ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

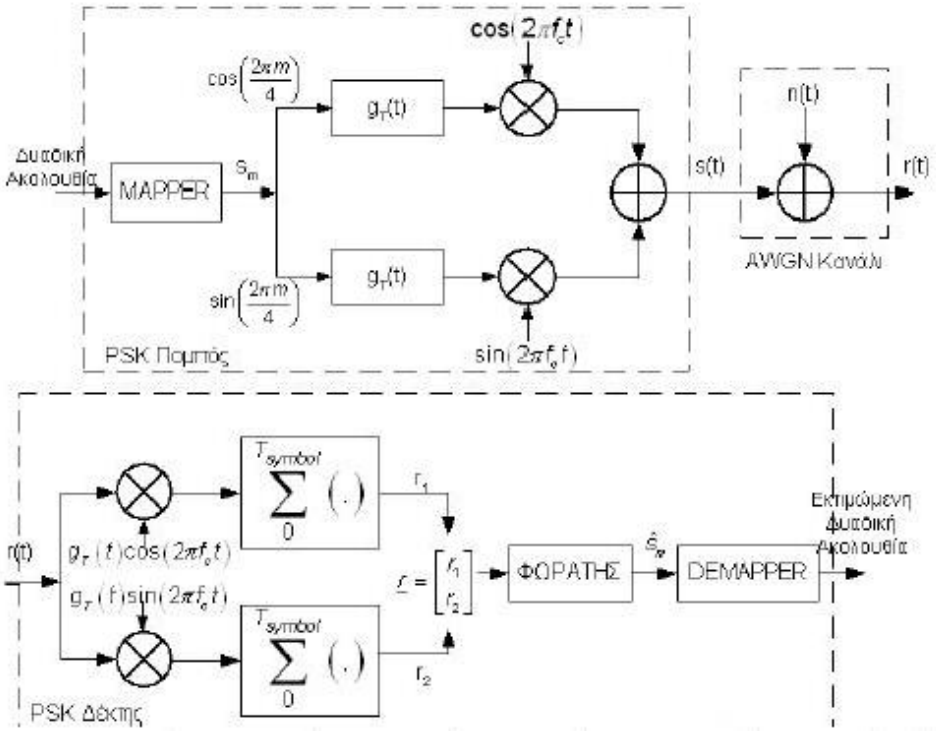
Ψηφιακές Τηλεπικοινωνίες

2η Άσκηση

Μάλλιος Χαράλαμπος 5343

Ακαδημαικό έτος 2015-2016

1. Με βάση τις υποδείξεις, υλοποιήθηκε το σύστημα 4-PSK και αναφέρονται τα βασικά του σημεία.



Δημιουργήθηκαν οι κάτωθι συναρτήσεις οι οποίες προσωμοιώνουν τη λειτουργία του .

A) **eisodos_dyadiki.m** Αρχικά παράγουμε τη δυαδική ακολουθία των bits 10^5 για να μεγαλύτερη ακρίβεια .

Κώδικας MATLAB :

```
function [ dyadiki_akolouthia ] = eisodos_dyadiki( eisodos)

%δημιουργία της δυαδικής ακολουθίας που θα χρησιμοποιηθεί ως είσοδος
dyadiki_akolouthia = randsrc(eisodos, 1, [0,1]);
end
```

B) **antistoixia.m** Αντιστοιχίζουμε τα σύμβολα σε Bits

Κώδικας MATLAB :

```
function symbola = antistoixia(dyadiki, epilogh, gray)

megethos = length(dyadiki);
```

```
%χωρίζουμε τη πρόταση σε δυο γκρούπ αυτές που διαιρούνται ή οχι με το
δυο
res = mod(megethos, 2);

%αν διαιρείται με το 2
div_dyad = dyadiki(1 : (megethos - res), :);

%κάνουμε reshape σε αυτή
resd = reshape(div_dyad, 2, (megethos - res) / 2);

%αντιστοιχία σε bit
for i = 1: (megethos - res) / 2
    symbola(i) = bin2dec(num2str(resd(:, i)'));
end
```

```
%αν αφήνει υπόλοιπο,  μετατροπή σε δυαδικό
if res ~= 0
    symbola(i + 1) = bin2dec(num2str(dyadiki(megethos - res + 1
:megethos, 1)'));
end

%Μετατροπή σε κώδικα GRAY
if gray == 1
    symbola = bin2gray(symbola, epilogh, 8);
end

end
```

Γ) **diamorfwtis.m** Διαμορφώνει τη κάθε συνιστώσα πολλαπλασιάζοντας με ορθογώνιο παλμό και τη διαμόρφωση γύρω απο τη φέρουσα συχνότητα ωστε να προκύψει το επιθυμητό ζωνοπερατό σήμα.

Κώδικας MATLAB :

```
function sym_dia = diamorfwtis(symbola, epilogh)

%μέγεθος των συμβόλων
megethos = length(symbola);


%περίοδος κάθε συμβόλου
periodos = 40;
%συχνότητα κάθε συμβόλου
syxnotita = 1 / periodos;


%περίοδος δείγματος
per_deigma = 1;
```

```
%περίοδος φέρουσας
per_fer = 4;
%συχνότητα φέρουσας
freq_fer = 1 / per_fer;
energeia = 1;



%ορθογώνιος παλμός
palmos = sqrt(2 * energeia / periodos);

% αρχικοποίηση
sym_dia = zeros(megethos, periodos / per_deigma);

% υπολογισμός
if epilogh == 'PSK'
    for i = 1: megethos
        for j = 1: periodos/per_deigma
            sym_dia(i, j) = palmos * cos( 2*pi*freq_fer*j -
2*pi*symbola(i)/4 );
        end
    end
elseif epilogh == 'FSK'
    for i = 1: megethos
        for j = 1: periodos/per_deigma
            sym_dia(i, j) = palmos * cos(2 * pi * (freq_fer +
symbola(i) * syxnotita) * j);
        end
    end
end

end
```

Δ) **thoribos.m** Προσθήκη θορύβου σύμφωνα με την εκφώνηση .

Κώδικας MATLAB :

```
function enthoribo = thoribos(sym_dia, snr)

%ενέργεια σήματος
Energeia = 1;
Eb = Energeia / 2;
%θόρυβος
No = Eb / (10^(snr/10));
[y,t] = size(sym_dia);
%κατανομή
n = 0;
s = sqrt(No / 2);
%προσθήκη θορύβου
thorubos = n + s * randn(y, t);
enthoribo = sym_dia + thorubos;

end
```

Ε) **apodiamorfwtis.m** Ο δέκτης αποδιαμορφώνει το ληφθέν σήμα .

Κώδικας MATLAB :

```
function apodia = apodiamorfwtis(lifthen, kwdikop)


%περίοδος συμβόλου και συχνότητα
periodos_sym = 40;
syxnotita_sym = 1 / periodos_sym;


%περίοδος δείγματος , φέρουσας και συχνότητα φέρουσας = 1;
periodos_dei = 1;
periodos_fer = 4;
syxnotita_fer = 1 / periodos_fer;
energria_sym = 1;



%δημιουργία παλμού
palmos = sqrt(2 * energria_sym / periodos_sym);

[m,periodos_sym ]= size(lifthen);

% Αποδιαμόρφωση
if kwdikop == 'PSK'
    for k = 1: periodos_sym
        y1(k, 1) = palmos * cos(2 * pi * syxnotita_fer * k);
        y2(k, 1) = palmos * sin(2 * pi * syxnotita_fer * k);
    end

    % αποδιαμόρφωση
    apodia = [lifthen * y1, lifthen * y2];
elseif kwdikop == 'FSK'
    for i = 1: 4
        for k = 1: periodos_sym
            y(i, k) = palmos * cos(2 * pi * ( syxnotita_fer + i *
syxnotita_sym) * k);
        end
    end

    % 4 συνιστώσες
    apodia = lifthen * y';
end

end
```

Ζ) **apofasi.m** Προσομοιώνει τη λειτουργία του φωρατή . Δέχεται το διάνυσμα r και αποφασίζει σε είναι πιό σύμβολο είναι πιό κοντά αφού συγχρονιστεί ο πομπός το δέκτη (πρέπει να γνωρίζει τη φάση φέρουσας),προκύπτει ένα διάνυσμα r με δυο συνιστώσες της φέρουσας. Αυτή θα είναι η αναμενόμενη τιμή του 4-PSK .

Κώδικας MATLAB :

```
function symbola = apofasi(r, kwdikop)

[grammes, ~] = size(r);

if kwdikop == 'PSK'

  % πιθανότητα συμβόλου

for i = 1: 4
        s(i, 1) = cos( 2 * pi * i / 4 );
        s(i, 2) = sin( 2 * pi * i / 4 );
    end

    % υπολογισμός της μεγαλύτερης πιθανότητας
    for j =1: grammes
        for i = 1: 4
            temp(i, 1) = norm([r(j,1), r(j,2)] - s(i,:));
        end
        [d, symbola(j, 1)] = min(temp);

    end

    % 4th = 0th
    symbola = mod(symbola,4);
elseif kwdikop == 'FSK'


    ypol = zeros(1, grammes);
       % υπολογισμός της μεγαλύτερης πιθανότητας

    ypol = [0: 3];
    for j =1: grammes
        deiktis = logical(round(abs(r(j, :))));
        if isempty(ypol(deiktis))
            symbola(j) = 0;
        else
            symbola(j) = max( ypol(deiktis) );
    end
end

    % 4th = 0th
symbola = mod(symbola,4);

end
```

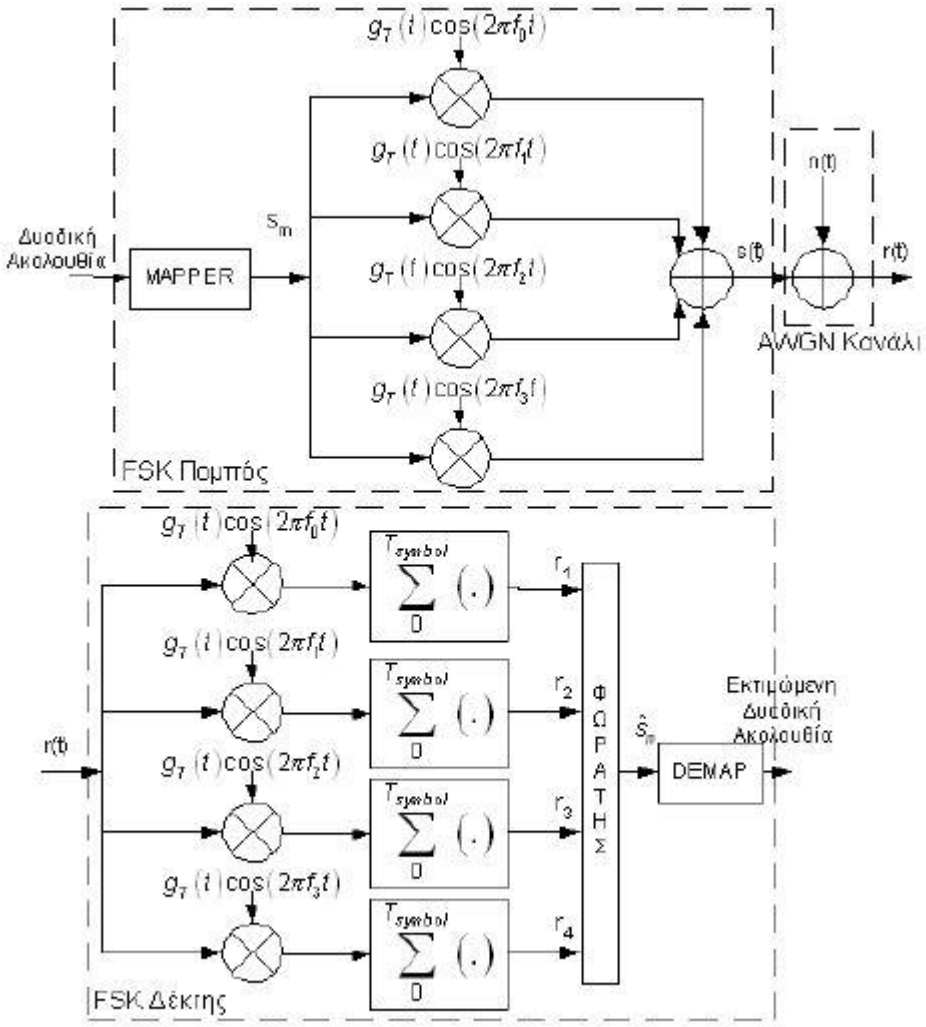
Η) **de_antistoixia.m** Αντιστοίχηση στην ακολουθία που εκτιμήσαμε .

Κώδικας MATLAB :

```
function symbola = apofasi(r, kwdikop)

[grammes, ~] = size(r);
```

```
if kwdikop == 'PSK'

  % πιθανότητα συμβόλου

for i = 1: 4
        s(i, 1) = cos( 2 * pi * i / 4 );
        s(i, 2) = sin( 2 * pi * i / 4 );
    end

    % υπολογισμός της μεγαλύτερης πιθανότητας
    for j =1: grammes
        for i = 1: 4
            temp(i, 1) = norm([r(j,1), r(j,2)] - s(i,:));
        end
        [d, symbola(j, 1)] = min(temp);

    end

    % 4th = 0th
    symbola = mod(symbola,4);
elseif kwdikop == 'FSK'


    ypol = zeros(1, grammes);
       % υπολογισμός της μεγαλύτερης πιθανότητας

    ypol = [0: 3];
    for j =1: grammes
        deiktis = logical(round(abs(r(j, :))));
        if isempty(ypol(deiktis))
            symbola(j) = 0;
        else
            symbola(j) = max( ypol(deiktis) );
    end
end

    % 4th = 0th
symbola = mod(symbola,4);

end
```

2. Με βάση τις υποδείξεις, υλοποιήθηκε το σύστημα 4-FSK και αναφέρονται τα βασικά του σημεία.



Χρησιμοποιούμε τις ίδιες συναρτήσεις με το σύστημα 4-PSK αλλά όταν τις καλούμε θέτουμε σαν επιλογή το 'FSK' για να εκτελεστεί το κομμάτι κώδικα που αφορά αυτό το σύστημα .

Α) **eisodos_dyadiki.m** Αρχικά παράγουμε τη δυαδική ακολουθία των bits

Β) **antistoixia.m** Αντιστοιχίζουμε τα σύμβολα σε Bits κατά κωδικοποίηση gray

Γ) **diamorfwtis.m** Διαμορφώνει τη κάθε συνιστώσα με 4 σήματα της μορφής :

Δ) **thoribos.m** Προσθήκη θορύβου

Ε) **apodiamorfwtis.m** Ο δέκτης αποδιαμορφώνει το ληφθέν σήμα

Ζ) **apofasi.m** Προσομοιώνει τη λειτουργία του φωράτη . Δέχεται το διάνυσμα r και αποφασίζει σε είναι πιό σύμβολο είναι πιό κοντά

Η) **de_antistoixia.m** Αντιστοίχηση στην ακολουθία που εκτιμήσαμε

Για να καλέσουμε τις συναρτήσεις και να υπολογίσουμε το BER για κάθε σύστημα χρησιμοποιούμε το script :

```
%======================== Ερώτημα 1 ============================
clear;
clc;
%counters που χρησιμοποιώ για τα δυο συστήματα
loopp=1;
loop = 1;
loopg=1;%για gray

%τα bits εισόδου
tabitseisodou = 10^5;

%αρχικοποίηση θεωρητικού psk
Pb(loop, 1) = 0;


%αρχικοποίηση για αποθήκευση ber για psk
BER_psk(loop, 1) = 0;

for SNR = 0:2:8


    dyadiki     = eisodos_dyadiki(tabitseisodou);
    symbola     = antistoixia(dyadiki, 'PSK',0);
    sym_dia     = diamorfwtis(symbola, 'PSK');

    enthoribo   = thoribos(sym_dia ,SNR);
    apodiam     = apodiamorfwtis(enthoribo, 'PSK');
    symbols     = apofasi(apodiam, 'PSK');
    eksodos     = de_antistoixia(symbols, 'PSK', 0);
    BER_psk(loop, 1) = biterror(dyadiki, eksodos);



 %υπολογισμός της θεωρητικής τιμής του ber για το 4-PSK
```

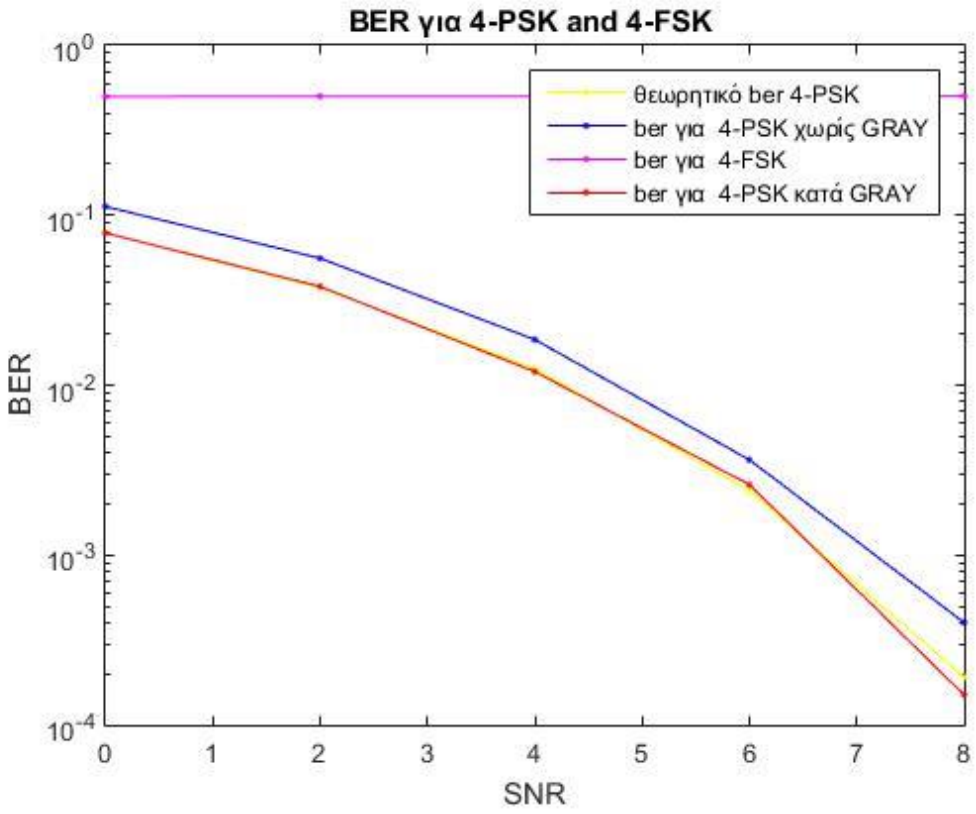
```
     Pb(loop, 1) = 1/2*erfc(sqrt( 10^(SNR/10)) );

    loop = loop + 1;


end
for SNR = 0:2:8


    dyadiki     = eisodos_dyadiki(tabitseisodou);
    symbola     = antistoixia(dyadiki, 'PSK',1);
    sym_dia     = diamorfwtis(symbola, 'PSK');

    enthoribo   = thoribos(sym_dia ,SNR);
    apodiam     = apodiamorfwtis(enthoribo, 'PSK');
    symbols     = apofasi(apodiam, 'PSK');
    eksodos     = de_antistoixia(symbols, 'PSK', 1);
    BER_psk_gray(loopg, 1) = biterror(dyadiki, eksodos);



    loopg = loopg + 1;


end






%======================== Ερώτημα 2 ========================


%αρχικοποίηση για αποθήκευση ber για fsk
BER_fsk(loopp, 1) = 0;
for SNR = 0:2:8


    dyadikif     = eisodos_dyadiki(tabitseisodou);
    symbolaf        = antistoixia(dyadikif, 'FSK', 0);
    sym_diaf        = diamorfwtis(symbolaf, 'FSK');

    enthoribof  = thoribos(sym_diaf ,SNR);
    apodiamf        = apodiamorfwtis(enthoribof, 'FSK');
    symbolsf    = apofasi(apodiamf, 'FSK');
    eksodosf    = de_antistoixia(symbolsf, 'FSK', 0);
    BER_fsk(loopp, 1) = biterror(dyadikif, eksodos);



    loopp = loopp + 1;


end
```

3. Μετρήσεις BER

Αφού υπολογίσουμε τις πιθανότητες σφάλματος τις παρουσιάζουμε γραφικά συναρτήσει του SNR .

Κώδικας MATLAB :

```
%======================== Ερώτημα 3=================================



%Σχεδιασμός BER συναρτήσει του SNR
s_N_R = [0: 2: 8];
semilogy(s_N_R', Pb, 'Y.-');
hold on;
semilogy(s_N_R', BER_psk, 'b.-');
semilogy(s_N_R', BER_fsk, 'm.-');
semilogy(s_N_R', BER_psk_gray, 'r.-');

legend('θεωρητικό ber 4-PSK','ber για  4-PSK χωρίς GRAY','ber για  4-
FSK','ber για  4-PSK κατά GRAY');
title('BER για 4-PSK and 4-FSK ');
xlabel('SNR');
ylabel('BER');
hold;

figure;
```



Βλέπουμε ότι το θεωρητικό 4-PSK παρουσιάζει μικρότερο σφάλμα σε σχέση με το πειραματικό καθώς αυξάνεται το SNR , ωστόσο και στα δυο συνεχώς το BER μειώνεται .**Τα τετραδικά αντίποδα είναι προτιμότερα** . Το 4-FSK παρουσιάζει το μεγαλύτερο BER καθώς τα ορθογώνια απαιτούν το διπλάσιο SNR για να έχουν το ίδιο BER σε σχέση με δυαδικά αντίποδα. Συγκεκριμένα απαιτεί : 10*log10(4) = 6 db παραπάνω .

Αυτό γιατί όσο μεγαλύτερη είναι η ελάχιστη απόσταση μεταξύ 2 σημείων του αστερισμού τόσο μικρότερη είναι η πιθανότητα σφάλματος.

4. Φάσμα Ισχύος

Κώδικας MATLAB :

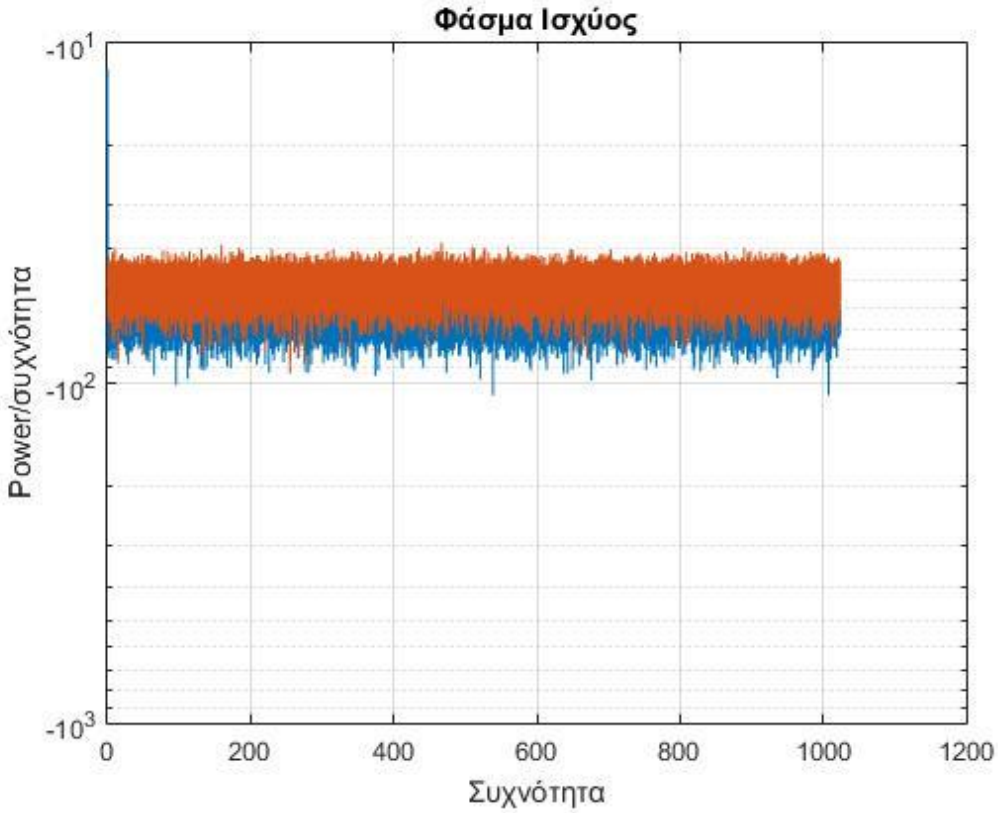
```
%======================== Ερώτημα 4 ==========================
```

```
%δείγματα
f = 2048;
t = 0:1/f:1-1/f;
%τα σύμβολα
s(:,:) = sym_dia(:,:);
%παίρνω το μέγεθος
m = length(s);
%κάνω FOURIER
four = fft(s);
four = four(1:m/2+1);
%τετράγωνο του μέτρου του
sfour = (1/(f*m)) * abs(four).^2;
%συχνότητες
sxynot = 0:f/length(s):f/2;
```

```
%δείγματα
f = 2048;
t = 0:1/f:1-1/f;
%τα σύμβολα
s(:,:) = sym_diaf(:,:);
%παίρνω το μέγεθος
m = length(s);
%κάνω FOURIER
four = fft(s);
four = four(1:m/2+1);
%τετράγωνο του μέτρου του
sfourf = (1/(f*m)) * abs(four).^2;
%συχνότητες
freqf = 0:f/length(s):f/2;
```

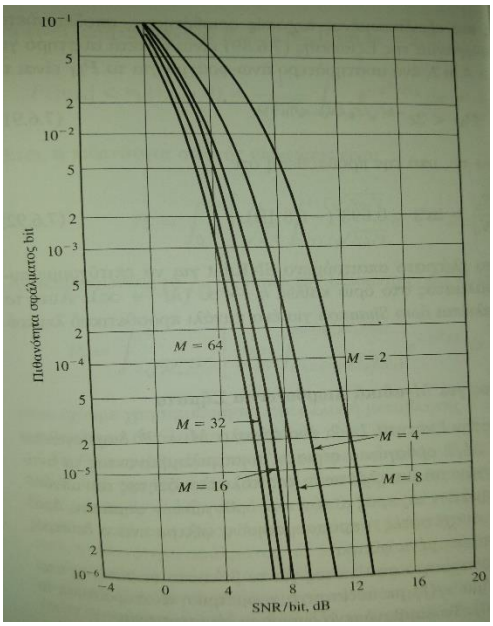
```
%σχεδίαση διαγράμματος
semilogy(freqf,10*log10(sfourf))
hold on
semilogy(sxynot,10*log10(sfour))
grid on
title('Φάσμα Ισχύος')
xlabel('Συχνότητα')
ylabel('Power/συχνότητα')
```

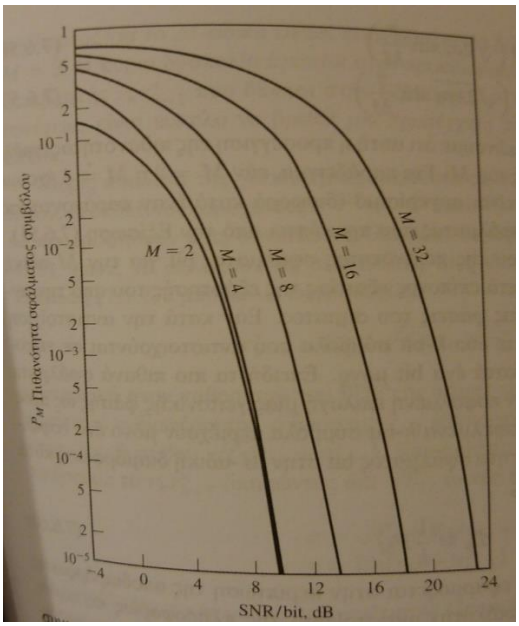
Γνωρίζουμε ότι όσον αφορά το εύρος ζώνης στα 4-FSK όταν αυξάνεται το M μειώνεται το SNR άρα και το έυρος ζώνης W αφού
W<= log(1+SNR) . Το αντίθετο συμβαίνει στα 4 -PSK.



Όπως παρατηρούμε από τη γραφική παράσραση στο 4-PSK χρησιμοποιείται μεγαλύτερο εύρος ζώνης και σχεδόν διπλάσιο από το 4 - FSK.

5. Σύγκριση εικονών 7.63 και 7.57





Για τα PSK όσο μεγαλώνει το M τόσο μεγαλώνει το SNR, δηλαδή τόσο περισσότερο μεγαλώνει η πιθανότητα σφάλματος συμβόλου SER, ενώ για τα FSK όσο μεγαλώνει το M τόσο μικραίνει το SNR, δηλαδή τόσο περισσότερο μικραίνει και η πιθανότητα σφάλματος BER.

Από την άλλη μεριά το βασικό χαρακτηριστικό του PSK είναι ότι **για καθορισμένο ρυθμό bit Rb , όταν αυξάνεται το M μειώνεται το απαιτούμενο εύρος ζώνης** για αυτό και αυξάνεται και η πιθανότητα SER. Οπότε η αύξηση του M στα PSK **αυξάνει την απόδοση του εύρους ζώνης** , και ταυτόχρονα μειώνεται η απόδοση ισχύος.Αυτό δικαιολογείται από την διάσταση του χώρου σημάτωνη οποία είναι σταθερή και ανεξάρτητη του M , ενώ στα FSK εξαρτάται απο το M.

Άρα για λίγα σύμβολα **συμφέρει** το PSK ενώ για πολλά το FSK.