



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
TECHNICAL UNIVERSITY OF CRETE

CDMA - Άσκηση 5(bonus)

Οικονομίδης Κωνσταντίνος (ΑΜ:2021030075)
Διδάσκων: Αθανασιος Π. Λιάβας

Πολυτεχνείο Κρήτης - Σχολή ΗΜΜΥ
Ασύρματες Επικοινωνίες

Χειμερινό εξάμηνο 2025-2026

Στο πρώτο μέρος της άσκησης, θα μελετήσουμε το uplink ενός απλού DS-CDMA συστήματος με K συγχρονισμένους χρήστες (ενδεικτικά, $K=1,2,3,4,5$).

Υποθέστε ότι η επικοινωνία διεξάγεται με πακέτα μήκους M δυαδικών ισοπίθανων συμβόλων από το αλφάβητο $X=\{+1,-1\}$ (ενδεικτικά $M=100$). Κάθε χρήστης χρησιμοποιεί κώδικα μήκους N (ενδεικτικά $N=32,64,128$) με χαρακτηριστικά λευκού θορύβου. Ένας απλοϊκός τρόπος δημιουργίας του κώδικα κάθε χρήστη έχει ως εξής:

$$c_k = \frac{1}{\sqrt{N}} \text{sign}(\text{rand}(N, 1)), \quad k = 1, \dots, K.$$

Ο κώδικας κάθε χρήστη παραμένει σταθερός καθόλη τη διάρκεια του πειράματος. Έστω ότι το κανάλι κάθε χρήστη έχει μιγαδική κρουστική απόκριση μήκους L (ενδεικτικά, $L=3,5$). Για κάθε πακέτο, η κρουστική απόκριση των καναλιών παραμένει σταθερή. Από πακέτο σε πακέτο οι κρουστικές αποκρίσεις αλλάζουν και οι τιμές που παίρνουν είναι ανεξάρτητες από τις προηγούμενες τιμές με κατανομή $h_k \sim CN(0, \frac{1}{L} \mathbf{I}_L)$, για $k=1, \dots, K$. Στην έξοδο του καναλιού, προστίθεται λευκός κυκλικός Gaussian θόρυβος με διασπορά σ_N^2 . Μας ενδιαφέρει η αποκωδικοποίηση του χρήστη 1 στον δέκτη (προφανώς, αντίστοιχη διαδικασία υλοποιείται για τους άλλους χρήστες). Συνεπώς, το κανάλι του χρήστη 1 θεωρείται γνωστό στον δέκτη.

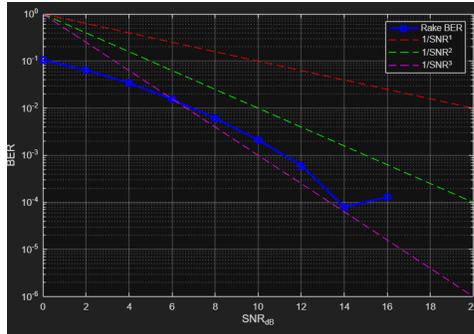
Αρχικά θεωρούμε $K=1$, δηλαδή, το σύστημα έχει μόνο ένα χρήστη.

1. Να υλοποιήσετε τον δέκτη Rake και να σχεδιάσετε το BER για $SNR_{dB}=[0:2:20]$, όπου:

$$SNR_{dB} = 10 \log_{10} \frac{1}{\sigma_N^2}$$

Να σχεδιάσετε τις συναρτήσεις $f_i = \frac{1}{SNR^i}$ για $i=1,2,3$ και να προσπαθήσετε να ανιχνεύσετε την τάξη διαφοροποίησης του σχήματος αυτού (δείτε ποιά είναι η κλίση του BER για αρκετά μεγάλο SNR).

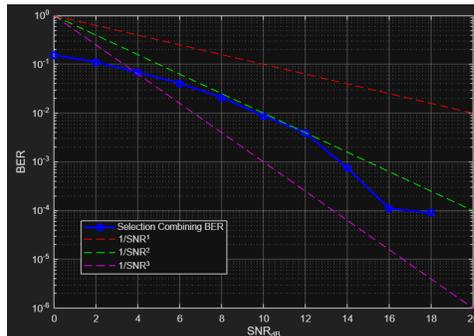
Σε αυτό το ερώτημα, υλοποιήθηκε ο δείκτης Rake και σχεδιάστηκε το BER για $SNR_{dB} = [0:2:20]$ καθώς και οι συναρτήσεις $f_i = \frac{1}{SNR^i}$ και πήραμε το εξής αποτέλεσμα από τον κώδικα:



Φαίνεται, ότι για έναν χρήστη η καμπύλη του BER μετά τα 10-12dB πέφτει με τον ίδιο ρυθμό με την συνάρτηση $\frac{1}{SNR^L}$. Άρα το σύστημα έχει diversity = L, όσες και οι διαδρομές του καναλιού.

2. Να υλοποιήσετε δέκτη Rake με 1 finger χρησιμοποιώντας μόνο τον πιο ισχυρό συντελεστή του καναλιού κάθε φορά (ο δέκτης αυτό καλείται selection combining) και να σχεδιάσετε το BER. Να σχεδιάσετε τις συναρτήσεις $f_i = \frac{1}{SNR^i}$ για $i=1,2,3$ και να προσπαθήσετε να ανιχνεύσετε την τάξη διαφοροποίησης του σχήματος αυτού (δείτε ποιά είναι η κλίση του BER για αρκετά μεγάλο SNR).

Εδώ, πήραμε το εξής αποτέλεσμα από τον κώδικα:

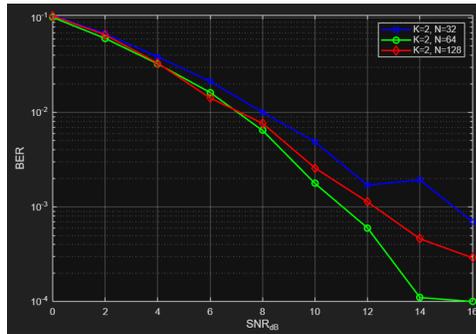


Εδώ ο δέκτης μας, όπως αναφέρεται και στην εκφώνηση, χρησιμοποιείται ο πιο ισχυρός συντελεστής του καναλιού, άρα είναι Selection Combining. Φαίνεται ότι η επίδοση του δεν είναι τόσο καλή, όσο με τον RAKE και προσεγγίζει την $\frac{1}{SNR^2}$ και επομένως δεν εκμεταλλεύεται και τις 3 διαδρομές του καναλιού.

Στη συνέχεια, να εισάγετε στο σύστημα δεύτερο χρήστη.

1. Να χρησιμοποιήσετε τον Rake με L fingers για την ανάκτηση της πληροφορίας του πρώτου χρήστη. Να υπολογίσετε το BER για $SNR_{dB} = [0:2:16]$, όπου το SNR ορίζεται όπως προγουμένως. Να συγκρίνεται το BER με αυτό του συστήματος με έναν χρήστη. Τι παρατηρείται για διαφορετικά N;

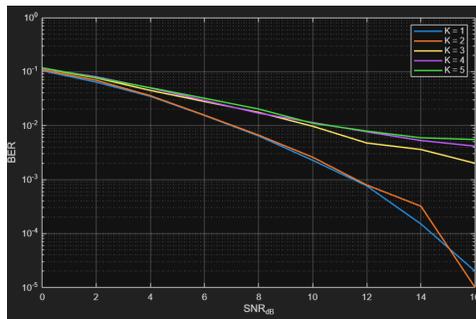
Με βάση τον κώδικα, λάβαμε το εξής αποτέλεσμα:



Η προσθήκη του δεύτερου χρήστη, επηρεάζει την απόδοση του συστήματος. Παρατηρείται ότι όσο μεγαλώνει το $N(32-64-128)$, τόσο το BER βελτιώνεται. Αυτό συμβαίνει καθώς οι μεγαλύτεροι κώδικες επιτρέπουν στον δέκτη να διαχωρίζει καλύτερα τα σήματα που λαμβάνει.

2. Στη συνέχεια, να εισάγετε περισσότερους χρήστες. Τι παρατηρείτε;

Ο κώδικας έδωσε το εξής αποτέλεσμα:



Η εισαγωγή περισσότερων χρηστών, επιφέρει μείωση στην επίδοση του συστήματος. Όσο ο αριθμός των χρηστών αυξάνεται, τόσο αυξάνεται και το BER.

Στο δεύτερο μέρος της άσκησης, θα μελετήσουμε τη διαμόρφωση OFDM.

1 Να δημιουργήσετε τυχαίο μιγαδικό κανάλι $h_l, l = 0, \dots, L - 1$, με h_i ανεξάρτητα όμοια κατανομημένα, $h_i \sim CN(0, \frac{1}{L})$ (ενδεικτικά, $L=4$).

Σε αυτό το ερώτημα, θεωρήσαμε κανάλι πολλαπλών διδρομών με $L=4$ συντελεστές. Αυτοί οι συντελεστές μοντελοποιήθηκαν ως ανεξάρτητες μιγαδικές Γκαουσιανές τυχαίες μεταβλητές με κατανομή $h_l \sim CN(0, \frac{1}{L})$.

2. Να δημιουργήσετε 4-QAM είσοδο $\tilde{d}[k], k=1, \dots, N$ με τιμές $\pm 1 \pm j$, ισοπίθανες (ενδεικτικά, $N=64, 128$).

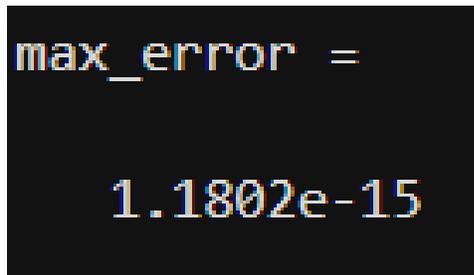
Εδώ δημιουργήθηκαν τα 4-QAM σύμβολα στο πεδίο των συχνοτήτων. Για κάθε OFDM σύμβολο εφαρμόστηκε IFFT με μέγεθος N για να πάρουμε το κατάλληλο σήμα στον χρόνο και προσθέσαμε το κατάλληλο κυκλικό πρόθεμα. Τέλος το σήμα μπαίνει μέσα από κανάλι h_i .

3. Να μεταδώσετε τα $\tilde{d}[k]$ μέσω του καναλιού με χρήση διαμόρφωσης OFDM.

Τα σύμβολα μεταδόθηκαν με κανάλι με πολλές διαδρομές με διαμόρφωση OFDM με τον εξής τρόπο. Εφαρμόζεται IFFT για να μετατραπούν στο πεδίο του χρόνου, προστέθηκε κυκλικό πρόθεμα και στη συνέχεια το σήμα μπήκε μέσα από κανάλι h_i .

4. Σε αθόρυβο περιβάλλον, να πεισθείτε ότι το frequency selective κανάλι έχει μετατραπεί σε N παράλληλα flat κανάλια.

Από τον κώδικα πήραμε το εξής αποτέλεσμα:

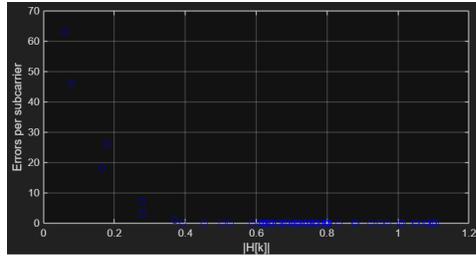


```
max_error =  
1.1802e-15
```

Σε αθόρυβο περιβάλλον φαίνεται ότι το frequency selective κανάλι μετατρέπεται σε N παράλληλα flat κανάλια μέσω OFDM. Αφού αφαιρέσαμε το κυκλικό πρόθεμα και εφαρμόσαμε FFT στον δέκτη, ισχύει η σχέση $Y[k] = H[k]\tilde{d}[k]$. Ο έλεγχος έδωσε μέγιστο σφάλμα $\max|Y[k] - H[k]\tilde{d}[k]| \approx 1.18 \cdot 10^{-15}$. Αυτό συμβαίνει καθώς το κυκλικό πρόθεμα μετατρέπει τη γραμμική συνέλιξη σε κυκλική, η οποία γίνεται γινόμενο στη συχνότητα.

5. Στην έξοδο του καναλιού, να προσθέσετε λευκό Gaussian θόρυβο ώστε να έχετε μέσο SNR (ενδεικτικά, SNR=15dB). Να συσχετίσετε τις θέσεις στις οποίες εμφανίζονται σφάλματα απόφασης με αυτές στις οποίες το κανάλι έχει "μικρό" μέτρο απόκρισης συχνότητων. Τι παρατηρείτε;

Η εκτέλεση του κώδικα έδωσε το εξής αποτέλεσμα:

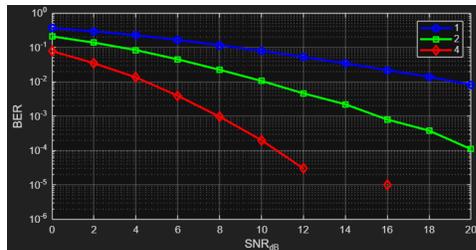


Με την προσθήκη λευκού Gaussian θορύβου για μέσο SNR=15dB, παρατηρήθηκε ότι τα σφάλματα συγκεντρώθηκαν σε υποφορείς όπου το μέτρο της απόκρισης συχνότητας $|H[k]|$ είναι μικρό. Στα deep fades του καναλιού ο ισοσταθμιστής διαιρεί με μικρή τιμή της απόκρισης συχνότητας και άρα ενισχύει τον θόρυβο. Για μεγάλες τιμές της απόκρισης συχνότητας, τα σφάλματα είναι πολύ λιγότερα.

Στη συνέχεια, θα προσπαθήσετε να εξάγετε διαφοροποίηση με "επανάληψη στις συχνότητες" χρησιμοποιώντας ανεξάρτητες υλοποιήσεις του καναλιού για διάφορες τιμές του μέσου λαμβανόμενου SNR.

1. Να χρησιμοποιήσετε επανάληψη στις συχνότητες για να εξάγετε διαφοροποίηση τάξης 1,2 και 4.

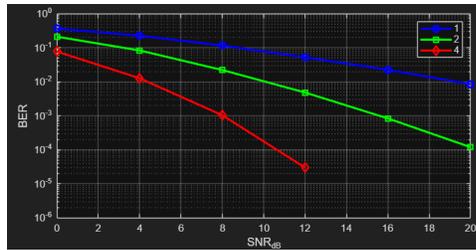
Ο κώδικας οδηγεί στο παρακάτω αποτέλεσμα:



Παρατηρείται ότι όσο αυξάνεται ο αριθμός των επαναλήψεων ($R=1,2,4$), τόσο η καμπύλη BER μειώνεται και η κλίση γίνεται πιο απότομη στα υψηλά SNR. Επομένως επιτυγχάνεται διαφοροποίηση τάξης R , μειώνοντας την πιθανότητα deep fading σε όλα τα κανάλια και άρα βελτιώνεται η αξιοπιστία της μετάδοσης.

2. Να σχεδιάσετε το BER για $SNR_{dB}=[0:4:20]$.

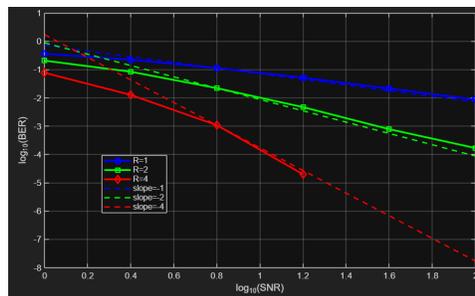
Από τον κώδικα, πήραμε το εξής αποτέλεσμα:



Παρατηρείται ότι και εδώ όσο αυξάνεται η τάξη διαφοροποίησης, τόσο η καμπύλη BER έχει μεγαλύτερη κλίση και μειώνεται πιο γρήγορα με το SNR. Επίσης επιτυγχάνεται διαφοροποίηση τάξης R.

3. Να σχεδιάσετε την ποσότητα $\log_{10} SNR$ ως προς την ποσότητα $\log_{10} BER$. Να προσπαθήσετε να επιβεβαιώσετε ότι επιτυγχάνεται την επιδιωκόμενη τάξη διαφοροποίησης συγκρίνοντας γραφικά την κλίση του γραφήματος με την κλίση κατάλληλα επιλεγμένων ευθειών γραμμών.

Τρέχοντας τον κώδικα, προκύπτει το παρακάτω αποτέλεσμα:



Από το γράφημα, παρατηρούμε ότι στα μεσαία και υψηλά SNR οι καμπύλες είναι σχεδόν παράλληλες με ευθείες κλίσης -1, -2, -4. Επομένως το BER μειώνεται με την σχέση $BER \propto SNR^{-R}$ και πετυχαίνουμε τάξη διαφοροποίησης R.

Παρακάτω ακολουθεί ο κώδικας που χρησιμοποιήθηκε για την υλοποίηση της άσκησης και βασίστηκε στον κώδικα που αναρτήθηκε στο eclass:

Listing 1: Κώδικας MATLAB

```
1
2 close all;
3 clc;
4
5 %% Erotima 1
6 K1 = 1;
7 M1 = 100;
8 N1 = 64;
9 L1 = 3;
10 SNR_dB_1 = 0:2:20;
11 SNR_lin_1 = 10.^(SNR_dB_1/10);
12 num_packets_1 = 1000;
13
14 c1 = (1/sqrt(N1)) * sign(randn(N1,1));
15 ber_rake = zeros(size(SNR_dB_1));
16
17 for s = 1:length(SNR_dB_1)
18     errors = 0;
19     sigma2_N = 1/SNR_lin_1(s);
20     for p = 1:num_packets_1
21         bits=sign(randn(M1, 1));
22         tx=kron(bits, c1);
23         h= (sqrt(1/(2*L1))) * (randn(L1,1) + 1i*randn(L1,1));
24         chan_out =conv(tx, h);
25         noise = sqrt(sigma2_N/2) * (randn(length(chan_out),1) + 1i*randn(
                length(chan_out),1));
26         rx=chan_out + noise;
27
28         dec = zeros(M1, 1);
29         for m = 1:M1
30             idx = (m-1)*N1 + (1:N1);
31             y_mrc = 0;
32             for l = 1:L1
33                 y_mrc = y_mrc + conj(h(l)) * (c1' * rx(idx + (l-1)));
34             end
35             dec(m) = sign(real(y_mrc));
36         end
37         errors = errors + sum(bits ~= dec);
38     end
39     ber_rake(s) = errors / (M1 * num_packets_1);
40 end
41
42 figure(1);
43 semilogy(SNR_dB_1, ber_rake, 'bo-', 'LineWidth', 2); hold on;
44 semilogy(SNR_dB_1, 1./SNR_lin_1.^3, 'm--');
45 grid on; xlabel('SNR_{dB}'); ylabel('BER');
```

```

46 legend('Rake MRC BER', '1/SNR^3');
47
48 %%Erotima 2
49 K2 = 1;
50 M2 = 100;
51 N2 = 64;
52 L2 = 3;
53 SNR_dB_2 = 0:2:20;
54 SNR_lin_2 = 10.^(SNR_dB_2/10);
55 num_packets_2 = 1000;
56
57 c2 = (1/sqrt(N2)) * sign(randn(N2,1));
58 ber_selection = zeros(size(SNR_dB_2));
59
60 for s = 1:length(SNR_dB_2)
61     errors = 0;
62     sigma2_N = 1/SNR_lin_2(s);
63     for p = 1:num_packets_2
64         bits = sign(randn(M2, 1));
65         tx = kron(bits, c2);
66         h = (sqrt(1/(2*L2))) * (randn(L2,1) + 1i*randn(L2,1));
67         [~, max_idx] = max(abs(h));
68         h_best = h(max_idx);
69
70         chan_out = conv(tx, h);
71         noise = sqrt(sigma2_N/2) * (randn(length(chan_out),1) + 1i*randn(
72             length(chan_out),1));
73         rx =chan_out + noise;
74
75         dec=zeros(M2, 1);
76         for m = 1:M2
77             idx=(m-1)*N2 + (1:N2);
78             y_sc=c2' * rx(idx + (max_idx-1));
79             dec(m)=sign(real(conj(h_best) * y_sc));
80         end
81         errors = errors + sum(bits ~= dec);
82     end
83     ber_selection(s) = errors / (M2 * num_packets_2);
84 end
85
86 figure(2);
87 semilogy(SNR_dB_2, ber_selection, 'ro-', 'LineWidth', 2); hold on;
88 semilogy(SNR_dB_2, 1./SNR_lin_2.^3, 'm--');
89 grid on; xlabel('SNR_{dB}'); ylabel('BER');
90 legend('Selection Combining BER', '1/SNR^3');
91
92 %% Erotima3
93 K3 = 2;
94 M3 = 100;
95 L3 = 3;

```

```

96 N_vals = [32, 64, 128];
97 SNR_dB_3 = 0:2:16;
98 SNR_lin_3 = 10.^(SNR_dB_3/10);
99 num_packets_3 = 1000;
100
101 figure(3);
102 colors = ['b', 'g', 'r']; markers = ['s', 'o', 'd'];
103
104 for n_idx = 1:length(N_vals)
105     N3 = N_vals(n_idx);
106     ber_k2 = zeros(size(SNR_dB_3));
107     c3 = (1/sqrt(N3))*sign(randn(N3, K3));
108
109     for s = 1:length(SNR_dB_3)
110         errors = 0;
111         sigma2_N = 1/SNR_lin_3(s);
112         for p = 1:num_packets_3
113             bits = sign(randn(M3, K3));
114             tx1 = kron(bits(:,1), c3(:,1));
115             tx2 = kron(bits(:,2), c3(:,2));
116             h1 = (sqrt(1/(2*L3))) * (randn(L3,1) + 1i*randn(L3,1));
117             h2 = (sqrt(1/(2*L3))) * (randn(L3,1) + 1i*randn(L3,1));
118             rx = conv(tx1, h1) + conv(tx2, h2) + sqrt(sigma2_N/2)*(randn(N3*M3+L3
119                 -1,1) + 1i*randn(N3*M3+L3-1,1));
120
121             dec = zeros(M3, 1);
122             for m = 1:M3
123                 idx = (m-1)*N3 + (1:N3);
124                 y_mrc = 0;
125                 for l = 1:L3
126                     y_mrc = y_mrc + conj(h1(l)) * (c3(:,1)' * rx(idx + (l-1)))
127                     ;
128                 end
129                 dec(m) = sign(real(y_mrc));
130
131             end
132             errors = errors + sum(bits(:,1) ~= dec);
133         end
134         ber_k2(s) = errors/(M3 * num_packets_3);
135     end
136
137     semilogy(SNR_dB_3, ber_k2, [colors(n_idx) '-' markers(n_idx)], 'LineWidth'
138         , 1.5); hold on;
139
140 end
141 grid on;
142 xlabel('SNR_{dB}');
143 ylabel('BER');
144 legend('K=2, N=32', 'K=2, N=64', 'K=2, N=128');
145
146 %% OFDM Erotima 1

```

```

144
145 ofdm_L = 4;    %arithmos taps kanaliou gia OFDM
146
147 % h_l = CN(0,1/L)
148 h_ofdm = sqrt(1/(2*ofdm_L)) *(randn(ofdm_L,1)+1i*randn(ofdm_L,1));
149
150 %% OFDM Erotima 2
151
152 ofdm_Nfft = 64;
153 ofdm_Nsym=100;
154 ofdm_cpLen=ofdm_L-1;
155
156 %4-QAM eisodos d_tilde[k] ({±1 ± j}/sqrt(2))
157 ofdm_a=sign(randn(ofdm_Nfft,ofdm_Nsym));
158 ofdm_b=sign(randn(ofdm_Nfft, ofdm_Nsym));
159 ofdm_d_tilde=(ofdm_a + 1i*ofdm_b)/sqrt(2);
160
161 %OFDM diamorfosi IFFT ana OFDM symbol
162 ofdm_x = ifft(ofdm_d_tilde, ofdm_Nfft, 1);
163
164 %Kykliko prothema
165 ofdm_x_cp=[ofdm_x(end-ofdm_cpLen+1:end, :); ofdm_x];
166
167 %Seiriopoiisi gia metadosi
168 ofdm_tx=ofdm_x_cp(:);
169
170 %Dieleusi apo kanali
171 ofdm_rx_noNoise =conv(ofdm_tx, h_ofdm);
172
173 %%OFDM Erotima 3
174
175 ofdm_Nfft = 64;
176 ofdm_Nsym = 100;
177 ofdm_cpLen = ofdm_L-1;
178
179 %IFFT
180 ofdm_x = ifft(ofdm_d_tilde, ofdm_Nfft, 1);
181
182 %Kykliko prothema
183 ofdm_x_cp = [ofdm_x(end-ofdm_cpLen+1:end, :); ofdm_x];
184
185 %Seiriopoiisi gia metadosi
186 ofdm_tx =ofdm_x_cp(:);
187
188 %Dieleusi apo kanali
189 ofdm_rx_noNoise=conv(ofdm_tx, h_ofdm);
190
191 %%OFDM erotima4
192
193 %Mikos block me CP
194 ofdm_blockLen = ofdm_Nfft+ofdm_cpLen;

```

```

195
196 %Bgazoume to epiplon tail apo ti syneliksi
197 ofdm_rx_trim=ofdm_rx_noNoise(1 : ofdm_blockLen * ofdm_Nsym);
198
199 %Epanafora se blocks
200 ofdm_rx_blocks =reshape(ofdm_rx_trim,ofdm_blockLen,ofdm_Nsym);
201
202 %Afairesi CP
203 ofdm_rx_noCP=ofdm_rx_blocks(ofdm_cpLen+1:end, :);
204
205 % FFT
206 ofdm_Y=fft(ofdm_rx_noCP, ofdm_Nfft, 1);
207
208 %Ypologismos H[k]
209 ofdm_H = fft(h_ofdm,ofdm_Nfft);
210
211 %Elegxos swesi Y[k] = H[k] * d_tilde[k]
212 ofdm_error=ofdm_Y-(ofdm_H .* ofdm_d_tilde);
213 max_error =max(abs(ofdm_error(:)))
214
215 %%OFDM AWGN me SNR=15dB
216
217 ofdm_SNRdB = 15;
218 ofdm_SNRlin = 10^(ofdm_SNRdB/10);
219
220 %Kanali xwris thorybo
221 ofdm_rx_noNoise=conv(ofdm_tx, h_ofdm);
222
223
224 %Prosthiki AWGN stin eksodo
225 ofdm_Psig = mean(abs(ofdm_rx_noNoise).^2);
226
227 %Thoryvos etsi wste Psig/Pn = SNR
228 ofdm_sigma2N =ofdm_Psig/ofdm_SNRlin;
229
230 ofdm_noise=sqrt(ofdm_sigma2N/2) * (randn(length(ofdm_rx_noNoise),1) + 1i*randn
    (length(ofdm_rx_noNoise),1));
231 ofdm_rx = ofdm_rx_noNoise + ofdm_noise;
232
233 %OFDM Receiver
234 ofdm_blockLen = ofdm_Nfft+ofdm_cpLen;
235
236 ofdm_rx_trim = ofdm_rx(1 : ofdm_blockLen * ofdm_Nsym);
237 ofdm_rx_blocks = reshape(ofdm_rx_trim, ofdm_blockLen, ofdm_Nsym);
238
239 ofdm_rx_noCP = ofdm_rx_blocks(ofdm_cpLen+1:end, :);
240 ofdm_Y = fft(ofdm_rx_noCP, ofdm_Nfft, 1);
241
242 %Frequency response H[k]
243 ofdm_H = fft(h_ofdm, ofdm_Nfft);
244 ofdm_H_mat = repmat(ofdm_H, 1, ofdm_Nsym);

```

```

245
246 %Isostathmisi d_hat = Y ./ H
247 ofdm_d_hat=ofdm_Y ./ ofdm_H_mat;
248
249 %4-QAM hard decision
250 ofdm_dec_real=sign(real(ofdm_d_hat));
251 ofdm_dec_imag=sign(imag(ofdm_d_hat));
252 ofdm_d_dec =(ofdm_dec_real+1i*ofdm_dec_imag)/sqrt(2);
253
254 %Lathoi apo ypoforea
255 ofdm_errMat = abs(ofdm_d_dec - ofdm_d_tilde) > 0;
256 ofdm_errCountPerSub = sum(ofdm_errMat, 2); % Nfft*1
257
258 %Grafima
259 figure;
260 plot(abs(ofdm_H), ofdm_errCountPerSub, 'bo');
261 grid on;
262 xlabel('|H[k]|');
263 ylabel('Errors per subcarrier');
264
265 %% OFDM Frequency Repetition Diversity gia 1,2,4
266
267 ofdmRep_SNRdB = 0:4:20;
268 ofdmRep_SNRlin=10.^(ofdmRep_SNRdB/10);
269 %ofdmRep_SNRdB =0:2:20; Antikathistw ta 2 panw me auto gia to erwtima 4.1
270 ofdmRep_SNRlin=10.^(ofdmRep_SNRdB/10);
271
272 ofdmRep_Rvals=[1 2 4];
273 ofdmRep_Nsym=200;
274 ofdmRep_numPackets=500;
275
276
277 ber_rep =zeros(length(ofdmRep_Rvals),length(ofdmRep_SNRdB));
278
279 for rIdx = 1:length(ofdmRep_Rvals)
280     ofdmRep_R = ofdmRep_Rvals(rIdx);
281
282
283     for s = 1:length(ofdmRep_SNRdB)
284         ofdmRep_err=0;
285         ofdmRep_tot=0;
286
287         ofdmRep_sigma2 = 1 / ofdmRep_SNRlin(s);
288
289         for p = 1:ofdmRep_numPackets
290             a = sign(randn(ofdmRep_Nsym,1));
291             b = sign(randn(ofdmRep_Nsym,1));
292             d = (a + 1i*b)/sqrt(2);
293
294             %Epanalipseis stis syxnotites
295             % H_r = CN(0,1) independent

```

```

296         H =(randn(ofdmRep_Nsym, ofdmRep_R) + 1i*randn(ofdmRep_Nsym,
297                ofdmRep_R))/sqrt(2);
298
299         N = sqrt(ofdmRep_sigma2/2) * (randn(ofdmRep_Nsym, ofdmRep_R) + 1i*
300                randn(ofdmRep_Nsym, ofdmRep_R));
301
302         Y =H .* repmat(d, 1, ofdmRep_R) + N;
303
304         num=sum(conj(H) .* Y, 2);
305         den=sum(abs(H).^2, 2);
306         z =num ./ den;
307
308         % Hard decision 4-QAM
309         d_hat = (sign(real(z))+1i*sign(imag(z)))/sqrt(2);
310
311         ofdmRep_err = ofdmRep_err + sum(d_hat ~= d);
312         ofdmRep_tot = ofdmRep_tot + ofdmRep_Nsym;
313     end
314
315     ber_rep(rIdx, s) =ofdmRep_err/ofdmRep_tot;
316 end
317
318 end
319
320 %Grafima
321 figure;
322 semilogy(ofdmRep_SNRdB, ber_rep(1,:), 'bo-', 'LineWidth', 2);
323 hold on;
324 semilogy(ofdmRep_SNRdB, ber_rep(2,:), 'gs-', 'LineWidth', 2);
325 semilogy(ofdmRep_SNRdB, ber_rep(3,:), 'rd-', 'LineWidth', 2);
326 grid on;
327 xlabel('SNR_{dB}'); ylabel('BER');
328 legend('1', '2', '4');
329 axis([0 20 1e-6 1]);
330
331 %% log10(BER) vs log10(SNR)
332
333 %Elegxos
334 mask1 = ber_rep(1,:)>0;
335 mask2 = ber_rep(2,:)>0;
336 mask4 =ber_rep(3,:)>0;
337
338 % log10(SNR)
339 x=log10(ofdmRep_SNRlin);
340
341 y1=log10(ber_rep(1,:));
342 y2=log10(ber_rep(2,:));
343 y4=log10(ber_rep(3,:));
344

```

```

345 figure;
346 plot(x(mask1), y1(mask1), 'bo-', 'LineWidth', 2);
347 hold on;
348 plot(x(mask2), y2(mask2), 'gs-', 'LineWidth', 2);
349 plot(x(mask4), y4(mask4), 'rd-', 'LineWidth', 2);
350 grid on;
351 xlabel('log_{10}(SNR)');
352 ylabel('log_{10}(BER)');
353 legend('R=1', 'R=2', 'R=4', 'Location','southwest');
354
355 %Eythies anaforas
356 snr_ref_dB = 8;
357 [~, idx_ref] = min(abs(ofdmRep_SNRdB-snr_ref_dB));
358
359 x_ref = x(idx_ref);
360
361
362 % Sygkrisi
363 yref1 = y1(idx_ref);
364 yref2 = y2(idx_ref);
365 yref4 =y4(idx_ref);
366
367
368 yline1 = yref1-1*(x-x_ref);
369 yline2 = yref2-2*(x-x_ref);
370 yline4 = yref4-4*(x-x_ref);
371
372 plot(x, yline1, 'b--', 'LineWidth', 1.5);
373 plot(x, yline2, 'g--', 'LineWidth', 1.5);
374 plot(x, yline4, 'r--', 'LineWidth', 1.5);
375
376 legend('R=1', 'R=2', 'R=4', 'slope=-1', 'slope=-2', 'slope=-4', 'Location','
southwest');

```