

Ψηφιακές Τηλεπικοινωνίες

Άσκηση 2

Κώστας Ζάκκας

AM:5292

Εξομοίωση Τηλεπικοινωνιακού Συστήματος Βασικής Ζώνης

1. Παρακάτω γίνεται μια περιγραφή των βασικών σημείων της υλοποίησης του ζητούμενου συστήματος. Για πραγματοποίηση πιο αξιόπιστων μετρήσεων χρησιμοποιείται ως είσοδος ένας αρκετά μεγάλος αριθμός δεδομένων και συγκεκριμένα η είσοδος έχει μέγεθος 10^5 .

Είσοδος: Αρχικά με χρήση της συνάρτησης `randn` παράγεται μια τυχαία δυαδική ακολουθία με ισοπίθανη την εμφάνιση του 0 και του 1 που θα δωθεί ως είσοδος στο σύστημα. Έπειτα μετατρέπουμε την δυαδική ακολουθία σε ακολουθία ακεραίων ανάλογα για την κάθε τιμή του M ενώ αν διαίρεση της ακολουθίας με το $\log M$ είναι ατελής συμπληρώνονται τα ψηφία που χρειάζονται με μηδενικά. Μετά αντιστοιχούμε την ακολουθία ακεραίων με τα σύμβολα του αστερισμού παίρνοντας τον μιγαδικό που προκύπτει από τις δύο συνιστώσες που υπολογίζονται με βάση τους τύπους της εκφώνησης. Επίσης για λόγους καλύτερης ψηφιακής αναπαράστασης γίνεται υπερδειγματοληψία της εισόδου.

Φίλτρα πομπού/δέκτη: Τα φίλτρα πομπού και δέκτη θα υλοποιηθούν ως φίλτρα τετραγωνικής ρίζας ανυψωμένου συνημιτόνου με παράγοντα αναδίπλωσης 0.3. Αυτό γίνεται με την χρήση της συνάρτησης `rcosdesign`. Έπειτα με χρήση της συνάρτησης `conv` γίνεται συνέλιξη του σήματος με το φίλτρο που κατασκευάστηκε.

Κανάλι: Στη συνέχεια αν το κανάλι είναι μη ιδανικό η κρουστική απόκριση που το χαρακτηρίζει υπερδειγματοληπτείται με τον ίδιο συντελεστή όπως το παραπάνω φίλτρο και γίνεται συνέλιξή του με το σήμα. Διαφορετικά αν το κανάλι είναι ιδανικό το σήμα παραμένει το ίδιο.

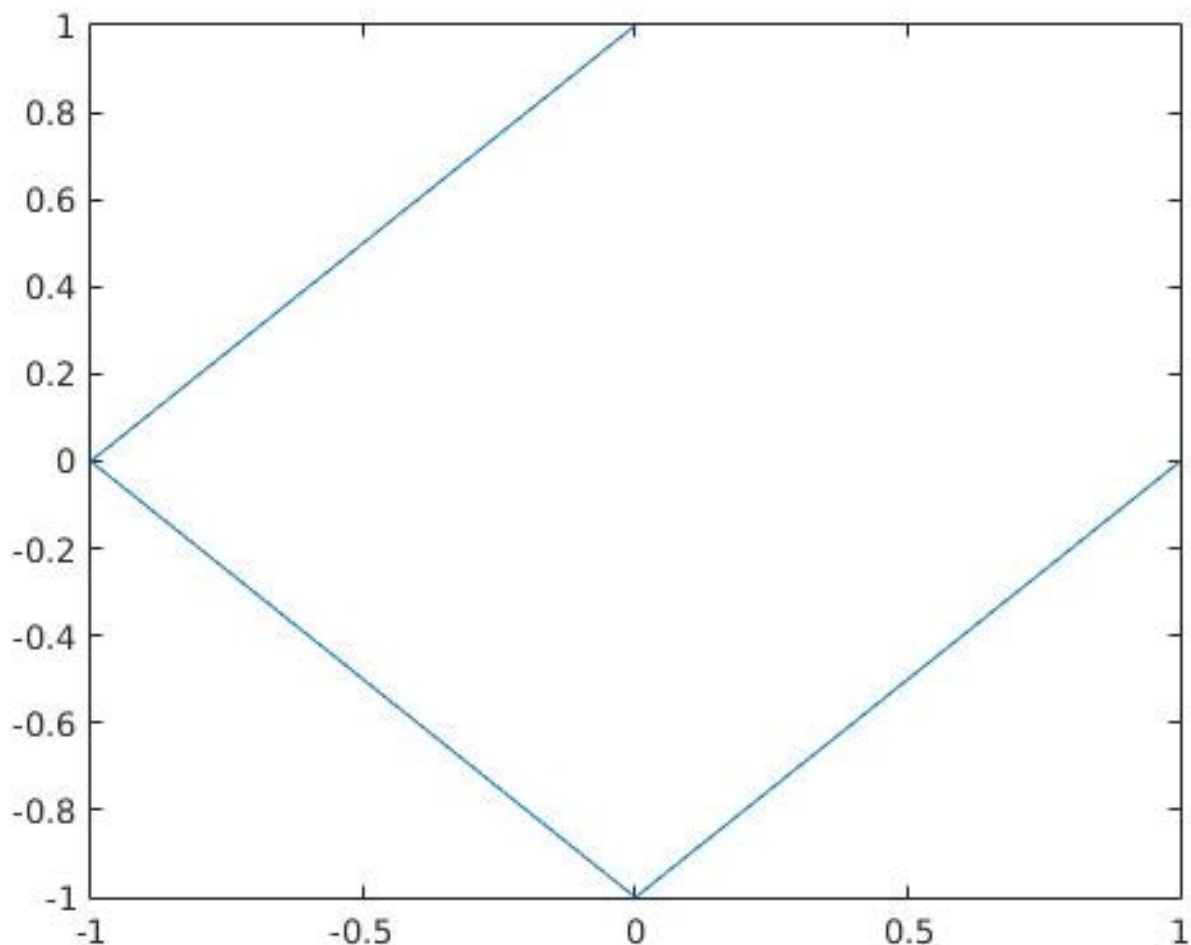
Προσθήκη θορύβου: Πριν το σήμα εισαχθεί στον δέκτη του προστίθεται θόρυβος. Η ισχύς του ισούται με τη διασπορά του και καθορίζεται από το SNR που έχουμε. Στην περίπτωση των μιγαδικών αστερισμών όπως εδώ ο θόρυβος πρέπει να προστίθεται και στο πραγματικό και στο φανταστικό των προς μετάδοση συμβόλων. Άρα οι μεταβλητές θορύβου που παράγονται θα είναι μιγαδικές διαιρεμένες με τη ρίζα 2, έτσι ώστε το πραγματικό και το φανταστικό μέρος των συμβόλων που πρόκειται να μεταδωθούν, να αλλοιώνεται από θόρυβο διασποράς $(\sigma^2)/2$. Γιαυτό γίνεται υπολογισμός της ισχύς του σήματος που είναι το άθροισμα των τετραγώνων των απόλυτων τιμών του διανύσματος του σήματος προς το μήκος του.

Δέκτης: Εφαρμόζεται στο εισερχόμενο σήμα το ίδιο φίλτρο με αυτό του πομπού και αφαίρεση της καθυστέρησης από την αρχή και το τέλος του σήματος. Έπειτα η ακολουθία των συμβόλων στην έξοδο του φίλτρου δέκτη υποδειγματοληπτείται.

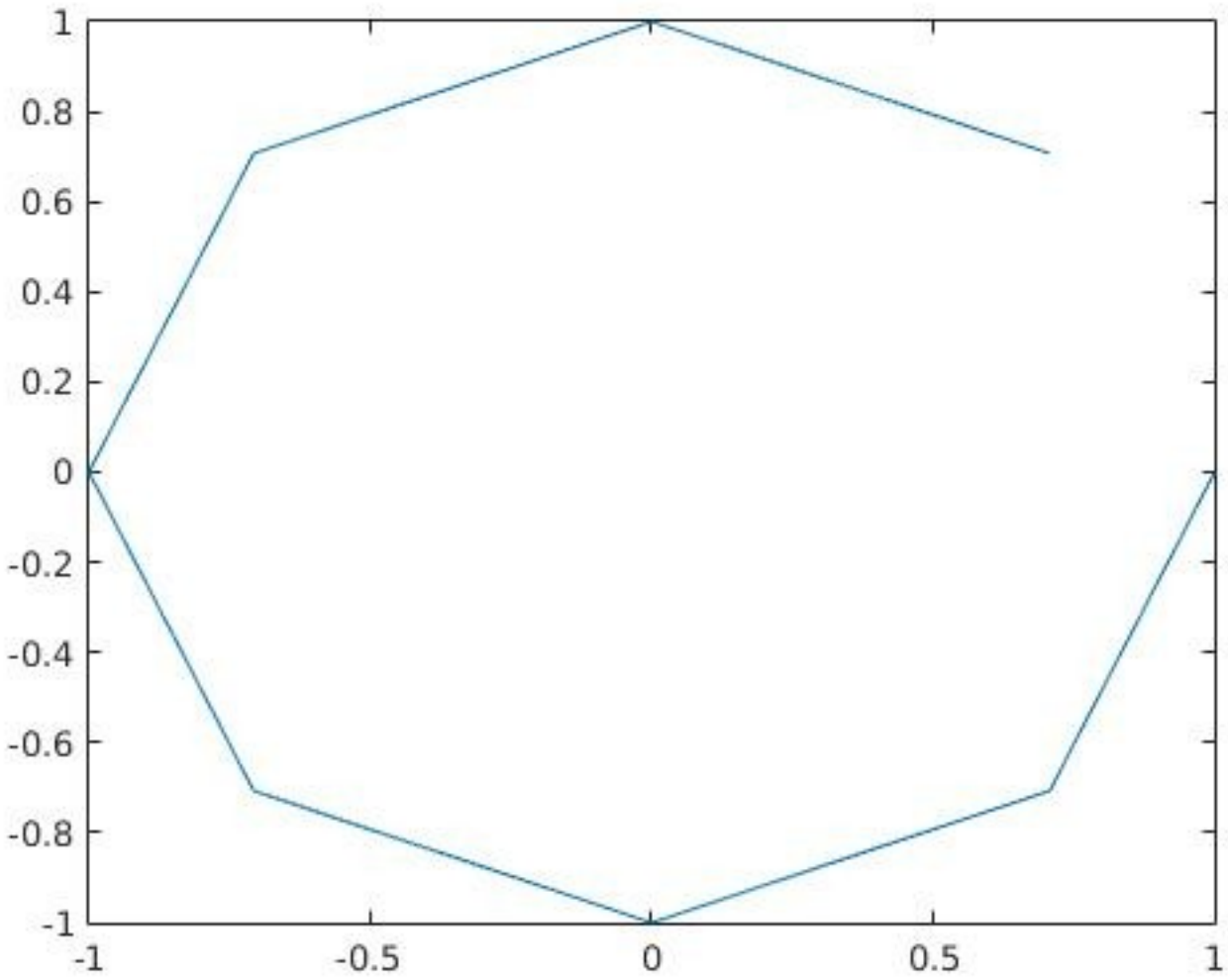
Διάταξη Απόφασης: Αρχικά γίνεται υπολογισμός των συμβόλων του αστερισμού. Έπειτα τα δείγματα περνούν από μία διάταξη απόφασης οπότε και αποφασίζεται ποια ήταν τα αντίστοιχα σύμβολα που στάλθηκαν. Αυτό γίνεται υπολογίζοντας τις ευκλείδειες αποστάσεις από τα σύμβολα του αστερισμού οπότε το σύμβολο που απέχει λιγότερο είναι αυτό που αντιστοιχεί στο στοιχείο της εξόδου. Έπειτα ανάλογα με το M και την αφαίρεση μηδενικών που είχαν προστεθεί αρχικά υπολογίζεται η δυαδική ακολουθία εξόδου, τα λάθη σε σχέση με την αρχική, το BER και το SER.

2. Σε αυτό το ερώτημα παραθέτουμε τα σχήματα των αστερισμών που υλοποιήθηκαν για τις διαμορφώσεις 4-PSK και 8-PSK.

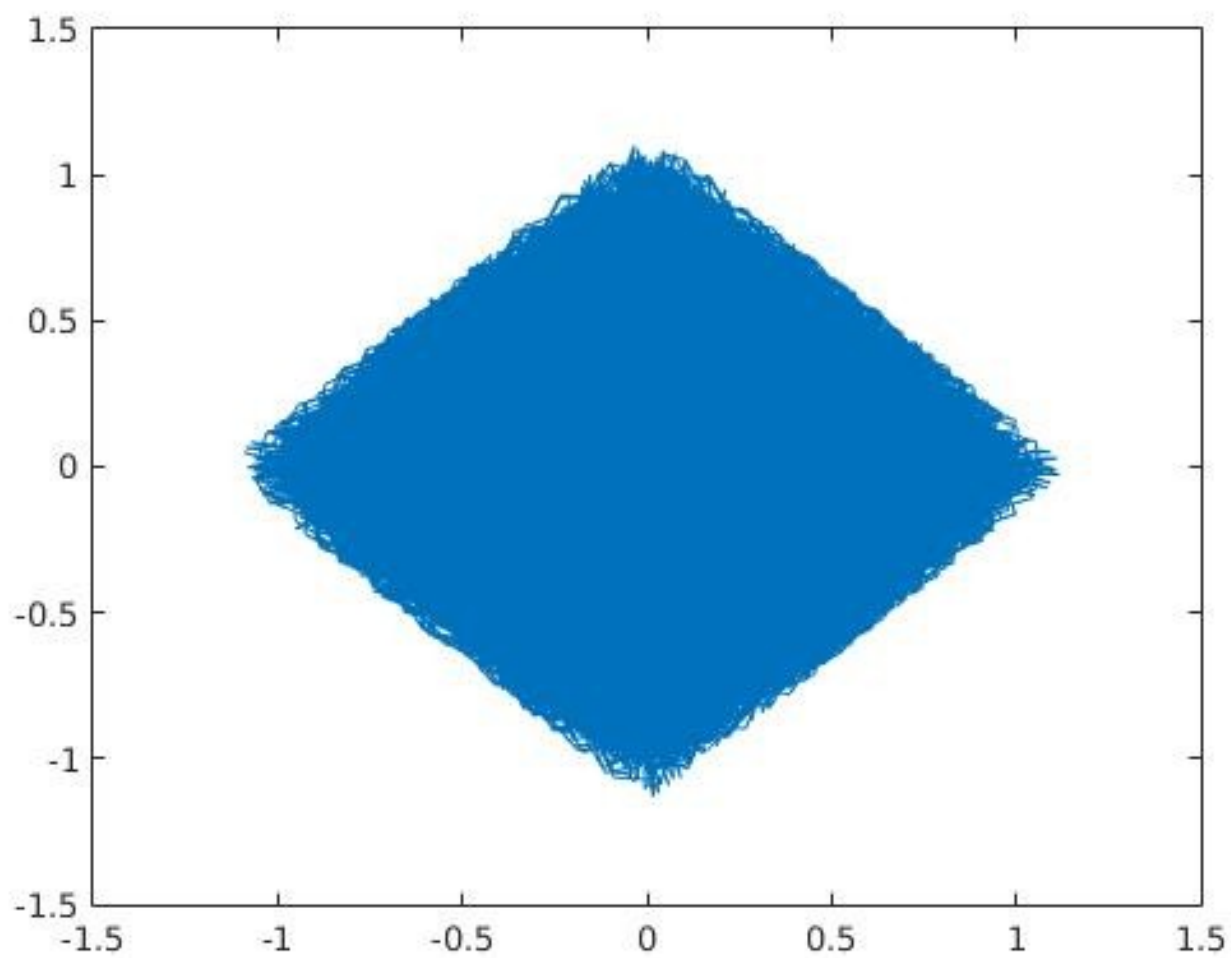
PSK-4



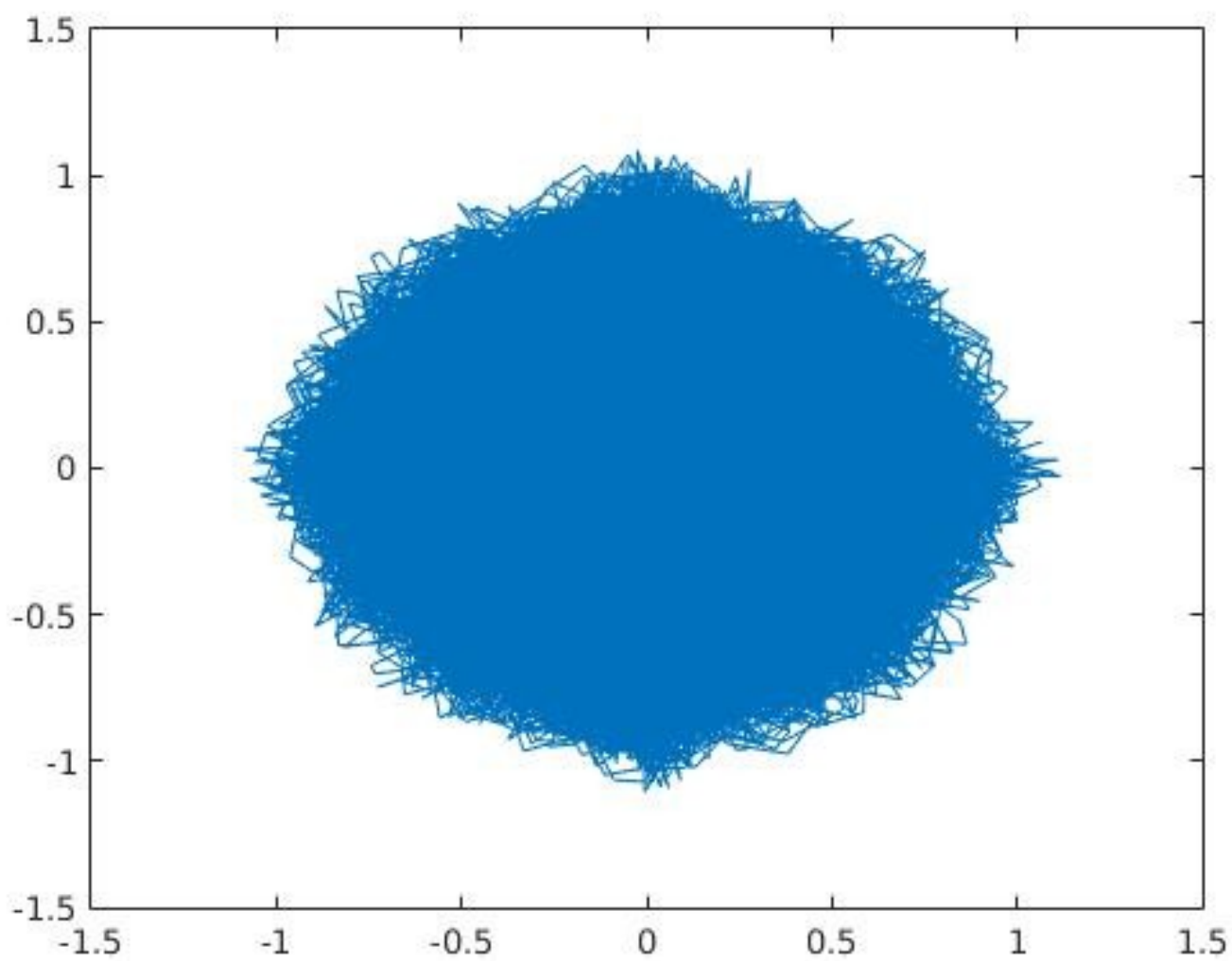
PSK-8



PSK-4(το σήμα που στάλθηκε)



PSK-8(το σήμα που στάλθηκε)



3,4. Παρακάτω παραθέτουμε σε πίνακες τις τιμές του SER και του BER για καθένα από τα δύο συστήματα και για κάθε ένα από τα δύο κανάλια, που αντιστοιχούν στις διάφορες τιμές του SNR καθώς και τα διαγράμματα με τις αντίστοιχες καμπύλες.

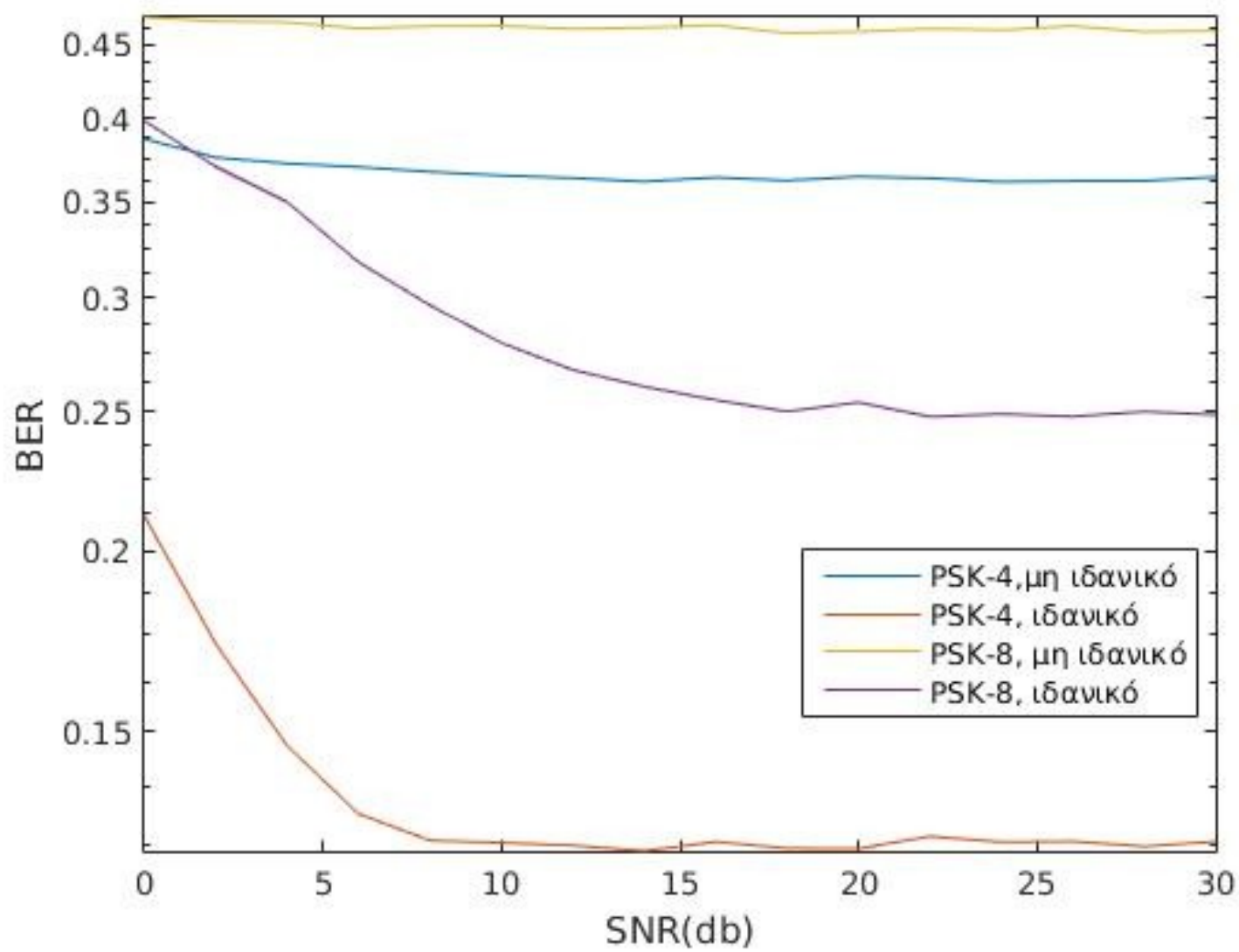
4-PSK

| SNR(db) | Ιδανικό Κανάλι | | Μη ιδανικό κανάλι | |
|---------|----------------|--------|-------------------|---------|
| | BER | SER | BER | SER |
| 0 | 0.21178 | 0.3393 | 0.38697 | 0.54548 |
| 2 | 0.17259 | 0.2988 | 0.37568 | 0.52944 |
| 4 | 0.14662 | 0.2728 | 0.37209 | 0.52326 |
| 6 | 0.13153 | 0.2562 | 0.36998 | 0.52052 |
| 8 | 0.12602 | 0.2497 | 0.36715 | 0.51668 |
| 10 | 0.12556 | 0.2432 | 0.36495 | 0.51316 |
| 12 | 0.12502 | 0.2357 | 0.36349 | 0.50976 |
| 14 | 0.12392 | 0.2130 | 0.36146 | 0.50748 |
| 16 | 0.12576 | 0.2003 | 0.36384 | 0.50964 |
| 18 | 0.12445 | 0.1876 | 0.36204 | 0.50836 |
| 20 | 0.12437 | 0.1746 | 0.36431 | 0.51056 |
| 22 | 0.12686 | 0.1673 | 0.36343 | 0.51052 |
| 24 | 0.12577 | 0.1684 | 0.36131 | 0.50766 |
| 26 | 0.12584 | 0.1597 | 0.36177 | 0.50858 |
| 28 | 0.12481 | 0.1510 | 0.36191 | 0.51096 |
| 30 | 0.12581 | 0.1601 | 0.36403 | 0.50965 |

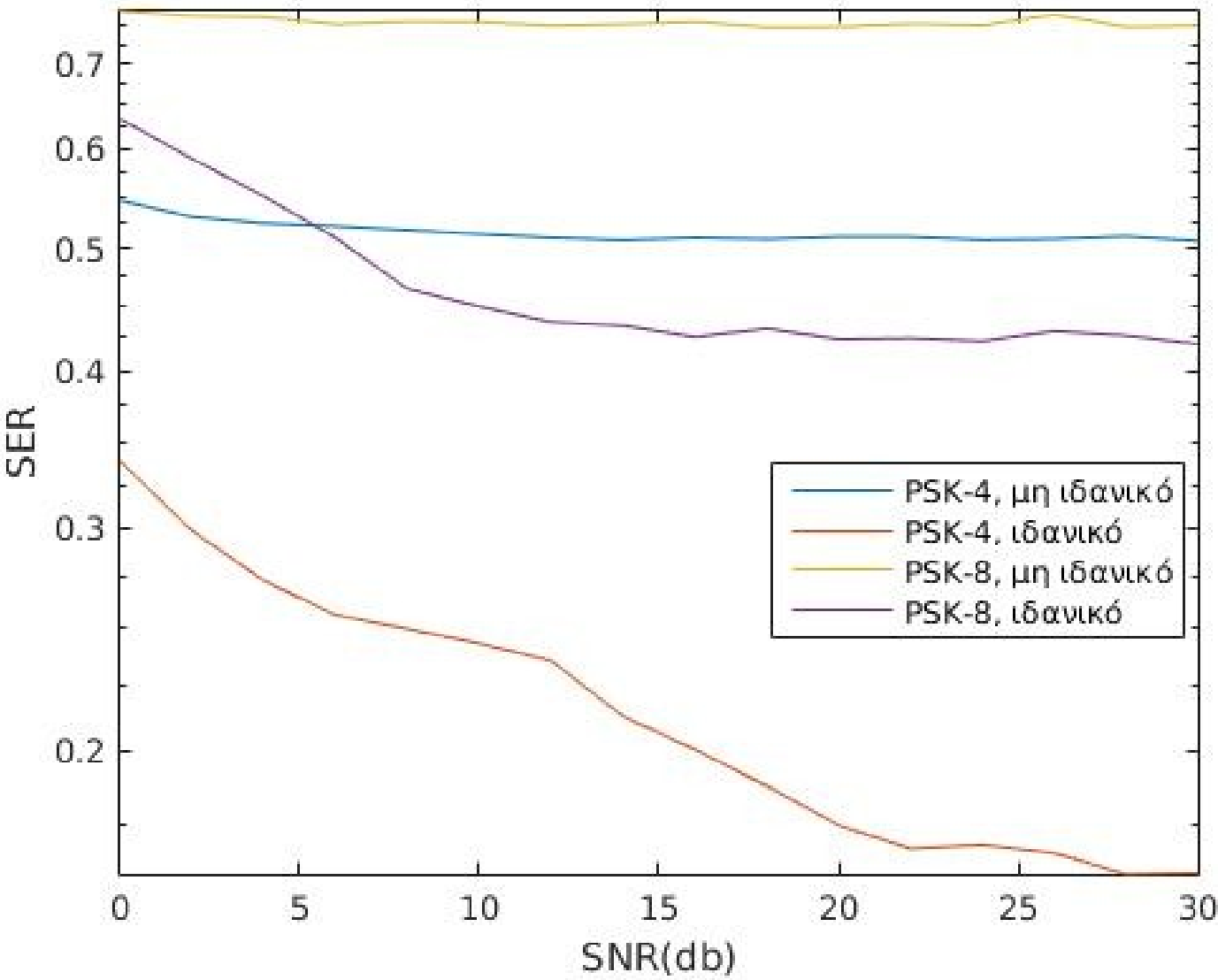
8-PSK

| SNR(db) | Ιδανικό Κανάλι | | Μη ιδανικό κανάλι | |
|---------|----------------|--------|-------------------|---------|
| | BER | SER | BER | SER |
| 0 | 0.39831 | 0.6332 | 0.47006 | 0.77035 |
| 2 | 0.37021 | 0.5889 | 0.467 | 0.76390 |
| 4 | 0.3998 | 0.5498 | 0.46633 | 0.76255 |
| 6 | 0.31771 | 0.5097 | 0.46177 | 0.75235 |
| 8 | 0.29663 | 0.4644 | 0.46339 | 0.75592 |
| 10 | 0.27925 | 0.4496 | 0.46368 | 0.75514 |
| 12 | 0.26745 | 0.4369 | 0.46146 | 0.75040 |
| 14 | 0.26039 | 0.4342 | 0.46231 | 0.75235 |
| 16 | 0.25487 | 0.4253 | 0.46416 | 0.75565 |
| 18 | 0.25024 | 0.4318 | 0.45831 | 0.74842 |
| 20 | 0.25392 | 0.4237 | 0.45934 | 0.74872 |
| 22 | 0.24816 | 0.4244 | 0.46158 | 0.75298 |
| 24 | 0.24925 | 0.4222 | 0.46062 | 0.75079 |
| 26 | 0.24824 | 0.4299 | 0.46352 | 0.76548 |
| 28 | 0.25017 | 0.4265 | 0.45928 | 0.74872 |
| 30 | 0.24893 | 0.4199 | 0.45997 | 0.75058 |

BER



SER



Από τις παραπάνω γραφικές παρατηρούμε ότι και το BER και το SER για τα ιδανικά κανάλια στις δύο διαμορφώσεις μειώνονται όσο αυξάνεται το SNR, που είναι λογικό καθώς όσο αυξάνεται το SNR τόσο πιο ισχυρό είναι το σήμα σε σχέση με τον θόρυβο. Έπειτα τα BER και SER στις δύο διαμορφώσεις στην περίπτωση του μη ιδανικού καναλιού είναι σχεδόν σταθερά και αντιστοιχούν σε μια οριζόντια σχεδόν ευθεία γραμμή, από μία για το καθένα. Επίσης βλέπουμε ότι η 4-PSK διαμόρφωση είναι καλύτερη από την 8-PSK σε όλες τις περιπτώσεις. Τέλος το SER έχει μεγαλύτερες τιμές από το BER για όλες τις περιπτώσεις που είναι λογικό καθώς οι επιλογές για το BER είναι δύο το 0 και το 1 ενώ τα σύμβολα έχουν πιο πολλές τιμές οπότε και μεγαλύτερη πιθανότητα λάθους.

Παρακάτω παρατίθεται ο κώδικας που αναπτύχθηκε ώστε να εξαχθούν τα παραπάνω δεδομένα.

```
%%ΕΙΣΟΔΟΣ
```

```
%Αριθμος bit εισοδου
```

```
n=1e5;
```

```
% τυχαία δυαδική ακολουθία μήκους n
```

```
% με χρήση της συνάρτησης rand
```

```
bin_seq = round(rand(1,n));
```

```
%
```

```
% μετατροπή των bits του bin_seq σε log2(M)-bit σύμβολα
```

```
%
```

```
for M=[4 8]
```

```
% κάθε k=log2(M) bit αντιστοιχούν σε ένα σύμβολο
```

```
k=log2(M);
```

```
% το μήκος της δυαδικής ακολουθίας να μην % διαιρείται ακριβώς με το log2(m)
```

```
% οπότε τότε θα συμπληρώσουμε στην αρχή της ακολουθίας μηδενικά ώστε να  
γίνεται
```

```
% η διαίρεση χωρίς να αλλάζει η αρχική ακολουθία
```

```
bin_seq2=0;
```

```
md=mod(n,k);
```

```
if md>0
```

```
    if mod(n+1,log2(M))~=0
```

```
        c=n+2;
```

```
    elseif mod(n+2,log2(M))~=0
```

```
        c=n+1;
```

```
    end
```

```
    for z=1:c
```

```
        if z>n
```

```
            bin_seq2(z)=0;
```

```

        else
            bin_seq2(z)=bin_seq(z);
        end
    end
elseif md==0
    bin_seq2=bin_seq;
end
lng=length(bin_seq2);
% χρήση της reshape ώστε σε κάθε γραμμή να υπάρχουν k=log2(M) bits
bin_seq1=reshape(bin_seq2,k,lng/k);
% μετατροπή κάθε γραμμής(log2(M) bit) σε έναν ακέραιο
dec_seq=bi2de(bin_seq1,'left-msb');
%

%%ΔΙΑΜΟΡΦΩΣΗ ΕΙΣΟΔΟΥ
Es=1;
sm=reshape(dec_seq,1,lng/k);
% τα σύμβολα θα είναι M οπότε θα αντιστοιχίσουμε
% κάθε ακέραιο με δύο συνιστώσες ώστε να έχει μια θέση στο χώρο
n=length(sm);
for i=1:n
    for l=1:M
        if sm(i)==(l-1);
            m=l;
            s1(i)=sqrt(Es)*cos((2*pi*m)/M);
            s2(i)=sqrt(Es)*sin((2*pi*m)/M);
        end
    end
end
% Συνθεση μιγαδικού αριθμού με τις δύο συνιστώσες
s_in=complex(s1,s2);
% Υπερδειγματοληψία του σήματος με βάση τον ρυθμό της εκφώνησης
s=upsample(s_in,4);
% Αφαιρούνται τα επιπλέον μηδενικά που είναι στο τέλος
s(length(s)-2:length(s))=[];
%

%%ΦΙΛΤΡΟ ΠΟΜΠΟΥ
%Κατασκευή του φίλτρου με χρήση της συνάρτησης rcosdesign
b=rcosdesign(0.3,6,4,'sqrt');
% Περνάμε το σήμα μέσα από το φίλτρο που δημιουργήσαμε με χρήση
% της συνάρτησης conv
s_f=conv(s,b);
%
```

```
%%ΦΙΛΤΡΟ ΚΑΝΑΛΙΟΥ
```

```
%Ελεγχος αν το καναλι είναι ιδανικο Για idl=1 είναι ιδανικο για idl=0 όχι
```

```
idl=0;
```

```
if idl==0;
```

```
% Θα γίνει συνελιξη με το παρακατω ιδανικο καναλι
```

```
h=[0.04 -0.05 0.07 -0.21 -0.5 0.72 0.36 0 0.21 0.03 0.07];
```

```
% Υπερδειγματοληψια του σηματος με βαση τον ρυθμο της εκφωνησης
```

```
h_ups=upsample(h,4);
```

```
% Αφαιρεση των 3 περιττων μηδενικων στο τελος
```

```
h_ups(length(h_ups)-2:length(h_ups))=[];
```

```
% Συνελιξη του σηματος με το ιδανικο καναλι με χρηση της συναρτησης conv
```

```
s_f=conv(s_f,h_ups);
```

```
end
```

```
%
```

```
%%ΘΟΡΥΒΟΣ
```

```
%προσθηκη θορυβου
```

```
for SNR=0:2:30
```

```
% Παιρνουμε το μετρο των στοιχειων του σηματος
```

```
s_f_abs=abs(s_f);
```

```
% Υπολογισμος της μεσης ισχυος του σηματος
```

```
P_S=0;
```

```
for i=1:length(s_f)
```

```
P_S=P_S + s_f_abs(i)^2;
```

```
end
```

```
P_S=P_S/length(s_f);
```

```
% Υπολογισμος της μεσης ισχυος του θορυβου για τα διαφορα SNR
```

```
% η οποια είναι το τετραγωνο της διασπορας του θορυβου
```

```
P_N=P_S/(10^(SNR/10));
```

```
% Υπολογισμος θορυβου
```

```
noise1=randn(length(s_f),1);
```

```
% Στη περιπτωση των μιγαδικων αστερισμων ο θορυβος θα πρεπει
```

```
% να προστεθει στο πραγματικο και στο φανταστικο μερος των συμβολων
```

```
%ωστε το πραγματικο και το φανταστικο μερος να εχουν αλλοιωθει
```

```
% απο θορυβο διασπορας  $\sigma^2/2=P_N/2$ 
```

```
if(isreal(s_f)==0)
```

```
    noise1=(noise1 + j*randn(length(s_f),1))/sqrt(2);
```

```
end
```

```
% Εξισωση θορυβου
```

```
noise=sqrt(P_N)*noise1;
```

```
% Προσθηκη του θορυβου στο σημα
```

```
s_f_n=s_f+noise';
```

```

%

%%ΔΕΚΤΗΣ
%Ο δεκτης εχει το ιδιο φιλτρο με τον πομπο

s_r=conv(s_f_n,b);
%
% Υπολογισμος της καθυστερησης για να αφαιρεθει
if idl==0
    delay=2*((length(b)-1)/2)+((length(h_ups)-1)/2);
elseif idl==1
    delay=2*((length(b)-1)/2);
end
% Αφαιρεση της καθυστερησης
s_r=s_r(delay+1:length(s_r)-delay);
% Υποδειγματοληψια του σηματος
s_r_d=downsample(s_r,4);
%

%%ΔΙΑΤΑΞΗ ΑΠΟΦΑΣΗΣ
%Υπολογισμος των συμβολων

for m=1:M
    s1_2(m)=sqrt(Es)*cos((2*pi*m)/M);
    s2_2(m)=sqrt(Es)*sin((2*pi*m)/M);
end
% Συνθεση μιγαδικου αριθμου με τις δυο συνιστωσες
s_2=complex(s1_2,s2_2);
% για το σημα που προεκυψε μετα την υποδειγματοποιηση
%γινεται ελεγχος καθε φορα απο ποιο συμβολο θα εχει την μικροτερη αποσταση
for j=1:length(s_r_d)
    % Βαζουμε για αρχικοποιηση της ελαχιστης τιμης για την ελαχιστη αποσταση
    % μια πολυ μεγαλη τιμη ωστε να γινει παρακατω σωστα η συγκριση
    apostash_min = 1e100;
    for a=0:M-1
        % υπολογισμος ευκλειδεις αποστασης του διανυσματος απο τα συμβολα
        % ωστε το συμβολο απο το οποιο θα εχει την μικροτερη αποσταση θα
        % ειναι το συμβολο που του αντιστοιχει και το m που αντιστοιχει σε
        % αυτο το συμβολο θα ειναι η αντιστοιχη αρχικη εισοδος
        apostash(a+1) = sqrt((s_r_d(j)-s_2(a+1))^2);
        if (apostash(a+1) < apostash_min)
            apostash_min = apostash(a+1);
            out=a;
        end
    end
end

```

```

    s_out(j)=out;
end
% Μετατροπή των συμβολών σε δυαδικό
bin_out=de2bi(s_out,'left-msb');
    bin_out = bin_out.';
    bin_out = bin_out(:).';
% Αφαίρεση των μηδενικών που προστεθηκαν
if md>0
    zer=lng-length(bin_seq);
    bin_out((lng-zer)+1:lng)=[];
end
%

%%ΥΠΟΛΟΓΙΣΜΟΣ BER και SER
ber=0;
ser=0;
for q=1:length(bin_seq)
    if bin_seq(q)==~bin_out(q)
        ber=ber+1;
    end
end
for v=1:length(dec_seq)
    if dec_seq(v)==s_out(v)
        ser=ser+1;
    end
end
BER=ber/length(bin_seq);
SER=(length(s_in)-ser)/length(s_in);
fprintf('Για M=%d', M);
fprintf(' και SNR=%d το BER είναι:%d\n αντίστοιχα το SER είναι:%d\n\n', SNR,
BER, SER);
%

end % του SNR

s_in=0;
s_out=0;
%

end % της for M=[4 8]

```