



Πανεπιστήμιο Πειραιώς

Τμήμα Ψηφιακών Συστημάτων

Π.Μ.Σ. " Διδακτική της Τεχνολογίας & Ψηφιακά Συστήματα "

**ΑΞΙΟΛΟΓΗΣΗ ΕΠΙΔΟΣΗΣ ΔΕΚΤΩΝ ΓΙΑ ΑΣΥΡΜΑΤΑ
ΣΥΣΤΗΜΑΤΑ ΠΟΛΛΑΠΛΩΝ ΕΙΣΟΔΩΝ ΠΟΛΛΑΠΛΩΝ
ΕΞΟΔΩΝ (ΜΙΜΟ)**

ΚΟΛΛΙΑΣ ΣΤΥΛΙΑΝΟΣ

ΕΠΙΒΛΕΠΩΝ: ΑΝΑΠΛΗΡΩΤΗΣ ΚΑΘΗΓΗΤΗΣ ΑΘΑΝΑΣΙΟΣ ΚΑΝΑΤΑΣ

Ιούνιος 2012

Περίληψη

Η παρούσα διπλωματική εργασία πραγματοποιεί μια αξιολόγηση της απόδοσης έξι δεκτών για συστήματα πολλαπλών εισόδων – πολλαπλών εξόδων (MIMO). Αρχικά παρουσιάζεται η εξέλιξη των ασύρματων επικοινωνιών που ώθησαν την επιστημονική κοινότητα στην ανακάλυψη νέων μεθόδων που θα επιτρέπουν την αύξηση της αξιοπιστίας και της χωρητικότητας των ασύρματων ζεύξεων. Στη συνέχεια παρουσιάζονται οι βασικές θεωρητικές αρχές πάνω στις οποίες βασίζεται η τεχνική MIMO. Ακολουθεί σύντομη περιγραφή των αποτελεσμάτων της χρησιμοποίησης της τεχνικής αυτής στη χωρητικότητα των ασύρματων ζεύξεων.

Στο κυριότερο μέρος της εργασίας παρουσιάζεται η λειτουργία των έξι διαφορετικών δεκτών. Η απόδοσή τους αξιολογείται μέσω προσομοιώσεων της λειτουργίας τους που υλοποιήθηκε με τη βοήθεια του λογισμικού Matlab. Μελετάται ο ρυθμός σφαλμάτων που εμφανίζεται για ασύρματο κανάλι που υπόκειται σε flat fading Rayleigh διαλείψεις με τη χρησιμοποίηση MQAM διαμόρφωσης για εύρος σηματοθορβικού λόγου από 0 έως 20 dB . Τέλος πραγματοποιείται μια σύγκριση της απόδοσης των διαφορετικών δεκτών για ταυτόχρονη αποκωδικοποίηση του ίδιου σήματος που διέρχεται μέσω του MIMO καναλιού.

Λέξεις Κλειδιά

MIMO, Χωρητικότητα, Δέκτης, Χωρική Πολυπλεξία, Zero Forcing, Minimum Mean Square Error, Ordered Successive Interference Cancellation, Maximum Likelihood, Sphere Decoder.

Abstract

The purpose of this thesis is to evaluate the performance of six receivers for multiple input – multiple output (MIMO) spatial multiplexing systems. Initially, we present the evolution of wireless communications which has prompted the scientific community in finding new methods for increasing reliability and capacity of wireless links. Following are the basic theoretical principles upon which the MIMO technique is based. Afterwards we present a brief description of the impact that the use of this technique has on the capacity of wireless links.

In the main part of this thesis we present the operation of the six different spatial multiplexing receivers. Their performance is evaluated with the use of simulations of their operation that are implemented in Matlab. We study the bit error rate that occurs in the transmission of data through a flat fading Rayleigh wireless channel with the use of MQAM modulation for a signal to interference ratio of 0 to 20 dB. Finally we present a performance comparison between the different receivers for the simultaneous decoding of the same signal that passes through the MIMO wireless channel.

Key Words

MIMO, Capacity, Receiver, Spatial Multiplexing, Zero Forcing, Minimum Mean Square Error, Ordered Successive Interference Cancellation, Maximum Likelihood, Sphere Decoder.

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στα πλαίσια του μεταπτυχιακού προγράμματος σπουδών Ψηφιακές Επικοινωνίες και Δίκτυα του Τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς.

Στο σημείο αυτό θα ήθελα να ευχαριστήσω τον επιβλέποντα αναπληρωτή καθηγητή του Πανεπιστημίου Πειραιώς κύριο Κανάτα Αθανάσιο για την καθοδήγηση που μου παρείχε κατά την εκπόνηση της διπλωματικής αλλά και καθ' όλη τη φοίτησή μου στο μεταπτυχιακό πρόγραμμα σπουδών. Επίσης θα ήθελα να ευχαριστήσω την οικογένειά μου χωρίς την υποστήριξη της οποίας δε θα ήταν δυνατόν να φτάσω σε αυτό το σημείο.

Περιεχόμενα

Περίληψη	i
Abstract	ii
Ευχαριστίες	iii
Περιεχόμενα	iv
Κεφάλαιο 1	1
Εισαγωγή.....	1
1.1 Η εξέλιξη των τηλεπικοινωνιών και της ασύρματης μετάδοσης δεδομένων.	1
1.2 Εισαγωγή στα Συστήματα MIMO.	2
1.2.1 Κέρδος διάταξης (Array Gain).....	2
1.2.2 Κέρδος διαφορισμού (Diversity Gain)	3
1.2.3 Κέρδος Χωρικής Πολυπλεξίας (Spatial Multiplexing Gain).....	6
1.3 Αντικείμενο Εργασίας	7
Κεφάλαιο 2	8
Θεωρία MIMO Συστημάτων	8
2.1 Χωρική Πολυπλεξία MIMO Καναλιού.....	8
2.2 Μετάδοση σήματος και βαθμοί ελευθερίας του συστήματος	11
2.2.1 SIMO (Single Input Multiple Output)	12
2.2.2 MISO (Multiple Input Single Output)	14
2.2.3 MIMO (Multiple Input Multiple Output)	15
2.3 Channel Matrix Condition	19
Κεφάλαιο 3	24
Χωρητικότητα Συστημάτων MIMO.....	24
3.1 Χωρητικότητα με γνωστό πίνακα καναλιού (Full CSI)	25
3.2 Χωρητικότητα όταν ο πομπός δεν έχει γνώση του καναλιού (No CSI at the transmitter).....	26
3.3 Εργοδική Χωρητικότητα.....	27
3.4 Χωρική Συσχέτιση	32
Κεφάλαιο 4	34

Δέκτες Χωρικής Πολυπλεξίας	34
4.1 Μοντέλο MIMO μετάδοσης.....	35
4.2 Γραμμικοί Δέκτες (Linear Receivers).....	40
4.2.1 Zero Forcing Receiver.....	40
4.2.2 MMSE Receiver.....	44
4.3 Successive Interference Cancellation Receivers	48
4.4 Δέκτης Μέγιστης Πιθανοφάνειας.....	58
4.5 Sphere Detection	61
Κεφάλαιο 5	74
Συμπεράσματα.....	74
Αναφορές.....	76
Παράρτημα	79
Κώδικες Matlab.....	79
Χωρητικότητα MIMO καναλιού	79
MIMO_Channel_Ergodig_Capacity_vs_SNR.m.....	79
Correlated_vs_Uncorrelated_MIMO_Channel_Capacity.m.....	79
OpenLoop_vs_ClosedLoop_Channel_Capacity.m	80
Προσομοίωση Λειτουργίας Δεκτών.....	82
Zero Forcing Receiver.....	82
MMSE Receiver	86
OSIC Zero Forcing Receiver	93
OSIC MMSE Receiver.....	101
ML Receiver.....	112
Sphere Decoder	120

Κεφάλαιο 1

Εισαγωγή

1.1 Η εξέλιξη των τηλεπικοινωνιών και της ασύρματης μετάδοσης δεδομένων.

Η μελέτη και η έρευνα πάνω στο θέμα της ασύρματης μετάδοσης πληροφοριών απασχολεί την επιστημονική κοινότητα ήδη από το 1896 όταν ο G. Marconi παρουσίασε αρχικά τον τηλεγράφο. Το 1901 υπήρξε επιτυχημένη μετάδοση μηνυμάτων που διέσχισαν μια απόσταση 1800 μιλίων πάνω από τον Ατλαντικό Ωκεανό. Στον αιώνα που ακολούθησε η πρόοδος της τεχνολογίας της ασύρματης μετάδοσης οδήγησε στο ραδιόφωνο, την τηλεόραση, τη δορυφορική επικοινωνία και τις κινητές τηλεπικοινωνίες.

Τη μεγαλύτερη επανάσταση έφεραν αποδεδειγμένα τα κινητά τηλέφωνα. Η μεγάλη έκρηξη παρουσιάστηκε τη δεκαετία του '90, όπου με την ανάπτυξη των δικτύων 2^{ης} γενιάς (2G) GSM και IS-95 το κινητό τηλέφωνο μπήκε δυναμικά στην καθημερινότητα των ανθρώπων. Η διείσδυση της κινητής τηλεφωνίας σήμερα έχει ξεπεράσει κάθε προσδοκία. Σύμφωνα με έρευνες σε σύνολο περίπου 6 δισεκατομμυρίων ανθρώπων του πληθυσμού της γης υπάρχουν πάνω από 5 δισεκατομμύρια συνδέσεις κινητής τηλεφωνίας. [2] Σε κάποιες χώρες το ποσοστό διείσδυσης ξεπερνάει το 130%.

Η μεγάλη αυτή διείσδυση δεν ήρθε εξαιτίας της απλής υπηρεσίας μετάδοσης φωνής. Η παράλληλη εξέλιξη της επιστήμης της πληροφορικής έπαιξε σπουδαίο ρόλο στην επιτυχία της κινητής τηλεφωνίας. Η συνεχής εξέλιξη σε ταχύτερους και μικρότερους υπολογιστές έχει οδηγήσει σήμερα τα κινητά τηλέφωνα να είναι υπολογιστές παλάμης εφάμιλλων δυνατοτήτων με τους φορητούς ηλεκτρονικούς υπολογιστές. Οι νέες συσκευές έχουν καινούριες δυνατότητες παροχής υπηρεσιών που συγχρόνως όμως είναι και πιο απαιτητικές. Προϋπόθεση για την παροχή τέτοιων

υπηρεσιών είναι η υποστήριξη υψηλότερων ρυθμών αξιόπιστης μετάδοσης δεδομένων. Προς αυτό το στόχο κινούνται όλες οι εξελίξεις στον τομέα της ασύρματης μετάδοσης δεδομένων. Η αύξηση του ρυθμού διαμόρφωσης που στο 64QAM κωδικοποιεί 8 bits/symbol αλλά και η διασπορά του φάσματος είναι τεχνικές που χρησιμοποιούνται για αυτό το λόγο. Προς αυτήν ακριβώς την κατεύθυνση κινείται και η εξέλιξη των συστημάτων MIMO (Multiple Input Multiple Output).

1.2 Εισαγωγή στα Συστήματα MIMO.

Το όνομα του συστήματος περιγράφει και το βασικό στοιχείο που το διαφοροποιεί από τις μέχρι τώρα τεχνικές μετάδοσης δεδομένων. Το στοιχείο που περιγράφεται, είναι η συμμετοχή στη μετάδοση πολλαπλών στοιχείων εισόδου, δηλαδή πολλαπλών κεραιών εκπομπής, και πολλαπλών στοιχείων εξόδου, δηλαδή πολλαπλών κεραιών λήψης. Η τεχνική αυτή εκμεταλλεύεται εκτός από τις διαστάσεις του χρόνου και της συχνότητας όπως τα μέχρι τώρα συνηθισμένα συστήματα, τη διάσταση του χώρου. Τα ηλεκτρομαγνητικά κύματα κατά τη διάδοσή τους ακολουθούν διαφορετικές διαδρομές. Για να εκμεταλλευτεί το γεγονός αυτό η τεχνική MIMO χρησιμοποιεί μεθόδους που αποσυνθέτουν τις διαδρομές από διαφορετικά στοιχεία κεραιών ώστε να προκύψουν περισσότερα του ενός ανεξάρτητα κανάλια μεταφοράς. Μέσω της κατάλληλης επεξεργασίας σήματος τα MIMO συστήματα επιτυγχάνουν είτε αύξηση του ρυθμού μετάδοσης είτε βελτίωση της αξιοπιστίας της μετάδοσης, ανάλογα με το ποια τεχνική χρησιμοποιείται για τον εκάστοτε από τους δυο στόχους. Η χρήση των πολλαπλών κεραιών εισόδου και εξόδου του συστήματος έχουν ορισμένα άμεσα οφέλη τα οποία παρατίθενται στη συνέχεια.

1.2.1 Κέρδος διάταξης (Array Gain)

Με τον όρο κέρδος διάταξης περιγράφεται η μέση αύξηση του σηματοθορυβικού λόγου που παρατηρείται στο δέκτη όταν χρησιμοποιούνται πολλαπλές κεραίες είτε στην πλευρά του πομπού είτε στην πλευρά του δέκτη. Είναι προφανές ότι τα αντίγραφα ενός σήματος που διαδίδεται μέσω διαφορετικών διαδρομών είτε γιατί προέρχονται από διαφορετικές κεραίες είτε γιατί καταφθάνουν

σε διαφορετικές κεραιές έχουν το καθένα τα δικά του διαφορετικά χαρακτηριστικά όσο αναφορά στο πλάτος και τη φάση με την οποία καταφθάνουν στο δέκτη. Το κέρδος διάταξης δηλαδή η αύξηση του μέσου σηματοθορυβικού λόγου προκύπτει μέσω του συμφασικού συνδυασμού (coherent detection) των διαφορετικών αντίγραφων του σήματος. Το κέρδος διάταξης αποτυπώνεται ως G_a

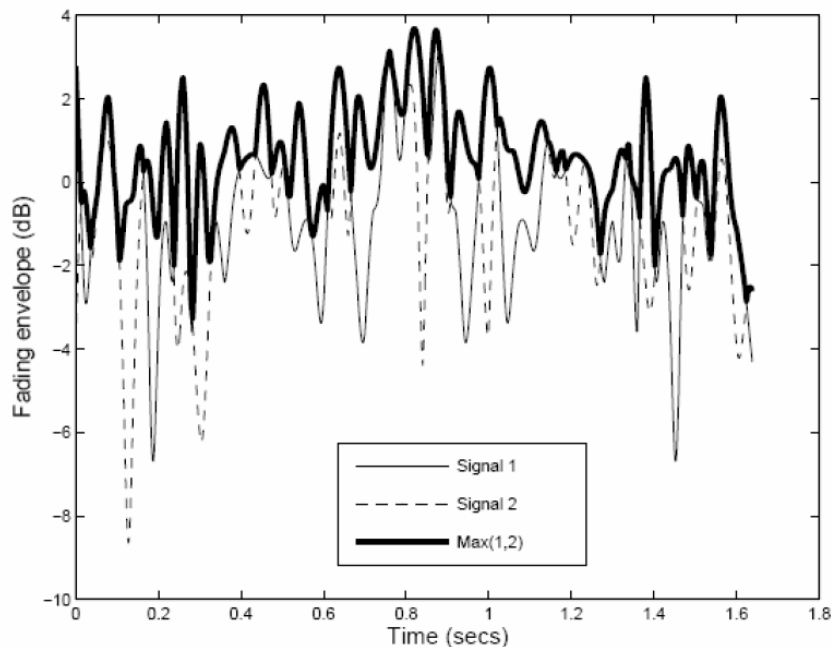
$$G_a = \frac{\rho_a}{\rho}$$

Όπου ρ είναι ο μέσος σηματοθορυβικός λόγος του κάθε ξεχωριστού αντίγραφου του σήματος που λαμβάνει η κεραιά του δέκτη και ρ_a ο μέσος ενεργός σηματοθορυβικός λόγος που προκύπτει μετά το συμφασικό συνδυασμό όλων των σημάτων. Ένας τέτοιος τύπος δέκτη είναι ο Rake receiver και στην περίπτωση κατά την οποία η χρήση πολλαπλών κεραιών αφορά μόνο την πλευρά λήψης, το κέρδος συστοιχίας είναι ανάλογο του αριθμού των κεραιών που χρησιμοποιούνται. Προϋπόθεση βεβαίως για το συμφασικό συνδυασμό είναι ο δέκτης να έχει πλήρη γνώση του διαύλου μεταφοράς του σήματος.

1.2.2 Κέρδος διαφορισμού (Diversity Gain)

Ο βασικός σκοπός του διαφορισμού είναι η αντιμετώπιση των διαλείψεων στις οποίες υπόκειται ένα σήμα κατά τη μετάδοσή του μέσα από έναν ασύρματο δίαυλο. Στα έως τώρα συστήματα επικοινωνιών έχει χρησιμοποιηθεί διαφορισμός στο πεδίο του χρόνου (time diversity) και το πεδίο της συχνότητας (frequency diversity). Τα MIMO συστήματα εισάγουν τη δυνατότητα διαφορισμού στο πεδίο του χώρου (spatial/antenna diversity). Ο διαφορισμός χώρου βελτιώνει την συμπεριφορά του συστήματος έναντι των βαθιών διαλείψεων που παρουσιάζονται κατά τη διάρκεια μετάδοσης του σήματος. Οι πολυδιαδρομικές συνιστώσες διάδοσης μπορούν υπό κατάλληλες προϋποθέσεις να δημιουργήσουν στατιστικά ανεξάρτητα υπο-κανάλια μεταφοράς με στατιστικά ανεξάρτητες συμπεριφορές. Τα διαφορετικά συνεπώς αντίγραφα ενός σήματος μπορούν να ανασυντεθούν και να βελτιώσουν την εικόνα του σήματος. Ο διαφορισμός χώρου διασπάται σε δυο υποκατηγορίες, σε διαφορισμό λήψης (receive diversity) και εκπομπής (transmit diversity).

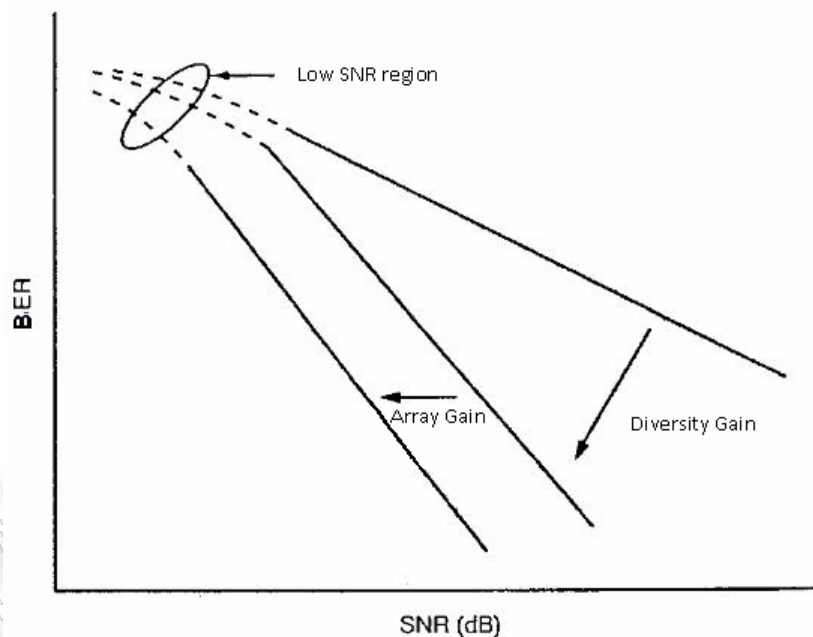
Στην πρώτη περίπτωση τα αντίγραφα ενός σήματος που προέρχονται από διαφορετικές, στατιστικώς ανεξάρτητες διαδρομές διάδοσης συνδυάζονται στην πλευρά του δέκτη. Εφόσον υπάρχει πλήρης γνώση του διαύλου διάδοσης και πραγματοποιείται ο κατάλληλος συνδυασμός έχουμε σαν αποτέλεσμα η περιβάλλουσα του σήματος που λαμβάνεται να παρουσιάζει μικρότερες διακυμάνσεις. Στην εικόνα 1.1 παρουσιάζεται το αποτέλεσμα που προκύπτει στην περιβάλλουσα ενός σήματος όταν εφαρμόζεται κατάλληλος συνδυασμός δύο ανεξάρτητων αντίγραφών του.



1.1 Βελτίωση της διακύμανσης περιβάλλουσας σήματος

Ο χωρικός διαφορισμός εκπομπής περιλαμβάνει προ-επεξεργασία των δεδομένων που μεταδίδονται. Η πλευρά εκπομπής καλείται να δημιουργήσει διαφορετικά σήματα τα οποία μέσω της υπέρθεσής τους στην πλευρά λήψης θα δημιουργήσουν ένα σήμα του οποίου η περιβάλλουσα θα παρουσιάζει μικρότερες διακυμάνσεις. Στο πεδίο αυτό έχουν διατυπωθεί μοντέλα μετάδοσης με χρήση χώρο – χρονικής κωδικοποίησης (Space – Time Coding). Υπάρχουν συστήματα μετάδοσης από την τεχνική Alamuti έως και τον Space – Time Trellis Encoder τα οποία

πραγματοποιούν τέτοιο χώρο – χρόνικη – κωδικοποίηση που αντιμετωπίζουν με επιτυχία το πρόβλημα των βαθιών διαλείψεων, εξομαλύνουν την περιβάλλουσα του σήματος που λαμβάνεται και επιτυγχάνουν μεγάλη βελτίωση στην αξιοπιστία της μετάδοσης. Η βελτίωση της αξιοπιστίας αξιολογείται μέσω του ρυθμού σφαλμάτων που προκύπτουν σε μια μετάδοση δεδομένων και το κέρδος διαφορισμού αποτυπώνεται σαν μια αλλαγή στην κλίση της καμπύλης που περιγράφει τη σχέση μεταξύ σηματοθορυβικού λόγου και ρυθμού σφαλμάτων. Στην εικόνα 1.2 παρουσιάζεται αυτό το κέρδος μαζί με το κέρδος διάταξης που παρουσιάστηκε στην προηγούμενη ενότητα και αποτυπώνεται σαν μετατόπιση της καμπύλης προς χαμηλότερες τιμές σηματοθορυβικού λόγου.



1.2 Κέρδος διάταξης (Array) και κέρδος διαφορισμού (Diversity)

1.2.3 Κέρδος Χωρικής Πολυπλεξίας (Spatial Multiplexing Gain)

Τα οφέλη που παρουσιάστηκαν στις προηγούμενες ενότητες αφορούν και περιπτώσεις κατά τις οποίες πολλαπλές κεραιές μπορούν να χρησιμοποιούνται είτε μόνο στην πλευρά του δέκτη είτε μόνο στην πλευρά του πομπού, δηλαδή αναφέρονται και σε συστήματα SIMO (Single Input Multiple Output), MISO (Multiple Input Single Output). Η ταυτόχρονη συμμετοχή πολλαπλών κεραιών και στις δυο πλευρές του συστήματος που είναι και το βασικό στοιχείο των MIMO συστημάτων παρέχει τη δυνατότητα χωρικής πολυπλεξίας. Σε ένα περιβάλλον πλήρους σκέδασης όπου οι πολυδιαδρομικές συνιστώσες των σημάτων δημιουργούν υπο-κανάλια στα οποία αναφερθήκαμε και νωρίτερα μπορούν να χρησιμοποιηθούν κατάλληλα ώστε τα ανεξάρτητα κανάλια που προκύπτουν να μεταφέρουν διαφορετικές ροές δεδομένων. Έτσι μια ροή δεδομένων προς μετάδοση μπορεί να χωριστεί σε τόσα τμήματα όσα και ο αριθμός των κεραιών που συμμετέχουν στην εκπομπή του σήματος τα οποία θα μεταδοθούν ταυτόχρονα χωρίς καμία αύξηση στο εύρος ζώνης του συστήματος. Όπως είναι προφανές η χωρική πολυπλεξία δίνει δυνατότητα για μεγάλη αύξηση της χωρητικότητας και της φασματικής απόδοσης ενός συστήματος μετάδοσης. Ωστόσο τα ανεξάρτητα σήματα που μεταδίδονται λαμβάνονται στην πλευρά του δέκτη από όλες τις κεραιές που συμμετέχουν συνδυασμένα. Η αποτελεσματικότητα της χωρικής πολυπλεξίας των MIMO συστημάτων εξαρτάται από την μέθοδο με την οποία ο δέκτης καλείται να αποπλέξει τα σήματα που λαμβάνει, να διαχωρίσει τις ανεξάρτητες ροές δεδομένων της κάθε κεραιάς εκπομπής και να ανασυνθέσει την αρχική ροή δεδομένων με όσο το δυνατόν λιγότερα σφάλματα. Στο σημείο αυτό πρέπει να σημειώσουμε πως το κέρδος πολυπλεξίας προκύπτει, όχι υποχρεωτικά αλλά συνήθως σε βάρος του κέρδους διαφορισμού και ανάλογα με τις ανάγκες του εκάστοτε συστήματος πρέπει να επιλέγεται ο στόχος με τον οποίο θα χρησιμοποιείται η MIMO μετάδοση και η κατάλληλη τεχνική. Στην περίπτωση που το επιθυμητό αποτέλεσμα είναι η αξιοπιστία της μετάδοσης ενδείκνυται η χρησιμοποίηση Space Time Block Codes

ενώ όταν επιθυμούμε την αύξηση της χωρητικότητας του συστήματος οι τεχνικές Spatial Multiplexing είναι αυτές που απαντούν στο συγκεκριμένο ζήτημα.

1.3 Αντικείμενο Εργασίας

Αντικείμενο της εργασίας αυτής είναι να παρουσιάσει έξι τεχνικές λήψης για συστήματα MIMO Spatial Multiplexing. Οι έξι διαφορετικοί δέκτες έχουν διαφορετικές ιδιότητες και διαφορετικές επιδόσεις όσο αναφορά στο ρυθμό σφαλμάτων που επιτυγχάνεται σε σχέση με το σηματοθορυβικό λόγο. Η αξιολόγηση των επιδόσεών τους γίνεται μέσω προσομοιώσεων της λειτουργίας τους με τη βοήθεια του λογισμικού Matlab.

Πριν όμως ασχοληθούμε με τους δέκτες αυτών των συστημάτων παρουσιάζουμε στις επόμενες ενότητες τη θεωρία που επιτρέπει τη δημιουργία των υπο-καναλιών που πραγματοποιούν τη μετάδοση των ανεξάρτητων ροών δεδομένων και οδηγούν στην μεγάλη αύξηση της χωρητικότητας.

Κεφάλαιο 2

Θεωρία MIMO Συστημάτων

2.1 Χωρική Πολυπλεξία MIMO Καναλιού

Ένα σύστημα μετάδοσης δεδομένων στο οποίο συμμετέχουν n_t κεραιές εκπομπής και n_r κεραιές στην μεριά του δέκτη περιγράφεται από έναν μιγαδικό πίνακα H διαστάσεων $n_r \times n_t$:

$$H = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1n_t} \\ h_{21} & h_{22} & \dots & h_{2n_t} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n_r 1} & h_{n_r 2} & \dots & h_{n_r n_t} \end{bmatrix}$$

Τα στοιχεία h_{ij} (όπου $i = 1, 2, \dots, n_r$ και $j = 1, 2, \dots, n_t$) του πίνακα που χαρακτηρίζει το κανάλι είναι οι συναρτήσεις μεταφοράς από τον j πομπό στον i δέκτη. Έτσι η μετάδοση των δεδομένων μέσω του MIMO καναλιού πλέον περιγράφεται από την εξίσωση:

$$y = Hx + w$$

Όπου x είναι διάνυσμα εισόδου του συστήματος διαστάσεων $n_t \times 1$. Το διάνυσμα $x = [x_1 \ x_2 \ \dots \ x_{n_t}]^T$ αντιπροσωπεύει τα σύμβολα που αποστέλλονται από τα n_t στοιχεία της κεραιάς εκπομπής. Το διάνυσμα w είναι αυτό που περιλαμβάνει τις μιγαδικές μεταβλητές του θορύβου που προστίθενται στο σήμα που συλλέγει η κάθε κεραιά λήψης. Το διάνυσμα αυτό αφορά κάθε κεραιά ξεχωριστά και είναι φυσικό να περιλαμβάνει τόσες μεταβλητές όσες και οι κεραιές λήψης, έχει δηλαδή διαστάσεις $n_r \times 1$. Το διάνυσμα $y = [y_1 \ y_2 \ \dots \ y_{n_r}]^T$ αναπαριστά το σήμα που λαμβάνει η εκάστοτε κεραιά λήψης. Το σήμα y_i που λαμβάνει ο δέκτης είναι το

αποτέλεσμα της υπέρθεσης όλων των σημάτων που αποστέλλονται από όλες τις κεραίες εκπομπής. Οι δυνατότητες χωρικής πολυπλεξίας αυτού του καναλιού περιγράφονται μέσω ενός μετασχηματισμού που ονομάζεται Singular Value Decomposition (SVD) ο οποίος μετατρέπει το MIMO κανάλι σε n_{\min} κανάλια SISO (Single Input Single output).

Σύμφωνα με τον SVD μετασχηματισμό:

$$H = U \Lambda V^*$$

Τα σύμβολα U και V αναπαριστούν δύο μιγαδικούς πίνακες διαστάσεων $n_r \times n_r$ και $n_t \times n_t$ αντίστοιχα οι οποίοι όταν πολλαπλασιαστούν με τον αναστροφοσυζυγή τους προκύπτει ο μοναδιαίος πίνακας των αντίστοιχων διαστάσεων, δηλαδή $UU^* = I$ και $VV^* = I$. Ο πίνακας Λ είναι ορθογώνιος πίνακας διαστάσεων $n_r \times n_t$ ο οποίος έχει όλα τα στοιχεία του μηδενικά εκτός από αυτά της διαγωνίου, σ_i με $i = 1, \dots$. Οι τιμές $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{n_{\min}}$ είναι μη μηδενικοί πραγματικοί αριθμοί και αποτελούν τις ιδιόμορφες τιμές του πίνακα H όπου $n_{\min} = \min(n_r, n_t)$ με την προϋπόθεση ότι η μετάδοση πραγματοποιείται σε περιβάλλον που περιέχει πολλούς σκεδαστές και τα σήματα έχουν αρκετές πολυδιαδρομικές συνιστώσες. Με την κατάλληλη επεξεργασία που περιγράφεται από τις τρεις παρακάτω εξισώσεις:

- $x = V \tilde{x}$
- $\tilde{y} = U^* y$
- $\tilde{w} = U^* w$

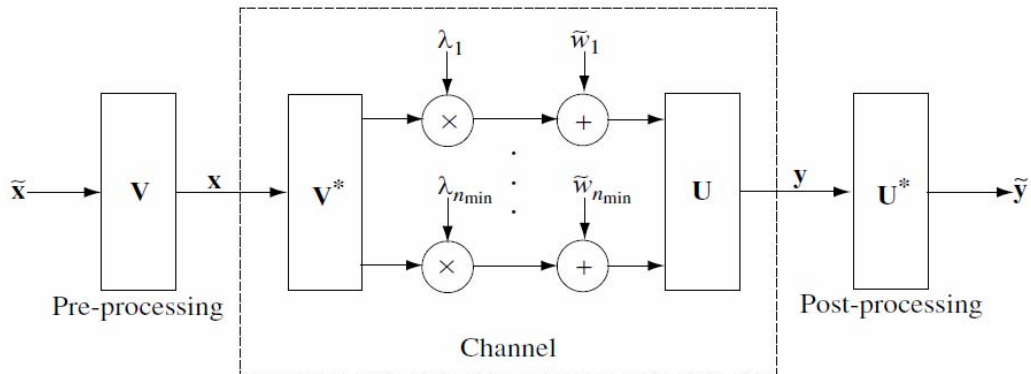
η εξίσωση που περιγράφει το κανάλι μετατρέπεται σε :

$$y = Hx + w \rightarrow U^* y = U^* (HV \tilde{x} + w)$$

Αντικαθιστώντας τον πίνακα H με το αποτέλεσμα του SVD μετασχηματισμού έχουμε

\tilde{y}

$$= U^* U \Lambda V^* V \tilde{x} + U^* w \Rightarrow \tilde{y} = \Lambda \tilde{x} + \tilde{w}$$



2.3 Μετατροπή MIMO καναλιού σε n_{\min} SISO με τη μέθοδο SVD[4]

Όπως φαίνεται και στην εικόνα 2.1 το MIMO κανάλι πλέον έχει μετατραπεί σε n_{\min} κανάλια SISO σύμφωνα με την εξίσωση:

$$y_i = \lambda_i x_i + \tilde{w}_i \quad \text{για} \quad i = 1, 2, \dots, n_{\min}$$

Οι μη μηδενικές τιμές λ_i^2 είναι οι ιδιοτιμές του πίνακα HH^* ή H^*H και το πλήθος των τιμών λ_i αποτελεί την τάξη του πίνακα H . Το πλήθος των τιμών αυτών αποτελεί τον αριθμό των βαθμών ελευθερίας του συστήματος που επιτρέπει τη μετάδοση ισάριθμων ανεξάρτητων ροών δεδομένων. Όπως αναφέρθηκε και νωρίτερα το πλήθος αυτό υπό τις κατάλληλες συνθήκες είναι ίσο με το μικρότερο από τους δυο αριθμούς κεραιών που βρίσκονται είτε από τη μεριά του πομπού είτε από αυτή του δέκτη, δηλαδή είναι ίσο με τη μικρότερη από τις δύο διαστάσεις του πίνακα H . Έχοντας πλήρη γνώση του χρονικά αμετάβλητου καναλιού και με πίνακα H πλήρους τάξης δηλαδή με το μέγιστο δυνατό αριθμό μη μηδενικών λ_i το κανάλι μπορεί να υποστηρίξει n_{\min} χωρικά πολυπλεγμένες ανεξάρτητες ροές δεδομένων. Η χωρητικότητα του καναλιού δίνεται από την εξίσωση:

$$C = \sum_{i=1}^{n_{\text{min}}} \log \left(1 + \frac{P_i * \lambda_i^2}{N_0} \right) \text{ bits/s/Hz}$$

όπου P_i είναι η ισχύς που ανατίθεται σε κάθε ροή δεδομένων που μεταδίδεται από κάθε στοιχείο της κεραίας εκπομπής και καθορίζεται $P_i = (\mu - N_0 / (\lambda_i^2 * 2))$ με χρήση της μεθόδου waterfilling[4] όπου μ έχει επιλεγεί έτσι ώστε το άθροισμα των P_i να μην υπερβαίνει τον περιορισμό της ισχύος που είναι προς μετάδοση από τον πομπό .

Εκτός από την τάξη του πίνακα H , δηλαδή το πλήθος των μη μηδενικών μοναδιαίων τιμών λ_i , για τη βέλτιστη χρήση του καναλιού εξετάζεται και η κατάσταση (condition) του πίνακα που περιγράφει το κανάλι. Τον αριθμό κατάστασης (condition number) καθορίζουν οι ίδιες οι τιμές λ_i και ορίζεται πως είναι ο λόγος $\max \lambda_i / \min \lambda_i$. Ο πίνακας H χαρακτηρίζεται well conditioned όταν ο λόγος αυτός πλησιάζει τη μονάδα και ill conditioned σε οποιαδήποτε άλλη περίπτωση. Ισχύει ότι σε περιπτώσεις που το κανάλι έχει υψηλό σηματοθορυβικό λόγο επιτυγχάνεται η μεγαλύτερη δυνατή χωρητικότητα όταν οι ιδιοτιμές λ_i είναι όμοιες μεταξύ τους [4]. Στην αντίθετη περίπτωση το βέλτιστο είναι να μεταδίδεται όλη η ισχύς πάνω από το ισχυρότερο ιδιοκανάλι δηλαδή το κανάλι με τη μέγιστη τιμή λ_i .

Από τα στοιχεία που παρουσιάσαμε καταλήγουμε στο συμπέρασμα πως κάτω από τις κατάλληλες συνθήκες ένα MIMO Spatial Multiplexing σύστημα μετάδοσης δεδομένων εκμεταλλευόμενο τη διάσταση του χώρου μπορεί να οδηγήσει σε μεγάλη αύξηση της χωρητικότητας. Στην επόμενη ενότητα περιγράφεται η θεωρία και οι προϋποθέσεις οι οποίες επιτρέπουν την δημιουργία των $n_{\text{min}} \times n_{\text{min}}$ ανεξάρτητων καναλιών από ένα MIMO κανάλι.

2.2 Μετάδοση σήματος και βαθμοί ελευθερίας του συστήματος

Στην ενότητα αυτή παρουσιάζεται ο τρόπος με τον οποίο προκύπτουν επιπλέον βαθμοί ελευθερίας σε ένα σύστημα που χρησιμοποιούνται πολλαπλές κεραίες για την μετάδοση των δεδομένων. Για να γίνει αυτό εξετάζουμε αναλυτικά

τις περιπτώσεις που πολλαπλές κεραιές χρησιμοποιούνται είτε στην πλευρά λήψης είτε στην πλευρά λήψης και στη συνέχεια την περίπτωση κατά την οποία συμμετέχουν πολλαπλές κεραιές και στις δυο πλευρές.

2.2.1 SIMO (Single Input Multiple Output)

Αρχικά εξετάζουμε την περίπτωση κατά την οποία υπάρχει μετάδοση σήματος από μια μόνο κεραιά στον πομπό σε πολλές κεραιές στην πλευρά του δέκτη μέσω μόνο του μονοπατιού οπτικής επαφής (Line of Sight) μεταξύ των δύο σημείων χωρίς να λαμβάνονται υπόψη οι επιδράσεις από ανακλάσεις και σκεδαστές που ενδεχομένως να υπάρχουν. Στην μετάδοση που περιγράφουμε τα στοιχεία της κεραιάς δέκτη είναι σε απόσταση $\Delta_r \lambda_c$ μεταξύ τους, όπου λ_c είναι το μήκος κύματος του σήματος και Δ_r η κανονικοποιημένη απόσταση μεταξύ των κεραιών δηλαδή ο αριθμός των μηκών κύματος που τις χωρίζει. Το SIMO κανάλι περιγράφεται από την εξίσωση:

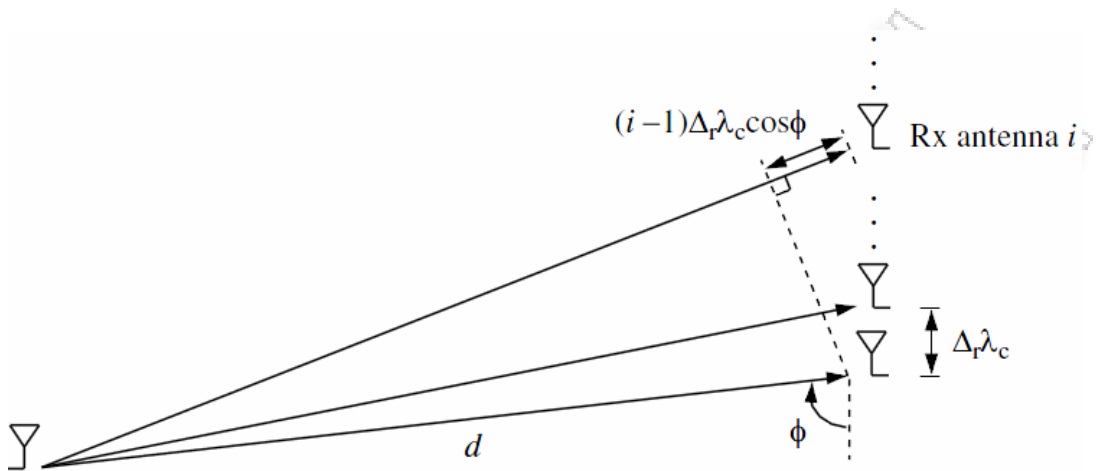
$$y = hx + w$$

όπου $h = [h_1 \quad h_2 \quad \dots \quad h_{n_r}]^T$ είναι το διάνυσμα των συναρτήσεων μεταφοράς από τον πομπό προς τους n_r διαφορετικούς δέκτες, x είναι το προς μετάδοση σύμβολο και w είναι ο θόρυβος που προστίθεται στη μεριά του δέκτη. Η συνάρτηση μεταφοράς που περιγράφει τη μετάδοση προς το i οστό στοιχείο της κεραιάς δέκτη περιγράφεται από την εξίσωση

$$h_i = a \exp\left(-\frac{j2\pi d_i}{\lambda_c}\right)$$

όπου a είναι η εξασθένιση του καναλιού και d_i η απόσταση του πομπού από το συγκεκριμένο στοιχείο. Όπως φαίνεται και στην εικόνα 2.2, το σημείο διαφοροποίησης είναι οι διαφορετικές αποστάσεις d_i που διανύει το σήμα οδεύοντας προς το κάθε στοιχείο της κεραιάς δέκτη. Η απόσταση αυτή δίνεται από την εξίσωση: $d_i \approx d + (i - 1)\Delta_r \lambda_c \cos\varphi$ για $i = 1, 2, \dots, n_r$ όπου d είναι η απόσταση του πομπού

από το πρώτο στοιχείο της κεραίας δέκτη και ϕ είναι η γωνία πρόσπτωσης του σήματος προς τη συστοιχία κεραίων του δέκτη.



2.4 Line of Sight SIMO μετάδοση

Ορίζοντας το κατευθυντικό συνήμιτονο $\Omega = \cos\phi$ η συνάρτηση μεταφοράς

$\mathbf{h} = [h_1, \dots, h_{n_r}]$ γράφεται:

$$\mathbf{h} = a \exp\left(-\frac{j2\pi d}{\lambda_c}\right) \begin{bmatrix} 1 \\ \exp(-2\pi\Delta_r\Omega) \\ \exp(-2\pi 2\Delta_r\Omega) \\ \vdots \\ \exp(-2\pi(n_r - 1)\Delta_r\Omega) \end{bmatrix}$$

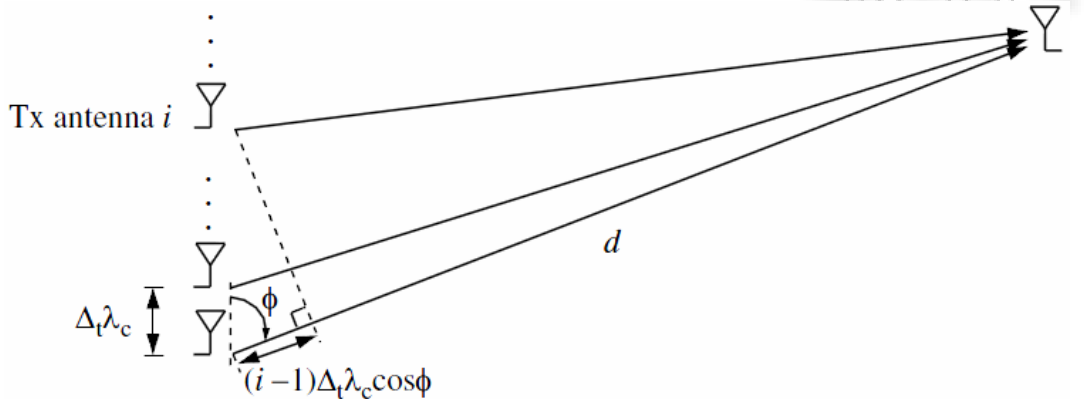
Τα σήματα που φτάνουν στις κεραίες του δέκτη έχουν διαφορετική φάση κατά λόγω της σχετικής καθυστέρησης στη διάδοση. Ορίζεται σαν χωρική υπογραφή το διάνυσμα :

$$\mathbf{s}_r(\Omega) = \frac{1}{\sqrt{n_r}} \begin{bmatrix} 1 \\ \exp(-2\pi\Delta_r\Omega) \\ \exp(-2\pi 2\Delta_r\Omega) \\ \vdots \\ \exp(-2\pi(n_r - 1)\Delta_r\Omega) \end{bmatrix}$$

Ο ιδανικός δέκτης συνδυάζει τα σήματα που φτάνουν στα διαφορετικά στοιχεία του δημιουργώντας ένα κέρδος ισχύος της τάξης n_r που αποτυπώνεται και στη χωρητικότητα του συστήματος :

$$C = \log \left(1 + \frac{P \|h\|^2}{N_0} \right) = \frac{\log \left(1 + \frac{P \alpha^2 n_r}{N_0} \right) b}{\frac{s}{Hz}}$$

2.2.2 MISO (Multiple Input Single Output)



2.5 Line of Sight MISO μετάδοση

Στην εικόνα 2.3 παρουσιάζεται επίσης η περίπτωση κατά την οποία οι πολλαπλές κεραιές βρίσκονται στη μεριά του πομπού (MISO). Σε αυτή την περίπτωση το σύστημα περιγράφεται από την εξίσωση: $\mathbf{y} = \mathbf{h}'\mathbf{x} + w$. Όταν τα στοιχεία της κεραιάς πομπού απέχουν μεταξύ τους κατά $\Delta t \lambda_c$ και όπως και στην SIMO περίπτωση η διάδοση του σήματος γίνεται μόνο μέσω της οπτικής επαφής πομπού και το διάνυσμα \mathbf{h} εκφράζεται αντίστοιχα ως:

$$\mathbf{h} = a \exp \left(-\frac{j2\pi d}{\lambda_c} \right) \begin{bmatrix} 1 \\ \exp(-2\pi \Delta_t \Omega) \\ \exp(-2\pi 2 \Delta_t \Omega) \\ \vdots \\ \exp(-2\pi (n_t - 1) \Delta_t \Omega) \end{bmatrix}$$

Το κατευθυντικό συννημίτονο Ω είναι $\Omega = \cos \phi$ όπου ϕ είναι η γωνία που σχηματίζει η διεύθυνση διάδοσης του σήματος με τη συστοιχία κεραιών του πομπού. Η ιδανική μετάδοση γίνεται και σε αυτή την περίπτωση κατά τη διεύθυνση $\mathbf{e}_r(\Omega)$ όπου

$$\mathbf{e}_t(\Omega) = \frac{1}{\sqrt{n_t}} \begin{bmatrix} 1 \\ \exp(-2\pi\Delta_t\Omega) \\ \exp(-2\pi 2\Delta_t\Omega) \\ \vdots \\ \exp(-2\pi(n_t - 1)\Delta_t\Omega) \end{bmatrix}$$

Με το αποτέλεσμα να είναι το ίδιο με την περίπτωση της SIMO μετάδοσης δηλαδή ένα κέρδος ισχύος της τάξης n_t .

2.2.3 MIMO (Multiple Input Multiple Output)

Ας υποθέσουμε τώρα πως έχουμε ένα σύστημα στο οποίο συμμετέχουν πολλαπλές κεραίες και στις δυο πλευρές, πομπού και δέκτη, αλλά η μετάδοση γίνεται μόνο μέσω του μονοπατιού οπτικής επαφής (LOS). Η συνάρτηση μεταφοράς για το μονοπάτι μετάδοσης από το $k = 1, 2, \dots, n_t$ στοιχείο του πομπού στο $i = 1, 2, \dots, n_r$ στοιχείο του δέκτη, σύμφωνα με τα όσα δείξαμε στις προηγούμενες ενότητες είναι:

$$h_{ik} = a \exp\left(-\frac{j2\pi d_{ik}}{\lambda_c}\right)$$

$$\text{με } d_{ik} = d + (i - 1)\Delta_r \lambda_c \cos\varphi_r - (k - 1)\Delta_t \lambda_c \cos\varphi_t$$

όπου d_{ik} είναι η απόσταση μεταξύ των στοιχείων i και k και d είναι η απόσταση μεταξύ των δύο συστοιχιών κεραιών. Αντικαθιστώντας την εξίσωση που περιγράφει την απόσταση d_{ik} προκύπτει :

$$h_{ik} = a \exp\left(\frac{-j2\pi d}{\lambda_c}\right) \exp(j2\pi(k - 1)\Delta_t\Omega_t) \exp(j2\pi(i - 1)\Delta_r\Omega_r)$$

Χρησιμοποιώντας τις εξισώσεις των χωρικών υπογραφών που χρησιμοποιήθηκαν στα συστήματα SIMO και MISO, μπορούμε να γράψουμε τον πίνακα που περιγράφει τις συναρτήσεις μεταφοράς για όλα τα μονοπάτια που περιλαμβάνει το MIMO σύστημα:

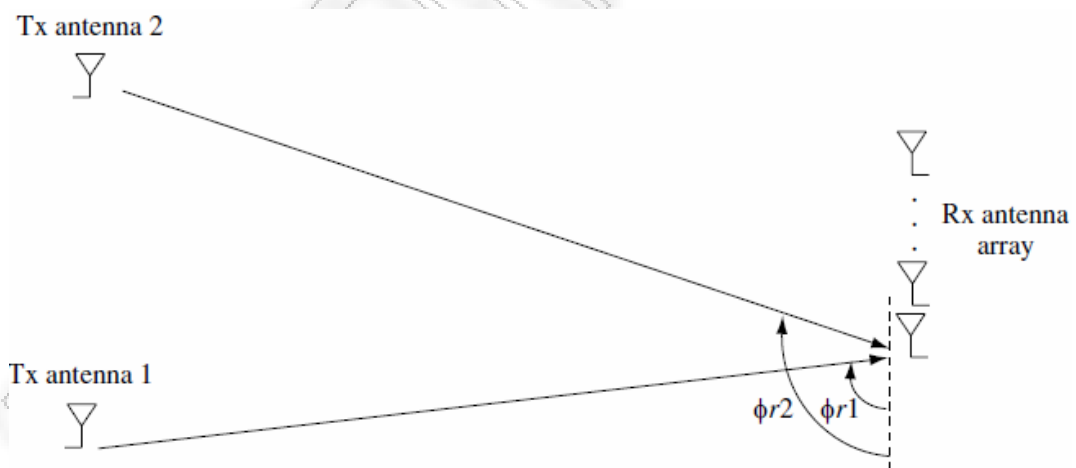
$$\mathbf{H} = a\sqrt{n_t n_r} \exp\left(\frac{-j2\pi d}{\lambda_c}\right) \mathbf{e}_r(\Omega_r) \mathbf{e}_t(\Omega_t)^T$$

Ο πίνακας που προκύπτει έχει μόνο μια μη μηδενική μοναδιαία τιμή $\lambda =$ και η χωρητικότητα του συστήματος διαμορφώνεται ως εξής:

$$C = \frac{\log\left(1 + \frac{P\alpha^2 n_t n_r}{N_0}\right) b}{Hz}$$

Παρατηρούμε πως παρά την ύπαρξη πολλαπλών κεραιών εισόδου και εξόδου στο σύστημά μας οι βαθμοί ελευθερίας δεν παρουσιάζουν καμία αύξηση και αυτό λόγω του ότι η διεύθυνση της διάδοσης των σημάτων σχηματίζει την ίδια γωνία με όλα τα στοιχεία των κεραιών. Έτσι το σύστημα έχει μόνο ένα βαθμό ελευθερίας και το μοναδικό κέρδος είναι ένα κέρδος ισχύος της τάξης $n_t n_r$.

Για να επιτευχθεί η αύξηση στους βαθμούς ελευθερίας του συστήματος τοποθετούμε τις κεραιές του πομπού σε απόσταση μεταξύ τους αρκετά μεγάλη, συγκρίσιμη με την απόσταση μεταξύ πομπού και δέκτη όπως φαίνεται και στην εικόνα 2.4



2.6 MIMO Σύστημα με κεραιές εκπομπής σε μεγάλη απόσταση μεταξύ τους

Όπως και στις προηγούμενες περιπτώσεις θεωρούμε πως η διάδοση γίνεται μόνο μέσω του μονοπατιού οπτικής επαφής μεταξύ πομπού και δέκτη. Επίσης θεωρούμε πως για τις δυο διαφορετικές διαδόσεις από τις κεραιές εκπομπής τα κανάλια $k=1,2$ έχουν απώλειες α_k γωνίες πρόσπτωσης φ_{rk} έτσι προκύπτει :

$$h_k = \alpha_k \sqrt{n_r} \exp\left(\frac{-j2\pi d_{1k}}{\lambda_c}\right) \mathbf{e}_r(\Omega_{rk}), \quad k = 1,2$$

Με d_{1k} συμβολίζεται η απόσταση μεταξύ κεραιάς εκπομπής k και της πρώτης κεραιάς της συστοιχίας των κεραιών του δέκτη. Λόγω της περιοδικότητας της συνάρτησης $\mathbf{e}_r(\Omega)$ η οποία αποδεικνύεται ότι έχει περίοδο $1/\Delta_r$ φτάνουμε στο συμπέρασμα ότι ο πίνακας $\mathbf{H}=[\mathbf{h}_1, \mathbf{h}_2]$ έχει γραμμικώς ανεξάρτητες στήλες εφόσον η διαφορά των γωνιών με τις οποίες προσπίπτουν τα σήματα στην κεραιά του δέκτη δεν είναι μηδενική και δεν είναι πολλαπλάσιο της τιμής

$$\Omega_r = \Omega_{r2} - \Omega_{r1} \neq 0 \text{ mod } \frac{1}{\Delta_r}.$$

Ο πίνακας \mathbf{H} έχει δυο μη μηδενικές ιδιοτιμές και τα σήματα που φεύγουν από τις δυο διαφορετικές κεραιές μπορούν να διαχωριστούν επιτυχώς στην πλευρά του δέκτη. Συνυπολογίζοντας το γεγονός ότι οι τιμές Ω_{r2} και Ω_{r1} αποτελούν κατευθυνικά συνημίτονα συμπεραίνουμε ότι για να επαληθευτεί η παραπάνω σχέση αρκεί να είναι διαφορετικά μεταξύ τους εφόσον η απόσταση μεταξύ των στοιχείων της κεραιάς δέκτη είναι μικρότερο από μισό μήκος κύματος δηλαδή $\Delta_r \leq 1/2$.

Αν αντιστρέψουμε τη διάταξη του προηγούμενου συστήματος μετάδοσης δεδομένων, δηλαδή αν έχουμε συστοιχία κεραιών στην πλευρά του πομπού και δύο απομακρυσμένες μεταξύ τους κεραιές στη μεριά του δέκτη, προκύπτει παρόμοιο σύστημα με το προηγούμενο. Οι ίδιοι κανόνες και τύποι που χρησιμοποιήθηκαν δημιουργούν ένα MIMO σύστημα με δυο βαθμούς ελευθερίας. Η διαφορά είναι πως

σε αυτή την περίπτωση ο πίνακας που περιγράφει το σύστημα είναι $\mathbf{H} = \begin{bmatrix} h'_1 \\ h'_2 \end{bmatrix}$ και

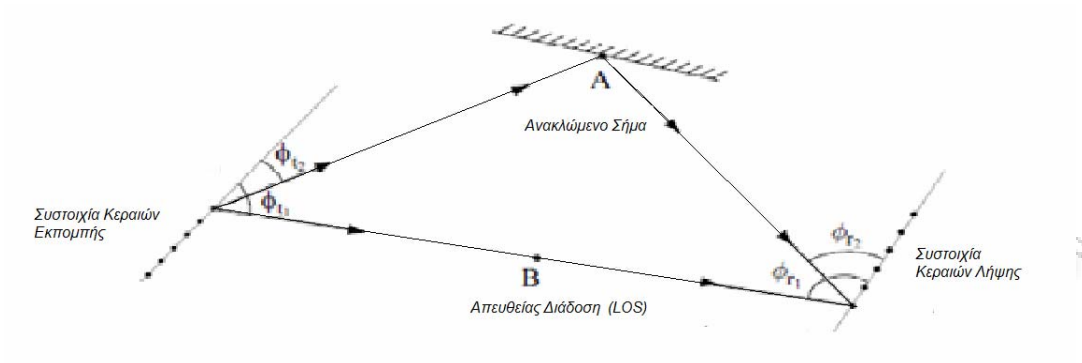
οι γραμμές του είναι ανεξάρτητες με την προϋπόθεση ότι για τις γωνίες εκπομπής

προς τις κεραιές λήψης ισχύει:

$$\Omega_t = \Omega_{t2} - \Omega_{t1} \neq 0 \pmod{\frac{1}{\Delta t}}$$

Στο ίδιο αποτέλεσμα μπορούμε να φτάσουμε χωρίς να τοποθετήσουμε σε μεγάλη απόσταση μεταξύ τους ούτε τις κεραιές του εκπομπής ούτε τις κεραιές λήψης, απλά λαμβάνοντας υπόψη στη μετάδοση εκτός από το μονοπάτι της οπτικής επαφής και ένα ακόμα μονοπάτι που προκύπτει από την ανάκλαση του σήματος σε κάποιο αντικείμενο. Το σύστημα αυτό όπως φαίνεται στην εικόνα 2.5 μπορεί να αναλυθεί σε δυο υποσυστήματα. Το πρώτο μέρος περιλαμβάνει τη συστοιχία κεραιών εκπομπής και σαν δέκτες απομακρυσμένους μεταξύ τους θεωρούμε τα σημεία A και B. Αυτό το κομμάτι περιγράφεται με τον πίνακα του προηγούμενου

μοντέλου $\mathbf{H}_1 = \begin{bmatrix} h_1^t \\ h_2^t \end{bmatrix}$. Το υπόλοιπο τμήμα της διάδοσης του σήματος αντιστοιχεί στο πρώτο μοντέλο μετάδοσης με απομακρυσμένες μεταξύ τους κεραιές εκπομπής στα σημεία A και B και δέκτη με συστοιχία κεραιών. Ο πίνακας περιγράφεται ως $\mathbf{H}_2 = \begin{bmatrix} h_1^r & h_2^r \end{bmatrix}$. Οι παραπάνω πίνακες όπως δείξαμε στα προηγούμενα παραδείγματα αντιστοιχούν σε συστήματα μετάδοσης με 2 βαθμούς ελευθερίας εξαιτίας της διαφορετικής γωνίας πρόσπτωσης των δυο σημάτων στις συστοιχίες των κεραιών πομπού και δέκτη αντίστοιχα. Το συνολικό σύστημα που προκύπτει $\mathbf{H} = \mathbf{H}_1 \mathbf{H}_2$ έχει και αυτό 2 βαθμούς ελευθερίας τους οποίους μπορούμε να εκμεταλλευτούμε είτε για την αύξηση της χωρητικότητας είτε για να δημιουργήσουμε μια πιο αξιόπιστη ζεύξη μεταξύ πομπού και δέκτη. [4]



2.7 MIMO μετάδοση με απευθείας οπτική επαφή και ένα ανακλώμενο σήμα. [4]

2.3 Channel Matrix Condition

Όπως είδαμε στην προηγούμενη ενότητα ο πίνακας H που περιγράφει το MIMO κανάλι ενός συστήματος που περιλαμβάνει 2 κεραίες εκπομπής και n_r κεραίες λήψης μπορεί υπό προϋποθέσεις να παρέχει δυο βαθμούς ελευθερίας. Για να ισχύει κάτι τέτοιο και ο πίνακας να είναι πλήρους τάξης, δηλαδή 2^{as} θα πρέπει για τα κατευθυντικά συνημίτονα να ισχύει : $\Omega_r = \Omega_{r2} - \Omega_{r1} \neq 0 \pmod{\frac{1}{\Delta_r}}$ δηλαδή $\cos\varphi_{r1} \neq \cos\varphi_{r2}$ για $\Delta_r \leq 0.5$. Ακόμα και όταν επαληθεύεται η σχέση αυτή θα πρέπει ο πίνακας να μπορεί να χαρακτηριστεί well – conditioned για να μπορεί να αυξηθεί αρκετά η χωρητικότητα του συστήματος.

Η κατάσταση (Condition) του πίνακα H εξαρτάται αποκλειστικά από τη σχέση μεταξύ των γωνιών με τις οποίες προσπίπτουν τα σήματα στις κεραίες. Όσο περισσότερο ευθυγραμμισμένες μεταξύ τους είναι οι χωρικές υπογραφές των σημάτων που προέρχονται από τις διαφορετικές κεραίες εκπομπής, τόσο χειρότερη η κατάσταση του πίνακα (ill conditioned H) και όσο λιγότερο ευθυγραμμισμένες τόσο καλύτερη (well conditioned). Αποδεικνύεται ότι για ένα συμβεί αυτό για την

περίπτωση που $\Delta_r \leq 0.5$ αρκεί $\cos\varphi_{r1} - \cos\varphi_{r2} \geq \frac{1}{L_r}$ όπου L_r είναι το κανονικοποιημένο μέγεθος της κεραίας $L_r = \Delta_r n_r$ δηλαδή το μήκος της συστοιχίας κεραίων μετρημένο σε μήκη κύματος. Ο δέκτης μπορεί να διαχωρίσει τα σήματα που έρχονται με διαφορά συνημίτονων γωνίας πρόσπτωσης μεγαλύτερη από . Αυτό

σημαίνει πως η δυνατότητα ενός δέκτη να διακρίνει σήματα που έρχονται από διαφορετικές κατευθύνσεις περιορίζεται από το μήκος του δέκτη και δεν εξαρτάται από τον αριθμό των στοιχείων που συγκροτούν την κεραία. Οι παραπάνω προϋποθέσεις ισχύουν και για τη γενικότερη περίπτωση που έχουμε δυο κεραίες με πολλαπλά στοιχεία και στην πλευρά του πομπού και την πλευρά του δέκτη ώστε να έχουμε πίνακα που περιγράφει το σύστημα πλήρους τάξης και well conditioned, δηλαδή το σύστημα να παρέχει το μέγιστο αριθμό βαθμών ελευθερίας και να είναι ικανό να εκμεταλλευτεί το γεγονός αυτό για την αύξηση της χωρητικότητας. Σύμφωνα με αυτές τις προϋποθέσεις μπορούμε εύκολα να συμπεράνουμε πως όταν τα σήματα μεταδίδονται με απευθείας LOS επαφή από τον πομπό στο δέκτη τότε οι γωνίες πρόσπτωσης και στις δυο πλευρές είναι περίπου ίσες και το σύστημα μπορεί να δώσει μόνο ένα βαθμό ελευθερίας. Το μόνο κέρδος που προκύπτει στη συγκεκριμένη περίπτωση είναι το κέρδος ισχύος. Για να μπορέσουμε να έχουμε το καλύτερο δυνατό αποτέλεσμα πρέπει τα διαφορετικά σήματα να μεταφέρονται διανύοντας διαφορετικές διαδρομές. Το MIMO σύστημα εκμεταλλεύεται τις πολυδιαδρομικές συνιστώσες που προκύπτουν όταν πραγματοποιείται μετάδοση σήματος σε ένα περιβάλλον που περιέχει διάφορους σκεδαστές. Το γεγονός πως οι γωνίες πρόσπτωσης πρέπει να διαφέρουν και στις δυο πλευρές πομπού και δέκτη δημιουργεί πρόβλημα σε περιπτώσεις όπως τα κινητά δίκτυα όπου η κεραία της κινητής συσκευής περιβάλλεται από σκεδαστές ενώ η κεραία του σταθμού βρίσκεται μακριά από αυτούς, συνήθως σε κάποιο υπερυψωμένο σημείο.

Η δυνατότητα διαχωρισμού των σημάτων από το δέκτη αποτυπώνεται από το διάγραμμα ακτινοβολίας της κεραίας που περιγράφεται από τη συνάρτηση $f_r(\cos\varphi - \cos\varphi_0)$ για το σήμα που έρχεται από την κατεύθυνση φ_0 . Η συνάρτηση f_r είναι :

- Περιοδική με περίοδο $\frac{\pi_r}{L_r} = \frac{1}{\Delta_r}$
- Μεγιστοποιείται στο σημείο $\cos\varphi = \cos\varphi_0$ $f_r(0) = 1$

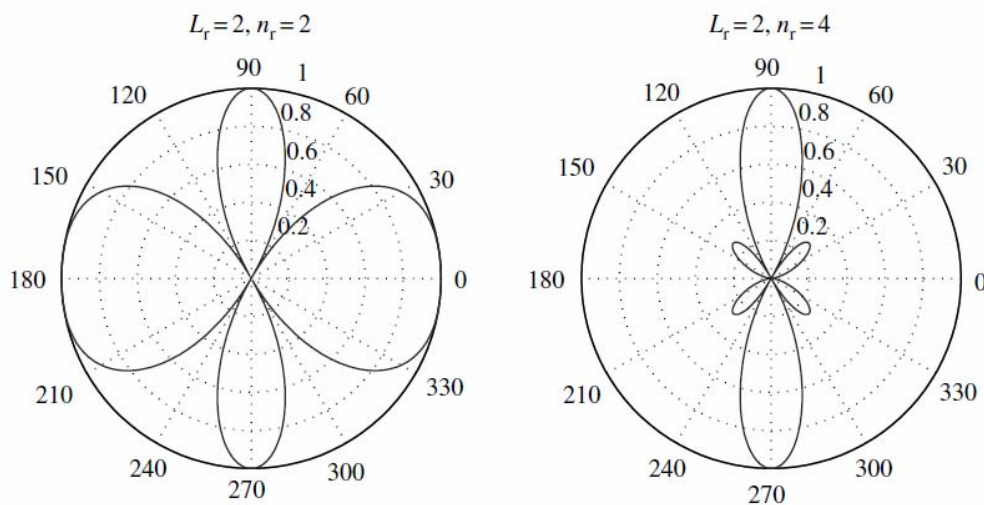
$$\cos\varphi - \cos\varphi_0 = \frac{k}{L_r} \text{ με } k = 1, 2, \dots, n_r - 1$$

- Μηδενίζεται στα σημεία όπου

Το διάγραμμα ακτινοβολίας σε πολικές συντεταγμένες ($\varphi, |f_r(\cos\varphi - \cos\varphi_0)|$) φαίνεται στην εικόνα 2.6 για τις περιπτώσεις όπου τα στοιχεία της κεραίας έχουν απόσταση μεταξύ τους $0,5\lambda$ και λ . Το διάγραμμα αυτό έχει κύριους λοβούς ακτινοβολίας στη γωνία φ_0 και σε οποιαδήποτε άλλη γωνία για την οποία ισχύει:

$$\cos\varphi = \cos\varphi_0 \text{ mod } \frac{1}{\Delta_r}$$

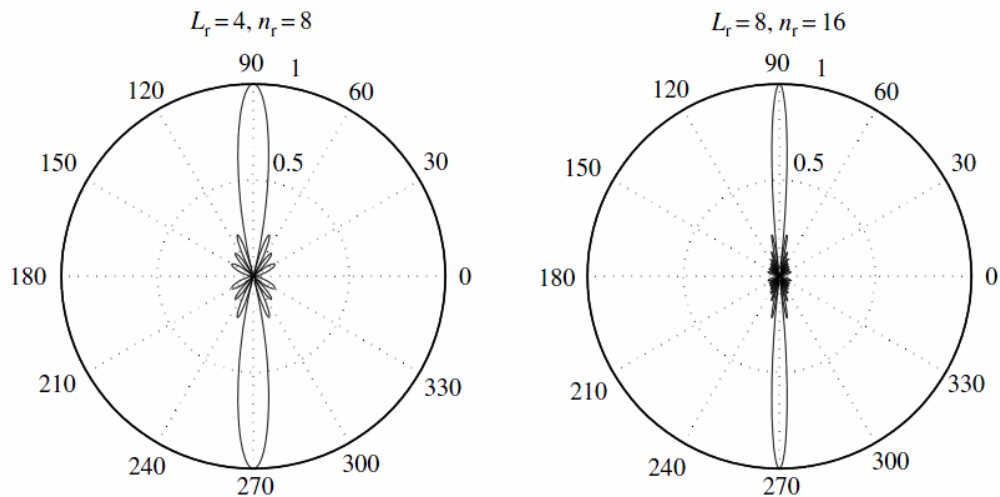
. Όσο η απόσταση μεταξύ των στοιχείων του δέκτη Δ_r είναι μικρότερη από $0,5$ τότε το διάγραμμα ακτινοβολίας έχει δυο κύριους λοβούς ακτινοβολίας, αυτόν στη γωνία φ_0 μαζί με το συμμετρικό του στη γωνία $-\varphi_0$. Όσο αυξάνεται η απόσταση μεταξύ των στοιχείων όπως είναι φυσιολογικό εμφανίζονται και επιπλέον λοβοί σε άλλες γωνίες που ικανοποιούν τις απαιτούμενες συνθήκες.



2.6 Διάγραμμα ακτινοβολίας [4]

Ο κύριος λοβός ακτινοβολίας έχει εύρος δέσμης ίσο με $2/L_r$ και οι ακτίνες σημάτων που λαμβάνονται από το δέκτη θα πρέπει για να είναι διαχωρίσιμες να διαφέρουν τα συνημίτονα των γωνιών με τις οποίες φτάνουν να έχουν διαφορά μεγαλύτερη από $1/L_r$. Στην εικόνα 2.7 παρακάτω αποτυπώνεται το γεγονός πως το εύρος δέσμης του

δέκτη εξαρτάται από το κανονικοποιημένο μέγεθος της κεραίας L_r . Όσο μεγαλύτερο γίνεται το μέγεθος της κεραίας τόσο πιο στενό γίνεται το εύρος δέσμης του δέκτη.



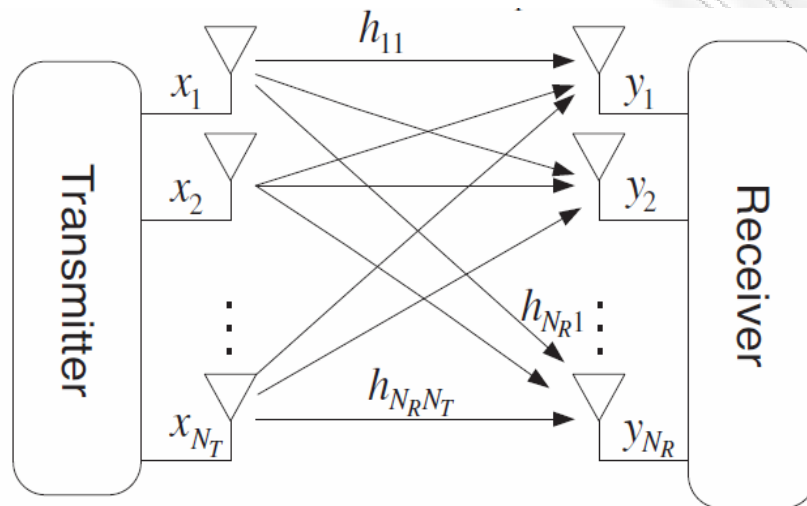
2.7 Εύρος δέσμης σε σχέση με το κανονικοποιημένο μέγεθος δέκτη.[4]

Ο ρόλος του μήκους της κεραίας δέκτη L_r είναι παρόμοιος με αυτόν του εύρους ζώνης ενός σήματος W . Η τιμή $1/L_r$ καθορίζει τη δυνατότητα διαχωρισμού των σημάτων στο πεδίο του χώρου ενώ η παράμετρος $1/W$ καθορίζει τη δυνατότητα διαχωρισμού στο πεδίο του χρόνου. Όταν δύο σήματα καταφθάνουν σε ένα δέκτη μέσω διαφορετικών διαδρομών με χρονική καθυστέρηση μικρότερη από $1/W$ τότε ο δέκτης δεν μπορεί να τα διαχωρίσει και να τα ανιχνεύσει ξεχωριστά ενώ όταν τα δυο σήματα καταφθάνουν στον δέκτη με διαφορά κατευθυντικών συνημίτονων μικρότερη από $1/L_r$ είναι συσχετισμένα και το σύστημα δεν παρέχει επιπλέον βαθμούς ελευθερίας.

ПАМЯТИ ИМЕНИТЕРА

Κεφάλαιο 3

Χωρητικότητα Συστημάτων MIMO



3.8 NR x NT MIMO system

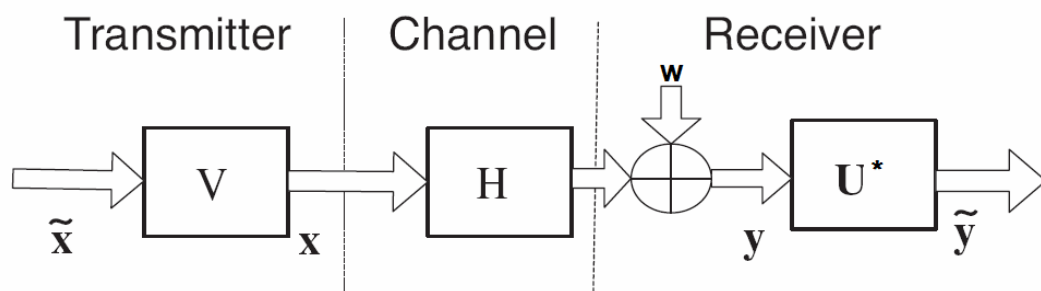
Στην εικόνα 3.1 φαίνεται η διάταξη ενός MIMO συστήματος με N_T κεραιές εκπομπής και N_R κεραιές στην πλευρά του δέκτη. Το κανάλι περιγράφεται από τον μιγαδικό πίνακα H διαστάσεων $N_R \times N_T$. Η πληροφορία που μεταδίδεται είναι το διάνυσμα $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_{N_T}]^T$ $N_T \times 1$ του οποίου οι γραμμές είναι τα ανεξάρτητα σύμβολα προς μετάδοση $x_1 \ x_2 \ \dots \ x_{N_T}$. Το σήμα που λαμβάνεται ο δέκτης συμβολίζεται με $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_{N_R}]^T$ είναι διαστάσεων $N_R \times 1$ και προκύπτει ως αποτέλεσμα της μετάδοσης που περιγράφεται από την εξίσωση: $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w}$. Με το διάνυσμα \mathbf{w} που έχει διαστάσεις ίδιες με αυτές του διανύσματος \mathbf{y} συμβολίζουμε το θόρυβο που προστίθεται στην πλευρά του δέκτη και είναι κυκλική συμμετρική γκαουσιανή μεταβλητή με μηδενική μέση τιμή. Αποδεικνύεται ότι η χωρητικότητα αυτών των συστημάτων δίνεται από την εξίσωση [4]:

$$C = \log_2 \det \left(\mathbf{I}_{N_R} + \frac{SNR}{N_T} \mathbf{H} \mathbf{R}_{\mathbf{x}\mathbf{x}} \mathbf{H}^* \right)$$

Με συμβολίζεται η αυτοσυσχέτιση του διανύσματος μετάδοσης. Αν υποθέσουμε πως η ισχύς που αποδίδεται σε κάθε ροή δεδομένων που μεταδίδεται από την εκάστοτε κεραία είναι 1 τότε το άθροισμα της διαγωνίου αυτού του πίνακα είναι ίσο με N_T .

3.1 Χωρητικότητα με γνωστό πίνακα καναλιού (Full CSI)

Η χωρητικότητα που μπορεί να παρέχει ένα MIMO κανάλι εξετάζεται για δύο διαφορετικές περιπτώσεις. Η παρούσα ενότητα αφορά στην περίπτωση κατά την οποία και οι δύο πλευρές επικοινωνίας έχουν πλήρη γνώση για την κατάσταση του MIMO καναλιού μέσα από το οποίο πραγματοποιείται η μετάδοση των δεδομένων. Όταν υπάρχει πλήρης γνώση του καναλιού, full CSI (Channel State Information) υπάρχει η δυνατότητα σύμφωνα με τα χαρακτηριστικά του καναλιού να γίνεται προεπεξεργασία των προς μετάδοση δεδομένων που οδηγεί σε υψηλή αποδοτικότητα φάσματος. Η εικόνα 3.2 δείχνει την διαδικασία που ακολουθείται για αυτό το σκοπό.



3.9 MIMO μετάδοση με χρήση του SVD μετασχηματισμού[5]

Οι πίνακες V και U^* οι οποίοι χρησιμοποιούνται για την επεξεργασία του σήματος πριν και μετά τη μετάδοσή είναι προϊόντα του SVD μετασχηματισμού του πίνακα H που περιγράφει το κανάλι. Με αυτόν τον τρόπο το MIMO κανάλι μπορεί να μοντελοποιηθεί ως $n = \min(N_T, N_R)$ SISO κανάλια. Η συνολική χωρητικότητα του καναλιού υπολογίζεται ως το άθροισμα των επιμέρους χωρητικότητων των απλών καναλιών. Η χωρητικότητα υπολογίζεται από την παρακάτω εξίσωση:

$$C = \frac{\sum_{i=1}^n \log_2 \left(1 + \frac{SNR * \gamma_i \lambda_i}{N_T} \right) bps}{Hz}$$

Όπου n είναι το πλήθος των ανεξάρτητων SISO καναλιών στα οποία αποσυντίθεται το MIMO κανάλι μέσω του SVD μετασχηματισμού. λ_i είναι οι ιδιοτιμές του πίνακα $\mathbf{H}\mathbf{H}^*$ ή $\mathbf{H}^*\mathbf{H}$ που περιγράφει το κανάλι. Η τιμή γ_i αφορά στο διαμοιρασμό την συνολικά εκπεμπόμενης ισχύος μεταξύ των διαφορετικών καναλιών που προκύπτουν. Ο συντελεστής γ_i καθορίζει την ισχύ που αποδίδεται σε κάθε στοιχείο της κεραίας εκπομπής. Ο υπολογισμός αυτού του συντελεστή γίνεται ιδανικά με τη μέθοδο water pouring. Σύμφωνα με τον αλγόριθμο αυτό ο υπολογισμός του συντελεστή γ_i γίνεται μέσω της σχέσης:

$$\gamma_i = \left(\mu - \frac{N_T}{SNR * \lambda_i} \right)^+ \text{ για } i = 1, 2, \dots, n$$

Η τιμή μ επιλέγεται με κατάλληλο τρόπο ώστε οι τιμές των συντελεστών γ_i να μην οδηγούν στην υπέρβαση του περιορισμού της συνολικά εκπεμπόμενης ισχύος. Η παρουσία του πρόσημου + στον εκθέτη της εξίσωσης σημαίνει πως ο συντελεστής γ_i παίρνει την τιμή που προκύπτει μόνο για τις περιπτώσεις που το αποτέλεσμα είναι θετικό ενώ στην αντίθετη περίπτωση μηδενίζεται. Σύμφωνα με την υπόθεση που κάναμε για το σύστημα οι συντελεστές που προκύπτουν μέσω του water pouring αλγορίθμου θα πρέπει να επαληθεύουν τη σχέση:

$$\sum_{i=1}^n \gamma_i = N_T$$

3.2 Χωρητικότητα όταν ο πομπός δεν έχει γνώση του καναλιού (No CSI at the transmitter)

Για να μπορεί ο πομπός να έχει γνώση της κατάστασης του καναλιού μετάδοσης απαιτούνται διαδικασίες εκτίμησής του. Έτσι είναι φυσικό στις περισσότερες περιπτώσεις το κανάλι και συνεπώς ο πίνακας \mathbf{H} που το περιγράφει να μην είναι γνωστός στην πλευρά εκπομπής και η ισχύς αποδίδεται όμοια σε όλες τις

κεραίες του πομπού. Η συνάρτηση αυτοσυσχέτισης του διανύσματος x με τα μεταδιδόμενα σύμβολα δίνεται $R_{xx} = I_{N_T}$. Η σχέση που περιγράφει τη χωρητικότητα αυτού του καναλιού γίνεται :

$$C = \log_2 \det \left(I_{N_R} + \frac{SNR}{N_T} H I_{N_T} H^* \right) \Rightarrow C = \log_2 \det \left(I_{N_R} + \frac{SNR}{N_T} H H^* \right)$$

Συνεχίζοντας τη σχέση αυτή και χρησιμοποιώντας τον SVD μετασχηματισμό του πίνακα H καταλήγουμε στη σχέση :

$$C = \sum_{i=1}^n \log_2 \left(1 + \frac{SNR}{N_T} \lambda_i \right) \frac{bps}{Hz}$$

όπου n είναι ο αριθμός των βαθμών ελευθερίας του συστήματος δηλαδή ο μέγιστος αριθμός ανεξάρτητων ροών δεδομένων που μπορούν να μεταδοθούν και λ_i οι ιδιοτιμές του πίνακα $H H^*$. Αποδεικνύεται ότι η μέγιστη χωρητικότητα επιτυγχάνεται όταν οι ιδιοτιμές αυτές είναι ίσες μεταξύ τους. Υποθέτοντας πως έχουμε ένα κανάλι

με συνολικό κέρδος $\|H\|^2 = \sum_{i=1}^n \lambda_i = \xi$ και το κανάλι είναι πλήρους τάξης με

$N_T=N_R=n$ τότε η χωρητικότητά του μεγιστοποιείται για τιμές $\lambda_i = \frac{\xi}{n}$ και γίνεται n φορές η χωρητικότητα των επιμέρους SISO καναλιών που υποστηρίζονται από το σύστημα.

$$C = n * \log_2 \left(1 + \frac{\xi * SNR}{n} \right) \frac{bps}{Hz}$$

3.3 Εργοδική Χωρητικότητα

Ένα κανάλι ασύρματης μετάδοσης δεν παραμένει σταθερό. Τα στοιχεία h_{ji} του πίνακα H που περιγράφει το MIMO κανάλι μεταβάλλονται κατά τη διάρκεια του χρόνου. Συνεπώς το κανάλι αυτό χαρακτηρίζεται ως μια εργοδική διαδικασία και η χωρητικότητα την οποία καλούμαστε να υπολογίσουμε, υπολογίζεται σαν η μέση τιμή της χωρητικότητας για πολλαπλές υλοποιήσεις ενός τυχαίου MIMO καναλιού.

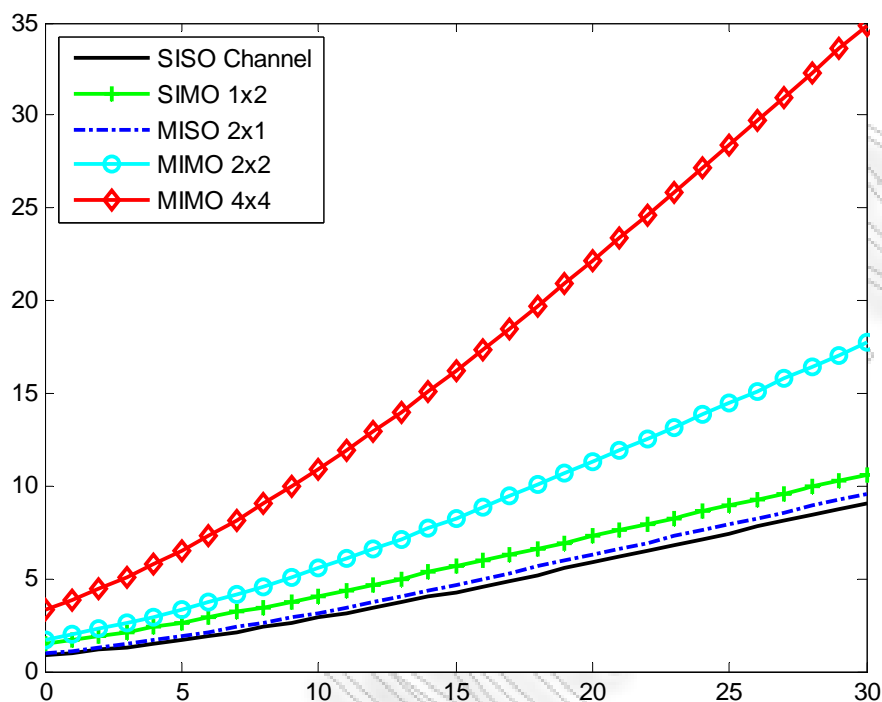
$$\bar{C} = E\{C(H)\} = E\left\{\log_2 \det\left(I_{N_R} + \frac{SNR}{N_T} HR_{xx}H^*\right)\right\} \frac{bps}{Hz}$$

Η χωρητικότητα του καναλιού όταν ο πομπός δεν γνωρίζει την κατάσταση του καναλιού (Open Loop Capacity) δίνεται από τη σχέση:

$$\bar{C}_{OL} = E\left\{\sum_{i=1}^n \log_2\left(1 + \frac{SNR}{N_T} \lambda_i\right)\right\} \frac{bps}{Hz}$$

Ενώ στην περίπτωση που ο πομπός έχει γνώση του πίνακα H (Closed Loop) μπορεί να κάνει χρήση του water pouring αλγορίθμου και αποδίδοντας κατάλληλες ισχύς στις διάφορες κεραιές διαμορφώνεται η χωρητικότητα:

$$\bar{C}_{CL} = E\left\{\sum_{i=1}^n \log_2\left(1 + \frac{SNR * \gamma_i}{N_T} \lambda_i\right)\right\} \frac{bps}{Hz}$$



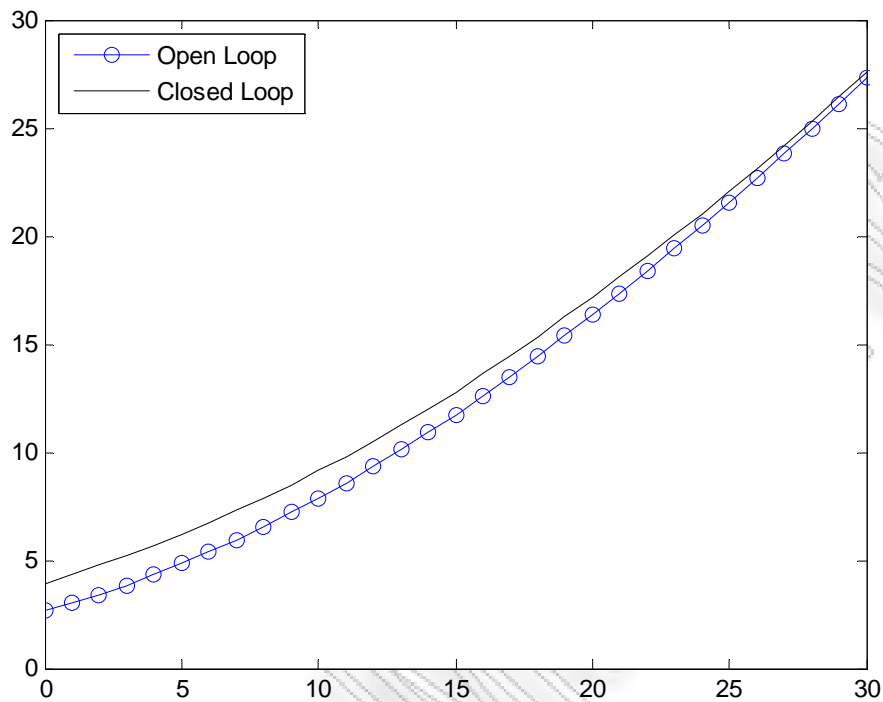
3.10 Χωρητικότητα MIMO καναλιού για συνδυασμούς κεραιών

Το γράφημα της εικόνας 3.3 έχει προκύψει από το πρόγραμμα του Matlab: MIMO_Channel_Ergodic_Capacity_vs_SNR.m. Στο γράφημα αυτό παρουσιάζεται η χωρητικότητα του MIMO καναλιού για διαφορετικούς συνδυασμούς, διατάξεις κεραιών. Εμφανίζεται η περίπτωση SISO καναλιού με μία κεραία εκπομπής και μία κεραία δέκτη, το SIMO κανάλι όπου η εκπομπή γίνεται με μια κεραία ενώ κατά τη λήψη χρησιμοποιούνται δύο, η αντίστροφη διάταξη MISO με δυο κεραίες εκπομπής και μια κεραία λήψης και δυο διατάξεις MIMO με δύο και τέσσερις κεραιές σε κάθε πλευρά. Το κανάλι που χρησιμοποιούμε περιγράφεται από τον πίνακα H σύμφωνα με το μοντέλο Rayleigh Fading Channel Model [5]. Τα στοιχεία του πίνακα δημιουργούνται με χρήση ενσωματωμένης συνάρτησης στο Matlab και οι γραμμές και οι στήλες του πίνακα είναι γραμμικώς ανεξάρτητες. Επίσης στη σχέση που χρησιμοποιείται για τον υπολογισμό της χωρητικότητας η τιμή του πίνακα R_{xx} έχει αντικατασταθεί από το μοναδιαίο πίνακα (Open Loop Capacity) καθώς η εργοδική

χωρητικότητα που υπολογίζουμε αναφέρεται σε συστήματα στα οποία η πλευρά του πομπού δεν έχει γνώση της κατάστασης του καναλιού (No CSI) . Στο παραπάνω διάγραμμα της εικόνας 3.3 παρατηρούμε πως το μεγαλύτερο όφελος στη χωρητικότητα του συστήματος εξαιτίας της χρήσης πολλαπλών κεραιών προκύπτει στις περιοχές υψηλού σηματοθορυβικού λόγου. Η εξίσωση υπολογισμού της χωρητικότητας του συστήματος είναι:

$$C = \begin{cases} \log_2 \frac{\det \left(I_{N_R} + \frac{SNR}{N_T} H H^* \right) bps}{Hz} & \text{για } N_R \leq N_T \\ \log_2 \frac{\det \left(I_{N_T} + \frac{SNR}{N_T} H^* H \right) bps}{Hz} & \text{για } N_R \geq N_T \end{cases}$$

Όπως παρουσιάστηκε στις προηγούμενες ενότητες όταν υπάρχει γνώση του καναλιού στην πλευρά του δέκτη υπάρχει η δυνατότητα να διαμοιραστεί κατάλληλα η ισχύς εκπομπής ανάμεσα στα στοιχεία της κεραίας ώστε να υπάρχουν καλύτερα αποτελέσματα. Το γράφημα της εικόνας 3.4 αποτυπώνει το πλεονέκτημα που έχει ένα τέτοιο σύστημα (Closed Loop) απέναντι σε ένα σύστημα όπου η ισχύς μοιράζεται όμοια σε όλες τις κεραιές εκπομπής (Open Loop).



3.11 Χωρητικότητα με και χωρίς CSI

Το διάγραμμα της εικόνας 3.4 είναι αποτέλεσμα του προγράμματος `OpenLoop_vs_ClosedLoop_Channel_Capacity.m` το οποίο υπολογίζει την εργοδική χωρητικότητα ενός MIMO συστήματος για την περίπτωση κατά την οποία η συνολική εκπεμπόμενη ισχύς μοιράζεται ανάμεσα σε τέσσερις κεραιές εκπομπής όμοια και για την περίπτωση κατά την οποία η ανάθεση ισχύος σε κάθε κεραιά γίνεται με τη χρήση του αλγορίθμου `water pouring`. Ο αλγόριθμος αυτός εκτελείται στο πρόγραμμα `Water_Pouring.m` ως συνάρτηση που καλείται από το κύριο πρόγραμμα. Όπως παρατηρούμε στο γράφημα υπάρχει σαφές όφελος με τη χρήση του αλγορίθμου `water pouring`, ωστόσο οι δύο γραφικές παραστάσεις συγκλίνουν όσο αυξάνεται η τιμή του σηματοθορυβικού λόγου.

3.4 Χωρική Συσχέτιση

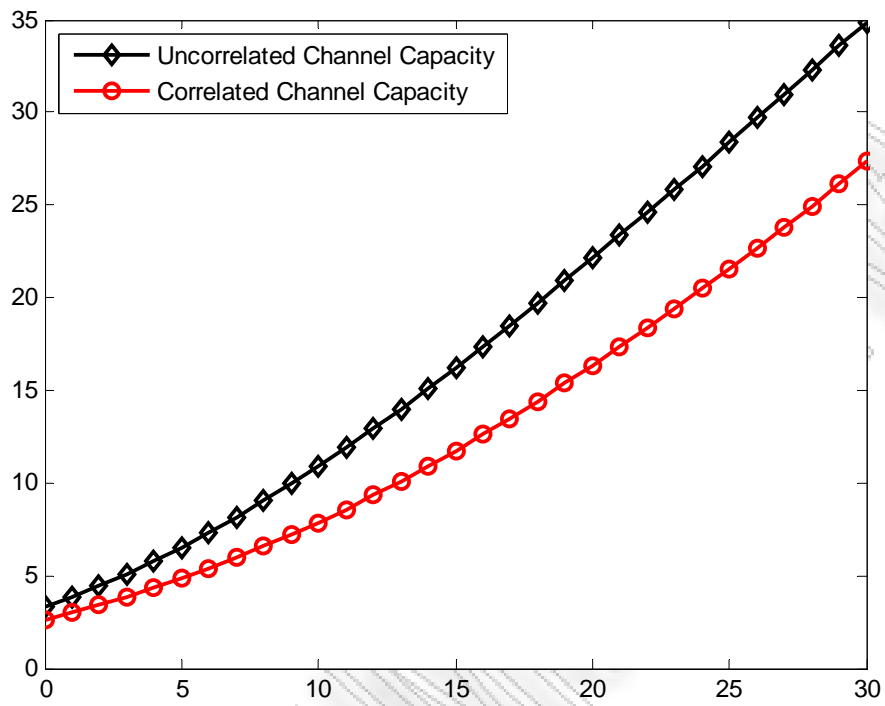
Έως αυτό το σημείο η μελέτη της χωρητικότητας αναφερόταν σε συστήματα ιδανικά σύμφωνα με τις προϋποθέσεις που παρουσιάστηκαν στο δεύτερο κεφάλαιο της εργασίας αυτής. Όταν πληρούνται αυτές οι προϋποθέσεις τότε προκύπτει ένα σύστημα που παρέχει τον μέγιστο αριθμό βαθμών ελευθερίας με χωρικά ασυσχέτιστες κεραίες εκπομπής και λήψης.

Είναι αυτονόητο πως ένα τέτοιο ιδανικό σύστημα είναι τουλάχιστον δύσκολο και σπάνιο να συναντηθεί σε πραγματικές συνθήκες μετάδοσης. Είναι πολύ πιθανό οι σκεδαστές που υπάρχουν στο περιβάλλον διάδοσης να μην επαρκούν και η διασπορά των γωνιών άφιξης των διαφορετικών σημάτων να μην επαρκεί ώστε να προκύψει πίνακας H που περιγράφει το κανάλι με γραμμικώς ανεξάρτητες γραμμές και στήλες. Στην περίπτωση που το κανάλι δεν είναι ιδανικό ο υπολογισμός του γίνεται με χρήση του μοντέλου γινομένου Kronecker (Kronecker product model) και εκφράζεται μέσω της εξίσωσης:

$$H = R_r^{\frac{1}{2}} H_w R_t^{\frac{1}{2}} \quad [5]$$

όπου R_r είναι οι συσχέτιση μεταξύ των στοιχείων της κεραίας δέκτη, δηλαδή η συσχέτιση μεταξύ των γραμμών του πίνακα H που περιγράφει το MIMO κανάλι, R_t η συσχέτιση μεταξύ των στοιχείων της κεραίας εκπομπής, δηλαδή μεταξύ των στηλών του πίνακα H και H_w είναι το ιδανικό κανάλι του οποίου οι γραμμές και οι στήλες είναι γραμμικώς ανεξάρτητες.

Στο γράφημα της εικόνας 3.5 που ακολουθεί αποτυπώνεται η υποβάθμιση της χωρητικότητας ενός MIMO συστήματος όταν υπάρχει τέτοια χωρική συσχέτιση. Το γράφημα της εικόνας παρουσιάζει τη χωρητικότητα ενός συστήματος με τέσσερις κεραίες εκπομπής και ισάριθμες κεραίες λήψης για τις δυο προαναφερθείσες περιπτώσεις.



3.12 Χωρητικότητα συσχετισμένου - ασυσχετίστου MIMO καναλιού

Κεφάλαιο 4

Δέκτες Χωρικής Πολυπλεξίας

Όπως αναφέρθηκε και στην αρχή της εργασίας, ο στόχος προς τον οποίο κινούνται κυρίως οι μελέτες στο χώρο των τηλεπικοινωνιών είναι η αύξηση της ταχύτητας μετάδοσης των δεδομένων, δηλαδή της χωρητικότητας των τηλεπικοινωνιακών καναλιών. Η MIMO τεχνολογία και συγκεκριμένα η τεχνική χωρικής πολυπλεξίας επιτρέπει αυτή την μεγάλη αύξηση στους ρυθμούς μετάδοσης εκμεταλλευόμενη το περιβάλλον που περιέχει πολλούς σκεδαστές και τις πολυδιαδρομικές συνιστώσες του σήματος. Η χωρική πολυπλεξία χρησιμοποιείται με δύο τρόπους. Η μία εκδοχή της χωρικής πολυπλεξίας καλείται SDMA (Space Division Multiple Access) πολλαπλή πρόσβαση διαχωρισμού χώρου. Χρησιμοποιείται για το διαμοιρασμό των πόρων που παρέχει το MIMO κανάλι ανάμεσα σε διαφορετικούς χρήστες. Η δεύτερη περίπτωση χωρικής πολυπλεξίας καλείται (Space Division Multiplexing) και είναι η περίπτωση με την οποία ασχολούμαστε στην εργασία αυτή. Μιλώντας για SDM αναφερόμαστε στην περίπτωση κατά την οποία διαφορετικά δεδομένα μεταδίδονται από τα διαφορετικά στοιχεία της κεραίας πομπού, η επικοινωνία ωστόσο πραγματοποιείται ανάμεσα σε δύο οντότητες (1 πομπός , 1 δέκτης) χωρίς να περιλαμβάνονται διαδικασίες πολλαπλής πρόσβασης. Η πλευρά του δέκτη καλείται στη συνέχεια να διαχωρίσει επιτυχώς τα σήματα που φθάνουν και να ανασυνθέσει το αρχικό σήμα. Στο κεφάλαιο αυτό παρουσιάζονται ορισμένοι δέκτες που κάνουν την προαναφερθείσα διαδικασία. Οι επιδόσεις τους αξιολογούνται μέσω γραφικών παραστάσεων που περιγράφουν τη συμπεριφορά των δεκτών σε ρυθμό σφαλμάτων σε σχέση με την τιμή του σηματοθορυβικού λόγου. Οι γραφικές παραστάσεις έχουν προκύψει από

προσομοιώσεις της λειτουργίας των δεκτών που έχουν υλοποιηθεί στο λογισμικό Matlab. Οι κώδικες όλων των προσομοιώσεων που χρησιμοποιούνται βρίσκονται στο παράρτημα της εργασίας. Πρώτα όμως παρουσιάζεται το μοντέλο μετάδοσης των δεδομένων που χρησιμοποιείται για της ανάγκες της προσομοίωσης.

4.1 Μοντέλο MIMO μετάδοσης.

Το σύστημα αποτελείται από N_T κεραιές εκπομπής και N_R κεραιές λήψης. Το περιβάλλον στο οποίο πραγματοποιείται η μετάδοση είναι πλούσιο σε σκεδαστές και η μετάδοση αναφέρεται στην επικοινωνία δυο οντοτήτων, δηλαδή δεν αναφερόμαστε σε πολλαπλή πρόσβαση χρηστών. Ο μιγαδικός πίνακας \mathbf{H} διαστάσεων $N_R \times N_T$ που περιγράφει την επίδραση του καναλιού στο σήμα είναι πλήρους τάξης. Η πλευρά του δέκτη έχει πλήρη γνώση του πίνακα αυτού (Channel State Information) ο οποίος είναι απαραίτητος για τον επιτυχή διαχωρισμό των σημάτων που προέρχονται από τα διαφορετικά στοιχεία του πομπού. Η σχέση που περιγράφει τη μετάδοση είναι $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$. Το διάνυσμα \mathbf{x} έχει διαστάσεις $N_T \times 1$ και είναι το διάνυσμα με τα σύμβολα που μεταδίδονται από τα διαφορετικά στοιχεία της κεραιάς του πομπού. Ο πίνακας \mathbf{H} όπως προαναφέρθηκε είναι διαστάσεων $N_R \times N_T$ του οποίου το κάθε στοιχείο h_{ij} περιγράφει την επίδραση του καναλιού από τον j πομπό στον i δέκτη. Τα διανύσματα \mathbf{n} και \mathbf{y} έχουν διαστάσεις $N_R \times 1$, το \mathbf{n} αποτελεί το θόρυβο ο οποίος είναι λευκός γκαουσιανός AWGN που προστίθεται στο κάθε στοιχείο δέκτη και το διάνυσμα \mathbf{y} είναι το τελικό διάνυσμα που εμφανίζεται στην πλευρά του δέκτη.

Η αξιολόγηση της επίδοσης των δεκτών που μελετάμε πραγματοποιείται μέσω προσομοιώσεων που δημιουργήθηκαν στο matlab. Οι προσομοιώσεις αυτές υπολογίζουν το ρυθμό σφαλμάτων κατά τη μετάδοση δεδομένων σε ένα MIMO σύστημα για διαφορετικές τιμές ενέργειας ανά δυαδικό σύμβολο προς φασματική πυκνότητα ισχύος θορύβου (E_b/N_0). Στο σημείο αυτό παρουσιάζονται οι αρχικές παράμετροι των προσομοιώσεων και η λειτουργία του κώδικα που προσομοιώνει την μετάδοση χωρίς την ειδική επεξεργασία που πραγματοποιείται από τον εκάστοτε τύπο δέκτη που χρησιμοποιείται.

Εφόσον αναφερόμαστε σε Spatial Multiplexing, κάθε χρονική στιγμή μεταδίδονται N_T διαφορετικά σύμβολα από τη συστοιχία κεραιών εκπομπής. Για να έχουμε αξιόπιστα στατιστικά αποτελέσματα στον κώδικα πραγματοποιούμε τη μετάδοση για πολλές τέτοιες χρονικές στιγμές οι οποίες καλούνται Timeslots. Συνεπώς τα σύμβολα προς μετάδοση δεν είναι ένα διάνυσμα αλλά ένας πίνακας στοιχείων x_{ij} με N_T σειρές και Timeslots στήλες :

$$\text{Symbols to be Transmitted} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1\text{Timeslots}} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N_T 1} & x_{N_T 2} & \dots & x_{N_T \text{Timeslots}} \end{bmatrix}$$

Τα σύμβολα αυτά προκύπτουν από τη διαμόρφωση μιας ακολουθίας δυαδικών ψηφίων. Στις προσομοιώσεις χρησιμοποιείται MQAM κωδικοποίηση και μετά τον ορισμό του πλήθους κεραιών εκπομπής και λήψης και τον καθορισμό του αριθμού των χρονοσχημάτων που πραγματοποιείται η μετάδοση, ανατίθεται τιμή στην μεταβλητή M . Η τιμή M αποτελεί τον αριθμό των διαφορετικών συμβόλων που μπορούν να μεταδοθούν και καθορίζει τον αριθμό των δυαδικών ψηφίων από το

οποία προκύπτει το κάθε σύμβολο x_{ij} σύμφωνα με τη σχέση $b = \log_2 M \frac{\text{bits}}{\text{symbols}}$.

Έχοντας τις τιμές b, N_T και Timeslots χρησιμοποιώντας την εντολή:

```
bitstream=randint(Timeslots*b*NT,1)
```

δημιουργείται η τυχαία ακολουθία δυαδικών ψηφίων τα οποία είναι ακριβώς όσα χρειάζονται για τη δημιουργία του πίνακα των προς μετάδοση συμβόλων. Στη συνέχεια τα δυαδικά ψηφία ομαδοποιούνται σε ομάδες των b bits μέσω των εντολών:

```
S=reshape(bitstream,b,length(bitstream)/b);
S=S.';
```

Ο πίνακας S αποτελείται από $N_T * Timeslots$ ομάδες των b bits δηλαδή το S είναι διάνυσμα $N_T * Timeslots$ συμβόλων. Τα σύμβολα διαμορφώνονται με τη χρήση των εντολών του matlab :

```
xsym = bi2de(S,'left-msb');  
xsym = modulate(modem.qammod(M),xsym);
```

Στην περίπτωση της MQAM κωδικοποίησης ανάλογα με την τιμή M διαφοροποιείται το εύρος των τιμών που παίρνουν τα διαμορφωμένα σύμβολα. Για το λόγο αυτό πρέπει να συμπεριληφθεί ένας παράγοντας που θα πραγματοποιεί κανονικοποίηση της μέσης ισχύος των μεταδιδόμενων συμβόλων.

```
QAM_nums=[0:M-1];  
QAM_syms=modulate(modem.qammod(M),QAM_nums);  
scale = modnorm(QAM_syms,'avpow',1);
```

Έχοντας υπολογίσει τον παράγοντα κανονικοποίησης της ισχύος scale πολλαπλασιάζεται με την ακολουθία συμβόλων xsym έτσι ώστε η μέση ισχύς των συμβόλων να είναι 1. Στη συνέχεια το διάνυσμα των συμβόλων μετατρέπεται στον πίνακα των προς μετάδοση συμβόλων με N_T σειρές και Timeslots στήλες.

```
xsym=scale*xsym;  
x=reshape(xsym,NT,Timeslots);
```

Έχουμε υποθέσει πως η μετάδοση των συμβόλων πραγματοποιείται μέσω διαύλου Rayleigh επίπεδων διαλείψεων (flat fading). Υποθέτουμε συνεπώς πως η μετάδοση πραγματοποιείται σε ιδανικό για τα MIMO συστήματα περιβάλλον πλούσιο σε σκεδαστές και συνεπώς οι πολυδιαδρομικές συνιστώσες του σήματος να μπορούν να θεωρηθούν ανεξάρτητες. Ο δίαυλος που προκύπτει ονομάζεται ανεξάρτητα πανομοιότυπα κατανεμημένος (Independent Identically Distributed, IID) δίαυλος

Rayleigh. Για να δημιουργηθεί ο διάυλος αυτός στις προσομοιώσεις, τα στοιχεία του πίνακα H είναι μιγαδικές τυχαίες μεταβλητές, κυκλικά συμμετρικές και ομοιόμορφα κατανομημένες που ακολουθούν κατανομή Gauss. Το κανάλι της μετάδοσης δημιουργείται μέσω της εντολής:

$$H = (\text{randn}(NR, NT, \text{Timeslots}) + 1i * \text{randn}(NR, NT, \text{Timeslots})) / \text{sqrt}(2);$$

Ο πίνακας H που δημιουργούμε έχει τυχαίες μεταβλητές Rayleigh με μοναδιαία μέση ισχύ. Δημιουργούμε πίνακα τριών διαστάσεων έτσι ώστε για κάθε χρονική στιγμή που μεταδίδεται κάποια στήλη του πίνακα συμβόλων να επιλέγεται το αντίστοιχο στιγμιότυπο δύο διαστάσεων του πίνακα H με τον οποίο πολλαπλασιάζεται πλέον το διάνυσμα των μεταδιδόμενων συμβόλων.

Το σήμα διέρχεται από διάυλο προσθετικού λευκού γκαουσιανού θορύβου. Σε κάθε στοιχείο της κεραίας δέκτη πρέπει να προστεθεί η τιμή αυτού του θορύβου. Η σχέση που περιγράφει το σηματοθορυβικό λόγο συναρτήσει της ενέργειας δυαδικού ψηφίου ανά φασματικής πυκνότητας ισχύος θορύβου είναι:

$$SNR = \frac{E_s}{N_o} = \frac{E_b}{N_o} \log_{\mathbf{2}} M$$

Όπου E_s είναι η ενέργεια μεταδιδόμενου συμβόλου δηλαδή είναι ίση με το γινόμενο E_b επί τον αριθμό των δυαδικών ψηφίων που κωδικοποιούνται ανά σύμβολο. Η σχέση αυτή σε dB μετατρέπεται:

$$SNR (dB) = \frac{E_b}{N_o} (dB) + 10 \log_{\mathbf{10}} b$$

Χρησιμοποιώντας τη συνάρτηση `randn()` όπως και για τη δημιουργία του διαύλου Rayleigh δημιουργούμε μιγαδικές μεταβλητές θορύβου με μέση ισχύ . Έχοντας υποθέσει ότι το σήμα που μεταδίδεται έχει μέση ισχύ ίση με 1 μπορούμε να ρυθμίσουμε την ισχύ του θορύβου στο δέκτη υπολογίζοντας το σ^2 σύμφωνα με τη σχέση:

$$SNR = \frac{1}{2\sigma^2} \Rightarrow \sigma = \frac{1}{\sqrt{2SNR}}$$

Στο ακόλουθο κομμάτι του κώδικα παράγεται ο πίνακας N_R γραμμών και Timeslots στηλών που περιέχει τις μεταβλητές που προστίθενται σε κάθε κεραία δέκτη για κάθε χρονική στιγμή μετάδοσης συμβόλων.

```
No=(randn(NR,Timeslots) + 1i*randn(NR,Timeslots));
SNR = 10^(SNR_dB(i)/10);
sigma = sqrt(1/(2*SNR));
noise = sigma * No;
```

Η μετάδοση αποτελεί μια επαναληπτική λειτουργία για όλες τις χρονικές στιγμές που έχουμε υποθέσει, εφόσον πλέον έχουν δημιουργηθεί οι πίνακες συμβόλων, θορύβου και καναλιού.

```
for T=1:Timeslots
Y(:,T)=H(:,T)*x(:,T);
Ynoisy(:,T)=Y(:,T)+noise(:,T);
end
```

Για κάθε χρονικό στιγμιότυπο, το αντίστοιχο διάνυσμα συμβόλων πολλαπλασιάζεται με τον κατάλληλο πίνακα, προστίθεται η αντίστοιχη τιμή θορύβου για κάθε στοιχείο της κεραίας δέκτη και προκύπτει ο πίνακας με τα σύμβολα που λαμβάνει ο δέκτης για όλες τις χρονικές στιγμές υλοποίησης της μετάδοσης. Ο πίνακας αυτός των ληφθέντων σημάτων έχει ίδιες διαστάσεις με τον πίνακα των μεταβλητών του θορύβου. Ο πίνακας αυτός εισέρχεται στον εκάστοτε δέκτη και μετά από κατάλληλη επεξεργασία προκύπτει ο πίνακας ίδιων διαστάσεων που περιλαμβάνει τα σύμβολα τα οποία εκτιμά ο δέκτης ότι απέστειλε ο πομπός. Για να υπολογιστεί ο ρυθμός σφαλμάτων της μετάδοσης τα σύμβολα αυτά πρέπει να μετατραπούν σε μια ακολουθία δυαδικών ψηφίων ώστε να συγκριθούν με την αρχική ακολουθία

bitstream. Αντιστρέφοντας τη διαδικασία που οδήγησε στη μετατροπή της ακολουθίας bitstream στον πίνακα των προς μετάδοση συμβόλων προκύπτει το τελικό διάνυσμα με τα δυαδικά ψηφία που έχει λάβει ο δέκτης. Στο παρακάτω τμήμα του κώδικα από την προσομοίωση του δέκτη Zero Forcing πραγματοποιείται αυτή η διαδικασία που περιγράψαμε η οποία είναι η ίδια που χρησιμοποιήθηκε για όλες τις προσομοιώσεις.

```
RecSymbols_zf=reshape(Yest_zf,prod(size(Yest_zf)),1);
RecSymbols_zf=(1/scale)*RecSymbols_zf;
Recy_zf = demodulate(modem.qamdemod(M),RecSymbols_zf);
Recy_zf= de2bi(Recy_zf,'left-msb');
Recy_zf=Recy_zf.';
Rbitstream_zf=reshape(Recy_zf,prod(size(Recy_zf)),1);
```

Τέλος έχοντας πλέον την ακολουθία των δυαδικών ψηφίων που έχει προκύψει από την επεξεργασία του σήματος του εκάστοτε δέκτη, η εντολή του matlab :

```
[NUM_ERRORS_zf(i) BER_zf(i)]=biterr(bitstream,Rbitstream_zf)
```

Πραγματοποιεί τη σύγκριση ψηφίο προς ψηφίο των δύο ακολουθιών και δίνει τον αριθμό των λαθασμένων bit που έχει αποκωδικοποιήσει ο δέκτης. Στις επόμενες ενότητες παρουσιάζεται η λειτουργία των δεκτών ξεχωριστά, καθώς και μία σύγκριση μεταξύ τους ως προς τις αποδόσεις τους.

4.2 Γραμμικοί Δέκτες (Linear Receivers)

4.2.1 Zero Forcing Receiver.

Η πρώτη κατά σειρά περίπτωση δέκτη που εξετάζουμε είναι αυτή του δέκτη Zero Forcing ή αλλιώς interference nulling receiver [7][8]. Σκοπός της διαδικασίας αυτής είναι να μηδενίσει την επίδραση της κάθε ανεξάρτητης ροής δεδομένων προς την άλλη. Δηλαδή να αντιστρέψει την επίδραση που έχει το κανάλι στο αρχικό

διάνυσμα του σήματος \mathbf{x} . Ο δέκτης έχοντας πλήρη γνώση του καναλιού μέσω του οποίου μεταδίδεται το σήμα αναζητά έναν μιγαδικό πίνακα \mathbf{W} τέτοιο ώστε $\mathbf{W}\mathbf{H} = \mathbf{I}$. Ο κατάλληλος πίνακας $\mathbf{W} = \mathbf{H}^+$ που ικανοποιεί την παραπάνω συνθήκη είναι ο Moore – Penrose pseudo-inverse του πίνακα \mathbf{H} ο οποίος προκύπτει μέσω της εξίσωσης :

$$\mathbf{W} = \mathbf{H}^+ = (\mathbf{H}^* \mathbf{H})^{-1} \mathbf{H}^*$$

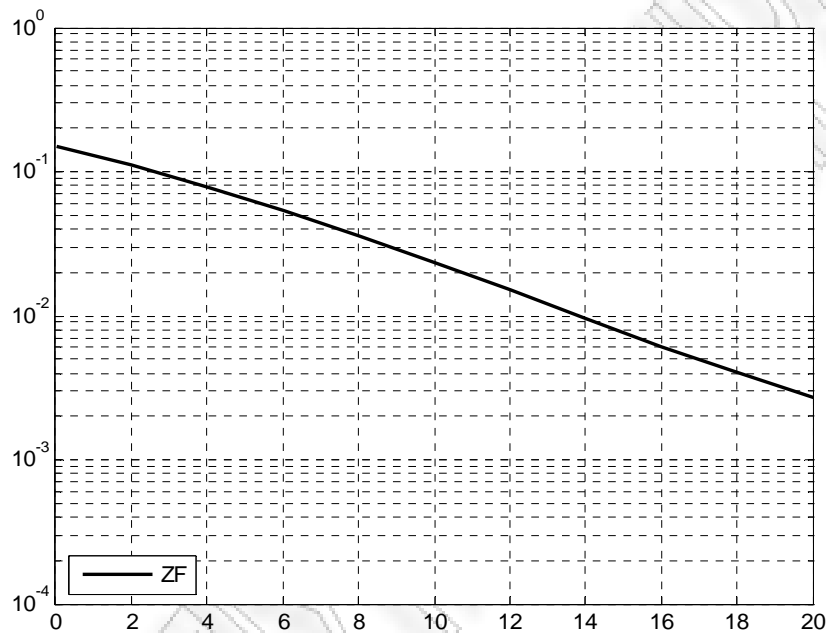
Για να υπάρχει αυτός ο πίνακας και να χρησιμοποιηθεί για την αντιστροφή της επίδρασης του καναλιού στο σήμα είναι προϋπόθεση ο αριθμός των κεραιών του πομπού να μην υπερβαίνει τον αριθμό αυτών στην πλευρά του δέκτη $N_T \leq N_R$. Ο πίνακας \mathbf{W} που προκύπτει έχει διαστάσεις $N_T \times N_R$. Τα αρχικά N_T σύμβολα που μεταδόθηκαν προκύπτουν από τον πολλαπλασιασμό του ληφθέντος σήματος \mathbf{y} με τον μιγαδικό πίνακα \mathbf{W} . Το αποτέλεσμα της πράξης αυτής θα είναι :

$$\mathbf{W}\mathbf{y} = \mathbf{W}\mathbf{H}\mathbf{x} + \mathbf{W}\mathbf{n} \Rightarrow \hat{\mathbf{x}} = \mathbf{x} + \mathbf{W}\mathbf{n}$$

Κατόπιν αυτής της επεξεργασίας, τα σύμβολα $\hat{x}_i = W_i y$ για $i = 1, 2, \dots, N_T$ αποτελούν τα σύμβολα που εκτιμά ο δέκτης ότι έστειλε ο πομπός όπου W_i αντιστοιχεί στην i -οστή σειρά του πίνακα. Ωστόσο κάθε σύμβολο \hat{x}_i που προκύπτει περιλαμβάνει εκτός από το μεταδοθέν x και έναν όρο $W\mathbf{n}$ ο οποίος αντιστοιχεί σε μια τιμή θορύβου \hat{n}_{ZF} που ενισχύεται μετά την εφαρμογή του φίλτρου W .

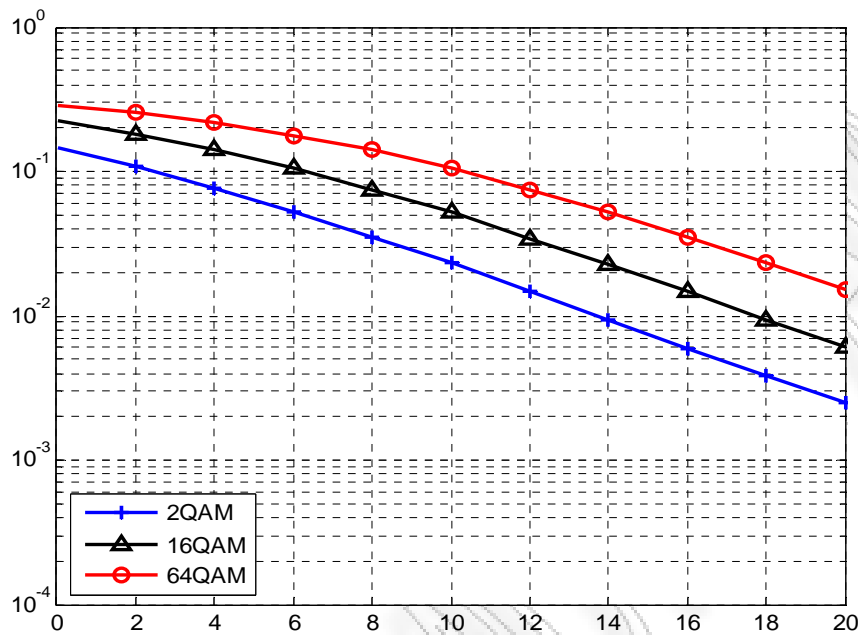
Στο πρόγραμμα του Matlab *mimo_ZF_RECEIVER.m*, προσομοιώνεται η μετάδοση δεδομένων σε ένα MIMO σύστημα με 2 κεραιές εκπομπής και 2 κεραιές λήψης. Η κωδικοποίηση των συμβόλων είναι 2QAM δηλαδή 1 bit per symbol. Το κανάλι μοντελοποιείται σύμφωνα με Rayleigh Fading Channel Model [5] και ο πίνακας \mathbf{H} που προκύπτει σε κάθε κύκλο επανάληψης της μετάδοσης δεδομένων είναι πλήρους τάξης. Στο παρακάτω γράφημα της εικόνας 4.1 παρουσιάζονται τα

αποτελέσματα της προσομοίωσης και αποτυπώνεται ο ρυθμός σφαλμάτων για διαφορετικές τιμές σηματοθρομβικού λόγου.



4.13 2x2 MIMO 2QAM Zero Forcing receiver performance

Το επόμενο γράφημα της εικόνας 4.2 είναι το αποτέλεσμα της προσομοίωσης της μετάδοσης για τρεις διαφορετικές περιπτώσεις διαμόρφωσης. Το πρόγραμμα MQAM_ZF_RECEIVER.m προσομοιώνει τη λειτουργία του δέκτη Zero Forcing για τις περιπτώσεις κατά τις οποίες χρησιμοποιείται 2, 16 και 64 -QAM διαμόρφωση. Στις περιπτώσεις αυτές κωδικοποιούνται 1, 4 και 6 δυαδικά ψηφία ανά μεταδιδόμενο σύμβολο.

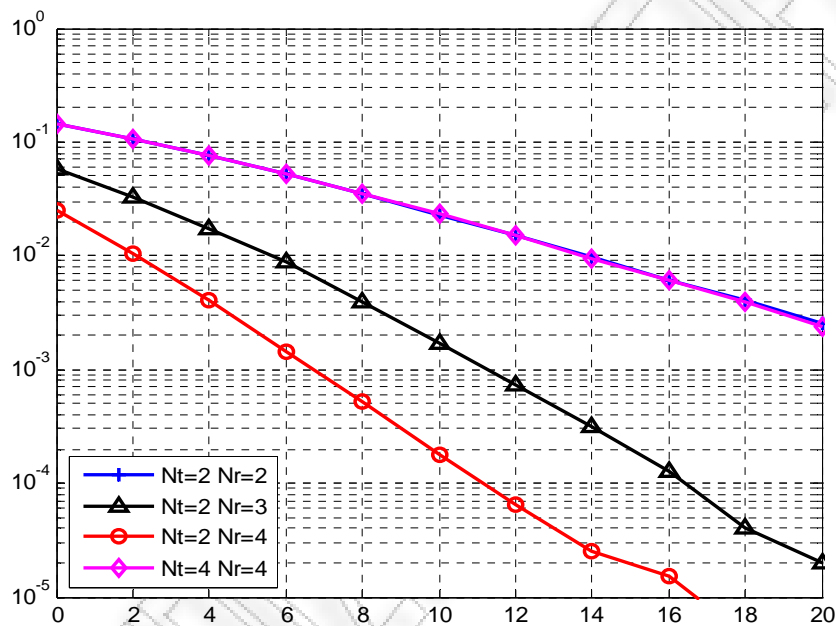


4.14 Απόδοση ZF Δέκτη για διαφορετικό σχήμα διαμόρφωσης

Παρατηρείται μια σημαντική υποβάθμιση της απόδοσης του δέκτη όσο αυξάνεται το σχήμα κωδικοποίησης και ο αριθμός των κωδικοποιημένων δυαδικών ψηφίων ανά σύμβολο μετάδοσης. Αυτό συμβαίνει γιατί στον κώδικα προσομοίωσης έχει συμπεριληφθεί ο παράγοντας scale ο οποίος κρατάει τη μέση ισχύ των μεταδιδόμενων συμβόλων στο 1. Συνεπώς όσο αυξάνεται το σχήμα κωδικοποίησης οι αποστάσεις μεταξύ των διαφορετικών συμβόλων μικραίνει με αποτέλεσμα ο θόρυβος ο οποίος έχει την ίδια ισχύ για όλες τις περιπτώσεις μετάδοσης να έχει χειρότερη επίδραση για κάθε επιπλέον δυαδικό ψηφίο ανά σύμβολο.

Το επόμενο γράφημα της εικόνας 4.3 έχει προκύψει από την εκτέλεση του προγράμματος ZF_RECEIVER_antenna_configuration.m το οποίο προσομοιώνει τη μετάδοση για τέσσερις διαφορετικές διατάξεις κεραιών. Για σύστημα με δύο κεραιές εκπομπής και δύο, τρεις και τέσσερις κεραιές λήψης και για συμμετρικό σύστημα τεσσάρων κεραιών εκπομπής και λήψης. Όσο αυξάνεται ο αριθμός των

κεραιών λήψης σε σχέση με αυτόν των κεραιών εκπομπής μειώνεται ο ρυθμός σφαλμάτων ενώ για το συμμετρικό σύστημα 4x4 η απόδοση του δέκτη είναι η ίδια με αυτήν του συστήματος 2x2. Τα αποτελέσματα επιβεβαιώνουν τη θεωρία που αναφέρει πως ο Zero Forcing Receiver επιτυγχάνει τάξη διαφορισμού ίση με $N_R - N_T + 1$ [7], όπου τάξη διαφορισμού i σημαίνει ότι για κάθε 10dB αύξηση σηματοθορυβικού λόγου επιτυγχάνεται μείωση του ρυθμού σφαλμάτων ίση με 10^{-i} .



4.15 Σύγκριση απόδοσης ZF για διαφορετικές διατάξεις κεραιών

4.2.2 MMSE Receiver.

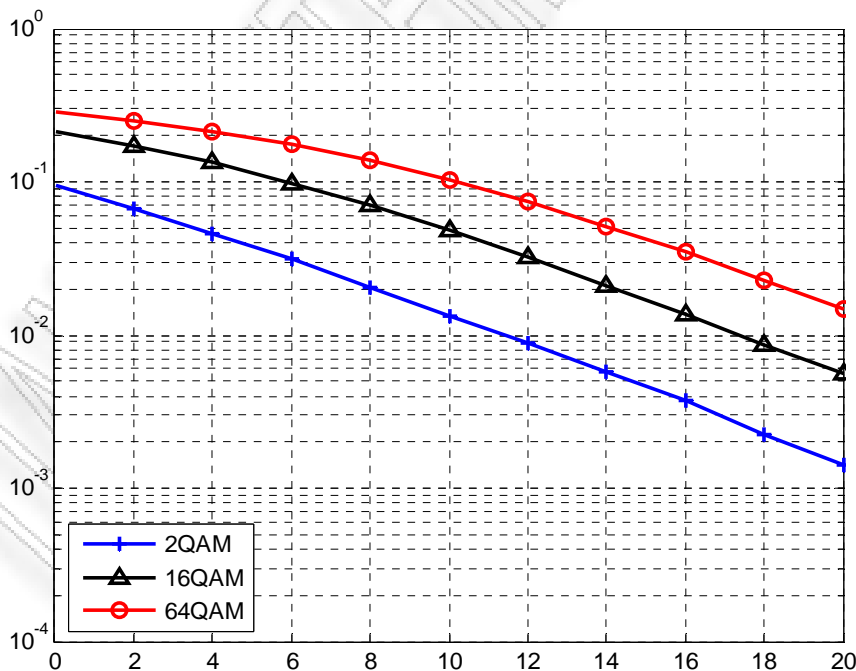
Παρόμοια με του Zero Forcing είναι η λειτουργία του MMSE (Minimum Mean Square Error) Receiver. [9] Στόχος της λειτουργίας του δέκτη αυτού είναι να ελαχιστοποιήσει το τετραγωνικό σφάλμα του σήματος \hat{x} που εκτιμάται πως έχει μεταδοθεί, δηλαδή να ελαχιστοποιήσει την τιμή:

$$e^2 = E[(\hat{x} - x)^* (\hat{x} - x)] = E[(\hat{x} - Wy)^* (\hat{x} - Wy)]$$

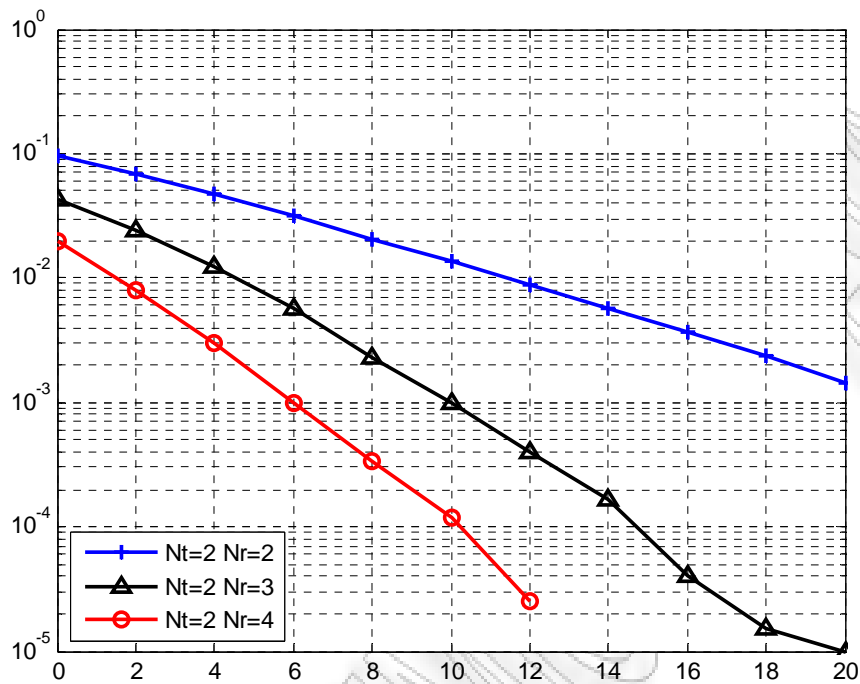
Ο πίνακας που δίνει λύση σε αυτό το πρόβλημα της ελαχιστοποίησης τετραγωνικού σφάλματος δίνεται από την εξίσωση :

$$\mathbf{W} = (\mathbf{H}^* \mathbf{H} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{H}^*$$

Ο δέκτης αυτός έχει στόχο να μειώσει την ενίσχυση του θορύβου που παρατηρείται στον Zero Forcing receiver κατά την εφαρμογή του φίλτρου. Το νέο φίλτρο \mathbf{W} που χρησιμοποιείται στον MMSE δέκτη χρησιμοποιεί την τιμή της μεταβλητότητας του θορύβου σ_n^2 . Όπως φαίνεται από την εξίσωση όταν η τιμή αυτή γίνει πολύ μικρή τότε το φίλτρο του MMSE δέκτη γίνεται ίδιο με αυτό του Zero Forcing. Ο δέκτης αυτός συμπεριφέρεται με τον ίδιο τρόπο, δηλαδή υποβαθμίζεται η απόδοσή του για κάθε αύξηση του σχήματος κωδικοποίησης και επιτυγχάνει τάξη διαφορισμού ίση με $N_R - N_T + 1$. Τα παρακάτω διαγράμματα στις εικόνες 4.4 και 4.5 που έχουν προκύψει από τα προγράμματα MQAM_MMSE_RECEIVER.m και MMSE_RECEIVER_antenna_configuration.m αποτυπώνουν αυτή τη συμπεριφορά.

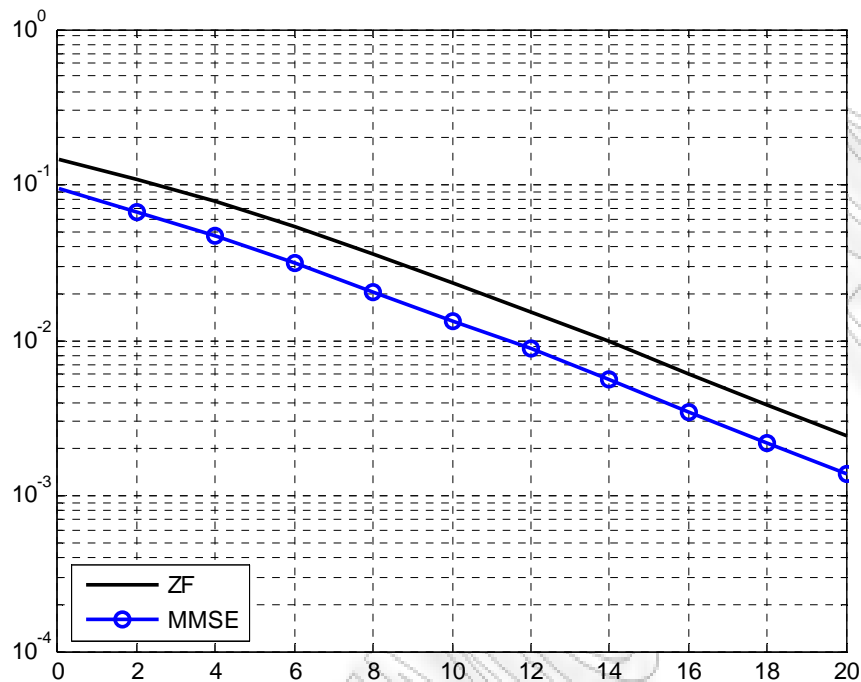


4. 16 Απόδοση MMSE Δέκτη για διαφορετικό σχήμα διαμόρφωσης



4.17 Σύγκριση απόδοσης MMSE για διαφορετικές διατάξεις κεραιών

Η προσομοίωση της λειτουργίας των δεκτών ZF και MMSE με το πρόγραμμα ZF_vs_MMSE.m δείχνει ότι η απόδοση του δέκτη MMSE εκφρασμένη σε ρυθμό σφαλμάτων σε σχέση με το σηματοθρομβικό λόγο είναι ελαφρώς καλύτερη από αυτή του Zero Forcing. Τα αποτελέσματα της προσομοίωσης παρουσιάζονται στο διάγραμμα της εικόνας 4.6 και η υπεροχή του MMSE δέκτη οφείλεται στην ενίσχυση του θορύβου που πραγματοποιείται με τη χρήση του φίλτρου \mathbf{W}_{ZF} (Zero Forcing).



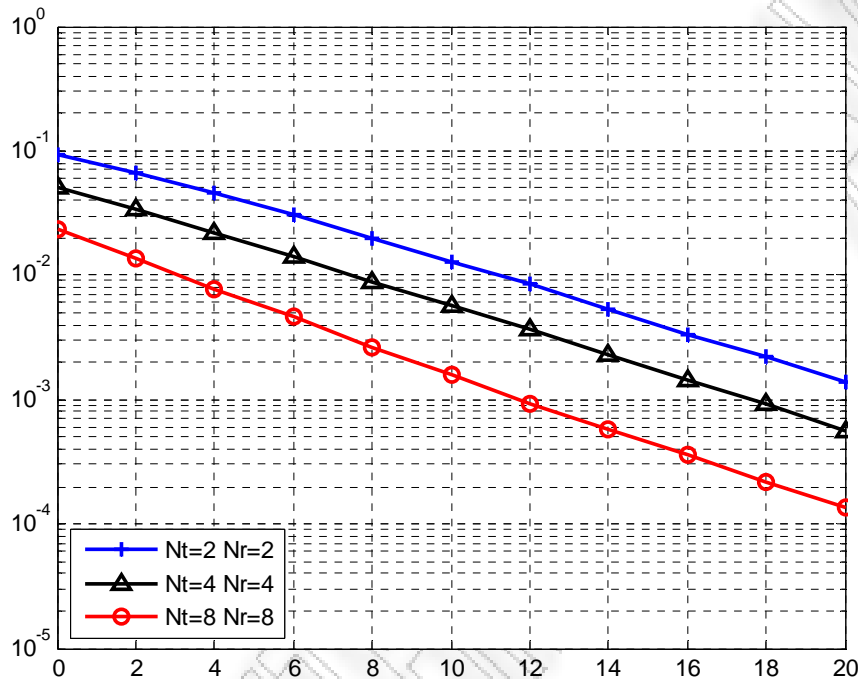
4.18 MMSE versus Zero Forcing Receiver

Η ισχύς του θορύβου για τους δέκτες που μελετάμε αποδεικνύεται πως είναι:

$$E\{\|r_{MMSE}\|^2\} = \frac{\sigma_n^2 \lambda_{\min}}{(\sigma_n^2 + \lambda_{\min})^2} \quad \text{και} \quad E\{\|r_{ZF}\|^2\} = \frac{\sigma_n^2}{\lambda_{\min}}$$

Όπου λ_{\min} είναι η ελάχιστη ιδιοτιμή του πίνακα $\mathbf{H}\mathbf{H}^*$ που περιγράφει τον διάυλο μεταφοράς. Όταν η μεταβλητότητα του θορύβου γίνεται πολύ μικρή, τότε η ενίσχυση του θορύβου είναι η ίδια και για τους δύο τύπους δεκτών. Όσο μικραίνει η ελάχιστη ιδιοτιμή λ_{\min} τότε η σχέση που περιγράφει την ισχύ του θορύβου για τον MMSE δέκτη δείχνει πως η ενίσχυση του θορύβου γίνεται μικρότερη. Η ελάχιστη ιδιοτιμή μικραίνει όσο ο πίνακας \mathbf{H} που περιγράφει το διάυλο παίρνει μεγαλύτερες διαστάσεις, γεγονός που εξηγεί και την αύξηση της επίδοσης του δέκτη όσο οι

κεραίες που συμμετέχουν στην μετάδοση των δεδομένων γίνονται περισσότερες όπως φαίνεται και από το γράφημα της εικόνας 4.7.



4.19 Σύγκριση απόδοσης MMSE για συμμετρικές διατάξεις διαφορετικού πλήθους κεραιών

4.3 Successive Interference Cancellation Receivers

Οι προηγούμενοι δέκτες που μελετήσαμε είναι ευρέως διαδεδομένοι στις ψηφιακές επικοινωνίες κυρίως λόγω της απλότητάς τους. Ωστόσο η απόδοσή τους είναι αρκετά μικρές. Μια βελτίωση της απόδοσής τους χωρίς πολύ μεγάλη αύξηση της πολυπλοκότητάς προσφέρει το μοντέλο μετάδοσης V-BLAST (Vertical Bell Laboratories Layered Space - Time) [8][10–13]. Πειραματική χρήση του συγκεκριμένου μοντέλου σε εσωτερικό περιβάλλον έχει δείξει ότι επιτυγχάνονται ταχύτητες 20-40 bps/Hz για τιμές σηματοθορυβικού λόγου SNR = 24 – 34 dB.

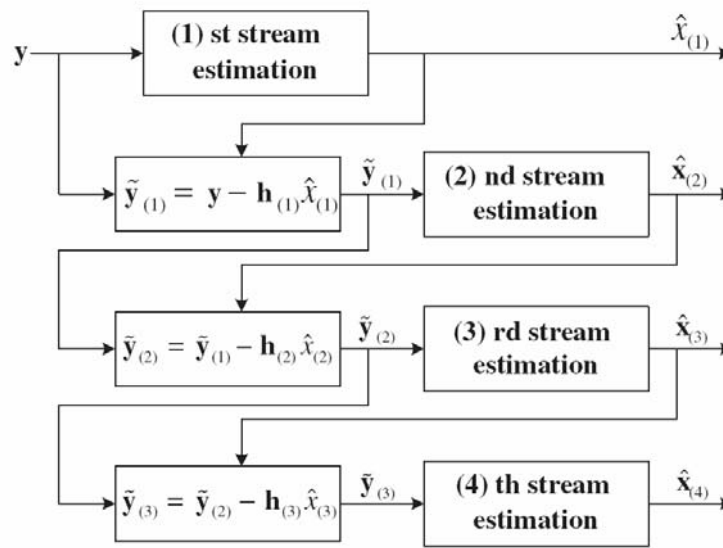
Στην εικόνα 4.8 παρουσιάζεται η λειτουργία των δεκτών που χρησιμοποιούν διατεταγμένη διαδοχική ακύρωση παρεμβολής (Ordered Successive Interference Cancellation) για ένα σύστημα στο οποίο πραγματοποιείται μετάδοση τεσσάρων ανεξάρτητων ροών δεδομένων από τις ισάριθμες κεραιές του πομπού. Η λειτουργία του συγκεκριμένου τύπου δέκτη περιγράφεται ως λειτουργία τεσσάρων γραμμικών δεκτών σε παράθεση οι οποίοι χρησιμοποιούν είτε φίλτρο Zero Forcing είτε MMSE. Το σήμα y που λαμβάνεται στην πλευρά του δέκτη καλείται να περάσει από το σύστημα τόσες φορές όσες και τα ανεξάρτητα σύμβολα που έχουν μεταδοθεί. Σε κάθε κύκλο λειτουργίας ανιχνεύεται ένα από τα σύμβολα που έχουν μεταδοθεί ταυτόχρονα. Η ανίχνευση γίνεται με απλό πολλαπλασιασμό με το φίλτρο W που έχει επιλεγεί να χρησιμοποιηθεί:

$$\hat{x}_i = W_i y$$

Όπου W_i είναι η γραμμή i του πίνακα $W = (H^* H)^{-1} H^*$ (ZF) όταν επιλέγεται η χρησιμοποίηση του φίλτρου Zero Forcing ή $W = (H^* H + \sigma_n^2 I)^{-1} H^*$ (MMSE) όταν η αποκωδικοποίηση γίνεται με τη χρήση του φίλτρου ελαχιστοποίησης τετραγωνικού σφάλματος (MMSE). Στο επόμενο στάδιο αφαιρείται η επίδραση που είχε η μετάδοση του συμβόλου αυτού από το συνολικό ληφθέν σήμα y . Το νέο σήμα $y^{(1)}$ που εισέρχεται στο δέκτη προκύπτει από την εξίσωση:

$$y^{(1)} = y - H_i \hat{x}_i$$

Με αυτόν τον τρόπο το σήμα $y^{(1)}$ που εισέρχεται στο δέκτη στον επόμενο κύκλο λειτουργίας του είναι απαλλαγμένο από την παρεμβολή που προκαλεί η μετάδοση του συμβόλου \hat{x}_i .



4.20 Δέκτης Διαδοχικής Ακύρωσης Παρεμβολής [5]

Για να είναι επιτυχής η αφαίρεση της παρεμβολής που προκαλεί το εκάστοτε σύμβολο που ανιχνεύεται από το συνολικό σήμα που εισέρχεται στον δέκτη κατά τον επόμενο κύκλο λειτουργίας του είναι προϋπόθεση το σύμβολο που ανιχνεύθηκε κατά τον προηγούμενο κύκλο να είναι σωστό. Αν υποθέσουμε ότι σε κάθε κύκλο i της λειτουργίας του αλγορίθμου ανιχνεύεται το αντίστοιχο σύμβολο του διανύσματος συμβόλων που μεταδόθηκε τότε η τάξη διαφορισμού του συγκεκριμένου συμβόλου είναι $N_R - N_T + i$ δηλαδή εξαρτάται από το ποιο κατά σειρά σύμβολο ανιχνεύεται. Αυτό συμβαίνει γιατί κάθε φορά που αφαιρείται η επίδραση του προηγούμενου συμβόλου από το ληφθέν σήμα y δημιουργείται ένα νέο σύστημα που περιγράφει τη μετάδοση του οποίου ο πίνακας \mathbf{H} έχει διαστάσεις $N_R \times N_T - i$. Οι διαστάσεις αυτές περιγράφουν ένα σύστημα με περισσότερες κεραιές λήψης από κεραιές εκπομπής και συνεπώς για κάθε επόμενο σύμβολο που αποκωδικοποιείται αυξάνεται η τάξη διαφορισμού. Στην περίπτωση κατά την οποία το σύμβολο που αποκωδικοποιήθηκε κατά τον προηγούμενο κύκλο του αλγορίθμου, αποκωδικοποιήθηκε εσφαλμένα το σφάλμα αυτό θα επηρεάσει και την αποκωδικοποίηση των υπόλοιπων συμβόλων που μεταδόθηκαν. Η αφαίρεση του λάθους συμβόλου περνάει στο νέο διάνυσμα \mathbf{y}^{new} που

εισέρχεται στον δέκτη κατά τον επόμενο κύκλο λειτουργίας. Για το λόγο αυτό είναι πολύ σημαντική η σειρά με την οποία ανιχνεύονται τα ανεξάρτητα σύμβολα και ο στόχος είναι σε κάθε κύκλο λειτουργίας να επιλέγεται το σύμβολο με την περισσότερη πιθανότητα να αποκωδικοποιηθεί σωστά λαμβάνοντας υπόψη την τιμή του σηματοθορυβικού λόγου και την τιμή της ισχύος σήματος προς θόρυβο και παρεμβολή.

Για τον Zero Forcing δέκτη η βέλτιστη επιλογή του συμβόλου που πρέπει να ανιχνευθεί με τη μεγαλύτερη προτεραιότητα γίνεται με βάση τον επιμέρους σηματοθορυβικό λόγο του συμβόλου ο οποίος είναι αντιστρόφως ανάλογος της ενίσχυσης του θορύβου που προκύπτει από τον πολλαπλασιασμό του ληφθέντος σήματος με το φίλτρο \mathbf{W} . Συνεπώς η επιλογή του κλάδου προς αποκωδικοποίηση γίνεται σύμφωνα με το ελάχιστο μέτρο του διανύσματος \mathbf{W}_i με το οποίο πολλαπλασιάζεται το σήμα. Μετά το σχηματισμό του φίλτρου μηδενισμού της παρεμβολής υπολογίζεται η νόρμα της κάθε γραμμής του πίνακα \mathbf{W} . Επιλέγεται η γραμμή με το μικρότερο αποτέλεσμα και πολλαπλασιάζεται με το σήμα \mathbf{y} . Το σύμβολο που προκύπτει πολλαπλασιάζεται με την αντίστοιχη ισότη στήλη του πίνακα \mathbf{H} που περιγράφει το κανάλι και στη συνέχεια αφαιρείται από το σήμα \mathbf{y} για να προκύψει το νέο $\mathbf{y}_{(1)}$. Η στήλη του πίνακα \mathbf{H} η οποία αντιστοιχεί στον δέκτη από τον οποίο προήλθε το σήμα που ανιχνεύτηκε αφαιρείται από τον νέο πίνακα \mathbf{H} που χρησιμοποιείται για τον δεύτερο κύκλο λειτουργίας του δέκτη και τον υπολογισμό του νέου φίλτρου \mathbf{W} . Η ίδια διαδικασία επαναλαμβάνεται μέχρι να ανιχνευτούν όλα τα σύμβολα που έχουν ληφθεί ταυτόχρονα από το δέκτη. Το παρακάτω τμήμα του κώδικα πραγματοποιεί την διαδικασία που περιγράψαμε και προέρχεται από τον κώδικα του προγράμματος `mimo_OSIC_ZF_RECEIVER.m`.

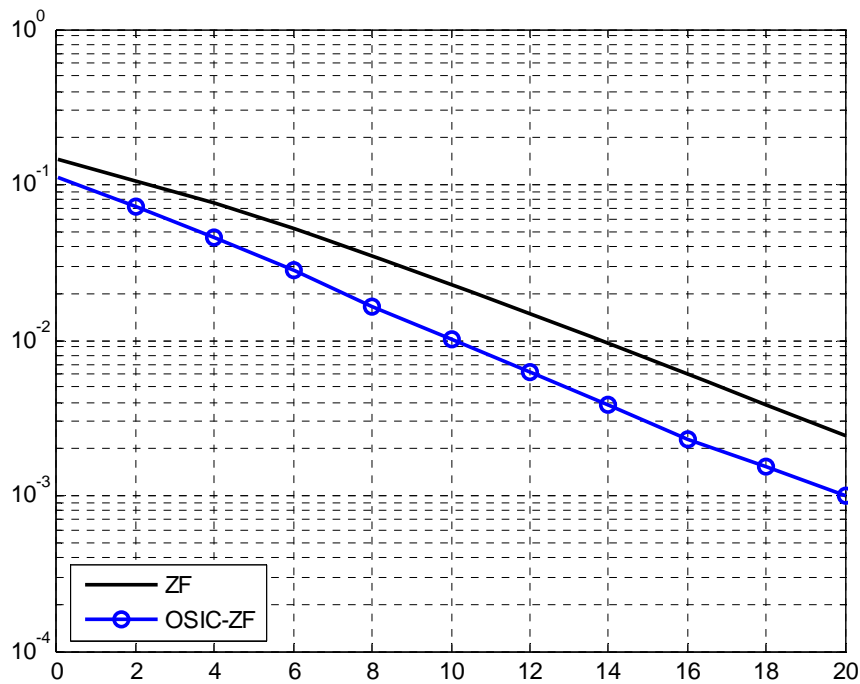
```
for T=1:Timeslots
    H_hat=H(:, :, T);
    Order=[];
    index_array=[1:NT];
    for stage=1:NT
```

```

W_osiczf = (H_hat'*H_hat)^-1 * H_hat';
norm_array=[];
    for i=1:NT-(stage-1)
        norm_array(i) = norm(W_osiczf(i,:));
    end
[val_min,index_min] = min(norm_array);
Order = [Order index_array(index_min)];
index_array = index_array([1:index_min-1
index_min+1:end]);
x_tmp(stage) = W_osiczf(index_min,:)*Ynoisy(:,T);
x_tmp(stage)=(1/scale)*x_tmp(stage);
X_tmp(stage) =
demodulate(modem.qamdmod(M),x_tmp(stage));
X_hat(stage)=
modulate(modem.qammod(M),X_tmp(stage));
X_hat(stage)=scale*X_hat(stage);
Ynoisy_tilde(:,T) = Ynoisy(:,T)...
...-H_hat(:,index_min)*X_hat(stage);
H_hat_tilde = H_hat(:,[1:index_min-1
index_min+1:NT-(stage...
...-1)]);
H_hat = H_hat_tilde; Ynoisy(:,T) =
Ynoisy_tilde(:,T);
    end
Yest_osiczf(Order,T) = X_hat;
end

```

Το διάγραμμα της εικόνας 4.9 έχει προκύψει από το πρόγραμμα ZF_vs_OSIC_ZF_receiver.m και αποτυπώνει τη βελτίωση του ρυθμού σφαλμάτων με τη χρήση του OSIC Zero Forcing δέκτη.



4.21 Σύγκριση δεκτών ZF και OSIC ZF

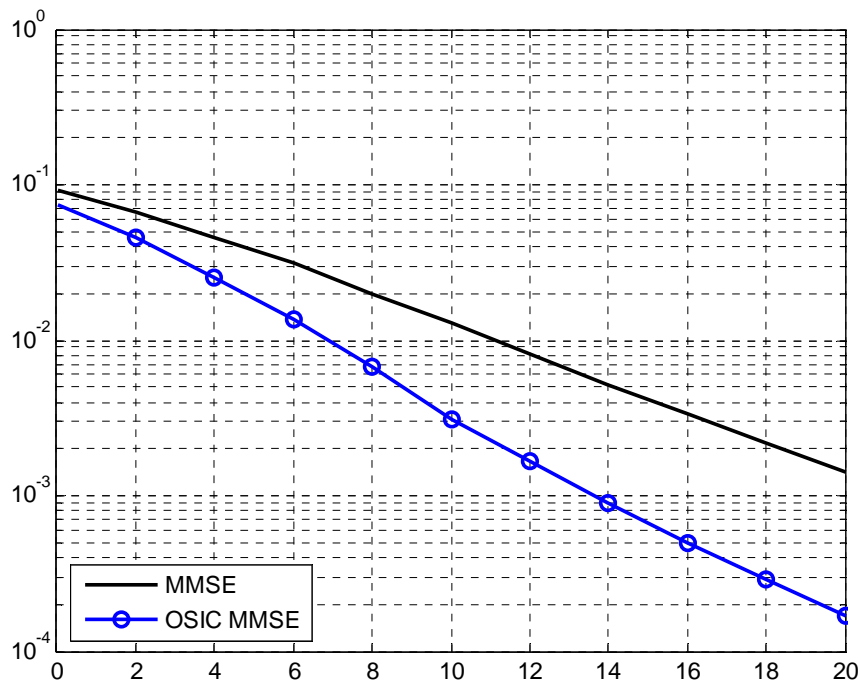
Αυτό που διαφοροποιείται όταν χρησιμοποιείται το φίλτρο MMSE είναι ο υπολογισμός του βέλτιστου κλάδου που θα ανιχνευτεί με τη μέγιστη προτεραιότητα. Στην περίπτωση αυτή η επιλογή γίνεται με βάση το λόγο του σήματος προς το θόρυβο και την παρεμβολή. Η συνάρτηση που δίνει το λόγο αυτό με βάση τον οποίο θα γίνει η επιλογή είναι:

$$SINR_i = \frac{|W_i h_i|^2}{\sum_{j \neq i} |W_i h_j|^2 + \sigma_n^2 |W_i|^2}$$

όπου \mathbf{W}_i είναι η i -οστή γραμμή του πίνακα \mathbf{W} ενώ το διάνυσμα \mathbf{h} αντιστοιχεί στις στήλες του πίνακα \mathbf{H} . [11] Το τμήμα του κώδικα που πραγματοποιεί αυτόν τον υπολογισμό το οποίο είναι και το μόνο που αλλάζει στον κώδικα του προγράμματος mimo_OSIC_MMSE_RECEIVER.m σε σχέση με το πρόγραμμα mimo_OSIC_ZF_RECEIVER.m είναι το ακόλουθο:

```
H_hat=H(:, :, T);
    Order=[];
    index_array=[1:NT];
    for stage=1:NT
        W_osicmmse =(H_hat'*H_hat+(sigma^2)*eye (NT+1-
stage))\H_hat';
        WH=W_osicmmse*H_hat;
        SINR=[];
        for i=1:NT-(stage-1)
            SINR(i)=(abs(WH(i,i))^2)/(sum(abs(WH(i,[1:i-1
i+1:NT-
            (stage-
1)]))^2)+(sigma^2)*norm(W_osicmmse(i,:))^2);
        end
```

Η βελτίωση του ρυθμού σφαλμάτων με τη χρήση της τεχνικής OSIC_MMSE σε σχέση με την απλή εφαρμογή του γραμμικού φίλτρου MMSE φαίνεται στο διάγραμμα της εικόνας 4.10 και έχει προκύψει από την εκτέλεση του προγράμματος MMSE_vs_OSIC_MMSE_receiver.m

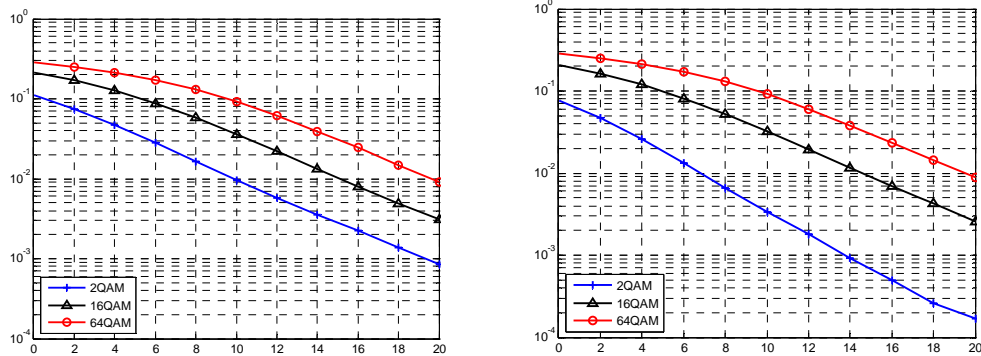


4.22 Σύγκριση δεκτών MMSE και OSIC MMSE

Το πλήθος των τιμών SINR που καλείται να υπολογίσει ο δέκτης ανέρχεται σε $N_T(N_T+1)/2$. Ο ίδιος αριθμός υπολογισμών απαιτείται και για τον OSIC ZF δέκτη με τη διαφορά ότι σε εκείνη την περίπτωση οι υπολογισμοί αφορούν το σηματοθορυβικό λόγο χωρίς να συνυπολογίζεται η παρεμβολή. Το πλήθος των υπολογισμών καθώς και η ανάγκη για κατασκευή νέου φίλτρου W σε κάθε κύκλο λειτουργίας του δέκτη αφού διαφοροποιείται και ο πίνακας H που περιγράφει τη μετάδοση, αυξάνει την πολυπλοκότητα των δεκτών αυτών σε σχέση με τις απλές περιπτώσεις ZF και MMSE.

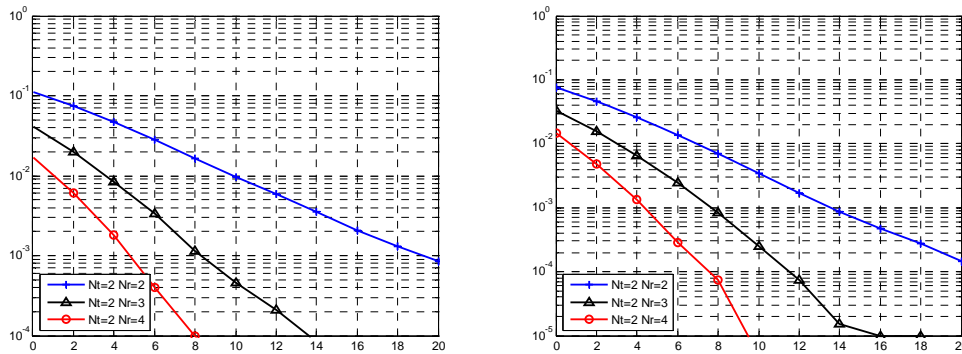
Οι δέκτες διατεταγμένης καταπίεσης της παρεμβολής (OSIC receivers) επιτυγχάνουν μεγαλύτερη αξιοπιστία σε σχέση με του γραμμικούς δέκτες ZF και MMSE που παρουσιάσαμε στην προηγούμενη ενότητα. Η συμπεριφορά τους ωστόσο κατά τη χρησιμοποίηση διαφορετικών σχημάτων κωδικοποίησης είναι όμοια με αυτή

των γραμμικών δεκτών. Στην εικόνα 4.11 παρουσιάζεται η συμπεριφορά των OSIC δεκτών σε ένα σύστημα με δύο κεραίες εκπομπής και δυο κεραίες λήψης για τις περιπτώσεις κατά τις οποίες η μετάδοση των δεδομένων γίνεται με 2, 16 και 64-QAM κωδικοποίηση.



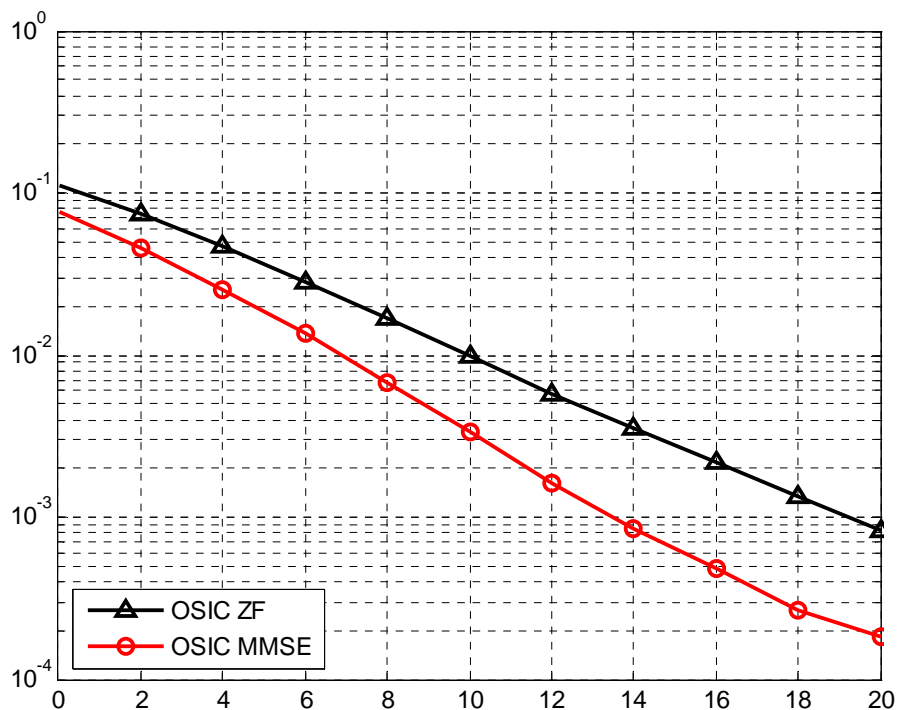
4.23 Απόδοση OSIC ZF & OSIC MMSE Δέκτη για διαφορετικό σχήμα διαμόρφωσης

Το ίδιο παρατηρούμε και στην εικόνα 4.12 όπου παρουσιάζονται τα αποτελέσματα της προσομοίωσης της λειτουργίας των OSIC δεκτών για διαφορετικές διατάξεις κεραιών. Η διαμόρφωση είναι 2 QAM και η τάξη διαφορισμού που επιτυγχάνεται είναι όπως και στους γραμμικούς δέκτες $N_R - N_T + 1$.



4.24 Απόδοση OSIC ZF & OSIC MMSE Δέκτη για διαφορετική διάταξη κεραιών

Όπως και στην περίπτωση των γραμμικών δεκτών έτσι και για τους OSIC δέκτες η απόδοση του δέκτη που χρησιμοποιεί το φίλτρο ελαχιστοποίησης τετραγωνικού σφάλματος είναι ελαφρώς καλύτερη από την απόδοση του OSIC ZF δέκτη. Στο διάγραμμα της εικόνας 4.13 παρουσιάζονται τα αποτελέσματα της προσομοίωσης μέσω του προγράμματος OSIC_ZF_VS_OSIC_MMSE_RECEIVER.m. Το πρόγραμμα αυτό προσομοιώνει τη λειτουργία των δύο δεκτών για το ίδιο σήμα. Το σύστημα περιλαμβάνει δύο κεραίες εκπομπής και δύο κεραίες λήψης ενώ το σχήμα κωδικοποίησης είναι 2QAM.



4.25 Σύγκριση απόδοσης OSIC ZF και OSIC MMSE δέκτη με 2QAM διαμόρφωση

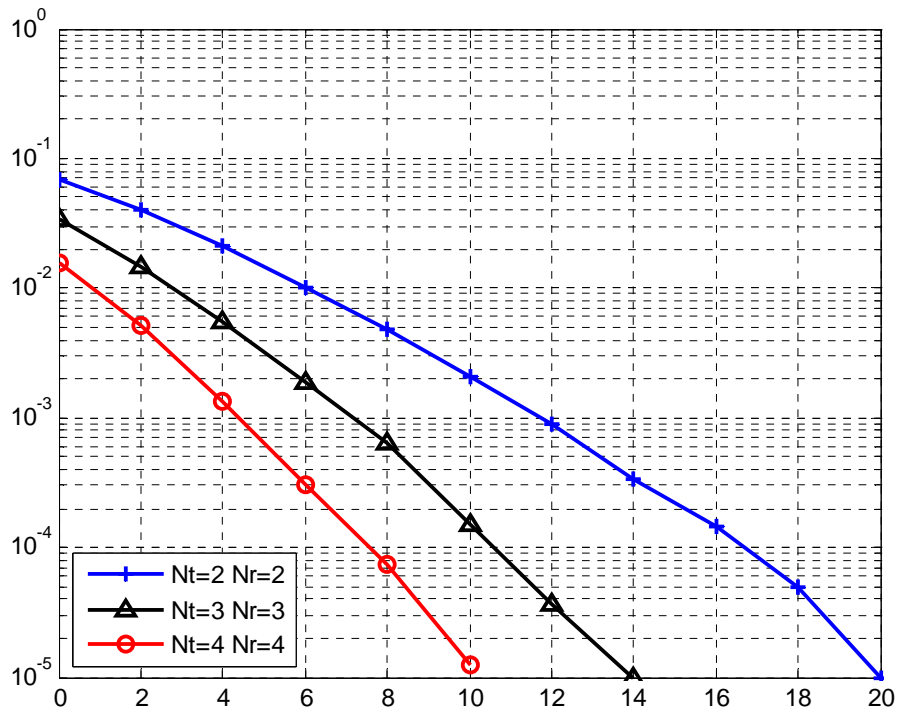
4.4 Δέκτης Μέγιστης Πιθανοφάνειας

Ο Δέκτης Μέγιστης Πιθανοφάνειας (Maximum Likelihood detector) [14] αποτελεί την πιο αξιόπιστη λύση για την ανίχνευση του διανύσματος συμβόλων που μεταδόθηκε. Ο δέκτης αυτός εξετάζει την ευκλείδεια απόσταση μεταξύ του διανύσματος y που λαμβάνει και του γινομένου όλων των πιθανών συνδυασμών συμβόλων με το κανάλι μεταφοράς. Προϋπόθεση είναι ο δέκτης να έχει πλήρη γνώση του πίνακα H που περιγράφει το κανάλι. Ο συνδυασμός συμβόλων που δίνει την ελάχιστη ευκλείδεια απόσταση από το ληφθέν σήμα y αποτελεί την λύση μέγιστης πιθανοφάνειας για το σήμα που μεταδόθηκε.

$$\hat{x}_{ML} = \arg \min_{x \in M^{NT}} \|y - Hx\|^2$$

Η εξαντλητική αναζήτηση είναι ο πιο αποτελεσματικός τρόπος για την αξιόπιστη εύρεση του πιθανότερου διανύσματος, αλλά είναι φυσικό να συνεπάγεται μεγάλη αύξηση της πολυπλοκότητας του αλγορίθμου. Η πολυπλοκότητα εξαρτάται από το πλήθος των συνδυασμών συμβόλων που προκύπτουν από το σχήμα κωδικοποίησης και τον αριθμό κεραιών που χρησιμοποιούνται κατά τη μετάδοση. Ο αριθμός αυτών των συνδυασμών είναι M^{NT} όπου NT ο αριθμός των κεραιών εκπομπής και M το πλήθος των πιθανών συμβόλων σύμφωνα με το σχήμα κωδικοποίησης. Είναι εύκολα κατανοητό πως για σχήμα κωδικοποίησης 16 QAM και υψηλότερο και κεραιές εκπομπής περισσότερες από 2 η πολυπλοκότητα του αλγορίθμου αυξάνει τόσο πολύ που η χρήση τέτοιου είδους δέκτη καθίσταται απαγορευτική.

Ο ML Decoder επιτυγχάνει τη μέγιστη δυνατή τάξη διαφορισμού ίση με NR δηλαδή ίση με το πλήθος των κεραιών του δέκτη. Αυτή η τάξη διαφορισμού επιτυγχάνεται με τη χρήση του δέκτη MRC (Maximum Ratio Combining) που είναι δέκτης σε συστήματα SIMO. Η προσομοίωση της λειτουργίας του δέκτη πραγματοποιείται μέσω του προγράμματος ML_receiver.m. Η εικόνα 4.14 παρουσιάζει τα αποτελέσματα της προσομοίωσης του δέκτη μέγιστης πιθανοφάνειας για διαφορετικές συμμετρικές διατάξεις κεραιών με χρήση κωδικοποίησης 2QAM.

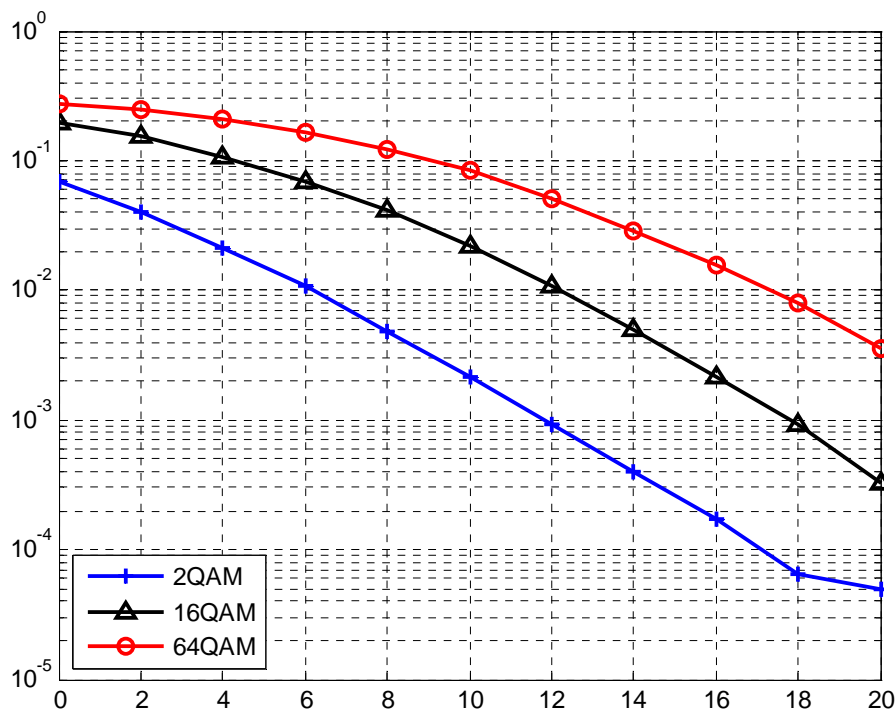


4.26 Απόδοση δέκτη Μέγιστης πιθανοφάνειας (ML) με διαμόρφωση 2QAM

Στις περιπτώσεις δεκτών που παρουσιάστηκαν στις προηγούμενες ενότητες η αύξηση της αξιοπιστίας τους ήταν αποτέλεσμα της αύξησης των κεραιών λήψης σε σχέση με τον αριθμό των κεραιών εκπομπής. Στην περίπτωση του δέκτη μέγιστης πιθανοφάνειας η αξιοπιστία αυξάνεται με την αύξηση των κεραιών λήψης που συνοδεύεται με ταυτόχρονη αύξηση των κεραιών εκπομπής, που σημαίνει και αύξηση της ταχύτητας μετάδοσης των δεδομένων αφού για κάθε επιπλέον κεραία εκπομπής μεταδίδεται ένα επιπλέον σύμβολο την ίδια χρονική στιγμή.

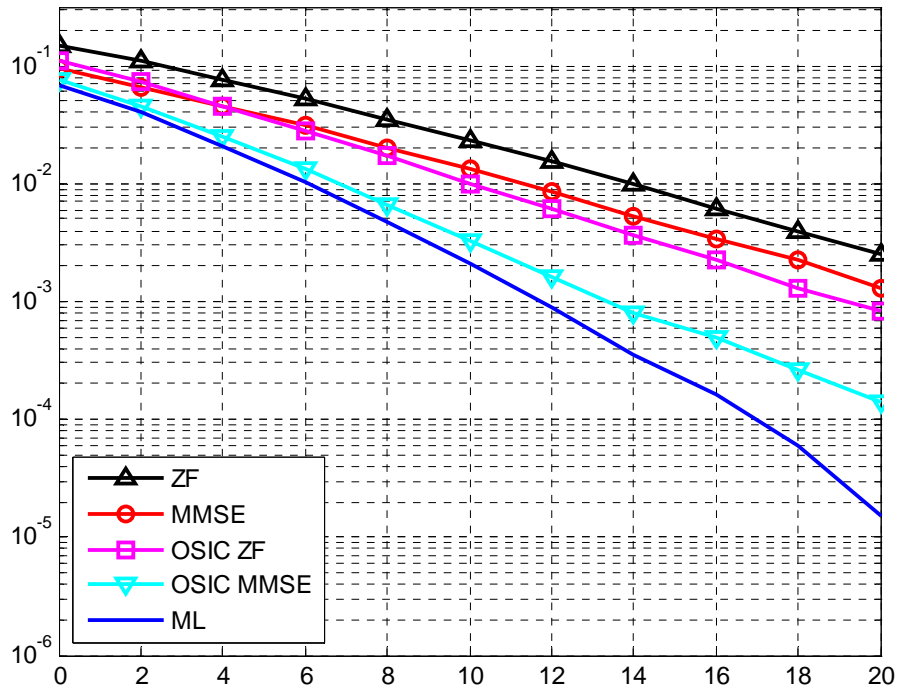
Είναι λογικό πως όπως και στις προηγούμενες περιπτώσεις δεκτών, έτσι και για τον δέκτη μέγιστης πιθανοφάνειας η απόδοσή του μειώνεται καθώς αυξάνεται το σχήμα κωδικοποίησης των δεδομένων. Το γεγονός αυτό αποτυπώνεται στο διάγραμμα της εικόνας 4.15. Το διάγραμμα αυτό παρουσιάζει τον ρυθμό σφαλμάτων ενός συστήματος με δύο κεραιές εκπομπής και λήψης για τις περιπτώσεις 2, 16 και

64QAM διαμόρφωσης και έχει προκύψει μέσω του προγράμματος MQAM_ML_receiver.m.



4.27 Απόδοση ML δέκτη για διαφορετικά σχήματα διαμόρφωσης.

Η αξιοπιστία του δέκτη μέγιστης πιθανοφάνειας είναι η μέγιστη δυνατή και η αξιοπιστία που παρέχει ο δέκτης αυτός χρησιμοποιείται σαν σημείο αναφοράς για την αξιολόγηση της επίδοσης των υπολοίπων δεκτών που χρησιμοποιούνται στα MIMO συστήματα. Η σύγκριση των αποδόσεων των δεκτών που παρουσιάστηκαν μέχρι αυτό το σημείο πραγματοποιείται με τη χρήση του προγράμματος ML_OSIC_MMSE_OSIC_ZF_MMSE_ZF_comparison.m. Η προσομοίωση αφορά σύστημα με δύο κεραίες εκπομπής και δύο κεραίες λήψης και το σχήμα διαμόρφωσης που χρησιμοποιείται είναι 2QAM. Τα αποτελέσματα της προσομοίωσης παρουσιάζονται στο διάγραμμα της εικόνας 4.16.



4.28 Σύγκριση απόδοσης δεκτών 2QAM

4.5 Sphere Detection

Όπως παρουσιάστηκε στην περίπτωση του ιδανικού δέκτη μέγιστης πιθανοφάνειας (ML Decoder), ο βέλτιστος τρόπος για την ανίχνευση του διανύσματος των συμβόλων που μεταδόθηκαν είναι η εξαντλητική αναζήτηση ανάμεσα σε όλους τους πιθανούς συνδυασμούς συμβόλων με κριτήριο την ελαχιστοποίηση της απόστασης $\|y - Hx\|^2$ όπου y είναι το διάνυσμα μιγαδικών συμβόλων που έχει λάβει ο δέκτης, H ο μιγαδικός πίνακας που περιγράφει το κανάλι και x το πιθανό διάνυσμα μιγαδικών συμβόλων που έστειλε ο πομπός. Ο αλγόριθμος του Sphere Decoder [17 – 20] έχει ως στόχο να περιορίσει την αναζήτηση μέσα στο περίγραμμα μιας σφαίρας που περιλαμβάνει ένα υποσύνολο των συνδυασμών των συμβόλων που ενδεχομένως έχει αποστείλει ο πομπός. Έτσι αντί για την εξαντλητική αναζήτηση που προαναφέραμε ο Sphere Decoder πραγματοποιεί την αναζήτηση για

τους συνδυασμούς συμβόλων που ικανοποιούν την σχέση $\|y - Hx\|^2 \leq C$ με $C > 0$ και \sqrt{C} είναι η ακτίνα της σφαίρας μέσα στην οποία γίνεται η αναζήτηση με κέντρο το σημείο y . Όσο μεγαλύτερη είναι αυτή η ακτίνα τόσο αυξάνεται ο αριθμός των συνδυασμών συμβόλων που πρέπει να εξεταστούν και συνεπώς η πολυπλοκότητα του αλγορίθμου. Αν η ακτίνα είναι ωστόσο πολύ μικρή τότε η σφαίρα δεν περικλείει κανένα σημείο και έτσι δεν είναι δυνατός ο εντοπισμός της ML λύσης. Η επιλογή της συγκεκριμένης ακτίνας αποτελεί ένα άλλο πρόβλημα στο οποίο δεν θα αναφερθούμε στη συγκεκριμένη εργασία. Ένας απλός τρόπος για την επιλογή της αρχικής ακτίνας μέσα στην οποία πραγματοποιείται η αναζήτηση είναι η επίλυση της εξίσωσης $C = \|y - Hx\|^2$ θέτοντας όπου x το διάνυσμα των συμβόλων που προκύπτει από την αποκωδικοποίηση του σήματος y με το φίλτρο του Zero Forcing Receiver. Με αυτόν τον τρόπο εξασφαλίζουμε ότι εντός του περιγράμματος της σφαίρας υπάρχει τουλάχιστον μία λύση αυτή που αντιστοιχεί στο διάνυσμα που ανιχνεύεται με τη μέθοδο zero forcing.

Πρώτη διαδικασία για την εφαρμογή του δέκτη Sphere Decoder είναι η μετατροπή του συστήματος που περιγράφει τη μετάδοση της πληροφορίας από ένα σύστημα μιγαδικών αριθμών σε σύστημα πραγματικών αριθμών. Αν υποθέσουμε ότι έχουμε ένα σύστημα μετάδοσης που αποτελείται από δυο κεραίες εκπομπής και δυο κεραίες λήψης που περιγράφεται από την εξίσωση : $y = Hx + n$. Τα στοιχεία της εξίσωσης είναι μιγαδικά, όπου y το μιγαδικό διάνυσμα συμβόλων που ελήφθη, x το αντίστοιχο που μεταδόθηκε, n ο λευκός θόρυβος που προστέθηκε στην πλευρά του δέκτη και H ο πίνακας που περιγράφει το κανάλι της μετάδοσης του οποίου τα στοιχεία είναι επίσης μιγαδικά. Σύμφωνα με την υπόθεση ότι το σύστημα είναι 2x2 μπορούμε να αναπτύξουμε τους πίνακες.

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}$$

Συμβολίζοντας το πραγματικό μέρος των στοιχείων προσθέτοντας το δείκτη R και το φανταστικό προσθέτοντας το δείκτη I προκύπτει:

$$\begin{bmatrix} y_{1R} + jy_{1I} \\ y_{2R} + jy_{2I} \end{bmatrix} = \begin{bmatrix} h_{11R} + jh_{11I} & h_{12R} + jh_{12I} \\ h_{21R} + jh_{21I} & h_{22R} + jh_{22I} \end{bmatrix} * \begin{bmatrix} x_{1R} + jx_{1I} \\ x_{2R} + jx_{2I} \end{bmatrix} + \begin{bmatrix} n_{1R} + jn_{1I} \\ n_{2R} + jn_{2I} \end{bmatrix}$$

Διαχωρίζοντας το πραγματικό και το φανταστικό μέρος αυτής της εξίσωσης προκύπτουν

A:

B:

$$\begin{bmatrix} y_{1I} \\ y_{2I} \end{bmatrix} = \begin{bmatrix} h_{11I} & h_{12I} & h_{11R} & h_{12R} \\ h_{21I} & h_{22I} & h_{21R} & h_{22R} \end{bmatrix} * \begin{bmatrix} x_{1R} \\ x_{2R} \\ x_{1I} \\ x_{2I} \end{bmatrix} + \begin{bmatrix} n_{1I} \\ n_{2I} \end{bmatrix}$$

Από τις εξισώσεις A&B προκύπτει το σύστημα:

$$\begin{bmatrix} y_{1R} \\ y_{2R} \\ y_{1I} \\ y_{2I} \end{bmatrix} = \begin{bmatrix} h_{11R} & h_{12R} & -h_{11I} & -h_{12I} \\ h_{21R} & h_{22R} & -h_{21I} & -h_{22I} \\ h_{11I} & h_{12I} & h_{11R} & h_{12R} \\ h_{21I} & h_{22I} & h_{21R} & h_{22R} \end{bmatrix} * \begin{bmatrix} x_{1R} \\ x_{2R} \\ x_{1I} \\ x_{2I} \end{bmatrix} +$$

Αναφερόμενοι πλέον στο νέο σύστημα πραγματικών αριθμών που περιγράφει τη μετάδοση και υποθέτοντας ότι τα δεδομένα κωδικοποιούνται σύμφωνα με 16QAM κωδικοποίηση, τα πιθανά ψηφία που αποτελούν τις τιμές των στοιχείων του διανύσματος x είναι: $\{-3 -1 1 3\}$. Ο Sphere Decoder εκμεταλλεύεται την ακόλουθη σχέση:

$$\|y - Hx\|^2 = (x - x_{ZF})^T R^T R (x - x_{ZF})$$

Ο πίνακας R είναι ο άνω τριγωνικός πίνακας που προκύπτει από τον QR μετασχηματισμό του πίνακα H και x_{ZF} είναι το αποτέλεσμα της επίδρασης του φίλτρου του Zero Forcing receiver στο ληφθέν σήμα y . Συνεπώς η πρώτη εργασία που πραγματοποιείται σε έναν τέτοιο δέκτη είναι η μετατροπή του συστήματος σε σύστημα πραγματικών αριθμών και η ανίχνευση του διανύσματος x_{ZF} . Το πρόγραμμα που πραγματοποιεί την αποκωδικοποίηση που περιγράφεται στην

ενότητα αυτή είναι το sphere_decoder.m. Στο τμήμα το κώδικα που ακολουθεί πραγματοποιείται η παραπάνω διαδικασία:

```

Y =[real(Y); imag(Y)];

H =[real(H) -(imag(H)) ; imag(H) real(H)];

x_hat=(((H'*H)^-1)*H')*Y;

x_hat=(1/scale)*x_hat;

[Q,R] = qr(H);

```

Το σήμα που έχει λάβει ο δέκτης είναι το \mathbf{Y} , \mathbf{H} είναι ο πίνακας που περιγράφει το κανάλι και $\mathbf{x_hat}$ είναι το σήμα που προκύπτει μέσω της εφαρμογής του φίλτρου \mathbf{W} του Zero Forcing δέκτη. Το σήμα $\mathbf{x_hat}$ πρέπει να πολλαπλασιαστεί με τον παράγοντα $(1/scale)$ για να αναστραφεί η κανονικοποίηση της ισχύος που επιβλήθηκε στην πλευρά εκπομπής. Το σήμα $\mathbf{x_hat}$ δεν περιλαμβάνει σύμβολα του αστερισμού 16QAM που χρησιμοποιούνται στο παράδειγμα που περιγράφουμε καθώς περιλαμβάνει θόρυβο. Πραγματοποιείται επίσης ο μετασχηματισμός του πίνακα \mathbf{H} για να προκύψει ο τριγωνικός πίνακας \mathbf{R} .

Σύμφωνα με την υπόθεση επιλέγουμε ως ακτίνα της σφαίρας μέσα στην οποία πραγματοποιείται η αναζήτηση της λύσης την ακτίνα της σφαίρας που περικλείει τη λύση που προκύπτει μέσω το Zero Forcing δέκτη. Ο υπολογισμός της ακτίνας αυτής γίνεται όταν αντικατασταθεί στην παράσταση $(\mathbf{x} - \mathbf{x}_{ZF})' \mathbf{R}' \mathbf{R} (\mathbf{x} - \mathbf{x}_{ZF}) = \mathbf{C}$ το διάνυσμα \mathbf{x} από τη λύση που προκύπτει μέσω του δέκτη αυτού. Όπως προαναφέραμε η τιμή του \mathbf{x}_{ZF} είναι αυτή που προκύπτει από την απλή αντιστροφή της επίδρασης του καναλιού και περιλαμβάνει θόρυβο. Για να φτάσουμε στο διάνυσμα που περιλαμβάνει σύμβολα του αστερισμού 16QAM θα πρέπει να πραγματοποιηθεί η διαδικασία slicing η οποία αντικαθιστά τις τιμές του \mathbf{x}_{ZF} με τις πλησιέστερες τιμές που αντιστοιχούν σε σύμβολα του αστερισμού. Το πρόγραμμα QAM_16_slicer.m πραγματοποιεί την διαδικασία αυτή.

```
function [x_sliced]=QAM_16_slicer(x_hat)
find(x_hat<=-2)=-3;
find((x_hat>=-2)&(x_hat<0))=-1;
find(x_hat>2)=3;
find((x_hat>0)&(x_hat<2))=1;
x_sliced=find';
```

Η συνάρτηση παίρνει ως είσοδο το διάνυσμα x_{hat} που προέκυψε από την αντιστροφή της επίδρασης του καναλιού. Όπως φαίνεται και από το τμήμα του κώδικα η διαδικασία είναι αρκετά απλή. Εφόσον η τιμές των συμβόλων είναι: **{-3 -1 1 3}** για τιμές μικρότερες από -2 αντικαθιστά με -3, για τιμές μικρότερες από 0 και μεγαλύτερες του -2 αντικαθιστά με -1, για τιμές μεγαλύτερες από 0 και μικρότερες του 2 αντικαθιστά με 1 και για τιμές μεγαλύτερες από 2 αντικαθιστά με την τιμή 3. Η έξοδος του προγράμματος αποτελεί το διάνυσμα των συμβόλων που θα ήταν η λύση ενός Zero Forcing δέκτη. Στο παράρτημα της εργασίας υπάρχουν ακόμα δύο περιπτώσεις παρόμοιων προγραμμάτων για τις περιπτώσεις 4- και 64QAM . Με δεδομένο πλέον το διάνυσμα x_{sliced} το πρόγραμμα sphere_decoder.m υπολογίζει την τιμή C.

```
radius_squared = norm(R*(x_sliced-x_hat))^2;
```

Η τιμή αυτή C είναι το τετράγωνο της ακτίνας της σφαίρας μέσα στην οποία πραγματοποιείται η αναζήτηση της ML λύσης και η σχέση που καλείται να λύσει ο sphere decoder γίνεται:

$$\|y - Hx\|^2 = (x - x_{ZF})^T R^T R (x - x_{ZF}) = \xi^T R^T R \xi = \|R\xi\|^2 \leq C$$

Όπως παρατηρούμε στην παραπάνω ανισότητα ο η τιμές των συμβόλων που αναζητά ο δέκτης περιλαμβάνονται σε άθροισμα τετραγώνων τα οποία πρέπει να είναι

μικρότερα από έναν θετικό αριθμό συνεπώς ο κάθε τετραγωνικός όρος πρέπει να είναι μικρότερος από την τιμή C. Αν αναπτύξουμε τη σχέση $\|R(x - x_{ZF})\|^2 \leq C$

$$x_4 - x_{4ZF}$$

Μπορούμε να αναλύσουμε τη λειτουργία του sphere decoder στα ακόλουθα βήματα:

Βήμα 1^ο

Ο αλγόριθμος ξεκινά με την αναζήτηση του συμβόλου x_4 . Για το σύμβολο αυτό πρέπει να επιλεγεί μια από τις πιθανές τιμές των πραγματικών συμβόλων $\{-3 -1 1 3\}$

. Η τιμή που θα επιλεγεί πρέπει να επαληθεύει τη σχέση:

$$|R_{44}(x_4 - x_{4ZF})|^2 \leq C \Rightarrow$$

$$x_{4ZF} - \frac{\sqrt{C}}{R_{44}} \leq x_4 \leq x_{4ZF} + \frac{\sqrt{C}}{R_{44}}$$

Βήμα 2^ο

Δεδομένης της επιλογής για την τιμή του συμβόλου x_4 ο αλγόριθμος συνεχίζει με την αναζήτηση της τιμής του x_3 σύμφωνα με τη σχέση :

$$|R_{44}(x_4 - x_{4ZF})|^2 + |R_{33}(x_3 - x_{3ZF}) + R_{34}(x_4 - x_{4ZF})|^2 \leq C \Rightarrow$$

$$x_{3ZF} - \frac{\sqrt{C - |R_{44}(x_4 - x_{4ZF})|^2 - R_{34}(x_4 - x_{4ZF})}}{R_{33}} \leq x_3 \leq x_{3ZF} + \frac{\sqrt{C - |R_{44}(x_4 - x_{4ZF})|^2 - R_{34}(x_4 - x_{4ZF})}}{R_{33}}$$

Η τιμή που ικανοποιεί την ανισότητα επιλέγεται ως τιμή του x_3 και ο αλγόριθμος συνεχίζει στα επόμενα βήματα.

Βήμα 3^ο

Παρόμοια εξίσωση προκύπτει για την επιλογή τιμής που θα ανατεθεί στο x_2 σύμφωνα με τις τιμές που έχουν ανατεθεί στα x_3 και x_4 .

Βήμα 4^ο

Με τον ίδιο τρόπο επιλέγεται τιμή για το σύμβολο x_1 δεδομένων των τιμών των υπολοίπων τριών συμβόλων.

Σύμφωνα με όσα παρουσιάστηκαν στα παραπάνω βήματα, ο sphere decoder αναζητά τις πιθανές τιμές των συμβόλων που μεταδόθηκαν μέσα σε κάποια όρια. Τα όρια αυτά υπολογίζονται στο πρόγραμμα του matlab bound2.m, προκύπτουν από τη σχέση:

$$\sum_{i=1}^N \left(R_{ii} \xi_i + \sum_{j=i+1}^N R_{ij} \xi_j \right)^2 \leq C$$

και καθορίζονται από τις παρακάτω εξισώσεις:

$$\left[-\sqrt{\frac{T_i}{Q_{ii}}} + S_i \right] \leq x_i \leq \left[\sqrt{\frac{T_i}{Q_{ii}}} + S_i \right] \quad (1)$$

Όπου $\lceil a \rceil$ είναι ο μικρότερος ακέραιος αριθμός μεγαλύτερος από a και $\lfloor a \rfloor$ ο μεγαλύτερος ακέραιος μικρότερος από a .

$$S_i = x_{ZF_i} - \sum_{l=i+1}^N Q_{il} \xi_l \quad (2)$$

$$T_i = C - \sum_{l=i+1}^N \left(\sum_{j=l}^N R_{lj} \xi_j \right)^2 \quad (3)$$

$$Q_{ij} = \begin{cases} R_{ii}^2 & \text{για } i = j \\ \frac{R_{ij}}{R_{ii}} & \text{για } i \neq j \end{cases} \quad (4)$$

Ο πίνακας \mathbf{Q} είναι άνω τριγωνικός όπως και ο πίνακας \mathbf{R} συνεπώς ενδείκνυται η αναζήτηση και επιλογή των τιμών του x_i να πραγματοποιείται με αφετηρία το $i=N$ όπου N το μέγεθος του διανύσματος \mathbf{x} και προχωρώντας αντίστροφα. Έτσι ο υπολογισμός των ορίων της σχέσης (1) γίνεται χρησιμοποιώντας τον τύπο:

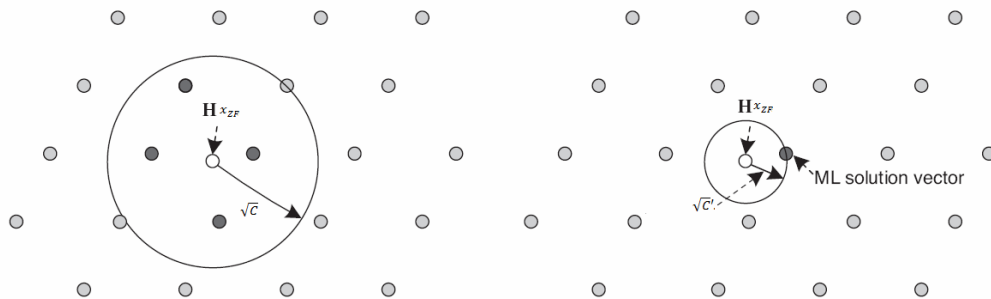
$$T_{i-1} = T_i - \left(\sum_{n=i}^N R_{in} x_n \right)^2 \quad (5)$$

Χρησιμοποιώντας τα όρια που προκύπτουν μέσω του προγράμματος bound2.m το πρόγραμμα stage_processing.m δημιουργεί σε κάθε βήμα του αλγορίθμου μία λίστα με τις πιθανές πραγματικές τιμές συμβόλων που βρίσκονται μέσα στα όρια αυτά. Το ίδιο πρόγραμμα επιλέγει την πρώτη από τις τιμές της λίστας που έχει δημιουργηθεί και την αναθέτει στο αντίστοιχο x_i ανάλογα με το βήμα στο οποίο βρίσκεται ο αλγόριθμος. Τέλος ενημερώνει τη λίστα με τις πιθανές τιμές, αφαιρώντας την τιμή την οποία έχει αναθέσει στο x_i . Αν τα όρια που έχουν τεθεί δεν περικλείουν καμία πραγματική τιμή συμβόλου τότε ο αλγόριθμος καλείται να οπισθοχωρήσει κατά ένα βήμα. Το πρόγραμμα bound2.m καλείται μόνο όταν ο αλγόριθμος προχωράει προς τα εμπρός. Όταν ο αλγόριθμος οπισθοχωρεί, οι λίστες με τις πιθανές τιμές συμβόλων προ-υπάρχουν και επιλέγεται η πρώτη τιμή από την εκάστοτε λίστα. Όταν ο αλγόριθμος φτάσει στο σημείο να μην υπάρχει πιθανή τιμή για το x_4 , δηλαδή για το πρώτο σύμβολο που καλείται να επιλέξει, τότε είναι απαραίτητος ο επαναπροσδιορισμός της ακτίνας της σφαίρας που έχει οριστεί να περιορίζει την αναζήτηση. Την διαδικασία αυτή την πραγματοποιεί το πρόγραμμα radius_control.m εφόσον χρειαστεί, διπλασιάζοντας την τιμή C που καθορίστηκε αρχικά.

Όταν ολοκληρωθούν και τα τέσσερα βήματα που παρουσιάστηκαν για το σύστημα 2x2, όλα τα σύμβολα x_i έχουν πραγματικές τιμές συμβόλων που ικανοποιούν τη σχέση

$$\sum_{i=1}^N \left(R_{ii} \xi_i + \sum_{j=i+1}^N R_{ij} \xi_j \right)^2 \leq C$$

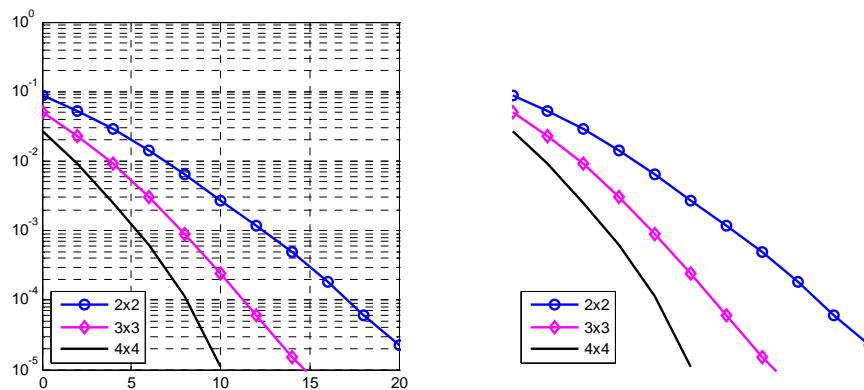
Ο αλγόριθμος επιστρέφει στο βήμα 1 αυτή τη φορά με νέα ακτίνα, εκείνη που προκύπτει από τη σχέση $(x - x_{ZF})^T R^T R (x - x_{ZF}) = C'$ όπου x αυτή τη φορά έχει αντικατασταθεί με το διάνυσμα που έχει προκύψει από τον προηγούμενο κύκλο λειτουργίας του αλγορίθμου. Με τον τρόπο αυτό ο αλγόριθμος σε κάθε κύκλο λειτουργίας του ελαττώνει την ακτίνα της σφαίρας μέσα στην οποία πραγματοποιεί την αναζήτηση μέχρι το σημείο που η ακτίνα περικλείει μόνο τη λύση ML. Τα στάδια από τον πρώτο έως τον τελευταίο κύκλο λειτουργίας του αλγορίθμου παρουσιάζονται στην εικόνα 4.17.



4.29 Τερματισμός του Sphere Decoder.

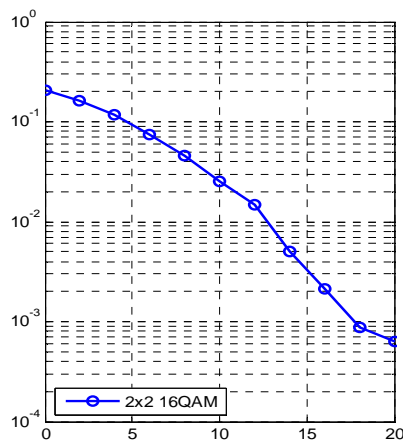
Για να ολοκληρωθεί η λειτουργία του αλγορίθμου θα πρέπει συνεπώς οι λίστες με τις πιθανές τιμές συμβόλων που δημιουργήθηκαν από το πρόγραμμα `stage_processing.m` να είναι κενές. Ο έλεγχος αυτός πραγματοποιείται στο πρόγραμμα `compare_vector_norm.m`. Στο ίδιο πρόγραμμα γίνεται και ο υπολογισμός της νέας ακτίνας C' σύμφωνα με την οποία ο δέκτης επιστρέφει στο βήμα 1 και ξεκινάει τον επόμενο κύκλο λειτουργίας του.

Ο sphere decoder επιτυγχάνει την τάξη διαφορισμού που επιτυγχάνεται και με τη χρήση του ML decoder, ίση δηλαδή με τον αριθμό των κεραιών λήψης N_T . Στην εικόνα 4.18 παρουσιάζονται τα γραφήματα που αποτυπώνουν τον ρυθμό σφαλμάτων που προκύπτουν από τη χρήση των δύο δεκτών για ένα MIMO σύστημα που χρησιμοποιεί 4 QAM διαμόρφωση. Η λειτουργία των δύο δεκτών εξετάζεται για τις περιπτώσεις συμμετρικών συστημάτων με 2, 3 και 4 κεραιές σε κάθε πλευρά της μετάδοσης και η κλίση των καμπυλών που αποτυπώνουν το ρυθμό σφαλμάτων σε σχέση με το σηματοθορυβικό λόγο επιβεβαιώνουν την τάξη διαφορισμού N_T .

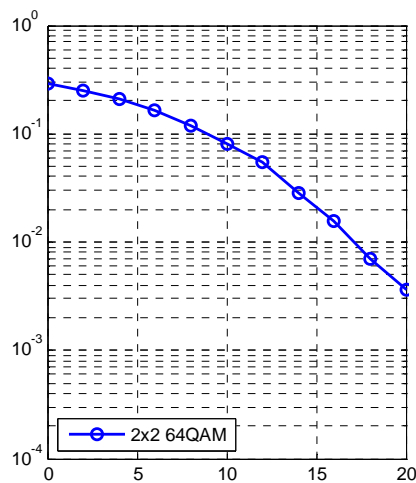


4.30 Απόδοση Sphere & ML decoder με 4QAM

Τα γραφήματα της εικόνας έχουν προκύψει από την εκτέλεση του προγράμματος sphere_vs_ml.m. Στην προσομοίωση αυτή το ίδιο σήμα που διέρχεται από το κανάλι αποκωδικοποιείται από τους δύο δέκτες. Όπως φαίνεται και στην εικόνα οι δυο δέκτες φαίνεται να έχουν την ίδια ακριβώς απόδοση. Κατά την εκτέλεση της προσομοίωσης οι δυο δέκτες δεν εμφάνισαν καμία διαφορά στον αριθμό σφαλμάτων σε καμία τιμή σηματοθορυβικού λόγου και για οποιαδήποτε από τις συμμετρικές διατάξεις των κεραιών. Ακριβώς το ίδιο αποτέλεσμα εμφανίζουν οι προσομοιώσεις και για τις περιπτώσεις που χρησιμοποιείται 16 και 64QAM διαμόρφωση τα γραφήματα των οποίων παρουσιάζονται στις εικόνες 4.19 και 4.20.



4.31 Απόδοση Sphere & ML decoder με 16QAM



4.32 Απόδοση Sphere & ML decoder με 64QAM

Οι προσομοιώσεις αυτές πραγματοποιούνται με μια τροποποιημένη εκδοχή του προγράμματος `sphere_vs_ml.m`. Το πρώτο προφανές σημείο τροποποίησης είναι η αλλαγή της τιμής της μεταβλητής M που προσδιορίζει το σχήμα διαμόρφωσης. Εκτός από την τιμή αυτή πρέπει να αντικατασταθεί η κλήση της συνάρτησης `QAM_4 slicer` της 34της γραμμής του προγράμματος `sphere_decoder.m` με την κατάλληλη συνάρτηση `QAM_16 slicer` ή `QAM_64 slicer`. Τέλος οι προσομοιώσεις

αυτές αφορούν σύστημα με μόνο δύο κεραίες εκπομπής και λήψης καθώς ο χρόνος εκτέλεσης της προσομοίωσης του ML δέκτη για τόσο υψηλά σχήματα διαμόρφωσης και για μεγαλύτερο αριθμό κεραιών εκπομπής από δύο είναι απαγορευτικός. Τα αποτελέσματα των γραφημάτων αποτυπώνουν το γεγονός ότι ο Sphere decoder παρουσιάζει τον ίδιο αριθμό σφαλμάτων με τον ML decoder ανεξάρτητα από το σχήμα διαμόρφωσης.

Ο sphere decoder όπως παρουσιάστηκε μέσα από τις προμοιώσεις μπορεί να παρέχει την ίδια αξιοπιστία με αυτή του δέκτη μέγιστης πιθανοφάνειας η οποία θεωρείται σημείο αναφοράς για την αξιολόγηση της αξιοπιστίας όλων των δεκτών. Το πρόβλημα του ML decoder είναι η μεγάλη πολυπλοκότητα που προσθέτει εξαιτίας της εξαντλητικής αναζήτησης που πραγματοποιείται ανάμεσα σε όλους τους πιθανούς συνδυασμούς συμβόλων. Η πολυπλοκότητα αυτή που προστίθεται λόγω των M^{N_T} συνδυασμών συμβόλων, όπου M το πλήθος των συμβόλων στον αστερισμό που χρησιμοποιείται και N_T το πλήθος των κεραιών εκπομπής, κάνει τη χρήση του ML δέκτη απαγορευτική για σχήμα διαμόρφωσης μεγαλύτερο από 16QAM και πλήθος κεραιών εκπομπής μεγαλύτερο των δύο. Ο sphere decoder με τον περιορισμό της αναζήτησης μέσα στην σφαίρα που καθορίζεται από την αρχική τιμή C που ορίζεται στον αλγόριθμο, καθιστά εφικτή την επίδοση παρόμοιας αξιοπιστίας και για συστήματα με υψηλότερα σχήματα διαμόρφωσης και περισσότερες κεραίες εκπομπής. Ωστόσο οι πολλές διεργασίες που εκτελούνται στον sphere decoder τον καθιστούν ασύμφορο για χαμηλά σχήματα διαμόρφωσης και συστήματα με 2 κεραίες εκπομπής και 2 κεραίες λήψης. Η μείωση της πολυπλοκότητας του Sphere Decoder σε σχέση με τον ML Decoder μπορεί να αποτυπωθεί μέσω του χρόνου εκτέλεσης της προσομοίωσης της λειτουργίας τους. Στους παρακάτω πίνακες αποτυπώνονται οι χρόνοι εκτέλεσης του εκάστοτε δέκτη ανάλογα με το σχήμα διαμόρφωσης και το πλήθος των κεραιών εκπομπής. Οι χρόνοι αυτοί έχουν προκύψει από την εκτέλεση των προγραμμάτων ML_receiver.m και Sphere_simulation.m για μία συγκεκριμένη τιμή σηματοθορυβικού λόγου με την μετάδοση των δεδομένων μέσω του MIMO καναλιού να πραγματοποιείται για 100 χρονοθυρίδες.

ML Decoder			
QAM Nt	4	16	64
2	0,05s	0,16s	2,34s
3	0,06s	2,47s	154,42s
4	0,17s	38,35s	9787,62s

Sphere Decoder			
QAM Nt	4	16	64
2	0,32s	0,33s	0,57s
3	0,41s	0,44s	0,95s
4	0,45s	0,49s	1,37s

Οι χρόνοι εκτέλεσης των προσομοιώσεων που αποτυπώνονται στους παραπάνω πίνακες είναι ενδεικτικοί, αφορούν την εκτέλεση του προγράμματος προσομοίωσης μόνο σε έναν συγκεκριμένο προσωπικό υπολογιστή και παρουσιάζονται μόνο για να καταδείξουν το ανυπέβλητο πρόβλημα που ανακύπτει λόγω της πολυπλοκότητας του ML decoder σε αντίθεση με τους χρόνους που παρουσιάζονται για τον Sphere decoder.

Η μεγάλη πολυπλοκότητα του δέκτη μέγιστης πιθανοφάνειας οφείλεται στην εξαντλητική αναζήτηση που πραγματοποιείται ανάμεσα στους πιθανούς συνδυασμούς των μεταδοθέντων συμβόλων που ανέρχονται σε M^{NT} . Η μείωση της πολυπλοκότητας που επιτυγχάνεται με τη χρήση του Sphere Decoder οφείλεται στη μείωση του αριθμού των πιθανών συνδυασμών συμβόλων μέσω του περιορισμού της αναζήτησης στα όρια της σφαίρας. Η πολυπλοκότητα του αλγορίθμου εξαρτάται από τον καθορισμό της κατάλληλης ακτίνας σφαίρας μέσα στην οποία πραγματοποιείται η αναζήτηση. Όσο μικρότερη είναι η ακτίνα της σφαίρας τόσο λιγότερα είναι τα πιθανά διανύσματα που εξετάζονται. Ωστόσο αν η σφαίρα που δημιουργείται δεν περικλείει κανένα διάνυσμα το ο αλγόριθμος επιβαρύνεται με επιπλέον βήματα προς τα πίσω διπλασιάζοντας την τιμή της ακτίνας έως ότου υπάρχουν πιθανά διανύσματα προς εξέταση. Στην ιδανική περίπτωση η σφαίρα θα πρέπει να περικλείει μόνο το διάνυσμα μέγιστης πιθανοφάνειας. Συνεπώς η μέγιστη αποδοτικότητα του αλγορίθμου του Sphere Decoder επιτυγχάνεται όταν η αρχική ακτίνα C που

επιλέγεται πλησιάζει την ευκλείδεια απόσταση του ληφθέντος σήματος \mathbf{y} από το γινόμενο του πίνακα \mathbf{H} με το διάνυσμα μέγιστης πιθανοφάνειας. $\mathbf{c} = \|\mathbf{y} - \mathbf{H}\mathbf{x}_{ML}\|^2$

Κεφάλαιο 5

Συμπεράσματα

Η τεχνολογία των MIMO συστημάτων βρίσκεται εδώ και αρκετό καιρό στο επίκεντρο του ενδιαφέροντος στον τομέα των ασύρματων επικοινωνιών. Η χρησιμοποίηση πολλαπλών κεραιών εκπομπής και λήψης εισάγουν νέες δυνατότητες στην ασύρματη μετάδοση δεδομένων. Οι τεχνικές διαφορισμού (diversity techniques) έχουν σαν αποτέλεσμα μεγάλη αύξηση της αξιοπιστίας του καναλιού. Η τεχνολογία χωρικής πολυπλεξίας (Spatial Multiplexing) δίνει τη δυνατότητα να επιτευχθούν πολύ υψηλές ταχύτητες μετάδοσης δεδομένων. Τα πλεονεκτήματα αυτά που επιφέρει η τεχνολογία MIMO προϋποθέτουν συγκεκριμένη διάταξη κεραιών και περιβάλλον πλούσιο σε σκεδαστές που θα επιτρέψει την μετατροπή των πολυδιαδρομικών συνιστωσών του σήματος σε ανεξάρτητες ροές δεδομένων. Σημαντικό ρόλο στην αξιοποίηση των δυνατοτήτων αυτών παίζει η τεχνική αποκωδικοποίησης του σήματος που λαμβάνει ο εκάστοτε δέκτης.

Η εργασία αυτή παρουσίασε τη λειτουργία έξι διαφορετικών δεκτών που χρησιμοποιούνται στα συστήματα MIMO Spatial Multiplexing. Η αξιολόγηση της απόδοσης των δεκτών πραγματοποιήθηκε μέσω προσομοιώσεων που υλοποιήθηκαν στο λογισμικό Matlab. Οι προσομοιώσεις αφορούν μετάδοση δεδομένων με τη χρήση

–QAM διαμόρφωσης μέσω καναλιού MIMO που υπόκειται σε flat fading διαλείψεις σύμφωνα με το μοντέλο Rayleigh. Οι δέκτες που μελετήθηκαν παρουσίασαν διαφορετικά επίπεδα αξιοπιστίας και διαφορετικά επίπεδα πολυπλοκότητας. Η τηλεπικοινωνιακές εφαρμογές ποικίλουν σε απαιτήσεις και οι διάφοροι δέκτες μπορούν να επιλεγούν ανάλογα με αυτές και ανάλογα με τους πόρους που διατίθενται για την υλοποίηση ενός MIMO συστήματος. Στο σημείο αυτό πρέπει να επισημάνουμε πως η μελέτη της μετάδοσης των δεδομένων που παρουσιάστηκε δεν περιλαμβάνει προ-επεξεργασία των δεδομένων η οποία μπορεί να οδηγήσει σε μεγαλύτερα επίπεδα αξιοπιστίας για την κάθε ζεύξη.

Όσο αναφορά στη μέγιστη αξιοπιστία που μπορεί να επιτευχθεί μέσω του δέκτη μέγιστης πιθανοφάνειας (ML decoder) που επιτυγχάνει τάξη διαφορισμού ίση με το πλήθος των κεραιών λήψης, φαίνεται πως το πρόβλημα της πολυπλοκότητας δεν είναι ανυπέρβλητο για την επίτευξή της. Ο δέκτης Sphere Decoder δείχνει να επιτυγχάνει την ίδια αξιοπιστία με ταυτόχρονη σημαντική μείωση της πολυπλοκότητας για υψηλά σχήματα διαμόρφωσης και μεγάλο αριθμό κεραιών λήψης.

Η έρευνα στο πεδίο των δεκτών MIMO συστημάτων χωρικής πολυπλεξίας δεν περιορίζεται στους δέκτες που παρουσιάστηκαν στην εργασία αυτή. Νέες τεχνικές δημιουργούνται, υπάρχουνες εξελίσσονται και συνδυασμός τεχνικών έρχονται να δώσουν λύσεις στα προβλήματα που ανακύπτουν με τη χρήση πολλαπλών κεραιών εκπομπής και λήψης.

Με την εξέλιξη αυτή των ασύρματων επικοινωνιών, στο μέλλον είναι δυνατό να μπορεί να παρέχεται ποιότητα υπηρεσίας μέσω ασύρματων ζεύξεων εφάμιλλη με αυτήν που παρέχεται από ενσύρματα μέσα σήμερα. Μια τέτοια εξέλιξη θα δώσει ακόμα περισσότερη ώθηση στον τομέα των επικοινωνιών καθώς η εγκατάσταση ασύρματων ζεύξεων είναι πολύ πιο απλή και προπάντων έχει πολύ μικρότερο κόστος.

Αναφορές

- [1] “ΑΣΥΡΜΑΤΕΣ ΕΠΙΚΟΙΝΩΝΙΕΣ ΚΑΙ ΔΙΚΤΥΑ” WILLIAM STALLINGS
Εκδόσεις Τζιόλα 2007
- [2] "BBC Measuring the Information Society 2010" (PDF). BBC News. 2010-07-09.
- [3] “ΣΥΣΤΗΜΑΤΑ ΚΙΝΗΤΩΝ ΕΠΙΚΟΙΝΩΝΙΩΝ” Α. Κανάτας Φ. Κωνσταντίνου Γ.
Πάντος Εκδόσεις Παπασωτηρίου 2008
- [4] “Fundamentals of Wireless Communication” David Tse Pramod Viswanath
Cambridge University Press 2005
- [5] “MIMO OOFDM Wireless Communications with Matlab” Yong Soo Cho,
Jaekwon Kim Won, Young Yang, Chung G. Kang
- [6]” MIMO System Technology for Wireless Communications” Edited by George
Tsoulos CRC Press Taylor & Francis Group 2006
- [7] “Space Division Multiplexing Algorithms” A. van Zelst, Electrotechnical
Conference, 2000. MELECON 2000. 10th Mediterranean

- [8] “Detection algorithm and initial laboratory results using V-BLAST space-time communication architecture” G.D. Golden, C.J. Foschini, R.A. Valenzuela and P.W. Wolniansky Electronics Letters Volume: 35 , Issue: 1
- [9] “Performance Analysis of MIMO System with Linear MMSE Receiver” Namshik Kim, Yusung Lee, and Hyuncheol Park Wireless Communications, IEEE Transactions on Volume: 7 , Issue: 11 , Part: 2
- [10] “ V-BLAST: An Architecture for Realizing Very High Data Rates Over the Rich-Scattering Wireless Channel” P. W. Wolniansky, G. J. Foschini, G. D. Golden, R. A. Valenzuela, Signals, Systems, and Electronics, 1998. ISSSE 98.
- [11] “Block-Iterative Generalized Decision Feedback Equalizers for Large MIMO Systems: Algorithm Design and Asymptotic Performance Analysis” Ying-Chang Liang, Sumei Sun, and Chin Keong Ho, Signal Processing, IEEE Transactions on Volume: 54 , Issue: 6 , Part: 1
- [12] “Performance analysis and comparison of $m \times n$ zero forcing and MMSE equalizer based receiver for mimo wireless channel” N. Sathish Kumar* and K. R. Shankar Kumar Songklanakar J. Sci. Technol. May - Jun. 2011
- [13] “A New OSIC Detection Algorithm for V-BLAST Systems” Zhang XiaoGe Xu ChengQi © 2007 IEEE
- [14] “Performance Analysis of Maximum Likelihood Detection in a MIMO Antenna System” Xu Zhu and Ross D. Murch, Senior Member, IEEE IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. 50, NO. 2, FEBRUARY 2002
- [15] “Τεχνικές και Αλγόριθμοι Επεξεργασίας Σήματος για τη Μείωση της Πολυπλοκότητας του Υλικού σε Ασύρματα Συστήματα Επικοινωνιών Πολλαπλών

Εισόδων – Πολλαπλών Εξόδων” Διδακτορική Διατριβή του Παναγιώτη – Θεόφιλου
Ν. Θεοφιλάκου Πειραιάς Ιούνιος 2009

[16] “On the decoding order of MIMO maximum-likelihood sphere decoder: linear and non-linear receivers” Kai-Kit Wong, Arogyaswami Paulraj Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th

[17] “A Sphere Decoding Algorithm for MIMO Channels” Chin-Yun Hung; Tzu-Hsien Sang
Signal Processing and Information Technology, 2006 IEEE

[18] “On the Sphere-Decoding Algorithm I.” Expected Complexity Babak Hassibi and Haris Vikalo IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 53, NO. 8, AUGUST 2005

[19] “On the Sphere-Decoding Algorithm II. Generalizations, Second-Order Statistics, and Applications to Communications Haris Vikalo and Babak Hassibi IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 53, NO. 8, AUGUST 2005

[20] “On the decoding order of MIMO maximum-likelihood sphere decoder: linear and non-linear receivers” Kai-Kit Wong and Arogyaswami Paulraj Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th

Παράρτημα

Κώδικες Matlab

Χωρητικότητα MIMO καναλιού

MIMO_Channel_Ergodig_Capacity_vs_SNR.m

```
% MIMO_Channel_Ergodig_Capacity_vs_SNR.m
clear all, close all
SNR_dB=[0:30]; SNR_linear=10.^(SNR_dB/10);
N_iter=1000; sq2 = sqrt(0.5); % 1000 υλοποιήσεις για τον υπολογισμό
μέσης τιμής
for Icase=1:5
    if Icase==1, nT=1; nR=1; % 1x1
    elseif Icase==2, nT=1; nR=2; % 1x2
    elseif Icase==3, nT=2; nR=1; % 2x1
    elseif Icase==4, nT=2; nR=2; % 2x2
    else nT=4; nR=4; %4x4
    end
    n=min(nT,nR); I = eye(n);
    C(Icase,:) = zeros(1,length(SNR_dB));
    for iter=1:N_iter
        H = sq2*(randn(nR,nT)+1i*randn(nR,nT)); %Δημιουργία τυχαίου
        καναλιού.
        if nR>=nT, HH = H'*H; else HH = H*H'; end
        for i=1:length(SNR_dB)
            C(Icase,i) = C(Icase,i)+log2(real(det(I+SNR_linear(i)/nT*HH)));
        end
    end
end
C = C/N_iter;
plot(SNR_dB,C(1,:), '-k', 'LineWidth', 1.5);
hold on, plot(SNR_dB,C(2,:), '-g+', 'LineWidth', 1.5);
hold on, plot(SNR_dB,C(3,:), '-.', 'LineWidth', 1.5);
hold on, plot(SNR_dB,C(4,:), '-co', 'LineWidth', 1.5);
hold on, plot(SNR_dB,C(5,:), '-rd', 'LineWidth', 1.5);
legend('SISO Channel', 'SIMO 1x2', 'MISO 2x1', 'MIMO 2x2', 'MIMO
4x4', 'Location', 'NorthWest')
title('Mimo Channel Ergodic Capacity')
xlabel('SNR[dB]'); ylabel('bps/Hz');
```

Correlated_vs_Uncorrelated_MIMO_Channel_Capacity.m

```
% Correlated_vs_Uncorrelated_MIMO_Channel_Capacity.m
clear all, close all;
```

```

SNR_dB=[0:30]; SNR_linear=10.^(SNR_dB/10.);
N_iter=1000; N_SNR=length(SNR_dB); % 1000 υλοποιήσεις για τον
υπολογισμό μέσης τιμής
nT=4; nR=4; n=min(nT,nR); % 4 κεραίες δέκτη + 4 κεραίες πομπού
I = eye(n); sq2=sqrt(0.5);
Rtx=[1 0.76*exp(0.17j*pi) 0.43*exp(0.35j*pi) 0.25*exp(0.53j*pi); %
πίνακας συσχέτισης
0.76*exp(-0.17j*pi) 1 0.76*exp(0.17j*pi) 0.43*exp(0.35j*pi);
0.43*exp(-0.35j*pi) 0.76*exp(-0.17j*pi) 1 0.76*exp(0.17j*pi);
0.25*exp(-0.53j*pi) 0.43*exp(-0.35j*pi) 0.76*exp(-0.17j*pi) 1];
Rrx=[1 0.76*exp(0.17j*pi) 0.43*exp(0.35j*pi) 0.25*exp(0.53j*pi); %
πίνακας συσχέτισης
0.76*exp(-0.17j*pi) 1 0.76*exp(0.17j*pi) 0.43*exp(0.35j*pi);
0.43*exp(-0.35j*pi) 0.76*exp(-0.17j*pi) 1 0.76*exp(0.17j*pi);
0.25*exp(-0.53j*pi) 0.43*exp(-0.35j*pi) 0.76*exp(-0.17j*pi) 1];
C_uncorr=zeros(1,N_SNR); C_corr=zeros(1,N_SNR);
for iter=1:N_iter
Hw = sq2*(randn(4,4) + 1i*randn(4,4)); % Ιδανικό Κανάλι
H = Rrx^(1/2)*Hw*Rtx^(1/2);
tmp1 = Hw'*Hw/nT; tmp2 = H'*H/nT;
for i=1:N_SNR
C_uncorr(i) = C_uncorr(i) + log2(det(I+SNR_linear(i)*tmp1));
C_corr(i) = C_corr(i) + log2(det(I+SNR_linear(i)*tmp2));
end
end
C_uncorr = real(C_uncorr)/N_iter;
C_corr = real(C_corr)/N_iter;
plot(SNR_dB,C_uncorr,'-kd','LineWidth',1.5);
hold on, plot(SNR_dB,C_corr,'-ro','LineWidth',1.5);
legend('Uncorrelated Channel Capacity','Correlated Channel
Capacity','Location','NorthWest')
title('Correlated vs Uncorrelated MIMO Channel Capacity')
xlabel('SNR[dB]'); ylabel('bps/Hz');

```

OpenLoop_vs_ClosedLoop_Channel_Capacity.m

```

clear all, close all;
SNR_dB=[0:30]; SNR_linear=10.^(SNR_dB/10.);
N_iter=1000; N_SNR=length(SNR_dB); % 1000 υλοποιήσεις για τον
υπολογισμό μέσης τιμής
nT=4; nR=4; n=min(nT,nR); % 4 κεραίες δέκτη + 4 κεραίες πομπού
I = eye(n); sq2=sqrt(0.5);
Rtx=[1 0.76*exp(0.17j*pi) 0.43*exp(0.35j*pi) 0.25*exp(0.53j*pi); %
πίνακας συσχέτισης
0.76*exp(-0.17j*pi) 1 0.76*exp(0.17j*pi) 0.43*exp(0.35j*pi);
0.43*exp(-0.35j*pi) 0.76*exp(-0.17j*pi) 1 0.76*exp(0.17j*pi);
0.25*exp(-0.53j*pi) 0.43*exp(-0.35j*pi) 0.76*exp(-0.17j*pi) 1];
Rrx=[1 0.76*exp(0.17j*pi) 0.43*exp(0.35j*pi) 0.25*exp(0.53j*pi); %
πίνακας συσχέτισης
0.76*exp(-0.17j*pi) 1 0.76*exp(0.17j*pi) 0.43*exp(0.35j*pi);
0.43*exp(-0.35j*pi) 0.76*exp(-0.17j*pi) 1 0.76*exp(0.17j*pi);
0.25*exp(-0.53j*pi) 0.43*exp(-0.35j*pi) 0.76*exp(-0.17j*pi) 1];
C_OL=zeros(1,length(SNR_dB));

```

```

C_CL=zeros(1,length(SNR_dB));
for iter=1:N_iter
Hw = sq2*(randn(4,4) + 1i*randn(4,4));
H = Rrx^(1/2)*Hw*Rtx^(1/2);
tmp = H'*H/nT;
SV = svd(H'*H);
for i=1:length(SNR_dB)
C_OL(i) = C_OL(i)+log2(det(I+SNR_linear(i)*tmp));
Gamma = Water_Pouring(SV,SNR_linear(i),nT);
C_CL(i) = C_CL(i) +
log2(det(I+SNR_linear(i)/nT*diag(Gamma)*diag(SV)));
end
end
C_OL = real(C_OL)/N_iter;
C_CL = real(C_CL)/N_iter;
plot(SNR_dB, C_OL, '-o', SNR_dB, C_CL, '-k');
legend('Open Loop', 'Closed Loop', 'Location', 'NorthWest')
title('Open vs Closed Loop MIMO Channel Capacity')
xlabel('SNR[dB]'); ylabel('bps/Hz');

```

```

% Water_Pouring.m
function [Gamma]=Water_Pouring(Lamda,SNR,nT)
Gamma=zeros(1,length(Lamda));
r=length(Lamda);
index=[1:r];
index_temp=index;
p=1;
while p<r+1
irp=1:r-p+1
temp = sum(1./Lamda(index_temp(irp)))
mu = nT/(r-p+1)*(1+1/SNR*temp)
Gamma(index_temp(irp)) = mu - nT./(SNR*Lamda(index_temp(irp)))
    if min(Gamma(index_temp))<0
        i=find(Gamma==min(Gamma));
        ii=find(index_temp==i);
        index_temp2=[index_temp([1:ii-1]) index_temp([ii+1:end])];
        clear index_temp;
        index_temp=index_temp2;
        p=p+1
        clear Gamma;
    else
        p=r+1;
    end
end
Gamma_t=zeros(1,length(Lamda));
Gamma_t(index_temp)=Gamma(index_temp);
Gamma=Gamma_t;

```

Προσομοίωση Λειτουργίας Δεκτών

Zero Forcing Receiver

```
%mimo_ZF_RECEIVER.m

Timeslots=100000;
NT=2; NR=2;
M=2; %Constellation Size
b = log2(M); % Number of bits per symbol

QAM_nums=[0:M-1];
QAM_syms=modulate(modem.qammod(M),QAM_nums);
scale = modnorm(QAM_syms,'avpow',1); %Power Normalization Factor

EbNo_dB=[0:2:20];
SNR_dB=EbNo_dB+10*log10(b);

%Δημιουργία διανύσματος των προς μετάδοση bits
%Διαμόρφωση των προς μετάδοση bits χρησιμοποιώντας M-QAM
bitstream=randint(Timeslots*b*NT,1);
S=reshape(bitstream,b,length(bitstream)/b);
S=S.';
xsym = bi2de(S,'left-msb');
xsym = modulate(modem.qammod(M),xsym);
xsym=scale*xsym;
x=reshape(xsym,NT,Timeslots); % Διαμόρφωση πίνακα των προς μετάδοση
στοιχείων

Y=zeros(NR,Timeslots);
Ynoisy=zeros(NR,Timeslots);

No=(randn(NR,Timeslots) + 1i*randn(NR,Timeslots));
H=(randn(NR,NT,Timeslots)+1i*randn(NR,NT,Timeslots))/sqrt(2); %
Δημιουργία MIMO καναλιού για κάθε χρονοσχισημή

for i=1:length(EbNo_dB)
    SNR = 10^(SNR_dB(i)/10);
    sigma = sqrt(1/(2*SNR));
    noise = sigma * No;

    for T=1:Timeslots
        Y(:,T)=H(:,:,T)*x(:,T);
        Ynoisy(:,T)=Y(:,T)+noise(:,T); % Προσθήκη θορύβου
    end
end
W_ZF=zeros(NT,NR,Timeslots);
Yest_zf=zeros(NT,Timeslots);
for T=1:Timeslots
    %Δημιουργία του πίνακα αναστροφής της επίδρασης του καναλιού.
    W_ZF(:,:,T)=(H(:,:,T)'*H(:,:,T))\H(:,:,T)'; % Pseudoinverse
Matrix for ZF decoder
    for nt=1:NT
```



```

        Yest_zf(nt,T)=W_ZF(nt,:,T)*Ynoisy(:,T); % Ανακατασκευή των
μεταδοθέντων συμβόλων
    end
end
%Μετατροπή συμβόλων σε bits
RecSymbols_zf=reshape(Yest_zf,prod(size(Yest_zf)),1);
RecSymbols_zf=(1/scale)*RecSymbols_zf;
Recy_zf = demodulate(modem.qamdemod(M),RecSymbols_zf);
Recy_zf= de2bi(Recy_zf,'left-msb'); % Convert integers to bits.
Recy_zf=Recy_zf.';
Rbitstream_zf=reshape(Recy_zf,prod(size(Recy_zf)),1);

% Έλεγχος Σφαλμάτων
[NUM_ERRORS_zf(i) BER_zf(i)]=biterr(bitstream,Rbitstream_zf)
end
figure
semilogy(EbNo_dB,BER_zf,'-k','LineWidth', 1.5)
legend('ZF','Location','SouthWest')
xlabel('Eb/No[dB]'); ylabel('BER');
title('MIMO ZF receiver performance')
grid on
axis([0 20 10^-4,10^0])

```

```

%MQAM_ZF_RECEIVER.m

Timeslots=100000;
NT=2; NR=2;
for Icase=1:3
if Icase==1, M=2;
elseif Icase==2, M=16;
else M=64;
end

b = log2(M); % Number of bits per symbol

QAM_nums=[0:M-1];
QAM_syms=modulate(modem.qammod(M),QAM_nums);
scale = modnorm(QAM_syms,'avpow',1); %Power Normalization Factor

EbNo_dB=[0:2:20];
SNR_dB=EbNo_dB+10*log10(b);

%Δημιουργία διανύσματος των προς μετάδοση bits
%Διαμόρφωση των προς μετάδοση bits χρησιμοποιώντας M-QAM
bitstream=randint(Timeslots*b*NT,1);
S=reshape(bitstream,b,length(bitstream)/b);
S=S.';
xsym = bi2de(S,'left-msb');
xsym = modulate(modem.qammod(M),xsym);
xsym=scale*xsym;
x=reshape(xsym,NT,Timeslots); % Διαμόρφωση πίνακα των προς μετάδοση
στοιχείων

```

```

Y=zeros(NR,Timeslots);
Ynoisy=zeros(NR,Timeslots);

No=(randn(NR,Timeslots) + 1i*randn(NR,Timeslots));
H=(randn(NR,NT,Timeslots)+1i*randn(NR,NT,Timeslots))/sqrt(2); %
Δημιουργία MIMO καναλιού για κάθε χρονοσχισμή

for i=1:length(EbNo_dB)
    SNR = 10^(SNR_dB(i)/10);
    sigma = sqrt(1/(2*SNR));
    noise = sigma * No;

    for T=1:Timeslots
        Y(:,T)=H(:, :, T)*x(:, T);
        Ynoisy(:, T)=Y(:, T)+noise(:, T); % Προσθήκη θορύβου
    end
    W_ZF=zeros(NT,NR,Timeslots);
    Yest_zf=zeros(NT,Timeslots);
    for T=1:Timeslots
        %Δημιουργία του πίνακα αναστροφής της επίδρασης του καναλιού.
        W_ZF(:, :, T)=(H(:, :, T)'*H(:, :, T))\H(:, :, T)'; % Pseudoinverse
Matrix for ZF decoder
        for nt=1:NT
            Yest_zf(nt, T)=W_ZF(nt, :, T)*Ynoisy(:, T); % Ανακατασκευή των
μεταδοθέντων συμβόλων
        end
    end
    %Μετατροπή συμβόλων σε bits
RecSymbols_zf=reshape(Yest_zf,prod(size(Yest_zf)),1);
RecSymbols_zf=(1/scale)*RecSymbols_zf;
Recy_zf = demodulate(modem.qamdemod(M),RecSymbols_zf);
Recy_zf= de2bi(Recy_zf, 'left-msb'); % Convert integers to bits.
Recy_zf=Recy_zf.';
Rbitstream_zf=reshape(Recy_zf,prod(size(Recy_zf)),1);

% Έλεγχος Σφαλμάτων
[NUM_ERRORS_zf(Icase,i)
BER_zf(Icase,i)]=biterr(bitstream,Rbitstream_zf)
end
end
figure
semilogy(EbNo_dB,BER_zf(1,:), '-+', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB,BER_zf(2,:), 'k-^', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB,BER_zf(3,:), 'ro-', 'LineWidth', 1.5)
legend('2QAM', '16QAM', '64QAM', 'Location', 'SouthWest')
xlabel('Eb/No[dB]'); ylabel('BER');
title('MIMO ZF receiver performance for various MQAM modulation
schemes')
grid on
axis([0 20 10^-4,10^0])

```

```

%ZF_RECEIVER_antenna_configuration.m

Timeslots=100000;
M=2;
b = log2(M); % Number of bits per symbol

QAM_nums=[0:M-1];
QAM_syms=modulate(modem.qammod(M),QAM_nums);
scale = modnorm(QAM_syms,'avpow',1); %Power Normalization Factor

EbNo_dB=[0:2:20];
SNR_dB=EbNo_dB+10*log10(b);

for Icase=1:4
if Icase==1, NT=2; NR=2;
elseif Icase==2, NT=2; NR=3;
elseif Icase==3, NT=2; NR=4;
else NT=4; NR=4;
end

%Δημιουργία διανύσματος των προς μετάδοση bits
%Διαμόρφωση των προς μετάδοση bits χρησιμοποιώντας M-QAM
bitstream=randint(Timeslots*b*NT,1);
S=reshape(bitstream,b,length(bitstream)/b);
S=S.';
xsym = bi2de(S,'left-msb');
xsym = modulate(modem.qammod(M),xsym);
xsym=scale*xsym;
x=reshape(xsym,NT,Timeslots); % Διαμόρφωση πίνακα των προς μετάδοση
στοιχείων

Y=zeros(NR,Timeslots);
Ynoisy=zeros(NR,Timeslots);

No=(randn(NR,Timeslots) + 1i*randn(NR,Timeslots));
H=(randn(NR,NT,Timeslots)+1i*randn(NR,NT,Timeslots))/sqrt(2); %
Δημιουργία MIMO καναλιού για κάθε χρονοσχισημή

for i=1:length(EbNo_dB)
    SNR = 10^(SNR_dB(i)/10);
    sigma = sqrt(1/(2*SNR));
    noise = sigma * No;

    for T=1:Timeslots
        Y(:,T)=H(:, :, T)*x(:,T);
        Ynoisy(:,T)=Y(:,T)+noise(:,T); % Προσθήκη Θορύβου
    end
end
W_ZF=zeros(NT,NR,Timeslots);
Yest_zf=zeros(NT,Timeslots);
for T=1:Timeslots
    %Δημιουργία του πίνακα αναστροφής της επίδρασης του καναλιού.

```

```

        W_ZF(:, :, T) = (H(:, :, T)' * H(:, :, T)) \ H(:, :, T)'; % Pseudoinverse
Matrix for ZF decoder
        for nt=1:NT
            Yest_zf(nt, T) = W_ZF(nt, :, T) * Ynoisy(:, T); % Ανακατασκευή των
μεταδοθέντων συμβόλων
        end
    end
    %Μετατροπή συμβόλων σε bits
RecSymbols_zf = reshape(Yest_zf, prod(size(Yest_zf)), 1);
RecSymbols_zf = (1/scale) * RecSymbols_zf;
Recy_zf = demodulate(modem.qamdemod(M), RecSymbols_zf);
Recy_zf = de2bi(Recy_zf, 'left-msb'); % Convert integers to bits.
Recy_zf = Recy_zf.';
Rbitstream_zf = reshape(Recy_zf, prod(size(Recy_zf)), 1);

% Έλεγχος Σφαλμάτων
[ NUM_ERRORS_zf(Icase, i)
BER_zf(Icase, i) ] = biterr(bitstream, Rbitstream_zf)
end
end
figure
semilogy(EbNo_dB, BER_zf(1, :), '-+', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB, BER_zf(2, :), 'k-^', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB, BER_zf(3, :), 'ro-', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB, BER_zf(4, :), 'md-', 'LineWidth', 1.5)
legend('Nt=2 Nr=2', 'Nt=2 Nr=3', 'Nt=2 Nr=4', 'Nt=4
Nr=4', 'Location', 'SouthWest')
xlabel('Eb/No[dB]'); ylabel('BER');
title('MIMO ZF receiver performance for various antenna configuration
schemes')
grid on
axis([0 20 10^-5, 10^0])

```

MMSE Receiver

```

% mimo_MMSE_RECEIVER.m

Timeslots = 100000;
NT = 2; NR = 2;
M = 16; %Constellation Size
b = log2(M); % Number of bits per symbol
QAM_nums = [0:M-1];
QAM_syms = modulate(modem.qammod(M), QAM_nums);
scale = modnorm(QAM_syms, 'avpow', 1);
EbNo_dB = [0:2:20];
SNR_dB = EbNo_dB + 10 * log10(b);

% Δημιουργία διανύσματος των προς μετάδοση bits
% Διαμόρφωση των προς μετάδοση bits χρησιμοποιώντας M-QAM
bitstream = randint(Timeslots * b * NT, 1);
S = reshape(bitstream, b, length(bitstream) / b);
S = S.';
xsym = bi2de(S, 'left-msb');

```

```

xsym = modulate(modem.qammod(M), xsym);
xsym=scale*xsym;
x=reshape(xsym,NT,Timeslots); % Διαμόρφωση πίνακα των προς μετάδοση
στοιχείων

Y=zeros(NR,Timeslots);
Ynoisy=zeros(NR,Timeslots);

No=(randn(NR,Timeslots) + 1i*randn(NR,Timeslots));
H=(randn(NR,NT,Timeslots)+1i*randn(NR,NT,Timeslots))/sqrt(2); %
Δημιουργία MIMO καναλιού για κάθε χρονοσχισμή

for i=1:length(EbNo_dB)
    SNR = 10^(SNR_dB(i)/10);
    sigma = sqrt(1/(2*SNR));
    noise = sigma * No;

    for T=1:Timeslots
        Y(:,T)=H(:, :, T)*x(:,T);
        Ynoisy(:,T)=Y(:,T)+noise(:,T);% Προσθήκη Θορύβου
    end
    W_MMSE=zeros(NT,NR,Timeslots);
    Yest_mmse=zeros(NT,Timeslots);
    for T=1:Timeslots
        %Δημιουργία του πίνακα αναστροφής της επίδρασης του καναλιού.
        W_MMSE(:, :, T)=(H(:, :, T)'*H(:, :, T)+(sigma^2)*eye(NT))\H(:, :, T)';
    %MMSE Filter Construction
        for nt=1:NT
            Yest_mmse(nt,T)=W_MMSE(nt, :, T)*Ynoisy(:,T);
        end
    end

    %Μετατροπή συμβόλων σε bits
    RecSymbols_mmse=reshape(Yest_mmse,prod(size(Yest_mmse)),1);
    RecSymbols_mmse=(1/scale)*RecSymbols_mmse;
    Recy_mmse = demodulate(modem.qamdemod(M), RecSymbols_mmse);
    Recy_mmse= de2bi(Recy_mmse, 'left-msb'); % Convert integers to bits.
    Recy_mmse=Recy_mmse.';
    Rbitstream_mmse=reshape(Recy_mmse,prod(size(Recy_mmse)),1);

    % Έλεγχος Σφαλμάτων
    [NUM_ERRORS_mmse(i) BER_mmse(i)]=biterr(bitstream,Rbitstream_mmse)
end
figure
semilogy(EbNo_dB, BER_mmse, '-k', 'LineWidth', 1.5)
legend('MMSE', 'Location', 'SouthWest')
xlabel('Eb/No[dB]'); ylabel('BER');
title('MIMO MMSE receiver perfomance')
grid on
axis([0 20 10^-5,10^0])

```

```

%MQAM_MMSE_RECEIVER.m

Timeslots=100000;
NT=2; NR=2;
for Icase=1:3
if Icase==1, M=2;
elseif Icase==2, M=16;
else M=64;
end

b = log2(M); % Number of bits per symbol

QAM_nums=[0:M-1];
QAM_syms=modulate(modem.qammod(M),QAM_nums);
scale = modnorm(QAM_syms,'avpow',1); %Power Normalization Factor

EbNo_dB=[0:2:20];
SNR_dB=EbNo_dB+10*log10(b);

%Δημιουργία διανύσματος των προς μετάδοση bits
%Διαμόρφωση των προς μετάδοση bits χρησιμοποιώντας M-QAM
bitstream=randint(Timeslots*b*NT,1);
S=reshape(bitstream,b,length(bitstream)/b);
S=S.';
xsym = bi2de(S,'left-msb');
xsym = modulate(modem.qammod(M),xsym);
xsym=scale*xsym;
x=reshape(xsym,NT,Timeslots); % Διαμόρφωση πίνακα των προς μετάδοση
στοιχείων

Y=zeros(NR,Timeslots);
Ynoisy=zeros(NR,Timeslots);

No=(randn(NR,Timeslots) + 1i*randn(NR,Timeslots));
H=(randn(NR,NT,Timeslots)+1i*randn(NR,NT,Timeslots))/sqrt(2); %
Δημιουργία MIMO καναλιού για κάθε χρονοσχισμή

for i=1:length(EbNo_dB)
SNR = 10^(SNR_dB(i)/10);
sigma = sqrt(1/(2*SNR));
noise = sigma * No;

for T=1:Timeslots
Y(:,T)=H(:, :, T)*x(:,T);
Ynoisy(:,T)=Y(:,T)+noise(:,T);% Προσθήκη θορύβου
end
W_MMSE=zeros(NT,NR,Timeslots);
Yest_mmse=zeros(NT,Timeslots);
for T=1:Timeslots
%Δημιουργία του πίνακα αναστροφής της επίδρασης του καναλιού.

```

```

        W_MMSE(:, :, T) = (H(:, :, T)' * H(:, :, T) + (sigma^2) * eye(NT)) \ H(:, :, T)';
%MMSE Filter Construction
    for nt=1:NT
        Yest_mmse(nt, T) = W_MMSE(nt, :, T) * Ynoisy(:, T);
    end
end

%Μετατροπή συμβόλων σε bits
RecSymbols_mmse = reshape(Yest_mmse, prod(size(Yest_mmse)), 1);
RecSymbols_mmse = (1/scale) * RecSymbols_mmse;
Recy_mmse = demodulate(modem.qamdemod(M), RecSymbols_mmse);
Recy_mmse = de2bi(Recy_mmse, 'left-msb'); % Convert integers to bits.
Recy_mmse = Recy_mmse.';
Rbitstream_mmse = reshape(Recy_mmse, prod(size(Recy_mmse)), 1);

% Έλεγχος Σφαλμάτων
[ NUM_ERRORS_mmse(Icase, i)
  BER_mmse(Icase, i) ] = biterr(bitstream, Rbitstream_mmse)
end
end
figure
semilogy(EbNo_dB, BER_mmse(1, :), '--', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB, BER_mmse(2, :), 'k-^', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB, BER_mmse(3, :), 'ro-', 'LineWidth', 1.5)
legend('2QAM', '16QAM', '64QAM', 'Location', 'SouthWest')
xlabel('Eb/No[dB]'); ylabel('BER');
title('MIMO MMSE receiver performance for various MQAM modulation
schemes')
grid on
axis([0 20 10^-4, 10^0])

```

```

%MMSE_RECEIVER_antenna_configuration.m

Timeslots=100000;
M=2;
b = log2(M); % Number of bits per symbol
QAM_nums=[0:M-1];
QAM_syms=modulate(modem.qammod(M), QAM_nums);
scale = modnorm(QAM_syms, 'avpow', 1); %Power Normalization Factor
EbNo_dB=[0:2:20];
SNR_dB=EbNo_dB+10*log10(b);

for Icase=1:3
    if Icase==1, NT=2; NR=2;
    elseif Icase==2, NT=2; NR=3;
    elseif Icase==3, NT=2; NR=4;
    else NT=4; NR=4;
    end

%Δημιουργία διανύσματος των προς μετάδοση bits
%Διαμόρφωση των προς μετάδοση bits χρησιμοποιώντας M-QAM
bitstream=randint(Timeslots*b*NT, 1);

```

```

S=reshape(bitstream,b,length(bitstream)/b);
S=S.';
xsym = bi2de(S,'left-msb');
xsym = modulate(modem.qammod(M),xsym);
xsym=scale*xsym;
x=reshape(xsym,NT,Timeslots); % Διαμόρφωση πίνακα των προς μετάδοση
στοιχείων

```

```

Y=zeros(NR,Timeslots);
Ynoisy=zeros(NR,Timeslots);

No=(randn(NR,Timeslots) + li*randn(NR,Timeslots));
H=(randn(NR,NT,Timeslots)+li*randn(NR,NT,Timeslots))/sqrt(2); %
Δημιουργία MIMO καναλιού για κάθε χρονοσχιμή

for i=1:length(EbNo_dB)
    SNR = 10^(SNR_dB(i)/10);
    sigma = sqrt(1/(2*SNR));
    noise = sigma * No;

    for T=1:Timeslots
        Y(:,T)=H(:,:,T)*x(:,T);
        Ynoisy(:,T)=Y(:,T)+noise(:,T);% Προσθήκη θορύβου
    end
    W_MMSE=zeros(NT,NR,Timeslots);
    Yest_mmse=zeros(NT,Timeslots);
    for T=1:Timeslots
        %Δημιουργία του πίνακα αναστροφής της επίδρασης του καναλιού.
        W_MMSE(:,:,T)=(H(:,:,T)'*H(:,:,T)+(sigma^2)*eye(NT))\H(:,:,T)';
    %MMSE Filter Construction
        for nt=1:NT
            Yest_mmse(nt,T)=W_MMSE(nt,:,T)*Ynoisy(:,T);
        end
    end

    %Μετατροπή συμβόλων σε bits
    RecSymbols_mmse=reshape(Yest_mmse,prod(size(Yest_mmse)),1);
    RecSymbols_mmse=(1/scale)*RecSymbols_mmse;
    Recy_mmse = demodulate(modem.qamdemod(M),RecSymbols_mmse);
    Recy_mmse= de2bi(Recy_mmse,'left-msb'); % Convert integers to bits.
    Recy_mmse=Recy_mmse.';
    Rbitstream_mmse=reshape(Recy_mmse,prod(size(Recy_mmse)),1);

    % Έλεγχος Σφαλμάτων
    [NUM_ERRORS_mmse(Icase,i)
    BER_mmse(Icase,i)]=biterr(bitstream,Rbitstream_mmse)
end
end
figure
semilogy(EbNo_dB,BER_mmse(1,:), '-+', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB,BER_mmse(2,:), 'k-^', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB,BER_mmse(3,:), 'ro-', 'LineWidth', 1.5)

```



```

hold on, semilogy(EbNo_dB,BER_mmse(4,:), 'md-', 'LineWidth', 1.5)
legend('Nt=2 Nr=2', 'Nt=2 Nr=3', 'Nt=2 Nr=4', 'Nt=4
Nr=4', 'Location', 'SouthWest')
xlabel('Eb/No[dB]'); ylabel('BER');
title('MIMO MMSE receiver performance for various antenna
configuration schemes')
grid on
axis([0 20 10^-5,10^0])

```

```

%ZF_vs_MMSE.m
Timeslots=100000;
NT=2; NR=2;
M=2; %Constellation Size
b = log2(M); % Number of bits per symbol

QAM_nums=[0:M-1];
QAM_syms=modulate(modem.qammod(M),QAM_nums);
scale = modnorm(QAM_syms,'avpow',1); %Power Normalization Factor

EbNo_dB=[0:2:20];
SNR_dB=EbNo_dB+10*log10(b);

%Δημιουργία διανύσματος των προς μετάδοση bits
%Διαμόρφωση των προς μετάδοση bits χρησιμοποιώντας M-QAM
bitstream=randint(Timeslots*b*NT,1);
S=reshape(bitstream,b,length(bitstream)/b);
S=S.';
xsym = bi2de(S,'left-msb');
xsym = modulate(modem.qammod(M), xsym);
xsym=scale*xsym;
x=reshape(xsym,NT,Timeslots); % Διαμόρφωση πίνακα των προς μετάδοση
στοιχείων

Y=zeros(NR,Timeslots);
Ynoisy=zeros(NR,Timeslots);

No=(randn(NR,Timeslots) + 1i*randn(NR,Timeslots));
H=(randn(NR,NT,Timeslots)+1i*randn(NR,NT,Timeslots))/sqrt(2); %
Δημιουργία MIMO καναλιού για κάθε χρονοσχιμή

for i=1:length(EbNo_dB)
    SNR = 10^(SNR_dB(i)/10);
    sigma = sqrt(1/(2*SNR));
    noise = sigma * No;

    for T=1:Timeslots
        Y(:,T)=H(:, :, T)*x(:,T);
        Ynoisy(:,T)=Y(:,T)+noise(:,T); % Προσθήκη θορύβου
    end
end
W_ZF=zeros(NT,NR,Timeslots);
Yest_zf=zeros(NT,Timeslots);

```

```

    for T=1:Timeslots
        %Δημιουργία του πίνακα αναστροφής της επίδρασης του καναλιού.
        W_ZF(:, :, T)=(H(:, :, T)'*H(:, :, T))\H(:, :, T)'; % Pseudoinverse
Matrix for ZF decoder
        for nt=1:NT
            Yest_zf(nt, T)=W_ZF(nt, :, T)*Ynoisy(:, T); % Ανακατασκευή των
μεταδοθέντων συμβόλων
        end
        end
        %Μετατροπή συμβόλων σε bits
RecSymbols_zf=reshape(Yest_zf,prod(size(Yest_zf)),1);
RecSymbols_zf=(1/scale)*RecSymbols_zf;
Recy_zf = demodulate(modem.qamdemod(M),RecSymbols_zf);
Recy_zf= de2bi(Recy_zf, 'left-msb'); % Convert integers to bits.
Recy_zf=Recy_zf.';
Rbitstream_zf=reshape(Recy_zf,prod(size(Recy_zf)),1);

% Έλεγχος Σφαλμάτων
[NUM_ERRORS_zf(i) BER_zf(i)]=biterr(bitstream,Rbitstream_zf)

W_MMSE=zeros(NT,NR,Timeslots);
Yest_mmse=zeros(NT,Timeslots);
    for T=1:Timeslots
        %Δημιουργία του πίνακα αναστροφής της επίδρασης του καναλιού.
        W_MMSE(:, :, T)=(H(:, :, T)'*H(:, :, T)+(sigma^2)*eye(NT))\H(:, :, T)';
%MMSE Filter Construction
        for nt=1:NT
            Yest_mmse(nt, T)=W_MMSE(nt, :, T)*Ynoisy(:, T);
        end
        end
        %Μετατροπή συμβόλων σε bits
RecSymbols_mmse=reshape(Yest_mmse,prod(size(Yest_mmse)),1);
RecSymbols_mmse=(1/scale)*RecSymbols_mmse;
Recy_mmse = demodulate(modem.qamdemod(M),RecSymbols_mmse);
Recy_mmse= de2bi(Recy_mmse, 'left-msb'); % Convert integers to bits.
Recy_mmse=Recy_mmse.';
Rbitstream_mmse=reshape(Recy_mmse,prod(size(Recy_mmse)),1);

% Έλεγχος Σφαλμάτων
[NUM_ERRORS_mmse(i) BER_mmse(i)]=biterr(bitstream,Rbitstream_mmse)
end
figure
semilogy(EbNo_dB,BER_zf, '-k', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB,BER_mmse, '-o', 'LineWidth', 1.5)
legend('ZF', 'MMSE', 'Location', 'SouthWest')
xlabel('Eb/No[dB]'); ylabel('BER');
title('MIMO ZF VS MMSE receiver perfomance')
grid on
axis([0 20 10^-4,10^0])

```

OSIC Zero Forcing Receiver

```
%mimo_OSIC_ZF_RECEIVER.m

Timeslots=100;
NT=2; NR=2;
M=2; %Constellation Size
b = log2(M); % Number of bits per symbol
QAM_nums=[0:M-1];
QAM_syms=modulate(modem.qammod(M),QAM_nums);
scale = modnorm(QAM_syms,'avpow',1);
EbNo_dB=[0:2:20];
SNR_dB=EbNo_dB+10*log10(b);

%Δημιουργία διανύσματος των προς μετάδοση bits
%Διαμόρφωση των προς μετάδοση bits χρησιμοποιώντας M-QAM
bitstream=randint(Timeslots*b*NT,1);
S=reshape(bitstream,b,length(bitstream)/b);
S=S.';
xsym = bi2de(S,'left-msb');
xsym = modulate(modem.qammod(M),xsym);
xsym=scale*xsym;
x=reshape(xsym,NT,Timeslots); % Διαμόρφωση πίνακα των προς μετάδοση
στοιχείων

Y=zeros(NR,Timeslots);
Ynoisy=zeros(NR,Timeslots);

No=(randn(NR,Timeslots) + 1i*randn(NR,Timeslots));
H=(randn(NR,NT,Timeslots)+1i*randn(NR,NT,Timeslots))/sqrt(2); %
Δημιουργία MIMO καναλιού για κάθε χρονοσχισμή

for l=1:length(EbNo_dB)
    SNR = 10^(SNR_dB(l)/10);
    sigma = sqrt(1/(2*SNR));
    noise = sigma * No;

    for T=1:Timeslots
        Y(:,T)=H(:, :, T)*x(:, T);
        Ynoisy(:,T)=Y(:,T)+noise(:,T); % Προσθήκη θορύβου
    end

    W_osiczf=zeros(NT,NR,Timeslots);
    Yest_osiczf=zeros(NT,Timeslots);
    for T=1:Timeslots
        H_hat=H(:, :, T);
        Order=[];
        index_array=[1:NT]; % Set of indices of signals to be detected
        for stage=1:NT
            %Δημιουργία του πίνακα αναστροφής της επίδρασης του καναλιού.
            W_osiczf = ((H_hat'*H_hat)^-1)*H_hat'; % Pseudoinverse Matrix
        end
        Yest_osiczf = W_osiczf*Ynoisy(:,T);
    end
end

for linear_decoder
    norm_array=[];
```

```

        for i=1:NT-(stage-1) % Detection ordering
            norm_array(i) = norm(W_osiczf(i,:));
        end
        [val_min,index_min] = min(norm_array); % Ordering in SNR
        Order = [Order index_array(index_min)];
        index_array = index_array([1:index_min-1 index_min+1:end]);
        x_tmp(stage) = W_osiczf(index_min,:)*Ynoisy(:,T); % Tx
signal estimation
        x_tmp(stage)=(1/scale)*x_tmp(stage);
        X_tmp(stage) = demodulate(modem.qamdemod(M),x_tmp(stage));%
Slicing
        X_hat(stage)= modulate(modem.qammod(M),X_tmp(stage));
        X_hat(stage)=scale*X_hat(stage);
        Ynoisy_tilde(:,T) = Ynoisy(:,T)-
H_hat(:,index_min)*X_hat(stage); % Interference subtraction
        H_hat_tilde = H_hat(:,[1:index_min-1 index_min+1:NT-(stage-
1)]); % New H
        H_hat = H_hat_tilde; Ynoisy(:,T) = Ynoisy_tilde(:,T);
    end
    Yest_osiczf(Order,T) = X_hat;
end
%Μετατροπή συμβόλων σε bits

RecSymbols_osiczf=reshape(Yest_osiczf,prod(size(Yest_osiczf)),1);
RecSymbols_osiczf=(1/scale)* RecSymbols_osiczf;
Recy_osiczf = demodulate(modem.qamdemod(M),RecSymbols_osiczf);
Recy_osiczf= de2bi(Recy_osiczf,'left-msb'); % Convert integers
to bits.
Recy_osiczf=Recy_osiczf.';

Rbitstream_osiczf=reshape(Recy_osiczf,prod(size(Recy_osiczf)),1);
% Ελεγχος Σφαλμάτων
[ NUM_ERRORS_osiczf(1)
BER_osiczf(1)]=biterr(bitstream,Rbitstream_osiczf)
end

figure
semilogy(EbNo_dB,BER_osiczf,'-k','LineWidth', 1.5)
legend('OSIC-ZF','Location','SouthWest')
xlabel('Eb/No[dB]'); ylabel('BER');
title('MIMO OSIC ZF receiver perfomance');
grid on
axis([0 20 10^-5,10^0])

```

```

%MQAM_OSIC_ZF_RECEIVER.m

```

```

Timeslots=100000;
NT=2; NR=2;
for Icase=1:3
    if Icase==1, M=2;
    elseif Icase==2, M=16;
    else M=64;

```

```

    end
    b = log2(M); % Number of bits per symbol
    QAM_nums=[0:M-1];
    QAM_syms=modulate(modem.qammod(M),QAM_nums);
    scale = modnorm(QAM_syms,'avpow',1);
    EbNo_dB=[0:2:20];
    SNR_dB=EbNo_dB+10*log10(b);

    %Δημιουργία διανύσματος των προς μετάδοση bits
    %Διαμόρφωση των προς μετάδοση bits χρησιμοποιώντας M-QAM
    bitstream=randint(Timeslots*b*NT,1);
    S=reshape(bitstream,b,length(bitstream)/b);
    S=S.';
    xsym = bi2de(S,'left-msb');
    xsym = modulate(modem.qammod(M),xsym);
    xsym=scale*xsym;
    x=reshape(xsym,NT,Timeslots); % Διαμόρφωση πίνακα των προς μετάδοση
    στοιχείων

    Y=zeros(NR,Timeslots);
    Ynoisy=zeros(NR,Timeslots);

    No=(randn(NR,Timeslots) + 1i*randn(NR,Timeslots));
    H=(randn(NR,NT,Timeslots)+1i*randn(NR,NT,Timeslots))/sqrt(2); %
    Δημιουργία MIMO καναλιού για κάθε χρονοσχισημή

    for l=1:length(EbNo_dB)
        SNR = 10^(SNR_dB(l)/10);
        sigma = sqrt(1/(2*SNR));
        noise = sigma * No;

        for T=1:Timeslots
            Y(:,T)=H(:,:,T)*x(:,T);
            Ynoisy(:,T)=Y(:,T)+noise(:,T);% Προσθήκη Θορύβου
        end

        W_osiczf=zeros(NT,NR,Timeslots);
        Yest_osiczf=zeros(NT,Timeslots);
        for T=1:Timeslots
            H_hat=H(:,:,T);
            Order=[];
            index_array=[1:NT]; % Set of indices of signals to be detected
            for stage=1:NT
                %Δημιουργία του πίνακα αναστροφής της επίδρασης του καναλιού.
                W_osiczf = ((H_hat'*H_hat)^-1)*H_hat'; % Pseudoinverse Matrix
            for linear decoder
                norm_array=[];
                for i=1:NT-(stage-1) % Detection ordering
                    norm_array(i) = norm(W_osiczf(i,:));
                end
                [val_min,index_min] = min(norm_array); % Ordering in SNR
                Order = [Order index_array(index_min)];
            end
        end
    end

```

```

        index_array = index_array([1:index_min-1 index_min+1:end]);
        x_tmp(stage) = W_osiczf(index_min,:) * Ynoisy(:,T); % Tx
signal estimation
        x_tmp(stage) = (1/scale) * x_tmp(stage);
        X_tmp(stage) = demodulate(modem.qamdemod(M), x_tmp(stage)); %
Slicing
        X_hat(stage) = modulate(modem.qammod(M), X_tmp(stage));
        X_hat(stage) = scale * X_hat(stage);
        Ynoisy_tilde(:,T) = Ynoisy(:,T) -
H_hat(:, index_min) * X_hat(stage); % Interference subtraction
        H_hat_tilde = H_hat(:, [1:index_min-1 index_min+1:NT-(stage-
1)]); % New H
        H_hat = H_hat_tilde; Ynoisy(:,T) = Ynoisy_tilde(:,T);
    end
    Yest_osiczf(Order,T) = X_hat;
end
%Μετατροπή συμβόλων σε bits

RecSymbols_osiczf = reshape(Yest_osiczf, prod(size(Yest_osiczf)), 1);
RecSymbols_osiczf = (1/scale) * RecSymbols_osiczf;
Recy_osiczf = demodulate(modem.qamdemod(M), RecSymbols_osiczf);
Recy_osiczf = de2bi(Recy_osiczf, 'left-msb'); % Convert integers
to bits.
Recy_osiczf = Recy_osiczf.';

Rbitstream_osiczf = reshape(Recy_osiczf, prod(size(Recy_osiczf)), 1);
% Ελεγχος Σφαλμάτων
[ NUM_ERRORS_osiczf(Icase, 1)
BER_osiczf(Icase, 1) ] = biterr(bitstream, Rbitstream_osiczf)
end
end
figure
semilogy(EbNo_dB, BER_osiczf(1,:), '-+', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB, BER_osiczf(2,:), 'k-^', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB, BER_osiczf(3,:), 'ro-', 'LineWidth', 1.5)
legend('2QAM', '16QAM', '64QAM', 'Location', 'SouthWest')
xlabel('Eb/No[dB]'); ylabel('BER');
title('MIMO OSIC ZF receiver performance for various MQAM modulation
schemes')
grid on
axis([0 20 10^-4, 10^0])

```

```

%OSIC_ZF_RECEIVER_antenna_configuration.m
Timeslots=100000;
M=2;
b = log2(M); % Number of bits per symbol

QAM_nums=[0:M-1];
QAM_syms=modulate(modem.qammod(M), QAM_nums);
scale = modnorm(QAM_syms, 'avpow', 1); %Power Normalization Factor

EbNo_dB=[0:2:20];

```

```

SNR_dB=EbNo_dB+10*log10(b);

for Icase=1:3
    if Icase==1, NT=2; NR=2;
    elseif Icase==2, NT=2; NR=3;
    else NT=2; NR=4;
    end

%Δημιουργία διανύσματος των προς μετάδοση bits
%Διαμόρφωση των προς μετάδοση bits χρησιμοποιώντας M-QAM
bitstream=randint(Timeslots*b*NT,1);
S=reshape(bitstream,b,length(bitstream)/b);
S=S.';
xsym = bi2de(S, 'left-msb');
xsym = modulate(modem.qammod(M), xsym);
xsym=scale*xsym;
x=reshape(xsym,NT,Timeslots); % Διαμόρφωση πίνακα των προς μετάδοση
στοιχείων

Y=zeros(NR,Timeslots);
Ynoisy=zeros(NR,Timeslots);

No=(randn(NR,Timeslots) + 1i*randn(NR,Timeslots));
H=(randn(NR,NT,Timeslots)+1i*randn(NR,NT,Timeslots))/sqrt(2); %
Δημιουργία MIMO καναλιού για κάθε χρονοσχισμή

for l=1:length(EbNo_dB)
    SNR = 10^(SNR_dB(l)/10);
    sigma = sqrt(1/(2*SNR));
    noise = sigma * No;

    for T=1:Timeslots
        Y(:,T)=H(:, :, T)*x(:,T);
        Ynoisy(:,T)=Y(:,T)+noise(:,T); % Προσθήκη Θορύβου
    end

    W_osiczf=zeros(NT,NR,Timeslots);
    Yest_osiczf=zeros(NT,Timeslots);
    for T=1:Timeslots
        H_hat=H(:, :, T);
        Order=[];
        index_array=[1:NT]; % Set of indices of signals to be detected
        for stage=1:NT
            %Δημιουργία του πίνακα αναστροφής της επίδρασης του καναλιού.
            W_osiczf = ((H_hat'*H_hat)^-1)*H_hat'; % Pseudoinverse Matrix
        for linear decoder
            norm_array=[];
            for i=1:NT-(stage-1) % Detection ordering
                norm_array(i) = norm(W_osiczf(i, :));
            end
            [val_min,index_min] = min(norm_array); % Ordering in SNR
            Order = [Order index_array(index_min)];
        end
    end
end

```

```

        index_array = index_array([1:index_min-1 index_min+1:end]);
        x_tmp(stage) = W_osiczf(index_min,:) * Ynoisy(:,T); % Tx
signal estimation
        x_tmp(stage) = (1/scale) * x_tmp(stage);
        X_tmp(stage) = demodulate(modem.qamdemod(M), x_tmp(stage)); %
Slicing
        X_hat(stage) = modulate(modem.qammod(M), X_tmp(stage));
        X_hat(stage) = scale * X_hat(stage);
        Ynoisy_tilde(:,T) = Ynoisy(:,T) -
H_hat(:, index_min) * X_hat(stage); % Interference subtraction
        H_hat_tilde = H_hat(:, [1:index_min-1 index_min+1:NT-(stage-
1)]); % New H
        H_hat = H_hat_tilde; Ynoisy(:,T) = Ynoisy_tilde(:,T);
        end
        Yest_osiczf(Order,T) = X_hat;
    end
    clear Ynoisy_tilde
    %Μετατροπή συμβόλων σε bits

RecSymbols_osiczf = reshape(Yest_osiczf, prod(size(Yest_osiczf)), 1);
RecSymbols_osiczf = (1/scale) * RecSymbols_osiczf;
Recy_osiczf = demodulate(modem.qamdemod(M), RecSymbols_osiczf);
Recy_osiczf = de2bi(Recy_osiczf, 'left-msb'); % Convert integers
to bits.
Recy_osiczf = Recy_osiczf.';

Rbitstream_osiczf = reshape(Recy_osiczf, prod(size(Recy_osiczf)), 1);
    % Έλεγχος Σφαλμάτων
    [NUM_ERRORS_osiczf(Icase, 1)] = biterr(bitstream, Rbitstream_osiczf)
end
end
figure
semilogy(EbNo_dB, BER_osiczf(1,:), '-+', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB, BER_osiczf(2,:), 'k-^', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB, BER_osiczf(3,:), 'ro-', 'LineWidth', 1.5)
legend('Nt=2 Nr=2', 'Nt=2 Nr=3', 'Nt=2 Nr=4', 'Location', 'SouthWest')
xlabel('Eb/No[dB]'); ylabel('BER');
title('OSIC ZF receiver performance for various antenna configuration
schemes')
grid on
axis([0 20 10^-4, 10^0])

```

```
%ZF_vs_OSIC_ZF_receiver.m
```

```

Timeslots=100000;
NT=2; NR=2;
M=2; %Constellation Size
b = log2(M); % Number of bits per symbol
QAM_nums=[0:M-1];
QAM_syms=modulate(modem.qammod(M), QAM_nums);
scale = modnorm(QAM_syms, 'avpow', 1);
EbNo_dB=[0:2:20];

```



```

SNR_dB=EbNo_dB+10*log10(b);

%Δημιουργία διανύσματος των προς μετάδοση bits
%Διαμόρφωση των προς μετάδοση bits χρησιμοποιώντας M-QAM
bitstream=randint(Timeslots*b*NT,1);
S=reshape(bitstream,b,length(bitstream)/b);
S=S.';
xsym = bi2de(S,'left-msb');
xsym = modulate(modem.qammod(M),xsym);
xsym=scale*xsym;
x=reshape(xsym,NT,Timeslots); % Διαμόρφωση πίνακα των προς μετάδοση
στοιχείων

Y=zeros(NR,Timeslots);
Ynoisy=zeros(NR,Timeslots);

No=(randn(NR,Timeslots) + 1i*randn(NR,Timeslots));
H=(randn(NR,NT,Timeslots)+1i*randn(NR,NT,Timeslots))/sqrt(2); %
Δημιουργία MIMO καναλιού για κάθε χρονοσχισμή

for l=1:length(EbNo_dB)
    SNR = 10^(SNR_dB(l)/10);
    sigma = sqrt(1/(2*SNR));
    noise = sigma * No;

    for T=1:Timeslots
        Y(:,T)=H(:,:,T)*x(:,T);
        Ynoisy(:,T)=Y(:,T)+noise(:,T); % Προσθήκη θορύβου
    end
    W_ZF=zeros(NT,NR,Timeslots);
    Yest_zf=zeros(NT,Timeslots);
    for T=1:Timeslots
        %Δημιουργία του πίνακα αναστροφής της επίδρασης του καναλιού.
        W_ZF(:,:,T)=(H(:,:,T)'*H(:,:,T))\H(:,:,T)'; % Pseudoinverse
Matrix for ZF decoder
        for nt=1:NT
            Yest_zf(nt,T)=W_ZF(nt,:,T)*Ynoisy(:,T); % Ανακατασκευή των
μεταδοθέντων συμβόλων
        end
    end
    %Μετατροπή συμβόλων σε bits
RecSymbols_zf=reshape(Yest_zf,prod(size(Yest_zf)),1);
RecSymbols_zf=(1/scale)*RecSymbols_zf;
Recy_zf = demodulate(modem.qamdemod(M),RecSymbols_zf);
Recy_zf= de2bi(Recy_zf,'left-msb'); % Convert integers to bits.
Recy_zf=Recy_zf.';
Rbitstream_zf=reshape(Recy_zf,prod(size(Recy_zf)),1);

% Έλεγχος Σφαλμάτων
[NUM_ERRORS_zf(1) BER_zf(1)]=biterr(bitstream,Rbitstream_zf)

W_osiczf=zeros(NT,NR,Timeslots);

```

```

Yest_osiczf=zeros(NT,Timeslots);
for T=1:Timeslots
    H_hat=H(:, :, T);
    Order=[];
    index_array=[1:NT]; % Set of indices of signals to be detected
    for stage=1:NT
        %Δημιουργία του πίνακα αναστροφής της επίδρασης του καναλιού.
        W_osiczf =((H_hat'*H_hat)^-1)*H_hat'; % Pseudoinverse Matrix
    for linear decoder
        norm_array=[];
        for i=1:NT-(stage-1) % Detection ordering
            norm_array(i) = norm(W_osiczf(i, :));
        end
        [val_min,index_min] = min(norm_array); % Ordering in SNR
        Order = [Order index_array(index_min)];
        index_array = index_array([1:index_min-1 index_min+1:end]);
        x_tmp(stage) = W_osiczf(index_min, :)*Ynoisy(:, T); % Tx
    signal estimation
        x_tmp(stage)=(1/scale)*x_tmp(stage);
        X_tmp(stage) = demodulate(modem.qamdemod(M), x_tmp(stage)); %
    Slicing
        X_hat(stage)= modulate(modem.qammod(M), X_tmp(stage));
        X_hat(stage)=scale*X_hat(stage);
        Ynoisy_tilde(:, T) = Ynoisy(:, T)-
    H_hat(:, index_min)*X_hat(stage); % Interference subtraction
        H_hat_tilde = H_hat(:, [1:index_min-1 index_min+1:NT-(stage-
    1)]); % New H
        H_hat = H_hat_tilde; Ynoisy(:, T) = Ynoisy_tilde(:, T);
    end
        Yest_osiczf(Order, T) = X_hat;
    end
    %Μετατροπή συμβόλων σε bits

RecSymbols_osiczf=reshape(Yest_osiczf,prod(size(Yest_osiczf)),1);
RecSymbols_osiczf=(1/scale)* RecSymbols_osiczf;
Recy_osiczf = demodulate(modem.qamdemod(M), RecSymbols_osiczf);
Recy_osiczf= de2bi(Recy_osiczf, 'left-msb'); % Convert integers
to bits.
Recy_osiczf=Recy_osiczf.';

Rbitstream_osiczf=reshape(Recy_osiczf,prod(size(Recy_osiczf)),1);
% Έλεγχος Σφαλμάτων
[ NUM_ERRORS_osiczf(1)
BER_osiczf(1)]=biterr(bitstream,Rbitstream_osiczf)
end

figure
semilogy(EbNo_dB,BER_zf, '-k', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB,BER_osiczf, '-o', 'LineWidth', 1.5)
legend('ZF', 'OSIC-ZF', 'Location', 'SouthWest')
xlabel('Eb/No[dB]'); ylabel('BER');
title('ZF VS OSIC ZF receiver perfomance');
grid on
axis([0 20 10^-4,10^0])

```

OSIC MMSE Receiver

```
%mimo_MMSE_OSIC_RECEIVER.m

Timeslots=100;
NT=2; NR=2;
M=2; %Constellation Size
b = log2(M); % Number of bits per symbol
QAM_nums=[0:M-1];
QAM_syms=modulate(modem.qammod(M),QAM_nums);
scale = modnorm(QAM_syms,'avpow',1);
EbNo_dB=[0:2:20];
SNR_dB=EbNo_dB+10*log10(b);

%Δημιουργία διανύσματος των προς μετάδοση bits
%Διαμόρφωση των προς μετάδοση bits χρησιμοποιώντας M-QAM
bitstream=randint(Timeslots*b*NT,1);
S=reshape(bitstream,b,length(bitstream)/b);
S=S.';
xsym = bi2de(S,'left-msb');
xsym = modulate(modem.qammod(M),xsym);
xsym=scale*xsym;
x=reshape(xsym,NT,Timeslots); % Διαμόρφωση πίνακα των προς μετάδοση
στοιχείων

Y=zeros(NR,Timeslots);
Ynoisy=zeros(NR,Timeslots);

No=(randn(NR,Timeslots) + 1i*randn(NR,Timeslots));
H=(randn(NR,NT,Timeslots)+1i*randn(NR,NT,Timeslots))/sqrt(2); %
Δημιουργία MIMO καναλιού για κάθε χρονοσχιμή

for l=1:length(EbNo_dB)
    SNR = 10^(SNR_dB(l)/10);
    sigma = sqrt(1/(2*SNR));
    noise = sigma * No;

    for T=1:Timeslots
        Y(:,T)=H(:,:,T)*x(:,T);
        Ynoisy(:,T)=Y(:,T)+noise(:,T);% Προσθήκη θορύβου
    end
    W_osicmmse=zeros(NT,NR,Timeslots);
    Y_est_osicmmse=zeros(NT,Timeslots);
    for T=1:Timeslots
        H_hat=H(:,:,T);
        Order=[];
        index_array=[1:NT]; % Set of indices of signals to be detected
        for stage=1:NT
            W_osicmmse = (H_hat'*H_hat+(sigma^2)*eye(NT+1-stage))\H_hat'; %
MMSE FILTER MATRIX
```

```

    WH=W_osicmmse*H_hat;
    SINR=[];
    for i=1:NT-(stage-1)

        SINR(i)=(abs(WH(i,i))^2)/(sum(abs(WH(i,[1:i-1 i+1:NT-(stage-1)]))^2)+(sigma^2)*norm(W_osicmmse(i,:))^2);
    end
    [val_max,index_tmp] = max(SINR); % Ordering using SINR
    Order = [Order index_array(index_tmp)];
    index_array = index_array([1:index_tmp-1
index_tmp+1:end]);

    x_tmp(stage) = W_osicmmse(index_tmp,:)*Ynoisy(:,T); % Tx
signal estimation
    x_tmp(stage)=(1/scale)*x_tmp(stage);
    X_tmp(stage) = demodulate(modem.qamdmod(M),x_tmp(stage));
    X_hat(stage)= modulate(modem.qammod(M),X_tmp(stage));%
Slicing
    X_hat(stage)=scale*X_hat(stage);
    Ynoisy_tilde(:,T) = Ynoisy(:,T)-
H_hat(:,index_tmp)*X_hat(stage); % Interference subtraction
    H_hat_tilde = H_hat(:,[1:index_tmp-1 index_tmp+1:NT-(stage-1)]); % New H
    H_hat = H_hat_tilde; Ynoisy(:,T) = Ynoisy_tilde(:,T);
end
    Y_est_osicmmse(Order,T) = X_hat;
end
    %Μετατροπή συμβόλων σε bits

RecSymbols_osicmmse=reshape(Y_est_osicmmse,prod(size(Y_est_osicmmse)
),1);
    RecSymbols_osicmmse=(1/scale)*RecSymbols_osicmmse;
    Recy_osicmmse =
demodulate(modem.qamdmod(M),RecSymbols_osicmmse);
    Recy_osicmmse= de2bi(Recy_osicmmse,'left-msb'); % Convert
integers to bits.
    Recy_osicmmse=Recy_osicmmse.';

Rbitstream_osicmmse=reshape(Recy_osicmmse,prod(size(Recy_osicmmse)),
1);
    % Έλεγχος Σφαλμάτων
    [NUM_ERRORS_osicmmse(1)
BER_osicmmse(1)]=biterr(bitstream,Rbitstream_osicmmse)
end

figure
semilogy(EbNo_dB,BER_osicmmse,'-k^','LineWidth', 1.5);
legend('OSIC MMSE','Location','SouthWest')
xlabel('Eb/No[dB]'); ylabel('BER');
title('MIMO OSIC MMSE receiver performance')
grid on
axis([0 20 10^-5,10^0])

```

```

%MQAM_OSIC_MMSE_RECEIVER.m

Timeslots=100000;
NT=2; NR=2;
for Icase=1:3
    if Icase==1, M=2;
    elseif Icase==2, M=16;
    else M=64;
    end

    b = log2(M); % Number of bits per symbol
    QAM_nums=[0:M-1];
    QAM_syms=modulate(modem.qammod(M),QAM_nums);
    scale = modnorm(QAM_syms,'avpow',1);
    EbNo_dB=[0:2:20];
    SNR_dB=EbNo_dB+10*log10(b);

    %Δημιουργία διανύσματος των προς μετάδοση bits
    %Διαμόρφωση των προς μετάδοση bits χρησιμοποιώντας M-QAM
    bitstream=randint(Timeslots*b*NT,1);
    S=reshape(bitstream,b,length(bitstream)/b);
    S=S.';
    xsym = bi2de(S,'left-msb');
    xsym = modulate(modem.qammod(M),xsym);
    xsym=scale*xsym;
    x=reshape(xsym,NT,Timeslots); % Διαμόρφωση πίνακα των προς μετάδοση
    στοιχείων

    Y=zeros(NR,Timeslots);
    Ynoisy=zeros(NR,Timeslots);

    No=(randn(NR,Timeslots) + 1i*randn(NR,Timeslots));
    H=(randn(NR,NT,Timeslots)+1i*randn(NR,NT,Timeslots))/sqrt(2); %
    Δημιουργία MIMO καναλιού για κάθε χρονοσχισημή

    for l=1:length(EbNo_dB)
        SNR = 10^(SNR_dB(l)/10);
        sigma = sqrt(1/(2*SNR));
        noise = sigma * No;

        for T=1:Timeslots
            Y(:,T)=H(:, :, T)*x(:,T);
            Ynoisy(:,T)=Y(:,T)+noise(:,T); % Προσθήκη θορύβου
        end
        W_osicmmse=zeros(NT,NR,Timeslots);
        Y_est_osicmmse=zeros(NT,Timeslots);
        for T=1:Timeslots
            H_hat=H(:, :, T);
            Order=[];
            index_array=[1:NT]; % Set of indices of signals to be detected
            for stage=1:NT

```

```

        W_osicmmse = (H_hat'*H_hat+(sigma^2)*eye(NT+1-stage))\H_hat'; %
MMSE FILTER MATRIX
        WH=W_osicmmse*H_hat;
        SINR=[];
        for i=1:NT-(stage-1)

            SINR(i)=(abs(WH(i,i))^2)/(sum(abs(WH(i,[1:i-1 i+1:NT-
(stage-1)]))^2)+(sigma^2)*norm(W_osicmmse(i,:))^2);
            end
            [val_max,index_tmp] = max(SINR); % Ordering using SINR
            Order = [Order index_array(index_tmp)];
            index_array = index_array([1:index_tmp-1
index_tmp+1:end]);

            x_tmp(stage) = W_osicmmse(index_tmp,:)*Ynoisy(:,T); % Tx
signal estimation
            x_tmp(stage)=(1/scale)*x_tmp(stage);
            X_tmp(stage) = demodulate(modem.qamdemod(M),x_tmp(stage));
            X_hat(stage)= modulate(modem.qammod(M),X_tmp(stage));%
Slicing
            X_hat(stage)=scale*X_hat(stage);
            Ynoisy_tilde(:,T) = Ynoisy(:,T)-
H_hat(:,index_tmp)*X_hat(stage); % Interference subtraction
            H_hat_tilde = H_hat(:,[1:index_tmp-1 index_tmp+1:NT-(stage-
1)]); % New H
            H_hat = H_hat_tilde; Ynoisy(:,T) = Ynoisy_tilde(:,T);
            end
            Y_est_osicmmse(Order,T) = X_hat;
        end
        %Μετατροπή συμβόλων σε bits

RecSymbols_osicmmse=reshape(Y_est_osicmmse,prod(size(Y_est_osicmmse)
),1);
RecSymbols_osicmmse=(1/scale)*RecSymbols_osicmmse;
Recy_osicmmse =
demodulate(modem.qamdemod(M),RecSymbols_osicmmse);
Recy_osicmmse= de2bi(Recy_osicmmse,'left-msb'); % Convert
integers to bits.
Recy_osicmmse=Recy_osicmmse.';

Rbitstream_osicmmse=reshape(Recy_osicmmse,prod(size(Recy_osicmmse)),
1);
    % Έλεγχος Σφαλμάτων
    [NUM_ERRORS_osicmmse(Icase,1)
BER_osicmmse(Icase,1)]=biterr(bitstream,Rbitstream_osicmmse)
end
end
figure
semilogy(EbNo_dB,BER_osicmmse(1,:), '-+', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB,BER_osicmmse(2,:), 'k-^', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB,BER_osicmmse(3,:), 'ro-', 'LineWidth', 1.5)
legend('2QAM', '16QAM', '64QAM', 'Location', 'SouthWest')
xlabel('Eb/No[dB]'); ylabel('BER');

```

```

title('OSIC MMSE receiver performance for various MQAM modulation
schemes')
grid on
axis([0 20 10^-4,10^0])

```

```

%OSIC_MMSE_RECEIVER_antenna_configuration.m

Timeslots=100000;
M=2;
b = log2(M); % Number of bits per symbol

QAM_nums=[0:M-1];
QAM_syms=modulate(modem.qammod(M),QAM_nums);
scale = modnorm(QAM_syms,'avpow',1); %Power Normalization Factor

EbNo_dB=[0:2:20];
SNR_dB=EbNo_dB+10*log10(b);

for Icase=1:3
    if Icase==1, NT=2; NR=2;
    elseif Icase==2, NT=2; NR=3;
    else NT=2; NR=4;
    end

    %Δημιουργία διανύσματος των προς μετάδοση bits
    %Διαμόρφωση των προς μετάδοση bits χρησιμοποιώντας M-QAM
    bitstream=randint(Timeslots*b*NT,1);
    S=reshape(bitstream,b,length(bitstream)/b);
    S=S.';
    xsym = bi2de(S,'left-msb');
    xsym = modulate(modem.qammod(M),xsym);
    xsym=scale*xsym;
    x=reshape(xsym,NT,Timeslots); % Διαμόρφωση πίνακα των προς μετάδοση
    στοιχείων

    Y=zeros(NR,Timeslots);
    Ynoisy=zeros(NR,Timeslots);

    No=(randn(NR,Timeslots)+1i*randn(NR,Timeslots));
    H=(randn(NR,NT,Timeslots)+1i*randn(NR,NT,Timeslots))/sqrt(2); %
    Δημιουργία MIMO καναλιού για κάθε χρονοσχισμή
    for l=1:length(EbNo_dB)
        SNR = 10^(SNR_dB(l)/10);
        sigma = sqrt(1/(2*SNR));
        noise = sigma * No;

        for T=1:Timeslots
            Y(:,T)=H(:,:,T)*x(:,T);
            Ynoisy(:,T)=Y(:,T)+noise(:,T); % Προσθήκη θορύβου
        end
        W_osicmmse=zeros(NT,NR,Timeslots);
    end

```

```

    Y_est_osicmmse=zeros (NT,Timeslots);
    for T=1:Timeslots
        H_hat=H(:, :,T);
        Order=[];
        index_array=[1:NT]; % Set of indices of signals to be detected
        for stage=1:NT
            W_osicmmse =(H_hat'*H_hat+(sigma^2)*eye (NT+1-stage))\H_hat'; %
MMSE FILTER MATRIX
            WH=W_osicmmse*H_hat;
            SINR=[];
            for i=1:NT-(stage-1)

                SINR(i)=(abs (WH (i,i))^2)/(sum(abs (WH (i,[1:i-1 i+1:NT-
(stage-1)]))^2)+(sigma^2)*norm(W_osicmmse(i,:))^2);
            end
            [val_max,index_tmp] = max(SINR); % Ordering using SINR
            Order = [Order index_array(index_tmp)];
            index_array = index_array([1:index_tmp-1
index_tmp+1:end]);

            x_tmp(stage) = W_osicmmse(index_tmp,:)*Ynoisy(:,T); % Tx
signal estimation
            x_tmp(stage)=(1/scale)*x_tmp(stage);
            X_tmp(stage) = demodulate(modem.qamdemod(M),x_tmp(stage));
            X_hat(stage)= modulate(modem.qammod(M),X_tmp(stage));%
Slicing
            X_hat(stage)=scale*X_hat(stage);
            Ynoisy_tilde(:,T) = Ynoisy(:,T)-
H_hat(:,index_tmp)*X_hat(stage); % Interference subtraction
            H_hat_tilde = H_hat(:,[1:index_tmp-1 index_tmp+1:NT-(stage-
1)]); % New H
            H_hat = H_hat_tilde; Ynoisy(:,T) = Ynoisy_tilde(:,T);
        end
        Y_est_osicmmse(Order,T) = X_hat;
    end
    clear Ynoisy_tilde
        %Μετατροπή συμβόλων σε bits

RecSymbols_osicmmse=reshape (Y_est_osicmmse,prod(size(Y_est_osicmmse)
),1);
    RecSymbols_osicmmse=(1/scale)*RecSymbols_osicmmse;
    Recy_osicmmse =
demodulate(modem.qamdemod(M),RecSymbols_osicmmse);
    Recy_osicmmse= de2bi(Recy_osicmmse,'left-msb'); % Convert
integers to bits.
    Recy_osicmmse=Recy_osicmmse.';

Rbitstream_osicmmse=reshape (Recy_osicmmse,prod(size(Recy_osicmmse)),
1);
    % Έλεγχος Σφαλμάτων
    [NUM_ERRORS_osicmmse(Icase,1)
BER_osicmmse(Icase,1)]=biterr(bitstream,Rbitstream_osicmmse)
end
end

```



```

figure
semilogy(EbNo_dB,BER_osicmmse(1,:), '--', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB,BER_osicmmse(2,:), 'k-^', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB,BER_osicmmse(3,:), 'ro-', 'LineWidth', 1.5)
legend('Nt=2 Nr=2', 'Nt=2 Nr=3', 'Nt=2 Nr=4', 'Location', 'SouthWest')
xlabel('Eb/No[dB]'); ylabel('BER');
title('OSIC MMSE receiver performance for various antenna
configuration schemes')
grid on
axis([0 20 10^-5,10^0])

```

```
%MMSE_VS_OSIC_MMSE_receiver.m
```

```

Timeslots=100000;
NT=2; NR=2;
M=2; %Constellation Size
b = log2(M); % Number of bits per symbol
QAM_nums=[0:M-1];
QAM_syms=modulate(modem.qammod(M), QAM_nums);
scale = modnorm(QAM_syms, 'avpow', 1);
EbNo_dB=[0:2:20];
SNR_dB=EbNo_dB+10*log10(b);

%Δημιουργία διανύσματος των προς μετάδοση bits
%Διαμόρφωση των προς μετάδοση bits χρησιμοποιώντας M-QAM
bitstream=randint(Timeslots*b*NT,1);
S=reshape(bitstream,b,length(bitstream)/b);
S=S.';
xsym = bi2de(S, 'left-msb');
xsym = modulate(modem.qammod(M), xsym);
xsym=scale*xsym;
x=reshape(xsym,NT,Timeslots); % Διαμόρφωση πίνακα των προς μετάδοση
στοιχείων

Y=zeros(NR,Timeslots);
Ynoisy=zeros(NR,Timeslots);

No=(randn(NR,Timeslots) + 1i*randn(NR,Timeslots));
H=(randn(NR,NT,Timeslots)+1i*randn(NR,NT,Timeslots))/sqrt(2); %
Δημιουργία MIMO καναλιού για κάθε χρονοσχισμή

for l=1:length(EbNo_dB)
    SNR = 10^(SNR_dB(l)/10);
    sigma = sqrt(1/(2*SNR));
    noise = sigma * No;

    for T=1:Timeslots
        Y(:,T)=H(:, :, T)*x(:,T);
        Ynoisy(:,T)=Y(:,T)+noise(:,T); % Προσθήκη Θορύβου
    end
    W_MMSE=zeros(NT,NR,Timeslots);
    Yest_mmse=zeros(NT,Timeslots);
    for T=1:Timeslots

```

```

%Δημιουργία του πίνακα αναστροφής της επίδρασης του καναλιού.
W_MMSE(:, :, T) = (H(:, :, T)' * H(:, :, T) + (sigma^2) * eye (NT)) \ H(:, :, T)';
%MMSE Filter Construction
for nt=1:NT
    Yest_mmse(nt, T) = W_MMSE(nt, :, T) * Ynoisy(:, T);
end
end

%Μετατροπή συμβόλων σε bits
RecSymbols_mmse = reshape(Yest_mmse, prod(size(Yest_mmse)), 1);
RecSymbols_mmse = (1/scale) * RecSymbols_mmse;
Recy_mmse = demodulate(modem.qamdemod(M), RecSymbols_mmse);
Recy_mmse = de2bi(Recy_mmse, 'left-msb'); % Convert integers to bits.
Recy_mmse = Recy_mmse.';
Rbitstream_mmse = reshape(Recy_mmse, prod(size(Recy_mmse)), 1);

% Έλεγχος Σφαλμάτων
[NUM_ERRORS_mmse(1) BER_mmse(1)] = biterr(bitstream, Rbitstream_mmse)

W_osicmmse = zeros (NT, NR, Timeslots);
Y_est_osicmmse = zeros (NT, Timeslots);
for T=1:Timeslots
    H_hat = H(:, :, T);
    Order = [];
    index_array = [1:NT]; % Set of indices of signals to be detected
    for stage=1:NT
        W_osicmmse = (H_hat' * H_hat + (sigma^2) * eye (NT+1-stage)) \ H_hat'; %
MMSE FILTER MATRIX
        WH = W_osicmmse * H_hat;
        SINR = [];
        for i=1:NT-(stage-1)

            SINR(i) = (abs(WH(i, i))^2) / (sum(abs(WH(i, [1:i-1 i+1:NT-(stage-1)]))^2) + (sigma^2) * norm(W_osicmmse(i, :))^2);
        end
        [val_max, index_tmp] = max(SINR); % Ordering using SINR
        Order = [Order index_array(index_tmp)];
        index_array = index_array([1:index_tmp-1
index_tmp+1:end]);

        x_tmp(stage) = W_osicmmse(index_tmp, :) * Ynoisy(:, T); % Tx
signal estimation
        x_tmp(stage) = (1/scale) * x_tmp(stage);
        X_tmp(stage) = demodulate(modem.qamdemod(M), x_tmp(stage));
        X_hat(stage) = modulate(modem.qammod(M), X_tmp(stage)); %
Slicing
        X_hat(stage) = scale * X_hat(stage);
        Ynoisy_tilde(:, T) = Ynoisy(:, T) -
H_hat(:, index_tmp) * X_hat(stage); % Interference subtraction
        H_hat_tilde = H_hat(:, [1:index_tmp-1 index_tmp+1:NT-(stage-1)]); % New H
        H_hat = H_hat_tilde; Ynoisy(:, T) = Ynoisy_tilde(:, T);

```

```

        end
        Y_est_osicmmse(Order,T) = X_hat;
    end
    %Μετατροπή συμβόλων σε bits

RecSymbols_osicmmse=reshape(Y_est_osicmmse,prod(size(Y_est_osicmmse)
),1);
    RecSymbols_osicmmse=(1/scale)*RecSymbols_osicmmse;
    Recy_osicmmse =
demodulate(modem.qamdemod(M),RecSymbols_osicmmse);
    Recy_osicmmse= de2bi(Recy_osicmmse,'left-msb'); % Convert
integers to bits.
    Recy_osicmmse=Recy_osicmmse.';

Rbitstream_osicmmse=reshape(Recy_osicmmse,prod(size(Recy_osicmmse)),
1);
    % Έλεγχος Σφαλμάτων
    [NUM_ERRORS_osicmmse(1)
BER_osicmmse(1)]=biterr(bitstream,Rbitstream_osicmmse)
end

figure
semilogy(EbNo_dB,BER_mmse,'-k','LineWidth', 1.5)
hold on, semilogy(EbNo_dB,BER_osicmmse,'-o','LineWidth', 1.5);
legend('MMSE','OSIC MMSE','Location','SouthWest')
xlabel('Eb/No[dB]'); ylabel('BER');
title('MMSE VS OSIC MMSE receiver perfomance')
grid on
axis([0 20 10^-4,10^0])

```

```

%OSIC_ZF_VS_OSIC_MMSE_RECEIVER.m

Timeslots=100000;
NT=2; NR=2;
M=2; %Constellation Size
b = log2(M); % Number of bits per symbol
QAM_nums=[0:M-1];
QAM_syms=modulate(modem.qammod(M),QAM_nums);
scale = modnorm(QAM_syms,'avpow',1);
EbNo_dB=[0:2:20];
SNR_dB=EbNo_dB+10*log10(b);

%Δημιουργία διανύσματος των προς μετάδοση bits
%Διαμόρφωση των προς μετάδοση bits χρησιμοποιώντας M-QAM
bitstream=randint(Timeslots*b*NT,1);
S=reshape(bitstream,b,length(bitstream)/b);
S=S.';
xsym = bi2de(S,'left-msb');
xsym = modulate(modem.qammod(M),xsym);
xsym=scale*xsym;
x=reshape(xsym,NT,Timeslots); % Διαμόρφωση πίνακα των προς μετάδοση
στοιχείων

```

```

Y=zeros(NR,Timeslots);
Ynoisy=zeros(NR,Timeslots);

No=(randn(NR,Timeslots) + li*randn(NR,Timeslots));
H=(randn(NR,NT,Timeslots)+li*randn(NR,NT,Timeslots))/sqrt(2); %
Δημιουργία MIMO καναλιού για κάθε χρονοσχισμή

for l=1:length(EbNo_dB)
    SNR = 10^(SNR_dB(l)/10);
    sigma = sqrt(1/(2*SNR));
    noise = sigma * No;

    for T=1:Timeslots
        Y(:,T)=H(:, :, T)*x(:, T);
        Ynoisy(:,T)=Y(:,T)+noise(:,T); % Προσθήκη θορύβου
    end
    Ynoisy1=Ynoisy;

    W_osiczf=zeros(NT,NR,Timeslots);
    Yest_osiczf=zeros(NT,Timeslots);
    for T=1:Timeslots
        H_hat=H(:, :, T);
        Order=[];
        index_array=[1:NT]; % Set of indices of signals to be detected
        for stage=1:NT
            %Δημιουργία του πίνακα αναστροφής της επίδρασης του καναλιού.
            W_osiczf = ((H_hat'*H_hat)^-1)*H_hat'; % Pseudoinverse Matrix
        for linear decoder
            norm_array=[];
            for i=1:NT-(stage-1) % Detection ordering
                norm_array(i) = norm(W_osiczf(i, :));
            end
            [val_min,index_min] = min(norm_array); % Ordering in SNR
            Order = [Order index_array(index_min)];
            index_array = index_array([1:index_min-1 index_min+1:end]);
            x_tmp(stage) = W_osiczf(index_min, :)*Ynoisy(:,T); % Tx
        signal estimation
            x_tmp(stage)=(1/scale)*x_tmp(stage);
            X_tmp(stage) = demodulate(modem.qamdemod(M),x_tmp(stage)); %
        Slicing
            X_hat(stage)= modulate(modem.qammod(M),X_tmp(stage));
            X_hat(stage)=scale*X_hat(stage);
            Ynoisy_tilde(:,T) = Ynoisy(:,T)-
            H_hat(:,index_min)*X_hat(stage); % Interference subtraction
            H_hat_tilde = H_hat(:, [1:index_min-1 index_min+1:NT-(stage-
            1)]); % New H
            H_hat = H_hat_tilde; Ynoisy(:,T) = Ynoisy_tilde(:,T);
        end
        Yest_osiczf(Order,T) = X_hat;
    end
    %Μετατροπή συμβόλων σε bits

```

```

RecSymbols_osiczf=reshape(Yest_osiczf,prod(size(Yest_osiczf)),1);
RecSymbols_osiczf=(1/scale)* RecSymbols_osiczf;
Recy_osiczf = demodulate(modem.qamdemod(M),RecSymbols_osiczf);
Recy_osiczf= de2bi(Recy_osiczf,'left-msb'); % Convert integers
to bits.
Recy_osiczf=Recy_osiczf.';

Rbitstream_osiczf=reshape(Recy_osiczf,prod(size(Recy_osiczf)),1);
% Ελεγχος Σφαλμάτων
[NUM_ERRORS_osiczf(1)
BER_osiczf(1)]=biterr(bitstream,Rbitstream_osiczf)

W_osicmmse=zeros(NT,NR,Timeslots);
Y_est_osicmmse=zeros(NT,Timeslots);
for T=1:Timeslots
    H_hat=H(:, :, T);
    Order=[];
    index_array=[1:NT]; % Set of indices of signals to be detected
    for stage=1:NT
        W_osicmmse = (H_hat'*H_hat+(sigma^2)*eye(NT+1-stage))\H_hat'; %
MMSE FILTER MATRIX
        WH=W_osicmmse*H_hat;
        SINR=[];
        for i=1:NT-(stage-1)

            SINR(i)=(abs(WH(i,i))^2)/(sum(abs(WH(i,[1:i-1 i+1:NT-
(stage-1)]))^2)+(sigma^2)*norm(W_osicmmse(i,:))^2);
            end
            [val_max,index_tmp] = max(SINR); % Ordering using SINR
            Order = [Order index_array(index_tmp)];
            index_array = index_array([1:index_tmp-1
index_tmp+1:end]);

            x_tmp(stage) = W_osicmmse(index_tmp,:)*Ynoisy1(:,T); % Tx
signal estimation
            x_tmp(stage)=(1/scale)*x_tmp(stage);
            X_tmp(stage) = demodulate(modem.qamdemod(M),x_tmp(stage));
            X_hat(stage)= modulate(modem.qammod(M),X_tmp(stage));%
Slicing
            X_hat(stage)=scale*X_hat(stage);
            Ynoisy_tilde(:,T) = Ynoisy1(:,T)-
H_hat(:,index_tmp)*X_hat(stage); % Interference subtraction
            H_hat_tilde = H_hat(:,[1:index_tmp-1 index_tmp+1:NT-(stage-
1)]); % New H
            H_hat = H_hat_tilde; Ynoisy1(:,T) = Ynoisy_tilde(:,T);
            end
            Y_est_osicmmse(Order,T) = X_hat;
        end
        %Μετατροπή συμβόλων σε bits
RecSymbols_osicmmse=reshape(Y_est_osicmmse,prod(size(Y_est_osicmmse)
),1);
RecSymbols_osicmmse=(1/scale)*RecSymbols_osicmmse;

```

```

    Recy_osicmmse =
demodulate(modem.qamdemod(M),RecSymbols_osicmmse);
    Recy_osicmmse= de2bi(Recy_osicmmse,'left-msb'); % Convert
integers to bits.
    Recy_osicmmse=Recy_osicmmse.';

Rbitstream_osicmmse=reshape(Recy_osicmmse,prod(size(Recy_osicmmse)),
1);
    % Έλεγχος Σφαλαμάτων
    [NUM_ERRORS_osicmmse(1)
BER_osicmmse(1)]=biterr(bitstream,Rbitstream_osicmmse)

end
figure
semilogy(EbNo_dB,BER_osiczf,'-k^','LineWidth', 1.5)
hold on, semilogy(EbNo_dB,BER_osicmmse,'ro-','LineWidth', 1.5)
legend('OSIC ZF','OSIC MMSE','Location','SouthWest')
xlabel('Eb/No[dB]'); ylabel('BER');
title('OSIC ZF VS OSIC MMSE receiver performace')
grid on
axis([0 20 10^-4,10^0])

```

ML Receiver

```

%ML_receiver.m

Timeslots=100000;
M=4; %Constellation Size
b = log2(M); % Number of bits per symbol
QAM_nums=[0:M-1];
QAM_syms=modulate(modem.qammod(M),QAM_nums);
scale = modnorm(QAM_syms,'avpow',1);
EbNo_dB=[0:2:20];
SNR_dB=EbNo_dB+10*log10(b);

for Icase=1:3
    if Icase==1, NT=2; NR=2;
    elseif Icase==2, NT=3; NR=3;
    else NT=4; NR=4;
    end
    %Δημιουργία διανύσματος των προς μετάδοση bits
    %Διαμόρφωση των προς μετάδοση bits χρησιμοποιώντας M-QAM
    bitstream=randint(Timeslots*b*NT,1);
    S=reshape(bitstream,b,length(bitstream)/b);
    S=S.';
    xsym = bi2de(S,'left-msb');
    xsym = modulate(modem.qammod(M),xsym);
    xsym=scale*xsym;
    x=reshape(xsym,NT,Timeslots); % Διαμόρφωση πίνακα των προς μετάδοση
στοιχείων

```

```

Y=zeros(NR,Timeslots);
Ynoisy=zeros(NR,Timeslots);

No=(randn(NR,Timeslots) + 1i*randn(NR,Timeslots));
H=(randn(NR,NT,Timeslots)+1i*randn(NR,NT,Timeslots))/sqrt(2); %
Δημιουργία MIMO καναλιού για κάθε χρονοσχηματισμό

for l=1:length(EbNo_dB)
    SNR = 10^(SNR_dB(l)/10);
    sigma = sqrt(1/(2*SNR));
    noise = sigma * No;

    for T=1:Timeslots
        Y(:,T)=H(:, :, T)*x(:,T);
        Ynoisy(:,T)=Y(:,T)+noise(:,T);% Προσθήκη Θορύβου
    end

    Yest_ml=zeros(NT,T);
    if NT==2

        for T=1:Timeslots
            metric = 100000;
            for i = 1:M
                x_tmp(1) = scale*QAM_syms(i);
                for m = 1:M
                    x_tmp(2) = scale*QAM_syms(m);
                    metric_tmp=(norm(Ynoisy(:,T)-(H(:, :, T)*x_tmp.')))^2;
                    if metric_tmp<metric
                        X_hat = x_tmp;
                        metric = metric_tmp;
                    end
                end
            end
            Yest_ml(:,T)=X_hat.';
        end
    elseif NT==3

        for T=1:Timeslots
            metric = 100000;
            for i = 1:M
                x_tmp(1) = scale*QAM_syms(i);
                for n = 1:M
                    x_tmp(2) = scale*QAM_syms(n);
                    for m = 1:M
                        x_tmp(3) = scale*QAM_syms(m);
                        metric_tmp=(norm(Ynoisy(:,T)-(H(:, :, T)*x_tmp.')))^2;
                        if metric_tmp<metric
                            X_hat = x_tmp;
                            metric = metric_tmp;
                        end
                    end
                end
            end
        end
    end
end
end

```

```

        Yest_ml(:,T)=X_hat.';
    end
    elseif NT==4

for T=1:Timeslots
    metric = 100000;
    for i = 1:M
        x_tmp(1) = scale*QAM_syms(i);
        for o = 1:M
            x_tmp(2) = scale*QAM_syms(o);
            for n = 1:M
                x_tmp(3) = scale*QAM_syms(n);
                for m = 1:M
                    x_tmp(4) = scale*QAM_syms(m);
                    metric_tmp=(norm(Ynoisy(:,T)-
(H(:, :, T)*x_tmp.')) ^2;
                    if metric_tmp<metric
                        X_hat = x_tmp;
                        metric = metric_tmp;
                    end
                end
            end
        end
    end
    Yest_ml(:,T)=X_hat.';
end
end
%Μετατροπή συμβόλων σε bits
RecSymbols_ml=reshape(Yest_ml,prod(size(Yest_ml)),1);
RecSymbols_ml=(1/scale)*RecSymbols_ml;
Recy_ml = demodulate(modem.qamdemod(M),RecSymbols_ml);
Recy_ml= de2bi(Recy_ml,'left-msb'); % Convert integers to bits.
Recy_ml=Recy_ml.';
Rbitstream_ml=reshape(Recy_ml,prod(size(Recy_ml)),1);
% Έλεγχος Σφαλμάτων
[ NUM_ERRORS_ml(Icase,1)
BER_ml(Icase,1)]=biterr(bitstream,Rbitstream_ml)
end
end
figure
semilogy(EbNo_dB,BER_ml(1,:), '-+', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB,BER_ml(2,:), 'k-^', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB,BER_ml(3,:), 'ro-', 'LineWidth', 1.5)
legend('Nt=2 Nr=2', 'Nt=3 Nr=3', 'Nt=4 Nr=4', 'Location', 'SouthWest')
xlabel('Eb/No[dB]'); ylabel('BER');
title('ML receiver performance')
grid on
axis([0 20 10^-5,10^0])

```

```

%MQAM_ML_receiver.m
clear all
Timeslots=100000;
NT=2; NR=2;

```



```

for Icase=1:3
    if Icase==1, M=2;
    elseif Icase==2, M=16;
    else M=64;
    end

    b = log2(M); % Number of bits per symbol
    QAM_nums=[0:M-1];
    QAM_syms=modulate(modem.qammod(M),QAM_nums);
    scale = modnorm(QAM_syms,'avpow',1);
    EbNo_dB=[0:2:20];
    SNR_dB=EbNo_dB+10*log10(b);
    %Δημιουργία διανύσματος των προς μετάδοση bits
    %Διαμόρφωση των προς μετάδοση bits χρησιμοποιώντας M-QAM
    bitstream=randint(Timeslots*b*NT,1);
    S=reshape(bitstream,b,length(bitstream)/b);
    S=S.';
    xsym = bi2de(S,'left-msb');
    xsym = modulate(modem.qammod(M),xsym);
    xsym=scale*xsym;
    x=reshape(xsym,NT,Timeslots); % Διαμόρφωση πίνακα των προς μετάδοση
    στοιχείων

    Y=zeros(NR,Timeslots);
    Ynoisy=zeros(NR,Timeslots);

    No=(randn(NR,Timeslots) + 1i*randn(NR,Timeslots));
    H=(randn(NR,NT,Timeslots)+1i*randn(NR,NT,Timeslots))/sqrt(2); %
    Δημιουργία MIMO καναλιού για κάθε χρονοσχισημή

    for l=1:length(EbNo_dB)
        SNR = 10^(SNR_dB(l)/10);
        sigma = sqrt(1/(2*SNR));
        noise = sigma * No;

        for T=1:Timeslots
            Y(:,T)=H(:, :, T)*x(:,T);
            Ynoisy(:,T)=Y(:,T)+noise(:,T); % Προσθήκη θορύβου
        end

        Yest_ml=zeros(NT,T);

        for T=1:Timeslots
            metric = 100000;
            for i = 1:M
                x_tmp(1) = scale*QAM_syms(i);
                for m = 1:M
                    x_tmp(2) = scale*QAM_syms(m);
                    metric_tmp=(norm(Ynoisy(:,T)-(H(:, :, T)*x_tmp.')))^2;
                    if metric_tmp<metric
                        X_hat = x_tmp;
                        metric = metric_tmp;
                    end
                end
            end
        end
    end

```

```

        end
        end
    end
    Yest_ml(:,T)=X_hat.';
end

%Μετατροπή συμβόλων σε bits
RecSymbols_ml=reshape(Yest_ml,prod(size(Yest_ml)),1);
RecSymbols_ml=(1/scale)*RecSymbols_ml;
Recy_ml = demodulate(modem.qamdemod(M),RecSymbols_ml);
Recy_ml= de2bi(Recy_ml,'left-msb'); % Convert integers to bits.
Recy_ml=Recy_ml.';
Rbitstream_ml=reshape(Recy_ml,prod(size(Recy_ml)),1);
% Έλεγχος Σφαλμάτων
[ NUM_ERRORS_ml(Icase,1)
BER_ml(Icase,1) ]=biterr(bitstream,Rbitstream_ml)
end
end
figure
semilogy(EbNo_dB,BER_ml(1,:),'-+', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB,BER_ml(2,:),'k-^', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB,BER_ml(3,:),'ro-', 'LineWidth', 1.5)
legend('2QAM', '16QAM', '64QAM', 'Location', 'SouthWest')
xlabel('Eb/No[dB]'); ylabel('BER');
title('ML receiver perfomance for various Modulation schemes')
grid on
axis([0 20 10^-5,10^0])

```

```

%ML_OSIC_MMSE_OSIC_ZF_MMSE_ZF_comparison.m

Timeslots=100000;
NT=2; NR=2;
M=2; %Constellation Size
b = log2(M); % Number of bits per symbol
QAM_nums=[0:M-1];
QAM_syms=modulate(modem.qammod(M),QAM_nums);
scale = modnorm(QAM_syms,'avpow',1);
EbNo_dB=[0:2:20];
SNR_dB=EbNo_dB+10*log10(b);

%Δημιουργία διανύσματος των προς μετάδοση bits
%Διαμόρφωση των προς μετάδοση bits χρησιμοποιώντας M-QAM
bitstream=randint(Timeslots*b*NT,1);
S=reshape(bitstream,b,length(bitstream)/b);
S=S.';
xsym = bi2de(S,'left-msb');
xsym = modulate(modem.qammod(M),xsym);
xsym=scale*xsym;
x=reshape(xsym,NT,Timeslots); % Διαμόρφωση πίνακα των προς μετάδοση
στοιχείων

```

```

Y=zeros(NR,Timeslots);
Ynoisy=zeros(NR,Timeslots);

No=(randn(NR,Timeslots) + 1i*randn(NR,Timeslots));
H=(randn(NR,NT,Timeslots)+1i*randn(NR,NT,Timeslots))/sqrt(2); %
Δημιουργία MIMO καναλιού για κάθε χρονοσχισμή

for l=1:length(EbNo_dB)
    SNR = 10^(SNR_dB(l)/10);
    sigma = sqrt(1/(2*SNR));
    noise = sigma * No;

    for T=1:Timeslots
        Y(:,T)=H(:, :, T)*x(:,T);
        Ynoisy(:,T)=Y(:,T)+noise(:,T); % Προσθήκη Θορύβου
    end
    Ynoisy_osicmmse=Ynoisy;
    Ynoisy_osiczf=Ynoisy;

    W_ZF=zeros(NT,NR,Timeslots);
    Yest_zf=zeros(NT,Timeslots);
    for T=1:Timeslots
        %Δημιουργία του πίνακα αναστροφής της επίδρασης του καναλιού.
        W_ZF(:, :, T)=(H(:, :, T)'*H(:, :, T))\H(:, :, T)'; % Pseudoinverse
Matrix for ZF decoder
        for nt=1:NT
            Yest_zf(nt,T)=W_ZF(nt, :, T)*Ynoisy(:,T); % Ανακατασκευή των
μεταδοθέντων συμβόλων
        end
    end
    %Μετατροπή συμβόλων σε bits
RecSymbols_zf=reshape(Yest_zf,prod(size(Yest_zf)),1);
RecSymbols_zf=(1/scale)*RecSymbols_zf;
Recy_zf = demodulate(modem.qamdemod(M),RecSymbols_zf);
Recy_zf= de2bi(Recy_zf,'left-msb'); % Convert integers to bits.
Recy_zf=Recy_zf.';
Rbitstream_zf=reshape(Recy_zf,prod(size(Recy_zf)),1);

% Έλεγχος Σφαλμάτων
[NUM_ERRORS_zf(1) BER_zf(1)]=biterr(bitstream,Rbitstream_zf)

    W_MMSE=zeros(NT,NR,Timeslots);
    Yest_mmse=zeros(NT,Timeslots);
    for T=1:Timeslots
        %Δημιουργία του πίνακα αναστροφής της επίδρασης του καναλιού.
        W_MMSE(:, :, T)=(H(:, :, T)'*H(:, :, T)+(sigma^2)*eye(NT))\H(:, :, T)';
%MMSE Filter Construction
        for nt=1:NT
            Yest_mmse(nt,T)=W_MMSE(nt, :, T)*Ynoisy(:,T);
        end
    end
    %Μετατροπή συμβόλων σε bits

```

```

RecSymbols_mmse=reshape(Yest_mmse,prod(size(Yest_mmse)),1);
RecSymbols_mmse=(1/scale)*RecSymbols_mmse;
Recy_mmse = demodulate(modem.qamdemod(M),RecSymbols_mmse);
Recy_mmse= de2bi(Recy_mmse,'left-msb'); % Convert integers to bits.
Recy_mmse=Recy_mmse.';
Rbitstream_mmse=reshape(Recy_mmse,prod(size(Recy_mmse)),1);

% Έλεγχος Σφαλμάτων
[NUM_ERRORS_mmse(1) BER_mmse(1)]=biterr(bitstream,Rbitstream_mmse)

W_osiczf=zeros(NT,NR,Timeslots);
Yest_osiczf=zeros(NT,Timeslots);
for T=1:Timeslots
    H_hat=H(:, :, T);
    Order=[];
    index_array=[1:NT]; % Set of indices of signals to be detected
    for stage=1:NT
        %Δημιουργία του πίνακα αναστροφής της επίδρασης του καναλιού.
        W_osiczf = ((H_hat'*H_hat)^-1)*H_hat'; % Pseudoinverse Matrix
    for linear decoder
        norm_array=[];
        for i=1:NT-(stage-1) % Detection ordering
            norm_array(i) = norm(W_osiczf(i, :));
        end
        [val_min,index_min] = min(norm_array); % Ordering in SNR
        Order = [Order index_array(index_min)];
        index_array = index_array([1:index_min-1 index_min+1:end]);
        x_tmp(stage) = W_osiczf(index_min,:)*Ynoisy_osiczf(:,T); %
    Tx signal estimation
        x_tmp(stage)=(1/scale)*x_tmp(stage);
        X_tmp(stage) = demodulate(modem.qamdemod(M),x_tmp(stage));%
    Slicing
        X_hat(stage)= modulate(modem.gammod(M),X_tmp(stage));
        X_hat(stage)=scale*X_hat(stage);
        Ynoisy_tilde(:,T) = Ynoisy(:,T)-
    H_hat(:,index_min)*X_hat(stage); % Interference subtraction
        H_hat_tilde = H_hat(:, [1:index_min-1 index_min+1:NT-(stage-
    1)]); % New H
        H_hat = H_hat_tilde; Ynoisy_osiczf(:,T) = Ynoisy_tilde(:,T);
    end
        Yest_osiczf(Order,T) = X_hat;
    end
    %Μετατροπή συμβόλων σε bits

RecSymbols_osiczf=reshape(Yest_osiczf,prod(size(Yest_osiczf)),1);
RecSymbols_osiczf=(1/scale)* RecSymbols_osiczf;
Recy_osiczf = demodulate(modem.qamdemod(M),RecSymbols_osiczf);
Recy_osiczf= de2bi(Recy_osiczf,'left-msb'); % Convert integers
to bits.
Recy_osiczf=Recy_osiczf.';

Rbitstream_osiczf=reshape(Recy_osiczf,prod(size(Recy_osiczf)),1);
% Έλεγχος Σφαλμάτων

```

```

    [NUM_ERRORS_osiczf(1)
BER_osiczf(1)]=biterr(bitstream,Rbitstream_osiczf)

    W_osicmmse=zeros(NT,NR,Timeslots);
    Y_est_osicmmse=zeros(NT,Timeslots);
    for T=1:Timeslots
        H_hat=H(:, :, T);
        Order=[];
        index_array=[1:NT]; % Set of indices of signals to be detected
        for stage=1:NT
            W_osicmmse = (H_hat'*H_hat+(sigma^2)*eye(NT+1-stage))\H_hat'; %
MMSE FILTER MATRIX
            WH=W_osicmmse*H_hat;
            SINR=[];
            for i=1:NT-(stage-1)

                SINR(i)=(abs(WH(i,i))^2)/(sum(abs(WH(i,[1:i-1 i+1:NT-(
(stage-1)]))^2)+(sigma^2)*norm(W_osicmmse(i,:))^2);
            end
            [val_max,index_tmp] = max(SINR); % Ordering using SINR
            Order = [Order index_array(index_tmp)];
            index_array = index_array([1:index_tmp-1
index_tmp+1:end]);

            x_tmp(stage) = W_osicmmse(index_tmp,:)*Ynoisy_osicmmse(:,T);
% Tx signal estimation
            x_tmp(stage)=(1/scale)*x_tmp(stage);
            X_tmp(stage) = demodulate(modem.qamdemod(M),x_tmp(stage));
            X_hat(stage)= modulate(modem.qammod(M),X_tmp(stage));%
Slicing
            X_hat(stage)=scale*X_hat(stage);
            Ynoisy_tilde(:,T) = Ynoisy_osicmmse(:,T)-
H_hat(:,index_tmp)*X_hat(stage); % Interference subtraction
            H_hat_tilde = H_hat(:,[1:index_tmp-1 index_tmp+1:NT-(stage-
1)]); % New H
            H_hat = H_hat_tilde; Ynoisy_osicmmse(:,T) =
Ynoisy_tilde(:,T);
        end
        Y_est_osicmmse(Order,T) = X_hat;
    end
    %Μετατροπή συμβόλων σε bits

RecSymbols_osicmmse=reshape(Y_est_osicmmse,prod(size(Y_est_osicmmse)
),1);
    RecSymbols_osicmmse=(1/scale)*RecSymbols_osicmmse;
    Recy_osicmmse =
demodulate(modem.qamdemod(M),RecSymbols_osicmmse);
    Recy_osicmmse= de2bi(Recy_osicmmse,'left-msb'); % Convert
integers to bits.
    Recy_osicmmse=Recy_osicmmse.';

Rbitstream_osicmmse=reshape(Recy_osicmmse,prod(size(Recy_osicmmse)),
1);
    % Έλεγχος Σφαλμάτων

```

```

        [NUM_ERRORS_osicmmse(1)
BER_osicmmse(1)]=biterr(bitstream,Rbitstream_osicmmse)

    Yest_ml=zeros(NT,T);

    for T=1:Timeslots
        metric = 100000;
        for i = 1:M
            x_tmp(1) = scale*QAM_syms(i);
            for m = 1:M
                x_tmp(2) = scale*QAM_syms(m);
                metric_tmp=(norm(Ynoisy(:,T)-(H(:, :, T)*x_tmp.')))^2;
                if metric_tmp<metric
                    X_hat = x_tmp;
                    metric = metric_tmp;
                end
            end
        end
        Yest_ml(:,T)=X_hat.';
    end

    %Μετατροπή συμβόλων σε bits
    RecSymbols_ml=reshape(Yest_ml,prod(size(Yest_ml)),1);
    RecSymbols_ml=(1/scale)*RecSymbols_ml;
    Recy_ml = demodulate(modem.qamdemod(M),RecSymbols_ml);
    Recy_ml= de2bi(Recy_ml, 'left-msb'); % Convert integers to bits.
    Recy_ml=Recy_ml.';
    Rbitstream_ml=reshape(Recy_ml,prod(size(Recy_ml)),1);
    % Έλεγχος Σφαλμάτων
    [NUM_ERRORS_ml(1) BER_ml(1)]=biterr(bitstream,Rbitstream_ml)

end
figure
semilogy(EbNo_dB,BER_zf, '-k^', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB,BER_mmse, 'ro-', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB,BER_osiczf, 'ms-', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB,BER_osicmmse, 'cv-', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB,BER_ml, '-', 'LineWidth', 1.5)
legend('ZF', 'MMSE', 'OSIC ZF', 'OSIC
MMSE', 'ML', 'Location', 'SouthWest')
xlabel('Eb/No[dB]'); ylabel('BER');
title('MIMO RECEIVERS PERFORMANCE COMPARISON')
grid on
axis([0 20 10^-6,10^-0.5])

```

Sphere Decoder

```

%SPHERE_DECODER.m
function[X_hat]=sphere_decoder(Y,H,NT,M)

global x_list;
global x_now;

```

```

global x_hat;
global x_sliced;
global x_pre;
global real_constellation;
global R;
global radius_squared;
global x_metric;
global len;
global S;
global T;
QAM_table=[0:M-1]';
QAM_table = modulate(modem.qammod(M),QAM_table)';
scale = modnorm(QAM_table,'avpow',1);
QAM_table2=[real(QAM_table) imag(QAM_table)];
real_constellation=unique(QAM_table2);
l=length(real_constellation);
T=zeros(len,1);
S=zeros(len,1);
len=NT*2;
x_list = zeros(len,1);
x_now = zeros(len,1);
x_pre = zeros(len,1);
x_metric = 0;

Y =[real(Y); imag(Y)];
H =[real(H) -(imag(H)) ; imag(H) real(H)];
x_hat=((H'*H)^-1)*H'*Y;
x_hat=(1/scale)*x_hat;
x_sliced=QAM_16_slicer(x_hat);
[Q,R] = qr(H);
radius_squared = norm(R*(x_sliced-x_hat))^2;
transition = 1;

flag = 1;
while(transition<len+2)
    if transition == 0
        [flag,transition,radius_squared]=
radius_control(radius_squared,transition);
    elseif transition <= len
        [flag,transition] = stage_processing(flag,transition);
    elseif transition == len+1
        [flag,transition] = compare_vector_norm(transition);
    end
end
ML = x_pre;
for i=1:NT
    X_hat(i) = ML(i)+1i*ML(i+NT);
end
X_hat=X_hat.';

```

```

%stage_processing.m
function [flag,transition] = stage_processing(flag,transition)

```

```

global x_list x_metric x_now len
global real_constellation radius_squared
stage_index = len -(transition-1);
if flag==2
    radius_squared = x_metric;
end
if flag~=0
    [bound_lower bound_upper] = bound2(transition);
    for i =1:length(real_constellation)
        if bound_lower <= real_constellation(i)&&
real_constellation(i) <= bound_upper
            list_len = list_length(x_list(stage_index,:));
            x_list(stage_index,list_len+1) = real_constellation(i);
        end
    end
end
list_len = list_length(x_list(stage_index,:));
if list_len==0
    if x_metric == 0 || transition ~= 1
        flag = 0;
        transition = transition-1;
    elseif x_metric ~= 0 && transition == 1
        transition = len +2;
    end
else
    x_now(stage_index) = x_list(stage_index,1);
    x_list(stage_index,:) = [x_list(stage_index,[2:end]) 0];
    flag = 1;
    transition = transition+1;
end
end

```

```

%bound2.m
function [bound_lower,bound_upper]=bound2(transition)
global R radius_squared x_now x_hat T S
len = length(x_hat);

QQ=zeros(len,len);
for i=1:len
    QQ(i,i)=R(i,i)^2;
    for k=i+1:len
        QQ(i,k)=R(i,k)/R(i,i);
    end
end

index = len-(transition-1);
T(len)=radius_squared;
b=0;
for l=index+1:len
    b=b-QQ(index,l)*(x_now(l)-x_hat(l));
end
S(index)=x_hat(index)+b;
a=0;

```



```

for k=index+1:len
    a=a+R(index+1,k)*(x_now(k)-x_hat(k));
end
if transition>1
    T(index)=T(index+1)-(a)^2;
end

bound_lower = fix(-(sqrt(T(index)/QQ(index,index)))+S(index));
bound_upper = ceil((sqrt(T(index)/QQ(index,index)))+S(index));

```

```

%compare_vector_norm.m
function [flag,transition] = compare_vector_norm(transition)
global x_list x_pre x_metric x_now x_hat R len radius_squared
real_constellation
vector_identity = vector_comparison(x_pre,x_now);
if vector_identity==1
    len_total = 0;
    for i=1:len
        len_total = len_total + list_length(x_list(i,:));
    end
    if len_total==0
        transition = len+2;
        flag = 1;
    else
        transition = transition-1;
        flag =0;
    end
else
    metric_temp = norm(R*(x_now-x_hat))^2;
    if metric_temp<=radius_squared
        x_pre = x_now; x_metric = metric_temp;
        transition = 1;
        flag = 2;
        l=length(real_constellation);
        x_list=zeros(len,1);
        x_now=zeros(len,1);
    else
        transition = transition-1;
        flag =0;
    end
end
end

```

```

%radius_control.m
function
[flag,transition,radius_squared]=radius_control(radius_squared,trans
ition)
radius_squared = radius_squared*2;
transition = transition+1;
flag = 1;

```

```

%list_length.m
function [len]=list_length(list)
global real_constellation
l=length(real_constellation);
len = 0;
for i=1:l
if list(i)==0, break; else len = len+1; end
end

```

```

%qam_slicer.m
function [x_sliced]=QAM_4_slicer(x_hat)
find(x_hat<0)=-1;
find(x_hat>0)=1;
x_sliced=find';

```

```

%qam_slicer.m
function [x_sliced]=QAM_16_slicer(x_hat)
find(x_hat<-2)=-3;
find((x_hat>-2)&(x_hat<0))=-1;
find(x_hat>2)=3;
find((x_hat>0)&(x_hat<2))=1;
x_sliced=find';

```

```

%qam64_slicer.m
function [x_sliced]=QAM_64_slicer(x_hat)
find(x_hat<-6)=-7;
find((x_hat>-6)&(x_hat<-4))=-5;
find((x_hat>-4)&(x_hat<-2))=-3;
find((x_hat>-2)&(x_hat<0))=-1;
find((x_hat>0)&(x_hat<2))=1;
find((x_hat>2)&(x_hat<4))=3;
find((x_hat>4)&(x_hat<6))=5;
find(x_hat>6)=7;
x_sliced=find';

```

```

%vector_comparison.m
function [check]=vector_comparison(vector_1,vector_2)
check = 0;
len1 = length(vector_1); len2 = length(vector_2);
if len1 ~=len2,
    error('Vector size is different');
end
for column_num = 1:len1
if vector_1(column_num,1)==vector_2(column_num,1)
check = check + 1;
end
end
if check == len1, check = 1;

```

```

else check = 0;
end

```

```

%sphere_simulation.m

Timeslots=10000;
NT=2; NR=2;
M=16; %Constellation Size
b = log2(M); % Number of bits per symbol
QAM_nums=[0:M-1];
QAM_syms=modulate(modem.qammod(M),QAM_nums);
scale = modnorm(QAM_syms,'avpow',1);
EbNo_dB=[0:2:20];
SNR_dB=EbNo_dB+10*log10(b);

%Δημιουργία διανύσματος των προς μετάδοση bits
%Διαμόρφωση των προς μετάδοση bits χρησιμοποιώντας M-QAM
bitstream=randint(Timeslots*b*NT,1);
S=reshape(bitstream,b,length(bitstream)/b);
S=S.';
xsym = bi2de(S,'left-msb');
xsym = modulate(modem.qammod(M),xsym);
xsym=scale*xsym;
x=reshape(xsym,NT,Timeslots); % Διαμόρφωση πίνακα των προς μετάδοση
στοιχείων

Y=zeros(NR,Timeslots);
Ynoisy=zeros(NR,Timeslots);

No=(randn(NR,Timeslots) + 1i*randn(NR,Timeslots));
H=(randn(NR,NT,Timeslots)+1i*randn(NR,NT,Timeslots))/sqrt(2); %
Δημιουργία MIMO καναλιού για κάθε χρονοσχισημή

for l=1:length(EbNo_dB)
    SNR = 10^(SNR_dB(l)/10);
    sigma = sqrt(1/(2*SNR));
    noise = sigma * No;

    for T=1:Timeslots
        Y(:,T)=H(:, :, T)*x(:,T);
        Ynoisy(:,T)=Y(:,T)+noise(:,T); % Προσθήκη θορύβου
    end

    Yest_sphere=zeros(NT,T);
    for T=1:Timeslots
        [X_hat]=sphere_decoder(Ynoisy(:,T),H(:, :, T),NT,M);
        Yest_sphere(:,T)=X_hat;
    end

    %Μετατροπή συμβόλων σε bits

```

```

RecSymbols_sphere=reshape(Yest_sphere,prod(size(Yest_sphere)),1);
    Recy_sphere = demodulate(modem.qamdemod(M),RecSymbols_sphere);
    Recy_sphere= de2bi(Recy_sphere,'left-msb'); % Convert integers
to bits.
    Recy_sphere=Recy_sphere.';

Rbitstream_sphere=reshape(Recy_sphere,prod(size(Recy_sphere)),1);
    % Έλεγχος Σφαλμάτων
    [NUM_ERRORS_sphere(1)
BER_sphere(1)]=biterr(bitstream,Rbitstream_sphere)
end
figure
semilogy(EbNo_dB,BER_sphere,'-k','LineWidth', 1.5)
legend('Sphere Decoding','Location','SouthWest')
xlabel('Eb/No[dB]'); ylabel('BER');
title('Sphere Decoder Performance');
grid on
axis([0 20 10^-5,10^0])

```

```

%sphere_vs_ml.m
Timeslots=10000;

M=16; %Constellation Size
b = log2(M); % Number of bits per symbol
QAM_nums=[0:M-1];
QAM_syms=modulate(modem.qammod(M),QAM_nums);
scale = modnorm(QAM_syms,'avpow',1);
EbNo_dB=[0:2:20];
SNR_dB=EbNo_dB+10*log10(b);

for Icase=1:3
    if Icase==1, NT=2; NR=2;
    elseif Icase==2, NT=3; NR=3;
    else NT=4; NR=4;
    end
    %Δημιουργία διανύσματος των προς μετάδοση bits
    %Διαμόρφωση των προς μετάδοση bits χρησιμοποιώντας M-QAM
    bitstream=randint(Timeslots*b*NT,1);
    S=reshape(bitstream,b,length(bitstream)/b);
    S=S.';
    xsym = bi2de(S,'left-msb');
    xsym = modulate(modem.qammod(M),xsym);
    xsym=scale*xsym;
    x=reshape(xsym,NT,Timeslots); % Διαμόρφωση πίνακα των προς μετάδοση
στοιχείων

Y=zeros(NR,Timeslots);
Ynoisy=zeros(NR,Timeslots);

No=(randn(NR,Timeslots) + 1i*randn(NR,Timeslots));

```

```

H=(randn(NR,NT,Timeslots)+1i*randn(NR,NT,Timeslots))/sqrt(2); %
Δημιουργία MIMO καναλιού για κάθε χρονοσχισμή

for l=1:length(EbNo_dB)
    SNR = 10^(SNR_dB(l)/10);
    sigma = sqrt(1/(2*SNR));
    noise = sigma * No;

    for T=1:Timeslots
        Y(:,T)=H(:, :, T)*x(:,T);
        Ynoisy(:,T)=Y(:,T)+noise(:,T); % Προσθήκη θορύβου
    end

    Yest_sphere=zeros(NT,T);
    for T=1:Timeslots
        [X_hat]=sphere_decoder(Ynoisy(:,T),H(:, :, T),NT,M);
        Yest_sphere(:,T)=X_hat;
    end
    %Μετατροπή συμβόλων σε bits

RecSymbols_sphere=reshape(Yest_sphere,prod(size(Yest_sphere)),1);
Recy_sphere = demodulate(modem.qamdemod(M),RecSymbols_sphere);
Recy_sphere= de2bi(Recy_sphere,'left-msb'); % Convert integers
to bits.
Recy_sphere=Recy_sphere.';

Rbitstream_sphere=reshape(Recy_sphere,prod(size(Recy_sphere)),1);
% Έλεγχος Σφαλμάτων
[NUM_ERRORS_sphere(Icase,l)
BER_sphere(Icase,l)]=biterr(bitstream,Rbitstream_sphere)

Yest_ml=zeros(NT,T);
if NT==2
for T=1:Timeslots
    metric = 100000;
    for i = 1:M
        x_tmp(1) = scale*QAM_syms(i);
        for m = 1:M
            x_tmp(2) = scale*QAM_syms(m);
            metric_tmp=(norm(Ynoisy(:,T)-(H(:, :, T)*x_tmp.')))^2;
            if metric_tmp<metric
                X_hat = x_tmp;
                metric = metric_tmp;
            end
        end
    end
    Yest_ml(:,T)=X_hat.';
end
elseif NT==3

for T=1:Timeslots
    metric = 100000;
    for i = 1:M

```

```

x_tmp(1) = scale*QAM_syms(i);
for n = 1:M
x_tmp(2) = scale*QAM_syms(n);
for m = 1:M
x_tmp(3) = scale*QAM_syms(m);
metric_tmp=(norm(Ynoisy(:,T)-(H(:, :, T)*x_tmp.')))^2;
if metric_tmp<metric
X_hat = x_tmp;
metric = metric_tmp;
end
end
end
end
Yest_ml(:,T)=X_hat.';
end
elseif NT==4

for T=1:Timeslots
metric = 100000;
for i = 1:M
x_tmp(1) = scale*QAM_syms(i);
for o = 1:M
x_tmp(2) = scale*QAM_syms(o);
for n = 1:M
x_tmp(3) = scale*QAM_syms(n);
for m = 1:M
x_tmp(4) = scale*QAM_syms(m);
metric_tmp=(norm(Ynoisy(:,T)-
(H(:, :, T)*x_tmp.')))^2;
if metric_tmp<metric
X_hat = x_tmp;
metric = metric_tmp;
end
end
end
end
end
end
Yest_ml(:,T)=X_hat.';
end
end
%Μετατροπή συμβόλων σε bits
RecSymbols_ml=reshape(Yest_ml,prod(size(Yest_ml)),1);
RecSymbols_ml=(1/scale)*RecSymbols_ml;
Recy_ml = demodulate(modem.qamdemod(M),RecSymbols_ml);
Recy_ml= de2bi(Recy_ml,'left-msb'); % Convert integers to bits.
Recy_ml=Recy_ml.';
Rbitstream_ml=reshape(Recy_ml,prod(size(Recy_ml)),1);
% Έλεγχος Σφαλμάτων
[NUM_ERRORS_ml(Icase,1)
BER_ml(Icase,1)]=biterr(bitstream,Rbitstream_ml)
end
end
figure

```

```

subplot(1,2,1), semilogy(EbNo_dB,BER_sphere(1,:), '-bo', 'LineWidth',
1.5)
hold on, semilogy(EbNo_dB,BER_sphere(2,:), '-md', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB,BER_sphere(3,:), '-k', 'LineWidth', 1.5)
legend('2x2', '3x3', '4x4', 'Location', 'SouthWest')
xlabel('Eb/No[dB]'); ylabel('BER');
title('Sphere Decoder Performance');
grid on
axis([0 20 10^-5,10^0])
subplot(1,2,2), semilogy(EbNo_dB,BER_ml(1,:), '-bo', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB,BER_ml(2,:), '-md', 'LineWidth', 1.5)
hold on, semilogy(EbNo_dB,BER_ml(3,:), '-k', 'LineWidth', 1.5)

legend('2x2', '3x3', '4x4', 'Location', 'SouthWest')
xlabel('Eb/No[dB]'); ylabel('BER');
title('ML Decoder Performance');
grid on
axis([0 20 10^-5,10^0])

```