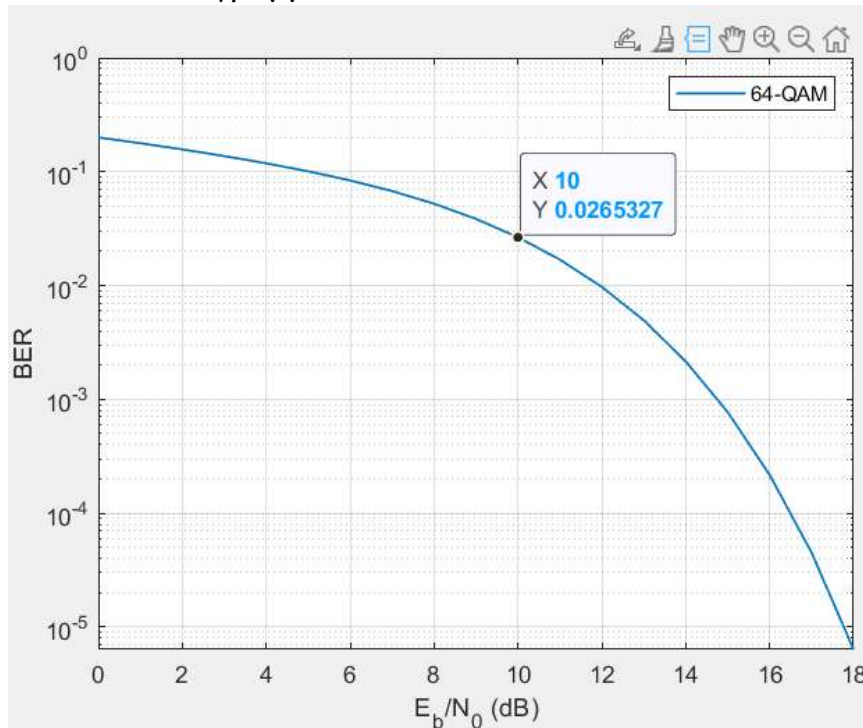
Ο κώδικας της συνάρτησης `ask_ber_func_qam_nyq` :

```

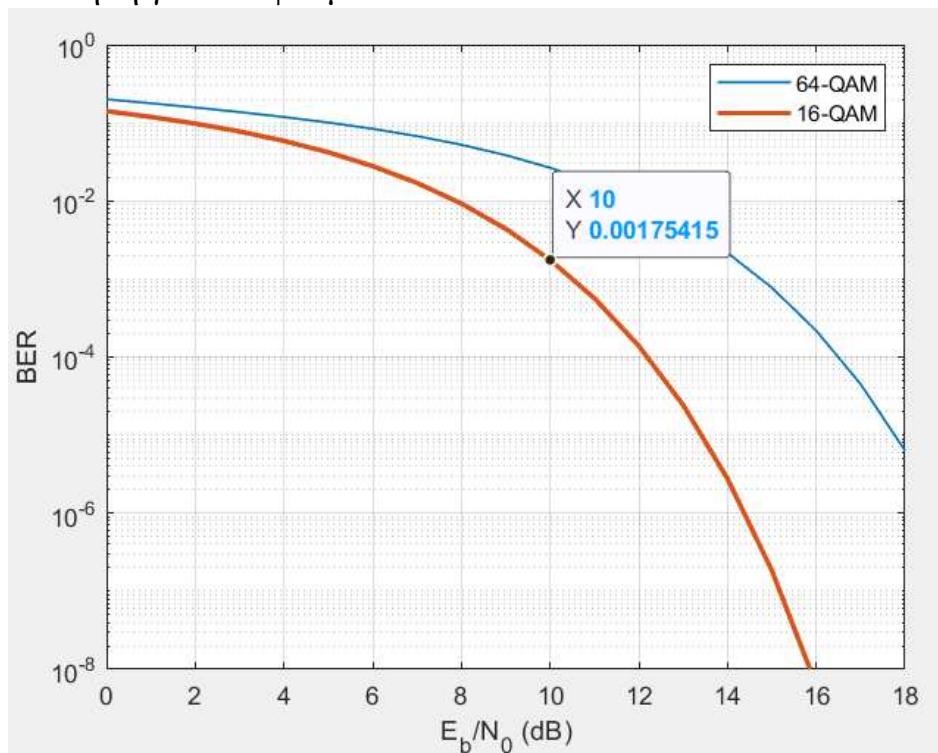
1 function [ber,numBits] = ask_ber_func_qam_nyq(EbNo, maxNumErrs, maxNumBits)
2 % Import Java class for BERTool.
3 import com.mathworks.toolbox.comm.BERTool.*;
4 % Initialize variables related to exit criteria.
5 totErr = 0; % Number of errors observed
6 numBits = 0; % Number of bits processed
7 % A. --- Set up parameters. ---
8 % --- INSERT YOUR CODE HERE.
9 M=64; % M-QAM
10 L=sqrt(M); l=log2(L); k=log2(M);
11 Nsymb=30000 % number of symbols in each run
12 nsamp=32; % oversampling, i.e. number of samples per T
13 rolloff=0.25; % rolloff for raised cosine filter
14 % Simulate until number of errors exceeds maxNumErrs
15 % or number of bits processed exceeds maxNumBits.
16 while((totErr < maxNumErrs) && (numBits < maxNumBits))
17 % Check if the user clicked the Stop button of BERTool.
18 %if (BERTool.getSimulationStop)
19 %break;
20 %end
21 % B. --- INSERT YOUR CODE HERE.
22 errors=ask_qam(M,Nsymb,nsamp,EbNo,rolloff);

```

- (3) Με την χρήση του συστήματος 64QAM δεν μπορούμε να επιτύχουμε πιθανότητα εσφαλμένου bit μικρότερη του 0.002, για τιμή του σηματοθορυβικού λόγου $E_b/N_0=10$, όπως φαίνεται στο ακόλουθο διάγραμμα



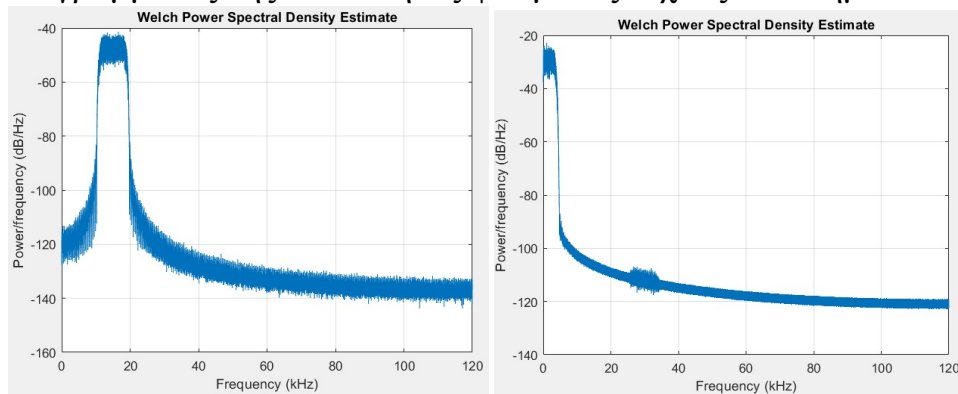
Με τη χρήση 16QAM παρατηρούμε ότι ικανοποιείται η ζητούμενη συνθήκη για το σφάλμα



- Ο μέγιστος ρυθμός μετάδοσης είναι :

$$R' = \frac{1}{T} \log_2 M' = 2 \cdot 4 = 8 \text{ Mbps}$$

- Οι παράμετροι που επηρεάζουν τα φασματικά χαρακτηριστικά baud rate T , roll off α , W καθώς και οι τιμές των nsamp , EbNo μένουν ίδιες, άρα δεν παρατηρούμε αλλαγή του σχετικού διαγράμματος της πυκνότητας φάσματος ισχύος των σημάτων.



- (4) Το καινούριο roll- off είναι : $\alpha' = \frac{0,25}{2} = 0,125$

Ο μέγιστος ρυθμός μετάδοσης θα είναι :

$$R' = \frac{W \cdot \log_2 M}{1 + \alpha'} = \frac{2.5 \cdot \log_2 16}{1 + 0,125} = \frac{10}{1,125} = 8.89 \text{ Mbps}$$