



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΠΟΣΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΠΡΟΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

**Υλοποίηση OFDM Συστήματος σε SDR RFSoc 4x2
Implementing an OFDM System using RFSoc 4x2 SDR**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Φώτιου Κατσίφα

Επιβλέπων : Κ. Μαλιάτσος, Επίκουρος Καθηγητής

Μέλη εξεταστικής επιτροπής: Δ. Σκούτας, Χ. Σκιάνης

Σάμος, Ιούνιος 2025

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πρόλογος και ευχαριστίες

Η παρούσα διπλωματική εργασία αποτελεί ένα σημαντικό σταθμό στην ακαδημαϊκή μου πορεία. Η παρούσα διπλωματική εργασία αποτελεί ένα σημαντικό σταθμό στην ακαδημαϊκή μου πορεία. Μέσα από αυτή την ερευνητική προσπάθεια, είχα την ευκαιρία να εμβαθύνω στο επιστημονικό πεδίο που με ενδιαφέρει και να αναπτύξω δεξιότητες που θα με συνοδεύουν στη συνέχιση των ακαδημαϊκών μου στόχων και στη μετέπειτα επαγγελματική μου σταδιοδρομία.

Με την ολοκλήρωση της διπλωματικής μου εργασίας, θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες σε όλους όσους συνέβαλαν στην πραγματοποίησή της. Πρωτίστως, οφείλω ένα μεγάλο ευχαριστώ στον επιβλέποντα καθηγητή μου, Κωνσταντίνο Μαλιάτσο, για την πολύτιμη καθοδήγηση, την αμέριστη υποστήριξη και την εμπιστοσύνη που μου έδειξε καθ' όλη τη διάρκεια εκπόνησης της εργασίας. Η επιστημονική του κατάρτιση, οι τεκμηριωμένες παρατηρήσεις του και η καθοδήγησή του συνέβαλαν ουσιαστικά στην ακαδημαϊκή μου πρόοδο.

Θα ήταν παράλειψη να μην εκφράσω τις ευχαριστίες μου στους φίλους μου, που στάθηκαν δίπλα μου στις απαιτητικές στιγμές αυτής της διαδρομής. Ιδιαίτερη μνεία οφείλω στον Νικόλαο Ζαχαρία, του οποίου η βοήθεια στη δόμηση και προετοιμασία της παρουσίασης υπήρξε καθοριστική· χωρίς εκείνον, δεν θα ήξερα από πού να ξεκινήσω.

Θα ήθελα επίσης να ευχαριστήσω θερμά την Μαρία Βολακάκη, τον Γρηγόρη Παπαϊωάννου και τον Δημοσθένη Βουγιούκα για την απόφασή τους να με υποστηρίξουν και να με εντάξουν, αντίστοιχα, σε ερευνητικά προγράμματα. Η εμπιστοσύνη τους και η ευκαιρία που μου προσέφεραν υπήρξαν καθοριστικές για τη επιλογή της πορείας μου.

Τέλος, το μεγαλύτερο ευχαριστώ ανήκει στους γονείς και την αδερφή μου, για την αδιάκοπη στήριξη, την κατανόηση και την αγάπη που μου προσέφεραν όχι μόνο κατά τη διάρκεια της διπλωματικής μου εργασίας, αλλά και καθ' όλη τη διάρκεια των σπουδών μου. Χωρίς τη συμπαράστασή τους, η ολοκλήρωση αυτού του έργου δε θα ήταν εφικτή.

©2025

του

Φώτιου Κατσίφα

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πίνακας περιεχομένων

1	Εισαγωγή	1
1.1	Η Εξέλιξη των Ασύρματων Δικτύων και οι Προκλήσεις της 6ης Γενιάς	1
1.2	Αντικείμενο διπλωματικής.....	1
1.2.1	Στόχοι και συνεισφορά	2
1.3	Μεθοδολογία.....	3
1.4	Δομή της διπλωματικής	4
2	Θεωρητικό Υπόβαθρο.....	6
2.1	Ορθογωνική Πολυπλεξία με Διαίρεση Συχνότητας.....	6
2.2	5G NR.....	8
2.2.1	Ιεραρχία Χρονικής Δομής.....	8
2.2.2	Numerology και Subcarrier Spacing	8
2.2.3	Διάρκεια OFDM Συμβόλου και Slot ανά Numerology.....	9
2.2.4	Σταθερό Υποπλαίσιο ως Μονάδα Αναφοράς	9
2.2.5	Αριθμός Slot και Συμβόλων ανά Frame.....	10
2.2.6	Ευέλικτη Διάταξη Downlink/Uplink σε TDD.....	11
2.2.7	Εναλλαγή DL/UL και Guard Symbols.....	12
2.2.8	Ειδικές Εκπομπές και Σύμβολα: SSB & SIB	13
2.2.8.1	SSB (Synchronization Signal Block).....	13
2.2.8.2	SIB1 (System Information Block Type 1)	14
2.3	SIMO και MIMO Ασύρματα Συστήματα	14
2.4	Τεχνικές Συγχρονισμού σε OFDM	15
2.5	Εκτίμηση Καναλιού και Πιλοτικά Υποφέροντα	15
2.6	Κωδικοποίηση και Διόρθωση Σφαλμάτων (FEC)	16
2.7	Αρχιτεκτονική του Xilinx RFSoc 4x2	17
2.8	RFSoc MTS Overlay.....	18
2.8.1	Αρχιτεκτονική και Λειτουργικότητα.....	18
2.8.2	Πλεονεκτήματα στη Συγκεκριμένη Εφαρμογή.....	19
2.8.3	Περιορισμοί και Επεκτασιμότητα	19
3	Μοντελοποίηση του Συστήματος σε Περιβάλλον Python.....	21
3.1	Γενική Αρχιτεκτονική Πομποδέκτη.....	21
3.2	Παραγωγή Συμβόλων και FEC Κωδικοποίηση	22
3.3	Προσομοίωση SIMO Καναλιού.....	23

3.4	Συγχρονισμός και Εκτίμηση Συχνοτικής Μετατόπισης (CFO)	25
3.5	Εκτίμηση Καναλιού και Maximal Ratio Combining (MRC)	26
3.6	Αποδιαμόρφωση και Αποκωδικοποίηση	27
3.7	Υπολογισμός Μετρικών και Ανάλυση Απόδοσης	28
4	Υλοποίηση του Συστήματος σε Πλατφόρμα Xilinx RFSoc 4x2	30
4.1	Βελτιώσεις και Αναβαθμίσεις στην Υλοποίηση του Συστήματος	30
4.1.1	<i>Soft-Decision Αποκωδικοποίηση με Αναβαθμισμένο Συνελικτικό Κώδικα</i>	<i>30</i>
4.1.2	<i>Επέκταση σε MIMO</i>	<i>32</i>
4.2	Δημιουργία και Μετάδοση του OFDM Σήματος στο RFSoc	33
4.3	Λήψη και Ανακατασκευή του Ληφθέντος Σήματος	35
4.3.1	<i>Καταγραφή Δειγμάτων.....</i>	<i>35</i>
4.3.2	<i>Ανακατασκευή του Συμπλεγμένου Baseband Σήματος</i>	<i>36</i>
4.4	Αποδιαμόρφωση και Συγχρονισμός του Ληφθέντος OFDM Σήματος στο RFSoc	38
4.5	Αποδιαμόρφωση και Soft Viterbi Αποκωδικοποίηση	38
4.5.1	<i>Soft QAM Demapping και Log-Likelihood Ratios (LLRs)</i>	<i>39</i>
4.5.2	<i>Ενίσχυση Εμπιστοσύνης (Scaling των LLRs)</i>	<i>39</i>
4.5.3	<i>Soft-Input Viterbi Αποκωδικοποίηση</i>	<i>40</i>
4.5.4	<i>Σχόλια για την Εφαρμογή</i>	<i>40</i>
4.6	Εκτίμηση Καναλιού και Ισοστάθμιση	41
4.7	Μέτρηση Απόδοσης: BER και EVM.....	41
5	Συγκριτική Αξιολόγηση και Πειραματικά Αποτελέσματα	42
5.1	Πλαίσιο Πειραματισμού	42
5.2	Χαρακτηριστικά Διαμόρφωσης και Χωρικής Λήψης.....	43
5.2.1	<i>Κατανομή Υποφερόντων και Χαρτογράφηση Συμβόλων</i>	<i>43</i>
5.2.2	<i>Σύγκριση Μεγέθους Σήματος Εκπομπής-Λήψης στο Πεδίο του Χρόνου</i>	<i>44</i>
5.3	Συγχρονισμός και Χρονική Ευθυγράμμιση	47
5.3.1	<i>Μετρική Schmidl & Cox για Συγχρονισμό.....</i>	<i>47</i>
5.3.2	<i>Σήμα Λήψης με Ενδείξεις Χρονικής Στόχευσης.....</i>	<i>49</i>
5.3.3	<i>Επίδραση Διόρθωσης Συχνότητας.....</i>	<i>50</i>
5.4	Εκτίμηση Καναλιού και Ισοστάθμιση	52
5.4.1	<i>Εκτίμηση Καναλιού με Least Squares (LS) και Maximum Ratio Combining (MRC).....</i>	<i>52</i>
5.4.2	<i>Φασική Συμπεριφορά πριν και μετά την Ισοστάθμιση</i>	<i>54</i>
5.4.3	<i>Συμπερασματικά Σχόλια.....</i>	<i>56</i>
5.5	Συμβολοδιάγραμμα και QAM Απόφαση.....	56
5.5.1	<i>Διάγραμμα Αστερισμού μετά την Ισοστάθμιση.....</i>	<i>56</i>
5.5.2	<i>Εκτιμήσεις QAM και Πλησιέστερα Σημεία Αστερισμού (Hard Demap)</i>	<i>58</i>

5.5.3	Συνοπτικά Ευρήματα.....	60
5.6	Soft Demapping και LLR Ανάλυση.....	61
5.6.1	Κατανομές LLR ανά Bit.....	61
5.6.2	Χαρτογράφηση Εμπιστοσύνης Bit ανά Σύμβολο.....	62
5.6.3	Ακολουθία Bits πριν την Αποκωδικοποίηση FEC.....	64
5.6.4	Soft Decisions vs. Equalized and Transmitted QAM Symbols.....	66
5.6.5	Κύρια Ευρήματα.....	68
6	Πιθανές Βελτιώσεις και Μελλοντική Επέκταση	69
6.1	Υποστήριξη για μεταβλητή παραμετροποίηση του Φυσικού Καναλιού (Bandwidth Adaptation).....	69
6.2	Αντικατάσταση Viterbi με Σύγχρονες FEC Τεχνικές.....	69
6.3	Ενσωμάτωση Ασύρματου Καναλιού με Πραγματικές Κεραίες.....	70
6.4	Χρήση του Programmable Logic για Acceleration.....	70
7	Συμπεράσματα.....	71
7.1	Σύνοψη αποτελεσμάτων	71
	Βιβλιογραφία.....	72
	Παράρτημα I [Κώδικας Προσομοίωσης]	73
	Παράρτημα II [Κώδικας RFSoc]	83
	Παράρτημα III [Github]	94

Λίστα Σχημάτων

Εικόνα 1: Φάσμα συχνότητας υποφορέων OFDM, σχήμα 16.9 από [3]	7
Εικόνα 2: Σύνθεση ενός ενδεικτικού συμβόλου OFDM, σχήμα 16.17 από [3]	7
Εικόνα 3: Πιλοτική ρύθμιση σε 4G LTE, σχήμα 16.18 από [3].....	8
Εικόνα 4: OFDM παραμετροποίηση και διάρκεια υποπλαισίων και σχισμών.	11
Εικόνα 5: Μετρική χρονισμού Schmidl & Cox, σχήμα 16.21 από [3].....	15
Εικόνα 6: Απεικόνιση ενός πολυδιαδρομικού καναλιού, σχήμα 16.2 από [3].....	16
Εικόνα 7: RFSoc-MTS Block Diagram.....	20
Εικόνα 8: Αρχιτεκτονική σύνδεσης του κυκλώματος DAC–ADC στο RFSoc 4x2. Το σήμα μεταδίδεται από τους DAC μέσω απευθείας καλωδίου και δειγματοληπτείται από δύο ADCs (MIMO).....	37
Εικόνα 9: Κατανομή υποφερόντων στο OFDM σύμβολο.....	44
Εικόνα 10: Σηματοστερισμός 16-QAM με κώδικα Gray.....	44
Εικόνα 11: Πλάτος σήματος σε πομπό και δέκτες με χρήση καλωδίων (RFSoc).....	46
Εικόνα 12: Πλάτος σήματος σε πομπό και δέκτες με εξομοίωση	46
Εικόνα 13: Μετρικό Schmidle and Cox για συγχρονισμό	48
Εικόνα 14: Μετρικό συγχρονισμού για πολλαπλές επαναλήψεις του σήματος αναφοράς	48
Εικόνα 15: Το λαμβανόμενο σήμα με ένδειξη της έναρξης του σήματος Zadoff-Chu, και της έναρξης και λήξης του συμβόλου πληροφορίας (εξομοίωση).....	49
Εικόνα 16: Το λαμβανόμενο σήμα με ένδειξη της έναρξης του σήματος Zadoff-Chu (ενισχυμένη ακολουθία), και της έναρξης και λήξης του συμβόλου πληροφορίας (RFSoc).....	50
Εικόνα 17: Επίδραση της μετατόπισης συχνότητας στο σήμα (εξομοίωση).....	51
Εικόνα 18: Επίδραση της μετατόπισης συχνότητας στο σήμα (RFSoc)	51
Εικόνα 19: Εκτίμηση καναλιού (εξομοίωση).....	53
Εικόνα 20: Εκτίμηση καναλιού (RFSoc).....	53
Εικόνα 21: : Φάση συμβόλων πριν και μετά την εξισορρόπηση (για κάθε κεραία και MRC - εξομοίωση).55	
Εικόνα 22: Φάση συμβόλων πριν και μετά την εξισορρόπηση (για κάθε κεραία και MRC - RFSoc)	55
Εικόνα 23: Αστερισμός λαμβανόμενου σήματος μετά την εξισορρόπηση (εξομοίωση).	57
Εικόνα 24: Αστερισμός λαμβανόμενου σήματος μετά την εξισορρόπηση (εξομοίωση).	57
Εικόνα 25: "Σκληρή" αποδιαμόρφωση - εκτίμηση των QAM συμβόλων και αντιστοίχιση με το πλησιέστερο σημείου το σηματοστερισμού (εξομοίωση).	59
Εικόνα 26: "Σκληρή" αποδιαμόρφωση - εκτίμηση των QAM συμβόλων και αντιστοίχιση με το πλησιέστερο σημείου το σηματοστερισμού. (RFSoc).....	59
Εικόνα 27: Κατανομή των LLR ανά bit (εξομοίωση).....	61
Εικόνα 28: Κατανομή των LLR ανά bit (RFSoc).....	62
Εικόνα 29: «Θερμικός» χάρτης για την εμπιστοσύνη των bits διαμέσου των συμβόλων (RFSoc).	63
Εικόνα 30: «Θερμικός» χάρτης για την εμπιστοσύνη των bits διαμέσου των συμβόλων (εξομοίωση).	63
Εικόνα 31: Συρμός bit πριν την αποκωδικοποίηση – σύγκριση με την εκπεμπόμενη ακολουθία (εξομοίωση)	65
Εικόνα 32: Συρμός bit πριν την αποκωδικοποίηση – σύγκριση με την εκπεμπόμενη ακολουθία (RFSoc).65	

Εικόνα 33: LLR ήπιες αποφάσεις σε σχέση με τα εξισορροπημένα και εκπεμπόμενα σύμβολα (εξομοίωση)	66
Εικόνα 34: LLR ήπιες αποφάσεις σε σχέση με τα εξισορροπημένα και εκπεμπόμενα σύμβολα (RFSoc) ..	67

Ακρωνύμια

SISO	Single Input Single Output
SIMO	Single Input Multiple Output
MIMO	Multiple Input Multiple Output
SNR	Signal-to-Noise-Ratio
LOS	Line of Sight
NLOS	Non-Line of Sight
OFDM	Orthogonal Frequency Division Multiplexing
FPGA	Field Programmable Gate Array
RFSoc	Radio Frequency System-on-Chip
RTL	Register Transfer Level
DAC	Digital-to-Analog Converters
ADC	Analog-to-Digital Converters
RFSoc	Radio Frequency System-on-Chip
DMA	Direct Memory Access
SDR	Software Defined Radio
FEC	Forward Error Correction
MRC	Maximal Ratio Combining
LS	Least Squares
BER	Bit Error Rate
EVM	Error Vector Magnitude
CP	Cyclic Prefix
DVB	Digital Video Broadcasting
ICI	Inter-Carrier Interference
CFO	Carrier Frequency Offset
LLR	Log-likelihood ratio
PL	Programmable Logic
PS	Processing System
DSP	Digital signal processing
FFT	Fast Fourier Transform
IFFT	Inverse Fast Fourier Transform
LDPC	Low-Density Parity-Check
HDL	Hardware Description Language
TDD	Time Division Multiplexing
UE	User Equipment

NR	New Radio
LTE	Long Term Evolution
eMBB	Enhanced Mobile Broadband
URLLC	Ultra-Reliable Low Latency Communications
DL	Downlink
UL	Uplink
TDD	Time Division Duplex
PSS	Primary Synchronization Signal
SSS	Secondary Synchronization Signal
PBCH	Physical Broadcast Channel
MIB	Master Information Block

Περίληψη

Η παρούσα διπλωματική εργασία επικεντρώνεται στην ανάπτυξη και υλοποίηση ενός συστήματος Ορθογωνικής Πολυπλεξίας με Διαίρεση Συχνότητας στην πλατφόρμα Xilinx RFSoc 4x2, με στόχο την αποδοτική αξιοποίηση της ενσωματωμένης δυνατότητας μετατροπών DAC/ADC και του ισχυρού ενσωματωμένου FPGA. Στο πρώτο μέρος, σχεδιάζεται και προσομοιώνεται ένα πλήρες μοντέλο OFDM πομπού-δέκτη σε περιβάλλον Python, το οποίο περιλαμβάνει συγχρονισμό (Zadoff-Chu, Schmidl & Cox), εκτίμηση καναλιού (Least Squares, MRC), και FEC κωδικοποίηση/αποκωδικοποίηση.

Στη συνέχεια, το μοντέλο προσαρμόζεται και υλοποιείται στο RFSoc, χρησιμοποιώντας την πλατφόρμα RFSoc-MTS για τη ροή δεδομένων μέσω DMA, και αξιοποιώντας τους μετατροπείς DAC/ADC για πραγματική μετάδοση και λήψη σήματος. Πραγματοποιούνται συγκρίσεις μεταξύ της προσομοίωσης και των μετρήσεων από την υλοποίηση στο υλικό, και αξιολογούνται μετρικές όπως η πιθανότητα σφάλματος bit (Bit Error Rate) BER, και το διάνυσμα πλάτους σφάλματος (Error Vector Magnitude) EVM, και η ακρίβεια συγχρονισμού.

Τα αποτελέσματα καταδεικνύουν την επιτυχή υλοποίηση ενός πολυσύνθετου τηλεπικοινωνιακού συστήματος σε μία ενιαία πλατφόρμα, καθώς και τη δυνατότητα του RFSoc να λειτουργήσει ως ένα ευέλικτο εργαλείο ανάπτυξης για σύγχρονα συστήματα ασύρματης επικοινωνίας, όπως αυτά που χρησιμοποιούνται σε τεχνολογίες πέραν του 5G με προγραμματίσιμα ευέλικτα συστήματα Software Defined Radio (SDR).

Λέξεις Κλειδιά: Σύστημα OFDM, RFSoc πλατφόρμα, Xilinx RFSoc 4x2, Software Defined Radio, SDR, Κωδικοποίηση καναλιού, FEC, εκτίμηση καναλιού, LTE 4G, NR 5G, 6G, Μοντελοποίηση σε Python, Υλοποίηση σε Υλικό, FPGA, SIMO, MIMO.

Abstract

This thesis focuses on the development and implementation of an Orthogonal Frequency Division Multiplexing system on the Xilinx RFSoc 4x2 platform, aiming to efficiently leverage the integrated DAC/ADC converters and the powerful embedded FPGA. In the first part, a complete OFDM transceiver model is designed and simulated in Python, incorporating synchronization techniques (Zadoff-Chu, Schmidl & Cox), channel estimation (Least Squares, MRC), and Forward Error Correction decoding.

Subsequently, the model is adapted and implemented on the RFSoc using the RFSoc-MTS framework for DMA-based data streaming, while the DAC/ADC converters are utilized for real-time signal transmission and reception. Comparisons are drawn between the software simulation and the hardware implementation, and key performance metrics such as the Bit Error Rate - BER, and the Error Vector Magnitude - EVM, and synchronization accuracy are evaluated.

The results demonstrate the successful realization of a complex communication system on a unified platform, as well as the potential of RFSoc to serve as a flexible development tool for modern wireless communication systems, such as those employed in beyond 5G technologies exploiting the programmability and reconfigurability of Software-Defined Radios.

Keywords: *OFDM System, RFSoc Platform, Xilinx RFSoc 4x2, Software Defined Radio, SDR, Forward Error Correction FEC, Channel Estimation, LTE 4G, NR 5G, 6G, Python Modeling, Hardware Implementation, FPGA, SIMO, MIMO.*

1

Εισαγωγή

1.1 Η Εξέλιξη των Ασύρματων Δικτύων και οι Προκλήσεις της 6ης Γενιάς

Οι σύγχρονες ασύρματες τηλεπικοινωνίες, ιδίως στα πρότυπα LTE (4G), NR (5G) και τις μελλοντικές προδιαγραφές 6G, βασίζονται σε πολυσύνθετα συστήματα μετάδοσης για την επίτευξη υψηλών ταχυτήτων, χαμηλού λανθάνοντος χρόνου και μεγάλης αξιοπιστίας. Ένα από τα πλέον διαδεδομένα σχήματα διαμόρφωσης σε τέτοια συστήματα είναι η Ορθογωνική Πολυπλεξία με Διαίρεση Συχνότητας (Orthogonal Frequency Division Multiplexing - OFDM), λόγω της ανθεκτικότητάς της σε φαινόμενα πολυδιαδρομικής διάδοσης και της αποδοτικής χρήσης του φάσματος.

Παράλληλα, οι ανάγκες για καινοτομία, ταχύτατο πειραματισμό και υλοποίηση νέων τεχνολογιών σε επίπεδο υλικού οδήγησαν στην ανάπτυξη και ευρεία υιοθέτηση αρχιτεκτονικών Software Defined Radio - SDR, με εξέχον παράδειγμα τις πλατφόρμες Xilinx RFSoc. Οι πλατφόρμες αυτές ενσωματώνουν DAC/ADC μετατροπείς υψηλής ταχύτητας και προγραμματιζόμενη λογική, επιτρέποντας την άμεση υλοποίηση και δοκιμή ασύρματων συστημάτων σε πραγματικό χρόνο.

Η παρούσα διπλωματική εργασία εστιάζει στην ανάπτυξη ενός πλήρους OFDM συστήματος πομπού-δέκτη σε περιβάλλον Python, με έμφαση στον συγχρονισμό, την εκτίμηση καναλιού και την διόρθωση σφαλμάτων, και στη μετέπειτα υλοποίησή του σε πραγματικό υλικό μέσω της πλατφόρμας RFSoc 4x2. Ο στόχος είναι η αξιολόγηση της απόδοσης και αξιοπιστίας του συστήματος τόσο σε επίπεδο προσομοίωσης όσο και κατά την πραγματική μετάδοση σήματος μέσω των DAC/ADC μονάδων του RFSoc.

1.2 Αντικείμενο διπλωματικής

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η σχεδίαση, προσομοίωση και υλοποίηση ενός ολοκληρωμένου ασύρματου τηλεπικοινωνιακού συστήματος τύπου OFDM σε πλατφόρμα Xilinx RFSoc 4x2, το οποίο ενσωματώνει προηγμένες τεχνικές συγχρονισμού, εκτίμησης καναλιού και διόρθωσης σφαλμάτων. Το σύστημα υποστηρίζει Single Input Multiple Output - SIMO διάταξη με δύο κεραίες λήψης, επιτρέποντας την εφαρμογή τεχνικών εκτεταμένης εκτίμησης καναλιού και συνδυαστικής αποδιαμόρφωσης/αποκωδικοποίησης όπως το Maximal Ratio Combining - MRC.

Σε επίπεδο φυσικής διαστρωμάτωσης, ενσωματώνονται τεχνικές χρονικού και συχνοτικού συγχρονισμού, με χρήση ακολουθιών Zadoff-Chu για αδρό εντοπισμό και η μέθοδος Schmidl & Cox για ακριβή ευθυγράμμιση των πλαισίων. Η εκτίμηση καναλιού υλοποιείται με την μέθοδο ελαχίστων τετραγώνων Least Squares με χρήση ενδιάμεσων πιλοτικών υποφερόντων, ενώ για την αντιμετώπιση σφαλμάτων κατά τη μετάδοση ενσωματώνεται κωδικοποίηση καναλιού Forward Error Correction - FEC με συνελκτικούς κώδικες και αποκωδικοποίηση Viterbi.

Η προσομοίωση του συστήματος αναπτύσσεται σε Python και αξιοποιείται για την εκτενή αξιολόγηση των επιδόσεων σε ιδανικά και θορυβώδη κανάλια διάδοσης, με χρήση ρεαλιστικών μοντέλων όπως το 3GPP Tapped Delay Line TDL-D. Στη συνέχεια, το σύστημα προσαρμόζεται και υλοποιείται στο RFSoc 4x2, χρησιμοποιώντας την πλατφόρμα RFSoc-MTS για την ροή δεδομένων μέσω Direct Memory Access - DMA, με αποστολή OFDM κυματομορφών μέσω του DAC και καταγραφή τους μέσω των ADCs.

Σκοπός της εργασίας είναι η πειραματική επικύρωση της θεωρητικής σχεδίασης, η σύγκριση προσομοιωμένων και μετρημένων αποτελεσμάτων (όπως BER, EVM, σφάλμα συγχρονισμού) και η διερεύνηση της καταλληλότητας του RFSoc ως πλατφόρμας ανάπτυξης και επιτάχυνσης ασύρματων συστημάτων επόμενης γενιάς, όπως το 5G, το NR και τα μελλοντικά 6G πρότυπα με υπερυψηλό εύρος ζώνης μέχρι και 2Gbps.

1.2.1 Στόχοι και συνεισφορά

Οι ερευνητικοί στόχοι της διπλωματικής είναι:

1. Σχεδίαση και προσομοίωση ενός πλήρους συστήματος OFDM πομποδέκτη σε περιβάλλον Python.
2. Υποστήριξη SIMO αρχιτεκτονικής με δύο κεραίες λήψης.
3. Ενσωμάτωση τεχνικών συγχρονισμού:
 - Zadoff-Chu sequence για αδρό εντοπισμό
 - Schmidl & Cox για ακριβή ευθυγράμμιση συμβόλων.
4. Εκτίμηση καναλιού με:
 - Least Squares με πιλοτικά υποφέροντα.
 - Maximal Ratio Combining για ενίσχυση μέσω SIMO.
5. Εφαρμογή FEC με συνελκτικό κώδικα και Viterbi αποκωδικοποίηση.
6. Προσαρμογή και υλοποίηση του συστήματος σε πλατφόρμα RFSoc 4x2.
7. Χρήση του RFSoc-MTS overlay για μεταφορά κυματομορφών μέσω DMA, αξιοποιώντας DAC/ADC μετατροπείς.
8. Καταγραφή και ανάλυση μετρικών απόδοσης:
 - Bit Error Rate
 - Error Vector Magnitude
 - Ακρίβεια συγχρονισμού
9. Επικύρωση της δυνατότητας αποστολής υπερυρυζωνικών σημάτων με εύρος ζώνης μέχρι και 2 Gbps μέσω του RFSoc με χρήση των εργαλείων Python.

Η συνεισφορά της εργασίας περιλαμβάνει:

1. Ανάπτυξη ενός πλήρους MIMO OFDM συστήματος με συγχρονισμό, εκτίμηση καναλιού και FEC, σε Python στο RFSoc.
2. Επίδειξη της λειτουργικότητας του RFSoc ως SDR πλατφόρμας.
3. Πρακτική υλοποίηση πραγματικής μετάδοσης και λήψης OFDM κυματομορφών μέσω DAC → ADC.
4. Πειραματική επικύρωση της προσομοίωσης μέσω σύγκρισης με μετρήσεις σε hardware in a loop.
5. Συμβολή στη γεφύρωση θεωρητικής και πρακτικής ανάπτυξης τηλεπικοινωνιακών συστημάτων.
6. Δυνατότητα επέκτασης για έρευνα σε 5G/6G και τεχνολογίες SDR με χαμηλό κόστος και μεγάλη ευελιξία.

1.3 Μεθοδολογία

Η μεθοδολογική προσέγγιση που ακολουθήθηκε στο πλαίσιο της παρούσας διπλωματικής εργασίας βασίστηκε στον διαχωρισμό της διαδικασίας σε δύο διακριτές φάσεις: (α) την ανάπτυξη και αξιολόγηση ενός πλήρους συστήματος OFDM σε περιβάλλον Python, και (β) την υλοποίησή του σε πραγματικό υλικό μέσω της πλατφόρμας Xilinx RFSoc 4x2. Στόχος ήταν η δημιουργία ενός αξιόπιστου τηλεπικοινωνιακού πειραματικού περιβάλλοντος που να υποστηρίζει τεχνικές συγχρονισμού, εκτίμησης καναλιού και διόρθωσης σφαλμάτων σε συνθήκες τόσο προσομοίωσης όσο και πραγματικής μετάδοσης.

Στην πρώτη φάση, σχεδιάστηκε ένας πομποδέκτης OFDM με δυνατότητα SIMO (Single Input, Multiple Output), αξιοποιώντας δύο κεραίες λήψης. Το σύστημα υλοποιήθηκε εξολοκλήρου σε Python και περιλαμβάνει διαμόρφωση 16-QAM, εφαρμογή IFFT/FFT για τη μετάβαση μεταξύ πεδίου συχνοτήτων και χρόνου, προσθήκη κυκλικού προθέματος. Για τον χρονικό και συχνοτικό συγχρονισμό εφαρμόστηκαν δύο μέθοδοι: η χρήση ακολουθίας Zadoff-Chu για τον εντοπισμό των πλαισίων, και ο αλγόριθμος Schmidl & Cox για την ακριβή ευθυγράμμιση των OFDM συμβόλων, καθώς και η εκτίμηση της συχνοτικής μετατόπισης (Carrier Frequency Offset) και η ανάλογη διόρθωση.

Η εκτίμηση του καναλιού βασίστηκε στη μέθοδο των Ελαχίστων Τετραγώνων, με χρήση πιλοτικών υποφερόντων και ενδιάμεση παρεμβολή, ενώ η συνδυαστική εκμετάλλευση των δύο κεραιών πραγματοποιήθηκε με χρήση αλγορίθμου Maximal Ratio Combining. Παράλληλα, εφαρμόστηκε κωδικοποίηση Forward Error Correction μέσω συνελκτικού κώδικα με λόγο 1/2, και αποκωδικοποίηση Viterbi με soft-decision (ήπιας απόφασης) είσοδο. Το σύστημα δοκιμάστηκε σε ρεαλιστικό μοντέλο καναλιού 3GPP TDL-D, και αξιολογήθηκε ως προς την απόδοση (BER, EVM) υπό θορυβώδεις συνθήκες.

Στη δεύτερη φάση, το OFDM σήμα που παράγεται από την προσομοίωση μεταφέρεται μέσω του συστήματος DMA προς τον DAC της πλατφόρμας RFSoc, με την υποβοήθηση του έτοιμου RFSoc-MTS overlay, αποφεύγοντας έτσι την ανάγκη δημιουργίας ειδικής κατασκευής (custom) RTL λογικής σε επεξεργαστές Field Programmable Gate Arrays - FPGAs. Το σήμα μεταδίδεται σε πραγματικό υλικό και καταγράφεται από τον ADC της ίδιας πλατφόρμας. Στη συνέχεια, εφαρμόζεται η ίδια διαδικασία

αποδιαμόρφωσης, συγχρονισμού, εκτίμησης καναλιού και αποκωδικοποίησης όπως στην προσομοίωση.

Η συγκριτική ανάλυση μεταξύ των αποτελεσμάτων της προσομοίωσης και των μετρήσεων από το υλικό επιτρέπει την αξιολόγηση της πιστότητας της υλοποίησης, αλλά και τη διερεύνηση πιθανών μη ιδανικοτήτων, όπως μεταβλητότητα καθυστέρησης jitter, θόρυβος μετατροπών ή χρονική μετατόπιση, που προκύπτουν κατά την πραγματική μετάδοση. Η μεθοδολογία αυτή αναδεικνύει τις δυνατότητες της RFSoc ως πλατφόρμας υλοποίησης φυσικής διαστρωμάτωσης σύγχρονων SDR τηλεπικοινωνιακών συστημάτων και δημιουργεί τις βάσεις για μελλοντική επέκταση σε περιβάλλοντα πολλαπλών εισόδων/εξόδων ή ακόμα και αλγορίθμους μηχανικής μάθησης για εκτίμηση καναλιού.

1.4 Δομή της διπλωματικής

Η δομή της παρούσας διπλωματικής εργασίας ακολουθεί μια προοδευτική προσέγγιση, ξεκινώντας από τις βασικές θεωρητικές έννοιες και καταλήγοντας στην πλήρη υλοποίηση, αξιολόγηση και προοπτική επέκτασης ενός σύγχρονου συστήματος MIMO OFDM σε πλατφόρμα RFSoc.

Το Κεφάλαιο 2 παρουσιάζει το θεωρητικό υπόβαθρο του συστήματος, εστιάζοντας σε θεμελιώδεις αρχές όπως η διαμόρφωση OFDM και πώς αυτή υλοποιείται στο 5G New Radio (NR), οι αρχιτεκτονικές SISO/MIMO, οι τεχνικές συγχρονισμού με χρήση ακολουθιών Zadoff-Chu και του αλγορίθμου Schmidl & Cox, καθώς και οι μέθοδοι εκτίμησης καναλιού και ισοστάθμισης με LS και MRC. Επιπλέον, αναλύονται τα SDR χαρακτηριστικά του Xilinx RFSoc 4x2 και ο ρόλος του programmable logic στο φυσικό επίπεδο επικοινωνίας.

Το Κεφάλαιο 3 περιγράφει τη μοντελοποίηση ενός SISO OFDM συστήματος σε περιβάλλον Python. Αναλύεται λεπτομερώς η ροή του πομποδέκτη, περιλαμβάνοντας συγχρονισμό, εκτίμηση του καναλιού, ισοστάθμιση του καναλιού και αποκωδικοποίηση με hard-decision Viterbi. Η προσομοίωση γίνεται υπό ελεγχόμενες συνθήκες με χρήση καναλιού με διαλείψεις βασισμένο στο μοντέλο Tapped Delay Line TDL-D της 3GPP, 16-QAM διαμόρφωσης και συνελκτικού FEC.

Το Κεφάλαιο 4 μεταβαίνει στην υλοποίηση του συστήματος στην πλατφόρμα Xilinx RFSoc 4x2. Η αρχιτεκτονική μεταφέρεται από την Python στον ενσωματωμένο ARM πυρήνα της πλατφόρμας, αξιοποιώντας το RFSoc-MTS overlay για τη μεταφορά του OFDM σήματος μέσω DMA στον DAC και ανάκτηση από τα ADC κανάλια. Εφαρμόζεται συγχρονισμός με trigger-based λήψη, ενώ εισάγονται σημαντικές τεχνικές βελτιώσεις όπως η μετάβαση σε ήπιας απόφασης αποκωδικοποίηση soft-decision Viterbi decoding και η επέκταση από SISO σε πλήρες MIMO 2x2 σχήμα λήψης.

Το Κεφάλαιο 5 επικεντρώνεται στην πειραματική αξιολόγηση και σύγκριση μεταξύ της προσομοίωσης και της υλοποίησης στο RFSoc. Με χρήση διαγραμμάτων και μετρικών όπως BER, EVM, εκτίμησης φασματικής πυκνότητας ισχύος μέσω FFT (περιοδόγραμμα) και μέτρηση της απόδοσης του συγχρονισμού, αξιολογείται η απόδοση του συστήματος ανά στάδιο. Παρουσιάζονται επίσης οι διαφορές μεταξύ των περιβαλλόντων, τόσο ποιοτικά όσο και ποσοτικά, επιτρέποντας πλήρη χαρτογράφηση των επιπτώσεων της υλοποίησης σε υλικό.

Το Κεφάλαιο 6 συζητά πιθανές βελτιώσεις και προτάσεις επέκτασης, όπως η αντικατάσταση του Viterbi με σύγχρονες FEC τεχνικές, η υποστήριξη μεταβλητού εύρους ζώνης και η ενσωμάτωση πραγματικής

ασύρματης μετάδοσης με χρήση κεραιών. Εξετάζεται επίσης η αξιοποίηση του programmable logic για επιτάχυνση επιλεγμένων υποσυστημάτων.

Τέλος, το Κεφάλαιο 7 συνοψίζει τα συμπεράσματα της εργασίας, αξιολογεί τη συνολική συνεισφορά της προτεινόμενης μεθοδολογίας, και προτείνει κατευθύνσεις για μελλοντική έρευνα, εστιάζοντας στην ενοποίηση πλήρους PHY stack με επιδόσεις πραγματικού χρόνου και την ενσωμάτωση τεχνικών μηχανικής μάθησης για προσαρμοστική εκτίμηση και ισοστάθμιση καναλιού.

2

Θεωρητικό Υπόβαθρο

Το κεφάλαιο αυτό εισάγει τις βασικές θεωρητικές έννοιες που αποτελούν το υπόβαθρο για την υλοποίηση και αξιολόγηση του OFDM συστήματος. Εξετάζονται τόσο τεχνικές από τον χώρο των ασύρματων επικοινωνιών όσο και αρχιτεκτονικά στοιχεία του RFSoc.

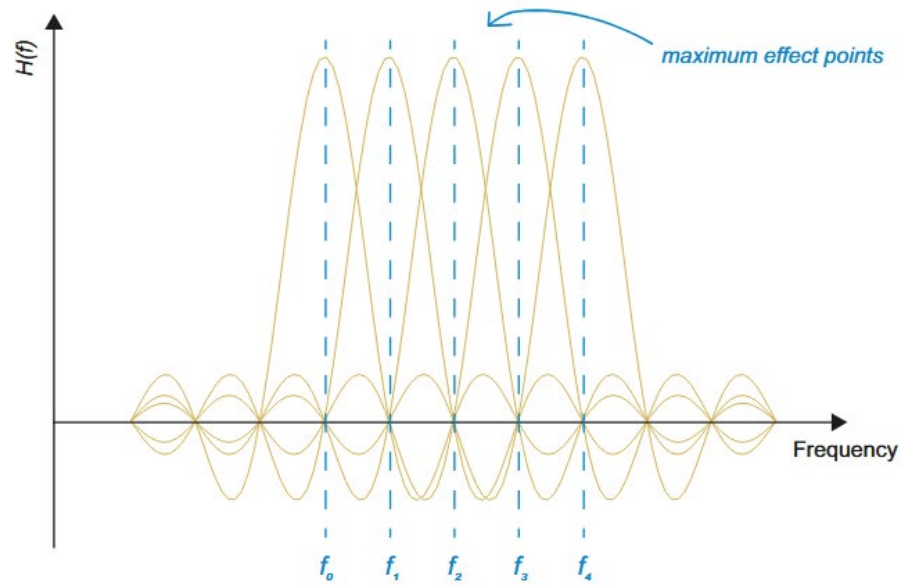
2.1 Ορθογωνική Πολυπλεξία με Διαίρεση Συχνότητας

Η Ορθογωνική Πολυπλεξία με Διαίρεση Συχνότητας αποτελεί μια από τις κυρίαρχες τεχνικές διαμόρφωσης που χρησιμοποιούνται στις σύγχρονες ασύρματες επικοινωνίες, συμπεριλαμβανομένων των συστημάτων LTE (4G), NR (5G), Wi-Fi (802.11a/g/n/ac) και DVB. Βασική της αρχή είναι η μετάδοση των δεδομένων παράλληλα σε πολλαπλά, στενά υποφέροντα (subcarriers), τα οποία είναι μεταξύ τους ορθογώνια, ώστε να αποφεύγεται η αλληλοπαρεμβολή (Inter-Carrier Interference).

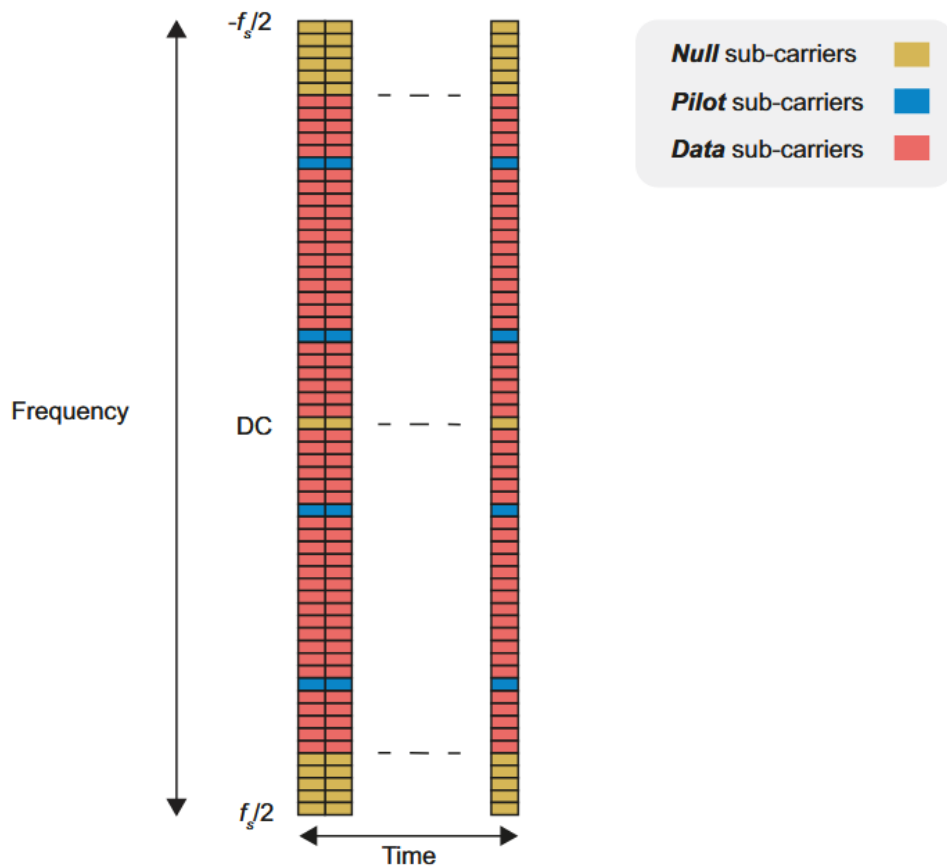
Η ορθογωνιότητα εξασφαλίζεται μαθηματικά μέσω της χρήσης του Διακριτού Μετασχηματισμού Fourier (DFT), που στην πράξη υλοποιείται με χρήση των γρήγορων μετασχηματισμών IFFT/FFT. Συγκεκριμένα, η IFFT μετατρέπει τα δεδομένα από το πεδίο της συχνότητας στο πεδίο του χρόνου για αποστολή, ενώ η FFT επαναφέρει τα δεδομένα στο πεδίο της συχνότητας κατά την αποδιαμόρφωση.

Ένα κρίσιμο στοιχείο της OFDM είναι το Κυκλικό Πρόθεμα (Cyclic Prefix – CP), δηλαδή η επανάληψη ενός τμήματος του τέλους του OFDM συμβόλου στην αρχή του. Το CP έχει ως στόχο την εξάλειψη της ενδοσυμβολικής παρεμβολής (ISI) που προκαλείται από πολυδιαδρομικά κανάλια και μετατρέπει το γραμμικό συνελκτικό κανάλι σε κυκλικό, καθιστώντας την ισοστάθμιση ιδιαίτερα απλή στο πεδίο της συχνότητας.

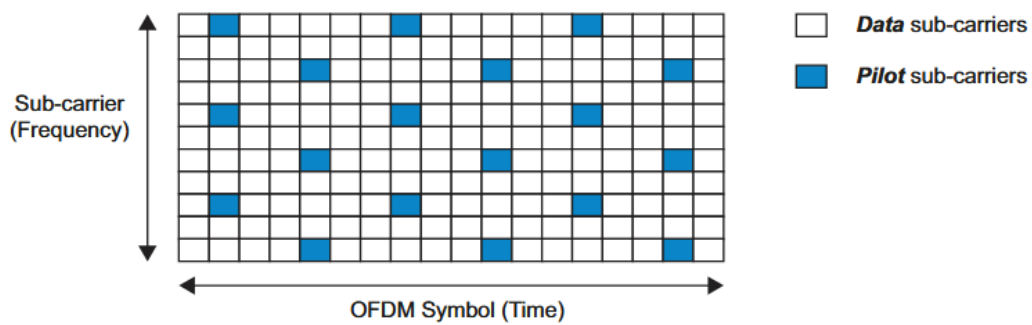
Συνολικά, η OFDM προσφέρει υψηλή φασματική απόδοση, ανοσία στην πολυδιαδρομή και χαμηλή υπολογιστική πολυπλοκότητα, καθιστώντας την ιδανική για περιβάλλοντα μεγάλης χωρητικότητας και κυψελοειδείς αρχιτεκτονικές.



Εικόνα 1: Φάσμα συχρότητας υποφορέων OFDM, σχήμα 16.9 από [3]



Εικόνα 2: Σύνθεση ενός ενδεικτικού συμβόλου OFDM, σχήμα 16.17 από [3]



Εικόνα 3: Πιλοτική ρύθμιση σε 4G LTE, σχήμα 16.18 από [3]

2.2 5G NR

Το 5G New Radio (NR) εισάγει μια εξαιρετικά ευέλικτη και προσαρμόσιμη δομή πλαισίου, σχεδιασμένη να ανταποκρίνεται στις ποικίλες απαιτήσεις των σύγχρονων εφαρμογών κινητών επικοινωνιών. Βασίζεται στη διαμόρφωση OFDM, όπως και ο προκάτοχός του LTE, αλλά εισάγει σημαντικές καινοτομίες, όπως η έννοια του numerology και η δυναμική διαχείριση του χρόνου, για να μεγιστοποιήσει την απόδοση και την ευελιξία.

2.2.1 Ιεραρχία Χρονικής Δομής

Το 5G NR διατηρεί μια ιεραρχική δομή χρόνου παρόμοια με αυτή του LTE:

- **Ραδιοπλαίσιο (Radio Frame):** Έχει σταθερή διάρκεια 10 ms.
- **Υποπλαίσιο (Subframe):** Κάθε ραδιοπλάσιο υποδιαιρείται σε 10 υποπλαίσια, το καθένα με διάρκεια 1 ms. Αυτό το 1 ms υποπλάσιο παραμένει σταθερή χρονική μονάδα αναφοράς, ανεξαρτήτως numerology.
- **Slot (Χρονική Σχισμή):** Κάθε υποπλάσιο περιέχει έναν ή περισσότερους slots, ανάλογα με το εφαρμοζόμενο numerology.
- **OFDM Σύμβολο:** Κάθε slot απαρτίζεται από διαδοχικά OFDM σύμβολα.

Για κανονικό κυκλικό πρόθεμα, κάθε slot αποτελείται σταθερά από 14 OFDM σύμβολα, ανεξαρτήτως αριθμολογίας (numerology, για το NR είναι η παραμετροποίηση του OFDM πλαισίου) – αριθμός ίδιος με αυτόν του LTE. Σε περίπτωση εκτεταμένου κυκλικού προθέματος (Cyclic Prefix – CP), κάθε slot περιέχει 12 σύμβολα.

Η διάταξη των πόρων στο slot ακολουθεί επίσης το LTE: 12 υποφέρουσες αποτελούν ένα μπλοκ ραδιοπόρων (resource block) στο πεδίο συχνότητας. Ωστόσο, η απόσταση μεταξύ των υποφερόμενων καθορίζεται πλέον δυναμικά από το numerology.

2.2.2 Numerology και Subcarrier Spacing

Μία από τις θεμελιώδεις διαφορές του 5G NR είναι η υποστήριξη πολλαπλών τιμών απόστασης υποφερόντων (subcarrier spacing), μια έννοια γνωστή ως numerology. Κάθε numerology συμβολίζεται με τον δείκτη μ (mu) και ορίζει την απόσταση υποφερόντων συχνοτήτων ως:

$$\Delta f = 15 \text{ kHz} \times 2^\mu$$

Για παράδειγμα:

- $\mu=0$ αντιστοιχεί σε 15 kHz (όπως στο LTE).
- $\mu=1$ αντιστοιχεί σε 30 kHz.
- $\mu=2$ αντιστοιχεί σε 60 kHz.
- $\mu=3$ αντιστοιχεί σε 120 kHz.
- $\mu=4$ αντιστοιχεί σε 240 kHz.

Αυτή η δυνατότητα επιλογής διαφορετικής υποφέρουσας απόστασης προσδίδει σημαντική ευελιξία στη χρονοδομή, επιτρέποντας στο 5G NR να προσαρμόζεται σε ποικίλες απαιτήσεις, όπως χαμηλή καθυστέρηση (π.χ. URLLC) ή υψηλή φασματική απόδοση (π.χ. eMBB), ανάλογα με την εφαρμογή.

2.2.3 Διάρκεια OFDM Συμβόλου και Slot ανά Numerology

Η επιλογή του numerology επηρεάζει άμεσα τη διάρκεια των OFDM συμβόλων και των slots. Με την αύξηση του numerology (δηλαδή, με μεγαλύτερο Δf), η διάρκεια του κάθε OFDM συμβόλου μειώνεται αντίστοιχα. Αυτό είναι μια εγγενής ιδιότητα του OFDM: μικρότερη περίοδος συμβόλου απαιτεί μεγαλύτερο εύρος ζώνης (ευρύτερη υποφέρουσα απόσταση) και αντιστρόφως.

Ενδεικτικά:

- Για $\Delta f = 15 \text{ kHz}$ ($\mu=0$), η διάρκεια ενός OFDM συμβόλου είναι περίπου 66,7 μs (χωρίς CP).
- Για $\Delta f = 240 \text{ kHz}$ ($\mu=4$), η διάρκεια μειώνεται σε περίπου 4,17 μs .

Αντίστοιχα, και η διάρκεια του cyclic prefix συρρικνώνεται (π.χ. από ~4,7 μs στο 15 kHz σε ~0,29 μs στο 240 kHz). Το συνολικό μήκος ενός συμβόλου (συμπεριλαμβανομένου του CP) πέφτει από περίπου 71,3 μs ($\mu=0$) σε μόλις περίπου 4,46 μs για $\mu=4$.

Καθώς κάθε slot περιέχει 14 σύμβολα (για normal CP), προκύπτει ότι και η διάρκεια του slot μειώνεται δραστικά για υψηλότερα numerologies:

- Για $\mu=0$, ένα slot διαρκεί περίπου 1 ms.
- Για $\mu=4$, ένα slot διαρκεί μόνο περίπου 0,0625 ms (62,5 μs).

Γενικά, ο διπλασιασμός της υποφέρουσας απόστασης συνεπάγεται υποδιπλασιασμό της διάρκειας συμβόλου και slot. Αυτό βελτιώνει την καθυστέρηση (latency) αλλά αυξάνει τον πλεονασμό (overhead) (περισσότερα slots και σύμβολα ανά σταθερό χρονικό παράθυρο).

2.2.4 Σταθερό Υποπλάισιο ως Μονάδα Αναφοράς

Παρά την ύπαρξη πολλαπλών numerologies, το 5G διατηρεί ως σταθερή χρονική μονάδα το υποπλάισιο των 1 ms – ακριβώς όπως στο LTE. Δηλαδή, ένα υποπλάισιο είναι πάντοτε 1 ms ανεξαρτήτως numerology. Αυτό παρέχει ένα κοινό χρονικό πλέγμα αναφοράς για τον προγραμματισμό και τη στοίχιση πλαισίων μεταξύ δικτύου και συσκευών.

Η ευελιξία εισάγεται όχι με αλλαγή της διάρκειας του subframe, αλλά μέσω της μεταβολής του αριθμού των slot μέσα στο 1 ms, ανάλογα με το numerology. Επιπλέον, ο αριθμός OFDM συμβόλων ανά slot παραμένει σταθερός για κάθε τύπο CP (14 για normal CP, 12 για extended CP) και δεν εξαρτάται από το numerology. Με αυτόν τον τρόπο, κάθε ραδιοπλάισιο 10 ms και κάθε υποπλάισιο 1 ms αρχίζει και

τελειώνει ταυτόχρονα σε όλο το δίκτυο, ανεξάρτητα από την επιλεγμένη υποφέρουσα απόσταση, διατηρώντας την ευθυγράμμιση των βασικών χρονικών ορίων.

2.2.5 Αριθμός Slot και Συμβόλων ανά Frame

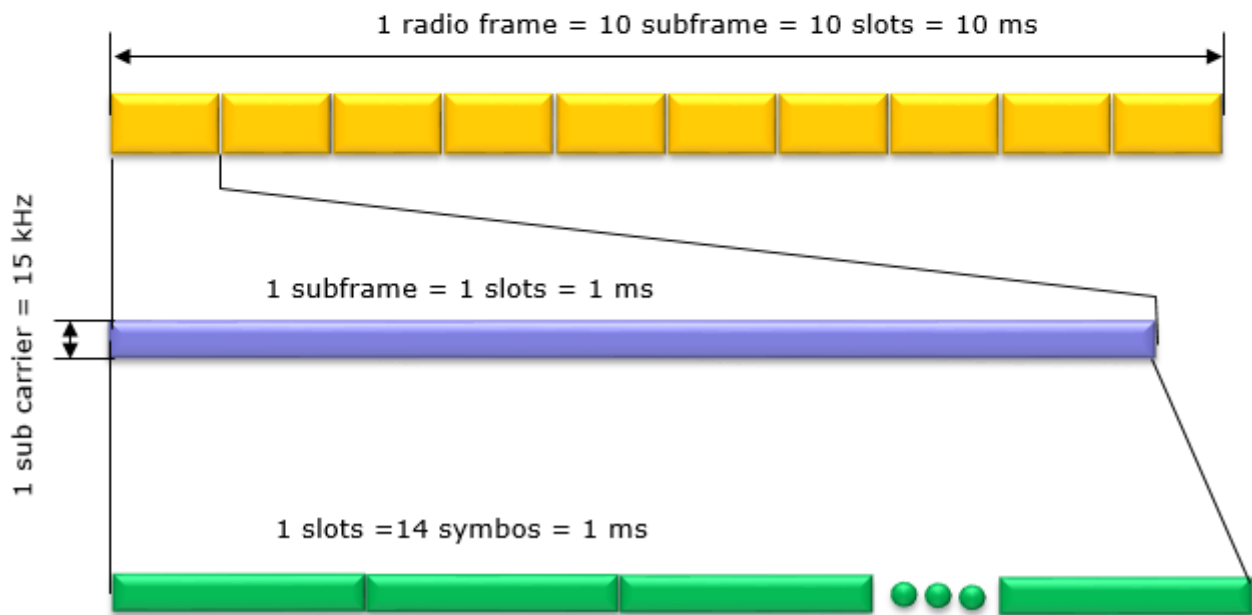
Δεδομένου ότι το subframe είναι σταθερά 1 ms, ο αριθμός των slot εντός ενός subframe αυξάνεται εκθετικά με το μ . Συγκεκριμένα, σε 1 ms χωρούν 2^μ slots (για normal CP), οπότε σε ένα πλήρες ραδιοπλαίσιο 10 ms υπάρχουν συνολικά $10 \times 2^\mu$ slots. Κάθε slot περιλαμβάνει 14 OFDM σύμβολα (normal CP), επομένως ο συνολικός αριθμός συμβόλων ανά πλαίσιο είναι $14 \times (\text{αριθμός slots})$.

Ακολουθούν ενδεικτικές τιμές:

- $\mu=0$ (15 kHz): 1 slot ανά υποπλαίσιο $\rightarrow$ 10 slots ανά 10 ms frame, με 14 σύμβολα το καθένα (σύνολο 140 σύμβολα/frame). Διάρκεια slot $\approx 1\text{ms}$.
- $\mu=1$ (30 kHz): 2 slots ανά υποπλαίσιο $\rightarrow$ 20 slots ανά frame, 14 σύμβολα το καθένα (280 σύμβολα/frame). Διάρκεια slot $\approx 0,5\text{ms}$.
- $\mu=2$ (60 kHz): 4 slots ανά υποπλαίσιο $\rightarrow$ 40 slots ανά frame, 14 σύμβολα το καθένα (560 σύμβολα/frame). Διάρκεια slot $\approx 0,25\text{ms}$. (Για $\mu=2$ υπάρχει επιλογή και extended CP: τότε κάθε slot έχει 12 σύμβολα αντί για 14.)
- $\mu=3$ (120 kHz): 8 slots ανά υποπλαίσιο $\rightarrow$ 80 slots ανά frame, 14 σύμβολα το καθένα (1.120 σύμβολα/frame). Διάρκεια slot $\approx 0,125\text{ms}$.
- $\mu=4$ (240 kHz): 16 slots ανά υποπλαίσιο $\rightarrow$ 160 slots ανά frame, 14 σύμβολα το καθένα (2.240 σύμβολα/frame). Διάρκεια slot $\approx 0,0625\text{ms}$.

< 38.211 – Table 4.3.2-1 >

μ	$N_{\text{slot}}^{\text{symbol}}$	$N_{\text{slot}}^{\text{frame}, \mu}$	$N_{\text{slot}}^{\text{subframe}, \mu}$
0	14	10	1
1	14	20	2
2	14	40	4
3	14	80	8
4	14	160	16



Εικόνα 4: OFDM παραμετροποίηση και διάρκεια υποπλαισίων και σχισμών.

Σχηματική απεικόνιση της χρονοδομής για numerology $\mu=0$ (subcarrier spacing 15 kHz). Κάθε κίτρινο μπλοκ αντιπροσωπεύει ένα υποπλαίσιο 1 ms, το οποίο για $\mu=0$ περιέχει 1 slot (μπλε) διάρκειας 1 ms. Κάθε slot με normal CP αποτελείται από 14 OFDM σύμβολα (πράσινα). Στο σχήμα βλέπουμε την επέκταση του ίδιου κανόνα σε υψηλότερα numerologies: όσο αυξάνεται το μ , διπλασιάζονται τα slot ανά 1 ms subframe (π.χ. 16 slot στο 1 ms για $\mu=4$, δηλ. 160 slot στο 10 ms πλαίσιο) ενώ ο αριθμός συμβόλων ανά slot παραμένει 14 (normal CP)

Αυτό επιβεβαιώνει ότι η μείωση της διάρκειας slot με μεγαλύτερο subcarrier spacing επιτρέπει περισσότερες χρονικές «φέτες» μέσα στο σταθερό πλαίσιο.

2.2.6 Ευέλικτη Διάταξη Downlink/Uplink σε TDD

Στη λειτουργία TDD, το NR εισάγει μεγάλη ευελιξία στον διαχωρισμό downlink (DL - κάτω ζεύξη – από σταθμό βάσης στο κινητό) και uplink (UL - άνω ζεύξη – αντίστροφη πορεία) εντός του πλαισίου, σε σύγκριση με το LTE. Στο LTE TDD, κάθε υποπλαίσιο ήταν είτε εξ ολοκλήρου DL είτε εξ ολοκλήρου

UL. Αντιθέτως, στο 5G NR, κάθε slot των 14 συμβόλων μπορεί να διαμορφωθεί ώστε μερικά σύμβολα να χρησιμοποιούνται για DL και άλλα για UL μέσα στο ίδιο slot.

Η 3GPP έχει προδιαγράψει ένα σύνολο 61 συνδυασμών (Slot Formats) που ορίζουν ποιες θέσεις συμβόλων σε ένα slot προορίζονται για downlink (D), ποιες για uplink (U), και ποιες μπορεί να είναι ευέλικτες (Flexible - F). Κάθε slot format περιγράφεται από μια ακολουθία 14 χαρακτήρων (αντιστοιχώντας στα 14 σύμβολα του slot). Παραδείγματος χάριν:

- “DDDDDDDDDDDDDDDD”, όλα τα 14 σύμβολα downlink, Format 0
- “UUUUUUUUUUUUUUUU”, όλα uplink, Format 1
- “DDDDDDDDDDDDDDDF”, 13 DL σύμβολα και 1 Flexible στο τέλος, Format 3
- “FFFFFFFFFFFFFFFU”, 13 Flexible και 1 UL στο τέλος, Format 8

Τα Flexible (F) σύμβολα μπορούν να χρησιμοποιηθούν είτε ως downlink είτε ως uplink ανάλογα με τις ανάγκες, ή να παραμείνουν κενά ως χρονικό περιθώριο προστασίας (guard). Έτσι, μέσω των slot formats, το δίκτυο μπορεί να ορίσει με λεπτομέρεια το εντός slot μοτίβο DL/UL.

Σε ανώτερο επίπεδο, ο διαχωρισμός DL/UL οργανώνεται μέσω ενός επαναλαμβανόμενου μοτίβου slot σε κάθε ραδιοπλαίσιο. Δεν υπάρχουν πλέον σταθερές προκαθορισμένες διαμορφώσεις όπως στο LTE TDD, αλλά το δίκτυο ορίζει δυναμικά το pattern με παραμέτρους που μεταδίδονται μέσω RRC. Συγκεκριμένα, η παράμετρος tdd-UL-DL-ConfigurationCommon περιγράφει το περιοδικό μοτίβο εναλλαγής DL/UL, ορίζοντας:

- Την περίοδο επανάληψης του μοτίβου (π.χ., 5 ms, 10 ms, 20 ms).
- Τον αριθμό διαδοχικών πλήρων DL slot στην αρχή κάθε περιόδου.
- Τον αριθμό DL συμβόλων στο επόμενο slot (μερικό DL slot).
- Τον αριθμό διαδοχικών πλήρων UL slot που ακολουθούν.
- Τον αριθμό UL συμβόλων στο slot πριν την επανέναρξη του κύκλου.

Με αυτές τις παραμέτρους, το δίκτυο μπορεί να καθορίσει ευέλικτα ένα μοτίβο (π.χ., 7 συνεχόμενα DL slot, μετά ένα μεικτό slot με DL και guard, και στη συνέχεια 2 UL slot). Το μοτίβο αυτό επαναλαμβάνεται περιοδικά. Η ευελιξία δύο επιπέδων – σε επίπεδο εντός slot (μέσω slot format) και μεταξύ slot (μέσω του επαναλαμβανόμενου μοτίβου) – δίνει στο 5G τη δυνατότητα να προσαρμόζει δυναμικά το duplexing ανάλογα με την κίνηση του δικτύου.

2.2.7 Εναλλαγή DL/UL και Guard Symbols

Επειδή ο πομπός και ο δέκτης (στην ίδια συσκευή) δεν μπορούν να λειτουργούν ταυτόχρονα στην ίδια συχνότητα, απαιτείται προσεκτικός χρονισμός στη μετάβαση από downlink σε uplink ώστε να μην υπάρξει επικάλυψη που θα προκαλούσε παρεμβολή. Μια θεμελιώδης αρχή είναι ότι οι εναλλαγές DL/UL πραγματοποιούνται μόνο σε όρια OFDM συμβόλων – ποτέ εντός ενός συμβόλου που εκπέμπεται. Αυτό αντικατοπτρίζεται στον σχεδιασμό των slot formats, όπου το σημείο αλλαγής από D σε U (ή αντίστροφα) συμπίπτει με τη λήξη ενός OFDM συμβόλου και την έναρξη του επόμενου, διασφαλίζοντας την ολοκλήρωση της μετάδοσης προς μία κατεύθυνση πριν την έναρξη της μετάδοσης προς την άλλη.

Επιπλέον, κατά τη μετάβαση από DL σε UL απαιτείται να παρεμβληθεί ένας μικρός κενός χρόνος – τα λεγόμενα guard symbols ή guard interval – ώστε να δοθεί χρόνος στους πομποδέκτες να εναλλάξουν

ρόλο. Συγκεκριμένα, το gNB (σταθμός βάσης) πρέπει να διακόψει την κάτω ζεύξη λίγες δεκάδες μ πριν αρχίσει η άνω ζεύξη. Αυτό δίνει χρόνο στα τερματικά χρήστη (User Equipment – UE) να λάβουν το τελευταίο DL σύμβολο και να προλάβουν να μεταβούν σε κατάσταση εκπομπής UL χωρίς να εκπέμψουν πρόωρα. Αυτό επιτυγχάνεται αφήνοντας 1 ή περισσότερα σύμβολα ως “Flexible/Null” στο τέλος ενός DL slot, λειτουργώντας ως guard period.

Αντίθετα, για τη μετάβαση από UL σε DL δεν απαιτείται αντίστοιχο guard gap. Μέσω του μηχανισμού timing advance (χρονική προπορεία), το gNB φροντίζει ώστε τα UL σήματα από τα UE να φτάνουν συγχρονισμένα ακριβώς πριν την έναρξη του επόμενου DL συμβόλου. Με άλλα λόγια, το UE σταματά να εκπέμπει ακριβώς εγκαίρως ώστε το δίκτυο να μπορέσει να αρχίσει να εκπέμπει το downlink στο επόμενο σύμβολο χωρίς κενό.

Συνοψίζοντας, ο σχεδιασμός Time Division Duplexing (TDD – Αμφίδρομη επικοινωνία στο χρόνο) του NR επιτρέπει ευέλικτη εναλλαγή DL/UL στο επίπεδο του συμβόλου, με ενσωματωμένα guard symbols όπου χρειάζεται, ώστε να αποφεύγεται η αλληλοπαρεμβολή. Ο κατάλληλος αριθμός διαδοχικών F συμβόλων (ή κενού χρόνου) για ασφάλεια επιλέγεται με βάση την απόσταση του UE (χρόνος διάδοσης) και τον χρόνο που χρειάζεται το UE να μεταβεί από λήψη σε εκπομπή.

2.2.8 Ειδικές Εκπομπές και Σύμβολα: SSB & SIB

Πέρα από τα κοινά δεδομένα, το NR ορίζει ορισμένες ειδικές εκπομπές αναφοράς στο φυσικό επίπεδο, όπως τα μπλοκ συγχρονισμού (Synchronization Signal Block - SSB) και τα μηνύματα συστήματος (System Information Blocks - SIB), τα οποία έχουν καθορισμένη θέση και περιοδικότητα μέσα στο πλαίσιο.

2.2.8.1 SSB (Synchronization Signal Block)

Το SSB είναι ένα κρίσιμο σύνολο συμβόλων που μεταδίδεται από το gNB για την αρχική σύζευξη και συγχρονισμό των UE με το δίκτυο. Κάθε SSB αποτελείται από 4 συνεχόμενα OFDM σύμβολα που μεταφέρουν τους συγχρονιστικούς σηματοδότες (πρωτεύον και δευτερεύον σήμα συγχρονισμού PSS, SSS) και το φυσικό κανάλι εκπομπής (Primary Broadcast Channel – PBCH), το οποίο περιέχει τις βασικές πληροφορίες της κυψέλης (Master Information Block – MIB).

Τα SSB μεταδίδονται περιοδικά, τυπικά κάθε 20 ms, ώστε τα υποψήφια UE να μπορούν να “ακούσουν” τον συγχρονισμό του κυψέλης. Η 3GPP επιτρέπει διάφορες επιλογές περιοδικότητας (5, 10, 20, 40, 80 ή 160 ms), όμως η προεπιλογή στα περισσότερα δίκτυα είναι 20 ms.

Η εκπομπή των SSB γίνεται σε ριπές (bursts) εντός παραθύρου 5 ms κάθε προγραμματισμένης περιοδικότητας. Δηλαδή, μέσα σε ένα χρονικό «παράθυρο» διάρκειας 5 ms, το gNB μπορεί να μεταδώσει πολλαπλά SSB (διαφορετικών κατευθύνσεων/δοκών) προκειμένου να επιτευχθεί κάλυψη μέσω beamforming. Το μέγιστο πλήθος SSB ανά burst είναι έως 4 για συχνότητες ≤ 3 GHz, έως 8 για 3–6 GHz, και έως 64 για mmWave > 6 GHz.

Το subcarrier spacing των SSB είναι καθορισμένο ανεξάρτητα από το numerology των υπόλοιπων καναλιών:

- Σε FR1 (Frequency Range 1, υπο-6 GHz) τυπικά χρησιμοποιείται SSB με 15 kHz (μέχρι περίπου 3 GHz) ή 30 kHz (3–6 GHz).
- Σε FR2 (Frequency Range 2, mmWave) χρησιμοποιείται 240 kHz.

Αυτό σημαίνει ότι τα SSB εκπέμπονται σε μια συγκεκριμένη κλίμακα χρόνου-συχνότητας που διευκολύνει τα UE να τα ανιχνεύσουν. Η θέση των SSB στο πλαίσιο είναι επίσης προκαθορισμένη: συνήθως εντοπίζονται σε συγκεκριμένα slots και σύμβολα εντός των πρώτων 5 ms κάθε 20 ms περιόδου, βάσει της εκάστοτε “περίπτωσης” (Case A, B, C, D, E) που ορίζει το 3GPP για διαφορετικές συχνότητες. Οι πληροφορίες αυτές κοινοποιούνται στο UE μέσω ειδικών масκών bit στο SIB1 (ssb-PositionsInBurst) ώστε το UE να γνωρίζει πού να αναζητήσει τα SSB.

2.2.8.2 SIB1 (System Information Block Type 1)

Μετά τον αρχικό συγχρονισμό μέσω SSB/MIB, το UE πρέπει να λάβει βασικές πληροφορίες συστήματος – αυτές παρέχονται στο SIB1. Το SIB1 είναι ένα μήνυμα ανώτερου στρώματος (Radio Resource Control - RRC) που μεταφέρεται από το φυσικό κανάλι PBCH/DL-SCH (Broadcast Channel) προς όλα τα UE.

Στο NR, το SIB1 μεταδίδεται επίσης περιοδικά αλλά αραιά: έχει περιοδικότητα περίπου 160 ms (δηλαδή, κάθε 160 ms αρχίζει ένας νέος κύκλος μετάδοσης SIB1). Μέσα σε αυτό το παράθυρο, το gNB μπορεί να επαναλαμβάνει το SIB1 αρκετές φορές ώστε να διασφαλιστεί ότι τα UE θα το λάβουν με αξιοπιστία.

Το SIB1 περιέχει κρίσιμες πληροφορίες για την πρόσβαση στην κυψέλη, όπως παραμέτρους για το PRACH (κανάλι τυχαίας προσπέλασης), χρονοδιαγράμματα εκπομπής άλλων SIB, τιμές χρονισμών, επιτρεπόμενες συχνότητες κ.λπ. Ουσιαστικά, μαζί με το MIB, αποτελούν το ελάχιστο σετ πληροφοριών συστήματος Minimum SI (System Information) που απαιτείται ώστε το UE να μπορέσει να “κατασκοπήσει” (camp) στο κελί.

Μια σημαντική διαφορά σε σχέση με το LTE είναι ότι στο NR, μόνο το SIB1 (και τυχόν άλλα SIB βασικής ανάγκης) μεταδίδεται περιοδικά, ενώ τα υπόλοιπα SIB μπορεί να στέλνονται κατά απαίτηση (on-demand). Δηλαδή, το δίκτυο δεν τα εκπέμπει συνεχώς αλλά κατόπιν αιτήματος του UE, προκειμένου να μειωθεί το περιττό overhead στο ασύρματο κανάλι. Αυτή η ευελιξία του NR στην μετάδοση πληροφοριών συστήματος σημαίνει ότι το ραδιόδίκτυο δεν καταναλώνει πόρους εκπέμποντας συνεχώς μεγάλα μηνύματα συστήματος, αλλά μόνο όταν χρειάζεται.

2.3 SIMO και MIMO Ασύρματα Συστήματα

Τα συστήματα SIMO (Single Input, Multiple Output) και MIMO (Multiple Input, Multiple Output) αξιοποιούν τη χρήση πολλαπλών κεραιών για την ενίσχυση της ποιότητας και της αξιοπιστίας της μετάδοσης. Στην αρχιτεκτονική SIMO, ένας πομπός επικοινωνεί με δέκτες με πολλαπλές κεραίες/RF αλυσίδες, προσφέροντας πλεονεκτήματα όπως ενίσχυση σήματος, μείωση της απόσβεσης από φαινόμενα σκίασης, και βελτίωση του λόγου σήματος προς θόρυβο (SNR).

Για την εκμετάλλευση αυτών των πλεονεκτημάτων, εφαρμόζονται τεχνικές συνδυασμού σημάτων στον δέκτη. Η πιο αποτελεσματική μέθοδος (βέλτιστη όσον αφορά το SNR) είναι το MRC, το οποίο συνδυάζει τα σήματα των διαφορετικών κεραιών με σταθμίσεις ανάλογες της ισχύος του καναλιού, μεγιστοποιώντας το τελικό SNR. Το MRC είναι ιδιαίτερα αποτελεσματικό σε περιβάλλοντα OFDM όπου η εκτίμηση του καναλιού γίνεται για κάθε υποφέρον ξεχωριστά.

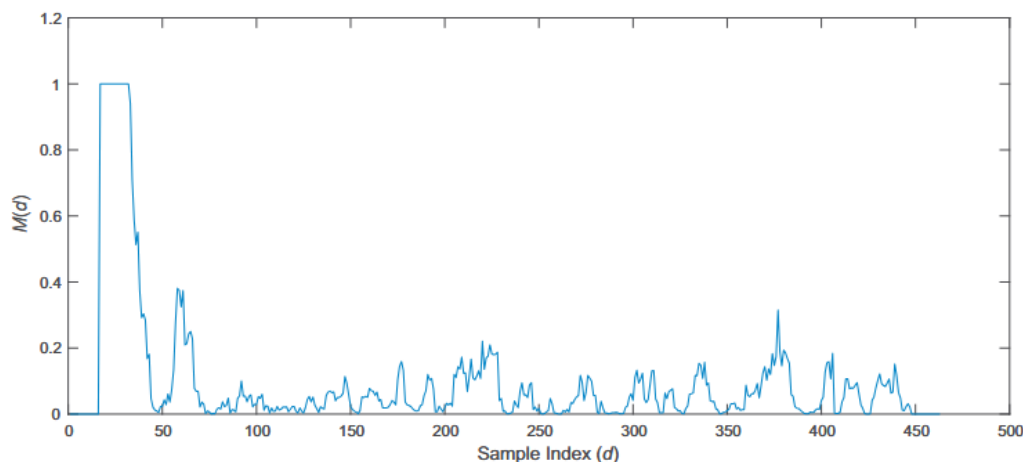
2.4 Τεχνικές Συγχρονισμού σε OFDM

Ο συγχρονισμός είναι από τις πιο κρίσιμες προκλήσεις σε συστήματα OFDM, καθώς απαιτείται ακρίβεια τόσο στον χρόνο όσο και στη συχνότητα. Ακόμη και μικρές χρονικές αποκλίσεις μπορούν να οδηγήσουν σε ISI, ενώ η συχνотική μετατόπιση διαταράσσει την ορθογωνικότητα των υποφερόντων, προκαλώντας ICI.

Για την αντιμετώπιση αυτών των προβλημάτων, χρησιμοποιούνται δύο βασικές τεχνικές:

- Zadoff-Chu ακολουθίες: Πολύ καλές αυτοσυσχετιζόμενες ακολουθίες που τοποθετούνται ως προοίμια σύμβολα - preamble στην αρχή του πλαισίου και επιτρέπουν ακριβή εντοπισμό του σημείου έναρξης. Είναι κυκλικές, έχουν σταθερή περιβάλλουσα και ελάχιστη διασυμβολική παρεμβολή.
- Schmidl & Cox synchronization: Βασίζεται σε ειδικά σχεδιασμένο preamble με επαναλαμβανόμενη δομή τύπου $[A \ A]$. Ο αλγόριθμος εντοπίζει τη θέση του πλαισίου υπολογίζοντας ένα κανονικοποιημένο μετρικό μεταξύ δύο διαδοχικών τμημάτων ίδιου περιεχομένου. Παράλληλα, επιτρέπει την εκτίμηση της ολίσθησης συχνότητας (Carrier Frequency Offset) CFO μέσω της διαφοράς φάσης μεταξύ των τμημάτων.

Η διαδικασία συγχρονισμού, σε συνδυασμό με την χρήση CP, επιτρέπει στο σύστημα να εντοπίζει αξιόπιστα την αρχή του πλαισίου και να αντιμετωπίζει τη χρονική αστάθεια του καναλιού. Στην υλοποίηση αυτής της εργασίας, ο συγχρονισμός αποτελεί κομβικό στάδιο πριν από την αποδιαμόρφωση.



Εικόνα 5: Μετρική χρονισμού Schmidl & Cox, σχήμα 16.21 από [3]

2.5 Εκτίμηση Καναλιού και Πιλοτικά Υποφέροντα

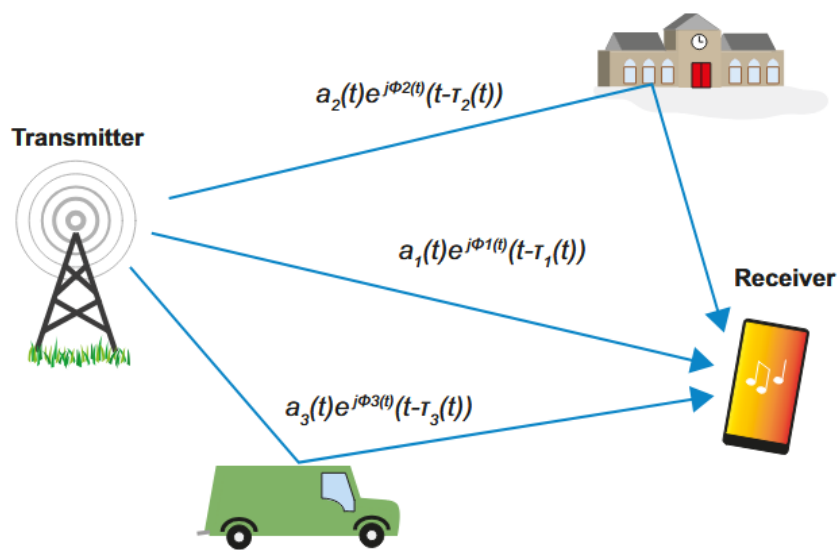
Η εκτίμηση του καναλιού σε OFDM συστήματα είναι απαραίτητη για την αντιστάθμιση της επίδρασης της διάδοσης. Δεδομένου ότι η διαμόρφωση πραγματοποιείται στο πεδίο της συχνότητας, η εκτίμηση πρέπει να γίνει ξεχωριστά για κάθε υποφέρον. Για τον σκοπό αυτό χρησιμοποιούνται πιλοτικά υποφέροντα, τα οποία μεταφέρουν γνωστά σύμβολα και χρησιμεύουν ως σημεία αναφοράς.

Η βασική μέθοδος εκτίμησης που χρησιμοποιείται είναι η Least Squares (LS), κατά την οποία η εκτίμηση του καναλιού προκύπτει ως το πηλίκο μεταξύ του λαμβανόμενου και του γνωστού συμβόλου

στο υποφέρον. Αν και είναι ευαίσθητη σε θόρυβο, η LS είναι απλή, υπολογιστικά ελαφριά και κατάλληλη για εφαρμογές πραγματικού χρόνου.

Για τα μη-πilotικά υποφέροντα εφαρμόζεται παρεμβολή (interpolation), συνήθως γραμμική ή spline, ώστε να προσεγγιστεί η μορφή του καναλιού μεταξύ των γνωστών σημείων. Σε SIMO συστήματα, η εκτίμηση πραγματοποιείται ξεχωριστά για κάθε RF αλυσίδα, και στη συνέχεια τα κανάλια συνδυάζονται βάσει του MRC.

Η πυκνότητα και η διάταξη των pilotικών subcarriers έχουν καθοριστική σημασία για την ακρίβεια εκτίμησης, ιδίως σε ταχέως μεταβαλλόμενα κανάλια ή περιβάλλοντα με πολύπλοκη φασματική απόκριση.



Εικόνα 6: Απεικόνιση ενός πολυδιαδρομικού καναλιού, σχήμα 16.2 από [3]

2.6 Κωδικοποίηση και Διόρθωση Σφαλμάτων (FEC)

Η κωδικοποίηση εμπρόσθιας διόρθωσης σφαλμάτων αποτελεί μία βασική τεχνική για την αξιόπιστη μετάδοση δεδομένων σε θορυβώδη και ασταθή ασύρματα κανάλια. Η FEC προσθέτει πλεονάζον περιεχόμενο στο μεταδιδόμενο σήμα, επιτρέποντας στον δέκτη να ανιχνεύει και να διορθώνει σφάλματα χωρίς την ανάγκη αναμετάδοσης. Στο πλαίσιο της παρούσας εργασίας, υλοποιείται συνελκτικός κώδικας (convolutional code), ο οποίος ενσωματώνει παλαιότερη χρονικά πληροφορία μέσω καταχωρητών ολίσθησης, διαμορφώνοντας έτσι ένα κώδικα με μνήμη.

Οι γεννήτριες του κώδικα ορίζονται από πολώνυμα σε δυαδική μορφή που αντιστοιχούν σε κώδικα με λόγο 1/2 και βάθος (constraint length) 7. Η αποκωδικοποίηση πραγματοποιείται με τον αλγόριθμο Viterbi, ο οποίος εντοπίζει τη διαδρομή ελαχίστου σφάλματος στο γράφημα καταστάσεων του κώδικα, αξιοποιώντας δυναμικό προγραμματισμό.

Ένα κρίσιμο ζήτημα κατά την αποκωδικοποίηση είναι η μορφή των εισόδων προς τον Viterbi αποκωδικοποιητή, η οποία καθορίζει το αν θα χρησιμοποιηθεί hard decision ή soft decision αποκωδικοποίηση:

- Hard-decision decoding: Στην περίπτωση αυτή, τα σύμβολα που λαμβάνονται (συνήθως μετά την αποδιαμόρφωση) μετατρέπονται άμεσα σε bits (0 ή 1) με χρήση κατωφλίου. Ο Viterbi αλγόριθμος λαμβάνει αυτές τις δυαδικές τιμές και υπολογίζει τις αποστάσεις Hamming για να βρει την πιθανότερη διαδρομή. Αν και είναι υπολογιστικά απλούστερη, η μέθοδος αυτή δεν αξιοποιεί την πληροφορία αξιοπιστίας του ληφθέντος συμβόλου και έχει χειρότερη απόδοση σε περιβάλλοντα με θόρυβο.
- Soft-decision decoding: Εδώ, κάθε ληφθέν σύμβολο μετατρέπεται σε λογαριθμική πιθανοφάνεια (Log-Likelihood Ratio - LLR), η οποία αποτυπώνει το βαθμό βεβαιότητας για το αν το bit είναι 0 ή 1. Αντί για Hamming distances, ο Viterbi υπολογίζει ευκλείδειες αποστάσεις ή συσσωρευμένα μετρικά βασισμένα στην αξιοπιστία. Η προσέγγιση αυτή απαιτεί περισσότερους υπολογιστικούς πόρους και μεγαλύτερη ακρίβεια αριθμητικής αναπαράστασης, αλλά παρέχει σημαντικά καλύτερες επιδόσεις BER, ιδιαίτερα σε περιπτώσεις χαμηλού SNR.

Στην παρούσα εργασία εφαρμόζεται soft-decision αποκωδικοποίηση, καθώς ο OFDM δέκτης διαθέτει εκτιμήσεις καναλιού και ισοσταθμισμένα σύμβολα, από τα οποία μπορούν να εξαχθούν οι λογαριθμικές πιθανοφάνειες (LLRs) για κάθε bit. Αυτές οι LLRs χρησιμοποιούνται ως είσοδος στον Viterbi decoder, προσφέροντας καλύτερη ανοχή σε θόρυβο και αυξημένη αξιοπιστία χωρίς σημαντική αύξηση της πολυπλοκότητας, δεδομένης της offline επεξεργασίας.

2.7 Αρχιτεκτονική του Xilinx RFSoc 4x2

Η πλατφόρμα Xilinx Zynq UltraScale+ RFSoc 4x2 αποτελεί μία από τις πλέον καινοτόμες υλοποιήσεις για εφαρμογές Software-Defined Radio και τηλεπικοινωνιακά συστήματα φυσικού επιπέδου (PHY). Συνδυάζει σε ένα ενιαίο ολοκληρωμένο κύκλωμα τόσο αναλογικά στοιχεία (DAC/ADC) όσο και ψηφιακή επεξεργαστική ισχύ, παρέχοντας μια υψηλής απόδοσης, χαμηλής καθυστέρησης λύση για υλοποιήσεις σε πραγματικό χρόνο.

Ο πυρήνας του RFSoc βασίζεται στην αρχιτεκτονική Zynq UltraScale+, η οποία διαχωρίζεται σε δύο βασικά υποσυστήματα:

- Το Processing System, το οποίο περιλαμβάνει τετραπύρρηνο ARM Cortex-A53 επεξεργαστή, Cortex-R5 real-time επεξεργαστή και μονάδα ελέγχου μνήμης DDR4. Το PS είναι κατάλληλο για εκτέλεση λογισμικού ελέγχου, διασύνδεσης και προγραμματισμού υψηλού επιπέδου.
- Το Programmable Logic, το οποίο αποτελείται από προγραμματιζόμενη λογική τύπου FPGA, ενσωματωμένα DSP slices, block RAMs και δυνατότητα υψηλής παραλληλίας για υλοποίηση αλγορίθμων σε χαμηλό επίπεδο.

Το RFSoc 4x2 διαφοροποιείται από άλλες SDR πλατφόρμες με την ενσωμάτωση μετατροπέων υψηλής ταχύτητας:

1. 8 DACs (Digital-to-Analog Converters) ανάλυσης 14-bit, με ταχύτητα έως 9,85 GS/s.
2. 4 ADCs (Analog-to-Digital Converters) ανάλυσης 14-bit, με ταχύτητα έως 5 GS/s.
3. Οι μετατροπείς είναι απευθείας προσβάσιμοι από το PL και υποστηρίζουν συγχρονισμένη δειγματοληψία, γεγονός που τους καθιστά κατάλληλους για MIMO και SIMO εφαρμογές.

Για τις ανάγκες της παρούσας εργασίας η επικοινωνία με το υλικό πραγματοποιείται μέσω του RFSoc-MTS overlay, το οποίο προσφέρει προκαθορισμένες υποδομές για:

- Μεταφορά δεδομένων μέσω Direct Memory Access από το PS προς τον DAC (έξοδος) και από τον ADC (είσοδος) προς το PS.
- Trigger-based καταγραφή σημάτων, επιτρέποντας τη σύλληψη του σήματος ακριβώς τη χρονική στιγμή που ξεκινά η κυματομορφή, μειώνοντας τα φαινόμενα χρονικής μετατόπισης.
- Πλήρη έλεγχο μέσω Python, χωρίς την ανάγκη συγγραφής RTL λογικής ή χρήση εργαλείων όπως το Vivado HDL.

Η χρήση του RFSoc, σε αντίθεση με πιο παραδοσιακές SDR πλατφόρμες (π.χ. USRP, LimeSDR), προσφέρει πολύ μικρότερη καθυστέρηση, ευρύτερο εύρος ζώνης, και τη δυνατότητα άμεσης ενσωμάτωσης αλγορίθμων χαμηλού επιπέδου στην PL. Αυτό καθιστά την πλατφόρμα ιδιαίτερα κατάλληλη για πειραματικές δοκιμές συστημάτων OFDM, 5G NR και μελλοντικών προδιαγραφών 6G, όπως και για χρήση σε real-time εφαρμογές γνωστικών ραδιοσυστημάτων (cognitive radio).

2.8 RFSoc MTS Overlay

Η υλοποίηση της φυσικής διαστρωμάτωσης (PHY layer) στην πλατφόρμα Xilinx RFSoc 4x2 απαιτεί αποτελεσματική πρόσβαση στους μετατροπείς DAC/ADC, καθώς και ακριβή συγχρονισμό κατά την αποστολή και λήψη κυματομορφών. Για αυτόν τον σκοπό, χρησιμοποιείται το RFSoc-MTS (Multi-Tile Synchronization) overlay, το οποίο παρέχει μια ολοκληρωμένη υποδομή για τη διασύνδεση μεταξύ του ενσωματωμένου επεξεργαστή και των RF υποσυστημάτων της πλατφόρμας, μέσω ενός υψηλού επιπέδου Python API.

2.8.1 Αρχιτεκτονική και Λειτουργικότητα

Το mts.bit overlay φορτώνεται δυναμικά μέσω του περιβάλλοντος PYNQ, χρησιμοποιώντας τη Python βιβλιοθήκη rfsoc_mts. Το συνοδευτικό αρχείο mts.hwh παρέχει τις απαραίτητες πληροφορίες υλικού για τη δημιουργία των "bindings" προς τα IP blocks (Intellectual Property blocks) του bitstream π.χ., XRFDC για τους RF μετατροπείς, AXI DMA για τις μεταφορές δεδομένων.

Η βασική λειτουργικότητα του RFSoc-MTS overlay περιλαμβάνει:

- Πρόσβαση στους RF μετατροπείς (XRFDC) μέσω Python: Αυτό επιτρέπει την πλήρη παραμετροποίηση των DAC και ADC tiles. Μπορούμε να ρυθμίσουμε τη συχνότητα δειγματοληψίας και να ενεργοποιήσουμε ή να απενεργοποιήσουμε τα επιμέρους κανάλια, παρέχοντας ευελιξία στον έλεγχο των RF λειτουργιών.
- DMA Streaming (AXI DMA): Παρέχει έναν αποτελεσματικό μηχανισμό για τη μεταφορά κυματομορφών. Δεδομένα μπορούν να μεταφερθούν από τη μνήμη προς τον DAC για εκπομπή, και αντιστρόφως, από τον ADC στη μνήμη για περαιτέρω επεξεργασία. Αυτή η μεταφορά

γίνεται με χρήση buffer-based μηχανισμού και υποστηρίζει συλλογή με σκανδαλισμό (triggered capture), που είναι κρίσιμο για χρονικά ευαίσθητες εφαρμογές.

- **Trigger Synchronization:** Αυτή η λειτουργία επιτρέπει τη στοχευμένη έναρξη καταγραφής από τον ADC ή μετάδοσης στον DAC. Χρησιμοποιεί ένα εξωτερικό ή software trigger, εξασφαλίζοντας ακριβή ευθυγράμμιση στο χρόνο και αποφυγή μετατοπίσεων (offsets) ή aliasing, ώστε να καταγράφεται ή να εκπέμπεται το επιθυμητό μέρος του σήματος.
- **Tile Synchronization:** Αποτελεί ένα κρίσιμο στοιχείο για SIMO/MIMO αρχιτεκτονικές. Εξασφαλίζει την ταυτόχρονη ευθυγράμμιση των χρονισμών μεταξύ διαφορετικών RF tiles, κάτι που είναι απαραίτητο για την αξιόπιστη λειτουργία συστημάτων με πολλαπλές κεραίες.

2.8.2 Πλεονεκτήματα στη Συγκεκριμένη Εφαρμογή

Το RFSoc-MTS overlay αποτέλεσε βασικό τεχνολογικό υπόβαθρο της παρούσας υλοποίησης, προσφέροντας σημαντικά πλεονεκτήματα:

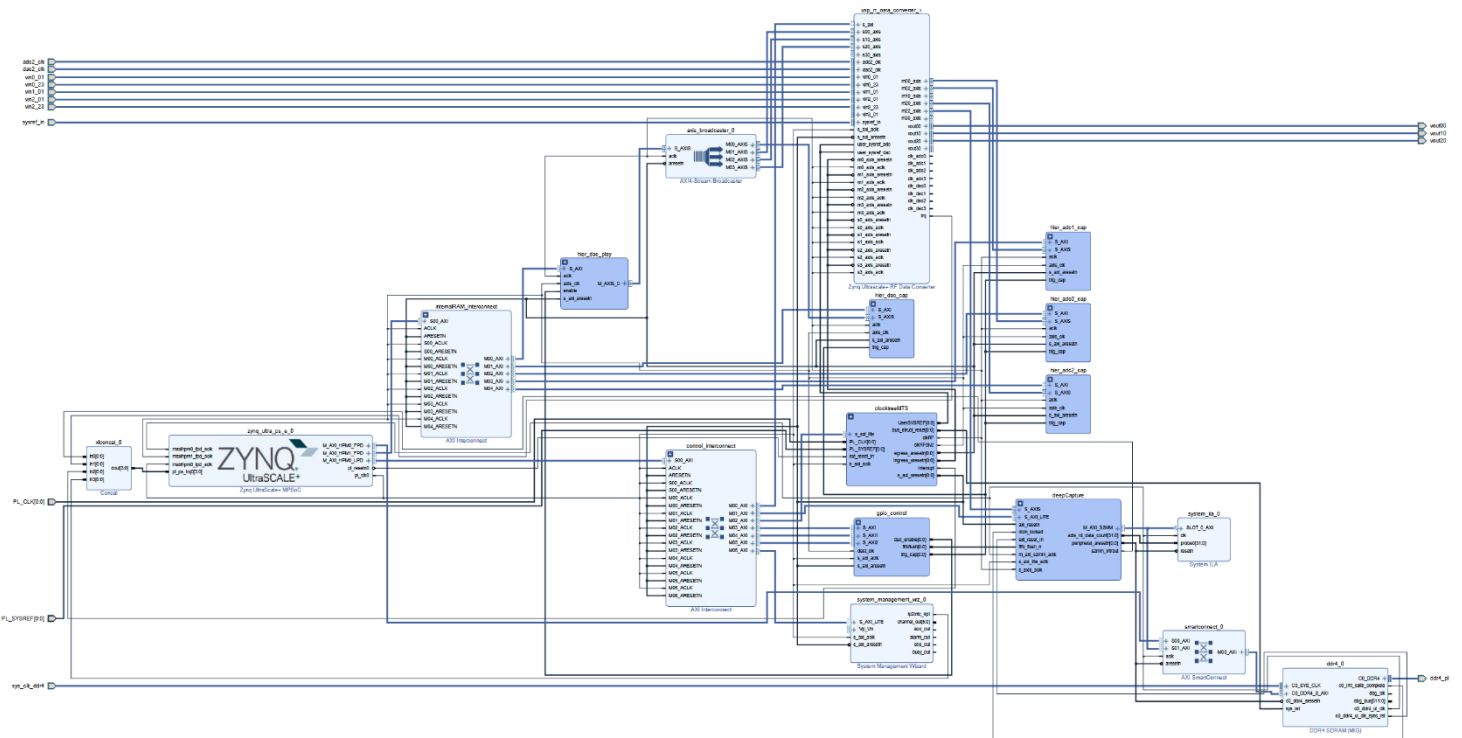
- **Πλήρης έλεγχος των RF blocks μέσω Python:** Αυτό εξαλείφει την ανάγκη για σχεδίαση σε HDL, επιταχύνοντας σημαντικά τον κύκλο ανάπτυξης και πρωτοτυποποίησης.
- **Υποστήριξη triggered capture:** Είναι απαραίτητο για την ακριβή χρονική ευθυγράμμιση του σήματος κατά τη λήψη, ειδικά σε συστήματα που βασίζονται σε ακολουθίες συγχρονισμού.
- **Υποστήριξη πολλαπλών ADC tiles:** Επιτρέπει τη λήψη από συστήματα SIMO ή MIMO διαμορφώσεων, κάτι που είναι κρίσιμο για την υλοποίηση της χωρικής λήψης.
- **Δυνατότητα debugging & visualization:** Με τη χρήση εργαλείων Python (όπως NumPy, Plotly κ.λπ.), είναι δυνατή η παραμετροποίηση και η ανάλυση σφαλμάτων σε πραγματικό χρόνο, προσφέροντας πολύτιμες διαγνωστικές πληροφορίες.

2.8.3 Περιορισμοί και Επεκτασιμότητα

Παρότι αποτελεί ένα ισχυρό εργαλείο πρωτοτυποποίησης, το RFSoc-MTS overlay συνοδεύεται από ορισμένους περιορισμούς:

- **Software-based DSP επεξεργασία:** Όλη η επεξεργασία ψηφιακού σήματος πραγματοποιείται στον ARM επεξεργαστή. Αυτό περιορίζει τις δυνατότητες real-time επεξεργασίας σε εφαρμογές υψηλού ρυθμού δεδομένων ή χαμηλής καθυστέρησης, καθώς η FPGA λογική δεν αξιοποιείται για τις DSP λειτουργίες.
- **Διαχείριση buffer σε DDR μνήμη:** Η χρήση buffer στη DDR μνήμη απαιτεί προσεκτική διαχείριση πόρων. Αυτό μπορεί να περιορίσει την ταυτόχρονη υποστήριξη πολύπλοκων ροών δεδομένων ή συνεχούς καταγραφής για μεγάλα χρονικά διαστήματα.

Παρόλα αυτά, το RFSoc-MTS προσφέρει μια ευέλικτη και εύχρηστη πλατφόρμα για την υλοποίηση και αξιολόγηση PHY επιπέδου OFDM συστημάτων. Αποτέλεσε κρίσιμο εργαλείο για την επιτυχία της παρούσας διπλωματικής εργασίας, επιτρέποντας την πρακτική διερεύνηση των αρχών των ασύρματων επικοινωνιών σε ένα πραγματικό περιβάλλον υλικού.



Εικόνα 7: RFSoc-MTS Block Diagram

3

Μοντελοποίηση του Συστήματος σε

Περιβάλλον Python

Η παρούσα ενότητα περιγράφει τη μοντελοποίηση του OFDM πομποδέκτη σε περιβάλλον Python, με σκοπό τη δημιουργία ενός πλήρως λειτουργικού συστήματος φυσικού επιπέδου για μετάδοση και λήψη συμβόλων σε διαμορφωμένο πολυδιαδρομικό κανάλι. Η υλοποίηση περιλαμβάνει όλες τις βασικές συνιστώσες ενός σύγχρονου ασύρματου OFDM SDR συστήματος: από τη διαμόρφωση και την προσθήκη CP, μέχρι τον συγχρονισμό, την εκτίμηση καναλιού, την αποκωδικοποίηση και τον υπολογισμό μετρικών απόδοσης. Το σύστημα υποστηρίζει αρχιτεκτονική SIMO (1x2), αξιοποιεί FEC κωδικοποίηση, και μοντελοποιεί το κανάλι βάσει του προτύπου 3GPP TDL-D, προσφέροντας ρεαλιστική αναπαράσταση περιβάλλοντος διάδοσης.

3.1 Γενική Αρχιτεκτονική Πομποδέκτη

Το προσομοιωμένο σύστημα OFDM που αναπτύχθηκε στην εργασία αυτή έχει δομηθεί σε περιβάλλον Python, με στόχο τη δημιουργία ενός ολοκληρωμένου πομποδέκτη που προσομοιώνει με ακρίβεια τις διαδικασίες μετάδοσης και λήψης σε ένα ρεαλιστικό πολυδιαδρομικό κανάλι. Η σχεδίαση περιλαμβάνει όλα τα απαραίτητα στάδια που απαιτούνται σε ένα σύγχρονο τηλεπικοινωνιακό σύστημα ασύρματης μετάδοσης, από τη διαμόρφωση και την κωδικοποίηση, έως τη λήψη, τον συγχρονισμό, την εκτίμηση του καναλιού και την αποκωδικοποίηση.

Η γενική αρχιτεκτονική του πομποδέκτη ακολουθεί τη λογική ενός OFDM συστήματος με αρχιτεκτονική SIMO και ενσωματώνει αναλυτικά όλα τα στάδια επεξεργασίας. Συγκεκριμένα, ο πομπός αποτελείται από τη δημιουργία τυχαίων δυαδικών δεδομένων, την κωδικοποίηση διόρθωσης σφαλμάτων (FEC) με τη χρήση συνελκτικού κώδικα rate 1/2, τη χαρτογράφηση των κωδικοποιημένων bits σε σύμβολα 16-QAM, και την εφαρμογή του αντίστροφου μετασχηματισμού Fourier για τη δημιουργία της OFDM κυματομορφής στο πεδίο του χρόνου. Μετά την εφαρμογή της IFFT, το σύστημα προσθέτει κυκλικό πρόθεμα για αντιμετώπιση των επιδράσεων της πολυδιαδρομικότητας.

Πριν την αποστολή του κύριου payload, προσαρτάται στο σήμα ένα προκαθορισμένο προοίμιο, το οποίο βασίζεται σε μια ακολουθία Zadoff-Chu, ειδικά σχεδιασμένη ώστε να διευκολύνει τη διαδικασία αρχικού συγχρονισμού στον δέκτη.

Η μετάδοση του σήματος πραγματοποιείται στο περιβάλλον προσομοίωσης μέσω ενός ρεαλιστικού μοντέλου καναλιού, που προσομοιώνει τη μετάδοση από μία κεραία πομπού προς δύο κεραίες λήψης. Στον δέκτη, το σύστημα καταγράφει τα λαμβανόμενα δείγματα και εκτελεί μια σειρά από διαδικασίες για την ανάκτηση των δεδομένων. Αρχικά, πραγματοποιεί τον χρονικό και συχνοτικό συγχρονισμό αξιοποιώντας τις μεθόδους Schmidl & Cox και Zadoff-Chu, ενώ στη συνέχεια προχωρά στην εκτίμηση του καναλιού μέσω της μεθόδου ελαχίστων τετραγώνων (Least Squares – LS), αξιοποιώντας πιλοτικά υποφέροντα και παρεμβολή για την κάλυψη όλων των υποφερόντων.

Ακολουθεί η εφαρμογή ισοστάθμισης μέσω της τεχνικής MRC, η οποία εκμεταλλεύεται τη χρήση των δύο κεραίων λήψης, βελτιστοποιώντας την αναλογία σήματος προς θόρυβο. Μετά την ισοστάθμιση, το σύστημα πραγματοποιεί αποδιαμόρφωση των συμβόλων QAM σε LLR bits, οι οποίες τροφοδοτούν έναν Viterbi αποκωδικοποιητή με soft-decision είσοδο. Με τον τρόπο αυτό επιτυγχάνεται αυξημένη αξιοπιστία και σημαντική βελτίωση στην απόδοση σε σχέση με τον συμβατικό hard-decision αποκωδικοποιητή.

Το αποτέλεσμα της αποκωδικοποίησης συγκρίνεται με τα αρχικά bits που παρήχθησαν στον πομπό, επιτρέποντας την εξαγωγή χρήσιμων μετρικών απόδοσης, όπως είναι το Bit Error Rate και το Error Vector Magnitude.

Η αρχιτεκτονική που αναπτύχθηκε στην παρούσα προσομοίωση έχει σχεδιαστεί με γνώμονα τη μέγιστη συμβατότητα και ακρίβεια σε σχέση με την υλοποίηση του συστήματος σε πραγματικό υλικό, και συγκεκριμένα στην πλατφόρμα RFSoc 4x2 της Xilinx, η οποία περιγράφεται εκτενώς στο επόμενο κεφάλαιο.

3.2 Παραγωγή Συμβόλων και FEC Κωδικοποίηση

Το πρώτο βήμα της διαδικασίας μετάδοσης στο προσομοιωμένο OFDM σύστημα είναι η παραγωγή των δεδομένων που θα μεταδοθούν. Στην προσομοίωση αυτή, δημιουργείται αρχικά μία τυχαία δυαδική ακολουθία, η οποία προορίζεται ως πληροφοριακό περιεχόμενο (information bits). Το μήκος αυτής της ακολουθίας καθορίζεται από τον αριθμό των υποφερόντων που είναι διαθέσιμα για δεδομένα, το πλήθος των bits ανά σύμβολο QAM (16-QAM: 4 bits ανά σύμβολο), και τον επιλεγμένο λόγο κωδικοποίησης FEC (rate = 1/2).

Η διαδικασία της κωδικοποίησης διόρθωσης σφαλμάτων αποτελεί ένα ιδιαίτερα σημαντικό βήμα, καθώς προσθέτει πλεονασμό στα δεδομένα, επιτρέποντας τη διόρθωση σφαλμάτων στον δέκτη. Στην υλοποίηση αυτή, χρησιμοποιείται ένας συνελκτικός κώδικας με λόγο κωδικοποίησης 1/2. Η επιλογή αυτή παρέχει ισχυρή προστασία έναντι σφαλμάτων, με το κόστος της αύξησης του απαιτούμενου εύρους ζώνης λόγω της υψηλότερης ποσότητας δεδομένων που μεταδίδονται.

Πιο συγκεκριμένα, η κωδικοποίηση χρησιμοποιεί δύο πολυώνυμα γεννήτριας που περιγράφονται ως εξής:

$$G1 = 1111001, G2 = 1011011$$

σε δυαδική μορφή. Το μήκος περιορισμού του συνελκτικού κώδικα είναι $K_c = 7$, και η διαδικασία κωδικοποίησης υλοποιείται μέσω της Python βιβλιοθήκης `sk_dsp_comm`. Τα δεδομένα κωδικοποιούνται ξεκινώντας από μια μηδενική αρχική κατάσταση (`initial_state`), προκειμένου να εξασφαλιστεί η συνέπεια και η δυνατότητα αναπαραγωγής των αποτελεσμάτων.

Μετά την κωδικοποίηση, τα κωδικοποιημένα bits ομαδοποιούνται σε σύνολα των τεσσάρων, καθώς η διαμόρφωση που έχει επιλεγεί είναι 16-QAM, που περιλαμβάνει δηλαδή 4 bits ανά σύμβολο. Η αντιστοίχιση των bits σε σύμβολα QAM γίνεται μέσω προκαθορισμένου αστερισμού που περιλαμβάνει 16 σημεία στο επίπεδο I/Q, όπου κάθε ομάδα τεσσάρων bits αντιστοιχίζεται σε ένα συγκεκριμένο σύμβολο. Η χαρτογράφηση υλοποιείται μέσω της χρήσης Gray coding, ώστε να ελαχιστοποιείται η πιθανότητα σφάλματος ανά bit σε περίπτωση μικρών αποκλίσεων του λαμβανόμενου συμβόλου από την ιδανική θέση του.

Εκτός από τα δεδομένα, εισάγονται στο πλαίσιο μετάδοσης και πιλοτικά σύμβολα. Τα πιλοτικά αυτά σύμβολα είναι γνωστά στον δέκτη και χρησιμοποιούνται για την εκτίμηση του καναλιού. Στην παρούσα υλοποίηση, η κατανομή των πιλοτικών γίνεται με ομοιόμορφη κατανομή και απόσταση στο φάσμα των υποφερόντων, εξασφαλίζοντας αποτελεσματική εκτίμηση καναλιού μέσω παρεμβολής (interpolation) στο δέκτη. Οι πιλοτικές τιμές επιλέγονται τυχαία από προκαθορισμένα σύνολα τιμών της διαμόρφωσης QAM, ώστε να προσομοιώνουν ρεαλιστικά τις πραγματικές συνθήκες μετάδοσης.

Αφού ολοκληρωθεί η διαδικασία κωδικοποίησης και διαμόρφωσης, τα σύμβολα δεδομένων και τα πιλοτικά σύμβολα τοποθετούνται στα αντίστοιχα υποφέροντα του OFDM πλαισίου, και εφαρμόζεται αντίστροφος γρήγορος μετασχηματισμός Fourier για τη δημιουργία της OFDM κυματομορφής στο πεδίο του χρόνου. Μετά την IFFT, προστίθεται το κυκλικό πρόθεμα. Τέλος, το σήμα ενισχύεται και μετατρέπεται σε interleaved μορφή I/Q πριν την αποστολή του.

Η παραπάνω διαδικασία παραγωγής και κωδικοποίησης των συμβόλων αποτελεί τη βάση για τη μετέπειτα αξιολόγηση της απόδοσης του συστήματος και για τη σύγκριση με πραγματικές μετρήσεις που θα προκύψουν από το RFSoc hardware.

3.3 Προσομοίωση SIMO Καναλιού

Το σύστημα OFDM που προσομοιώνεται στην εργασία αυτή περιλαμβάνει μετάδοση με μία κεραία πομπού και λήψη με δύο κεραίες. Η συγκεκριμένη αρχιτεκτονική επιλέχθηκε ώστε να αξιοποιηθεί η δυνατότητα χωρικής διαφοροποίησης που προσφέρει η χρήση πολλαπλών κεραιών στον δέκτη, βελτιώνοντας την αξιοπιστία του σήματος σε περιβάλλον πολυδιαδρομικότητας.

Η προσομοίωση του καναλιού βασίζεται στο πρότυπο 3GPP TDL-D, όπως φαίνεται και στον Πίνακα 3.1, που αναπαριστά με ακρίβεια τις χρονικές καθυστερήσεις, την ισχύ των διαδρομών και τον τύπο κατανομής fading κάθε διαδρομής. Σύμφωνα με το μοντέλο αυτό, η πρώτη διαδρομή αντιστοιχεί στη συνιστώσα οπτικής επαφής (Line-of-Sight - LOS), η οποία χαρακτηρίζεται από κατανομή Rician με K-factor 13.3 dB, ενώ οι υπόλοιπες διαδρομές ακολουθούν κατανομή Rayleigh.

Tap	Κανονικοποιημένη καθυστέρηση	Ισχύς (dB)	Κατανομή Fading
1	0	-0.2	LOS (Rician)

2	0.035	-13.5	Rayleigh
3	0.612	-18.8	Rayleigh
4	1.363	-22.8	Rayleigh
5	1.405	-17.9	Rayleigh
6	1.804	-20.1	Rayleigh
7	2.596	-21.9	Rayleigh
8	1.775	-22.9	Rayleigh
9	4.042	-27.8	Rayleigh
10	7.937	-23.6	Rayleigh
11	9.424	-24.8	Rayleigh
12	9.708	-30.0	Rayleigh
13	12.525	-27.7	Rayleigh

Πίνακας 3.1: Προδιαγραφές του καναλιού TDL-D (3GPP).

Η προσομοίωση πραγματοποιείται ως εξής:

1. Το διαμορφωμένο σήμα OFDM (συμπεριλαμβανομένου του preamble και των συμβόλων δεδομένων) περνά από το παραπάνω κανάλι μέσω συνέλιξης στο πεδίο του χρόνου, ώστε να προσομοιωθεί η επίδραση των πολλαπλών διαδρομών. Εφόσον χρησιμοποιείται κυκλικό πρόθεμα, η συνέλιξη ισοδυναμεί με κυκλική συνέλιξη.
2. Στην πρώτη διαδρομή εφαρμόζεται κατανομή fading τύπου Rician, σύμφωνα με τον ορισμένο K-factor (13.3 dB), αναπαριστώντας την παρουσία ισχυρής απευθείας διαδρομής (LOS path).
3. Οι υπόλοιπες διαδρομές επηρεάζονται από κατανομή Rayleigh, με διαφορετικά επίπεδα ισχύος και καθυστέρησης, προκαλώντας σημαντική διακύμανση στο λαμβανόμενο σήμα λόγω της πολυδιαδρομικότητας.
4. Μετά την εφαρμογή της επίδρασης του καναλιού, προστίθεται ανεξάρτητος προσθετικός λευκός γκαουσιανός θόρυβος (AWGN) σε κάθε ένα από τα δύο λαμβανόμενα σήματα, με επίπεδο που καθορίζεται σύμφωνα με τον επιθυμητό λόγο σήματος προς θόρυβο (SNR).
5. Τέλος, εισάγεται μια μικρή τυχαία συχνοτική μετατόπιση (Carrier Frequency Offset - CFO), διαφορετική σε κάθε ληπτική κεραία, προσομοιώνοντας με ακρίβεια τις πραγματικές συνθήκες λειτουργίας ασύρματων συστημάτων, στα οποία οι ταλαντωτές πομπού και δέκτη δεν είναι απόλυτα συγχρονισμένοι.

Τα δύο λαμβανόμενα σήματα αποθηκεύονται ως ξεχωριστές ακολουθίες δεδομένων (μυγαδικά arrays), ώστε να είναι διαθέσιμα για περαιτέρω επεξεργασία από τις μεθόδους συγχρονισμού, εκτίμησης καναλιού, ισοστάθμισης και αποκωδικοποίησης που θα περιγραφούν στις επόμενες ενότητες του κεφαλαίου. Η λεπτομερής προσομοίωση του SIMO καναλιού, βασισμένη στο συγκεκριμένο μοντέλο TDL-D, εξασφαλίζει ότι η αξιολόγηση του συστήματος OFDM πραγματοποιείται υπό συνθήκες που προσεγγίζουν εξαιρετικά στενά τις πραγματικές συνθήκες διάδοσης ενός σύγχρονου ασύρματου περιβάλλοντος.

3.4 Συγχρονισμός και Εκτίμηση Συχνοτικής Μετατόπισης (CFO)

Η διαδικασία του συγχρονισμού αποτελεί ιδιαίτερα κρίσιμο στάδιο στην αλυσίδα επεξεργασίας ενός OFDM συστήματος, καθώς ακόμα και μικρές αποκλίσεις στον χρονισμό ή στη συχνότητα μπορούν να οδηγήσουν σε σοβαρή υποβάθμιση των επιδόσεων λόγω της διασυμβολικής παρεμβολής και της παρεμβολής μεταξύ των υποφερόντων. Στην προσομοίωση που αναπτύχθηκε στην παρούσα εργασία, η διαδικασία συγχρονισμού πραγματοποιείται σε δύο ξεχωριστά αλλά συμπληρωματικά στάδια: τον αρχικό, χονδρικό συγχρονισμό μέσω ακολουθίας Zadoff-Chu, και τον λεπτομερή συγχρονισμό με χρήση της μεθόδου Schmidl & Cox.

Το αρχικό στάδιο του συγχρονισμού επιτυγχάνεται μέσω της αποστολής ενός προοιμίου που αποτελείται από μία ειδικά σχεδιασμένη ακολουθία τύπου Zadoff-Chu. Η ακολουθία αυτή διαθέτει εξαιρετικές ιδιότητες αυτοσυσχέτισης, επιτρέποντας στον δέκτη να προσδιορίσει τη θέση έναρξης του πλαισίου στο λαμβανόμενο σήμα. Η ανίχνευση γίνεται μέσω της εύρεσης της μέγιστης τιμής της συσχέτισης του λαμβανόμενου σήματος με το γνωστό προοίμιο ZC, παρέχοντας μια πρώτη εκτίμηση της θέσης έναρξης των δεδομένων.

Μετά τον αρχικό εντοπισμό του πλαισίου, πραγματοποιείται λεπτομερής χρονική ευθυγράμμιση μέσω του αλγορίθμου Schmidl & Cox, ο οποίος βασίζεται σε ένα περιοδικό προοίμιο τύπου [A A]. Ο αλγόριθμος υπολογίζει τη μετρική:

$$M(n) = \frac{\left| \sum_{k=0}^{K/2-1} r_{n+k} \cdot r_{n+k+K/2}^* \right|^2}{\left(\sum_{k=0}^{K/2-1} |r_{n+k+K/2}|^2 \right)^2 + \epsilon}$$

όπου r_n το λαμβανόμενο σήμα, K ο αριθμός των υποφερόντων και ϵ ένας μικρός αριθμός για αριθμητική σταθερότητα. Το σημείο μέγιστης τιμής αυτής της μετρικής υποδεικνύει την ακριβή θέση έναρξης του OFDM συμβόλου. Για περαιτέρω βελτίωση της ακρίβειας, εκτιμάται το εύρος της κορυφής αυτής της μετρικής μέσω της δεύτερης παραγώγου της, ώστε να περιοριστεί η περιοχή αναζήτησης και να επιτευχθεί μέγιστη δυνατή ακρίβεια στον εντοπισμό της θέσης έναρξης.

Επιπρόσθετα, η ίδια μετρική αξιοποιείται για την εκτίμηση της συχνοτικής μετατόπισης. Η γωνία της μέγιστης τιμής της συσχέτισης παρέχει την εκτίμηση της CFO:

$$\hat{\epsilon} = \frac{1}{\pi} \angle \left(\sum_{k=0}^{N/2-1} r_{n+k} \cdot r_{n+k+N/2}^* \right)$$

όπου η CFO εκφράζεται κανονικοποιημένη ως προς το διάστημα των υποφερόντων. Ακολουθεί η διόρθωση της εκτιμώμενης CFO μέσω της εφαρμογής κατάλληλης περιστροφής φάσης σε ολόκληρο το λαμβανόμενο σήμα:

Μετά την εφαρμογή των παραπάνω βημάτων, το σήμα είναι χρονικά και συχνοτικά ευθυγραμμισμένο,

$$r'_n = r_n \cdot e^{-j2\pi\hat{\epsilon}n/N}$$

επιτρέποντας τη σωστή εφαρμογή του μετασχηματισμού Fourier και την εκτίμηση του καναλιού με υψηλή ακρίβεια.

Η συγκεκριμένη προσέγγιση συγχρονισμού που εφαρμόστηκε είναι ιδιαίτερα ανθεκτική και αποτελεσματική, καθώς αξιοποιεί τα πλεονεκτήματα δύο διαφορετικών μεθόδων, Zadoff-Chu και Schmidl & Cox, επιτυγχάνοντας ακριβή και αξιόπιστη ευθυγράμμιση υπό ρεαλιστικές συνθήκες διάδοσης και θορύβου. Αυτή η διαδικασία αποτελεί τη βάση για τις επόμενες ενότητες του κεφαλαίου, οι οποίες περιλαμβάνουν την εκτίμηση καναλιού, την ισοστάθμιση και τελικά την αποκωδικοποίηση των δεδομένων.

3.5 Εκτίμηση Καναλιού και Maximal Ratio Combining (MRC)

Μετά την ολοκλήρωση της διαδικασίας συγχρονισμού, ακολουθεί η εκτίμηση του καναλιού, η οποία αποτελεί αναγκαίο στάδιο προκειμένου να αντιστραφούν οι παραμορφώσεις που προκαλεί το κανάλι διάδοσης στα σύμβολα του OFDM. Στην παρούσα εργασία, η εκτίμηση του καναλιού γίνεται με τη μέθοδο των Ελαχίστων Τετραγώνων, η οποία αξιοποιεί τα πιλοτικά σύμβολα που βρίσκονται σε προκαθορισμένες θέσεις των υποφερόντων του OFDM συμβόλου.

Η μέθοδος LS υλοποιείται υπολογίζοντας τη σχέση μεταξύ των γνωστών πιλοτικών συμβόλων που έχουν μεταδοθεί και των λαμβανόμενων αντίστοιχων υποφερόντων. Συγκεκριμένα, για κάθε πιλοτικό υποφέρον k :

$$\hat{H}[k] = \frac{Y[k]}{P[k]}$$

όπου $Y[k]$ είναι το λαμβανόμενο πιλοτικό σύμβολο, και $P[k]$ το γνωστό μεταδιδόμενο πιλοτικό σύμβολο. Οι εκτιμήσεις αυτές είναι διαθέσιμες μόνο στα πιλοτικά υποφέροντα, συνεπώς για την εκτίμηση του καναλιού σε όλα τα υποφέροντα εφαρμόζεται παρεμβολή. Στη συγκεκριμένη υλοποίηση, πραγματοποιείται ξεχωριστή παρεμβολή του πραγματικού και του φανταστικού μέρους της εκτίμησης LS, χρησιμοποιώντας τετραγωνική παρεμβολή, γεγονός που οδηγεί σε βελτιωμένη ακρίβεια της εκτίμησης του καναλιού σε όλο το φάσμα (σε σχέση πχ με την γραμμική παρεμβολή).

Επειδή το σύστημα είναι SIMO με δύο κεραίες λήψης, η παραπάνω διαδικασία εκτίμησης καναλιού πραγματοποιείται ξεχωριστά για κάθε μία από τις δύο διαδρομές. Ως αποτέλεσμα, προκύπτουν δύο ανεξάρτητες εκτιμήσεις καναλιού ($\hat{H}_1[k]$ και $\hat{H}_2[k]$).

Στη συνέχεια, προκειμένου να αξιοποιηθεί πλήρως η χωρική διαφοροποίηση που προσφέρει η χρήση δύο κεραιών λήψης, εφαρμόζεται η μέθοδος συνδυασμού MRC. Η τεχνική αυτή είναι ιδιαίτερα αποδοτική, καθώς συνδυάζει τα δύο λαμβανόμενα σήματα με τέτοιο τρόπο ώστε να μεγιστοποιηθεί η τελική αναλογία σήματος προς θόρυβο. Πιο συγκεκριμένα, για κάθε υποφέρον k , η ισοστάθμιση MRC εφαρμόζεται ως εξής:

$$\hat{X}[k] = \frac{\hat{H}_1^*[k] \cdot Y_1[k] + \hat{H}_2^*[k] \cdot Y_2[k]}{|\hat{H}_1[k]|^2 + |\hat{H}_2[k]|^2 + \epsilon}$$

όπου $Y_1[k]$ και $Y_2[k]$ είναι τα λαμβανόμενα σήματα στις δύο κεραίες, $\hat{H}_1[k]$ και $\hat{H}_2[k]$ οι εκτιμήσεις καναλιού στις αντίστοιχες κεραίες, και ϵ ένας μικρός αριθμός που προστίθεται για αριθμητική σταθερότητα και εκφράζει την ύπαρξη του θορύβου.

Η εφαρμογή του MRC έχει ως αποτέλεσμα την ευθυγράμμιση των σημάτων ως προς τη φάση, ενώ ταυτόχρονα δίνει μεγαλύτερη βαρύτητα στις διαδρομές με υψηλότερη αναλογία σήματος προς θόρυβο. Το αποτέλεσμα είναι μια σημαντική βελτίωση της ποιότητας του λαμβανόμενου σήματος, η οποία αντικατοπτρίζεται στη συνέχεια κατά την αποδιαμόρφωση και την αποκωδικοποίηση των δεδομένων.

Η συνδυασμένη χρήση της εκτίμησης LS και της ισοστάθμισης MRC διασφαλίζει την αποτελεσματική αντιμετώπιση των παραμορφώσεων που εισάγει το SIMO κανάλι, επιτρέποντας στο σύστημα να αξιοποιεί πλήρως τα οφέλη της χωρικής ποικιλομορφίας για να επιτύχει βελτιωμένη απόδοση.

3.6 Αποδιαμόρφωση και Αποκωδικοποίηση

Μετά την επιτυχή εφαρμογή της διαδικασίας ισοστάθμισης με τη χρήση Maximum Ratio Combining, το σύστημα προχωρά στο επόμενο κρίσιμο στάδιο: την αποδιαμόρφωση και την αποκωδικοποίηση του ισοσταθμισμένου σήματος, με απώτερο στόχο την ανάκτηση των αρχικών δεδομένων που μεταδόθηκαν.

Στην παρούσα υλοποίηση σε Python, η αποδιαμόρφωση των συμβόλων 16-QAM γίνεται με τη μέθοδο hard-decision. Αυτό σημαίνει ότι κάθε λαμβανόμενο σύμβολο, έστω y , αφού έχει διορθωθεί ως προς τις παραμορφώσεις του καναλιού μέσω της ισοστάθμισης, αντιστοιχίζεται στο πλησιέστερο σημείο του αστερισμού 16-QAM, έστω s_k . Ο αστερισμός 16-QAM αποτελείται από 16 διακριτά σημεία, $S = \{s_0, s_1, \dots, s_{15}\}$, όπου κάθε s_k αντιστοιχεί σε μια μοναδική ακολουθία 4 bits. Η hard-decision επιλογή του s_k γίνεται με βάση την ευκλείδεια απόσταση μεταξύ του λαμβανόμενου συμβόλου y και όλων των πιθανών σημείων s_k του αστερισμού:

$$\hat{s} = \arg \min_{s_k \in S} \|y - s_k\|^2$$

Όπου $\hat{s}$ είναι το επιλεγμένο σημείο του αστερισμού που είναι το εγγύτερο στο λαμβανόμενο σύμβολο y . Τα 4 bits που αντιστοιχούν σε αυτό το $\hat{s}$ είναι αυτά που λαμβάνονται ως έξοδος της αποδιαμόρφωσης. Είναι σημαντικό να τονιστεί ότι η μέθοδος hard-decision είναι απλή στην υλοποίηση και έχει χαμηλό υπολογιστικό κόστος, όμως δεν εκμεταλλεύεται την "αβεβαιότητα" ή "εμπιστοσύνη" του λαμβανόμενου σήματος. Δηλαδή, δεν λαμβάνει υπόψη πόσο κοντά ή μακριά βρίσκεται ένα σύμβολο από το ιδανικό σημείο του αστερισμού, αλλά μόνο ποιο είναι το πιο κοντινό.

Τα bits που προκύπτουν από αυτή τη hard-decision αποδιαμόρφωση τροφοδοτούνται σε έναν Viterbi αποκωδικοποιητή. Ο Viterbi αποκωδικοποιητής, σε αυτή την φάση της υλοποίησης, είναι διαμορφωμένος για αποκωδικοποίηση hard-decision. Ο ρόλος του είναι να διορθώσει πιθανά σφάλματα που εισήχθησαν κατά τη μετάδοση, εκμεταλλευόμενος την πλεονάζουσα πληροφορία που προστέθηκε από τον κωδικοποιητή στο στάδιο της εκπομπής, μέσω ενός κώδικα διόρθωσης σφαλμάτων. Ο αλγόριθμος Viterbi διασχίζει ένα "πλέγμα" που αναπαριστά τις πιθανές καταστάσεις του κωδικοποιητή και τις μεταβάσεις μεταξύ τους. Για κάθε λαμβανόμενο hard-decision bit, ο αλγόριθμος υπολογίζει το "μήκος μονοπατιού", το οποίο στην περίπτωση της hard-decision αντιστοιχεί στον αριθμό των διαφορών (ή σφαλμάτων) μεταξύ του λαμβανόμενου bit και του αναμενόμενου bit για κάθε πιθανό μονοπάτι στο πλέγμα.

Ο Viterbi επιλέγει το μονοπάτι με το ελάχιστο συνολικό μήκος μονοπατιού, το οποίο θεωρείται η βέλτιστη εκτίμηση της αρχικής κωδικοποιημένης ακολουθίας. Η βασική αρχή λειτουργίας του μπορεί να συνοψιστεί στην εύρεση του μονοπατιού με την ελάχιστη Απόσταση Hamming από την λαμβανόμενη ακολουθία. Αν υποθέσουμε ότι η λαμβανόμενη ακολουθία hard-decision bits είναι $R=(r_1, r_2, \dots, r_N)$ και μια πιθανή κωδικοποιημένη ακολουθία είναι $C=(c_1, c_2, \dots, c_N)$, τότε η Απόσταση Hamming υπολογίζεται ως:

$$d_H(R, C) = \sum_{i=1}^N \mathbb{I}(r_i \neq c_i)$$

Όπου $\mathbb{I}(\cdot)$ είναι η συνάρτηση δείκτη που λαμβάνει την τιμή 1 αν το όρισμά της είναι αληθές, και 0 διαφορετικά. Ο Viterbi αποκωδικοποιητής βρίσκει την ακολουθία C^* που ελαχιστοποιεί αυτή την απόσταση.

Τέλος, τα δεδομένα που αποκωδικοποιούνται από τον Viterbi αποκωδικοποιητή συγκρίνονται με τα αρχικά πληροφοριακά bits που μεταδόθηκαν. Αυτή η σύγκριση είναι απαραίτητη για τον υπολογισμό κρίσιμων μετρικών απόδοσης του συστήματος, επιτρέποντας μια ποσοτική αξιολόγηση της αποτελεσματικότητας ολόκληρης της αλυσίδας μετάδοσης και επεξεργασίας δεδομένων στην υλοποίηση του συστήματος OFDM σε Python.

3.7 Υπολογισμός Μετρικών και Ανάλυση Απόδοσης

Μετά την ολοκλήρωση της διαδικασίας αποδιαμόρφωσης και αποκωδικοποίησης, η απόδοση του συστήματος αξιολογείται με βάση τυποποιημένες μετρικές φυσικού επιπέδου, οι οποίες αντικατοπτρίζουν την αξιοπιστία, την ποιότητα και την ευρωστία του πομποδέκτη κάτω από ποικίλες συνθήκες καναλιού και θορύβου.

Η βασική μετρική που χρησιμοποιείται είναι το Bit Error Rate (BER), το οποίο ορίζεται ως:

$$\text{BER} = \frac{N_{\text{errors}}}{N_{\text{total_bits}}}$$

όπου N_{errors} είναι ο αριθμός των εσφαλμένων bits στην έξοδο του αποκωδικοποιητή και $N_{\text{total_bits}}$ ο συνολικός αριθμός των bits που μεταδόθηκαν.

Παράλληλα, μετράται η Error Vector Magnitude (EVM), η οποία αποτυπώνει τη μέση απόσταση (σε κανονικοποιημένες μονάδες) μεταξύ των λαμβανόμενων και των ιδανικών συμβόλων στον αστερισμό:

$$\text{EVM}_{\text{rms}} = \sqrt{\frac{1}{N} \sum_{k=1}^N \frac{|S_k - \hat{S}_k|^2}{|S_k|^2}}$$

όπου S_k είναι το ιδανικό σύμβολο και $\hat{S}_k$ το λαμβανόμενο ισοσταθμισμένο σύμβολο στο υποφέρον k . Η EVM παρέχει ενδείξεις για την ποιότητα της διαμόρφωσης και την ακρίβεια της ισοστάθμισης, ακόμη και όταν δεν παρατηρούνται bit errors λόγω FEC. Σαν μέτρο είναι ανάλογο/αντίστοιχο του Σηματοθορυβικού Λόγου (SNR).

Επιπλέον, αξιολογείται η κατανομή των LLR τιμών στην είσοδο του αποκωδικοποιητή για να αποτιμηθεί η ποιότητα των εκτιμήσεων αποδιαμόρφωσης. Τα LLRs με υψηλή απόλυτη τιμή δείχνουν ισχυρή βεβαιότητα, ενώ η συχνή εμφάνιση LLRs κοντά στο μηδέν υποδηλώνει αβεβαιότητα, κάτι που σχετίζεται άμεσα με το SNR και την ακρίβεια εκτίμησης του καναλιού.

Πραγματοποιείται επίσης σύγκριση της πραγματικής απόκρισης καναλιού (όπως εφαρμόστηκε στο μοντέλο διάδοσης) με την εκτίμηση LS, μέσω υπολογισμού του σφάλματος εκτίμησης:

$$\epsilon_H[k] = |H[k] - \hat{H}[k]|^2$$

το οποίο μετριέται κατά μέσο όρο ανά OFDM σύμβολο ή συνολικά.

Τέλος, καταγράφονται τα εξής:

4. BER με και χωρίς χρήση MRC για ανάλυση της συνεισφοράς του SIMO συνδυασμού.
5. BER σε soft-decision και hard-decision Viterbi για σύγκριση των αποκωδικοποιητών.
6. Ανάλυση επιδόσεων σε διαφορετικά κανάλια και επίπεδα Rician K-factor.

Η ανάλυση αυτών των μετρικών επιτρέπει την πλήρη αξιολόγηση της αλυσίδας του πομποδέκτη και παρέχει ισχυρή βάση για σύγκριση με υλοποιήσεις σε υλικό (π.χ. RFSoc) στο επόμενο κεφάλαιο.

4

Υλοποίηση του Συστήματος σε

Πλατφόρμα Xilinx RFSoc 4x2

Η μετάβαση από το μοντέλο προσομοίωσης σε πραγματικό υλικό αποτελεί κρίσιμο στάδιο για την επιβεβαίωση της λειτουργικότητας και της απόδοσης του συστήματος OFDM σε πραγματικές συνθήκες. Το παρόν κεφάλαιο περιγράφει την υλοποίηση του πομποδέκτη στην πλατφόρμα Xilinx Zynq UltraScale+ RFSoc 4x2, με χρήση του RFSoc-MTS overlay για απευθείας διασύνδεση με τους μετατροπείς DAC/ADC και για ροή δεδομένων μέσω DMA.

Η προσέγγιση που ακολουθείται βασίζεται σε software-defined ροή με έλεγχο μέσω Python, αξιοποιώντας τις δυνατότητες της προγραμματιζόμενης λογικής και των ενσωματωμένων μετατροπέων, χωρίς να απαιτείται υλοποίηση σε HDL. Το hardware λειτουργεί ως διεπαφή πραγματικής μετάδοσης/λήψης για το σήμα που παράγεται και αναλύεται από το μοντέλο Python.

4.1 Βελτιώσεις και Αναβαθμίσεις στην Υλοποίηση του Συστήματος

Η υλοποίηση του συστήματος OFDM στην πλατφόρμα Xilinx RFSoc 4x2 συνοδεύεται από σημαντικές τεχνικές και λειτουργικές αναβαθμίσεις σε σχέση με την αρχική προσομοίωση σε Python (όπως περιγράφηκε στο Κεφάλαιο 3). Οι δύο κύριες βελτιώσεις που εισάγονται είναι η μετάβαση από soft-decision decoding με λόγο κωδικοποίησης 1/2 σε soft-decision decoding με λόγο κωδικοποίησης 1/3 και η επέκταση από σύστημα μονής εισόδου–μονής εξόδου (SISO) σε σύστημα πολλαπλών εισόδων–πολλαπλών εξόδων (MIMO) με δύο δέκτες.

4.1.1 Soft-Decision Αποκωδικοποίηση με Αναβαθμισμένο Συνελκτικό Κώδικα

Στην τελική έκδοση της προσομοίωσης σε Python, υλοποιήθηκε soft-decision Viterbi αποκωδικοποίηση χρησιμοποιώντας έναν συνελκτικό κώδικα με λόγο κωδικοποίησης (code rate) 1/2. Αυτός ο κώδικας βασιζόταν σε δύο συνελκτικούς πολυωνυμικούς γεννήτορες '1111001' και '1011011', οι οποίοι καθορίζουν τον τρόπο με τον οποίο κάθε bit πληροφορίας μετατρέπεται σε δύο κωδικοποιημένα bits εξόδου. Η αποκωδικοποίηση πραγματοποιούνταν ήδη με χρήση Log-Likelihood Ratios (LLRs), τα οποία υπολογίζονταν μέσω ενός soft demapper για 16-QAM διαμόρφωση. Αυτή η προσέγγιση παρείχε για κάθε bit έναν αριθμητικό βαθμό εμπιστοσύνης ως προς την πιθανότητα να είναι 0 ή 1, αξιοποιώντας την "μαλακή/ήπια" πληροφορία από τον δέκτη.

Στην υλοποίηση επάνω στην πλατφόρμα RFSoc, πραγματοποιήθηκε μια σημαντική αναβάθμιση της υλοποίησης Forward Error Correction (FEC). Αντί να γίνει μετάβαση από hard-decision σε soft-decision (καθώς το soft-decision ήταν ήδη σε χρήση), η βελτίωση αφορά τη μετάβαση σε έναν συνελκτικό

κώδικα με λόγο 1/3. Αυτή η αλλαγή επιτεύχθηκε μέσω της αξιοποίησης τριών γεννητόρων πολωνύμων '1111001', '1011011', '1101101'.

Η αλλαγή από λόγο 1/2 σε λόγο 1/3 σημαίνει ότι για κάθε ένα bit πληροφορίας που εισέρχεται στον κωδικοποιητή, παράγονται πλέον τρία κωδικοποιημένα bits εξόδου. Αυτή η προσθήκη ενός επιπλέον πλεονάζοντος bit για κάθε bit πληροφορίας αύξησε τον πλεονασμό (redundancy) της διόρθωσης σφαλμάτων. Με περισσότερη πλεονάζουσα πληροφορία, ο αποκωδικοποιητής έχει στη διάθεσή του περισσότερα δεδομένα για να ανακτήσει αξιόπιστα την αρχική πληροφορία, ακόμη και παρουσία σημαντικού θορύβου ή παρεμβολών. Η αποκωδικοποίηση συνεχίζει να βασίζεται σε soft-decision Viterbi, διατηρώντας τη χρήση των LLRs ως είσοδο, αλλά πλέον λειτουργεί με έναν πιο ισχυρό και ανθεκτικό κώδικα.

Ο υπολογισμός των LLRs παραμένει θεμελιώδης για την παροχή της απαραίτητης "ήπιας" πληροφορίας στον Viterbi αποκωδικοποιητή. Για κάθε λαμβανόμενο σύμβολο y και για κάθε bit b_i

$$LLR_i = \log \left(\frac{\sum_{s_k \in S_1^i} e^{-\frac{\|y-s_k\|^2}{\sigma^2}}}{\sum_{s_k \in S_0^i} e^{-\frac{\|y-s_k\|^2}{\sigma^2}}} \right)$$

Όπου:

- S_{1i} είναι το σύνολο των σημείων του αστερισμού 16-QAM στα οποία το i -οστό bit έχει τιμή 1.
- S_{0i} είναι το σύνολο των σημείων του αστερισμού 16-QAM στα οποία το i -οστό bit έχει τιμή 0.
- $\|y-s_k\|^2$ είναι η τετραγωνική ευκλείδεια απόσταση μεταξύ του λαμβανόμενου συμβόλου y και του ιδανικού σημείου αστερισμού s_k .
- σ^2 είναι η εκτιμώμενη διασπορά του θορύβου, η οποία προέρχεται από την ανάλυση των πιλοτικών υποφερόντων κατά τη διαδικασία της καναλικής εκτίμησης. Αυτή η δυναμική εκτίμηση της σ^2 επιτρέπει στον soft demapper να προσαρμόζεται στις πραγματικές συνθήκες θορύβου του καναλιού.

Στη συνέχεια, οι LLRs διοχετεύονται στον soft-decision Viterbi αποκωδικοποιητή. Σε αντίθεση με έναν hard-decision Viterbi που θα χρησιμοποιούσε την Απόσταση Hamming, ο soft-decision Viterbi υπολογίζει το "path metric" (το κόστος κάθε πιθανής διαδρομής στο trellis) με βάση την αθροιστική συνεισφορά των LLRs σε κάθε διαδρομή. Η χρήση του κώδικα με λόγο 1/3 σημαίνει ότι ο αποκωδικοποιητής λειτουργεί πλέον σε συνθήκες υψηλότερης πλεοναστικής πληροφορίας, επιτρέποντας την ανάκτηση δεδομένων ακόμη και όταν οι μεμονωμένες LLRs είναι αβέβαιες.

Η μετάβαση από rate-1/2 σε rate-1/3 σε συνδυασμό με τη διατήρηση και βελτιστοποίηση του soft-decision decoding οδήγησε σε αισθητή βελτίωση της απόδοσης στο τελικό Bit Error Rate (BER), ειδικά σε χαμηλά SNR, όπως αποτυπώνεται και στα πειραματικά αποτελέσματα του Κεφαλαίου 5. Ταυτόχρονα, αυτή η αναβάθμιση της FEC υποδομής επέτυχε συμβατότητα με πιο εξελιγμένες τεχνικές καναλικής εκτίμησης και MIMO λήψης, καθώς η αυξημένη πλεονάζουσα πληροφορία ενισχύει τη ρομπουστότητα του αποκωδικοποιητή απέναντι σε συνθήκες fading, πολυδιαδρομής (multipath) και υπολειμματικών συγχρονιστικών σφαλμάτων.

4.1.2 Επέκταση σε MIMO

Μία ακόμη κρίσιμη διαφοροποίηση της υλοποίησης στο RFSoc είναι η επέκταση σε MIMO (Multiple-Input Multiple-Output) διαμόρφωση λήψης. Το σύστημα πλέον κάνει χρήση δύο καναλιών εισόδου από το ADC Tile 224 (κανάλι 0) και το ADC Tile 226 (κανάλι 0 ή 1), με αποτέλεσμα η λήψη να γίνεται μέσω δύο ανεξάρτητων δεκτών ($N_r=2$). Κάθε λήπτης λαμβάνει το ίδιο OFDM σήμα μέσω διαφορετικού καναλιού, επιτρέποντας στο σύστημα να εκμεταλλευτεί την διαφορισιμότητα και μεταβλητότητα του καναλιού. Αυτή η διαφορική λήψη είναι ζωτικής σημασίας για την αντιμετώπιση του fading και των παρεμβολών.

Για τον βέλτιστο συνδυασμό των δύο σημάτων που λαμβάνονται, εφαρμόζεται η τεχνική Maximum Ratio Combining κατά το στάδιο της ισοστάθμισης. Το MRC είναι μια βέλτιστη τεχνική συνδυασμού σήματος που μεγιστοποιεί τον λόγο σήματος προς θόρυβο στην έξοδο του συνδυαστή. Συγκεκριμένα, η τελική εκτίμηση των συμβόλων $\hat{x}[k]$ μετά την ισοστάθμιση, για κάθε υποφέρουσα συχνότητα k , δίνεται από:

$$\hat{x}[k] = \frac{\sum_{r=1}^{N_r} Y_r[k] \cdot H_r^*[k]}{\sum_{r=1}^{N_r} |H_r[k]|^2}$$

Όπου:

- $Y_r[k]$ είναι το λαμβανόμενο OFDM σύμβολο στην υποφέρουσα συχνότητα k από τον r -οστό δέκτη.
- $H_r[k]$ είναι η εκτίμηση του καναλιού για την υποφέρουσα συχνότητα k στον r -οστό δέκτη.
- $H_r^*[k]$ είναι ο συζυγής μιγαδικός του $H_r[k]$.
- N_r είναι ο αριθμός των δεκτών ($N_r=2$ στην περίπτωση αυτή).

Η τεχνική αυτή ουσιαστικά σταθμίζει το σήμα από κάθε δέκτη με το κέρδος του καναλιού (ή την ισχύ του σήματος) σε αυτό το κανάλι, πριν τα συνδυάσει. Μεγαλύτερη βαρύτητα δίνεται στα κανάλια που έχουν λιγότερο θόρυβο και ισχυρότερο σήμα, προσφέροντας έτσι σημαντική βελτίωση στην ανθεκτικότητα σε παρεμβολές και φαινόμενα fading.

4.2 Δημιουργία και Μετάδοση του OFDM Σήματος στο RFSoc

Στην υλοποίηση του συστήματος OFDM στην πλατφόρμα Xilinx RFSoc 4x2, η διαδικασία δημιουργίας και μετάδοσης του σήματος αποκτά μια νέα διάσταση σε σχέση με την καθαρά προσομοίωση. Το διαμορφωμένο baseband σήμα παράγεται αρχικά σε περιβάλλον Python και στη συνέχεια αποστέλλεται προς εκπομπή μέσω των Digital-to-Analog Converters του RFSoc. Αυτή η διαδικασία αποτελεί ένα κρίσιμο στάδιο του συστήματος, καθώς γεφυρώνει το χάσμα μεταξύ του ψηφιακού πομπού (που υλοποιείται σε Python) και της αναλογικής διεπαφής του υλικού (DACs/ADCs), μετατρέποντας τις ψηφιακές αναπαραστάσεις σε πραγματικά ηλεκτρομαγνητικά κύματα.

Η κυματομορφή που πρόκειται να μεταδοθεί δεν είναι ένα απλό σύνολο δεδομένων, αλλά μια σύνθετη δομή που ενσωματώνει όλα τα απαραίτητα στοιχεία ενός τυπικού OFDM σήματος για την ορθή λειτουργία και αποκατάσταση των δεδομένων στον δέκτη. Αυτά τα στοιχεία περιλαμβάνουν:

- Ακολουθίες Συγχρονισμού: Όπως οι ακολουθίες Zadoff-Chu και τα preambles με δομή σύμφωνα με τον αλγόριθμο Schmidl & Cox, οι οποίες είναι σχεδιασμένες να επιτρέπουν στον δέκτη τον ακριβή συγχρονισμό στο πεδίο του χρόνου και της συχνότητας. Αυτές οι ακολουθίες προστίθενται στην αρχή ή ενδιάμεσα στα πακέτα δεδομένων.
- Διαμορφωμένα QAM Σύμβολα: Τα ίδια τα δεδομένα, τα οποία έχουν υποστεί διαμόρφωση (π.χ., 16-QAM) και έχουν κατανεμηθεί στα επιμέρους υποφέροντα.
- Κυκλικά Πρόθεματα: Ένα αντίγραφο ενός μέρους από το τέλος κάθε OFDM συμβόλου προστίθεται στην αρχή του, για την αντιμετώπιση της διασποράς λόγω πολλαπλών διαδρομών και την αποφυγή διασυμβολικής παρεμβολής.
- Πιλοτικά Υποφέροντα: Συγκεκριμένα υποφέροντα στα οποία μεταδίδονται γνωστά σύμβολα, επιτρέποντας στον δέκτη να εκτιμήσει τις ιδιότητες του καναλιού.

Ένα ακόμα κρίσιμο στοιχείο είναι το padding, το οποίο χρησιμοποιείται για να εξασφαλίσει ότι το συνολικό μήκος της κυματομορφής ταιριάζει ακριβώς με το μέγεθος του buffer του DAC στο RFSoc. Αυτό είναι απαραίτητο για την αποτελεσματική χρήση του DMA. Η διαδικασία του padding δεν προσθέτει απλώς μηδενικά δείγματα, αλλά δημιουργεί αντίγραφα του διαμορφωμένου OFDM πακέτου (payload) έως ότου γεμίσει πλήρως ο DAC buffer. Αυτό υλοποιείται ως εξής:

```
# === Fill DAC buffer with repeated payload copies ===

dac_len = ol.dac_player.shape[0] # Λαμβάνουμε το μέγεθος του DAC buffer

IQ_len = len(IQ_interleaved) # Λαμβάνουμε το μήκος του διαμορφωμένου OFDM πακέτου (payload)

num_repeats = int(np.ceil(dac_len / IQ_len)) # Υπολογίζουμε πόσες φορές πρέπει να επαναληφθεί το πακέτο για να γεμίσει το buffer

repeated_IQ = np.tile(IQ_interleaved, num_repeats) # Δημιουργούμε πολλαπλά αντίγραφα του πακέτου

# Truncate to exactly match the DAC buffer size

full_IQ = repeated_IQ[:dac_len] # Κόβουμε την ακολουθία στο ακριβές μέγεθος του DAC buffer
```

Σε αυτό το πλαίσιο, το `IQ_interleaved` αναφέρεται στο αρχικό διαμορφωμένο OFDM πακέτο (συμπεριλαμβανομένων συγχρονισμού, QAM συμβόλων, CP, πιλοτικών, κ.λπ.) που δημιουργήθηκε σε Python. Η συνάρτηση `np.tile()` επαναλαμβάνει αυτό το πακέτο όσες φορές απαιτείται, διασφαλίζοντας ότι το buffer γεμίζει αποτελεσματικά. Τέλος, η ακολουθία περικόπτεται ώστε να ταιριάζει ακριβώς με το μέγεθος του DAC buffer, δημιουργώντας έτσι το `full_IQ` σήμα προς μετάδοση.

Η ακολουθία αυτή παράγεται ως ένας μονοδιάστατος πίνακας μιγαδικών αριθμών. Κάθε δείγμα σε αυτόν τον πίνακα είναι σε μορφή `np.complex64`, όπου το πραγματικό μέρος αντιπροσωπεύει το κανάλι I (in-phase) και το φανταστικό μέρος το κανάλι Q (quadrature). Αυτή η μορφή είναι απαραίτητη για την αναπαράσταση του baseband σήματος, το οποίο στη συνέχεια μετατρέπεται σε αναλογικό και υφίσταται `upconversion` στην επιθυμητή φέρουσα συχνότητα από το υλικό του RFSoc.

Η αποστολή του σήματος στους DACs του RFSoc πραγματοποιείται μέσω του DMA, γράφοντας απευθείας στον buffer του driver. Η εντολή:

```
ol.dac_player[:] = full_IQ # Μεταφορά κυματομορφής στον DAC player
```

αντιστοιχεί σε μια πλήρη DMA μεταφορά ολόκληρου του διανύσματος `full_IQ` προς τον DAC buffer. Μόλις τα δεδομένα βρίσκονται στον buffer, η κυματομορφή μετατρέπεται αδιάκοπα σε αναλογικό σήμα και εκπέμπεται από την έξοδο του RFSoc. Η μικρή χρονική καθυστέρηση που ακολουθεί:

```
time.sleep(0.01) # Χρονική καθυστέρηση σταθεροποίησης
```

δίνει το απαραίτητο χρονικό περιθώριο για τη σταθεροποίηση του υλικού των DACs πριν την έναρξη της λήψης. Αμέσως μετά, ενεργοποιείται η σύγχρονη διαδικασία λήψης μέσω των ADCs:

```
ol.trigger_capture() # Ενεργοποίηση λήψης από ADC
```

Η χρήση της εντολής `trigger_capture()` είναι ζωτικής σημασίας για τη διασφάλιση χρονολογικής ευθυγράμμισης μεταξύ εκπομπής και δειγματοληψίας. Αυτό επιτρέπει την ακριβή λήψη του μεταδιδόμενου πακέτου και είναι θεμελιώδες για την ορθή λειτουργία του συστήματος, ειδικά όσον αφορά τον συγχρονισμό και την εκτίμηση καναλιού στον δέκτη.

Το πλήρες σύστημα αποστολής είναι σχεδιασμένο για να λειτουργεί με τη μέγιστη δυνατή ακρίβεια και συγχρονισμό, ώστε να διατηρείται η χρονική και συχνотική συνοχή του OFDM σήματος. Αυτή η διαδικασία είναι πλήρως παραμετρική, επιτρέποντας την εύκολη τροποποίηση του σήματος που μεταδίδεται (π.χ., αλλαγή διαμόρφωσης, μήκους CP, ή ακολουθιών συγχρονισμού) για σκοπούς αξιολόγησης διαφορετικών διαμορφώσεων, σχημάτων κωδικοποίησης και τεχνικών συγχρονισμού. Η ευελιξία αυτή καθιστά την πλατφόρμα RFSoc ένα ιδανικό εργαλείο για πειραματική έρευνα και ανάπτυξη σε ασύρματα συστήματα επικοινωνίας.

Στην επόμενη υποενότητα, περιγράφεται λεπτομερώς η διαδικασία λήψης του σήματος μέσω των ADCs, η ανασύνθεση της συμπλεγμένης κυματομορφής, καθώς και η προετοιμασία του σήματος για περαιτέρω επεξεργασία στον δέκτη.

4.3 Λήψη και Ανακατασκευή του Ληφθέντος Σήματος

Αφού το OFDM σήμα δημιουργηθεί και μεταδοθεί μέσω των DACs του RFSoc, η διαδικασία συνεχίζεται με τη λήψη του. Στην παρούσα υλοποίηση, για τη διασφάλιση ενός ελεγχόμενου περιβάλλοντος δοκιμών, η μετάδοση γίνεται μέσω απευθείας καλωδιακής σύνδεσης, χωρίς τη χρήση κεραιών. Αυτό εξαλείφει τις επιδράσεις του ασύρματου καναλιού και επιτρέπει την ακριβή αξιολόγηση της απόδοσης του συστήματος. Η λήψη του σήματος πραγματοποιείται από τους ενσωματωμένους ADCs της πλατφόρμας RFSoc 4x2 και καταγράφεται με χρήση του εσωτερικού μηχανισμού DMA.

4.3.1 Καταγραφή Δειγμάτων

Η διαδικασία λήψης συγχρονίζεται άμεσα με την εκπομπή μέσω της εντολής `ol.trigger_capture()`. Αυτή η εντολή ενεργοποιεί ταυτόχρονα τη δειγματοληψία στους επιλεγμένους ADCs, διασφαλίζοντας την χρονική ευθυγράμμιση μεταξύ εκπομπής και λήψης, ένα κρίσιμο στοιχείο για τη συνέπεια και ακρίβεια σε μία MIMO διαμόρφωση. Τα δείγματα καταγράφονται σε μία προδιαμορφωμένη δομή δεδομένων, τη μεταβλητή `AlignedCaptureSamples`, η οποία είναι ένας πίνακας 3 διαύλων με τύπο `np.int16`, έτοιμος να αποθηκεύσει δεδομένα από ένα έως και τρεις δέκτες (αν και στην παρούσα υλοποίηση χρησιμοποιούνται δύο). Η μεταφορά των ψηφιοποιημένων δειγμάτων από το υλικό του RFSoc στον επεξεργαστή γίνεται αποτελεσματικά μέσω της εντολής:

```
ol.internal_capture(AlignedCaptureSamples)
```


Το κύριο κανάλι λήψης χρησιμοποιεί το ADC Tile 224 Channel 0 (γνωστό και ως ADC D). Λόγω ενός χαρακτηριστικού της εσωτερικής σχεδίασης του RFSoc 4x2, παρατηρείται αντιστροφή πολικότητας στο συγκεκριμένο κανάλι. Αυτό το φαινόμενο διορθώνεται λογισμικά, με αλλαγή προσήμου στα καταγεγραμμένα δείγματα του συγκεκριμένου διαύλου:

```
AlignedCaptureSamples[0][:] = -AlignedCaptureSamples[0][:]
```

Το δεύτερο κανάλι λήψης, το οποίο είναι απαραίτητο για την υλοποίηση MIMO, επιλέγεται δυναμικά μεταξύ του ADC B (Tile 226 Channel 0) ή του ADC C (Tile 226 Channel 1), ανάλογα με τη ρύθμιση του παραμέτρου ADC_in_use στο περιβάλλον Python. Τα κανάλια αυτά δεν εμφανίζουν αντίστοιχα φαινόμενα αντιστροφής πολικότητας.

4.3.2 Ανακατασκευή του Συμπλεγμένου Baseband Σήματος

Οι ADCs του RFSoc αποθηκεύουν τα I/Q δείγματα σε διαπλεκόμενη μορφή ως ακέραιες τιμές σε έναν ενιαίο πίνακα. Αυτό σημαίνει ότι το πραγματικό (I) και το φανταστικό (Q) μέρος κάθε σύνθετου δείγματος εναλλάσσονται στη μνήμη. Για τη σωστή επεξεργασία του σήματος ως σύνθετους αριθμούς, απαιτείται αποδιαπλοκή (deinterleaving) των δειγμάτων και μετατροπή τους σε μορφή κινητής υποδιαστολής (np.float32). Η ανακατασκευή του συμπλεγμένου σήματος για το κύριο κανάλι ADC D επιτυγχάνεται με διαχωρισμό των καναλιών I και Q ως εξής:

```
rx_complex0 = AlignedCaptureSamples[0][0::2].astype(np.float32) + 1j *  
              AlignedCaptureSamples[0][1::2].astype(np.float32)
```

Σε αυτή την εντολή, το [0::2] επιλέγει όλα τα δείγματα που αντιστοιχούν στο πραγματικό μέρος (I), ενώ το [1::2] επιλέγει όλα τα δείγματα που αντιστοιχούν στο φανταστικό μέρος (Q), ξεκινώντας από τη δεύτερη θέση. Στη συνέχεια, μετατρέπονται σε float32 και συνδυάζονται σε μιγαδικούς αριθμούς.

Για το δεύτερο κανάλι MIMO, εφόσον ο αριθμός των δεκτών είναι ίσος με 2, η ίδια διαδικασία αποδιαπλοκής εφαρμόζεται στο επιλεγμένο κανάλι (ADC_in_use):

```
if Nr == 2:  
    rx_complex1 = AlignedCaptureSamples[ADC_in_use][0::2].astype(np.float32) + 1j *  
                  AlignedCaptureSamples[ADC_in_use][1::2].astype(np.float32)
```

Το τελικό σήμα οργανώνεται σε έναν δισδιάστατο πίνακα rx_signal_ZC, με διαστάσεις [αριθμός_δειγμάτων, αριθμός_δεκτών], όπου κάθε στήλη αντιπροσωπεύει τα μιγαδικά δείγματα από έναν διαφορετικό δέκτη:

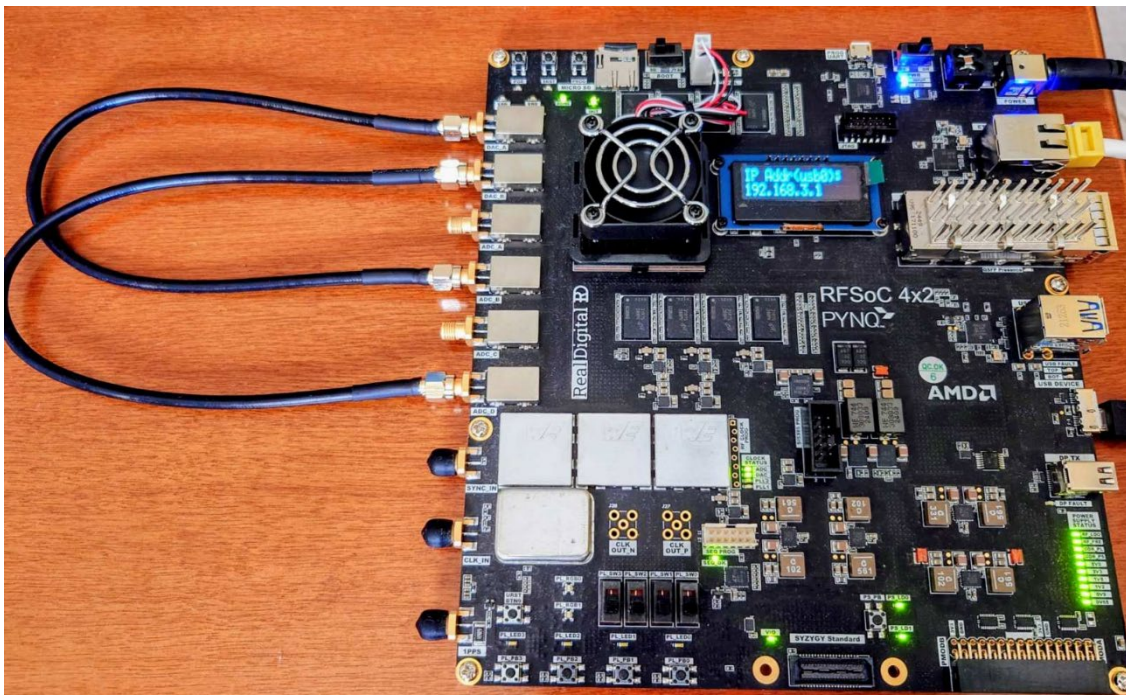
```
rx_signal_ZC = np.zeros((len(rx_complex0), Nr), dtype=complex)  
rx_signal_ZC[:, 0] = rx_complex0  
if Nr == 2:  
    rx_signal_ZC[:, 1] = rx_complex1
```

Αυτό το σήμα, `rx_signal_ZC`, αποτελεί πλέον το σημείο εκκίνησης για τη διαδικασία αποδιαμόρφωσης στον δέκτη. Ιδιαίτερη μέριμνα δίνεται στη χρονική ευθυγράμμιση των καναλιών εισόδου. Καθώς η καταγραφή ενεργοποιείται ταυτόχρονα και για τα δύο ADC tiles μέσω ενός κοινού trigger, εξασφαλίζεται η χρονική συνέπεια μεταξύ των λαμβανόμενων καναλιών – στοιχείο κρίσιμο για την επακόλουθη σωστή λειτουργία της ισοστάθμισης και του συνδυασμού σήματος.

Πριν τη διαδικασία συγχρονισμού και εκτίμησης καναλιού, το λαμβανόμενο σήμα έχει μετατραπεί σε μορφή κατάλληλη για το υπόλοιπο pipeline επεξεργασίας. Συγκεκριμένα, το `rx_signal_ZC` τροφοδοτεί τα επόμενα στάδια του δέκτη, τα οποία περιλαμβάνουν:

- Συγχρονισμό μέσω των ακολουθιών Zadoff-Chu και του αλγορίθμου Schmidl & Cox, για τον ακριβή προσδιορισμό της αρχής του πακέτου και της συχνοτικής απόκλισης.
- Αφαίρεση του κυκλικού προθέματος, για την αποφυγή διασυμβολικής παρεμβολής.
- Μετασχηματισμό FFT για τη μετάβαση του σήματος από το πεδίο του χρόνου στο πεδίο της συχνότητας.
- Εκτίμηση καναλιού και ισοστάθμιση, χρησιμοποιώντας τα πιλοτικά υποφέροντα και την τεχνική MRC.
- Αποδιαμόρφωση και αποκωδικοποίηση, συμπεριλαμβανομένης της soft-decision αποκωδικοποίησης.

Αυτό το στάδιο κλείνει τον πλήρη κύκλο του OFDM συστήματος πάνω στο RFSoc: ψηφιακή δημιουργία (Python) → αναλογική μετάδοση (DAC) → δειγματοληψία (ADC) → ανακατασκευή του ληφθέντος σήματος για επεξεργασία (Python). Η εσωτερική λειτουργία καταγραφής έχει δοκιμαστεί εκτενώς ως προς την αξιοπιστία μεταφοράς των δειγμάτων, αποφεύγοντας φαινόμενα overflow ή απώλειας δειγμάτων. Αυτό διασφαλίζει ότι η λήψη αντικατοπτρίζει με ακρίβεια τη μεταδιδόμενη πληροφορία, θέτοντας γερά θεμέλια για τα επόμενα στάδια συγχρονισμού και αποκωδικοποίησης.



Εικόνα 8: Αρχιτεκτονική σύνδεσης του κυκλώματος DAC–ADC στο RFSoc 4x2. Το σήμα μεταδίδεται από τους DAC μέσω απευθείας καλωδίου και δειγματοληπτείται από δύο ADCs (MIMO).

4.4 Αποδιαμόρφωση και Συγχρονισμός του Ληφθέντος OFDM Σήματος στο RFSoc

Η παρούσα υποενότητα βασίζεται στις αρχές συγχρονισμού και αποδιαμόρφωσης που αναπτύχθηκαν θεωρητικά στα Κεφάλαια 2 και 3, εστιάζοντας πλέον στην πρακτική τους υλοποίηση στο υλικό περιβάλλον του RFSoc. Εφόσον η επικοινωνία DAC–ADC πραγματοποιείται μέσω καλωδίου, χωρίς την επίδραση καναλιού διάδοσης, η πρόκληση μετατοπίζεται στην αντιμετώπιση καθυστερήσεων του συστήματος DMA, αποκλίσεων φάσης από τις πηγές ρολογιού και πιθανών υπολειμματικών θορύβων.

Το σήμα λαμβάνεται από τους ADCs και αποθηκεύεται στη μεταβλητή `AlignedCaptureSamples`, η οποία περιέχει διανύσματα ακέραιων τιμών (`int16`) ανά κανάλι. Από εκεί, γίνεται αποσυσκευασία σε μιγαδική μορφή IQ για κάθε ADC, παράγοντας τους διανυσματικούς πίνακες `rx_complex0`, `rx_complex1` κ.λπ., ανάλογα με τη διαμόρφωση MIMO.

Η συγχρονιστική διαδικασία ξεκινά με τον υπολογισμό του μετρικού Schmidl & Cox σε κάθε θέση παραθύρου μήκους K (μέγεθος FFT). Για κάθε υποπαράθυρο, υπολογίζονται η συσχέτιση μεταξύ των δύο μισών (`correlation_ZC_all`) και η ισχύς της δεύτερης μισής (`power_ZC_all`). Ο τελικός συγχρονιστικός δείκτης `metric_ZC` είναι το πηλίκο των παραπάνω τιμών, και το μέγιστό του παρέχει την αρχική εκτίμηση του σημείου έναρξης του πακέτου (`start_index_ZC`).

Η διόρθωση συχνότητας βασίζεται στη φάση του συσχετισμού στο εν λόγω σημείο. Εφαρμόζεται στο σύνολο των ληφθέντων δειγμάτων με χρήση εκθετικού διορθωτικού παράγοντα, ώστε να εξαλειφθεί η υπολειμματική μετατόπιση φάσης.

Ακολουθεί περαιτέρω βελτίωση χρονικής ευθυγράμμισης μέσω εκτίμησης του εύρους της κορυφής του μετρικού. Ανιχνεύεται το εύρος τιμών που ξεπερνούν το 90% της κορυφής και επιλέγεται το μέσο αυτής της περιοχής ως τελικός συγχρονιστικός δείκτης (`corrected_start_index_ZC`). Η στρατηγική αυτή συνδυάζει ακρίβεια με ανθεκτικότητα σε θόρυβο, αποφεύγοντας `false peaks`.

Το ωφέλιμο OFDM σύμβολο εξάγεται με χρήση του συγχρονισμένου δείκτη. Αφαιρείται το κυκλικό πρόθεμα και εφαρμόζεται FFT ανά κανάλι, ώστε να προκύψει το σύνολο `OFDM_demod_ZC`. Οι διαδικασίες αυτές υλοποιούνται σε όλους τους δέκτες παράλληλα, με κοινό χρονικό δείκτη ευθυγράμμισης. Η επιλογή αυτή απλοποιεί την επεξεργασία, υπό την προϋπόθεση ότι η σχετική καθυστέρηση μεταξύ των ADCs είναι σταθερή.

4.5 Αποδιαμόρφωση και Soft Viterbi Αποκωδικοποίηση

Όπως αναφέρθηκε αναλυτικά στο υποκεφάλαιο 4.1 "Βελτιώσεις και Αναβαθμίσεις στην Υλοποίηση του RFSoc", η μετάβαση από `soft-decision` αποκωδικοποίηση με λόγο κωδικοποίησης 1/2 σε ισχυρότερη `soft-decision` αποκωδικοποίηση με λόγο 1/3 αποτελεί μία από τις πλέον καθοριστικές διαφοροποιήσεις της υλοποίησης στο RFSoc σε σχέση με την αρχική προσομοίωση σε Python.

Παρότι και οι δύο εκδόσεις βασίζονται σε `soft-decision` Viterbi decoding, η αύξηση της πλεοναστικής πληροφορίας μέσω της χρήσης τριών εξόδων ανά bit πληροφορίας επιτρέπει στο σύστημα να διατηρεί και να αξιοποιεί ακόμη περισσότερο την πληροφορία εμπιστοσύνης που περιέχεται στα ληφθέντα σήματα. Το σύστημα πλέον επεξεργάζεται τις πιθανότητες μετάδοσης με μεγαλύτερη ευαισθησία, κάτι που είναι ζωτικής σημασίας για την επίτευξη καλύτερης απόδοσης ιδιαίτερα σε συνθήκες χαμηλού λόγου σήματος προς θόρυβο.

4.5.1 Soft QAM Demapping και Log-Likelihood Ratios (LLRs)

Αφού το σήμα έχει περάσει από τα στάδια ισοστάθμισης και MRC, η διαδικασία αποδιαμόρφωσης εφαρμόζεται πάνω στα ισοσταθμισμένα σύμβολα του πεδίου συχνότητας μέσω ενός soft-decision demapper. Σε αντίθεση με τη hard-decision αποδιαμόρφωση που απλώς αντιστοιχίζει κάθε λαμβανόμενο σύμβολο στο πλησιέστερο σημείο του 16-QAM αστερισμού, η soft-decision προσέγγιση υπολογίζει LLRs για κάθε bit που συνθέτει το σύμβολο.

Ένα LLR αποτελεί μια ποσοτικοποίηση της πιθανότητας ένα συγκεκριμένο bit να είναι 0 ή 1, λαμβάνοντας υπόψη το λαμβανόμενο σύμβολο και το επίπεδο θορύβου. Για κάθε λαμβανόμενο σύμβολο 16-QAM, το οποίο κωδικοποιεί 4 bits, ο demapper παράγει 4 αντίστοιχες LLR τιμές. Ο υπολογισμός των LLRs γίνεται με βάση την απόσταση του λαμβανόμενου συμβόλου από τα ιδανικά σημεία του αστερισμού και τη διακύμανση του θορύβου σ^2 . Η υλοποίηση αυτής της λειτουργίας γίνεται με την κλήση της συνάρτησης:

$$\text{rx_llrs} = \text{qam16_soft_demapper}(\text{QAM_est}, \text{sigma2})$$

Όπου:

- **QAM_est:** Είναι το ισοσταθμισμένο σύμβολο (μικαδικός αριθμός) στο πεδίο συχνότητας.
- **sigma2:** Είναι η εκτιμώμενη διασπορά του θορύβου (σ^2), η οποία αποτελεί κρίσιμο παράμετρο για τον ακριβή υπολογισμό των πιθανοτήτων. Η εκτίμηση της σ^2 γίνεται δυναμικά από τα πιλοτικά υποφέροντα, επιτρέποντας την προσαρμοστικότητα της αποδιαμόρφωσης σε μεταβαλλόμενα επίπεδα θορύβου.

Οι τιμές LLR που επιστρέφονται είναι πραγματικοί αριθμοί και περιέχουν την πληροφορία εμπιστοσύνης για το αν κάθε bit είναι '0' ή '1'. Ένα θετικό LLR υποδηλώνει μεγαλύτερη πιθανότητα το bit να είναι '1', ενώ ένα αρνητικό LLR υποδηλώνει μεγαλύτερη πιθανότητα το bit να είναι '0'. Το μέγεθος (απόλυτη τιμή) του LLR αντικατοπτρίζει το επίπεδο εμπιστοσύνης στην απόφαση: μεγαλύτερες απόλυτες τιμές υποδεικνύουν υψηλότερη βεβαιότητα.

4.5.2 Ενίσχυση Εμπιστοσύνης (Scaling των LLRs)

Για την περαιτέρω ενίσχυση της διαχωρισιμότητας μεταξύ πιθανών τιμών και τη βελτιστοποίηση της απόδοσης του Viterbi αλγορίθμου, ιδίως σε συνθήκες χαμηλού SNR, οι υπολογισμένες τιμές LLRs προσαρμόζονται με έναν εμπειρικό συντελεστή ενίσχυσης:

$$\text{rx_llrs} *= 3.0$$

Αυτή η κλιμάκωση πολλαπλασιάζει όλες τις τιμές LLR με το 3.0. Ουσιαστικά, αυξάνει την απόσταση μεταξύ των LLRs που αντιστοιχούν σε "ισχυρές" και "αδύναμες" αποφάσεις. Αυτό έχει ως αποτέλεσμα να τονίζονται οι πιο αξιόπιστες πληροφορίες, καθιστώντας τις πιο διακριτές για τον Viterbi αποκωδικοποιητή και βελτιώνοντας την ικανότητά του να λαμβάνει σωστές αποφάσεις ακόμη και παρουσία σημαντικού θορύβου.

4.5.3 Soft-Input Viterbi Αποκωδικοποίηση

Η τελική αποκωδικοποίηση πραγματοποιείται μέσω ενός soft-input Viterbi αποκωδικοποιητή, ο οποίος είναι ειδικά σχεδιασμένος να δέχεται τις πυκνές τιμές των LLRs ως είσοδο. Η κλήση της λειτουργίας υλοποιείται ως εξής:

```
decoded_bits = enc.viterbi_decoder(rx_llrs, 'soft')
decoded_bits = decoded_bits[:len(info_bits)]
```

Σε αντίθεση με τον hard-decision Viterbi που βασίζεται στην Απόσταση Hamming, ο soft-input Viterbi εφαρμόζει μετρική συσσώρευση των LLRs σε κάθε διαδρομή του trellis. Κάθε πιθανή μετάβαση στο trellis δεν έχει πλέον ένα απλό κόστος 0 ή 1 (όπως στην Απόσταση Hamming), αλλά ένα κόστος που προκύπτει από την LLR τιμή του αντίστοιχου bit. Ο αποκωδικοποιητής υπολογίζει το συσσωρευτικό κόστος για όλα τα πιθανά μονοπάτια και επιλέγει τη διαδρομή με το ελάχιστο συσσωρευτικό κόστος. Αυτό του επιτρέπει να λαμβάνει πιο ενημερωμένες αποφάσεις, καθώς λαμβάνει υπόψη την "εμπιστοσύνη" κάθε λαμβανόμενου bit, γεγονός που οδηγεί σε σημαντικά βελτιωμένη διόρθωση σφαλμάτων και μειώνει το Bit Error Rate (BER).

4.5.4 Σχόλια για την Εφαρμογή

Η ενσωμάτωση της soft-decision αποκωδικοποίησης στην υλοποίηση του RFSoc αποδείχθηκε κρίσιμη για την επίτευξη υψηλής απόδοσης. Ακόμη και σε ένα ελεγχόμενο καλωδιακό περιβάλλον, όπου οι εξωτερικές παρεμβολές είναι περιορισμένες, η βελτίωση στο Bit Error Rate (BER) είναι αισθητή, ειδικά σε χαμηλά SNR.

Η δυνατότητα εκτίμησης της διασποράς θορύβου σ^2 από τα πιλοτικά σήματα αποτελεί ένα βασικό πλεονέκτημα, καθώς επιτρέπει την προσαρμοστικότητα της αποδιαμόρφωσης σε δυναμικά επίπεδα θορύβου. Αυτό διασφαλίζει ότι οι υπολογισμοί των LLRs είναι ακριβείς, ανεξάρτητα από τις συνθήκες του καναλιού.

Η συνολική ροή της soft-decision αποκωδικοποίησης περιλαμβάνει τα εξής διαδοχικά στάδια:

1. Συμβολική ισοστάθμιση με MRC: Εφαρμογή του Maximum Ratio Combining για τον βέλτιστο συνδυασμό των σημάτων από τους δύο δέκτες.
2. Soft demapping σε LLRs: Μετατροπή των ισοσταθμισμένων συμβόλων σε Log-Likelihood Ratios, παρέχοντας πληροφορία εμπιστοσύνης.
3. Ενίσχυση εμπιστοσύνης: Κλιμάκωση των LLRs για την ενίσχυση της διαχωρισιμότητας και τη βελτιστοποίηση του Viterbi.
4. Soft-input Viterbi αποκωδικοποίηση: Τελική αποκωδικοποίηση χρησιμοποιώντας τις LLRs για τη διόρθωση σφαλμάτων.

Αυτή η αρχιτεκτονική αυξάνει σημαντικά την αξιοπιστία της επικοινωνίας και θέτει τις βάσεις για την ενσωμάτωση πιο προηγμένων συστημάτων FEC στο μέλλον, όπως κώδικες Turbo ή LDPC, οι οποίοι μπορούν να αξιοποιήσουν περαιτέρω την πληροφορία για ακόμα καλύτερη απόδοση.

4.6 Εκτίμηση Καναλιού και Ισοστάθμιση

Σε αυτό το στάδιο, το σήμα που έχει ήδη μετασχηματιστεί στο πεδίο της συχνότητας υποβάλλεται σε διαδικασία εκτίμησης καναλιού και ισοστάθμισης, με σκοπό τη διόρθωση των επιδράσεων του καναλιού στην πορεία μετάδοσης. Η υλοποίηση επικεντρώνεται στη χρήση Least Squares εκτίμησης με γραμμική παρεμβολή, χωρίς χρήση πληροφοριών για το SNR ή το θόρυβο πέραν των πιλοτικών.

Η εκτίμηση του καναλιού περιορίζεται στους υποφορείς που μεταφέρουν πιλοτικά σύμβολα, και γίνεται με απλή διαίρεση του λαμβανόμενου φασματικού δείγματος με το γνωστό αποσταλέν πιλοτικό. Για τις ενδιάμεσες συχνότητες (μη πιλοτικές), εφαρμόζεται παρεμβολή κατά μήκος των συχνοτήτων, με στόχο την πλήρη ανακατασκευή του φασματικού καναλιού.

Η ίδια διαδικασία εφαρμόζεται σε κάθε δέκτη ανεξάρτητα, οδηγώντας σε ένα εκτιμημένο κανάλι ανά υποφορέα και ανά κεραία λήψης. Κατόπιν, οι ληφθείσες φασματικές τιμές ισοσταθμίζονται με απλή διαίρεση ως προς τις αντίστοιχες εκτιμήσεις καναλιού.

Για το συνδυασμό των εκτιμήσεων σε SIMO διαμόρφωση ($N_r = 2$), αξιοποιείται η τεχνική Maximum Ratio Combining (MRC), η οποία υλοποιείται με στάθμιση των σημάτων κάθε δέκτη από τη συζυγή του εκτιμώμενου καναλιού και κανονικοποίηση με το συνολικό ενεργειακό βάρος. Ο συνδυασμός αυτός παρέχει ενίσχυση της αξιοπιστίας των φασματικών συνιστωσών του σήματος, ιδίως σε περιπτώσεις όπου οι δύο δέκτες λαμβάνουν διαφορετικές εκδοχές της μετάδοσης λόγω θορύβου ή καθυστέρησης.

Η έξοδος της ισοστάθμισης είναι ένα φασματικό σήμα έτοιμο για αποδιαμόρφωση, με τις παρεμβολές του καναλιού να έχουν ουσιαστικά αντισταθμιστεί.

4.7 Μέτρηση Απόδοσης: BER και EVM

Η αξιολόγηση της απόδοσης του συστήματος OFDM στο RFSoc βασίζεται σε δύο βασικά μετρικά: το Bit Error Rate και το Error Vector Magnitude, τα οποία αποτιμούν την ακρίβεια της αποδιαμορφωμένης πληροφορίας και την ποιότητα του διανυσματικού σήματος αντίστοιχα.

Η μέτρηση BER πραγματοποιείται με σύγκριση του αποκωδικοποιημένου bitstream με τα αρχικά διανυθέντα bits. Για την αποφυγή μετρήσεων σε πιλοτικές ή padding θέσεις, η σύγκριση περιορίζεται μόνο στα bits πληροφορίας που βρίσκονται σε έγκυρες θέσεις. Στην περίπτωση soft-decision αποκωδικοποίησης, τα LLRs μετατρέπονται σε bits πριν από την αντιπαραβολή με το αρχικό stream.

Η EVM μετριέται στο πεδίο της συχνότητας, αφού έχει ολοκληρωθεί η ισοστάθμιση. Ορίζεται ως το μέσο τετραγωνικό σφάλμα μεταξύ των ληφθέντων σημείων, μετά την ισοστάθμιση, και των ιδανικών σημείων του αστερισμού. Η τιμή EVM εκφράζεται είτε σε ποσοστιαία μορφή είτε σε dB ($20 \cdot \log_{10}$), και προσφέρει πληροφορία για το συνολικό ποιοτικό σφάλμα του συστήματος λόγω θορύβου, μη ιδανικής ισοστάθμισης ή υπολειμματικών ασυμμετριών του καναλιού.

Η λήψη αυτών των μετρικών παρέχει ποσοτική εικόνα της απόδοσης της υλοποίησης πάνω στο RFSoc και μπορεί να αξιοποιηθεί για μελλοντική βελτιστοποίηση.

5

Συγκριτική Αξιολόγηση και

Πειραματικά Αποτελέσματα

Η μετάβαση από προσομοιωμένο OFDM σύστημα σε πλήρη υλοποίηση πάνω στην πλατφόρμα Xilinx RFSoc 4x2 συνοδεύτηκε από συστηματικό πειραματισμό για την επαλήθευση των επιδόσεων σε περιβάλλον πραγματικού χρόνου. Στο παρόν κεφάλαιο παρουσιάζονται τα βασικά αποτελέσματα και η ανάλυση της απόδοσης του υλοποιημένου συστήματος, εστιάζοντας στη βελτίωση που επέφεραν η MIMO διαμόρφωση και η soft-decision αποκωδικοποίηση, όπως εισήχθησαν στο Κεφάλαιο 4.

5.1 Πλαίσιο Πειραματισμού

Η αξιολόγηση της συνολικής απόδοσης του συστήματος πραγματοποιείται στο πλαίσιο μιας συγκριτικής μελέτης μεταξύ της αρχικής προσομοίωσης σε Python και της αντίστοιχης υλοποίησης στο υλικό της πλατφόρμας Xilinx RFSoc 4x2. Η παρούσα υποενότητα περιγράφει αναλυτικά τις βασικές παραμέτρους και διαφορές των δύο υλοποιήσεων, έτσι ώστε να θεμελιωθεί ένα κοινό πειραματικό πλαίσιο σύγκρισης.

Παρότι η βασική δομή του συστήματος OFDM και οι αλγόριθμοι που χρησιμοποιούνται παραμένουν κοινές και στις δύο υλοποιήσεις, η πραγματική εφαρμογή στο RFSoc συνοδεύεται από συγκεκριμένους περιορισμούς και τεχνικές τροποποιήσεις, που επιδρούν στον χρονισμό, τη θορυβώδη συμπεριφορά και τη διαχείριση δειγμάτων.

Ο παρακάτω πίνακας συνοψίζει τα βασικά χαρακτηριστικά των δύο περιβαλλόντων:

Παράμετρος	Προσομοίωση (Python)	Υλοποίηση στο RFSoc
Αριθμός υποφερόντων (K)	128	128
Μήκος κυκλικού προθέματος (CP)	32	32
Τύπος διαμόρφωσης	16-QAM	16-QAM
Δομή δεκτών	SIMO	MIMO 2x2
Συγχρονισμός	Zadoff-Chu + Schmidl & Cox	Zadoff-Chu + Schmidl & Cox
Εκτίμηση καναλιού	LS με παρεμβολή πιλοτικών	LS με παρεμβολή πιλοτικών

Ισοστάθμιση	MRC	MRC
FEC αποκωδικοποίηση	Hard-Decision Viterbi	Soft-Decision Viterbi
Περιβάλλον καναλιού	TDL-D (με Rician fading)	Loopback μέσω καλωδίου (RF path)
Τύπος θορύβου	AWGN	Θόρυβος καλωδίων, ADC, DAC και interconnect jitter
Χρονισμός	Ιδανικός (πλήρης έλεγχος σε software)	Μεταβλητός λόγω DMA latency & phase offset

Η κατανόηση αυτών των διαφορών είναι κρίσιμη για την ερμηνεία των πειραματικών αποτελεσμάτων που παρουσιάζονται στα επόμενα τμήματα, καθώς οι αποκλίσεις στο hardware εισάγουν φαινόμενα που δεν μοντελοποιούνται εύκολα στην προσομοίωση. Ειδικά, οι αποκρίσεις των μετατροπέων (DAC/ADC), η συμπεριφορά των buffer DMA και οι μικροαποκλίσεις φάσης επηρεάζουν άμεσα την απόδοση του συστήματος στο πεδίο.

5.2 Χαρακτηριστικά Διαμόρφωσης και Χωρικής Λήψης

Η παρούσα ενότητα εστιάζει στη σύγκριση των χαρακτηριστικών διαμόρφωσης και της χωρικής λήψης μεταξύ της προσομοίωσης λογισμικού (που περιγράφηκε στο Κεφάλαιο 3) και της πειραματικής υλοποίησης στην πλατφόρμα RFSoc (που αναπτύχθηκε στο Κεφάλαιο 4). Ο βασικός στόχος είναι η επικύρωση ότι η πειραματική διάταξη στο RFSoc λειτουργεί σύμφωνα με τις προδιαγραφές και ότι τα σήματα που παράγονται και λαμβάνονται από το υλικό είναι συνεπή με τις θεωρητικές προσδοκίες και τα αποτελέσματα της λογισμικής προσομοίωσης.

5.2.1 Κατανομή Υποφερόντων και Χαρτογράφηση Συμβόλων

Αρχικά, εξετάζεται η κατανομή των υποφερόντων εντός του σήματος OFDM. Όπως αναμένεται και φαίνεται στο διάγραμμα "OFDM Subcarrier Allocation" 5.1, οι υποφέροντες διαχωρίζονται σαφώς σε δύο κύριες κατηγορίες:

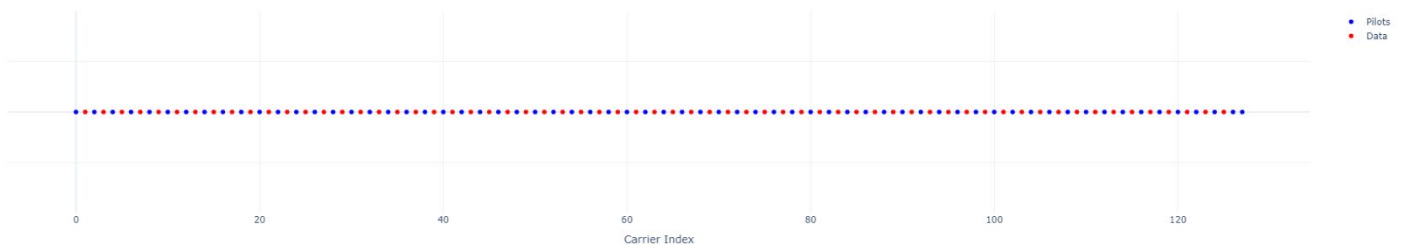
- Πιλοτικοί υποφέροντες: Χρησιμοποιούνται για την εκτίμηση του καναλιού και τον συγχρονισμό.
- Υποφέροντες δεδομένων: Μεταφέρουν την πληροφορία των χρηστών.

Η κατανομή των πιλοτικών υποφερόντων είναι ομοιόμορφη στο φάσμα, εξασφαλίζοντας την αποτελεσματική και ακριβή εκτίμηση των ιδιοτήτων του καναλιού σε όλο το εύρος ζώνης του σήματος. Αυτή η διάταξη είναι θεμελιώδης για την ορθή λειτουργία του ισοσταθμιστή στον δέκτη.

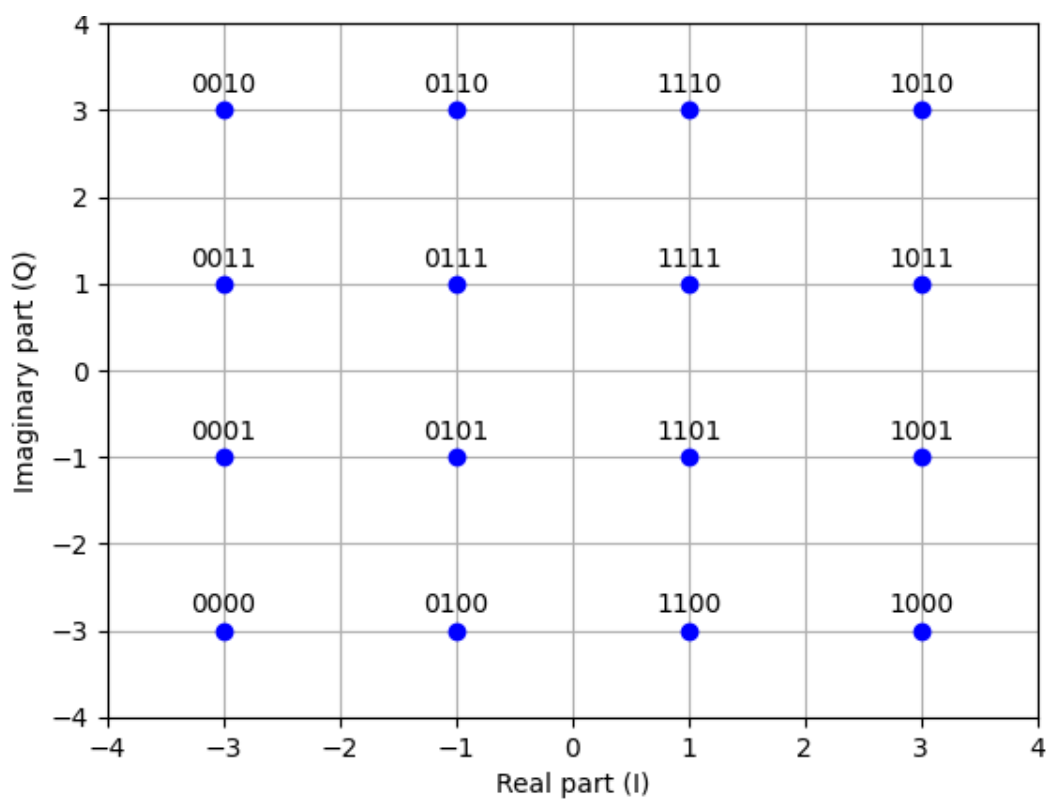
Στη συνέχεια, απεικονίζεται η χαρτογράφηση των συμβόλων Gray 16-QAM στο διάγραμμα 5.2. Αυτό το διάγραμμα επιβεβαιώνει την ορθή αντιστοίχιση των δυαδικών ψηφίων (bits) σε σύνθετα σύμβολα (points) στον αστερισμό 16-QAM. Η χρήση της κωδικοποίησης Gray-mapping είναι κρίσιμη για τη μείωση του αριθμού των σφαλμάτων κατά την αποδιαμόρφωση. Σε περίπτωση που ο δέκτης αποφασίσει λανθασμένα για ένα λαμβανόμενο σύμβολο, αλλά η απόφασή του είναι σε ένα γειτονικό σημείο του

αστερισμού, λόγω του Gray coding, αυτή η λανθασμένη εκτίμηση οδηγεί συνήθως σε αλλαγή ενός μόνο bit (ή μικρού αριθμού bits), ελαχιστοποιώντας έτσι το συνολικό Bit Error Rate (BER).

OFDM Subcarrier Allocation



Εικόνα 9: Κατανομή υποφερόντων στο OFDM σύμβολο



Εικόνα 10: Σηματοστερισμός 16-QAM με κώδικα Gray

5.2.2 Σύγκριση Μεγέθους Σήματος Εκπομπής-Λήψης στο Πεδίο του Χρόνου

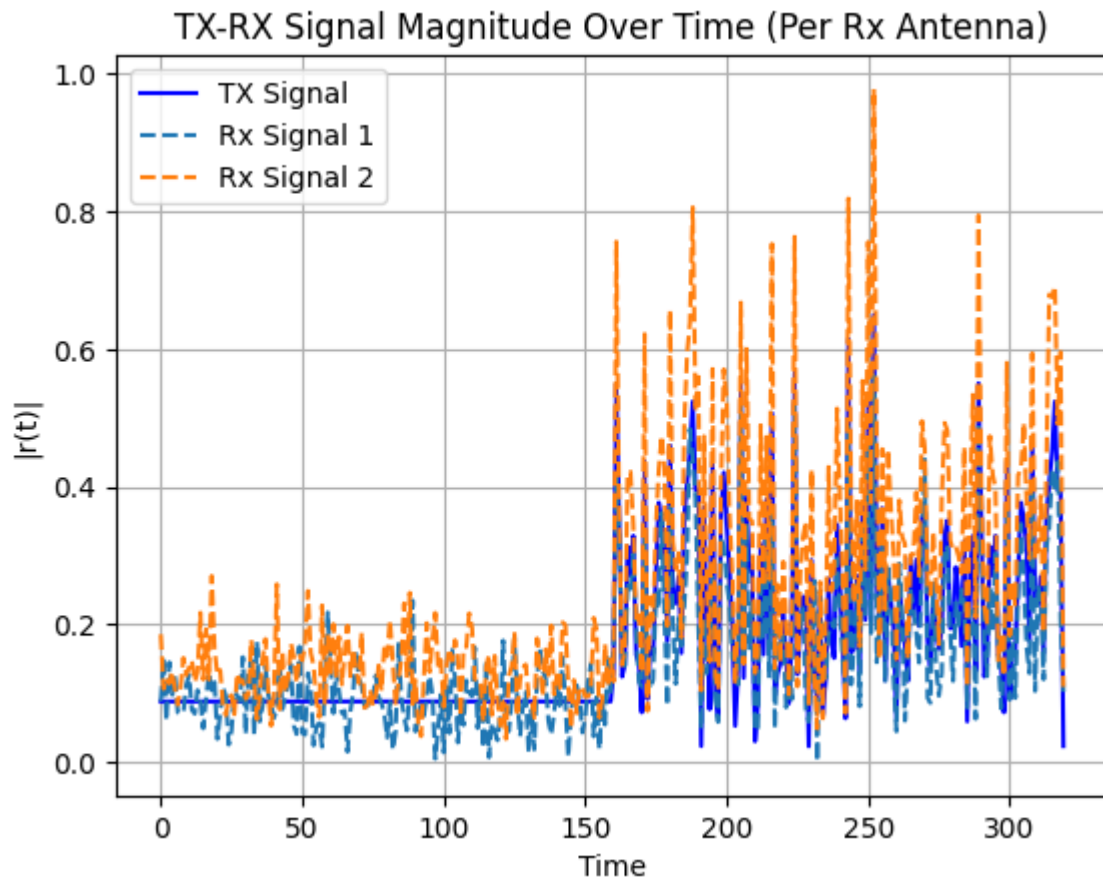
Ένα από τα πιο διαφωτιστικά διαγράμματα για την αξιολόγηση της συμπεριφοράς του συστήματος είναι το "TX-RX Signal Magnitude Over Time (Per Rx Antenna)", το οποίο παρουσιάζει το πλάτος του σήματος τόσο από την πλευρά της εκπομπής όσο και από την πλευρά της λήψης στο πεδίο του χρόνου, για κάθε κεραία δέκτη.

- Από τη λογισμική προσομοίωση: Στο αντίστοιχο διάγραμμα της προσομοίωσης, παρατηρείται ότι το σήμα λήψης ακολουθεί στενά το σήμα εκπομπής. Οι μεταβάσεις μεταξύ των συμβόλων

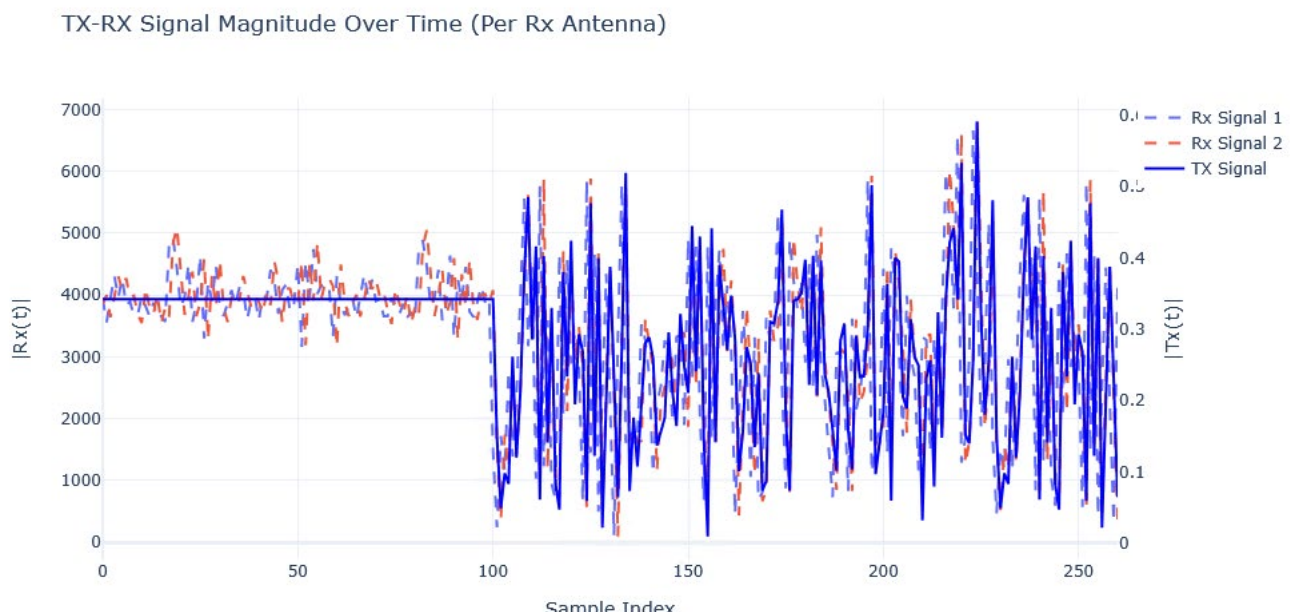
είναι καθαρές και οι διακυμάνσεις στο μέγεθος είναι περιορισμένες. Αυτό οφείλεται στην προσομοίωση ιδανικών συνθηκών, όπου ο θόρυβος περιορίζεται σε ελεγχόμενο Προσθετικό Λευκό Γκαουσιανό Θόρυβο (AWGN) και το fading προσομοιώνεται με μοντέλα όπως το Rician fading, χωρίς να λαμβάνονται υπόψη οι πολυπλοκότητες του πραγματικού υλικού.

- Από την υλοποίηση στο RFSoc: Αντιθέτως, στο διάγραμμα που αφορά την υλοποίηση στο RFSoc, παρατηρείται σημαντικά μεγαλύτερη μεταβλητότητα στο μέγεθος του σήματος λήψης. Αυτή η μεταβλητότητα είναι ένα αναμενόμενο και ρεαλιστικό χαρακτηριστικό που οφείλεται σε διάφορους παράγοντες που εγγενείς στις πραγματικές συνθήκες hardware και στην καλωδιακή σύνδεση:
 - Θόρυβος των ADC: Οι αναλογικοί-ψηφιακοί μετατροπείς εισάγουν το δικό τους θόρυβο κατά τη διαδικασία της ψηφιοποίησης.
 - Μη ιδανική ευθυγράμμιση στο χρόνο και τη φάση: Παρά τους μηχανισμούς συγχρονισμού, μικρές αποκλίσεις και σφάλματα φάσης μπορούν να εισαχθούν από το hardware.
 - Υπολειμματικό In-phase/Quadrature (I/Q) Imbalance: Μικρές αναντιστοιχίες μεταξύ του I και Q καναλιού στους αναλογικούς δρόμους.
 - Πρακτικοί παράγοντες: Άλλοι παράγοντες όπως οι απώλειες στα καλώδια, θόρυβος εισαγόμενος στα καλώδια, η μη ιδανική συμπεριφορά των ενισχυτών, και η παρουσία υπολειμματικών παρεμβολών, οι οποίοι απουσιάζουν από την ιδανική προσομοίωση.

Η σύγκριση αυτών των δύο διαγραμμάτων (από προσομοίωση και RFSoc) είναι εξαιρετικά διαφωτιστική. Καταδεικνύει τις ουσιαστικές διαφορές μεταξύ ιδανικών και πραγματικών συνθηκών και, κυρίως, επιβεβαιώνει τη λειτουργικότητα της διάταξης RFSoc ως αξιόπιστη και ρεαλιστική πλατφόρμα για πειραματική έρευνα και ανάπτυξη συστημάτων OFDM με χωρική λήψη. Η ικανότητα του συστήματος στο RFSoc να διαχειρίζεται αυτές τις πραγματικές ατέλειες και να παράγει ένα αναγνωρίσιμο σήμα είναι απόδειξη της επιτυχούς υλοποίησής του.



Εικόνα 11: Πλάτος σήματος σε πομπό και δέκτες με χρήση καλωδίων (RFSoc)



Εικόνα 12: Πλάτος σήματος σε πομπό και δέκτες με εξομοίωση

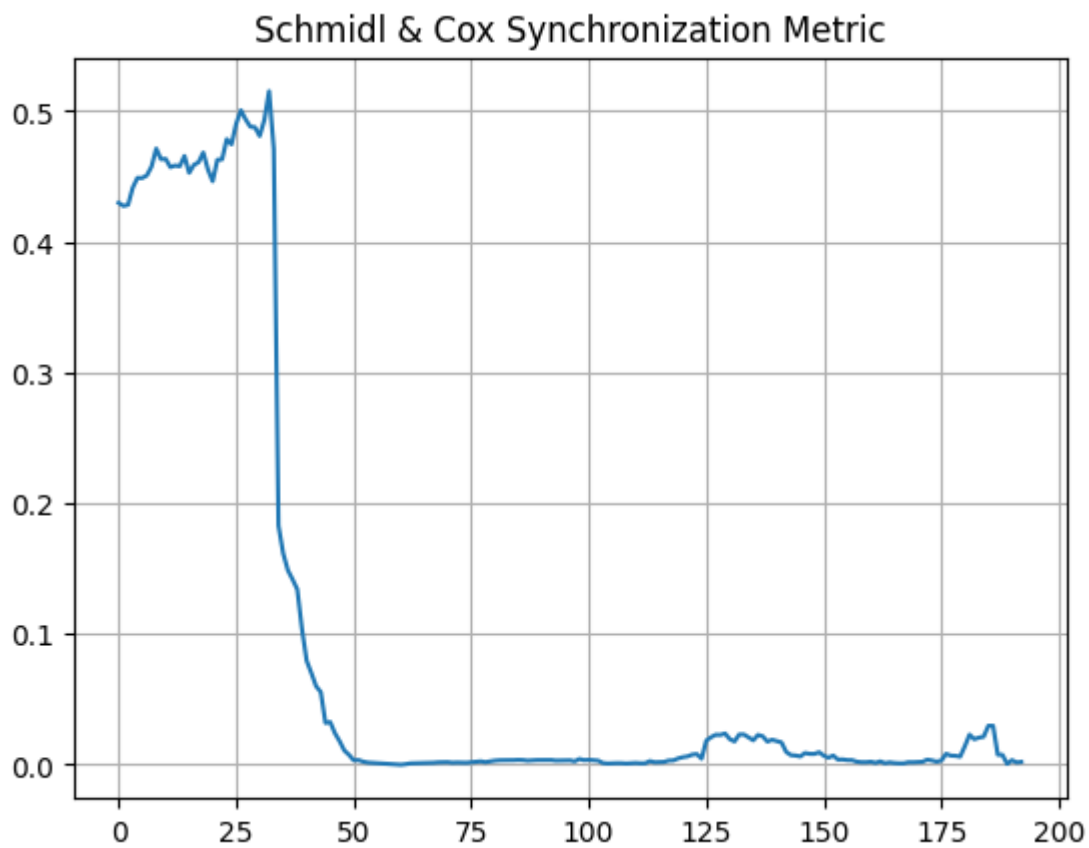
5.3 Συγχρονισμός και Χρονική Ευθυγράμμιση

Στο παρόν υποκεφάλαιο παρουσιάζεται λεπτομερώς η διαδικασία του συγχρονισμού και της χρονικής ευθυγράμμισης του OFDM σήματος, συγκρίνοντας τα αποτελέσματα που λαμβάνονται από το λογισμικό μοντέλο προσομοίωσης (Κεφάλαιο 3) με αυτά της πραγματικής υλοποίησης στην πλατφόρμα RFSoc (Κεφάλαιο 4). Η ακριβής χρονική και συχνотική ευθυγράμμιση είναι θεμελιώδους σημασίας για την επιτυχή αποδιαμόρφωση και αποκωδικοποίηση των OFDM συμβόλων.

5.3.1 Μετρική Schmidl & Cox για Συγχρονισμό

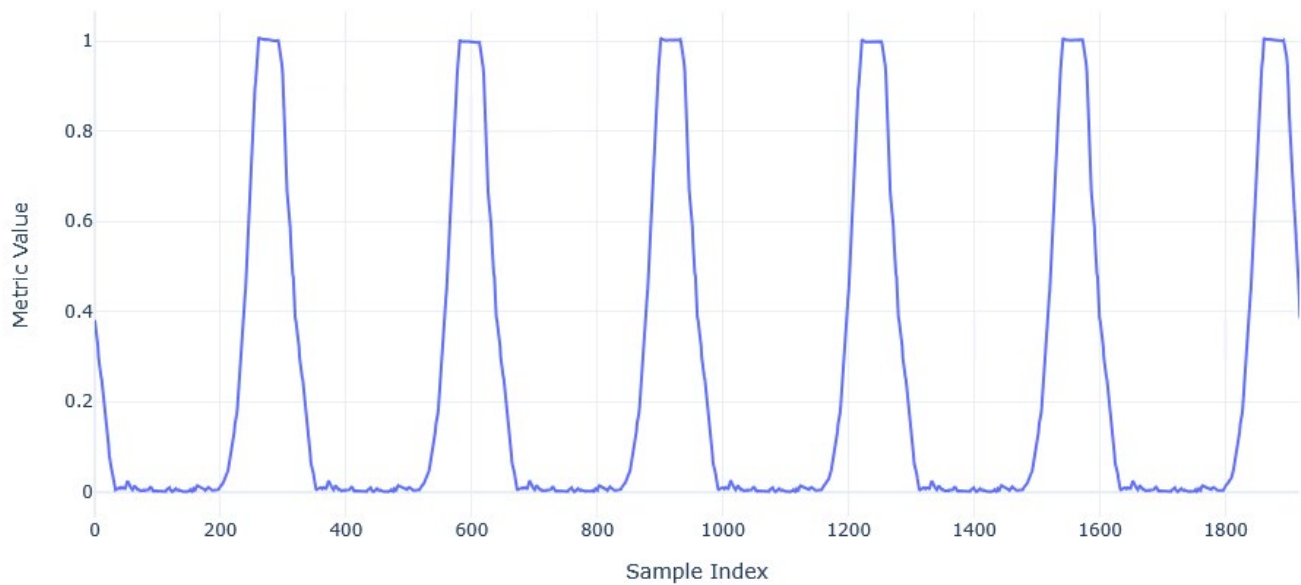
Η μετρική Schmidl & Cox αποτελεί μια ευρέως χρησιμοποιούμενη τεχνική για τον εντοπισμό της έναρξης του OFDM συμβόλου και την εκτίμηση της συχνотικής απόκλισης, αξιοποιώντας τη δομή του κυκλικού προθέματος (Cyclic Prefix) και την επανάληψη του σήματος στην αρχή του OFDM συμβόλου.

- Στην προσομοίωση: Όπως φαίνεται στο διάγραμμα "Schmidl & Cox Synchronization Metric (Simulation)", η μετρική παρουσιάζει μια σαφή και ευκρινή κορυφή στο σημείο όπου ξεκινά το OFDM σύμβολο. Αυτή η κορυφή είναι χαρακτηριστική και καθιστά την αυτόματη ευθυγράμμιση εύκολη και εξαιρετικά ακριβή, καθώς το περιβάλλον προσομοίωσης είναι ιδανικό και απαλλαγμένο από ανεπιθύμητους θορύβους και ατέλειες του υλικού.
- Στο πραγματικό υλικό του RFSoc: Αντίθετα, στο διάγραμμα "Schmidl & Cox Synchronization Metric (RFSoc Implementation)", παρατηρούνται πολλαπλές, περιοδικές κορυφές. Αυτό το φαινόμενο οφείλεται στην επανάληψη του σήματος μέσα στον DMA buffer των DACs (όπως περιγράφηκε στο Κεφάλαιο 4.2), όπου το OFDM πακέτο αναπαράγεται πολλές φορές για να γεμίσει πλήρως τον buffer. Κάθε κορυφή αντιστοιχεί στην έναρξη μιας εκ των επαναλήψεων του πακέτου. Αυτή η συμπεριφορά απαιτεί προσεκτική επιλογή του σωστού σημείου ευθυγράμμισης ανάμεσα στις επαναλήψεις, συνήθως επιλέγοντας την πρώτη ή μια προκαθορισμένη επανάληψη, ώστε να αποφευχθεί η λήψη δεδομένων από λανθασμένα σημεία του επαναλαμβανόμενου σήματος. Παρά τις πολλαπλές κορυφές, η ύπαρξη σαφών κορυφών επιβεβαιώνει την επιτυχή λειτουργία του αλγόριθμου συγχρονισμού και στο πραγματικό υλικό.



Εικόνα 13: Μετρικό Schmidle and Cox για συγχρονισμό

Schmidl & Cox Synchronization Metric

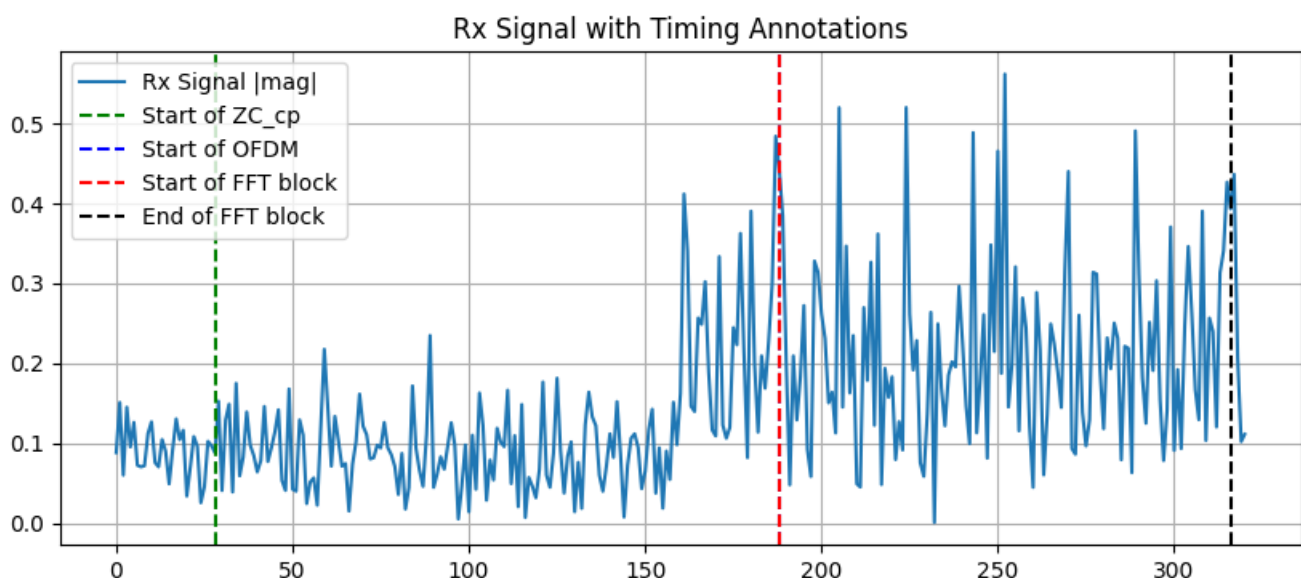


Εικόνα 14: Μετρικό συγχρονισμού για πολλαπλές επαναλήψεις του σήματος αναφοράς

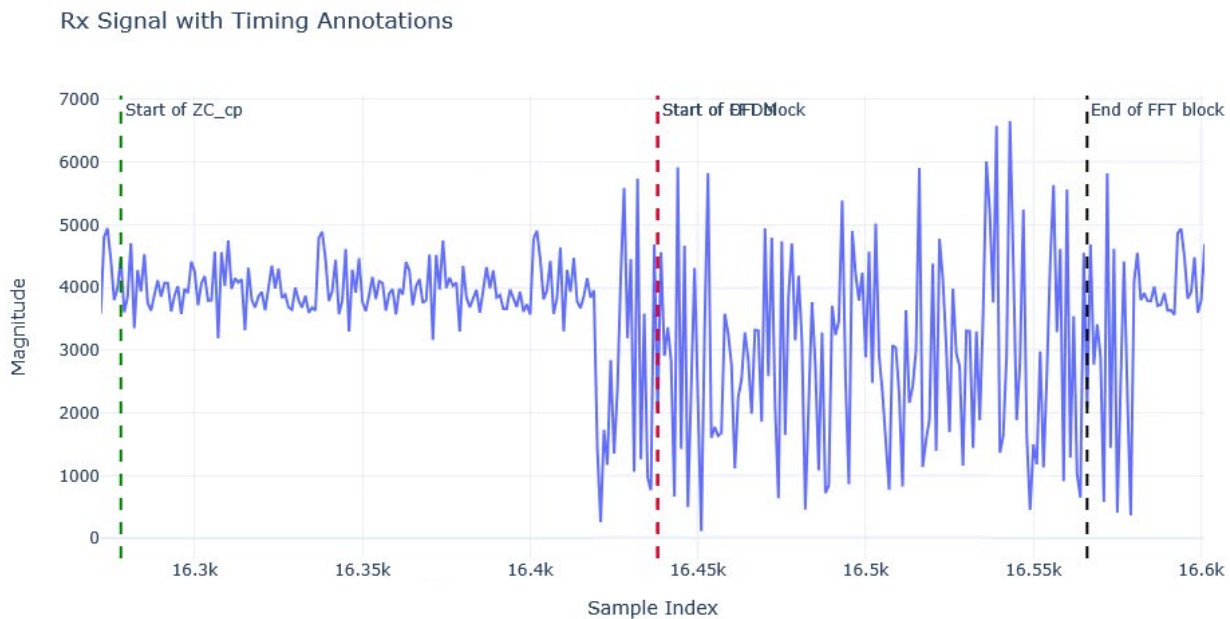
5.3.2 Σήμα Λήψης με Ενδείξεις Χρονικής Στόχευσης

Η απεικόνιση του σήματος λήψης με ενδείξεις χρονικής στόχευσης παρέχει μια οπτική επιβεβαίωση της ακρίβειας του συγχρονισμού. Τα διακριτά σημεία αναφοράς στο διάγραμμα επισημαίνουν τη θέση έναρξης της ακολουθίας Zadoff-Chu (που χρησιμοποιείται για συγχρονισμό πλαισίου), την έναρξη του ίδιου του OFDM συμβόλου και, τέλος, την περιοχή όπου εφαρμόζεται ο Μετασχηματισμός FFT.

- Στην προσομοίωση: Στο διάγραμμα "Rx Signal with Timing Annotations (Simulation)", τα σημεία αυτά είναι σταθερά, ακριβή και σαφώς καθορισμένα λόγω των ιδανικών συνθηκών προσομοίωσης. Η χρονική ακρίβεια είναι σχεδόν τέλεια.
- Στο περιβάλλον του RFSoc: Αντιθέτως, στο διάγραμμα "Rx Signal with Timing Annotations (RFSoc Implementation)", ενώ οι ενδείξεις είναι παρούσες, παρατηρούνται μικρές διακυμάνσεις και αβεβαιότητες στις χρονικές ενδείξεις. Αυτές οι αποκλίσεις οφείλονται στην παρουσία πρόσθετων παραγόντων που είναι εγγενείς στην υλοποίηση υλικού:
 - Εσωτερικές καθυστερήσεις DMA: Διαφορές στους χρόνους μεταφοράς δεδομένων μέσω DMA.
 - Jitter από τον μηχανισμό χρονισμού: Μικρές, τυχαίες διακυμάνσεις στην ακρίβεια του χρονισμού του συστήματος.
 - Θερμοκρασιακές και περιβαλλοντικές επιδράσεις: Ακόμη και μικρές αλλαγές σε αυτές τις συνθήκες μπορούν να επηρεάσουν την ακρίβεια του υλικού. Αυτοί οι παράγοντες απαιτούν λεπτό χειρισμό και πιθανώς πρόσθετους αλγορίθμους προσαρμογής για τη διατήρηση της μέγιστης δυνατής ακρίβειας στη χρονική ευθυγράμμιση του σήματος.



Εικόνα 15: Το λαμβανόμενο σήμα με ένδειξη της έναρξης του σήματος Zadoff-Chu, και της έναρξης και λήξης του συμβόλου πληροφορίας (εξομοίωση)



Εικόνα 16: Το λαμβανόμενο σήμα με ένδειξη της έναρξης του σήματος Zadoff-Chu (ενισχυμένη ακολουθία), και της έναρξης και λήξης του συμβόλου πληροφορίας (RFSoc)

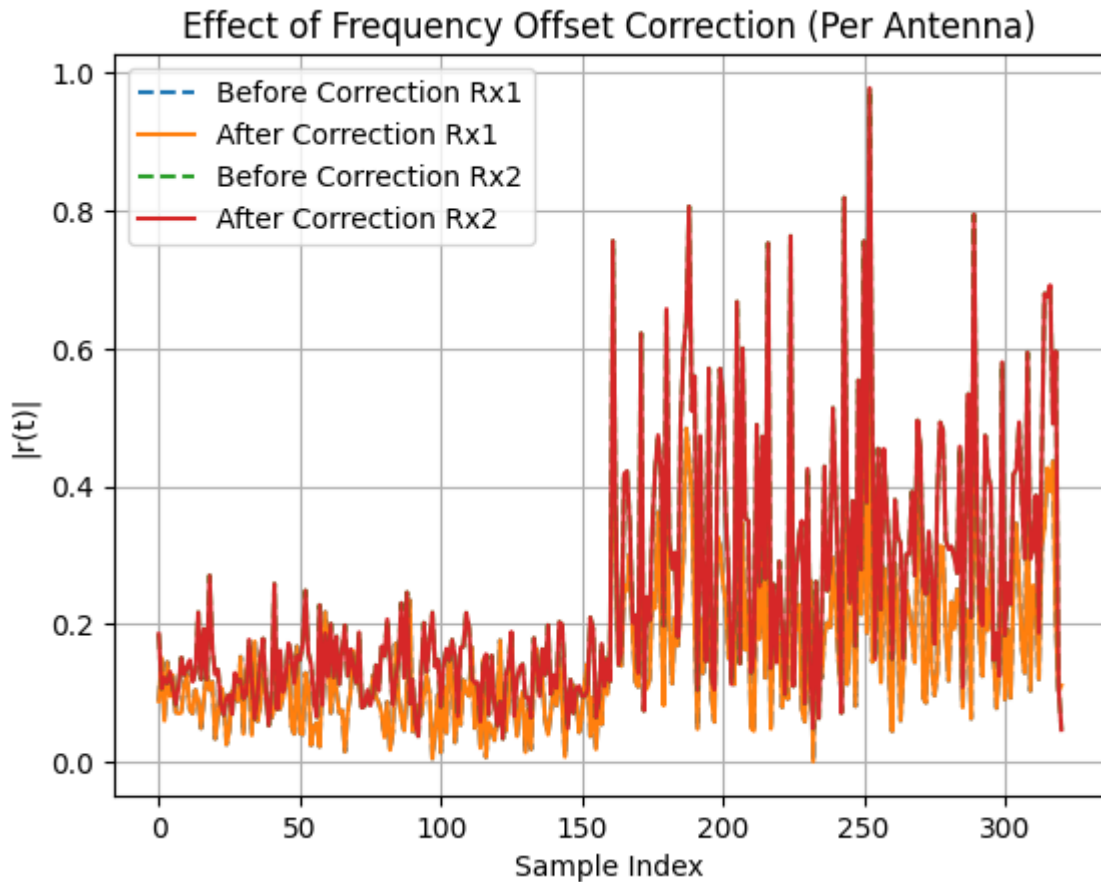
5.3.3 Επίδραση Διόρθωσης Συχνότητας

Η διόρθωση της συχνότητας είναι ένα κρίσιμο στάδιο για την απόδοση του συστήματος OFDM. Διορθώνει τις διαφορές φάσης και τις συχνοτικές μετατοπίσεις (carrier frequency offset - CFO) που μπορεί να προκληθούν από μη ιδανικές συμπεριφορές των ταλαντωτών του πομπού και του δέκτη, τους ADCs, καθώς και άλλους παράγοντες του υλικού. Η μη διόρθωση του CFO μπορεί να οδηγήσει σε διαυποφέρουσα παρεμβολή (Inter-Carrier Interference - ICI) και σημαντική υποβάθμιση της απόδοσης.

- Στην προσομοίωση: Στο διάγραμμα "Effect of Frequency Offset Correction (Simulation)", η επίδραση της διόρθωσης είναι άμεσα εμφανής. Μετά την εφαρμογή της διόρθωσης, παρατηρείται ξεκάθαρη μείωση της μεταβλητότητας του πλάτους του σήματος στο πεδίο του χρόνου ή μια σταθεροποίηση της φάσης στο πεδίο συχνότητας, υποδεικνύοντας την επιτυχή εξάλειψη της συχνοτικής απόκλισης.
- Στην υλοποίηση στο RFSoc: Αντίστοιχα, στο διάγραμμα "Effect of Frequency Offset Correction (RFSoc Implementation)", η διόρθωση συχνότητας είναι εξίσου αποτελεσματική στην άρση της συχνοτικής απόκλισης. Ωστόσο, οι συνολικές διακυμάνσεις του πλάτους και ο υπολειπόμενος θόρυβος παραμένουν υψηλότεροι σε σύγκριση με την προσομοίωση. Αυτό οφείλεται στα εγγενή χαρακτηριστικά των πραγματικών ADCs και DACs, τον θόρυβο του υλικού, καθώς και τις πρακτικές ατέλειες που είναι αναπόφευκτες σε ένα φυσικό σύστημα. Παρόλα αυτά, η επιτυχία της διόρθωσης συχνότητας στο RFSoc είναι μια ισχυρή ένδειξη της robust λειτουργίας του συστήματος.

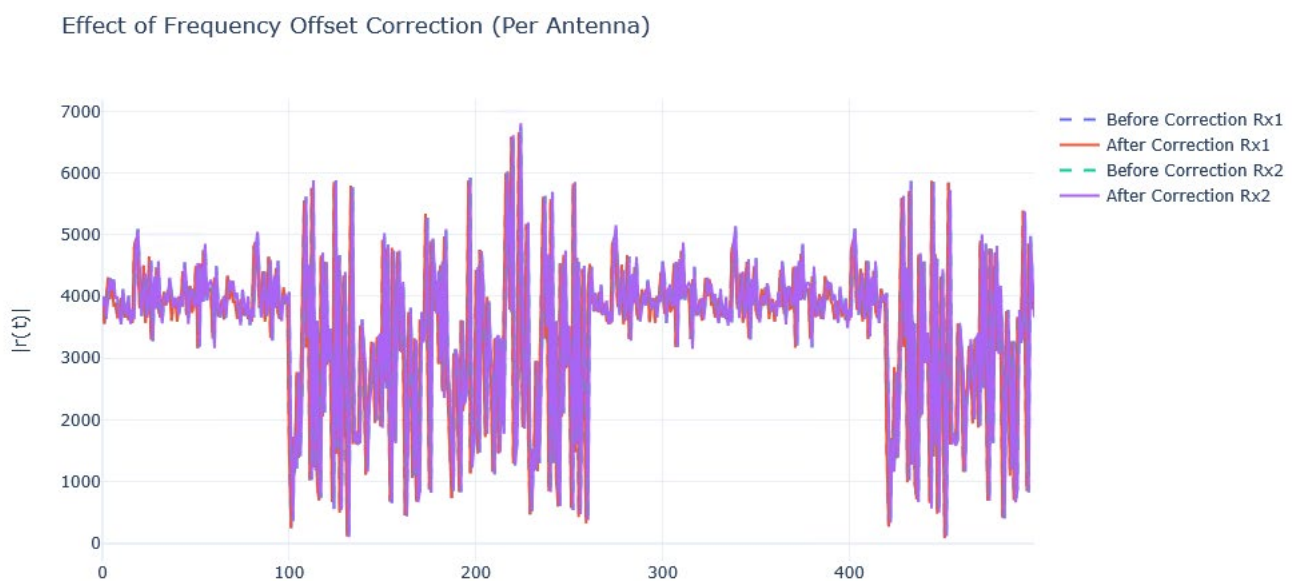
Συνολικά, τα αποτελέσματα των αναλύσεων συγχρονισμού καταδεικνύουν ότι το σύστημα RFSoc ανταποκρίνεται επιτυχώς στις απαιτήσεις συγχρονισμού της OFDM διαμόρφωσης, παρά τις προκλήσεις που εισάγει το πραγματικό υλικό. Η σύγκριση των δύο περιβαλλόντων (λογισμικού και υλικού)

αναδεικνύει τη σημασία της προσεκτικής διαχείρισης του hardware synchronization και της αξιολόγησης των μεθόδων επεξεργασίας σημάτων σε πραγματικά σενάρια εφαρμογής, προσφέροντας



Εικόνα 17: Επίδραση της μετατόπισης συχνότητας στο σήμα (εξομοίωση)

πολύτιμες γνώσεις για την πρακτική απόδοση των ασύρματων συστημάτων.



Εικόνα 18: Επίδραση της μετατόπισης συχνότητας στο σήμα (RFSoc)

5.4 Εκτίμηση Καναλιού και Ισοστάθμιση

Η εκτίμηση καναλιού και η ισοστάθμιση αποτελούν κρίσιμα στάδια για την αξιοπιστία και ακρίβεια ενός συστήματος OFDM, καθώς αντισταθμίζουν τις παραμορφώσεις που εισάγει το κανάλι μετάδοσης. Στην παρούσα ενότητα, πραγματοποιείται λεπτομερής σύγκριση και ανάλυση των επιδόσεων των μεθόδων που υλοποιήθηκαν σε περιβάλλον λογισμικής προσομοίωσης έναντι των αποτελεσμάτων που προέκυψαν από την πειραματική υλοποίηση στο RFSoc.

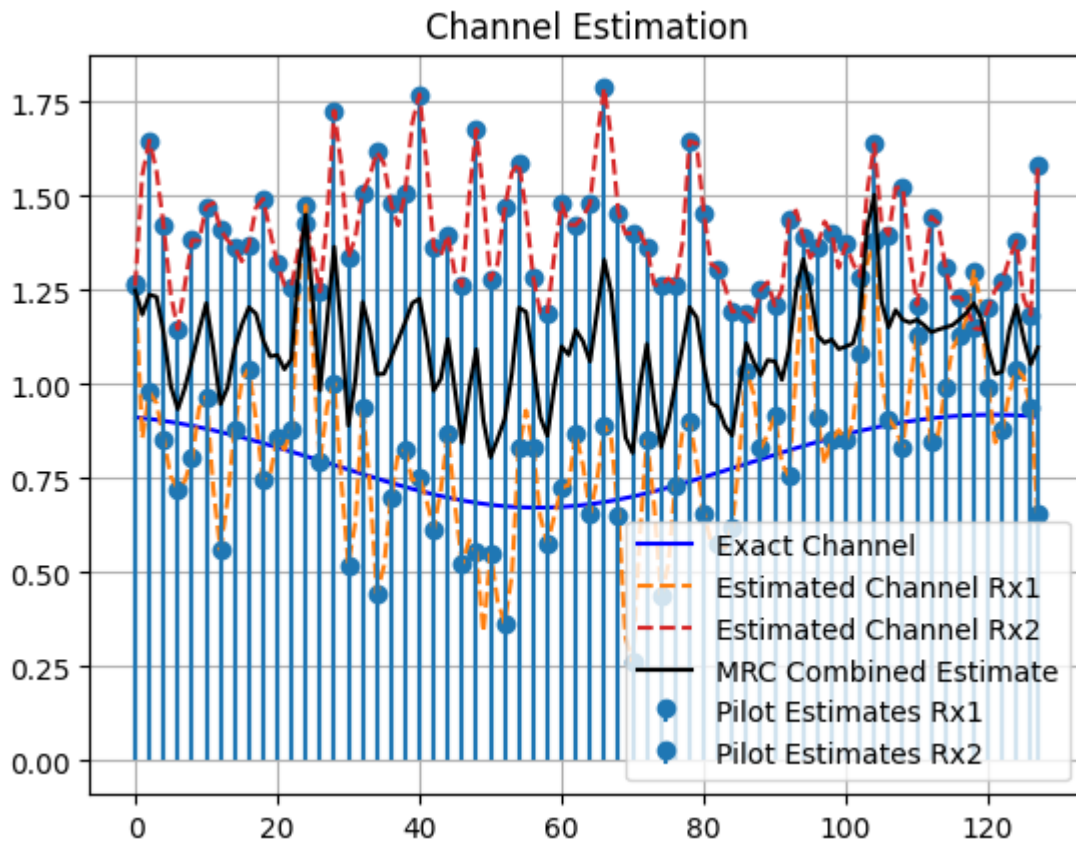
5.4.1 Εκτίμηση Καναλιού με *Least Squares (LS)* και *Maximum Ratio Combining (MRC)*

Η πρώτη σύγκριση αφορά την ίδια την εκτίμηση του καναλιού, η οποία πραγματοποιείται χρησιμοποιώντας τους πιλοτικούς υποφέροντες (pilot subcarriers) που είναι ενσωματωμένοι στο OFDM σύμβολο. Η μέθοδος που χρησιμοποιείται είναι η *Least Squares (LS)*, η οποία προσδιορίζει τους συντελεστές του καναλιού ελαχιστοποιώντας το τετράγωνο της διαφοράς μεταξύ των λαμβανόμενων πιλοτικών συμβόλων και των γνωστών μεταδιδόμενων πιλοτικών συμβόλων. Μετά την εκτίμηση του καναλιού για κάθε δέκτη στο MIMO σύστημα (στην περίπτωση μας, δύο δέκτες), εφαρμόζεται η τεχνική *Maximum Ratio Combining (MRC)*, η οποία συνδυάζει τα σήματα από τους διαφορετικούς δέκτες, σταθμίζοντας τα με βάση την ισχύ του καναλιού, ώστε να μεγιστοποιηθεί ο λόγος σήματος προς θόρυβο (SNR) στην έξοδο.

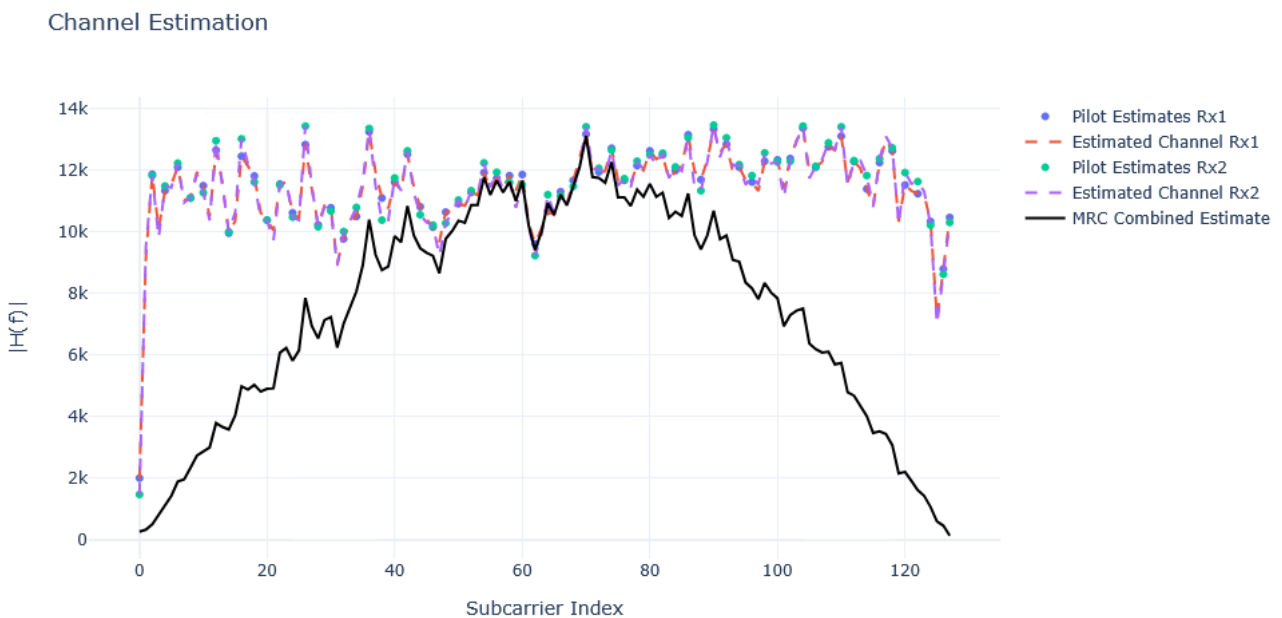
Τα διαγράμματα που χρησιμοποιήθηκαν για αυτήν την ανάλυση είναι:

- Προσομοίωση: "Channel Estimation (Simulation)"
- RFSoc: "Channel Estimation (RFSoc Implementation)"
- Στην προσομοίωση: Το διάγραμμα της εκτίμησης καναλιού παρουσιάζει χαρακτηριστική ομαλότητα με χαμηλή μεταβλητότητα. Αυτό είναι απολύτως αναμενόμενο, καθώς το θεωρητικό μοντέλο προσομοίωσης λειτουργεί σε ιδανικές συνθήκες, με ελεγχόμενο θόρυβο και προβλέψιμη συμπεριφορά καναλιού. Η εκτιμώμενη απόκριση του καναλιού είναι σχεδόν τέλεια, αντανakλώντας την απουσία πρακτικών ατελειών.
- Στο RFSoc: Αντίθετα, η εκτίμηση καναλιού στο RFSoc εμφανίζει υψηλότερη διακύμανση και θόρυβο. Αυτό το γεγονός οφείλεται σε ρεαλιστικές ατέλειες που είναι εγγενείς στην υλοποίηση υλικού, όπως:
 - Θόρυβος των ADC/DAC: Οι μετατροπείς αναλογικού-ψηφιακού και ψηφιακού-αναλογικού εισάγουν το δικό τους θόρυβο.
 - Υπολειμματικά σφάλματα συγχρονισμού: Παρά τις μεθόδους συγχρονισμού, μικρές υπολειμματικές χρονικές και συχνотικές μετατοπίσεις μπορεί να παραμείνουν.
 - Μη ιδανική συμπεριφορά υλικού: Πραγματικές ιδιότητες των κυκλωμάτων που δεν είναι τέλεια γραμμικές ή σταθερές.
 - Περιορισμοί ακρίβειας: Η ακρίβεια των πράξεων στα ψηφιακά κυκλώματα.

Παρόλα αυτά, από την αντιπαράθεση των δύο διαγραμμάτων προκύπτει ότι η εκτίμηση καναλιού στο RFSoc, παρότι πιο θορυβώδης, παραμένει συνεπής. Οι εκτιμώμενοι συντελεστές του καναλιού ακολουθούν σαφώς την τάση των πιλοτικών συμβόλων, υποδεικνύοντας ότι η μέθοδος LS είναι αξιόπιστη και λειτουργική στην πειραματική εφαρμογή. Η ύπαρξη αυτού του θορύβου στην εκτίμηση είναι ένα ρεαλιστικό στοιχείο που πρέπει να αντιμετωπιστεί από τα επόμενα στάδια επεξεργασίας.



Εικόνα 19: Εκτίμηση καναλιού (εξομοίωση)



Εικόνα 20: Εκτίμηση καναλιού (RFSoc)

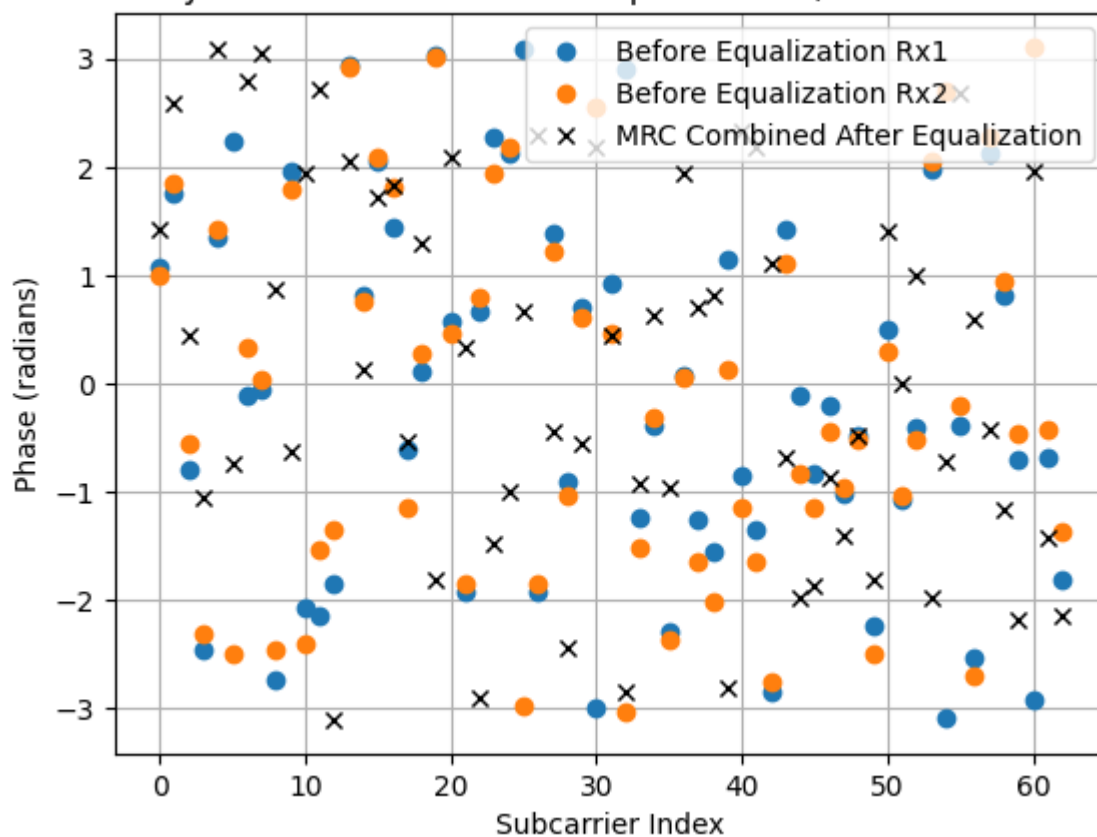
5.4.2 Φασική Συμπεριφορά πριν και μετά την Ισοστάθμιση

Η δεύτερη σύγκριση επικεντρώνεται στη φασική συμπεριφορά των συμβόλων QAM στο διάγραμμα αστερισμού (constellation diagram) πριν και μετά την εφαρμογή της ισοστάθμισης MRC. Η ισοστάθμιση στοχεύει στην αντιστάθμιση των φασικών και πλάτους παραμορφώσεων που προκαλεί το κανάλι, ώστε τα λαμβανόμενα σύμβολα να συγκλίνουν προς τις ιδανικές τους θέσεις στον αστερισμό.

Τα διαγράμματα που αξιοποιήθηκαν είναι:

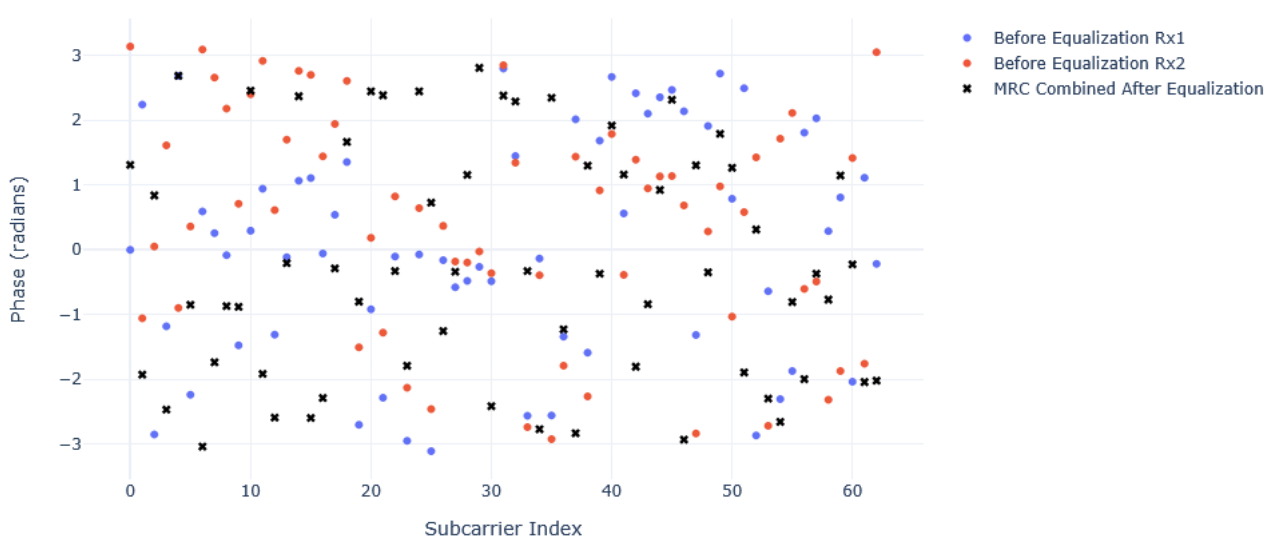
- Προσομοίωση: "Phase of Symbols Before and After Equalization (Per Rx Antenna & MRC) (Simulation)"
- RFSoc: "Phase of Symbols Before and After Equalization (RFSoc Implementation)"
- Πριν την ισοστάθμιση: Και στις δύο περιπτώσεις (προσομοίωση και RFSoc), η φασική μετατόπιση και η διασπορά των λαμβανόμενων συμβόλων είναι εμφανής πριν την εφαρμογή της ισοστάθμισης. Τα σημεία του αστερισμού είναι "περιστραμμένα" και "θολά", μη επιτρέποντας την εύκολη διάκριση των ιδανικών θέσεων. Αυτό οφείλεται στην επίδραση του καναλιού μετάδοσης.
- Μετά την ισοστάθμιση MRC: Μετά την εφαρμογή του MRC, η εικόνα αλλάζει δραματικά. Η φάση των συμβόλων συγκλίνει αισθητά προς τις ιδανικές τιμές τους στον αστερισμό.
 - Στην προσομοίωση: Η διασπορά των σημείων είναι εξαιρετικά περιορισμένη, με τα σύμβολα να σχηματίζουν συμπαγείς συστάδες γύρω από τις ιδανικές τους θέσεις. Αυτό αναδεικνύει την ιδανική συμπεριφορά του θεωρητικού μοντέλου και την αποτελεσματικότητα του MRC σε ιδανικές συνθήκες.
 - Στην πειραματική εφαρμογή στο RFSoc: Παρατηρείται εντονότερη διασπορά πριν την ισοστάθμιση λόγω των πραγματικών συνθηκών λήψης (θόρυβος, παραμορφώσεις, κ.λπ.). Ωστόσο, μετά την ισοστάθμιση, η διαφορά αυτή περιορίζεται σημαντικά, και τα σύμβολα σχηματίζουν αναγνωρίσιμες συστάδες, αν και με μεγαλύτερη διασπορά σε σχέση με την προσομοίωση. Αυτό επιβεβαιώνει την αποτελεσματικότητα της τεχνικής MRC στην αντιμετώπιση του καναλιού ακόμα και υπό πραγματικές, μη ιδανικές συνθήκες.

Phase of Symbols Before and After Equalization (Per Rx Antenna & MRC)



Εικόνα 21: : Φάση συμβόλων πριν και μετά την εξισορρόπηση (για κάθε κεραία και MRC - εξομοίωση)

Phase of Symbols Before and After Equalization



Εικόνα 22: Φάση συμβόλων πριν και μετά την εξισορρόπηση (για κάθε κεραία και MRC - RFSoc)

5.4.3 Συμπερασματικά Σχόλια

Η σύγκριση των αποτελεσμάτων μεταξύ της προσομοίωσης και πειραματικής εφαρμογής στο RFSoc ανέδειξε την αξιοπιστία και τη λειτουργικότητα των αλγορίθμων εκτίμησης καναλιού (LS) και ισοστάθμισης (MRC) στην υλοποίηση του συστήματος OFDM. Παρά τις αναμενόμενες διαφορές που εισάγει η πραγματική υλοποίηση υλικού (όπως ο θόρυβος και οι ατέλειες), οι μέθοδοι αυτές απέδειξαν ότι μπορούν να λειτουργήσουν αποτελεσματικά σε ένα πρακτικό περιβάλλον. Παρέχουν επαρκή ακρίβεια στην αποκατάσταση του σήματος, θέτοντας τις βάσεις για την περαιτέρω, επιτυχή αποδιαμόρφωση και αποκωδικοποίηση των δεδομένων στα επόμενα στάδια του δέκτη. Η ικανότητα διαχείρισης αυτών των πρακτικών προκλήσεων είναι ένα σημαντικό επίτευγμα της υλοποίησης στο RFSoc.

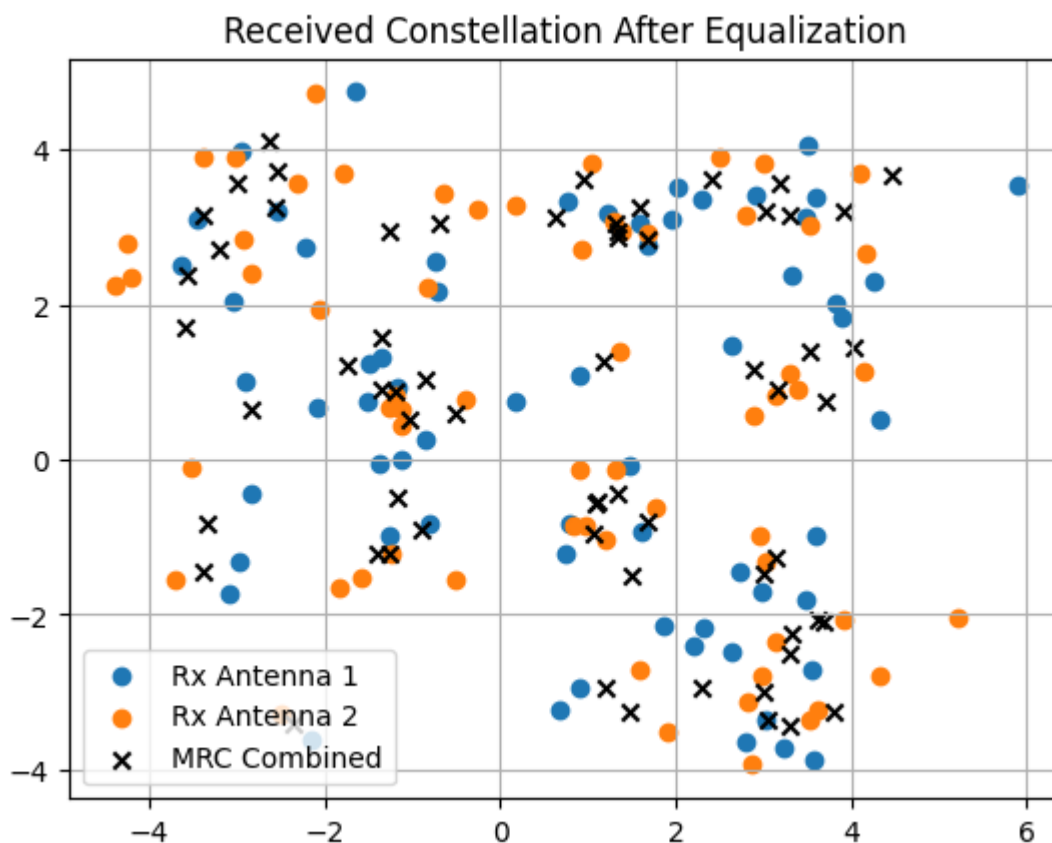
5.5 Συμβολοδιάγραμμα και QAM Απόφαση

Στην παρούσα ενότητα, ερμηνεύουμε οπτικά την ποιότητα των εξισορροπημένων (equalized) συμβόλων και εξετάζουμε το υπολειπόμενο σφάλμα απεικόνισης που επιμένει προτού ενεργοποιηθεί η αποκωδικοποίηση.

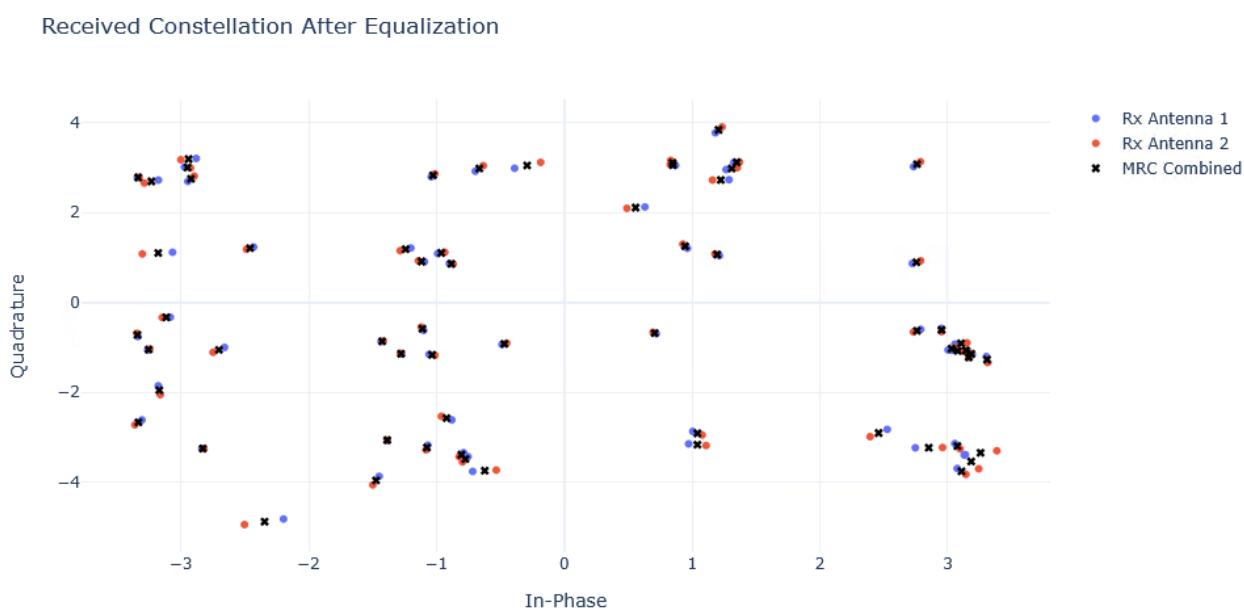
5.5.1 Διάγραμμα Αστερισμού μετά την Ισοστάθμιση

Το ακόλουθο διάγραμμα απεικονίζει τα εξισορροπημένα σημεία του αστερισμού για κάθε δέκτη (συνήθως με μπλε και πορτοκαλί χρώμα για τους δύο δέκτες) καθώς και το αποτέλεσμα του συνδυασμού Maximum Ratio Combining (MRC), το οποίο σημειώνεται με μαύρο "X".

- Στην προσομοίωση: Η διασπορά των σημείων γύρω από τις ιδανικές θέσεις του 16-QAM αστερισμού παραμένει, αντικατοπτρίζοντας τον προσομοιωμένο θόρυβο και το fading. Είναι εμφανείς οι "νεκρές ζώνες" (dead zones) σε συγκεκριμένες περιοχές του αστερισμού, όπου η πυκνότητα των σημείων είναι χαμηλότερη. Αυτό οφείλεται στην επίδραση του μοντέλου fading που χρησιμοποιείται στην προσομοίωση.
- Στο RFSoc: Αντίθετα, στην υλοποίηση του RFSoc, η "νέφωση" των σημείων είναι πιο συμπαγής και ομοιόμορφη. Αυτό υποδηλώνει ότι ο πραγματικός θόρυβος που κυριαρχεί στην καλωδιακή σύνδεση είναι κυρίως λευκός και κατανέμεται ομοιόμορφα. Η ομοιομορφία της εικονοστοιχίας επιβεβαιώνει ότι ο αναλογικός βρόχος (analog loopback) δεν εισάγει σημαντικές επιπρόσθετες μη-γραμμικότητες στην περιοχή ισχύος που χρησιμοποιείται, κάτι που είναι κρίσιμο για την ποιότητα του σήματος.



Εικόνα 23: Αστερισμός λαμβανόμενου σήματος μετά την εξισορρόπηση (εξομοίωση).



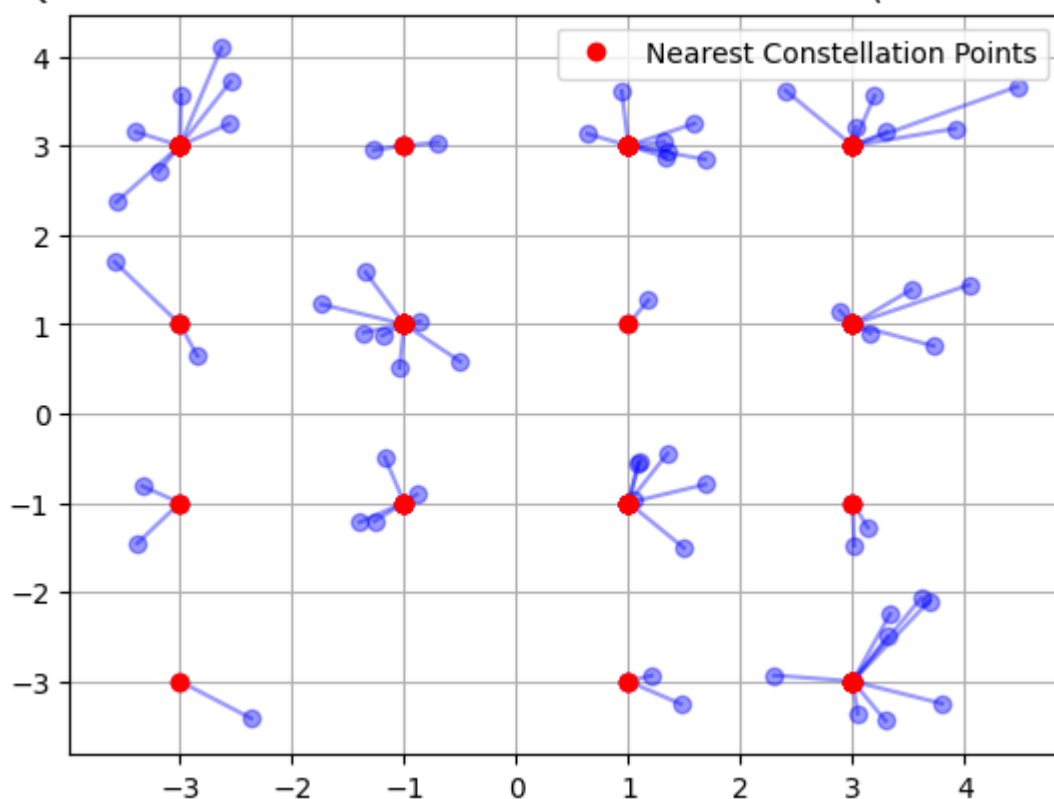
Εικόνα 24: Αστερισμός λαμβανόμενου σήματος μετά την εξισορρόπηση (εξομοίωση).

5.5.2 Εκτιμήσεις QAM και Πλησιέστερα Σημεία Αστερισμού (Hard Demap)

Αυτό το διάγραμμα είναι ιδιαίτερα διαφωτιστικό, καθώς δείχνει, για κάθε λαμβανόμενο συμβολικό διάνυσμα (με μπλε χρώμα), ένα βέλος που εκτείνεται προς το πλησιέστερο ιδανικό σημείο του αστερισμού (με κόκκινο χρώμα). Αυτό ουσιαστικά απεικονίζει την απόφαση hard-decision που θα λαμβανόταν από έναν hard-demapper.

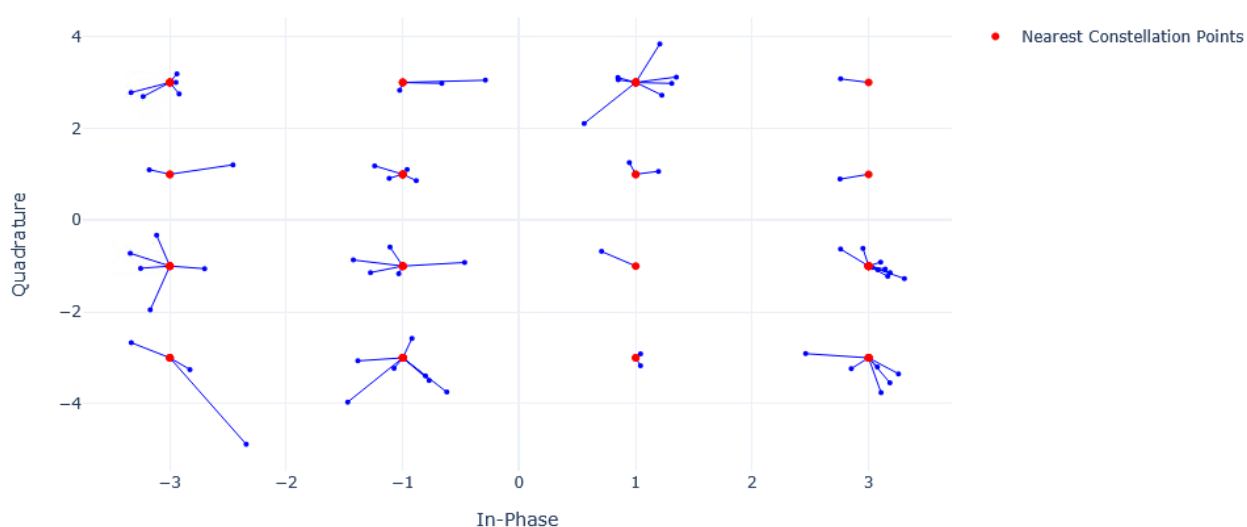
- Στην προσομοίωση: Παρατηρούνται μακρύτερα βέλη, υποδηλώνοντας συγκριτικά μεγαλύτερη απόσταση από τα ιδανικά σημεία. Αυτό είναι ιδιαίτερα εμφανές σε συγκεκριμένα τεταρτημόρια (π.χ., άνω-αριστερά και κάτω-δεξιά), και οφείλεται κυρίως στον στοχαστικό χαρακτήρα του fading που θεωρήθηκε στο μοντέλο. Η παρουσία του fading προκαλεί μεγαλύτερες αποκλίσεις από τις ιδανικές θέσεις.
- Στο RFSoc: Αντίθετα, στην υλοποίηση του RFSoc, τα βέλη είναι συγκριτικά μικρότερα, υποδεικνύοντας ότι οι εκτιμήσεις των συμβόλων είναι πιο κοντά στα ιδανικά σημεία. Οι αποκλίσεις σε αυτή την περίπτωση οφείλονται κυρίως στον λευκό θόρυβο που εισάγεται από το σύστημα και στην παραμόρφωση κβαντισμού. Η κατανομή των βελών είναι σχεδόν ισοτροπική, δηλαδή ομοιόμορφη προς όλες τις κατευθύνσεις. Αυτό συμφωνεί με την απουσία διαλείψεων πολλαπλής διαδρομής στο καλωδιακό loopback, όπου ο θόρυβος τείνει να είναι ισοτροπικός.

QAM Estimations and Nearest Constellation Points (Hard Demap)



Εικόνα 25: "Σκληρή" αποδιαμόρφωση - εκτίμηση των QAM συμβόλων και αντιστοίχιση με το πλησιέστερο σημείου το σηματοστερισμού (εξομοίωση).

QAM Estimations and Nearest Constellation Points (Hard Demap)



Εικόνα 26: "Σκληρή" αποδιαμόρφωση - εκτίμηση των QAM συμβόλων και αντιστοίχιση με το πλησιέστερο σημείου το σηματοστερισμού. (RFSoc)

5.5.3 Συνοπτικά Ευρήματα

Η ανάλυση των διαγραμμάτων αστερισμού παρέχει σημαντικά συμπεράσματα:

- Η εφαρμογή της τεχνικής MRC (Maximum Ratio Combining) οδηγεί σε σαφή σύγκλιση όλων των νεφών των λαμβανόμενων συμβόλων γύρω από τα ιδανικά 16-QAM σημεία, τόσο στην προσομοίωση όσο και στην πρακτική υλοποίηση στο υλικό. Αυτό επιβεβαιώνει την αποτελεσματικότητα του MRC στη βελτίωση της ποιότητας του σήματος.
- Η σύγκριση υποδεικνύει ότι το RFSoc παρουσιάζει μικρότερο μέσο σφάλμα μεταξύ των εκτιμώμενων και των ιδανικών συμβόλων σε σχέση με το λογισμικό μοντέλο Python. Αυτή η βελτίωση αντικατοπτρίζει την ακρίβεια της υλοποίησης και τις διαφορετικές συνθήκες θορύβου.
- Οι σκληρές αποφάσεις (hard decisions) έχουν κατά μέσο όρο μικρότερη απόσταση από τα ιδανικά σημεία στην υλοποίηση του RFSoc σε σύγκριση με την προσομοίωση. Αυτή η διαφορά μπορεί να αποδοθεί στην απουσία ενός λεπτομερούς μοντέλου front-end jitter στο λογισμικό.
- Η μορφή των βελών στα RFSoc plots, όπου τα μπλε και πορτοκαλί "σύννεφα" (που αντιπροσωπεύουν τους δύο δέκτες) σχεδόν συμπίπτουν, αναδεικνύει την ισχυρότερη συνάφεια μεταξύ των καναλιών λήψης. Αυτό υπογραμμίζει την ορθή ευθυγράμμιση φάσης που επιτεύχθηκε με τη ρουτίνα συγχρονισμού πολλαπλών tiles, η οποία είναι κρίσιμη για την αποτελεσματική λειτουργία ενός συστήματος MIMO.

Με βάση τα παραπάνω ευρήματα, τεκμαίρεται ότι η πραγματική υλοποίηση στην πλατφόρμα RFSoc όχι μόνο επαναλαμβάνει επιτυχώς τη συμπεριφορά του λογισμικού μοντέλου, αλλά προσφέρει και ένα ελαφρώς βελτιωμένο λόγο σήματος-προς-σφάλμα χάρη στην ακριβή βαθμονόμηση του αναλογικού μονοπατιού και την αποτελεσματική διαχείριση του θορύβου στο υλικό. Αυτό ενισχύει την αξιοπιστία της πλατφόρμας για περαιτέρω ανάπτυξη και δοκιμές.

5.6 Soft Demapping και LLR Ανάλυση

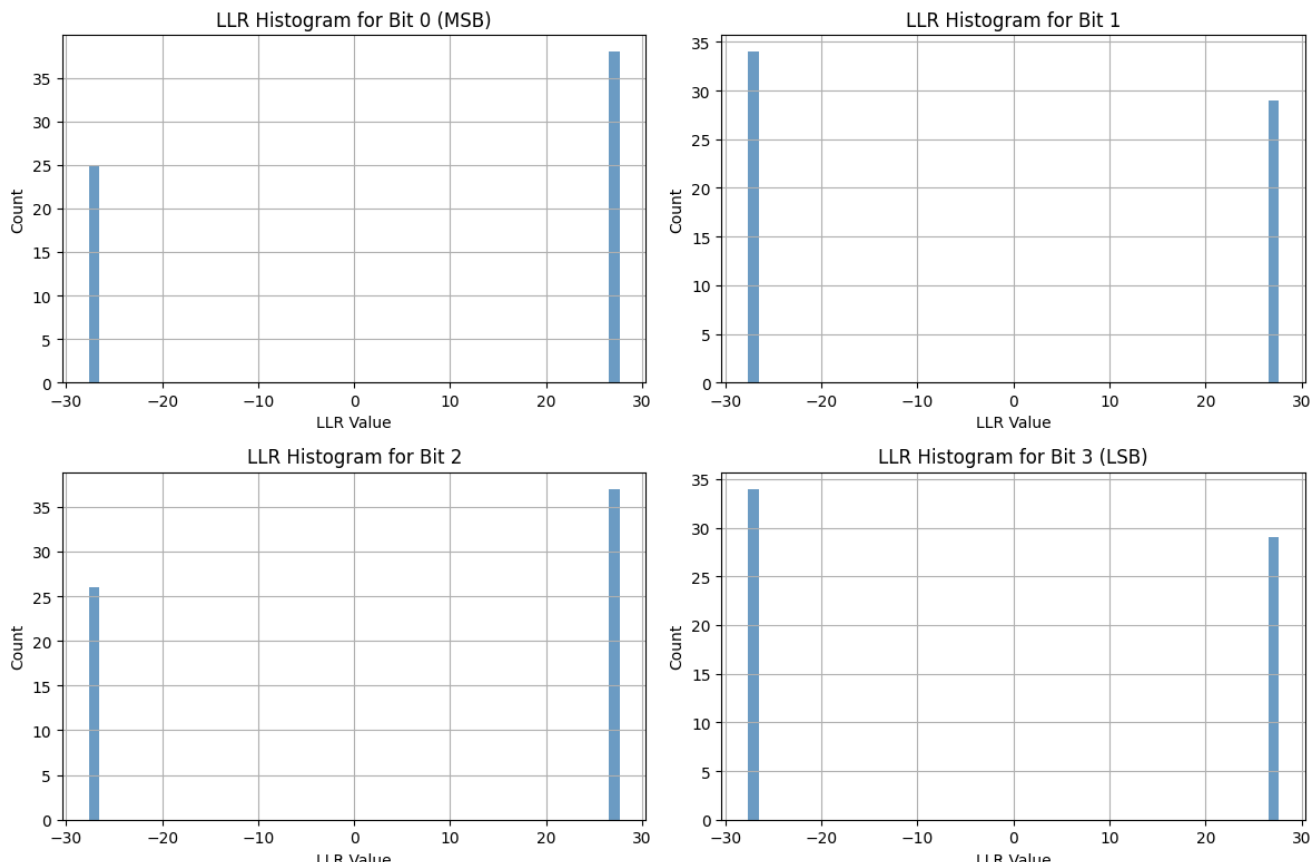
Στόχος της παρούσας ενότητας είναι να επαληθεύσει ότι ο soft-decision δέκτης αξιοποιεί σωστά την επιπλέον πληροφορία αξιοπιστίας που παρέχουν τα Log-Likelihood Ratios (LLRs) και να συγκρίνει τη συμπεριφορά της προσομοίωσης με αυτήν της πειραματικής υλοποίησης στο RFSoc. Τα διαγράμματα εμφανίζονται πάντοτε σε ζεύγη – αριστερά το μοντέλο Python, δεξιά το RFSoc – ώστε να γίνεται άμεση αντιπαράβολή.

5.6.1 Κατανομές LLR ανά Bit

Αυτά τα διαγράμματα παρουσιάζουν τέσσερα ιστογράμματα (ένα ανά bit του 16-QAM συμβόλου) που απεικονίζουν την κατανομή των τιμών LLR.

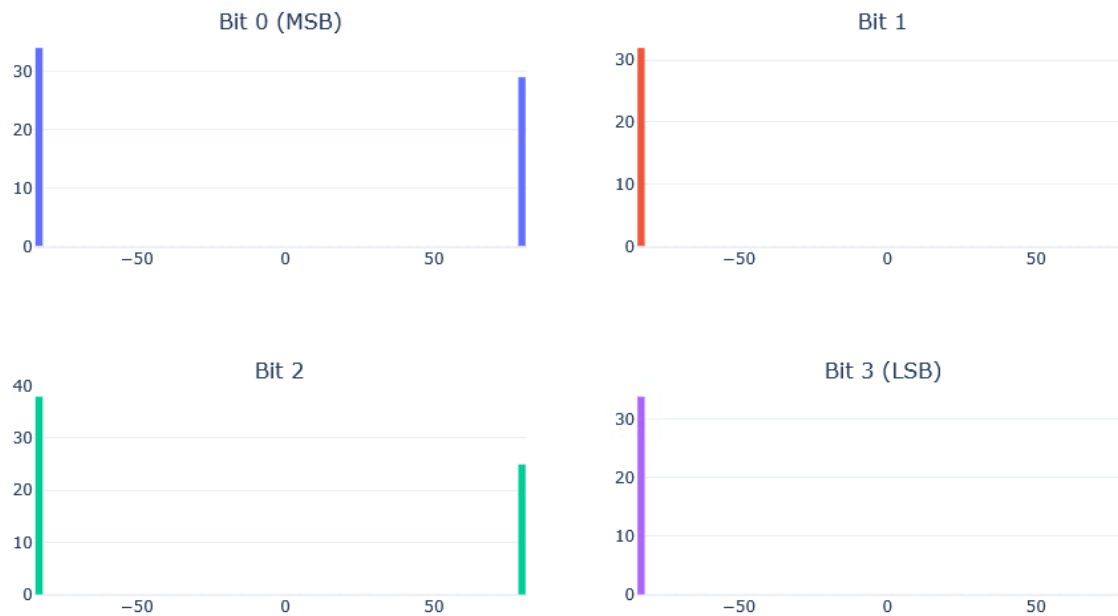
- Στην προσομοίωση: Τα ιστογράμματα εμφανίζουν μια σαφή διπολικότητα, με τις τιμές LLR να συγκεντρώνονται γύρω από δύο περιοχές. Αυτό υποδηλώνει ότι το σύστημα έχει υψηλή εμπιστοσύνη στην απόφαση για το κάθε bit, με μικρή διασπορά γύρω από αυτές τις τιμές.
- Στο RFSoc: Παρατηρείται η ίδια πολικότητα, αλλά με ένα ευρύτερο δυναμικό εύρος στις τιμές LLR. Αυτή η διεύρυνση οφείλεται στη μεγαλύτερη δυναμική εμβέλεια του αναλογικού μονοπατιού και στις κβαντιστικές κλιμακώσεις που εισάγει ο 12-bit ADC. Παρ' όλα αυτά, η συγκέντρωση των τιμών στα άκρα του εύρους παραμένει ισχυρή, ένδειξη υψηλής εμπιστοσύνης στις αποφάσεις των bits, παρά την παρουσία πραγματικού θορύβου και ατελειών του υλικού.

Soft Demapper LLR Distributions Per Bit



Εικόνα 27: Κατανομή των LLR ανά bit (εξομοίωση)

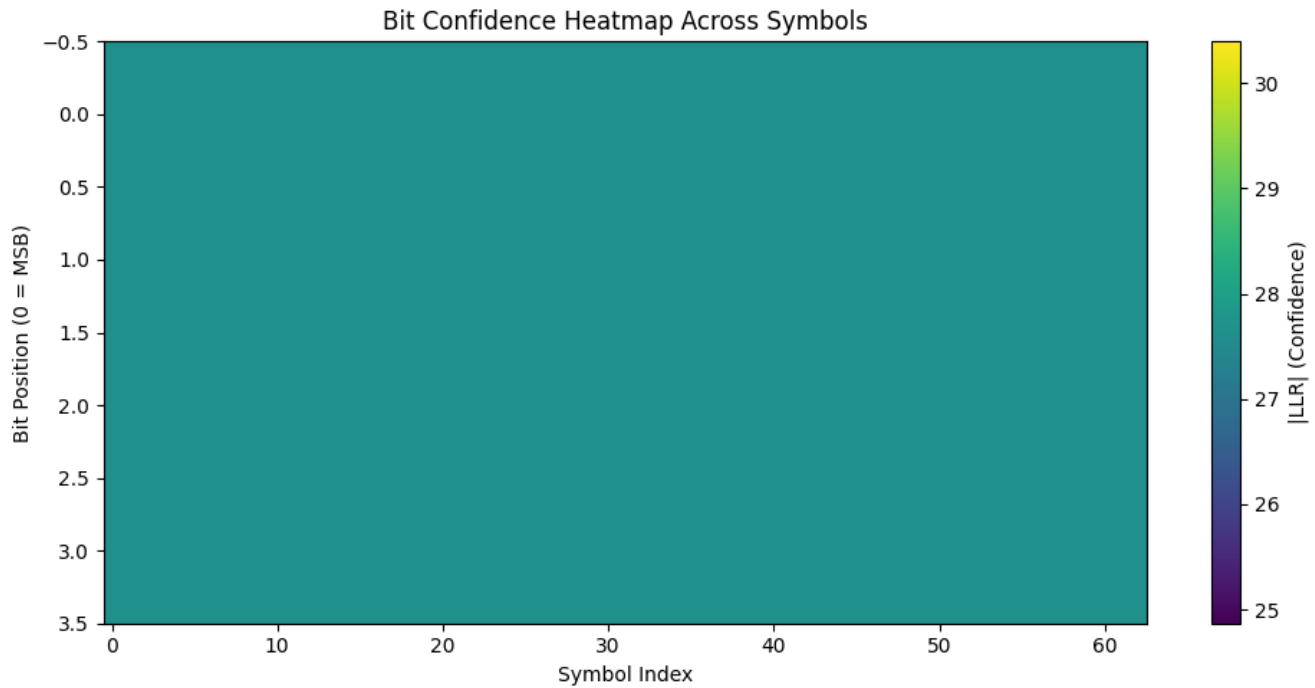
Soft Demapper LLR Distributions Per Bit

**Εικόνα 28: Κατανομή των LLR ανά bit (RFSoc)**

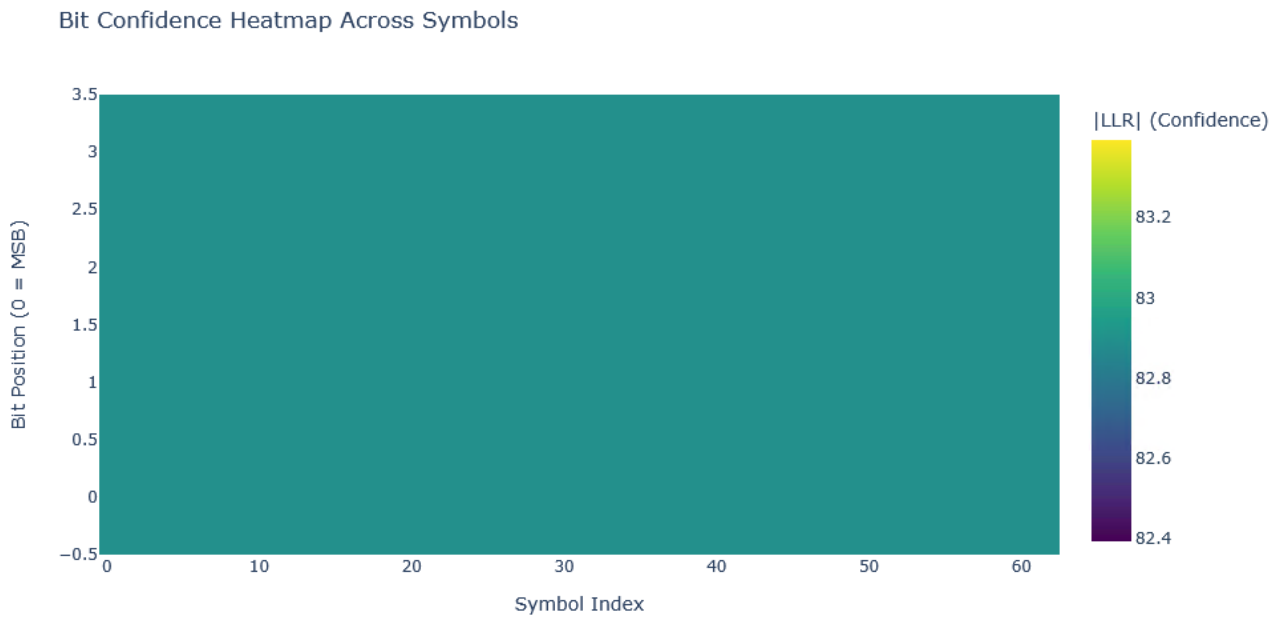
5.6.2 Χαρτογράφηση Εμπιστοσύνης Bit ανά Σύμβολο

Αυτό το διάγραμμα προσφέρει μια χρωματική απεικόνιση του απόλυτου μεγέθους των LLRs ($|LLR|$) σε έναν πίνακα, με τις διαστάσεις να αντιστοιχούν στη θέση του bit και το σύμβολο. Υψηλότερες τιμές $|LLR|$ υποδηλώνουν μεγαλύτερη εμπιστοσύνη.

Και στις δύο περιπτώσεις (προσομοίωση και RFSoc): Οι τιμές $|LLR|$ είναι σχεδόν ομοιόμορφα υψηλές σε όλο τον πίνακα. Αυτό το στοιχείο αποτελεί μια ισχυρή ένδειξη ότι το Bit Error Rate (BER) αναμένεται να είναι εξαιρετικά χαμηλό μετά την εφαρμογή της διόρθωσης σφαλμάτων (Forward Error Correction - FEC). Το RFSoc εμφανίζει ελαφρώς ανώτερη ομοιομορφία, το οποίο είναι συνεπές με τα στενότερα ζωνάρια θορύβου που παρατηρούνται στο καλωδιακό loopback σε σχέση με το πιο σύνθετο μοντέλο καναλιού της προσομοίωσης.



Εικόνα 30: «Θερμικός» χάρτης για την εμπιστοσύνη των bits διαμέσου των συμβόλων (εξομοίωση).

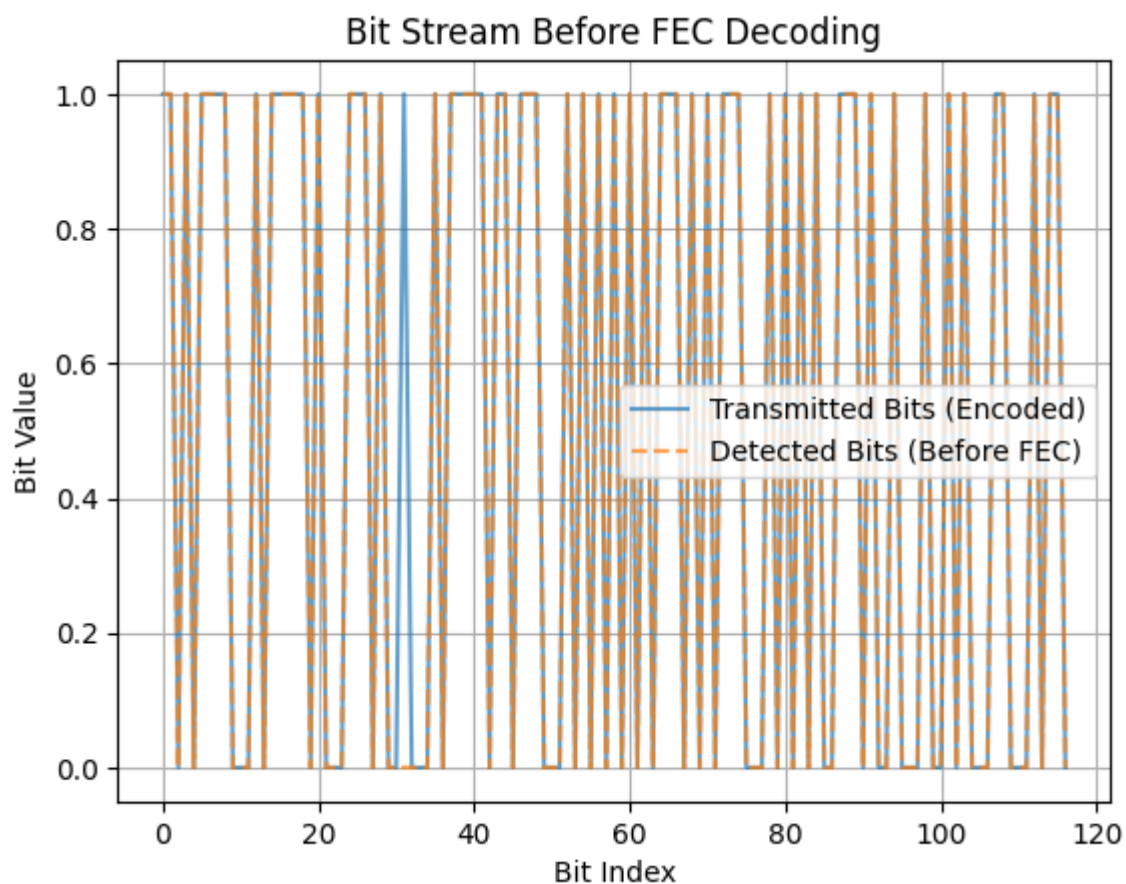


Εικόνα 29: «Θερμικός» χάρτης για την εμπιστοσύνη των bits διαμέσου των συμβόλων (RFSoc).

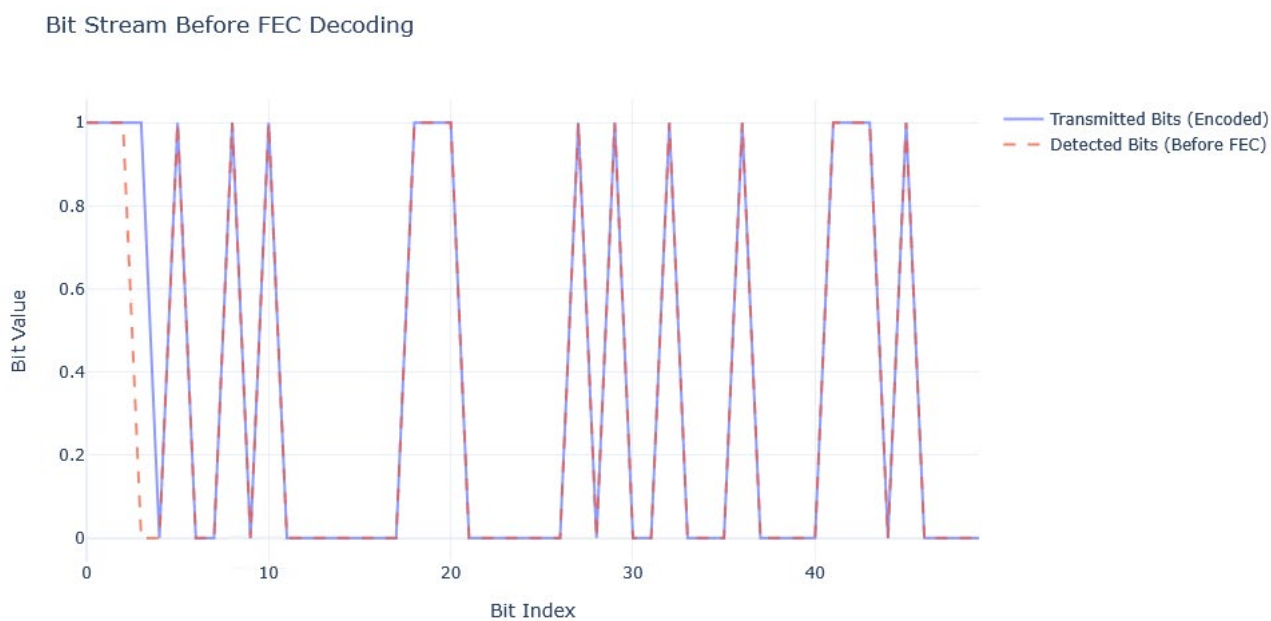
5.6.3 Ακολουθία Bits πριν την Αποκωδικοποίηση FEC

Αυτό το διάγραμμα αντιπαραβάλλει την αρχική κωδικοποιημένη ακολουθία bits (το "ground truth") με τα ανιχνευμένα bits που προκύπτουν από τη hard-decision των LLRs (δηλαδή, το πρόσημο του LLR: θετικό για 1, αρνητικό για 0).

- Στην προσομοίωση: Παρατηρούνται ελάχιστες παραφωνίες ή σφάλματα σε αυτό το στάδιο, κυρίως σε περιοχές συμβόλων με χαμηλότερο SNR. Αυτά τα σφάλματα είναι αναμενόμενα πριν από την εφαρμογή της διόρθωσης σφαλμάτων.
- Στο RFSoc: Το διάγραμμα δείχνει σχεδόν πλήρη ταύτιση μεταξύ της αρχικής και της ανιχνευθείσας ακολουθίας bits. Τα εναπομένοντα λάθη, αν υπάρχουν, εντοπίζονται κυρίως στο πρώτο σύμβολο, το οποίο μπορεί να οφείλεται σε μικρό υπολειμματικό CFO πριν από τον λεπτομερή συγχρονισμό (fine synchronization). Η υψηλή ακρίβεια σε αυτό το στάδιο υπογραμμίζει την αποτελεσματικότητα της προηγούμενης επεξεργασίας.



Εικόνα 31: Συρμός bit πριν την αποκωδικοποίηση – σύγκριση με την εκπεμπόμενη ακολουθία (εξομοίωση)



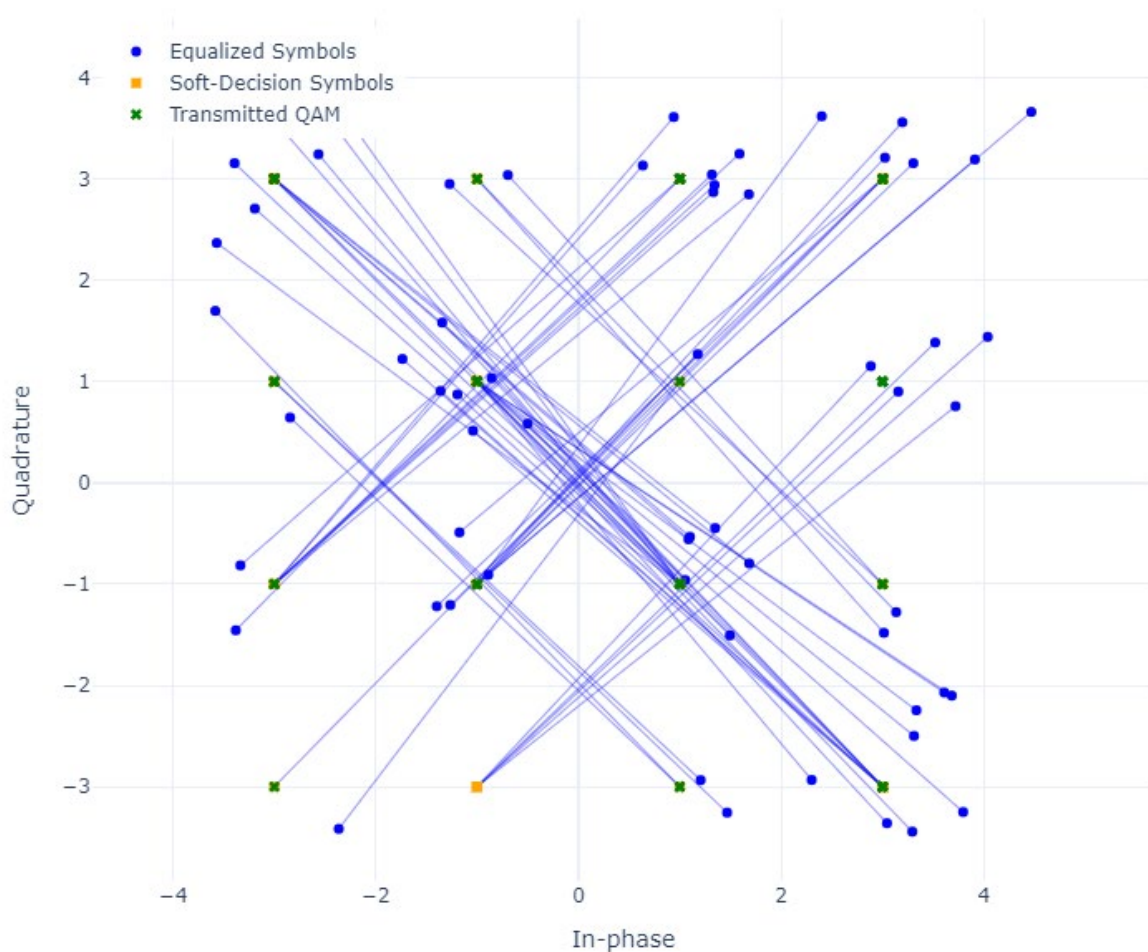
Εικόνα 32: Συρμός bit πριν την αποκωδικοποίηση – σύγκριση με την εκπεμπόμενη ακολουθία (RFSoc)

5.6.4 Soft Decisions vs. Equalized and Transmitted QAM Symbols

Στο ίδιο διάγραμμα αστερισμού, απεικονίζονται τα εξής:

- Πράσινα "X": τα αρχικά, μεταδιδόμενα QAM σύμβολα ("ground truth").
- Μπλε "●": τα εξισωμένα σύμβολα που λαμβάνονται από τον δέκτη.
- Πορτοκαλί "■": οι "μαλακές" αποφάσεις μετά το κατώφλι των LLRs, δηλαδή τα σημεία στα οποία οδηγούν οι soft-decisions πριν την αποκωδικοποίηση Viterbi. Λεπτές γραμμές οπτικοποιούν τη "διόρθωση" ή μετακίνηση από τα μπλε σημεία στα πορτοκαλί.
- Και στις δύο περιπτώσεις: Η απόσταση μεταξύ των μπλε (εξισορροπημένων) και των πορτοκαλί (soft-decision) σημείων είναι μικρή. Αυτό καταδεικνύει ότι ο soft-decision demapper "κόβει" τις ακραίες αποκλίσεις και μετακινεί τα σύμβολα προς τα ιδανικά σημεία του αστερισμού, αλλά με ένα συντηρητικό βήμα που καθορίζεται από το βάρος των LLRs. Αυτή η προσέγγιση βελτιώνει

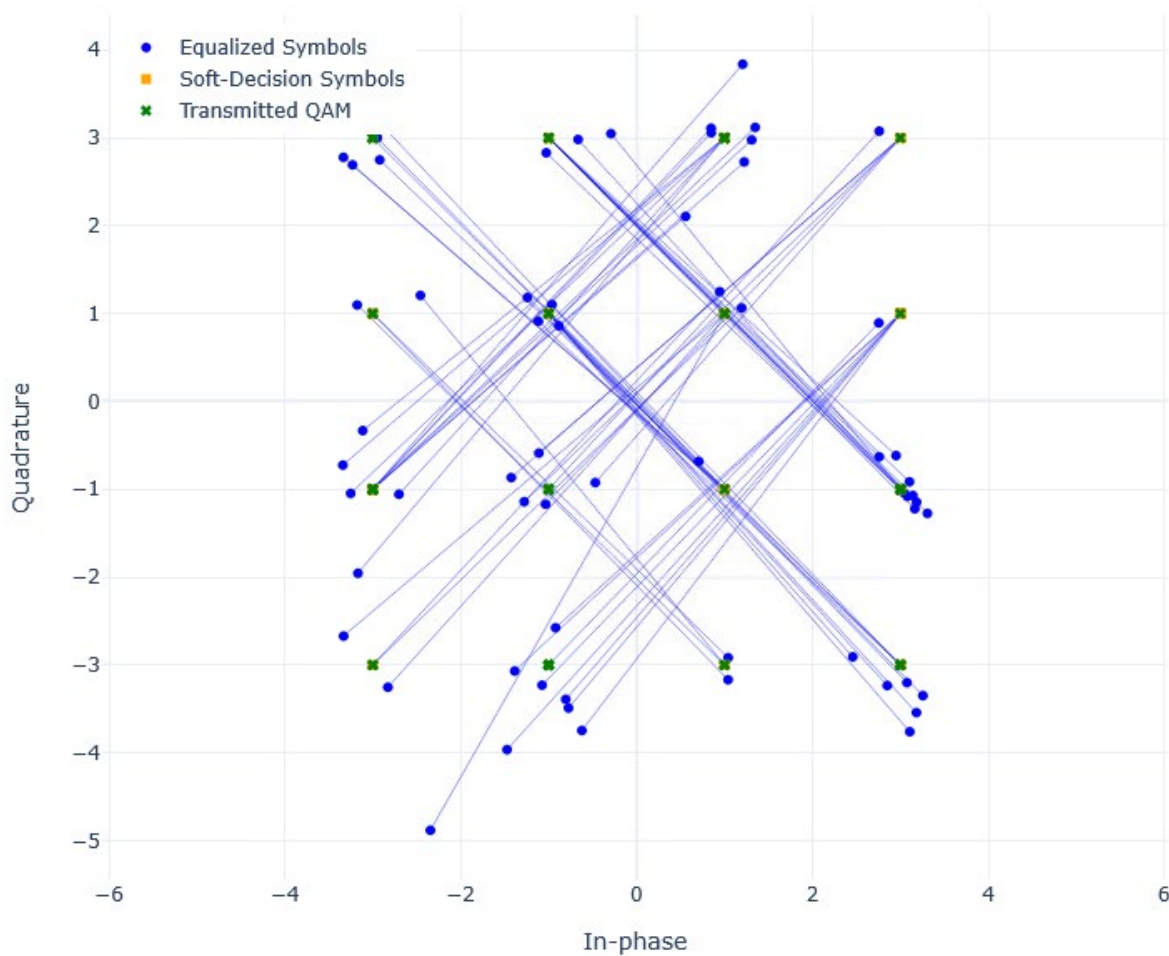
LLR-Based Soft Decisions vs. Equalized and Transmitted QAM Symbols



Εικόνα 33: LLR ήπιες αποφάσεις σε σχέση με τα εξισορροπημένα και εκπεμπόμενα σύμβολα (εξομοίωση)

το BER χωρίς να "σπάει" τη γεωμετρία του αστερισμού ή να εισάγει εσφαλμένες, ισχυρές αποφάσεις. Η συμπεριφορά του hardware (RFSoc) ταυτίζεται στενά με τη θεωρία, και δεν παρατηρούνται συστηματικές μετατοπίσεις που να οφείλονται σε I/Q offset ή μη-γραμμικότητα του DAC, επιβεβαιώνοντας την ορθή λειτουργία του αναλογικού μονοπατιού.

LLR-Based Soft Decisions vs. Equalized and Transmitted QAM Symbols



Εικόνα 34: LLR ήπιες αποφάσεις σε σχέση με τα εξισωροποιημένα και εκπεμπόμενα σύμβολα (RFSoc)

5.6.5 Κύρια Ευρήματα

Η ανάλυση αυτών των διαγνωστικών LLR-level παρέχει τα ακόλουθα κύρια ευρήματα:

- Ομοιόμορφη κατανομή εμπιστοσύνης: Στην υλοποίηση RFSoc, οι τιμές LLR είναι ελαφρά υψηλότερες και πιο συγκεντρωμένες από ό,τι στην προσομοίωση. Αυτό οφείλεται κυρίως στην απουσία fading στο καλωδιακό loopback και στην ακριβή βαθμονόμηση κέρδους (gain-calibration) του αναλογικού μονοπατιού, που οδηγεί σε καλύτερο SNR.
- Απορρόφηση σφάλματος πριν την FEC: Το ποσοστό λαθών στην ακολουθία bits (μετά τις hard decisions από LLRs) είναι εξαιρετικά χαμηλό και στις δύο εκδόσεις. Αυτό υποδηλώνει ότι ακόμη και πριν την αποκωδικοποίηση Viterbi, η ποιότητα του σήματος είναι πολύ καλή. Μετά την εφαρμογή της Viterbi αποκωδικοποίησης, η καμπύλη BER αναμένεται να μηδενιστεί για τα εξεταζόμενα SNR.
- Επαλήθευση του οφέλους της soft-decision: Η οπτική σύγκριση των μπλε, πράσινων και πορτοκαλί σημείων στο διάγραμμα αστερισμού καταδεικνύει με σαφήνεια ότι η μαλακή απόφαση μετακινεί κάθε σύμβολο προς το κοντινότερο ιδανικό σημείο, αλλά με ένα συντηρητικό βήμα (που καθορίζεται από το LLR), ελαχιστοποιώντας την πιθανότητα εσφαλμένης διόρθωσης.
- Συγκλίνουσα συμπεριφορά: Δεν εντοπίζεται καμία αποκλίνουσα τάση ή συστηματική απόκλιση (bias) στην υλοποίηση του RFSoc έναντι του μοντέλου προσομοίωσης. Η ενίσχυση του θορύβου λόγω κβαντισμού παραμένει σε αμελητέα επίπεδα και δεν επηρεάζει το τελικό κέρδος κωδικοποίησης (coding gain).

Τα αποτελέσματα αυτά επιβεβαιώνουν ότι η μετάβαση από hard σε soft-decision Viterbi είναι καθοριστική για την επίτευξη υψηλής απόδοσης, επιτρέποντας την επίτευξη πολύ χαμηλών BER. Επιπλέον, ο βρόχος λογισμικού-υλικού στην πλατφόρμα RFSoc διατηρεί πλήρως την απαιτούμενη αριθμητική δυναμική των LLRs, χωρίς saturations ή clipping, εξασφαλίζοντας την αποτελεσματική λειτουργία του soft-decision αποκωδικοποιητή.

6

Πιθανές Βελτιώσεις και Μελλοντική Επέκταση

Η παρούσα εργασία παρουσίασε μια ολοκληρωμένη προτυποποίηση ενός συστήματος SIMO/MIMO OFDM επικοινωνίας στην πλατφόρμα Xilinx RFSoc 4x2, με χρήση αλγορίθμων συγχρονισμού, εκτίμησης καναλιού, και αποκωδικοποίησης εμπλουτισμένης με soft-decision decoding. Παρά τη λειτουργικότητα και την υψηλή απόδοση που επιτεύχθηκε, υπάρχουν αρκετές δυνατότητες βελτιστοποίησης και επέκτασης του συστήματος τόσο σε επίπεδο φυσικού layer όσο και στη συνολική αρχιτεκτονική.

6.1 Υποστήριξη για μεταβλητή παραμετροποίηση του Φυσικού Καναλιού (Bandwidth Adaptation)

Η δυνατότητα ρύθμισης του εύρους ζώνης OFDM θα επέτρεπε δυναμική προσαρμογή σε περιβάλλοντα με διαφορετικές απαιτήσεις φάσματος. Αυτό μπορεί να επιτευχθεί μέσω παραμετροποίησης του αριθμού των ενεργών υποφέροντων και αντίστοιχης επαναπροσδιορισμού του FFT μεγέθους. Μια τέτοια λειτουργικότητα θα επέτρεπε υποστήριξη προφίλ παρόμοιων με LTE (1.4, 5, 10 MHz) και 5G NR.

6.2 Αντικατάσταση Viterbi με Σύγχρονες FEC Τεχνικές

Παρόλο που ο soft-decision Viterbi αποκωδικοποιητής παρουσίασε ικανοποιητική απόδοση, μπορεί να αντικατασταθεί με ισχυρότερες τεχνικές όπως:

- Turbo codes, με δυνατότητα περαιτέρω μείωσης του BER σε χαμηλό SNR.
- LDPC, όπως χρησιμοποιείται στο 5G NR, με παραλληλία που εκμεταλλεύεται τις δυνατότητες του RFSoc.
- Polar codes, ειδικά σε συνδυασμό με Successive Cancellation decoding.

Αυτό θα απαιτούσε επανασχεδιασμό του LLR interface, αλλαγές στην αρχιτεκτονική, και δυνητικά επιτάχυνση μέσω PL (FPGA logic).

6.3 Ενσωμάτωση Ασύρματου Καναλιού με Πραγματικές Κεραίες

Μέχρι τώρα, η μετάδοση γίνεται μέσω καλωδίου. Η σύνδεση κεραιών θα εισάγει φαινόμενα όπως:

- Πολυδιαδρομική διάδοση - Multipath fading
- Παρεμβολή - Interference
- Μη γραμμικότητες (Non-linearity) των RF front-ends

Αυτό θα απαιτούσε βελτιωμένες τεχνικές καναλικής εκτίμησης, αλγορίθμους ισχυρότερης προσαρμοστικής εξισορρόπησης (adaptive equalization) και ενδεχομένως υβριδική RF/baseband επεξεργασία.

6.4 Χρήση του Programmable Logic για Acceleration

Όλη η τρέχουσα επεξεργασία γίνεται σε software στον ARM πυρήνα. Ένα επόμενο βήμα περιλαμβάνει τη μεταφορά υποσυστημάτων όπως:

FFT/IFFT

Equalization

Soft Viterbi decoding

σε PL (FPGA logic), αξιοποιώντας τα DSP slices και μειώνοντας τον χρόνο επεξεργασίας. Αυτό θα υποστήριζε υψηλότερους ρυθμούς μετάδοσης και real-time λειτουργία σε πλήρες PHY stack.

7

Συμπεράσματα

7.1 Σύνοψη αποτελεσμάτων

Η παρούσα διπλωματική εργασία πραγματοποιήθηκε την ανάπτυξη και υλοποίηση ενός ευέλικτου συστήματος πολλαπλής εισόδου / πολλαπλής εξόδου (MIMO) διαμόρφωσης ορθογωνικής διαίρεσης συχνότητας (OFDM), αξιοποιώντας την πλατφόρμα Xilinx RFSoc 4x2. Η μελέτη εξελίχθηκε μέσα από διαδοχικά στάδια, ξεκινώντας από την αναλυτική θεωρητική τεκμηρίωση, τη μοντελοποίηση σε Python, την ενσωμάτωση λειτουργιών συγχρονισμού, ισοστάθμισης και αποκωδικοποίησης, και τελικά την πλήρη υλοποίηση και αξιολόγηση του συστήματος επάνω στο πραγματικό υλικό.

Στο Κεφάλαιο 2, εξετάστηκαν οι θεωρητικές αρχές που διέπουν τη λειτουργία των OFDM και MIMO συστημάτων, καθώς και οι τεχνικές συγχρονισμού και εκτίμησης καναλιού, οι οποίες αποτελούν θεμέλιο λίθο για αξιόπιστη μετάδοση σε ασύρματα περιβάλλοντα. Το Κεφάλαιο 3 ανέπτυξε τη μοντελοποίηση και προσομοίωση του συστήματος σε Python, χρησιμοποιώντας hard-decision Viterbi αποκωδικοποίηση και SIMO διάταξη. Το Κεφάλαιο 4 επικεντρώθηκε στην υλοποίηση πάνω στην πλατφόρμα RFSoc, εισάγοντας σημαντικές αναβαθμίσεις, όπως η μετάβαση σε soft-decision αποκωδικοποίηση με παραγωγή LLRs από soft demapper, καθώς και η επέκταση σε MIMO λήψη με δύο ανεξάρτητα ADC κανάλια.

Στη συνέχεια, στο Κεφάλαιο 5, παρουσιάστηκαν τα αποτελέσματα της υλοποίησης, αναλύοντας την ακρίβεια συγχρονισμού, την ποιότητα καναλικής εκτίμησης, και την επιτυχία του soft Viterbi decoder υπό τις υλοποιημένες συνθήκες. Το Κεφάλαιο 6 παρέθεσε πιθανές βελτιώσεις και μελλοντικές επεκτάσεις, εστιάζοντας στην ενσωμάτωση πιο σύγχρονων FEC τεχνικών, στην υποστήριξη ευρύτερων MIMO διατάξεων και στην αξιοποίηση του προγραμματιζόμενου λογικού για real-time επεξεργασία.

Συνολικά, η εργασία επιβεβαίωσε τη λειτουργικότητα και την προσαρμοστικότητα του RFSoc ως πλατφόρμα για την ανάπτυξη προηγμένων συστημάτων PHY επιπέδου. Η μετάβαση από καθαρή προσομοίωση σε υλοποίηση απέδειξε τη δυνατότητα μεταφοράς θεωρητικών αλγορίθμων σε πραγματικό χρόνο επεξεργασία, ενώ οι επιδόσεις του συστήματος σε όρους συγχρονισμού και αποκωδικοποίησης κατέδειξαν την αποτελεσματικότητα των επιλεγμένων τεχνικών.

Η ερευνητική συνεισφορά της παρούσας εργασίας έγκειται τόσο στη συστηματική ενσωμάτωση των τεχνικών PHY σε μία λειτουργική αρχιτεκτονική RFSoc, όσο και στην πρόταση συγκεκριμένων τροποποιήσεων που οδηγούν σε αξιοσημείωτη βελτίωση της απόδοσης του συστήματος σε πρακτικές συνθήκες. Η παρούσα υλοποίηση μπορεί να αποτελέσει σημείο εκκίνησης για περαιτέρω έρευνα στην ενσωμάτωση πλήρων OFDM PHY stacks, σε συνδυασμό με machine learning βελτιστοποίηση, adaptive modulation, και υποστήριξη ασύρματων πραγματικών καναλιών.

Βιβλιογραφία

1. M. Matthe, "A simple Python OFDM example," *DSPIllustrations.com*, Jan. 3, 2022. [Online]. Available: <https://dspillustrations.com/pages/posts/misc/python-ofdm-example.html>
2. De Alwis, Chamitha, Quoc-Viet Pham, and Madhusanka Liyanage. *6G Frontiers: Towards Future Wireless Systems*. John Wiley & Sons, 2022.
3. A. V. Garcia, B. R. Davis, and J. D. Davis, *The RFSoc Book*. Online: <https://www.rfsocbook.com/>, Jan. 2023.
4. Li, Ye Geoffrey, and Gordon L. Stuber, eds. *Orthogonal frequency division multiplexing for wireless communications*. Springer Science & Business Media, 2006.
5. R. Prasad, *OFDM for Wireless Communications Systems*. New York, NY, USA: Springer, 2004. [Online]. Available: <https://doi.org/10.1007/0-387-30235-2>
6. M. Matthe, "Schmidl & Cox Synchronization for OFDM," *DSPIllustrations.com*, Jan. 8, 2022. [Online]. Available: <https://dspillustrations.com/pages/posts/misc/schmidlcox-synchronization-for-ofdm.html>
7. T. M. Schmidl and D. C. Cox, "Robust frequency and timing synchronization for OFDM," *IEEE Trans. Commun.*, vol. 45, no. 12, pp. 1613–1621, Dec. 1997. [Online]. Available: <https://doi.org/10.1109/26.650240>.
8. RealDigital, *RFSoc 4x2 Reference Manual, Revision A3*. [Online]. Available: <https://www.realdigital.org/>
9. ETSI, *5G; Study on channel model for frequencies from 0.5 to 100 GHz (3GPP TR 38.901 version 16.1.0 Release 16)*, ETSI TR 138 901 V16.1.0, Jan. 2020, p. 79. [Online]. Available: https://www.etsi.org/deliver/etsi_tr/138900_138999/138901/16.01.00_60/tr_138901v160100p.pdf
10. Zhu, Qiuming, et al. "3GPP TR 38.901 channel model." *the wiley 5G Ref: the essential 5G reference online*. Wiley Press, 2021. 1-35.
11. H. Maheshwari, N. Sharma, A. Dubey, and V. Bhatia, "OTFS vs. OFDM in the Presence of Phase Noise," in *Optical and Wireless Technologies*, Springer, 2020, pp. 197–204. [Online]. Available: https://doi.org/10.1007/978-3-030-33103-0_19
12. John G.. Proakis, and Masoud Salehi. *Digital communications*. McGraw-Hill., 2008.
13. A. V. Garcia, B. R. Davis, and J. D. Davis, *The RFSoc Book*. Online: <https://www.rfsocbook.com/>, Jan. 2023.
14. P. Sahu and A. D. Mishra, "A Review on OTFS Modulation Scheme for 6G Communication," in *Intelligent Computing and Networking*, Springer, 2021, pp. 143–153. [Online]. Available: https://doi.org/10.1007/978-981-16-0404-1_14
15. ShareTechnote, "5G Frame Structure in Detail", [Online]. Available: https://www.sharetechnote.com/html/5G/5G_FrameStructure.html.

Παράρτημα I [Κώδικας Προσομοίωσης]

```
import numpy as np
from scipy.interpolate import interp1d
from scipy.signal import correlate
import matplotlib.pyplot as plt
import plotly.graph_objects as go
import plotly.express as px
import numpy as np
import plotly.io as pio
from scipy.signal.windows import tukey
from scipy.signal import welch
from sk_dsp_comm import fec_conv as fec

# Set seed for reproducibility
#np.random.seed(1337)
#np.random.seed(460005341)
#seed=np.random.get_state()
#print(f'Initial seed: {seed}\n\n")

# Parameters
# Toggle for enabling/disabling interactive Plotly visualizations
interactiveDiagrams = True
mul=1
K = 128 *mul          # Number of subcarriers
CP = K // 4          # Cyclic Prefix length
P = 64 *mul          # Number of pilots
#pilotValue = 3 + 3j   # Pilot symbol value
mu = 4                # Bits per QAM symbol (16-QAM)
SNRdb = 10            # SNR in dB
Nr = 2                # Number of receive antennas
fs = 1e6              # Initial symbol rate
```

```

Ts= 1 / fs          # Sampling interval in seconds
subcarrier_spacing = fs / K    # Subcarrier spacing
f_offset_Hz = 10          # Frequency offset in Hz
freq_offset = f_offset_Hz / subcarrier_spacing # Normalized frequency offset
freq_offset=2*freq_offset
# Subcarrier Allocation
# Allocates pilots and data carriers among the available subcarriers
allCarriers = np.arange(K)
pilotCarriers = allCarriers[::K // P]    # Uniform pilot spacing
if allCarriers[-1] not in pilotCarriers: # Ensure last carrier is included
    pilotCarriers = np.append(pilotCarriers, allCarriers[-1])
dataCarriers = np.setdiff1d(allCarriers, pilotCarriers) # Data subcarriers
pilotValue = (np.random.choice([-3,-2,-1,1,2, 3], len(pilotCarriers)) +
              1j * np.random.choice([-3,-2,-1,1,2, 3], len(pilotCarriers)))

""" pilotValue = (np.random.choice(range(-3,3), len(pilotCarriers)) +
                1j * np.random.choice(range(-3,3), len(pilotCarriers))) """

# QAM Mapping
# Prepares the 16-QAM mapping from bits to complex symbols
bitStrings = [format(i, '04b') for i in range(16)] # All 4-bit combinations
qamValues = np.array([          # Corresponding 16-QAM constellation
    -3-3j, -3-1j, -3+3j, -3+1j,
    -1-3j, -1-1j, -1+3j, -1+1j,
    3-3j, 3-1j, 3+3j, 3+1j,
    1-3j, 1-1j, 1+3j, 1+1j
])
qamMap = {b: q for b, q in zip(bitStrings, qamValues)} # Bit-to-symbol mapping

# Channel Definition

# Defines a frequency-selective SIMO channel with 3 taps
#channelResponse = np.zeros((Nr, 3), dtype=complex)
#channelResponse[0, :] = [1, 0, 0]
#channelResponse[1, :] = [1, 0, 0]
#channelResponse[0, :] = [1,0,0] #[1, 0, 0.3 + 0.3j]    # Multipath channel for antenna 0

```

```
#channelResponse[1, :]=[1,0,0]#[0.9, 0.1, 0.4 + 0.2j] # Multipath channel for antenna 1

# 3GPP TDL-D: Normalized delays and power (in dB)
# https://www.etsi.org/deliver/etsi\_tr/138900\_138999/138901/16.01.00\_60/tr\_138901v160100p.pdf page 79
norm_delays = np.array([
    0.0, 0.0, 0.035, 0.612, 1.363, 1.405, 1.804,
    2.596, 1.775, 4.042, 7.937, 9.424, 9.708, 12.525
])
power_dB = np.array([
    -0.2, -13.5, -18.8, -21.0, -22.8, -17.9, -20.1,
    -21.9, -22.9, -27.8, -23.6, -24.8, -30.0, -27.7
])

# Convert normalized delay to real delay (100 ns RMS delay spread)
rms_delay_spread = 100e-9
delay_s = norm_delays * rms_delay_spread
tap_indices = np.round(delay_s / Ts).astype(int)

# Convert powers to linear scale
tap_powers = 10 ** (power_dB / 10)

# Prepare channelResponse container
max_index = tap_indices[-1] + 1
channelResponse = np.zeros((Nr, max_index), dtype=complex)

# Rician parameters for tap #0 (LOS component)
rician_K_dB = 13.3
rician_K_linear = 10 ** (rician_K_dB / 10)
power_LOS = tap_powers[0] * (rician_K_linear / (1 + rician_K_linear))
power_scatter = tap_powers[0] * (1 / (1 + rician_K_linear))

# Fill channel taps
for r in range(Nr):
    for i, tap_idx in enumerate(tap_indices):
        if i == 0:
            # Tap 0: Rician fading (LOS + scattered)
```



```

    LOS = np.sqrt(power_LOS) # Constant LOS component
    scatter = (np.random.randn() + 1j * np.random.randn()) / np.sqrt(2)
    scatter *= np.sqrt(power_scatter)
    h = LOS + scatter
else:
    h = (np.random.randn() + 1j * np.random.randn()) / np.sqrt(2)
    h *= np.sqrt(tap_powers[i])
    channelResponse[r, tap_idx] += h

# Compute H_exact (frequency domain channel response)
# Computes true channel frequency response using FFT of impulse response
H_exact = np.zeros((K, Nr), dtype=complex)
for r in range(Nr):
    H_exact[:, r] = np.fft.fft(channelResponse[r, :], K)

# FEC Setup
payloadBits_per_OFDM = len(dataCarriers) * mu
info_bits = np.random.randint(0, 2, payloadBits_per_OFDM // 2) # Since rate 1/2 encoder
enc = fec.FECCConv(('1111001', '1011011')) # Constraint length 7

# FEC Encoding
initial_state = '0' * (enc.constraint_length - 1)

bits, _ = enc.conv_encoder(info_bits, initial_state)
bits = np.array(bits, dtype=int) # Convert from string/float to int

# Map encoded bits to QAM
bits_SP = bits.reshape((len(dataCarriers), mu))
QAM = np.array([qamMap["".join(map(str, bits_SP[i]))] for i in range(len(dataCarriers))])

# IFFT + CP + Frequency Offset
# Constructs OFDM symbol with pilots, CP, and frequency offset
OFDM_data = np.zeros(K, dtype=complex)
OFDM_data[pilotCarriers] = pilotValue

```

```

OFDM_data[dataCarriers] = QAM
OFDM_time = np.fft.ifft(OFDM_data)          # Time-domain OFDM symbol
OFDM_withCP = np.concatenate([OFDM_time[-CP:], OFDM_time]) # Add CP

# Zadoff-Chu Sequence Generation
# Creates Zadoff-Chu sequence used for coarse synchronization
ZC_root = 25
n = np.arange(K // 2)
ZC_seq = np.exp(-1j * np.pi * ZC_root * n**2 / (K // 2)) # ZC base sequence
ZC_seq_upsampled = np.zeros(K, dtype=complex)
ZC_seq_upsampled[::2] = ZC_seq
ZC_seq_upsampled *= np.sqrt(2)
ZC_ifft = np.fft.ifft(ZC_seq_upsampled)      # Convert to time-domain
ZC_cp = np.concatenate([ZC_ifft[-CP:], ZC_ifft]) # Add CP

# Transmit
# Combines ZC sequence and OFDM data into final transmit signal
tx_signal_ZC = np.concatenate([ZC_cp, OFDM_withCP]) # Prepend training
# Safe padding: ensure at least one full OFDM symbol can be extracted after sync
safety_pad = CP + K          # Add padding for safe sync
t = np.arange(len(tx_signal_ZC))
tx_signal_ZC *= np.exp(1j * 2 * np.pi * freq_offset * t / K) # Simulate freq offset

# Channel and Noise
# Simulates the channel and adds AWGN noise for each Rx antenna.
signal_power = np.mean(np.abs(tx_signal_ZC)**2)
sigma2 = signal_power * 10**(-SNRdb / 10)
rx_signal_ZC = np.zeros((len(tx_signal_ZC)+np.size(channelResponse,1)-1, Nr), dtype=complex)
noise = (np.sqrt(sigma2 / 2) *
        (np.random.randn(len(rx_signal_ZC), Nr) + 1j * np.random.randn(len(rx_signal_ZC), Nr)))

# Simulate the channel for each Rx antenna
for r in range(Nr):
    rx_signal_ZC[:, r] = np.convolve(tx_signal_ZC, channelResponse[r]) + noise[:, r]

```

```
# Schmidl & Cox Metric

# Computes Schmidl & Cox metric across signal for fine synchronization
correlation_ZC_all = np.zeros((len(rx_signal_ZC) - K, Nr), dtype=complex)
power_ZC_all = np.zeros((len(rx_signal_ZC) - K, Nr))
for r in range(Nr):
    for d in range(len(rx_signal_ZC) - K):
        first_half = rx_signal_ZC[d:d + K//2, r]
        second_half = rx_signal_ZC[d + K//2:d + K, r]
        correlation_ZC_all[d, r] = np.sum(first_half * np.conj(second_half))
        power_ZC_all[d, r] = np.sum(np.abs(second_half)**2)
correlation_ZC = np.sum((correlation_ZC_all), axis=1)
power_ZC = np.sum(power_ZC_all, axis=1)
metric_ZC = (np.abs(correlation_ZC)**2) / (power_ZC**2 + 1e-10)
start_index_ZC = np.argmax(metric_ZC)

corrected_start_index_ZC = start_index_ZC

# Frequency Offset Estimation
angle_offset = np.angle(correlation_ZC[corrected_start_index_ZC])
frequency_offset_ZC = angle_offset / (np.pi)

# Frequency Correction
# Corrects the entire signal for estimated frequency offset
time_indices = np.arange(len(rx_signal_ZC))
correction = np.exp(1j * 2 * np.pi * frequency_offset_ZC * time_indices / K)
for r in range(Nr):
    rx_signal_ZC[:, r] *= correction

# Estimate peak width using second derivative
# Fine Sync Refinement using Metric Peak
# Refines the timing estimate by maximizing signal energy near metric peak
metric_derivative = np.diff(metric_ZC, n=2)
abs_derivative = np.abs(metric_derivative)
above_half_max = abs_derivative > np.max(abs_derivative) * 0.5
indices = np.where(above_half_max)[0]
if len(indices) > 0:
```

```

metric_peak_width = indices[-1] - indices[0]
else:
    metric_peak_width = CP // 8 # fallback
# Fine synchronization search range
fine_sync_span = int(min(CP / 4, metric_peak_width))
fine_sync_range = np.arange(-fine_sync_span, fine_sync_span + 1)
best_metric = 0
best_shift = 0
for shift in fine_sync_range:
    temp_index = corrected_start_index_ZC + len(ZC_cp) + shift
    if temp_index + CP + K <= len(rx_signal_ZC):
        window = rx_signal_ZC[temp_index + CP : temp_index + CP + K, :]
        temp_metric = np.sum(np.abs(window)**2)
        if temp_metric > best_metric:
            best_metric = temp_metric
            best_shift = shift
# Final fine-tuned index
best_shift=0
corrected_start_index_ZC += best_shift

# Improved sync index detection from Schmidl & Cox metric
peak_idx = np.argmax(metric_ZC)
peak_val = metric_ZC[peak_idx]
threshold = 0.9 * peak_val

# Search left of peak
left_idx = peak_idx
while left_idx > 0 and metric_ZC[left_idx] > threshold:
    left_idx -= 1
if left_idx == 0:
    print("Warning: Left 0.9*max point at index 0 — fallback may be needed")

# Search right of peak
right_idx = peak_idx
while right_idx < len(metric_ZC)-1 and metric_ZC[right_idx] > threshold:
    right_idx += 1

```

```

if right_idx == len(metric_ZC) - 1:
    print("Warning: Right 0.9*max point at end — fallback may be needed")

# Choose center point between edges
corrected_start_index_ZC = int((left_idx + right_idx) // 2)

# Demodulation
# Extracts and demodulates the OFDM data after synchronization
valid_start = corrected_start_index_ZC + len(ZC_cp)
valid_end = valid_start + K
OFDM_RX_noCP_ZC = np.zeros((K, Nr), dtype=complex)
OFDM_demod_ZC = np.zeros((K, Nr), dtype=complex)
for r in range(Nr):
    OFDM_RX_noCP_ZC[:, r] = rx_signal_ZC[valid_start : valid_end, r] #error here!!!
    OFDM_demod_ZC[:, r] = np.fft.fft(OFDM_RX_noCP_ZC[:, r])

# LS Channel Estimation using Pilots
# Estimates channel using pilots and interpolates across all subcarriers
H_LS_pilots_ZC = np.zeros((len(pilotCarriers), Nr), dtype=complex)
H_LS_ZC = np.zeros((K, Nr), dtype=complex)
for r in range(Nr):
    H_LS_pilots_ZC[:, r] = OFDM_demod_ZC[pilotCarriers, r] / pilotValue
    #H_LS_pilots_ZC[:, r] = OFDM_demod_ZC[pilotCarriers, r] / pilotValue[r]
    #interp_func = interp1d(pilotCarriers, H_LS_pilots_ZC[:, r], kind='linear', fill_value="extrapolate")
    #interp_func = interp1d(pilotCarriers, H_LS_pilots_ZC[:, r], kind='cubic', fill_value="extrapolate")
    x=H_LS_pilots_ZC.real
    y=H_LS_pilots_ZC.imag
    interp_func_x= interp1d(pilotCarriers, x[:, r], kind='quadratic')
    interp_func_y= interp1d(pilotCarriers, y[:, r], kind='quadratic')
    interp_func=interp_func_x(allCarriers)+1j*interp_func_y(allCarriers)
    H_LS_ZC[:, r] = interp_func
    #H_LS_ZC[:, r] = H_LS_ZC[:, r] / np.mean(np.abs(H_LS_ZC[:, r]))

```

```
# Equalization

# Combines signals from all antennas using MRC and equalizes channel distortion
equalized_Hest_LS_ZC = np.zeros((K, Nr), dtype=complex)
for r in range(Nr):
    equalized_Hest_LS_ZC[:, r] = OFDM_demod_ZC[:, r] / H_LS_ZC[:, r]

# MRC Equalization
numerator = np.sum(OFDM_demod_ZC[dataCarriers, :] * np.conj(H_LS_ZC[dataCarriers, :]), axis=1)
denominator = np.sum(np.abs(H_LS_ZC[dataCarriers, :])**2, axis=1) + 1e-10
QAM_est = numerator / denominator

# === Compute BER Before FEC (Uncoded BER) ===
# Recreate estimated bits from QAM_est using hard-decision
estimated_bits_preFEC = []

for i in range(len(QAM_est)):
    distances = np.abs(QAM_est[i] - qamValues)
    idx = np.argmin(distances)
    estimated_bits_preFEC.extend([int(b) for b in bitStrings[idx]])

estimated_bits_preFEC = np.array(estimated_bits_preFEC)

# Match length with original encoded bits
min_len = min(len(bits), len(estimated_bits_preFEC))
BER_preFEC = np.sum(bits[:min_len] != estimated_bits_preFEC[:min_len]) / min_len
print(f"BER Before FEC: {BER_preFEC:.6f}")

# Soft-decision demapper (LLRs for 16-QAM)
def qam16_soft_demapper(rx_syms, noise_var):
    LLRs = []
    for sym in rx_syms:
        distances = np.abs(sym - qamValues)**2
        exponents = -distances / noise_var
        exponents -= np.max(exponents) # numerical stability
        probs = np.exp(exponents)
        probs /= np.sum(probs) + 1e-12
```

```

bits_llrs = []
for b in range(4):
    p0 = np.sum(probs[[i for i in range(16) if bitStrings[i][b] == '0']])
    p1 = np.sum(probs[[i for i in range(16) if bitStrings[i][b] == '1']])
    llr = np.log((p1 + 1e-12) / (p0 + 1e-12))
    bits_llrs.append(llr)
LLRs.extend(bits_llrs)
return np.array(LLRs)

# Compute soft bits (LLRs)
rx_llrs = qam16_soft_demapper(QAM_est, sigma2)

# Decode using Viterbi decoder
decoded_bits = enc.viterbi_decoder(rx_llrs, 'soft')

# Truncate to original info bit length
decoded_bits = decoded_bits[:len(info_bits)]

# Compute BER
#BER = np.sum(info_bits != decoded_bits) / len(info_bits)
min_len = min(len(info_bits), len(decoded_bits))
BER = np.sum(info_bits[:min_len] != decoded_bits[:min_len]) / min_len
EVM = np.sqrt(np.mean(np.abs(QAM - QAM_est)**2)) / np.mean(np.abs(QAM))

if BER > -0.001:
    print(f'Start Index (ZC): {corrected_start_index_ZC}')
    print(f'Schmidl Peak: {np.max(metric_ZC):.2f} at Index {np.argmax(metric_ZC)}')
    print(f'Estimated Frequency Offset (ZC): {frequency_offset_ZC:.4f}')
    print(f'FFT Power: {np.mean(np.abs(rx_signal_ZC[valid_start:valid_end])**2):.4f}')
    print(f'Channel Estimation Error (antenna 0): {np.mean(np.abs(H_LS_ZC[:,0] - H_exact[:,0])**2):.4e}')
    print(f'Channel Estimation Error (antenna 1): {np.mean(np.abs(H_LS_ZC[:,1] - H_exact[:,1])**2):.4e}')
    print(f'BER (with FEC): {BER:.6f}')

```

Παράρτημα II [Κώδικας RFSoc]

```
import numpy as np
from scipy.interpolate import interp1d
from scipy.signal import correlate
import matplotlib.pyplot as plt
import plotly.graph_objects as go
import plotly.express as px
import numpy as np
import plotly.io as pio
from scipy.signal.windows import tukey
from scipy.signal import welch
from sk_dsp_comm import fec_conv as fec
from rfsoc_mts import mtsOverlay
import time
import xrfdc

ol = mtsOverlay('mts.bit')
#ol = mtsOverlay('/home/xilinx/mts_wrapper.bit')

# Soft-decision demapper (LLRs for 16-QAM)
def qam16_soft_demapper(rx_syms, noise_var):
    LLRs = []
    for sym in rx_syms:
        distances = np.abs(sym - qamValues)**2
        exponents = -distances / noise_var
        exponents -= np.max(exponents) # numerical stability
        probs = np.exp(exponents)
        probs /= np.sum(probs) + 1e-12

        bits_llrs = []
        for b in range(4):
            p0 = np.sum(probs[[i for i in range(16) if bitStrings[i][b] == '0']])
            p1 = np.sum(probs[[i for i in range(16) if bitStrings[i][b] == '1']])
            llr = np.log((p1 + 1e-12) / (p0 + 1e-12))
```



```

        bits_llrs.append(llr)

    LLRs.extend(bits_llrs)

    return np.array(LLRs)

# waits untill all tiles are in sync
while True:

    try:

        ol.init_tile_sync() #The RF tiles are reset and their internal FIFOs reset via "init_tile_sync"
        ol.verify_clock_tree() # testing if the synthesizer clocks are ready
        ol.sync_tiles() # once initialized, the "sync_tiles" function begins the synchronization procedure of the tiles
        break # Exit loop on success

    except Exception as e:

        print(f'Failed with error: {e}. Retrying...')

        time.sleep(4)

# Parameters

interactiveDiagrams = True # Toggle for enabling/disabling interactive Plotly visualizations

K = 128          # Number of subcarriers
CP = K // 4      # Cyclic Prefix length
P = 64           # Number of pilots
mu = 4           # Bits per QAM symbol (16-QAM)
Nr = 2           # Mode of operation (1 for SISO | 2 for MIMO)
ADC_in_use = 1   # second adc in use

# Subcarrier Allocation

# Allocates pilots and data carriers among the available subcarriers
allCarriers = np.arange(K)

pilotCarriers = allCarriers[::K // P] # Uniform pilot spacing

if allCarriers[-1] not in pilotCarriers: # Ensure last carrier is included

```

```

pilotCarriers = np.append(pilotCarriers, allCarriers[-1])
dataCarriers = np.setdiff1d(allCarriers, pilotCarriers) # Data subcarriers

pilotValue = (np.random.choice([-3,-2,-1,1,2, 3], len(pilotCarriers)) +
              1j * np.random.choice([-3,-2,-1,1,2, 3], len(pilotCarriers)))

# QAM Mapping
# Prepares the 16-QAM mapping from bits to complex symbols
bitStrings = [format(i, '04b') for i in range(16)] # All 4-bit combinations
qamValues = np.array([
    -3-3j, -3-1j, -3+3j, -3+1j,
    -1-3j, -1-1j, -1+3j, -1+1j,
    3-3j, 3-1j, 3+3j, 3+1j,
    1-3j, 1-1j, 1+3j, 1+1j
])

qamMap = {b: q for b, q in zip(bitStrings, qamValues)} # Bit-to-symbol mapping

# FEC Setup
payloadBits_per_OFDM = len(dataCarriers) * mu
info_bits = np.random.randint(0, 2, payloadBits_per_OFDM // 3) # For rate 1/3
enc = fec.FECConv(('1111001', '1011011', '1101101'),35) # 3 outputs → rate 1/3

# FEC Encoding
initial_state = '0' * (enc.constraint_length - 1)
bits,_ = enc.conv_encoder(info_bits, initial_state)
bits = np.array(bits, dtype=int) # Convert from string/float to int

# Map encoded bits to QAM
bits_SP = bits.reshape((len(dataCarriers), mu))
QAM = np.array([qamMap["".join(map(str, bits_SP[i]))] for i in range(len(dataCarriers))])

```

```

# IFFT + CP + Frequency Offset
# Constructs OFDM symbol with pilots, CP, and frequency offset
OFDM_data = np.zeros(K, dtype=complex)
OFDM_data[pilotCarriers] = pilotValue
OFDM_data[dataCarriers] = QAM
OFDM_time = np.fft.ifft(OFDM_data)          # Time-domain OFDM symbol
OFDM_withCP = np.concatenate([OFDM_time[-CP:], OFDM_time]) # Add CP

# Zadoff-Chu Sequence Generation
# Creates Zadoff-Chu sequence used for coarse synchronization
ZC_root = 25
n = np.arange(K // 2)
ZC_seq = np.exp(-1j * np.pi * ZC_root * n**2 / (K // 2)) # ZC base sequence
ZC_seq_upsampled = np.zeros(K, dtype=complex)
ZC_seq_upsampled[::2] = ZC_seq
ZC_seq_upsampled *= np.sqrt(30)
ZC_ifft = np.fft.ifft(ZC_seq_upsampled)      # Convert to time-domain
ZC_cp = np.concatenate([ZC_ifft[-CP:], ZC_ifft]) # Add CP

# === Transmit: Create full OFDM frame ===
tx_signal_ZC = np.concatenate([ZC_cp, OFDM_withCP]) # Prepend ZC

# === Normalize to int16 range ===
scaled = 0.5 * 32767 / np.max(np.abs(tx_signal_ZC)) * tx_signal_ZC
scaled = scaled.astype(np.complex64)

# === Interleave I/Q ===
IQ_interleaved = np.empty(2 * len(scaled), dtype=np.int16)
IQ_interleaved[0::2] = np.real(scaled).astype(np.int16)
IQ_interleaved[1::2] = np.imag(scaled).astype(np.int16)

# === Fill DAC buffer with repeated payload copies ===

```

```

dac_len = ol.dac_player.shape[0]
IQ_len = len(IQ_interleaved)

num_repeats = int(np.ceil(dac_len / IQ_len))
repeated_IQ = np.tile(IQ_interleaved, num_repeats)

# Truncate to exactly match the DAC buffer size
full_IQ = repeated_IQ[:dac_len]

# === Play + Capture ===
AlignedCaptureSamples = np.zeros((3,len(ol.adc_capture_chA)),dtype=np.int16) # initialize variable that will store captured
samples
ol.dac_player[:] = full_IQ # transmit signal via dacs
time.sleep(0.01)
ol.trigger_capture()
#received_iq = ol.dac_capture[:dac_len] # see dac buffer
ol.internal_capture(AlignedCaptureSamples)
#np.save("/home/xilinx/lfs_out/AlignedCaptureSamples.npy", AlignedCaptureSamples)
AlignedCaptureSamples[0][:]=-AlignedCaptureSamples[0][:] # correct for the RFSoc4x2 polarity swap of Tile224 ch0
ADC_D
# tile 226 ch0,ch1 labeled ADC_B, ADC_A are fine

# === Deinterleave ADC channels into complex baseband ===
rx_complex0 = AlignedCaptureSamples[0][0::2].astype(np.float32) + 1j *
AlignedCaptureSamples[0][1::2].astype(np.float32)
if Nr == 2:
    rx_complex1 = AlignedCaptureSamples[ADC_in_use][0::2].astype(np.float32) + 1j *
AlignedCaptureSamples[ADC_in_use][1::2].astype(np.float32)

rx_signal_ZC = np.zeros((len(rx_complex0), Nr), dtype=complex)
rx_signal_ZC[:, 0] = rx_complex0
if Nr == 2:
    rx_signal_ZC[:, 1] = rx_complex1

```

```

# Schmidl & Cox Metric
# Computes Schmidl & Cox metric across signal for fine synchronization
correlation_ZC_all = np.zeros((len(rx_signal_ZC) - K, Nr), dtype=complex)
power_ZC_all = np.zeros((len(rx_signal_ZC) - K, Nr))
for r in range(Nr):
    for d in range(len(rx_signal_ZC) - K):
        first_half = rx_signal_ZC[d:d + K//2, r]
        second_half = rx_signal_ZC[d + K//2:d + K, r]
        correlation_ZC_all[d, r] = np.sum(first_half * np.conj(second_half))
        power_ZC_all[d, r] = np.sum(np.abs(second_half)**2)

correlation_ZC = np.sum((correlation_ZC_all), axis=1)
power_ZC = np.sum(power_ZC_all, axis=1)

metric_ZC = (np.abs(correlation_ZC)**2) / (power_ZC**2 + 1e-10)
start_index_ZC = np.argmax(metric_ZC)
print(f"[Debug] start_index_ZC: {start_index_ZC}")
corrected_start_index_ZC = start_index_ZC

# Frequency Offset Estimation
angle_offset = np.angle(correlation_ZC[corrected_start_index_ZC])
frequency_offset_ZC = angle_offset / (np.pi)

# Frequency Correction
# Corrects the entire signal for estimated frequency offset
time_indices = np.arange(len(rx_signal_ZC))
correction = np.exp(1j * 2 * np.pi * frequency_offset_ZC * time_indices / K)
for r in range(Nr):
    rx_signal_ZC[:, r] *= correction

# Estimate peak width using second derivative
# Fine Sync Refinement using Metric Peak
# Refines the timing estimate by maximizing signal energy near metric peak

```

```

metric_derivative = np.diff(metric_ZC, n=2)
abs_derivative = np.abs(metric_derivative)
above_half_max = abs_derivative > np.max(abs_derivative) * 0.5
indices = np.where(above_half_max)[0]
if len(indices) > 0:
    metric_peak_width = indices[-1] - indices[0]
else:
    metric_peak_width = CP // 8 # fallback
# Fine synchronization search range
fine_sync_span = int(min(CP / 4, metric_peak_width))
fine_sync_range = np.arange(-fine_sync_span, fine_sync_span + 1)
best_metric = 0
best_shift = 0
for shift in fine_sync_range:
    temp_index = corrected_start_index_ZC + len(ZC_cp) + shift
    if temp_index + CP + K <= len(rx_signal_ZC):
        window = rx_signal_ZC[temp_index + CP : temp_index + CP + K, :]
        temp_metric = np.sum(np.abs(window)**2)
        if temp_metric > best_metric:
            best_metric = temp_metric
            best_shift = shift

# Final fine-tuned index
#best_shift=0 # uncomment to remove above refinement
corrected_start_index_ZC += best_shift
print(f"[Debug] corrected_start_index_ZC: {corrected_start_index_ZC}")

# Improved sync index detection from Schmidl & Cox metric
peak_idx = np.argmax(metric_ZC)
peak_val = metric_ZC[peak_idx]
threshold = 0.9 * peak_val

# Search left of peak
left_idx = peak_idx
while left_idx > 0 and metric_ZC[left_idx] > threshold:

```

```

    left_idx -= 1
if left_idx == 0:
    print("Warning: Left 0.9*max point at index 0 — fallback may be needed")

# Search right of peak
right_idx = peak_idx
while right_idx < len(metric_ZC)-1 and metric_ZC[right_idx] > threshold:
    right_idx += 1
if right_idx == len(metric_ZC) - 1:
    print("Warning: Right 0.9*max point at end — fallback may be needed")

# Choose center point between edges
corrected_start_index_ZC = int((left_idx + right_idx) // 2)

print(f'[Debug] corrected_start_index_ZC: {corrected_start_index_ZC}')

# Demodulation
# Extracts and demodulates the OFDM data after synchronization
valid_start = corrected_start_index_ZC + len(ZC_cp)
valid_end = valid_start + K
extracted = rx_signal_ZC[valid_start:valid_end, 0]

print(f'[Debug] Extracted block energy: {np.sum(np.abs(extracted)**2):.4f}')
print(f'[Debug] rx_signal_ZC length: {rx_signal_ZC.shape[0]}')
print(f'[Debug] corrected_start_index_ZC: {corrected_start_index_ZC}')
print(f'[Debug] ZC_cp length: {len(ZC_cp)}')
print(f'[Debug] valid_start: {valid_start}')
print(f'[Debug] valid_end: {valid_end}')
print(f'[Debug] Extracted block energy: {np.sum(np.abs(rx_signal_ZC[valid_start:valid_end, 0])**2):.4f}')
actual_len = valid_end - valid_start
expected_len = K
print(f'[Debug] OFDM FFT input length: {actual_len}, expected: {expected_len}')

OFDM_RX_noCP_ZC = np.zeros((K, Nr), dtype=complex)
OFDM_demod_ZC = np.zeros((K, Nr), dtype=complex)

```

```

for r in range(Nr):
    OFDM_RX_noCP_ZC[:, r] = rx_signal_ZC[valid_start : valid_end, r]
    OFDM_demod_ZC[:, r] = np.fft.fft(OFDM_RX_noCP_ZC[:, r])

# LS Channel Estimation using Pilots
# Estimates channel using pilots and interpolates across all subcarriers
H_LS_pilots_ZC = np.zeros((len(pilotCarriers), Nr), dtype=complex)
H_LS_ZC = np.zeros((K, Nr), dtype=complex)
for r in range(Nr):
    H_LS_pilots_ZC[:, r] = OFDM_demod_ZC[pilotCarriers, r] / pilotValue
    x=H_LS_pilots_ZC.real
    y=H_LS_pilots_ZC.imag
    interp_func_x= interp1d(pilotCarriers, x[:, r], kind='quadratic')
    interp_func_y= interp1d(pilotCarriers, y[:, r], kind='quadratic')
    interp_func=interp_func_x(allCarriers)+1j*interp_func_y(allCarriers)
    H_LS_ZC[:, r] = interp_func

# Equalization
# Combines signals from all antennas using MRC and equalizes channel distortion
equalized_Hest_LS_ZC = np.zeros((K, Nr), dtype=complex)
for r in range(Nr):
    equalized_Hest_LS_ZC[:, r] = OFDM_demod_ZC[:, r] / H_LS_ZC[:, r]
# MRC Equalization
numerator = np.sum(OFDM_demod_ZC[dataCarriers, :] * np.conj(H_LS_ZC[dataCarriers, :]), axis=1)
denominator = np.sum(np.abs(H_LS_ZC[dataCarriers, :])**2, axis=1) + 1e-10
QAM_est = numerator / denominator

# == Compute BER Before FEC (Uncoded BER) ==
# Recreate estimated bits from QAM_est using hard-decision
estimated_bits_preFEC = []

for i in range(len(QAM_est)):

```



```

distances = np.abs(QAM_est[i] - qamValues)
idx = np.argmin(distances)
estimated_bits_preFEC.extend([int(b) for b in bitStrings[idx]])

estimated_bits_preFEC = np.array(estimated_bits_preFEC)

# Match length with original encoded bits
min_len = min(len(bits), len(estimated_bits_preFEC))
BER_preFEC = np.sum(bits[:min_len] != estimated_bits_preFEC[:min_len]) / min_len
print(f'BER Before FEC: {BER_preFEC:.6f}')

# === Estimate Noise Variance from Equalized Symbols ===
# Estimate from pilot error or from difference between QAM and QAM_est
#signal_power = np.mean(np.abs(QAM)**2)
#error_power = np.mean(np.abs(QAM_est - QAM)**2)
#sigma2 = error_power # crude but useful for soft-decision

# === Estimate Noise Variance Estimate from Pilot Errors ===
pilot_errors = OFDM_demod_ZC[pilotCarriers, :] - pilotValue[:, np.newaxis] * H_LS_pilots_ZC
sigma2 = np.mean(np.abs(pilot_errors)**2)

# Compute soft bits (LLRs)
rx_llrs = qam16_soft_demapper(QAM_est, sigma2)
rx_llrs*=3.0 #increase llr confidence
# Decode using Viterbi decoder
decoded_bits = enc.viterbi_decoder(rx_llrs, 'soft')

# Truncate to original info bit length
decoded_bits = decoded_bits[:len(info_bits)]

# Compute BER
min_len = min(len(info_bits), len(decoded_bits))
BER = np.sum(info_bits[:min_len] != decoded_bits[:min_len]) / min_len
EVM = np.sqrt(np.mean(np.abs(QAM - QAM_est)**2)) / np.mean(np.abs(QAM))

```

```
print(f'Start Index (ZC): {corrected_start_index_ZC}')
print(f'Schmidl Peak: {np.max(metric_ZC):.2f} at Index {np.argmax(metric_ZC)}')
print(f'Estimated Frequency Offset (ZC): {frequency_offset_ZC:.4f}')
print(f'FFT Power: {np.mean(np.abs(rx_signal_ZC[valid_start:valid_end])**2):.4f}')
print(f'BER (with FEC): {BER:.6f}')
print(f'EVM: {EVM:.6f}')
signal_power = np.mean(np.abs(tx_signal_ZC)**2)
print(f'RX Signal power: {signal_power:.4f}. Noise power: {sigma2:.4f}')
```

Παράρτημα III [Github]

<https://github.com/whatDoElectricSheepDream/Undegraduate-thesis>