



Πανεπιστήμιο Πειραιώς  
Τμήμα Ψηφιακών Συστημάτων  
ΠΜΣ Ψηφιακές Επικοινωνίες και Δίκτυα

Υλοποίηση MIMO Πομποδέκτη με  
Χωρο-Χρονική Κωδικοποίηση σε Επαναπρογραμματιζόμενους  
Πομποδέκτες

Development of an STC MIMO transceiver in SDR

Καρανάτσιος Δημήτριος  
Αριθμός Μητρώου: ME 11080

Επιβλέπων: Κανάτας Αθανάσιος

Πειραιάς  
Σεπτέμβριος 2015



## Περίληψη

Τα συστήματα SDR (Software Defined Radio), δεν αποτελούν μία καινούρια ιδέα στον χώρο των τηλεπικοινωνιών. Τα τελευταία χρόνια, όμως, λόγω της μεγάλης ανάπτυξης των επιστημών της πληροφορικής και των ηλεκτρονικών, έχουν αποκτήσει μεγάλες δυνατότητες και γνωρίζουν όλο και μεγαλύτερη εφαρμογή στην ασύρματη μετάδοση δεδομένων. Σε αυτό συμβάλλουν τα μεγάλα πλεονεκτήματα που χαρακτηρίζουν τα συστήματα αυτά, τα σημαντικότερα εκ των οποίων είναι το χαμηλό κόστος, η ευκολία της εγκατάστασης και η εύκολη παραμετροποίηση της λειτουργικότητάς τους.

Αντικείμενο της συγκεκριμένης διπλωματικής εργασίας αποτέλεσε η μελέτη των συστημάτων SDR και ειδικότερα των συσκευών USRP (Universal Software Radio Peripherals). Επιπλέον, έγινε η ανάπτυξη εφαρμογής σε γλώσσα C++ ενός transceiver (πομπός-δέκτης), χωροχρονικού συστήματος MIMO 2x2, με διαμόρφωση OFDM και κωδικοποίηση Alamouti.

## Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον Κωνσταντίνο Μαλιάτσο, η βοήθεια του οποίου υπήρξε σημαντικότερη για την ολοκλήρωση της συγκεκριμένης διπλωματικής εργασίας.

# Περιεχόμενα

<b>Περίληψη.....</b>	<b>3</b>
<b>Ευχαριστίες.....</b>	<b>3</b>
Περιεχόμενα .....	4
<b>Κατάλογος Σχημάτων .....</b>	<b>6</b>
<b>Κατάλογος Πινάκων .....</b>	<b>7</b>
<b>Συνομογραφίες.....</b>	<b>8</b>
<b>Κεφάλαιο 1 - Περιγραφή Συστήματος .....</b>	<b>9</b>
1.1 Software Defined Radio (SDR) .....	9
1.2 Universal Software Radio Peripheral (USRP) .....	10
1.3 Βασικά Χαρακτηριστικά Συστήματος .....	13
1.4 Τοπολογία Συστήματος.....	14
<b>Κεφάλαιο 2 - Γραφικό Περιβάλλον Εφαρμογής (GUI) .....</b>	<b>15</b>
2.1 Γραφικό Περιβάλλον Πομπού.....	15
<b>Κεφάλαιο 3 - Διαγραμματική Απεικόνιση Λειτουργίας Πομπού-Δέκτη .....</b>	<b>21</b>
3.1 Διάγραμμα Λειτουργίας Πομπού .....	21
3.2 Διάγραμμα λειτουργίας Δέκτη .....	23
<b>Κεφάλαιο 4 - Ψηφιακή Διαμόρφωση Σήματος .....</b>	<b>25</b>
4.1 Είδη Τεχνικών Ζωνοπερατής Ψηφιακής Διαμόρφωσης.....	25
4.2 Διαμόρφωση Μετατόπισης Φάσης.....	25

(PSK-Phase Shift Keying) .....	25
4.3 Ορθογώνια Διαμόρφωση Μετατόπισης Φάσης.....	26
(Quadrature Phase Shift Keying - QPSK) .....	26
<b>Κεφάλαιο 5 - Συστήματα Πολλαπλών Κεραιών Εκπομπής -Λήψης .....</b>	<b>33</b>
5.1 Χωροχρονική Κωδικοποίηση Alamouti.....	33
5.2 Συστήματα Πολλαπλών Κεραιών Εκπομπής – Λήψης (MIMO).....	34
5.3 Κωδικοποίηση OFDM - Alamouti 2x2 .....	37
<b>Κεφάλαιο 6 - Ορθογωνική Πολυπλεξία Διαίρεσης Συχνότητας (OFDM).....</b>	<b>39</b>
6.1 Γενική Περιγραφή .....	39
6.2 Προβλήματα Συγχρονισμού.....	40
6.3 Κυκλικό Πρόθεμα.....	42
6.4 Βασική Υλοποίηση Συστήματος OFDM.....	43
6.4.1 Πομπός.....	43
6.4.2 Δέκτης .....	44
6.5 Επιλογή παραμέτρων συστήματος.....	45
<b>Κεφάλαιο 7 - Ανάλυση Κώδικα Εφαρμογής.....</b>	<b>47</b>
7.1 Ανάλυση Κώδικα Πομπού.....	47
7.1.1 Βήμα 1: Δημιουργία των δεδομένων (παράλληλη μορφή).....	47
7.1.2 Βήμα 2: Διαμόρφωση QPSK.....	48
7.1.3 Βήμα 3: Κωδικοποίηση Alamouti .....	50
7.1.4 Βήμα 4: IFFT-(Αντίστροφος Ταχύς Μετασχηματισμός Fourier).....	53
7.1.5 Βήμα 5: Μετατροπή σήματος από παράλληλο σε σειριακό .....	55
7.1.6 Βήμα 6: Προσθήκη Κυκλικού προθέματος .....	55
7.1.7 Βήμα 7: Interpolation.....	57
7.1.8 Βήμα 8: Συνελκτικό Φιλτράρισμα - Convolution Filtering (Square Root Raised Cosine Filtering) 58	
7.1.9 Βήμα 9: Μετάδοση σήματος .....	60
7.2 Ανάλυση Κώδικα Δέκτη .....	62
7.2.1 Βήμα 1: Λήψη Σήματος.....	62
7.2.2 Βήμα 2: Συνελκτικό Φιλτράρισμα - Convolution Filtering (Square Root Raised Cosine Filtering) 64	
7.2.3 Βήμα 3: Decimation .....	65

7.2.4	Βήμα 4: Συγχρονισμός Πομπού – Δέκτη .....	67
7.2.5	Βήμα 5: Αφαίρεση Κυκλικού Προθέματος .....	70
7.2.6	Βήμα 6: Μετατροπή σειριακού σήματος σε παράλληλο .....	72
7.2.7	Βήμα 7: Fast Fourier Transform (Ταχύς Μετασχηματισμός Fourier) .....	73
7.2.8	Βήμα 8 - 9: Εκτίμηση καναλιού – Αποκωδικοποίηση Alamouti .....	75
<b>Κεφάλαιο 8 - Εργαστηριακές Μετρήσεις Συστήματος .....</b>		<b>79</b>
8.1	Εργαστηριακή Διάταξη .....	79
<b>Βιβλιογραφία.....</b>		<b>85</b>

## Κατάλογος Σχημάτων

1.1	Βασικό σχεδιάγραμμα λειτουργίας συστημάτων SDR (σελ. 9)
1.2	Βασικά Block λειτουργίας συστήματος SDR-USRP2 (A,B,C) (σελ. 11)
1.3	Διάγραμμα λειτουργίας DDC (Digital Down Converter) (σελ. 12)
1.4	Τοπολογία συστήματος SDR-USRP 2 (σελ. 14)
2.1	Γραφικό περιβάλλον πομπού (σελ. 16)
2.2	Γραφικό περιβάλλον δέκτη (σελ. 18)
3.1	Διάγραμμα λειτουργίας πομπού (σελ. 22)
3.2	Διάγραμμα λειτουργίας δέκτη (σελ. 24)
4.1	QPSK αστερισμός (constellation) (σελ.27)
4.2	Διαμορφωτής QPSK (σελ. 28)
4.3	Διαμόρφωση QPSK (σελ. 29)
4.4	Αποδιαμορφωτής QPSK (σελ. 30)
5.1	Σύστημα Alamouti STC 2x2 (σελ. 34)
5.2	Μετασχηματισμός ιδιάζουσων τιμών σε δίαυλο MIMO (σελ. 37)
6.1	Σειριακή και παράλληλη μετάδοση δεδομένων (σελ. 39)
6.2	Σύστημα OFDM με 3 υποκανάλια (σελ. 40)
6.3	Προσθήκη Κυκλικού Προθέματος (Cyclic Prefix) (σελ. 43)
7.1	Bits Δεδομένων (σελ. 48)
7.2	Δημιουργία vector qpskModulatedVector (σελ. 50)

- 7.3 Δεδομένα μετά την διαμόρφωση QPSK (σελ. 51)
- 7.4 Δεδομένα μετά την κωδικοποίηση Alamouti (σελ. 52)
- 7.5 Vector alamoutiSymbolsVectorAntenna (σελ. 52)
- 7.6 Εκτέλεση IFFT (σελ. 54)
- 7.7 Παράλληλο σήμα (σελ. 56)
- 7.8 Προσθήκη Κυκλικού Προθέματος (σελ. 56)
- 7.9 Σειριακό σήμα μετά την προσθήκη Κυκλικού Προθέματος (σελ. 56)
- 7.10 Σειριακό σήμα μετά το Interpolation (σελ. 57)
- 7.11 Λαμβανόμενο σήμα 1<sup>ης</sup> κεραίας δέκτη (σελ. 63)
- 7.12 Μορφή του λαμβανόμενου σήματος (1<sup>η</sup> κεραία) μετά το Convolution Filtering (σελ. 65)
- 7.13 Λαμβανόμενο σήμα (1<sup>η</sup> κεραία) μετά το RRC filtering (σελ. 66)
- 7.14 Λαμβανόμενο σήμα (1<sup>η</sup> κεραία) μετά το decimation (σελ. 66)
- 7.15 Χρονική επικάλυψη αποστολής και λήψης σήματος (σελ. 67)
- 7.16 Διαδικασία IFFT στα σύμβολα του preamble του πρώτου OFDM frame (σελ. 68)
- 7.17 Διάρθρωση του λαμβανόμενου σήματος (1<sup>η</sup> κεραία) σε windows (σελ. 68)
- 7.18 Υπολογισμός Σημείου Συγχρονισμού SP (σελ. 69)
- 7.19 Αφαίρεση Κυκλικού Προθέματος (σήμα 1<sup>ης</sup> κεραίας) (σελ. 71)
- 7.20 Σύμβολα Κυκλικού Προθέματος που αφαιρούνται (σήμα 1<sup>ης</sup> κεραίας) (σελ. 72)
- 7.21 Σειριακή μορφή σήματος (1<sup>η</sup> κεραία) (σελ. 72)
- 7.22 Παράλληλη μορφή σήματος (σελ. 73)
- 7.23 Σύμβολα 1<sup>ης</sup> κεραίας μετά τον μετασχηματισμό IFFT (σελ. 74)
- 7.24 Σύμβολα 1<sup>ου</sup> υποκαναλιού OFDM (1<sup>η</sup> – 2<sup>η</sup> κεραία) (σελ. 76)
- 7.25 Ληφθέντα σύμβολα μετά την αποκωδικοποίηση Alamouti (σελ. 78)

## Κατάλογος Πινάκων

- 1.1 Βασικά χαρακτηριστικά συσκευών USRP-N210 (σελ. 9)
- 4.1 Συντεταγμένες σήματος QPSK (σελ. 27)
- 7.1 Αντιστοίχιση bits σε σύμβολα QPSK (σελ. 49)
- 8.1 Μηδενικά subcarriers (σελ. 79)

# Συντομογραφίες

<b>ADC</b>	Analog to Digital converter
<b>AGC</b>	Automatic Gain Controller
<b>ASK</b>	Amplitude Shift Keying
<b>DAC</b>	Digital to Analog Converter
<b>DDC</b>	Digital Down Converter
<b>DUC</b>	Digital Up Converter
<b>FDD</b>	Frequency Division Duplex
<b>FDM</b>	Frequency Division Multiplexing
<b>FFT</b>	Fast Fourier Transform
<b>FPGA</b>	Field Programmable Gate Array
<b>GUI</b>	Graphical User Interface
<b>IFFT</b>	Inverse Fast Fourier Transform
<b>MIMO</b>	Multiple Input-Multiple Output
<b>MISO</b>	Multiple Input-Single Output
<b>NCO</b>	Numerically Controlled Oscillator
<b>OFDM</b>	Orthogonal Frequency Division Multiplexing
<b>PSK</b>	Phase Shift Keying
<b>QAM</b>	Quadrature Amplitude Modulation
<b>QPSK</b>	Quadrature Phase Shift Keying
<b>RF</b>	Radio Frequency
<b>RRC</b>	Root Raised Cosine
<b>SDR</b>	Software Defined Radio
<b>SIMO</b>	Single Input-Multiple Output
<b>SMA</b>	Subminiature Model A
<b>STC</b>	Space Time Coding
<b>SVD</b>	Singular Value Decomposition
<b>TDD</b>	Time Division Duplex
<b>UHD</b>	Usrp Hardware Driver
<b>USRP</b>	Universal Software Radio Peripheral

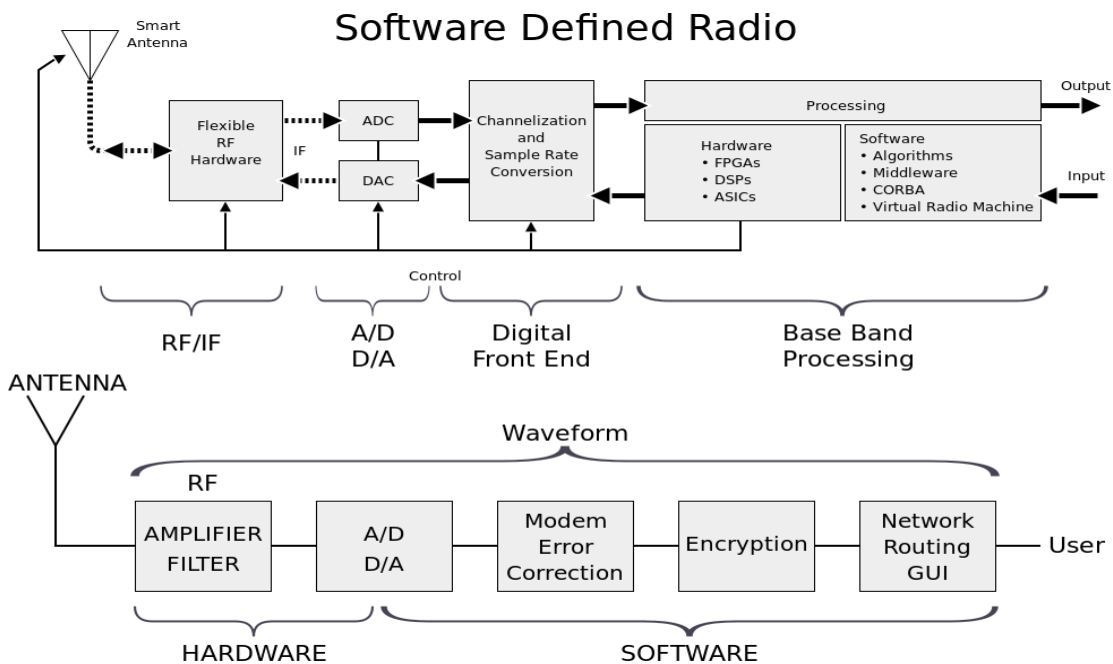


# Κεφάλαιο 1 - Περιγραφή Συστήματος

## 1.1 Software Defined Radio (SDR)

Στα κλασικά ασύρματα συστήματα, οι περισσότερες λειτουργίες που αφορούν την αποστολή και τη λήψη σήματος πραγματοποιούνται με τη βοήθεια ηλεκτρονικών κυκλωμάτων ή στοιχείων hardware, όπως είναι οι μίκτες, τα φίλτρα, οι ενισχυτές, οι διαμορφωτές/αποδιαμορφωτές κ.α. Αντίθετα, στα συστήματα SDR οι λειτουργίες αυτές πραγματοποιούνται από λογισμικό, γραμμένο σε κάποια κοινή γλώσσα προγραμματισμού, το οποίο εκτελείται σε ειδικούς ενσωματωμένους επεξεργαστές ή ακόμη και σε απλούς προσωπικούς υπολογιστές (υπολογιστές-hosts).

Όπως φαίνεται στο Σχήμα 1, ένα σύστημα SDR, στην πιο απλή του μορφή αποτελείται από έναν υπολογιστή-host, μία κεραία και τους αντίστοιχους μετατροπείς ADC (Analog to Digital Converter στον δέκτη) ή DAC (Digital to Analog Converter στον πομπό), οι οποίοι για τα συστήματα SDR καλούνται και Digitizers. Η επεξεργασία του σήματος από λογισμικό, προσδίδει στα συστήματα SDR μεγάλη προσαρμοστικότητα και ευχρηστία, διότι με τον τρόπο αυτό, μπορούν να υποστηριχθούν διαφορετικές τεχνικές μετάδοσης και πρωτόκολλα επικοινωνίας.



Σχήμα 1.1 – Βασικό σχεδιάγραμμα λειτουργίας συστημάτων SDR

Όπως φαίνεται στο Σχήμα 1.1, τα κυριότερα στάδια της επεξεργασίας του σήματος, τόσο κατά την εκπομπή όσο και κατά την λήψη, εκτελούνται από ρουτίνες λογισμικού. Συνεπώς, η οποιαδήποτε αλλαγή στη μορφή του σήματος, συνεπάγεται την αλλαγή συγκεκριμένων modules λογισμικού, κάτι που μειώνει τον απαιτούμενο χρόνο και το κόστος. Στην περίπτωση των κλασικών hardware συστημάτων, αντίστοιχες αλλαγές απαιτούν την αντικατάσταση ολόκληρων κυκλωμάτων και συσκευών, κάτι που συνεπάγεται αυξημένο κόστος αλλά και χρόνο υλοποίησης. Στην πράξη, παρά την συνεχή βελτίωση των συστημάτων SDR, η ασύρματη μετάδοση και λήψη σήματος χωρίς την χρήση επιπλέον ηλεκτρονικών κυκλωμάτων και στοιχείων, δεν είναι εύκολα εφικτή. Κύριος λόγος για αυτό, είναι η δυσκολία μετατροπής του αναλογικού σήματος σε ψηφιακό (και αντιστρόφως από ψηφιακό σε αναλογικό), επιτυγχάνοντας ταυτόχρονα, υψηλό ρυθμό και μεγάλη ακρίβεια, διότι η επεξεργαστική ισχύς των συστημάτων SDR, η οποία συνήθως είναι περιορισμένη, επιβάλλει ένα άνω όριο στις επιδόσεις αυτών.

## 1.2 Universal Software Radio Peripheral (USRP)



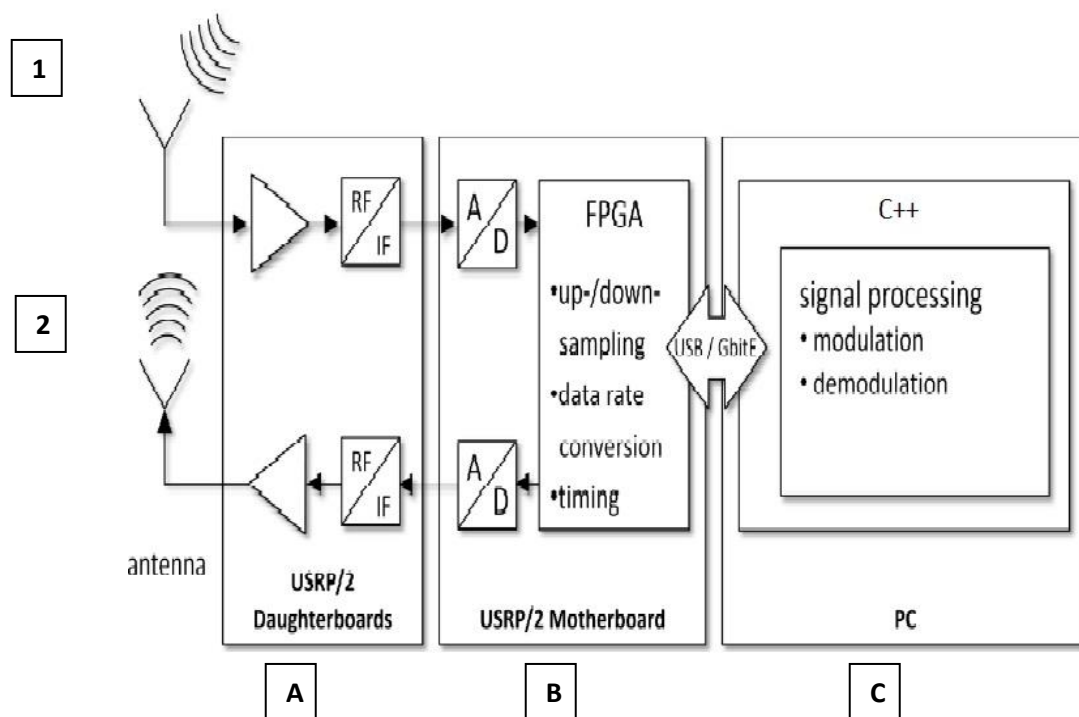
**Εικόνα 1.1 – Συσκευή USRP (σειρά N210)**

Οι συσκευές USRP (Universal Software Radio Peripheral), αποτελούν μία από τις συνηθέστερες επιλογές για την εγκατάσταση και αξιολόγηση συστημάτων SDR, καθώς και την ζεύξη του ψηφιακού σήματος (από τον υπολογιστή-host) με το αναλογικό (RF-κεραία). Μία από τις πρόσφατες σειρές συσκευών USRP είναι η USRP-N210 (Εικόνα 1.1), η οποία χρησιμοποιήθηκε για την υλοποίηση της συγκεκριμένης διπλωματικής εργασίας. Τα βασικά χαρακτηριστικά των συσκευών της σειράς αυτής είναι τα παρακάτω:

<b>ADC</b>	100 MS/14 bit
<b>DAC</b>	400 MS/16 bit
<b>Mixer</b>	(decimation-interpolation) Προγραμματιζόμενα
<b>Μέγιστο Bandwidth</b>	50 MHz
<b>Σύνδεση με PC</b>	Gigabit Ethernet (1000 Mbit/s)
<b>RF Range</b>	DC-6 GHz, εξαρτάται από την ενσωματωμένη κάρτα που χρησιμοποιείται
<b>Firmware version</b>	12.3
<b>FPGA version</b>	10.0

**Πίνακας 1.1 - Βασικά χαρακτηριστικά συσκευών USRP-N210**

Η λειτουργία των συσκευών USRP, όπως και άλλων συσκευών SDR, μπορεί να χωριστεί σε τρία βασικά block, όπως φαίνεται στο Σχήμα 1.2:

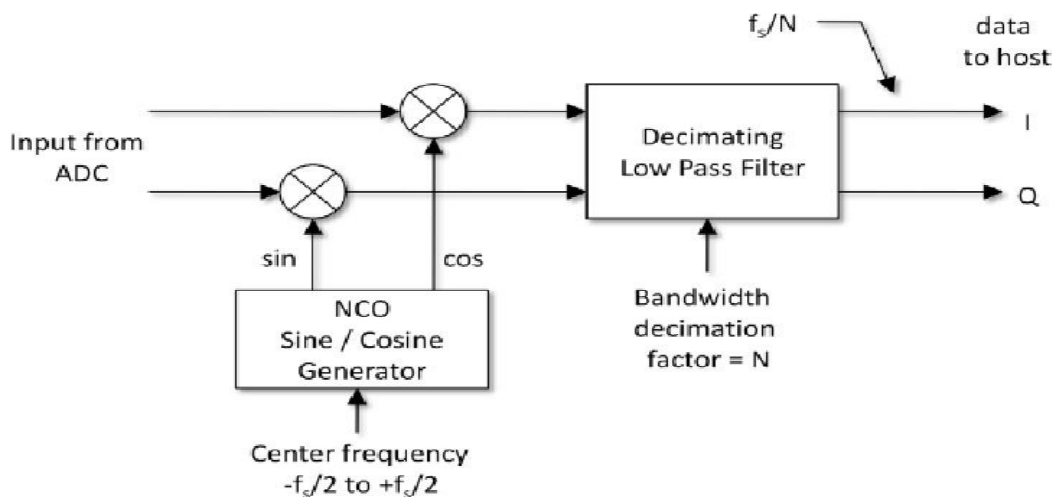


**Σχήμα 1.2 – Βασικά Block λειτουργίας συστήματος SDR-USRP2 (A,B,C).  
(1-Λήψη, 2-Εκπομπή)**

Το block A αποτελεί την πύλη σύνδεσης (front-end) του συστήματος με το αναλογικό πεδίο (RF). Το αναλογικό σήμα λαμβάνεται (1) ή εκπέμπεται (2) από απλές κεραίες ή από ενσωματωμένες κάρτες (daughter boards), που συνδέονται με την συσκευή μέσω ειδικών υποδοχών SMA (Subminiature Model A). Γενικά, η εκπομπή και η λήψη σήματος εκτελούνται

κατά τρόπο ανεξάρτητο μεταξύ τους (σε ορισμένες συσκευές USRP, μπορούν να εκτελεστούν ταυτόχρονα). Η συχνότητα λειτουργίας (DC-6 GHz), μπορεί να παραμετροποιηθεί ενώ εξαρτάται και από τον τύπο ενσωματωμένης κάρτας που χρησιμοποιείται .

Η μετατροπή του αναλογικού σήματος σε ψηφιακό και αντίστροφα, εκτελείται από το block B. Κατά την λήψη του σήματος (1), η δειγματοληψία και μετατροπή του εκτελείται από τους μετατροπείς της μητρικής κάρτας της συσκευής USRP. Τα ψηφιακά σύμβολα που προκύπτουν προωθούνται στους ψηφιακούς μετατροπείς DDC του FPGA (Field Programmable Gate Array), απομονώνοντας τις απαιτούμενες, από τον χρήστη, τιμές συχνότητας στον επιθυμητό ρυθμό μετάδοσης. Κατά την εκπομπή του σήματος πραγματοποιείται η αντίστροφη διαδικασία (2), με χρήση μετατροπέων DAC. Στο Σχήμα 1.3 παρουσιάζεται η λειτουργία ενός μετατροπέα DDC (Digital Down Converter).



**Σχήμα 1.3: Διάγραμμα λειτουργίας DDC (Digital Down Converter)**

Κατά την λήψη του σήματος (1), τα ψηφιακά δείγματα που δημιουργούνται από τον μετατροπέα ADC, πολλαπλασιάζονται με τιμές συναρτήσεων ημιτόνου και συνημιτόνου, διαχωρίζοντας και παράγοντας τις τιμές I και Q του συμβόλου. Η συχνότητα δημιουργείται από έναν ψηφιακό ταλαντωτή NCO (Numerically Controlled Oscillator), ο οποίος παράγει μία κυματομορφή διακριτού χρόνου και πλάτους, εντός του FPGA. Η χρήση του ταλαντωτή NCO, επιτρέπει την ταχύτατη μεταβολή συχνότητας. Τέλος εκτελείται μείωση του αριθμού των δειγμάτων (down-sampling) κατά έναν παράγοντα  $1/N$ . Στην περίπτωση της εκπομπής σήματος (2), πραγματοποιείται η αντίστροφη διαδικασία με χρήση μετατροπέων DUC (Digital Up Converter).

Το block C, περιλαμβάνει τη επεξεργασία του σήματος από το λογισμικό που χρησιμοποιείται στο σύστημα SDR. Ο προγραμματισμός των συσκευών USRP μπορεί να γίνει

αποκλειστικά με χρήση της γλώσσας C++, όπως έγινε στην συγκεκριμένη διπλωματική, είτε με συνδυασμό των γλωσσών C++ και Python (βιβλιοθήκη GNU-Radio). Η μεταφορά δεδομένων μεταξύ της συσκευής USRP με τον υπολογιστή-host γίνεται μέσω σύνδεσης Gigabit Ethernet, η οποία επιτρέπει την μεταφορά δεδομένων με μέγιστο ρυθμό 50 MS/s.

### 1.3 Βασικά Χαρακτηριστικά Συστήματος

Στους υπολογιστές-host του πομπού και του δέκτη χρησιμοποιήθηκε το λειτουργικό σύστημα **Ubuntu 14.04.2 LTS**.

Για την ανάπτυξη της εφαρμογής, χρησιμοποιήθηκε η γλώσσα προγραμματισμού C++ και πιο συγκεκριμένα οι παρακάτω βιβλιοθήκες:

- 1) C++ 11: Πρότυπη βιβλιοθήκη της C++ (έκδοση 2011).
- 2) UHD (USRP HARDWARE DRIVER - version 003.008.005-release): Αποτελεί το API (Application Program Interface) για την επικοινωνία μεταξύ του υπολογιστή-host και των συσκευών USRP που είναι συνδεδεμένες με αυτόν. Το download και η εγκατάσταση της βιβλιοθήκης γίνεται με χρήση της παρακάτω εντολής στο command line interface του υπολογιστή-host:

```
sudo apt-get install libuhd-dev libuhd003 uhd-host
```

Με την εγκατάσταση της βιβλιοθήκης UHD, εκτελείται η εγκατάσταση της βιβλιοθήκης boost (version 105400) της γλώσσας C++, η οποία είναι απαραίτητη για την λειτουργία των συσκευών USRP.

- 3) FFTW-3 (version 3.3.4): Χρησιμοποιείται στον Αντίστροφο Μετασχηματισμό Fourier (πομπός) αλλά και στον Ταχύ Μετασχηματισμό Fourier στον δέκτη. Το download και η εγκατάσταση της βιβλιοθήκης γίνεται με χρήση της παρακάτω εντολής στο command line interface του υπολογιστή-host:

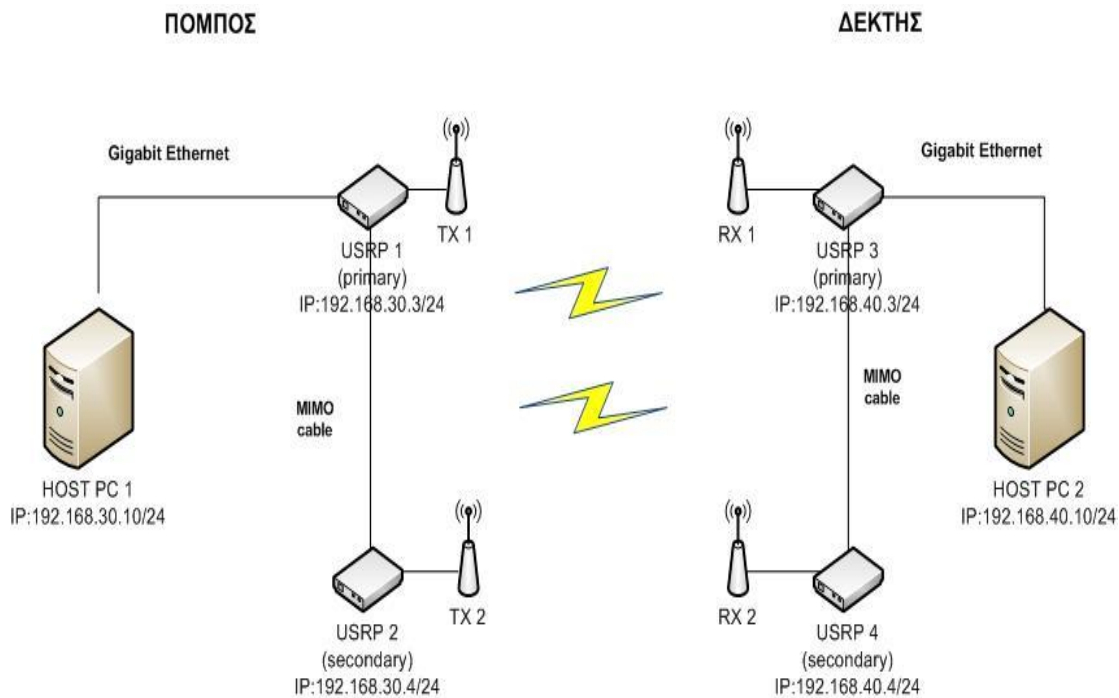
```
sudo apt-get install fftw3 fftw3-dev pkg-config
```

- 4) EIGEN (version 3.2.5): Χρησιμοποιείται στην εκτέλεση των αλγεβρικών πράξεων κατά τον συγχρονισμό πομπού-δέκτη, αλλά και κατά τον υπολογισμό των channel coefficients. Το download και η εγκατάσταση της βιβλιοθήκης γίνεται με χρήση της παρακάτω εντολής στο command line interface του υπολογιστή host:

```
sudo apt-get install libeigen3-dev
```

## 1.4 Τοπολογία Συστήματος

Η τοπολογία του συστήματος SDR που αναπτύχθηκε στα πλαίσια της συγκεκριμένης εργασίας είναι αυτή του Σχήματος 1.4:



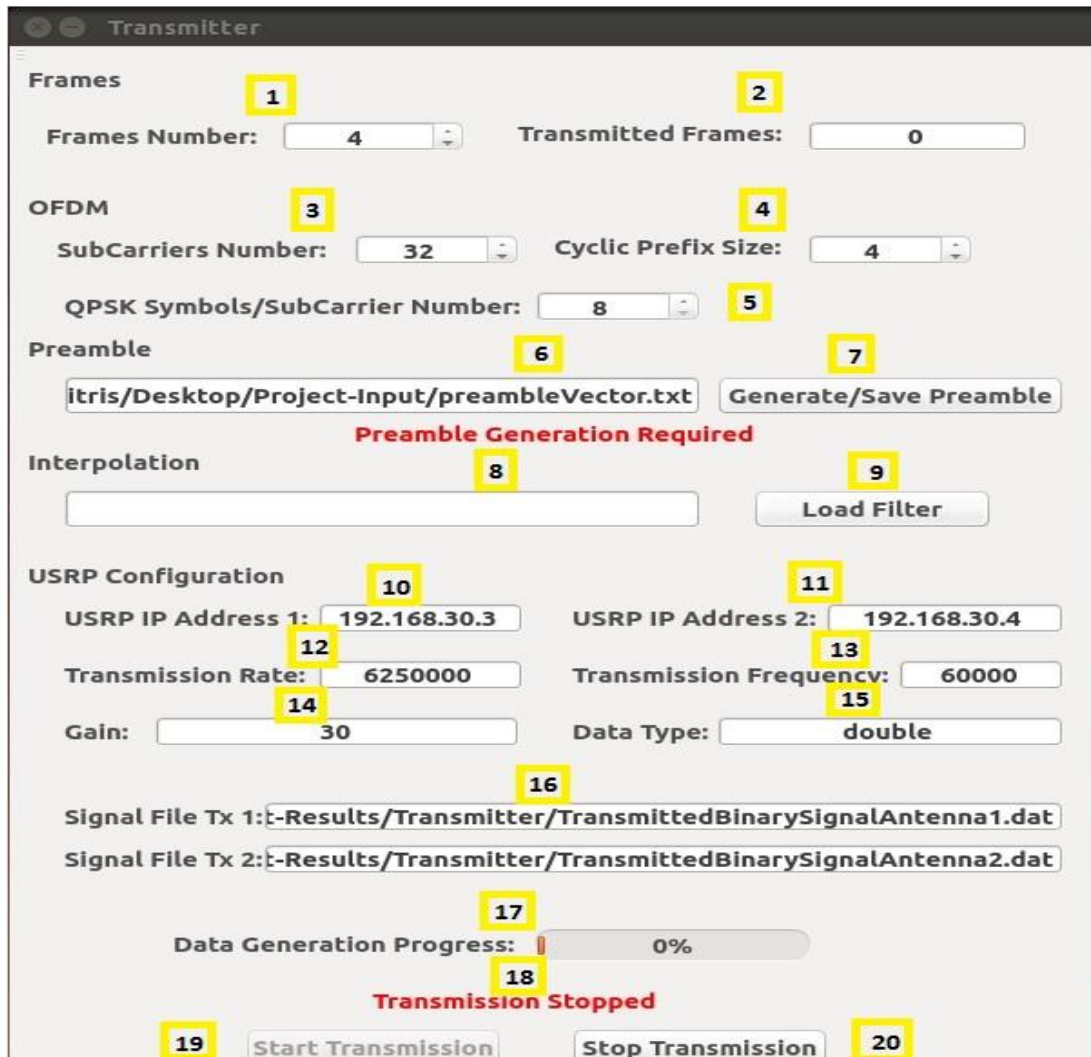
**Σχήμα 1.4: Τοπολογία συστήματος SDR-USRP 2**

Για την υλοποίηση συστήματος MIMO 2x2, απαιτούνται 2 συσκευές USRP-N210, τόσο στον πομπό όσο και στο δέκτη. Το σύστημα αποτελείται από δύο αυτόνομα υποσυστήματα SDR, που αντιστοιχούν στον πομπό και τον δέκτη. Σε κάθε συσκευή αντιστοιχεί μία κεραία εκπομπής (στον πομπό) ή λήψης (στον δέκτη). Ένα ακόμη χαρακτηριστικό των συσκευών της σειράς N210 είναι η χρήση του MIMO cable ώστε να πραγματοποιείται συγχρονισμένη MIMO εκπομπή και λήψη μεταξύ των δυο RF βαθμίδων. Με την διασύνδεση αυτή, οι δύο φυσικές συσκευές λειτουργούν εικονικά ως μία (1 virtual device), ο απαραίτητος συγχρονισμός της οποίας πραγματοποιείται αποκλειστικά από την primary-πρωτεύουσα φυσική συσκευή. Η επικοινωνία του υπολογιστή-host με τις συσκευές USRP, πραγματοποιείται μέσω του Gigabit Ethernet link που τον συνδέει με την primary συσκευή USRP. Θα πρέπει να σημειωθεί, πως η χρήση του MIMO cable, καθιστά μη απαραίτητη την απευθείας ζεύξη της δεύτερης φυσικής συσκευής USRP με τον υπολογιστή-host, διότι όλα τα απαραίτητα δεδομένα διέρχονται από αυτό.

# Κεφάλαιο 2 - Γραφικό Περιβάλλον Εφαρμογής (GUI)

## 2.1 Γραφικό Περιβάλλον Πομπού

Για την ευκολότερη εκτέλεση της εφαρμογής αναπτύχθηκε γραφικό περιβάλλον (GUI), τόσο για τον πομπό όσο και για τον δέκτη. Με τον τρόπο αυτό, καθίσταται γρηγορότερη η αλλαγή των παραμέτρων εκτέλεσης της εφαρμογής από τον χρήστη. Η ανάπτυξη του γραφικού περιβάλλοντος έγινε με χρήση της βιβλιοθήκης QT (έκδοση 5.3) της C++. Ακολουθεί η περιγραφή της λειτουργίας των επιμέρους τμημάτων (components) του γραφικού περιβάλλοντος του πομπού (Σχήμα 2.1):



Σχήμα 2.1 – Γραφικό περιβάλλον πομπού

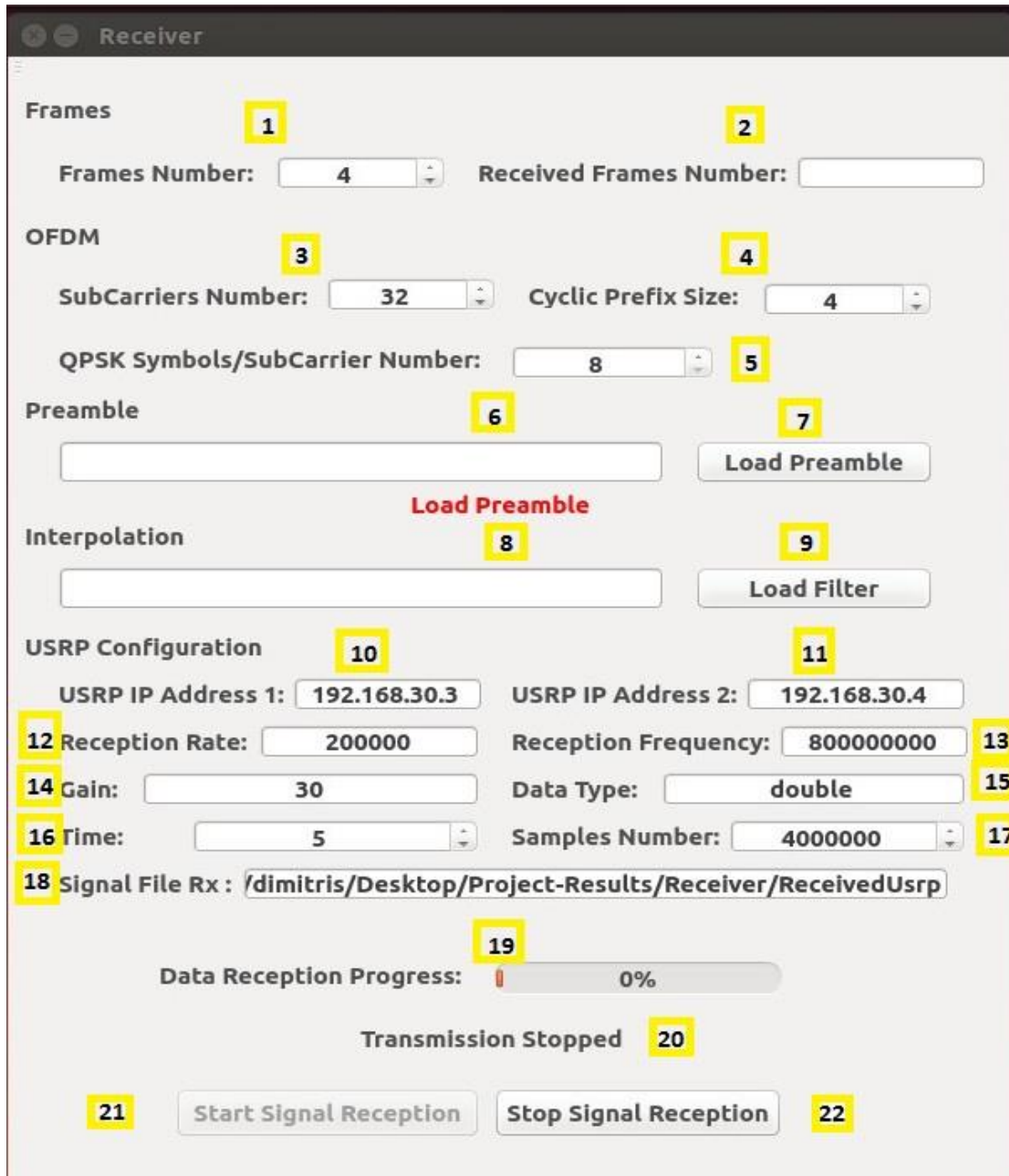
- 1) Αριθμός των OFDM frames που δημιουργούνται και μεταδίδονται στον δέκτη.
- 2) Αριθμός των OFDM frames που έχουν δημιουργηθεί πριν την έναρξη της μετάδοσης.
- 3) Αριθμός των υποκαναλιών (subcarriers) OFDM των OFDM frames.
- 4) Μέγεθος του Κυκλικού Προθέματος-Cyclic Prefix (σε δείγματα).
- 5) Αριθμός συμβόλων δεδομένων που περιέχονται σε κάθε OFDM subcarrier.
- 6) File path του αρχείου txt στο οποίο αποθηκεύεται το preamble που δημιουργείται πριν την αποστολή (το ίδιο preamble χρησιμοποιείται από τον δέκτη για τον απαραίτητο συγχρονισμό του με τον πομπό). Με αυτόν τον τρόπο το preamble θεωρείται γνωστό σε πομπό και δέκτη.
- 7) QButton για την δημιουργία και αποθήκευση του preamble.



- 8) Πεδίο για την απόδοση του file path του αρχείου txt στο οποίο βρίσκονται αποθηκευμένες οι τιμές του φίλτρου Root Raised Cosine (ρίζα ανορθωμένου συνημιτόνου), για την μορφοποίηση του σήματος πριν την αποστολή (Pulse Shaping).
- 9) QPushButton για την ανάγνωση των τιμών του Root Raised Cosine φίλτρου.
- 10) Διεύθυνση IP της primary συσκευής USRP (στην οποία συνδέεται απευθείας το Ethernet καλώδιο).
- 11) Διεύθυνση IP της secondary συσκευής USRP (η οποία συνδέεται με την primary μέσω του MIMO cable).
- 12) Ρυθμός μετάδοσης.
- 13) Φέρουσα συχνότητα μετάδοσης.
- 14) Κέρδος ενίσχυσης κεραιών των συσκευών USRP.
- 15) Αριθμητικός τύπος δειγμάτων που μεταδίδονται (π.χ. double, float κτλ).
- 16) File path των αρχείων txt στον οποίο αποθηκεύονται τα δείγματα που δημιουργούνται και μεταδίδονται. Το πρώτο πεδίο καθορίζει το file path των δειγμάτων που μεταδίδονται από την πρώτη κεραία ενώ το δεύτερο καθορίζει το file path των δειγμάτων που μεταδίδονται από την δεύτερη κεραία.
- 17) QProgressBar που δείχνει το ποσοστό των προς μετάδοση δειγμάτων που έχουν δημιουργηθεί πριν την έναρξη της μετάδοσης.
- 18) QLabel που δείχνει την τρέχουσα κατάσταση της μετάδοσης (αν βρίσκεται σε εξέλιξη ή έχει ολοκληρωθεί).
- 19) QPushButton για την έναρξη της δημιουργίας δεδομένων και την πραγματοποίηση της μετάδοσης. Σε περίπτωση που ισχύει κάποια από τις παρακάτω περιπτώσεις, τότε εμφανίζονται alert dialogs και εφαρμογή δεν εκτελείται:
  - Το μέγεθος του Κυκλικού Προθέματος είναι μεγαλύτερο ή ίσο από τον αριθμό των OFDM subcarriers του OFDM frame.
  - Δεν έχει δημιουργηθεί το αρχείο txt που περιέχει το preamble.
  - Δεν έχει επιλεχθεί το αρχείο txt που περιέχει τις τιμές του Root Raised Cosine φίλτρου.
  - Δεν έχει συνδεθεί κάποια συσκευή USRP με τον υπολογιστή-host στον οποίο εκτελείται η εφαρμογή.
- 20) QPushButton για την διακοπή της δημιουργίας δεδομένων και της μετάδοσης.

## 2.2 Γραφικό Περιβάλλον Δέκτη

Ακολουθεί η περιγραφή των επιμέρους τμημάτων (components) του γραφικού περιβάλλοντος του δέκτη (Σχήμα 2.2):



Σχήμα 2.2 - Γραφικό περιβάλλον δέκτη

- 1) Αριθμός των OFDM frames που αναμένονται για επεξεργασία στον δέκτη.
- 2) Αριθμός των OFDM frames που έχουν ληφθεί πριν την έναρξη της εφαρμογής.
- 3) Αριθμός των OFDM subcarriers των OFDM frames.
- 4) Μέγεθος του Κυκλικού Προθέματος-Cyclic Prefix (σε δείγματα).
- 5) Αριθμός συμβόλων δεδομένων που περιέχονται σε κάθε υποφέρον OFDM .

- 6) File path του αρχείου txt στο οποίο είναι αποθηκευμένο το preamble που έχει δημιουργηθεί από τον πομπό.
- 7) QPushButton για την ανάγνωση του preamble.
- 8) Πεδίο για την απόδοση του file path του αρχείου txt στο οποίο βρίσκονται αποθηκευμένες οι τιμές του Root Raised Cosine φίλτρου.
- 9) QPushButton για την ανάγνωση των τιμών του Root Raised Cosine φίλτρου.
- 10) Διεύθυνση IP της primary συσκευής USRP.
- 11) Διεύθυνση IP της secondary συσκευής USRP.
- 12) Ρυθμός λήψης.
- 13) Φέρουσα συχνότητα λήψης.
- 14) Κέρδος ενίσχυσης κεραιών των συσκευών USRP.
- 15) Αριθμητικός τύπος δειγμάτων που λαμβάνονται (π.χ. double, float κτλ).
- 16) Χρονικό διάστημα στην έναρξη της λήψης, κατά το οποίο οι συσκευές USRP αγνοούν το λαμβανόμενο σήμα. Κατά την εκκίνηση της λειτουργίας του δέκτη και μέχρις ότου τα ηλεκτρονικά κυκλώματα του USRP έρθουν σε κατάσταση κανονικής λειτουργίας, ο AGC (Automatic Gain Controller) και οι ταλαντωτές συμπεριφέρονται ανώμαλα. Για αυτό το λόγο πρέπει να ληφθεί ένα διάστημα φύλαξης στην αρχή της λειτουργίας για το οποίο τα λαμβανόμενα δείγματα θεωρούνται λανθασμένα και αγνοούνται. Το διάστημα αυτό μπορεί να είναι και μικρότερο του 1sec.
- 17) Συνολικός αριθμός δειγμάτων που λαμβάνονται από τον δέκτη.
- 18) File path των αρχείων txt στο οποίο αποθηκεύονται τα δείγματα που έχουν ληφθεί από τον δέκτη.
- 19) QProgressBar που δείχνει το ποσοστό των ληφθέντων δειγμάτων που έχει αναλυθεί από τον δέκτη.
- 20) QLabel που δείχνει την τρέχουσα κατάσταση της λήψης (αν βρίσκεται σε εξέλιξη ή έχει ολοκληρωθεί).
- 21) QPushButton για την έναρξη της λήψης. Σε περίπτωση που ισχύει κάποια από τις παρακάτω περιπτώσεις, τότε εμφανίζονται alert dialogs και εφαρμογή δεν εκτελείται:
  - Το μέγεθος του Κυκλικού Προθέματος είναι μεγαλύτερο ή ίσο από τον αριθμό των OFDM subcarriers του OFDM frame.
  - Δεν έχει γίνει ανάγνωση του αρχείου txt που περιέχει το preamble.
  - Δεν έχει επιλεγθεί το αρχείο txt που περιέχει τις τιμές του Root Raised Cosine φίλτρου.
  - Δεν έχει συνδεθεί κάποια συσκευή USRP με τον υπολογιστή-host στον οποίο εκτελείται η εφαρμογή.
- 22) QPushButton για την διακοπή της λήψης.

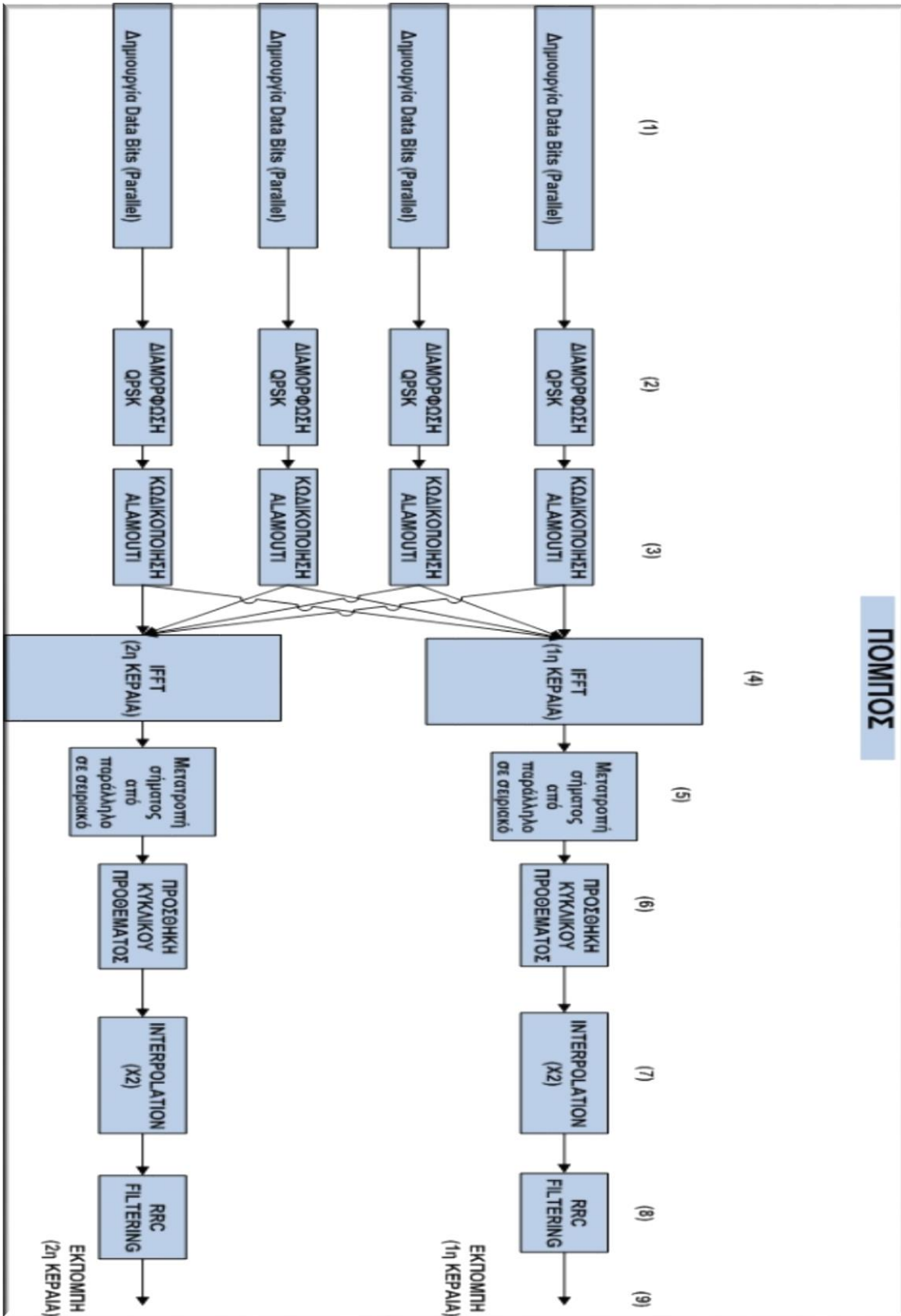
Σημειώνεται ότι για την ορθή λειτουργία του συστήματος εκπομπής και λήψης, οι παράμετροι στον πομπό και στον δέκτη πρέπει να είναι εναρμονισμένοι. Δηλαδή, τα χαρακτηριστικά της OFDM κυματομορφής, η φέρουσα συχνότητα, το εύρος ζώνης, το preamble και το φίλτρο μορφοποίησης, θα πρέπει να είναι όμοια τόσο στον πομπό όσο και στο δέκτη. Διαφορετικά η λήψη του σήματος δεν θα οδηγήσει σε επιτυχή αποδιαμόρφωση.

# Κεφάλαιο 3 - Διαγραμματική Απεικόνιση Λειτουργίας Πομπού-Δέκτη

## 3.1 Διάγραμμα Λειτουργίας Πομπού

Ακολουθεί η διαγραμματική απεικόνιση της λειτουργίας του πομπού και η ανάλυση των επιμέρους τμημάτων (modules-μέθοδοι λογισμικού), από τα οποία αποτελείται (Σχήμα 3.1):

## ΠΟΜΠΟΣ



Σχήμα 3.1 – Διάγραμμα λειτουργίας πομπού

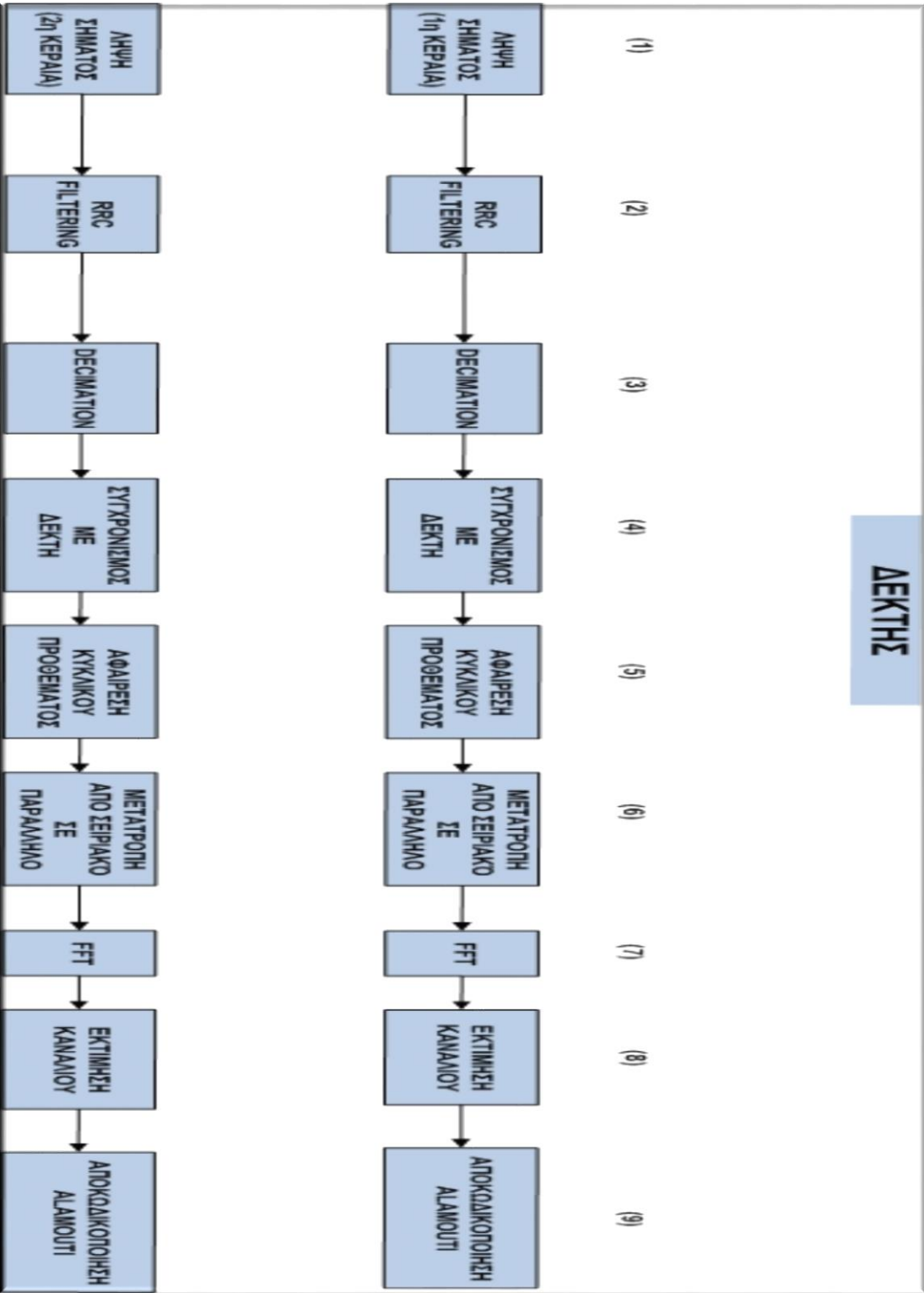
- 1) Δημιουργία των bits πληροφορίας.
- 2) Μετατροπή των bits πληροφορίας σε σύμβολα, χρησιμοποιώντας την μέθοδο ψηφιακής διαμόρφωσης QPSK.
- 3) Χωροχρονική κωδικοποίηση Alamouti των συμβόλων πληροφορίας.
- 4) Εκτέλεση Αντίστροφου Ταχέως Μετασχηματισμού Fourier (IFFT).
- 5) Μετατροπή σήματος από παράλληλο σε σειριακό.
- 6) Προσθήκη Κυκλικού Προθέματος.
- 7) Interpolation (Αύξηση του ρυθμού επεξεργασίας κατά 2).
- 8) Root Raised Cosine Filtering.
- 9) Μετάδοση σήματος (συσσκευές USRP).

### 3.2 Διάγραμμα λειτουργίας Δέκτη

Ακολουθεί η διαγραμματική απεικόνιση της λειτουργίας του δέκτη και η ανάλυση των επιμέρους τμημάτων (modules–μέθοδοι λογισμικού), από τα οποία αποτελείται (Σχήμα 3.2):

- 1) Λήψη σήματος (συσσκευές USRP).
- 2) Root Raised Cosine Filtering.
- 3) Decimation (Μείωση του ρυθμού επεξεργασίας κατά 2)..
- 4) Συγχρονισμός πομπού-δέκτη.
- 5) Αφαίρεση Κυκλικού Προθέματος.
- 6) Μετατροπή σήματος από σειριακό σε παράλληλο.
- 7) Εκτέλεση Ταχέως μετασχηματισμού Fourier (FFT).
- 8) Εκτίμηση καναλιού (channel estimation)
- 9) Αποκωδικοποίηση Alamouti.

## ΔΕΚΤΗΣ



Σχήμα 3.2 – Διάγραμμα λειτουργίας δέκτη



## Κεφάλαιο 4 - Ψηφιακή Διαμόρφωση Σήματος

Ως διαμόρφωση σήματος, ορίζεται η διαδικασία μεταβολής ενός περιοδικού σήματος, συνήθως υψηλής συχνότητας, με στόχο την κωδικοποίηση σε αυτό, ενός σήματος χαμηλής συχνότητας, το οποίο μεταφέρει την πληροφορία. Το σήμα αυτό ονομάζεται φέρον (carrier signal) και συνήθως έχει απλή συχνότητα, δηλαδή ημιτονοειδή κυματομορφή. Η διαμόρφωση είναι βασικής σημασίας για την δίοδο του σήματος μέσω του καναλιού, (π.χ καλώδιο στις ενσύρματες επικοινωνίες, ελεύθερος χώρος στις ασύρματες), το εύρος ζώνης του οποίου δεν επικαλύπτεται με το εύρος ζώνης του σήματος.

### 4.1 Είδη Τεχνικών Ζωνοπερατής Ψηφιακής Διαμόρφωσης

Η ζωνοπερατή διαμόρφωση είναι η διαδικασία κατά την οποία το σήμα που περιέχει την πληροφορία (φέρον σήμα), μετατρέπεται σε μία ημιτονοειδή κυματομορφή η οποία έχει τρία βασικά χαρακτηριστικά μεγέθη:

- 1) Πλάτος
- 2) Συχνότητα
- 3) Φάση

Κατά την διαδικασία της διαμόρφωσης ένα ή περισσότερα από τα παραπάνω χαρακτηριστικά μεταβάλλονται.

Οι κυριότερες μέθοδοι ψηφιακής διαμόρφωσης είναι οι εξής:

- 1) Διαμόρφωση μετατόπισης συχνότητας (FSK)
- 2) Διαμόρφωση μετατόπισης πλάτους (ASK)
- 3) Διαμόρφωση μετατόπισης φάσης (PSK)
- 4) Ορθογωνική διαμόρφωση πλάτους (QAM), η οποία αποτελεί συνδυασμό των μεθόδων PSK και ASK.

### 4.2 Διαμόρφωση Μετατόπισης Φάσης (PSK-Phase Shift Keying)

Στην διαμόρφωση μετατόπισης φάσης, για την διαμόρφωση της πληροφορίας χρησιμοποιείται ένας συγκεκριμένος αριθμός διαφορετικών φάσεων, κάθε μία εκ των οποίων

αντιστοιχεί σε μία συγκεκριμένη ακολουθία από bits. Κάθε ακολουθία από bits αποτελεί ένα σύμβολο. Υπάρχουν δύο βασικές κατηγορίες τεχνικών διαμόρφωσης φάσης:

- 1) Αυτές κατά τις οποίες για την διαμόρφωση της πληροφορίας χρησιμοποιείται η φάση αυτή καθ' αυτή.
- 2) Αυτές κατά τις οποίες για τη διαμόρφωση της πληροφορίας χρησιμοποιείται η αλλαγή φάσης μεταξύ των διαδοχικών συμβόλων (διαφορικές τεχνικές).

### 4.3 Ορθογώνια Διαμόρφωση Μετατόπισης Φάσης (Quadrature Phase Shift Keying - QPSK)

Στην συγκεκριμένη εργασία, ως μέθοδος διαμόρφωσης χρησιμοποιείται η QPSK. Η QPSK αποτελεί μία ειδικότερη κατηγορία των μεθόδων M-PSK, συνεπώς και για αυτήν ισχύει:

$$s_i(t) = A \cos(2\pi f_c t + \theta_i) \quad (4)$$

Για τον σχηματισμό των συμβόλων χρησιμοποιούνται  $k=2$  bits, ενώ ο αριθμός των διαφορετικών συμβόλων είναι  $N_s=2^k=2^2=4$  σύμβολα. Η εξίσωση 4 μπορεί να γραφεί και ως εξής:

$$s_i(t) = A \cos \theta_i \cos 2\pi f_c t - A \sin \theta_i \sin 2\pi f_c t \quad (5)$$

$$\text{όπου } \varphi_1(t) = \sqrt{\frac{2}{T}} \cos 2\pi f_c t \quad (6) \text{ και } \varphi_2(t) = \sqrt{\frac{2}{T}} (\sin 2\pi f_c t) \quad \mu\epsilon \quad 0 \leq t \leq T \quad (7)$$

Από τις εξισώσεις (4), (5) και (6) προκύπτει:

$$s_{i1} = \sqrt{E \cos(\theta_i)} \quad (8), \quad s_{i2} = \sqrt{E \sin(\theta_i)} \quad (9) \text{ και } \theta_i = \tan^{-1}(s_{i2}/s_{i1}) \quad (10)$$

όπου  $E = A^2 \frac{T}{2}$  είναι η ενέργεια του συμβόλου. Συνεπώς, τα τέσσερα διαφορετικά σύμβολα της μεθόδου QPSK, αντιστοιχούν σε ισάριθμα σημεία ή διανύσματα στο επίπεδο συντεταγμένων:

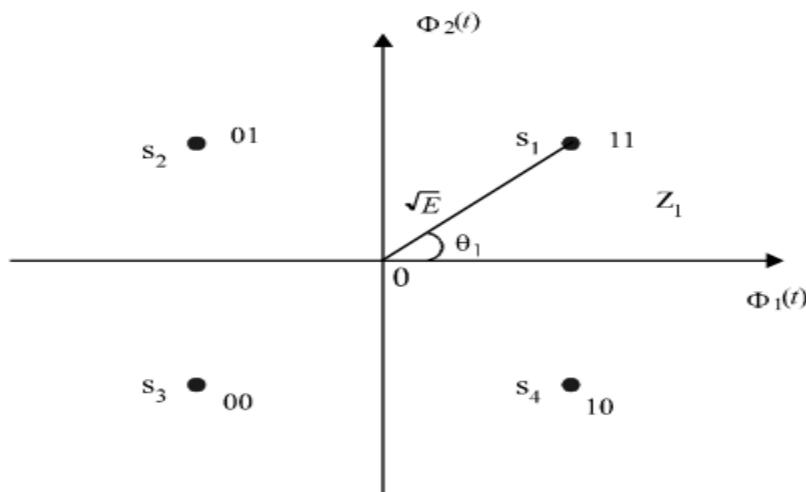
$$s_i = \begin{bmatrix} s_{i1} \\ s_{i2} \end{bmatrix}, \quad i=1,2,3,4 \quad (11)$$

Η γωνία των διανυσμάτων με τον οριζόντιο άξονα συντεταγμένων είναι ίδια με την αρχική φάση  $\theta_i$  ενώ το μέτρο τους είναι ίσο με  $\sqrt{E}$ .

Σύμβολο	Φάση $\theta_i$	$s_{i1}=\sqrt{E}\cos\theta_i$	$s_{i2}=\sqrt{E}\sin\theta_i$
11	$\pi/4$	$\sqrt{E/2}$	$\sqrt{E/2}$
01	$3\pi/4$	$-\sqrt{E/2}$	$\sqrt{E/2}$
00	$-3\pi/4$	$-\sqrt{E/2}$	$-\sqrt{E/2}$
10	$-\pi/4$	$\sqrt{E/2}$	$-\sqrt{E/2}$

**Πίνακας 4.1 – Συντεταγμένες σήματος QPSK**

Σύμφωνα με τα παραπάνω, ο αστερισμός (constellation diagram) της μεθόδου QPSK είναι αυτός που φαίνεται στο Σχήμα 4.1. Στην συγκεκριμένη περίπτωση χρησιμοποιείται κωδικοποίηση Gray, κατά την οποία κάθε φάση, συνεπώς και κάθε σύμβολο, διαφέρει από το προηγούμενο και το επόμενο μόνο κατά ένα bit, μειώνοντας την πιθανότητα σφάλματος. Τα πιθανότερα σφάλματα που προκαλούνται από τον θόρυβο, οδηγούν στην εσφαλμένη επιλογή μιας φάσης γειτονικής ως προς την μεταδιδόμενη φάση. Με την κωδικοποίηση Gray, μόνο ένα bit μπορεί να είναι εσφαλμένο στην ακολουθία των  $k$  bits. (Η παρακάτω αντιστοιχία συμβόλων-bits χρησιμοποιήθηκε και σε αυτήν την εργασία).

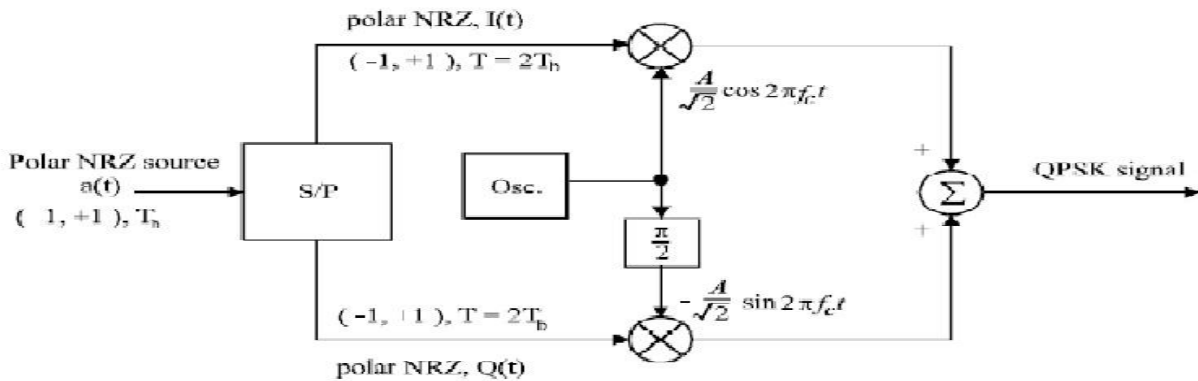


**Σχήμα 4.1 - QPSK αστερισμός (constellation)**

Όπως φαίνεται στον Πίνακα 4.1, η τιμή 1 αντιστοιχεί σε τετμημένη  $\sqrt{E/2}$ , ενώ η τιμή 0 σε τεταγμένη  $-\sqrt{E/2}$ . Επίσης κάνουμε την παραδοχή πως τα περιττά bits αντιστοιχούν στην τιμή  $s_{i1}$ , ενώ τα άρτια στην τιμή  $s_{i2}$ . Συνεπώς από την εξίσωση 5 προκύπτει:

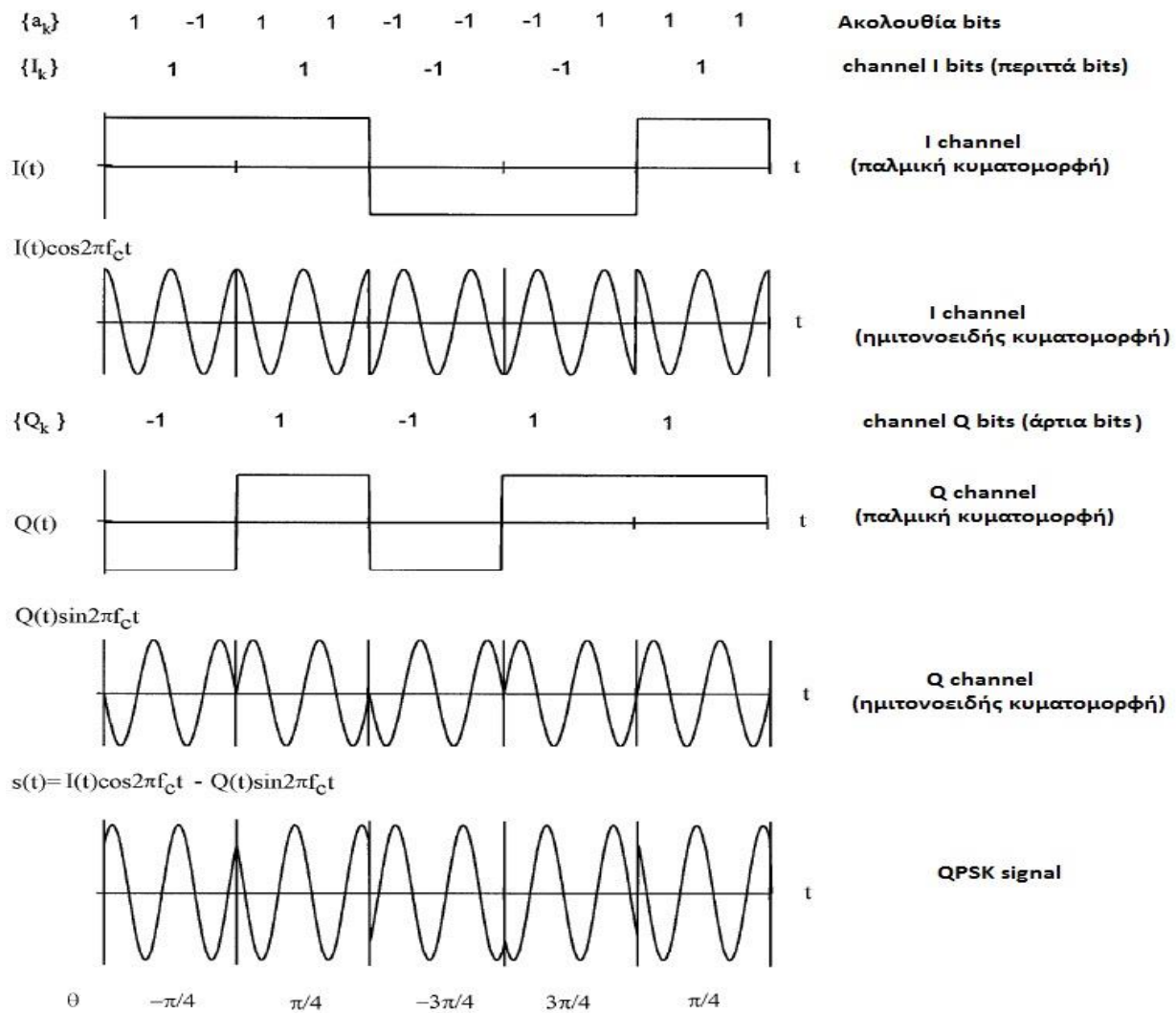
$$s(t) = \frac{A}{\sqrt{2}}I(t) \cos 2\pi f_c t - \frac{A}{\sqrt{2}}Q(t) \sin 2\pi f_c t \quad \mu\epsilon -\infty \leq t \leq \infty \quad (12)$$

όπου  $I(t)$  και  $Q(t)$  είναι οι παλμικές κυματομορφές των περιττών και άρτιων bits αντίστοιχα. Η εξίσωση (12), αποτελεί τη βάση για τη λειτουργία του διαμορφωτή QPSK του Σχήματος 4.2:



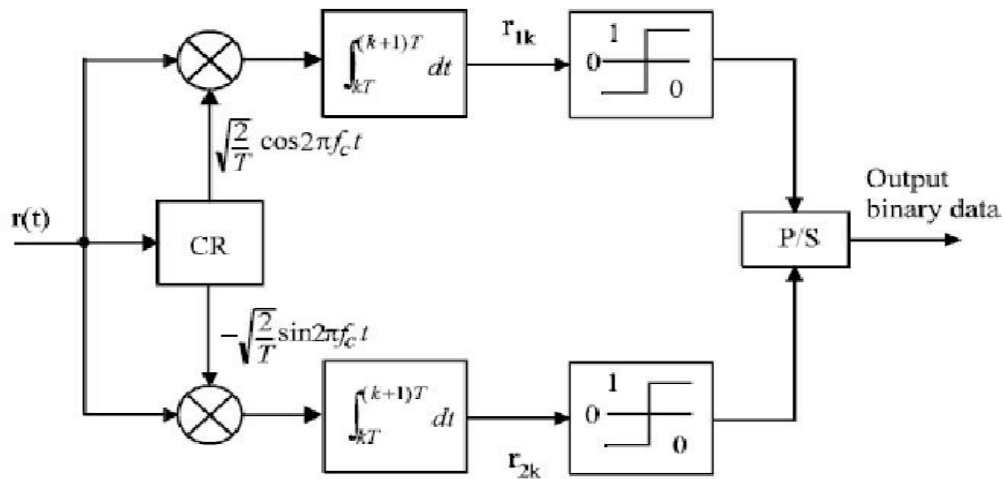
**Σχήμα 4.2: Διαμορφωτής QPSK**

Το κανάλι του ημιτόνου ονομάζεται κανάλι I (Inphase - συμφασικό), ενώ το κανάλι του συνημιτόνου ονομάζεται κανάλι Q (Quadrature - ορθογώνιο). Τα δεδομένα διαχωρίζονται από τον μετατροπέα σειριακού σήματος σε παράλληλο (S/P), έτσι ώστε τα περιττά bits να αποτελούν το κανάλι I, ενώ τα άρτια bits να αποτελούν το κανάλι Q. Τα bits με τιμή 1 μετατρέπονται σε θετικούς παλμούς, ενώ τα bits με τιμή 0 σε αρνητικούς. Οι παλμοί έχουν το ίδιο πλάτος και ίδια χρονική διάρκεια  $T$ . Οι παλμοί του καναλιού I πολλαπλασιάζονται με τον παράγοντα  $\cos(2\pi f_c t)$ , ενώ του καναλιού Q με τον παράγοντα  $\sin(2\pi f_c t)$ . Τελικά, οι κυματομορφές που προκύπτουν αθροίζονται από τον αθροιστή, σχηματίζοντας το τελικό σήμα QPSK (Σχήμα 4.3):



**Σχήμα 4.3: Διαμόρφωση QPSK**

Τα κανάλια I και Q μπορούν να αποδιαμορφωθούν ως δύο ανεξάρτητα σήματα BPSK (Σχήμα 4.4). Ένας μετατροπέας παράλληλου σήματος σε σειριακό, συνδυάζει τα δύο σήματα σε ένα. Αυτό είναι δυνατό λόγω της ένα προς ένα αντιστοιχίας μεταξύ των bits δεδομένων και του σήματος των καναλιών I και Q, αλλά και της ορθογωνιότητας που υπάρχει μεταξύ των τελευταίων.



**Σχήμα 4.4: Αποδιαμορφωτής QPSK**

Η μέση πιθανότητα του ρυθμού σφαλμάτων (bit error rate) δίνεται από τον τύπο (13):

$$P_b = Q\left(\sqrt{\frac{E}{N_0}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad (13)$$

Κάθε σύμβολο QPSK αποτελείται από 2 bits, ένα από το κανάλι I και ένα από το κανάλι Q. Θεωρείται, λοιπόν, εσφαλμένο, όταν ένα από τα δύο bits παρουσιάζει σφάλμα, οπότε η πιθανότητα του σφάλματος συμβόλου (symbol error) δίνεται από τον τύπο (14):

$$\begin{aligned} P_s &= 1 - P_r(\text{both bits are correct}) \\ &= 1 - (1 - P_b)^2 = 2P_b - P_b^2 \\ &= 2Q\left(\sqrt{\frac{E}{N_0}}\right) - \left[Q\left(\sqrt{\frac{E}{N_0}}\right)\right]^2 \end{aligned} \quad (14)$$

Για υψηλές τιμές SNR ισχύει:

$$\left[Q\left(\sqrt{\frac{E}{N_0}}\right)\right]^2 \approx 0 \quad (15)$$

Για υψηλές τιμές SNR, χρησιμοποιώντας κωδικοποίηση Gray, ένα λανθασμένο σύμβολο οδηγεί στην επιλογή ενός γειτονικού συμβόλου, το οποίο διαφέρει από το σωστό κατά ένα bit. Συνεπώς για την πιθανότητα του bit error ισχύει:

$$P_b = \frac{1}{2}P_s = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad (16)$$





# Κεφάλαιο 5 - Συστήματα Πολλαπλών Κεραίων Εκπομπής -Λήψης

## 5.1 Χωροχρονική Κωδικοποίηση Alamouti

Πολλές φορές, είναι πρακτικότερη και οικονομικότερη η ύπαρξη μίας κεραίας λήψης στον δέκτη, λόγω περιορισμών στο κόστος και τον όγκο του εξοπλισμού. Στην περίπτωση αυτή μπορεί να χρησιμοποιείται διαφορική εκπομπή, με πολλαπλές εισόδους (τις κεραίες του πομπού) και μία έξοδο (την κεραία του δέκτη). Συστήματα αυτού του είδους ονομάζονται MISO (Multiple Input- Single Output). Αντίθετα, τα συστήματα διαφορικής λήψης (μία είσοδος – πολλές έξοδοι), αναφέρονται ως SIMO. Η εκπομπή του σήματος από τις πολλαπλές κεραίες του πομπού ορίζεται από ένα σχήμα χωρο-χρονικής κωδικοποίησης (Space Time Coding).

Ο απλούστερος χωροχρονικός κώδικας είναι ο κώδικας Alamouti. Στην πιο απλή μορφή του θεωρούμε ένα σύστημα  $2 \times 1$ , δηλαδή ένα σύστημα δύο κεραίων εκπομπής και μίας κεραίας λήψης. Για την ολοκλήρωση της κωδικοποίησης αυτής απαιτούνται δύο χρονικές περίοδοι. Έστω ότι στην πρώτη περίοδο η πρώτη κεραία εκπέμπει το σύμβολο  $X_1$  και η δεύτερη το  $X_2$ . Στην επόμενη χρονική περίοδο η πρώτη κεραία εκπέμπει το σύμβολο  $-X_2^*$ , ενώ η δεύτερη κεραία εκπέμπει το σύμβολο  $X_1^*$  (όπου  $X^*$  ο συζυγής μιγαδικός του  $X$ ). Στην δεύτερη χρονική περίοδο δεν εκπέμπονται νέα σύμβολα, συνεπώς ο ρυθμός μετάδοσης παραμένει αμετάβλητος.

Το σήμα που λαμβάνει η κεραία του δέκτη κατά την πρώτη χρονική περίοδο, δίνεται από τον τύπο:

$$Y_1 = h_1 X_1 + h_2 X_2 + n_1 \quad (1)$$

όπου  $h_i$  το κέρδος του διαύλου μεταξύ της  $i$ -οστής κεραίας εκπομπής και της κεραίας λήψης, ενώ  $n_i$  είναι ο προστιθέμενος θόρυβος. Αν θεωρήσουμε ότι ο δίαυλος παραμένει αμετάβλητος κατά τις δύο χρονικές περιόδους, τότε το λαμβανόμενο σήμα κατά την επόμενη χρονική περίοδο είναι:

$$Y_2 = h_1 (-X_2^*) + h_2 X_1^* + n_1 \quad (2)$$

Ο δέκτης συνδυάζει τα σήματα που λαμβάνει στις δύο χρονικές περιόδους και υπολογίζει τα μεγέθη:

$$\widehat{Y}_1 = h_1^* Y_1 + h_2 Y_2^* \quad (3)$$

$$\widehat{Y}_2 = h_2^* Y_1 - h_1 Y_2^* \quad (4)$$

Αντικαθιστώντας τις εξισώσεις (1) και (2) στις (3) και (4), έχουμε:

$$\widehat{Y}_1 = (|h_1|^2 + |h_2|^2)X_1 + h_1^*n_1 + h_2n_2 \quad (5)$$

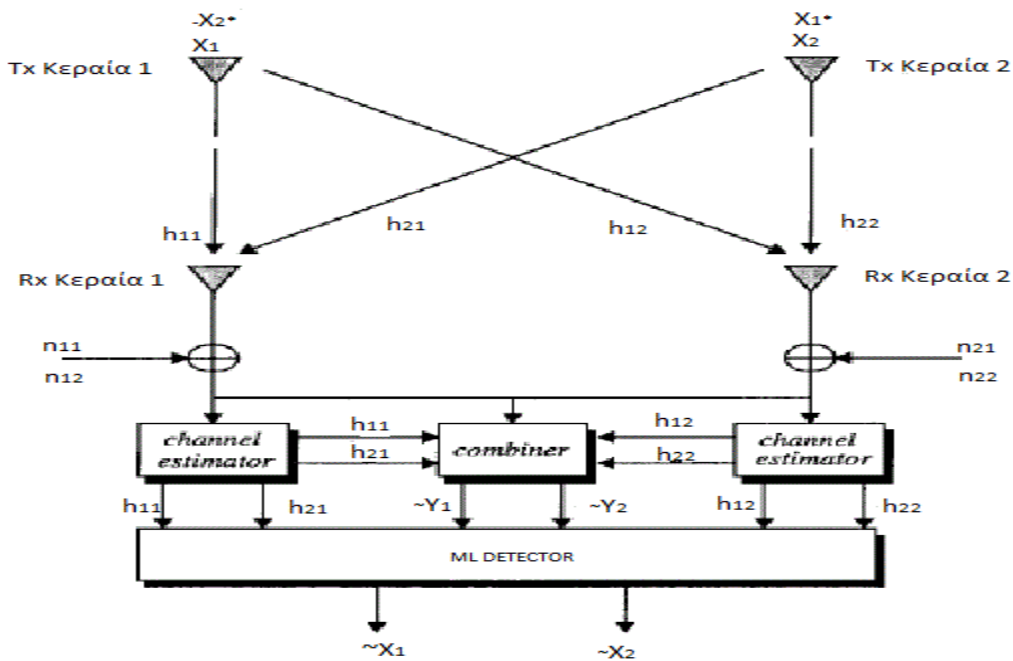
$$\widehat{Y}_2 = (|h_1|^2 + |h_2|^2)X_2 - h_1^*n_2 + h_2^*n_1 \quad (6)$$

Η επιλογή των συμβόλων από τον δέκτη γίνεται βάσει του παρακάτω κριτηρίου απόφασης:

$$\widehat{X}_1 = \arg_{x \in S} \min \{ |\widehat{Y}_1 - (|h_1|^2 + |h_2|^2)x|^2 \} \quad (7)$$

$$\widehat{X}_2 = \arg_{x \in S} \min \{ |\widehat{Y}_2 - (|h_1|^2 + |h_2|^2)x|^2 \} \quad (8)$$

## 5.2 Συστήματα Πολλαπλών Κεραίων Εκπομπής - Λήψης (MIMO)



Σχήμα 5.1 – Σύστημα Alamouti STC 2x2

Έστω ένα σύστημα με  $N_t$  κεραίες εκπομπής και  $N_r$  κεραίες λήψης, δηλαδή  $N_t$  εισόδους και  $N_r$  εξόδους. Ένα σύστημα τέτοιου είδους ονομάζεται σύστημα πολλαπλών εισόδων – πολλαπλών εξόδων (Multiple Input Multiple Output – MIMO System). Έστω ότι τα σήματα που εκπέμπονται από τις κεραίες εκπομπής παρίστανται με το διάνυσμα:

$$\mathbf{x} = [x_1(\mathbf{n}), x_2(\mathbf{n}), \dots, x_{N_t}(\mathbf{n})]^T \quad (9)$$

Υποθέτουμε πως κάθε ένα από τα σήματα  $\{x_i\}_{i=1}^{N_t}$  έχει μηδενική μέση τιμή και όλα έχουν κοινή συνδιακύμανση  $\sigma_x^2$ . Η συνολική ισχύς που εκπέμπει ο πομπός είναι ίση με:

$$\mathbf{P}_t = N_t \sigma_x^2 \quad (10)$$

Από τον εξίσωση (10) και την ανάγκη η ισχύς αυτή να είναι σταθερή, εξάγεται το συμπέρασμα πως η συνδιακύμανση  $\sigma_x^2$  είναι αντιστρόφως ανάλογη του πλήθους των κεραιών εκπομπής.

Θεωρούμε πως το εύρος συνοχής του διαύλου είναι πολύ μεγαλύτερο από το εύρος ζώνης των εκπεμπόμενων σημάτων, συνεπώς δεν υπάρχουν διαλείψεις επιλεκτικές ως προς την συχνότητα, οπότε ο δίαυλος μεταξύ της  $k$ -οστής κεραιάς εκπομπής ( $1 \leq k \leq N_t$ ) περιγράφεται από τον μιγαδικό συντελεστή  $h_{ik}$ . Συνολικά, οι  $N_r \times N_t$  ζεύξεις περιγράφονται από τον πίνακα  $\mathbf{H}$ :

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1N_t} \\ h_{21} & h_{22} & \dots & h_{2N_t} \\ \vdots & \vdots & \vdots & \vdots \\ h_{N_r 1} & h_{N_r 2} & \dots & h_{N_r N_t} \end{bmatrix}$$

Κάθε γραμμή του πίνακα  $\mathbf{H}$  αντιστοιχεί σε μια κεραιά λήψης, ενώ κάθε στήλη σε μια κεραιά εκπομπής. Τα σήματα που λαμβάνουν οι κεραιές λήψης παρίστανται με το διάνυσμα:

$$\mathbf{y} = [y_1, y_2, \dots, y_{N_r}]^T \quad (11)$$

Το  $i$ -οστό στοιχείο του διανύσματος  $\mathbf{y}$  εκφράζει το συνολικό σήμα που λαμβάνει η  $i$ -οστή κεραιά λήψης από όλες τις κεραιές εκπομπής και είναι:

$$y_i = \sum_{k=1}^{N_t} h_{ik} x_k + n_i \quad (12)$$

όπου  $x_k$  το σήμα που εκπέμπεται από την  $k$ -οστή κεραιά εκπομπής και  $n_i$  ο λευκός προσθετικός θόρυβος στην  $i$ -οστή κεραιά λήψης. Για τον θόρυβο θεωρούμε πως έχει μηδενική μέση τιμή, φασματική πυκνότητα ισχύος  $\frac{N_0}{2}$  και πίνακα συνδιακύμανσης  $\sigma^2 \mathbf{I}_{N_r}$ , όπου  $\mathbf{I}_{N_r}$  ο  $N_r$ -μοναδιαίος πίνακας. Αν οι τιμές του θορύβου συγκεντρωθούν στο διάνυσμα:

$$\mathbf{n} = [n_1, n_2, \dots, n_{N_r}]^T \quad (13)$$

τότε το σύστημα μπορεί να περιγραφεί με την διανυσματική εξίσωση:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \quad (14)$$

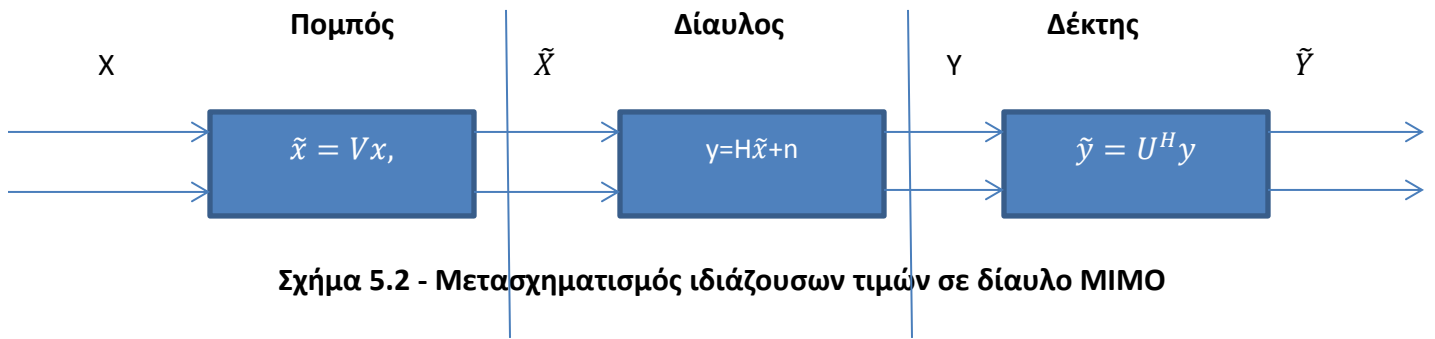
Η εξίσωση (14) περιγράφει το βασικό μιγαδικό μοντέλο διαύλου για συστήματα MIMO, με επίπεδες διαλείψεις. Σημαντική παράμετρος για την επίδοση των συστημάτων MIMO είναι κατά πόσο ο πομπός και ο δέκτης γνωρίζουν τον πίνακα  $H$  του διαύλου. Όταν λειτουργεί ως πομπός, αξιοποιεί την πληροφορία που ήδη έχει για το  $H$  (για συστήματα Time Division Duplex-TDD) ή μέσω ανάδρασης λαμβάνει την πληροφορία αυτή από την άλλη πλευρά της ζεύξης (για συστήματα Frequency Division Duplex-FDD). Αν ο πίνακας  $H$  δεν είναι γνωστός στον πομπό και στον δέκτη, τότε γίνεται κάποια υπόθεση για την στατιστική συμπεριφορά του.

Με χρήση διαφορισμού σε πομπό και δέκτη, μπορούμε να επιτύχουμε βελτίωση της απόδοσης του συστήματος κατά έναν παράγοντα που ονομάζεται κέρδος διαφορισμού (diversity gain). Κάθε δίαυλος MIMO μπορεί να αναλυθεί σε  $R$  παράλληλους και ανεξάρτητους διαύλους. Με πολυπλεξία των δεδομένων στους διαύλους αυτούς, μπορεί να επιτευχθεί αύξηση της χωρητικότητας κατά  $R$ .

Έστω ότι για ένα σύστημα MIMO ( $N_t, N_r$ ), ο πίνακας του διαύλου  $H$  είναι γνωστός, τόσο στον πομπό όσο και στον δέκτη. Ο πίνακας  $H$  μπορεί να αναλυθεί με χρήση ιδιάζουσων τιμών (Singular Value Decomposition-SVD) ως εξής:

$$\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H \quad (15)$$

Οι πίνακες  $U$ , διαστάσεων  $N_r \times N_r$  και  $V$ , διαστάσεων  $N_t \times N_t$  είναι ορθομοναδιαίοι, δηλαδή ισχύει  $U^H U = I_{N_r}$  και  $V^H V = I_{N_t}$ . Ο πίνακας  $\Sigma$  είναι διαγώνιος πίνακας που περιέχει τις ιδιάζουσες τιμές  $\{\sigma_i\}$  του πίνακα  $H$ . Η ιδιάζουσα τιμή  $\sigma_i$  είναι ίση με την τετραγωνική τιμή της  $i$ -οστής μεγαλύτερης ιδιάζουσας τιμής του πίνακα  $HH^H$ , δηλαδή ισχύει  $\sigma_i = \sqrt{\lambda_i}$ . Οι μη μηδενικές ιδιάζουσες τιμές είναι  $R_H$ , όπου  $R_H$  είναι η τάξη του πίνακα  $H$ . Η τάξη ενός πίνακα δεν μπορεί να υπερβαίνει το πλήθος των γραμμών και των στηλών του, άρα  $R_H \leq \min(N_t, N_r)$ . Η ισότητα ισχύει όταν ο  $H$  είναι πλήρους τάξης, κάτι που ισχύει σε διαύλους όπου η διάδοση επηρεάζεται από πολλούς σκεδαστές. Αντίθετα, αν οι σκεδαστές είναι λίγοι ή τα στοιχεία του  $H$  έχουν υψηλή συσχέτιση μεταξύ τους, τότε η τάξη του  $H$  μπορεί να πάρει τιμή μέχρι και 1. Η ανάλυση σε παράλληλους ανεξάρτητους διαύλους γίνεται μετασχηματίζοντας τα εκπεμπόμενα σήματα πριν την εφαρμογή τους στις κεραίες εκπομπής και τα λαμβανόμενα σήματα μετά τις κεραίες λήψης. Αν το διάνυσμα των εκπεμπόμενων σημάτων είναι το  $x$ , τότε εφαρμόζεται ο μετασχηματισμός  $\tilde{x} = Vx$ , ενώ στα λαμβανόμενα σήματα  $y$ , εφαρμόζεται ο μετασχηματισμός  $\tilde{y} = U^H y$ . Στο Σχήμα 5.2 φαίνεται η διαδικασία του μετασχηματισμού ιδιάζουσων τιμών σε δίαυλο MIMO.



### 5.3 Κωδικοποίηση OFDM - Alamouti 2x2

Η κωδικοποίηση Alamouti αποτελεί την πιο απλή χωροχρονική τεχνική διαφορισμού. Το σύστημα που εξετάζεται αποτελείται από δύο κεραίες εκπομπής και δύο κεραίες λήψης. Στην περίπτωση αυτή, η κωδικοποίηση Alamouti είναι η μοναδική που εξασφαλίζει μεγιστοποίηση του κέρδους διαφορισμού (diversity gain), ενώ το κανονικοποιημένο data rate διατηρείται ίσο με 1.

Πιο ειδικά, χρησιμοποιούνται δύο κεραίες εκπομπής για την αποστολή δύο OFDM συμβόλων και των συζυγών τους, σε δύο διαδοχικές χρονικές στιγμές. Με τον τρόπο αυτό επιτυγχάνεται κέρδος διαφορισμού, χωρίς μείωση του data rate. Στον δέκτη, πραγματοποιείται διανυσματικός συνδυασμός των συμβόλων που λαμβάνουν οι δύο κεραίες του δέκτη. Αφού η εκπομπή γίνεται σε δύο χρονικές περιόδους, η αποκωδικοποίηση στον δέκτη γίνεται και αυτή σε ισάριθμες χρονικές στιγμές.

Έστω ότι το λαμβανόμενο σήμα στον δέκτη ισούται με το διάνυσμα  $Y$ . Τότε για την πρώτη χρονική στιγμή ισχύει:

$$Y = \begin{bmatrix} y_1^1 \\ y_2^1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} n_1^1 \\ n_2^1 \end{bmatrix} \quad (9)$$

Για την δεύτερη χρονική περίοδο ισχύει:

$$Y = \begin{bmatrix} y_1^2 \\ y_2^2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} -x_2^* \\ x_1^* \end{bmatrix} + \begin{bmatrix} n_1^2 \\ n_2^2 \end{bmatrix} \quad (10)$$

όπου  $\begin{bmatrix} y_1^1 \\ y_2^1 \end{bmatrix}$  είναι το λαμβανόμενο OFDM σύμβολο κατά την πρώτη χρονική περίοδο από τις κεραίες 1 και 2 αντίστοιχα. Συνδυάζοντας τις εξισώσεις (9) και (10) προκύπτει η εξίσωση (11):

$$Y = \begin{bmatrix} y_1^1 \\ y_2^1 \\ y_1^{2*} \\ y_2^{2*} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \\ h_{12}^* & -h_{11}^* \\ h_{22}^* & -h_{21}^* \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} n_1^1 \\ n_2^1 \\ n_1^{2*} \\ n_2^{2*} \end{bmatrix} \quad (11)$$

Στόχος μας είναι να απομονώσουμε τα μεταδιδόμενα σύμβολα  $x_1$  και  $x_2$ , κάτι που γίνεται ευκολότερο χρησιμοποιώντας την εκτίμηση του καναλιού. Για να γίνει αυτό θα πρέπει να μειώσουμε τον αριθμό των αγνώστων, χρησιμοποιώντας την εκτίμηση καναλιού, προκειμένου να υπολογίσουμε τους αντίστοιχους συντελεστές.

Στην συγκεκριμένη περίπτωση η εκτίμηση καναλιού γίνεται για κάθε υποκανάλι OFDM, όλων

των OFDM frames. Με δεδομένο τον πίνακα  $H = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \\ h_{12}^* & -h_{11}^* \\ h_{22}^* & -h_{21}^* \end{bmatrix}$ , μπορούμε να απομονώσουμε

τα σύμβολα  $x_1$  και  $x_2$  πολλαπλασιάζοντας τον πίνακα  $Y$  με τον αντίστροφο πίνακα του  $H$ , δηλαδή τον  $H^{-1}$ . Επειδή όμως ο πίνακας  $H$  δεν είναι τετραγωνικός, χρειαζόμαστε τον Moore-Penrose ψευδοαντίστροφο πίνακα του  $H$ , τον  $H^+$ . Ισχύει:

$$H^+ = (H^H H)^{-1} H^H \quad (12)$$

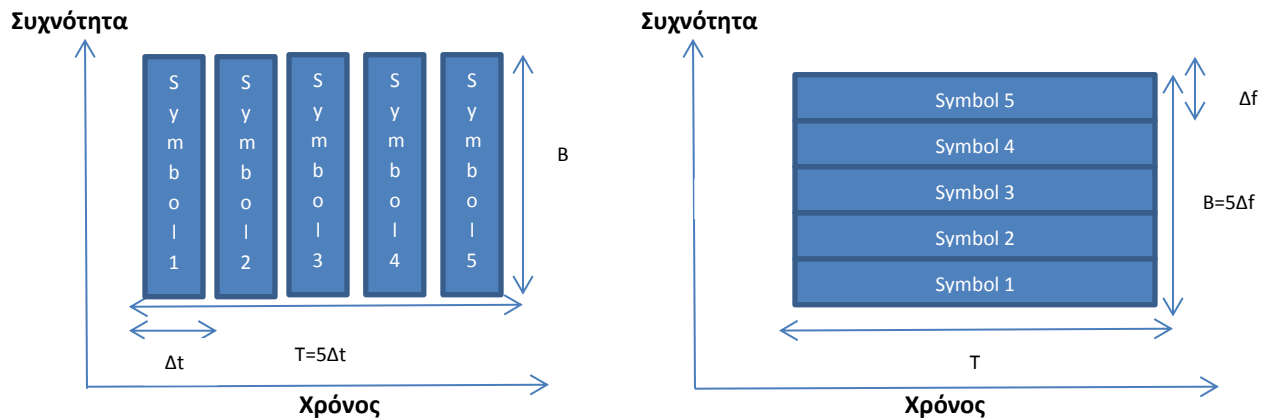
Με τον τρόπο αυτό, τα εκτιμώμενα μεταδιδόμενα σύμβολα μπορούν να υπολογιστούν χρησιμοποιώντας την εξίσωση (13):

$$\begin{bmatrix} \widehat{x}_1 \\ \widehat{x}_2 \end{bmatrix} = H^+ \begin{bmatrix} y_1^1 \\ y_2^1 \\ y_1^{2*} \\ y_2^{2*} \end{bmatrix} \quad (13)$$

# Κεφάλαιο 6 - Ορθογωνική Πολυπλεξία Διαίρεσης Συχνότητας (OFDM)

## 6.1 Γενική Περιγραφή

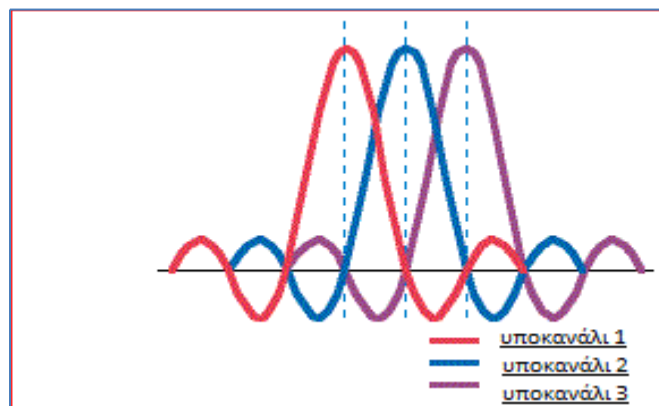
Βασικό χαρακτηριστικό της ορθογωνικής πολυπλεξίας διαίρεσης συχνότητας (OFDM) είναι η διαίρεση του προς μετάδοση ευρυζωνικού σήματος (wideband) σε μεγάλο αριθμό παράλληλων σημάτων στενής ζώνης. Στα συμβατικά συστήματα σειριακής μετάδοσης, γίνεται κατάληψη όλου του διαθέσιμου φάσματος με παλμούς πολύ μικρής διάρκειας. Αντίθετα, στην τεχνική OFDM, τα προς μετάδοση ψηφιακά δεδομένα διοχετεύονται σε παράλληλα υποκανάλια (OFDM subchannels/subcarriers) και μεταδίδονται με χαμηλότερο ρυθμό μετάδοσης. Με τον τρόπο αυτό δημιουργούνται πολλαπλά παράλληλα υποσυστήματα στενής ζώνης (narrowband), τα οποία μεταδίδουν ταυτόχρονα ένα μικρό μέρος της πληροφορίας με χαμηλό ρυθμό μετάδοσης, ενώ ο συνολικός ρυθμός μετάδοσης παραμένει σταθερός.



Σχήμα 6.1: Σειριακή και παράλληλη μετάδοση δεδομένων

Στο Σχήμα 6.1 απεικονίζεται ένα σειριακό σύστημα μετάδοσης και ένα σύστημα με 5 υποκανάλια. Στο σειριακό σύστημα, η μετάδοση του κάθε συμβόλου έχει πολύ μικρή χρονική διάρκεια ( $\Delta t$ ) και καταλαμβάνει εύρος συχνοτήτων  $B$ . Στο παράλληλο σύστημα, η χρονική διάρκεια της μετάδοσης ενός συμβόλου έχει πενταπλάσια διάρκεια ( $T = 5\Delta t$ ), αλλά καταλαμβάνει το 1/5 του διαθέσιμου φάσματος ( $\Delta f = B/5$ ). Παρατηρούμε πως και στις δύο περιπτώσεις, ο αριθμός των συμβόλων που μεταδίδονται είναι ίδιος (5 σύμβολα). Στο παράλληλο σύστημα, όμως, κάθε σύμβολο μεταδίδεται σε διάυλο στενής ζώνης, κάτι που καθιστά την ισοστάθμιση στον δέκτη ευκολότερη.

Ένα από τα βασικά πλεονεκτήματα της τεχνικής OFDM, σε σύγκριση με την απλή πολυπλεξία διαίρεσης συχνότητας (FDM), είναι η οικονομία φάσματος, η οποία επιτυγχάνεται λόγω της αλληλοκάλυψης των υποκαναλιών. Η αλληλοκάλυψη των καναλιών γίνεται εφικτή, λόγω της μεταξύ τους ορθογωνιότητας. Η ορθογωνιότητα εξασφαλίζει πως στα σημεία όπου το φάσμα ενός υποκαναλιού παρουσιάζει κορυφή, το φάσμα των υπολοίπων καναλιών παρουσιάζει μηδενισμό (Σχήμα 6.2). Για να ισχύσει αυτό, θα πρέπει οι φέρουσες συχνότητες των υποκαναλιών να αποτελούν ακέραια πολλαπλάσια μιας βασικής συχνότητας. Φυσικά, για να ισχύει η συνθήκη ορθογωνιότητας, θα πρέπει να υπάρχει πολύ καλός συγχρονισμός (δηλαδή ευθυγράμμιση των subcarriers). Σε περίπτωση που υπάρχουν σφάλματα συγχρονισμού, τότε η ορθογωνιότητα παύει να ισχύει, με αποτέλεσμα την αλλοίωση του σήματος. Συνεπώς, η εξασφάλιση της ορθογωνιότητας των υποκαναλιών είναι ένα πολύ σημαντικό ζητούμενο στα συστήματα OFDM.



**Σχήμα 6.2 – Σύστημα OFDM με 3 υποκανάλια.**

**Στο σημείο όπου ένα από αυτά παρουσιάζει κορυφή, τα υπόλοιπα παρουσιάζουν μηδενισμό.**

## 6.2 Προβλήματα Συγχρονισμού

Τα προβλήματα συγχρονισμού μπορούν να δημιουργηθούν κυρίως από τους εξής παράγοντες:

- 1) Ταλαντωτές που χρησιμοποιούνται στον πομπό και στον δέκτη (carrier frequency offset).
- 2) Αποκλίσεις μεταξύ των κυκλωμάτων που χρησιμοποιούνται στον πομπό και στον δέκτη (sampling clock offset)
- 3) Σφάλματα στην εκτίμηση έναρξης των συμβόλων OFDM στον δέκτη (timing offset).



4) Θόρυβο φάσης των ταλαντωτών (phase offset).

Η επίδραση των σφαλμάτων συγχρονισμού εξαρτάται από την σχετική απόκλιση συχνότητας, δηλαδή από την πραγματική απόκλιση κανονικοποιημένη ως προς το εύρος του κάθε υποκαναλιού. Έτσι, μια συγκεκριμένη τιμή απόκλισης συχνότητας μπορεί, για παράδειγμα, να έχει αμελητέα επίδραση σε ένα σύστημα OFDM με μεγάλο εύρος υποκαναλιών, αλλά να είναι καταστροφική για ένα σύστημα με υποκανάλια μικρού εύρους.

Έστω ότι το RF σήμα που εκπέμπεται είναι  $s(t)$ . Αν  $r(t)$  είναι το λαμβανόμενο σήμα, τότε ισχύει:

$$r(t) = z(t)s(t) \quad (1)$$

όπου  $z(t)$  ο παράγοντας που μοντελοποιεί την αλλοίωση του σήματος λόγω απώλειας συγχρονισμού.

Στην περίπτωση της απόκλισης μεταξύ των συχνοτήτων των τοπικών ταλαντωτών πομπού και δέκτη, αν η σχετική απόκλιση είναι  $\varepsilon = \delta f / \Delta f$ , όπου  $\delta f$  η απόκλιση συχνότητας και  $\Delta f$  το κέρδος του κάθε υποκαναλιού, τότε ισχύει:

$$z(t) = e^{\frac{j2\pi\varepsilon t}{T_s}} \quad (2)$$

Αμελώντας τον θόρυβο, η  $m$ -οστή έξοδος του FFT στον δέκτη είναι:

$$Y_m = a s_m - \sum_{k \neq m} \frac{\sin(\pi\varepsilon)}{\pi(m-k-\varepsilon)} e^{j\pi\varepsilon} s_k \quad (3)$$

όπου ο συντελεστής  $a$  εκφράζει την εξασθένηση του σήματος που προκαλείται από τον δίαυλο, ενώ το άθροισμα αντιστοιχεί στην ενδοσυμβολική παρεμβολή που προκαλείται στο  $m$ -οστό υποκανάλι από τα σήματα των υπολοίπων υποκαναλιών, λόγω έλλειψης ορθογωνιότητας.

Αν η περίοδος δειγματοληψίας στον δέκτη είναι  $(1 + \beta)T_{s\text{amp}}$ , όπου  $T_{s\text{amp}}$  είναι η περίοδος δειγματοληψίας στον πομπό, τότε η μέση ισχύς της ενδοσυμβολικής παρεμβολής στο  $m$ -οστό υποκανάλι είναι:

$$P(m) = \frac{\pi}{3} \beta^2 m^2 \quad (4)$$

Στην περίπτωση που υπάρχει σφάλμα χρονισμού  $\Delta t$  στην εκτίμηση των ορίων ενός λαμβανόμενου συμβόλου OFDM και το σφάλμα αυτό είναι μικρότερο από τη διάρκεια

φύλαξης ( $\Delta t \leq T_G$ ), τότε στο λαμβανόμενο σήμα προκαλείται στροφή φάσης, η οποία για το  $m$ -οστό υποκανάλι είναι ίση με  $e^{\frac{j2\pi m\Delta t}{T_s}}$ .

### 6.3 Κυκλικό Πρόθεμα

Ένα ακόμη σημαντικό πλεονέκτημα της διαμόρφωσης OFDM είναι η προστασία του σήματος από τη διασυμβολική παρεμβολή, η οποία προκαλείται από την πολυδιαδρομική διάδοση. Αυτό γίνεται εφικτό, λόγω της πολύ μεγαλύτερης χρονικής διάρκειας του συμβόλου, σε σχέση με τα σειριακά συστήματα. Εάν το OFDM σύστημα χρησιμοποιεί  $N$  υποκανάλια, τότε η χρονική διάρκεια του συμβόλου είναι  $N$  φορές μεγαλύτερη. Με τον τρόπο αυτό καθίσταται δυνατή η χρήση χρονικών διαστημάτων φύλαξης.

Η χρονική διάρκεια του διαστήματος φύλαξης θα πρέπει να είναι ίση ή μεγαλύτερη με τη χρονική διασπορά που εισάγει ο διάυλος, ώστε να αποφευχθεί η διασυμβολική παρεμβολή. Φυσικά, κατά τη διάρκεια του, δεν μεταδίδεται πληροφορία, κάτι που υποβαθμίζει το ρυθμό μετάδοσης δεδομένων. Αν η διάρκεια του διαστήματος φύλαξης είναι  $T_G$  και η διάρκεια του συμβόλου είναι  $T_U$ , ο ρυθμός μετάδοσης μειώνεται κατά  $T_U/(T_U + T_G)$ . Ως εμπειρικός κανόνας, έχει οριστεί η μέγιστη χρονική διάρκεια του διαστήματος φύλαξης στο 20% της συνολικής διάρκειας του OFDM συμβόλου, δηλαδή ισχύει:

$$T_{tot} = T_U + T_G \quad (5) \quad \text{και} \quad T_G = T_{tot}/4 \quad (6)$$

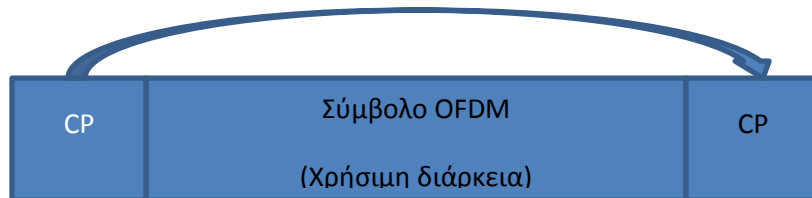
Συνεπώς ο ρυθμός μετάδοσης μειώνεται κατά:

$$10 \log_{10} \left( \frac{T_U}{T_U + T_G} \right) = 10 \log_{10} \left( \frac{T_s}{T_s + \frac{1}{4}T_s} \right) = 10 \log_{10} 0.8 = -1 \text{ dB} \quad (7)$$

Η μείωση του ρυθμού μετάδοσης είναι πολύ μικρή, κάτι που δεν είναι εφικτό στα συστήματα σειριακής μετάδοσης, διότι η μέγιστη χρονική διασπορά του διαύλου είναι αρκετά μεγάλη σε σχέση με την χρονική διάρκεια του συμβόλου. Στην περίπτωση αυτή, λοιπόν, δεν συνίσταται η χρήση διαστημάτων φύλαξης, διότι υποβιβάζεται κατά πολύ ο ρυθμός μετάδοσης δεδομένων.

Κατά τη διάρκεια του διαστήματος φύλαξης, ο δέκτης απορρίπτει όποια σήματα λαμβάνει, ενώ ο πομπός μπορεί να μην εκπέμπει καθόλου (zero-prefix). Όμως, σε ένα σύστημα OFDM, που υλοποιείται με μετασχηματισμό Fourier, είναι αποδοτικότερο, κατά τη διάρκεια του διαστήματος φύλαξης να γίνεται εκπομπή ενός τμήματος του συμβόλου. Πιο ειδικά, γίνεται εκπομπή μιας κυκλικής επέκτασής του (Cyclic Prefix), καθιστώντας την

ισοστάθμιση στον δέκτη πιο αποδοτική, αφού αυτή μετατρέπεται σε μία απλή μιγαδική διαίρεση, λόγω της ευκρίτητας του πολλαπλασιασμού στη συχνότητα με την κυκλική συνέλιξη στο πεδίο του χρόνου. Συνεπώς το διάστημα φύλαξης καταλαμβάνεται από ένα τμήμα της αρχής του προς μετάδοση συμβόλου, το οποίο δημιουργείται με την μορφή κυκλικού προθέματος.



Σχήμα 6.3 – Προσθήκη Κυκλικού Προθέματος στο τέλος (Cyclic Prefix)

## 6.4 Βασική Υλοποίηση Συστήματος OFDM

### 6.4.1 Πομπός

Η βασική υλοποίηση του πομπού ενός συστήματος OFDM αποτελείται από τα εξής βήματα:

- 1) Κωδικοποίηση του προς μετάδοση σήματος και διαχωρισμός του σε πακέτα.
- 2) Διαμόρφωση του σήματος στη βασική ζώνη, με μία μέθοδο ψηφιακής διαμόρφωσης (στην συγκεκριμένη περίπτωση χρησιμοποιήθηκε διαμόρφωση QPSK).
- 3) Μετατροπή του σήματος από σειριακό σε παράλληλο (serial to parallel conversion), ώστε να εισαχθεί στον IFFT (Αντίστροφος Ταχύς Μετασχηματισμός Fourier).
- 4) Εκτέλεση του Αντίστροφου Μετασχηματισμού Fourier. Με τον τρόπο αυτό, τα σύμβολα από το πεδίο της συχνότητας, πριν την μετατροπή, μεταφέρονται στο πεδίο του χρόνου. Στην έξοδο, λοιπόν, του IFFT, έχουμε χρονικά δείγματα του προς μετάδοση OFDM συμβόλου.
- 5) Προσθήκη Κυκλικού Προθέματος σε κάθε σύμβολο OFDM.
- 6) Μετατροπή του σήματος εξόδου του IFFT από παράλληλο σε σειριακό (parallel to serial conversion). Αν ο IFFT έχει μέγεθος  $N$ , ενώ το διάστημα φύλαξης αντιστοιχεί σε  $G$  δείγματα, τότε για την υλοποίηση του κυκλικού προθέματος, ο μετατροπέας έχει μέγεθος  $N + G$ .
- 7) Μετατροπή του σήματος σε αναλογικό (Digital to Analog conversion-DAC), ώστε να μεταφερθεί στις συχνότητες εκπομπής.

Αν τα προς μετάδοση ψηφιακά σύμβολα είναι  $\{X_k\}_{k=0}^{N-1}$ , τότε το εκπεμπόμενο OFDM σήμα βασικής ζώνης είναι η έξοδος του IFFT. Η  $n$ -οστή έξοδος του IFFT είναι:

$$x_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X_k e^{j2\pi nk/N}, \quad 0 \leq n \leq N-1 \quad (8)$$

### 6.4.2 Δέκτης

Στον δέκτη ακολουθείται η αντίστροφη διαδικασία:

- 1) Λήψη του σήματος και μετατροπή του στην βασική ζώνη.
- 2) Μετατροπή του σήματος σε ψηφιακό (Analog to Digital conversion-ADC).
- 3) Αφαίρεση του Κυκλικού Προθέματος από κάθε σύμβολο OFDM.

Η χρήση του Κυκλικού Προθέματος συμβάλλει σημαντικά στην αναίρεση της διασυμβολικής παρεμβολής. Στον πομπό, το εκπεμπόμενο σύμβολο  $x$  προκύπτει με αντίστροφο μετασχηματισμό Fourier (IFFT) του διανύσματος εισόδου  $X$ . Ακολουθεί η κυκλική επέκταση του διανύσματος, οπότε η συνέλιξη του διανύσματος  $x^{CP}$  με την κρουστική απόκριση του διαύλου  $h$  είναι ισοδύναμη με την κυκλική συνέλιξη του διανύσματος  $x$  με την κρουστική απόκριση του διαύλου  $h$ :

$$y = x^{CP} h = x \otimes h \quad (9)$$

(Το σύμβολο  $\otimes$  εκφράζει την πράξη της κυκλικής συνέλιξης). Στον δέκτη λοιπόν, έχουμε:

$$Y = FFT\{y\} = FFT\{x \otimes h\} = FFT\{x\} * FFT\{h\} = X * H \quad (10)$$

όπου  $H = FFT\{h\}$  (Το σύμβολο  $*$  εκφράζει τον πολλαπλασιασμό σημείο προς σημείο). Γνωρίζοντας το λαμβανόμενο σήμα  $Y$  και τον πίνακα  $H$  (με χρήση τεχνικών εκτίμησης διαύλου), είναι εφικτή η ανάκτηση του, αν διαιρέσουμε σημείο προς σημείο τα  $Y$  και  $H$ . Χωρίς την χρήση Κυκλικού Προθέματος, η συνέλιξη μεταξύ του εκπεμπόμενου σήματος και του διαύλου θα ήταν γραμμική, κάτι που θα καθιστούσε πολύ δυσκολότερη την ανάκτηση του εκπεμπόμενου σήματος.

- 4) Μετατροπή του σήματος από σειριακό σε παράλληλο (serial to parallel conversion), ώστε να εισαχθεί στον FFT (Ταχύς Μετασχηματισμός Fourier).
- 5) Εκτέλεση Μετασχηματισμού Fourier. Με τον τρόπο αυτό, από το πεδίο του χρόνου, μεταφερόμαστε στο πεδίο της συχνότητας. Η  $m$ -οστή έξοδος του FFT είναι:

$$Y_m = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} y_n e^{-j2\pi nm/N} = \frac{1}{N} \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} X_k e^{j2\pi nk/N} e^{-j2\pi nm/N} =$$

$$= \frac{1}{N} \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} X_k e^{\frac{j2\pi n(k-m)}{N}} = \frac{1}{N} \sum_{n=0}^{N-1} X_m = X_m, \quad 0 \leq m \leq N-1 \quad (11)$$

- 6) Εκτέλεση ισοστάθμισης του σήματος σύμφωνα με τα αναφερόμενα στο σημείο 3.
- 7) Μετατροπή του σήματος από παράλληλο σε σειριακό (parallel to serial conversion).
- 8) Αποδιαμόρφωση/αποκωδικοποίηση του σήματος.

## 6.5 Επιλογή παραμέτρων συστήματος

Για την σωστή λειτουργία ενός συστήματος OFDM η επιλογή των βασικών παραμέτρων πρέπει να πληροί ορισμένες προϋποθέσεις. Η χρονική διάρκεια του διαστήματος φύλαξης ( $T_G$ ) πρέπει να είναι τουλάχιστον ίση με τη μέγιστη χρονική διασπορά του διαύλου. Επιπλέον, η χρονική διάρκεια ( $T_U$ ), που ονομάζεται χρήσιμη διάρκεια του συμβόλου OFDM, πρέπει να είναι, κατά κανόνα, τουλάχιστον τετραπλάσια από τη διάρκεια του διαστήματος φύλαξης. Με τον τρόπο αυτό, εξασφαλίζεται η προστασία του σήματος από την διασυμβολική παρεμβολή, χωρίς μεγάλη μείωση του ρυθμού μετάδοσης.

Η χρήσιμη διάρκεια του συμβόλου καθορίζει άμεσα τον αριθμό των φερόντων που θα χρησιμοποιηθούν. Η τιμή της είναι αντιστρόφως ανάλογη της απόστασης μεταξύ των φερόντων των γειτονικών υποκαναλιών  $\Delta f = \frac{1}{T_U}$ . Όταν αυξάνεται η τιμή της  $T_U$  μειώνεται η απόσταση μεταξύ των φερόντων. Γενικά, το εύρος των υποκαναλιών πρέπει να είναι αρκετά μεγάλο, έτσι ώστε να μην επηρεάζεται από ενδεχόμενες μετατοπίσεις στις συχνότητες των φερόντων. Επιπλέον, η μέγιστη χρονική διασπορά του διαύλου  $\tau_{max}$  καθορίζει ένα κάτω όριο για την χρήσιμη διάρκεια  $T_U$ , αφού μεγάλες τιμές της  $\tau_{max}$  απαιτούν τη χρήση μεγάλου διαστήματος φύλαξης  $T_G$ , άρα και μεγάλη χρήσιμη διάρκεια συμβόλου  $T_U$ .



# Κεφάλαιο 7 - Ανάλυση Κώδικα Εφαρμογής

## 7.1 Ανάλυση Κώδικα Πομπού

Στο κεφάλαιο αυτό, γίνεται βηματικά η ανάλυση των μεθόδων που χρησιμοποιήθηκαν για την υλοποίηση του πομπού (Κεφάλαιο 3 – σχήμα 1). Τα δεδομένα εισόδου που χρησιμοποιούνται είναι τα εξής:

Αριθμός OFDM Frames\* = 1 frame

Αριθμός υποκαναλιών OFDM\* = 4 υποκανάλια

Μέγεθος Κυκλικού Προθέματος\* = 1 δείγμα

Αριθμός συμβόλων δεδομένων (QPSK) σε κάθε υποκανάλι OFDM\* = 8 σύμβολα

(Με αστερίσκο σημειώνονται οι μεταβλητές των οποίων οι τιμές καταχωρούνται από τον χρήστη μέσω του GUI της εφαρμογής. Θα πρέπει να αναφερθεί πως για τις μετρήσεις χρησιμοποιήθηκαν διαφορετικές παράμετροι που θα αναφερθούν στο κεφάλαιο 8).

### 7.1.1 Βήμα 1: Δημιουργία των δεδομένων (παράλληλη μορφή)

Η δημιουργία της δεδομένων (data bits) που θα μεταδοθούν από τον πομπό γίνεται από την μέθοδο:

```
vector <vector<int>> dataBitsGenerator (int OFDM_SUBCARRIERS_NUMBER, int BITS_PER_SUBCARRIER_NUMBER)
```

#### Δεδομένα εισόδου:

- 1) **int OFDM\_SUBCARRIERS\_NUMBER**: αριθμός των υποκαναλιών OFDM που έχει το προς μετάδοση OFDM frame.
- 2) **int BITS\_PER\_SUBCARRIER\_NUMBER**: αριθμός των bits δεδομένων που περιέχονται σε κάθε OFDM frame.

Η μέθοδος dataBitsGenerator δημιουργεί τα data bits, δηλαδή ακέραιους αριθμούς με τιμές 0 ή 1, που περιέχονται σε κάθε frame. Για την δημιουργία τους ακολουθείται ομοιόμορφη κατανομή, της οποίας η συνάρτηση πυκνότητας πιθανότητας είναι:

$$P(i|a, b) = \frac{1}{b-a+1} \quad \text{με} \quad a \leq i \leq b$$

#### Δεδομένα εξόδου:

**vector<vector<int>> dataBitsVector**: vector διαστάσεων (OFDM\_SUBCARRIERS\_NUMBER x BITS\_PER\_SUBCARRIER\_NUMBER). Στην ουσία είναι ένα frame που περιέχει την προς μετάδοση πληροφορία (data bits frame), σε παράλληλη μορφή.

Στην συγκεκριμένη περίπτωση έχουμε:

OFDM\_SUBCARRIERS\_NUMBER=4

BITS\_PER\_SUBCARRIER\_NUMBER=2\*(Αριθμό συμβόλων QPSK)=2\*8=16 bits

Η έξοδος της μεθόδου είναι το vector dataBitsVector που έχει την παρακάτω μορφή:

<b>b<sub>11</sub></b>	<b>b<sub>12</sub></b>	<b>b<sub>13</sub></b>	<b>b<sub>14</sub></b>	...	<b>b<sub>116</sub></b>
<b>b<sub>21</sub></b>	<b>b<sub>22</sub></b>	<b>b<sub>23</sub></b>	<b>b<sub>24</sub></b>	...	<b>b<sub>216</sub></b>
<b>b<sub>31</sub></b>	<b>b<sub>32</sub></b>	<b>b<sub>33</sub></b>	<b>b<sub>34</sub></b>	...	<b>b<sub>316</sub></b>
<b>b<sub>41</sub></b>	<b>b<sub>42</sub></b>	<b>b<sub>43</sub></b>	<b>b<sub>44</sub></b>	...	<b>b<sub>416</sub></b>

**Σχήμα 7.1 – vector dataBitsVector**

όπου  $b_{ij}$  τα data bits έχουν δημιουργηθεί. Στο συγκεκριμένο παράδειγμα το dataBitsVector είναι διαστάσεων 4x16.

### 7.1.2 Βήμα 2: Διαμόρφωση QPSK

Η διαμόρφωση των data bits του βήματος (1) σε σύμβολα QPSK πραγματοποιείται από την μέθοδο:

```
vector <vector<complex<int>>> qpskModulator (vector <vector<int>> dataBitsVector,  
int OFDM_SUBCARRIERS_NUMBER, int QPSK_SYMBOLS_PER_SUBCARRIER_NUMBER, int  
frameLoopsNumber, vector <vector <complex<int>>> preambleVector)
```

#### Δεδομένα εισόδου:

- 1) **vector**<**vector** <**int**>> **dataBitsVector**: περιέχει τα προς μετάδοση data bits του βήματος 1.
- 2) **int** **OFDM\_SUBCARRIERS\_NUMBER**: αριθμός των υποκαναλιών OFDM που έχει το προς μετάδοση OFDM frame.
- 3) **int** **QPSK\_SYMBOLS\_PER\_SUBCARRIER\_NUMBER**: αριθμός των συμβόλων QPSK (δεδομένα) που περιέχονται σε κάθε υποκάνάλι OFDM.
- 4) **int** **frameLoopsNumber**: αριθμός τρέχουσας επανάληψης (αριθμός δημιουργηθέντος OFDM frame).
- 5) **vector** <**vector** <**complex**<**int**>>> **preambleVector**: περιέχει τα σύμβολα QPSK που αποτελούν το preamble για κάθε OFDM frame.

Η μετατροπή των bits σε σύμβολα QPSK γίνεται ακολουθώντας την παρακάτω αντιστοιχία:



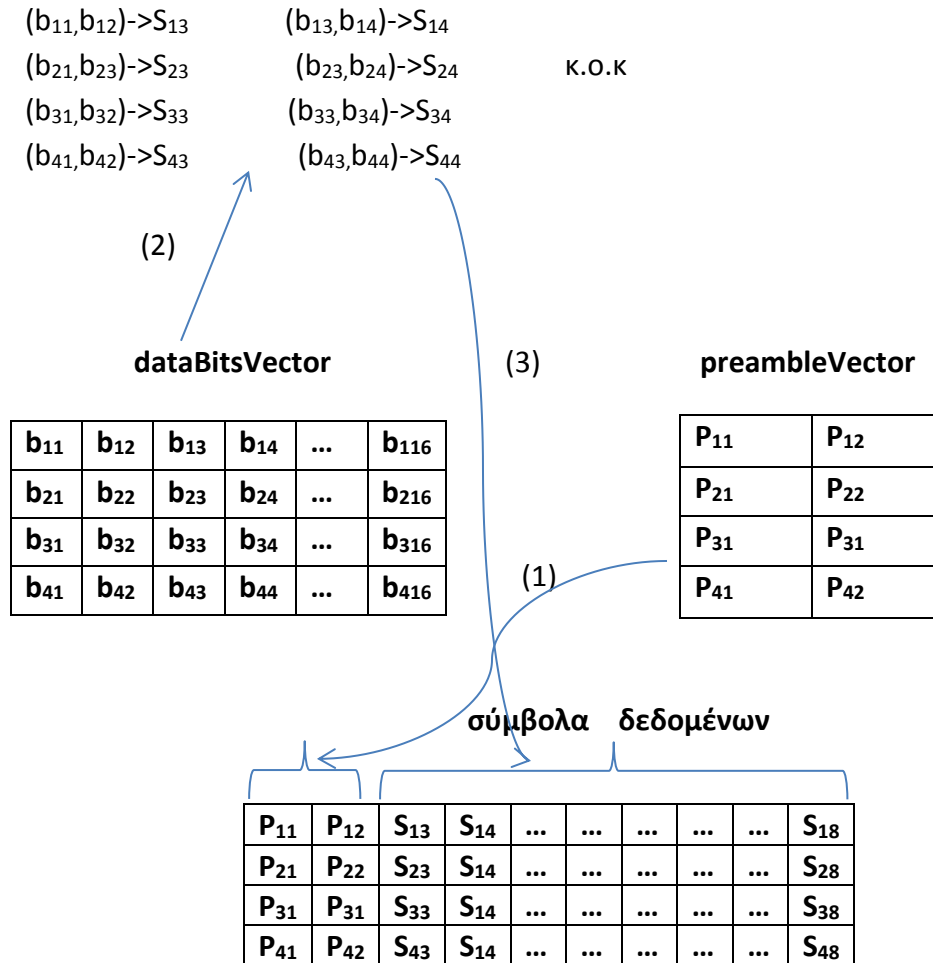
Ακολουθία data bits	Σύμβολο QPSK
(1,1)	(1+1i)
(0,1)	(-1+1i)
(0,0)	(-1-1i)
(1,0)	(1-1i)

**Πίνακας 7.1 – Αντιστοίχιση bits σε σύμβολα QPSK**

Όπως παρατηρούμε, η διαμόρφωση ακολουθεί την κωδικοποίηση Gray, κατά την οποία κάθε σύμβολο διαφέρει από το προηγούμενο και το επόμενο του, μόνο κατά ένα bit.

Στην αρχή κάθε υποκαναλιού OFDM, στην θέση των δύο πρώτων συμβόλων QPSK, γίνεται η εισαγωγή των συμβόλων του preamble, το οποίο είναι κοινό και γνωστό, τόσο στον πομπό όσο και στον δέκτη. Έπειτα γίνεται η μετατροπή των data bits σε σύμβολα QPSK, δηλαδή σε μιγαδικούς αριθμούς, βάσει της αντιστοιχίας του Πίνακα 7.1. Η διαδικασία εμφανίζεται στο Σχήμα 7.2 που ακολουθεί:

- 1) Προσθήκη των συμβόλων του preamble στο vector `qpskModulatedVector`
- 2) Μετατροπή των data bits ( $b_{ij}$ ) σε σύμβολα QPSK ( $S_{uv}$ )
- 3) Προσθήκη των συμβόλων στο vector `qpskModulatedvector`:



**Σχήμα 7.2 – Δημιουργία vector qpskModulatedVector**

**Δεδομένα Εξόδου:** `vector<vector<complex<int>>> qpskModulatedVector`, που είναι ένα vector διαστάσεων `(OFDM_SUBCARRIERS_NUMBER x QPSK_SYMBOLS_PER_SUBCARRIER_NUMBER)`. Το `qpskModulatedVector` περιέχει την προς μετάδοση πληροφορία, μετά την ψηφιακή διαμόρφωση (QPSK modulation). Στο συγκεκριμένο παράδειγμα το vector `qpskModulatedVector` έχει διαστάσεις 4x8.

### 7.1.3 Βήμα 3: Κωδικοποίηση Alamouti

Η κωδικοποίηση Alamouti πραγματοποιείται από 2 μεθόδους οι οποίες είναι:

- 1) `vector<vector<complex<double>> >alamoutiSymbolsVectorSpaceTimeCoderAntenna1` (vector < vector < complex < int > > > `qpskModulatedVector`, int `OFDM_SUBCARRIERS_NUMBER`, int `QPSK_SYMBOLS_PER_SUBCARRIERS_NUMBER`).

Η συγκεκριμένη μέθοδος πραγματοποιεί την κωδικοποίηση Alamouti για τα δεδομένα που πρόκειται να μεταδοθούν από την πρώτη κεραία του πομπού.

2) **vector<vector<complex<double>> >alamoutiSymbolsVectorSpaceTimeCoderAntenna2**  
(vector < vector < complex < int > > > qpskModulatedVector, int OFDM\_SUBCARRIERS\_NUMBER, int QPSK\_SYMBOLS\_PER\_SUBCARRIERS\_NUMBER) .

Η μέθοδος αυτή πραγματοποιεί την κωδικοποίηση Alamouti για τα δεδομένα που πρόκειται να μεταδοθούν από την δεύτερη κεραία του πομπού.

### Δεδομένα εισόδου:

Οι δύο μέθοδοι δέχονται τα ίδια δεδομένα εισόδου:

- 1) **vector <vector<complex<int>>> qpskModulatedVector:** περιέχει τα σύμβολα δεδομένων που πρόκειται να μεταδοθούν (είναι το vector που δημιουργείται στο βήμα 2).
- 2) **int OFDM\_SUBCARRIERS\_NUMBER:** αριθμός των υποκαναλιών OFDM που έχει το προς μετάδοση OFDM frame.
- 3) **int QPSK\_SYMBOLS\_PER\_SUBCARRIER\_NUMBER:** αριθμός των συμβόλων QPSK που περιέχονται σε κάθε υποκανάλι OFDM.

Στην συγκεκριμένη περίπτωση, από το βήμα 2 έχουμε το vector qpskModulatedVector με διαστάσεις (OFDM\_SUBCARRIERS\_NUMBER x QPSK\_SYMBOLS\_PER\_SUBCARRIER\_NUMBER).

Ισχύει:

OFDM\_SUBCARRIERS\_NUMBER=4 υποκανάλια

QPSK\_SYMBOLS\_PER\_SUBCARRIER\_NUMBER=8 σύμβολα

Το qpskModulatedVector έχει την παρακάτω μορφή:

$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$	$X_7$	$X_8$
$X_9$	$X_{10}$	$X_{11}$	$X_{12}$	$X_{13}$	$X_{14}$	$X_{15}$	$X_{16}$
$X_{17}$	$X_{18}$	$X_{19}$	$X_{20}$	$X_{21}$	$X_{22}$	$X_{23}$	$X_{24}$
$X_{25}$	$X_{26}$	$X_{27}$	$X_{28}$	$X_{29}$	$X_{30}$	$X_{31}$	$X_{32}$

**Σχήμα 7.3 – (vector qpskModulatedVector)**

όπου  $X_i$  είναι τα σύμβολα QPSK του frame που πρόκειται να μεταδοθεί.

Σύμφωνα με την κωδικοποίηση Alamouti, η αποστολή δύο διαδοχικών συμβόλων από τις δύο κεραίες του πομπού (MIMO 2x2) γίνεται σε δύο διαδοχικές χρονικές περιόδους με τον ακόλουθο τρόπο:

### Χρονική περίοδος

	$t_1$	$t_2$
Κεραία	$X_1$	$-X_2^*$
	$X_2$	$X_1^*$

Πίνακας 7.2

Βάσει του Πίνακα 7.2 το αποτέλεσμα της κωδικοποίησης Alamouti του vector `qpskModulatedVector` για τα σύμβολα που θα μεταδοθούν από την πρώτη κεραία είναι το εξής:

### Χρονική Περίοδος

	$t_1=0$	$t_2=1$	$t_3=2$	$t_4=3$	$t_5=4$	$t_6=5$	$t_7=6$	$t_8=7$
Μεταδιδόμενα Σύμβολα	$X_1$	$-X_2^*$	$X_3$	$-X_4^*$	$X_5$	$-X_6^*$	$X_7$	$-X_8^*$
	$X_9$	$-X_{10}^*$	$X_{11}$	$-X_{12}^*$	$X_{13}$	$-X_{14}^*$	$X_{15}$	$-X_{16}^*$
	$X_{17}$	$-X_{18}^*$	$X_{19}$	$-X_{20}^*$	$X_{21}$	$-X_{22}^*$	$X_{23}$	$-X_{24}^*$
	$X_{25}$	$-X_{26}^*$	$X_{27}$	$-X_{28}^*$	$X_{29}$	$-X_{30}^*$	$X_{31}$	$-X_{32}^*$

Σχήμα 7.4 (vector `alamoutiSymbolsVectorAntenna1`)

Το αποτέλεσμα της κωδικοποίησης Alamouti του vector `qpskModulatedVector` για τα σύμβολα που θα μεταδοθούν από την δεύτερη κεραία, εμφανίζεται στο Σχήμα 7.5:

### Χρονική Περίοδος

	$t_1=0$	$t_2=1$	$t_3=2$	$t_4=3$	$t_5=4$	$t_6=5$	$t_7=6$	$t_8=7$
Μεταδιδόμενα Σύμβολα	$X_2$	$X_1^*$	$X_4$	$X_3^*$	$X_6$	$X_5^*$	$X_8$	$X_7^*$
	$X_{10}$	$X_9^*$	$X_{12}$	$X_{11}^*$	$X_{14}$	$X_{13}^*$	$X_{16}$	$X_{15}^*$
	$X_{18}$	$X_{17}^*$	$X_{20}$	$X_{19}^*$	$X_{22}$	$X_{21}^*$	$X_{24}$	$X_{25}^*$
	$X_{26}$	$X_{25}^*$	$X_{28}$	$X_{27}^*$	$X_{30}$	$X_{29}^*$	$X_{32}$	$X_{31}^*$

Σχήμα 7.5 (vector `alamoutiSymbolsVectorAntenna2`)

### Δεδομένα Εξόδου:

1) `vector<vector<complex<double>>> alamoutiSymbolsVectorAntenna1`, που είναι ένα vector διαστάσεων (OFDM\_SUBCARRIERS\_NUMBER x QPSK\_SYMBOLS\_PER\_SUBCARRIER\_NUMBER).

Αποτελεί την έξοδο της μεθόδου **alamoutiSymbolsVectorSpaceTimeCoderAntenna1** και περιέχει τα σύμβολα, μετά την κωδικοποίηση Alamouti, που θα μεταδοθούν από την πρώτη κεραία του πομπού. Στο συγκεκριμένο παράδειγμα το vector **alamoutiSymbolsVectorAntenna1** έχει διαστάσεις 4x8.

2) **vector<vector<complex<double>>> alamoutiSymbolsVectorAntenna2**, που είναι ένα vector διαστάσεων (OFDM\_SUBCARRIERS\_NUMBER x QPSK\_SYMBOLS\_PER\_SUBCARRIER\_NUMBER).

Αποτελεί την έξοδο της μεθόδου **alamoutiSymbolsVectorSpaceTimeCoderAntenna2** και περιέχει τα σύμβολα, μετά την κωδικοποίηση Alamouti, που θα μεταδοθούν από την δεύτερη κεραία του πομπού. Στο συγκεκριμένο παράδειγμα το vector **alamoutiSymbolsVectorAntenna2** έχει διαστάσεις 4x8.

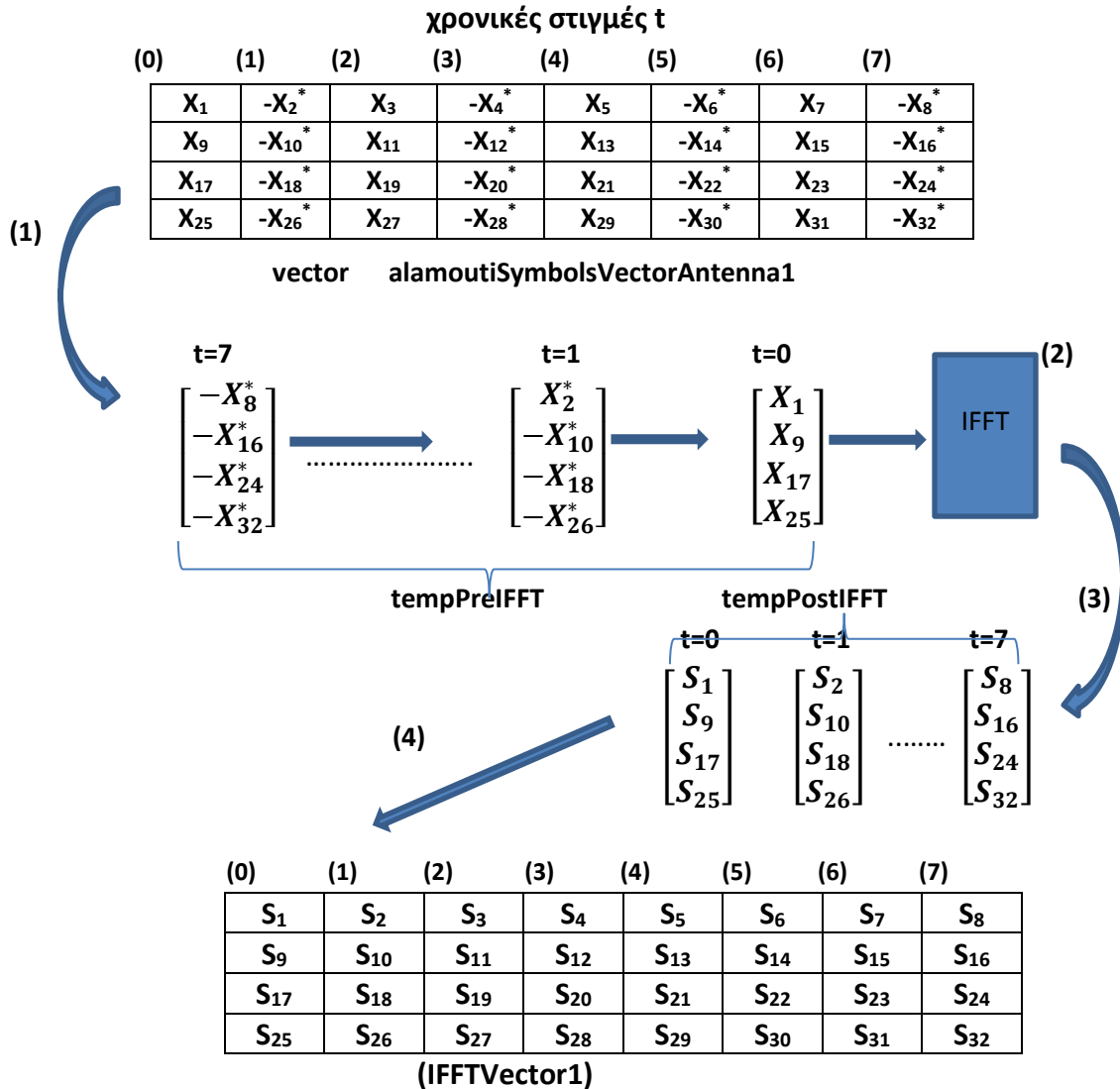
#### 7.1.4 Βήμα 4: IFFT-(Αντίστροφος Ταχύς Μετασχηματισμός Fourier)

Στο βήμα αυτό εκτελείται ο Αντίστροφος Ταχύς Μετασχηματισμός Fourier, με τον οποίο τα σύμβολα στο πεδίο της συχνότητας μετατρέπονται σε δείγματα (samples), στο πεδίο του χρόνου. Ο μετασχηματισμός εκτελείται από την μέθοδο **vector < vector <complex <double>> > IFFT (int OFDM\_SUBCARRIERS\_NUMBER, int QPSK\_SYMBOLS\_PER\_SUBCARRIER\_NUMBER, vector <vector<complex<double>>> symbolsVector)**. Οι αλγεβρικές πράξεις του μετασχηματισμού FFT εκτελούνται με κλήση των μεθόδων της βιβλιοθήκης FFTW-3.

##### Δεδομένα Εισόδου:

- 1) **int OFDM\_SUBCARRIERS\_NUMBER**: αριθμός των υποκαναλιών OFDM που έχει το προς μετάδοση OFDM frame.
- 2) **int QPSK\_SYMBOLS\_PER\_SUBCARRIER\_NUMBER**: αριθμός των συμβόλων QPSK που περιέχονται σε κάθε υποκανάλι OFDM.
- 3) **vector <vector<complex<double>>> symbolsVector**: περιέχει τα προς μετάδοση σύμβολα, μετά και την κωδικοποίηση Alamouti. Στην περίπτωση της πρώτης κεραίας, ως είσοδος εισάγεται το **vector<vector<complex<double>>> alamoutiSymbolsVectorAntenna1**, ενώ για αυτό της δεύτερης, η μέθοδος καλείται δεύτερη φορά, όπου εισάγεται το **vector<vector<complex<double>>> alamoutiSymbolsVectorAntenna2**.

Στο Σχήμα 7.6 παρουσιάζεται η εκτέλεση του Αντίστροφου Ταχέως Μετασχηματισμού Fourier για το σύμβολα της πρώτης κεραίας (vector **alamoutiSymbolsVectorAntenna1**):



**Σχήμα 7.6 – Εκτέλεση IFFT**

- 1) Τα δείγματα κάθε στήλης του vector alamoutiSymbolsVectorAntenna1 αντιγράφονται στον πίνακα (array) δυναμικής ανάθεσης μνήμης tempPreIFFT της βιβλιοθήκης FFTW, διαστάσεων (1 x OFDM\_SUBCARRIERS\_NUMBER).
- 2) Στα περιεχόμενα του πίνακα tempPreIFFT, εκτελείται Ταχύς μετασχηματισμός Fourier σε διαδοχικές χρονικές περιόδους  $t=0,1,\dots,(QPSK\_SYMBOLS\_PER\_SUBCARRIER\_NUMBER-1)$ .
- 3) Τα αποτελέσματα του μετασχηματισμού, δηλαδή τα δείγματα που πρόκειται να μεταδοθούν από την πρώτη κεραία, αποθηκεύονται στον πίνακα (array) δυναμικής ανάθεσης μνήμης tempPostIFFT της βιβλιοθήκης FFTW, διαστάσεων (1 x OFDM\_SUBCARRIERS\_NUMBER).

4) Τα δείγματα του πίνακα tempPostIFFT, αποθηκεύονται στο `vector<vector<complex<double>>> IFFTVector1`, διαστάσεων (OFDM\_SUBCARRIERS\_NUMBER x QPSK\_SYMBOLS\_PER\_SUBCARRIER\_NUMBER).

#### Δεδομένα Εξόδου:

**vector < vector < complex < double > > > FFTVector1:** Η μέθοδος IFFT, για την περίπτωση της πρώτης κεραίας επιστρέφει το `vector IFFTVector1`, το οποίο περιέχει τα δείγματα που τελικά θα μεταδοθούν από αυτήν. Στην περίπτωση της δεύτερης κεραίας, από την μέθοδο επιστρέφεται το `vector FFTVector2`. Τα δύο `vectors` είναι διαστάσεων (OFDM\_SUBCARRIERS\_NUMBER x QPSK\_SYMBOLS\_PER\_SUBCARRIER\_NUMBER). Στην συγκεκριμένη περίπτωση οι διαστάσεις τους είναι (4x8).

### 7.1.5 Βήμα 5: Μετατροπή σήματος από παράλληλο σε σειριακό Βήμα 6: Προσθήκη Κυκλικού προθέματος

Τα βήματα 5 και 6 πραγματοποιούνται από την μέθοδο `vector <complex<double>> parallelToSerial_CpAdder (int QPSK_SYMBOLS_PER_SUBCARRIER_NUMBER, int OFDM_SUBCARRIERS_NUMBER, int SERIAL_CP_VECTOR_SIZE, int CYCLIC_PREFIX_SIZE, vector <vector<complex<double>>> ifftVector)`.

#### Δεδομένα Εισόδου:

**1) int QPSK\_SYMBOLS\_PER\_SUBCARRIER\_NUMBER:** αριθμός των συμβόλων QPSK που περιέχονται σε κάθε υποκανάλι OFDM.

**2) int OFDM\_SUBCARRIERS\_NUMBER:** αριθμός των υποκαναλιών OFDM που έχει το προς μετάδοση OFDM frame.

**3) int SERIAL\_CP\_VECTOR\_SIZE:** μέγεθος του προς μετάδοση σήματος (αριθμός δειγμάτων) μετά την μετατροπή του σε παράλληλο και την προσθήκη του Κυκλικού προθέματος. Για την μεταβλητή SERIAL\_CP\_VECTOR\_SIZE ισχύει:

$SERIAL\_CP\_VECTOR\_SIZE = (OFDM\_SUBCARRIERS\_NUMBER + CYCLIC\_PREFIX\_SIZE) * QPSK\_SYMBOLS\_PER\_SUBCARRIER\_NUMBER$ , όπου CYCLIC\_PREFIX\_SIZE είναι το μέγεθος του

Κυκλικού προθέματος που έχει επιλεγεί από τον χρήστη για κάθε σύμβολο OFDM.

**4) int CYCLIC\_PREFIX\_SIZE:** μέγεθος του Κυκλικού Προθέματος (σε δείγματα), όπως αυτό έχει οριστεί από τον χρήστη.

**5) vector < vector<complex<double>>> ifftVector:** `vector` που περιέχει τα δείγματα που προκύπτουν μετά την εκτέλεση του IFFT. Για το σήμα που πρόκειται να μεταδοθεί από την πρώτη κεραία στην μέθοδο εισάγεται το `vector ifftVector1`, ενώ για το σήμα της δεύτερης κεραίας η μέθοδος καλείται δεύτερη φορά, όπου και εισάγεται το `vector ifftVector2`.

Η μέθοδος **parallelToSerial\_CpAdder** λαμβάνει τα δείγματα που προκύπτουν από την εκτέλεση του IFFT (ifftVector1 και ifftVector2) , τα οποία βρίσκονται σε παράλληλη μορφή. Το σήμα εισόδου μετατρέπεται σε σειριακό, ενώ ταυτόχρονα γίνεται και η προσθήκη του Κυκλικού προθέματος.

Σύμφωνα με το προηγούμενο βήμα το vector ifftVector1 έχει την παρακάτω μορφή:

(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>
S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>
S <sub>17</sub>	S <sub>18</sub>	S <sub>19</sub>	S <sub>20</sub>	S <sub>21</sub>	S <sub>22</sub>	S <sub>23</sub>	S <sub>24</sub>
S <sub>25</sub>	S <sub>26</sub>	S <sub>27</sub>	S <sub>28</sub>	S <sub>29</sub>	S <sub>30</sub>	S <sub>31</sub>	S <sub>32</sub>

**Σχήμα 7.7 – Παράλληλο σήμα (vector ifftVector1)**

Συνεπώς το πρώτο δείγμα (με κόκκινα γράμματα), κάθε στήλης του διανύσματος προστίθεται στο τέλος της, αποτελώντας το Κυκλικό πρόθεμα, μεγέθους ενός δείγματος.

(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>
S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>14</sub>	S <sub>15</sub>	S <sub>16</sub>
S <sub>17</sub>	S <sub>18</sub>	S <sub>19</sub>	S <sub>20</sub>	S <sub>21</sub>	S <sub>22</sub>	S <sub>23</sub>	S <sub>24</sub>
S <sub>25</sub>	S <sub>26</sub>	S <sub>27</sub>	S <sub>28</sub>	S <sub>29</sub>	S <sub>30</sub>	S <sub>31</sub>	S <sub>32</sub>
S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>

**Σχήμα 7.8 – Προσθήκη Κυκλικού Προθέματος**

Μετά την προσθήκη του Κυκλικού προθέματος, το σήμα μετατρέπεται από παράλληλο σε σειριακό και αποθηκεύεται στο vector cpSerialAntenna1 διαστάσεων (1x SERIAL\_CP\_VECTOR\_SIZE). Στην συγκεκριμένη περίπτωση ισχύει: SERIAL\_CP\_VECTOR\_SIZE= (4+1)\*8=40 samples

(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(35)	(36)	(37)	(38)	(39)					
S <sub>1</sub>	S <sub>9</sub>	S <sub>17</sub>	S <sub>25</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>10</sub>	S <sub>18</sub>	S <sub>26</sub>	S <sub>2</sub>	....	....	....	....	....	S <sub>8</sub>	S <sub>16</sub>	S <sub>24</sub>	S <sub>32</sub>	S <sub>8</sub>

**Σχήμα 7.9 – Σειριακό σήμα μετά την προσθήκη Κυκλικού Προθέματος (vector cpSerialAntenna1)**

**Δεδομένα Εξόδου:**

**vector <complex < double > > cpSerialAntenna1:** περιέχει το σήμα που πρόκειται να μεταδοθεί από την πρώτη κεραία του πομπού, μετά την διαδικασία της μετατροπής του από



παράλληλο σε σειριακό και την προσθήκη του Κυκλικού προθέματος. Έχει διαστάσεις (1x SERIAL\_CP\_VECTOR\_SIZE).

Στην περίπτωση του σήματος που πρόκειται να μεταδοθεί από την δεύτερη κεραία του πομπού, έξοδος της μεθόδου **parallelToSerial\_CpAdder** είναι το vector cpSerialAntenna2.

### 7.1.6 Βήμα 7: Interpolation

Κατά την φάση του Interpolation, γίνεται προσθήκη πλεοναζόντων (dummy) δειγμάτων με μηδενική τιμή μεταξύ των δειγμάτων του σήματος, με τελικό στόχο την αύξηση του εύρους ζώνης και την μείωση των πλευρικών λοβών της OFDM κυματομορφής. Η προσθήκη των δειγμάτων αυτών γίνεται μετά από κάθε δείγμα του προς μετάδοση σήματος. Το Interpolation εκτελείται από την μέθοδο: **vector<complex<double>> zeroInsertion** (int VECTOR\_SIZE\_AFTER\_ZERO\_INSERTION, int INTERPOLATION\_ZEROS\_NUMBER, vector < complex < double > > serialCpVector)

#### Δεδομένα εισόδου:

- 1) **int VECTOR\_SIZE\_AFTER\_ZERO\_INSERTION:** μέγεθος του προς μετάδοση σήματος, (cpSerialAntenna1 και cpSerialAntenna2), μετά και την προσθήκη των πλεοναζόντων δειγμάτων κατά την διαδικασία του Interpolation. Ισχύει: VECTOR\_SIZE\_AFTER\_ZERO\_INSERTION = 2 x SERIAL\_CP\_VECTOR\_SIZE
- 2) **int INTERPOLATION\_ZEROS\_NUMBER:** αριθμός των πλεοναζόντων δειγμάτων που εισάγονται στο προς μετάδοση σήμα κατά την διαδικασία του Interpolation. Ο αριθμός των δειγμάτων που εισάγονται είναι ίσος με το μέγεθος του σήματος, καθώς η αύξηση του ρυθμού είναι ανά 2, δηλαδή ισχύει: INTERPOLATION\_ZEROS\_NUMBER = SERIAL\_CP\_VECTOR\_SIZE
- 3) **vector < complex < double > > serialCpVector:** περιέχει τα δείγματα που προκύπτουν από την διαδικασία της μετατροπής του σήματος από παράλληλο σε σειριακό και την προσθήκη του Κυκλικού προθέματος (βήμα 5 και 6). Για το σήμα που πρόκειται να μεταδοθεί από την πρώτη κεραία εισάγεται το vector serialCpVector1 ενώ για αυτό της δεύτερης κεραίας εισάγεται το vector serialCpVector2.

Κατά την διαδικασία του Interpolation, μετά από κάθε δείγμα του σήματος, εισάγεται ένα πλεονάζον (dummy) δείγμα, δηλαδή ένας μηδενικός μιγαδικός αριθμός  $S_d=(0+0i)$ . Στο τέλος της διαδικασίας, το προς μετάδοση σήμα έχει την παρακάτω μορφή:

(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)					(74)	(75)	(76)	(77)	(78)	(79)
$S_1$	$S_d$	$S_9$	$S_d$	$S_{17}$	$S_d$	$S_{25}$	$S_d$	...	...	...	...	$S_{24}$	$S_d$	$S_{32}$	$S_d$	$S_8$	$S_d$

Σχήμα 7.10 – Σειριακό σήμα μετά το Interpolation (vector zeroInsertedAntenna1)

Μετά την ολοκλήρωση του συγκεκριμένου βήματος, το προς μετάδοση σήμα της πρώτης κεραίας αποθηκεύεται στο vector `zeroInsertedAntenna1`, που είναι ένα vector διαστάσεων ( $1 \times \text{VECTOR\_SIZE\_AFTER\_ZERO\_INSERTION}$ ). Στην συγκεκριμένη περίπτωση ισχύει  $\text{VECTOR\_SIZE\_AFTER\_ZERO\_INSERTION} = 2 \times \text{SERIAL\_CP\_VECTOR\_SIZE} = 2 \times 40 = 80$  δείγματα

#### **Δεδομένα Εξόδου:**

**vector < complex < double >> zeroInsertedAntenna1**, το οποίο είναι ένα vector διαστάσεων ( $1 \times \text{VECTOR\_SIZE\_AFTER\_ZERO\_INSERTION}$ ). Το vector **zeroInsertedAntenna1**, περιέχει το σήμα που θα μεταδοθεί από την πρώτη κεραία του πομπού, μετά την διαδικασία του Interpolation.

Στην περίπτωση του σήματος που θα μεταδοθεί από την δεύτερη κεραία, από την μέθοδο **zeroInsertion**, επιστρέφεται το vector **zeroInsertedAntenna2**

#### **7.1.7 Βήμα 8: Συνελκτικό Φιλτράρισμα - Convolution Filtering (Square Root Raised Cosine Filtering)**

Το Συνελκτικό φιλτράρισμα εκτελείται από την μέθοδο **vector < complex < double >> filterConvolution (int filterSize, int samplesNumber, int CONVOLUTION\_SIZE, vector < complex <double>> samplesVector, vector < double > filterCoefficientsVector)**.

#### **Δεδομένα εισόδου:**

- 1) int filterSize:** αριθμός συντελεστών του Square Root Raised Cosine φίλτρου (μέγεθος του vector `filterCoefficientsVector`, με μέγεθος `CONVOLUTION_FILTER_SIZE`)
- 2) int samplesNumber:** μέγεθος του προς μετάδοση σήματος πριν την εκτέλεση του convolution filtering. (Μέγεθος των vectors `zeroInsertedAntenna1` και `zeroInsertedAntenna2` του βήματος 8).
- 3) int CONVOLUTION\_SIZE:** μέγεθος του προς μετάδοση σήματος μετά την ολοκλήρωση του convolution filtering. Αν  $M$  είναι το μέγεθος του σήματος πριν το filtering και  $N$  το μέγεθος του φίλτρου, τότε ισχύει:

$\text{CONVOLUTION\_SIZE} = M + N - 1 = \text{VECTOR\_SIZE\_AFTER\_ZERO\_INSERTION} + \text{CONVOLUTION\_FILTER\_SIZE} - 1$  (δείγματα)

- 4) vector < complex <double>> samplesVector:** περιέχει το προς μετάδοση σήμα πριν την εκτέλεση του convolution filtering (vector `zeroInsertedAntenna1` και `zeroInsertedAntenna2`)
- 5) vector < double > filterCoefficientsVector:** περιέχει τις τιμές του Square Root Raised Cosine φίλτρου, το οποίο χρησιμοποιείται για την εκτέλεση του convolution filtering.

### Αλγόριθμος:

Ο κώδικας εκτέλεσης είναι ο παρακάτω (μέθοδος filterConvolution):

```
double a;
complex < double > complexB=complex <double> (0,0);
vector< complex < double > > convolutedvectorAntenna;
complex < double > complexY=complex <double> (0,0);

int minusValue=0;

for(int i=0;i<CONVOLUTION_SIZE;i++)//FOR LOOP 1 (1)
{
    for (int j=0;j<filterSize;j++)//FOR LOOP 2 (2)
    {
        a=filterCoefficientsVector[j]; (3)
        minusValue=i-j; (4)

        if ((minusValue<0) || (minusValue>=samplesNumber))//IF 1 (5)
        {
            complexB=complex < double > (0,0);
        }IF 1

        if ((minusValue>=0)&&(minusValue<samplesNumber))//IF 2 (6)
        {
            complexB=complex<double>(real(samplesVector[minusValue]),imag(samplesVector[minusValue]));
        }//IF 2
        complexY=complex<double>(real(complexY)+a*real(complexB),imag(complexY)+a*imag(complexB)); (7)

    }//FOR LOOP 2

    convolutedVectorAntenna[i]=complex <double> (real(complexY),imag(complexY)); (8)
    complexY=complex<double>(0,0);
}//FOR LOOP 1
```

- 1) Έναρξη του loop 1. Ο αριθμός των επαναλήψεων είναι ίσος με το μέγεθος του σήματος μετά την ολοκλήρωση του filtering.
- 2) Έναρξη του loop 2. Ο αριθμός των επαναλήψεων είναι ίσος με το μέγεθος του φίλτρου .
- 3) Ανάθεση τιμής στην μεταβλητή double a, η οποία είναι ίση με την j-οστή τιμή του vector filterCoefficientsVector.
- 4) Υπολογισμός της τιμής minusValue η οποία είναι ίση με την διαφορά των τιμών i-j.

- 5) Περίπτωση που η τιμή της μεταβλητής `minusValue` είναι μικρότερη από μηδέν ή είναι μικρότερη ή ίση από το μέγεθος του σήματος πριν το `filtering`. Στην περίπτωση αυτή στην μιγαδική μεταβλητή `complexB` αποδίδεται η τιμή μηδέν.
- 6) Περίπτωση που η τιμή της μεταβλητής `minusValue` είναι μεγαλύτερη ή ίση με μηδέν και ταυτόχρονα μικρότερη από το μέγεθος του σήματος πριν το `filtering`. Στη μιγαδική μεταβλητή `complexB` αποδίδεται η τιμή της (i-j)-οστής θέσης του σήματος πριν το `filtering`.
- 7) Στην μεταβλητή `complexY`, αποδίδεται η τιμή της μεταβλητής `complexB`.
- 8) Η τιμή της μεταβλητής `complexY`, αποθηκεύεται στην i-οστή θέση του `vector convolutedVectorAntenna`, που περιέχει το σήμα μετά την ολοκλήρωση του `filtering`.

Μετά την ολοκλήρωση του συγκεκριμένου βήματος το προς μετάδοση σήμα της πρώτης κεραίας έχει μέγεθος: (1x CONVOLUTION\_SIZE). Για την συγκεκριμένη περίπτωση ισχύει: CONVOLUTION\_SIZE=VECTOR\_SIZE\_AFTER\_ZERO\_INSERTION+ CONVOLUTION\_FILTER\_SIZE-1= 80+51-1=130 δείγματα

#### Δεδομένα Εξόδου:

**vector < complex < double >> convolutedVector1:** Περιέχει το σήμα που πρόκειται να μεταδοθεί από την πρώτη κεραία, μετά την ολοκλήρωση του `filtering`. Το `vector convolutedVector1` έχει διαστάσεις (1 x CONVOLUTION\_SIZE). Στην περίπτωση του σήματος της δεύτερης κεραίας, το `vector` που εξάγεται από την μέθοδο **filterConvolution** είναι το **convolutedVector2**.

### 7.1.8 Βήμα 9: Μετάδοση σήματος

Στο βήμα αυτό εκτελείται η ασύρματη μετάδοση του σήματος, μέσω των δύο συσκευών USRP που αποτελούν τον MIMO πομπό. Για την σωστή πραγματοποίηση της μετάδοσης είναι απαραίτητος ο ορισμός των τιμών των παρακάτω παραμέτρων, για τις συσκευές USRP-N210 του πομπού (με αστερίσκο σημειώνονται οι μεταβλητές των οποίων οι τιμές καταχωρούνται από τον χρήστη μέσω του `transmitter GUI`):

- 1) **string args\***: `string` που περιέχει τις διευθύνσεις IP των συσκευών USRP, οι οποίες δίνονται από τον χρήστη μέσω του `transmitter GUI`. Η σειρά με την οποία καταχωρούνται οι διευθύνσεις IP των συσκευών, καθορίζει και τον ρόλο που έχουν κατά τη διάρκεια της μετάδοσης. Αυτή που καταχωρείται πρώτη, αποτελεί την `primary` συσκευή του πομπού, αναλαμβάνοντας των συγχρονισμό και των δύο. Στην συγκεκριμένη περίπτωση, ορίζεται ως `primary` η συσκευή με διεύθυνση IP = 192.168.30.3 και ως `secondary` η συσκευή με διεύθυνση IP: 192.168.30.4. Η τελική τιμή της μεταβλητής `args` είναι:

**args= "addr0=192.168.30.3, addr1=192.168.30.4"**

- 2) **string type\***: καθορίζει το πρότυπο των δειγμάτων του σήματος βασικής ζώνης στον host υπολογιστή. Στην συγκεκριμένη περίπτωση, τα δείγματα του σήματος βασικής ζώνης είναι τύπου complex-float 32, δηλαδή μιγαδικοί δεκαδικοί αριθμοί, με πραγματικό και φανταστικό μέρος μεγέθους 32 bits μνήμης (συνολικά 64 bits). Όσο μεγαλύτερο είναι το μέγεθος των bits μνήμης που καταλαμβάνει το δείγμα, τόσο μεγαλύτερη είναι η ακρίβεια που επιτυγχάνεται, εις βάρος της ταχύτητας επεξεργασίας των δεδομένων από τον υπολογιστή- host.
- 3) **string wirefmt**: αποτελεί τον τύπο των δειγμάτων, κατά την μετάδοση μεταξύ των συσκευών USRP (data link layer). Στην συγκεκριμένη περίπτωση, τα δείγματα είναι complex-short 16. Μπορεί να χρησιμοποιηθούν δείγματα τύπου complex-short 8, κάτι που αυξάνει την ταχύτητα μετάδοσης, αλλά μειώνει την ακρίβεια των υπολογισμών.
- 4) **string file1, file2\***: strings που περιέχουν τα file paths των binary αρχείων που περιέχουν το προς μετάδοση σήμα της πρώτης και δεύτερης συσκευής, αντίστοιχα. Μετά το τέλος του βήματος 8 (Convolution Filtering), τα προς μετάδοση δείγματα και των δύο συσκευών, αποθηκεύονται σε δύο διαφορετικά binary αρχεία, κωδικοποιημένα σε αριθμητικές τιμές τύπου complex-float 32. Η μέθοδος template<typename **samp\_type**> void **send\_from\_file** της βιβλιοθήκης UHD, εκτελεί την ανάγνωση των δεδομένων από τα συγκεκριμένα αρχεία και την ασύρματη μετάδοσή τους στον δέκτη.
- 5) **double rate\***: καθορίζει τον ρυθμό μετάδοσης των δειγμάτων από τον πομπό.
- 6) **double freq\***: καθορίζει την κεντρική συχνότητα μετάδοσης των δειγμάτων από τον πομπό.
- 7) **double gain\***: καθορίζει το κέρδος ενίσχυσης μετάδοσης του πομπού.
- 8) **string synch\***: καθορίζει την μέθοδο συγχρονισμού μεταξύ των συσκευών USRP που αποτελούν τον πομπό. Για την μετάδοση MIMO, η τιμή της μεταβλητής MIMO είναι:

**sync="mimo"**

που σημαίνει ότι χρησιμοποιείται mimo cable για τον συγχρονισμό τους.

- 9) **string channel\_list**: καθορίζει τα USRP channels, που χρησιμοποιούνται κατά την μετάδοση των δειγμάτων. Για την λειτουργία ενός πομπού MIMO, ισχύει:

**channel\_list="0,1"**

- 10) **size\_t spb**: Καθορίζει το μέγεθος του buffer εκπομπής των συσκευών USRP του πομπού. Με δεδομένο ότι τα USRP επεξεργάζονται και στέλνουν πλαίσια πληροφορίας συγκεκριμένου μεγέθους, το μέγεθος αυτό ορίζεται από την μεταβλητή αυτή.

## 7.2 Ανάλυση Κώδικα Δέκτη

Ακολουθεί η βηματική ανάλυση των μεθόδων που χρησιμοποιήθηκαν για την υλοποίηση του δέκτη (Κεφάλαιο 3 – σχήμα 2). Όπως και στον πομπό τα δεδομένα εισόδου που χρησιμοποιούνται στα παραδείγματα του δέκτη είναι τα εξής:

Αριθμός OFDM Frames\* = 1 frame

Αριθμός υποκαναλιών OFDM\* = 4 υποκανάλια

Μέγεθος Κυκλικού Προθέματος\* = 1 δείγμα

Αριθμός συμβόλων QPSK σε κάθε υποκανάλι OFDM\* = 8

Μέγεθος λαμβανόμενου σήματος για κάθε κεραία = 130 samples

(Με αστερίσκο σημειώνονται οι μεταβλητές των οποίων οι τιμές καταχωρούνται από τον χρήστη μέσω του GUI της εφαρμογής. Θα πρέπει να αναφερθεί πως για τις μετρήσεις χρησιμοποιήθηκαν διαφορετικές παράμετροι, ίδιες όμως με τις παραμέτρους που χρησιμοποιήθηκαν στον πομπό και οι οποίες αναφέρονται στο κεφάλαιο 8).

Στην περίπτωση αυτή θεωρούμε πως το μέγεθος του λαμβανόμενου σήματος είναι ακριβώς ίσο με το μέγεθος του σήματος που μεταδίδεται από τον πομπό. Στην πραγματικότητα, για την ορθότερη λειτουργία των συσκευών USRP, το λαμβανόμενο σήμα πρέπει να οριστεί αρκετά μεγαλύτερο.

### 7.2.1 Βήμα 1: Λήψη Σήματος

Οι τιμές των μεταβλητών που πρέπει να οριστούν για την σωστή λειτουργία του πομπού είναι οι παρακάτω (με αστερίσκο σημειώνονται οι μεταβλητές των οποίων οι τιμές αποδίδονται από τον χρήστη μέσω του receiver GUI):

**1) string args\***: string που περιέχει τις διευθύνσεις IP των συσκευών USRP. Η σειρά με την οποία καταχωρούνται οι διευθύνσεις IP των συσκευών, καθορίζει και τον ρόλο που έχουν κατά τη διάρκεια της μετάδοσης. Αυτή που καταχωρείται πρώτη, αποτελεί την primary συσκευή του πομπού, αναλαμβάνοντας των συγχρονισμό και των δύο. Στην συγκεκριμένη περίπτωση, ορίζεται ως primary, η συσκευή με διεύθυνση IP = 192.168.40.3 και ως secondary η συσκευή με διεύθυνση IP: 192.168.40.4. Η τελική τιμή της μεταβλητής args είναι:

**args= "addr0=192.168.40.3, addr1=192.168.40.4"**

**2) string type\***: καθορίζει τον τύπο των δειγμάτων του σήματος στον δέκτη. Όπως συμβαίνει και στον πομπό, τα δείγματα του είναι complex-float 32, δηλαδή μιγαδικοί δεκαδικοί αριθμοί, με πραγματικό και φανταστικό μέρος μεγέθους 32 bits μνήμης (συνολικά 64 bits).

**3) string file\***: string που περιέχει τα file paths των binary αρχείων που περιέχουν το λαμβανόμενο σήμα της πρώτης και δεύτερης κεραίας του δέκτη. . Μετά το τέλος της λήψης ,

τα δείγματα που έχουν ληφθεί από τις δύο κεραίες, αποθηκεύονται σε δύο διαφορετικά binary αρχεία, κωδικοποιημένα σε αριθμητικές τιμές τύπου complex-float 32.

**4) double rate\***: καθορίζει τον ρυθμό λήψης των δειγμάτων από τον δέκτη. Η τιμή της μεταβλητής rate πρέπει να είναι ίδια τόσο στον πομπό, όσο και στον δέκτη.

**5) double freq\***: καθορίζει την κεντρική συχνότητα λήψης των δειγμάτων από τον δέκτη. Η τιμή της μεταβλητής freq πρέπει να είναι ίδια σε πομπό και δέκτη.

**6) double gain\***: καθορίζει το κέρδος ενίσχυσης του δέκτη.

**7) string sync**: καθορίζει την μέθοδο συγχρονισμού των συσκευών USRP που αποτελούν τον δέκτη. Η μεταβλητή sync πρέπει να έχει την τιμή "mimo" και στον δέκτη, για διασύνδεση των συσκευών USRP με mimo cable.

**8) string channel\_list**: καθορίζει τα USRP channels, που χρησιμοποιούνται κατά την λήψη των δειγμάτων. Η μεταβλητή channel\_list πρέπει να έχει την τιμή "0,1" και στον δέκτη. Αν η λήψη γίνεται μόνο από το primary, η τιμή είναι "0". Αν η λήψη γίνεται μόνο από τον secondary δέκτη, η τιμή είναι "1". Στην περίπτωση που λαμβάνουν και οι δυο ταυτόχρονα, όπως στο 2x2 Alamouti σύστημα που υλοποιήθηκε η τιμή είναι "0,1".

**9) double seconds\_in\_future**: καθορίζει το χρονικό διάστημα καθυστέρησης της έναρξης λήψης σήματος από τον δέκτη, , ώστε να ομαλοποιηθούν τα αρχικά φαινόμενα αστάθειας για τον AGC και τους μίκτες.

**10) size\_t total\_num\_samps**: καθορίζει τον συνολικό αριθμό των δειγμάτων που πρέπει να έχει λάβει ο δέκτης. Όταν ολοκληρωθεί η λήψη αυτών των δειγμάτων, σταματά πλήρως η λειτουργία του δέκτη.

Τα λαμβανόμενα δείγματα κάθε κεραίας, αποθηκεύονται σε binary files. Η μέθοδος **vector < complex<double>> binaryToDoubleConverter(string filePath)** μετατρέπει τα δείγματα σε αριθμητικές τιμές complex <double> και τα αποθηκεύει στο **vector<complex<double>> receivedSignal1**. Το λαμβανόμενο σήμα της πρώτης κεραίας αποθηκεύεται στο **vector<complex<double>> receivedSignal2**.

Η μορφή του λαμβανόμενου σήματος της πρώτης κεραίας είναι αυτή του Σχήματος 7.11 (σε παρενθέσεις η τιμή του vector index:

(0)	(1)	(2)	(3)	(4)	(5)			(128)	(129)
R <sub>1</sub>	R <sub>2</sub>	R <sub>3</sub>	R <sub>4</sub>	R <sub>5</sub>	R <sub>6</sub>	...	...	R <sub>129</sub>	R <sub>130</sub>

**Σχήμα 7.11: Λαμβανόμενο σήμα 1<sup>ης</sup> κεραίας δέκτη (vector receivedSignal1)**

## 7.2.2 Βήμα 2: Συνελικτικό Φιλτράρισμα - Convolution Filtering (Square Root Raised Cosine Filtering)

Το Συνελικτικό Φιλτράρισμα στο λαμβανόμενο σήμα εκτελείται από την μέθοδο `vector < complex < double>> filterConvolution(int filterSize, int samplesNumber, int CONVOLUTION_SIZE, vector <complex<double>> samplesVector, vector <double> filterCoefficientsVector)`.

### Δεδομένα εισόδου:

- 1) **int filterSize:** αριθμός των συντελεστών του Square Root Raised Cosine Φίλτρου (μέγεθος του vector `filterCoefficientsVector`, με μέγεθος `CONVOLUTION_FILTER_SIZE`).
- 2) **int samplesNumber:** μέγεθος του λαμβανόμενου σήματος πριν την εκτέλεση του Συνελικτικού Φιλτραρίσματος. (Μέγεθος των vectors `receivedSignalVector1` και `receivedSignalVector2` του βήματος 1).
- 3) **int CONVOLUTION\_SIZE:** μέγεθος του προς μετάδοση σήματος μετά την ολοκλήρωση του Συνελικτικού Φιλτραρίσματος. Αν  $M$  είναι το μέγεθος του σήματος πριν το φιλτράρισμα και  $N$  το μέγεθος του φίλτρου, τότε ισχύει:

$$\text{CONVOLUTION\_SIZE} = M + N - 1 = \text{RECEIVED\_SIGNAL\_SIZE} + \text{CONVOLUTION\_FILTER\_SIZE} - 1 \quad \text{δείγματα} \quad (1)$$

- 4) **vector < complex <double> > samplesVector:** περιέχει το λαμβανόμενο σήμα πριν την εκτέλεση του Συνελικτικού Φιλτραρίσματος (για το σήμα της πρώτης κεραίας εισάγεται το vector `receivedSignalVector1`, ενώ για αυτό της δεύτερης, το vector `receivedSignalVector2` του βήματος 1).
- 5) **vector < double > filterCoefficientsVector:** περιέχει τις τιμές του Square Root Raised Cosine φίλτρου, το οποίο χρησιμοποιείται για την εκτέλεση του convolution filtering.

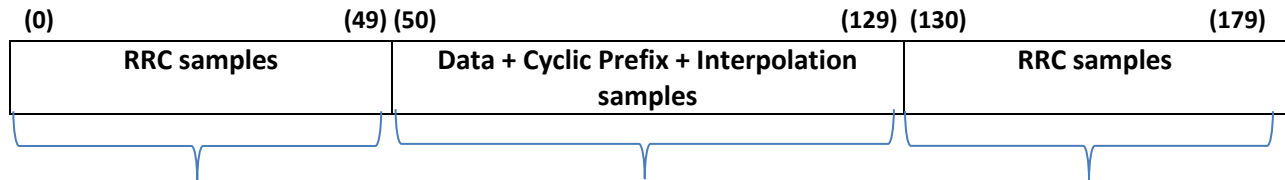
### Αλγόριθμος

Ο αλγόριθμος που εκτελείται είναι ο ίδιος με αυτόν που εκτελείται στο βήμα 8 του πομπού. Οι τιμές του φίλτρου είναι οι ίδιες με αυτές που χρησιμοποιήθηκαν για την εκτέλεση της αντίστοιχης διαδικασίας και στον πομπό. . Με αυτόν τον τρόπο και με δεδομένο ότι τα Root Raised Cosine φίλτρα έχουν τις ιδιότητες των Nyquist φίλτρων μηδενικής διασυμβολικής παρεμβολής, πομπός και δέκτης λειτουργούν με ταιριασμένα φίλτρα (matched filters). Μετά την ολοκλήρωση του συγκεκριμένου βήματος το λαμβανόμενο σήμα της πρώτης κεραίας έχει μέγεθος:  $(1 \times \text{CONVOLUTION\_SIZE})$ . Από τον τύπο (1) για την συγκεκριμένη περίπτωση ισχύει:



$CONVOLUTION\_SIZE=RECEIVED\_SIGNAL\_SIZE + CONVOLUTION\_FILTER\_SIZE-1= 130 +51 -1 =180$  samples

Η μορφή του σήματος φαίνεται στο Σχήμα 7.12 που ακολουθεί:



Σχήμα 7.12: Μορφή του λαμβανόμενου σήματος (1<sup>η</sup> κεραία) μετά το Convolution Filtering (vector convolutedVector1)

#### Δεδομένα Εξόδου:

**vector < complex < double >> convolutedVector1:** Περιέχει το σήμα που έχει ληφθεί από την πρώτη κεραία, μετά την ολοκλήρωση του filtering. Το vector convolutedVector1 έχει διαστάσεις (1 x CONVOLUTION\_SIZE). Στην περίπτωση του σήματος της δεύτερης κεραίας, το vector που εξάγεται από την μέθοδο **filterConvolution** είναι το **convolutedVector2**.

### 7.2.3 Βήμα 3: Decimation

Το decimation στον δέκτη, αποτελεί ακριβώς την αντίστροφη διαδικασία από αυτήν του interpolation στον πομπό (βήμα 7 του πομπού). Αποτέλεσμα της, είναι η αφαίρεση των επιπρόσθετων πλεοναζόντων δειγμάτων που προστέθηκαν στον πομπό πριν την τελική αποστολή του σήματος, με σκοπό τη μείωση της διασυμβολικής παρεμβολής. Το decimation, για το σήμα της πρώτης κεραίας, εκτελείται από την μέθοδο **vector <complex<double>> receiverZerosRemover (int DOWNSIZED\_SIGNAL\_SIZE, int CONVOLUTION\_SIZE, vector < complex < double > convolutedVector1)**.

#### Δεδομένα εισόδου:

**1) int DOWNSIZED\_SIGNAL\_SIZE:** Μέγεθος του λαμβανόμενου σήματος, μετά την εκτέλεση του decimation. Ισχύει:

$$DOWNSIZED\_SIGNAL\_SIZE= CONVOLUTION\_SIZE/2$$

δηλαδή, το μέγεθος του σήματος μετά το decimation μειώνεται στο μισό.

**2) int CONVOLUTION\_SIZE:** μέγεθος του σήματος αμέσως μετά την εκτέλεση του convolution filtering.

3) **vector <complex <double> convolutedVector1**: vector που περιέχει το σήμα της πρώτης κεραίας μετά την εκτέλεση του convolution filtering.

Το σήμα που έχει ληφθεί από την πρώτη κεραία, αμέσως μετά την εκτέλεση του convolution filtering, έχει την μορφή του Σχήματος 7.13 (με κόκκινο χρώμα εμφανίζονται τα δείγματα dummy που αφαιρούνται):

(0)	(1)	(2)	(3)	(4)	(5)	(6)			(179)
C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>	C <sub>5</sub>	C <sub>6</sub>	C <sub>7</sub>	...	...	C <sub>180</sub>

**Σχήμα 7.13 – Λαμβανόμενο σήμα (1<sup>η</sup> κεραία) μετά το RRC filtering (vector convolutedVector1)**

Μετά την εκτέλεση του decimation, το σήμα της πρώτης κεραίας έχει την μορφή του Σχήματος 7.14:

(0)	(1)	(2)	(3)	(4)	(5)	(6)			(89)
C <sub>1</sub>	C <sub>3</sub>	C <sub>5</sub>	C <sub>7</sub>	C <sub>9</sub>	C <sub>11</sub>	C <sub>13</sub>	...	...	C <sub>90</sub>

**Σχήμα 7.14 - Λαμβανόμενο σήμα (1<sup>η</sup> κεραία) μετά το decimation (vector decimatedSignal1)**

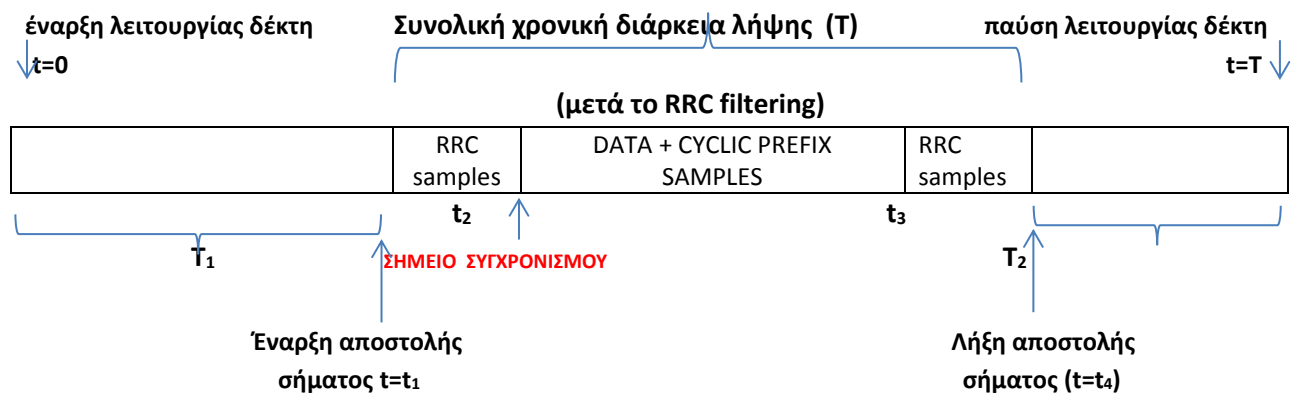
Στην συγκεκριμένη περίπτωση το μέγεθος του σήματος της πρώτης κεραίας πριν το decimation είναι 180 δείγματα. Μετά την αφαίρεση των dummy δειγμάτων, το μέγεθος του σήματος μειώνεται σε 90 δείγματα.

**Δεδομένα Εξόδου:**

**vector < complex < double > > decimatedSignal1**: Περιέχει το σήμα που λαμβάνεται από την πρώτη κεραία, μετά την εκτέλεση του decimation. Έχει διαστάσεις (1x DOWNSIZED\_SIGNAL\_SIZE). Στην περίπτωση του σήματος της δεύτερης κεραίας, από την μέθοδο receiverZerosRemover επιστρέφεται το **vector <complex<double> > decimatedSignal2**, το οποίο είναι και αυτό διαστάσεων (1x DOWNSIZED\_SIGNAL\_SIZE).

## 7.2.4 Βήμα 4: Συγχρονισμός Πομπού – Δέκτη

Όπως είδαμε στο βήμα 1, το χρονικό διάστημα λειτουργίας του δέκτη είναι αρκετά μεγαλύτερο από το χρονικό διάστημα που απαιτείται για την αποστολή του σήματος από τον πομπό του συστήματος, όπως φαίνεται παρακάτω:



Σχήμα 7.15 - Χρονική επικάλυψη αποστολής και λήψης σήματος

Όπως φαίνεται στο Σχήμα 7.15, η λειτουργία του δέκτη αρχίζει τη χρονική στιγμή  $t=0$  και διαρκεί έως τη χρονική στιγμή  $t=T$ . Η λειτουργία του πομπού, δηλαδή η αποστολή σήματος, αρχίζει τη χρονική στιγμή  $t_1$  και διαρκεί έως τη χρονική στιγμή  $t_4$ . Κατά τα «νεκρά», για τον πομπό, χρονικά διαστήματα  $T_1$  και  $T_2$ , ο δέκτης εξακολουθεί να λαμβάνει και να αποθηκεύει το σήμα, το οποίο φυσικά δεν προέρχεται από τον πομπό του συστήματος (μπορεί να προέρχεται από γειτονικές κεραιές, παρεμβολές, θόρυβο κτλ.). Επιπλέον, όπως είδαμε και στο βήμα 2, στην αρχή και το τέλος του «ωφέλιμου σήματος», δηλαδή αυτού που προέρχεται από τον πομπό του συστήματος, μεταξύ των χρονικών στιγμών  $t_1$  και  $t_4$ , υπάρχουν τα δείγματα που έχουν δημιουργηθεί κατά την διαδικασία του Root Raised Cosine Filtering και τα οποία φυσικά δεν περιέχουν δεδομένα (χρονικά διαστήματα  $t_1-t_2$  και  $t_3-t_4$ ), τα λεγόμενα tail samples. Η διαδικασία του συγχρονισμού επιτρέπει τον ακριβή υπολογισμό του σημείου όπου αρχίζει η λήψη των δειγμάτων δεδομένων και κυκλικού προθέματος από τον πομπό. Με τον τρόπο αυτό όλα τα υπόλοιπα δείγματα που έχουν ληφθεί απορρίπτονται από την επεξεργασία του σήματος που ακολουθεί στα επόμενα βήματα.

Ο αλγόριθμος συγχρονισμού αποτελείται από τα εξής βήματα (εξετάζεται η περίπτωση του σήματος που λαμβάνεται από την πρώτη κεραία):



$$\text{SLIDING\_WINDOWS\_NUMBER}=90 - (4+1)+1=90-5+1=86$$

- 3) Τα δείγματα που περιέχει κάθε sliding window πολλαπλασιάζονται σημείο προς σημείο με τα περιεχόμενα του vector  $\text{IFFTPreambleVector1}^*$ . Έπειτα υπολογίζεται η απόλυτη τιμή του αθροίσματος των επι μέρους γινομένων για κάθε sliding window:

$$\begin{array}{l}
 \mathbf{t=0} \\
 \begin{bmatrix} C_1 & * & S_1^* \\ C_3 & * & S_9^* \\ C_5 & * & S_{17}^* \\ C_7 & * & S_{25}^* \\ C_9 & * & S_1^* \end{bmatrix} \longrightarrow \text{abs}_1(\mathbf{N}_0)_1 = \text{abs} |C_1 * S_1^* + C_3 * S_9^* + C_5 * S_{17}^* + C_7 * S_{25}^* + C_9 * S_1^*| \\
 \\
 \mathbf{t=1} \\
 \begin{bmatrix} C_3 & * & S_1^* \\ C_5 & * & S_9^* \\ C_7 & * & S_{17}^* \\ C_9 & * & S_{25}^* \\ C_{11} & * & S_1^* \end{bmatrix} \longrightarrow \text{abs}_1(\mathbf{N}_1)_1 = \text{abs} |C_3 * S_1^* + C_5 * S_9^* + C_7 * S_{17}^* + C_9 * S_{25}^* + C_{11} * S_1^*| \\
 \dots \\
 \dots \\
 \mathbf{t=86} \\
 \begin{bmatrix} C_{81} & * & S_1^* \\ C_{83} & * & S_9^* \\ C_{85} & * & S_{17}^* \\ C_{87} & * & S_{25}^* \\ C_{89} & * & S_1^* \end{bmatrix} \longrightarrow \text{abs}_1(\mathbf{N}_{86})_1 = |C_{86} * S_1^* + C_{83} * S_9^* + C_{85} * S_{17}^* + C_{87} * S_{25}^* + C_{89} * S_1^*|
 \end{array}$$

Σχήμα 7.18 – Υπολογισμός Σημείου Συγχρονισμού SP

Τα βήματα (1)-(3) επαναλαμβάνονται για το σήμα της δεύτερης κεραίας:

- 1) Στο 1<sup>ο</sup> βήμα εκτελείται Ταχύς Αντίστροφος Μετασχηματισμός Fourier στο preamble της δεύτερης κεραίας ( $\text{PreambleVector2}$ ), οπότε προκύπτει το vector  $\text{IFFTPreambleVector2}^*$ .
- 2) Στο 2<sup>ο</sup> βήμα ως είσοδος χρησιμοποιείται το λαμβανόμενο σήμα της 2<sup>ης</sup> κεραίας (vector  $\text{decimatedSignalVector2}$ ).
- 3) Υπολογίζεται η απόλυτη τιμή του αθροίσματος των επί μέρους γινομένων για κάθε sliding window του βήματος 2.

- 4) Το σημείο Συγχρονισμού SP είναι αυτό για το οποίο ισχύει:

$$\text{abs}(\mathbf{N}_k)_1 + \text{abs}(\mathbf{N}_k)_2 = \max(\text{abs}_1(\mathbf{N}_i)_1 + \text{abs}_2(\mathbf{N}_i)_2), \text{ με } 0 \leq i \leq \text{SLIDING\_WINDOWS\_NUMBER}$$

δηλαδή το σημείο στο οποίο μεγιστοποιείται το άθροισμα των απολύτων τιμών των επί μέρους τιμών που έχουν υπολογιστεί.

Στην συγκεκριμένη περίπτωση σημείο συγχρονισμού πομπού-δέκτη είναι το 26ο σύμβολο (δηλαδή το στοιχείο με vector index=25).

Ο συγχρονισμός πομπού-δέκτη πραγματοποιείται από την μέθοδο **int Receiver (int DOWNSIZED\_SIGNAL\_SIZE, int PREAMBLE\_VECTOR\_SIZE, vector < complex < double > > decimatedVector1, vector < complex < double > > preambleVector)**. Για την ευκολότερη εκτέλεση των αλγεβρικών πράξεων χρησιμοποιήθηκαν μέθοδοι της βιβλιοθήκης Eigen 3.

#### Δεδομένα εισόδου:

- 1) **int DOWNSIZED\_SIGNAL\_SIZE**: μέγεθος του λαμβανόμενου σήματος, μετά την εκτέλεση του decimation.
- 2) **int PREAMBLE\_VECTOR\_SIZE**: Μέγεθος του preamble.
- 3) **vector < complex < double > > decimatedVector1**: μέγεθος του vector που περιέχει το σήμα που λαμβάνεται από την πρώτη κεραία, μετά την εκτέλεση του decimation. Στην περίπτωση του σήματος της δεύτερης κεραίας, στη μέθοδο εισάγεται το vector **decimatedVector2**.
- 4) **vector < complex < double > > preambleVector**: περιέχει τα σύμβολα του preamble.

#### Δεδομένα εξόδου:

**int maximumIndex1**: Ακέραια τιμή στην οποία αποδίδεται το σημείο (vector index), στο οποίο σημειώνεται ο συγχρονισμός πομπού-δέκτη. (Στην περίπτωση του σήματος που λαμβάνει η δεύτερη κεραία, η μέθοδος επιστρέφει την τιμή maximumIndex2). Η τιμή αυτή χρησιμοποιείται στο επόμενο βήμα (βήμα 5), για τον διαχωρισμό των δειγμάτων δεδομένων του σήματος από τα δείγματα του Κυκλικού Προθέματος και από αυτά που προστέθηκαν κατά τη διαδικασία του Convolution Filtering (βήμα 5).

### 7.2.5 Βήμα 5: Αφαίρεση Κυκλικού Προθέματος

Στο βήμα αυτό, αφαιρείται το Κυκλικό Προθέμα, το οποίο έχει προστεθεί στον πομπό πριν την μετάδοση του σήματος. Η αφαίρεση του Κυκλικού Προθέματος, στην περίπτωση του λαμβανόμενου σήματος της πρώτης κεραίας, εκτελείται από την μέθοδο **vector <complex<double>> cyclicPrefixRemover (int CYCLIC\_PREFIX\_SIZE, int OFDM\_SUBCARRIERS\_NUMBER, int maximumIndex, int VECTOR\_SIZE\_AFTER\_CYCLIC\_PREFIX\_REMOVAL, int FINAL\_INDEX\_FOR\_CYCLIC\_PREFIX\_REMOVAL, vector<complex<double>> decimatedSignalVector1)**.



$$\text{FINAL\_INDEX\_FOR\_CYCLIC\_PREFIX\_REMOVAL}=25 \times (1 \times (4+1) \times 8) = 65$$

Συνεπώς, τα τμήμα του vector decimatedSignalVector1 που περιέχει τα δείγματα δεδομένων και Κυκλικού Προθέματος, έχει μέγεθος 40 δειγμάτων (VECTOR\_SIZE\_AFTER\_CYCLIC\_PREFIX\_REMOVAL) και εκτείνεται από την θέση 25 έως και θέση 65. Από το τμήμα αυτό αφαιρούνται τα δείγματα που περιέχουν το Κυκλικό Πρόθεμα, όπως φαίνεται στο Σχήμα 7.20 (με κόκκινο χρώμα εμφανίζονται τα δείγματα του Κυκλικού Προθέματος που αφαιρούνται):

(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(17)	(18)		(39)
D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>	D <sub>8</sub>	D <sub>9</sub>	D <sub>10</sub>	D <sub>11</sub>	D <sub>12</sub>	D <sub>13</sub>	D <sub>14</sub>	D <sub>15</sub>	D <sub>16</sub>	D <sub>17</sub>	D <sub>18</sub>	...	D <sub>40</sub>

**Σχήμα 7.20 – Σύμβολα Κυκλικού Προθέματος που αφαιρούνται (σήμα 1<sup>ης</sup> κεραίας)  
(vector decimatedSignalVector1)**

Στην περίπτωση αυτή, αφαιρούνται συνολικά 8 δείγματα Κυκλικού Προθέματος συνεπώς ισχύει:

$$\text{VECTOR\_SIZE\_AFTER\_CYCLIC\_PREFIX\_REMOVAL}=40-8=32 \text{ δείγματα πληροφορίας}$$

**Δεδομένα Εξόδου:**

**vector <complex<double>> dataNoCyclicPrefixVector1:** Για το σήμα που λαμβάνει η πρώτη κεραία μετά την αφαίρεση του Κυκλικού Προθέματος, η μέθοδος επιστρέφει το vector <complex <double>> dataNoCyclicPrefixVector1, το οποίο έχει διαστάσεις (1x VECTOR\_SIZE\_AFTER\_CYCLIC\_PREFIX\_REMOVAL). Στην περίπτωση της δεύτερης κεραίας, επιστρέφεται το vector <complex <double>> **dataNoCyclicPrefixVector2.**

**7.2.6 Βήμα 6: Μετατροπή σειριακού σήματος σε παράλληλο**

Στο βήμα αυτό γίνεται η μετατροπή του σήματος από σειριακό σε παράλληλο, για να εκτελεστεί, στη συνέχεια, ο Ταχύς Μετασχηματισμός Fourier-FFT (βήμα 7). Για την πραγματοποίηση της μετατροπής, δεν χρησιμοποιείται κάποια βοηθητική μέθοδος, όπως στα προηγούμενα βήματα. Η σειριακή μορφή του σήματος είναι αυτή που φαίνεται στο Σχήμα 7.21:

(0)	(1)	(2)	(3)	(4)	(5)	(6)			(31)
D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>	...	...	D <sub>32</sub>

**Σχήμα 7.21 – Σειριακή μορφή σήματος (1<sup>η</sup> κεραία)  
(vector dataNoCyclicPrefixVector1)**



Το παράλληλο σήμα που δημιουργείται, μπορεί να θεωρηθεί ως πίνακας διαστάσεων  $M \times N$ , όπου:

$$M = \text{OFDM\_SUBCARRIERS\_NUMBER} \text{ και } N = \text{QPSK\_SYMBOLS\_PER\_SUBCARRIER\_NUMBER}.$$

Στην περίπτωση που εξετάζουμε, ισχύει:

$\text{OFDM\_SUBCARRIERS\_NUMBER} = 4$  υποκανάλια OFDM και

$\text{QPSK\_SYMBOLS\_PER\_SUBCARRIER\_NUMBER} = 8$  σύμβολα

Συνεπώς, το παράλληλο σήμα, τόσο της πρώτης και της δεύτερης κεραίας, μπορεί να θεωρηθεί ως ένας πίνακας διαστάσεων  $4 \times 8$ , όπως φαίνεται στο Σχήμα 7.22:

(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
$D_1$	$D_5$	$D_9$	$D_{13}$	$D_{17}$	$D_{21}$	$D_{25}$	$D_{29}$
$D_2$	$D_6$	$D_{10}$	$D_{14}$	$D_{18}$	$D_{22}$	$D_{26}$	$D_{30}$
$D_3$	$D_7$	$D_{11}$	$D_{15}$	$D_{19}$	$D_{23}$	$D_{27}$	$D_{31}$
$D_4$	$D_8$	$D_{12}$	$D_{16}$	$D_{20}$	$D_{24}$	$D_{28}$	$D_{32}$

**Σχήμα 7.22 - (parallelVector1)**

Το παράλληλο σήμα των δύο κεραίων, αποθηκεύεται στα vectors **parallelVector1** και **parallelVector2** αντίστοιχα, τα οποία είναι διαστάσεων ( $\text{OFDM\_SUBCARRIERS\_NUMBER} \times \text{QPSK\_SYMBOLS\_PER\_SUBCARRIER\_NUMBER}$ ).

### 7.2.7 Βήμα 7: Fast Fourier Transform (Ταχύς Μετασχηματισμός Fourier)

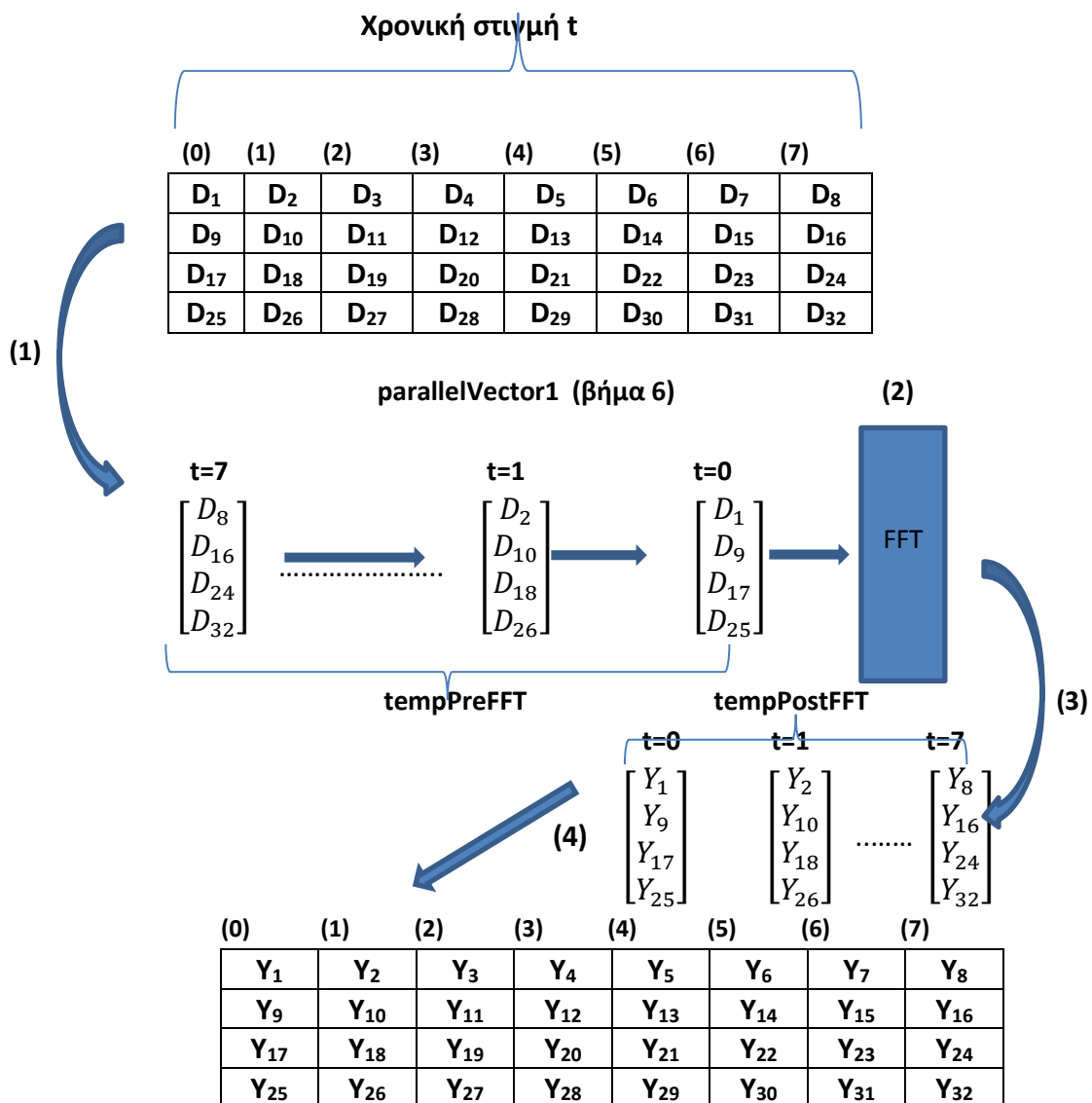
Στο βήμα αυτό, εκτελείται ο Ταχύς Μετασχηματισμός Fourier, κάτι που έχει ως αποτέλεσμα την μετάβαση του σήματος από το πεδίο του χρόνου στο πεδίο της συχνότητας. Η μετατροπή πραγματοποιείται από την μέθοδο `vector < vector < complex < double >>> FFT (int OFDM_SUBCARRIERS_NUMBER, int QPSK_SYMBOLS_PER_SUBCARRIER_NUMBER, vector< vector < complex < double >>> vec)`. Οι αλγεβρικές πράξεις του μετασχηματισμού FFT εκτελούνται με κλήση των μεθόδων της βιβλιοθήκης FFTW-3.

#### Δεδομένα εισόδου

**1) int OFDM\_SUBCARRIERS\_NUMBER:** αριθμός των υποκαναλιών OFDM που έχει το λαμβανόμενο OFDM frame.

- 2) `int QPSK_SYMBOLS_PER_SUBCARRIER_NUMBER`: αριθμός των συμβόλων QPSK (data) που περιέχονται σε κάθε υποκανάλι OFDM.
- 3) `vector< vector<complex<double >> >> vec`: vector που περιέχει το σήμα μετά την μετατροπή του από σειριακή σε παράλληλη μορφή. Για το σήμα της πρώτης κεραίας εισάγεται το `vector parallelVector1`, ενώ για το σήμα της δεύτερης κεραίας η μέθοδος FFT καλείται δεύτερη φορά, όπου και εισάγεται το `vector parallelVector2`.

Στο Σχήμα 7.23 παρουσιάζεται η εκτέλεση του Ταχέως Μετασχηματισμού Fourier για το σήμα της πρώτης κεραίας (`vector parallelVector1`):



Σχήμα 7.23 – Σύμβολα 1<sup>ης</sup> κεραίας μετά τον μετασχηματισμό IFFT - (`vector FFTVector1`)

- 1) Τα δείγματα κάθε στήλης του vector parallelvector1 αντιγράφονται στον πίνακα (array) δυναμικής ανάθεσης μνήμης tempPreFFT της βιβλιοθήκης FFTW, διαστάσεων (1x OFDM\_SUBCARRIERS\_NUMBER).
- 2) Στα περιεχόμενα του πίνακα tempPreFFT, εκτελείται Ταχύς μετασχηματισμός Fourier σε διαδοχικές χρονικές περιόδους  $t = 0, 1, \dots, (\text{QPSK\_SYMBOLS\_PER\_SUBCARRIER\_NUMBER} - 1)$ .
- 3) Τα αποτελέσματα του μετασχηματισμού, δηλαδή τα σύμβολα που έχουν ληφθεί από την πρώτη κεραία, αποθηκεύονται στον πίνακα (array) δυναμικής ανάθεσης μνήμης tempPostFFT της βιβλιοθήκης FFTW, διαστάσεων (1x OFDM\_SUBCARRIERS\_NUMBER).
- 4) Τα σύμβολα του πίνακα tempPostFFT, αποθηκεύονται στο vector `< vector < complex < double > > >` FFTVector1, διαστάσεων (OFDM\_SUBCARRIERS\_NUMBER x QPSK\_SYMBOLS\_PER\_SUBCARRIER\_NUMBER).

### Δεδομένα Εξόδου

`vector < vector < complex < double > > >` **FFTVector1**: Η μέθοδος IFFT, για το σήμα της πρώτης κεραίας επιστρέφει το vector FFTVector1, το οποίο περιέχει τα σύμβολα που προκύπτουν, μετά την ολοκλήρωση του Ταχέως Μετασχηματισμού Fourier. Στην περίπτωση του σήματος της δεύτερης κεραίας, από την μέθοδο επιστρέφεται το vector **FFTVector2**. Τα δύο vectors είναι διαστάσεων (OFDM\_SUBCARRIERS\_NUMBER x QPSK\_SYMBOLS\_PER\_SUBCARRIER\_NUMBER). Στην συγκεκριμένη περίπτωση οι διαστάσεις τους είναι (4x8).

### 7.2.8 Βήμα 8 - 9: Εκτίμηση καναλιού – Αποκωδικοποίηση Alamouti

Κατά την αποκωδικοποίηση Alamouti, για την εκτίμηση των λαμβανόμενων συμβόλων είναι απαραίτητη η γνώση του πίνακα  $H^+$  (Moore-Penrose ψευδοαντίστροφος πίνακας του πίνακα των channel coefficients  $H$ ) για τον οποίο ισχύει:

$$H^+ = (H^H H)^{-1} H^H \quad (k) \quad \text{όπου } H = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \\ h_{12}^* & -h_{11}^* \\ h_{22}^* & -h_{21}^* \end{bmatrix}$$

Αρχικά, λοιπόν, απαιτείται ο υπολογισμός των τιμών των channel coefficients  $h_{11}, h_{12}, h_{21}, h_{22}$  για κάθε OFDM frame, αλλά και για κάθε υποκανάλι ενός OFDM frame. Η διαδικασία αυτή ονομάζεται εκτίμηση καναλιού (channel estimation).

Τα προς μετάδοση σύμβολα του πρώτου υποκαναλιού OFDM της πρώτης και δεύτερης κεραίας, μετά την κωδικοποίηση Alamouti, είναι τα εξής:

	$t_1=0$	$t_2=1$	$t_3=2$	$t_4=3$	$t_5=4$	$t_6=5$	$t_7=6$	$t_8=7$
1 <sup>η</sup> κεραία	$X_1$	$-X_2^*$	$X_3$	$-X_4^*$	$X_5$	$-X_6^*$	$X_7$	$-X_8^*$
2 <sup>η</sup> κεραία	$X_2$	$X_1^*$	$X_4$	$X_3^*$	$X_6$	$X_5^*$	$X_8$	$X_7^*$

Σχήμα 7.24- Σύμβολα 1<sup>ου</sup> υποκαναλιού OFDM (1<sup>η</sup> – 2<sup>η</sup> κεραία)

Οι τιμές των συμβόλων  $X_1$  και  $X_2$  είναι γνωστές, διότι εισάγονται από το vector **preambleVector**. Τα σύμβολα  $X_1$  και  $X_2$ , και γενικότερα τα δύο πρώτα σύμβολα κάθε υποκαναλιού OFDM, είναι αυτά που χρησιμοποιούνται για τον υπολογισμό των αντίστοιχων channel coefficients.

Για το σύμβολο που λαμβάνει η 1<sup>η</sup> κεραία την χρονική στιγμή  $t_1=0$  ισχύει:

$$y_1^1 = [h_{11} \quad h_{12}] \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \quad (1)$$

ενώ για το σύμβολο που λαμβάνει την χρονική στιγμή  $t_2=1$ :

$$y_1^2 = [h_{11} \quad h_{12}] \begin{bmatrix} -X_2^* \\ X_1^* \end{bmatrix} \quad (2)$$

Από τις εξισώσεις (1) και (2) προκύπτει:

$$\begin{bmatrix} Y_1^1 \\ Y_1^2 \end{bmatrix} = \begin{bmatrix} X_1 & X_2 \\ -X_2^* & X_1^* \end{bmatrix} \rightarrow \begin{bmatrix} h_{11} \\ h_{12} \end{bmatrix} = \begin{bmatrix} X_1 & X_2 \\ -X_2^* & X_1^* \end{bmatrix}^{-1} \begin{bmatrix} y_1^1 \\ y_1^2 \end{bmatrix} \quad (3)$$

Ομοίως, για το σύμβολο που λαμβάνει η 2<sup>η</sup> κεραία την χρονική στιγμή  $t_1=0$  ισχύει:

$$y_2^1 = [h_{21} \quad h_{22}] \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \quad (4)$$

Για το σύμβολο που λαμβάνει η 2<sup>η</sup> κεραία την χρονική στιγμή  $t_2=1$  ισχύει:

$$y_2^2 = [h_{21} \quad h_{22}] \begin{bmatrix} -X_2^* \\ X_1^* \end{bmatrix} \quad (5)$$

Από τις εξισώσεις (3) και (4) προκύπτει:

$$\begin{bmatrix} Y_2^1 \\ Y_2^2 \end{bmatrix} = \begin{bmatrix} X_1 & X_2 \\ -X_2^* & X_1^* \end{bmatrix} \rightarrow \begin{bmatrix} h_{21} \\ h_{22} \end{bmatrix} = \begin{bmatrix} X_1 & X_2 \\ -X_2^* & X_1^* \end{bmatrix}^{-1} \begin{bmatrix} y_2^1 \\ y_2^2 \end{bmatrix} \quad (6)$$

Με τον τρόπο αυτό, υπολογίζονται οι τιμές του πίνακα  $H = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \\ h_{12}^* & -h_{11}^* \\ h_{22}^* & -h_{21}^* \end{bmatrix}$ .

Ο πίνακας  $H$  δεν είναι τετραγωνικός, για αυτό πρέπει να υπολογιστεί, ο πίνακας  $H^+ = (H^H H)^{-1} H^H$ , όπου  $H^H$  είναι ο ανάστροφος συζυγής πίνακας του  $H$ . Πλέον, η εκτίμηση των όλων των λαμβανόμενων συμβόλων του υποκαναλιού OFDM είναι δυνατή με την εξίσωση 7:

$$\begin{bmatrix} \widehat{x}_1 \\ \widehat{x}_2 \end{bmatrix} = H^+ \begin{bmatrix} y_1^1 \\ y_2^1 \\ y_1^{2*} \\ y_2^{2*} \end{bmatrix} \quad (7)$$

Η εκτίμηση του καναλιού και η αποκωδικοποίηση Alamouti εκτελούνται από την μέθοδο `vector <vector<complex<double>>> channelEstimator_AlamoutiDecoder (int OFDM_SUBCARRIERS_NUMBER, int QPSK_SYMBOLS_PER_SUBCARRIER_NUMBER, vector <vector<complex<double>> transmitterSymbolVector1, vector <vector<complex<double>>> transmitterSymbolVector2, vector <vector<complex <double >>> FFTVector1, vector<vector<complex <double>>> FFTVector2)`. (Για την εκτέλεση των μεθόδων `channelEstimator_AlamoutiDecoder` και `moorePenroseMatrixCalculator` χρησιμοποιείται η βιβλιοθήκη Eigen, για τον ευκολότερο υπολογισμό των αλγεβρικών πράξεων που απαιτούνται).

### Δεδομένα Εισόδου

- 1) **int OFDM\_SUBCARRIERS\_NUMBER:** αριθμός των υποκαναλιών OFDM που έχει το λαμβανόμενο OFDM frame (ορίζεται από τον χρήστη).
- 2) **int QPSK\_SYMBOLS\_PER\_SUBCARRIER\_NUMBER:** αριθμός των συμβόλων QPSK (data) που περιέχονται σε κάθε υποκανάλι OFDM.
- 3) **vector <vector<complex<double>> transmitterSymbolVector1:** vector που περιέχει τα σύμβολα που μεταδίδονται από την πρώτη κεραία τις χρονικές στιγμές  $t_1=0$  και  $t_2=1$ .
- 4) **vector <vector<complex<double>>> transmitterSymbolVector2:** vector που περιέχει τα σύμβολα που μεταδίδονται από την δεύτερη κεραία κατά τις χρονικές στιγμές  $t_1=0$  και  $t_2=1$ .

- 5) **vector <vector<complex <double >>> FFTVector1:** vector που περιέχει τα σύμβολα του σήματος που λαμβάνει η πρώτη κεραία μετά την ολοκλήρωση του Ταχέως Μετασχηματισμού Fourier.
- 6) **vector <vector<complex <double >>> FFTVector2:** vector που περιέχει τα σύμβολα του σήματος που λαμβάνει η δεύτερη κεραία μετά την ολοκλήρωση του Ταχέως Μετασχηματισμού Fourier.

Κατά την διάρκεια εκτέλεσης της μεθόδου **channelEstimator\_AlamoutiDecoder** καλείται η μέθοδος **MatrixXcd** **moorePenroseMatrixCalculator** (**MatrixXcd channelCoefficientsMatrix**). Η μέθοδος χρησιμοποιείται για τον υπολογισμό του πίνακα  $H^+$  των υποκαναλιών OFDM. Ως είσοδο λαμβάνει τον πίνακα (MatrixXcd – πίνακας της βιβλιοθήκης Eigen που περιέχει τιμές τύπου complex<double>) με τις τιμές των channel coefficients των υποκαναλιών OFDM, οι οποίες έχουν υπολογιστεί από την μέθοδο channelEstimator\_AlamoutiDecoder. Έξοδος της μεθόδου είναι ο πίνακας  $H^+$ , ο οποίος χρησιμοποιείται στη συνέχεια από την μέθοδο channelEstimator\_AlamoutiDecoder, για τον τελικό υπολογισμό των λαμβανόμενων συμβόλων.

### Δεδομένα Εξόδου

**vector < vector < complex < double > > retrievedFromAlamoutiQPSKSymbolsVector:** vector που περιέχει τα λαμβανόμενα σύμβολα QPSK, μετά την αποκωδικοποίηση Alamouti. Το vector έχει διαστάσεις (OFDM\_SUBCARRIERS\_NUMBER x QPSK\_SYMBOLS\_PER\_SUBCARRIER\_NUMBER), είναι δηλαδή ίσων διαστάσεων με το vector qpskModulatedVector (βήμα 2, πομπός).

$\widehat{X}_1$	$\widehat{X}_2$	$\widehat{X}_3$	$\widehat{X}_4$	$\widehat{X}_5$	$\widehat{X}_6$	$\widehat{X}_7$	$\widehat{X}_8$
$\widehat{X}_9$	$\widehat{X}_{10}$	$\widehat{X}_{11}$	$\widehat{X}_{12}$	$\widehat{X}_{13}$	$\widehat{X}_{14}$	$\widehat{X}_{15}$	$\widehat{X}_{16}$
$\widehat{X}_{17}$	$\widehat{X}_{18}$	$\widehat{X}_{19}$	$\widehat{X}_{20}$	$\widehat{X}_{21}$	$\widehat{X}_{22}$	$\widehat{X}_{23}$	$\widehat{X}_{24}$
$\widehat{X}_{25}$	$\widehat{X}_{26}$	$\widehat{X}_{27}$	$\widehat{X}_{28}$	$\widehat{X}_{29}$	$\widehat{X}_{30}$	$\widehat{X}_{31}$	$\widehat{X}_{32}$

Σχήμα 7.25 – Ληφθέντα σύμβολα μετά την αποκωδικοποίηση Alamouti (vector retrievedFromAlamoutiQPSKSymbolsVector)

# Κεφάλαιο 8 - Εργαστηριακές Μετρήσεις Συστήματος

## 8.1 Εργαστηριακή Διάταξη

Η ορθότητα λειτουργίας του συστήματος, ελέγχθηκε με την ανάπτυξη ενός πραγματικού συστήματος SDR στο Εργαστήριο Τηλεπικοινωνιακών Συστημάτων του Πανεπιστημίου Πειραιώς, κατά το πρότυπο του Σχήματος 1.4. Οι τιμές των παραμέτρων του συστήματος που χρησιμοποιήθηκαν κατά τις εργαστηριακές δοκιμές είναι οι ακόλουθες:

- Αριθμός OFDM frames: 10
- Αριθμός subcarriers/OFDM frame: 64
- Μέγεθος Κυκλικού Προθέματος: 16 δείγματα
- Αριθμός Συμβόλων δεδομένων (QPSK)/subcarrier: 32 σύμβολα
- Ρυθμός μετάδοσης/λήψης: 2Mbps
- Συχνότητα μετάδοσης/λήψης: 700 MHz
- Κέρδος ενίσχυσης πομπού/δέκτη: 30 dBi
- Τύπος δεδομένων: double
- Χρονικό διάστημα απόρριψης σήματος στον δέκτη: 1 sec
- Συνολικός αριθμός λαμβανόμενων δειγμάτων στον δέκτη:  $5 \cdot 10^5$  δείγματα

Σε κάθε OFDM frame, τα παρακάτω subcarriers κρατήθηκαν μηδενικά.

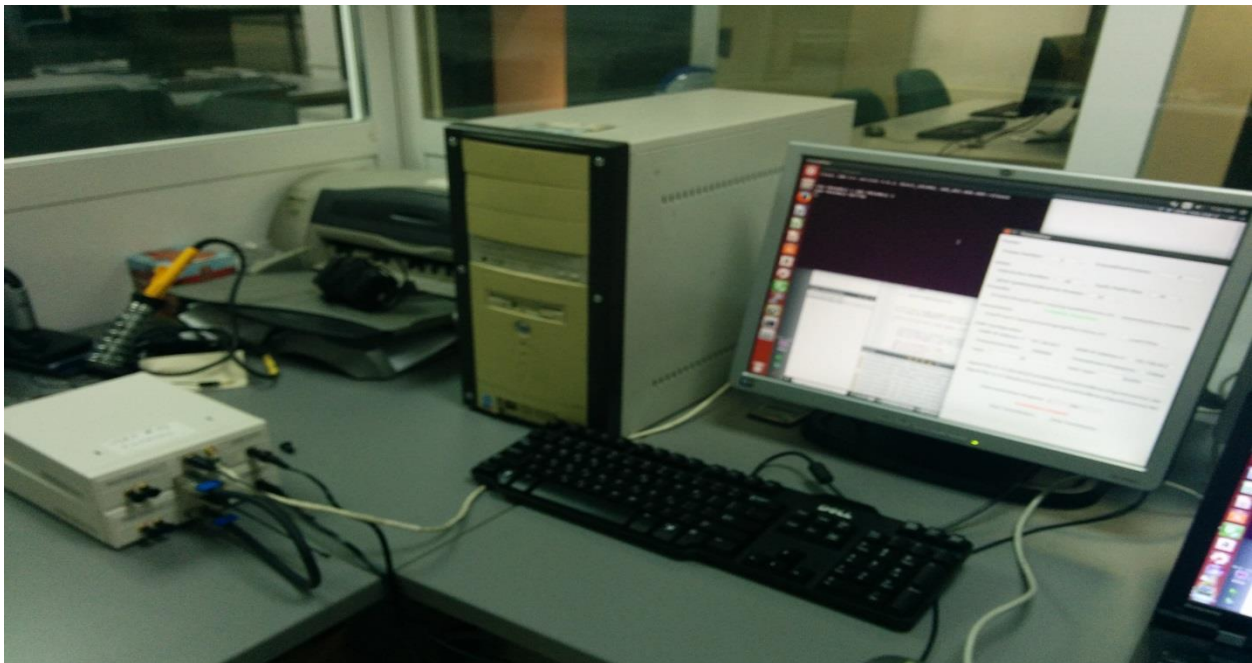
-32 έως -26
0
25 έως 31

**Πίνακας 8.1 – Μηδενικά subcarriers**

Το σύστημα SDR που αναπτύχθηκε παρουσιάζεται στις εικόνες που ακολουθούν:

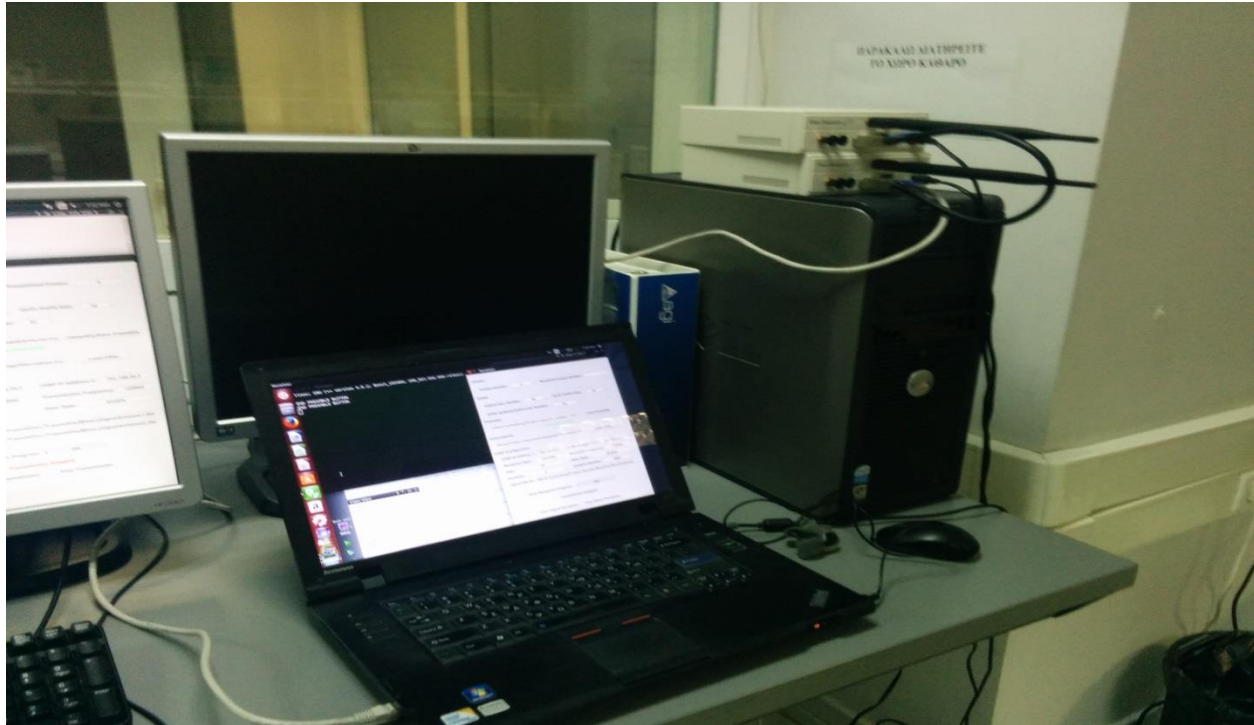


**Εικόνα 8.1 – Γενική άποψη του συστήματος SDR MIMO 2x2 που αναπτύχθηκε στο εργαστήριο. Αριστερά διακρίνεται ο υπολογιστής – host του πομπού και δεξιά ο υπολογιστής host του δέκτη, κάθε ένας εκ των οποίων είναι συνδεδεμένος με 2 συσκευές USRP.**



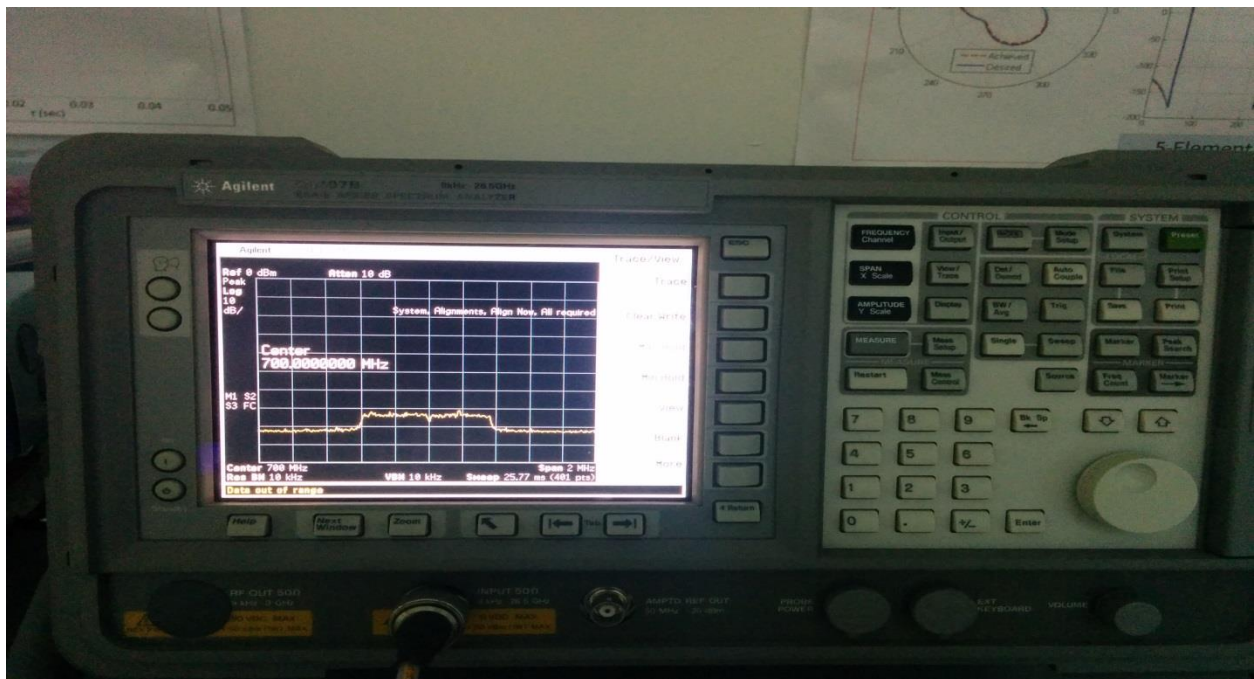
**Εικόνα 8.2 – Πομπός**



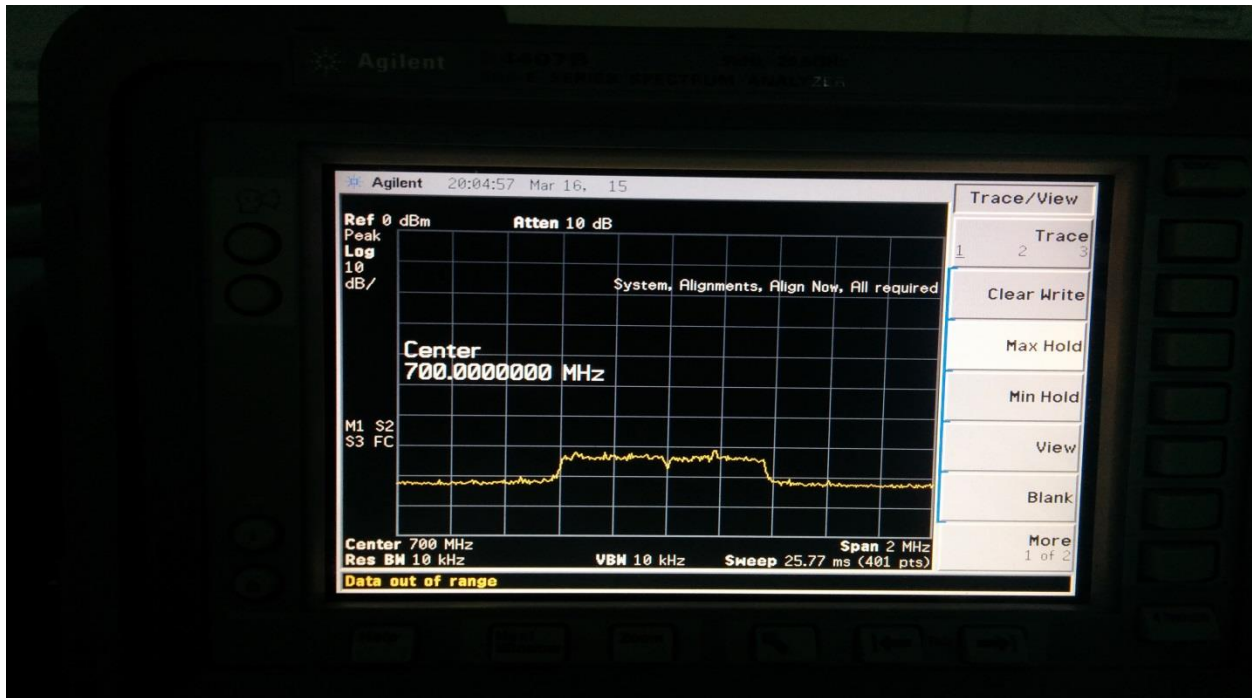


**Εικόνα 8.3 – Δέκτης**

Στις Εικόνες 8.4 και 8.5 παρουσιάζεται η μέτρηση του εκπεμπόμενου σήματος από τον παλμογράφο του εργαστηρίου:

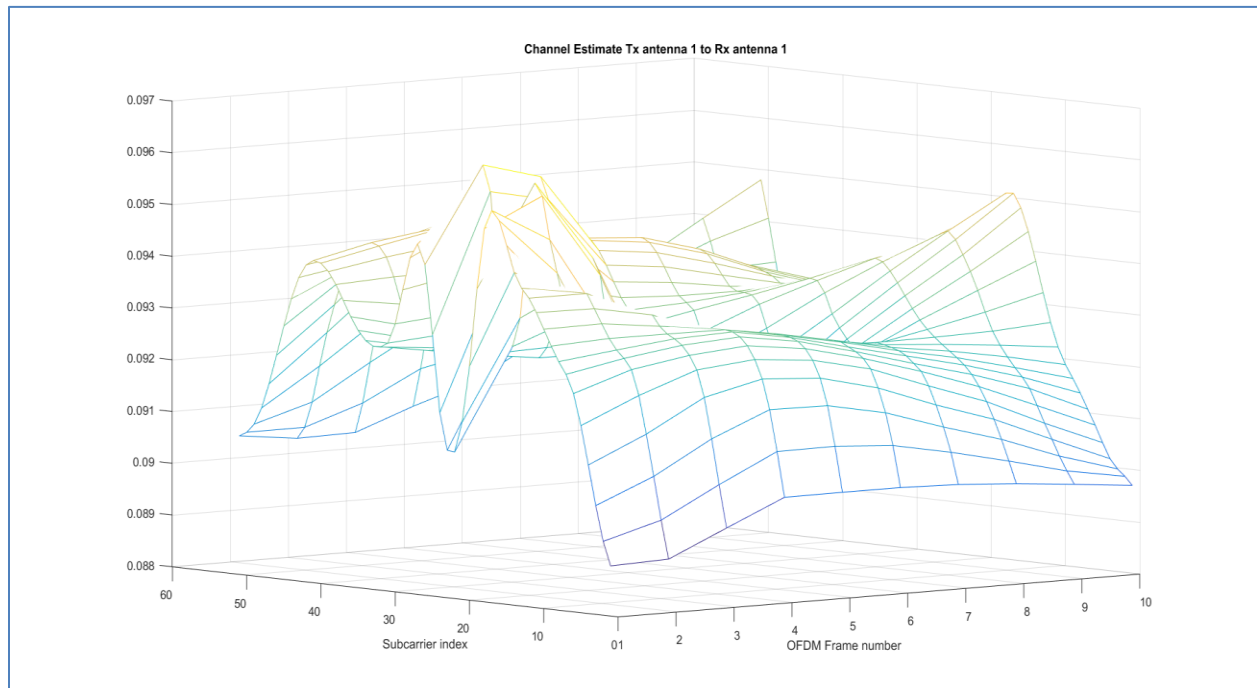


**Εικόνα 8.4 - Εκπεμπόμενο σήμα**



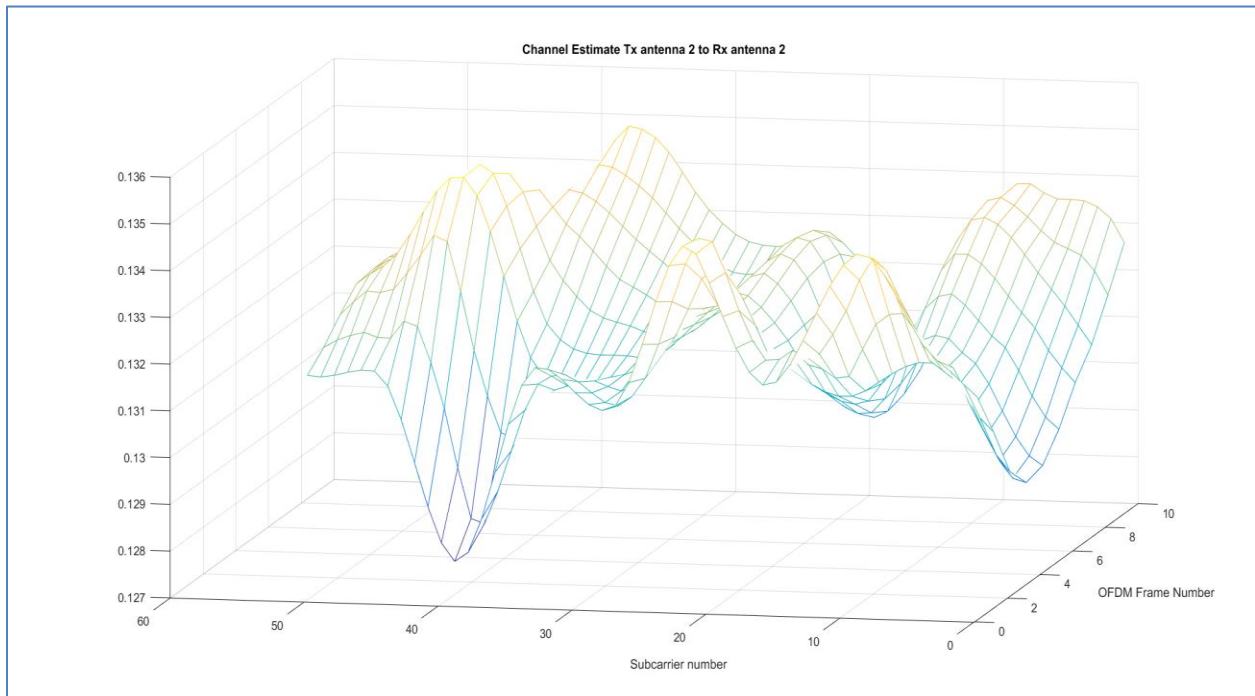
Εικόνα 8.5 – Εκπεμπόμενο σήμα

Ακολουθούν οι εκτιμώμενες τιμές του διαύλου για κάθε OFDM subcarrier μεταξύ της 1<sup>ης</sup> κεραίας του πομπού και δέκτη: (τιμή  $h_{11}$  του πίνακα συντελεστών διαύλου  $H$ ):



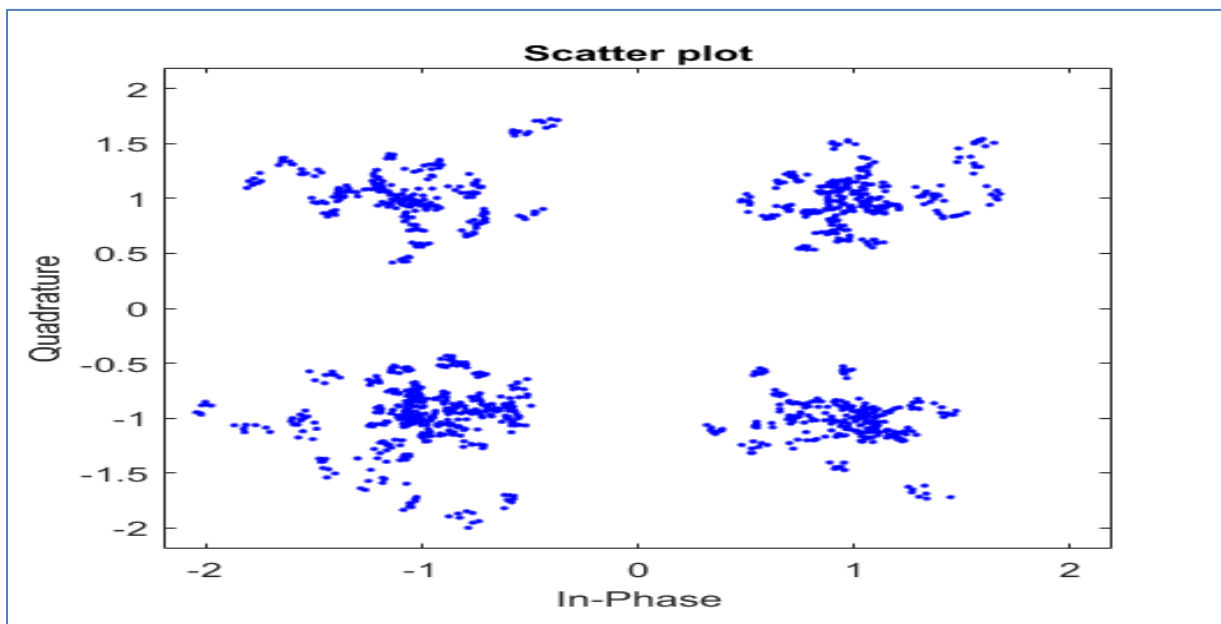
Εικόνα 8.6 – Εκτιμώμενες τιμές καναλιού  $h_{11}$

Αντιστοίχως, οι εκτιμούμενες τιμές του διαύλου για κάθε OFDM subcarrier μεταξύ της 2<sup>ης</sup> κεραίας του πομπού και δέκτη, εμφανίζονται στην Εικόνα 8.7:



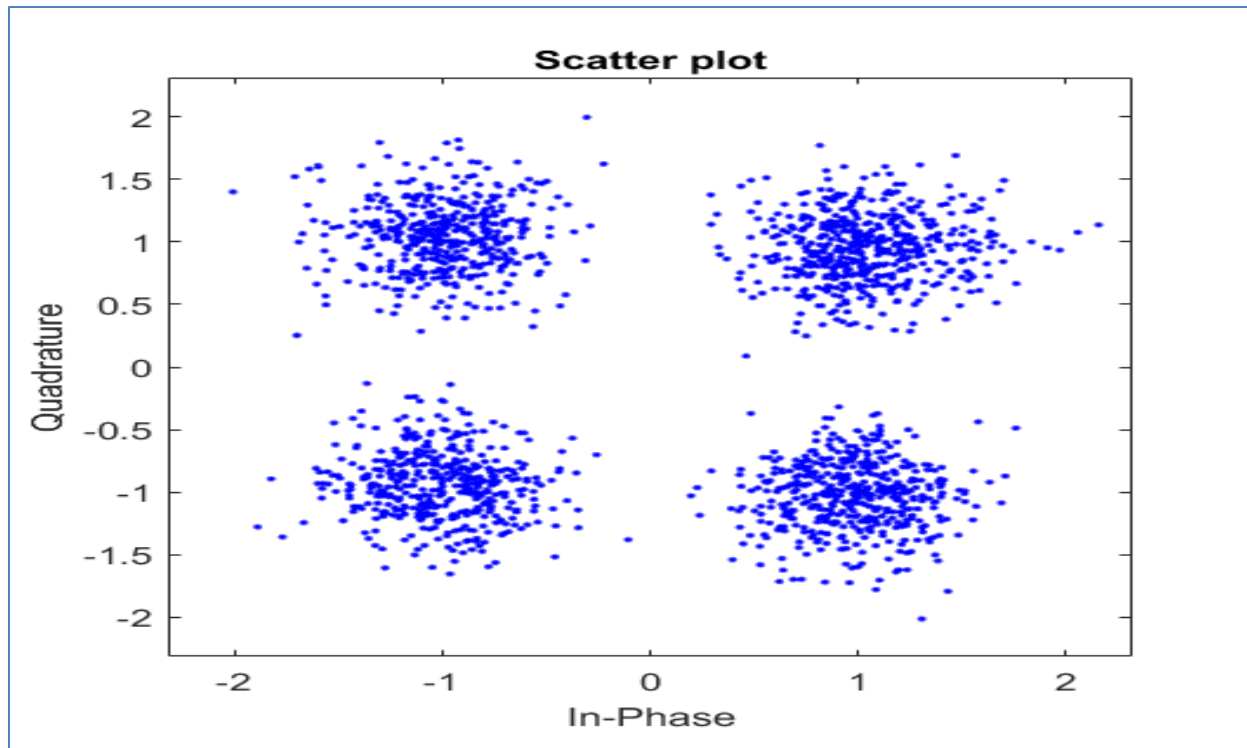
**Εικόνα 8.7 – Εκτιμούμενες τιμές καναλιού  $h_{22}$**

Ακολουθεί ο αστερισμός QPSK (QPSK constellation) του ληφθέντος σήματος της πρώτης κεραίας του δέκτη (Εικόνα 8.8):



**Εικόνα 8.8**

Ο αστερισμός του ληφθέντος σήματος της δεύτερης κεραίας του δέκτη είναι ο ακόλουθος (Εικόνα 8.9):



Εικόνα 8.9

# Βιβλιογραφία

- [1] J.Proakis, M. Salehi, Telecommunication System Engineering, Prentice Hall, Inc., New Jersey, 2002.
- [2] Fuking Xiong, Digital Modulation Techniques, Artech House, Inc., Norwood, 2004.
- [3]
- [4] J.R. Hampton, Introduction to MIMO Communications, Cambridge University Press, Cambridge, 2014.
- [5] A. Mohammadi, F. M. Ghannouchi, RF Transceiver Design for MIMO Wireless Communications, Springer, Berlin, 2012.
- [6] R. Prasad, OFDM for Wireless Communications Systems, Artech House, Inc., Boston, 2004
- [7] T.D Chiueh, P.Y. Tsai, OFDM Baseband Receiver Design for Wireless Communications, John Wiley & Sons, Singapore, 2007
- [8] Α. Κανάτας, Φ.Κωνσταντίνου, Γ. Πάντος, Συστήματα Κινητών Επικοινωνιών, Παπασωτηρίου, Αθήνα, 2008.
- [9] Herbert Schild, C++Programming CookBook, McGraw-Hill, New York, 2008
- [10] [www.ettus.com](http://www.ettus.com).
- [11], National Instruments, An introduction to Software Defined Radio Systems, 2013, (<http://eduscol.education.fr/sti/sites/eduscol.education.fr.sti/files/ressources/techniques/3801/3801-manuel-tp-usrp-3h.pdf>)
- [12] Din Kelly, The Universal Hardware Driver, 2011, ([http://people.bu.edu/mrahaim/NEWSDR/Presentations/NEWSDR\\_Kelly.pdf](http://people.bu.edu/mrahaim/NEWSDR/Presentations/NEWSDR_Kelly.pdf)).