



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Ανάπτυξη Αξιόπιστου Δέκτη με Τεχνικές Βαθιάς Μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Κιουράνα Αναστάσιου

Επιβλέπων : Βουγιούκας Δημοσθένης

Μέλη εξεταστικής επιτροπής:

Γκουμόπουλος Χρήστος

Σκούτας Δημήτριος

Σάμος, Οκτώβριος 2020

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πρόλογος και ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στα πλαίσια του προπτυχιακού προγράμματος σπουδών του τμήματος Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων του Πανεπιστημίου Αιγαίου. Η διαδικασία εκπόνησης της διπλωματικής είναι κομβικό σημείο για την πορεία κάθε φοιτητή. Για αυτό τον λόγο θα ήθελα να ευχαριστήσω ιδιαίτερα τον κύριο Δημοσθένη Βουγιούκα, επιβλέποντα καθηγητή της διπλωματικής, για την πολύτιμη βοήθεια του στην εκπόνηση της διπλωματικής καθώς επίσης και για την καθοδήγηση, τον χρόνο και το κουράγιο του σε όλη την διάρκεια αυτής της προσπάθειας. Επιπλέον, θα ήθελα να ευχαριστήσω τον κύριο Νικόλαο Σοφούλη, που με βοήθησε με τις συμβουλές του να ξεπεράσω κάποιες δυσκολίες που προέκυψαν. Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου για την στήριξη και τις συμβουλές που μου έδιναν καθ' όλη την διάρκεια της προσπάθειας μου για την εκπόνηση της διπλωματικής εργασίας.

© 2020

του

Κιουράνα Αναστάσιου

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Η σελίδα αυτή είναι σκόπιμα λευκή

Πίνακας περιεχομένων

Πρόλογος και ευχαριστίες	3
Λίστα Σχημάτων	8
Λίστα Πινάκων	10
Ακρωνύμια.....	11
Περίληψη	13
Abstract.....	14
1 Εισαγωγή	15
1.1 Η χρήση της Μηχανικής Μάθησης στο κλάδο των Τηλεπικοινωνιών.....	15
1.2 Αντικείμενο της Διπλωματικής.....	15
1.3 Δομή της Διπλωματικής.....	16
2 Ασύρματες επικοινωνίες.....	17
2.1 Διαμόρφωση σήματος.....	17
2.2 Αναλογικές Διαμορφώσεις	17
2.2.1 Διαμόρφωση Πλάτους (AM).....	17
2.2.2 Διαμόρφωση συχνότητας (FM).....	18
2.2.3 Διαμόρφωση Φάσης (PM)	18
2.3 Ψηφιακές Διαμορφώσεις	18
2.3.1 Διαμόρφωση μετατόπισης συχνότητας (FSK).....	19
2.3.2 Διαμόρφωση μετατόπισης πλάτους (ASK).....	19
2.3.3 Διαμόρφωση μετατόπισης φάσης (PSK)	20
2.3.4 Διαμόρφωση M-QAM.....	20
2.4 Δέκτης Συσχέτισης	21
2.5 Ρυθμοί μετάδοσης σφαλμάτων για όλα τα είδη διαμόρφωσης.....	22
2.6 Τεχνικές Πολλαπλής Πρόσβασης.....	24
2.7 Πολυπλεξία	25
2.7.1 Πολυπλεξία με Διαίρεση Χρόνου (TDM).....	26
2.7.2 Πολυπλεξία με Διαίρεση Συχνότητας (FDM).....	26
2.8 Rayleigh Fading Channel.....	27
3 Εισαγωγή στο OFDM.....	28
3.1 Μετάδοση απλού φέροντος έναντι πολλαπλών φερόντων	28
3.1.1 Μετάδοση απλού φέροντος.....	28
3.1.2 Μετάδοση Πολλαπλών φερόντων.....	31

3.1.3 Μετάδοση απλού φέροντος έναντι πολλαπλών φερόντων	36
3.2 Βασική αρχή του OFDM	36
3.2.1 Διαμόρφωση και Αποδιαμόρφωση OFDM.....	36
3.2.2 Διάστημα Φύλαξης OFDM.....	40
3.2.3 Ζώνη Φύλαξης OFDM.....	44
3.2.4 BER of OFDM Scheme	48
3.3 Εφαρμογές OFDM.....	50
3.4 Πλεονεκτήματα και Μειονεκτήματα της OFDM	50
4 Μηχανική Μάθηση.....	51
4.1 Τεχνικές Μηχανικής Μάθησης.....	51
4.1.1 Αναδρομή.....	51
4.1.2 Κατηγοριοποίηση (Classification)	52
4.1.3 Clustering	52
4.1.4 Μείωση Διαστάσεων (Dimensionality Reduction).....	52
4.1.5 Νευρωνικά Δίκτυα και Βαθιά Μάθηση	53
4.1.6 Μεταφορά Μάθησης (Transfer Learning)	54
4.1.7 Word Embeddings.....	55
4.1.8 Επεξεργασία Φυσικής Γλώσσας	55
5 Υλοποίηση TensorFlow και Matlab	56
5.1 Εισαγωγή στο TensorFlow	56
5.2 CUDA	57
5.2.1 Γενικά για το CUDA.....	57
5.2.2 CUDA στην Βαθιά Μάθηση	57
5.2.3 Χρησιμότητα του CUDA για την παρούσα διπλωματική.....	57
5.3 Βήματα Σχεδιασμού και Υλοποίησης του Μοντέλου	58
5.4 Διαδικασία της διαμόρφωσης OFDM στο Matlab και δημιουργία του Dataset	58
5.5 Ανάπτυξη μοντέλου στο περιβάλλον του PyCharm	69
5.6 Επεξεργασία δεδομένων αρχείων csv.....	71
5.7 Preprocessing των δεδομένων από το TensorFlow	71
5.8 Μοντέλα που χρησιμοποιούνται στις τεχνικές Βαθιάς Μάθησης	73
5.8.1 Autoencoders.....	74
5.8.2 Deep Belief Net.....	74
5.8.3 Convolutional Neural Network	74
5.8.4 Recurrent Neural Networks.....	75
5.8.4.1 Long Short Term Memory	75

5.8.5 Ενισχυτική Μάθηση στα νευρωνικά δίκτυα	76
5.9 Επιλογή κατάλληλου μοντέλου	76
5.10 Activation Function	79
5.11 Αξιολόγηση και Προβλέψεις	80
5.12 Συμπεράσματα προσομοίωσης	92
6 Μελλοντικές Επεκτάσεις	93
6.1 UPMC	93
6.2 OFDMA	93
6.3 CDMA	93
6.4 Πρακτικές Εφαρμογές	94
BIBΛΙΟΓΡΑΦΙΑ	95
ΠΑΡΑΡΤΗΜΑ	97
Κώδικας Python για ανάπτυξη μοντέλου	97
Κώδικας για επεξεργασία των csv αρχείων	107

Λίστα Σχημάτων

Εικόνα 1 : Αστερισμοί διαμορφώσεων M-QAM και M-PSK.....	21
Εικόνα 2 : Δέκτης συσχέτισης.....	22
Εικόνα 3 : Σύστημα OFDM.....	28
Εικόνα 4 : Χαρακτηριστικά δομής και συχνότητας ενός πολυκαναλικού συστήματος μετάδοσης.....	32
Εικόνα 5 : Χαρακτηριστικά δομής και φάσματος ενός πολυκαναλικού συστήματος μετάδοσης.....	33
Εικόνα 6 : Χαρακτηριστικά δομής και φάσματος ενός σχήματος μετάδοσης OFDM.....	35
Εικόνα 7 : Χαρακτηριστικά δομής και φάσματος ενός σχήματος μετάδοσης OFDM.....	35
Εικόνα 8 : Διάγραμμα διαμόρφωσης/αποδιαμόρφωσης OFDM για N=6.....	39
Εικόνα 9 : Διάγραμμα πομπού και δέκτη σε ένα OFDM σύστημα.....	39
Εικόνα 10 : Κρουστική απόκριση καναλιού διακριτού χρόνου.....	41
Εικόνα 11 : Αποτέλεσμα καναλιού πολλαπλών διαδρομών στο ληφθέν σήμα χωρίς το διάστημα προστασίας.....	41
Εικόνα 12 : OFDM σύμβολα με CP και περιγραφή του χρόνου/συχνότητας αυτών.....	43
Εικόνα 13 : Επίδραση του καναλιού πολλαπλών διαδρομών στα OFDM σύμβολα με CP.....	44
Εικόνα 14 : Επίδραση του καναλιού πολλαπλών διαδρομών στα OFDM σύμβολα με ZP.....	45
Εικόνα 15 : Αντιγραφή και πρόσθεση του διαστήματος προστασίας του επόμενου συμβόλου στο head κομμάτι του τρέχοντος συμβόλου.....	46
Εικόνα 16 : Ισχύς φάσματος 2 OFDM συμβόλων, ένα με ZP ένα με CP.....	46
Εικόνα 17 : Raised cosine window for OFDM symbol.....	47
Εικόνα 18 : Φάσμα ισχύος των παραθύρων RC.....	48
Εικόνα 19 : OFDM Διαμορφωτής.....	49
Εικόνα 20 : OFDM Αποδιαμορφωτής.....	49
Εικόνα 21 : Νευρωνικό Δίκτυο με ένα κρυμμένο επίπεδο.....	53
Εικόνα 22 : Νευρωνικό δίκτυο με πολλά κρυμμένα επίπεδα.....	54
Εικόνα 23 : Υπολογισμοί στους κόμβους του γραφήματος.....	56
Εικόνα 24 : Μοντέλο 16QAM-OFDM.....	59
Εικόνα 25 : Μοντέλο προσομοίωσης 16QAM-OFDM με Rayleigh Fading Channel.....	59
Εικόνα 26 : Random Integer Generator.....	60
Εικόνα 27 : Rectangular QAM Modulator Baseband.....	61
Εικόνα 28 : OFDM Modulator.....	62
Εικόνα 29 : Rayleigh Fading Channel.....	63
Εικόνα 30 : AWGN Channel.....	64
Εικόνα 31 : OFDM Demodulator.....	65
Εικόνα 32: Αποδιαμορφωτής 16-QAM.....	66
Εικόνα 33 : Error Rate Calculation.....	67
Εικόνα 34 : Το Workspace.....	68
Εικόνα 35 : Μετατροπή σε csv αρχείο.....	68
Εικόνα 36 : Δημιουργία περιβάλλοντος για TensorFlow.....	69
Εικόνα 37 : Πακέτα του περιβάλλοντος.....	70

Εικόνα 38 : Περιβάλλον διερμηνέα για το project	70
Εικόνα 39 : Preprocessing	72
Εικόνα 40 : Δημιουργία του dataset για την εκπαίδευση	73
Εικόνα 41 : Dense επίπεδο	77
Εικόνα 42 : Επίπεδα μοντέλου που επιλέχθηκε	78
Εικόνα 43 : Optimizer	78
Εικόνα 44 : loss	78
Εικόνα 45 : ReLu vs Sigmoid.....	79
Εικόνα 46 : Softmax	80
Εικόνα 47 : Εκτύπωση προβλέψεων	80
Εικόνα 48 : Μερικές προβλέψεις του μοντέλου.....	81
Εικόνα 49 : BER value by EbNo value	82
Εικόνα 50 : BPSK-OFDM BER by EbNo value	83
Εικόνα 51 : Διάγραμμα ακρίβειας.....	84
Εικόνα 52 : Διάγραμμα απώλειας	84
Εικόνα 53 : Γράφημα επιπέδων.....	85
Εικόνα 54 : Μεταδεδομένα επιπέδου cu_dnnlstm	86
Εικόνα 55 : Επίπεδα μοντέλου	87
Εικόνα 56 : Αποτελέσματα μοντέλου του δέκτη.....	87
Εικόνα 57 : Κώδικας Python για τα αποτελέσματα του μοντέλου.....	88
Εικόνα 58 : Κώδικας Python για δημιουργία διαγραμμάτων.....	89
Εικόνα 59 : Αξιοπιστία Ασύρματου Δέκτη	90
Εικόνα 60 : Ακρίβεια Μοντέλου	91
Εικόνα 61: MSE	91

Λίστα Πινάκων

Πίνακας 1 : Ρυθμοί μετάδοσης σφαλμάτων 1/2	23
Πίνακας 2 : Ρυθμοί μετάδοσης σφαλμάτων 2/2	24

Ακρωνύμια

BER	Bit Error Ratio
SNR	Signal to Noise Ratio
OFDM	Orthogonal Frequency Division Multiplexing
LSTM	Long Short Term Memory
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
CUDA	Compute Unified Device Architecture
FFT	Fast Fourier Transform
ISI	Inter Symbol Interference
DFT	Discrete Fourier Transform
IDFT	Inverse Discrete Fourier Transform
IFFT	Inverse Fast Fourier Transform
QAM	Quadrature Amplitude Modulation
CP	Cyclic Prefix
ICI	Inter Channel Interference
ACI	Adjacent Channel Interference
AWGN	Additive White Gaussian Noise
RELU	Rectified Linear Unit
TPU	Tensor Processing Unit
ML	Machine Learning
FIR	Finite Impulse Response
MIMO	Multiple Input Multiple Output
MLSD	Maximum Likelihood Sequence Detection
IoT	Internet of Things
RBM	Restricted Boltzmann Machines
TDM	Time Division Multiplexing
DAMA	Demand Assigned Multiple Access

PAMA	Pre-Assigned Multiple Access
CS	Cyclic Suffix
BPF	Berkeley Packet Filter
RC	Raised Cosine
SC	Sub-Carriers
NLP	Natural Language Processing
UFMC	Universal Filtered Multi-Carrier
OFDMA	Orthogonal Frequency Division Multiple Access
CDMA	Code Division Multiple Access
SC-OFDMA	Single Carrier - Orthogonal Frequency Division Multiple Access
AM	Amplitude Modulation
FM	Frequency Modulation
PM	Phase Modulation
PSK	Phase Shift Keying
FSK	Frequency Shift Keying
CPM	Continuous Phase Modulation
BPSK	Binary Phase Shift Keying
OFSK	Offset Frequency Shift Keying
MSK	Minimum Shift Keying
ASK	Amplitude Shift Keying
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
UMTS	Universal Mobile Telecommunications System

Περίληψη

Ο χώρος των Τηλεπικοινωνιών και πιο συγκεκριμένα των Ασύρματων Επικοινωνιών, έχει μια ραγδαία εξέλιξη τα τελευταία χρόνια. Η τεχνολογία των ασύρματων τηλεπικοινωνιών αναζητά μεθόδους ώστε να ικανοποιήσει τις απαιτήσεις για γρήγορη μετάδοση της πληροφορίας αξιοποιώντας όσο καλύτερα γίνεται το διαθέσιμο φάσμα και μειώνοντας το κόστος και την πολυπλοκότητα, ενώ ταυτόχρονα να παρέχει υψηλή ποιότητα υπηρεσίας. Η μέθοδος Πολυπλεξίας Ορθογωνικής Διαίρεσης Συχνότητας αποτελεί μία ευέλικτη μέθοδος πολυπλεξίας καναλιών με αυξημένη χωρητικότητα που εφαρμόστηκε στις τηλεπικοινωνίες στα τέλη του προηγούμενου αιώνα. Με τη τεχνική αυτή επιτεύχθηκε υψηλός ρυθμός μετάδοσης ακόμα και σε πολυδιαδρομικά περιβάλλοντα διάδοσης. Η μέθοδος αυτή βασίστηκε στη μετάδοση πολλαπλών φερόντων με τη διατήρηση της ορθογωνικότητας μεταξύ τους ώστε να αντιμετωπιστεί το πρόβλημα της διασυμβολικής παρεμβολής κατά τις ασύρματες μεταδόσεις μέσω πολυδιαδρομικών διαύλων.

Αντίστοιχο φαινόμενο παρατηρείται και στον χώρο της Μηχανικής Μάθησης (Machine Learning) όπου οι εφαρμογές της ολοένα αναπτύσσονται και χρησιμοποιούνται ευρέως και όλο και πιο πολύ στην καθημερινότητά μας. Πτυχές της μηχανικής γνώσης όπως η Βαθιά Μάθηση (Deep Learning), μπορούν να χρησιμοποιηθούν στο πεδίο των ασύρματων τηλεπικοινωνιών για την ανάπτυξη αξιόπιστων μοντέλων. Με την χρήση κατάλληλων εργαλείων όπως το TensorFlow καθώς και της γλώσσας προγραμματισμού Python μπορούν να δημιουργηθούν και να εκπαιδευθούν μοντέλα για την αντιμετώπιση διάφορων προβλημάτων.

Στην παρούσα διπλωματική παρουσιάζεται η ανάπτυξη ενός αξιόπιστου δέκτη με την βοήθεια των τεχνικών της Βαθιάς Μάθησης οι οποίες μπορούν να απλοποιήσουν τον τρόπο λειτουργίας των δεκτών και να παρέχουν μεγάλη ακρίβεια στα σήματα λήψης. Με τις τεχνικές αυτές θα αναπτυχθεί ένα μοντέλο Βαθιάς Μάθησης το οποίο θα βοηθήσει στη σωστή λειτουργία του δέκτη ώστε να παρέχει αξιοπιστία. Επίσης, με τη μέθοδο OFDM θα γίνει η προσομοίωση του δέκτη στο MATLAB για την παραγωγή του Dataset που θα χρησιμοποιηθεί για την εκπαίδευση και τον έλεγχο του μοντέλου. Ακολουθεί η ανάπτυξη ενός μοντέλου με τη χρήση του TensorFlow, ο έλεγχος, και η επίδοση του όσον αφορά την ακρίβεια καθώς και η αξιολόγηση των αποτελεσμάτων. Τέλος, θα εξεταστούν διαγράμματα με συγκρίσεις τριών μεθόδων διαμόρφωσης σε συνδυασμό με την OFDM σχετικά την τιμή προσομοίωσης BER (Bit Error Ratio), καθώς επίσης και της θεωρητικής τιμής BER αυτών.

Λέξεις Κλειδιά: Λόγος Σφαλμάτων Bit, TensorFlow, Πολυπλεξία Ορθογωνικής Διαίρεσης Συχνότητας, Μηχανική Μάθησης, Βαθιά Μάθηση, Python, MATLAB, Δέκτης, Παρεμβολή, Φέρον, Σύμβολο

Abstract

The field of Telecommunications, and more specifically of Wireless Communications, has a rapid development in recent years. Wireless telecommunications technology is looking for methods to meet the requirements for fast transmission of information by making the best use of the available spectrum and reducing costs and complexity, while providing high quality service. Orthogonal Frequency Division Multiplexing is a flexible method of multiplexing channels with increased capacity that was applied to telecommunications at the end of the last century. With this technique a high transmission rate was achieved even in multidirectional propagation environments. This method was based on the transmission of multiple carriers while maintaining the orthogonality between them in order to address the problem of inter-symbol interference during wireless transmission via multi-channel channels.

A similar phenomenon is observed in the field of Machine Learning where its applications are increasingly developed and used widely and more and more in our daily lives. Aspects of engineering knowledge such as Deep Learning can be used in the field of wireless telecommunications to develop reliable models. Using appropriate tools such as TensorFlow as well as Python, models can be created and trained to deal with various problems.

This thesis presents the development of a reliable receiver with the help of Deep Learning techniques which can simplify the operation of the receivers and provide high accuracy of received signals. These techniques will develop a Deep Learning model that will help the receiver to function properly to provide reliability. Also, with the OFDM method, the receiver will be simulated in MATLAB for the production of Dataset that will be used for training and testing the model. This is followed by the development of a model using TensorFlow, its testing, and performance in terms of accuracy as well as the evaluation of results. Finally, diagrams with comparisons of three modulation methods in conjunction with OFDM regarding the BER (Bit Error Ratio) simulation value as well as their theoretical BER value will be considered.

Keywords: *Bit Error Ratio, TensorFlow, MATLAB, Orthogonal Frequency Division Multiplexing, Machine Learning, Receiver, Interference, Symbol, Carrier, Python*

1 Εισαγωγή

1.1 Η χρήση της Μηχανικής Μάθησης στο κλάδο των Τηλεπικοινωνιών

Μηχανική Μάθηση είναι η μελέτη αλγορίθμων των υπολογιστών που βελτιώνονται αυτόματα μέσω της εμπειρίας που αποκτάται από τα μοντέλα που κατασκευάζουν. Θεωρείται υποσύνολο της Τεχνητής Νοημοσύνης. Οι αλγόριθμοι μηχανικής μάθησης κατασκευάζουν μαθηματικά μοντέλα βασισμένα σε δεδομένα δειγμάτων, τα οποία είναι γνωστά ως «training data», για να κάνουν προβλέψεις ή να πάρουν αποφάσεις χωρίς να είναι ρητά αποφασισμένο να το πράξουν. Η Μηχανική Μάθηση είναι έτοιμη να αναδιαμορφώσει κάθε πτυχή της βιομηχανίας των Τηλεπικοινωνιών, χωρίς να πρέπει να παραμεληθούν ούτε τα γεγονότα ούτε η διαφημιστική εκστρατεία. Οι πληροφορίες σε πραγματικό χρόνο, που τροφοδοτούνται από την Μηχανική Μάθηση και την Τεχνητή Νοημοσύνη, αλλάζουν το παιχνίδι για τη μείωση του κόστους, τη δημιουργία νέων ροών εσόδων, την ενίσχυση της εμπειρίας των πελατών και την επίτευξη της κλίμακας που απαιτείται από τις τεχνολογίες IoT και 5G. Οι Τηλεπικοινωνίες δεν περιορίζονται πλέον στην παροχή βασικών τηλεφωνικών και διαδικτυακών υπηρεσιών. Στην εποχή του IoT υπάρχουν όλο και περισσότερες δυνατότητες. Έτσι τώρα μπορεί κάποιος να πάρει ένα smartphone (τηλέφωνο και υπηρεσίες διαδικτύου), καθώς και ευρυζωνική σύνδεση, και μια συνδρομή στο Netflix, συνδεδεμένες συσκευές – για παράδειγμα ένα smartwatch, και ούτω καθεξής. Με τον αυξανόμενο αριθμό ευκαιριών και με τόσα πολλά δεδομένα, είναι σαφές ότι όλες αυτές οι πληροφορίες μπορούν εύκολα να ξεφύγουν από τον έλεγχο. Οι αλγόριθμοι της Μηχανικής Μάθησης επιτρέπουν να ανακτηθεί ο έλεγχος του τι μπορεί να γίνει με τα διαθέσιμα δεδομένα. Αυτό είναι ζωτικής σημασίας δεδομένου ότι οι επιχειρηματικές αποφάσεις, γενικά, θα πρέπει να βασίζονται σε πραγματικές πληροφορίες.[12]

1.2 Αντικείμενο της Διπλωματικής

Η παρούσα διπλωματική αφορά την ανάπτυξη ενός αξιόπιστου δέκτη με χρήση τεχνικών βαθιάς μάθησης. Στον χώρο των Τηλεπικοινωνιών και ειδικότερα των Ασύρματων Επικοινωνιών παρατηρούνται προβλήματα που αφορούν την αξιοπιστία του σήματος λόγω του θορύβου που υφίσταται στο σήμα. Πιο συγκεκριμένα, κατά την μετάδοσή του αντιμετωπίζει φυσικά εμπόδια, όπως ο αέρας, βουνά ή κάποιος τοίχος, τα οποία προκαλούν θόρυβο στο σήμα, καθιστώντας το ορισμένες φορές δύσκολο να ανιχνεύσει ο δέκτης αν ένα σήμα είναι αξιόπιστο. Αυτό το πρόβλημα καλείται να λύσει η διπλωματική αυτή, με τη βοήθεια τεχνικών βαθιάς μάθησης. Προσομοιώνοντας ένα μοντέλο διαμόρφωσης OFDM και χρησιμοποιώντας βέλτιστες τεχνικές Βαθιάς Μάθησης για την εκπαίδευση ενός μοντέλου, θα αναπτυχθεί ένας αξιόπιστος δέκτης ο οποίος θα μπορεί με τη βοήθεια του μοντέλου να λειτουργεί σωστά και να ανακτά με

ακρίβεια το σήμα λήψης. Το μοντέλο θα κάνει επίσης προβλέψεις σχετικά με την αξιοπιστία του σήματος, έχοντας βέβαια την κατάλληλη «εμπειρία». Όμως, υπήρχαν δυσκολίες σε αυτό το εγχείρημα που αντιμετωπίστηκαν σε μεγάλο βαθμό. Κάποιες από αυτές ήταν το περιβάλλον της προσομοίωσης της διαμόρφωσης καθώς επίσης και η υπολογιστική ισχύ που απαιτούνταν για την βέλτιστη χρήση του TensorFlow.

1.3 Δομή της Διπλωματικής

Τα κεφάλαια που ακολουθούν αναφέρονται γενικότερα στους κλάδους που εμπλέκονται στο αντικείμενο της διπλωματικής και πιο συγκεκριμένα στα κομμάτια αυτών που χρειάστηκαν για της διεκπεραίωσή της. Στο 2ο κεφάλαιο γίνεται μια αναφορά στις ασύρματες επικοινωνίες που αφορούν τη λειτουργία του δέκτη. Στο επόμενο κεφάλαιο, αναλύεται η διαδικασία της OFDM διαμόρφωσης και αποδιαμόρφωσης καθώς επίσης και ο υπολογισμός του BER που αφορούν την λειτουργία του ασύρματου δέκτη. Το 4ο κεφάλαιο αναλύει και εξηγεί την Μηχανική Μάθησης και αναφέρει μερικές τεχνικές της. Στη συνέχεια, το 5ο κεφάλαιο αναφέρεται στην υλοποίηση του μοντέλου με τη χρήση της γλώσσας Python καθώς και στη διαδικασία της προσομοίωσης της διαμόρφωσης OFDM του δέκτη στο MATLAB για την δημιουργία του dataset που θα χρησιμοποιηθεί για την εκπαίδευση του μοντέλου. Επιπλέον, αναφέρεται το CUDA καθώς και η συμβολή του στην εκπαίδευση του μοντέλου. Στο 6ο κεφάλαιο αναφέρονται μελλοντικές επεκτάσεις της διπλωματικής καθώς και ορισμένες εφαρμογές της.

2 Ασύρματες επικοινωνίες

2.1 Διαμόρφωση σήματος

Στις Τηλεπικοινωνίες διαμόρφωση ονομάζεται η διαδικασία μεταβολής ενός περιοδικού σήματος, συνήθως ενός σήματος υψηλής συχνότητας, με στόχο την κωδικοποίηση σε αυτό ενός σήματος χαμηλής συχνότητας το οποίο μεταφέρει κωδικοποιημένη πληροφορία. Το σήμα αυτό υψηλής συχνότητας τότε καλείται φέρον και συνήθως είναι σήμα απλής συχνότητας (π.χ. μία ημιτονοειδής κυματομορφή). Η διαμόρφωση απαιτείται για να μπορέσει να διέλθει ένα σήμα από κάποιο τηλεπικοινωνιακό κανάλι (ένα καλώδιο, στις ενσύρματες επικοινωνίες, ή ο ελεύθερος χώρος, στις ασύρματες επικοινωνίες) το εύρος ζώνης του οποίου δεν επικαλύπτεται με το εύρος ζώνης του σήματος. Στο άλλο άκρο της επικοινωνίας, στον παραλήπτη, λαμβάνει χώρα η αντίστροφη διαδικασία προκειμένου να ανακτηθεί το αρχικό σήμα, η αποδιαμόρφωση. Μια συσκευή η οποία μπορεί να επιτελεί και τις δύο διεργασίες, έτσι ώστε να λειτουργεί και ως αποστολέας και ως παραλήπτης, ονομάζεται μόντεμ (modem, εκ των αγγλικών λέξεων Modulator-Demodulator).[1]

2.2 Αναλογικές Διαμορφώσεις

Στις αναλογικές διαμορφώσεις το σήμα πληροφορίας που διαμορφώνεται είναι σε αναλογική μορφή.

2.2.1 Διαμόρφωση Πλάτους (AM)

Στη διαμόρφωση πλάτους μεταβάλλεται το πλάτος του φέροντος κύματος υψηλής συχνότητας ανάλογα με το πλάτος του σήματος πληροφορίας. Το διαμορφωμένο σήμα που προκύπτει έχει σταθερή συχνότητα και μεταβαλλόμενο πλάτος. Η κύρια εφαρμογή της διαμόρφωσης AM είναι στην ραδιοφωνία. Η τεχνική της AM όμως μειονεκτεί σε σχέση με την FM στο ότι επηρεάζεται εντονότερα από τον ηλεκτρονικό θόρυβο (παράσιτα). Τα κυκλώματά της όμως πλεονεκτούν σε χαμηλές συχνότητες και έτσι εξακολουθεί να χρησιμοποιείται σε εκπομπές στην ζώνη των μακρών και των μεσαίων κυμάτων. Με διαμόρφωση AM διαμορφώνεται και το σήμα εικόνας του αναλογικού τηλεοπτικού σήματος (το σήμα ήχου διαμορφώνεται με FM). Η διαμόρφωση AM χωρίζεται σε:

- a) Διαμόρφωση μονής πλευρικής ζώνης (SSB)
και
- b) Διαμόρφωση διπλής πλευρικής ζώνης (DSB)

2.2.2 Διαμόρφωση συχνότητας (FM)

Στην διαμόρφωση συχνότητας η συχνότητα του σήματος υψηλής συχνότητας (φέρων κύμα) μεταβάλλεται ανάλογα με το πλάτος του σήματος πληροφορίας (ακουστικό σήμα). Το διαμορφωμένο σήμα που προκύπτει έχει σταθερό πλάτος αλλά μεταβαλλόμενη συχνότητα και μοιάζει να παρουσιάζει «πυκνώματα» και «αραιώματα». Ο τρόπος με τον οποίο το πλάτος του ακουστικού σήματος επηρεάζει το φέρων είναι ο εξής. Όπου το ακουστικό σήμα έχει μεγάλο πλάτος έχουμε αύξηση της συχνότητας του φέροντος και όπου το ακουστικό σήμα έχει μικρό πλάτος έχουμε μείωση της συχνότητας του φέροντος.

2.2.3 Διαμόρφωση Φάσης (PM)

Η διαμόρφωση φάσης είναι μια μορφή διαμόρφωσης που αντιπροσωπεύει πληροφορίες όπως τις διακυμάνσεις στην στιγμιαία φάση ενός κύματος. Σε αντίθεση με τη πιο διαδεδομένη μορφή διαμόρφωσης, τη διαμόρφωση συχνότητας, η PM δε χρησιμοποιείται ιδιαίτερα για ραδιοφωνικές μεταδόσεις. Αυτό συμβαίνει επειδή τείνει να απαιτείται πιο περίπλοκη χρήση υλικού και μπορεί να υπάρξουν προβλήματα ασάφειας στο να καθοριστεί αν, για παράδειγμα, το σήμα έχει αλλάξει φάση από $+180^\circ$ ή -180° . Η διαμόρφωση φάσης χρησιμοποιείται, ωστόσο, σε ψηφιακούς συνθέτες μουσικής, ακόμα και αν τα μέσα αυτά συνήθως αναφέρονται ως "FM" συνθέτες (synthesizers) (και οι δύο αυτοί τύποι διαμόρφωσης (PM και FM) ακούγονται πολύ παρόμοιοι, αλλά η PM είναι συνήθως ευκολότερο να εφαρμοστεί σε αυτόν τον τομέα).

2.3 Ψηφιακές Διαμορφώσεις

Όπως στις αναλογικές διαμορφώσεις, έτσι και στις ψηφιακές, η διαδικασία της διαμόρφωσης είναι η μεταβολή ενός από τα τρία χαρακτηριστικά (πλάτος, φάση, συχνότητα) ενός φέροντος σήματος από το σήμα πληροφορίας. Το σήμα πληροφορίας πλέον είναι η ροή δυαδικών ψηφίων όπως ουσιαστικά εξέρχεται από τη διαδικασία της μορφοποίησης ή της κωδικοποίησης. Στη γενική περίπτωση ο ψηφιακός διαμορφωτής επεξεργάζεται m συνεχόμενα δυαδικά ψηφία και παράγει μία από τις διαθέσιμες κυματομορφές. Κατά συνέπεια, ο διαμορφωτής "αντιστοιχίζει" μία ακολουθία από m διαδοχικά bits της εξόδου του κωδικοποιητή διαύλου σε μία από τις καταστάσεις συμβόλων με τέτοιο τρόπο και με τέτοια δομή, ώστε να είναι κατάλληλες προς μετάδοση. Η επιλογή της τεχνικής της ψηφιακής διαμόρφωσης επιλέγεται ανάλογα με την αποδοτικότητα φάσματος σε σχέση με τον ρυθμό μετάδοσης των παρεχόμενων υπηρεσιών, την ευκολία υλοποίησης και την ανθεκτικότητα σε σφάλματα. Συνήθως, στις δορυφορικές επικοινωνίες απαιτείται διαμόρφωση σταθερής περιβάλλουσας, ώστε να ελαχιστοποιούνται τα φαινόμενα μη-γραμμικής ενίσχυσης στους ενισχυτές υψηλής ισχύος. Γι' αυτό το λόγο προτιμάται η διαμόρφωση φάσης (PSK), έχοντας το πλεονέκτημα της σταθερής περιβάλλουσας και της καλύτερης φασματικής απόδοσης

σε σχέση με άλλες τεχνικές (π.χ. FSK). Ως φασματική απόδοση Γ (bits/sec/Hz) ορίζεται ο λόγος R_c/B , όπου B είναι το καταλαμβανόμενο εύρος ζώνης μετάδοσης του ψηφιακού σήματος από τον πομπό. Το πρόβλημα των PSK τεχνικών είναι η ασυνεχής μετάβαση της φάσης από σύμβολο σε σύμβολο. Για τον λόγο αυτό, χρησιμοποιούνται ευρέως και οι τεχνικές διαμόρφωσης συνεχούς φάσης (CPM), όπως οι OQPSK, MSK, $\pi/4$ -QPSK, οι οποίες παρουσιάζουν λιγότερη ενέργεια στους πλευρικούς λοβούς του φάσματος. Η κατάλληλη τεχνική ψηφιακής διαμόρφωσης επιλέγεται με βάση την αποδοτικότητα του φάσματος και της ισχύος ή, ισοδύναμα, την πιθανότητα σφάλματος κατά τη λήψη.[2] Η πιστότητα της μετάδοσης και κατά συνέπεια η αξιοπιστία του ψηφιακού τηλεπικοινωνιακού συστήματος, προσδιορίζεται από τον ρυθμό σφαλμάτων (Bit-Error-Rate, BER) και την πιθανότητα σφαλμάτων (Bit-Error-Probability, BEP) και, γενικά, εξαρτάται από:

- A. την ισχύ εκπομπής του σήματος,
- B. το σχήμα διαμόρφωσης,
- C. το σχήμα κωδικοποίησης,
- D. την πυκνότητα ισχύος θορύβου του διαύλου,
- E. την απόσταση πομπού και δέκτη,
- F. τη συνολική παραμόρφωση που εισάγεται κατά την επεξεργασία του σήματος από τον πομπό και τον δέκτη.

2.3.1 Διαμόρφωση μετατόπισης συχνότητας (FSK)

Διαμόρφωση μετατόπισης συχνότητας ονομάζεται ο τύπος διαμόρφωσης σήματος όπου ψηφιακά δεδομένα παρουσιάζονται ως αλλαγές στη συχνότητα ενός φέροντος. Τα περισσότερα από τα πρώτα μοντέλα modem χρησιμοποιούσαν διαμόρφωση FSK για να στείλουν και να λάβουν δεδομένα με ρυθμούς μέχρι 300, 600 ή 1200 bits το δευτερόλεπτο. Μερικοί μικρο-υπολογιστές χρησιμοποιούσαν μια ειδική μορφή διαμόρφωσης FSK, το πρότυπο Kansas City, για αποθήκευση δεδομένων σε κασέτες ήχου. Η διαμόρφωση FSK χρησιμοποιείται ακόμη στο ερασιτεχνικό ραδιόφωνο γιατί επιτρέπει μεταφορά δεδομένων από μη τροποποιημένο εξοπλισμό για μετάδοση φωνής. Η ψηφιακή διαμόρφωση συχνότητας έχει χρησιμοποιηθεί έως τελευταία κατά κόρον για τη δημιουργία αλλά και την ανίχνευση, ενώ παράλληλα δεν επηρεάζεται από τις διακυμάνσεις πλάτους που προκαλεί το κανάλι. Η μέθοδος FSK μεταφέρει τα δεδομένα χρησιμοποιώντας φορείς με διακριτές συχνότητες, ώστε να αναπαραστήσει με αυτές, τις καταστάσεις συμβόλων. Μια σημαντική ιδιότητα της FSK είναι ότι το πλάτος του διαμορφωμένου σήματος είναι σταθερό.

2.3.2 Διαμόρφωση μετατόπισης πλάτους (ASK)

Η πιο απλή εφαρμογή ASK είναι η λεγόμενη *on-off* όπου λειτουργεί ως διακόπτης. Η ύπαρξη φέροντος σήματος υποδηλώνει δυαδικό 1 ενώ η απουσία του δυαδικό 0. Η συγκεκριμένη εφαρμογή χρησιμοποιείται ιδιαίτερα για την επικοινωνία μέσω κώδικα Μορς σε ραδιοφωνικές συχνότητες. Σε πιο εξειδικευμένες εφαρμογές όπου υπάρχουν περισσότερα λογικά επίπεδα, δηλαδή χρησιμοποιούνται δεδομένα άνω

του ενός bit, το πλάτος του φέροντος σήματος μπορεί να πάρει περισσότερες διακριτές τιμές. Για ένα bit παίρνει 2 διακριτές τιμές, για 2 bits 4 διακριτές τιμές, για 3 bits 8 διακριτές κ.ο.κ. δηλαδή αν θέλουμε να μεταφέρουμε ένα δεδομένο n bits σε έναν παλμό τότε το πλάτος του φέροντος σήματος ιδανικά μπορεί να πάρει 2^n διακριτές τιμές. Για την μεταφορά δεδομένων πολλών bit σε έναν παλμό πρέπει να λαμβάνεται υπόψιν ο λόγος σήματος/θορύβου (SNR) ώστε να είναι δυνατή η σωστή αποδιαμόρφωση του σήματος. Το φάσμα ενός σήματος ASK μπορεί εύκολα να προσδιοριστεί εάν είναι γνωστό το φάσμα του σήματος δεδομένων βασικής ζώνης. Τότε η διαδικασία της διαμόρφωσης ASK μπορεί να θεωρηθεί ως πολλαπλασιασμός του σήματος βασικής ζώνης με το φέρον.

2.3.3 Διαμόρφωση μετατόπισης φάσης (PSK)

Η διαμόρφωση μετατόπισης φάσης ονομάζεται ο τύπος διαμόρφωσης όπου η πληροφορία περιέχεται στη στιγμιαία φάση του διαμορφωμένου φέροντος σήματος. Στη μέθοδο αυτή η φάση του φέροντος σήματος μεταβάλλεται για να απεικονιστεί το bit ενώ η συχνότητα και το πλάτος παραμένουν σταθερά. Για παράδειγμα αν θεωρηθεί ότι η φάση 0° είναι το 0 τότε μεταβάλλοντας τη φάση κατά 180° είναι το 1. Η φάση μένει σταθερή για κάθε δυαδικό (συντομογραφία για δυαδικό ψηφίο) διάστημα.

Επίσης, η διαμόρφωση μετατόπισης φάσης είναι ένα ψηφιακό σύστημα διαμόρφωσης που μεταφέρει δεδομένα με την αλλαγή ή την αυξομείωση της φάσης του σήματος. Τα ψηφιακά δεδομένα κωδικοποιούνται σε τμήματα ημιτονοειδών σημάτων ίδιου χρονικού εύρους, ίδιας συχνότητας και διαφορετικής αρχικής φάσης. Η PSK χρησιμοποιεί έναν πεπερασμένο αριθμό των φάσεων και στη κάθε φάση αποδίδεται ένας μοναδικός τύπος δυαδικών bits. Συνήθως, κάθε φάση κωδικοποιεί ίσο αριθμό bits. Ο αποδιαμορφωτής, ο οποίος έχει σχεδιαστεί ειδικά για τα σύμβολα που χρησιμοποιούνται από τον διαμορφωτή, καθορίζει τη φάση του λαμβανόμενου σήματος και την αποδιαμορφώνει στο σύμβολο που αντιπροσωπεύει και έτσι γίνεται η ανάκτηση των αρχικών στοιχείων προς αποστολή.

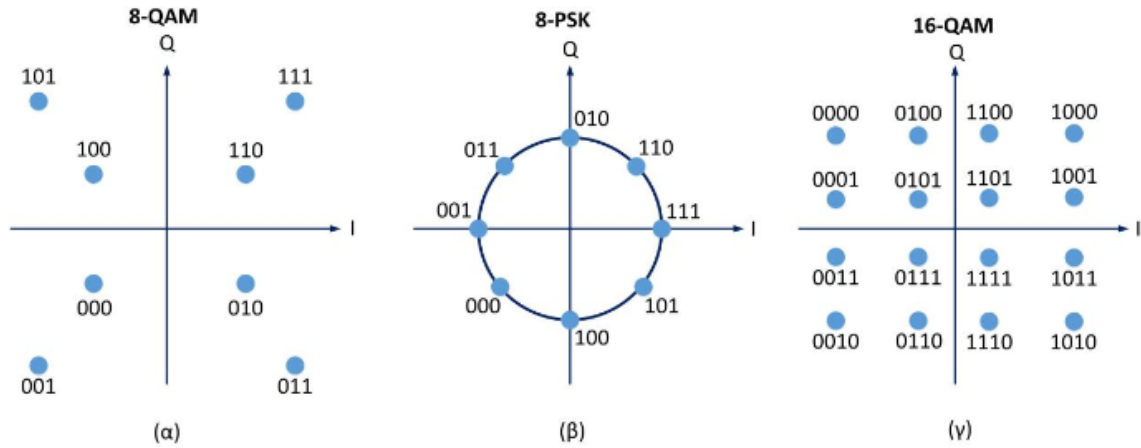
Στην Ψηφιακή Διαμόρφωση Φάσης η πληροφορία περιέχεται στη στιγμιαία φάση του διαμορφωμένου φέροντος. Αυτή η φάση συνήθως ενσωματώνεται στο φέρον και εκτιμάται ως προς ένα σταθερό φέρον αναφοράς γνωστής φάσης, και για αυτό ονομάζεται σύμφωνη PSK (coherent PSK). Στη δυαδική διαμόρφωση PSK (BPSK) χρησιμοποιούνται οι καταστάσεις φάσης 0° και 180° . Είναι επίσης δυνατή η μετάδοση δεδομένων κωδικοποιημένων με τη μορφή μεταβολών φάσης (διαφοράς φάσης) ανάμεσα σε διαδοχικά σύμβολα. Αυτή η μέθοδος ονομάζεται Διαφορικά Σύμφωνη Ψηφιακή διαμόρφωση Φάσης (Differentially Coherent PSK).

2.3.4 Διαμόρφωση M-QAM

Αυτή η διαμόρφωση είναι επίσης μια πολυεπίπεδη (δηλαδή υψηλότερη από δυαδική) μέθοδος διαμόρφωσης, στην οποία το πλάτος και η φάση του φέροντος διαμορφώνονται, αποτελώντας έτσι συνδυασμό των διαμορφώσεων ASK/PSK.

Ονομάζεται και πολυπλεξία ορθογωνικών φερόντων (quadrature-carrier multiplexing). Μπορεί, επίσης, να θεωρηθεί ότι προκύπτει από δύο διαμορφωμένα AM-DSB-SC σήματα, προερχόμενα από την εφαρμογή δύο ανεξάρτητων σημάτων πληροφορίας με διαφορά φάσης 180° , στα οποία επιτρέπεται να καταλάβουν το ίδιο εύρος ζώνης μετάδοσης. Συνεπώς, είναι μια διάταξη εξοικονόμησης εύρους ζώνης (bandwidth-conservation scheme).[2] Ανάλογα με το μέγεθος του αλφάβητου, ο αστερισμός μπορεί να διευρύνεται, χρησιμοποιώντας μεγαλύτερες τιμές πλάτους, όπως φαίνεται στην Εικόνα 1. Ακόμα, όμως, και για το ίδιο αλφάβητο, το σχήμα του αστερισμού μπορεί να διαφέρει. Η γενική μορφή μιας διαμορφωμένης M-QAM κυματομορφής για το i-οστό σύμβολο είναι

$$S_i^{M-QAM}(t) = \sqrt{\frac{2*E_{s,i}}{T_s}} * \cos[2\pi f_0 + \varphi_i], \quad 0 \leq t < T_s, \quad i = 0, 1, \dots, M-1 \quad (I)$$

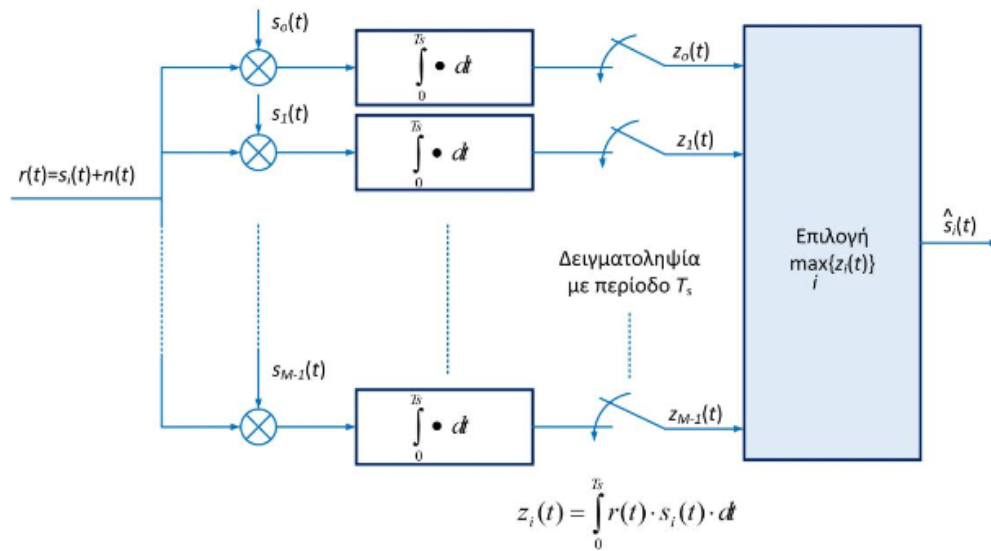


Εικόνα 1 : Αστερισμοί διαμορφώσεων M-QAM και M-PSK

2.4 Δέκτης Συσχέτισης

Ο δέκτης συσχέτισης που φαίνεται στην εικόνα 2, αποτελείται από M διαφορετικούς κλάδους, όσο και το μέγεθος της M-αδικής διαμόρφωσης. Το λαμβανόμενο σήμα τροφοδοτείται παράλληλα σε όλους τους κλάδους. Σε κάθε κλάδο πολλαπλασιάζεται με την αντίστοιχη κυματομορφή που έχει μεταδοθεί και το γινόμενο διέρχεται από έναν ολοκληρωτή, ενώ η έξοδός του δειγματοληπτείται με περίοδο δειγματοληψίας T_s . Τα δείγματα (τυχαίες μεταβλητές) $\bar{z} = \{z_i\}_{i=0}^{M-1}$ (2)

που προκύπτουν κατευθύνονται στον ανιχνευτή μέγιστης πιθανοφάνειας, όπου επιλέγεται η κυματομορφή, που αντιστοιχεί στη μεγαλύτερη τιμή του z_i . Επειδή τα σήματα $r(t)$ και $s_i(t)$ συσχετίζονται, η διάταξη του πολλαπλασιαστή και ολοκληρωτή ονομάζεται συσχετιστής. Η τελική εκτιμώμενη κυματομορφή $s_i(t)$ προκύπτει από το σύνολο των κυματομορφών $\{s_i(t)\}_{i=0}^{M-1}$ που ταιριάζει περισσότερο στο λαμβανόμενο σήμα $r(t)$. [2][3]



Εικόνα 2 : Δέκτης συσχέτισης

2.5 Ρυθμοί μετάδοσης σφαλμάτων για όλα τα είδη διαμόρφωσης

Σε αυτήν την ενότητα απεικονίζονται δύο πίνακες όπου μπορούμε να δούμε τους ρυθμούς μετάδοσης σφάλματος για όλα τα είδη διαμόρφωσης. Καταλαβαίνουμε λοιπόν, ότι για κάθε είδος χρησιμοποιείται διαφορετική συνάρτηση για την εύρεση των SER και BER. Επίσης, παρατηρείται ότι για τα BER και SER της κάθε διαμόρφωσης το μόνο που αλλάζει όσον αφορά τον τύπο είναι ότι για τον λόγο SER χρησιμοποιείται η ενέργεια του συμβόλου E_s ενώ για τον λόγο BER η ενέργεια του bit E_b . [2]

Σχήμα Διαμόρφωσης	Symbol Error Rate (SER)	Bit Error Rate (BER)
OOK	$Q\left(\sqrt{\frac{E_s}{N_o}}\right)$	$Q\left(\sqrt{\frac{E_b}{N_o}}\right)$
FSK (σύμφωνη)	$Q\left(\sqrt{\frac{E_s}{N_o}}\right)$	$Q\left(\sqrt{\frac{E_b}{N_o}}\right)$
FSK (ασύμφωνη)	$\frac{1}{2} \exp\left(-\frac{E_s}{2N_o}\right)$	$\frac{1}{2} \exp\left(-\frac{E_b}{2N_o}\right)$
BPSK	$Q\left(\sqrt{\frac{2E_s}{N_o}}\right)$	$Q\left(\sqrt{\frac{2E_b}{N_o}}\right)$
DE-BPSK	$\approx 2Q\left(\sqrt{\frac{2E_b}{N_o}}\right)$	$\approx 2Q\left(\sqrt{\frac{2E_b}{N_o}}\right)$
DBPSK	$\frac{1}{2} \exp\left(-\frac{E_s}{N_o}\right)$	$\frac{1}{2} \exp\left(-\frac{E_b}{N_o}\right)$
QPSK, OQPSK, 4-QAM	$\approx 2Q\left(\sqrt{\frac{E_s}{N_o}}\right)$	$Q\left(\sqrt{\frac{2E_b}{N_o}}\right)$

Πίνακας 1 : Ρυθμοί μετάδοσης σφαλμάτων 1/2

$\pi/4$ -QPSK	$\approx 2Q\left(\sqrt{\frac{E_s}{N_o}}\right)$	$\approx 2Q\left(\sqrt{\frac{2E_b}{N_o}}\right)$
DE-QPSK	$\approx 2Q\left(\sqrt{\frac{E_s}{N_o}}\right)$	$\approx 2Q\left(\sqrt{\frac{2E_b}{N_o}}\right)$
DQPSK	$\approx Q\left(\sqrt{\frac{2E_s}{N_o}} \sin \frac{\pi}{4\sqrt{2}}\right)$	$\approx Q\left(\sqrt{\frac{4E_b}{N_o}} \sin \frac{\pi}{4\sqrt{2}}\right)$
MSK	$Q\left(\sqrt{\frac{2E_s}{N_o}}\right)$	$Q\left(\sqrt{\frac{2E_b}{N_o}}\right)$
16-QAM	$3Q\left(\sqrt{\frac{E_s}{5N_o}}\right)$	$\frac{3}{4}Q\left(\sqrt{\frac{4E_b}{5N_o}}\right)$
M-PSK ($M>4$)	$\approx 2Q\left(\sqrt{\frac{2E_s}{N_o}} \cdot \sin \frac{\pi}{M}\right)$	$\approx \frac{2}{m}Q\left(\sqrt{\frac{2mE_b}{N_o}} \cdot \sin \frac{\pi}{M}\right)$
DM-PSK ($M>4$)	$\approx 2Q\left(\sqrt{\frac{2E_s}{N_o}} \cdot \sin \frac{\pi}{\sqrt{2}M}\right)$	$\approx \frac{2}{m}Q\left(\sqrt{\frac{2mE_b}{N_o}} \cdot \sin \frac{\pi}{\sqrt{2}M}\right)$

Πίνακας 2 : Ρυθμοί μετάδοσης σφαλμάτων 2/2

2.6 Τεχνικές Πολλαπλής Πρόσβασης

Υπάρχουν τέσσερα βασικά συστήματα πολλαπλής πρόσβασης που εκχωρούν αποκλειστικούς δορυφορικούς διαύλους σε πολλαπλούς χρήστες ή αντίστοιχα που εκχωρούν αποκλειστικούς δορυφορικούς διαύλους από πολλαπλούς επίγειους σταθμούς σε έναν δορυφορικό αναμεταδότη:

1. η πολλαπλή πρόσβαση διαίρεσης συχνότητας FDMA (Frequency Division Multiple Access)
2. η πολλαπλή πρόσβαση διαίρεσης χρόνου TDMA (Time Division Multiple Access)
3. η πολλαπλή πρόσβαση διαίρεσης κώδικα CDMA (Code Division Multiple Access)
4. η πολλαπλή πρόσβαση διαίρεσης χώρου SDMA (Spatial Division Multiple Access)

Οι μέθοδοι αυτές καθορίζουν τον τρόπο πρόσβασης στο κοινό δορυφορικό κανάλι. Η απόκτηση πρόσβασης καθορίζεται, επίσης, και από δύο άλλες δευτερεύουσες μεθόδους, που συνήθως υλοποιούνται μαζί με μία ή περισσότερες από τις προαναφερθείσες τεχνικές πολλαπλής πρόσβασης, για να παρέχουν πρόσβαση στον αναμεταδότη. Οι δευτερεύουσες αυτές μέθοδοι περιλαμβάνουν:

1. τη σταθερή εκχώρηση (fixed assignment, FA) ή προ-εκχώρηση (pre-assigned, PA) από την οποία προκύπτει η αντίστοιχη προ-εκχωρημένη τεχνική πολλαπλής πρόσβασης PAMA (Pre-Assigned Multiple Access)
2. τη δυναμική εκχώρηση (dynamic assignment, DA) από την οποία προκύπτει η αντίστοιχη κατ' αίτηση εκχώρηση τεχνική πολλαπλής πρόσβασης DAMA (Demand Assigned Multiple Access).

Στην περίπτωση της προ-εκχωρημένης πολλαπλής πρόσβασης PAMA, ο αναμεταδότης εκχωρείται στον μεμονωμένο χρήστη, είτε μόνιμα για ολόκληρη τη διάρκεια ζωής του δορυφόρου ή τουλάχιστον για μεγάλες περιόδους. Η προ-εκχώρηση μπορεί να αφορά μια ορισμένη ζώνη συχνοτήτων, χρονοσχισμή ή κώδικα. Όταν, όμως, σπάνια χρησιμοποιείται μία ζεύξη με προ-εκχωρημένα κανάλια, αυτή η ζεύξη δεν είναι μόνο δαπανηρή για τον χρήστη, αλλά και μη αποδοτική.

Η μέθοδος πολλαπλής πρόσβασης με κατ' αίτηση εκχώρηση DAMA επιτρέπει σε πολλαπλούς χρήστες να μοιραστούν μία κοινή ζεύξη, όπου κάθε χρήστης χρειάζεται μόνο να θέσει ένα αίτημα προς τον σταθμό ελέγχου, όταν απαιτεί τη σύνδεση που θα χρησιμοποιηθεί. Η ζεύξη του καναλιού ολοκληρώνεται μόνο όπως αιτήθηκε ο χρήστης και η συχνότητα του καναλιού εκχωρείται από τις διαθέσιμες συχνότητες εντός της ζώνης συχνοτήτων του αναμεταδότη. Η μέθοδος είναι πολύ αποδοτική για εκείνους τους χρήστες, που πρέπει να πληρώσουν για τη χρήση της χωρητικότητας του αναμεταδότη μόνο για το χρονικό διάστημα που πραγματικά χρησιμοποιήθηκε.[2]

2.7 Πολυπλεξία

Η έννοια της πολυπλεξίας μπορεί να εφαρμοστεί τόσο στο πεδίο συχνότητας όσο και στο πεδίο του χρόνου. Είναι μια μέθοδος στην οποία πολλαπλά αναλογικά ή ψηφιακά σήματα συνδυάζονται σε ένα σήμα σε ένα shared medium (κοινόχρηστο μέσο: κανάλι μεταφοράς πληροφορίας που εξυπηρετεί παραπάνω από έναν χρήστες ταυτόχρονα). Ο στόχος είναι να μοιραστεί ένας πόρος σε ανεπάρκεια. Για παράδειγμα, στις τηλεπικοινωνίες, πολλές τηλεφωνικές κλήσεις μπορούν να πραγματοποιούνται χρησιμοποιώντας ένα καλώδιο.

Το σήμα πολυπλεξίας μεταδίδεται μέσω ενός καναλιού επικοινωνίας όπως ένα καλώδιο. Η πολυπλεξία διαιρεί την χωρητικότητα του καναλιού επικοινωνίας σε διάφορα λογικά κανάλια, ένα για κάθε σήμα μηνύματος ή ροή δεδομένων προς μεταφορά. Μια αντίστροφη διαδικασία, γνωστή ως αποπολυπλεξία, εξάγει τα αρχικά κανάλια στο άκρο του δέκτη.

Η συσκευή που εκτελεί την πολυπλεξία ονομάζεται πολυπλέκτης (MUX) και αυτή που εκτελεί την αντίστροφη διαδικασία λέγεται αποπολυπλέκτης (DEMUX ή IMUX).

Η αντίστροφη πολυπλεξία (IMUX) έχει τον αντίθετο στόχο ως πολυπλεξία, δηλαδή τη διάσπαση μιας ροής δεδομένων σε πολλές ροές, τη μεταφορά τους ταυτόχρονα σε πολλά κανάλια επικοινωνίας και την αναδημιουργία της αρχικής ροής δεδομένων.[7]

2.7.1 Πολυπλεξία με Διαίρεση Χρόνου (TDM)

Η TDM είναι ένα είδος ψηφιακής ή (σπανίως) αναλογικής πολυπλεξίας στην οποία δύο ή περισσότερα σήματα ή ψηφία μεταδίδονται ταυτόχρονα ως υπό-κανάλια σε ένα κανάλι επικοινωνίας. Το πεδίο του χρόνου χωρίζεται σε πολλά επαναλαμβανόμενα timeslots σταθερού μήκους, ένα για κάθε υπό-κανάλι. Το δείγμα bytes ή το data block του υπό-καναλιού 1 εκπέμπεται κατά το timeslot 1, του υπό-καναλιού 2 κατά το timeslot 2, κλπ. Κάθε σήμα, λοιπόν, καταλαμβάνει το διάυλο για κλάσμα t/T του συνολικού χρόνου ενώ είναι πιθανόν ο διάυλος να παραμένει αδρανής για κάποιο χρονικό ποσοστό.

Κατά την πολυπλεξία TDM σημάτων λαμβάνονται ανά ορισμένα χρονικά διαστήματα διάρκειας T δείγματα διάρκειας t και μεταδίδονται μέσω του διαύλου με συγκεκριμένη χρονική αλληλουχία. Έτσι καταλαμβάνουν διαδοχικές μη επικαλυπτόμενες χρονικές θυρίδες διάρκειας t σε επαναλαμβανόμενα χρονικά πλαίσια διάρκειας T . Το κάθε σήμα καταλαμβάνει το διάυλο για κλάσμα t/T του συνολικού χρόνου, ενώ, ενδεχομένως, για κάποιο ποσοστό του χρόνου ο διάυλος δε χρησιμοποιείται. Όταν τα σήματα που πολυπλέκονται είναι εξ αρχής ψηφιακά, γίνεται πολυπλεξία των ψηφίων τους με κατάλληλο τρόπο, ο οποίος κατά κύριο λόγο λαμβάνει υπόψη τις ενδεχόμενες διαφορές στο ρυθμό μετάδοσης των πηγών πληροφορίας. Ένα σημαντικό πρόβλημα που προκύπτει είναι η ανάγκη για ακριβή συγχρονισμό μεταξύ πομπού και δέκτη, ώστε τα διαδοχικά δείγματα που λαμβάνονται από το δέκτη να διοχετεύονται στο κατάλληλο προορισμό. Ο συγχρονισμός επιτυγχάνεται με χρήση πολύπλοκων ψηφιακών κυκλωμάτων. Επίσης, ένα άλλο πρόβλημα που αντιμετωπίζει η πολυπλεξία TDM είναι η διασυμβολική παρεμβολή (ISI) που είναι ενδεχόμενο να δημιουργηθεί λόγω χρονικής επικάλυψης των δειγμάτων. Αυτή αντιμετωπίζεται με κατάλληλη σχεδίαση των φίλτρων που χρησιμοποιούνται.[16]

2.7.2 Πολυπλεξία με Διαίρεση Συχνότητας (FDM)

Στην περίπτωση της πολυπλεξίας με διαίρεση συχνότητας (Frequency Division Multiplexing) κάθε ένα από τα μεταδιδόμενα σήματα μετατοπίζεται φασματικά καθώς η φέρουσα συχνότητά του επιλέγεται έτσι, ώστε να καταλάβει μια ζώνη μεταξύ διαδοχικών και μη επικαλυπτόμενων ζωνών του διατιθέμενου φάσματος . Από το άθροισμα των επιμέρους σημάτων προκύπτει ένα σύνθετο σήμα με πολύ μεγαλύτερο εύρος ζώνης, το οποίο διαμορφώνεται με κάποιον από τους συνήθεις τρόπους, αφήνοντας όμως ενδεχομένως κάποιες διαθέσιμες ζώνες συχνοτήτων αχρησιμοποίητες. Χαρακτηριστικό απλό παράδειγμα μετάδοσης πολυπλεγμένου FDM σήματος είναι η μετάδοση τηλεφωνικών σημάτων μέσω ομοαξονικού καλωδίου ή ασύρματης μικροκυματικής ζεύξης. Το σήμα αυτό περιλαμβάνει πολλά απλά τηλεφωνικά σήματα πολυπλεγμένα κατά συχνότητα. Επίσης, και το σήμα της τηλεόρασης αποτελεί ειδική περίπτωση πολυπλεξίας FDM δυο σημάτων, του ακουστικού σήματος διαμορφωμένου κατά FM και του σήματος εικόνας διαμορφωμένου κατά AM-VSB (Amplitude Modulation Vestigial Sideband). Η δημιουργία σύνθετων μικροκυματικών σημάτων, όπως για παράδειγμα τα σήματα των

δορυφορικών επικοινωνιών, είναι και αυτή διαδικασία πολυπλεξίας FDM. Σε αρκετές περιπτώσεις δορυφορικών ζεύξεων, διαφορετικής προέλευσης σύνθετα σήματα, όπως πολυάριθμα τηλεφωνικά σήματα που έχουν ήδη πολυπλεχθεί, ή ένα ή περισσότερα τηλεοπτικά σήματα και μεγάλος όγκος ψηφιακών δεδομένων δημιουργούν, πολυπλεκόμενα κατά FDM, το συνολικό δορυφορικό σήμα.

Το βασικό πρόβλημα που αντιμετωπίζει η πολυπλεξία FDM είναι η έκθεση σε διαλείψεις επιλεκτικές ως προς τη συχνότητα καθώς και η ανάγκη χρησιμοποίησης ενισχυτών με μεγάλο εύρος ζώνης λειτουργίας λόγω αυτής ακριβώς της μεγάλης φασματικής διεύρυνσης.[20]

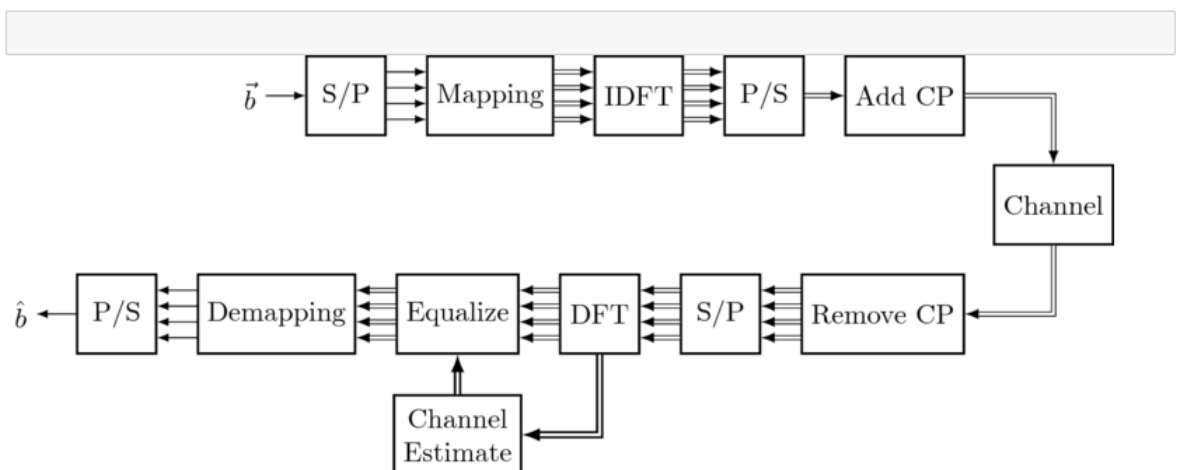
2.8 Rayleigh Fading Channel

Όπως έχει προαναφερθεί η διαδρομή που ακολουθεί ένα σήμα για να φτάσει από το πομπό στο δέκτη χαρακτηρίζεται από πολλά εμπόδια με αποτέλεσμα να υπάρχει μια σειρά ανακλάσεων, περιθλάσεων και σκεδάσεων. Σε ένα εξωτερικό χώρο υπάρχουν κτίρια, δέντρα, οχήματα ενώ σε ένα εσωτερικό χώρο υπάρχουν έπιπλα, συσκευές μεγάλου όγκου, τοίχοι κτλ. Συνεπώς το σήμα που καταφθάνει στο δέκτη είναι επηρεασμένο και συχνά εξασθενημένο. Παρόλο που το κανάλι στο σύνολο του είναι επιλεκτικής συχνότητας, το κανάλι που «αντιμετωπίζει» κάθε subcarrier ενός OFDM συμβόλου χαρακτηρίζεται από επίπεδη σκίαση με το κάθε subcarrier να αντιμετωπίζει μια Rayleigh σκίαση. Σε αυτή τη περίπτωση το μήκος κάθε διαδρομής είναι διαφορετικό. Άλλο μήκος για το απευθείας κύμα, άλλο για τα ανακλώμενα κύματα, άλλο για τα περιθλώμενα, άλλο για τα σκεδασμένα και άλλο για αυτά που έχουν υποστεί μια σειρά σκεδάσεων. Μαζί με το μήκος διαδρομής αλλάζει και ο χρόνος άφιξης κάθε κύματος. Στο δέκτη λοιπόν φτάνουν κύματα με διαφορετική φάση και διαφορετικό χρόνο άφιξης ο οποίος θα πρέπει φυσικά να είναι μικρότερος της εξάπλωσης καθυστέρησης. Αφού αφιχθούν τα κύματα στο δέκτη, πιθανόν να έχουμε ενίσχυση του σήματος ή αποδυνάμωση. Στην αποδυνάμωση που οφείλεται από αυτές τις πολλαπλές διαδρομές αυξάνεται, όπως είναι λογικό, το ποσοστό των λαθών. Το Rayleigh Fading είναι ένα στατιστικό μοντέλο για τη μελέτη της επίδρασης του περιβάλλοντος πολλαπλών διαδρομών σε σήματα ασύρματων μεταδόσεων.

3 Εισαγωγή στο OFDM

Χρησιμοποιείται στις ασύρματες επικοινωνίες και είναι η πιο κοινή μέθοδος διαμόρφωσης. Το OFDM ανήκει στην κατηγορία των συστημάτων διαμόρφωσης πολλαπλών φερόντων. Το OFDM αποσυνθέτει τη ζώνη συχνότητας μετάδοσης σε μια ομάδα στενότερων συνεχόμενων υποζωνών (φέροντα) και κάθε φέρων διαμορφώνεται ξεχωριστά. Μπορεί να εφαρμοστεί αυτός ο τύπος διαμόρφωσης με έναν αντίστροφο γρήγορο μετασχηματισμό Fourier (IFFT). Με τη χρήση στενών ορθογώνιων υποφερόντων, το σήμα OFDM αποκτά αντοχή έναντι ενός καναλιού εξασθένισης επιλεκτικής συχνότητας.

Στο άκρο λήψης, το σήμα OFDM μπορεί να αποδιαμορφωθεί με έναν γρήγορο μετασχηματισμό Fourier (FFT) και να εξισωθεί με ένα πολύπλοκο κέρδος σε κάθε υποφέρων. Ο συνδυασμός OFDM με MIMO μπορεί να βελτιώσει την ταχύτητα επικοινωνίας χωρίς να αυξήσει τη ζώνη συχνότητας.



Εικόνα 3 : Σύστημα OFDM

3.1 Μετάδοση απλού φέροντος έναντι πολλαπλών φερόντων

3.1.1 Μετάδοση απλού φέροντος

3.1.1.1 Μετάδοση απλού φέροντος ζώνης βάσης: Μοντέλο Συστήματος

Θεωρούμε ένα κανάλι περιορισμένης ζώνης $h(t)$ με εύρος ζώνης W . Τα σύμβολα μετάδοσης $a\{n\}$, το καθένα με περίοδο συμβόλων T δευτερολέπτων, δηλαδή, ρυθμός

δεδομένων $R = 1 / T$, έχουν σχήμα παλμού από ένα φίλτρο μετάδοσης $g_T(t)$ στον πομπό. Μετά τη λήψη τους μέσω του καναλιού, επεξεργάζονται με το φίλτρο λήψης, τον εξισορροπητή και τον ανιχνευτή στον δέκτη. Τα $g_T(t)$, $g_R(t)$, και $h^{-1}(t)$ δηλώνουν την απόκριση σε κρουστικό παλμό του φίλτρου μετάδοσης, του φίλτρου λήψης και του εξισορροπητή, αντίστοιχα. Η έξοδος του εξισορροπητή είναι η

$$y(t) = \sum_{m=-\infty}^{\infty} a_m g(t - mT) + z(t) \quad (3)$$

Όπου το $z(t)$ είναι προσθετικός θόρυβος και το $g(t)$ είναι η κρουστική απόκριση του συνολικού συστήματος από άκρο σε άκρο που δίνεται ως

$$g(t) = g_T(t) * h(t) * g_R(t) * h^{-1}(t) \quad (4)$$

Ο εξισορροπητής έχει σχεδιαστεί για να αντισταθμίζει το αποτέλεσμα του καναλιού. Εδώ, απλώς υποθέτουμε ότι το αποτέλεσμα του καναλιού αντισταθμίζεται τέλεια από τον εξισωτή σύμφωνα με την εξίσωση 5. Επομένως, η συνολική κρουστική απόκριση υπόκειται μόνο στη μετάδοση και τη λήψη φίλτρων. Όταν αγνοείται ο όρος θορύβου, το σήμα εξόδου του δείγματος του εξισορροπητή μπορεί να εκφραστεί ως

$$y(t_n) = \sum_{m=-\infty}^{\infty} a_m g((n-m)T) \quad t_n = nT \quad (5)$$

Απομονώνοντας το n -οστο δείγμα a_n , η εξίσωση 7 μπορεί να γραφεί ως

$$y(t_n) = a_n g(0) + \sum_{m=-\infty}^{\infty} a_m g(n-m)T \quad (6)$$

Παρατηρείται ότι το $g(t)$ δεν είναι περιορισμένου χρόνου λόγω του πεπερασμένου εύρους του καναλιού. Σε περίπτωση που το $g((n-m)T) \neq 0$ για $\forall m \neq n$, ο δεύτερος όρος παραμένει ως ISI στο a_n . Στην πραγματικότητα, το ISI προκαλείται από ένα ίχνος της συνολικής παλμικής απόκρισης, η οποία θα μπορούσε να υποβαθμίσει την απόδοση ενός ψηφιακού συστήματος επικοινωνίας. Επομένως, το φίλτρο μετάδοσης και το φίλτρο λήψης πρέπει να σχεδιάζονται σκόπιμα έτσι ώστε να ελαχιστοποιείται ή να εξαλείφεται πλήρως το ISI σε ένα πρακτικό σύστημα. Η έκταση του ISI εξαρτάται από τη διάρκεια μιας περιόδου συμβόλων T : όσο μικρότερη είναι η περίοδος συμβόλων, τόσο μεγαλύτερη είναι η επιρροή του ISI. Αυτό υποδηλώνει ότι αν δεν ισχύει το $g((n-m)T) \neq 0$ για $\forall m \neq n$, το ISI γίνεται σημαντικό όσο ο ρυθμός μετάδοσης δεδομένων αυξάνεται σε ένα σύστημα ενός φέροντος.

3.1.1.2 Κριτήριο ISI και Nyquist

Η ISI μπορεί να εξαλειφθεί πλήρως ικανοποιώντας την παρακάτω συνθήκη στο πεδίο του χρόνου στην συνολική κρουστική απόκριση:

$$g(nT) = \delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \quad (7)$$

Ισοδύναμα στο πεδίο συχνότητας:

$$\sum_{i=-\infty}^{\infty} G\left(f - \frac{i}{T}\right) = T \quad (8)$$

Όπου το $G(f)$ είναι ο μετασχηματισμός Fourier του $g(t)$, που αναπαριστά την συνολική απόκριση συχνότητας.

Αυτό που φαίνεται στις εξισώσεις 9 ή 10 είναι το Nyquist κριτήριο που εξασφαλίζει επικοινωνία χωρίς ISI ακόμα και με μικρή περίοδο συμβόλου T για υψηλό ρυθμό μετάδοσης σε ένα σύστημα μετάδοσης ενός φέροντος. Τα φίλτρα που ικανοποιούν το κριτήριο Nyquist αναφέρονται ως Nyquist φίλτρα (ILPF).

3.1.1.3 Περιορισμός της μετάδοσης απλού φέροντος για υψηλό ρυθμό δεδομένων

Για να υποστηριχθεί ο ρυθμός των R_s συμβόλων το δευτερόλεπτο, το ελάχιστο εύρος που απαιτείται είναι το Nyquist, το οποίο δίνεται από το $R_s/2$ Hz.

Αυτό συνεπάγεται ότι απαιτείται μεγαλύτερο εύρος ζώνης για την υποστήριξη υψηλότερου ρυθμού δεδομένων σε μια μετάδοση ενός φορέα. Μέχρι στιγμής, θεωρείται ότι το κανάλι αντισταθμίζεται τέλεια από τον εξισορροπητή. Ωστόσο, καθώς ο ρυθμός συμβολισμού αυξάνεται, το εύρος ζώνης σήματος γίνεται μεγαλύτερο. Όταν το εύρος ζώνης σήματος γίνεται μεγαλύτερο από το εύρος ζώνης συνοχής στο ασύρματο κανάλι, ο σύνδεσμος υποφέρει από εξασθένιση πολλαπλών διαδρομών, που προκαλεί παρεμβολές μεταξύ συμβόλων (ISI). Σε γενικές γραμμές, οι προσαρμοστικοί εξισορροπητές απασχολούνται για να αντιμετωπίσουν το ISI που προκύπτει από το χρονικά μεταβαλλόμενο κανάλι πολλαπλών διαδρομών. Επιπλέον, η πολυπλοκότητα ενός εξισορροπητή αυξάνεται με τον ρυθμό δεδομένων. Πιο συγκεκριμένα, οι προσαρμοστικοί εξισορροπητές εφαρμόζονται με φίλτρα πεπερασμένης απόκρισης παλμών (FIR) με τους συντελεστές προσαρμοστικού χαρτιού που προσαρμόζονται έτσι ώστε να ελαχιστοποιείται η επίδραση του ISI. Στην πραγματικότητα, απαιτείται περισσότερη τάση εξισορροπητή καθώς το ISI γίνεται σημαντικό, για παράδειγμα, όταν αυξάνεται ο ρυθμός δεδομένων. Ο βέλτιστος ανιχνευτής για το κανάλι διαλείψεων πολλαπλών διαδρομών είναι ένας τομέας αλληλουχίας μέγιστης πιθανότητας (MLSD), ο οποίος βασίζει τις αποφάσεις του στην παρατήρηση μιας ακολουθίας λαμβανόμενων συμβόλων σε διαδοχικά διαστήματα συμβόλων, υπέρ της μεγιστοποίησης της εκ των υστέρων πιθανότητας. Ας σημειωθεί ότι η πολυπλοκότητά του εξαρτάται από τη σειρά διαμόρφωσης και τον αριθμό των πολλαπλών διαδρομών. Τα M και L δηλώνουν τον αριθμό των πιθανών σημείων σήματος για κάθε σύμβολο διαμόρφωσης και το εύρος του ISI που προκύπτουν πάνω από το κανάλι διαλείψεως πολλαπλών διαδρομών, αντίστοιχα. Λόγω της μνήμης μήκους L για την έκταση του ISI, πρέπει να

αξιολογηθούν οι αντίστοιχες μετρήσεις διαδρομής της Ευκλείδιας απόστασης M^L για να επιλεγεί η καλύτερη ακολουθία στο MLSD. Όταν επιδιώκεται μια πιο αποτελεσματική μετάδοση αυξάνοντας το M και εφαρμόζεται υψηλός ρυθμός δεδομένων, η πολυπλοκότητα του βέλτιστου εξισορροπητή καθίσταται απαγορευτική, για παράδειγμα, $M^L = 64^{16}$ για $L \approx 16$ με 64-QAM στο ρυθμό δεδομένων 10Mbps μέσω του καναλιού πολλαπλών διαδρομών με καθυστέρηση διάδοσης 10ms. Όταν τα M και L είναι πολύ μεγάλα, μπορούν να χρησιμοποιηθούν και άλλοι πιο πρακτικοί αλλά μη βέλτιστοι εξισορροπητές, όπως MMSE ή LS. Ωστόσο, η πολυπλοκότητα αυτών των υποβέλτιστων εξισορροπητών εξακολουθεί να είναι πολύ τεράστια για να εφαρμοστεί καθώς το ISI αυξάνεται με τον ρυθμό δεδομένων. Αυτή η συγκεκριμένη κατάσταση μπορεί να εξηγηθεί από το γεγονός ότι η αντίστροφη συνάρτηση (μια απόκριση πεδίου συχνότητας εξισορροπητή) γίνεται πιο έντονη καθώς αυξάνεται η επιλεκτικότητα συχνότητας του καναλιού. Συμπερασματικά, η μετάδοση ενός φέροντος με υψηλό ρυθμό δεδομένων ενδέχεται να μην είναι εφικτή λόγω της υπερβολικής πολυπλοκότητας του εξισορροπητή στον δέκτη.

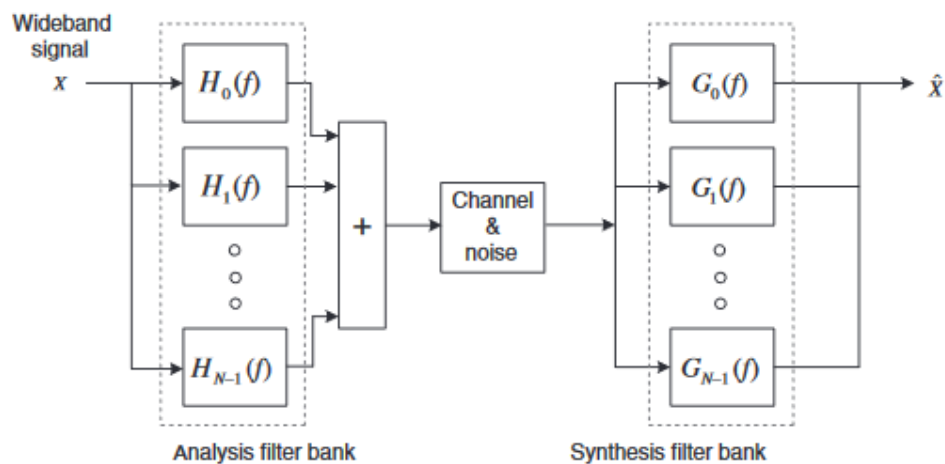
3.1.2 Μετάδοση Πολλαπλών φερόντων

3.1.2.1 Βασική δομή ενός συστήματος μεταφοράς πολλαπλών φερόντων

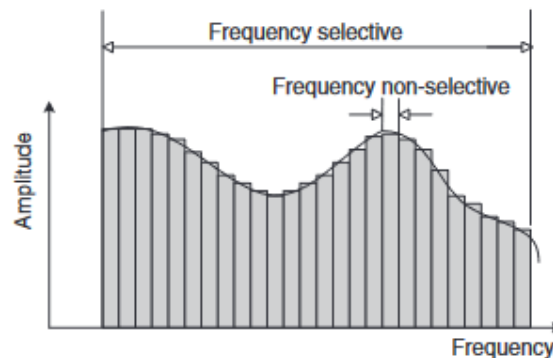
Για να ξεπεραστεί η επιλεκτικότητα συχνότητας του καναλιού ευρείας ζώνης που βιώθηκε από την μετάδοση ενός φορέα, πολλαπλά φέροντα μπορούν να χρησιμοποιηθούν για υψηλού ρυθμού μετάδοση δεδομένων. Η εικόνα 5(a) δείχνει τη βασική δομή και την έννοια ενός συστήματος μετάδοσης πολλαπλών φερόντων. Εδώ, ένα ευρείας ζώνης σήμα αναλύεται (μέσω φίλτρου στενής ζώνης πολλαπλών $H_k(f)$'s) σε μερικά σήματα στενής ζώνης στον πομπό και συντίθεται (μέσω φίλτρου στενής ζώνης πολλαπλών $G_k(f)$'s, το κάθε ένα αντιστοιχισμένο σε $H_k(f)$) στον δέκτη έτσι ώστε το επιλεκτικής συχνότητας κανάλι ευρείας ζώνης να μπορεί να προσεγγιστεί από πολλαπλά επίπεδης συχνότητας κανάλια στενής ζώνης όπως απεικονίζονται στην εικόνα 5(b). Ας σημειωθεί ότι η συχνότητα μη επιλεκτικότητας των καναλιών στενής ζώνης μειώνει την πολυπλοκότητα του εξισορροπητή για κάθε υποκανάλι. Όσο διατηρείται η ορθογωνικότητα μεταξύ αυτών των υποκαναλιών, το ICI (παρεμβολή μεταξύ φερόντων) μπορεί να κατασταλεί, οδηγώντας σε μετάδοση χωρίς παρεμβολή.

Σε αυτό το σύστημα πολλαπλών καναλιών, ας χωρίσουμε τα ευρείας ζώνης σε N υποκανάλια στενής ζώνης, τα οποία έχουν συχνότητα υποφέροντος f_k , $k=0,1,2,\dots,N-1$. Η εικόνα 6(a) δείχνει την βασική δομή ενός σχεδίου επικοινωνίας πολλαπλού φέροντος, το οποίο είναι μια ειδική μορφή του συστήματος πολυκαναλιού, όπου τα διαφορετικά σύμβολα μεταδίδονται με ορθογωνικά υποκανάλια σε παράλληλη μορφή. Αυτό συνεπάγεται ότι η μετάδοση πολλαπλών φερόντων μπορεί να θεωρηθεί ως ένα είδος μεθόδου FDMA (πολλαπλή πρόσβαση διαίρεσης συχνότητας). Το Σχήμα 6(b) απεικονίζει ένα φάσμα μεταδιδόμενου σήματος στο σύστημα μετάδοσης πολλαπλών φερόντων, το οποίο καλύπτει πολλαπλές υποζώνες ίσου εύρους ζώνης, η καθεμία στο κέντρο της διαφορετικής συχνότητας φέροντος. Εάν κάθε υποκανάλι είναι απεριόριστο

όπως απεικονίζεται στο Σχήμα 6(b), γίνεται Μετάδοση FMT (Filtered Multi-Tone). Ενώ ένας τύπος συστήματος μετάδοσης πολλαπλών φερόντων τύπου FMT μπορεί να αντιμετωπίσει την επιλεκτικότητα συχνότητας ενός καναλιού ευρείας ζώνης, η εφαρμογή του καθίσταται περίπλοκη καθώς περιλαμβάνει περισσότερους κωδικοποιητές / αποκωδικοποιητές και ταλαντωτές, και φίλτρα υψηλότερης ποιότητας καθώς αυξάνεται ο αριθμός των υποφερόντων.

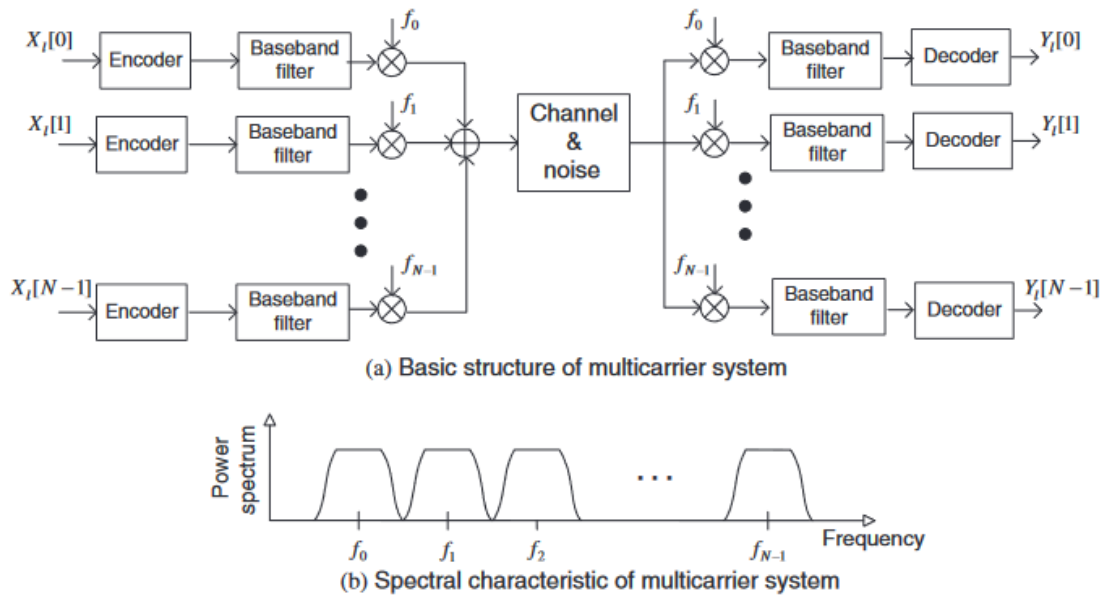


(a) The structure of multichannel transmission system



(b) The frequency response of multichannel transmission system

Εικόνα 4 : Χαρακτηριστικά δομής και συχνότητας ενός πολυκαναλικού συστήματος μετάδοσης

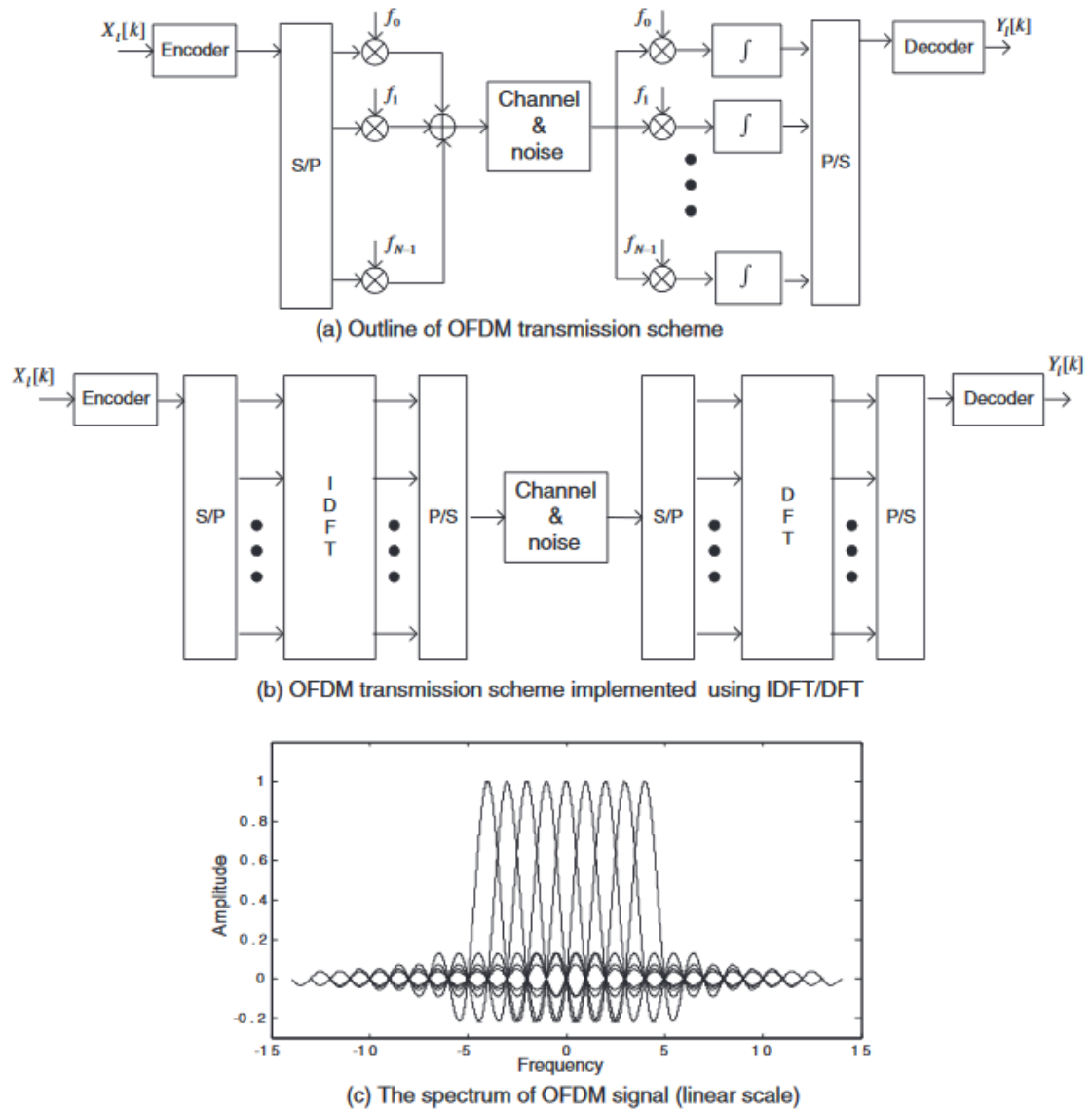


Εικόνα 5 : Χαρακτηριστικά δομής και φάσματος ενός πολυκαναλικού συστήματος μετάδοσης

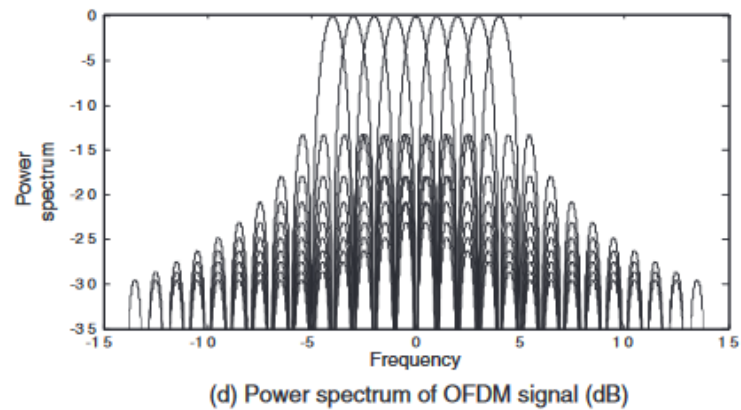
3.1.2.2 Σχήμα Μετάδοσης OFDM

Το σχήμα μετάδοσης πολυπλεξίας ορθογώνιας διαίρεσης συχνότητας είναι ένας άλλος τύπος συστήματος πολλαπλών καναλιών, το οποίο είναι παρόμοιο με το σχήμα μετάδοσης FMT με την έννοια ότι το στοιχείο χρησιμοποιεί πολλαπλά υποφέροντα. Όπως φαίνεται στο Σχήμα 6(a), δεν χρησιμοποιεί μεμονωμένα φίλτρα ζώνης και ταλαντωτές για κάθε υπο-κανάλι και επιπλέον, τα φάσματα των υπο-φερόντων αλληλεπικαλύπτονται για απόδοση εύρους ζώνης, σε αντίθεση με το σχήμα FMT όπου η ευρυζωνική ζώνη χωρίζεται πλήρως σε στενοζωνικά δευτερεύοντα κανάλια. Τα πολλαπλά ορθογώνια σήματα υποφερόντων, τα οποία επικαλύπτονται στο φάσμα, μπορούν να παραχθούν γενικεύοντας το κριτήριο Nyquist απλού φέροντος στην Εξίσωση 10 στο κριτήριο πολλαπλών φερόντων. Στην πράξη, οι διαδικασίες διακριτού μετασχηματισμού Fourier (DFT) και αντίστροφης DFT (IDFT) είναι χρήσιμες για την εφαρμογή αυτών των ορθογώνιων σημάτων. Ας σημειωθεί ότι το DFT και το IDFT μπορούν να εφαρμοστούν αποτελεσματικά χρησιμοποιώντας το γρήγορο μετασχηματισμό Fourier (FFT) και το αντίστροφο γρήγορο μετασχηματισμό Fourier (IFFT), αντίστοιχα. Στο σύστημα μετάδοσης OFDM, το σημείο NF IFFT λαμβάνεται για τα μεταδιδόμενα σύμβολα $\{X_I[k]\}_{k=0}^{N-1}$, έτσι ώστε να δημιουργηθούν $\{x[n]\}_{n=0}^{N-1}$, τα δείγματα για το άθροισμα των N ορθογώνιων σημάτων υποφερόντων. Το $y[n]$ δηλώνει το ληφθέν δείγμα που αντιστοιχεί στο $x[n]$ με προσθετικό θόρυβο $w[n]$ (i.e. $y[n] = x[n] + w[n]$). Λαμβάνοντας το N-point FFT των ληφθέντων δειγμάτων $\{y[n]\}_{n=0}^{N-1}$, η θορυβώδης μετατροπή των μεταδιδόμενων συμβόλων $\{Y_I[k]\}_{k=0}^{N-1}$ μπορεί να ληφθεί στον δέκτη. Το σχήμα 7 (c) δείχνει τη δομή μετάδοσης OFDM που υλοποιείται από IDFT / DFT. Τα εγγενή πλεονεκτήματα της μετάδοσης OFDM θα

αναλυθούν αργότερα. Δεδομένου ότι όλα τα υπο-φέροντα είναι της πεπερασμένης διάρκειας T , το φάσμα του σήματος OFDM μπορεί να θεωρηθεί ως το άθροισμα των συναρτήσεων μετατόπισης συχνότητας στην περιοχή συχνοτήτων όπως απεικονίζεται στο Σχήμα 7 (c), όπου οι αλληλεπικαλυπτόμενες γειτονικές συναρτήσεις χωρίζονται κατά $1/T$. Το διακριτό σχήμα πολλαπλών τόνων (DMT) που χρησιμοποιείται στο ADSL (Asymmetric Digital Subscriber Line) και το VDSL που βασίζεται σε φερμουάρ (Very high-rate Data Digital Subscriber Line) έχει επίσης την ίδια δομή με το OFDM. Δεδομένου ότι κάθε σήμα subcarrier έχει χρονοπεριορισμό για κάθε σύμβολο (δηλ. όχι περιορισμένη ζώνη), σε ένα σήμα OFDM μπορεί να προκύψει ακτινοβολία εκτός ζώνης, η οποία προκαλεί μη αμελητέα γειτονική παρεμβολή καναλιών (ACI). Φαίνεται σαφώς από το Σχήμα 8 (d) ότι ο πρώτος πλευρικός λοβός δεν είναι τόσο μικρός σε σύγκριση με τον κύριο λοβό στα φάσματα. Επομένως, το σχήμα OFDM τοποθετεί μια ζώνη φρουράς στους δρομολογητές, που ονομάζονται εικονικοί φορείς (VCs), γύρω από τη ζώνη συχνοτήτων για τη μείωση της ακτινοβολίας εκτός ζώνης. Το σχήμα OFDM εισάγει επίσης ένα διάστημα φύλαξης στον τομέα χρόνου, που ονομάζεται κυκλικό πρόθεμα (CP), το οποίο μετριάζει την ISI μεταξύ των OFDM συμβόλων.



Εικόνα 6 : Χαρακτηριστικά δομής και φάσματος ενός σχήματος μετάδοσης OFDM



Εικόνα 7 : Χαρακτηριστικά δομής και φάσματος ενός σχήματος μετάδοσης OFDM

3.1.3 Μετάδοση απλού φέροντος έναντι πολλαπλών φερόντων

Στις προηγούμενες υποενότητες, συζητήσαμε τα συστήματα μεταφοράς απλού φέροντος και πολλαπλών φερόντων. Είναι σαφές ότι κάθε ένα από αυτά τα συστήματα έχει τα δικά του πλεονεκτήματα και μειονεκτήματα. Το σύστημα απλού φέροντος μπορεί να μην είναι χρήσιμο για μια ασύρματη μετάδοση υψηλού ποσοστού, απλά επειδή απαιτεί έναν εξισορροπητή υψηλής πολυπλοκότητας για να αντιμετωπίσει το πρόβλημα παρεμβολής μεταξύ των συμβόλων στο κανάλι εξασθένησης πολλαπλών διαδρομών ή ισοδύναμα, επιλεκτικό κανάλι εξασθένησης συχνότητας. Κατά μέσο όρο, το σύστημα πολλαπλών φερόντων είναι χρήσιμο για μια ασύρματη μετάδοση υψηλής ταχύτητας, η οποία δεν επηρεάζει την πολυπλοκότητα της εξίσωσης καναλιών. [5]

3.2 Βασική αρχή του OFDM

3.2.1 Διαμόρφωση και Αποδιαμόρφωση OFDM

3.2.1.1 Ορθογωνικότητα

Εννοιολογικά, το OFDM είναι μια εξειδικευμένη μέθοδος πολυπλεξίας διαίρεσης συχνότητας (FDM), με τον πρόσθετο περιορισμό ότι όλα τα σήματα υποφέροντος εντός ενός καναλιού επικοινωνίας είναι ορθογώνια μεταξύ τους.

Θεωρούμε τα περιορισμένου χρόνου μιγαδικά εκθετικά σήματα $\{e^{j2\pi f_k t}\}_{k=0}^{N-1}$, τα οποία αναπαριστούν τα διαφορετικά υποφέροντα στο $f_k = k/T_{sym}$ στο OFDM σήμα, όπου $0 \leq t \leq T_{sym}$. Αυτά τα σήματα ορίζονται να είναι ορθογώνια αν το ολοκλήρωμα του προϊόντος για την κοινή τους περίοδο είναι 0, δηλαδή,

$$\begin{aligned} \frac{1}{T_{sym}} \int_0^{T_{sym}} e^{j2\pi f_k t} * e^{-j2\pi f_i t} dt &= \frac{1}{T_{sym}} \int_0^{T_{sym}} e^{j2\pi \frac{k}{T_{sym}} t} * e^{-j2\pi \frac{i}{T_{sym}} t} dt = \\ \frac{1}{T_{sym}} \int_0^{T_{sym}} e^{j2\pi \frac{(k-i)}{T_{sym}} t} dt &= \begin{cases} 1, & \forall \text{ integer } k=i \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (9)$$

Παίρνοντας, τα διακριτά δείγματα με τα στιγμιότυπα δειγματοληψίας στα $t=nT_s=nT_s/N$, $n=0,1,2,\dots,N-1$, η εξίσωση 11 μπορεί να γραφεί στο διακριτό πεδίο του χρόνου ως

$$\begin{aligned} \frac{1}{N} \sum_{n=0}^{N-1} e^{j2\pi \frac{k}{T_{sym}} * nT_s} * e^{-j2\pi \frac{i}{T} * nT_s} &= \frac{1}{N} \sum_{n=0}^{N-1} e^{j2\pi \frac{k}{T_{sym}} * \frac{nT}{N}} * e^{-j2\pi \frac{i}{T_{sym}} * \frac{nT_{sym}}{N}} = \\ \frac{1}{N} \sum_{n=0}^{N-1} e^{j2\pi \frac{(k-i)}{N} * n} &= \begin{cases} 1, & \forall \text{ ακέραιο } k=i \\ 0, & \text{αλλιώς} \end{cases} \end{aligned} \quad (10)$$

Η παραπάνω ορθογωνικότητα είναι απαραίτητη προϋπόθεση για το σήμα OFDM να είναι χωρίς ICI.[11]

3.2.1.2 Διαμόρφωση και Αποδιαμόρφωση OFDM

Ο OFDM πομπός χαρτογραφεί τα bits μηνύματος σε μια ακολουθία PSK ή QAM συμβόλων τα οποία θα μετατραπούν ακολούθως σε N παράλληλες ροές. Κάθε ένα από τα N σύμβολα από την σειριακή σε παράλληλη (S/P) μετατροπή διεξάγεται από το διαφορετικό υποφέρων. Το $X_l[k]$ δηλώνει το l th σύμβολο μετάδοσης στο k th υποφέρων, $l=0,1,2,\dots,\infty$, $k=0,1,2,\dots,N-1$. Λόγω της S/P μετατροπής, η διάρκεια του χρόνου μετάδοσης για N σύμβολα επεκτείνεται κατά NT_s , το οποίο σχηματίζει ένα μόνο OFDM σύμβολο με μήκος T_{sym} (i.e. $T_{sym} = NT_s$). Το $\Psi_{l,k}$ δηλώνει το l th OFDM σήμα στο k th υποφέρων, το οποίο δίνεται ως

$$\Psi_{l,k} = \begin{cases} e^{j2\pi f_k(t-lT_{sym})}, & 0 < t \leq T_{sym} \\ 0, & \text{αλλού} \end{cases} \quad (11)$$

Τότε τα bandpass και baseband OFDM σήματα στο πεδίο συνεχούς χρόνου μπορούν να εκφραστούν αντίστοιχα ως

$$x_l(t) = \text{Re} \left\{ \frac{1}{T_{sym}} \sum_{l=0}^{\infty} \left\{ \sum_{k=0}^{N-1} x_l[k] \Psi_{l,k}(t) \right\} \right\} \quad (12)$$

Και

$$x_l(t) = \sum_{l=0}^{\infty} \sum_{k=0}^{N-1} x_l[k] e^{j2\pi f_k(t-lT_{sym})} \quad (13)$$

Το συνεχούς χρόνου OFDM σήμα βασικής ζώνης στην εξίσωση 14 μπορεί να δειγματοληπτηθεί κατά $t = lT_{sym} + nT_s$ με $T_s = T_{sym}/N$ και $f_k = k/T_{sym}$ για να αποδώσει το αντίστοιχο διακριτού χρόνου OFDM σύμβολο ως

$$x_l[n] = \sum_{k=0}^{N-1} x_l[k] e^{\frac{j2\pi kn}{N}} \quad \text{για } n = 0, 1, \dots, N-1 \quad (14)$$

Ας σημειώσουμε ότι η εξίσωση 16 φαίνεται ότι είναι το N -σημείο IDFT των PSK ή των QAM δεδομένων συμβόλων $X_l[k]_{k=0}^{N-1}$, και μπορεί να υπολογιστεί αποδοτικά χρησιμοποιώντας τον IFFT (Inverse Fast Fourier Transform) αλγόριθμο.

Θεωρούμε το ληφθέν OFDM σύμβολο βασικής ζώνης

$$y_l(t) = \sum_{k=0}^{N-1} X_l[k] e^{j2\pi f_k(t-lT_{sym})} \quad (15)$$

, $lT_{sym} < t \leq lT_{sym} + nT_s$, από το οποίο το μεταδιδόμενο σύμβολο $X_l[k]$ μπορεί να ανακατασκευαστεί από την ορθογωνικότητα μεταξύ των υποφερόντων στην εξίσωση 11 ως :

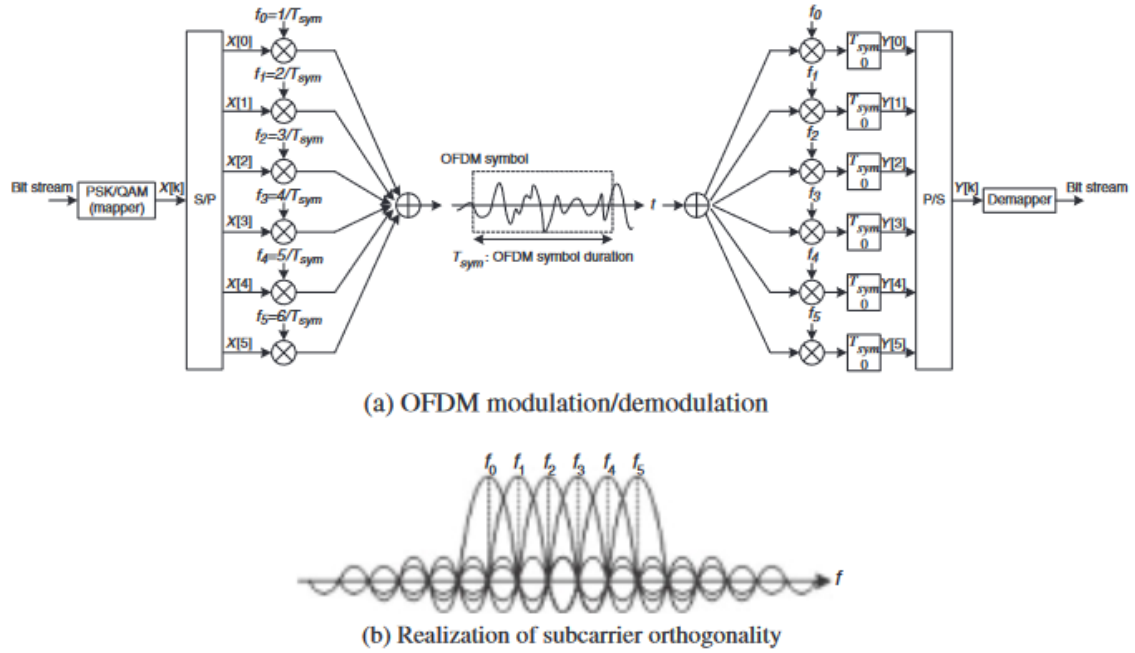
$$\begin{aligned} Y_l[k] &= \frac{1}{T_{sym}} \int_{-\infty}^{\infty} y_l(t) e^{j2\pi f_k(t-lT_{sym})} dt = \\ &= \frac{1}{T_{sym}} \int_{-\infty}^{\infty} \left\{ \sum_{i=0}^{N-1} X_l[i] e^{j2\pi f_i(t-lT_{sym})} \right\} e^{j2\pi f_k(t-lT_{sym})} dt = \\ &= \sum_{i=0}^{N-1} X_l[i] \left\{ \frac{1}{T_{sym}} \int_0^{T_{sym}} e^{j2\pi(f_i-f_k)(t-lT_{sym})} dt \right\} = X_l[k] \end{aligned} \quad (16)$$

όπου τα αποτελέσματα του καναλιού και του θορύβου δεν λαμβάνονται υπόψιν. Το $\{y_l[n]\}_{n=0}^{N-1}$ είναι οι τιμές δειγμάτων του ληφθέν OFDM συμβόλου $y_l(t)$ στο $t = lT_{sym} + nT_s$. Τότε, το ολοκλήρωμα στην διαδικασία διαμόρφωσης της εξίσωσης 18 μπορεί να αναπαρασταθεί στο διακριτό χρόνο ως:

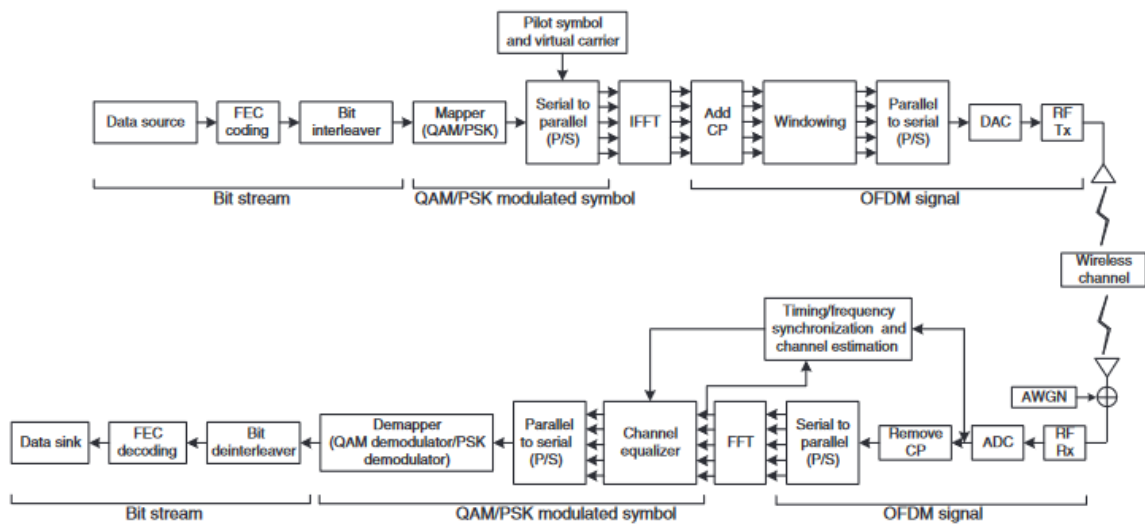
$$\begin{aligned} Y_l[k] &= \sum_{n=0}^{N-1} y_l[n] e^{-\frac{j2\pi kn}{N}} = \sum_{n=0}^{N-1} \left\{ \frac{1}{N} \sum_{i=0}^{N-1} X_l[i] e^{\frac{j2\pi in}{N}} \right\} e^{-\frac{j2\pi kn}{N}} = \\ &= \frac{1}{N} \sum_{n=0}^{N-1} \sum_{i=0}^{N-1} X_l[i] e^{\frac{j2\pi(i-k)n}{N}} = X_l[k] \end{aligned} \quad (17)$$

Στην πραγματικότητα, η εξίσωση 19 είναι το DFT N-σημείο του $\{y_l[n]\}_{n=0}^{N-1}$ και μπορεί να υπολογιστεί αποδοτικά χρησιμοποιώντας τον FFT (Fast Fourier Transform) αλγόριθμο.

Σύμφωνα με τα παραπάνω, η διαμόρφωση και η αποδιαμόρφωση OFDM μπορούν να απεικονιστούν από το διάγραμμα μπλοκ στην Εικόνα 9, το οποίο δείχνει ότι το σύμβολο περιοχής συχνότητας $X[k]$ διαμορφώνει τον υπο-φορέα με συχνότητα $f_k = k/T_{sym}$, για $N=6$ (i.e. $k=0,1,2,\dots,5$), ενώ μπορεί να αποδιαμορφώνεται χρησιμοποιώντας την ορθογωνικότητα μεταξύ των υποφερόντων στο δέκτη. Ας σημειωθεί ότι το αρχικό σύμβολο $X[k]$ έχει διάρκεια T_s , αλλά το μήκος του έχει επεκταθεί κατά $T_{sym} = NT_s$ μεταδίδοντας N σύμβολα σε παράλληλη μορφή, η οποία έχει τώρα διάρκεια T_{sym} . Το Σχήμα 9 (b) απεικονίζει μια τυπική πραγματοποίηση ορθογωνικότητας μεταξύ όλων των υποφερόντων. Επιπλέον, έχει αποδειχθεί ότι αυτή η διαμόρφωση πολλαπλών φερόντων μπορεί να εφαρμοστεί από IFFT και FFT στον πομπό και τον δέκτη, αντίστοιχα. Το Σχήμα 10 δείχνει ένα πλήρες διάγραμμα με μπλοκς



Εικόνα 8 : Διάγραμμα διαμόρφωσης/αποδιαμόρφωσης OFDM για $N=6$



Εικόνα 9 : Διάγραμμα πομπού και δέκτη σε ένα OFDM σύστημα

του τυπικού OFDM πομπού και δέκτη, περιλαμβάνοντας τις IFFT και FFT λειτουργίες.
[4][5]

3.2.2 Διάστημα Φύλαξης OFDM

3.2.2.1 Επίδραση του καναλιού πολλαπλών διαδρομών στα σύμβολα OFDM

Θεωρούμε το l -th OFDM σήμα

$$x_l(t) = \sum_{k=0}^{N-1} X_l[k] e^{j2\pi f_k(t-lT_{sym})}, \quad lT_{sym} < t \leq lT_{sym} + nT_s \quad (18)$$

Για το κανάλι με κρουστική απόκριση $h_l(t)$, το ληφθέν σήμα δίνεται ως :

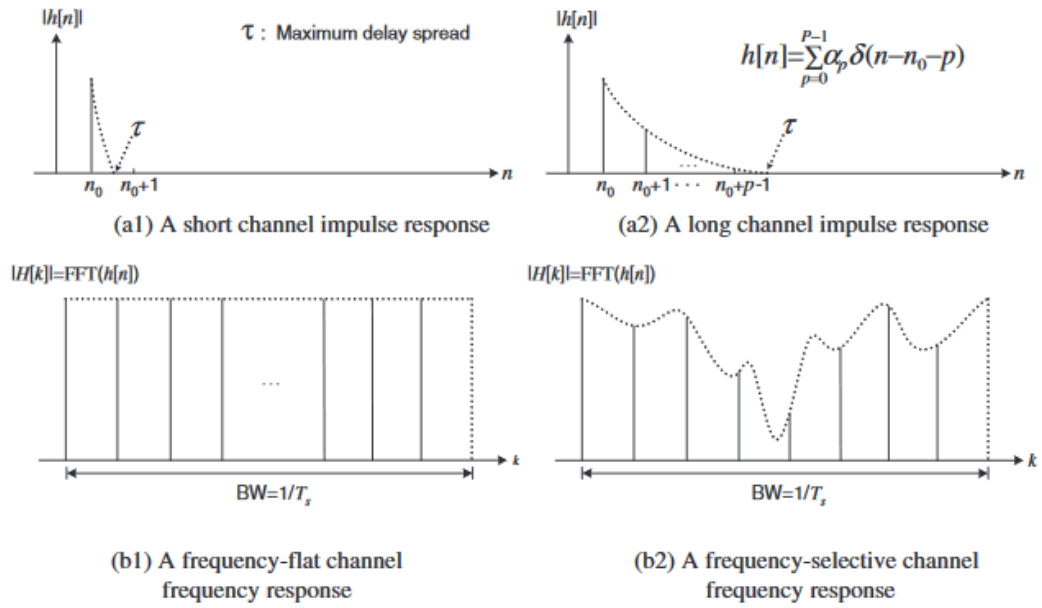
$$y_l(t) = x_l(t) * h_l(t) + z_l(t) = \int_0^\infty h_l(\tau) x_l(t - \tau) d\tau + z_l(t), \quad lT_{sym} < t \leq lT_{sym} + nT_s \quad (19)$$

Όπου το $z_l(t)$ είναι η διαδικασία προσθετικού λευκού Gaussian θορύβου (AWGN). Παίρνοντας τα δείγματα της εξίσωσης 21 στο $nT_s = nT_{sym}/N$, η εξίσωση 21 μπορεί να αναπαρασταθεί σε διακριτό χρόνο ως:

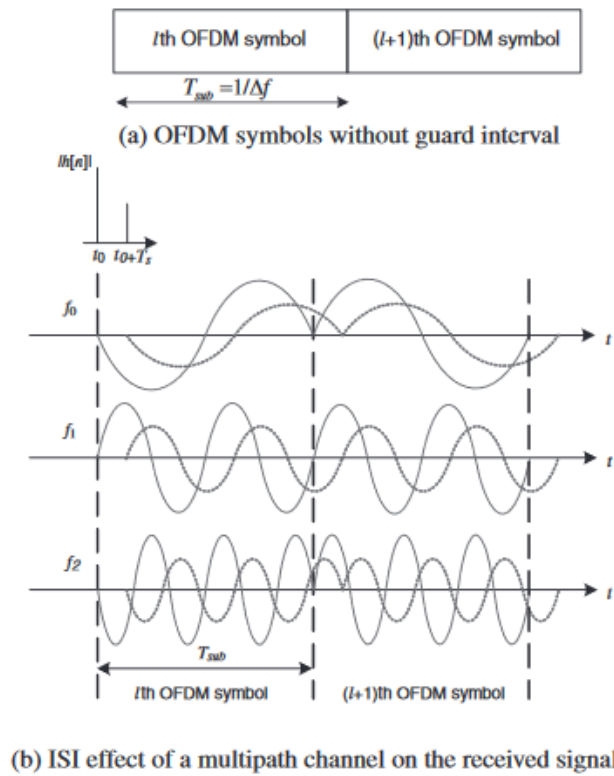
$$y_l[n] = x_l[n] * h_l[n] + z_l[n] = \sum_{m=0}^\infty h_l[m] x_l[n - m] + z_l[n] \quad (20)$$

Όπου $x_l[n] = x_l(nT_s)$, $y_l[n] = y_l(nT_s)$, $h_l[n] = h_l(nT_s)$ και $z_l[n] = z_l(nT_s)$.

Για να κατανοήσουμε ένα φαινόμενο ISI του καναλιού πολλαπλών διαδρομών, εξετάζουμε τα επεξηγηματικά παραδείγματα για το κανάλι διακριτού χρόνου στο Σχήμα 11, όπου εμφανίζονται δύο παλμικές αποκρίσεις με διαφορετικά μήκη μαζί με τις αποκρίσεις συχνότητας. Το σχήμα 12 απεικονίζει ένα φαινόμενο ISI του καναλιού πολλαπλών διαδρομών σε δύο διαδοχικά σύμβολα OFDM. Το T_{sub} δηλώνει τη διάρκεια του πραγματικού συμβόλου OFDM χωρίς χρονικό διάστημα προστασίας. Αφού το $W=1/T_s$ και έτσι, $\Delta f=W/N=1/(NT_s)$ και $T_{sub}=NT_s=1/\Delta f$. Με την επέκταση της διάρκειας του συμβόλου κατά N φορές (δηλαδή, $T_{sub} = NT_s$), το αποτέλεσμα του καναλιού εξασθένησης πολλαπλών διαδρομών μειώνεται σημαντικά στο σύμβολο OFDM. Ωστόσο, η επίδρασή του παραμένει ως επιβλαβής παράγοντας που μπορεί να σπάσει την ορθογωνικότητα μεταξύ των υποφερόντων του συστήματος OFDM. Όπως φαίνεται στο Σχήμα 12 (b), το πρώτο λαμβανόμενο σύμβολο (γράφεται σε μια συμπαγή γραμμή) αναμιγνύεται με το δεύτερο λαμβανόμενο σύμβολο (γράφεται σε μια διακεκομμένη γραμμή), το οποίο δημιουργεί το ISI. Είναι προφανές ότι όλα τα υποφέροντα δεν είναι πλέον ορθογώνια κατά τη διάρκεια κάθε συμβόλου OFDM. Για να διασφαλιστεί η απόδοση του OFDM, πρέπει να υπάρχουν κάποια μέσα αντιμετώπισης του ISI μέσω του καναλιού πολλαπλών διαδρομών. Όπως συζητείται στη συνέχεια, θα είναι απαραίτητο ένα διάστημα προστασίας μεταξύ δύο διαδοχικών συμβόλων OFDM.



Εικόνα 10 : Κρουστική απόκριση καναλιού διακριτού χρόνου



Εικόνα 11 : Αποτέλεσμα καναλιού πολλαπλών διαδρομών στο ληφθέν σήμα χωρίς το διάστημα προστασίας

3.2.2.2 Κυκλικό Πρόθεμα (Cyclic Prefix)

Το διάστημα προστασίας OFDM μπορεί να εισαχθεί με δύο διαφορετικούς τρόπους. Το ένα είναι η μηδενική επένδυση (ZP) που γεμίζει το διάστημα προστασίας με μηδενικά. Το άλλο είναι η κυκλική επέκταση του συμβόλου OFDM (για κάποια συνέχεια) με CP (κυκλικό πρόθεμα) ή CS (κυκλικό επίθημα). Το CP επεκτείνει το OFDM σύμβολο αντιγράφοντας τα τελευταία δείγματα του συμβόλου OFDM στο μπροστινό μέρος του. Ας θεωρηθεί ότι το T_G δηλώνει το μήκος του CP σε δείγματα. Στη συνέχεια, τα εκτεταμένα σύμβολα OFDM έχουν τώρα διάρκεια $T_{sym} = T_{sub} + T_G$. Το σχήμα 4.13 (a) δείχνει δύο διαδοχικά σύμβολα OFDM, καθένα από τα οποία έχει CP μήκους T_G , ενώ απεικονίζει το σύμβολο OFDM μήκους $T_{sym} = T_{sub} + T_G$. Εν τω μεταξύ, το Σχήμα 13 (b) τα απεικονίζει από κοινού στους τομείς χρόνου και συχνότητας. Το Σχήμα 14 (c) δείχνει τα ISI αποτελέσματα ενός καναλιού πολλαπλών διαδρομών σε μερικά υποφέροντα του συμβόλου OFDM. Από αυτό το σχήμα μπορεί να φανεί ότι εάν το μήκος του διαστήματος προστασίας έχει οριστεί μεγαλύτερο ή ίσο με τη μέγιστη καθυστέρηση ενός καναλιού πολλαπλών διαδρομών, το αποτέλεσμα ISI ενός συμβόλου OFDM (σχεδιάζεται σε μια διακεκομμένη γραμμή) στο επόμενο σύμβολο περιορίζεται εντός του διαστήματος προστασίας, ώστε να μην επηρεάζει το FFT του επόμενου συμβόλου OFDM, που λαμβάνεται για τη διάρκεια του T_{sub} . Αυτό συνεπάγεται ότι το διάστημα προστασίας που είναι μεγαλύτερο από τη μέγιστη καθυστέρηση του καναλιού πολλαπλών διαδρομών επιτρέπει τη διατήρηση της ορθογωνικότητας μεταξύ των υποφερόντων. Δεδομένου ότι η συνέχεια του κάθε καθυστερημένου υποφέροντος εγγυήθηκε από το CP, η ορθογωνικότητα του με όλα τα υπόλοιπα υποφέροντα διατηρείται πάνω από το T_{sub} , έτσι ώστε

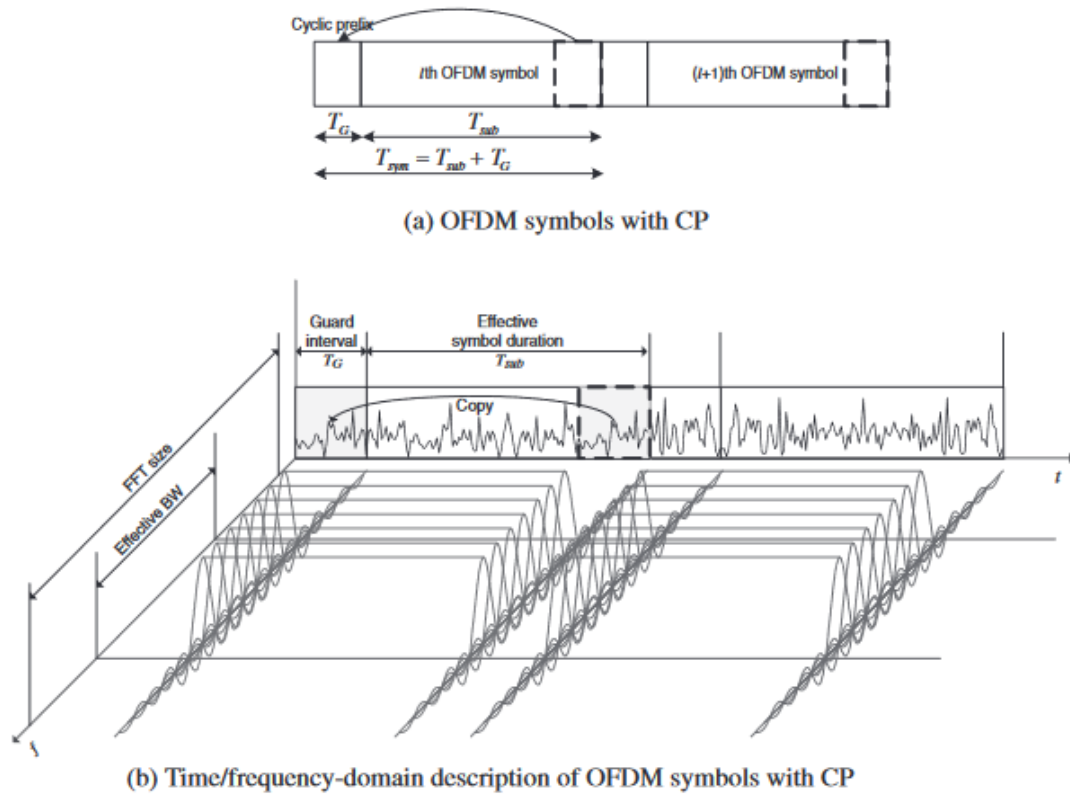
$$\frac{1}{T_{sub}} \int_0^{T_{sub}} e^{j2\pi f_k(t-t_0)} e^{-j2\pi f_i(t-t_0)} dt = 0, \quad k \neq i \quad (21)$$

για το πρώτο OFDM σήμα που φτάνει με καθυστέρηση t_0 , και

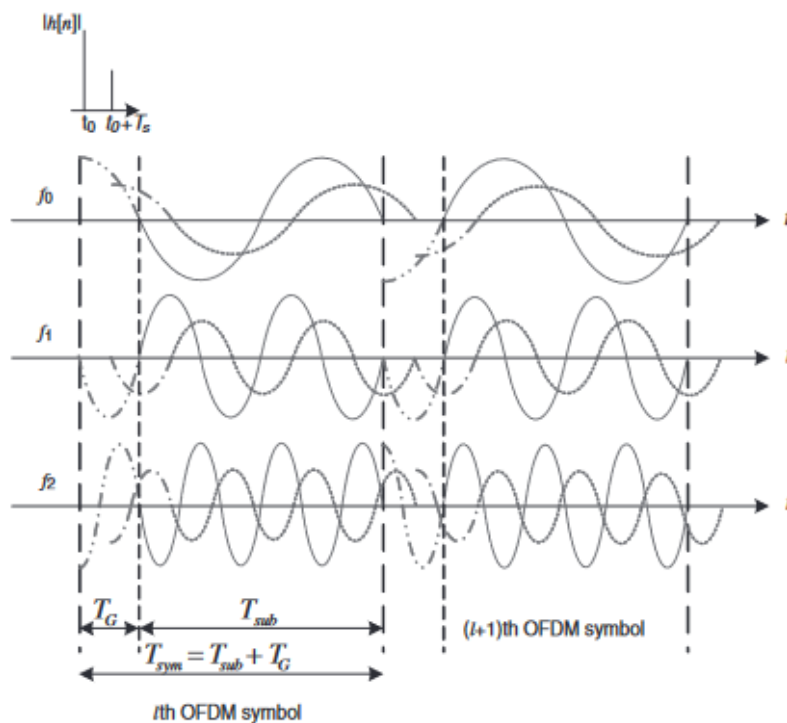
$$\frac{1}{T_{sub}} \int_0^{T_{sub}} e^{j2\pi f_k(t-t_0)} e^{-j2\pi f_i(t-t_0-T_s)} dt = 0, \quad k \neq i \quad (22)$$

για το δεύτερο OFDM σήμα που φτάνει με καθυστέρηση $t_0 + T_s$.

Το Σχήμα 13(a) δείχνει ότι εάν το μήκος του διαστήματος προστασίας έχει ρυθμιστεί μικρότερο από τη μέγιστη καθυστέρηση ενός καναλιού πολλαπλών διαδρομών, το τμήμα της ουράς ενός συμβόλου OFDM (που υποδηλώνεται από ένα τέταρτο κύκλο) επηρεάζει το τμήμα κεφαλής του επόμενου συμβόλου, με αποτέλεσμα το ISI. Στην πράξη, μπορεί να συμβεί μετατόπιση χρονισμού συμβόλων (STO), το οποίο εμποδίζει το head ενός συμβόλου OFDM να συμπέσει με το σημείο εκκίνησης του παραθύρου FFT.



Εικόνα 12 : OFDM σύμβολα με CP και περιγραφή του χρόνου/συχνότητας αυτών

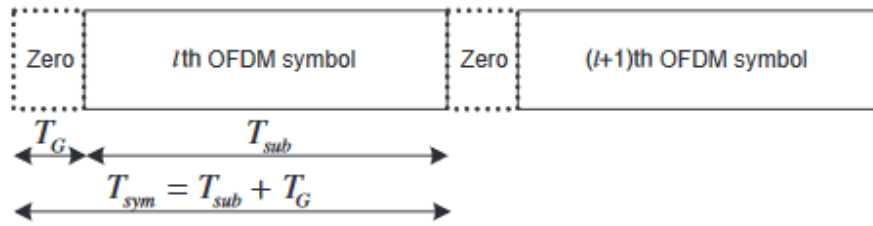


(c) ISI effect of a multipath channel for each subcarrier

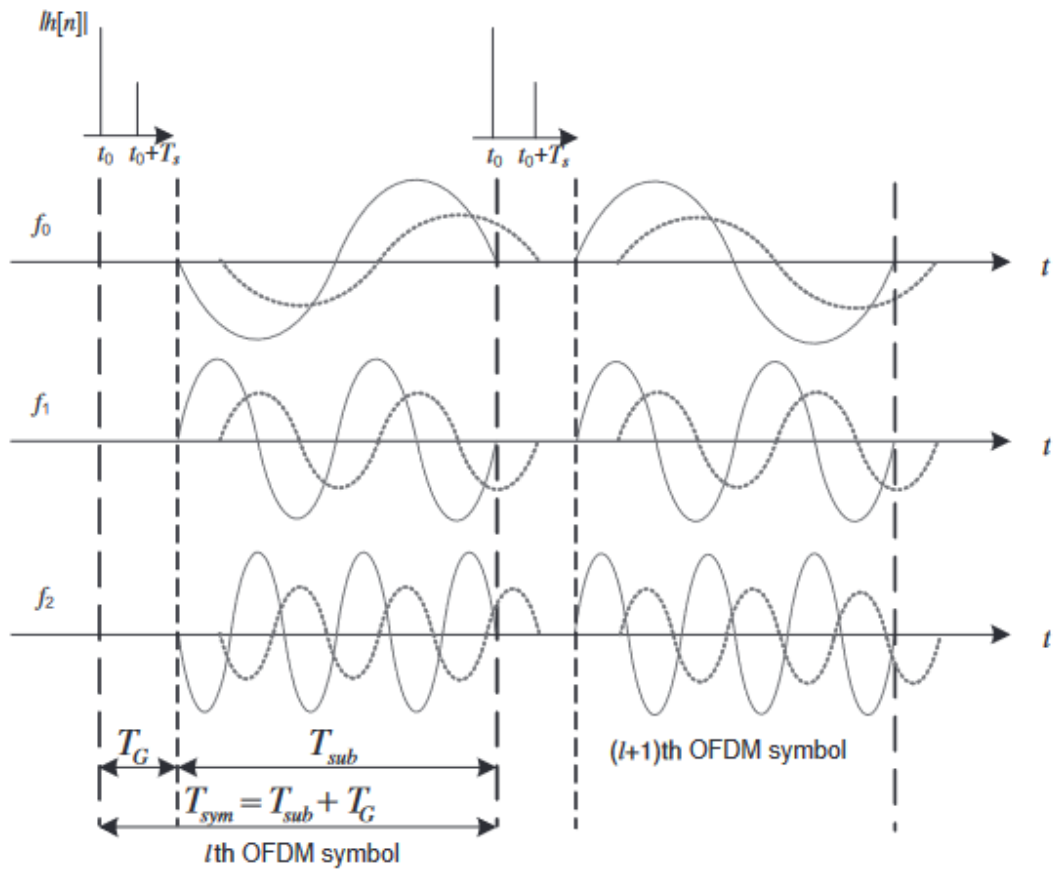
Εικόνα 13 : Επίδραση του καναλιού πολλαπλών διαδρομών στα OFDM σύμβολα με CP

3.2.3 Ζώνη Φύλαξης OFDM

Κάθε υποφορέας ενός OFDM συμβόλου με την αποτελεσματική διάρκεια T_{sub} μπορεί να θεωρηθεί ως ένα μονού τόνου σήμα πολλαπλασιασμένο με ένα ορθογώνιο παράθυρο μήκους T_{sub} , του οποίου το φάσμα είναι μια sinc συνάρτηση με zero-crossing εύρος $2/T_{sub}$. Έτσι, η ισχύς του φάσματος ενός OFDM σήματος είναι το άθροισμα πολλών μετατοπισμένης συχνότητας sinc συναρτήσεων, οι οποίες έχουν μεγάλη εκτός ζώνης ισχύ τόσο ώστε συμβαίνει ACI. Ως αποτέλεσμα, απαιτείται μια ζώνη φύλαξης (guard band) για να μειώσει την επίδραση του ACI στο OFDM σύστημα.



(a) OFDM symbols with ZP



(b) The ISI effect of a multipath channel for each subcarrier

Εικόνα 14 : Επίδραση του καναλιού πολλαπλών διαδρομών στα OFDM σύμβολα με ZP

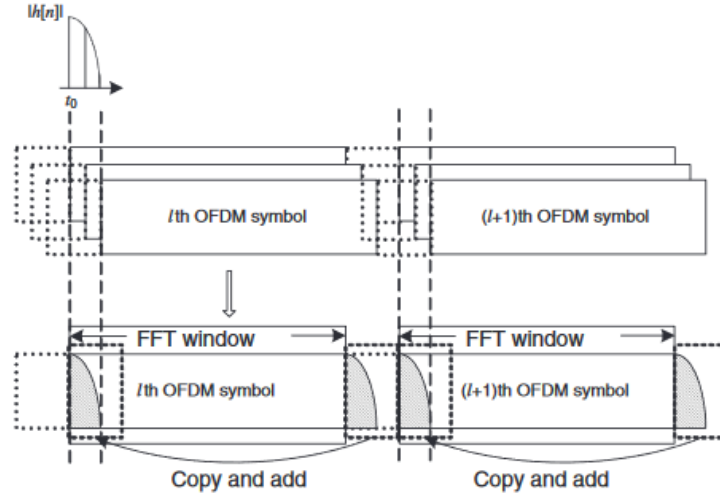
Για να μειωθεί η ισχύς εκτός του εύρους των συμβόλων OFDM, μπορεί να χρησιμοποιηθεί ένα BPF, αλλά μπορεί να απαιτεί τεράστιο υπολογισμό και υψηλή πολυπλοκότητα για να γίνει αποδεκτή η απόδοση φιλτραρίσματος. Εναλλακτικά, μπορεί να χρησιμοποιηθεί μια λειτουργία διαμόρφωσης περιοχών χρόνου όπως το υψωμένο συνημίτονο (RC). Τα passband και baseband σήματα για το l th OFDM

σύμβολο, που διαμορφώνεται από ένα παράθυρο RC $\Psi_{l,k}(t)$ γράφονται όπως φαίνεται παρακάτω:

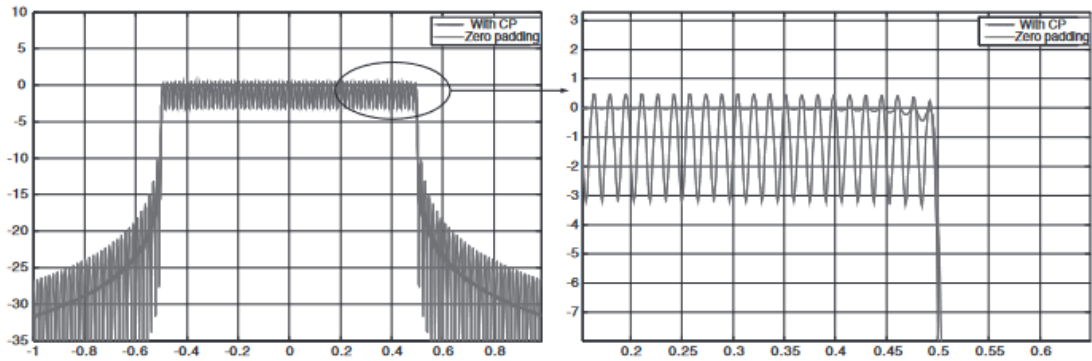
$$x_l(t) = \text{Re}\{h_{RC}(t - lT_{sym}) \sum_{k=0}^{N-1} X_{l,k} \Psi_{l,k}(t)\} \quad (23)$$

και

$$x_l^{rc} = h_{RC}(t - lT_{sym}) \sum_{k=0}^{N-1} X_{l,k} e^{j2\pi k \Delta f (t - lT_{sym})} \quad (24)$$



Εικόνα 15 : Αντιγραφή και πρόσθεση του διαστήματος προστασίας του επόμενου συμβόλου στο head κομμάτι του τρέχοντος συμβόλου



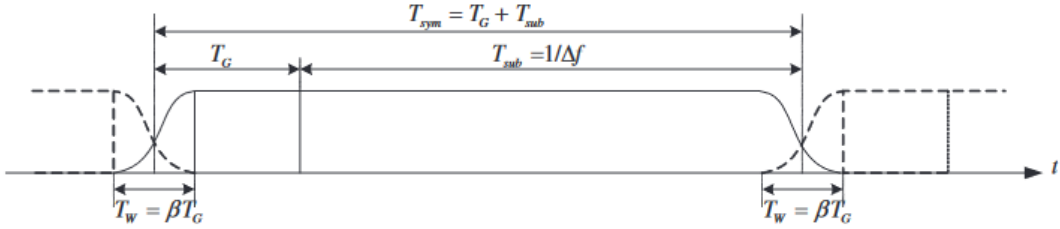
Εικόνα 16 : Ισχύς φάσματος 2 OFDM συμβόλων, ένα με ZP ένα με CP

όπου

$$\psi_{l,k}(t) = \begin{cases} e^{j2\pi f_k(t-lT_{sym})} & , \quad for - (T_G + \frac{T_W}{2}) \leq t \leq (T_{sub} + \frac{T_W}{2}) \\ 0 & , \quad otherwise \end{cases} \quad (25)$$

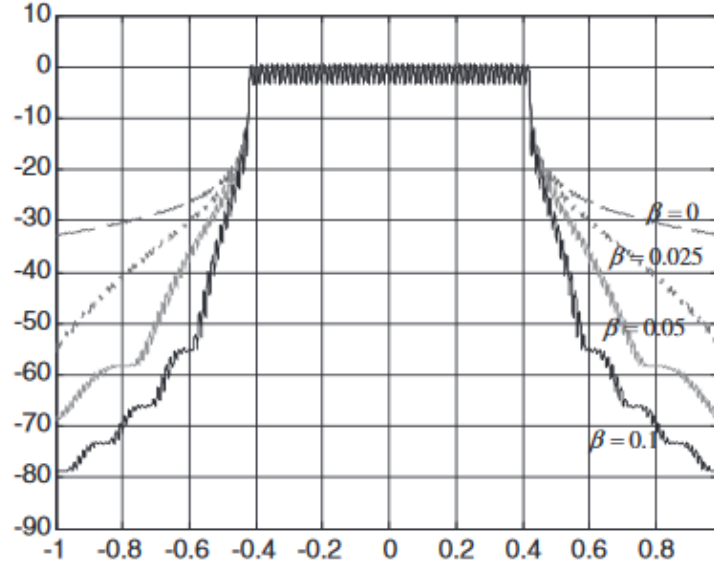
και

$$h_{RC}(t) = \begin{cases} 0.5 + 0.5 \cos\left(\frac{\pi(t+\beta T_{sym}+T_G)}{\beta T_{sym}}\right) & for - \left(T_G + \frac{\beta T_{sym}}{2}\right) \leq t < - \left(T_G - \frac{\beta T_{sym}}{2}\right) \\ 1.0 & for - \left(T_G + \frac{\beta T_{sym}}{2}\right) \leq t < - \left(T_{sub} - \frac{\beta T_{sym}}{2}\right) \\ 0.5 + 0.5 \cos(\pi(t - T_{sub} + \beta T_{sym})/\beta T_{sym}) & for \left(T_{sub} - \frac{\beta T_{sym}}{2}\right) \leq t < - \left(T_{sub} + \frac{\beta T_{sym}}{2}\right) \end{cases} \quad (26)$$



Εικόνα 17 : Raised cosine window for OFDM symbol

Το Σχήμα 18 δείχνει ένα παράθυρο υψωμένου συνημίτονου που χρησιμοποιείται για τη διαμόρφωση των συμβόλων OFDM για τη μείωση των ισχύων εκτός ζώνης. Καθώς ο roll-off (παράγοντας κύλισης) παράγοντας β αυξάνεται, το μεταβατικό μέρος του παραθύρου RC γίνεται πιο ομαλό, έτσι ώστε το ACI να μπορεί να μειωθεί σε αντάλλαγμα για ένα πιο αποτελεσματικό διάστημα προστασίας. Ένα άλλο μέτρο κατά του ACI είναι η χρήση των εικονικών φερόντων (VC), οι οποίοι είναι τα αχρησιμοποίητα υποφέροντα και στα δύο άκρα της ζώνης μετάδοσης. Δεν απαιτείται πρόσθετη επεξεργασία όταν χρησιμοποιούνται τα εικονικά φέροντα. Ωστόσο, η φασματική απόδοση (εύρος ζώνης) μειώνεται κατά N_{used} / N_{times} λόγω των αχρησιμοποίητων υποφερόντων, όπου το N_{used} είναι ο αριθμός των υποφερόντων που χρησιμοποιούνται για τη μετάδοση δεδομένων. Τα εικονικά φέροντα μπορούν να χρησιμοποιηθούν σε συνδυασμό με το RC παράθυρο για τη μείωση της ισχύος εκτός ζώνης και τελικά για την καταπολέμηση του ACI. Το σχήμα 19 δείχνει το φάσμα ισχύος των παραθύρων RC με διαφορετικούς παράγοντες roll-off, όπου 54 από τα 64 συνολικά υποφέροντα χρησιμοποιούνται για τη μετάδοση δεδομένων (εξαιρουμένων των 10 εικονικών υποφερόντων).[5] Από αυτό το σχήμα φαίνεται ότι η ισχύς εκτός ζώνης μειώνεται καθώς ο παράγοντας roll-off γίνεται μεγαλύτερος:



Εικόνα 18 Φάσμα ισχύος των παραθύρων RC

3.2.4 BER of OFDM Scheme

Οι αναλυτικές BER εκφράσεις για M-ary σηματοδότηση σε AWGN και Rayleigh κανάλια δίνονται αντίστοιχα ως:

$$P_e = \frac{2(M-1)}{M \log_2 M} Q \left(\sqrt{\frac{6E_b}{N_0} \cdot \frac{\log_2 M}{M^2-1}} \right) \quad (27)$$

$$P_e = \frac{M-1}{M \log_2 M} \left(1 - \sqrt{\frac{\frac{3\gamma \log_2 M}{M^2-1}}{\frac{3\gamma \log_2 M}{M^2-1} + 1}} \right) \quad (28)$$

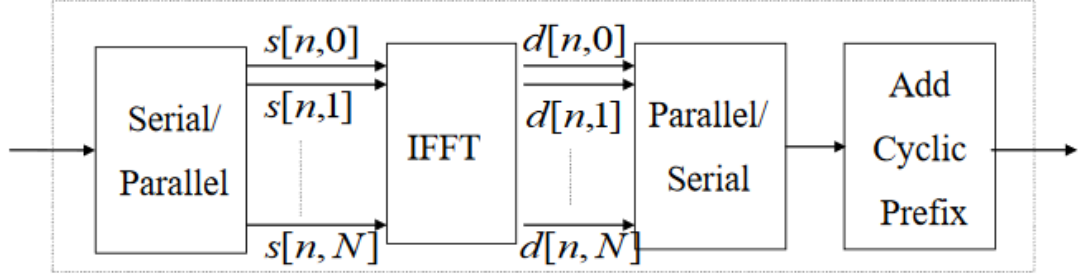
Όπου τα γ και M δηλώνουν το E_b/N_0 και την τάξη διαμόρφωσης, αντίστοιχα, όπου το $Q(\cdot)$ είναι η καθιερωμένη Q συνάρτηση που ορίζεται ως:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt \quad (29)$$

Παρατηρείται ότι αν τα N_{used} υποφέροντα από τα συνολικά N (FFT size) υποφέροντα (εκτός από $N_{vc}=N-N_{used}$ εικονικά υποφέροντα) χρησιμοποιούνται για να μεταφέρουν

δεδομένα, το SNR στο πεδίο του χρόνου, SNR_t , διαφέρει από το SNR στο πεδίο συχνότητας, SNR_f , όπως φαίνεται παρακάτω:

$$SNR_t = SNR_f + 10 \log \frac{N_{used}}{N} [dB] \quad (30)$$

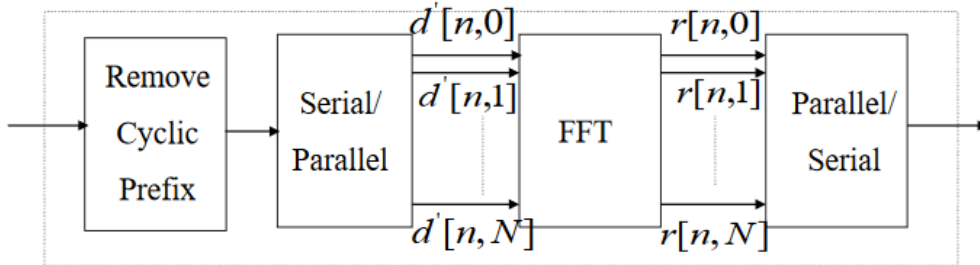


Εικόνα 19 : OFDM Διαμορφωτής

Με τη βοήθεια του IFFT, το σήμα που καταλαμβάνει συγκεκριμένο φάσμα, μετατρέπεται σε σήμα συναρτήσει του χρόνου.

Προστίθεται στη συνέχεια το διάστημα φύλαξης.

$$d[n, i] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} s[n, k] e^{j2\pi i \frac{k}{N}} \quad (31)$$



Εικόνα 20 : OFDM Αποδιαμορφωτής

Αφαιρείται το διάστημα φύλαξης.

Με τη βοήθεια του FFT, δημιουργεί σήμα συναρτήσει της συχνότητας.[5]

$$r[n, k] = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} d[n, i] e^{-j2\pi k \frac{i}{N}} \quad (32)$$

3.3 Εφαρμογές OFDM

Η Διαμόρφωση OFDM έχει διάφορες εφαρμογές όσον αφορά τον τομέα των Τηλεπικοινωνιών λόγω των δυνατοτήτων που προσφέρει. Μερικές από αυτές είναι τα ADSL modems, UMTS, Wi-Fi καθώς επίσης και στα δίκτυα κινητής τηλεφωνίας 4ης γενιάς WiMAX και LTE. Επιπλέον, άλλες εφαρμογές αφορούν την ψηφιακή μετάδοση ήχου και βίντεο (DAB και DVB-T) όπως επίσης και τα συστήματα IEEE 802.11a&g (WLAN).

3.4 Πλεονεκτήματα και Μειονεκτήματα της OFDM

Η χρήση της διαμόρφωσης πολυπλεξίας ορθογώνιας διαίρεσης συχνότητας προσφέρει πολλά πλεονεκτήματα. Αρχικά, χρησιμοποιεί αποτελεσματικά το εύρος ζώνης και είναι ανθεκτική στην εξασθένηση συχνότητας. Επίσης, εξουδετερώνει την διασυμβολική παρεμβολή (ISI). Η εξισορρόπηση του καναλιού γίνεται απλούστερη σε σχέση με άλλες τεχνικές που χρησιμοποιούν κυκλώματα εξισορροπητών. Με τη χρήση τεχνικών FFT η OFDM γίνεται ακόμα πιο αποτελεσματική. Επιπλέον, παρέχει καλή προστασία στις παρεμβολές μεταξύ των καναλιών. Τέλος, οι διαδικασίες για τη διατήρηση της ορθογωνικότητας είναι πιο εύκολα υλοποιήσιμες από άλλες διαδικασίες που λαμβάνουν χώρα σε τεχνικές όπως CDMA, TDMA. Εφαρμόζεται κυρίως όταν οι απαιτήσεις είναι υψηλές καθώς το κόστος συντήρησης και λειτουργίας είναι αρκετά χαμηλό.[16]

Παρόλα τα οφέλη που προσφέρει, η OFDM έχει και μερικά μειονεκτήματα. Είναι περισσότερο ευαίσθητη στην αντιστάθμιση συχνότητας φέροντος και στην ολίσθηση συχνότητας, σε σχέση με τα συστήματα που χρησιμοποιούν ένα μόνο φέρον, εξ' αιτίας της διαρροής ενέργειας με το DFT. Επίσης, στα σήματα OFDM προστίθεται κρουστικός θόρυβος ο οποίος εξαρτάται από το πλάτος των σημάτων και για το λόγο αυτό απαιτεί την ύπαρξη RF ενισχυτών ισχύος με μεγάλο λόγο πλάτους ισχύος προς μέση τιμή.[16] Επιπλέον, παρατηρείται ευπάθεια σε προβλήματα συγχρονισμού λόγω

- a. Φαινομένου Doppler : Είναι μια απόκλιση συχνότητας του λαμβανόμενου σήματος που οφείλεται στη σχετική κίνηση του δέκτη ως προς τον πομπό.
- b. Απόκλισης συχνότητας ταλαντωτών.

4 Μηχανική Μάθηση

Ένας αλγόριθμος Μηχανικής Μάθησης είναι μια μαθηματική έκφραση που αναπαριστά δεδομένα στην μορφή ενός προβλήματος, συνήθως ένα επιχειρηματικό πρόβλημα. Ο στόχος είναι να πάει από τα δεδομένα στη διορατικότητα. Για παράδειγμα, εάν ένας ηλεκτρονικός λιανοπωλητής θέλει να προβλέψει τις πωλήσεις για το επόμενο τρίμηνο, μπορεί να χρησιμοποιήσει έναν αλγόριθμο μηχανικής εκμάθησης που προβλέπει αυτές τις πωλήσεις με βάση προηγούμενες πωλήσεις και άλλα σχετικά δεδομένα. Ομοίως, ένας κατασκευαστής ανεμόμυλου μπορεί να παρακολουθεί οπτικά σημαντικό εξοπλισμό και να τροφοδοτεί τα δεδομένα βίντεο μέσω αλγορίθμων που έχουν εκπαιδευτεί για τον εντοπισμό επικίνδυνων ρωγμών. Παρακάτω θα αναλυθούν δέκα μέθοδοι οι οποίοι μπορούν να προσφέρουν μια επισκόπηση της έννοιας της Μηχανικής Μάθησης.

Μπορεί να γίνει μια διάκριση μεταξύ δύο γενικών κατηγοριών μηχανικής μάθησης: υπό επίβλεψη και χωρίς επίβλεψη. Εφαρμόζουμε υπό επίβλεψη τεχνικές ML όταν έχουμε ένα κομμάτι των δεδομένων που θέλουμε να προβλέψουμε ή να εξηγήσουμε. Το πράττουμε χρησιμοποιώντας προηγούμενα δεδομένα εισροών και εκροών για να προβλέψουμε μια έξοδο με βάση μια νέα είσοδο. Για παράδειγμα, οι υπό επίβλεψη τεχνικές ML μπορούν να χρησιμοποιηθούν για την βοήθεια μιας επιχείρησης παροχής υπηρεσιών που θέλει να προβλέψει τον αριθμό των νέων χρηστών που θα εγγραφούν στην υπηρεσία τον επόμενο μήνα. Αντίθετα, το ML χωρίς επίβλεψη εξετάζει τρόπους συσχέτισης και ομαδοποίησης σημείων δεδομένων χωρίς τη χρήση μεταβλητής στόχου για την πρόβλεψη. Με άλλα λόγια, αξιολογεί τα δεδομένα από την άποψη των χαρακτηριστικών και χρησιμοποιεί τα χαρακτηριστικά για να σχηματίσουν συμπλέγματα στοιχείων που είναι παρόμοια μεταξύ τους. Για παράδειγμα, οι τεχνικές ML χωρίς επίβλεψη θα μπορούσαν να χρησιμοποιηθούν για την βοήθεια ενός λιανοπωλητή που θέλει να τμηματογραφεί προϊόντα με παρόμοια χαρακτηριστικά — χωρίς να χρειαστεί να καθοριστούν εκ των προτέρων ποια χαρακτηριστικά θα χρησιμοποιηθούν. Στην προκειμένη περίπτωση η κατηγορία που θα χρησιμοποιήσουμε στην διπλωματική είναι οι τεχνικές Μηχανικής Μάθησης υπό επίβλεψη.

4.1 Τεχνικές Μηχανικής Μάθησης

4.1.1 Αναδρομή

Οι μέθοδοι αναδρομής εντάσσονται στην κατηγορία των υπό επίβλεψη τεχνικών ML. Βοηθούν στην πρόβλεψη ή την επεξήγηση μιας συγκεκριμένης αριθμητικής τιμής με βάση ένα σύνολο προηγούμενων δεδομένων, για παράδειγμα την πρόβλεψη της τιμής ενός ακινήτου με βάση προηγούμενα δεδομένα τιμολόγησης για παρόμοιες ιδιότητες.

Η απλούστερη μέθοδος είναι η γραμμική παλινδρόμηση όπου χρησιμοποιούμε τη μαθηματική εξίσωση της γραμμής ($y = m * x + b$) για να μοντελοποιήσουμε ένα σύνολο δεδομένων. Εκπαιδεύουμε ένα μοντέλο γραμμικής παλινδρόμησης με πολλά ζεύγη δεδομένων (x, y) υπολογίζοντας τη θέση και την κλίση μιας γραμμής που ελαχιστοποιεί τη συνολική απόσταση μεταξύ όλων των σημείων δεδομένων και της γραμμής. Με άλλα λόγια, υπολογίζουμε την κλίση (m) και το σημείο τομής y (b) για μια γραμμή που προσεγγίζει καλύτερα τις παρατηρήσεις στα δεδομένα. Οι τεχνικές παλινδρόμησης καλύπτουν το φάσμα από απλό (όπως γραμμική παλινδρόμηση) σε πολύπλοκο (όπως η κανονικοποιημένη γραμμική παλινδρόμηση, η πολυωνυμική παλινδρόμηση, τα δέντρα αποφάσεων και οι τυχαίες παλινδρομήσεις δάσους, τα νευρωνικά δίκτυα, μεταξύ άλλων).

4.1.2 Κατηγοριοποίηση (Classification)

Μια άλλη κατηγορία Μηχανικής Μάθησης υπό επίβλεψη είναι οι μέθοδοι κατηγοριοποίησης, οι οποίες προβλέπουν ή εξηγούν μια τιμή κλάσης. Για παράδειγμα, μπορούν να σας βοηθήσουν να προβλέψετε αν ένας ηλεκτρονικός πελάτης θα αγοράσει ένα προϊόν ή όχι. Αλλά οι μέθοδοι κατηγοριοποίησης δεν περιορίζονται σε δύο κατηγορίες. Για παράδειγμα, μια μέθοδος κατηγοριοποίησης θα μπορούσε να βοηθήσει στην αξιολόγηση του κατά πόσον μια δεδομένη εικόνα περιέχει αυτοκίνητο ή φορτηγό. Σε αυτή την περίπτωση, η έξοδος θα είναι 3 διαφορετικές τιμές: 1) η εικόνα περιέχει ένα αυτοκίνητο, 2) η εικόνα περιέχει ένα φορτηγό, ή 3) η εικόνα δεν περιέχει ούτε ένα αυτοκίνητο ούτε ένα φορτηγό. Ο απλούστερος αλγόριθμος κατηγοριοποίησης είναι η καταγραφή παλινδρόμησης η οποία υπολογίζει την πιθανότητα εμφάνισης ενός συμβάντος με βάση μία ή περισσότερες εισόδους.[17]

4.1.3 Clustering

Με τις μεθόδους συμπλέγματος, μπαίνουμε στην κατηγορία των μη επιτηρούμενων ML επειδή στόχος τους είναι να ομαδοποιήσουν παρατηρήσεις που έχουν παρόμοια χαρακτηριστικά. Οι μέθοδοι συμπλέγματος δεν χρησιμοποιούν πληροφορίες εξόδου για εκπαίδευση, αλλά αντίθετα επιτρέπουν στον αλγόριθμο να καθορίσει την έξοδο. Στις μεθόδους συμπλέγματος, μπορούμε να χρησιμοποιήσουμε μόνο απεικονίσεις για να ελέγξουμε την ποιότητα της λύσης. Η πιο δημοφιλής μέθοδος συμπλέγματος είναι το K-Means, όπου το "K" αντιπροσωπεύει τον αριθμό των συμπλεγμάτων που επιλέγει να δημιουργήσει ο χρήστης.[17]

4.1.4 Μείωση Διαστάσεων (Dimensionality Reduction)

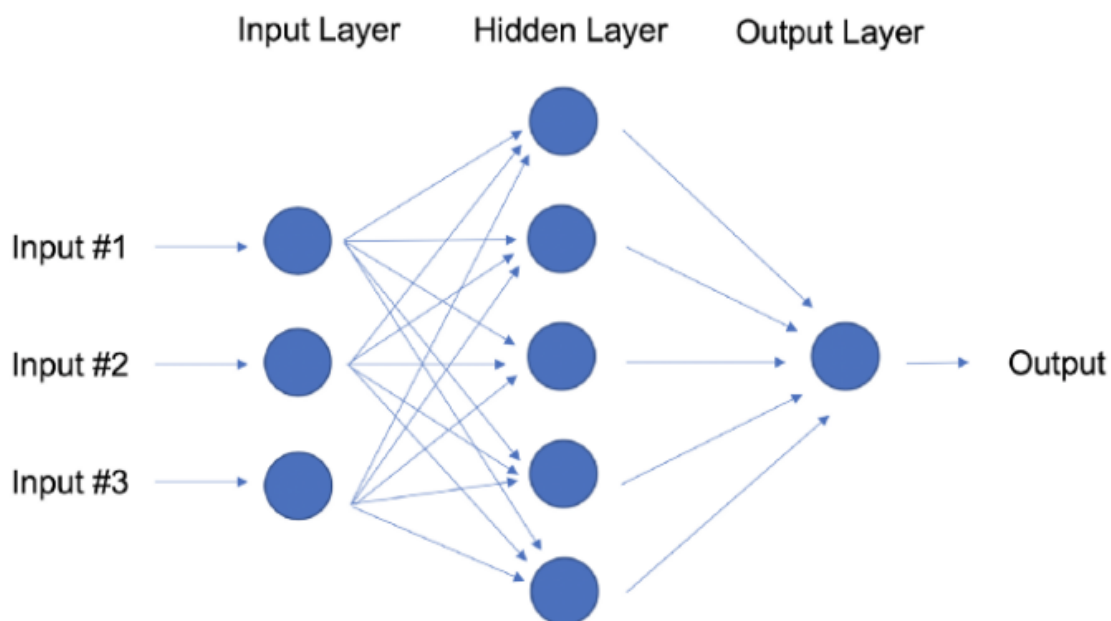
Χρησιμοποιούμε την μείωση διαστάσεων για να αφαιρέσουμε τη λιγότερο σημαντική πληροφορία από ένα σύνολο δεδομένων (dataset). Είναι θεμιτό και πρακτικό όταν σε ένα σύνολο δεδομένων υπάρχουν εκατοντάδες ή χιλιάδες στήλες (χαρακτηριστικά),

οπότε η μείωση του αριθμού τους είναι σημαντικό. Για παράδειγμα, οι εικόνες μπορεί να περιέχουν χιλιάδες pixels, από τα οποία κάποια μπορεί να μην χρειάζονται στην ανάλυση που θα γίνει. Ή κατά τη δοκιμή μικροτσίπ στο πλαίσιο της διαδικασίας κατασκευής, μπορεί να έχουμε χιλιάδες μετρήσεις και δοκιμές που εφαρμόζονται σε κάθε τσιπ, πολλές από τις οποίες παρέχουν περιττές πληροφορίες. Σε αυτές τις περιπτώσεις χρειάζονται αλγόριθμοι Μείωσης Διαστάσεων για να κάνουν το σύνολο δεδομένων διαχειρίσιμο. Η πιο δημοφιλής μέθοδος μείωσης διαστάσεων είναι η Ανάλυση Κύριων Στοιχείων (PCA), η οποία μειώνει τη διάσταση του χώρου δυνατοτήτων βρίσκοντας νέα διανύσματα που μεγιστοποιούν τη γραμμική παραλλαγή των δεδομένων. Η PCA μπορεί να μειώσει τη διάσταση των δεδομένων δραματικά και χωρίς να χάσει πάρα πολλές πληροφορίες όταν οι γραμμικές συσχετίσεις των δεδομένων είναι ισχυρές.[17]

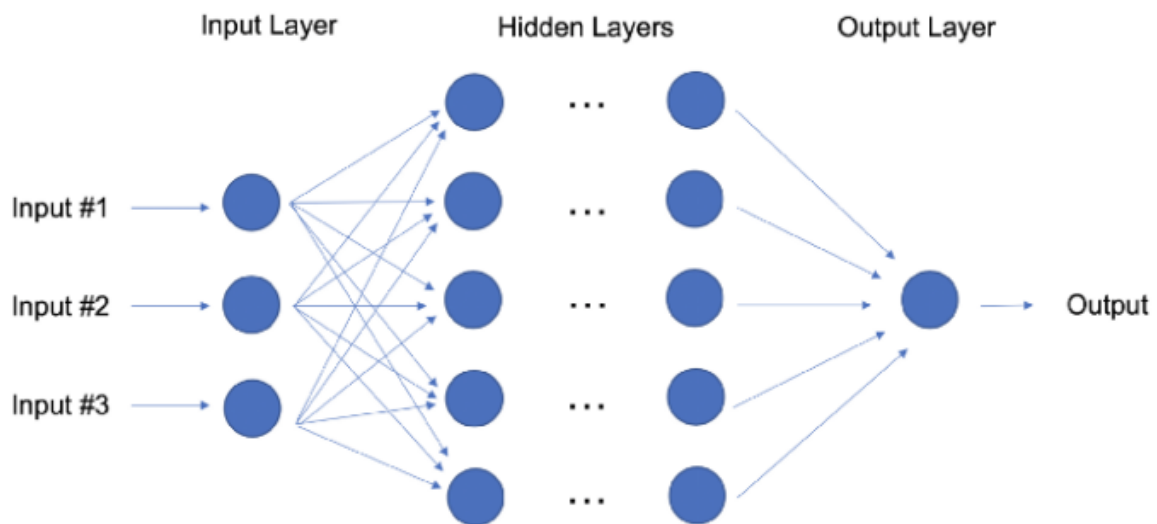
4.1.5 Νευρωνικά Δίκτυα και Βαθιά Μάθηση

Ο στόχος των Νευρωνικών δικτύων είναι να συλλάβουν μη γραμμικά μοτίβα στα δεδομένα με την προσθήκη επιπέδων των παραμέτρων στο μοντέλο. Στην παρακάτω εικόνα, το απλό νευρωνικό δίκτυο έχει τρεις εισόδους, ένα μόνο κρυφό στρώμα με πέντε παραμέτρους και ένα επίπεδο εξόδου.

Ο όρος Βαθιά Μάθηση προέρχεται από ένα νευρωνικό δίκτυο με πολλά κρυμμένα επίπεδα και συμπυκνώνει μια μεγάλη ποικιλία αρχιτεκτονικών.



Εικόνα 21 : Νευρωνικό Δίκτυο με ένα κρυμμένο επίπεδο



Εικόνα 22 : Νευρωνικό δίκτυο με πολλά κρυμμένα επίπεδα

Για την καλύτερη απόδοση, οι τεχνικές βαθιάς μάθησης απαιτούν πολλά δεδομένα και πολλή υπολογιστική δύναμη, δεδομένου ότι η μέθοδος είναι self-tuning με πολλές παραμέτρους μέσα σε τεράστιες αρχιτεκτονικές. Αυτό καθιστά επιτακτική την χρήση πολύ ισχυρών υπολογιστών ενισχυμένους με GPU (γραφικές μονάδες επεξεργασίας). Το CUDA εδώ αποτελεί κομβική λύση παρέχοντας τα οφέλη του σχετικά με την υπολογιστική ισχύ.

4.1.6 Μεταφορά Μάθησης (Transfer Learning)

Αναφέρεται στην επαναχρησιμοποίηση προηγούμενων εκπαιδευμένων νευρωνικών δικτύων και την προσαρμογή του σε ένα νέο αλλά παρόμοιο έργο. Συγκεκριμένα, μόλις εκπαιδευτεί ένα νευρωνικό δίκτυο χρησιμοποιώντας δεδομένα για μια εργασία, μπορεί να μεταφερθεί ένα τμήμα των εκπαιδευμένων στρωμάτων και να συνδυαστούν με μερικά νέα στρώματα που μπορούν να εκπαιδευτούν χρησιμοποιώντας τα δεδομένα του νέου έργου. Με την προσθήκη μερικών στρωμάτων, το νέο νευρωνικό δίκτυο μπορεί να μάθει και να προσαρμοστεί γρήγορα στο νέο έργο. Το βασικό πλεονέκτημα της Μεταφοράς Μάθησης είναι ότι χρειάζεται λιγότερα δεδομένα για την εκπαίδευση του νευρωνικού δικτύου, το οποίο είναι ιδιαίτερα σημαντικό, διότι η εκπαίδευση για αλγόριθμους βαθιάς μάθησης είναι δαπανηρή τόσο από την άποψη του χρόνου όσο και από την άποψη των χρημάτων (υπολογιστικοί πόροι) - και φυσικά είναι συχνά πολύ δύσκολο να βρεθούν δεδομένα τα οποία έχουν «χαρακτηριστεί» για την εκπαίδευση.[17]

4.1.7 Word Embeddings

Οι ενσωματώσεις κειμένου μπορούν να καταγράψουν το περιβάλλον μιας λέξης σε ένα έγγραφο. Με τη λέξη πλαίσιο, οι ενσωματώσεις μπορούν να ποσοτικοποιήσουν την ομοιότητα μεταξύ των λέξεων, η οποία με τη σειρά της επιτρέπει την αριθμητική με λέξεις. Το Word2Vec είναι μια μέθοδος που βασίζεται σε νευρωνικά δίκτυα που αντιστοιχίζει λέξεις σε ένα σώμα με ένα αριθμητικό διάνυσμα. Στη συνέχεια, μπορούν να χρησιμοποιηθούν αυτά τα διανύσματα για την εύρεση συνωνύμων, την εκτέλεση αριθμητικών πράξεων με λέξεις ή την αναπαράσταση εγγράφων κειμένου (λαμβάνοντας το μέσο όρο όλων των διανυσματικών λέξεων σε ένα έγγραφο). Οι αναπαραστάσεις λέξεων επιτρέπουν την εύρεση ομοιοτήτων μεταξύ λέξεων, υπολογίζοντας την ομοιότητα συνημίτονου μεταξύ της διανυσματικής αναπαράστασης δύο λέξεων. Η ομοιότητα συνημίτονου μετρά τη γωνία μεταξύ δύο διανυσμάτων.[17]

4.1.8 Επεξεργασία Φυσικής Γλώσσας

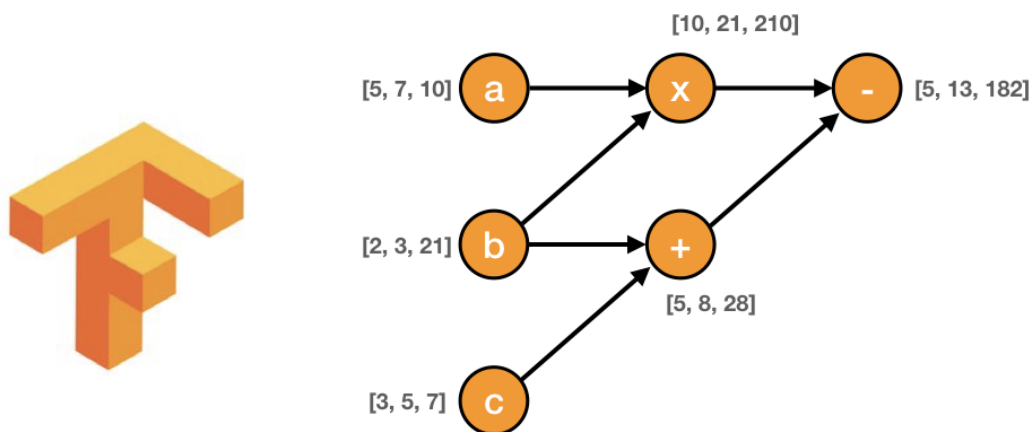
Η Επεξεργασία Φυσικής Γλώσσας (NLP) δεν είναι μια μέθοδος μηχανικής μάθησης από μόνη της, αλλά μια ευρέως χρησιμοποιούμενη τεχνική για την προετοιμασία κειμένου για Μηχανική Μάθηση. Υπάρχουν τόνοι εγγράφων κειμένου σε μια ποικιλία μορφών (Word, online blogs, ...). Τα περισσότερα από αυτά τα έγγραφα κειμένου θα είναι γεμάτα τυπογραφικά λάθη, χαρακτήρες που λείπουν και άλλες λέξεις που έπρεπε να φιλτραριστούν. Ο απλούστερος τρόπος αντιστοίχισης κειμένου σε μια αριθμητική αναπαράσταση είναι ο υπολογισμός της συχνότητας κάθε λέξης μέσα σε κάθε έγγραφο κειμένου. Ας σκεφτούμε έναν πίνακα ακεραίων όπου κάθε γραμμή αντιπροσωπεύει ένα έγγραφο κειμένου και κάθε στήλη αντιπροσωπεύει μια λέξη. Αυτή η αναπαράσταση πίνακα συχνότητας των λέξεων ονομάζεται συνήθως Πίνακας Συχνότητας Όρους (TFM). Από εκεί, μπορούμε να δημιουργήσουμε μια άλλη δημοφιλής αναπαράσταση πίνακα ενός εγγράφου κειμένου διαιρώντας κάθε καταχώρηση στον πίνακα με ένα βάρος του πόσο σημαντική είναι κάθε λέξη μέσα σε ολόκληρο το σώμα των εγγράφων.[17]

5 Υλοποίηση TensorFlow και Matlab

5.1 Εισαγωγή στο TensorFlow

Το TensorFlow είναι μια δωρεάν βιβλιοθήκη λογισμικού ανοιχτού κώδικα για ροή δεδομένων και διαφοροποιημένο προγραμματισμό σε μια σειρά εργασιών. Δημιουργήθηκε αρχικά για να αντιμετωπίσει το πρόβλημα των διεργασιών που είχαν πολύπλοκους αριθμητικούς υπολογισμούς. Είναι μια συμβολική βιβλιοθήκη μαθηματικών και χρησιμοποιείται επίσης για εφαρμογές μηχανικής μάθησης, όπως τα νευρωνικά δίκτυα.[6] Το TensorFlow αναπτύχθηκε από την ομάδα του Google Brain για εσωτερική χρήση στην Google. Κυκλοφόρησε με το Apache License 2.0 στις 9 Νοεμβρίου 2015. Το backend του είναι γραμμένο σε C/C++ γεγονός που το καθιστά να εκτελείται αρκετά πιο γρήγορα από την γλώσσα προγραμματισμού Python. Επίσης, παρέχει API για Python και C++, ωστόσο της Python είναι πιο ολοκληρωμένο και εύκολο στην χρήση.

Όσον αφορά την αρχιτεκτονική του TensorFlow, είναι ευέλικτη επιτρέποντας μας να κάνουμε υπολογισμούς σε μία ή περισσότερες CPUs ή GPUs καθώς επίσης και σε έναν διακομιστή ή μια κινητή συσκευή. Όλα αυτά μπορούν να γίνουν χρησιμοποιώντας ένα μόνο API. Το TensorFlow επιτρέπει στους προγραμματιστές να δημιουργούν γραφήματα ροής δεδομένων (δομές που περιγράφουν τον τρόπο με τον οποίο τα δεδομένα κινούνται μέσω γραφήματος ή μια σειρά κόμβων επεξεργασίας). Κάθε κόμβος στο γράφημα αντιπροσωπεύει μια μαθηματική λειτουργία και κάθε σύνδεση ή ακμή μεταξύ των κόμβων είναι ένας πολυδιάστατος πίνακας δεδομένων ή ένας tensor που είναι το αντικείμενο που χειρίζεται το TensorFlow.[7] Τέλος, το TensorFlow είναι κατάλληλο για εφαρμογές Βαθιάς Μάθησης καθώς παρέχει εκτεταμένη υποστήριξη για αυτήν καθώς επίσης και εργαλεία και μαθηματικές συναρτήσεις που είναι χρήσιμα για νευρωνικά δίκτυα.



Εικόνα 23 : Υπολογισμοί στους κόμβους του γραφήματος

5.2 CUDA

5.2.1 Γενικά για το CUDA

Το CUDA είναι μια πλατφόρμα παράλληλου υπολογισμού και ένα μοντέλο προγραμματισμού το οποίο αναπτύχθηκε από την εταιρεία Nvidia για γενικούς υπολογισμούς στις δικές της GPUs. Το CUDA επιτρέπει στους προγραμματιστές να επιταχύνουν τις εφαρμογές υπολογισμών υψηλής έντασης, αξιοποιώντας την ισχύ των GPUs για το κομμάτι παραλληλισμού του υπολογισμού. Παρόλο που υπήρχαν και άλλα προτεινόμενα APIs για τις GPUs, όπως το OpenCL, και υπάρχουν ανταγωνιστικές GPUs από άλλες εταιρείες, όπως η AMD, ο συνδυασμός των CUDA και Nvidia GPUs κυριαρχεί σε αρκετούς τομείς εφαρμογών, συμπεριλαμβανομένης της Βαθιάς Μάθησης, και αποτελεί θεμέλιο για ορισμένους από τους ταχύτερους υπολογιστές στον κόσμο.[8]

5.2.2 CUDA στην Βαθιά Μάθηση

Η Βαθιά Μάθηση έχει υπερβολική ανάγκη την ταχύτητα των υπολογισμών. Για παράδειγμα, για να εκπαιδεύσουν τα μοντέλα για τη Μετάφραση Google το 2016, οι ομάδες του Google Brain και της Μετάφρασης Google έκαναν εκατοντάδες TensorFlow εκτελέσεις μιας εβδομάδας χρησιμοποιώντας GPU. Είχαν αγοράσει 2.000 GPU επιπέδου διακομιστή από τη Nvidia για το σκοπό αυτό. Χωρίς GPU, αυτές οι εκτελέσεις εκπαίδευσης θα απαιτούσαν μήνες και όχι μια εβδομάδα για σύγκλιση. Για την ανάπτυξη παραγωγής αυτών των μοντέλων μετάφρασης TensorFlow, η Google χρησιμοποίησε ένα νέο τσιπ προσαρμοσμένης επεξεργασίας, το TPU (μονάδα επεξεργασίας tensor). Εκτός από το TensorFlow, πολλά άλλα πλαίσια (frameworks) DL βασίζονται στο CUDA για την υποστήριξη GPU τους, συμπεριλαμβανομένων των Caffe2, CNTK, Databricks, H2O.ai, Keras, MXNet, PyTorch, Theano και Torch. Στις περισσότερες περιπτώσεις χρησιμοποιούν τη βιβλιοθήκη cuDNN για τους υπολογισμούς του βαθιού νευρωνικού δικτύου. Αυτή η βιβλιοθήκη είναι τόσο σημαντική για την εκπαίδευση των πλαισίων βαθιάς μάθησης που όλα τα πλαίσια που χρησιμοποιούν μια δεδομένη έκδοση του cuDNN έχουν ουσιαστικά τους ίδιους αριθμούς απόδοσης για αντίστοιχες περιπτώσεις χρήσης. Όταν το CUDA και το cuDNN βελτιώνονται από έκδοση σε έκδοση, όλα τα πλαίσια βαθιάς μάθησης που ενημερώνονται στη νέα έκδοση βλέπουν τα κέρδη απόδοσης.[9]

5.2.3 Χρησιμότητα του CUDA για την παρούσα διπλωματική

Απαραίτητο συστατικό για την διεκπεραίωση της διπλωματικής φάνηκε να είναι η χρήση του CUDA. Αναλυτικότερα, για τη χρήση της κάρτας γραφικών και των cuda cores της πρέπει να εγκαταστήσουμε στο περιβάλλον του διερμηνευτή που

χρησιμοποιούμε στο project το Tensorflow-gpu. Αυτό γίνεται εγκαθιστώντας το Anaconda Navigator και δημιουργώντας ένα περιβάλλον με εγκατεστημένα τα libraries του CUDA για το TensorFlow. Έπειτα, από το περιβάλλον του Pycharm, στην προκειμένη περίπτωση, επιλέγουμε στις ρυθμίσεις του project ως διερμηνευτή αυτόν από το Anaconda Navigator. Στη συνέχεια, οι υπολογισμοί γίνεται με χρήσης της παράλληλης επεξεργασίας της κάρτας γραφικών και ως αποτέλεσμα είναι κατά πολύ γρηγορότεροι. Για το μοντέλο, το layer που χρησιμοποιείται είναι το CuDNNLSTM για την πραγματοποίηση των υπολογισμών από τους CUDA cores της κάρτας γραφικών. Η διάρκεια των υπολογισμών είναι και στην ουσία αυτή της εκπαίδευσης του μοντέλου η οποία μειώνεται σημαντικά σε σχέση με την χρήση των πυρήνων της CPU.

5.3 Βήματα Σχεδιασμού και Υλοποίησης του Μοντέλου

Πρώτα πρέπει να δημιουργηθεί το κατάλληλο dataset για την εκπαίδευση και τον έλεγχο του μοντέλου που θα υλοποιηθεί. Αυτό θα γίνει με προσομοίωση της διαμόρφωσης στο περιβάλλον του MATLAB. Θα ακολουθήσει η χρήση του εργαλείου Simulink όπου με κατάλληλη επιλογή των blocks και με κατάλληλη παραμετροποίηση θα εξάγουμε τα αντίστοιχα αποτελέσματα που θα χρησιμοποιήσουμε ως dataset. Τα δεδομένα αυτά πριν τα χρησιμοποιήσουμε στο κομμάτι του preprocessing θα τα επεξεργαστούμε για να ονοματίσουμε τις στήλες. Πριν φτάσουμε στο σημείο του preprocessing, έχουμε αναπτύξει ένα παράθυρο, όπου επιλέγοντας μεταξύ των 3 διαμορφώσεων αυτή που θέλουμε, επιλέγεται το αντίστοιχο dataset της διαμόρφωσης που επιλέχθηκε. Στη συνέχεια, θα προχωρήσουμε στο preprocessing των δεδομένων του dataset και θα τα χωρίσουμε σε training και testing δεδομένα. Ακολουθεί η επιλογή του μοντέλου και η δημιουργία του με χρήση των κατάλληλων επιπέδων. Το μοντέλο διεξάγει προβλέψεις για τα σήματα που χρησιμοποιήθηκαν για testing. Τέλος, γίνεται η αξιολόγηση του μοντέλου και του δέκτη και εξάγονται στατιστικά σχετικά με την απόδοση και την ακρίβεια τους.

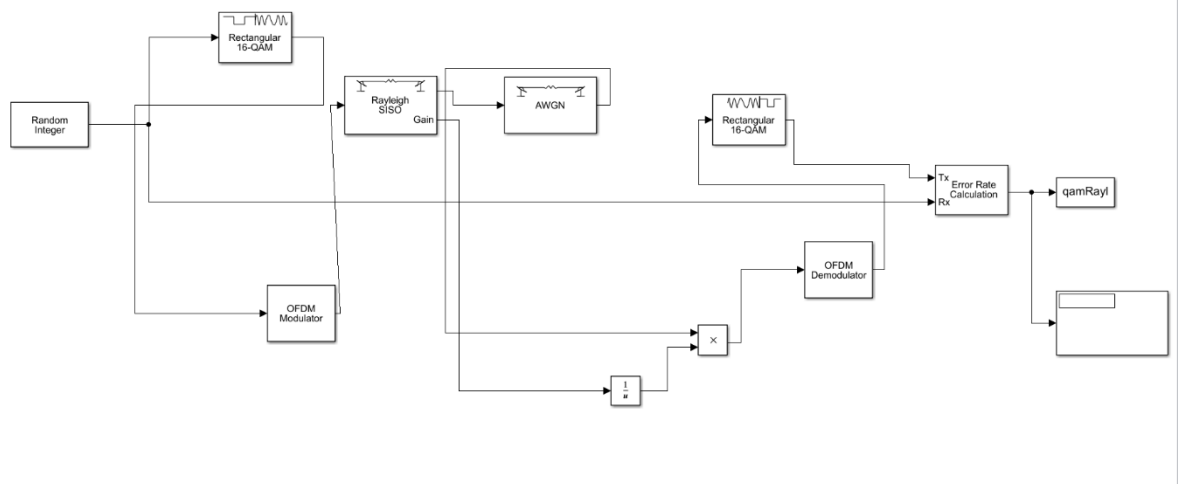
5.4 Διαδικασία της διαμόρφωσης OFDM στο Matlab και δημιουργία του Dataset

Για την πραγματοποίηση της προσομοίωσης ακολουθούμε τα παρακάτω βήματα:

1. Αρχικά χρησιμοποιούμε ένα μπλοκ για την παραγωγή τυχαίων ακέραιων αριθμών με αριθμό δειγμάτων ανά frame 53 καθώς το διάστημα φύλαξης είναι 11 και το μήκος του FFT 64.
2. Στη συνέχεια έχουμε μια 16-QAM διαμόρφωση βασικής ζώνης η οποία ακολουθείται από την OFDM διαμόρφωση του σήματος εισόδου. Αντίστοιχα, έχουμε στην θέση της 16-QAM τις άλλες 2 διαμορφώσεις BPSK και QPSK.

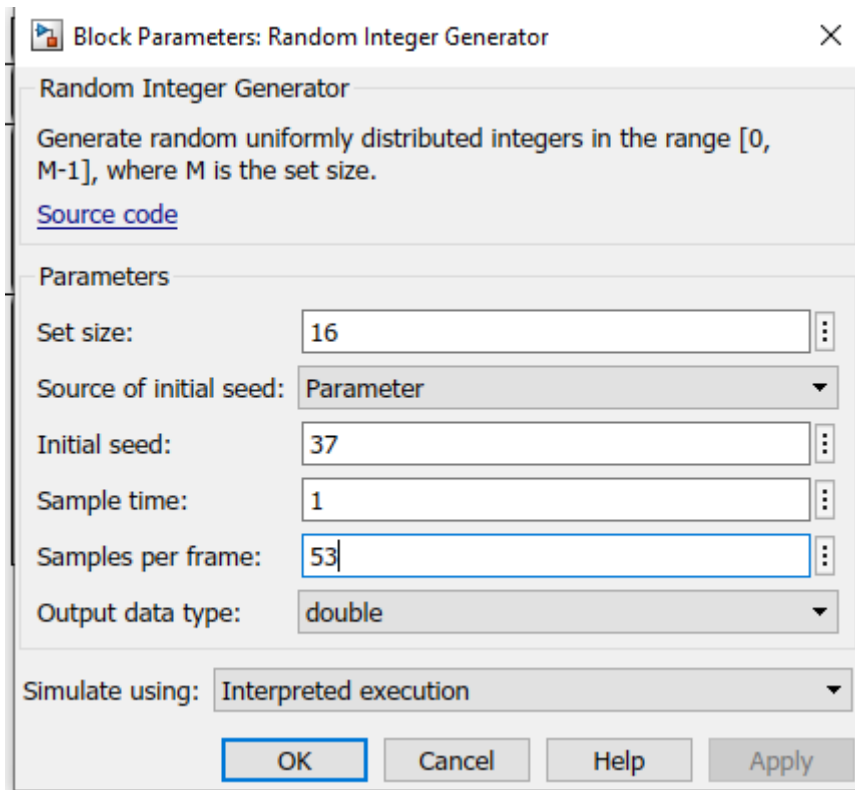
3. Αμέσως μετά έχουμε ένα Rayleigh Fading κανάλι με μέγιστη ολίσθηση Doppler 0.001Hz.
4. Έπειτα, προστίθεται λευκός προσθετικός Gaussian θόρυβος (AWGN) με λόγο Eb/No 25 dB και 4 bits ανά σύμβολο. Αντίστοιχα για τις BPSK και QPSK έχουμε 1 και 2 bits ανά σύμβολο.
5. Μετά πραγματοποιείται αποδιαμόρφωση OFDM και αποδιαμόρφωση βασικής ζώνης 16-QAM.
6. Τέλος, υπολογίζουμε τον ρυθμό σφαλμάτων για να υπολογιστεί το BER (εσφαλμένα bits/συνολικά bits) και αποθηκεύουμε σε ένα πίνακα τα εσφαλμένα bits, το BER και τα συνολικά bits.

Η προσομοίωση θα πραγματοποιηθεί για 100.000 δείγματα με χρήση του εργαλείου Simulink όπου με την χρήση blocks θα δημιουργηθεί το μοντέλο. Η προσομοίωση αφορά την τεχνική διαμόρφωσης OFDM αφού πρώτα πραγματοποιηθεί διαμόρφωση βασικής ζώνης 16-QAM (Quadrature Amplitude Modulation). Στο παρακάτω στιγμιότυπο απεικονίζεται το μοντέλο προσομοίωσης στο Matlab.



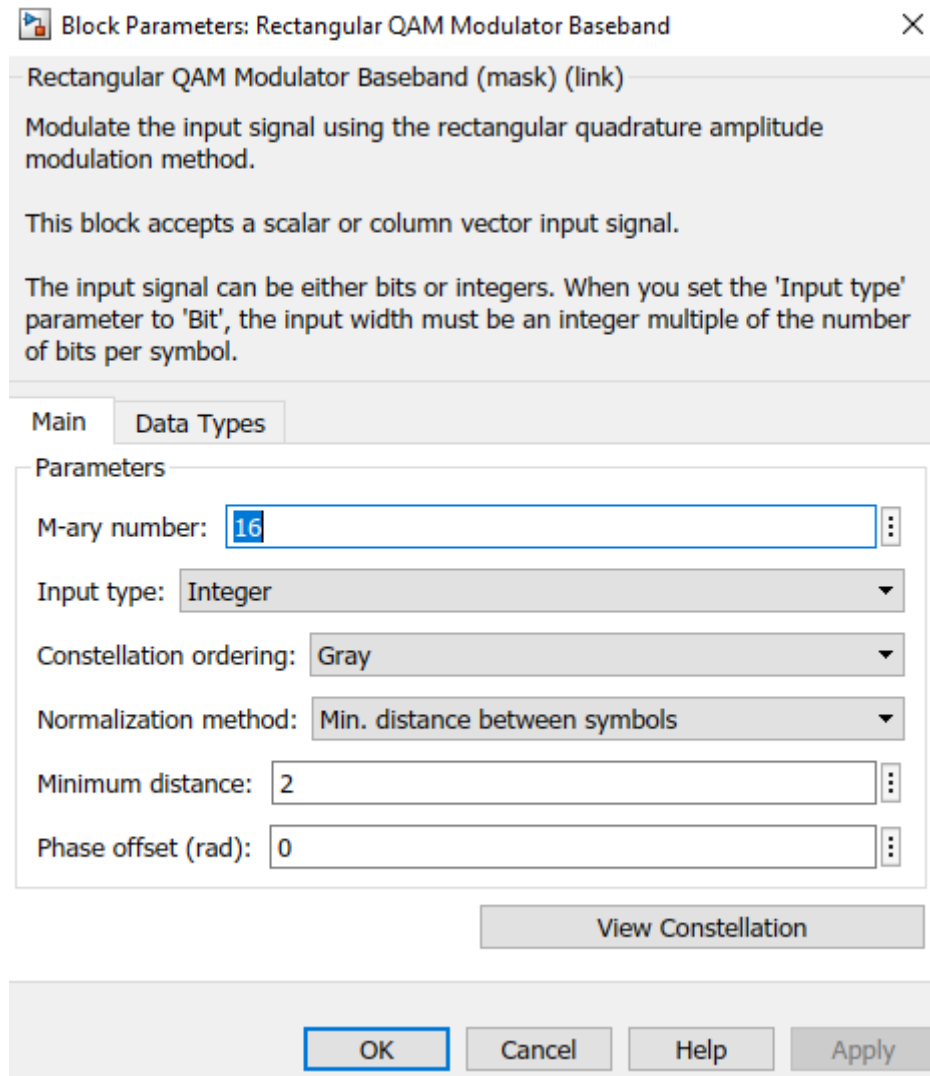
Εικόνα 24 : Μοντέλο 16QAM-OFDM

Για την συγκεκριμένη προσομοίωση παραμετροποιήθηκαν κατάλληλα τα blocks του μοντέλου ως εξής:

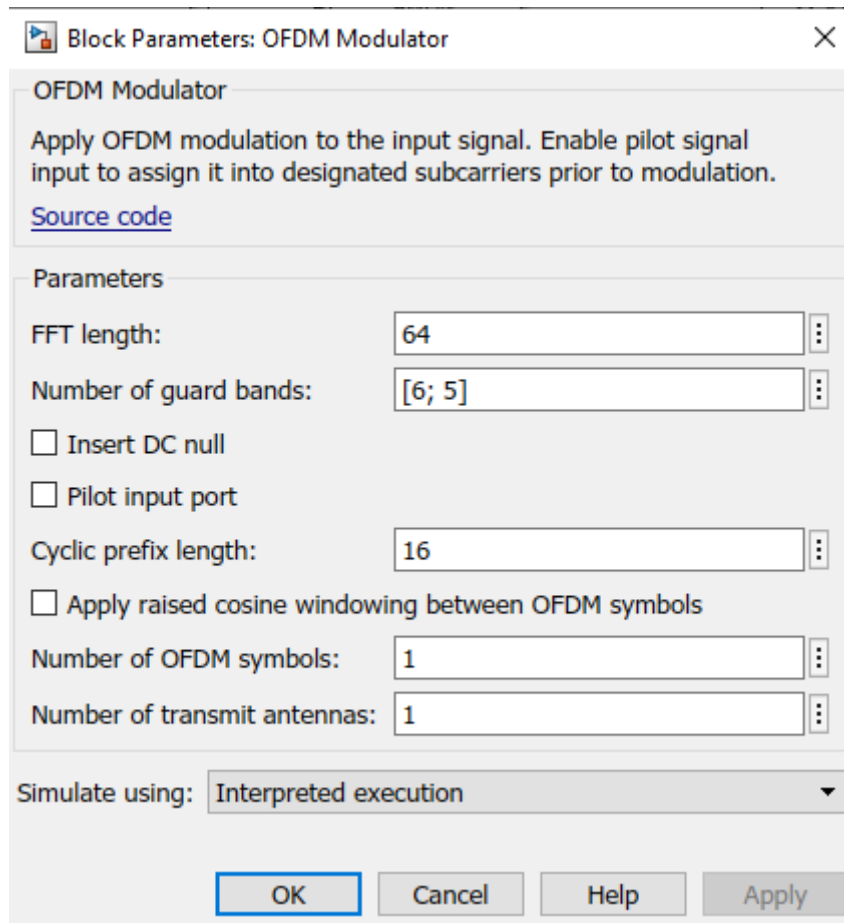


Εικόνα 26 : Random Integer Generator

Ο αριθμός 16 αφορά τον αριθμό των συμβόλων (κάθε σύμβολο έχει 4 bits) και το 53 τα δείγματα ανά frame. Προκύπτει από την αφαίρεση των guard bands από το μήκος του FFT ($64 - 11$).

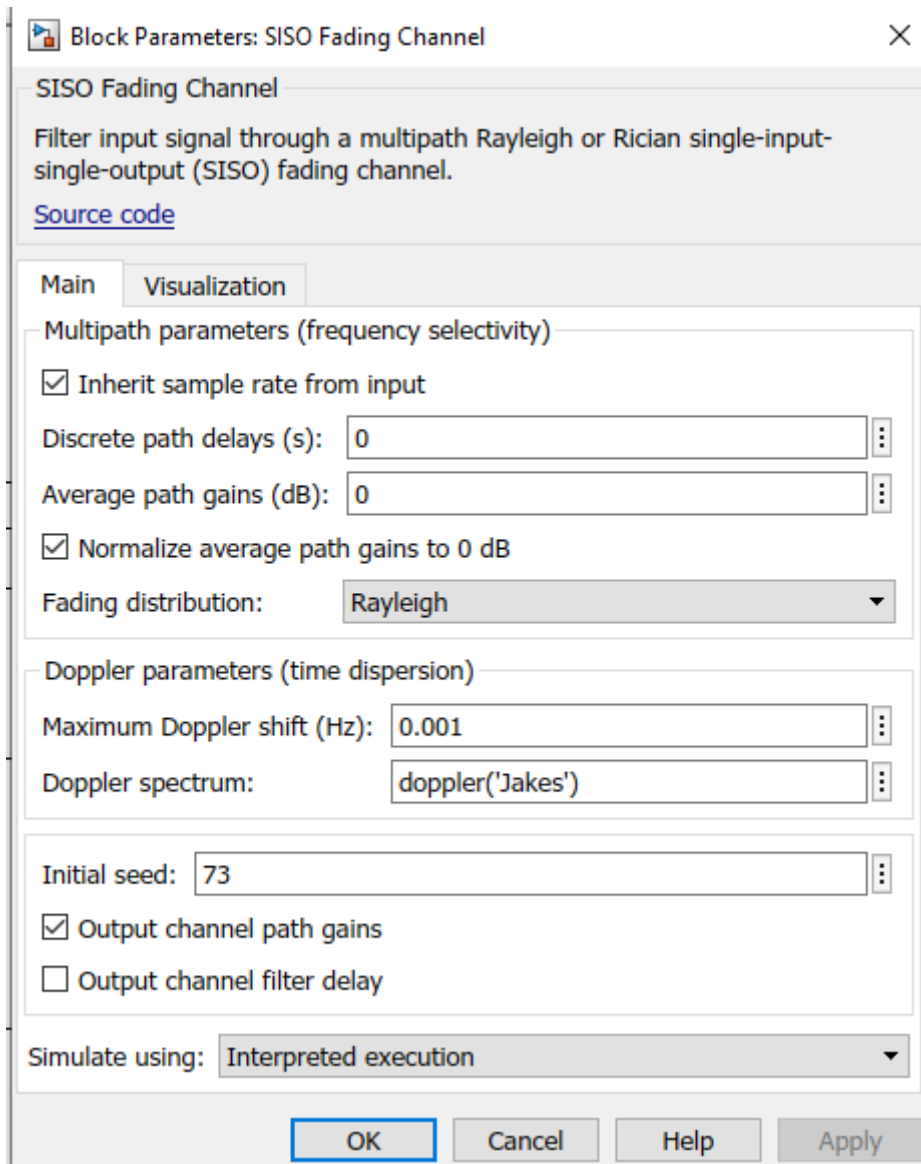


Εικόνα 27 : Rectangular QAM Modulator Baseband



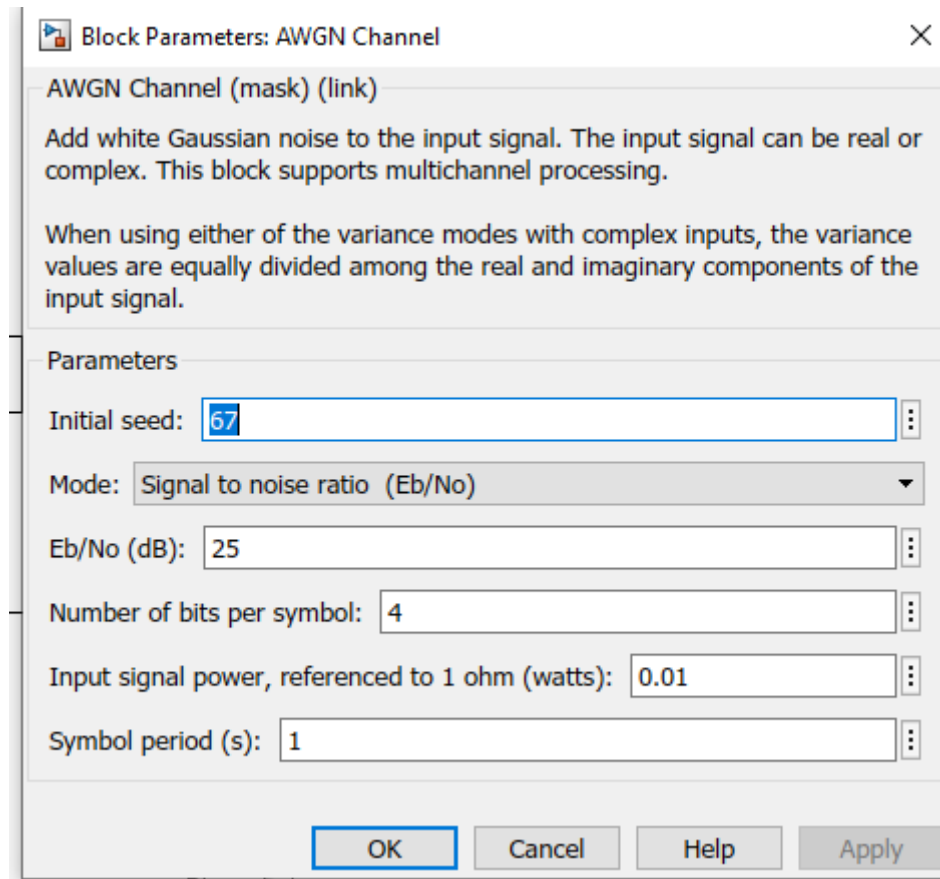
Εικόνα 28 : OFDM Modulator

Αφού παραχθεί το σήμα (οι ακέραιοι από την γεννήτρια) υφίσταται διαμόρφωση 16-QAM βασικής ζώνης όπου έχουμε τα QAM σύμβολα. Έπειτα έχουμε OFDM διαμόρφωση όπου κάθε σύμβολο μεταφέρεται από ένα συγκεκριμένο φέρον. Στη συνέχεια το σήμα γίνεται από ψηφιακό αναλογικό για την μετάδοση του.



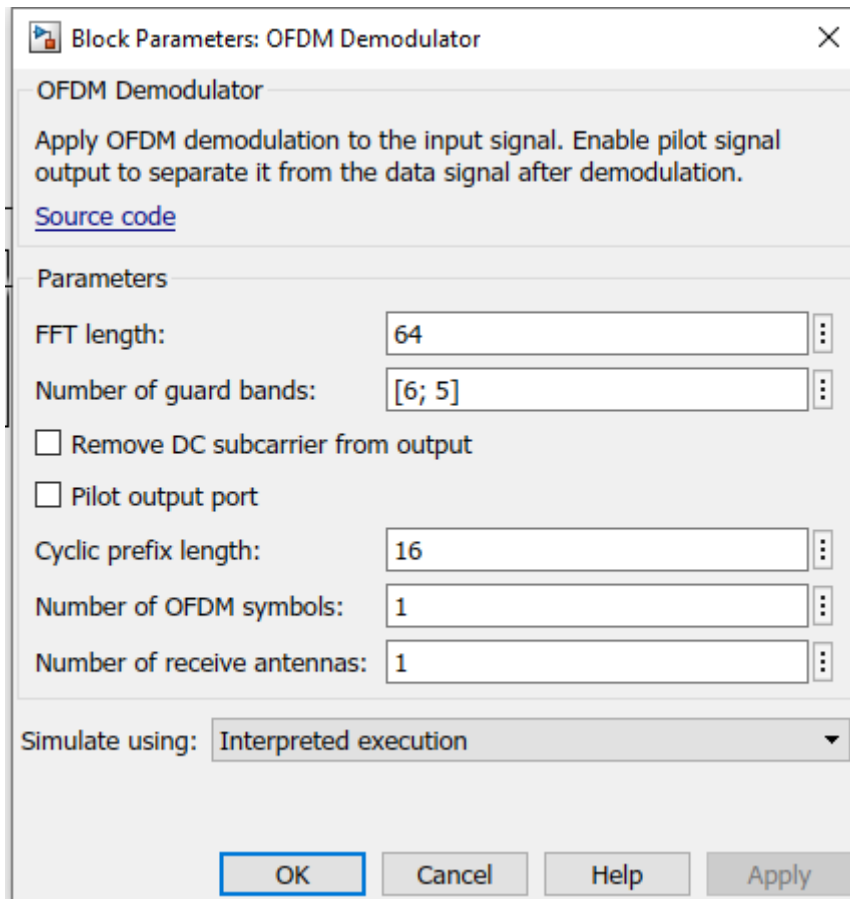
Εικόνα 29 : Rayleigh Fading Channel

Στη συνέχεια το σήμα εισόδου περνάει από ένα Rayleigh Fading κανάλι πολυδιαδρομής όπου κατά την μετάδοση του προστίθεται προσθετικός λευκός γκαουσιανός θόρυβος όπως παρατηρείται παρακάτω.

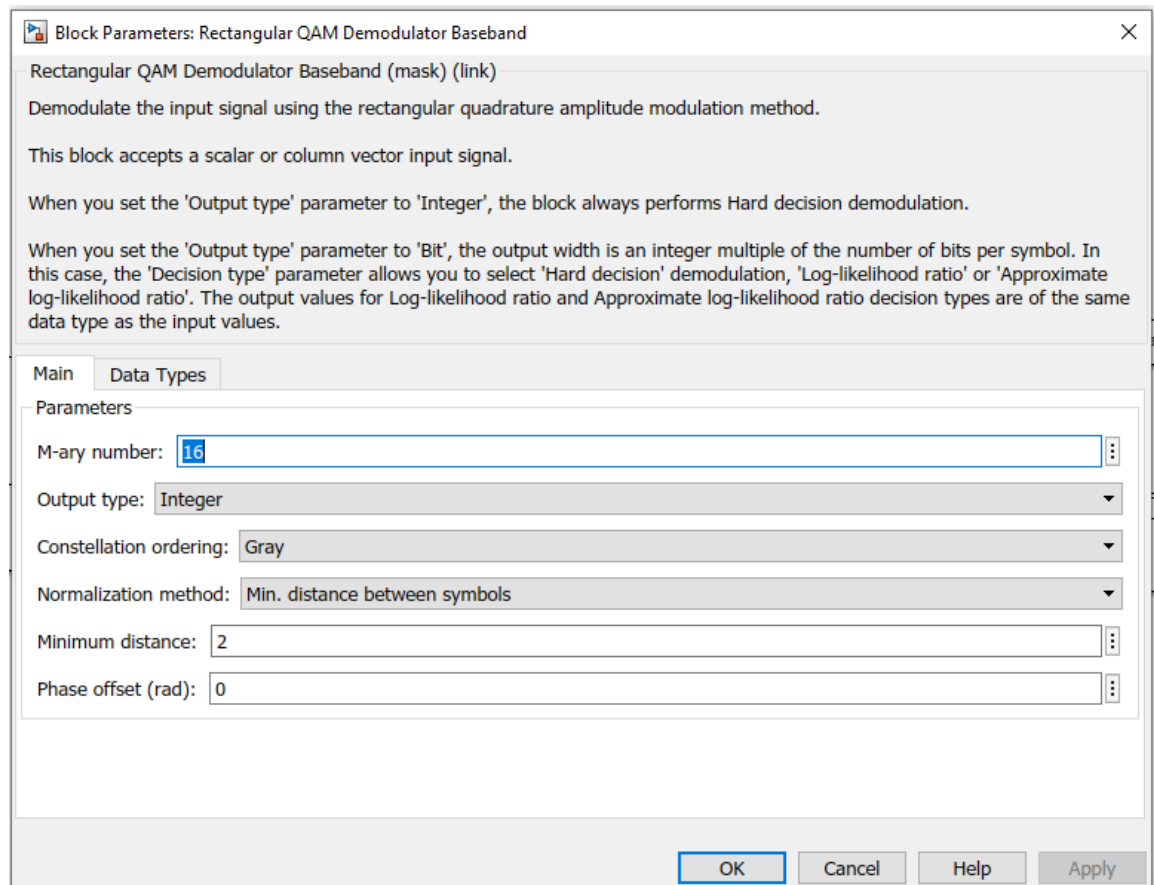


Εικόνα 30 : AWGN Channel

Εδώ γίνεται παραμετροποίηση του λόγου E_b/N_0 (SNR ανά bit) της ενέργειας bit ανά σύμβολο και της φασματικής πυκνότητας του θορύβου. Για την 16-QAM διαμόρφωση έχουμε 4 bits ανά σύμβολο.

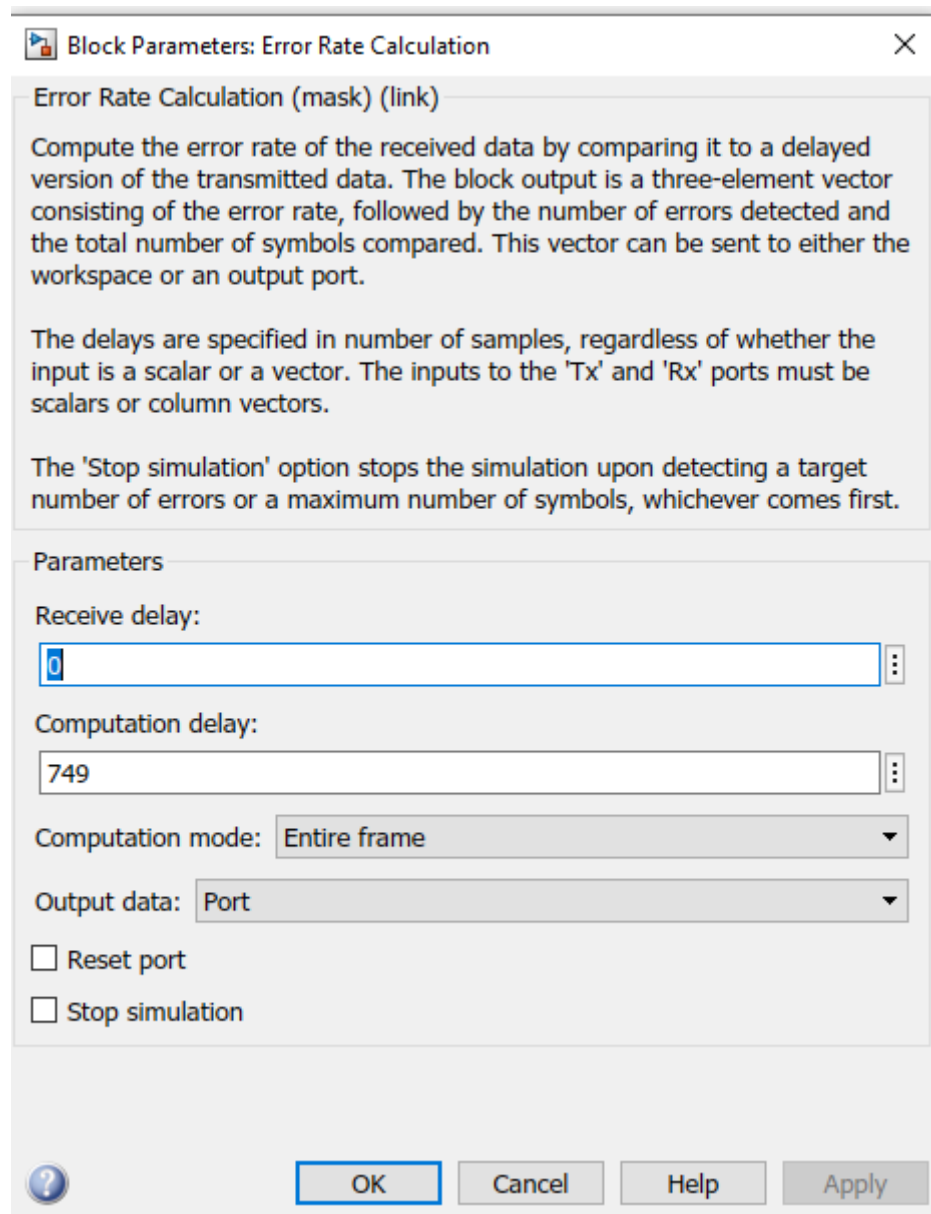


Εικόνα 31 : OFDM Demodulator



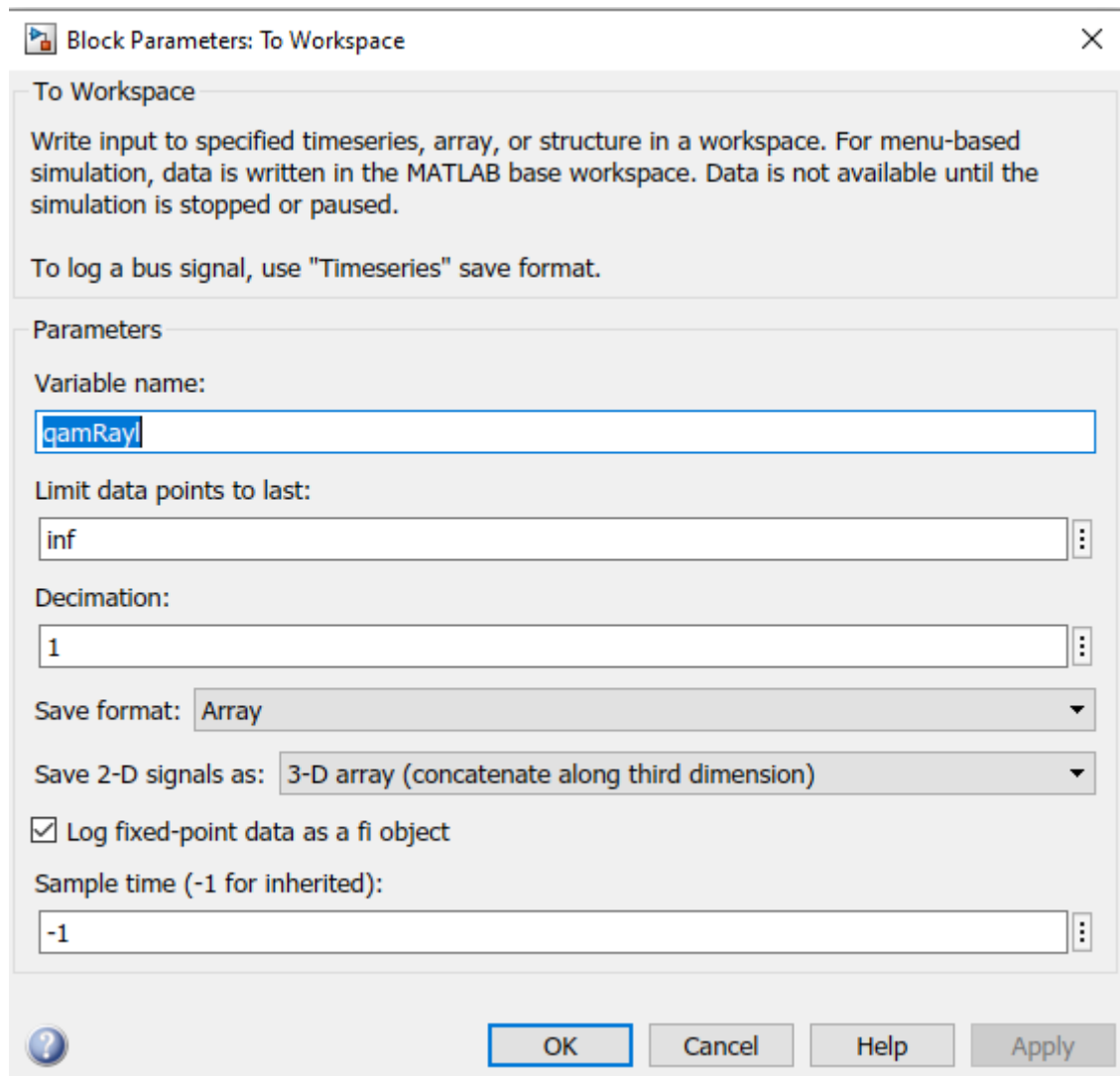
Εικόνα 32: Αποδιαμορφωτής 16-QAM

Εδώ το σήμα εισόδου αφού μετατραπεί από αναλογικό σε ψηφιακό αποδιαμορφώνεται και στην συνέχεια υπολογίζεται ο ρυθμός σφάλματος.



Εικόνα 33 : Error Rate Calculation

Από τον υπολογισμό αυτό εξάγουμε αποτελέσματα που αφορούν την τιμή BER, τα σφάλματα και τα συνολικά σύμβολα που στάλθηκαν.



Εικόνα 34 : To Workspace

Αφού υπολογιστεί το BER, αποθηκεύονται τα αποτελέσματα στο Workspace και ώστε να χρησιμοποιηθεί ως dataset για την εκπαίδευση του μοντέλου. Τα αποτελέσματα από την προσομοίωση εξάγονται σε μορφή δυσδιάστατου πίνακα με 3 στήλες. Για να μπορέσουμε να τα χρησιμοποιήσουμε στο μοντέλο μας αργότερα τα μετατρέπουμε σε csv αρχεία με την παρακάτω εντολή.

```
csvwrite('l6qam-ofdm_ray1_snr25.csv', qamRay1);
```

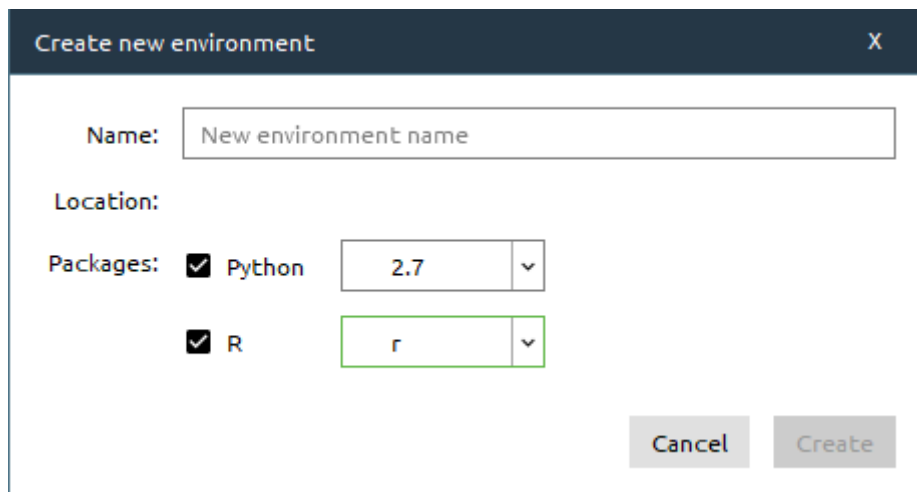
Εικόνα 35 : Μετατροπή σε csv αρχείο

Για την δημιουργία του dataset χρειαστήκαμε 100,000 δείγματα τα οποία για να προκύψουν έγινε πολλαπλασιασμός των δειγμάτων ανά frame με τον αριθμό του dataset μείον 1 καθώς ξεκινάμε από το 0. Για να είναι αξιόπιστο το dataset ήταν θεμιτό να είναι μεγάλος ο όγκος των δειγμάτων.

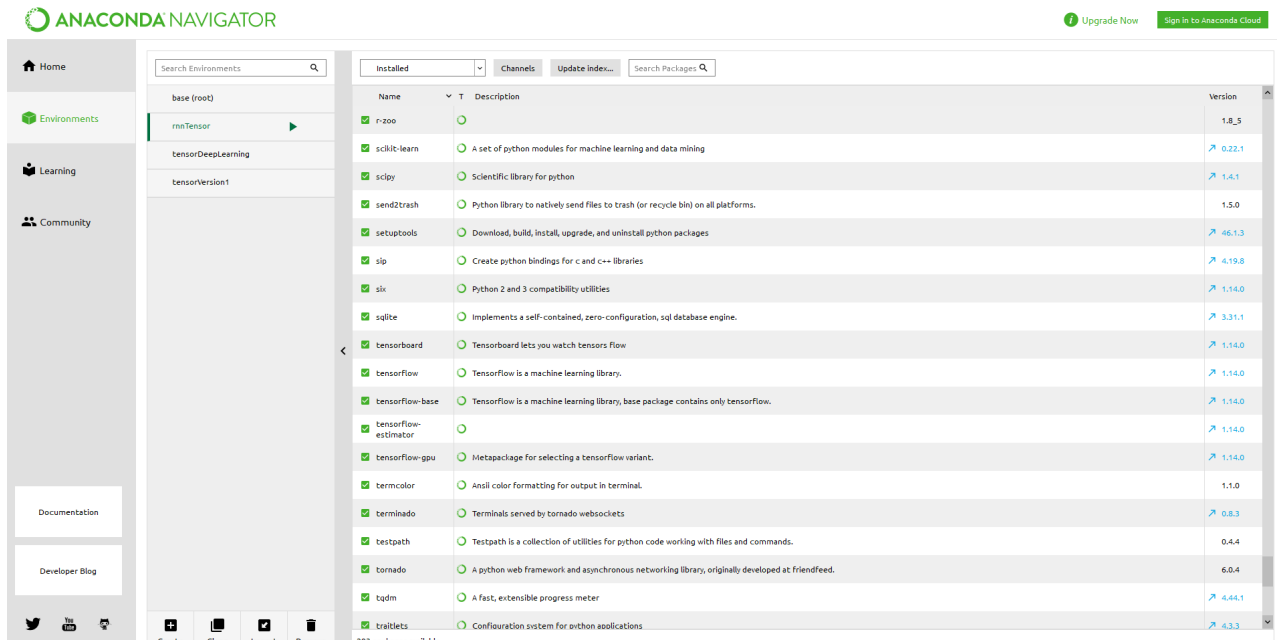
Αντίστοιχη διαδικασία και παραμετροποίηση γίνεται και για την δημιουργία και των άλλων μοντέλων προσομοίωσης (BPSK, QPSK) αντίστοιχα με τις απαιτήσεις της κάθε τεχνικής διαμόρφωσης.

5.5 Ανάπτυξη μοντέλου στο περιβάλλον του PyCharm

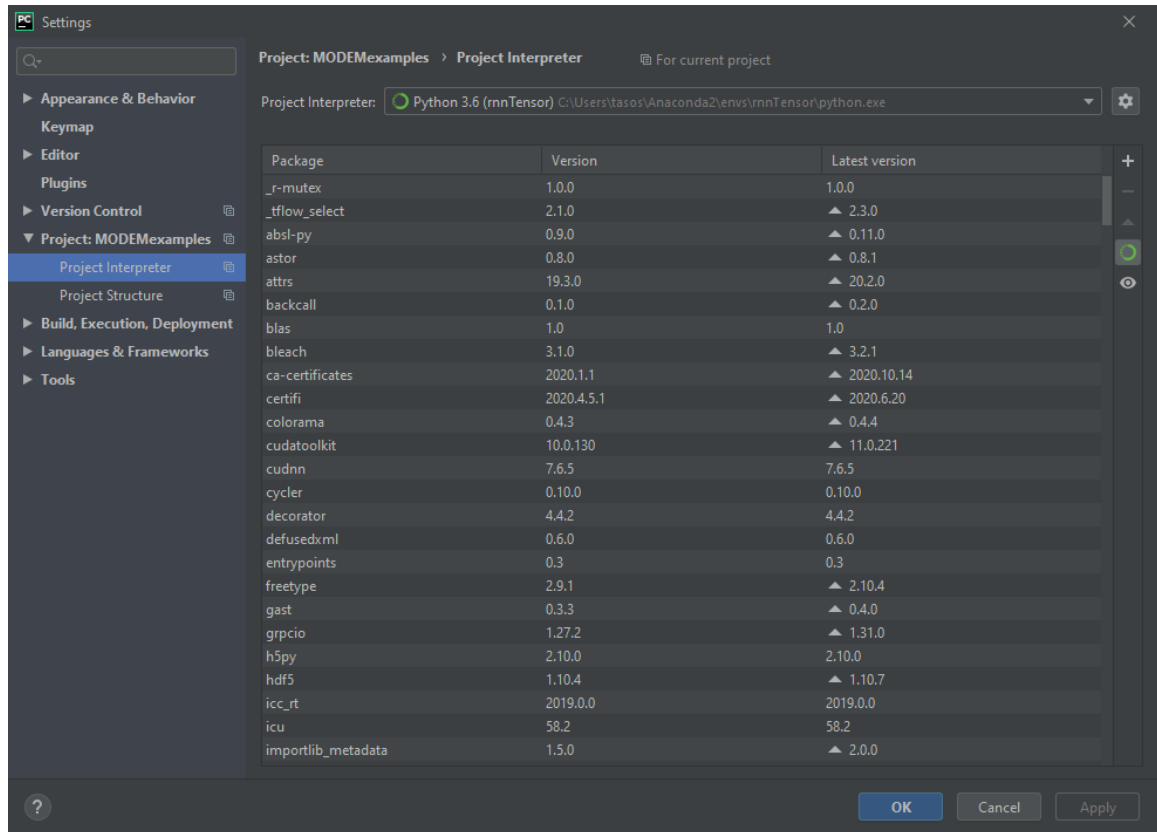
Με χρήση του εργαλείου PyCharm θα γίνει η ανάπτυξη του μοντέλου με την βιβλιοθήκη TensorFlow χρησιμοποιώντας ως γλώσσα προγραμματισμού την Python. Πρώτα, αφού δημιουργήσουμε το project στο PyCharm πρέπει να γίνει από τις ρυθμίσεις προσθήκη του κατάλληλου διερμηνέα για να χρησιμοποιηθεί το TensorFlow και το CUDA. Αυτό θα γίνει με τη βοήθεια του Anaconda Navigator. Θα δημιουργήσουμε εκεί ένα περιβάλλον όπου θα εγκαταστήσουμε τα packages για το TensorFlow και το CUDA. Στην συνέχεια, θα προστεθεί στον διερμηνέα του project όπου θα γράψουμε τον κώδικα για την ανάπτυξη του μοντέλου ώστε να μπορέσουμε να χρησιμοποιήσουμε τις βιβλιοθήκες και τις δυνατότητες που προσφέρει το TensorFlow και το CUDA αντίστοιχα. Στις παρακάτω εικόνες φαίνεται η διαδικασία αυτή.



Εικόνα 36 : Δημιουργία περιβάλλοντος για TensorFlow



Εικόνα 37 : Πακέτα του περιβάλλοντος



Εικόνα 38 : Περιβάλλον διαμορφωμένο για το project

5.6 Επεξεργασία δεδομένων αρχείων csv

Τα δεδομένα που εξάχθηκαν από το MATLAB με την διεξαγωγή των προσομοιώσεων της διαμόρφωσης 16QAM-OFDM, BPSK-OFDM και QPSK-OFDM αντίστοιχα χρειάζεται να επεξεργαστούν. Αναλυτικότερα, επειδή τα δεδομένα είναι στη μορφή στηλών χωρίς ετικέτα (label) που καθορίζει το τι είναι οι αριθμοί της κάθε στήλης, επεξεργαζόμαστε το αρχείο προσθέτοντας τις στήλες : Ber, Errors, Bits (Symbols). Αυτές οι στήλες αφορούν την τιμή BER, τον αριθμό των εσφαλμένων bits και τον συνολικό αριθμό (συμβόλων) bits που στάλθηκαν. Με αυτό τον τρόπο μπορούμε να διαχειριστούμε καλύτερα τα δεδομένα στο κομμάτι του preprocessing, όπου θα προετοιμαστούν για να τροφοδοτήσουν το μοντέλο.

5.7 Preprocessing των δεδομένων από το TensorFlow

Όσον αφορά τα δεδομένα, συνήθως σκεφτόμαστε ορισμένα μεγάλα σύνολα δεδομένων με τεράστιο αριθμό σειρών και στηλών. Αν και αυτό είναι πιθανό σενάριο, δεν συμβαίνει πάντα - τα δεδομένα θα μπορούσαν να είναι σε πολλές διαφορετικές μορφές: Δομημένοι πίνακες, εικόνες, αρχεία ήχου, βίντεο κ.λπ. Τα μηχανήματα δεν καταλαβαίνουν τα ελεύθερα δεδομένα κειμένου, εικόνας ή βίντεο όπως είναι, καταλαβαίνουν 1 και 0. Επομένως, πιθανόν δεν θα είναι αρκετά καλό να κάνουμε μια προβολή διαφανειών όλων των εικόνων μας και να αναμένουμε ότι το μοντέλο μηχανικής μάθησης θα εκπαιδευτεί απλώς από αυτό. Ένα σύνολο δεδομένων μπορεί να θεωρηθεί ως συλλογή αντικειμένων δεδομένων, τα οποία συχνά ονομάζονται επίσης ως αρχεία, σημεία, διανύσματα, μοτίβα, συμβάντα, περιπτώσεις, δείγματα, παρατηρήσεις ή οντότητες. Τα αντικείμενα δεδομένων περιγράφονται από μια σειρά χαρακτηριστικών, που καταγράφουν τα βασικά χαρακτηριστικά ενός αντικειμένου, όπως η μάζα ενός φυσικού αντικειμένου ή ο χρόνος κατά τον οποίο συνέβη ένα συμβάν, κλπ. Τα χαρακτηριστικά γνωρίζονται συχνά ως μεταβλητές, χαρακτηριστικά, πεδία, χαρακτηριστικά ή διαστάσεις.[19]

Τα χαρακτηριστικά μπορεί να είναι:

- I. Κατηγορικά: Χαρακτηριστικά των οποίων οι τιμές λαμβάνονται από ένα καθορισμένο σύνολο τιμών. Για παράδειγμα, ημέρες σε μια εβδομάδα: {Δευτέρα, Τρίτη, Τετάρτη, Πέμπτη, Παρασκευή, Σάββατο, Κυριακή} είναι μια κατηγορία επειδή η τιμή της λαμβάνεται πάντα από αυτό το σύνολο. Ένα άλλο παράδειγμα θα μπορούσε να είναι το Boolean set: {True, False}
- II. Αριθμητικά: Χαρακτηριστικά των οποίων οι τιμές είναι συνεχείς ή ακέραιες τιμές. Αντιπροσωπεύονται από αριθμούς και διαθέτουν τις περισσότερες ιδιότητες των αριθμών. Για παράδειγμα, ο αριθμός των βημάτων που περπατάτε σε μια μέρα ή η ταχύτητα με την οποία οδηγείτε το αυτοκίνητό σας.

Σε αυτό το βήμα επεξεργαζόμαστε το dataset που έχουμε δημιουργήσει με το Matlab. Κατηγοριοποιούμε τις τιμές BER σε δύο κλάσεις δημιουργώντας την στήλη «Target» η οποία αποτελείται από τιμές 0 και 1. Το 1 αφορά τις τιμές BER που είναι πολύ μικρές και άρα αξιόπιστες και το 0 αυτές που δεν είναι με βάση ένα κατώφλι. Αυτό είναι η μέση τιμή του BER όλου dataset. Τέλος, τα κάνουμε shuffle για να γίνει καλύτερα το preprocessing. Στην παρακάτω εικόνα φαίνεται η διαδικασία που αναφέρθηκε.

```
##### PREPROCESSING #####

# Here we find median of BER values
# it didnt work due to decimal problems, i had to ceil down the decimal part for better results
# median = np.median(dataset['Ber'])

avg_sum = 0
for sums in bers:
    avg_sum = avg_sum + float(sums)

# the addition of the second part was made based on experience cause the main purpose was to ceil on the decimal part
avg_ber = (avg_sum/len(bers)) + (5*10**(-5))

# I split the signals to those which have small BER value (for that reason they are reliable)
# and to those which have bigger BER values (unreliable)
# based on the median threshold.

def classify(ber):
    if ber <= avg_ber:
        return 1
    else:
        return 0

main_ds = pd.DataFrame()

if len(main_ds) == 0:
    main_ds = dataset
else:
    main_ds = main_ds.join(dataset)

main_ds['target'] = list(map(classify, main_ds['Ber'])) # Mapping Bers and classify the result to a new column target

# shuffle dataset for better preprocessing
main_ds = main_ds.sample(frac=1)
```

Εικόνα 39 : Preprocessing

Στη συνέχεια τα φέρνουμε στην επιθυμητή μορφή ώστε να αποτελέσουν κατάλληλη είσοδο στο μοντέλο. Τα φέρνουμε δηλαδή στη μορφή ενός tensor και στην συνέχεια τα βάζουμε στην ίδια κλίμακα. Χωρίζουμε τα δεδομένα σε train και test. Με τη χρήση της συνάρτησης `train_test_split()` ορίζουμε το μέγεθος των δεδομένων για την εκπαίδευση και για τον έλεγχο του μοντέλου και βάζουμε μια αυθαίρετη σταθερά για είναι σταθερά τα αποτελέσματα.

```
# we drop the column Ber cause we want the other two columns
# for the input data of our model
main_ds.drop('Ber', 1)
for col in main_ds.columns:
    if col != 'target':
        main_ds[col] = preprocessing.scale(main_ds[col].values) # algorithm puts data in one scale
        main_ds[col].dropna() # drop rows with zeros in ber column

# Save preprocessed dataset to new csv file
main_ds.to_csv('preprocessedData.csv')

# lists for train and test data
X = []
Y = []

x = main_ds.drop('target', 1)
y = main_ds.target

# The following progress is to bring the data to form
# so we can manage them as a tensor with the form of [[][]]
for i in x.values:
    X.append([np.array(i)])

for j in y.values:
    Y.append([np.array(j)])

X = np.array(X)
Y = np.array(Y)

# Split the data of the dataset to 70% training and 30% testing
train_x, test_x, train_y, test_y = train_test_split(X, Y, test_size=0.3, random_state=10)
```

Εικόνα 40 : Δημιουργία του dataset για την εκπαίδευση

5.8 Μοντέλα που χρησιμοποιούνται στις τεχνικές Βαθιάς Μάθησης

Σε αυτό το σημείο πρέπει να επιλέξουμε μεταξύ διάφορων μοντέλων Βαθιάς Μάθησης αυτό που θα χρησιμοποιήσουμε για την προσομοίωση του δέκτη. Υπάρχουν διάφορα είδη μοντέλων. Τα πιο γνωστά είναι τα:

1. Autoencoders
2. Deep Belief Net
3. CNN (Convolutional Neural Networks)
4. RNN (Recurrent Neural Network)
5. Reinforcement Learning to Neural Networks

5.8.1 Autoencoders

Ο αυτόματος κωδικοποιητής είναι ένα τεχνητό νευρωνικό δίκτυο που είναι ικανό να μάθει διάφορα πρότυπα κωδικοποίησης. Η απλή μορφή του αυτόματου κωδικοποιητή είναι ακριβώς όπως το πολυστρωματικό perceptron, που περιέχει ένα επίπεδο εισόδου ή ένα ή περισσότερα κρυφά στρώματα ή ένα επίπεδο εξόδου. Η σημαντική διαφορά μεταξύ του τυπικού πολυστρωματικού perceptron και του τροφοδοτικού νευρικού δικτύου και του αυτόματου κωδικοποιητή είναι στον αριθμό των κόμβων στο επίπεδο εξόδου. Στην περίπτωση του αυτόματου κωδικοποιητή, το επίπεδο εξόδου περιέχει την ίδια ποσότητα κόμβων με το επίπεδο εισόδου. Αντί να προβλέπει τιμές-στόχους σύμφωνα με το διάνυσμα εξόδου, ο αυτόματος κωδικοποιητής πρέπει να προβλέψει τις εισόδους του. [14]

5.8.2 Deep Belief Net

Ένα δίκτυο βαθιάς πεποίθησης είναι μια λύση στο πρόβλημα του χειρισμού μη κυρτών αντικειμενικών συναρτήσεων και τοπικών ελαχίστων κατά τη χρήση του τυπικού πολυστρωματικού perceptron. Είναι ένας εναλλακτικός τύπος βαθιάς μάθησης που αποτελείται από πολλαπλά επίπεδα λανθάνουσας μεταβλητής με σύνδεση μεταξύ των επιπέδων. Το δίκτυο βαθιάς πεποίθησης μπορεί να θεωρηθεί ως περιορισμένες μηχανές Boltzmann (RBM), όπου το κρυφό επίπεδο κάθε υποδικτύου ενεργεί ως ορατό επίπεδο εισόδου για το παρακείμενο επίπεδο του δικτύου. Κάνει το χαμηλότερο ορατό επίπεδο ένα σετ εκπαίδευσης για το παρακείμενο επίπεδο του δικτύου. Με αυτόν τον τρόπο, κάθε στρώμα του δικτύου εκπαιδεύεται ανεξάρτητα και άπληστα. Οι κρυφές μεταβλητές χρησιμοποιούνται ως παρατηρούμενες μεταβλητές για να εκπαιδεύσουν κάθε επίπεδο της βαθιάς δομής.[21]

5.8.3 Convolutional Neural Network

Στη βαθιά μάθηση, ένα συνελικτικό νευρωνικό δίκτυο (CNN ή ConvNet) είναι μια κατηγορία βαθιών νευρωνικών δικτύων, τα οποία συνήθως χρησιμοποιούνται για την ανάλυση οπτικών εικόνων. Έχουν εφαρμογές στην αναγνώριση εικόνας και βίντεο, συστήματα σύστασης, ταξινόμηση εικόνας, ανάλυση ιατρικής εικόνας, επεξεργασία φυσικής γλώσσας, και οικονομικές χρονοσειρές. Τα CNN είναι κανονικές εκδόσεις των πολυεπίπεδων perceptrons. Τα πολυστρωματικά perceptrons συνήθως σημαίνουν πλήρως συνδεδεμένα δίκτυα, δηλαδή, κάθε νευρώνας σε ένα στρώμα συνδέεται με όλους τους νευρώνες στο επόμενο στρώμα. Τυπικοί τρόποι κανονικοποίησης περιλαμβάνουν την προσθήκη κάποιας μορφής μέτρησης βάρους στη λειτουργία απώλειας. Τα CNN υιοθετούν μια διαφορετική προσέγγιση προς την κανονικοποίηση: εκμεταλλεύονται το ιεραρχικό μοτίβο στα δεδομένα και συγκεντρώνουν πιο περίπλοκα μοτίβα χρησιμοποιώντας μικρότερα και απλούστερα μοτίβα. Επομένως, στην κλίμακα της σύνδεσης και της πολυπλοκότητας, τα CNN βρίσκονται στο κάτω άκρο. [13]

5.8.4 Recurrent Neural Networks

Ένα επαναλαμβανόμενο νευρωνικό δίκτυο (RNN) είναι μια κατηγορία τεχνητών νευρικών δικτύων όπου οι συνδέσεις μεταξύ κόμβων σχηματίζουν ένα κατευθυνόμενο γράφημα κατά μήκος μιας χρονικής ακολουθίας. Αυτό του επιτρέπει να εμφανίζει χρονική δυναμική συμπεριφορά. Προερχόμενα από τροφοδοτικά (feedforward) νευρωνικά δίκτυα, τα RNN μπορούν να χρησιμοποιήσουν την εσωτερική τους κατάσταση (μνήμη) για την επεξεργασία ακολουθιών μεταβλητού μήκους εισόδων. Αυτό τα καθιστά εφαρμόσιμα σε εργασίες όπως η μη καταχωρισμένη, συνδεδεμένη αναγνώριση γραφής ή η αναγνώριση ομιλίας. Ο όρος «επαναλαμβανόμενο νευρωνικό δίκτυο» χρησιμοποιείται αδιάκριτα για να αναφέρεται σε δύο ευρείες κατηγορίες δικτύων με παρόμοια γενική δομή, όπου η μία είναι πεπερασμένη ώθηση και η άλλη είναι άπειρη ώθηση. Και οι δύο κατηγορίες δικτύων εμφανίζουν χρονική δυναμική συμπεριφορά. Ένα περιοδικό δίκτυο πεπερασμένων παλμών είναι ένα κατευθυνόμενο ακυκλικό γράφημα που μπορεί να ξετυλιχθεί και να αντικατασταθεί με ένα αυστηρά τροφοδοτούμενο νευρωνικό δίκτυο, ενώ ένα άπειρο επαναλαμβανόμενο δίκτυο παλμών είναι ένα κατευθυνόμενο κυκλικό γράφημα που δεν μπορεί να ξετυλιχθεί. Τα επαναλαμβανόμενα δίκτυα πεπερασμένης ώθησης και απεριόριστης ώθησης μπορούν να έχουν επιπλέον αποθηκευμένες καταστάσεις και η αποθήκευση μπορεί να τεθεί υπό άμεσο έλεγχο από το νευρωνικό δίκτυο. Ο χώρος αποθήκευσης μπορεί επίσης να αντικατασταθεί από άλλο δίκτυο ή γράφημα, εάν αυτό περιλαμβάνει χρονικές καθυστερήσεις ή έχει βρόχους ανατροφοδότησης. Τέτοιες ελεγχόμενες καταστάσεις αναφέρονται ως περιφραγμένη κατάσταση ή περιφραγμένη μνήμη, και αποτελούν μέρος μακροχρόνιων δικτύων βραχυπρόθεσμης μνήμης (LSTM) και επαναλαμβανόμενων μονάδων με πύλη. Αυτό ονομάζεται επίσης Feedback Neural Network.[21]

5.8.4.1 Long Short Term Memory

Τα δίκτυα LSTM εφευρέθηκαν από τους Hochreiter και Schmidhuber το 1997 και έθεσαν αρχεία ακρίβειας σε πολλούς τομείς εφαρμογών. Γύρω στο 2007, το LSTM άρχισε να φέρνει επανάσταση στην αναγνώριση ομιλίας, ξεπερνώντας τα παραδοσιακά μοντέλα σε ορισμένες εφαρμογές ομιλίας. Το 2009, ένα δίκτυο LSTM με κατάρτιση Connectionist Temporal Classification (CTC) ήταν το πρώτο RNN που κέρδισε διαγωνισμούς αναγνώρισης μοτίβων όταν κέρδισε αρκετούς διαγωνισμούς για την αναγνώριση συνδεδεμένης γραφής. Το 2014, ο κινεζικός γίγαντας αναζήτησης Baidu χρησιμοποίησε RNN εκπαιδευμένα σε CTC για να σπάσει το σημείο αναφοράς αναγνώρισης ομιλίας Switchboard Hub5'00 χωρίς να χρησιμοποιήσει παραδοσιακές μεθόδους επεξεργασίας ομιλίας. Το LSTM βελτίωσε επίσης την αναγνώριση ομιλίας μεγάλου λεξιλογίου και τη σύνθεση κειμένου σε ομιλία και χρησιμοποιήθηκε στο Google Android. Το 2015, η αναγνώριση ομιλίας της Google παρουσίασε ένα δραματικό άλμα 49% μέσω του εκπαιδευμένου CTC LSTM, το οποίο χρησιμοποιήθηκε από τη φωνητική αναζήτηση της Google. Η LSTM έσπασε ρεκόρ για βελτιωμένη αυτόματη μετάφραση, Language Modeling και Multilingual Language

Processing. Το LSTM σε συνδυασμό με συνελκτικά νευρωνικά δίκτυα (CNN) βελτίωσαν την αυτόματη καταγραφή εικόνας.[21]

5.8.5 Ενισχυτική Μάθηση στα νευρωνικά δίκτυα

Η εκμάθηση ενίσχυσης είναι ένα είδος υβριδισμού δυναμικού προγραμματισμού και εποπτευόμενης μάθησης. Τυπικά στοιχεία της προσέγγισης είναι το περιβάλλον, ο πράκτορας, οι ενέργειες, η πολιτική και οι λειτουργίες κόστους. Ο πράκτορας ενεργεί ως ελεγκτής του συστήματος. Η πολιτική καθορίζει τις ενέργειες που πρέπει να αναληφθούν και η συνάρτηση ανταμοιβής καθορίζει τον συνολικό στόχο του μαθησιακού προβλήματος ενίσχυσης. Ένας πράκτορας, που λαμβάνει τη μέγιστη δυνατή ανταμοιβή, μπορεί να θεωρηθεί ότι εκτελεί την καλύτερη ενέργεια για μια δεδομένη κατάσταση. Εδώ, ένας πράκτορας αναφέρεται σε μια αφηρημένη οντότητα, είτε αντικείμενο είτε υποκείμενο (αυτόνομα αυτοκίνητα, ρομπότ, άνθρωποι, chatbots υποστήριξης πελατών κ.λπ.), η οποία εκτελεί ενέργειες. Η κατάσταση ενός πράκτορα αναφέρεται στη θέση και την κατάσταση του να βρίσκεται στο αφηρημένο περιβάλλον του. Για παράδειγμα, μια συγκεκριμένη θέση σε έναν κόσμο εικονικής πραγματικότητας, ένα κτίριο, μια σκακιέρα ή τη θέση και την ταχύτητα σε μια πίστα. Η βαθιά μάθηση ενίσχυσης υπόσχεται μια πολύ γενικευμένη διαδικασία μάθησης που μπορεί να μάθει χρήσιμη συμπεριφορά με πολύ λίγη ανατροφοδότηση. Είναι ένας συναρπαστικός και απαιτητικός τομέας, ο οποίος αναμφίβολα θα είναι ουσιαστικό μέρος του μελλοντικού ΑΙ τοπίου.[21]

5.9 Επιλογή κατάλληλου μοντέλου

Για την παρούσα διπλωματική επιλέχθηκε ένα είδος μοντέλου Sequential, το οποίο είναι ένα διαδοχικό μοντέλο κατάλληλο για μια απλή στοίβα στρωμάτων όπου κάθε στρώμα έχει ακριβώς ένα tensor εισόδου και ένα tensor εξόδου. Πιο συγκεκριμένα χρησιμοποιήθηκε ένα RNN μοντέλο το οποίο είναι το LSTM καθώς προσφέρει μεγαλύτερη ακρίβεια.

Αξίζει να σημειωθεί ότι το μοντέλο που χρησιμοποιήθηκε είχε Sequence to Vector αρχιτεκτονική όπου ως είσοδο έχουμε τις τιμές του BER και ως έξοδο το αν είναι αξιόπιστος ο δέκτης και το αν έγινε σωστή πρόβλεψη για την αξιοπιστία του σήματος.

Όσον αφορά τα επίπεδα, προστέθηκαν ένα επίπεδο LSTM, ένα Dropout, ένα BatchNormalization και ένα Dense.

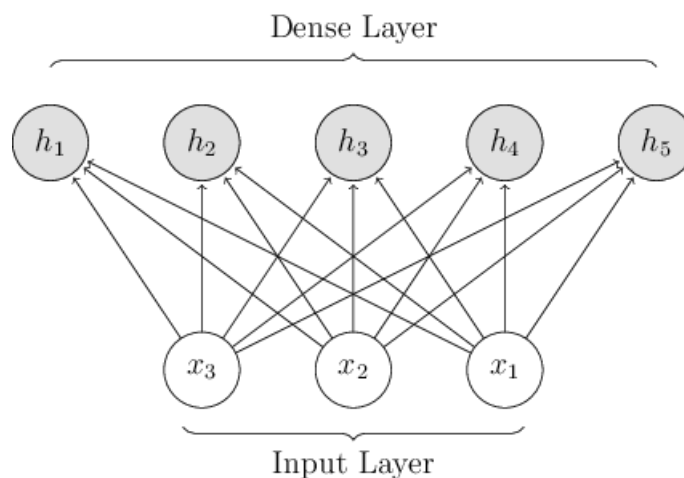
- I. Dropout: Είναι μια μέθοδος κανονικοποίησης της οποίας η βασική ιδέα είναι να γίνει τυχαία ρίψη μονάδων (μαζί με τις συνδέσεις τους) από το νευρωνικό δίκτυο κατά τη διάρκεια της εκπαίδευσης. Αυτό εμποδίζει την υπερβολική προσαρμογή των μονάδων. Κατά τη διάρκεια της εκπαίδευσης, έχουμε Dropout δείγματα από εκθετικό αριθμό διαφορετικών «αραιωμένων» δικτύων. Κατά τη διάρκεια της δοκιμής, είναι εύκολο να προσεγγίσουμε το αποτέλεσμα του

μέσου όρου των προβλέψεων όλων αυτών των αραιωμένων δικτύων χρησιμοποιώντας απλά ένα απλό δίκτυο που έχει μικρότερα βάρη. Αυτό μειώνει σημαντικά την υπερβολική τοποθέτηση και δίνει σημαντικές βελτιώσεις σε σχέση με άλλες μεθόδους κανονικοποίησης. Το Dropout έχει αποδειχθεί ότι βελτιώνει την απόδοση των νευρωνικών δικτύων σε εποπτευόμενες μαθησιακές εργασίες στην όραση, την αναγνώριση ομιλίας, την ταξινόμηση εγγράφων και την υπολογιστική βιολογία, επιτυγχάνοντας αποτελέσματα τελευταίας τεχνολογίας σε πολλά σύνολα δεδομένων αναφοράς.

- II. BatchNormalization: Κατά τη διάρκεια της οπίσθιας διάδοσης, αυτά τα φαινόμενα προκαλούν απόσπαση της προσοχής στις κλίσεις, που σημαίνει ότι οι κλίσεις πρέπει να αντισταθμίσουν τα ακραία επίπεδα, πριν μάθουν τα βάρη για την παραγωγή απαιτούμενων εξόδων. Η ομαλοποίηση παρτίδας κανονικοποιεί αυτές τις κλίσεις από την απόσπαση της προσοχής στα ακραία επίπεδα και ρέει προς τον κοινό στόχο (ομαλοποιώντας τα) εντός ενός εύρους μικρο-παρτίδας.

Πρόβλημα μαθησιακού ρυθμού: Γενικά, τα ποσοστά εκμάθησης διατηρούνται μικρά, έτσι ώστε μόνο ένα μικρό μέρος των βαθμίδων να διορθώνει τα βάρη, ο λόγος είναι ότι οι διαβαθμίσεις για τις ακραίες ενεργοποιήσεις δεν πρέπει να επηρεάζουν τις ενεργοποιημένες ενεργοποιήσεις. Με την ομαλοποίηση κατά παρτίδες, αυτές οι ακραίες ενεργοποιήσεις μειώνονται και ως εκ τούτου υψηλότερα ποσοστά μάθησης μπορούν να χρησιμοποιηθούν για την επιτάχυνση της μαθησιακής διαδικασίας.

- III. Dense: Ένα πυκνό στρώμα είναι απλώς ένα κανονικό στρώμα νευρώνων σε ένα νευρωνικό δίκτυο. Κάθε νευρώνας λαμβάνει είσοδο από όλους τους νευρώνες στο προηγούμενο στρώμα, συνδέοντας έτσι πυκνά. Παρακάτω φαίνεται ένα παράδειγμα ενός Dense επιπέδου.



Εικόνα 41 : Dense επίπεδο

```
model = Sequential()

# LSTM model with given input shape
model.add(CuDNNLSTM(128, input_shape=X.shape[1:], return_sequences=True))
model.add(Dropout(0.2))
model.add(BatchNormalization())

model.add(CuDNNLSTM(128, return_sequences=True))
model.add(Dropout(0.2))
model.add(BatchNormalization())

model.add(CuDNNLSTM(128))
model.add(Dropout(0.2))
model.add(BatchNormalization())

model.add(Dense(32, activation='relu'))
model.add(Dropout(0.2))

model.add(Dense(32, activation='relu'))

# 2 neurons for output layer afou exw 0 or 1
model.add(Dense(2, activation='softmax')) # when categorical_crossentropy
```

Εικόνα 42 : Επίπεδα μοντέλου που επιλέχθηκε

Επίσης, ως optimizer για το μοντέλο χρησιμοποιήθηκε ο “Adams” ο οποίος είναι ένας αλγόριθμος βελτιστοποίησης ο οποίος ενημερώνει τα βάρη του νευρωνικού δικτύου επαναληπτικά με βάση τα δεδομένα εκπαίδευσης. Επίσης, είναι ο πιο ευρέως χρησιμοποιούμενος. Ως loss χρησιμοποιήθηκε το “sparse_categorical_crossentropy”, που σημαίνει ότι κάθε είσοδος δεδομένων μπορεί να ανήκει σε μία κλάση. Δηλαδή είναι αυτό που θέλουμε εδώ καθώς έχουμε 2 κλάσεις (0 και 1).

```
opt = tensorflow.keras.optimizers.Adam(lr=0.001, decay=1e-6)
```

Εικόνα 43 : Optimizer

```
model.compile(loss='sparse_categorical_crossentropy', optimizer=opt, metrics=['accuracy'])
```

Εικόνα 44 : loss

5.10 Activation Function

Την χρησιμοποιούμε για να πάρουμε την έξοδο ενός κόμβου. Είναι επίσης γνωστή ως συνάρτηση μεταφοράς.

- ReLU: Το Rectified Linear Unit είναι η πιο συχνά χρησιμοποιούμενη λειτουργία ενεργοποίησης στον κόσμο αυτή τη στιγμή δεδομένου ότι χρησιμοποιείται σε σχεδόν όλα τα συνελκτικά νευρωνικά δίκτυα ή τη βαθιά μάθηση. Όπως μπορείτε να δείτε, το ReLU είναι μισό ανορθωμένο (από κάτω). Το $f(z)$ είναι μηδέν όταν το z είναι μικρότερο από το μηδέν και το $f(z)$ είναι ίσο με το z όταν το z είναι πάνω ή ίσο με το μηδέν.

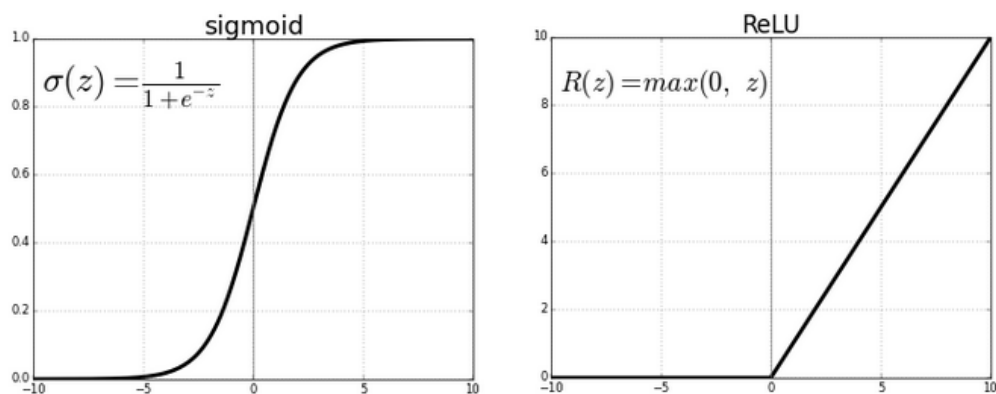


Fig: ReLU v/s Logistic Sigmoid

Εικόνα 45 : ReLu vs Sigmoid

Εύρος: $[0 \text{ έως το } \infty)$

Αλλά το ζήτημα είναι ότι όλες οι αρνητικές τιμές γίνονται αμέσως μηδενικές, γεγονός που μειώνει την ικανότητα του μοντέλου να ταιριάζει ή να εκπαιδεύεται σωστά από τα δεδομένα. Αυτό σημαίνει ότι οποιαδήποτε αρνητική είσοδος που δίνεται στη συνάρτηση ενεργοποίησης ReLU μετατρέπει την τιμή σε μηδέν αμέσως στο γράφημα, το οποίο με τη σειρά του επηρεάζει το γράφημα που προκύπτει μη αντιστοιχίζοντας κατάλληλα τις αρνητικές τιμές. Στην προκειμένη περίπτωση δεν έχουμε αρνητικές τιμές στο dataset οπότε είναι η κατάλληλη συνάρτηση μεταφοράς για το μοντέλο μας.

- Softmax: Η συνάρτηση softmax συμπιέζει τις εξόδους κάθε μονάδας μεταξύ 0 και 1, όπως και η λειτουργία σιγμοειδούς. Αλλά διαιρεί επίσης κάθε έξοδο έτσι ώστε το συνολικό άθροισμα των εξόδων να είναι ίσο με 1 όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 46 : Softmax

Η έξοδος της λειτουργίας softmax, η οποία είναι ισοδύναμη με μια κατηγορική κατανομή πιθανότητας, μας λέει την πιθανότητα ότι οποιαδήποτε από τις τάξεις είναι αληθής.

Μαθηματικά η λειτουργία softmax φαίνεται παρακάτω, όπου το z είναι ένα διάνυσμα των εισόδων στο επίπεδο εξόδου (εάν έχουμε 10 μονάδες εξόδου, τότε υπάρχουν 10 στοιχεία στο z). Και πάλι, το j ευρετηριάζει τις μονάδες εξόδου, έτσι $j = 1, 2, \dots, K$.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad (33)$$

5.11 Αξιολόγηση και Προβλέψεις

Αφού επιλεγεί το μοντέλο, στην προκειμένη περίπτωση το LSTM, θα πρέπει να γίνει αξιολόγηση με τη χρήση των δεδομένων που έχουμε για testing για τα αποτελέσματα των προβλέψεων του μοντέλου. Κάθε φορά που το μοντέλο κάνει μια σωστή πρόβλεψη θεωρείται επιτυχία και άρα προσδίδει στην ακρίβεια του μοντέλου. Πάντα θα υπάρχει και ένα μικρό ποσοστό λάθους, αλλά το σημαντικό είναι η ακρίβεια του μοντέλου να είναι κοντά στο ένα και σίγουρα αρκετά μεγαλύτερη από το 0.5. Αν το μοντέλο έχει τέτοια ακρίβεια θεωρείται ότι παρέχει ακρίβεια και αξιοπιστία. Διατρέχοντας αρκετές φορές το dataset βοηθάει στο να εκπαιδευτεί καλύτερα το μοντέλο και να επιτύχει ακολούθως μεγαλύτερη ακρίβεια. Αυτό γίνεται στη ρύθμιση με τη χρήση του “epoch”. Επιπλέον, χρησιμοποιώντας το “model.predict()” μπορούμε να πάρουμε τις προβλέψεις που έκανε το μοντέλο και να δούμε ποιες ήταν σωστές και ποιες όχι για διαγνωστικούς σκοπούς. Αυτό φαίνεται στην παρακάτω εικόνα.

```
84 predictions = model.predict(test_x)
85
86 for i in range(len(predictions)):
87     print('Test:{} Prediction:{} '.format(test_y[i], np.argmax(predictions[i])))
88
```

Εικόνα 47 : Εκτύπωση προβλέψεων

Όπως θα φανεί στο παρακάτω στιγμιότυπο τα αποτελέσματα από τις προβλέψεις είναι 0 ή 1 (κατηγοριοποίηση που έγινε για την καλύτερη εκπαίδευση του μοντέλου). Το 1 σημαίνει ότι το BER είναι μικρότερο από το κατώφλι που θέσαμε, το οποίο είναι η μέση τιμή του BER των συνολικών δειγμάτων του dataset, ώστε να είναι αξιόπιστος ο δέκτης ενώ το 0 ότι είναι μεγαλύτερο αυτού. Σε γενικές γραμμές βλέπουμε ότι με χρήση του OFDM και του κατάλληλου μοντέλου ο δέκτης είναι αρκετά αξιόπιστος.

```
Test:[0] Prediction:0
Test:[0] Prediction:0
Test:[1] Prediction:1
Test:[0] Prediction:0
Test:[0] Prediction:0
Test:[0] Prediction:0
Test:[0] Prediction:0
Test:[0] Prediction:0
Test:[0] Prediction:0
Test:[1] Prediction:1
Test:[0] Prediction:0
```

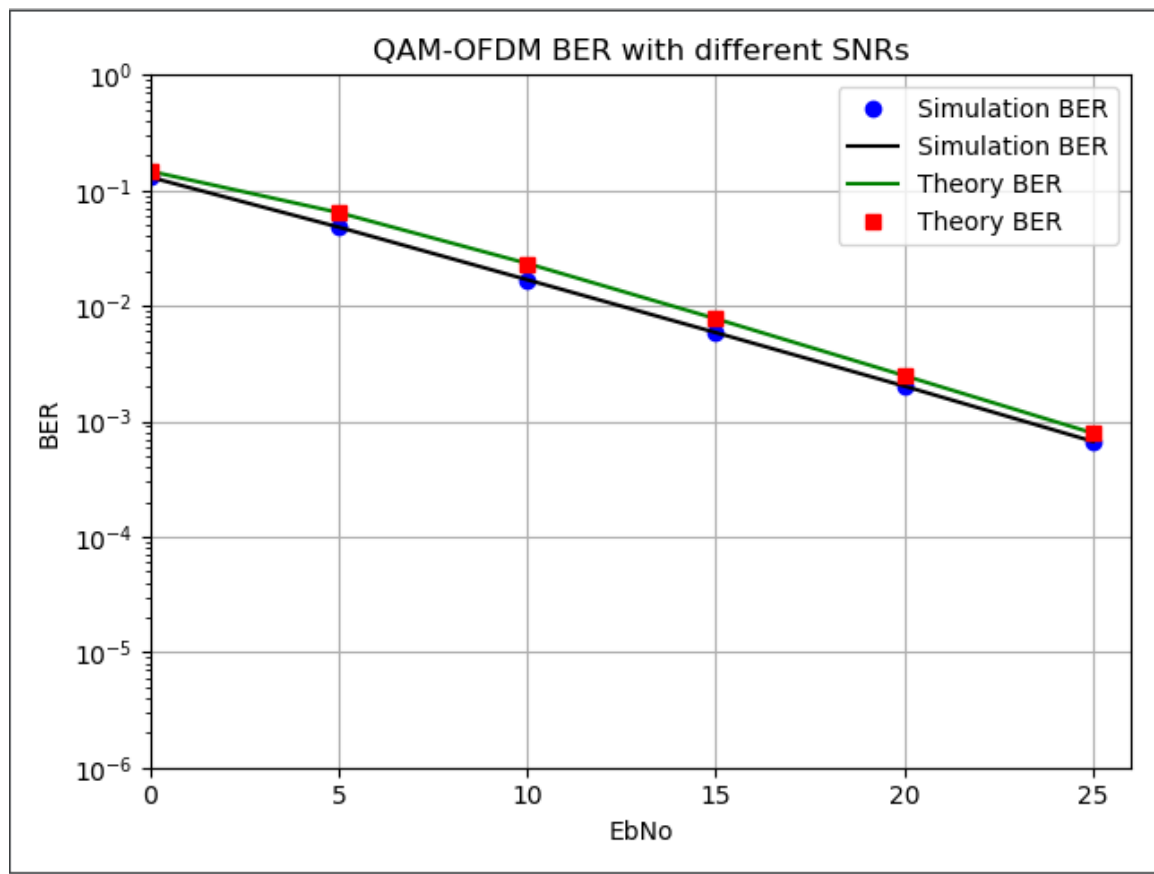
Εικόνα 48 : Μερικές προβλέψεις του μοντέλου

Όσον αφορά τις τιμές του BER από την προσομοίωση στο Matlab, υπολογίσαμε για κάθε dataset με το αντίστοιχο EbNo την μέση τιμή του BER των δειγμάτων.

Για την θεωρητική τιμή χρησιμοποιήθηκε η παρακάτω εξίσωση η οποία υπολογίζει το BER για κάθε ξεχωριστή τιμή EbNo:

```
EbNoLin = mh.pow(10, (ebno / 10))
theoryBers.append(0.5 * (1 - mh.sqrt(EbNoLin / (EbNoLin + 1))))
```

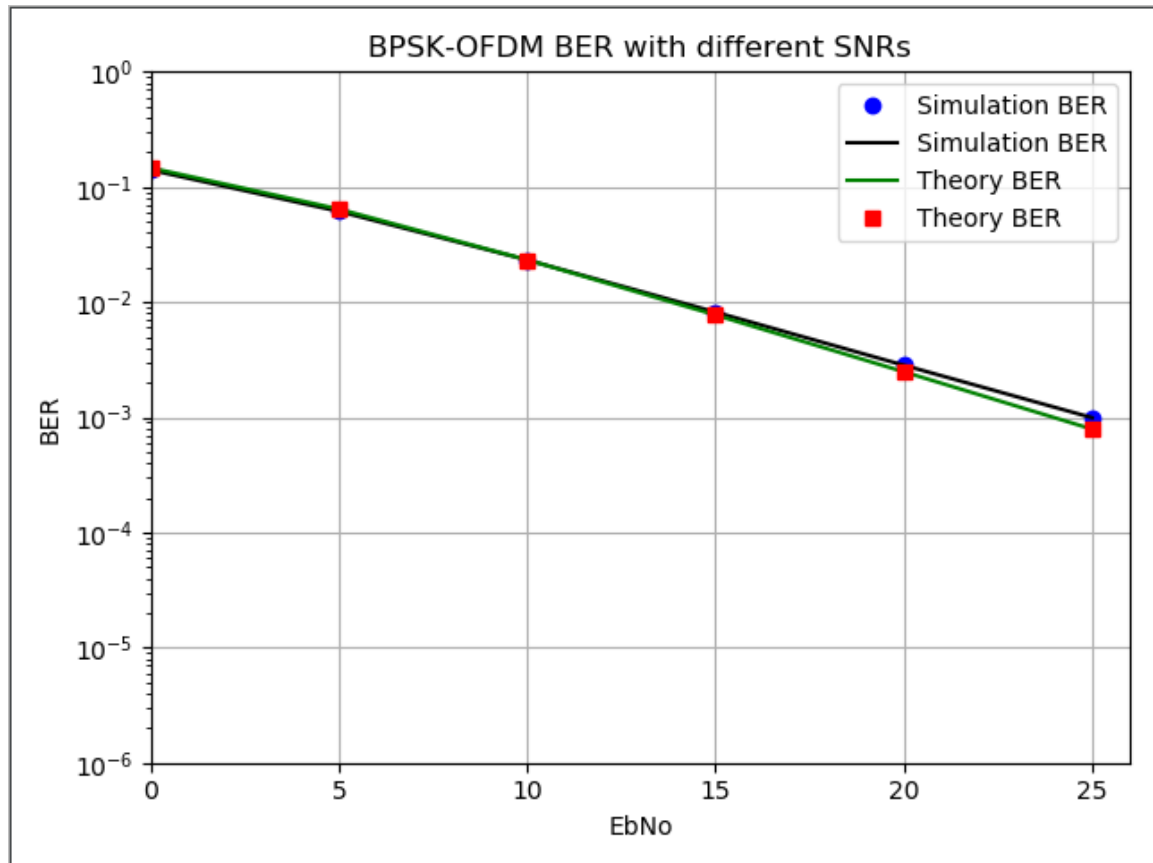
Όπου, ebno είναι η κάθε τιμή EbNo στο εύρος τιμών [0,25] με βήμα 5 και το theoryBers η λίστα όπου αποθηκεύεται η κάθε διαφορετική τιμή BER για το αντίστοιχο EbNo.



Εικόνα 49 : BER value by E_b/N_0 value

Το παραπάνω διάγραμμα παρουσιάζει την σύγκριση μεταξύ της θεωρητικής τιμής BER και της τιμής της προσομοίωσης για την διαμόρφωση 16QAM-OFDM. Οι τιμές σχεδόν συμπίπτουν στα περισσότερα σημεία γεγονός που επιβεβαιώνει την αξιοπιστία των αποτελεσμάτων. Επίσης, φαίνεται ότι όσο πιο μεγάλο είναι το E_b/N_0 τόσο μικρότερη είναι η τιμή του BER. Με αυτόν τον τρόπο εκφράζεται ότι όσο μικρότερος είναι ο ρυθμός σφάλματος των bits τόσο πιο αξιόπιστος είναι ο δέκτης μας. Παρακάτω

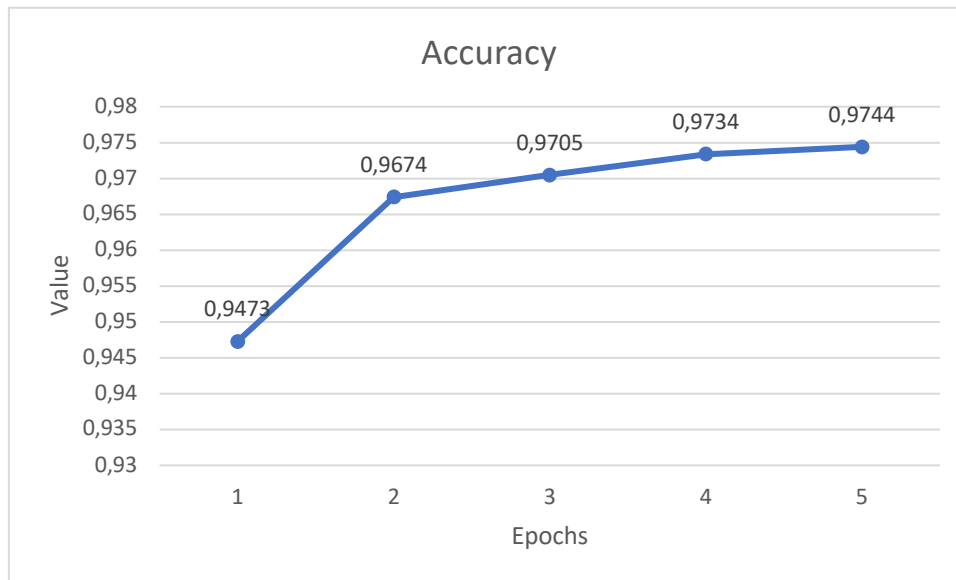
μπορούμε να δούμε και την σύγκριση των τιμών BER για την BPSK-OFDM διαμόρφωση όπου παρατηρείται και εδώ ότι οι τιμές συμπίπτουν.



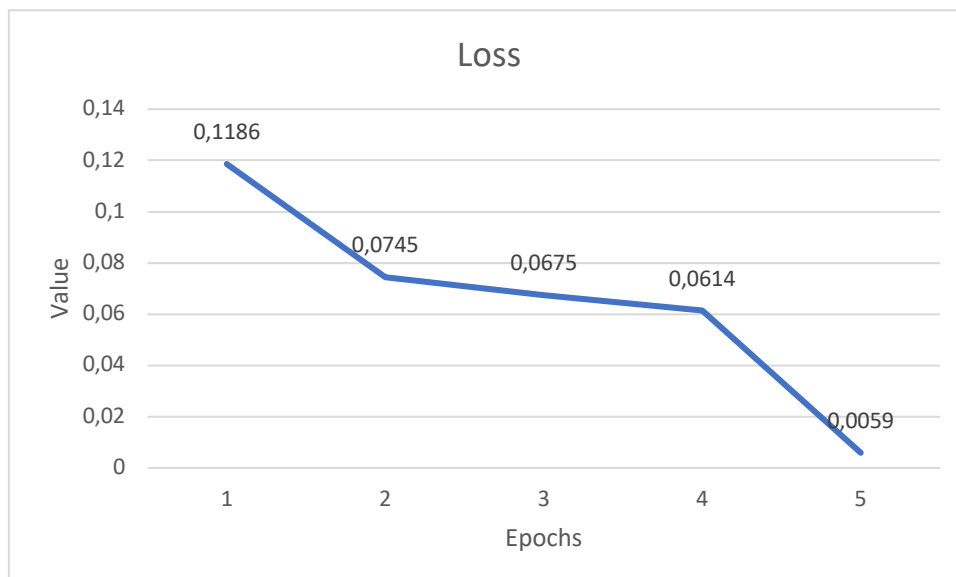
Εικόνα 50 : BPSK-OFDM BER by E_b/N_0 value

Το TensorFlow επίσης, έχει ένα dashboard (Tensorboard) όπου μπορεί να δει κάποιος τα διαγράμματα που αφορούν την ακρίβεια και την απώλεια του μοντέλου κατά την διαδικασία της εκπαίδευσης του καθώς και το γράφημα που αφορά τα διάφορα επίπεδα του μοντέλου. Με την εντολή ‘tensorboard - --logdir = logs’ στο τερματικό του προγραμματιστικού περιβάλλοντος που χρησιμοποιούμε μεραφερόμαστε στο Tensorboard όπου μπορούμε να δούμε τα παραπάνω διαγράμματα.

Όπως φαίνεται παρακάτω στο διάγραμμα ακρίβειας, η ακρίβεια του μοντέλου αυξάνεται σε κάθε epoch γεγονός που μας επιβεβαιώνει ότι η εκπαίδευση του μοντέλου έγινε σωστά. Κάθε φορά που διατρέχεται το dataset το μοντέλο βελτιώνεται καθώς αποκτά εμπειρία και κατά συνέπεια έχει μεγαλύτερη ακρίβεια. Αξίζει να σημειωθεί ότι μετά από δύο epochs πετυχαίνουμε καλύτερη ακρίβεια, η οποία μεγιστοποιείται μέσω δοκιμών στα 10 epochs.

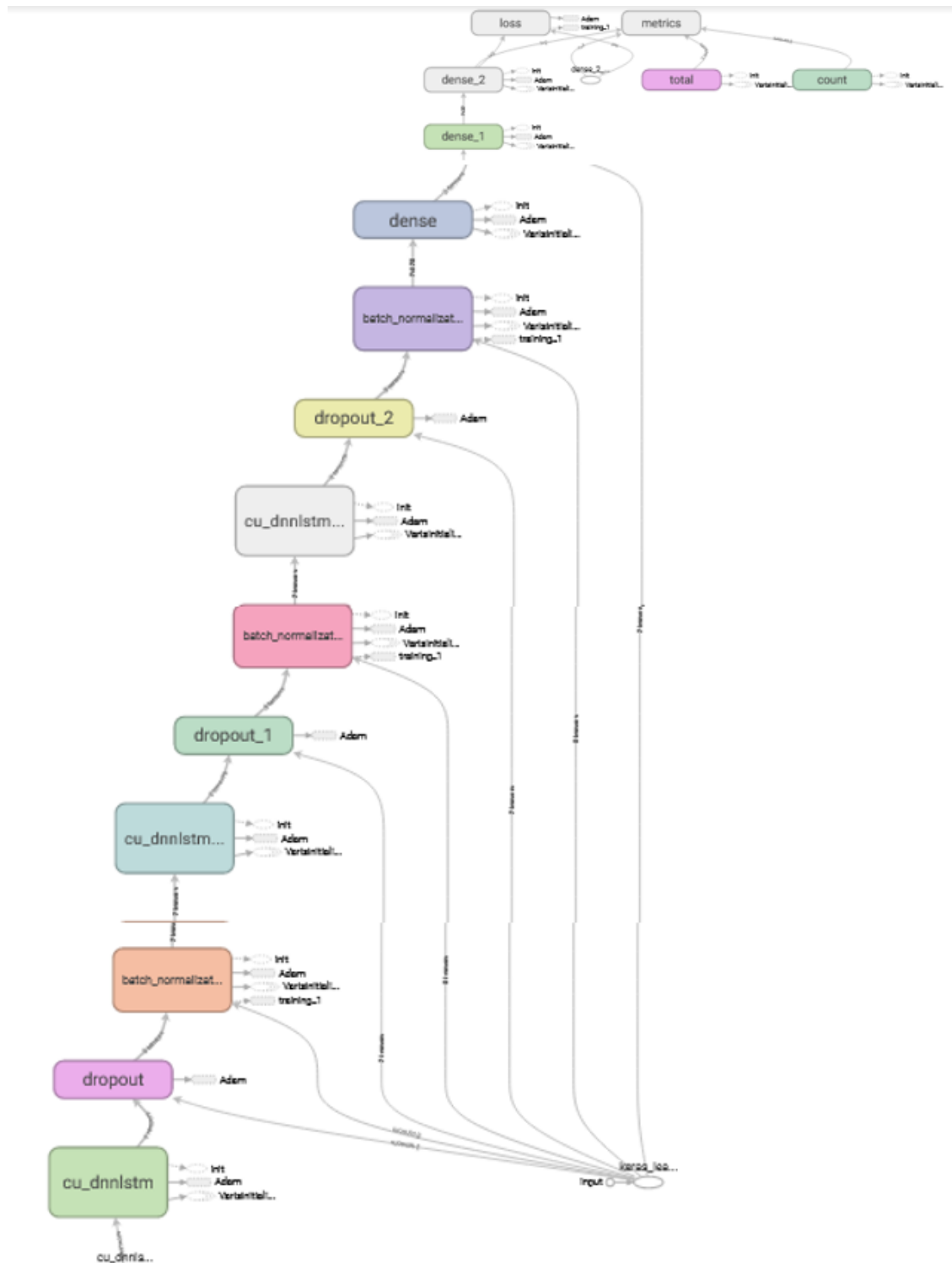


Εικόνα 51 : Διάγραμμα ακρίβειας



Εικόνα 52 : Διάγραμμα απώλειας

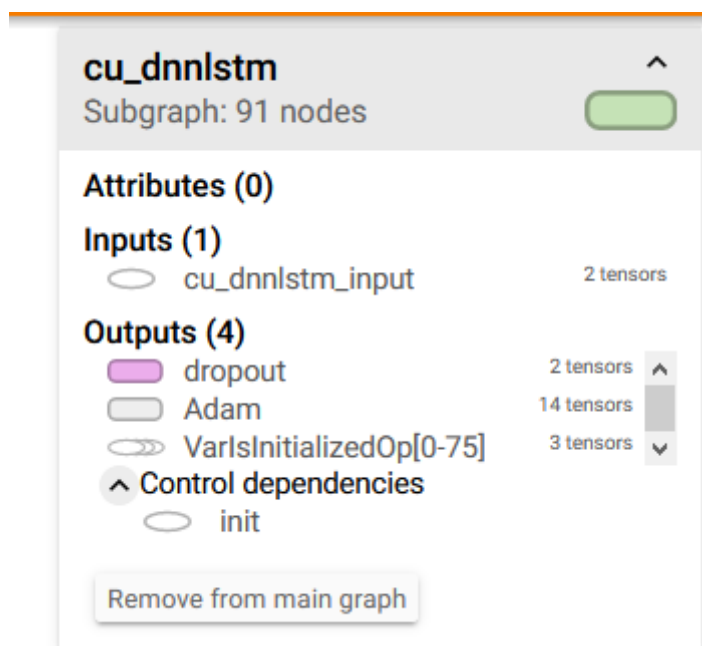
Το αντίστροφο ισχύει για την απώλεια του μοντέλου καθώς σε κάθε epoch μειώνεται λόγω της βελτίωσης του μοντέλου μέσω της εμπειρίας που αποκτάται.



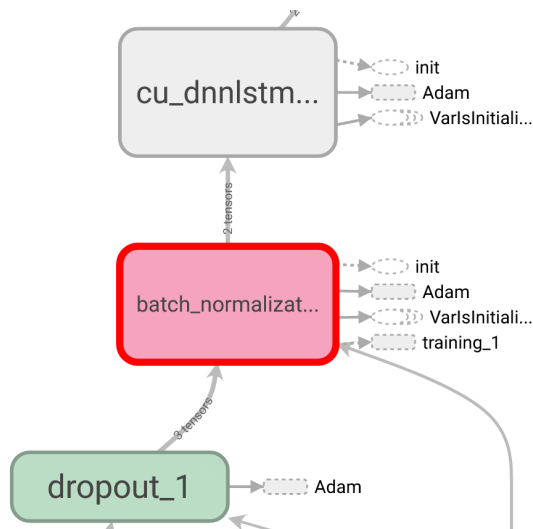
Εικόνα 53 : Γράφημα επιπέδων

Η παραπάνω εικόνα παρουσιάζει τα διάφορα επίπεδα του μοντέλου που εκπαιδεύθηκε καθώς και την λειτουργία του κάθε επιπέδου. Πιο συγκεκριμένα αποπροσωπεί τα

επίπεδα που χρησιμοποιήσαμε στον κώδικα για την ανάπτυξη και εκπαίδευση του μοντέλου, τα οποία φαίνονται αναλυτικότερα στην εικόνα 43. Στην παρακάτω εικόνα απεικονίζεται ένα κομμάτι του γραφήματος όπου παρουσιάζονται αναλυτικότερα τα επίπεδα του μοντέλου. Βλέπουμε ότι έχουμε τους κύριους κόμβους όπως ο “dropout_1” και δίπλα από κάθε κύριο τους βοηθητικούς όπως ο “training_1” ή “Adam”. Οι κόμβοι συνδέονται μεταξύ τους με συνδέσεις οι οποίες είναι τα tensors αντικείμενα του TensorFlow. Όλοι αυτοί οι κόμβοι περιγράφουν λειτουργίες του γραφήματος, τα επίπεδα δηλαδή που χρησιμοποιήσαμε. Γενικά, το γράφημα του Tensorboard μας δίνει μια απεικόνιση του keras μοντέλου που ορίσαμε. Επίσης, αξίζει να αναφερθεί ότι πατώντας σε κάθε κόμβο μπορεί κάποιος να δει μεταδεδομένα. Δηλαδή, μας επιτρέπει να δούμε τις διάφορες εισόδους και εξόδους, καθώς επίσης και άλλες πληροφορίες για το μοντέλο όπως φαίνεται στην παρακάτω εικόνα όπου έχουμε κάνει κλικ στον κόμβο cu_dnnlstm.



Εικόνα 54 : Μεταδεδομένα επιπέδου cu_dnnlstm



Εικόνα 55 : Επίπεδα μοντέλου

Αφού το μοντέλο εκπαιδευθεί, ένα μέρος του dataset, όπως είχαμε αναφέρει και παραπάνω, χρησιμοποιείται για τον έλεγχο του μοντέλου. Από αυτό τον έλεγχο προκύπτει ότι είναι ακριβές όσον αφορά την αξιοπιστία του δέκτη και αυτό μπορεί να φανεί από τα παρακάτω αποτελέσματα.

```
Reliable Receiver's Model with 29351 right guesses

From the correct predicted samples the reliable ones are: 84%

Model Training duration: 56.5599 seconds

Model's loss: 0.0509
Model's accuracy: 0.9784
Mean Squared Error (MSE) of model's testing: 0.021633333333333334
```

Εικόνα 56 : Αποτελέσματα μοντέλου του δέκτη

Ο έλεγχος γίνεται συγκρίνοντας την πρόβλεψη με την πραγματική τιμή και αν είναι σωστή τότε έχουμε μια σωστή πρόβλεψη. Έπειτα, βλέποντας σε ποια κατηγορία είναι αυτή η πρόβλεψη (1 $\rightarrow$ αξιοπιστία δέκτη, 0 $\rightarrow$ αναξιόπιστος), ξεχωρίζουμε τις τιμές BER που είναι καλές για την αξιοπιστία του δέκτη και σε αυτές που δεν είναι.

```
for i in range(len(predictions)):
    if test_y[i] == np.argmax(predictions[i]) and test_y[i] == 0:
        correct_zeros += 1

    if test_y[i] == np.argmax(predictions[i]) and test_y[i] == 1:
        correct_ones += 1

    if test_y[i] != np.argmax(predictions[i]) and test_y[i] == 1:
        wrong_ones += 1

    if test_y[i] != np.argmax(predictions[i]) and test_y[i] == 0:
        wrong_zeros += 1

if (correct_ones + correct_zeros) > (wrong_zeros + wrong_ones):
    print("\nReliable Receiver's Model with %d right guesses" % (correct_ones + correct_zeros))
else:
    print("\nUnreliable receiver's model accuracy!!!")

print('\nReceiver achieved reliability: %d' % ((correct_ones*100)/(correct_zeros+correct_ones))+ '%')
```

Εικόνα 57 : Κώδικας Python για τα αποτελέσματα του μοντέλου

```

bar_width = 0.35
opacity = 0.8 # αδιαφάνεια baras

# Model's Receiver statistics of correct predicted reliable and unreliable signals
rect1 = plt.bar('Correct Predicted Reliable signals', correct_ones, bar_width, alpha=opacity)
rect2 = plt.bar('Correct Predicted Unreliable signals', correct_zeros, bar_width, alpha=opacity)
plt.title("BAR CHART OF RECEIVER'S RELIABILITY")
plt.ylabel('Predictions')
plt.xlabel('Samples')

# attach a label above each bar displaying its height.
def autolabel(rects):
    for rect in rects:
        height = rect.get_height()
        plt.annotate('{}' .format(height),
                    xy=(rect.get_x() + rect.get_width() / 2, height),
                    xytext=(0, 3), # 3 points vertical offset
                    textcoords="offset points",
                    ha='center', va='bottom')

autolabel(rect1)
autolabel(rect2)
plt.tight_layout()
plt.show()

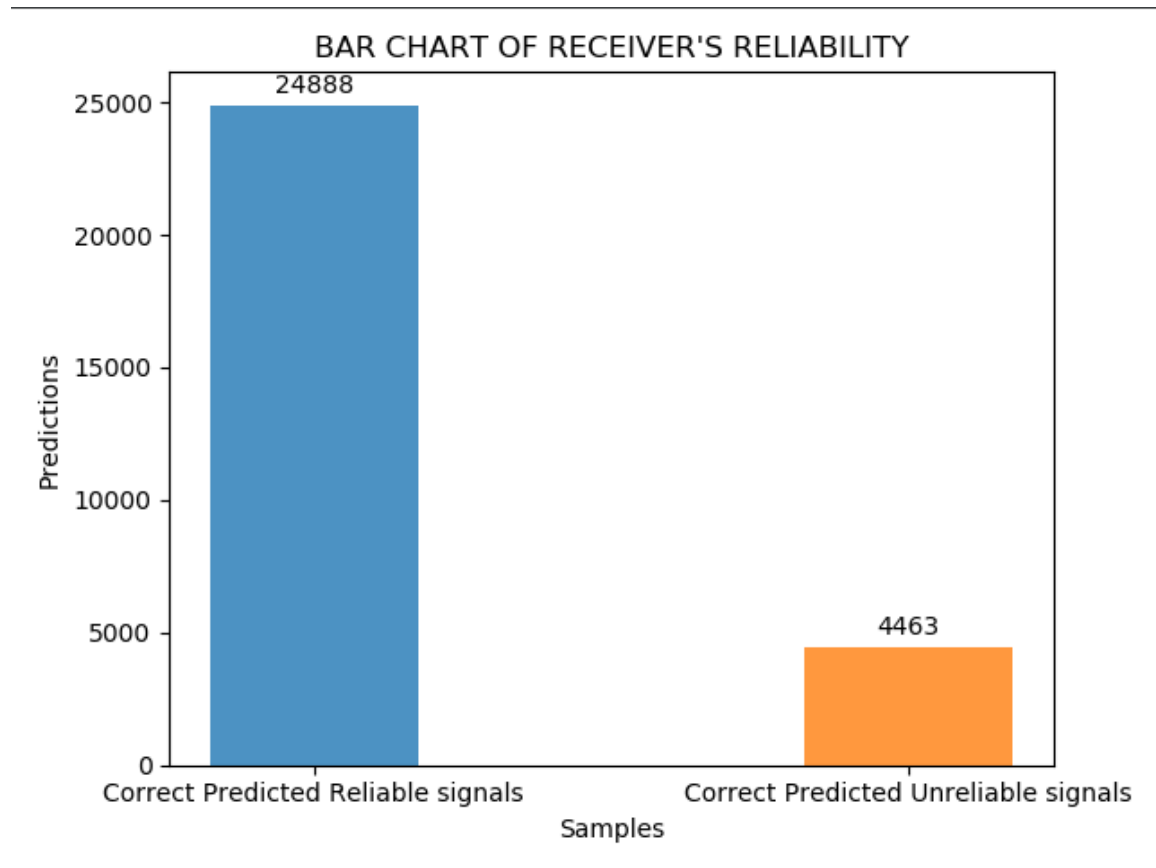
# Bar chart of overall correct and wrong predictions
rect3 = plt.bar('Correct predictions', (correct_ones+correct_zeros), bar_width, alpha=opacity)
rect4 = plt.bar('Wrong Predictions', (wrong_zeros+wrong_ones), bar_width, alpha=opacity)
plt.title("BAR CHART OF MODEL'S ACCURACY")
plt.ylabel('Predictions')
plt.xlabel('Samples')
autolabel(rect3)
autolabel(rect4)
plt.tight_layout()
plt.show()

```

Εικόνα 58 : Κώδικας Python για δημιουργία διαγραμμάτων

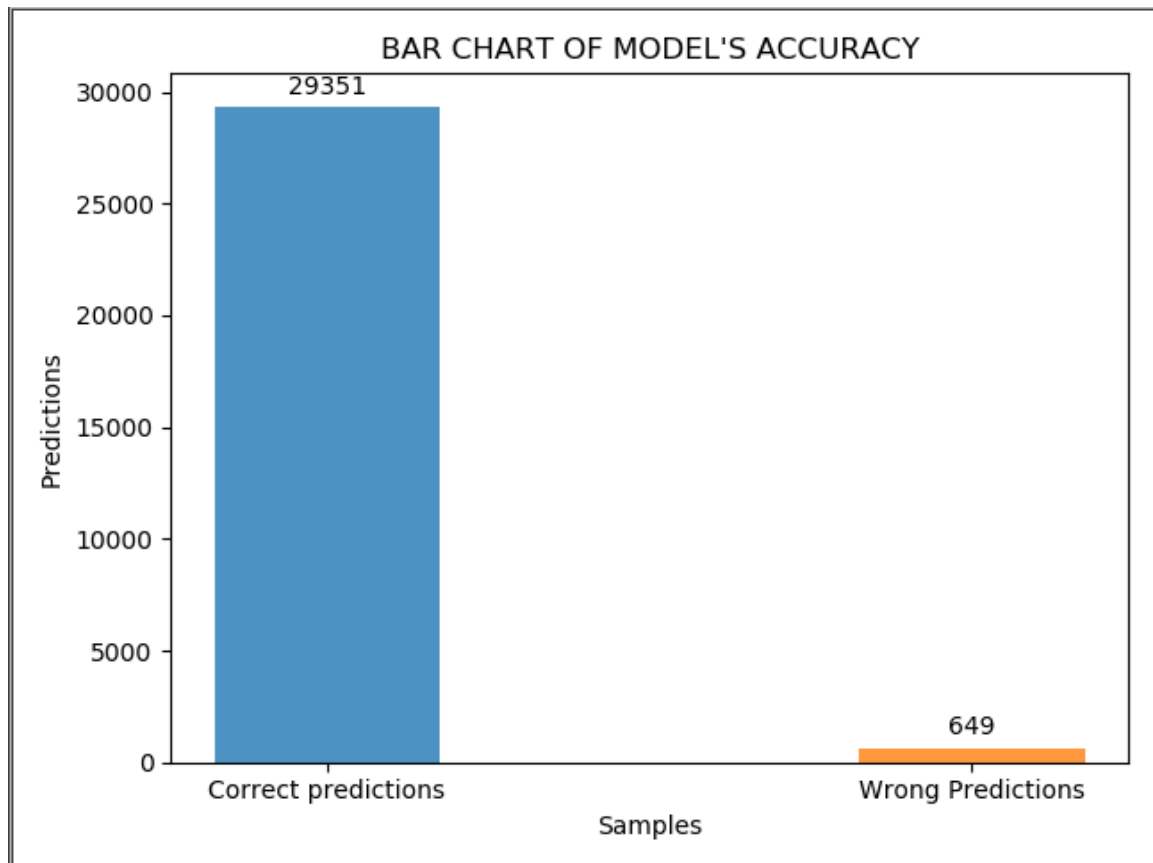
Στην παραπάνω εικόνα φαίνεται ο κώδικας που χρησιμοποιήθηκε για την δημιουργία των παρακάτω διαγραμμάτων. Χρησιμοποιήθηκε η βιβλιοθήκη matplotlib.pyplot για το σκοπό αυτό, δηλαδή την δημιουργία των bar charts.

Στο πρώτο διάγραμμα απεικονίζεται ο αριθμός των σημάτων που θεωρούνται αξιόπιστα και έχουν προβλεφθεί σωστά από το μοντέλο καθώς επίσης και αυτά που είναι μη αξιόπιστα και έχουν προβλεφθεί σωστά. Ουσιαστικά, βλέπουμε από τις προβλέψεις που έγιναν σωστά ποια σήματα παρέχουν αξιοπιστία και ποια όχι.



Εικόνα 59 : Αξιοπιστία Ασύρματου Δέκτη

Στο δεύτερο μπορούμε να δούμε τον αριθμό των σωστών και λάθος προβλέψεων του μοντέλου. Είναι φανερά περισσότερες οι σωστές γεγονός που επιβεβαιώνει την ακρίβεια που παρέχει το μοντέλο.



Εικόνα 60 : Ακρίβεια Μοντέλου

Στα στατιστικά στοιχεία, το μέσο τετράγωνο σφάλμα ή η μέση τετραγωνική απόκλιση μετρά τον μέσο όρο των τετραγώνων των σφαλμάτων, δηλαδή τη μέση τετραγωνική

```
# For calculating mean square error
predictions_list = []
tests_list = []
for i in range(len(predictions)):
    predictions_list.append(np.argmax(predictions[i]))
    tests_list.append(test_y[i])

from sklearn.metrics import mean_squared_error
# Mean Squared Error for comparison
print('Mean Squared Error (MSE): {}'.format(mean_squared_error(tests_list, predictions_list)))
```

Εικόνα 61: MSE

διαφορά μεταξύ των εκτιμώμενων τιμών και της πραγματικής τιμής. Το mean square error εδώ είναι επίσης αρκετά μικρό, γεγονός που επαληθεύει την καλή επίδοση του μοντέλου.

Όπως φαίνεται η ακρίβεια του μοντέλου είναι αρκετά ικανοποιητική σε ποσοστό 98%.

Επίσης ικανοποιητική είναι και η αξιοπιστία του δέκτη σε ποσοστό 84%.

5.12 Συμπεράσματα προσομοίωσης

Με τη βοήθεια του MATLAB αναπτύχθηκαν και προσομοιώθηκαν οι 2 τεχνικές διαμόρφωσης με επιτυχία. Επίσης, με την χρήση τεχνικών Βαθιάς μάθησης το μοντέλο που αναπτύχθηκε, καθώς επίσης και τα επίπεδα που επιλέχθηκαν για αυτό, αποδείχθηκε ότι είναι ακριβές και μπορεί να ανταπεξέλθει στις απαιτήσεις μας. Με την βοήθεια του CUDA ο χρόνος εκπαίδευσης μειώθηκε σημαντικά καθώς το μοντέλο εκπαιδεύθηκε σε μόλις 56 δευτερόλεπτα, διαδικασία που διαφορετικά θα έπαιρνε συνήθως τον πενταπλάσιο χρόνο καθώς ο όγκος των δεδομένων είναι αρκετά μεγάλος και ο αριθμός των epochs ήταν επίσης υψηλός (5 epochs). Επιπλέον, τα αποτελέσματα που προέκυψαν σχετικά με την αξιοπιστία που παρέχει ο δέκτης καθώς και με την ακρίβεια του μοντέλου είναι ικανοποιητικά. Πιο συγκεκριμένα, τα ποσοστά που πέτυχε το μοντέλο όσον αφορά την ακρίβεια και την αξιοπιστία είναι υψηλά σε βαθμό 98% και 84% αντίστοιχα. Το ίδιο υψηλά είναι και για τις άλλες δύο διαμορφώσεις που ελέγχθηκαν. Επιπρόσθετα, συμπεραίνουμε ότι το μοντέλο αυτό μπορεί να βοηθήσει στην σωστή λειτουργία του δέκτη διακρίνοντας τα σήματα με πολλά ή ελάχιστα σφάλματα και ακολούθως παρέχοντας καλή ακρίβεια των σημάτων λήψης. Έτσι, έχουμε έναν αξιόπιστο δέκτη όπου τα σήματα λαμβάνονται με ακρίβεια. Τέλος, παρατηρούμε ότι ο ρυθμός των bit σφαλμάτων μειώνεται σταδιακά με την αύξηση του σηματοθορυβικού λόγου. Κάτι ακόμα όμως που αξίζει να προσέξουμε είναι η αύξηση των σφαλμάτων που παρατηρείται σε ένα πιο ρεαλιστικό μοντέλο σε σχέση με αυτό του απλού προσθετικού λευκού γκαουσιανού θορύβου.

6 Μελλοντικές Επεκτάσεις

6.1 UFMC

Είναι μια εναλλακτική κυματομορφή της OFDM η οποία θα χρησιμοποιηθεί σε 5G συστήματα επικοινωνίας και η οποία θα αντιμετωπίσει κάποια προβλήματα της OFDM που αφορούν κυρίως την απώλεια στην απόδοση του φάσματος. Σε αντίθεση με το OFDM, το UFMC ομαδοποιεί τα υποφέροντα σε διάφορες υποζώνες, εφαρμόζοντας ξεχωριστά το φιλτράρισμα. Με αυτόν τον τρόπο, είναι δυνατό να επιτευχθεί ανθεκτικότητα έναντι της εσφαλμένης ευθυγράμμισης χρόνου και συχνότητας χωρίς επιβάρυνση στο κυκλικό πρόθεμα και ούτω καθεξής, βελτίωση της φασματικής απόδοσης και μείωση των εκπομπών εκτός ζώνης χωρίς απώλεια των πλεονεκτημάτων της OFDM. Όλες αυτές οι ιδιότητες είναι ιδιαίτερα εφαρμόσιμες σε ένα τυπικό σενάριο IoT, όπου μικρή ποσότητα δεδομένων δημιουργείται σποραδικά από μια συσκευή IoT. Πρώτα απ' όλα, η κυματομορφή εγγυάται μια σαφώς καθορισμένη κατοχή φάσματος χωρίς παρεμβολές σε παρακείμενες επικοινωνίες και επομένως αποφεύγοντας οποιαδήποτε σπατάλη ενέργειας. [22]

6.2 OFDMA

Πολλαπλά υποφέροντα μεταφέρουν το σήμα τα οποία τηρούν την ορθογωνικότητα μεταξύ τους γεγονός που επιτρέπει να έχουν το ίδιο εύρος ζώνης. Δεν προτιμάται όταν έχουμε να κάνουμε με ανέβασμα δεδομένων καθώς παρατηρείται μεγάλη κατανάλωση ισχύος για την παραγωγή του σήματος και οι κινητές συσκευές έχουν περιορισμένη χωρητικότητα ισχύος. Για αυτό το λόγο στο LTE χρησιμοποιείται η SC-OFDMA. [18]

$$\sum_{n=0}^N x_n e^{j\omega t n} \quad (34)$$

6.3 CDMA

Είναι πολύ διαφορετική από την πολυπλεξία χρόνου και συχνότητας. Σε αυτό το σύστημα ο χρήστης μπορεί να χρησιμοποιήσει όλο το εύρος ζώνης καθόλη τη διάρκεια. Η βασική αρχή είναι ότι χρησιμοποιούνται διαφορετικοί μοναδικοί CDMA κωδικοί για να ξεχωρίζουν τους διαφορετικούς χρήστες μεταξύ τους.

Για αυτό το λόγο η CDMA είναι ένα από τα πιο ασφαλή συστήματα επικοινωνίας. Το εύρος ζώνης μετάδοσης είναι περισσότερο σε σύγκριση με αυτό του αρχικού σήματος. Εδώ, το σήμα στον πομπό διαδίδεται, διαμορφώνεται μέσω ενός καναλιού και στο

δέκτη το σήμα αποδιαδίδεται και αποδιαμορφώνεται στο ληφθέν αρχικό σήμα από το διαμορφωμένο κύμα.[19]

6.4 Πρακτικές Εφαρμογές

Ο Δέκτης που αναπτύχθηκε μπορεί να φανεί χρήσιμος, όσον αφορά την αξιοπιστία της λειτουργίας του, σε πολλές περιπτώσεις. Μπορεί να χρησιμοποιηθεί σε ένα IoT δίκτυο επικοινωνίας. Ο όγκος των δεδομένων που ανταλλάσσονται μεταξύ των συσκευών και των αισθητήρων είναι τεράστιος όπου μεγάλη σημασία πρέπει να δοθεί στην αξιοπιστία της αποστολής και λήψης των δεδομένων. Επίσης, μπορεί να χρησιμοποιηθεί σε ένα δίκτυο με έξυπνες συσκευές όπως τα smartphones και τα smartwatches. Πιο συγκεκριμένα, και εδώ ανταλλάσσονται πολλά δεδομένα μεταξύ των συσκευών όπου η ύπαρξη εμποδίων μπορεί να προκαλέσει την ύπαρξη θορύβου στην επικοινωνία μεταξύ τους. Για παράδειγμα, τέτοια δεδομένα μπορεί να είναι διάφορες μετρήσεις ή ενδείξεις του σώματος που στέλνονται και λαμβάνονται μεταξύ ενός smartwatch και ενός κινητού.

BIBΛΙΟΓΡΑΦΙΑ

- [1] Bernard Sklar, Digital Communications: Fundamentals and Applications, Prentice Hall, 2nd edition 2017.
- [2] Δημοσθένης Βουγιούκας, Δορυφορικές Επικοινωνίες: Τεχνολογίες, Συστήματα και Εφαρμογές, ΣΕΑΒ, 2015
- [3] Νικόλαος Μήτρου, Ψηφιακές Επικοινωνίες, ΣΕΑΒ, 2015
- [4] Ramjee Prasad, OFDM for Wireless Communications Systems, Artech House, 2004
- [5] Yong Soo Cho, Jaekwon Kim, Won Young Yang, Chung-GU Kang, MIMO-OFDM Wireless Communications with MATLAB, John Wiley & Sons (Asia) Pte LTD, 2010
- [6] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, Google Brain, TensorFlow: A System for Large-Scale Machine Learning, the 12th USENIX Symposium on Operating Systems Design and Implementation, 2016
- [7] Nishant Shukla, Machine Learning with TensorFlow, Manning Publications Co., 2018
- [8] D. Luebke, "CUDA: Scalable parallel programming for high-performance scientific computing," 2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Paris, 2008
- [9] Rivenson, Y., Zhang, Y., Günaydin, H. *et al.* Phase recovery and holographic image reconstruction using deep learning in neural networks. *Light Sci Appl* **7**, 17141 (2018).
- [10] Charles M. Hannum; The NetBSD Foundation (1998). Computers System Research Group(1994).
- [11] Robertson, P.; Kaiser, S. (1999). "The effects of Doppler spreads in OFDM(A) mobile radio systems". Gateway to 21st Century Communications Village. VTC 1999-Fall. IEEE VTS 50th Vehicular Technology Conference
- Haas, R.; Belfiore, J.C. (1997). "A Time-Frequency Well-localized Pulse for Multiple Carrier Transmission". Wireless Personal Communications
- Damien Roque, Cyrille Siclet , Performances of Weighted Cyclic Prefix OFDM with Low-Complexity Equalization
- Jeon, W.G.; Chang, K.H.; Cho, Y.S. (1999). "An equalization technique for orthogonal frequency-division multiplexing systems in time-variant multipath channels". IEEE Transactions on Communications

- [12] S. A. Qureshi, A. S. Rehman, A. M. Qamar, A. Kamal and A. Rehman, "Telecommunication subscribers' churn prediction model using machine learning," Eighth International Conference on Digital Information Management (ICDIM 2013), Islamabad, 2013
- [13] T. O'Shea and J. Hoydis, "An Introduction to Deep Learning for the Physical Layer," in IEEE Transactions on Cognitive Communications and Networking, vol. 3, no. 4, pp. 563-575, Dec. 2017, doi: 10.1109/TCCN.2017.2758370.
- [14] E. Balevi and J. G. Andrews, "One-Bit OFDM Receivers via Deep Learning," in IEEE Transactions on Communications, vol. 67, no. 6, pp. 4326-4336, June 2019, doi: 10.1109/TCOMM.2019.2903811.
- [15] B. Uengtrakul and D. Bunnjaweht, "A cost efficient software defined radio receiver for demonstrating concepts in communication and signal processing using Python and RTL-SDR," 2014 Fourth International Conference on Digital Information and Communication Technology and its Applications (DICTAP), Bangkok, 2014, pp. 394-399, doi: 10.1109/DICTAP.2014.6821718. Dept. of Electrical and Computer Engineering Thammasat University, Pathumthani 12120, Thailand
- [16] Καφεντζής Βασίλειος, Μελέτη της επίδοσης του OFDM σε ασύρματα κανάλια (Μεταπτυχιακή διπλωματική Πανεπιστημίου Πειραιά)
- [17] Giuseppe Bonaccorso, Machine Learning Algorithms, Packt Publishing, 2017
- [18] Arun Kumar Manisha Gupta. Design, comparative study and analysis of CDMA for different modulation techniques, Egyptian Informatics Journal, 2015
- [19] Mosa Ali Abu-Rgheff, Introduction to CDMA Wireless Communications, Academic Press, 1st edition 2007
- [20] Salvador García, Julián Luengo, Francisco Herrera, Data preprocessing in Data Mining, Springer, Cham, 2015
- [21] S. Chakraborty et al., "Interpretability of deep learning models: A survey of results," 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), San Francisco, CA, 2017, pp. 1-6, doi: 10.1109/UIC-ATC.2017.8397411.
- [22] Vitiello, C., Luise, M., Knopp, R., & Kaltenberger, F. (2016). Universal filtered multicarrier for machine type communications in 5G.

ΠΑΡΑΡΤΗΜΑ

Κώδικας Python για ανάπτυξη μοντέλου

```
import pandas as pd
import tensorflow
import numpy as np
import time
import matplotlib.pyplot as plt
import random
import statistics as st
import csv
import math as mh
import datetime
import tkinter as tk

from tensorflow.contrib.learn.python.learn.estimators.sklearn import train_test_split
from keras.callbacks import TensorBoard, EarlyStopping
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, CuDNNLSTM, Dropout, BatchNormalization
from scipy.interpolate import interpolate
from sklearn import preprocessing
from sklearn.metrics import mean_squared_error

##### PYTHON GUI, WINDOW TO CHOOSE MODULATION #####
LARGE_FONT = ("Verdana", 14)
clicked1 = False
clicked2 = False
clicked3 = False

class Modulation(tk.Tk):
    def __init__(self, *args, **kwargs):
        tk.Tk.__init__(self, *args, **kwargs)
        container = tk.Frame(self)
        container.pack(side="top", fill="both", expand=True)
        container.grid_rowconfigure(0, weight=1)
        container.grid_columnconfigure(0, weight=1)
        self.frames = {}
        frame = StartPage(container, self)
        self.frames[StartPage] = frame
        frame.grid(row=0, column=0, sticky="nsew") # sticky = alignment + stretch
        self.show_Frame(StartPage)

    def show_Frame(self, cont):
        frame = self.frames[cont]
        frame.tkraise()
```

```

def isClicked1(con):
    global clicked1
    clicked1 = True
    buttState(1, con)
    return clicked1

def isClicked2(con):
    global clicked2
    clicked2 = True
    buttState(2, con)
    return clicked2

def isClicked3(con):
    global clicked3
    clicked3 = True
    buttState(3, con)
    return clicked3

def buttState(butNum, con):
    for btn in buttons:
        if str(btn['state']) == 'normal':
            btn.configure(state='disabled')

    buttons[butNum-1].configure(state='normal')
    con.destroy()

buttons = []

class StartPage(tk.Frame):
    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)
        label = tk.Label(self, text="Choose Modulation", font=LARGE_FONT)
        label.pack(pady=10, padx=10)

        button1 = tk.Button(self, text="16QAM-QFDM", command=lambda: isClicked1(controller))
        button1.pack()
        buttons.append(button1)

```

```
# for plotting purpose
errors = []
bits = []
bers = []

with open(chosenDataset) as csvfile:
    reader = csv.reader(csvfile)
    for row in reader:
        errors.append(row[1])
        bits.append(row[2])
        bers.append(row[0])
csvfile.close()

# Remove first items of lists cause they are strings
bits.remove('Bits')~# bits -> symbols
bers.remove('Ber')
errors.remove('Errors')
plottingBits = []
plottingber = []
plottingerrors = []

for i in range(len(bits)):
    # zeros is not important to plot
    # first values of ber are zeros
    if float(errors[i]) != 0:
        # cast to float
        plottingBits.append(1.0*float(bits[i]))
        plottingber.append(1.0*float(bers[i]))
        plottingerrors.append(1.0*float(errors[i]))

#print(dataset['Ber'].astype(float)) # convert column values to float type
maxError = max(plottingerrors)

# Plot of BER and of the bits send
plt.plot(plottingBits, plottingerrors, 'r-', label='Errors')
plt.title('Errors Occurred with EbNo 25dB')
plt.axis([0, plottingBits[-1], 1, maxError])~# for QAM-BPSK -> last element of the list
plt.xscale('linear')
plt.yscale('log')
plt.xlabel('Bits Send')
plt.ylabel('Errors')
plt.legend()
plt.show()
```

```
# Plot of BER and EbNo
EbNo_range_qam = range(0, 26, 5)
EbNo = 25.0 # 25dB SNR for each bit
EbNo_qpsk = 30.0
dataset_for_snr = []

if chosenCsvs == "QPSK":
    d1 = pd.read_csv('ofdm-qpsk_ray1\\qpsk-ofdm_ray1_snr0_2.csv')
    d2 = pd.read_csv('ofdm-qpsk_ray1\\qpsk-ofdm_ray1_snr5_2.csv')
    d3 = pd.read_csv('ofdm-qpsk_ray1\\qpsk-ofdm_ray1_snr10_2.csv')
    d4 = pd.read_csv('ofdm-qpsk_ray1\\qpsk-ofdm_ray1_snr15_2.csv')
    d5 = pd.read_csv('ofdm-qpsk_ray1\\qpsk-ofdm_ray1_snr20_2.csv')
    d6 = pd.read_csv('ofdm-qpsk_ray1\\qpsk-ofdm_ray1_snr25_2.csv')
    d7 = pd.read_csv('ofdm-qpsk_ray1\\qpsk-ofdm_ray1_snr30_2.csv')

    dataset_for_snr.append(d1)
    dataset_for_snr.append(d2)
    dataset_for_snr.append(d3)
    dataset_for_snr.append(d4)
    dataset_for_snr.append(d5)
    dataset_for_snr.append(d6)
    dataset_for_snr.append(d7)

if chosenCsvs == "16QAM":
    d1 = pd.read_csv('ofdm-qam_ray1\\16qam-ofdm_ray1_snr0_3.csv')
    d2 = pd.read_csv('ofdm-qam_ray1\\16qam-ofdm_ray1_snr5_3.csv')
    d3 = pd.read_csv('ofdm-qam_ray1\\16qam-ofdm_ray1_snr10_3.csv')
    d4 = pd.read_csv('ofdm-qam_ray1\\16qam-ofdm_ray1_snr15_3.csv')
    d5 = pd.read_csv('ofdm-qam_ray1\\16qam-ofdm_ray1_snr20_3.csv')
    d6 = pd.read_csv('ofdm-qam_ray1\\16qam-ofdm_ray1_snr25_3.csv')

    dataset_for_snr.append(d1)
    dataset_for_snr.append(d2)
    dataset_for_snr.append(d3)
    dataset_for_snr.append(d4)
    dataset_for_snr.append(d5)
    dataset_for_snr.append(d6)
```

```

if chosenCsvs == "BPSK":
    d1 = pd.read_csv('ofdm-bpsk_ray1\\bpsk-ofdm_ray1_snr0_2_64.csv')
    d2 = pd.read_csv('ofdm-bpsk_ray1\\bpsk-ofdm_ray1_snr5_2_64.csv')
    d3 = pd.read_csv('ofdm-bpsk_ray1\\bpsk-ofdm_ray1_snr10_2_64.csv')
    d4 = pd.read_csv('ofdm-bpsk_ray1\\bpsk-ofdm_ray1_snr15_2_64.csv')
    d5 = pd.read_csv('ofdm-bpsk_ray1\\bpsk-ofdm_ray1_snr20_2_64.csv')
    d6 = pd.read_csv('ofdm-bpsk_ray1\\bpsk-ofdm_ray1_snr25_2_64.csv')

    dataset_for_snr.append(d1)
    dataset_for_snr.append(d2)
    dataset_for_snr.append(d3)
    dataset_for_snr.append(d4)
    dataset_for_snr.append(d5)
    dataset_for_snr.append(d6)

# We find the minimum ber of every dataset
# and store it in a list
medians_ber = []
for col in dataset_for_snr:
    #col = col[col['Ber'] != 0]
    medians_ber.append(col['Ber'].mean())

bersum = 0
for i in range(len(plottingber)):
    bersum = bersum + i

medianBer = bersum / len(plottingber)

# Theory Ber of different SNRs [0,25]
theoryBers = []

# 16-QAM OFDM
if chosenCsvs == "16QAM":
    # Theoretical BER for QAM
    for ebno in EbNo_range_qam:
        EbNoLin = mh.pow(10, (ebno / 10)) # -> best one
        theoryBers.append(0.5 * (1 - mh.sqrt(EbNoLin / (EbNoLin + 1))))

plt.plot(EbNo_range_qam, medians_ber, 'bo', EbNo_range_qam, medians_ber, 'k', label='Simulation BER')
plt.plot(EbNo_range_qam, theoryBers, 'g-', EbNo_range_qam, theoryBers, 'rs', label='Theory BER')

```

```

plt.axis([0, EbNo + 1, 1e-6, 1]) # Theory BERs reach 10^-6 for QAM
plt.title('QAM-OFDM BER with different SNRs')

# BPSK OFDM
if chosenCsvs == "BPSK":
    # Theoretical BER for BPSK
    EbNo_range_bpsk = range(0, 26, 5)
    for ebno in EbNo_range_bpsk:
        EbNoLin = mh.pow(10, (ebno / 10))
        theoryBers.append(0.5 * (1 - mh.sqrt(EbNoLin / (EbNoLin + 1))))

    plt.plot(EbNo_range_bpsk, medians_ber, 'bo', EbNo_range_bpsk, medians_ber, 'k', label='Simulation BER')
    plt.plot(EbNo_range_bpsk, theoryBers, 'g-', EbNo_range_bpsk, theoryBers, 'rs', label='Theory BER')

    plt.axis([0, EbNo+1, 1e-6, 1]) # Theory BERs reach 10^-6 for BPSK
    plt.title('BPSK-OFDM BER with different SNRs')

# QPSK OFDM
if chosenCsvs == "QPSK":
    # Theoretical BER for QPSK
    EbNo_range_qpsk = range(0, 31, 5)
    for ebno in EbNo_range_qpsk:
        EbNoLin = mh.pow(10, (ebno/10))
        theoryBers.append(0.5*(1-mh.sqrt(EbNoLin/(EbNoLin+1))))_# best one so far, same type for BPSK also

    plt.plot(EbNo_range_qpsk, medians_ber, 'bo', EbNo_range_qpsk, medians_ber, 'k', label='Simulation BER')
    plt.plot(EbNo_range_qpsk, theoryBers, 'g-', EbNo_range_qpsk, theoryBers, 'rs', label='Theory BER')

    plt.axis([0, EbNo_qpsk + 1, 1e-6, 1]) # Theory BERs reach 10^-6 for QPSK
    plt.title('QPSK-OFDM BER with different SNRs')

plt.xscale('linear')
plt.yscale('log')
plt.xlabel('EbNo')
plt.ylabel('BER')

plt.grid(True)
plt.legend()
plt.show()

```

```

##### PREPROCESSING #####

# Here we find median of BER values
# it had decimal problems, so i had to ceil down the decimal part for better results
# median = np.median(dataset['Ber'])

avg_sum = 0
for sums in bers:
    avg_sum = avg_sum + float(sums)

# the addition of the second part was made based on experience cause the main purpose was to ceil on the decimal part
avg_ber = (avg_sum/len(bers)) + (5*10**(-5))

# I split the signals to those which have small BER value (for that reason they are reliable)
# and to those which have bigger BER values (unreliable)
# based on the median threshold.

def classify(ber):
    if ber <= avg_ber:
        return 1
    else:
        return 0

main_ds = pd.DataFrame()

if len(main_ds) == 0:
    main_ds = dataset
else:
    main_ds = main_ds.join(dataset)

main_ds['target'] = list(map(classify, main_ds['Ber'])) # Mapping Bers and classify the result to a new column target

# shuffle dataset for better preprocessing
main_ds = main_ds.sample(frac=1)

# we drop the column Ber cause we want the other two columns
# for the input data of our model
main_ds.drop('Ber', 1)

for col in main_ds.columns:
    if col != 'target':
        main_ds[col] = preprocessing.scale(main_ds[col].values) # algorithm puts data in one scale
        main_ds[col].dropna(inplace=True) # drop rows with zeros in ber column
    
```

```
# Save preprocessed dataset to new csv file
main_ds.to_csv('preprocessedData.csv')

# lists for train and test data
X = []
Y = []

x = main_ds.drop('target', 1)
y = main_ds.target

# The following progress is to bring the data to form
# so we can manage them as a tensor
for i in x.values:
    X.append([np.array(i)])

for j in y.values:
    Y.append([np.array(j)])

X = np.array(X)
Y = np.array(Y)

# Split the data of the dataset to 70% training and 30% testing
train_x, test_x, train_y, test_y = train_test_split(X, Y, test_size=0.3, random_state=10)

##### MODEL IMPLEMENTATION #####

model = Sequential()

# LSTM model with given input shape
model.add(CuDNNLSTM(128, input_shape=X.shape[1:], return_sequences=True))
model.add(Dropout(0.2))
model.add(BatchNormalization())

model.add(CuDNNLSTM(128, return_sequences=True))
model.add(Dropout(0.2))
model.add(BatchNormalization())

model.add(CuDNNLSTM(128))
model.add(Dropout(0.2))
model.add(BatchNormalization())

model.add(Dense(32, activation='relu'))
model.add(Dropout(0.2))
```

```

model.add(Dense(32, activation='relu'))

# 2 neurons for output layer cause we have 0 or 1
model.add(Dense(2, activation='softmax')) # when categorical_crossentropy

# Summary of model architecture
print(model.summary())

# Load model
model = tensorflow.keras.models.load_model('ofdmModel.model')

opt = tensorflow.keras.optimizers.Adam(lr=0.001, decay=1e-6) # optimizer for the model accuracy

# Use sparse categorical_crossentropy when your classes
# are mutually exclusive (e.g. when each sample belongs exactly to one class)
model.compile(loss='sparse_categorical_crossentropy', optimizer=opt, metrics=['accuracy'])

# Type in terminal: tensorboard --logdir=logs
# to get localhost for accessing Tensorboard
tensorboard = TensorBoard(log_dir="logs\\{}".format(NAME))

# stops training when there is no improvement
early_stop = EarlyStopping(monitor='val_loss', patience=10)

startTime = time.time() # time in seconds
history = model.fit(train_x, train_y, batch_size=BATCH_SIZE, epochs=EPOCHS, callbacks=[early_stop, tensorboard])
endTime = time.time()
predictions = model.predict(test_x)

##### PLOT THE MODELS RESULTS AND BAR CHARTS #####
##### FOR RELIABILITY AND ACCURACY AND OTHER LKE RIGHT PREDICTIONS #####

# 1 -> Reliable sample    0 -> Unreliable sample based on BER
# If reliable samples are more than unreliable samples -> Receiver Reliable

# Correct is for the prediction and 1 or 0 for the category of the sample signal
# 1 -> small BER | 0 -> big BER
correct_ones = 0
correct_zeros = 0
wrong_ones = 0
wrong_zeros = 0

```

```

1 # Print the Test value and the Prediction
2 # for i in range(len(predictions)):
3     # print('Test:{}'.format(test_y[i]), np.argmax(predictions[i]))

4 for i in range(len(predictions)):
5     if test_y[i] == np.argmax(predictions[i]) and test_y[i] == 0:
6         correct_zeros += 1

7     if test_y[i] == np.argmax(predictions[i]) and test_y[i] == 1:
8         correct_ones += 1

9     if test_y[i] != np.argmax(predictions[i]) and test_y[i] == 1:
10        wrong_ones += 1

11    if test_y[i] != np.argmax(predictions[i]) and test_y[i] == 0:
12        wrong_zeros += 1

13
14 if (correct_ones + correct_zeros) > (wrong_zeros + wrong_ones):
15     print("\nReliable Receiver's Model with %d right guesses" % (correct_ones + correct_zeros))
16
17 else:
18     print("\nUnreliable receiver's model accuracy!!!")
19
20 print('\nFrom the correct predicted samples the reliable ones are: %d' % ((correct_ones*100)/(correct_zeros+correct_ones))+ '%')
21
22 print('\nModel Training duration: %.4f seconds' % (endTime - startTime))
23
24 score = model.evaluate(test_x, test_y, verbose=0)
25 print("\nModel's loss: %.4f" % score[0])
26 print("Model's accuracy: %.4f" % score[1])
27
28 # Save model
29 model.save("ofdmModel.model")
30
31 # For calculating mean square error
32 predictions_list = []
33 tests_list = []
34 for i in range(len(predictions)):
35     predictions_list.append(np.argmax(predictions[i]))
36     tests_list.append(test_y[i])

```

```
opacity = 0.8_# bar opacity
|
# Model's Receiver statistics of correct predicted reliable and unreliable signals
rect1 = plt.bar('Correct Predicted Reliable signals', correct_ones, bar_width, alpha=opacity)
rect2 = plt.bar('Correct Predicted Unreliable signals', correct_zeros, bar_width, alpha=opacity)
plt.title("BAR CHART OF RECEIVER'S RELIABILITY")
plt.ylabel('Predictions')
plt.xlabel('Samples')

# attach a label above each bar displaying its height.
def autolabel(rects):
    for rect in rects:
        height = rect.get_height()
        plt.annotate('{}' .format(height),
                      xy=(rect.get_x() + rect.get_width() / 2, height),
                      xytext=(0, 3), # 3 points vertical offset
                      textcoords="offset points",
                      ha='center', va='bottom')

autolabel(rect1)
autolabel(rect2)
plt.tight_layout()
plt.show()

# Bar chart of overall correct and wrong predictions
rect3 = plt.bar('Correct predictions', (correct_ones+correct_zeros), bar_width, alpha=opacity)
rect4 = plt.bar('Wrong Predictions', (wrong_zeros+wrong_ones), bar_width, alpha=opacity)
plt.title("BAR CHART OF MODEL'S ACCURACY")
plt.ylabel('Predictions')
plt.xlabel('Samples')
autolabel(rect3)
autolabel(rect4)
plt.tight_layout()
plt.show()
```

Κώδικας για επεξεργασία των csv αρχείων

```
import csv

with open('ofdm-qam_rayl\\16qam-ofdm_rayl_snr20.csv', 'r', newline='') as f:
    r = csv.reader(f)
    data = [line for line in r]

with open('ofdm-qam_rayl\\16qam-ofdm_rayl_snr20.csv', 'w', newline='') as file:
    filewriter = csv.writer(file, delimiter=',', quotechar='|', quoting=csv.QUOTE_MINIMAL)
    filewriter.writerow(['Ber', 'Errors', 'Bits'])
    filewriter.writerows(data)

file.close()
```