Ψηφιακές Τηλεπικοινωνίες

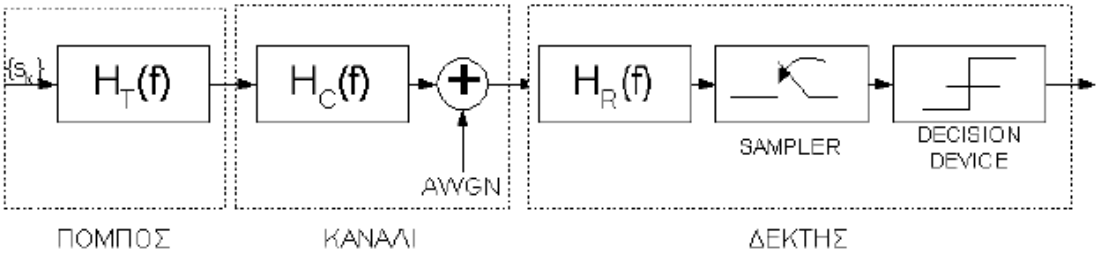
2η Εργαστηριακή Άσκηση
Εξομοίωση Τηλεπικοινωνιακού Συστήματος Βασικής Ζώνης

Παπαρροδοπούλου Αναστασία ΑΜ 3873
26/02/14

Εισαγωγή:

Το περιεχόμενο της δεύτερης αυτής εργαστηριακής άσκησης, είναι η εξομοίωση ενός τηλεπικοινωνιακού συστήματος βασικής ζώνης και η εξέταση της επίδοσής του για διάφορα είδη διαμόρφωσης σε ιδανικό και μη ιδανικό κανάλι. Συγκεκριμένα εξετάζουμε την επίδοση για 2-PAM, 4-PAM και 4-QAM διαμορφώσεις.

Το σύστημά μας έχει την εξής διάταξη:



Στη συνέχεια, στο Α' μέρος της άσκησής μας, θα αναλύσουμε το κάθε επιμέρους κομμάτι.

Α' Μέρος: Περιγραφή Τηλεπικοινωνιακού Συστήματος

Όπως αναφέραμε λοιπόν και παραπάνω, θα κάνουμε μία σύντομη ανάλυση της διαδικασίας του τηλεπικοινωνιακού συστήματος.

Φίλτρα Πομπού/Δέκτη

Γνωρίζουμε από τη θεωρία ότι η ακολουθία συμβόλων s(t) δεν αποστέλλεται απευθείας στο κανάλι, καθώς έτσι θα αύξανε σημαντικά το απαιτούμενο εύρος ζώνης. Μορφοποιείται λοιπόν το σήμα από ένα φίλτρο πομπού ώστε να περιοριστεί το συχνοτικό του περιεχόμενο. Όμως, το κανάλι από το οποίο θα περάσει το σήμα μας είναι άγνωστο, οπότε είναι αδύνατο να υπολογίσουμε τα βέλτιστα φίλτρα πομπού και δέκτη. Έτσι τα υλοποιούμε ως φίλτρα τετραγωνικής ρίζας ανυψωμένου συνημιτόνου με παράγοντα αναδίπλωσης 0.3, η κρουστική απόκριση των οποίων συνελίσσεται με το εκάστοτε σήμα. Ιδανικά, η κρουστική απόκριση έχει άπειρους συντελεστές, ωστόσο στην πράξη επιλέγεται ένας συνολικός 6-8 περιόδων συμβόλου (6-8Ts). Συμπερασματικά, για 6 περιόδους χρησιμοποιούμε το διάστημα [-3Ts, 3Ts].

Στη MATLAB χρησιμοποιούμε την ενδογενή συνάρτηση rcosfir(), το αποτέλεσμα της οποίας το κανονικοποιούμε έτσι ώστε η νόρμα του διανύσματος να είναι μοναδιαία. Με αυτό τον τρόπο τα φίλτρα δε θα μεταβάλλουν την ισχύ του σήματος εισόδου.

Υπερδειγματοληψία Ακολουθιών Εισόδου

Η είσοδος του συστήματος είναι μία ακολουθία συμβόλων τα οποία για κάθε περίπτωση αλφαβήτου είναι ισοπίθανα. Η περίοδος συμβόλου είναι Ts=1sec, όμως η προσομοίωση θα πρέπει να λειτουργήσει με υπερδειγματοληψία, άρα για σήμα εισόδου s(t), σε διάστημα T=Ts/4 λαμβάνουμε s(nT). Επομένως, στην πηγή μας προσθέτουμε 3 μηδενικά ανάμεσα σε κάθε δύο στοιχεία αυτης.

Στη MATLAB χρησιμοποιούμε την ενδογενή συνάρτηση randsrc() για να παράξουμε ακολουθία ισοπίθανων συμβόλων, βάσει αλφαβήτου που δίνουμε ως όρισμα. Επίσης, με την upsample() υπερδειγματολειπτούμε ανά 4 και αποκόπτουμε τα τελευταία 3 μηδενικά.

Κανάλι

Για τα πειράματά μας χρησιμοποιούμε δύο είδη καναλιών, ένα ιδανικό κι ένα μη ιδανικό. Στο ιδανικό απλά προστίθεται στο σήμα που εισήλθε στο κανάλι ο θόρυβος. Το μη ιδανικό κανάλι τώρα, χαρακτηρίζεται από τη δοσμένη από την εκφώνηση, κρουστική απόκριση.

h(-6 : 6) = [0.01 0.04 -0.05 0.06 -0.22 -0.5 0.72 0.36 0 0.21 0.04 0.08 0.02]

Αντίστοιχα με τα άλλα φίλτρα, το κανάλι θα πρέπει να υπερδειγματοληπτηθεί κατά 4 και μετέπειτα να υπολογιστεί η συνέλιξή του με το σήμα.

Θόρυβος Συστήματος

Στην έξοδο κάθε καναλιού και προτού εισαχθεί το σήμα στον δέκτη, προστίθεται θόρυβος σε αυτό. Ο θόρυβος είναι λευκός προσθετικός Gaussian με μηδενική μέση τιμή. Η ισχύς του ισούται με τη διασπορά του και καθορίζεται από το SNR που επιθυμούμε να έχουμε. Στη γνωστή μας από τη θεωρία μας εξίσωση, λύνουμε ως προς την διασπορά του θορύβου και στη συνέχεια πολλαπλασιάζοντας με αυτή

τυχαίες τιμές κανονικά κατανεμημένες στο [-1,1], μπορούμε να προσομοιώσουμε το θόρυβο που εν τέλει προσθέτουμε στο σήμα μας.

Αυτό που πρέπει να προσέξουμε είναι ότι στην περίπτωση των μιγαδικών αστερισμών, 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 $\sigma_n^2/2$.

Στη MATLAB υπολογίσουμε αρχικά την ισχύ του σήματος η οποία είναι ίση με το άθροισμα των τετραγώνων των απόλυτων τιμών του διανύσματος του σήματος διαιρεμένο με το μήκος του.

Διάταξη Απόφασης

Ανάλογα με την σχέση σήματος εξόδου-πιθανών τιμών των στοιχείων του, «μαντεύουμε» βάσει κριτηρίου ML για το ποιά τιμή είναι αυτή που εστάλη. Η ευκλείδια απόσταση $D(r,s_m)$ ελαχιστοποιείται όταν επιλέγουμε την πιο πιθανή έκφραση, δηλαδή τη μικρότερη τιμή του τετραγώνου της διαφοράς κάθε στοιχείου του διανύσματος εξόδου με κάθε στοιχείο του αλφαβήτου της πηγής.

Στη MATLAB έχουμε ένα προσωρινό πίνακα που σε κάθε iteration αποθηκεύει στις θέσεις του αυτά τα τετράγωνα. Σύμφωνα με το ποιο τετράγωνο είναι μικρότερο, επιλέγεται το αντίστοιχο στοιχείο ως σωστή εκπομπή.

Δέκτης

Όμοια με το φίλτρο πομπού εφαρμόζουμε το φίλτρο δέκτη στο εισερχόμενο σήμα. Στο σήμα που έχει προκύψει μετά από όλες τις συνελίξεις, αρκετοί συντελεστές αναφέρονται σε μελλοντικά δείγματα. Τα αιτιατά φίλτρα δεν είναι υλοποιήσιμα. Άρα για να μπορέσουμε να έχουμε συγχρονισμό εισόδου-εξόδου και για να μπορέσουμε να πάρουμε αργότερα απόφαση για το σήμα, μετακινούμε όλους τους συντελεστές του ώστε να είναι όλοι αιτιατοί. Αυτό έχει ως αποτέλεσμα την πρόσθεση κάποιας καθυστέρησης την οποία αφαιρούμε από την αρχή και το τέλος του διανύσματος του σήματος. Τέλος, υποδειγματοληπτούμε το σήμα μας.

Στη MATLAB αρχικά εφαρμόζουμε το φίλτρο το οποίο είναι ίδιο με το φίλτρο πομπού και μετά υπολογίζουμε τις καθυστερήσεις φίλτρου πομπού και φίλτρου δέκτη. Στην περίπτωση των μη ιδανικών καναλιών προσθέτουμε και την καθυστέρηση λόγω της συνέλιξης του καναλιού. Τέλος, η υποδειγματοληψία γίνεται ξεκινώντας από το πρώτο στοιχείο μετά τη καθυστέρηση και ανά 4 επιλέγουμε τα στοιχεία μέχρι να φτάσουμε το μέγεθος του διανύσματος εισόδου και πετάμε τα υπόλοιπα τελευταία.

-----Η διαδικασία αυτή υλοποιείται στο κομμάτι κώδικα με όνομα meros_A.m, και βρίσκεται στο παράρτημα με συγκεντρωμένους όλους τους κώδικες της εργαστηριακής άσκησης, στο τελευταίο κομμάτι της αναφοράς, μετά τη θεωρητική ανάλυση των πέντε υποερωτημάτων. -----

Β' Μέρος: Δυαδικό PAM

Το αλφάβητο της πηγής για το 2-PAM μας δίνεται από την εκφώνηση ως {-1,+1}. Για κάθε τιμή του SNR, από το 0 με βήμα 2 μέχρι το 30, εκτελούμε το κομμάτι κώδικα με όνομα meros_B.m και αποθηκεύουμε τα αποτελέσματα ξεχωριστά για τα ιδανικά και τα μη-ιδανικά κανάλια, έτσι ώστε κάθε στήλη να αντιπροσωπεύει το αποτέλεσμα του αντίστοιχου SNR.

Ακολούθως μετατρέπουμε τα σύμβολα σε bits. Έχουμε δυαδικό αλφάβητο άρα εύκολα το -1 αντιστοιχίζεται στο 0, και το +1 στο 1. Η διαδικασία αυτή δεν είναι απαραίτητη, αλλά την κάναμε για γενίκευση της μεθόδου.

Για τον υπολογισμό του BER αθροίσαμε το σύνολο των XOR λογικών πράξεων κάθε στήλης κάθε πίνακα με το διάνυσμα πηγής και διαιρέσαμε με το μήκος της. Έτσι καταλήγουμε σε δύο διανύσματα που αντιπροσωπεύουν τα BER για κάθε τιμή SNR και τα δίνουμε ως έξοδο.

Έχουμε επίσης συγγράψει κώδικα για τη γραφική απεικόνιση των παραπάνω.

Για να πάρουμε τα αποτελέσματά μας λοιπόν, εκτελούμε την εντολή:

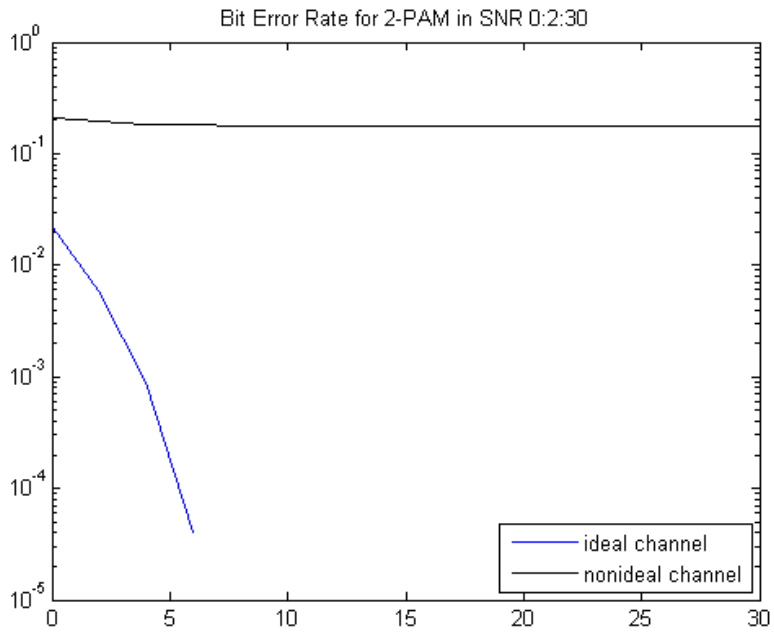
```
[ PAM2_ideal_BER , PAM2_nonideal_BER ] = meros_B
```

(η οποία συμπεριλαμβάνεται στον κώδικα του 5ου μέρους της εργαστηριακής άσκησης)

Ο πίνακας λοιπόν είναι ο εξής:

SNR(dB)	Ιδανικό Κανάλι	Μη-Ιδανικό Κανάλι
0	0.0223	0.2072
2	0.0058	0.1962
4	0.0009	0.1849
6	0.0000	0.1818
8	0	0.1769
10	0	0.1760
12	0	0.1756
14	0	0.1743
16	0	0.1746
18	0	0.1747
20	0	0.1752
22	0	0.1749
24	0	0.1757
26	0	0.1753
28	0	0.1758
30	0	0.1752

Η γραφική απεικόνιση τώρα των αποτελεσμάτων για το 2-PAM, είναι:



-----Η παραπάνω διαδικασία υλοποιείται στο κομμάτι κώδικα με όνομα meros_B.m, και βρίσκεται στο παράρτημα με τους κώδικες στο τελευταίο κομμάτι της αναφοράς.-----

Γ' Μέρος: Τετραδικό PAM

Το αλφάβητο της πηγής για το 4-PAM μας δίνεται από το την εκφώνηση του γ' μέρους της άσκησης ως τα 4 συγκεκριμένα κλάσματα. Αυτή η επιλογή τιμών φαίνεται να έγινε σκόπιμα ώστε να είναι δυνατή η σύγκριση με το 2-PAM.

Συγκεκριμένα, τα σύμβολά μας είναι ισοπίθανα και θέλουμε να διατηρήσουμε σταθερή τη μοναδιαία μέση ενέργεια του σήματος. Δηλαδή:

- για το 2-PAM έχουμε: $P_{s,2\text{-}PAM}=\sum p_i s_i=(1/2)(+1)^2+(1/2)(-1)^2=1$
- για το 4-PAM είναι: $P_{s,4\text{-}PAM}=\sum p_i s_i =(1/4)*(9\alpha^2+\alpha^2+\alpha^2+9\alpha^2)=1$

Λύνοντας το σύστημα βρίσκουμε πως το α ισούται με 1 προς τη ρίζα του 5. Αντιστοιχίζουμε λοιπόν στις τιμές του αλφαβήτου.

Το σύστημα εκτελείται όμοια με το 2-PAM, οπότε και μετατρέπουμε τα σύμβολα πηγής και εξόδων ως εξής:

$-3/\sqrt{(5)} \rightarrow 00$
$-1/\sqrt{(5)} \rightarrow 01$
$1/\sqrt{(5)} \rightarrow 10$
$3/\sqrt{(5)} \rightarrow 11$

Σημειώνουμε ότι χρησιμοποιήσαμε ξεχωριστά διπλάσιου ύψους διανύσματα σε σχέση με της αρχικής πηγής, διότι κάθε σύμβολο κωδικοποιείται σε 2 bits. Για τον υπολογισμό του BER αθροίσαμε το σύνολο των XOR λογικών πράξεων κάθε στήλης κάθε πίνακα, με το διάνυσμα που εκφράζει το σήμα της πηγής με δυαδικά ψηφία ως bits, και διαιρέσαμε με το μήκος της. Έτσι καταλήξαμε σε δύο διανύσματα που αντιπροσωπεύουν τα BER για κάθε τιμή SNR και τα δώσαμε ως έξοδο.

Έχουμε και πάλι συμπεριλάβει κώδικα για τη γραφική απεικόνιση των παραπάνω.

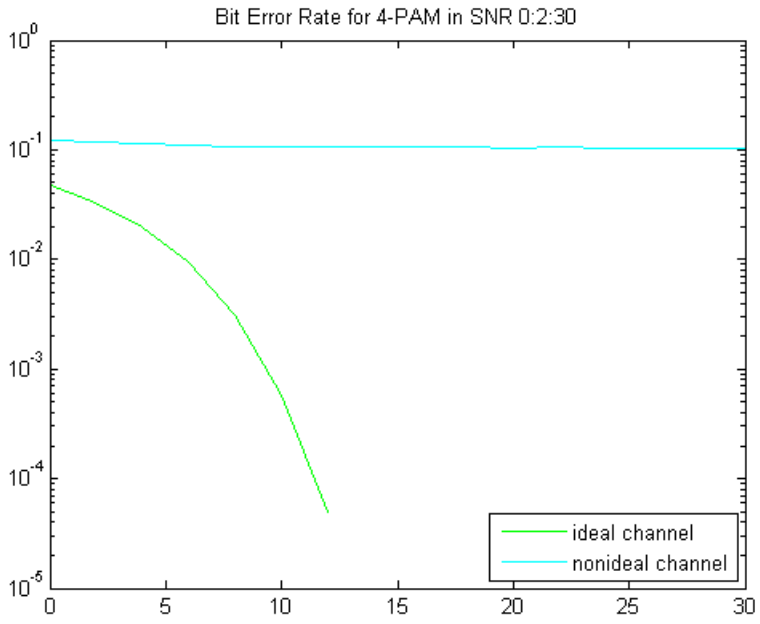
Επίσης όμοια με πριν, εκτελούμε την αποθήκευση των αποτελεσμάτων την παρακάτω εντολή:

```
[ PAM4_ideal_BER , PAM4_nonideal_BER ] = meros_C
```

Σε αυτή την περίπτωση τώρα, ο πίνακας είναι ο εξής:

SNR(dB)	Ιδανικό Κανάλι	Μη-Ιδανικό Κανάλι
0	0.0478	0.1227
2	0.0333	0.1169
4	0.0198	0.1131
6	0.0094	0.1104
8	0.0031	0.1082
10	0.0006	0.1069
12	0.0001	0.1063
14	0	0.1057
16	0	0.1050
18	0	0.1050
20	0	0.1048
22	0	0.1050
24	0	0.1048
26	0	0.1046
28	0	0.1046
30	0	0.1045

Ενώ η γραφική απεικόνιση των αποτελεσμάτων για το 4-PAM, είναι:



-----Η διαδικασία αυτή υλοποιείται στο κομμάτι κώδικα με όνομα meros_C.m, και βρίσκεται στο παράρτημα με τους κώδικες στο τελευταίο κομμάτι της αναφοράς.-----

Δ' Μέρος: Τετραδικό QAM

Το αλφάβητο της πηγής για το 4-QAM μας δίνεται από την εκφώνηση. Η επιλογή αυτών των τιμών πάλι φαίνεται να μην ήταν τυχαία, καθώς θέλουμε να είναι δυνατή η σύγκριση με 4-PAM και 2-PAM. Επομένως πάλι θέλουμε μοναδιαία μέση ενέργεια σήματος.

Πράγματι, βάσει της θεωρίας:

$P_{s,4\text{-}QAM}=\sum p_i s_i^2=1$

Το σύστημα εκτελείται όμοια με τις προηγούμενες δύο διαμορφώσεις, αλλά με τις διαφορές που αναφέραμε στο Α' μέρος για μιγαδικούς αστερισμούς συμβόλων.

Μετατρέπουμε λοιπόν τα σύμβολα πηγής και εξόδων ως εξής:

$(-1-j)/\sqrt{(2)} \rightarrow 00$
$(-1+j)/\sqrt{(2)} \rightarrow 01$
$(1-j)/\sqrt{(2)} \rightarrow 10$
$(1+j)/\sqrt{(2)} \rightarrow 11$

Όμοια με το 4-PAM χρησιμοποιήσαμε ξεχωριστά διπλάσιου ύψους διανύσματα σε σχέση με της αρχικής πηγής, διότι κάθε σύμβολο κωδικοποιείται σε 2 bits. Για τον υπολογισμό του BER αθροίσαμε το σύνολο των XOR λογικών πράξεων κάθε στήλης κάθε πίνακα με το διάνυσμα της πηγής με δυαδικά ψηφία ως bits, και διαιρέσαμε με το μήκος της. Έτσι καταλήξαμε σε δύο διανύσματα που αντιπροσωπεύουν τα BER για κάθε τιμή SNR και τα δώσαμε ως έξοδο.

Έχουμε και πάλι συμπεριλάβει κώδικα για τη γραφική απεικόνιση των παραπάνω.

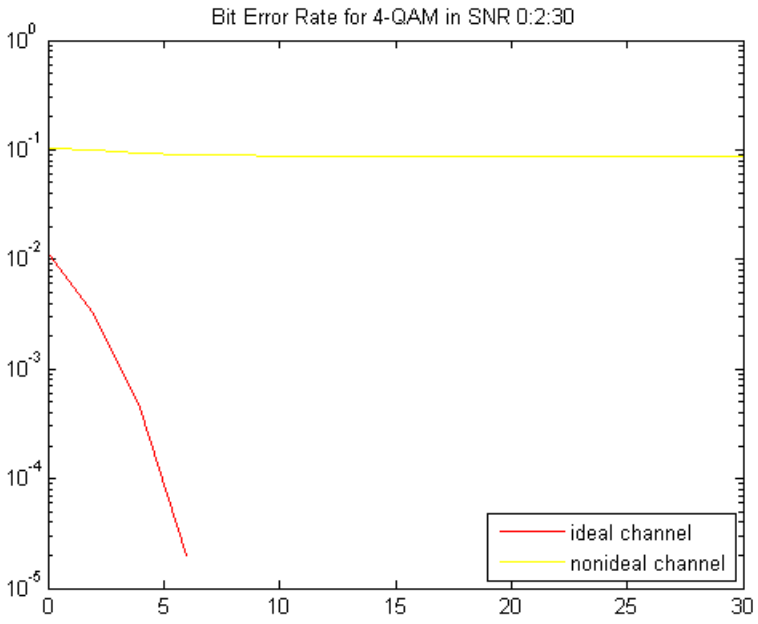
Επίσης όμοια με πριν, εκτελούμε την αποθήκευση των αποτελεσμάτων την παρακάτω εντολή:

```
[ QAM4_ideal_BER , QAM4_nonideal_BER ] = meros_D
```

Σε αυτή την περίπτωση, ο πίνακας είναι ο εξής:

SNR(dB)	Ιδανικό Κανάλι	Μη-Ιδανικό Κανάλι
0	0.0113	0.1038
2	0.0032	0.0986
4	0.0004	0.0936
6	0.0000	0.0908
8	0	0.0888
10	0	0.0882
12	0	0.0878
14	0	0.0881
16	0	0.0877
18	0	0.0881
20	0	0.0880
22	0	0.0883
24	0	0.0879
26	0	0.0882
28	0	0.0883
30	0	0.0880

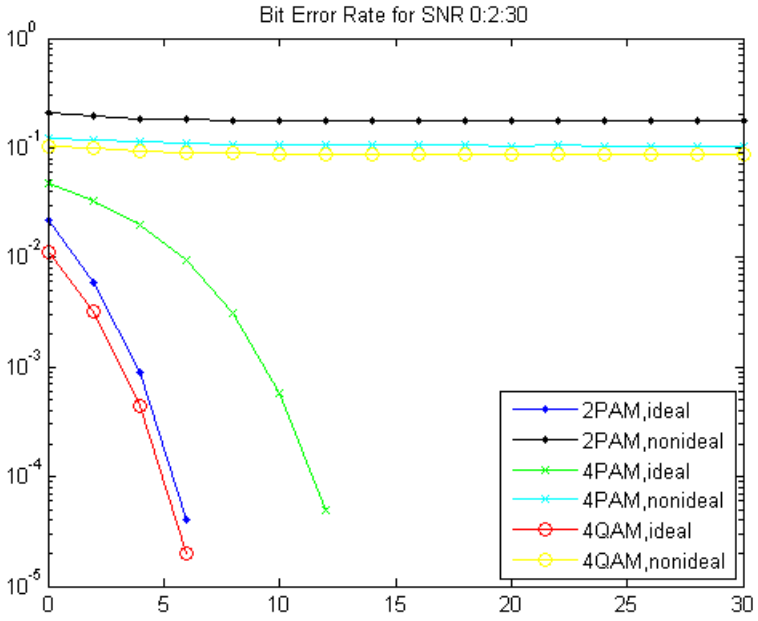
Η γραφική απεικόνιση των αποτελεσμάτων για το 4-QAM, είναι:



-----Η διαδικασία αυτή υλοποιείται στο κομμάτι κώδικα με όνομα meros_D.m, και βρίσκεται στο παράρτημα με τους κώδικες στο τελευταίο κομμάτι της αναφοράς.-----

Ε' Μέρος: Σύγκριση της επίδοσης των διαμορφώσεων

Πριν προχωρήσουμε σε κάποιες παρατηρήσεις σχετικά με τις διαμορφώσεις, παραθέτουμε μία γραφική απεικόνιση όλων μαζί.



Με τη βοήθεια της γραφικής παράστασης λοιπόν, μπορούμε να κάνουμε τις εξής παρατηρήσεις:

Παρατήρηση 1:
Όσο αυξάνεται το SNR|dB μειώνεται το σφάλμα. Αυτό είναι προφανές από τον ορισμό του SNR|dB που εκφράζει το ποσοστό σήματος προς θόρυβο. Επομένως όταν είναι μεγαλύτερο το σήμα μας, είναι και πιο καθαρό.

Παρατήρηση 2:
Για τα ιδανικά κανάλια, το BER μηδενίζεται σε κάποια τιμή του SNR|dB και μετέπειτα. Κι αυτό είναι αναμενόμενο καθώς από ένα σημείο και μετά, αναλόγως τη διαμόρφωση, το σήμα μεταδίδεται σωστά.

Παρατήρηση 3:
Τα μη-ιδανικά κανάλια έχουν σχεδόν σταθερό BER. Αυτό είναι προφανές επίσης, καθότι η συνέλιξη του μη-ιδανικού καναλιού με το σήμα προσθέτει μεγάλης τάξεως θόρυβο στο σήμα.

Παρατήρηση 4:
Για τα PAM ισχύει ότι όσο αυξάνεται ο αριθμός των bits, τόσο αυξάνεται και το BER. Αυτό είναι προφανές καθότι για δοσμένο εύρος ζώνης, έχουμε περισσότερα στιγμιότυπα σε μεγαλύτερου αλφαβήτου σήματα και ανάγκη περισσότερων bit απεικόνισης. Όμως οι ευκλείδιες αποστάσεις μεταξύ των σημείων μικραίνουν, επηρεάζονται οι μετρικές, κι έτσι συμβαίνουν συχνότερα σφάλματα στα σύμβολα.

Παρατήρηση 5:
Το 4-QAM είναι καλύτερο από άποψη σφαλμάτων. Η διαμόρφωση QAM είναι ορθογώνια και διαμορφώνει το σήμα και κατά πλάτος αλλά και κατά φάση, και επιτυγχάνει ταυτόχρονη διαβίβαση (k1+k2) δυαδικών στοιχείων συνολικά. Επομένως, μπορεί να επιτυγχάνει ίδιο bitrate με το 4-PAM, αλλά με σχεδόν την ίδια πιθανότητα σφάλματος με το 2-PAM.

Καταλήγουμε ότι καλύτερο για την υλοποίησή μας ήταν το 4-QAM καθότι κατάφερε μεγαλύτερη, και καλύτερης ποιότητας, μεταφορά bit για όλες τις περιπτώσεις.

-----Η διαδικασία αυτή υλοποιείται στο κομμάτι κώδικα με όνομα meros_E.m, και βρίσκεται στο παράρτημα με τους κώδικες στο τελευταίο κομμάτι της αναφοράς.-----

---

Παράρτημα με Κώδικες:

Στο σημείο αυτό παρουσιάζουμε τους κώδικες που υλοποιήσαμε για την εξομοίωση ενός τηλεπικοινωνιακού συστήματος βασικής ζώνης, ώστε να εξετάσουμε την επίδοσή του για διάφορα είδη διαμόρφωσης σε ιδανικό και μη-ιδανικό κανάλι.

-----meros_A.m-----

```
function [ out ] = meros_A( in, rate, ro_factor, T, ideal, SNR_db, s )
%meros_A: Modulation of input signal through system containing a filter
%for transmission, a channel with additive white gaussian noise, a filter
%in receiver, a sampler and a decision device.
%   ~ Inputs:
%   in:         input signal
%   rate:       the rate in which the signal is upsampled
%   ro_factor: the roll-off factor for raised cosine function
%   T:          transmission period
%   ideal:      'y' for ideal channel, 'n' for nonideal channel
%   SNR_db:     SNR in decibels (can be given as vector)
%   s:          the alphabet of the source

%~~~~~~~~~~~~~~~~~~~~~~ INPUT | UPSAMPLING ~~~~~~~~~~~~~~~~~~~~~~
%---upsample input according to rate:
signal_up = upsample(in,rate);
%remove last nonusable '0s'
signal_up = signal_up(1:length(signal_up)-rate+1);

%~~~~~~~~~~~~~~~~~~~~~~ TRANSMITTER | FILTER ~~~~~~~~~~~~~~~~~~~~
%---transmitter's filter (Ts = 1sec)
f = rcosfir(ro_factor , [-T/2 , T/2] , rate , 1 , 'sqrt');
%normalization of filter so that norm(f) = 1
f = f./norm(f);
%---signal is passed through the filter (convolution):
TF_sig = conv(f , signal_up);

%~~~~~~~~~~~~~~~~~~~~~~ CHANNEL | FILTER ~~~~~~~~~~~~~~~~~~~~~~~~~
%---check if channel is ideal or not;
%if it isn't, add the channel's upsampled response
if (ideal == 'n')
    %non-ideal channel's impulse response is given as:
    h = [0.01 , 0.04 , -0.05 , 0.06 , -0.22 , -0.5 , ...
         0.72 , 0.36 , 0 , 0.21 , 0.04 , 0.08 , 0.02];
    h_up = upsample(h,rate);
    %remove extra '0s'
    h_up = h_up(1:length(h_up)-rate+1);
    %convolution with signal
    C_sig = conv(TF_sig , h_up);
elseif (ideal == 'y')
    C_sig = TF_sig;
end
```

```
%~~~~~~~~~~~~~~~~~~~~ CHANNEL | NOISE ~~~~~~~~~~~~~~~~~~~~~~~~
%---additive AWGN noise:
%calculate average power of channeled signal:
absC_sig = abs(C_sig);
Ps = sum(absC_sig.^2) / length(C_sig);
%calculate average power of channel's noise based on given SNR|db:
Pn = Ps / (10^(SNR_db/10));
%calculate noise based on Pn which is the square of noise's deviation:
noiserand = randn(length(C_sig),1);
%if the constellation is complex, noise must be added in both imaginary and
%real parts:
if(isreal(C_sig) == 0)
    noiserand = (noiserand + 1j*randn(length(C_sig),1)) / sqrt(2);
end
noise = sqrt(Pn) * noiserand;
%---channel's output with added noise:
CAWGN_sig = C_sig + noise;

%~~~~~~~~~~~~~~~~~~~~~ RECEIVER | FILTER ~~~~~~~~~~~~~~~~~~~~~~
%---receiver's filter is the same as transmitter's:
RF_sig = conv(f , CAWGN_sig);

%~~~~~~~~~~~~~~~~~~~ RECEIVER | DOWNSAMPLING ~~~~~~~~~~~~~~~~~~~
%---calculate delay according to channel's type to be excluded from
%incoming signal:
if (ideal == 'n')
    %displacement:
    delay = 2*floor(length(f)/2) + floor(length(h_up)/2);
elseif (ideal == 'y')
    delay = 2*floor(length(f)/2);
end
%downsample signal:
S_sig = RF_sig(delay+1 : rate : length(RF_sig) - delay);

%~~~~~~~~~~~~~~~~~~~~ RECEIVER | DECISION ~~~~~~~~~~~~~~~~~~~~~
out = zeros(length(S_sig),1);
D = zeros(length(s),1);
%for each element of the signal vector, decide by minimum square distance
%to which value of the alphabet it is closer:
for i = 1 : 1 : length(S_sig)
    for j = 1 : 1 : length(s)
        D(j,1) = (S_sig(i,1) - s(j,1))^2;
    end
    for k = 1 : 1 : length(D)
        if(D(k,1) == min(D))
            out(i,1) = s(k,1);
        end
    end
end

end
```

-----meros_B.m-----

```
function [ ideal_BER , nonideal_BER ] = meros_B(  )
%meros_B :: SIMULATION OF 2-PAM

%script for excercise 2.B:
%~~~~~~~~~~~~~~~~~~~~~~~~~ SIMULATION OF 2-PAM ~~~~~~~~~~~~~~~~~~~~~~~~~~
%Given parameters:
a = 0.3;           %roll-off factor
T = 6;             %sampling period width centralized around 0 (T=6Ts=6)
rate = 4;          %over-sampling rate
SNR = (0:2:30);    %SNR values for which calculations will be made
numval = 10^5;     %number of input values
s = [-1 ; 1];      %alphabet of 2-PAM (source symbols)

%1-D random input source of equal probable symbols (-1 , +1) for 2-PAM:
src = randsrc(numval,1,s');

%Initialization of system ouputs; each column contains a vector of the
%output values for different given input SNRs.
%Each matrix is simulation of ideal and non-ideal channels:
ideal_output = zeros(length(src),length(SNR));
nonideal_output = zeros(length(src),length(SNR));
ideal_BER = zeros(length(SNR),1);
nonideal_BER = zeros(length(SNR),1);

%Source signal transmitted through the system for different values of SNR:
for i = 1 : 1 : length(SNR)
    ideal_output(:,i) = modulation(src, rate, a, T, 'y', SNR(1,i), s);
    nonideal_output(:,i) = modulation(src, rate, a, T, 'n', SNR(1,i), s);
end

%~~~~~~~~~~~~~~~~~~~~~~~~~ CONVERSION TO BINARY ~~~~~~~~~~~~~~~~~~~~~~~~~~
%for each element in source input and channel's outputs convert symbols to
%binary digits as follows:
% -1 --> 0
% +1 --> 1
for i = 1 : 1 : length(src)
    %source's conversion to 1s and 0s:
    if(src(i,1) == -1)
        src(i,1) = 0;
    elseif(src(i,1) ~= 1)
        error('ERROR: value out of permitted for source!');
    end
    %channels' concersions to 1s and 0s:
    for j = 1 : 1 : length(SNR)
        if(ideal_output(i,j) == -1)
            ideal_output(i,j) = 0;
        elseif(ideal_output(i,j) ~= 1)
            error('ERROR: value out of permitted for ideal output!');
        end
        if(nonideal_output(i,j) == -1)
            nonideal_output(i,j) = 0;
        elseif(nonideal_output(i,j) ~= 1)
            error('ERROR: value out of permitted for nonideal output!');
```

```
        end
    end
end

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ BIT ERROR RATE ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%Calculation of BER:
for i = 1 : 1 : length(SNR)
    ideal_BER(i,1) = sum(xor(ideal_output(:,i),src)) / length(src);
    nonideal_BER(i,1) = sum(xor(nonideal_output(:,i),src)) / length(src);
end
%Graphical representation of BER:
figure;
semilogy(SNR , ideal_BER , 'b');
hold on;
semilogy(SNR , nonideal_BER , 'k');
title('Bit Error Rate for 2-PAM in SNR 0:2:30');
legend('ideal channel' , 'nonideal channel' , 'Location' , 'SouthEast');
hold off;

end
```

-----meros_C.m-----

```
function [ ideal_BER , nonideal_BER ] = meros_C(  )
%meros_C :: SIMULATION OF 4-PAM

%script for excercise 2.C:
%~~~~~~~~~~~~~~~~~~~~~~~~~~ SIMULATION OF 4-PAM ~~~~~~~~~~~~~~~~~~~~~~~~~~~
%Given parameters:
a = 0.3;           %roll-off factor
T = 6;             %sampling period width centralized around 0 (T = 6Ts)
rate = 4;          %over-sampling rate
SNR = (0:2:30);    %SNR values for which calculations will be made
numval = 10^5;     %number of input values
s = [-3/sqrt(5); -1/sqrt(5); 1/sqrt(5); 3/sqrt(5)]; %alphabet of 4-PAM

%1-D random input source of equal probable symbols (-3/sqrt(5), -1/sqrt(5),
% 1/sqrt(5), 3/sqrt(5)) for 4-PAM:
src = randsrc(numval,1,s');

%Initialization of system ouputs; each column contains a vector of the
%output values for different given input SNRs.
%Each matrix is simulation of ideal and non-ideal channels:
ideal_output = zeros(length(src),length(SNR));
nonideal_output = zeros(length(src),length(SNR));
ideal_BER = zeros(length(SNR),1);
nonideal_BER = zeros(length(SNR),1);

%Source signal transmitted through the system for different values of SNR:
for i = 1 : 1 : length(SNR)
    ideal_output(:,i) = modulation(src, rate, a, T, 'y', SNR(1,i), s);
    nonideal_output(:,i) = modulation(src, rate, a, T, 'n', SNR(1,i), s);
end
```

```
%~~~~~~~~~~~~~~~~~~~~~~~~~ CONVERSION TO BINARY ~~~~~~~~~~~~~~~~~~~~~~~~~
%for each element in source input and channel's outputs convert symbols to
%binary digits as follows:
% -3/sqrt(5) --> 00
% -1/sqrt(5) --> 01
%  1/sqrt(5) --> 10
%  3/sqrt(5) --> 11
source = zeros(length(src)*2,1);             %extra to store source's bits
ideal = zeros(length(src)*2,length(SNR));    %extra for ideal chan. bits
nonideal = zeros(length(src)*2,length(SNR)); %extra for nonideal chan. bits
for i = 1 : 1 : length(src)
    %source's conversion to 1s and 0s:
    if(src(i,1) == -3/sqrt(5))
        source(i,1) = 0;
        source(i+1,1) = 0;
    elseif(src(i,1) == -1/sqrt(5))
        source(i,1) = 0;
        source(i+1,1) = 1;
    elseif(src(i,1) == 1/sqrt(5))
        source(i,1) = 1;
        source(i+1,1) = 0;
    elseif(src(i,1) == 3/sqrt(5))
        source(i,1) = 1;
        source(i+1,1) = 1;
    else
        error('ERROR: value out of permitted for source!');
    end
    %channels' concersions to 1s and 0s:
    for j = 1 : 1 : length(SNR)
        if(ideal_output(i,j) == -3/sqrt(5))
            ideal(i,j) = 0;
            ideal(i+1,j) = 0;
        elseif(ideal_output(i,j) == -1/sqrt(5))
            ideal(i,j) = 0;
            ideal(i+1,j) = 1;
        elseif(ideal_output(i,j) == 1/sqrt(5))
            ideal(i,j) = 1;
            ideal(i+1,j) = 0;
        elseif(ideal_output(i,j) == 3/sqrt(5))
            ideal(i,j) = 1;
            ideal(i+1,j) = 1;
        else
            error('ERROR: value out of permitted for ideal output!');
        end

        if(nonideal_output(i,j) == -3/sqrt(5))
            nonideal(i,j) = 0;
            nonideal(i+1,j) = 0;
        elseif(nonideal_output(i,j) == -1/sqrt(5))
            nonideal(i,j) = 0;
            nonideal(i+1,j) = 1;
        elseif(nonideal_output(i,j) == 1/sqrt(5))
            nonideal(i,j) = 1;
            nonideal(i+1,j) = 0;
        elseif(nonideal_output(i,j) == 3/sqrt(5))
```

```
            nonideal(i,j) = 1;
            nonideal(i+1,j) = 1;
        else
            error('ERROR: value out of permitted for nonideal output!');
        end
    end
end

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ BIT ERROR RATE ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%Calculation of BER (Bit Error Rate):
for i = 1 : 1 : length(SNR)
    ideal_BER(i,1) = sum(xor(ideal(:,i),source)) / length(source);
    nonideal_BER(i,1) = sum(xor(nonideal(:,i),source)) / length(source);
end
%Graphical representation of BER:
figure;
semilogy(SNR , ideal_BER , 'g');
hold on;
semilogy(SNR , nonideal_BER , 'c');
title('Bit Error Rate for 4-PAM in SNR 0:2:30');
legend('ideal channel' , 'nonideal channel' , 'Location' , 'SouthEast');
hold off;

end
```

-----meros_D.m-----

```
function [ ideal_BER , nonideal_BER ] = meros_D(  )
%meros_D :: SIMULATION OF 4-QAM

%script for excercise 2.D:
%~~~~~~~~~~~~~~~~~~~~~~~~~~ SIMULATION OF 4-QAM ~~~~~~~~~~~~~~~~~~~~~~~~~~~
%Given parameters:
a = 0.3;           %roll-off factor
T = 6;             %sampling period width centralized around 0 (T = 6Ts)
rate = 4;          %over-sampling rate
SNR = (0:2:30);    %SNR values for which calculations will be made
numval = 10^5;     %number of input values
%alphabet of 4-QAM:
s = [(-1-1j)/sqrt(2); (-1+1j)/sqrt(2); (1-1j)/sqrt(2); (1+1j)/sqrt(2)];

%1-D random input source of equal probable symbols ((-1-1j)/sqrt(2);
% (-1+1j)/sqrt(2); (1-1j)/sqrt(2); (1+1j)/sqrt(2)) for 4-QAM:
src = randsrc(numval,1,s');

%Initialization of system ouputs; each column contains a vector of the
%output values for different given input SNRs.
%Each matrix is simulation of ideal and non-ideal channels:
ideal_output = zeros(length(src),length(SNR));
nonideal_output = zeros(length(src),length(SNR));
ideal_BER = zeros(length(SNR),1);
nonideal_BER = zeros(length(SNR),1);

%Source signal transmitted through the system for different values of SNR:
```

```
for i = 1 : 1 : length(SNR)
    ideal_output(:,i) = modulation(src, rate, a, T, 'y', SNR(1,i), s);
    nonideal_output(:,i) = modulation(src, rate, a, T, 'n', SNR(1,i), s);
end

%~~~~~~~~~~~~~~~~~~~~~~~~~ CONVERSION TO BINARY ~~~~~~~~~~~~~~~~~~~~~~~~~
%for each element in source input and channel's outputs convert symbols to
%binary digits as follows:
% (-1-j)/sqrt(2) --> 00
% (-1+j)/sqrt(2) --> 01
%  (1-j)/sqrt(2) --> 10
%  (1+j)/sqrt(2) --> 11
source = zeros(length(src)*2,1);              %extra to store source's bits
ideal = zeros(length(src)*2,length(SNR));     %extra for ideal chan. bits
nonideal = zeros(length(src)*2,length(SNR)); %extra for nonideal chan. bits
for i = 1 : 1 : length(src)
    %source's conversion to 1s and 0s:
    if(src(i,1) == (-1-1j)/sqrt(2))
        source(i,1) = 0;
        source(i+1,1) = 0;
    elseif(src(i,1) == (-1+1j)/sqrt(2))
        source(i,1) = 0;
        source(i+1,1) = 1;
    elseif(src(i,1) == (1-1j)/sqrt(2))
        source(i,1) = 1;
        source(i+1,1) = 0;
    elseif(src(i,1) == (1+1j)/sqrt(2))
        source(i,1) = 1;
        source(i+1,1) = 1;
    else
        error('ERROR: value out of permitted for source!');
    end
    %channels' concersions to 1s and 0s:
    for j = 1 : 1 : length(SNR)
        if(ideal_output(i,j) == (-1-1j)/sqrt(2))
            ideal(i,j) = 0;
            ideal(i+1,j) = 0;
        elseif(ideal_output(i,j) == (-1+1j)/sqrt(2))
            ideal(i,j) = 0;
            ideal(i+1,j) = 1;
        elseif(ideal_output(i,j) == (1-1j)/sqrt(2))
            ideal(i,j) = 1;
            ideal(i+1,j) = 0;
        elseif(ideal_output(i,j) == (1+1j)/sqrt(2))
            ideal(i,j) = 1;
            ideal(i+1,j) = 1;
        else
            error('ERROR: value out of permitted for ideal output!');
        end
        if(nonideal_output(i,j) == (-1-1j)/sqrt(2))
            nonideal(i,j) = 0;
            nonideal(i+1,j) = 0;
        elseif(nonideal_output(i,j) == (-1+1j)/sqrt(2))
            nonideal(i,j) = 0;
            nonideal(i+1,j) = 1;
        elseif(nonideal_output(i,j) == (1-1j)/sqrt(2))
            nonideal(i,j) = 1;
```

```
            nonideal(i+1,j) = 0;
        elseif(nonideal_output(i,j) == (1+1j)/sqrt(2))
            nonideal(i,j) = 1;
            nonideal(i+1,j) = 1;
        else
            error('ERROR: value out of permitted for nonideal output!');
        end
    end
end
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~ BIT ERROR RATE ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%Calculation of BER (Bit Error Rate):
for i = 1 : 1 : length(SNR)
    ideal_BER(i,1) = sum(xor(ideal(:,i),source)) / length(source);
    nonideal_BER(i,1) = sum(xor(nonideal(:,i),source)) / length(source);
end
%Graphical representation of BER:
figure;
semilogy(SNR , ideal_BER , 'r');
hold on;
semilogy(SNR , nonideal_BER , 'y');
title('Bit Error Rate for 4-QAM in SNR 0:2:30');
legend('ideal channel' , 'nonideal channel' , 'Location' , 'SouthEast');
hold off;
end
```

-----meros_E.m-----

```
function [  ] = meros_E(  )
%meros_E :: COMPARISON OF PERFOMANCES
%script for excercise 2.E:
%~~~~~~~~~~~~~~~~~~~~~~~ COMPARISON OF PERFOMANCES ~~~~~~~~~~~~~~~~~~~~~~~~
SNR = (0:2:30);
[ PAM2_ideal_BER , PAM2_nonideal_BER ] = meros_B
[ PAM4_ideal_BER , PAM4_nonideal_BER ] = meros_C
[ QAM4_ideal_BER , QAM4_nonideal_BER ] = meros_D
figure;
semilogy(SNR, PAM2_ideal_BER , 'b.-');
hold on;
semilogy(SNR, PAM2_nonideal_BER, 'k.-');
hold on;
semilogy(SNR, PAM4_ideal_BER , 'gx-');
hold on;
semilogy(SNR, PAM4_nonideal_BER, 'cx-');
hold on;
semilogy(SNR, QAM4_ideal_BER , 'ro-');
hold on;
semilogy(SNR, QAM4_nonideal_BER, 'yo-');
title('Bit Error Rate for SNR 0:2:30');
legend('2PAM,ideal' , '2PAM,nonideal' , '4PAM,ideal' , '4PAM,nonideal', ...
       '4QAM,ideal' , '4QAM,nonideal' , 'Location', 'SouthEast');
hold off;

end
```