



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ

**Μελέτη συστημάτων Echo State Networks με σκοπό την
βελτίωση των επιδόσεων Ασύρματων Επικοινωνιών.**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Τσαλίκη Παράσχου

Επιβλέπων : Χάρης Μεσαριτάκης

Μέλη εξεταστικής επιτροπής:

Χάρης Μεσαριτάκης,
Χαράλαμπος Σκιάνης,
Θεόδωρος Κωστούλας.

Σάμος, 10/2024

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πρόλογος και ευχαριστίες

Με την ολοκλήρωση της παρούσας διπλωματικής εργασίας ένα σημαντικό κεφάλαιο της ζωής μου φτάνει στο τέλος του. Αυτό των πρώτων μου σπουδών, στη Σάμο, στο τμήμα Μηχανικών Πληροφοριακών & Επικοινωνιακών Συστημάτων.

Αρχικά θα ήθελα να ευχαριστήσω τον κύριο Χάρη Μεσαριτάκη, για την πολύτιμη καθοδήγησή του κατά τη διάρκεια της εκπόνησης της εργασίας μου.

Επίσης θα ήθελα να δώσω ένα τεράστιο ευχαριστώ στους γονείς μου, Παναγιώτη και Ιωάννα, που με στήριξαν σε όλη τη πορεία των σπουδών μου.

Τέλος θα ήθελα να πω ένα μεγάλο ευχαριστώ στους φίλους και συνάδελφους που γνώρισα, οι οποίοι με στήριξαν και με βοήθησαν κατά τη διάρκεια των σπουδών μου.

Στο όμορφο νησί, εις το επανιδείν.

© 2024

του

Τσαλίκη Παράσχου

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πίνακας Περιεχομένων

1	Εισαγωγή στις τηλεπικοινωνίες.....	1
1.1	Η ιστορία της ανθρώπινης επικοινωνίας.....	1
1.2	Η ιστορία των τηλεπικοινωνιών.....	1
1.3	Η έννοια της επικοινωνίας.....	2
1.4	Τα σήματα και η ανάγκη για διαμόρφωση.....	3
1.5	Διαμορφώσεις αναλογικού σήματος.....	4
1.5.1	Διαμόρφωση Πλάτους - <i>Amplitude Modulation (AM)</i>	4
1.5.2	Διαμορφώσεις Γωνίας: Διαμόρφωση συχνότητας - <i>Frequency Modulation (FM)</i> και Διαμόρφωση Φάσης- <i>Phase Modulation(PM)</i>	5
1.6	Διαμορφώσεις ψηφιακού σήματος.....	5
1.6.1	Διαμόρφωση Πλάτους Παλμού (<i>Pulse Amplitude Modulation - PAM</i>).....	6
1.6.2	Διαμόρφωση Ολίσθησης Συχνότητας (<i>Frequency Shift Keying - ASK</i>).....	8
1.6.3	Διαμόρφωση Ολίσθησης Φάσης (<i>Phase Shift Keying - PSK</i>).....	8
1.6.4	Διαμόρφωση Μ-αδικής Ορθογωνικής μεταλλαγής Πλάτους - <i>QAM (Quadrature Amplitude Modulation)</i>	8
1.7	Η Αποδιαμόρφωση (<i>Demodulation</i>).....	9
2	Η ασύρματη επικοινωνία.....	10
2.1	Ο ασύρματος δίαυλος επικοινωνίας.....	10
2.1.1	Η λειτουργία της κεραίας.....	11
2.1.2	Φαινόμενα που επηρεάζουν το σήμα κατά τη διάδοση.....	11
2.2	Μοντελοποίηση φαινομένων διάδοσης.....	12
2.2.1	Κανάλι Προσθετικού Λευκού Γκαουσιανού Θορύβου – <i>Additive White Gaussian Noise (AWGN)</i>	12
2.2.2	Κανάλι <i>Rayleigh</i>	13
2.2.3	Κανάλι <i>Rice</i>	16
2.2.4	Κανάλι <i>Nakagami-m</i>	17
3	Συστήματα Ανάκτησης Σήματος.....	18
3.1	Τεχνικές Ισοστάθμισης (<i>Equalization Techniques</i>).....	18
3.1.1	<i>Tapped Delay Line (TDL)</i> ή <i>Finite Impulse Response (FIR) Filter</i>	18
3.1.2	Ισοσταμιστής πρόσθιας τροφοδότησης (<i>Feedforward Equalizer - FFE</i>).....	19
3.1.3	Ισοσταμιστής ανάδρασης απόφασης (<i>Decision Feedback Equalizer - DFE</i>).....	20
3.1.4	Μη Γραμμικός Ισοσταμιστής <i>Volterra (Volterra Nonlinear Equalizer - VNLE)</i>	21
3.1.5	Προσαρμοστικοί Αλγόριθμοι Ισοσταθμιστών.....	23
3.1.6	Βέλτιστοι Ισοστάθμιστές (<i>Optimal Equalizer</i>).....	23

3.1.7.	Άλλες σύγχρονες τεχνικές ισοστάθμισης:	23
3.2	Ισοστάθμιση με Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks - ANN):	24
3.2.1	Νευρωνικά Δίκτυα Πρόσθιας Τροφοδότησης (Feedforward Neural Networks – FNN):	25
3.2.2	Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks – CNN):	27
3.2.3	Επαναλαμβανόμενα Νευρωνικά Δίκτυα (Recurrent Neural Networks – RNN):	29
3.2.3.1	Δίκτυα μακροπρόθεσμης-βραχυπρόθεσμης μνήμης (Long-Short Term Memory Networks - LSTM):	32
3.2.3.2	Δίκτυα αναδρομικής μονάδας με πύλες (Gated Recurrent Unit Networks - GRU):	33
3.2.3.3	Echo State Networks - (ESN):	35
4	Εφαρμογή Reservoir Computing σε Rayleigh Fading Channel	39
4.1	Πρόβλεψη NARMA ακολουθίας με Reservoir Computing:	39
4.1.1	Αρχικοποίηση ακολουθίας	39
4.1.2	Εκπαίδευση του ESN	41
4.1.3	Δοκιμή του ESN	42
4.1.4	Αποτελέσματα δοκιμών του ESN για τη NARMA10	44
4.2	Πρόβλεψη διαμορφωμένης ακολουθίας 4-PAM μέσα από κανάλι εξασθένησης Rayleigh και AWGN με χρήση του ESN:	45
4.2.1	Δημιουργία Rayliegh Channel	45
4.2.2	Διαμόρφωση της ακολουθίας με 4-PAM	46
4.2.3	Δοκιμή του ESN	47
4.2.4	Προσθήκη ψηφιακής μνήμης στο ESN για βελτίωση της πρόβλεψης	49
4.2.5	Βελτιωμένη δοκιμή και αποτελέσματα του ESN	51
5	Συμπέρασμα.	52
	Βιβλιογραφία & Πηγές Πληροφορίας:	53
	Σύνδεσμοι Εικόνων:	55
	Παράρτημα Κώδικα Ι:	56
	Παράρτημα Κώδικα ΙΙ:	59
	Παράρτημα Κώδικα ΙΙΙ:	60

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Γενικό Μοντέλο Επικοινωνίας κατά Shannon & Weaver.....	3
Εικόνα 2: Διάγραμμα Αστερισμού Διαμόρφωσης M-QAM.	8
Εικόνα 3: Μοντέλο ασύρματης ζεύξης.	10
Εικόνα 4: Αναπαράσταση κεραίας.	11
Εικόνα 5: Αναπαράσταση Gaussian Κατανομής.	12
Εικόνα 6: Δίαυλος NLOS επικοινωνίας.	13
Εικόνα 7: Διάγραμμα IQ πολυδιαδρομικής συνιστώσας από Rayleigh Fading Channel.....	14
Εικόνα 8: Αναπαράσταση κατανομής Rayleigh.....	15
Εικόνα 9: Δίαυλος με πολλαπλές συνιστώσες NLOS και μιας συνιστώσας LOS επικοινωνίας.	16
Εικόνα 10: Αναπαράσταση κατανομής Rice.....	17
Εικόνα 11: Αναπαράσταση κατανομής Nakagami-m.....	17
Εικόνα 12: Μοντέλο καναλιού διακριτού χρόνου με διασυμβολική παρεμβολή και AWGN.	18
Εικόνα 13: Γραμμικό εγκάρσιο φίλτρο προσαρμοστικών συντελεστών.....	19
Εικόνα 14: Γενικό μοντέλο αρχιτεκτονικής ενός DFE.....	20
Εικόνα 15: Η αρχιτεκτονική ενός DFE με 2 feedforward taps και 1 feedback tap.....	20
Εικόνα 16: Μη γραμμικός ισοσταθμιστής Volterra.	21
Εικόνα 17: Ο βιολογικός νευρώνας συγκριτικά με το βασικό Perceptron.....	24
Εικόνα 18: FNN με 1 στρώμα εισόδου, 2 κρυφά στρώμα και 1 στρώμα εξόδου.	25
Εικόνα 19: Αρχιτεκτονική ενός CNN.....	27
Εικόνα 20: Απλή αναπαράσταση RNN (αριστερά) και "ξεδιπλωμένη" στο χρόνο.....	29
Εικόνα 21: Αναπαράσταση της λειτουργίας ενός LSTM.	32
Εικόνα 22: Δομή του GRU συγκριτικά με LSTM.....	33
Εικόνα 23: Αναπαράσταση ενός ESN.....	35
Εικόνα 24: Διάγραμμα ροής σε κλασσικό ESN.....	36
Εικόνα 25: Παράμετροι του ESN.....	39
Εικόνα 26: Αρχικοποίηση του dataset.....	39
Εικόνα 27: Δημιουργία NARMA ακολουθίας.....	39
Εικόνα 28: Προσαρμογή Spectral Radius.....	40
Εικόνα 29: Διαχωρισμός training & testing dataset.....	41
Εικόνα 30: Εκπαίδευση ESN.....	41
Εικόνα 31: Υπολογισμός πίνακα καταστάσεων.....	41
Εικόνα 32: Υπολογισμός πίνακα τιμών του teacher.....	42
Εικόνα 33: Υπολογισμός weights εξόδων με pseudoinverse.....	42
Εικόνα 34: Δοκιμή του ESN.....	42
Εικόνα 35: Δημιουργία καταστάσεων της ακολουθίας εκπαίδευσης.....	42
Εικόνα 36: Προβλεπόμενη έξοδος.....	43
Εικόνα 37: Activation Function του ESN.....	43
Εικόνα 38: Αναστροφή της κλιμάκωσης και μετατόπισης (de-scaling and de-shifting).....	43
Εικόνα 39: Γενικό Surface Diagram συμπεριφοράς ESN.....	44
Εικόνα 40: Πανοραμικό Surface Diagram συμπεριφοράς ESN.....	44

Εικόνα 41: Δημιουργία Rayleigh Channel	45
Εικόνα 42: Ορισμός path Delays	45
Εικόνα 43: Ορισμός των average path Gains	46
Εικόνα 44: 4-PAM Modulation	46
Εικόνα 45: Δημιουργία φέροντος	46
Εικόνα 46: Signal through Rayleigh Channel	46
Εικόνα 47: Signal through AWGN channel	47
Εικόνα 48: Είσοδος & Έξοδος στο ESN	47
Εικόνα 49: Σύγκριση BER Cases 1 & 2	48
Εικόνα 50: Interpolation	49
Εικόνα 51: Plot με τα στάδια του Σήματος	49
Εικόνα 52: Αποτελέσματα τελικής δοκιμής του ESN	51

Ακρωνύμια

AM	Amplitude Modulation
ANN	Artificial Neural Network
ARMA	AutoRegressive Moving Average
ASK	Amplitude Shift Keying
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
BPTT	Backpropagation Through Time
CNN	Convolutional Neural Networks
dB	Decibel
DFE	Decision Feedback Equalizer
DML	Direct Modulation Laser
ESN	Echo State Network
FFE	Feedforward Equalizer
FIR	Finite Impulse Response
FM	Frequency Modulation
FNN	Feedforward Neural Networks
FSK	Frequency Shift Keying
GD	Gradient Descent
GHz	Giga Hertz
GRU	Gated Recurrent Unit
Hz	Hertz
ISI	Intersymbol Interference
KHz	Kilo Hertz
LMS	Least Mean Squares
LOS	Line Of Sight
LSM	Liquid State Machine
LSTM	Long-Short Term Memory
MLPS	Multi-layer Perceptrons
MLSE	Maximum Likelihood Sequence Estimation
NARMA	Nonlinear AutoRegressive Moving Average
NLOS	No Line Of Sight
NN	Neural Network
NRMSE	Normalized Root Mean Square Error

PAM	Pulse Amplitude Modulation
PDF	Probability Density Function
PM	Phase Modulation
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RC	Reservoir Computing
ReLU	<i>Rectified Linear Unit</i>
RLS	<i>Recursive-Least Square</i>
RNN	Recurrent Neural Network
SGD	<i>Stochastic Gradient Descent</i>
SNR	Signal to Noise Ratio
TDL	Tapped Delay Line
VNLE	Volterra Nonlinear Equalizer
W	Weights
ΣΠΠ	Συνάρτηση Πυκνότητας Πιθανότητας
ΦΠΠ	Φασματική Πυκνότητα Ισχύος

Περίληψη

Η παρούσα διπλωματική εργασία πραγματεύεται ταυτόχρονα το πεδίο των ασύρματων επικοινωνιών και των τεχνητών νευρωνικών δικτύων. Πιο συγκεκριμένα στα πλαίσια αυτής της εργασίας διερευνήθηκε η αξιοποίηση της μεθόδου υπολογισμού δεξαμενής (Reservoir Computing) ως ένα χρήσιμο εργαλείο για την αντιμετώπιση διαλείψεων που ενδέχεται να εμφανιστούν σε ένα ασύρματο τηλεπικοινωνιακό σύστημα.

Η δομή της παρούσας εργασίας έχει ως εξής: στην αρχή παρουσιάζεται μια εισαγωγή σε βασικές έννοιες των τηλεπικοινωνιών και αναλύονται οι λειτουργίες διάφορων βασικών διατάξεων όπως φίλτρα, ισοσταθμιστές καθώς και πραγματοποιείται μια σύντομη εισαγωγή στα νευρωνικά δίκτυα.

Στη συνέχεια, ακολουθεί η ανάλυση της χρήσης ενός τεχνητού νευρωνικού δικτύου Reservoir Computing για την πρόβλεψη μιας ψευδοτυχαίας ακολουθίας NARMA και εξάγονται αποτελέσματα, αναδεικνύοντας την ικανότητα του Reservoir Computing να αναλύει χρονοσειρές.

Στη συνέχεια εξετάζεται η χρήση του Reservoir στη περίπτωση ενός ασύρματου δίαυλου επικοινωνίας (Wireless Link), για τη πρόβλεψη ενός εκπεμπόμενου σήματος, ψηφιακά διαμορφωμένου κατά πλάτος βαθμού 4 (4-Pulse Amplitude Modulation), σε κανάλι εξασθένησης Rayleigh (Rayleigh Fading Channel) πολλαπλών διαδρομών.

Σε τελικό στάδιο το Reservoir εκπαιδεύεται κατάλληλα, ώστε να επιτευχθεί πρόβλεψη της αρχικής κατάστασης του λαμβανόμενου σήματος, εφόσον έχει περάσει μέσα από κανάλι εξασθένησης Rayleigh πολλαπλών διαδρομών και λευκό προσθετικό γκαουσιανό θόρυβο (Additive White Gaussian Noise).

Λέξεις Κλειδιά: *Ασύρματος δίαυλος, Φίλτρο, Ισοστάθμιστής, Reservoir Computing, Echo State Network, NARMA, Pulse Amplitude Modulation, Rayleigh Fading Channel, Additive White Gaussian Noise.*

Abstract

This thesis focuses on the field of telecommunications and artificial neural networks. More specifically, we investigate the use of Reservoir Computing (RC) as a useful tool for dealing with intermittencies that may occur in a wireless telecommunication system.

In this work, we provide an introduction to telecommunications, we discuss different communication components, such as filters, equalizers and especial focus is invested on the use of artificial neural networks for such deployments.

Furthermore, we evaluated the ESN's (Echo State Network) capability to predict time-series such as NARMA sequence.

The next step consisted of evaluating the capabilities of the ESN model to address the impairments of a wireless channel, subject to fading. In this context, a multi-path Rayleigh fading channel was emulated, whereas a digital 4-pulse amplitude modulated signal (4-PAM) was transmitted over our wireless link.

The last step, our ESN was trained so as to predict the initial state of the of the received signal, given that it has passed through a multipath channel with Rayleigh fading and Additive White Gaussian Noise (AWGN).

Keywords: *Wireless Link, Filter, Equalizer, Reservoir Computing, Echo State Network, NARMA, Pulse Amplitude Modulation, Rayleigh Fading Channel, Additive White Gaussian Noise.*

1

Εισαγωγή στις τηλεπικοινωνίες

1.1 Η ιστορία της ανθρώπινης επικοινωνίας.

Ένα από τα πρώτα είδη ανθρωπιδων που έδειξαν σημαντική εξέλιξη με χαρακτηριστικά που περιλαμβάνουν την όρθια στάση, τη χρήση εργαλείων και τις **πρώτες ενδείξεις μιας πιο σύνθετης επικοινωνίας**, ήταν ο Homo Erectus ο οποίος εξαφανίστηκε πριν από περίπου 110 χιλιάδες χρόνια. Η κατασκευή και η χρήση εργαλείων απαιτούσε ένα βασικό επίπεδο συνεννόησης και μετάδοσης γνώσης από γενιά σε γενιά. Υπάρχουν ενδείξεις ότι ταξίδευαν σε μεγάλες αποστάσεις και δημιουργούσαν κοινότητες, κάτι που ενίσχυε την ανάγκη για επικοινωνία. Μετά από χιλιάδες χρόνια εξέλιξης και τελικά με την εμφάνιση του Homo Sapiens, στο σημείο του σήμερα, έχουν αναπτυχθεί οπτικοακουστικές τέχνες και περίπλοκα κοινωνικά συστήματα, έννοιες όπως αυτή της διαφύλαξης της ιστορίας της εξέλιξης του ανθρώπου, δηλαδή της μνήμης του είδους από γενιά σε γενιά. Προκύπτει εύλογα το συμπέρασμα ότι η **επικοινωνία** αποτέλεσε ένα στοιχείο άρρηκτα συνδεδεμένο με τη φύση του ανθρώπου και την μετεξέλιξή του σε αυτό που είναι σήμερα. [1]

1.2 Η ιστορία των τηλεπικοινωνιών.

Η ιστορία των τηλεπικοινωνιών έχει καθιερωθεί να χωρίζεται σε δύο σημαντικές περιόδους. Σε αυτή πριν και μετά τη χρήση του ηλεκτρισμού από τον άνθρωπο. Κάποιες σημαντικές μέθοδοι απομακρυσμένης επικοινωνίας πριν τη χρήση του ηλεκτρισμού οι οποίες αξίζει να αναφερθούν είναι:

- **Οι Φρυκτωρίες:** Με τη χρήση της φωτιάς και το άναμμα πυρσών πάνω σε σημεία με αρκετό υψόμετρο, ώστε να είναι ορατά από χιλιόμετρα, μεταφερόταν μια δυαδική πληροφορία για το αν συνέβη κάτι ή όχι, δίχως κάποια περαιτέρω κωδικοποίηση.
- **Ο Οπτικός τηλεγράφος:** Με τη περαιτέρω χρήση της φωτιάς για την επίτευξη απομακρυσμένης επικοινωνίας, δεν άργησαν να αναπτυχθούν μέθοδοι κωδικοποίησης του αλφάβητου. Ο Κλεοξένης και ο Δημόκλειτος παρουσίασαν μια επαναστατική μέθοδο. Έχοντας ως γνωστή παραδοχή πομπού και δέκτη έναν πίνακα με γραμμές και στήλες, όπου κάθε γράμμα είχε τη θέση του, ανάλογα με πόσους πυρσούς άναβαν και σε ποιο σημείο των τειχών εμφανίζονταν. Ο δέκτης καταλάβαινε σε ποια γραμμή ή στήλη του πίνακα αντιστοιχούσε το οπτικό ερέθισμα που λάμβανε.

- **Ο Υδραυλικός τηλεγράφος:** Ο Αινείας ο Τακτικός τοποθέτησε δύο δοχεία με νερό με μια κάθετη ράβδο μέσα τους, πάνω στην οποία ανά έξι εκατοστά αναγράφονταν κωδικοποιημένα μηνύματα στρατιωτικής φύσεως. Όταν ο πομπός δήλωνε την πρόθεση για επικοινωνία τότε και ο δέκτης με το άναμμα πυρσού, δήλωνε την ετοιμότητα του. Στη συνέχεια ανασήκωναν ταυτόχρονα τους πυρσούς και άνοιγαν τις βαλβίδες των δοχείων με το νερό ώστε να αδειάσουν μέχρι το σημείο της χρονικής στιγμής που ο αποστολέας θα κατέβαζε το πυρσό του με αποτέλεσμα και στα δύο σημεία η στάθμη να ήταν η ίδια λόγω του συγχρονισμού εκ των δύο.
- **Η μέθοδος Ταμ-Ταμ:** Αφρικανικές φυλές μετέφεραν μηνύματα μέσω του ήχου τυμπάνων.
- **Τα Σήματα καπνού:** Οι ιθαγενείς της Αμερικής με τη χρήση μιας κουβέρτας με την οποία σκέπαζαν και ξεσκέπαζαν μια εστία φωτιάς δημιουργούσαν σύννεφα καπνού τα οποία μετέφεραν κωδικοποιημένα μηνύματα.
- **Τα Ταχυδρομικά περιστέρια:** Καθώς τα περιστέρια διαθέτουν ειδικούς υποδοχείς στο οργανισμό τους που περιέχουν μικροσκοπικά κρυσταλλίδια μαγνητίτη (ορυκτό ευαίσθητο σε αλλαγές του μαγνητικού πεδίου), έχουν τη δυνατότητα να χαρτογραφούν με βάση διαφορετικά χαρακτηριστικά όπως την ένταση και τη κλίση του μαγνητικού πεδίου της Γης τα οποία διαφέρουν από περιοχή σε περιοχή. [2] Μεταφέρονταν έτσι σε απομακρυσμένες περιοχές με σκοπό να γυρίσουν στη γενέτειρά τους με τα γνώριμα μαγνητικά χαρακτηριστικά φέροντας ένα γραπτό μήνυμα.
- **Ο Μηχανικός Τηλέγραφος:** Αποτελούσε μηχανή με κινητούς βραχίονες που με τις διαφορετικές θέσεις που λαμβάνονταν η κωδικοποιημένη πληροφορία μεταξύ σταθμών.

Μετά τη χρήση του ηλεκτρισμού τρόποι απομακρυσμένης επικοινωνίας άξιοι αναφοράς ήταν:

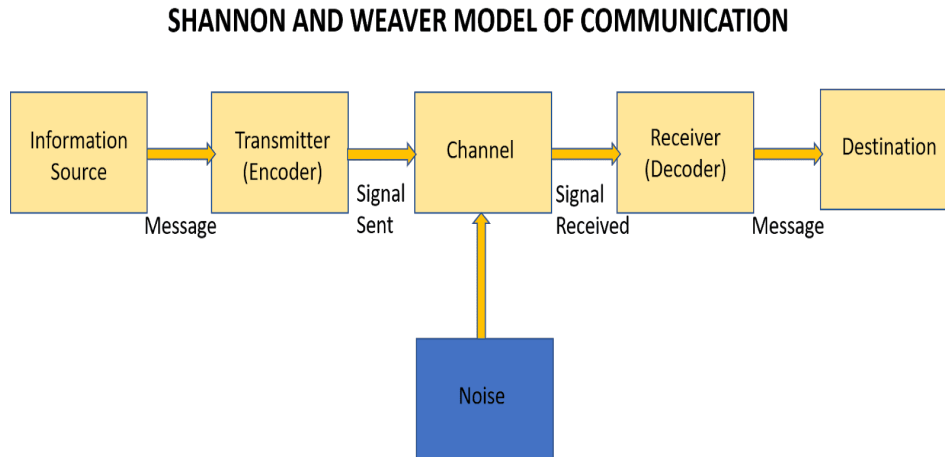
- **Ο Ηλεκτρικός τηλεγράφος:** Ήταν το πρώτο μέσο απομακρυσμένης επικοινωνίας που χρησιμοποίησε τον ηλεκτρισμό, εκμεταλλευόμενο τις ηλεκτρικές μεταβολές που προκαλεί ο πομπός στο μέσο διάδοσης. Όταν οι μεταβολές αυτές έφταναν στον δέκτη, τότε σύμφωνα με τον κώδικα Morse, τα σήματα αποκωδικοποιούνταν.
- **Το Τηλέφωνο:** Οι Bell και Gray παρατήρησαν την εξέλιξη του τηλεγράφου, ο οποίος με τις ερευνητικές ενέργειες των Stearns και Edison το 1870, μπορούσε πλέον να αποστείλει ταυτόχρονα πολλαπλά τηλεγραφικά σήματα και άρα θα μπορούσε να μεταδοθεί η φωνή.
- **Ο Ασύρματος τηλεγράφος:** Ήταν η πρώτη ασύρματη συσκευή μετάδοσης πληροφορίας. Η υλοποίηση του δεν θα ήταν δυνατή δίχως τις θεωρητικές έρευνες των Maxwell το 1860 και Hertz το 1887. Παρουσιάστηκε για πρώτη φορά το 1896 από τον Marconi και τοποθετήθηκε σε πλοία. Το 1901 πραγματοποιήθηκε ασύρματη υπερατλαντική επικοινωνία μεταξύ Αγγλίας και Καναδά. [3]

1.3 Η έννοια της επικοινωνίας.

Πολλοί είναι οι ορισμοί για την έννοια της επικοινωνίας (communication), αναλόγως και το επιστημονικό πεδίο που μελετάται. Στο πλαίσιο της συγκεκριμένης εργασίας είναι επαρκές να οριστεί κατά το πρότυπο Shannon και Weaver (1948) το οποίο αποτελεί το πιο θεμελιώδες μοντέλο όλων. Σε αυτό παρουσιάστηκαν έννοιες όπως: η πηγή πληροφορίας (information source), το μήνυμα πληροφορίας (information message), ο πομπός (transmitter), ο δέκτης (receiver), το κανάλι

(channel), η κωδικοποίηση (coding), η πιθανότητα σφάλματος (probability of error), η χωρητικότητα του καναλιού (channel capacity), η εντροπία (entropy), κ.α. [3]

Το συγκεκριμένο μοντέλο έγινε αποδεκτό στον τομέα των τηλεπικοινωνιών αλλά και σε πολλούς άλλους τομείς επιστημών.



Εικόνα 1: Γενικό Μοντέλο Επικοινωνίας κατά Shannon & Weaver.

1.4 Τα σήματα και η ανάγκη για διαμόρφωση.

Σε ένα σύνθετο τηλεπικοινωνιακό σύστημα πραγματοποιούνται συνεχώς εκπομπές και λήψεις σημάτων. Τα σήματα αποτελούν μετρήσιμες μεταβολές που έχει μια φυσική ποσότητα (π.χ. τάση και ένταση ηλεκτρικού ρεύματος, φωτεινότητα των pixel μιας φωτογραφίας κ.α.). Στον τομέα των τηλεπικοινωνιών συνήθως πρόκειται για μεταβολές τάσης και έντασης ως προς τον χρόνο, τη συχνότητα ή το χώρο. Έτσι αναπαρίστανται ως συναρτήσεις της μορφής $i(t)$, με πεδίο τιμών την τάση, την ένταση ή την φωτεινότητα, και πεδίο ορισμού το χρόνο, τη συχνότητα και τον χώρο.

Τα σήματα χωρίζονται σε αναλογικά, διακριτού χρόνου και ψηφιακά. Τα αναλογικά σήματα έχουν πεδία ορισμού και τιμών τα οποία αποτελούν διαστήματα πραγματικών αριθμών. Στα σήματα διακριτού χρόνου η ανεξάρτητη μεταβλητή t λαμβάνει διακριτές τιμές, ενώ η εξαρτημένη i λαμβάνει τιμές από διάστημα πραγματικών αριθμών. Τέλος στα ψηφιακά σήματα, τόσο η εξαρτημένη όσο και η ανεξάρτητη μεταβλητή λαμβάνουν διακριτές τιμές.

Για να καταφέρει να φτάσει η χρήσιμη πληροφορία από τον πομπό στο δέκτη, θα πρέπει να κωδικοποιηθεί το σήμα μας πριν εκπεμφθεί, με τέτοιο τρόπο ώστε οι αντίστοιχες μεταβολές στο πλάτος, τη συχνότητα και τη φάση, να μπορούν να ερμηνευθούν από τον παραλήπτη. Ανάλογα το είδος της πηγής, παράγονται σήματα βασικής ζώνης είτε αναλογικά είτε ψηφιακά. Στη συνέχεια τα σήματα πληροφορίας διέρχονται από ένα ενσύρματο ή ασύρματο δίαυλο επικοινωνίας, όπου επιφέρει διαφορετικά χαρακτηριστικά και επιδράσεις πάνω στα μεταδιδόμενα σήματα.

Είναι λοιπόν αναγκαίο να τροποποιηθούν τα σήματα πληροφορίας που λαμβάνονται, έτσι ώστε να εκμεταλλεύονται τις ιδιότητες του εκάστοτε διαύλου. Συνήθως τα σήματα πληροφορίας ή αλλιώς σήματα διαμόρφωσης (*modulation signals*) είναι βασικής ζώνης (με τιμές γύρω από συχνότητα $f=0$), ενώ τα σήματα που εκπέμπονται μπορούν να είναι και βασικής ζώνης αλλά και

ζωνοπερατά (με τιμές γύρω από μια συχνότητα f_c , όπου $f_c \gg 0$ ή $f_c \ll 0$). [4] Η διαδικασία κατά την οποία το σήμα διαμόρφωσης μεταβάλλει τα χαρακτηριστικά ενός φέροντος σήματος (*carrier signal*), ονομάζεται *διαμόρφωση* (*modulation*) και διαφέρει ανάλογα με το είδος του σήματος και το επιθυμητό τρόπο διαμόρφωσης. Ουσιαστικά το φέρον σήμα (συνήθως ημιτονοειδές) πλέον φέρει τη πληροφορία του σήματος διαμόρφωσης. Το αποτέλεσμα αυτής της διαδικασίας είναι το *διαμορφωμένο σήμα* (*modulated signal*). Γίνεται αντιληπτό ότι από τη βασική ζώνη συχνοτήτων με κεντρική συχνότητα το 0, γίνεται μετατόπιση σε ζώνη υψηλότερων συχνοτήτων με κέντρο τη συχνότητα αυτή του φέροντος σήματος.

Κάποιοι συνήθεις λόγοι για την εφαρμογή της *διαμόρφωσης* είναι:

- Η μετατόπιση του σήματος πληροφορίας, από τη περιοχή βασικής ζώνης σε διαφορετική περιοχή συχνοτήτων όπου εκεί το μέσο διάδοσης (καλώδιο χαλκού, οπτική ίνα, ατμόσφαιρα) έχει ευνοϊκότερες συνθήκες για τη μετάδοση.
- Η μετατόπιση σε περιοχή συχνοτήτων η οποία έχει οριστεί κατάλληλη για συγκεκριμένους σκοπούς εκπομπής, από τους διεθνούς οργανισμούς προτυποποίησης.
- Η μετατόπιση σε υψηλότερη περιοχή συχνοτήτων έτσι ώστε να έχουμε μικρότερα μήκη κύματος, καθώς είναι ανάλογα των μεγεθών των κεραιών εκπομπής και λήψης που καλούμαστε να κατασκευάσουμε.
- Η μετατόπιση σε μια περιοχή συχνοτήτων έτσι ώστε σε μια διάταξη που διαθέτουμε να μπορέσουμε να εκπέμψουμε και άλλα σήματα σε διαφορετικές περιοχές συχνοτήτων, με αποτέλεσμα να μη παρεμβάλλονται μεταξύ τους.
- Η δυνατότητα να προσδώσουμε ένα είδος αντοχής του σήματος στον θόρυβο και τις παρεμβολές, διευρύνοντας το εύρος ζώνης του σήματος διαμόρφωσης σε σχέση με αυτό του σήματος πληροφορίας.
- Η δυνατότητα να πετύχουμε *πολυπλεξία* (*multiplexing*), δηλαδή τη ταυτόχρονη μετάδοση σημάτων μέσα από τον ίδιο δίαυλο επικοινωνίας με διαφορετικές πηγές. [5]

1.5 Διαμορφώσεις αναλογικού σήματος.

Οι αναλογικές επικοινωνίες δεν χρησιμοποιούνται πλέον σε τόσο μεγάλο βαθμό όσο στο παρελθόν, όμως καθίστανται βασικές για τη θεμελίωση των αντίστοιχων τεχνικών που χρησιμοποιούνται στις ψηφιακές επικοινωνίες. Παρακάτω παρουσιάζονται: η διαμόρφωση κατά πλάτος (Amplitude Modulation - AM), η διαμόρφωση κατά συχνότητα (Frequency Modulation - FM) και η διαμόρφωση κατά φάση (Phase Modulation - PM). [5]

1.5.1 Διαμόρφωση Πλάτους - Amplitude Modulation (AM):

Σε αυτή τη περίπτωση διαμόρφωσης η πληροφορία κωδικοποιείται στις διαφορετικές τιμές που λαμβάνει το πλάτος του σήματος διαμόρφωσης.

Θεωρείται φέρον σήμα $c(t)$ της μορφής:

$$c(t) = A_c \cos(2\pi f_c t + \varphi_c)$$

όπου A_c το πλάτος, f_c η συχνότητα και φ_c η φάση του.

Πριν πραγματοποιηθεί η εκπομπή του, θα μεταβληθεί κάποιο από αυτά τα χαρακτηριστικά του σήματος πληροφορίας ή συνδυασμός τους. Στη συγκεκριμένη περίπτωση πρόκειται για αναλογικό, οπότε η μεταβολή που θα πραγματοποιηθεί είναι συνεχής και ονομάζεται *αναλογική διαμόρφωση (analog modulation)*. Το πλάτος $c(t)$ αποτελεί γραμμική συνάρτηση του σήματος πληροφορίας βασικής ζώνης $m(t)$. Το διαμορφωμένο σήμα $s(t)$ θα έχει τη μορφή:

$$s(t) = A_c [1 + km(t)] \cdot \cos(2\pi f_c t + \varphi_c)$$

δηλαδή το πλάτος του είναι $a(t) = [1 + km(t)]$, όπου k ο δείκτης διαμόρφωσης (*modulation index*) και ο όρος 1, χρησιμοποιείται για να εξασφαλίζεται ότι το $a(t)$ είναι πάντα θετικό, ώστε η ανάκτηση του σήματος πληροφορίας να γίνεται ευκολότερα.

1.5.2 Διαμορφώσεις Γωνίας: Διαμόρφωση συχνότητας - Frequency Modulation (FM) και Διαμόρφωση Φάσης-Phase Modulation(PM):

Οι δύο επιπλέον περιπτώσεις αναλογικών διαμορφώσεων FM και PM αποτελούν τις *διαμορφώσεις γωνίας (angle modulation)*. Σε αυτές τις περιπτώσεις η γωνία του φέροντος σήματος $c(t)$ μεταβάλλεται σύμφωνα με το σήμα πληροφορίας $m(t)$, ενώ το πλάτος A_c μένει σταθερό. Το διαμορφωμένο σήμα $s(t)$ είναι της μορφής:

$$s(t) = A_c \cos[2\pi f_c t + \varphi_i(t)]$$

Εάν η στιγμιαία φάση $\varphi_i(t)$ μεταβάλλεται γραμμικά με το σήμα πληροφορίας $m(t)$ τότε έχω *διαμόρφωση φάσης*, δηλαδή:

$$\varphi_i(t) = k_p \cdot m(t)$$

όπου k_p (rad/volt), είναι η *ευαισθησία φάσης (phase sensitivity)* του διαμορφωτή.

Εάν η στιγμιαία συχνότητα f_c μεταβάλλεται γραμμικά με το σήμα πληροφορίας τότε έχουμε *διαμόρφωση συχνότητας*, δηλαδή:

$$f_i(t) = f_c + k_f \cdot m(t)$$

όπου k_f η ευαισθησία συχνότητας (*frequency sensitivity*) του διαμορφωτή.

1.6 Διαμορφώσεις ψηφιακού σήματος

Στη περίπτωση της ψηφιακής διαμόρφωσης (*digital modulation*) το σήμα πληροφορίας είναι ψηφιακό, άρα και οι μεταβολές που προκαλεί στα χαρακτηριστικά του φέροντος σήματος είναι διακριτών τιμών.

Η επίδοση ενός τρόπου διαμόρφωσης εξαρτάται από την *αποδοτικότητα της ισχύος (power efficiency)* και την *φασματική απόδοση (bandwidth efficiency)*. Η αποδοτικότητα ισχύος ερμηνεύεται από τον λόγο της ενέργειας του bit, προς τη φασματική πυκνότητα θορύβου (E_b/N_0). Η φασματική απόδοση ερμηνεύεται από τον λόγο R_b/B_T bps/Hz. Σύμφωνα με το θεώρημα του Shannon, για να επιτευχθεί μια μικρή πιθανότητα σφάλματος θα πρέπει η μέγιστη αποδοτικότητα σφάλματος η_{bmax} να είναι ίση με $\log_2(1+SNR)$, δηλαδή θέτεται περιορισμός από τον σηματοθορυβικό λόγο SNR (Signal to Noise Ratio).

Στα ψηφιακά συστήματα επικοινωνίας το σήμα πληροφορίας αναπαρίσταται από bits, δηλαδή τα χαρακτηριστικά του φέροντος αλλάζουν σύμφωνα με τη μετάδοση των bits. Τα μεταδιδόμενα bits ομαδοποιούνται και σχηματίζουν τα λεγόμενα *ψηφιακά σύμβολα M (digital symbol)*, όπου το

πλήθος τους είναι $M=2^n$. Το ψηφιακά διαμορφωμένο σήμα είναι ημιτονοειδές με διάρκεια T_s και αναπαριστά ένα σύμβολο μήκους n bits, άρα ο ρυθμός μετάδοσης συμβόλου (*baud rate*||*symbol rate*) είναι $R=1/T_s$ bits/sec. [5] Η γενική μορφή της κυματομορφής προκύπτει ως:

$$s(t) = \sqrt{\frac{2E_{s,i}}{T_s}} \cos[2\pi f_c t + \varphi_i], 0 \leq t < T_s.$$

Αλλιώς γράφεται ως:

$$s(t) = A_i \cdot g_{T_s}(t) \cdot \cos[2\pi f_i t + \varphi_i]$$

όπου g_{T_s} αποτελεί τη κρουστική απόκριση του φίλτρου στο πομπό, με διάρκεια T_s . Ο παλμός g_{T_s} καθορίζει το φάσμα του σήματος διαμόρφωσης. Μπορούν να χρησιμοποιηθούν πολλών ειδών παλμοί για τη βελτίωση των φασματικών χαρακτηριστικών του σήματος διαμόρφωσης, ωστόσο για λόγους απλότητας διατηρούνται οι ορθογωνικοί παλμοί και παραλείπεται η εισαγωγή του συμβόλου i , καθώς ισχύει ότι:

$$g_{T_s}(t) = \begin{cases} \sqrt{\frac{2E_s}{T_s}}, 0 \leq t < T_s \\ 0, \text{αλλού.} \end{cases}$$

1.6.1 Διαμόρφωση Πλάτους Παλμού (Pulse Amplitude Modulation - PAM):

Σε αυτή τη περίπτωση ψηφιακής διαμόρφωσης τα bits του σήματος πληροφορίας αντιστοιχούν στο πλάτος κάθε παλμού. Τα σύμβολα που εκπέμπονται διαφέρουν μεταξύ τους ως προς το πλάτος των παλμών, άρα και στην ενέργειά τους. Η ανίχνευση από τον δέκτη βασίζεται σε αυτή ακριβώς τη μέτρηση της ενέργειας του παλμού. Η μέθοδος PAM μπορεί να υλοποιηθεί σε βασική ζώνη αλλά και ως ζωνοπερατή. Η πληροφορία που μεταφέρεται είναι ίδια και στις δύο περιπτώσεις, η κύρια διαφορά βρίσκεται στη συχνότητα λειτουργίας και στον τρόπο διάδοσης του σήματος, αλλά οι παλμοί και τα σύμβολα αντιπροσωπεύουν την ίδια πληροφορία.

Για να εκπεμφθεί το M-PAM (M πλήθος ψηφιακών συμβόλων) στη βασική ζώνη χρησιμοποιούνται κυματομορφές $s_i(t), i=1,2,\dots,M=2^n$ και αντιστοιχούν σε n bits το κάθε σύμβολο. Διαμορφώνουν έτσι το πλάτος του βασικού παλμού $g(t)$ με διάρκεια T και ενέργεια που δίνεται παρακάτω:

$$\varepsilon_g = \int_0^T |g(t)|^2 dt$$

Ένα σύστημα M-PAM με άρτιο είτε περιττό αριθμό συμβόλων M μπορεί να περιγραφεί από τη σχέση:

$$s_i(t) = \begin{cases} (2j - M - 1) \cdot g(t), t \in [0, T] \\ 0, \text{αλλού.} \end{cases}$$

Η επιλογή του είδους του παλμού $g(t)$, γίνεται αφού ληφθεί υπόψιν το εύρος ζώνης που διατίθεται, η απόκριση του καναλιού και η διασυμβολική παρεμβολή (*intersymbol interference*). Σύμφωνα με την μέθοδο Gram-Schmidt έχει αποδειχθεί ότι ο χώρος των σημάτων M-PAM της προηγούμενης σχέσης είναι:

$$\varphi(t) = \frac{g(t)}{\sqrt{\varepsilon_g}}$$

ενώ όταν έχουμε binary PAM η ενέργεια παλμού βασικής ζώνης είναι ίση με τη ενέργεια του bit ($\varepsilon_g = \varepsilon_b$).

Επομένως προκύπτει πως οι κυματομορφές που προκύπτουν είναι:

$$s_i(t) = (2i - M - 1) \cdot \sqrt{\varepsilon_g} \cdot \varphi(t), \text{ για } i = 1, 2, \dots, M$$

καθώς τα σύμβολα θα είναι:

$$s_i = \{(2i - M - 1) \cdot \sqrt{\varepsilon_g}\}$$

Αντίστοιχα για το BPAM είναι:

$$s_i = \{-\sqrt{2}, +\sqrt{2}\}$$

και για το 4-PAM:

$$s_i = \{-3\sqrt{2}, -\sqrt{2}, +\sqrt{2}, +3\sqrt{2}\}.$$

Η ζωνοπερατή έκδοσή του, είναι γνωστή και ως *Διαμόρφωση Μεταλλαγής Μετατόπισης Πλάτους* (*Amplitude Shift Keying – ASK*), η οποία υλοποιείται με τον πολλαπλασιασμό του σήματος βασικής ζώνης με ένα φέρον. Έτσι σε αυτή τη περίπτωση για $i=1, \dots, M$ προκύπτει ότι:

$$u_i(t) = \begin{cases} (2i - M - 1)g(t) \cdot \cos 2\pi f_c t, & t \in [0, T] \\ 0, & \text{αλλού.} \end{cases}$$

Επιπλέον η συνάρτηση βάσης του χώρου σημάτων του M-PAM υπολογίζεται:

$$\varphi(t) = \frac{\sqrt{2}g(t) \cos 2\pi f_c t}{\sqrt{\varepsilon_g}}$$

Όταν το σήμα βασικής ζώνης μεταφέρεται σε υψηλότερες συχνότητες μέσω διαμόρφωσης, η συνολική ενέργεια κατανέμεται τόσο στη θετική όσο και στην αρνητική πλευρά του φάσματος της συχνότητας. Αυτή η συμμετρία έχει ως αποτέλεσμα, η ενέργεια του ζωνοπερατού σήματος να είναι το μισό της ενέργειας του σήματος βασικής ζώνης. Δηλαδή, η ενέργεια κάθε ανεξάρτητου μέρους του φάσματος είναι η μισή.

Το ζωνοπερατό M-PAM μεταφέρει την ίδια πληροφορία με το βασικό M-PAM, αλλά μεταφέρεται σε υψηλότερη συχνότητα. Ο αστερισμός, οι περιοχές απόφασης, και η πιθανότητα σφάλματος μένουν ίδιες. Η διαφορά έγκειται κυρίως στο γεγονός ότι το σήμα βρίσκεται σε διαφορετική περιοχή του φάσματος. Μια διαφορά ακόμα έγκειται στη περίπτωση συστήματος πομπού και δέκτη, όπου ο πομπός χρειάζεται πολλαπλασιασμό με το φέρον και ο δέκτης απαιτεί ανάκτηση του φέροντος. [3]

1.6.2 Διαμόρφωση Ολίσθησης Συχνότητας (Frequency Shift Keying - ASK):

Σε αυτή τη μέθοδο διαμόρφωσης η πληροφορία αντιστοιχείται στη συχνότητα των ημιτονοειδών κυματομορφών. Εδώ τα σύμβολα είναι πλήθους $M=2^n$, διάρκειας T και αποτελούνται από n bits το καθένα.

Το σήμα για $i=1,2,\dots,M$ θα είναι της μορφής:

$$s_i(t) = \begin{cases} \frac{\sqrt{2\varepsilon_s}}{T} \cos(2\pi f_i t + \theta_i), & t \in [0, T] \\ 0, & \text{αλλού.} \end{cases}$$

Προκύπτει ότι όλες οι κυματομορφές έχουνε ίδια ενέργεια ίση με:

$$\varepsilon_s = \int_0^T |s_i(t)|^2 dt.$$

Επιπλέον αποδεικνύεται ότι ο χώρος των σημάτων είναι της μορφής:

$$\varphi_i(t) = \sqrt{\frac{2}{T}} \cos(2\pi f_i t + \theta_i).$$

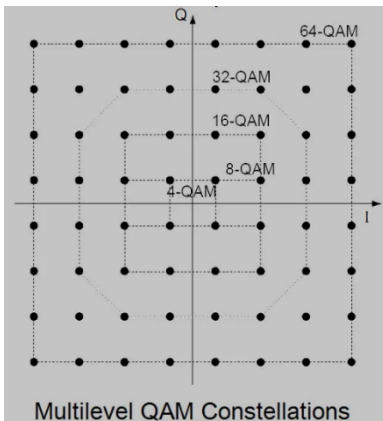
1.6.3 Διαμόρφωση Ολίσθησης Φάσης (Phase Shift Keying - PSK):

Στη συγκεκριμένη τεχνική διαμόρφωσης η πληροφορία μεταφέρεται στη φάση ημιτονοειδών κυματομορφών ίδιας συχνότητας και πλάτους. Χρησιμοποιείται πλήθος $M=2^n$ κυματομορφών και για $i=1,\dots,M$ προκύπτει ότι:

$$s_i(t) = \begin{cases} \sqrt{\frac{2\varepsilon_s}{T}} \cos(2\pi f_c t + \theta_i), & t \in [0, T] \\ 0, & \text{αλλού.} \end{cases}$$

όπου $\theta_i = 2\pi * [(i-1)/M]$. Επίσης προκύπτει ότι όλες οι κυματομορφές $s_i(t)$ έχουν σταθερή περιβάλλουσα ίση με: $V(t) = \sqrt{\frac{2\varepsilon_s}{T}}$ και ίδια ενέργεια συμβόλου ε_s .

1.6.4 Διαμόρφωση Μ-αδικής Ορθογωνικής μεταλλαγής Πλάτους -QAM (Quadrature Amplitude Modulation):



Η μέθοδος αυτή αποτελεί ένα συνδυασμό των τεχνικών PSK και ASK, όπου τα ψηφιακά σύμβολα έχουν διαφορετικά πλάτη και φάσεις. Εναλλακτικά μπορεί να θεωρηθεί ότι προκύπτει από δύο διαμορφώσεις πλάτους, με τις φάσεις να διαφέρουν κατά 180 μοίρες. Δηλαδή σχηματίζονται σε δύο κάθετους μεταξύ τους άξονες, γεγονός πάνω στο οποίο οφείλεται και η ονομασία της διαμόρφωσης. Το σήμα διαχωρίζεται σε δύο συνιστώσες, τη συνιστώσα In-phase (I), που είναι το κομμάτι του σήματος που μεταδίδεται χωρίς μετατόπιση φάσης, και τη συνιστώσα Quadrature (Q), που είναι το κομμάτι του σήματος που μεταδίδεται με μετατόπιση φάσης 90 μοιρών.

Εικόνα 2: Διάγραμμα Αστερισμού Διαμόρφωσης M-QAM.

Το παραπάνω διάγραμμα αστερισμού της M-QAM αναπαριστά τις πιθανές τιμές της φάσης και του πλάτους του σήματος ως σημεία στο επίπεδο. [5]

Η γενική μορφή της είναι:

$$s_{ij}(t) = \frac{\sqrt{2E_s}}{T_s} \cdot A_i \cdot \cos(2\pi f_0 t + \varphi_j), \text{ για } i = 1, \dots, M_1 \text{ και } j = 1, \dots, M_2, 0 \leq t < T.$$

1.7 Η Αποδιαμόρφωση (Demodulation)

Η διαδικασία της αποδιαμόρφωσης είναι ουσιαστικά η αντίστροφη της διαμόρφωσης, όπου το σήμα πληροφορίας (όπως ήχος, δεδομένα ή βίντεο) ανακτάται από το μεταδιδόμενο σήμα που έχει μεταδοθεί στη συχνότητα φέροντος και κωδικοποιηθεί αναλόγως με τη ανάλογη μέθοδο διαμόρφωσης. Υπάρχουν είδη αποδιαμορφωτών ανάλογα με το είδος της διαμόρφωσης που έχει χρησιμοποιηθεί για τη μετάδοση του σήματος.

Μερικοί από αυτούς είναι:

- Αποδιαμορφωτής Πλάτους (AM Demodulator): Χρησιμοποιείται για την αποδιαμόρφωση σημάτων διαμορφωμένων κατά πλάτος (AM).
- Αποδιαμορφωτής Συχνότητας (FM Demodulator): Χρησιμοποιείται για την αποδιαμόρφωση σημάτων διαμορφωμένων κατά συχνότητα (FM), όπου η πληροφορία μεταφέρεται μέσω της διακύμανσης της συχνότητας του φέροντος σήματος.
- Αποδιαμορφωτής Φάσης (PM Demodulator): Αποδιαμορφώνει σήματα διαμορφωμένα κατά φάση (PM).

Οι αποδιαμορφωτές που απαιτούνται για ψηφιακά διαμορφωμένα σήματα θεωρούνται πιο σύνθετοι από τους παραπάνω.

Κάποιοι από αυτούς είναι:

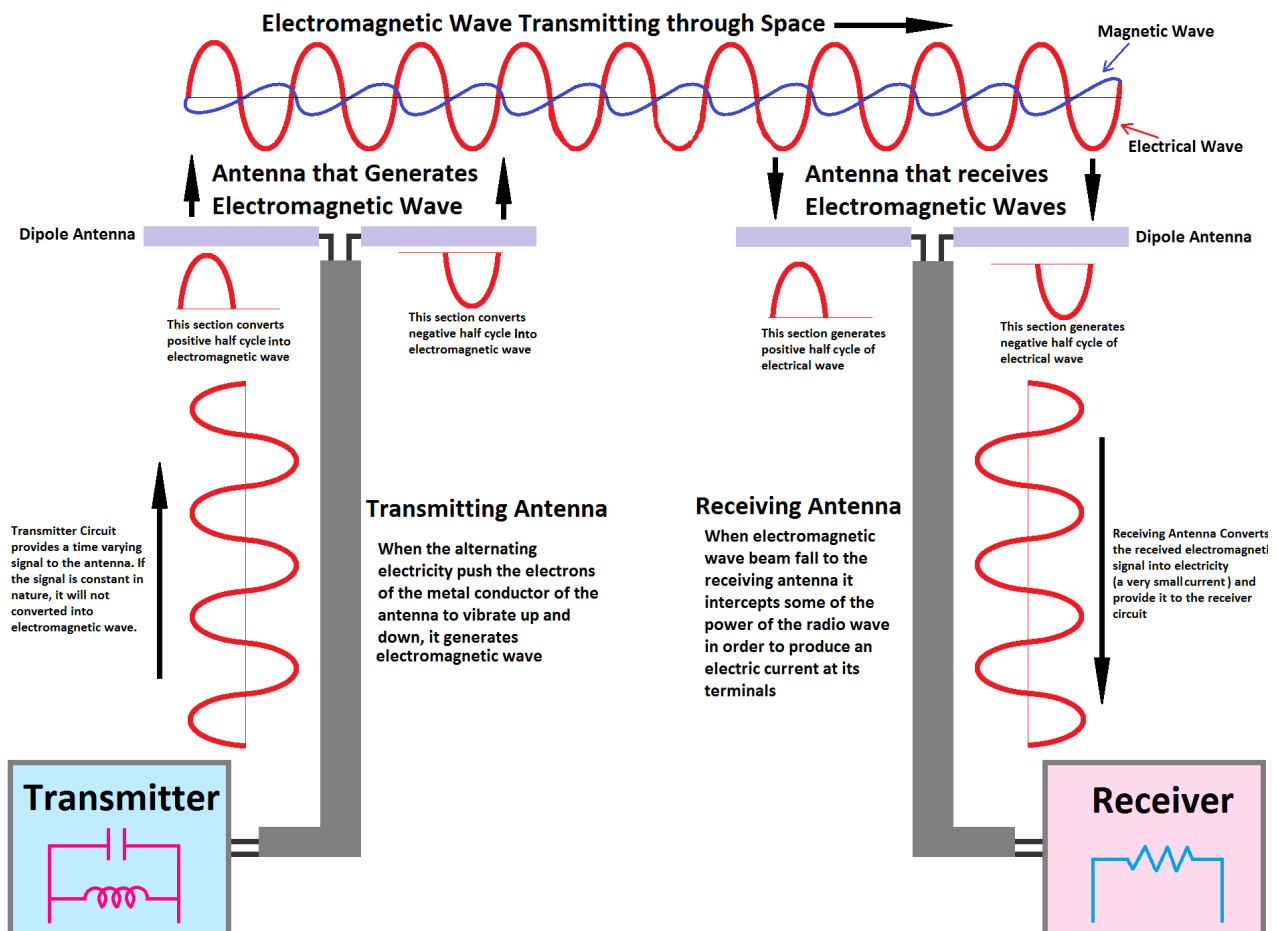
- Αποδιαμορφωτής ASK: Φιλτράρει το λαμβανόμενο σήμα χρησιμοποιώντας ένα ζωνοπερατό φίλτρο για να αφαιρέσει τυχόν θόρυβο και ανεπιθύμητες συχνότητες. Έπειτα ένας ανιχνευτής πλάτους μετράει τις αλλαγές στο πλάτος του σήματος και αποφασίζει σε ποιες περιοχές απόφασης ανήκουν για το αντίστοιχο σύμβολο.
- Αποδιαμόρφωση FSK: Φιλτράρει το λαμβανόμενο σήμα χρησιμοποιώντας δύο ζωνοπερατά φίλτρα που αντιστοιχούν στις δύο συχνότητες του "1" και του "0".
- Αποδιαμόρφωση PSK: Συγκρίνει τη φάση του λαμβανόμενου σήματος με τη φάση ενός σήματος αναφοράς. Όταν γίνεται αλλαγή στη φάση, την αναγνωρίζει και την μεταφράζει ως "0" ή "1". Στην BPSK (Binary PSK) υπάρχουν δύο φάσεις διαφοράς 180 μοιρών, και στη QPSK (Quadrature PSK), χρησιμοποιούνται τέσσερις διαφορετικές φάσεις, διασφαλίζοντας τη δυνατότητα για κωδικοποίηση δύο bit πληροφορίας ανά αλλαγή της φάσης.
- Αποδιαμόρφωση QAM: Χρησιμοποιεί δύο εισόδους μια για τη συνιστώσα φάσης και μια για τον πλάτους. Έπειτα αναλύονται και μεταφράζεται το αποτέλεσμα σε ένα σύνολο από δυαδικά bits. [3] [5]

2

Η ασύρματη επικοινωνία

2.1 Ο ασύρματος διάυλος επικοινωνίας.

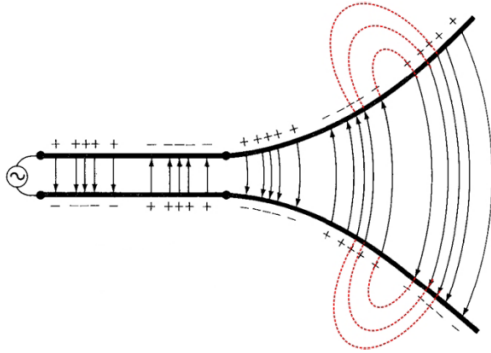
Η αρχιτεκτονική ενός ασύρματου συστήματος επικοινωνίας υπάγεται και αυτή στο γενικό μοντέλο των Shannon και Weaver που αναφέρθηκε παραπάνω. Οι κύριες διαφορές ενός ασύρματου διαύλου επικοινωνίας και ενός επίγειου δικτύου επικοινωνίας με οπτικές ίνες ή συμβατικά καλώδια χαλκού, εντοπίζονται στο κομμάτι του τρόπου εκπομπής και λήψης ενός σήματος, καθώς και τη φύση του καναλιού μετάδοσης, μαζί με τα διάφορα φαινόμενα που επιδρούν σε αυτό. Δηλαδή καλούνται να αντιμετωπιστούν οι παράγοντες που προκύπτουν λόγω των διατάξεων των κεραιών που λειτουργούν ως πομποί ή δέκτες και οι παράγοντες που προκύπτουν λόγω της μεταφοράς της πληροφορίας μέσα από το μέσο διάδοσης (π.χ.: καλώδιο, οπτική ίνα, ατμόσφαιρα, κ.α.). [6][7]



Εικόνα 3: Μοντέλο ασύρματης ζεύξης.

2.1.1 Η λειτουργία της κεραίας.

Η κεραία ουσιαστικά δίνει τη δυνατότητα να μετατραπεί ένα *καθοδηγούμενο σήμα* (*guided wave*), το οποίο μεταφέρεται μέσω ενός κυματοδηγού, γραμμής μεταφοράς ή καλώδιο τροφοδοσίας, σε ένα κύμα το οποίο μπορεί και ταξιδεύει στον ελεύθερο χώρο. Υπάρχει η δυνατότητα να γίνει και το αντίστροφο, δηλαδή να λαμβάνει ένα ηλεκτρομαγνητικό κύμα που ταξιδεύει στον ελεύθερο χώρο και να το μετατρέψει σε καθοδηγούμενο, ώστε στη συνέχεια να πραγματοποιηθεί αποδιαμόρφωση ή κάποια άλλη επεξεργασία απαραίτητη για την ανάκτηση της πληροφορίας.



Εικόνα 4: Αναπαράσταση κεραίας.

Πιο συγκεκριμένα μια κεραία αποτελείται από ένα σύστημα αγωγών με κατάλληλο σχήμα, διεγερόμενο κατάλληλα από ρεύματα υψηλής συχνότητας. Όπου κατά αυτό το τρόπο δημιουργεί ηλεκτρομαγνητικά κύματα ίδιας συχνότητας στο περιβάλλοντα χώρο. Μέσω αυτών επιτυγχάνεται η μετάδοση της ηλεκτρομαγνητικής ενέργειας. Το ηλεκτρομαγνητικό πεδίο της κεραίας που σχηματίζεται, εξαρτάται από τη πυκνότητα του ρεύματος που επάγεται στην επιφάνεια της κεραίας, δηλαδή εξαρτάται άμεσα με τη γεωμετρία της κατασκευής της κεραίας. Επομένως τα κύρια χαρακτηριστικά που απασχολούν είναι ο τρόπος που διεγείρουμε τις κεραίες, η συχνότητα και το σχήμα τους.[6][7]

2.1.2 Φαινόμενα που επηρεάζουν το σήμα κατά τη διάδοση.

Το σήμα καθώς ταξιδεύει προς τον προορισμό του, είναι πιθανό να επηρεαστεί από μια ποικιλία φαινομένων, έτσι το λαμβανόμενο σήμα είναι παραμορφωμένο κατά πλάτος, κατά φάση και σε ορισμένες περιπτώσεις κατά συχνότητα. Ορισμένα από αυτά παρουσιάζονται παρακάτω:

- **Ανάκλαση (Reflection):** Όταν τα ηλεκτρομαγνητικά κύματα πέφτουν σε λείες επιφάνειες πολύ μεγαλύτερες από το μήκος κύματός τους.
- **Απορρόφηση (Absorption):** Όταν παρατηρείται μείωση του πλάτους του κύματος, λόγω μεταφοράς μέρους της ενέργειάς του σήματος στο μέσο διάδοσης.
- **Διάθλαση (Refraction):** Όταν ο δείκτης διάθλασης του μέσου διάδοσης μεταβάλλεται στο χώρο και παρατηρούνται αλλαγές στη κατεύθυνση διάδοσης του κύματος.
- **Περίθλαση (Diffraction):** Όταν τα κύματα καμπυλώνουν γύρω από φυσικά ή τεχνητά εμπόδια, λόγω της παρουσίας άλλων ηλεκτρομαγνητικών κυμάτων.
- **Σκέδαση (Scattering):** Όταν τα ηλεκτρομαγνητικά κύματα πέφτουν πάνω σε τραχιές επιφάνειες ή σε επιφάνειες με συγκρίσιμες διαστάσεις με το μήκος κύματός τους και διασκορπίζεται η ενέργειά τους σε διάφορες κατευθύνσεις.
- **Διαλείψεις (Fading):** Όταν παρατηρούνται μεταβολές στο πλάτος και την ισχύ του λαμβανόμενου σήματος. Χωρίζονται σε Μεγάλης, Μεσαίας και Μικρής Κλίμακας και υπάγονται στη κατηγορία πολλαπλασιαστικού θορύβου.
- **Φαινόμενα Doppler:** Όταν ο δέκτης πλησιάζει είτε απομακρύνεται με μια ταχύτητα v από τον πομπό, τότε αντιλαμβάνεται τις συχνότητες των λαμβανόμενων σημάτων ως αυξημένες ή μειωμένες. [6]

2.2 Μοντελοποίηση φαινόμενων διάδοσης.

2.2.1 Κανάλι Προσθετικού Λευκού Γκαουσιανού Θορύβου – Additive White Gaussian Noise (AWGN):

Πρόκειται για την πιο απλή πρακτική εκδοχή ενός ασύρματου δίαυλου επικοινωνίας. Είναι ένα γενικό μοντέλο καναλιού το οποίο χρησιμοποιείται για να προσομοιωθεί ο θερμικός θόρυβος που προκαλείται μέσα στο υλικό των ηλεκτρονικών διατάξεων από τη κίνηση των ηλεκτρονίων λόγω της θερμικής ενέργειας που αποκτά συνήθως ένας δέκτης. Τέτοιου είδους φαινόμενα μετά από μεγάλο πλήθος μετρήσεων δίνουν προσθετικό θόρυβο, με ένταση που ακολουθεί τη κανονική ή Γκαουσιανή κατανομή (Gaussian distribution). Ο θόρυβος προστίθεται στο σήμα χωρίς να το αλλοιώνει, δηλαδή το τελικό λαμβανόμενο σήμα είναι το αρχικό σήμα συν τον θόρυβο:

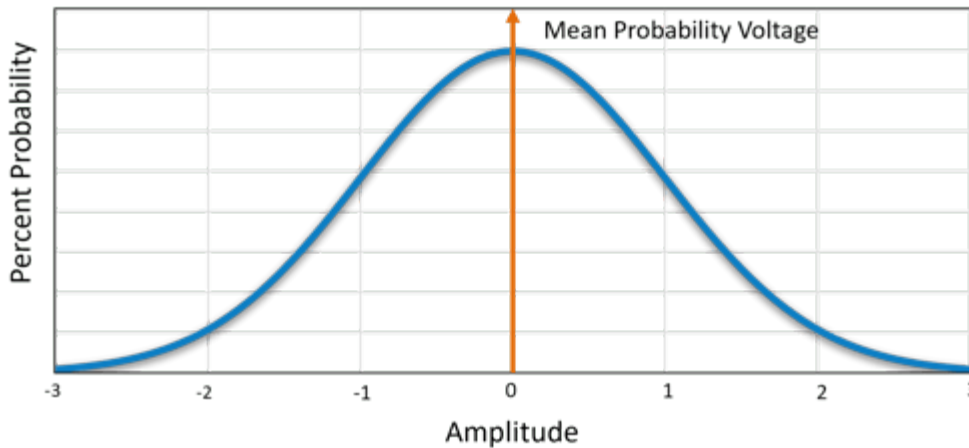
$$y(t) = x(t) + n(t)$$

όπου: $y(t)$ είναι το σήμα που λαμβάνεται, $x(t)$ είναι το αρχικό σήμα, $n(t)$ είναι ο θόρυβος.

Ονομάζεται Λευκός διότι η Φασματική Πυκνότητα Ισχύος (ΦΠΙ) του είναι σταθερή σε όλο το φάσμα συχνοτήτων δηλαδή έχει ίδια ενέργεια και ισχύ σε όλες τις συχνότητες. Η ΦΠΙ είναι της μορφής:

$$S_n(f) = \frac{N_0}{2}, f \in (-\infty, +\infty)$$

όπου N_0 είναι η ισχύς θορύβου ανά μονάδα εύρους ζώνης (W/Hz), και είναι σταθερή σε όλες τις συχνότητες.



Εικόνα 5: Αναπαράσταση Gaussian Κατανομής.

Οι τιμές του θορύβου σε κάθε χρονική στιγμή κατανέμονται σύμφωνα με την καμπύλη της κανονικής κατανομής, η οποία έχει μέσο όρο μηδέν και διακύμανση που αντιστοιχεί στην ισχύ του θορύβου. Έτσι η πιθανότητα να έχει ο θόρυβος μια συγκεκριμένη τιμή $n(t)$ δίνεται από την σχέση:

$$f(n) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{n^2}{2\sigma^2}}$$

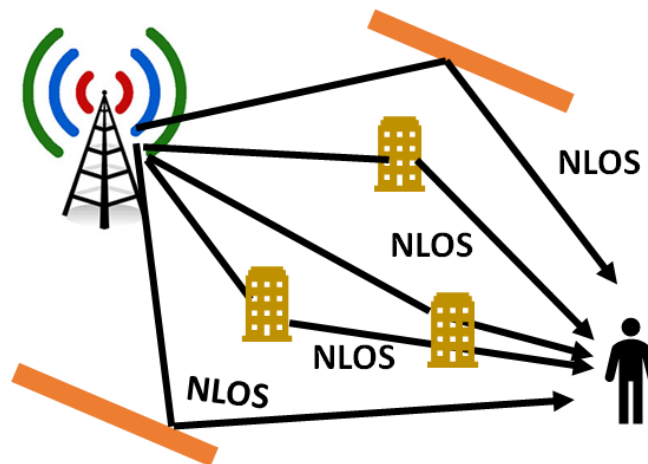
όπου σ^2 είναι η διακύμανση του θορύβου, που σχετίζεται με την ισχύ του και ο μέσος όρος είναι 0, επειδή ο θόρυβος είναι συμμετρικός γύρω από το μηδέν.

Ο λόγος που το συγκεκριμένο μοντέλο χρησιμοποιείται ευρέως στις τηλεπικοινωνίες είναι το γεγονός ότι σε πολλές πρακτικές εφαρμογές αντιμετωπίζονται μόνο κανάλια θορύβου, όπως μικροκυματικές ζεύξεις, δορυφορικές επικοινωνίες σε καλές καιρικές συνθήκες και κανάλια επικοινωνίας ευρείας ζώνης με ομοαξονικά καλώδια.

Σε άλλες περιπτώσεις καναλιών με φαινόμενα όπως διαλείψεις, πολλαπλές οδεύσεις κ.α., ο θόρυβος συνεχίζει να προστίθεται και να μοντελοποιείται ως AWGN. Έτσι θέτεται η καλύτερη περίπτωση καναλιού που αποτελεί ένα άνω όριο της επίδοσης του συστήματός, όταν υπάρχουν επιπλέον φαινόμενα που την επηρεάζουν.[3][6]

2.2.2 Κανάλι Rayleigh.

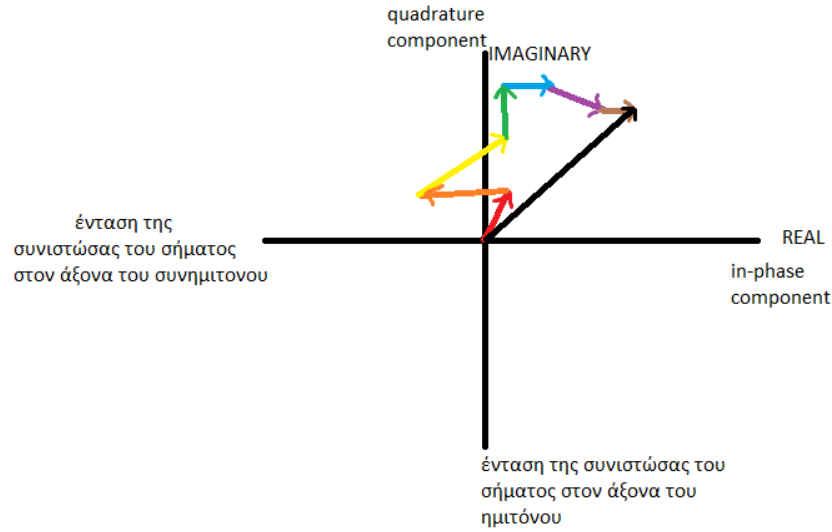
Το συγκεκριμένο μοντέλο χρησιμοποιείται ευρέως για την προσομοίωση πολυδιαδρομικών συνιστωσών σε ασύρματο δίαυλο. Πιο συγκεκριμένα, καλύπτει τη περίπτωση που δεν υπάρχουν απευθείας συνιστώσες. Δηλαδή δεν υπάρχει επικοινωνία με απευθείας με οπτική επαφή (No Line Of Sight - NLOS) όπως φαίνεται το παρακάτω σχήμα:



Εικόνα 6: Δίαυλος NLOS επικοινωνίας.

Το σήμα που φτάνει στον δέκτη έχει ακολουθήσει πολλαπλές διαδρομές λόγω ανακλάσεων σε εμπόδια (κτίρια, δέντρα, κ.λπ.). Αυτές οι πολυδιαδρομικές συνιστώσες αλληλοεπιδρούν μεταξύ τους, παρουσιάζοντας παρεμβολές που οδηγούν σε τυχαίες διακυμάνσεις του πλάτους του λαμβανόμενου σήματος.

Το λαμβανόμενο σήμα τη στιγμή t αποτελεί μια τυχαία συνισταμένη h_t , της οποίας μπορεί να σχεδιαστεί το διάγραμμα IQ (Εικόνα 7). Στο διάγραμμα φαίνονται τα επιμέρους διανύσματα (με χρώμα) από διαφορετικές διαδρομές και η λαμβανόμενη συνισταμένη τους (με μαύρο). Τα επιμέρους διανύσματα έχουν διαφορετική κατεύθυνση και μέτρο που είναι τυχαία. Επομένως η h_t θα αποτελείται από τυχαίες τιμές που ακολουθούν κανονική κατανομή στον άξονα I (Inphase component), δηλαδή $I \sim N(0, \sigma^2)$ και τυχαίες τιμές που ακολουθούν κανονική κατανομή στον άξονα Q (Quadrature component), δηλαδή $Q \sim N(0, \sigma^2)$.



Εικόνα 7: Διάγραμμα IQ πολυδιαδρομικής συνιστώσας από Rayleigh Fading Channel

Δηλαδή οι δύο ανεξάρτητες τυχαίες μεταβλητές (μία για κάθε άξονα) ακολουθούν Gaussian κατανομή. Επομένως η από κοινού Συνάρτηση Πυκνότητας Πιθανότητας (ΣΠΠ) ή Probability Distribution Function (PDF) θα έχει τη μορφή:

$$p(I, Q) = \frac{1}{2\pi\sigma^2} e^{-\frac{I^2+Q^2}{2\sigma^2}}$$

Το μέγεθος του λαμβανόμενου σήματος είναι το μέτρο του διανύσματος της ακτίνας r που σχηματίζεται από τις συνιστώσες I και Q . Από το πυθαγόρειο θεώρημα ισχύει ότι:

$$r = \sqrt{I^2 + Q^2}$$

Για να βρεθεί η ΣΠΠ του r πρέπει να αλλάξει το σύστημα συντεταγμένων από καρτεσιανές (I, Q) σε πολικές συντεταγμένες (r, θ) , άρα θα είναι:

$$r = \sqrt{I^2 + Q^2}$$

και

$$\theta = \tanh^{-1}\left(\frac{Q}{I}\right)$$

Για να γίνει η μετάβαση πρέπει να ληφθεί υπόψιν ο Ιακωβιανός πίνακας (Jacobian matrix) του μετασχηματισμού. Η σχέση μεταξύ των συναρτήσεων πυκνότητας πιθανότητας είναι:

$$p(r, \theta) = p(I, Q) \left| \frac{\partial(I, Q)}{\partial(r, \theta)} \right|$$

όπου ο Ιακωβιανός πίνακας για την μετάβαση είναι $\left| \frac{\partial(I, Q)}{\partial(r, \theta)} \right| = r$.

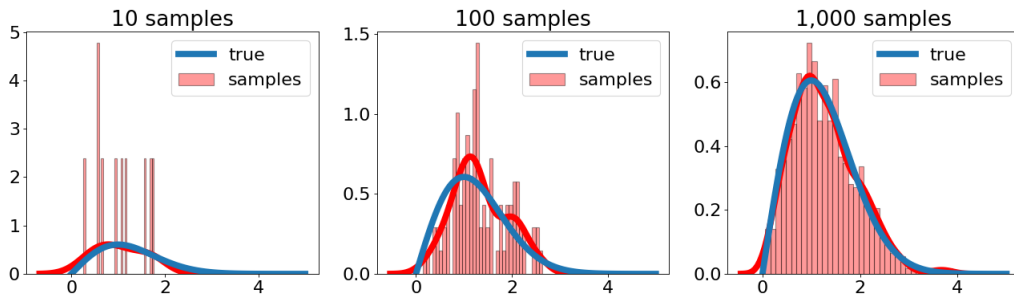
Μπορεί να παραλειφθεί η κατανομή της γωνίας θ , καθώς είναι ομοιόμορφα κατανεμημένη στο διάστημα $[0, 2\pi]$. Η ΣΠΠ της r δίνεται από τη σχέση:

$$p(r) = \int_0^{2\pi} p(r, \theta) d\theta \Rightarrow$$

Στη τελική μορφή της, η κατανομή Rayleigh γίνεται:

$$p(r) = \frac{r}{\sigma^2} e^{-\frac{r^2}{2\sigma^2}}$$

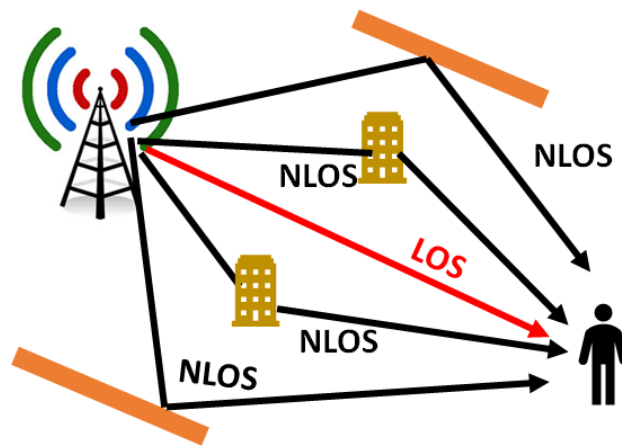
Η κατανομή Rayleigh εκφράζει την πιθανότητα, το μέγεθος r να λάβει μια συγκεκριμένη τιμή, όταν οι συνιστώσες I και Q είναι ανεξάρτητες κανονικές τυχαίες μεταβλητές με μέσο όρο 0 και διασπορά σ^2 . Παρακάτω απεικονίζονται πραγματικές μετρήσεις σε σύγκριση με τη θεωρητική μορφή της κατανομής.[6][8][9]



Εικόνα 8: Αναπαράσταση κατανομής Rayleigh.

2.2.3 Κανάλι Rice.

Σε αυτή τη περίπτωση το λαμβανόμενο σήμα αποτελείται από πολυδιαδρομικές συνιστώσες NLOS επικοινωνίας, όμως υπάρχει και μία συνιστώσα που επικοινωνεί με απευθείας οπτική επαφή (Line Of Sight - LOS). Η ισχύς της είναι μεγαλύτερη από τις υπόλοιπες. Ο δείκτης K (Rice Factor) εκφράζει τον λόγο της κυρίαρχης συνιστώσας (με LOS επικοινωνία) προς την συνισταμένη των πολυδιαδρομικών συνιστωσών. Όταν η κυρίαρχη συνιστώσα αρχίζει και έχει σημαντικά μεγαλύτερη ισχύ από τις υπόλοιπες ανακλώμενες διαδρομές, τότε από τη κατανομή Rayleigh γίνεται μετάβαση στη κατανομή Rice για να μοντελοποιηθεί καλύτερα το κανάλι. Αυτό είναι σημαντικό επειδή, για μικρές τιμές της κυρίαρχης συνιστώσας (μικρό K), η συμπεριφορά του καναλιού συνεχίζει να μοιάζει με τη περίπτωση του καναλιού Rayleigh.



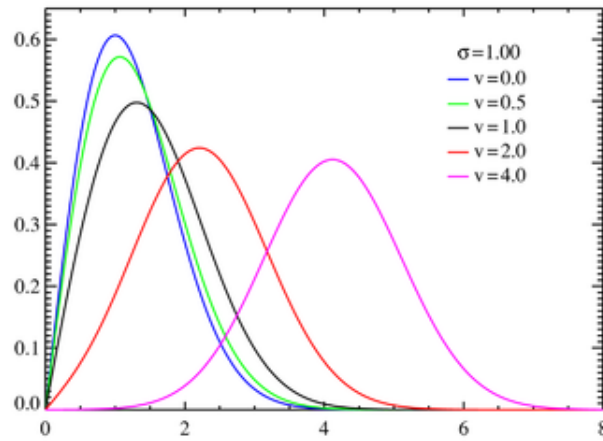
Εικόνα 9: Διάυλος με πολλαπλές συνιστώσες NLOS και μιας συνιστώσας LOS επικοινωνίας.

Στο αντίστοιχο διάγραμμα IQ η απευθείας συνιστώσα θα λαμβάνει σταθερή μετατόπιση s στον άξονα I, ενώ στον άξονα Q παραμένει τυχαία με μέση τιμή το 0. Άρα το I γίνεται κανονική τυχαία μεταβλητή με μέση τιμή s και διακύμανση σ^2 , και το Q παραμένει με μέση τιμή 0 και διακύμανση σ^2 . Ακολουθεί μετάβαση του συστήματος συντεταγμένων όπως στη περίπτωση του καναλιού Rayleigh, και προκύπτει η σχέση:

$$p(r) = \frac{r}{\sigma^2} e^{-(r^2+s^2)/2\sigma^2} \cdot I_0\left(\frac{rs}{\sigma^2}\right)$$

με $K = \frac{s^2}{2\sigma^2}$, όπου ο λόγος εκφράζει την ισχύ της άμεσης συνιστώσας s^2 σε σχέση με τη συνολική ισχύ της πολυδιαδρομικής συνισταμένης $2\sigma^2$.

Η μέθοδος αυτή ονομάζεται και Nakagami-Rice ή Nakagami-n. [6][8][9]



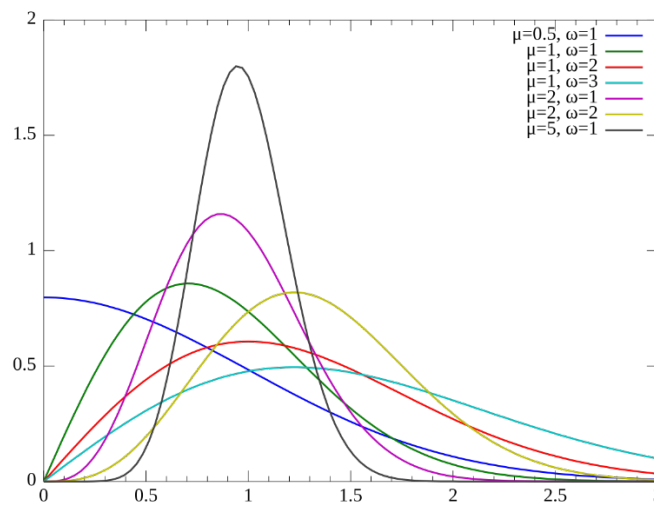
Εικόνα 10: Αναπαράσταση κατανομής Rice.

2.2.4 Κανάλι Nakagami-m.

Ορισμένοι ασύρματοι δίαυλοι δεν υπάγονται ούτε στη περίπτωση καναλιού Rayleigh, ούτε στη περίπτωση καναλιού Rice. Αν για παράδειγμα υπάρχουν 2 συνιστώσες με LOS επικοινωνία, συγκρίσιμες μεταξύ τους και ισχυρότερες από τις υπόλοιπες, τότε η προσέγγιση από τη κατανομή Rice δεν επαρκεί. Μια εναλλακτική τεχνική για αυτό το σενάριο είναι η κατανομή Nakagami-m. Είναι χρήσιμη σε περιπτώσεις όπου άλλα μοντέλα εξασθένησης, όπως το Rayleigh και το Rice, δεν προσφέρουν επαρκή ακρίβεια. Πρόκειται για μια πιο ευέλικτη κατανομή που μπορεί να προσαρμοστεί για να περιγράψει διάφορους τύπους εξασθένησης (fading) με τη χρήση της παραμέτρου m, η οποία καθορίζει τον βαθμό της εξασθένησης. Η ΣΠΠ της είναι:

$$p_R(r) = \frac{2}{\Gamma(m)} \left(\frac{m}{\Omega}\right)^m \cdot r^{2m-1} \cdot e^{-\frac{m}{\Omega}r^2}, r \geq 0.$$

όπου ο παράγοντας m καθορίζει της μορφή της κατανομής και λαμβάνει τιμές από το 0.5 έως το άπειρο. Όταν m=1 τότε το κανάλι προσεγγίζει τη περίπτωση του καναλιού Rayleigh. Όταν m>1 τότε προσεγγίζεται η περίπτωση του καναλιού Rice. Τέλος όταν το m τείνει στο άπειρο τότε υπάρχει LOS επικοινωνία. Το $\Omega=E(r^2)$ είναι η ροπή δεύτερης τάξης και $\Gamma(m)$ η συνάρτηση Gamma.[6][8][9]



Εικόνα 11: Αναπαράσταση κατανομής Nakagami-m.

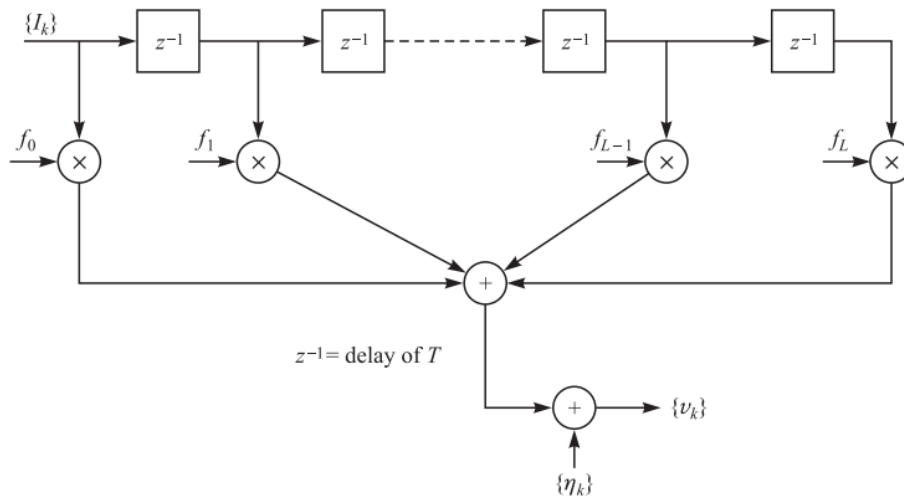
3

Συστήματα Ανάκτησης Σήματος

3.1 Τεχνικές Ισοστάθμισης (Equalization Techniques):

Στο προηγούμενο κεφάλαιο παρουσιάστηκαν μέθοδοι αναπαράστασης διαφορετικών ειδών καναλιών. Αυτοί οι μέθοδοι είναι χρήσιμοι καθώς με τη εφαρμογή τους, παράγουν δεδομένα τα οποία έχουν παραμορφωθεί με παρόμοιο τρόπο όπως θα συνέβαινε σε πραγματικές συνθήκες. Ο δέκτης καλείται να επεξεργαστεί τα δεδομένα αυτά έτσι ώστε να ανακτηθεί το σήμα πληροφορίας και να επιτευχθεί αξιόπιστη επικοινωνία. Η ισοστάθμιση είναι η διαδικασία αντιστάθμισης των επιπτώσεων των παραμορφώσεων και της διασυμβολικής παρεμβολής (*Inter-Symbol Interference - ISI*) που προκαλείται λόγω της διάδοσης σημάτων μέσα από ένα κανάλι επικοινωνίας. Το πρόβλημα της ISI προκύπτει από τη παραμόρφωση πλάτους και φάσης. Η ισοστάθμιση επιτρέπει τη βελτίωση της ποιότητας του σήματος και τη μείωση των σφαλμάτων. Η πρώτη και πιο απλή μέθοδος ισοστάθμισης είναι η χρήση γραμμικών φίλτρων, θα παρουσιαστούν και οι μη γραμμικοί ισοσταθμιστές DFE (Decision Feedback Equalizers) και VNLE (Volterra Nonlinear Equalizers).

3.1.1 Tapped Delay Line (TDL) ή Finite Impulse Response (FIR) Filter:



Εικόνα 12: Μοντέλο καναλιού διακριτού χρόνου με διασυμβολική παρεμβολή και AWGN.

Η μοντελοποίηση ενός καναλιού με διασυμβολική παρεμβολή, είναι εφικτή με τη χρήση ενός γραμμικού φίλτρου *FIR* (*Finite Impulse Response*) μήκους L . Το μήκος του εξαρτάται από τα μελλοντικά ή τα παρελθοντικά σύμβολα ανάλογα την υλοποίηση. Το εικονιζόμενο μοντέλο αποτελείται από μια σειρά **διακριτών καθυστερήσεων (tapped delays)** που δεσμεύουν

παρελθοντικές τιμές ενός σήματος, οι οποίες στη συνέχεια πολλαπλασιάζονται με τα αντίστοιχα **βάρη των διακριτών συντελεστών (tap weight)**. Η έξοδος του φίλτρου δίνεται από τη σχέση:

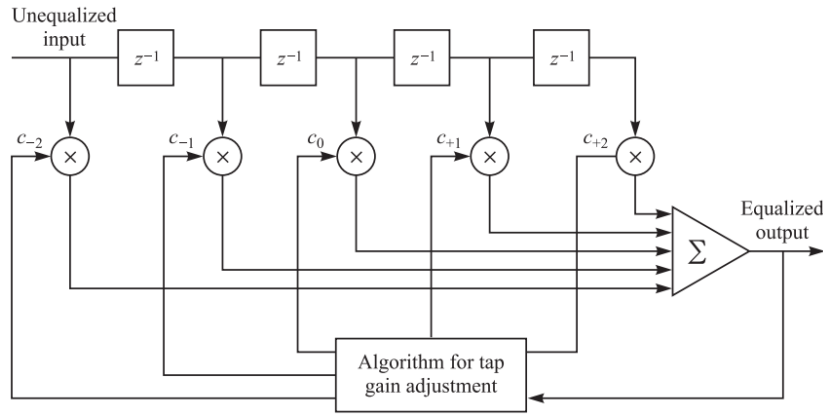
$$v_k = \sum_{n=0}^L f_n I_{k-n} + \eta_k$$

όπου v_k είναι έξοδος του συστήματος στο χρονικό βήμα k , I_{k-n} η είσοδος του συστήματος στο χρονικό βήμα $k-n$, f_n τα βάρη των taps για κάθε βήμα, η_k προσθετικός θόρυβος και L ο συνολικός αριθμός των taps. [9][10]

3.1.2. Ισοσταμιστής πρόσθιας τροφοδότησης (Feedforward Equalizer - FFE):

Οι γραμμικοί ισοσταθμιστές χωρίζονται σε **ισοσταθμιστές προϋθμίσης (Preset Equalizers)** και **προσαρμοστικούς ισοσταθμιστές (Adaptive Equalizers)**. Αν το κανάλι είναι χρονικά σταθερό, οι διακριτοί συντελεστές και τα βάρη τους παραμένουν σταθεροί ως παράμετροι, ενώ αν το κανάλι είναι χρονικά μεταβαλλόμενο, οι παράμετροι αυτοί είναι αναγκαίο να προσαρμόζονται συνεχώς. Άρα όταν οι παράμετροι ενός γραμμικού φίλτρου FIR προσαρμόζονται δυναμικά με βάση κάποιο κριτήριο (π.χ. την ελαχιστοποίηση του σφάλματος), τότε γίνεται προσαρμοστικό φίλτρο.

Όταν η κρουστική απόκριση του καναλιού μεταβάλλεται με το χρόνο, το προσαρμοστικό φίλτρο του δέκτη γίνεται φίλτρο μεταβαλλόμενου χρόνου. Στην περίπτωση αυτή, ο συνδυασμός των χρονικών μεταβολών του καναλιού και του προσαρμοσμοστικού φίλτρου, δημιουργούν ένα φίλτρο διακριτού χρόνου με χρονικά μεταβλητούς συντελεστές και χρονικά μεταβλητά φαινόμενα διασυμβολικής παρεμβολής, τα οποία μπορούν να μοντελοποιηθούν από ένα φίλτρο όπως αυτό της παρακάτω εικόνας το οποίο αποτελεί το μοντέλο ενός **ισοσταθμιστή πρόσθιας τροφοδότησης (FFE)**, όπου τα βάρη των συντελεστών μεταβάλλονται χρονικά βάσει κάποιου κριτηρίου:



Εικόνα 13: Γραμμικό εγκάρσιο φίλτρο προσαρμοστικών συντελεστών.

Όταν οι διακριτοί συντελεστές που εφαρμόζονται στις εκδοχές του λαμβανόμενου σήματος έχουν καθυστέρηση ίση με τη περίοδο T , τότε πρόκειται για έναν **ισοσταθμιστή απόστασης συμβόλων (symbol-spaced equalizer)**, ενώ αν η καθυστέρηση ισούται με υποπολλαπλάσιο του T , τότε πρόκειται για **κλασματικό ισοσταθμιστή απόστασης (fractionally spaced equalizer)**. Η έξοδος του φίλτρου εκφράζεται από τη σχέση:

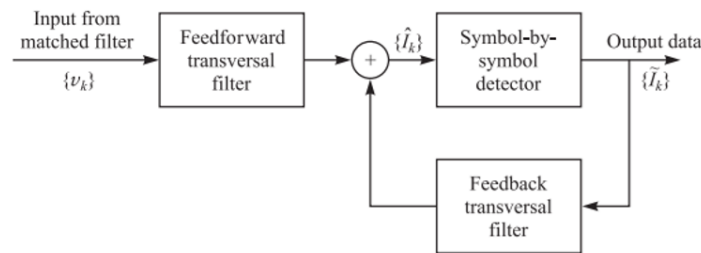
$$\hat{I}_k = \sum_{j=-K}^K c_j v_{k-j}$$

όπου $\{c_j\}$ είναι οι $2K+1$ μιγαδικοί συντελεστές βάρους, $\hat{I}_k$ η εκτιμώμενη τιμή κβαντισμένη στο κοντινότερο σύμβολο πληροφορίας ώστε να παρθεί το αναμενόμενο $\tilde{I}_k$. Αν το $\tilde{I}_k$ δεν είναι ίδιο με το εκπεμπόμενο σύμβολο I_k τότε υπάρχει σφάλμα.

Εφαρμόζονται αρκετά κριτήρια για τη αποδοτικότερη ρύθμιση των συντελεστών c_j . Το κριτήριο ελάχιστου μέσου τετραγωνικού σφάλματος (*Minimum Mean Square Error Criterion - MMSE*) είναι το πιο αποδοτικό καθώς, ελαχιστοποιεί το μέσο τετραγωνικό σφάλμα της εξόδου του ισοσταθμιστή σε σχέση με την λαμβανόμενη ακολουθία και λαμβάνει υπόψη του τόσο τη διασυμβολική παρεμβολή όσο και τον προσθετικό θόρυβο. Ωστόσο ο υπολογισμός της πιθανότητας αυτής γίνεται σταδιακά μια πιο περίπλοκη μη γραμμική συνάρτηση του c_j . Κανάλια πολλαπλών διαδρομών είναι δύσκολο να ισοσταθμιστούν με τον FFE. Στο πεδίο της συχνότητας, αυτά τα κανάλια προκαλούν μεγάλες εναλλαγές πλάτους, έτσι όταν γίνεται αντιστροφή του σήματος με σκοπό την ισοστάθμιση, προκαλείται ενίσχυση του θορύβου. Ο μη γραμμικός ισοσταθμιστής ανάδρασης απόφασης, ο οποίος αναλύεται στη συνέχεια, έχει πολύ καλύτερη απόδοση σε τέτοιου είδους κανάλια. [9][10][11]

3.1.3. Ισοσταθμιστής ανάδρασης απόφασης (*Decision Feedback Equalizer - DFE*):

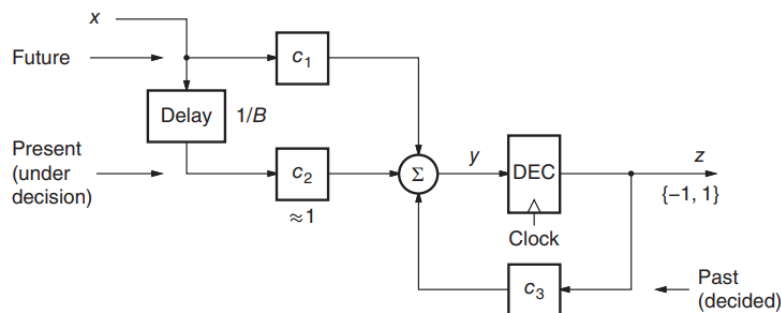
Ο μη γραμμικός ισοσταθμιστής ανατροφοδότησης απόφασης (*DFE*) αποτελεί μια συνηθέστερη και πιο ισχυρή εναλλακτική.



Εικόνα 14: Γενικό μοντέλο αρχιτεκτονικής ενός DFE..

Το γενικό μοντέλο (Εικόνα 14) αποτελείται από δύο γραμμικά φίλτρα FIR, το πρώτο είναι *φίλτρο πρόσθιας τροφοδότησης (feedforward filter)* και το δεύτερο *φίλτρο ανάδρασης (feedback filter)*.

Ο εικονιζόμενος DFE (Εικόνα 15), διαθέτει δύο **διακριτούς συντελεστές πρόσθιας τροφοδότησης (feedforward taps)** που τροφοδοτούνται από το σήμα εισόδου, και ένα **διακριτό συντελεστή ανάδρασης (feedback tap)** που τροφοδοτείται από την έξοδο z του κυκλώματος λήψης αποφάσεων. Η έξοδος y αποτελεί το άθροισμα των εξόδων των δύο γραμμικών φίλτρων FIR που τροφοδοτεί το κύκλωμα λήψης απόφασης.



Εικόνα 15: Η αρχιτεκτονική ενός DFE με 2 feedforward taps και 1 feedback tap.

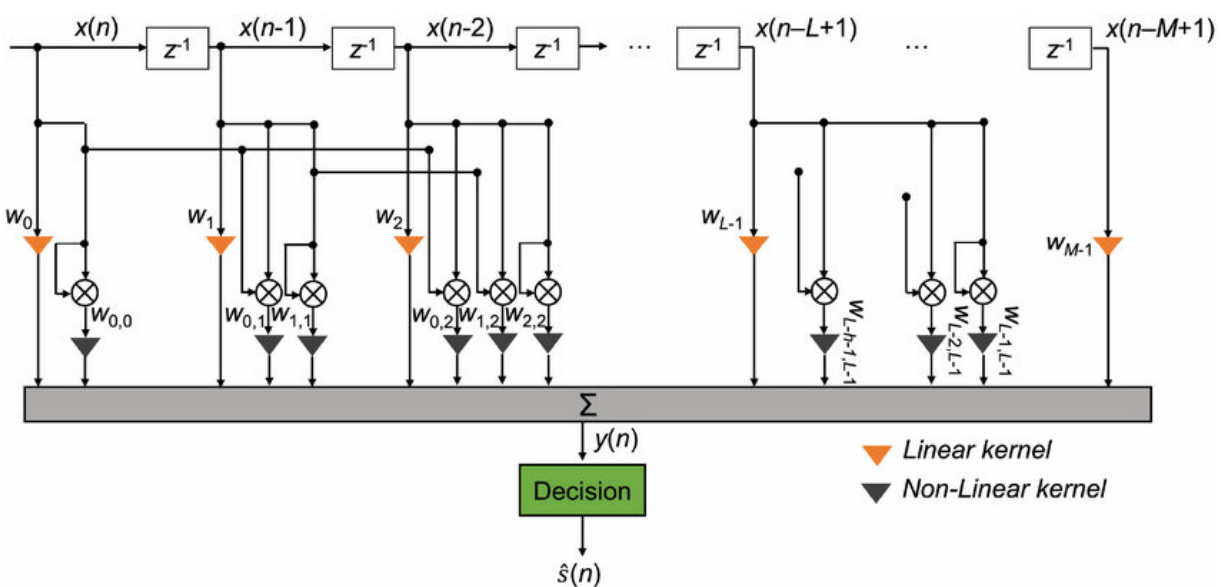
Ο DFE αντιμετωπίζει την διασυμβολική παρεμβολή αφαιρώντας τόσο την **παρεμβολή από προηγούμενα σύμβολα (precursors)**, όσο και την **παρεμβολή από επόμενα σύμβολα (postcursors)**. Για την αντιμετώπιση των precursors, ο DFE χρησιμοποιεί μηχανισμό ανάλογο με του FFE. Το πλεονέκτημά του, έγκειται στο γεγονός ότι για τα postcursors, χρησιμοποιεί την έξοδο του κυκλώματος απόφασης αντί για την έξοδο ενός γραμμικού φίλτρου. Είναι πολύ αποδοτικός επειδή χρησιμοποιεί τις αποφάσεις που έχουν παρθεί, οι οποίες θεωρούνται καθαρές από θόρυβο και ISI. Αυτό σημαίνει ότι, αν η απόφαση για ένα προηγούμενο σύμβολο είναι σωστή, η ISI μπορεί να αντισταθμιστεί εντελώς, χωρίς να προστεθεί θόρυβος. Η έξοδός του είναι της μορφής:

$$\hat{I}_k = \sum_{j=-K_1}^0 c_j v_{k-j} + \sum_{j=1}^{K_2} c_j \tilde{I}_{k-j}$$

όπου v_{k-j} , τα λαμβανόμενα δείγματα από το κανάλι, $\tilde{I}_{k-j}$ οι προηγούμενες αποφάσεις (*feedback component*), c_j οι συντελεστές βάρους, K_1 το μέγιστο χρονικό όριο για τα δείγματα από το κανάλι (feedforward φίλτρου), K_2 το μέγιστο χρονικό όριο για τις προηγούμενες αποφάσεις (feedback φίλτρου).

Τελικά στη περίπτωση του DFE, εμφανίζεται το πρόβλημα της *διάδοσης σφάλματος (error propagation)*. Δηλαδή αν γίνει ένα λάθος στο κύκλωμα απόφασης, τότε το λάθος αυτό θα μεταφερθεί και θα δημιουργήσει περισσότερη ISI στο επόμενο σύμβολο, οδηγώντας σε ακόμα περισσότερα λάθη στην αλληλουχία. Παρά το μειονέκτημα αυτό, τα πλεονεκτήματά του DFE συνήθως υπερτερούν αυτά του FFE. Επειδή το κύκλωμα αποφάσεων περιλαμβάνεται στη δομή του DFE και επηρεάζει τη λειτουργία του, ο DFE θεωρείται μη γραμμικός ισοσταθμιστής. Επιπλέον, οι περισσότερες πρακτικές εφαρμογές απαιτούν την ισοστάθμιση πιο σύνθετων καναλιών με μεγαλύτερες χρονικές αποκρίσεις, άρα οι DFE συχνά χρειάζονται περισσότερους από τρεις διακριτούς συντελεστές (taps) για είναι αποδοτικοί.[9][11][12]

3.1.4. Μη Γραμμικός Ισοσταθμιστής Volterra (Volterra Nonlinear Equalizer - VNLE):



Εικόνα 16: Μη γραμμικός ισοσταθμιστής Volterra.

Η μη γραμμική απόκριση ενός συστήματος δεν μπορεί να περιγραφεί μόνο με βάση την κρουστική απόκριση, καθώς με αυτό το τρόπο μπορούν να αντιμετωπιστούν μόνο οι γραμμικές παρεμβολές της ISI. Οι μαθηματικές σειρές Volterra χρησιμοποιούνται για να μοντελοποιήσουν τα μη γραμμικά φαινόμενα συστημάτων που παρουσιάζουν φαινόμενα μνήμης, δηλαδή υπάρχει συσχέτιση της τρέχουσας τιμής του σήματος με προηγούμενες τιμές του.

Ο μη γραμμικός ισοσταθμιστής Volterra (VNLE) συνδυάζει ένα γραμμικό φίλτρο και διάφορες πολυωνυμικές εκφράσεις, οι οποίες συσχετίζουν τα τρέχοντα με τα προηγούμενα σύμβολα της εισόδου. Οι γραμμικοί πυρήνες (*linear kernels*) πραγματοποιούν στην αντιστάθμιση των γραμμικών παρεμβολών. Οι γραμμικοί συντελεστές βάρους $w_{m1}(n)$ εφαρμόζονται στην τρέχουσα και προηγούμενη τιμή εισόδου, παρέχοντας τη βασική γραμμική αντιστάθμιση. Οι μη γραμμικοί πυρήνες (*nonlinear kernels*) πραγματοποιούν την αντιστάθμιση των μη γραμμικών παρεμβολών.

Οι γραμμικοί πυρήνες αντιστοιχούν στη παραγωγή του πρώτου όρου της σειράς Volterra. Έχουν παρόμοια λειτουργία με τα γραμμικά φίλτρα FIR, τα οποία προσαρμόζονται απευθείας σε ένα σήμα και στις καθυστερήσεις του. Ο γραμμικός όρος που προκύπτει είναι της μορφής:

$$h_1(0) x(n) + h_1(1) x(n-1) + h_1(2) x(n-2) + \dots$$

Όπως φαίνεται στο διάγραμμα του VNLE (Εικόνα 16), οι γραμμικοί όροι υπολογίζονται για κάθε διακριτό συντελεστή (*tap*) με τα αντίστοιχα βάρη:

$$w_0, w_1, \dots, w_{(M-1)}$$

Κάθε συντελεστής εφαρμόζεται στην αντίστοιχη καθυστέρηση του σήματος και προστίθεται στην έξοδο $y(n)$.

Οι μη γραμμικοί πυρήνες (*Non-Linear Kernels*) συνδέονται με πολλές τιμές από διαφορετικές διακριτές καθυστερήσεις (*tapped delays*). Αυτοί οι πυρήνες εκφράζουν τις μη γραμμικές αλληλεπιδράσεις μεταξύ των συμβόλων και είναι υπεύθυνοι για την αντιστάθμιση των μη γραμμικών παρεμβολών. Ο μη γραμμικός όρος που προκύπτει είναι:

$$h_2(0,1) x(n) x(n-1) + h_2(1,2) x(n-1) x(n-2) + \dots$$

Όπως φαίνεται στο διάγραμμα VNLE (Εικόνα 16), οι μη γραμμικοί πυρήνες λαμβάνουν τις τιμές από δύο ή περισσότερες διακριτές καθυστερήσεις και εφαρμόζουν αντίστοιχους συντελεστές βάρους ($w_{0,0}, w_{1,1}, w_{2,2}, \dots$), για να παραχθούν τα αποτελέσματα.

Η τελική έξοδος $y(n)$ προκύπτει από το άθροισμα όλων των γραμμικών και μη γραμμικών όρων. Όλοι οι γραμμικοί και μη γραμμικοί πυρήνες προσθέτουν τα αποτελέσματά τους και σχηματίζουν την συνολική έξοδο:

$$y(n) = \sum_{m_1=0}^{M-1} w_{m_1}(n)x(n-m_1) + \sum_{m_1=0}^{L-1} \sum_{m_2=m_1}^{\min(m_1+h, L-1)} w_{m_1, m_2}(n)x(n-m_1)x(n-m_2)$$

όπου $y(n)$, η έξοδος του συστήματος στο χρονικό δείγμα n , $x(n)$ η είσοδος στο χρονικό δείγμα n , $w_{m_1}(n)$ οι γραμμικοί πυρήνες που αντιστοιχούν στον όρο 1^{ης} τάξης της εξίσωσης, $w_{m_1, m_2}(n)$ οι μη γραμμικοί πυρήνες του συστήματος, οι οποίοι περιλαμβάνουν συνδυασμό των εισόδων με τις καθυστερήσεις m_1 και m_2 (για όρο 2^{ης} τάξης).

Για να ισχύει η παραπάνω εξίσωση, ο VNLE πρέπει να προσαρμόσει τα βάρη των αντίστοιχων γραμμικών και μη γραμμικών διακριτών καθυστερήσεων. Αυτά τα βάρη προσαρμόζονται με τη χρήση προσαρμοστικών αλγορίθμων, όπως του αλγόριθμου **ελάχιστων μέσων τετραγώνων (Least-Mean Square - LMS)** και του αλγόριθμου **αναδρομικών ελάχιστων τετραγώνων (Recursive-Least Square - RLS)**. Οι τιμές του παραμορφωμένου σήματος περνούν από τις γραμμικές και μη γραμμικές περιοχές του VNLE για να παραχθεί η εκτίμηση του μεταδιδόμενου συμβόλου. Στη συνέχεια, αυτή η εκτίμηση συγκρίνεται με το σύμβολο της εκπαίδευσης, και το στιγμιαίο σφάλμα που προκύπτει από τη σύγκριση αυτή, χρησιμοποιείται για την ενημέρωση των βαρών. [13][14][15]

3.1.5. Προσαρμοστικοί Αλγόριθμοι Ισοσταθμιστών:

- **Least Mean Squares (LMS) Equalizer:** Ενημερώνει δυναμικά τις παραμέτρους του ισοσταθμιστή και ελαχιστοποιεί το σφάλμα. Ενώ είναι υπολογιστικά απλός, η απόδοσή του περιορίζεται από τη ταχύτητα προσαρμογής του στο πρόβλημα.
- **Recursive Least Squares (RLS) Equalizer:** Είναι ταχύτερος από τον LMS και αποδοτικότερος, αλλά με υψηλότερη υπολογιστική πολυπλοκότητα. [9]

3.1.6. Βέλτιστοι Ισοστάθμιστές (Optimal Equalizer):

- **Εκτίμηση ακολουθίας μέγιστης πιθανοφάνειας (Maximum Likelihood Sequence Estimation - MLSE):** Αυτή η μέθοδος εκτιμά την πιο πιθανή αλληλουχία συμβόλων, αλλά η υπολογιστική της πολυπλοκότητα είναι υψηλή.
- **Αλγόριθμος Viterbi:** Αποτελεί μια διαφορετική υλοποίηση του **MLSE**, η οποία μειώνει την πολυπλοκότητα της ισοστάθμισης με χρήση δυναμικού προγραμματισμού. Χρησιμοποιείται σε συστήματα κινητής τηλεφωνίας και στις δορυφορικές επικοινωνίες. [9][11]

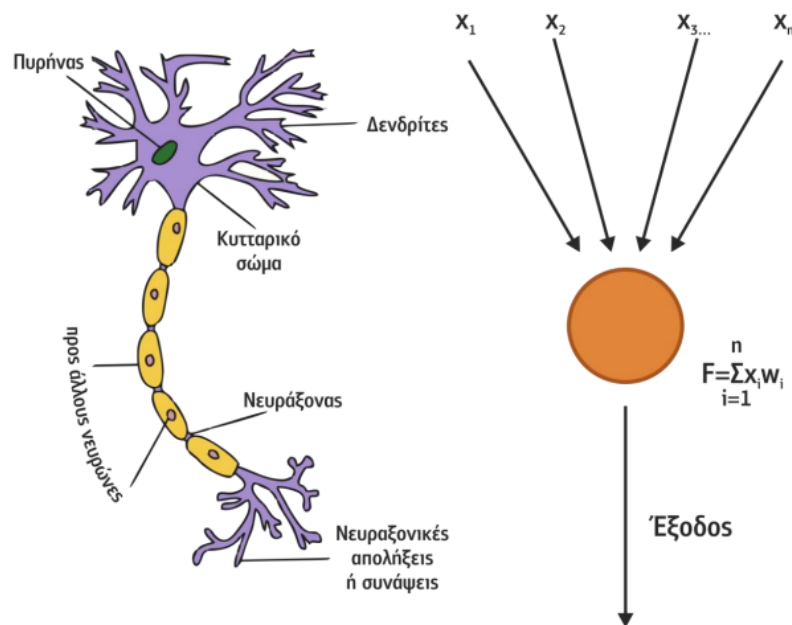
3.1.7. Άλλες σύγχρονες τεχνικές ισοστάθμισης:

- **Ισοστάθμιση Υπερτροφοδότησης (Turbo Equalization):** Συνδυάζει ισοστάθμιση και αποκωδικοποίηση μέσω επαναληπτικών αλγορίθμων, βελτιώνοντας την απόδοση σε κανάλια με έντονη ISI.
- **Τυφλή Ισοστάθμιση (Blind Equalization):** Πραγματοποιεί μια εκτίμηση για το κανάλι χωρίς τη χρήση δεδομένων εκπαίδευσης και είναι ιδανική για συστήματα όπου δεν υπάρχει πρόσβαση σε γνωσιακά δεδομένα. [9][10][12]

3.2 Ισοστάθμιση με Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks - ANN):

Η Βαθιά Μάθηση (Deep Learning) είναι μια υποκατηγορία της Μηχανικής Μάθησης (Machine Learning) στον τομέα της Τεχνητής Νοημοσύνης (Artificial Intelligence), η οποία ασχολείται με τα Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks - ANN) πολλαπλών επιπέδων (multi-layer ANN), τα οποία εκπαιδεύονται από μεγάλο όγκο πολυδιάστατων δεδομένων και είναι αποδοτικότερα από τους αλγορίθμους μηχανικής μάθησης, σε πολυσύνθετα προβλήματα.

Το θεμελιώδες δομικό στοιχείο ενός ANN είναι το Perceptron, το οποίο προσπαθεί να μιμηθεί τη λειτουργία ενός βιολογικού νευρώνα του ανθρώπινου εγκεφάλου, όπως φαίνεται στη παρακάτω εικόνα:



Εικόνα 17: Ο βιολογικός νευρώνας συγκριτικά με το βασικό Perceptron.

Το στοιχειώδες Perceptron είναι ουσιαστικά ένα ANN με μόνο ένα νευρώνα. Η έξοδός του για διάνυσμα εισόδου $x = \{x_1, \dots, x_n\}$ δίνεται από τη σχέση:

$$y = g\left(\sum_{i=1}^n x_i w_i\right)$$

όπου g η συνάρτηση ενεργοποίησης (activation function) του νευρώνα και w τα βάρη των συνάψεων. Μερικές φορές εκτός από τα εισερχόμενα σήματα x και τα αντίστοιχα βάρη w , ο νευρώνας έχει και ένα εσωτερικό βάρος b που τον χαρακτηρίζει. Το εσωτερικό αυτό βάρος λέγεται **προδιάθεση (bias)** του νευρώνα. Θεωρείται πάντα ότι η τιμή του σήματος που περνάει από όλα τα εσωτερικά βάρη είναι 1. Έτσι, οι μονάδες της προδιάθεσης θα είναι οι ίδιες με τις μονάδες του γινομένου xw . Στη γενική μορφή η έξοδός του προκύπτει από τη σχέση:

$$y = g\left(b + \sum_{i=1}^n x_i w_i\right)$$

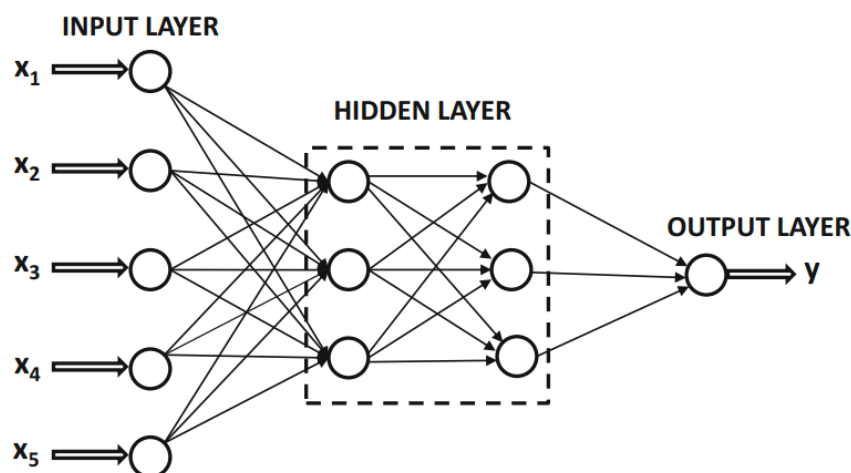
Η προδιάθεση δεν έχει συγκεκριμένη φυσική έννοια και δεν είναι απαραίτητο να τροφοδοτείται στον επόμενο νευρώνα. Μερικές φορές θεωρείται ότι είναι ένα εξωτερικό ερέθισμα, το οποίο προστίθεται στο υπόλοιπο άθροισμα για να δοθεί το σωστό y στην έξοδο.

Η συνάρτηση ενεργοποίησης περιορίζεται από ένα κριτήριο κατωφλίου μιας τιμής θ . Εάν η τιμή του y ικανοποιεί το κριτήριο κατωφλίου τότε ο νευρώνας ενεργοποιείται και τροφοδοτεί την τιμή y ως τιμή εισόδου στον επόμενο νευρώνα. Η ενεργοποίηση του νευρώνα εξαρτάται από τρεις παραμέτρους: τα βάρη των συνδέσεων, τις τιμές των εισόδων και την τιμή του κατωφλίου. Αυτό που μαθαίνει το σύστημά να κάνει, αποθηκεύεται στα βάρη των συνδέσεων που μεταβάλλονται συνεχώς κατά την διάρκεια της εκπαίδευσης.

Κατά την εκπαίδευση ενός τεχνητού νευρωνικού δικτύου, οι αλλαγές στα βάρη γίνονται με τους εξής τρόπους:

- Με *εποπτευόμενη εκπαίδευση (supervised training)* δίνονται οι τιμές των εισόδων και των στόχων που πρέπει να μάθει το δίκτυο, δηλαδή παρουσιάζονται τα πρότυπα στο δίκτυο. Γίνεται αρχικοποίηση με τυχαίες τιμές στα βάρη w . Κατά την διαδικασία εκπαίδευσης το δίκτυο αλλάζει τις τιμές των βαρών και τις διορθώνει ανάλογα με το λαμβανόμενο σφάλμα (διαφορά από τον στόχο). Ο στόχος είναι τελικά να ελαχιστοποιηθεί η διαφορά μεταξύ της επιθυμητής εξόδου και της τρέχουσας τιμής της εξόδου μετά από διαδοχικές αλλαγές των βαρών (ανακυκλώσεις διορθώσεων).
- Με *μη-εποπτευόμενη εκπαίδευση (unsupervised training)*. Σε αυτή το μέθοδο δεν δίνονται αντίστοιχοι στόχοι όπως στην περίπτωση της εποπτευόμενης εκπαίδευσης, έτσι δεν πραγματοποιείται κανένας έλεγχος ή σύγκριση με το σφάλμα. Το τεχνητό νευρωνικό δίκτυο δεν χρησιμοποιεί καμία εξωτερική παράμετρο για την αλλαγή των βαρών. Μπορεί να χρησιμοποιηθεί ένας εσωτερικός έλεγχος, ώστε να βρεθεί κάποιο μοτίβο στα σήματα εισόδου και να προσαρμοστεί ανάλογα έτσι ώστε οι εξοδοί να έχουν τα ίδια χαρακτηριστικά με αυτά. Αυτή η μέθοδος εκπαίδευσης λέγεται αυτό-εποπτευόμενη εκπαίδευση (self-supervised training), διότι το δίκτυο ελέγχει τον εαυτό του και διορθώνει τα σφάλματα με ένα μηχανισμό ανάδρασης. Σε κάθε περίπτωση όταν το δίκτυο δεν αλλάζει πλέον τις τιμές των βαρών, τότε θεωρείται ότι η εκπαίδευση έχει επιτευχθεί. [16][17]

3.2.1 Νευρωνικά Δίκτυα Πρόσθιας Τροφοδότησης (Feedforward Neural Networks – FNN):



Εικόνα 18: FNN με 1 στρώμα εισόδου, 2 κρυφά στρώμα και 1 στρώμα εξόδου.

Τα νευρωνικά δίκτυα πρόσθιας τροφοδότησης (FNN) ονομάζονται και Perceptrons πολλαπλών στρωμάτων (Multi-Layer Perceptrons - MLPs). Αποτελούν τα απλούστερα μοντέλα στον τομέα της βαθιάς μάθησης (Deep Learning) και των **βαθέων νευρωνικών δικτύων (Deep Neural Networks-DNN)**. Η διάταξη των νευρώνων χωρίζεται σε τρία είδη στρωμάτων, το *Input Layer* (Στρώμα Εισόδου), *Hidden Layer* (Κρυφό Στρώμα) και *Output Layer* (Στρώμα Εξόδου). Στόχος του FNN είναι η προσέγγιση μιας συνάρτησης f^* , όπως για παράδειγμα σε ένα πρόβλημα ταξινόμησης στο οποίο η είσοδος x αντιστοιχίζεται στη κατηγορία των y με τη διαδικασία f , σύμφωνα με τη σχέση:

$$y = f(x)$$

Το FNN προσεγγίζει την f^* όσο πιο κοντά γίνεται στην f , ορίζοντας μια απεικόνιση $y = f^*(x; \theta)$ και μαθαίνει τις τιμές του θ οι οποίες οδηγούν στην καλύτερη προσέγγιση της διαδικασίας f .

Η πληροφορία μεταφέρεται μόνο από την είσοδο x , προς στην έξοδο y , χωρίς ανατροφοδότηση. Οι νευρώνες μεταξύ των επιπέδων είναι πλήρως συνδεδεμένοι, δηλαδή κάθε νευρώνας σε ένα επίπεδο είναι συνδεδεμένος με όλους τους νευρώνες του επόμενου επιπέδου. Κάθε σύνδεση μεταξύ τους έχει ένα *συντελεστή βάρους (weight)*, του οποίου η τιμή προσαρμόζεται κατά τη διαδικασία της *εκπαίδευσης (teaching)*. Οι τιμές των βαρών επηρεάζουν τη συμβολή της κάθε εισόδου στην τελική έξοδο. Κάθε νευρώνας στο κρυφό στρώμα και στο στρώμα εξόδου εφαρμόζει μία συνάρτηση ενεργοποίησης, η οποία αποφασίζει για την ενεργοποίηση ενός νευρώνα ή όχι. Συνήθεις συναρτήσεις ενεργοποίησης είναι, η συνάρτηση **Ανορθωμένης Γραμμικής Μονάδας (Rectified Linear Unit - ReLU)**, η **σιγμοειδής συνάρτηση (Sigmoid function)** και η **συνάρτηση υπερβολικής εφαπτομένης (Tanh function)**. Η συνάρτηση ενεργοποίησης βοηθά στην εισαγωγή μη γραμμικών χαρακτηριστικών, ώστε το νευρωνικό δίκτυο να μπορεί να προσομοιώνει και να μαθαίνει πιο πολύπλοκες σχέσεις.

Η μάθηση γίνεται μέσω της **αντίστροφης διάδοσης (backpropagation)** σφάλματος από το τέλος του δικτύου προς την αρχή του δικτύου, όπου υπολογίζονται οι μερικές παράγωγοι και το δίκτυο υπολογίζει την απόκλιση της πρόβλεψης από την πραγματική τιμή, δηλαδή το σφάλμα (*error*). Το σφάλμα διοχετεύεται προς τα πίσω μέσω του δικτύου, προσαρμόζοντας τα βάρη ανάλογα με το μέγεθος και την κατεύθυνση του σφάλματος. Η εκπαίδευσή του είναι η επαναληπτική εφαρμογή αυτής της διαδικασίας, όπου το FNN βελτιώνει σταδιακά την ικανότητά του να προβλέπει. Η αποδοτικότερη ρύθμιση των βαρών κατά την διαδικασία της εκπαίδευσης, επιτυγχάνεται με τη εφαρμογή διάφορων αλγοριθμικών μεθόδων όπως της **επικλινούς καθόδου (Gradient Descent - GD)**.

Η διαδικασία ξεκινά με την είσοδο των δεδομένων στο νευρωνικό δίκτυο από το στρώμα εισόδου, και στη συνέχεια από το κρυφό στρώμα. Οι νευρώνες λαμβάνουν τις εισόδους από το προηγούμενο στρώμα, πολλαπλασιάζονται με τα αντίστοιχα βάρη, εφαρμόζεται η συνάρτηση ενεργοποίησης και τα αποτελέσματα οδηγούνται στο επόμενο επίπεδο. Αν το επόμενο επίπεδο είναι το στρώμα εξόδου τότε αυτά τα αποτελέσματα είναι η πρόβλεψη του FNN. Έπειτα εφαρμόζεται η διαδικασία αντίστροφης διάδοσης σφάλματος.

Πιο αναλυτικά, κατά τη διάρκεια της εκπαίδευσης, το δίκτυο υπολογίζει μια συνάρτηση κόστους $J(\theta)$, η οποία εκφράζει κατά πόσο αποκλίνει η πρόβλεψη του δικτύου $\hat{y}$, από την πραγματική τιμή y . Το κόστος προκύπτει συναρτήσεων των παραμέτρων του δικτύου (π.χ.: των βαρών w και των προδιαθέσεων b) που συμβολίζονται με θ , και ο στόχος της εκπαίδευσης είναι να

ελαχιστοποιηθεί. Για να επιτευχθεί αυτό, χρησιμοποιείται η μέθοδος επικλινούς καθόδου (GD) για τον υπολογισμό της κλίσης (*gradient*) της συνάρτησης κόστους $\nabla_{\theta} J(\theta)$, η οποία εκφράζει τη κατεύθυνση και το μέγεθος της μεταβολής για κάθε παράμετρο θ . Αυτό γίνεται κατά τη διαδικασία της αντίστροφης διάδοσης σφάλματος. Η αντίστροφη διάδοση σφάλματος δεν είναι ο αλγόριθμος μάθησης, αλλά η διαδικασία για τον υπολογισμό των μερικών παραγώγων. Για την πραγματική μάθηση, εφαρμόζεται ο αλγόριθμος της **στοχαστικής επικλινούς καθόδου (Stochastic Gradient Descent - SGD)**, ο οποίος ρυθμίζει τα βάρη του νευρωνικού δικτύου, βάσει των υπολογισμένων βαθμών κλίσης.

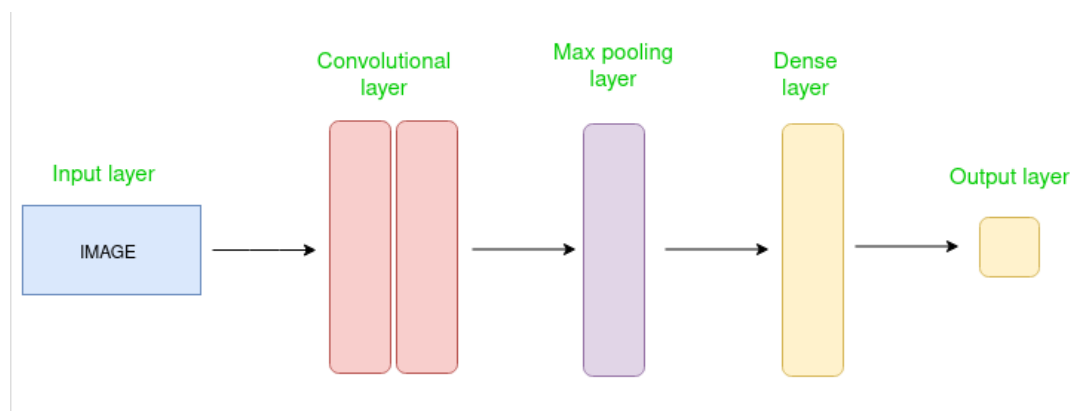
Η ενημέρωση των βαρών δίνεται από τη σχέση:

$$W_I^{s+1} = W_I^s + a_{lr} \nabla \hat{L}(W_I^s)$$

όπου W_I^s και W_I^{s+1} , τα βάρη του στρώματος I στο s -στό και στο $(s+1)$ -οστό βήμα εκπαίδευσης του αλγορίθμου αντίστοιχα, a_{lr} ο ρυθμός μάθησης (*learning rate*) που καθορίζει πόσο απότομες θα είναι οι αλλαγές στα βάρη σε κάθε βήμα εκπαίδευσης, $\hat{L}$ η τιμή της συνάρτησης του κόστους και $\nabla \hat{L}$ η κλίση (*gradient*) της συνάρτησης του κόστους ως προς τα βάρη W_I , η οποία δείχνει την κατεύθυνση της μέγιστης αύξησης ή μείωσης του κόστους. Ο ρυθμός εκμάθησης a_{lr} είναι πολύ σημαντικός για τη διαδικασία, καθώς αν είναι πολύ μεγάλος, μπορεί να προκαλέσει αστάθεια (δηλαδή το δίκτυο να μη συγκλίνει σε λύση), ενώ αν είναι πολύ μικρός, η εκπαίδευση θα πραγματοποιείται με αργό ρυθμό.

Παρά το γεγονός ότι η διαδικασία αντίστροφης διάδοσης σφάλματος αναπτύχθηκε για νευρωνικά δίκτυα πολλαπλών στρωμάτων, μπορεί να χρησιμοποιηθεί και για τον υπολογισμό των παραγώγων οποιασδήποτε συνάρτησης και δεν περιορίζεται μόνο στην εκπαίδευση των παραμέτρων των νευρωνικών δικτύων. Μπορεί να υπολογιστεί ο βαθμός κλίσης μιας συνάρτησης με ένα μόνο αποτέλεσμα, αλλά και ο **Ιακωβιανός πίνακας (Jacobian Matrix)** για συναρτήσεις με πολλαπλές εξόδους. [18][19]

3.2.2 Συνελκτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks – CNN):



Εικόνα 19: Αρχιτεκτονική ενός CNN

Τα συνελικτικά νευρωνικά δίκτυα (CNN) είναι ένας εξειδικευμένος τύπος νευρωνικών δικτύων που σχεδιάστηκαν για να επεξεργάζονται δεδομένα με γνωστή διασυνδεδεμένη τοπολογία. Χρησιμοποιούνται κυρίως για δεδομένα που έχουν δομή πλέγματος, όπως χρονοσειρές (μονοδιάστατο πλέγμα) ή εικόνες (δισδιάστατο πλέγμα). Χρησιμοποιούν τη μαθηματική πράξη της *συνέλιξης* (*convolution*) αντί για τη πράξη του πολλαπλασιασμού που χρησιμοποιείται FNN. Η πράξη της συνέλιξης επιτρέπει στα CNN να αναγνωρίζουν πρότυπα, σχήματα και χαρακτηριστικά σε δεδομένα, π.χ., άκρες σε εικόνες, ή μεταβολές σε χρονοσειρές. Τα CNN έχουν μια πιο εξειδικευμένη αρχιτεκτονική σε σχέση με τα FNN, καθώς χρησιμοποιούν τρεις βασικούς τύπους στρωμάτων, τα *συνελικτικά στρώματα* (*convolutional layers*), τα *συγκεντρωτικά στρώματα* (*pooling layers*) και τα *πλήρως συνδεδεμένα στρώματα* (*fully connected layers*).

Το κύριο χαρακτηριστικό των CNN είναι η χρήση *πυρήνων* (*kernels*) στα συνελικτικά στρώματα. Οι πυρήνες *συνελίσσονται* (*convolve*) με την είσοδο, δηλαδή σαρώνουν την είσοδο για να εξάγουν σημαντικά *χαρακτηριστικά* (*features*). Οι πυρήνες είναι πίνακες βαρών που εκπαιδεύονται μέσω του αλγορίθμου *επικλινούς καθόδου* (*GD*), ώστε να μάθουν να ανιχνεύουν συγκεκριμένα χαρακτηριστικά. Οι πυρήνες έχουν μέγεθος $L_{ker} \times N_{ch}$, όπου L_{ker} είναι το μήκος του πυρήνα και N_{ch} , ο αριθμός των εισόδων των δεδομένων. Η πράξη συνέλιξης εφαρμόζεται σε κάθε είσοδο, με τα βάρη του πυρήνα να υπολογίζονται και να εκπαιδεύονται έτσι ώστε να ανιχνεύουν τα χαρακτηριστικά που είναι σημαντικά για την πρόβλεψη. Το κάθε στοιχείο του πυρήνα (πίνακας βαρών) δίνεται από τη σχέση:

$$x_L(m) = \sigma_L(m)[(z_L(m) + b_L(m))]$$

όπου το $z^l(m)$ είναι το αποτέλεσμα της συνέλιξης, το $b^l(m)$ είναι η προσθήκη προδιάθεσης, και η σ^l είναι η συνάρτηση ενεργοποίησης.

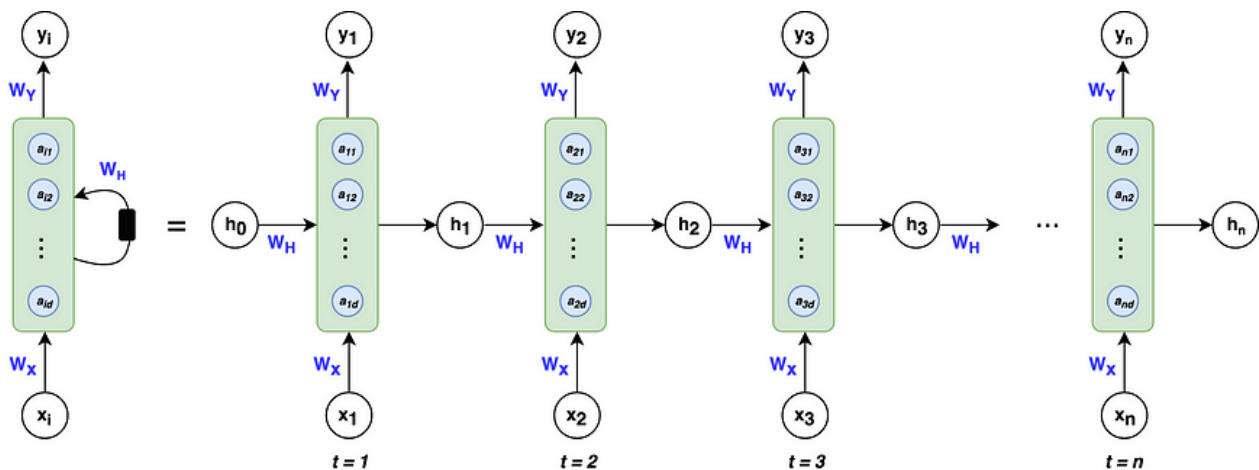
Τα CNN εκμεταλλεύονται τη χωρική κατανομή των δεδομένων και μετά από τη διαδικασία της συνέλιξης, εφαρμόζεται μια συνάρτηση ενεργοποίησης, συνήθως η *ReLU* γιατί κρατά τις θετικές τιμές και μηδενίζει τις αρνητικές, δίνοντας τη δυνατότητα στο δίκτυο να μάθει σύνθετες σχέσεις.

Στα συγκεντρωτικά στρώματα (*pooling layers*) οι διαστάσεις της εισόδου μειώνονται με την *υποδειγματοληψία* (*downsampling*) των δεδομένων, έτσι ώστε να μειωθεί η υπολογιστική πολυπλοκότητα και να διατηρηθούν μόνο τα πιο σημαντικά χαρακτηριστικά. Δύο συνήθη είδη *συγκέντρωσης* (*pooling*) είναι η *μέγιστη συγκέντρωση* (*max pooling*) και η *μέση συγκέντρωση* (*average pooling*), από τις οποίες κάθε περιοχή της εικόνας, κρατείται η μέγιστη τιμή και η μέση τιμή αντίστοιχα, απορρίπτοντας τις άλλες πληροφορίες.

Τελικά τα χαρακτηριστικά που έχουν ανιχνευτεί από τα προηγούμενα στρώματα (*feature maps*), περνούν μέσα από τα πλήρως συνδεδεμένα στρώματα, τα οποία λειτουργούν όπως ένα FNN. Αυτά τα επίπεδα πραγματοποιούν την τελική ταξινόμηση ή πρόβλεψη με βάση τα εξαγόμενα χαρακτηριστικά. Τα CNN κατασκευάζουν μια ιεραρχία χαρακτηριστικών, από απλά σε πιο σύνθετα χαρακτηριστικά, επιτρέποντας την ανίχνευση πιο αφηρημένων ιδιοτήτων. [19]

3.2.3 Επαναλαμβανόμενα Νευρωνικά Δίκτυα (Recurrent Neural Networks – RNN):

Τα επαναλαμβανόμενα νευρωνικά δίκτυα (RNN) είναι ένας τύπος τεχνητού νευρωνικού δικτύου που έχει σχεδιαστεί για να επεξεργάζεται σειριακά ή χρονικά εξαρτημένα δεδομένα. Η βασική τους διαφορά σε σχέση με τα FNN και τα CNN είναι ότι μπορούν να ανακαλούν πληροφορίες από προηγούμενα χρονικά βήματα και χρησιμοποιούν αυτή τη μνήμη για να παράγουν την έξοδο. Αυτό καθιστά τα RNN ιδανικά για εφαρμογές όπως, αναγνώριση φωνής, επεξεργασία φυσικής γλώσσας ή πρόβλεψη χρονοσειρών. Στο παρακάτω σχήμα φαίνεται η λειτουργία ενός απλού RNN ή Vanilla RNN. Όσο τα χρονικά βήματα αυξάνονται, είναι αναγκαίο να εμφανιστεί σχηματικά σε βάθος χρόνου, όπως γίνεται παρακάτω:



Εικόνα 20: Απλή αναπαράσταση RNN (αριστερά) και "ξεδιπλωμένη" στο χρόνο.

Πρόκειται για αναδρομικές μονάδες (units) των οποίων οι καταστάσεις επηρεάζονται από τα τρέχοντα, παρελθοντικά ή μελλοντικά (σε περιπτώσεις αμφίδρομων RNN) δεδομένα. Η ικανότητα του δικτύου να διατηρεί και να χρησιμοποιεί παρελθοντικές καταστάσεις, αποτελεί την έννοια της μνήμης του RNN και υλοποιείται με τη βοήθεια της **κρυφής κατάστασης (hidden state)**.

Η κρυφή κατάσταση είναι ένα διάνυσμα στο οποίο καταγράφεται η εσωτερική κατάσταση του δικτύου σε ένα συγκεκριμένο χρονικό βήμα. Περιέχει τις πληροφορίες που έχουν δει μέχρι εκείνη τη στιγμή οι μονάδες του δικτύου και επιτρέπει στο RNN να λαμβάνει υπόψη τόσο την τρέχουσα είσοδο όσο και τις προηγούμενες. Είναι ένας εσωτερικός μηχανισμός μνήμης που ενημερώνεται καθώς το δίκτυο επεξεργάζεται νέα δεδομένα. Το RNN επεξεργάζεται σειριακά τα εισερχόμενα στοιχεία της κρυφής κατάστασης και η πληροφορία συσσωρεύεται για όλες τις προηγούμενες καταστάσεις της ακολουθίας. Η πληροφορία διατηρείται στην κρυφή κατάσταση του RNN, η οποία συμβάλει στην επεξεργασία κάθε νέου χρονικού βήματος.

Η εξίσωση της κρυφής κατάστασης δίνεται από τη σχέση:

$$h_t = \sigma (W_{xh}x_t + W_{hh}h_{t-1})$$

όπου h_t η κρυφή κατάσταση σε κάθε χρονική στιγμή t , που εξαρτάται από την προηγούμενη κρυφή κατάσταση h_{t-1} και την τρέχουσα είσοδο x_t , W_{xh} είναι ο πίνακας βαρών μεταξύ της εισόδου και

της κρυφής κατάστασης, W_{hh} είναι ο πίνακας βαρών που συνδέει τις κρυφές καταστάσεις μεταξύ τους, η συνάρτηση ενεργοποίησης σ εφαρμόζεται στοιχείο προς στοιχείο στο αποτέλεσμα.

Η εξίσωση των εξόδων είναι:

$$y_t = W_{hy}h_t$$

όπου η έξοδος y_t σε κάθε χρονική στιγμή παράγεται από την κρυφή κατάσταση h_t με τη χρήση του πίνακα βαρών W_{hy} . Το y_t είναι ένα διάνυσμα πιθανοτήτων για τις πιθανές επόμενες τιμές, και συνήθως εφαρμόζεται η συνάρτηση *softmax* για να μετατραπούν οι τιμές αυτές σε έγκυρες πιθανότητες.

Η αναπαράσταση του Vanilla RNN σε βάθος χρόνου, είναι χρήσιμη κυρίως κατά την εκπαίδευση του δικτύου. Σε αυτή τη διαδικασία, το σφάλμα που παράγεται στο τέλος της ακολουθίας διαδίδεται προς τα πίσω μέσα από τα χρονικά βήματα, χρησιμοποιώντας μια διαδικασία που ονομάζεται **αντίστροφη χρονική διάδοση σφάλματος (Backpropagation Through Time - BPTT)**. Ουσιαστικά, κάθε χρονικό βήμα έχει στην είσοδο του, ένα αντίγραφο της τοπολογίας και των βαρών του δικτύου. Στη συνέχεια το σφάλμα υπολογίζεται και συσσωρεύεται σε κάθε χρονικό στάδιο και ανανεώνονται τα βάρη βάσει του συνολικού συσσωρευμένου σφάλματος.

Λόγω της επαναληπτικής φύσης της εξίσωσης της κρυφής κατάστασης, τα RNN μπορούν να επεξεργάζονται ακολουθίες μεταβλητού μήκους. Αυτό είναι ιδιαίτερα χρήσιμο για εφαρμογές, όπως η μοντελοποίηση φυσικής γλώσσας, όπου οι προτάσεις μπορεί να έχουν διαφορετικά μήκη. Το δίκτυο προσαρμόζεται ανάλογα με το μήκος της ακολουθίας, χωρίς να χρειάζεται να αλλάξει η αρχιτεκτονική του. Τα RNN είναι πλήρεις μηχανές Turing, που σημαίνει ότι μπορούν να προσομοιώσουν οποιονδήποτε αλγόριθμο, αρκεί να υπάρχει αρκετή υπολογιστική δύναμη και δεδομένα.

Όταν ο αριθμός των χρονικών βημάτων αυξάνεται, η διαδικασία αντίστροφης διάδοσης σφάλματος (backpropagation) προκαλεί προβλήματα **μηδενισμού και απειρισμού των μερικών παραγώγων (vanishing and exploding gradient)**, έτσι η εκπαίδευση γίνεται πιο δύσκολη για μεγάλες ακολουθίες. Πιο αναλυτικά, οι μερικές παράγωγοι γίνονται πολύ μικρές καθώς μεταδίδονται προς τα πίσω, και έτσι το δίκτυο δυσκολεύεται να μάθει μακροχρόνιες εξαρτήσεις. Αυτό σημαίνει ότι τα βάρη των πρώτων στρωμάτων (πιο κοντά στην είσοδο) δεν ενημερώνονται επαρκώς, με αποτέλεσμα το δίκτυο να μην μπορεί να μάθει αποδοτικά από τα πρώιμα δεδομένα. Το πρόβλημα αυτό κάνει τη διαδικασία αντίστροφης διάδοσης σφάλματος προβληματική, επειδή υποβαθμίζει εκθετικά τις εξαρτήσεις που εκτείνονται σε μεγάλο χρόνο σε σχέση με τις πιο πρόσφατες. Το πρόβλημα επιδεινώνεται ακόμα περαιτέρω από την κοινή χρήση των ίδιων βαρών σε κάθε χρονικό βήμα του RNN, το οποίο σημαίνει ότι τα σφάλματα επαναλαμβάνονται σε όλα τα χρονικά βήματα.

Το πρόβλημα του μηδενισμού των μερικών παραγώγων προκύπτει επειδή, τα νευρωνικά δίκτυα χρησιμοποιούν συναρτήσεις ενεργοποίησης όπως είναι η *sigmoid* ή η *tanh*, για να καθορίσουν αν ένας νευρώνας θα ενεργοποιηθεί ή όχι. Αυτές οι συναρτήσεις έχουν περιορισμένο εύρος τιμών (π.χ., η σιγμοειδής παίρνει τιμές μεταξύ 0 και 1). Έτσι όταν οι μερικοί παράγωγοι αυτών των συναρτήσεων υπολογίζονται κατά τη διαδικασία αντίστροφης διάδοσης σφάλματος (backpropagation), οι τιμές τους γίνονται πολύ μικρές. Έτσι, όταν μεταφέρονται πίσω μέσω πολλών στρωμάτων (ή χρονικών βημάτων στη περίπτωση του RNN), οι τιμές τους γίνονται όλο και μικρότερες, οδηγώντας σε σχεδόν μηδενική ενημέρωση των βαρών για τα πρώιμα χρονικά βήματα.

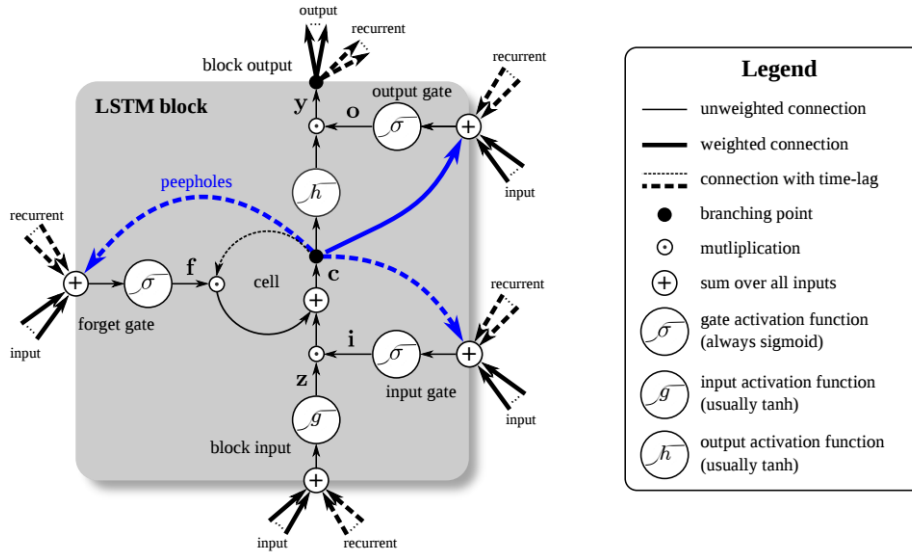
Ως αποτέλεσμα το RNN αδυνατεί να μάθει τις μακροπρόθεσμες εξαρτήσεις σε μεγάλες ακολουθίες από δεδομένα (π.χ., χρονοσειρές).

Το πρόβλημα του απειρισμού των μερικών παραγώγων είναι ακριβώς το αντίθετο, όπου οι τιμές των μερικών παραγώγων, γίνονται υπερβολικά μεγάλες κατά τη διάρκεια της διαδικασίας αντίστροφης διάδοσης σφάλματος. Αυτό οδηγεί σε πολύ απότομες ενημερώσεις των βαρών, κάτι το οποίο μπορεί να κάνει το δίκτυο να παρουσιάσει ασταθή συμπεριφορά και να μη συγκλίνει καθόλου σε αποφάσεις. Δηλαδή όταν οι μερικές παράγωγοι των συναρτήσεων ενεργοποίησης ή των βαρών έχουν μεγάλες τιμές και αυτές διογκώνονται καθώς μεταφέρονται προς τα πίσω. Έτσι οι ενημερώσεις στα βάρη μπορεί να είναι τόσο απότομες που το δίκτυο αναπηδάει στον χώρο των παραμέτρων και δεν μπορεί να συγκλίνει σε ένα σωστό αποτέλεσμα.

Για την αντιμετώπιση αυτών των προβλημάτων έχουν αναπτυχθεί διάφορες τεχνικές όπως:

- **Περιοπή μερικών παραγώγων (Gradient clipping):** Περιορίζει τις τιμές των μερικών παραγώγων (gradients), ώστε να μην υπερβαίνουν ένα συγκεκριμένο όριο, για την αποφυγή του προβλήματος απειρισμού των μερικών παραγώγων (*exploding gradients*).
- Ορισμός της ReLU (Rectified Linear Unit) ως συνάρτησης ενεργοποίησης: Είναι πιο ανθεκτική στο πρόβλημα του μηδενισμού των μερικών παραγώγων (*vanishing gradients*), καθώς οι παράγωγοί της είναι σταθεροί για θετικές τιμές.
- Μπορεί να χρησιμοποιηθεί μια παραλλαγή της διαδικασίας *backpropagation* που ονομάζεται **αποκομμένη (Truncated) BPTT**, όπου η διαδικασία εφαρμόζεται μόνο σε έναν περιορισμένο αριθμό χρονικών βημάτων (π.χ., αντί να υπολογισθούν όλες οι μερικές παράγωγοι, γίνεται η εκπαίδευση μόνο για τα τελευταία βήματα).
- **Δίκτυα μακροπρόθεσμης-βραχυπρόθεσμης μνήμης (LSTM) και δίκτυα αναδρομικής μονάδας με πύλες (GRU)** είναι ειδικές αρχιτεκτονικές RNN που έχουν σχεδιαστεί για να αποθηκεύουν και να μεταφέρουν σημαντικές πληροφορίες μακροχρόνια, κάτι που βοηθά στην αποτροπή του *vanishing gradient*. Χρησιμοποιούν μηχανισμούς ελέγχου της ροής της πληροφορίας μέσω πυλών, επιτρέποντας την αποθήκευση και τη διαγραφή μνήμης. Είναι ιδανικά για προβλήματα με μακροχρόνιες εξαρτήσεις, αλλά είναι πιο πολύπλοκα και απαιτούν περισσότερους υπολογιστικούς πόρους. [19][20]

3.2.3.1 Δίκτυα μακροπρόθεσμης-βραχυπρόθεσμης μνήμης (Long-Short Term Memory Networks - LSTM):



Εικόνα 21: Αναπαράσταση της λειτουργίας ενός LSTM.

Τα απλά RNN παρουσιάζουν το πρόβλημα του μηδενισμού των μερικών παραγώγων (*vanishing gradients*) και το πρόβλημα του απειρισμού των μερικών παραγώγων (*exploding gradients*) κατά τη διαδικασία της αντίστροφης διάδοσης σφάλματος (*backpropagation*). Άρα δυσκολεύονται να διατηρήσουν τη μνήμη σε μακροχρόνιες ακολουθίες, καθώς οι διαδοχικές ενημερώσεις της κατάστασης οδηγούν σε απώλεια σημαντικής πληροφορίας.

Τα δίκτυα **μακροπρόθεσμης-βραχυπρόθεσμης μνήμης (LSTM)** προτάθηκαν ως λύση στο πρόβλημα. Αποτελούν μια πιο σύνθετη μορφή RNN η οποία διαθέτει ειδικές μονάδες που ονομάζονται **κύτταρα LSTM (LSTM cells)**. Τα κύτταρα διαθέτουν τρεις **πύλες (gates)**, την **πύλη λήθης (Forget gate)** η οποία αποφασίζει ποια πληροφορία θα διαγραφεί από τη μνήμη, την **πύλη εισόδου (Input gate)** η οποία καθορίζει ποια νέα πληροφορία θα αποθηκευτεί στη μνήμη, και την **πύλη εξόδου (Output gate)** που καθορίζει ποια πληροφορία θα χρησιμοποιηθεί για τον υπολογισμό της εξόδου. Αυτός ο μηχανισμός επιτρέπει στα LSTM να διατηρούν πληροφορίες για μεγάλα χρονικά διαστήματα και να αποφασίζουν ποιες είναι χρήσιμες για κάθε χρονικό βήμα.

Η κύρια κατάσταση μνήμης ονομάζεται **κατάσταση κυττάρου (Cell State - C_t)** μέσα στην οποία αποθηκεύεται η μακροπρόθεσμη μνήμη του LSTM, η οποία διατηρεί τη πληροφορία για πολλά χρονικά βήματα, με ελεγχόμενες ενημερώσεις της.

Οι ενημερώσεις στο LSTM βασίζονται σε διαδοχικές πράξεις με τις παρακάτω εξισώσεις:

$$[i, f, o, c] = f(W^{(k)}[h_{t-1}, x_t]^T)$$

όπου i, f, o είναι οι πύλες *input*, *forget*, *output* και c είναι η κατάσταση του κυττάρου αντίστοιχα.

Η Forget gate ελέγχει αν θα διατηρηθεί η προηγούμενη κατάσταση μνήμης c_{t-1} , η κατάσταση μνήμης δίνεται από τη σχέση:

$$c_t = f \odot c_{t-1} + i \odot c$$

όπου $f \odot c_{t-1}$ σημαίνει ότι θυμάται ένα μέρος της προηγούμενης κατάστασης, ενώ το $i \odot c$ είναι η προσθήκη νέας πληροφορίας.

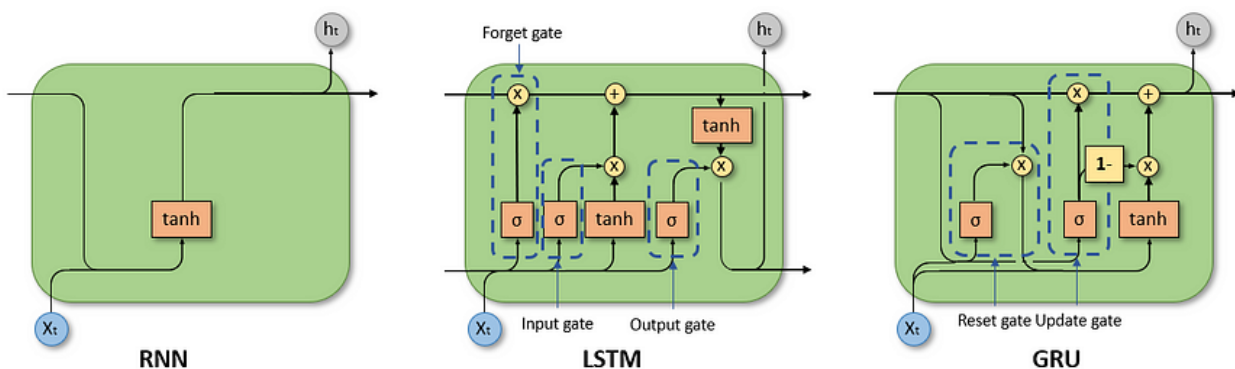
Η Output Gate αποφασίζει πόση πληροφορία από την κατάσταση μνήμης c_t θα εμφανιστεί στη κρυφή κατάσταση με τον εξής τρόπο:

$$h_t = o \odot f(c_t)$$

Το LSTM επιλύει το πρόβλημα του *vanishing gradient* διατηρώντας μια κατάσταση κυττάρου (cell state) που ενημερώνεται προσθετικά αντί για πολλαπλασιαστικά, κάτι που βοηθά στην καλύτερη ροή των τιμών των μερικών παραγώγων στο δίκτυο. Επιπλέον, οι πύλες ελέγχου (input, forget, output) επιτρέπουν στο LSTM να αποφασίζει ποιες πληροφορίες να κρατήσει ή να ξεχάσει, δίνοντάς του τη δυνατότητα να διαχειρίζεται μακροχρόνιες εξαρτήσεις.

Τα LSTM είναι ιδανικά για γλωσσική μοντελοποίηση, όπου είναι κρίσιμο να διατηρούνται πληροφορίες για μεγάλα χρονικά διαστήματα και να ανιχνεύονται εξαρτήσεις από προηγούμενες λέξεις, κάτι που τα απλά RNN δεν μπορούν να κάνουν αποτελεσματικά. [19][20]

3.2.3.2 Δίκτυα αναδρομικής μονάδας με πύλες (Gated Recurrent Unit Networks - GRU):



Εικόνα 22: Δομή του GRU συγκριτικά με LSTM

Το **δίκτυο αναδρομικής μονάδας με πύλες (GRU)** αποτελεί μια απλοποιημένη εκδοχή του LSTM δικτύου, η οποία είναι και αυτή σχεδιασμένη για να αντιμετωπίσει προβλήματα που σχετίζονται με την εκπαίδευση των RNN.

Σε αντίθεση με τα LSTM, τα GRU δεν χρησιμοποιούν ξεχωριστές εσωτερικές καταστάσεις (*cell states*). Αντ' αυτού, η ενημέρωση της κατάστασης γίνεται άμεσα στο κρυφό επίπεδο (*hidden state*). Αυτό μειώνει την πολυπλοκότητα και τον αριθμό των παραμέτρων που απαιτούνται, καθώς δεν περιέχονται ξεχωριστές πύλες για εισόδους και εξόδους (input/output gates) όπως έχει το LSTM.

Οι δύο βασικές πύλες που διαθέτουν είναι:

- Η Πύλη Επαναφοράς (*Reset Gate, r_t*): Καθορίζει πόσο από το προηγούμενο κρυφό επίπεδο (*hidden state*) θα διατηρηθεί για την ενημέρωση της νέας κατάστασης. Όταν η

πύλη έχει τιμή κοντά στο 0, διαγράφει το παρελθόν και ανανεώνει την κατάσταση με νέα πληροφορία.

- Και η *Πύλη Ενημέρωσης (Update Gate, z_t)*: Καθορίζει πόσο από το προηγούμενο κρυφό επίπεδο (*hidden state*) θα αντιγραφεί στην τρέχουσα κατάσταση. Με αυτήν την πύλη, το μοντέλο αποφασίζει αν θα διατηρήσει την παλιά πληροφορία ή αν θα την ενημερώσει με νέα.

Οι πύλες αυτές υπολογίζονται με σιγμοειδείς συναρτήσεις ενεργοποίησης, οι οποίες επιστρέφουν τιμές μεταξύ 0 και 1, αποφασίζοντας ποσοστιαία πόσο θα επιδράσει κάθε κομμάτι πληροφορίας.

Ο υπολογισμός της *πύλης ενημέρωσης (update gate)* δίνεται από τη σχέση:

$$z_t = \sigma(W_z x_t + U_z h_{t-1})$$

όπου η πύλη ενημέρωσης αποφασίζει πόσο από το παρελθόν (h_{t-1}) θα μεταφερθεί στο παρόν, ανάλογα με τις τρέχουσες εισόδους x_t .

Ο υπολογισμός της *πύλης επαναφοράς (reset gate)* δίνεται από τη σχέση:

$$r_t = \sigma(W_r x_t + U_r h_{t-1})$$

όπου η πύλη επαναφοράς καθορίζει πόσο από την προηγούμενη κρυφή κατάσταση θα διαγραφεί και δεν θα επηρεάσει τη νέα ενημέρωση.

Η ενημέρωση της κρυφής κατάστασης (*hidden state*) είναι:

$$\tilde{h}_t = f(W_h x_t + r_t \cdot U_h h_{t-1})$$

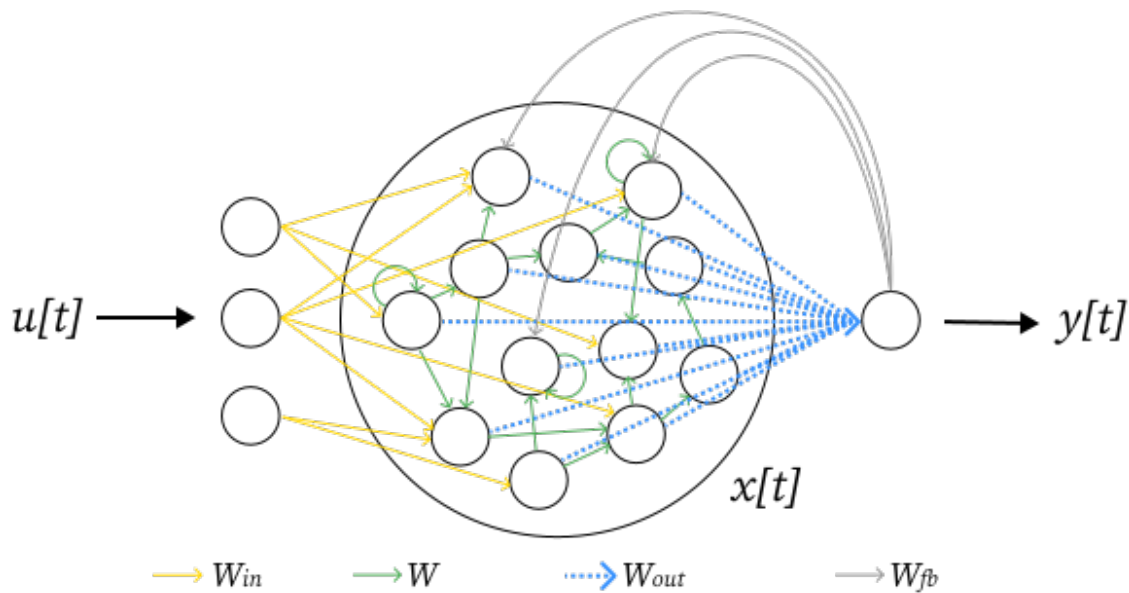
όπου η προηγούμενη κρυφή κατάσταση h_{t-1} πολλαπλασιάζεται με την πύλη επαναφοράς r_t , περιορίζοντας την επιρροή της και επιτρέποντας την ενημέρωση της νέας κατάστασης $\tilde{h}_t$.

Η *τελική κρυφή κατάσταση (final hidden state)* θα είναι:

$$h_t = z_t \cdot h_{t-1} + (1 - z_t) \cdot \tilde{h}_t$$

όπου το τελικό κρυφό επίπεδο h_t προκύπτει από έναν συνδυασμό της προηγούμενης κατάστασης h_{t-1} και της νέας υπολογισμένης κατάστασης $\tilde{h}_t$, ελεγχόμενη από την πύλη ενημέρωσης z_t . [20]

3.2.3.3 Echo State Networks - (ESN):



Εικόνα 23: Αναπαράσταση ενός ESN

Τα Echo State Networks (ESN) αποτελούν μια διαφορετική προσέγγιση των RNN. Η βασική ιδέα πίσω από τα ESN είναι η χρήση ενός μεγάλου, τυχαία αρχικοποιημένου κρυφού στρώματος που αποκαλείται δεξαμενή (*reservoir*). Για αυτό το λόγο υπάγονται στο τομέα υπολογισμών δεξαμενής (*Reservoir Computing- RC*) μαζί με τα *Liquid State Machines (LSM)*.

Η δομή του ESN περιέχει τρία στρώματα τα οποία είναι:

- *Στρώμα Εισόδου (Input layer)*: Είναι το στρώμα το οποίο προσπελαύνει την είσοδο στο στρώμα που βρίσκεται η δεξαμενή.
- *Δεξαμενή (reservoir)*: Είναι ένα τυχαία αρχικοποιημένο και συνδεδεμένο δίκτυο νευρώνων, του οποίου οι καταστάσεις ανανεώνονται αναδρομικά. Δεν γίνεται εκπαίδευση στο reservoir, αλλά οι καταστάσεις που παράγονται είναι εξαιρετικά μη γραμμικές και είναι χρήσιμες για πολλές εφαρμογές.
- *Στρώμα εξόδου (Output layer)*: Είναι το μόνο στρώμα που εκπαιδεύεται για να προβλέπει την επιθυμητή έξοδο.

Το γεγονός ότι η δεξαμενή παραμένει σταθερή και δεν εκπαιδεύεται, μειώνει τον υπολογιστικό φόρτο της διαδικασίας της εκπαίδευσης συγκριτικά με άλλα RNN, αφού εκπαιδεύονται μόνο τα βάρη του στρώματος εξόδου.

Η συμπεριφορά του ESN εξαρτάται από τρέχοντα και προηγούμενα δεδομένα εισόδου, και αντιστέκεται στη μείωση της επιρροής των πρώιμων εισόδων με την πάροδο των χρονικών βημάτων λόγω του προβλήματος μηδενισμού μερικών παραγώγων (*vanishing gradients*), αποτρέποντας έτσι τη συσσώρευση σφαλμάτων και την ασταθή συμπεριφορά. Αυτή η ιδιότητα ονομάζεται *Echo State Property*. Η ιδιότητα αυτή όπως θα αναλυθεί και στη συνέχεια, διασφαλίζεται όταν η *φασματική ακτίνα (spectral radius)* του πίνακα βαρών W_{res} είναι μικρότερη από 1, όμως προτείνεται να έχει τιμή σχετικά κοντά στο 1 για να διασφαλίζεται ότι η ιδιότητα διατηρείται από το σύστημα.

Η δυναμική κατάσταση της δεξαμενής ενός ESN ανανεώνεται σε κάθε χρονικό βήμα με την ακόλουθη εξίσωση:

$$x(t+1) = f(W_{\text{res}}x(t) + W_{\text{in}}u(t+1) + W_{\text{fb}}y(t))$$

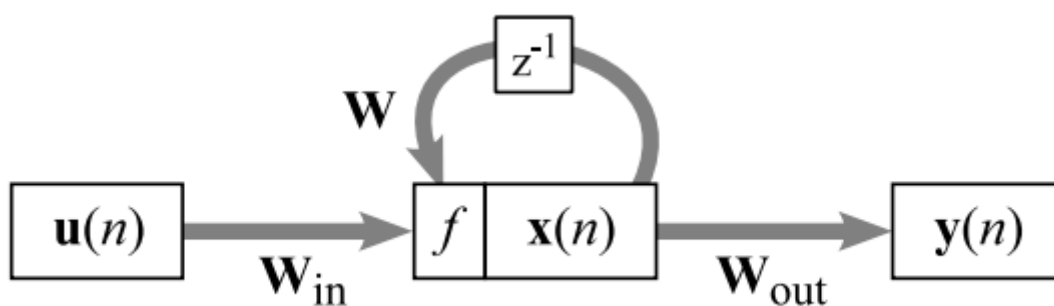
όπου $x(t+1)$ είναι το διάνυσμα κατάστασης της δεξαμενής στο χρονικό βήμα $t+1$, W_{res} είναι ο πίνακας βαρών των συνδέσεων μεταξύ των νευρώνων της δεξαμενής, $u(t+1)$ είναι το σήμα εισόδου στο χρονικό βήμα $t+1$, W_{in} είναι ο πίνακας βαρών των συνδέσεων εισόδου, W_{fb} είναι ο πίνακας βαρών της ανάδρασης από την έξοδο προς τη δεξαμενή (προαιρετικό, αν χρησιμοποιείται ανάδραση), $y(t)$ είναι η έξοδος του δικτύου στο χρόνο t (αν υπάρχει ανάδραση) και η f είναι μια μη γραμμική συνάρτηση ενεργοποίησης (π.χ. $\tanh$ ή sigmoid).

Η έξοδος του δικτύου υπολογίζεται γραμμικά από τις καταστάσεις της δεξαμενής:

$$y(t) = W_{\text{out}}x(t)$$

όπου $y(t)$ είναι το διάνυσμα εξόδου στο χρόνο t , W_{out} είναι ο πίνακας βαρών του στρώματος εξόδου και $x(t)$ είναι το διάνυσμα κατάστασης της δεξαμενής.

Για τη διασύνδεση των εισόδων προς τις κρυφές μονάδες του ESN είναι σημαντικό να γίνει σωστή παραμετροποίηση των βαρών (W_{in}), διαφορετικά οι νέες εισροές σε κάθε χρονικό στίγμα (*time stamp*) μπορούν να καταστρέψουν τις πληροφορίες που μεταφέρονται από προηγούμενα χρονικά στίγματα. Χρησιμοποιούνται τυχαία βάρη στις συνδέσεις μεταξύ των εισόδων προς τις κρυφές μονάδες (input units - to - hidden units) και στις συνδέσεις μεταξύ των κρυφών μονάδων (hidden unit - to - hidden unit). Η διάσταση του κρυφού στρώματος (reservoir) είναι συνήθως πολύ μεγαλύτερη από αυτή του στρώματος εισόδου.



Εικόνα 24: Διάγραμμα ροής σε κλασικό ESN

Σημαντικό χαρακτηριστικό το οποίο επηρεάζει την αρχιτεκτονική των ESN, είναι η εκπαίδευση μόνο του στρώματος εξόδου, συνήθως αυτό το στρώμα περιλαμβάνει γραμμικές μονάδες επεξεργασίας, ειδικά όταν οι τιμές εξόδου σχηματίζουν ακολουθία. Η εκπαίδευση στο στρώμα εξόδου ισοδυναμεί με την επίλυση ενός προβλήματος **γραμμικής παλινδρόμησης (Linear**

Regression). Μια συνήθης μέθοδος που εφαρμόζεται είναι η **παλινδρόμηση κορυφής (ridge regression)**, κατά την οποία η εκπαίδευση των βαρών εξόδου πραγματοποιείται με την ακόλουθη σχέση:

$$W_{out} = (X^T X + \lambda I)^{-1} X^T Y$$

όπου X είναι ο πίνακας των καταστάσεων της δεξαμενής για όλες τις χρονικές στιγμές, Y είναι ο πίνακας των επιθυμητών εξόδων για όλες τις χρονικές στιγμές, λ είναι η παράμετρος κανονικοποίησης της μεθόδου ridge regression, I είναι μοναδιαίος πίνακας. Η εκπαίδευση γίνεται με την ελαχιστοποίηση του σφάλματος μεταξύ των επιθυμητών εξόδων και των εξόδων που παράγονται από το ESN, συνυπολογίζοντας τα βάρη του στρώματος εξόδου W_{out} .

Στα κρυφά στρώματα των ESN, οι ενεργοποιήσεις των νευρώνων είναι μη γραμμικές. Η *φασματική ακτίνα (spectral radius)* του πίνακα βαρών των κρυφών συνδέσεων (W_{hh}) είναι πολύ σημαντική για τη σταθερότητα του συστήματος, δηλαδή για τη διατήρηση της Echo State Property. Πιο συγκεκριμένα, αποτελεί τη *μέγιστη ιδιοτιμή (eigenvalue)* του πίνακα W_{hh} που πρέπει να είναι 1. Αυτό μπορεί να επιτευχθεί με την *κανονικοποίηση (normalization)* του πίνακα W_{hh} βάσει της απόλυτης τιμής της μέγιστης ιδιοτιμής του.

Η παραδοχή ότι η φασματική ακτίνα θα πρέπει να είναι ακριβώς ίση με 1 μπορεί να είναι αυστηρή, καθώς οι μη γραμμικές ενεργοποιήσεις των νευρώνων είναι αυτές που μειώνουν την επιρροή των πρώιμων καταστάσεων στο δίκτυο, με την εμφάνιση του προβλήματος του μηδενισμού των μερικών παραγώγων τους (vanishing gradients). Αυτή η ρύθμιση της φασματικής ακτίνας προσφέρει καλύτερη απόδοση πρόβλεψης σε ακολουθίες που διαθέτουν δεδομένα με χρονική εξάρτηση, και θέτουν χαρακτηριστικά μνήμης.

Ένα άλλο ιδιαίτερο χαρακτηριστικό των ESN είναι η πολυπλοκότητα στις διασυνδέσεις, που θέτει ο πίνακας βαρών W_{hh} των διασυνδέσεων των κρυφών μονάδων (hidden units) της δεξαμενής και το γεγονός ότι ένα μέρος των βαρών στον πίνακα έχουν μηδενική τιμή. Αυτή η τεχνική βοηθά στη μείωση της επιρροής που προσδίδουν οι κρυφές καταστάσεις και βελτιώνει την σταθερότητα των υπολογισμών. Μια στρατηγική είναι να χωριστούν οι κρυφές μονάδες σε ομάδες και να επιτρέπεται η σύνδεση μόνο μεταξύ μονάδας μιας συγκεκριμένης κατηγορίας, μέθοδος που οδηγεί στη δημιουργία ανεξάρτητων υποδικτύων. Αυτή η τεχνική μπορεί να θεωρηθεί ως ένα είδος ομαδοποίησης πολλών ESN, όπου κάθε υποδίκτυο μοντελοποιεί και μαθαίνει διαφορετικές δυναμικές σχέσεις.

Τα ESN αποδίδουν πολύ καλά όταν το στρώμα εισόδου τους είναι μικρής διάστασης. Αυτό συμβαίνει επειδή, η δομή τους τους επιτρέπει να επεξεργάζονται αποτελεσματικά δεδομένα σε μεγάλους χρονικούς άξονες με επαναλήψεις, όμως δεν προσαρμόζονται καλά όταν οι διαστάσεις των εισόδων είναι μεγάλες.

Μπορούν να χρησιμοποιηθούν και για την αρχικοποίηση των βαρών άλλων αναδρομικών νευρωνικών δικτύων. Έχουν εξαιρετικές επιδόσεις σε προβλήματα πρόβλεψης χρονοσειρών. Ένα συνηθισμένο τέχνασμα είναι η εκπαίδευση του μοντέλου ώστε να προβλέπει τιμές του μέλλοντος, δηλαδή η εκπαίδευση να γίνεται με εισόδους από τη χρονική στιγμή t και ως στόχο να έχει τιμές από τη χρονική στιγμή $t+k$, όπου το k είναι ο χρονικός άξονας της πρόβλεψης. Αυτή η προσέγγιση καθιστά τα ESN ισχυρά μη γραμμικά παλινδρομικά αυτόματα.

Παρόλο που τα ESN δεν είναι κατάλληλα για την επεξεργασία εισόδων μεγάλης διάστασης (π.χ. κείμενα), μπορούν να χρησιμοποιηθούν για την αρχικοποίηση απλών RNN. Είναι πιο σύνθετα στη παραμετροποίησή τους, καθώς η επιτυχία τους εξαρτάται από τη δομή της δεξαμενής και τις κατάλληλες τιμές για τις συνδέσεις, που πρέπει να επιλεγούν κατάλληλα. Έπειτα πρέπει να εκπαιδευτεί το στρώμα εξόδου και στη συνέχεια να ξεκινήσει η διαδικασία της εκπαίδευσης με τη διαδικασία αντίστροφης διάδοσης σφάλματος.

Τα Echo State Networks προσφέρουν μια εξαιρετικά αποδοτική προσέγγιση για την εκπαίδευση νευρωνικών δικτύων πάνω σε χρονοσειρές, καθώς αξιοποιούν την τυχαιότητα στο κρυφό στρώμα (δεξαμενή) για να αναπτύξουν πολύπλοκους συνδυασμούς. Είναι ιδανικά για την πρόβλεψη χρονοσειρών σε προβλήματα μικρής ή μέτριας κλίμακας, όπου οι είσοδοι δεν έχουν μεγάλες διαστάσεις. Παράλληλα, η δυνατότητα χρήσης τους ως προ-εκπαιδευτικά βήματα για τα RNN τα καθιστά χρήσιμα και σε πρακτικές εφαρμογές. [19][20][21][22]

4

Εφαρμογή Reservoir Computing σε Rayleigh Fading Channel

4.1 Πρόβλεψη NARMA ακολουθίας με Reservoir Computing:

4.1.1 Αρχικοποίηση ακολουθίας

Χρησιμοποιήθηκε το πακέτο κώδικα για την εκπαίδευση τεχνητού νευρωνικού δικτύου ESN, από το ινστιτούτο Fraunhofer IAIS (Intelligent Analysis and Information Systems), του H. Jaeger σε περιβάλλον λογισμικού MATLAB. Αρχικά εξετάστηκαν οι δυνατότητες του συγκεκριμένου είδους τεχνητού νευρωνικού δικτύου, μελετώντας τις παραμέτρους της δεξαμενής του (reservoir parameters).

Στην παρακάτω εικόνα παρουσιάζεται ο τρόπος που δημιουργήθηκε η δομή (struct) του ESN με τις εξής παραμέτρους:

```
esn = generate_esn(nInputUnits, nInternalUnits, nOutputUnits, ...
    'spectralRadius', 0.9, 'inputScaling', inputScalingVector, 'inputShift', inputShiftVector, ...
    'teacherScaling', 0.3, 'teacherShift', -0.2, 'feedbackScaling', 0, ...
    'type', 'plain_esn', 'methodWeightCompute', 'pseudoinverse');
```

Εικόνα 25: Παράμετροι του ESN

Στη συνέχεια δημιουργήθηκε ένα σύνολο δεδομένων διακριτού μήκους sequenceLength ακολούθως:

```
sequenceLength = 10000;
[inputSequence outputSequence] = generate_NARMA_sequence(sequenceLength, systemOrder);
```

Εικόνα 26: Αρχικοποίηση του dataset

Με τον ακόλουθο τρόπο παράχθηκε μια τυχαία ακολουθία εισόδου (inputSequence) και μια ακολουθία NARMA ως έξοδος (outputSequence).

```
%%% create input

inputSequence = [ones(sequenceLength,1) rand(sequenceLength,1)];
% use the input sequence to drive a NARMA equation

outputSequence = 0.1*ones(sequenceLength,1);

for i = memoryLength + 1 : sequenceLength
    % insert suitable NARMA equation on r.h.s., this is just an ad hoc
    % example
    outputSequence(i,1) = 0.7 * inputSequence(i-memoryLength,2) + 0.1 ...
        + (1 - outputSequence(i-1,1)) * outputSequence(i-1,1);
end
```

Εικόνα 27: Δημιουργία NARMA ακολουθίας

Η *NARMA* (*Nonlinear AutoRegressive Moving Average*) είναι ένα δυναμικό μαθηματικό σύστημα που μοντελοποιεί μη γραμμικά συστήματα μεταβλητής τάξης (*system order*) χρησιμοποιώντας προηγούμενες τιμές, τόσο της εισόδου όσο και της εξόδου. Ουσιαστικά, πρόκειται για μια επέκταση των γραμμικών ARMA μοντέλων, αλλά με μη γραμμικούς όρους. Χρησιμοποιείται για να περιγράψει συστήματα των οποίων η έξοδος $y(t)$ κάποιας χρονικής στιγμής t εξαρτάται από:

- Προηγούμενες τιμές της εξόδου (autoregressive).
- Προηγούμενες τιμές της εισόδου (moving average).
- Μη γραμμικούς όρους που επηρεάζουν την αλληλεπίδραση των προηγούμενων τιμών εισόδου/εξόδου.

Η εξίσωση που προκύπτει η έξοδος από ένα NARMA σύστημα είναι της μορφής:

$$y(t+1) = f[y(t), y(t-1), \dots, y(t-p), u(t), u(t-1), \dots, u(t-q)]$$

όπου $y(t)$ είναι η έξοδος του συστήματος στη χρονική στιγμή t , $u(t)$ είναι η είσοδος στη χρονική στιγμή t , p εκφράζει πόσες προηγούμενες τιμές εξόδου λαμβάνονται υπόψη, q εκφράζει πόσες προηγούμενες τιμές εισόδου λαμβάνονται υπόψη. Η συνάρτηση f είναι μη γραμμική και μπορεί να περιλαμβάνει πολλούς διαφορετικούς τύπους όρων.

Η τάξη (SystemOrder) μιας NARMA αναφέρεται στον αριθμό των προηγούμενων τιμών εισόδου ή εξόδου που επηρεάζουν την τρέχουσα έξοδο. Στη συγκεκριμένη περίπτωση εφαρμόστηκε ένα σύστημα τάξης 10 (NARMA10), το οποίο εξαρτάται από τις προηγούμενες 10 τιμές εισόδου ή/και εξόδου.

Στη περίπτωση της δοκιμής που εκτελέστηκε, η είσοδος `inputSequence` αποτελείται από μια στήλη με άσσους και μια δεύτερη στήλη που περιέχει τυχαίους αριθμούς παραγόμενους από τη συνάρτηση `rand(sequenceLength,1)`, οι οποίοι κυμαίνονται στο διάστημα $[0,1]$.

Η έξοδος (`outputSequence`) που παράχθηκε εκφράζεται από τη σχέση:

$$y(n) = 0.7 \cdot u(n-10) + 0.1 + (1 - y(n-1)) \cdot y(n-1)$$

όπου $y(n)$ είναι η έξοδος στη χρονική στιγμή n , $u(n-10)$ είναι η είσοδος τη χρονική στιγμή $n-10$, δηλαδή η είσοδος με καθυστέρηση 10 χρονικών βημάτων (επειδή το `memoryLength=10`), ο όρος $0.7 \cdot u(i-10)$ υπολογίζει τη συνεισφορά της εισόδου με καθυστέρηση 10 χρονικών βημάτων. Ο μη γραμμικός όρος $(1-y(i-1)) \cdot y(i-1)$ είναι το πιο σύνθετο μέρος, καθώς η έξοδος $y(i-1)$ υπολογίζεται με βάση την προηγούμενη τιμή της εξόδου και μπορεί να προκαλέσει αυξομειώσεις στην τιμή της εξόδου, ανάλογα με την αρχική συνθήκη και τη δυναμική του συστήματος. Το εύρος τιμών θα είναι περίπου από 0 έως 1.

[25][26]

Στη συνέχεια προσαρμόστηκε η τιμή της φασματικής ακτίνας (*spectral radius*) στα βάρη των εσωτερικών μονάδων, με τη τιμή του αντικείμενου `esn.spectralRadius` ακολούθως:

```
esn.internalWeights = esn.spectralRadius * esn.internalWeights_UnitSR;
```

Εικόνα 28: Προσαρμογή Spectral Radius

4.1.2 Εκπαίδευση του ESN

Στη συνέχεια η ακολουθία `inputSequence` χωρίστηκε σε δύο τμήματα. Το ένα τμήμα χρησιμοποιήθηκε για την εκπαίδευση του ESN και το άλλο για τη δοκιμή του στη πρόβλεψη. Το ίδιο έγινε και με την ακολουθία `outputSequence`.

```
train_fraction = 0.5 ; % use 50% in training and 50% in testing
[trainInputSequence, testInputSequence] = ...
    split_train_test(inputSequence, train_fraction);
[trainOutputSequence, testOutputSequence] = ...
    split_train_test(outputSequence, train_fraction);
```

Εικόνα 29: Διαχωρισμός training & testing dataset

Στη συνέχεια το ESN αρχικοποιήθηκε με είσοδο την ακολουθία `trainInputSequence` και με έξοδο την ακολουθία `trainOutputSequence`. Η `trainOutputSequence` αποτελεί τις τιμές-στόχους που θα χρησιμοποιηθούν στην εξαναγκασμένη εκπαίδευση (teacher forcing). Επιπλέον προσπελάσσεται η δομή του esn με τις παραμέτρους του και ένα πλήθος συμβόλων (`nForgetPoints`) τα οποία το νευρωνικό σβήνει με σκοπό την αποφυγή αστάθειας από αρχικά δεδομένα.

```
[trainedEsn stateMatrix] = ...
    train_esn(trainInputSequence, trainOutputSequence, esn, nForgetPoints) ;
```

Εικόνα 30: Εκπαίδευση ESN

Έπειτα με τη χρήση της συνάρτησης `compute_statematrix()`, υπολογίστηκε ο πίνακας καταστάσεων (`stateCollection`). Αυτός ο πίνακας περιέχει τις εσωτερικές καταστάσεις του δικτύου καθώς και τα αντίστοιχα εισαγόμενα δεδομένα, τα οποία χρησιμοποιούνται για την εκπαίδευση του ESN.

```
stateCollection = compute_statematrix(trainInput, trainOutput, trained_esn, nForgetPoints);
```

Εικόνα 31: Υπολογισμός πίνακα καταστάσεων.

Ο πίνακας `stateCollection` είναι υπεύθυνος για τη συλλογή των καταστάσεων του δικτύου. Το μέγεθός του εξαρτάται από τη ποσότητα των δεδομένων και τον αριθμό των νευρώνων του δικτύου (μονάδες εισόδου και εσωτερικές μονάδες). Η συνάρτηση ελέγχει αν υπάρχει κάποια αρχική κατάσταση που δίνεται εξωτερικά μέσω κάποιας παραμέτρου. Αν δεν υπάρχει, ξεκινάει με μηδενική κατάσταση για όλες τις μονάδες του δικτύου (εσωτερικές μονάδες, μονάδες εισόδου και εξόδου).

Πριν ξεκινήσει η εκπαίδευση, το δίκτυο μπορεί να αγνοήσει τα πρώτα δεδομένα (`nForgetPoints`) και να τροφοδοτήσει την ίδια είσοδο πολλές φορές στο δίκτυο, ώστε να σταθεροποιηθεί η εσωτερική του κατάσταση. Αυτή η διαδικασία είναι χρήσιμη όταν, είναι επιθυμητό να προσαρμοστεί το δίκτυο πριν αρχίσει η συλλογή καταστάσεων. Οι είσοδοι προσαρμόζονται σύμφωνα με τις παραμέτρους του ESN (`inputScaling`, `inputShift`).

Στην συνέχεια με τη χρήση της συνάρτησης `compute_teacher()` υπολογίζεται και δημιουργείται ένας πίνακας που περιέχει τις επιθυμητές τιμές εξόδου (τιμές εκπαίδευσης) του ESN. Αυτές οι τιμές χρησιμοποιούνται για την εκπαίδευση του δικτύου, όπως φαίνεται παρακάτω:

```
teacherCollection = compute_teacher(trainOutput, trained_esn, nForgetPoints) ;
```

Εικόνα 32: Υπολογισμός πίνακα τιμών του *teacher*

Αντί για τη μέθοδο *Wiener Hopf*, η οποία από μαθηματικής άποψης είναι πρόβλημα επίλυσης γραμμικής παλινδρόμησης (Linear Regression) και έχει μεγάλη αριθμητική αστάθεια, χρησιμοποιήθηκε η μέθοδος **ψευδοαντιστροφής (*pseudoinverse*)**, η οποία προσφέρει έναν απλό και άμεσο τρόπο για την εκπαίδευση των βαρών του στρώματος εξόδου (*outputWeights*), ειδικά όταν η γραμμική εξίσωση δεν είναι πλήρως ορισμένη στις διαστάσεις της ή υπάρχει **υπεραναγνώριση των δεδομένων (*overfitting*)**. Είναι μια προσέγγιση της λύσης για το πρόβλημα επίλυσης της γραμμικής παλινδρόμησης, όταν το σύστημα των εξισώσεων είναι αδύνατο να λυθεί απευθείας με τη μορφή που έχει. Είναι δηλαδή μια μορφή γραμμικής παλινδρόμησης που χρησιμοποιείται ειδικά όταν δεν μπορεί να εφαρμοστεί απευθείας, η κανονική εξίσωση της παλινδρόμησης, λόγω προβλημάτων στην αντιστρεψιμότητά της. Η διαδικασία αυτή πραγματοποιήθηκε με τη χρήση της ακόλουθης συνάρτησης *feval()*, η οποία είναι μεταβλητή συνάρτηση και στη συγκεκριμένη περίπτωση έχει προκαθορισμένη τιμή (*default value*) αυτή της *Pseudoinverse*:

```
trained_esn.outputWeights = feval(trained_esn.methodWeightCompute, stateCollection, teacherCollection) ;
```

Εικόνα 33: Υπολογισμός *weights* εξόδων με *pseudoinverse*

Χρησιμοποιήθηκε δηλαδή για να καλέσει μια συγκεκριμένη συνάρτηση που έχει οριστεί στη δομή *esn{}*, με βάση τα χαρακτηριστικά του δικτύου, σε αυτή τη περίπτωση καλέστηκε η μέθοδος που έχει καθοριστεί στην παράμετρο *esn.methodWeightCompute*, που όπως αναφέρθηκε προηγουμένως έχει προκαθορισμένη τιμή την *Pseudoinverse*.

4.1.3 Δοκιμή του ESN

Η δοκιμή του ESN πραγματοποιήθηκε χρησιμοποιώντας τη συνάρτηση *test_esn()* για την ακολουθία εισόδου εκπαίδευσης (*trainInputSequence*) και για την άγνωστη ακολουθία (*testInputSequence*) που προορίζεται για τη δοκιμή όπως φαίνεται στη παρακάτω εικόνα:

```
predictedTrainOutput = test_esn(trainInputSequence, trainedEsn, nForgetPoints);
predictedTestOutput = test_esn(testInputSequence, trainedEsn, nForgetPoints);
```

Εικόνα 34: Δοκιμή του ESN

Η συγκεκριμένη συνάρτηση υπολογίζει τις καταστάσεις του δικτύου (*stateCollection*) που αντιστοιχούν στην ακολουθία εισόδου που είχαμε τροφοδοτήσει προηγουμένως για τη διαδικασία της εκπαίδευσης, με τον ακόλουθο τρόπο:

```
stateCollection = compute_statematrix(inputSequence, [], esn, nForgetPoints) ;
```

Εικόνα 35: Δημιουργία καταστάσεων της ακολουθίας εκπαίδευσης

Στη συνέχεια προκύπτει η προβλεπόμενη ακολουθία `outputSequence`, πολλαπλασιάζοντας τις παραπάνω καταστάσεις (`stateCollection`) με τα εκπαιδευμένα βάρη του στρώματος εξόδου, με τον εξής τρόπο:

```
outputSequence = stateCollection * esn.outputWeights' ;
```

Εικόνα 36: Προβλεπόμενη έξοδος

Αφού υπολογίστηκε η έξοδος, εφαρμόζεται η αντίστοιχη συνάρτηση ενεργοποίησης εξόδου (που έχει οριστεί ως αντικείμενο `esn.outputActivationFunction`), για να μετατραπούν οι έξοδοι στις τελικές προβλέψεις:

```
outputSequence = feval(esn.outputActivationFunction, outputSequence);
```

Εικόνα 37: Activation Function του ESN

Εδώ η προκαθορισμένη τιμή του αντικειμένου `esn.outputActivationFunction` είναι η συνάρτηση `identity()`, η οποία απλώς δίνει την έξοδο όπως είναι.

Τέλος, η έξοδος αναπροσαρμόζεται και αναστρέφεται η μετατόπιση (που είχε εφαρμοστεί κατά την εκπαίδευση) για να επιστρέψει στις αρχικές διαστάσεις και τιμές της, όπως φαίνεται ακολούθως:

```
outputSequence = outputSequence - repmat(esn.teacherShift',[nOutputPoints 1]) ;  
outputSequence = outputSequence / diag(esn.teacherScaling) ;
```

Εικόνα 38: Αναστροφή της κλιμάκωσης και μετατόπισης (de-scaling and de-shifting)

Για να γίνει κατανοητό πως συμπεριφέρεται η δομή που μόλις κατασκευάστηκε, πραγματοποιήθηκε δοκιμή του ESN για διαφορετικές τιμές εσωτερικών μονάδων (από 0 έως 1000), και για διαφορετικές τιμές της φασματικής ακτίνας (από 0 έως 2), καθώς καταγράφηκε το μέτρο σφάλματος NRMSE (Normalized Root Mean Square Error) για κάθε συνδυασμό. Όσο περισσότερες μονάδες εφαρμόζονται, τόσο πιο πολύ αργούν οι υπολογισμοί του ESN καθώς γίνεται πιο περίπλοκη η δομή της δεξαμενής.

Το NRMSE υπολογίζεται με τον εξής τύπο:

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{N} \sum_{n=1}^N (y_{\text{pred},n} - y_{\text{true},n})^2}}{\sigma_{y_{\text{true}}}}$$

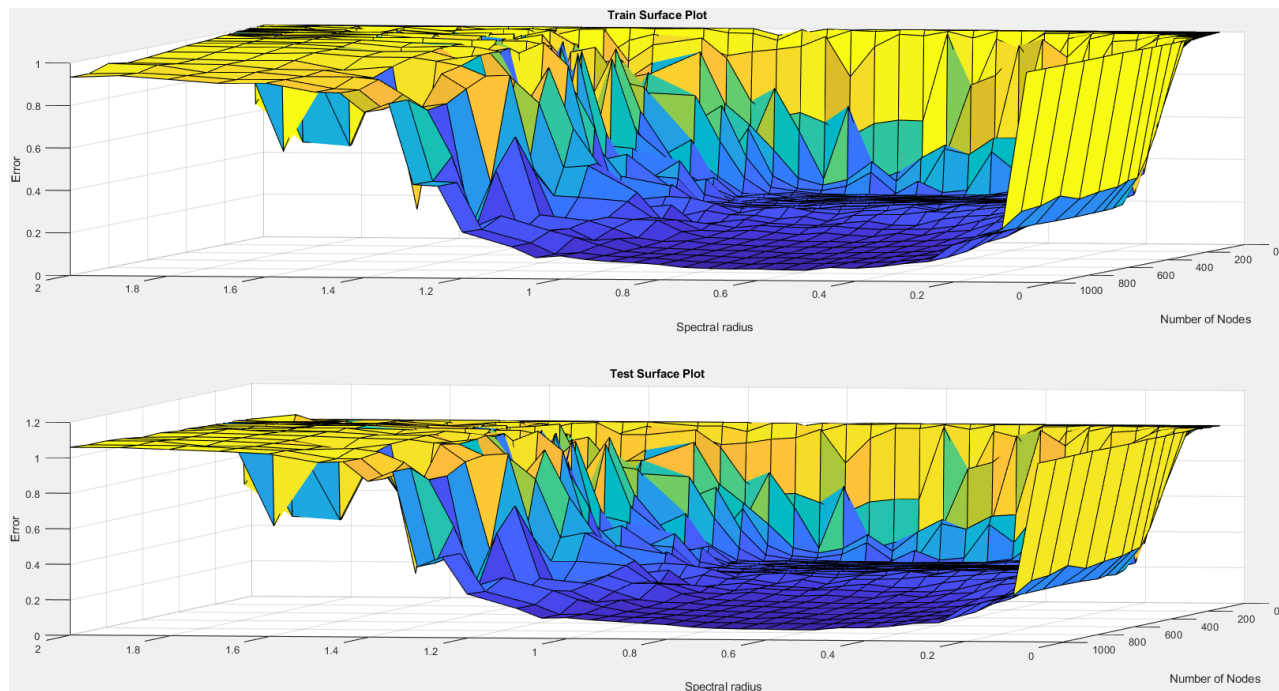
όπου $y_{\text{pred},n}$ είναι η εκτιμώμενη τιμή (predicted output) για το n-οστό δείγμα, $y_{\text{true},n}$ είναι η πραγματική τιμή (true output) για το n-οστό δείγμα, N είναι ο αριθμός δειγμάτων και $\sigma_{y_{\text{true}}}$ είναι η τυπική απόκλιση των πραγματικών τιμών (true values) και υπολογίζεται ως:

$$\sigma_{y_{\text{true}}} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_{\text{true},i} - \overline{y_{\text{true}}})^2}$$

όπου $\overline{y_{\text{true}}}$ είναι η μέση τιμή των πραγματικών τιμών. [23][24][25]

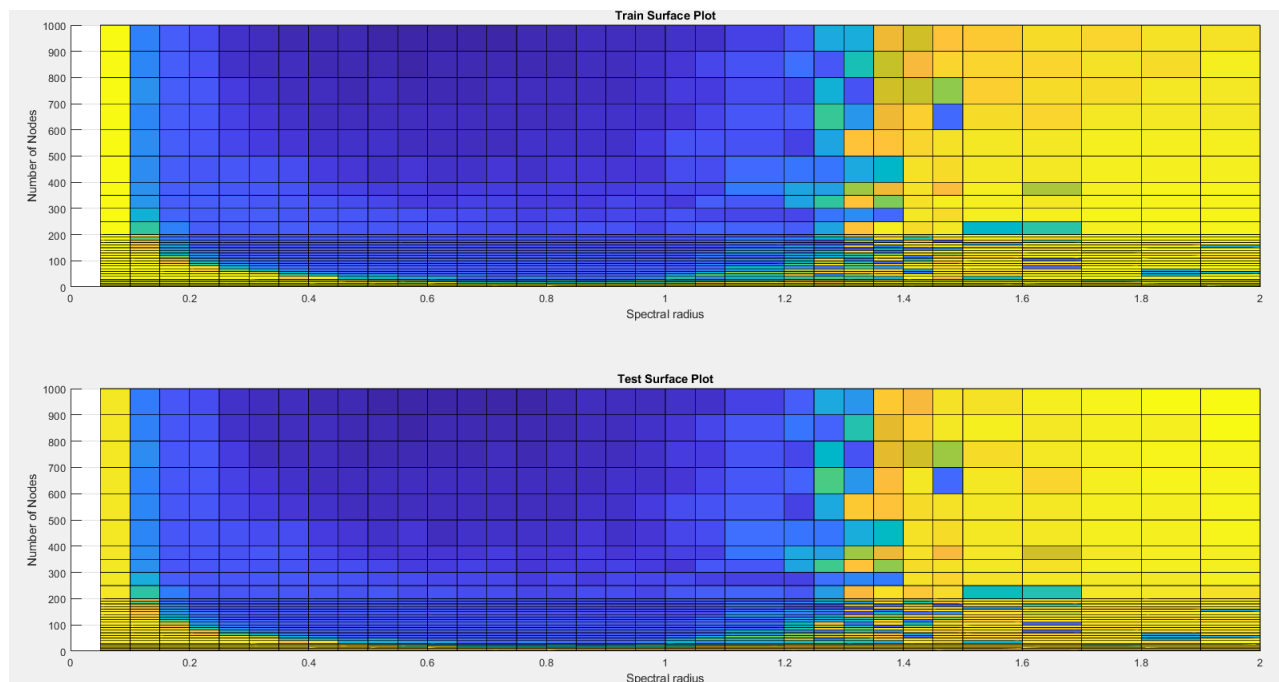
4.1.4 Αποτελέσματα δοκιμών του ESN για τη NARMA10

Η δοκιμή του ESN, πραγματοποιήθηκε για τη πρόβλεψη της ακολουθίας `trainInputSequence` με την οποία εκπαιδεύτηκε και για τη πρόβλεψη της ακολουθίας `testInputSequence`, η οποία είναι άγνωστη στο ESN. Τα αποτελέσματα παρουσιάζονται παρακάτω:



Εικόνα 39: Γενικό Surface Diagram συμπεριφοράς ESN

Στα συγκεκριμένα διαγράμματα, όσο πιο σκούρο εμφανίζεται το χρώμα τόσο μικρότερο είναι το σφάλμα πρόβλεψης. Παρατηρείται πως όσο αυξάνεται το πλήθος των εσωτερικών μονάδων και όσο η τιμή της φασματικής ακτίνας κυμαίνεται περίπου στο διάστημα $[0.6, 0.9]$, τόσο καλύτερη θα είναι η απόδοση που θα έχει το ESN σε προβλέψεις.



Εικόνα 40: Πανοραμικό Surface Diagram συμπεριφοράς ESN

4.2 Πρόβλεψη διαμορφωμένης ακολουθίας 4-PAM μέσα από κανάλι εξασθένισης Rayleigh και AWGN με χρήση του ESN.

Σε αυτό το κεφάλαιο παρουσιάζονται οι δυνατότητες που έχει το ESN, στη πρόβλεψη μίας ακολουθίας που προκύπτει, όταν εφαρμόζεται μοντέλο καναλιού Rayleigh, το οποίο προσομοιώνει τις πραγματικές συνθήκες του μέσου διάδοσης, που αντιμετωπίζει ένα εκπεμπόμενο ηλεκτρομαγνητικό σήμα στη πορεία του για να φτάσει στον δέκτη. Η βασική αρχιτεκτονική και ο τρόπος υπολογισμών παρέμεινε ίδιος με του προηγούμενου κεφαλαίου (4.1), όπου εκπαιδεύτηκε και δοκιμάστηκε το ESN με μια ακολουθία NARMA10. Επομένως σε αυτό το κεφάλαιο θα αναλυθούν κυρίως τα νέα κομμάτια κώδικα που δημιουργήθηκαν και οι μετατροπές που πραγματοποιήθηκαν στα ήδη υπάρχοντα αρχεία.

4.2.1 Δημιουργία Rayleigh Channel

Κατασκευάστηκε μια ακολουθία 10.000 συμβόλων. Αντί για τη συνάρτηση της NARMA αυτή τη φορά δημιουργήθηκε η συνάρτηση myChannel() μέσα στην οποία δημιουργείται η δομή ενός καναλιού Rayleigh, με τον εξής τρόπο που φαίνεται στην παρακάτω εικόνα:

```
% Creating a Rayleigh channel object
Rayleigh = comm.RayleighChannel(...
    'SampleRate', 2*channelRate, ...
    'PathDelays', pathDelays, ...
    'AveragePathGains', averagePathGains, ...
    'NormalizePathGains', true, ...
    'PathGainsOutputPort', true);
```

Εικόνα 41: Δημιουργία Rayleigh Channel

Αρχικά ορίστηκε ο ρυθμός δειγματοληψίας του καναλιού (SampleRate) ως 2 φορές τη συχνότητα του σήματος που είναι ορισμένη στα 3 GHz το οποίο αποτελεί τη συχνότητα που δειγματοληπτεί το comm.RayleighChannel, την ακολουθία που προσπελάσσεται από αυτό. Η συχνότητα δειγματοληψίας καθορίζει πόσα από τα δείγματα του σήματος θα υποβληθούν σε επεξεργασία από το κανάλι. Εάν η συχνότητα δειγματοληψίας είναι χαμηλή, το κανάλι μπορεί να παραλείψει σημαντικά τμήματα τα ακολουθίας του σήματος. Εάν είναι υψηλή, θα επεξεργάζεται κάθε δείγμα πιο συχνά, δίνοντας μια πιο λεπτομερή προσομοίωση των επιδράσεων του καναλιού.

Για παράδειγμα στη περίπτωση ενός σήματος με ρυθμό 1 KHz (1000 δείγματα το δευτερόλεπτο). Εάν οριστεί συχνότητα δειγματοληψίας του καναλιού Rayleigh σε 500 Hz, τότε μόνο τα μισά από τα δείγματα του σήματος θα περάσουν από την επεξεργασία του καναλιού. Αντίθετα, αν η συχνότητα είναι 2 KHz, το κανάλι θα επεξεργαστεί όλα τα δείγματα και θα εφαρμόζει τα φαινόμενα εξασθένισης πιο συχνά, κάτι που μπορεί να δείχνει πιο ρεαλιστικό για περιβάλλοντα που έχουν απότομες μεταβολές (διαλείψεις) ή πολλαπλά μονοπάτια. Έτσι εφαρμόστηκε 2 φορές αυτή τη συχνότητα ικανοποιείται το όριο του κριτηρίου Nyquist.

Για να λειτουργήσει σωστά το κανάλι, ορίστηκαν οι καθυστερήσεις των διαδρομών (pathDelays), οι οποίες είναι 4 διαδρομές:

```
% Populate pathDelays array using a for loop
for i = 0:numPaths-1
    pathDelays(i+1) = i * (1/channelRate); % Indexing starts from 1 in MATLAB
end
```

Εικόνα 42: Ορισμός pathDelays

Στο παραπάνω κομμάτι κώδικα ορίζεται ότι, η πρώτη διαδρομή θα έχει καθυστέρηση $1T=1/\text{channelRate}$, η δεύτερη θα έχει $2T=2/\text{channelRate}$ κ.ο.κ.

Η κάθε διαδρομή διαθέτει και το αντίστοιχο κέρδος (gain) της. Ορίζουμε τα κέρδη όπως βλέπουμε παρακάτω:

```
averagePathGains = [-1 -5 -10 -20]; % Path gains
```

Εικόνα 43: Ορισμός των averagePathGains

Οι τιμές αυτές έχουν μονάδα μέτρησης το $\text{dB} = 10 \log_{10} \left(\frac{P(\text{out})}{P(\text{in})} \right)$, άρα σε όλες τις διαδρομές εφαρμόζονται οι αντίστοιχες αποδυναμώσεις στο σήμα.

4.2.2 Διαμόρφωση της ακολουθίας με 4-PAM

Συνέχεια θα χρησιμοποιήθηκε τη συνάρτηση pammod() για τη διαμόρφωση μιας τυχαία παραγόμενης ακολουθίας όπως φαίνεται στη συγκεκριμένη φωτογραφία:

```
% Generating random PAM-modulated sequence
random_values = randi([0 M-1], sequenceLength, 1);
pam_values = pammod(random_values, M, 0, 'gray'); % PAM modulation
```

Εικόνα 44: 4-PAM Modulation

Το διαμορφωμένο σήμα (pam_values) έχει πλέον κωδικοποιηθεί με τυχαίες τιμές του 4-PAM (-3, -1, 1, 3) και κατά κώδικα Gray, ώστε κάθε σύμβολο να έχει τουλάχιστον διαφορά 1 bit από το προηγούμενο και το επόμενο, με σκοπό τη καλύτερη ανίχνευσή τους στο τέλος και τη μείωση σφαλμάτων.

Έπειτα πολλαπλασιάστηκε με ένα φέρον για να παρομοιάσει πιο πιστά τη διαδικασία της ασύρματης εκπομπής αν και θα μπορούσε να παραληφθεί στα πλαίσια της προσομοίωσης, καθώς ένα PAM βασικής ζώνης με ένα ζωνοπερατό επιφέρει τα ίδια αποτελέσματα στη συγκεκριμένη περίπτωση.

```
%From BaseBand to Bandpass
t = (0:length(pam_values)-1).' / channelRate;
carrier = cos(2*pi*channelRate*t);
bandpass_signal = real(pam_values) .* carrier;
```

Εικόνα 45: Δημιουργία φέροντος

Ουσιαστικά η καθορισμένη συχνότητα εκπομπής χρειάζεται για να μεταδοθεί το σήμα σε μια συχνότητα κατά την οποία επιτρέπεται η εκπομπή και είναι πιο ανθεκτική σε φυσικά φαινόμενα που επιδρούν στο σήμα τα οποία όμως εμείς δεν λαμβάνουμε υπόψιν στη συγκεκριμένη εργασία.

Έπειτα το κανάλι Rayleigh εφαρμόστηκε στο διαμορφωμένο σήμα (pam_values) με τον ακόλουθο τρόπο:

```
% Passing the signal through the Rayleigh channel
[received_signal, ~] = Rayleigh(bandpass_signal);
```

Εικόνα 46: Signal through Rayleigh Channel

Επιπρόσθετα εφαρμόστηκε AWGN, θέτοντας το σηματοθορυβικό λόγο SNR (Signal to Noise Ratio) σε μονάδες dB, η οποία αποτελεί σημαντική παράμετρο σε συστήματα επικοινωνιών και επεξεργασίας σήματος, που μετρά την ισχύ του επιθυμητού σήματος σε σχέση με την ισχύ του θορύβου.

```
SNR_dB = 20; % Signal-to-Noise Ratio (SNR) in dB
noisy_signal = awgn(received_signal, SNR_dB, 'measured');
```

Εικόνα 47: Signal through AWGN channel

Ένα υψηλό SNR σημαίνει ότι το σήμα είναι ισχυρό σε σχέση με τον θόρυβο, ενώ ένα χαμηλό SNR υποδηλώνει ότι ο θόρυβος είναι συγκρίσιμος ή μεγαλύτερος από το σήμα, κάτι που μπορεί να οδηγήσει σε λάθη στην επικοινωνία. Η σχέση που το εκφράζει είναι:

$$SNR_{dB} = 10 \log_{10} \left(\frac{P_{signal}}{P_{noise}} \right)$$

Στη συγκεκριμένη περίπτωση θέτεται ένα SNR ώστε να υπολογιστεί από τη συνάρτηση `awgn()` η ισχύς του αντίστοιχου ισχύος θορύβου σε σχέση με την μέση ισχύ του σήματός.

Στο τέλος προσπελαύνεται το σήμα με το θόρυβο (`noisy_signal`) ως `InputSequence` και το διαμορφωμένο αρχικό σήμα `pam_values` ως `outputSequence`, ώστε να ακολουθήσει η διαδικασία της εκπαίδευσης και της δοκιμής του ESN, όπως είδαμε στο κεφάλαιο (4.1).

```
% Creating inputSequence
inputSequence = [ones(sequenceLength, 1), (real(noisy_signal))]; |
% Creating outputSequence
outputSequence = pam_values; % Output sequence is the original PAM-modulated symbols
```

Εικόνα 48: Είσοδος & Έξοδος στο ESN

4.2.3 Δοκιμή του ESN

Για να μετρηθεί καλύτερα η απόδοση του ESN πάνω στο `noisy_signal` που δημιουργήθηκε, θα χρησιμοποιηθεί η μέθοδος μέτρησης σφαλμάτων Bit Error Rate (BER). Αποτελεί μια μετρική που χρησιμοποιείται σε συστήματα ψηφιακών επικοινωνιών για να μετρηθεί η απόδοση του συστήματος. Εκφράζει το ποσοστό των bits που λαμβάνονται λανθασμένα σε σχέση με τα συνολικά bits που μεταδίδονται. Το BER εξαρτάται από πολλούς παράγοντες, όπως είναι το SNR, η πολυπλοκότητα της διαμόρφωσης, οι συνθήκες του καναλιού και ο αλγόριθμος ανίχνευσης που εφαρμόζεται. Η θεωρητική σχέση που το εκφράζει σε διαμόρφωση M-PAM είναι:

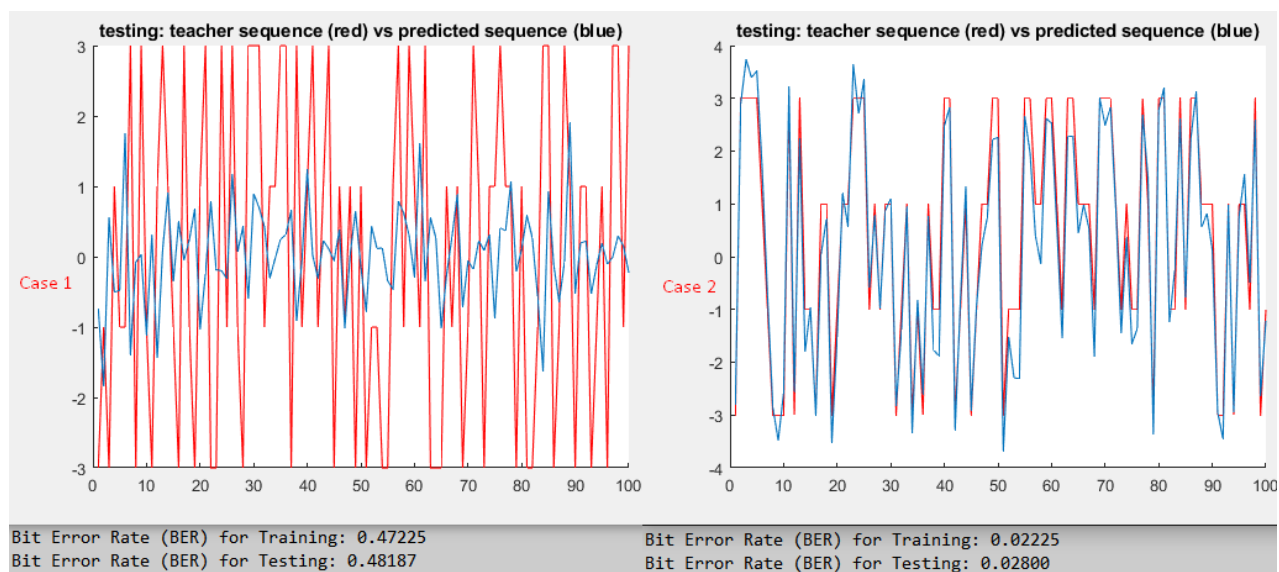
$$BER = \frac{2(M-1)}{M \log_2(M)} \cdot Q \left(\sqrt{\frac{6 \log_2(M)}{M^2-1} \cdot \frac{E_b}{N_0}} \right)$$

όπου M οι βαθμίδες του PAM, $Q(x)$ είναι η συνάρτηση Q η οποία δίνει την πιθανότητα ότι η τιμή ενός τυχαίου σήματος θα υπερβεί το εσωτερικό της όρισμα x , E_b η ενέργεια ανά bit και N_0 είναι η φασματική πυκνότητα ισχύος θορύβου.

Στη συγκεκριμένη περίπτωση όμως χρησιμοποιείται η συνάρτηση `biterr()` η οποία ακολουθεί την εξής σχέση για τον υπολογισμό του BER:

$$BER = \frac{\text{Number of bit errors}}{\text{Total Number of Transmitted bits}}$$

Κατά τη δοκιμή του ESN στο συγκεκριμένο σημείο, λήφθηκαν μετρήσεις οι οποίες είχαν τεράστιες αποκλίσεις μεταξύ τους όπως το παρακάτω παράδειγμα (Case 1 και Case 2):



Εικόνα 49: Σύγκριση BER Cases 1 & 2

Στις δύο αυτές περιπτώσεις χρησιμοποιήθηκαν 200 μονάδες και φασματική ακτίνα 0.9. Η μόνη διαφορά ήταν στο σπόρο (seed) της γεννήτριας συνάρτησης που παράγει τη τυχαία ακολουθία.

Παρατηρείται ότι το ESN αντιμετωπίζει πρόβλημα μνήμης και δεν μπορεί να παράσχει τις απαραίτητες συσχετίσεις για κάθε τυχαία ακολουθία που του δίνεται.

Για το λόγο αυτό έγινε προσθήκη μνήμης στο σύστημα, προσθέτοντας χρήσιμη πληροφορία μέσα στα τυχαία δεδομένα, ώστε το ESN να έχει σταθερότερη απόδοση στη πρόβλεψη τυχαίων ακολουθιών.

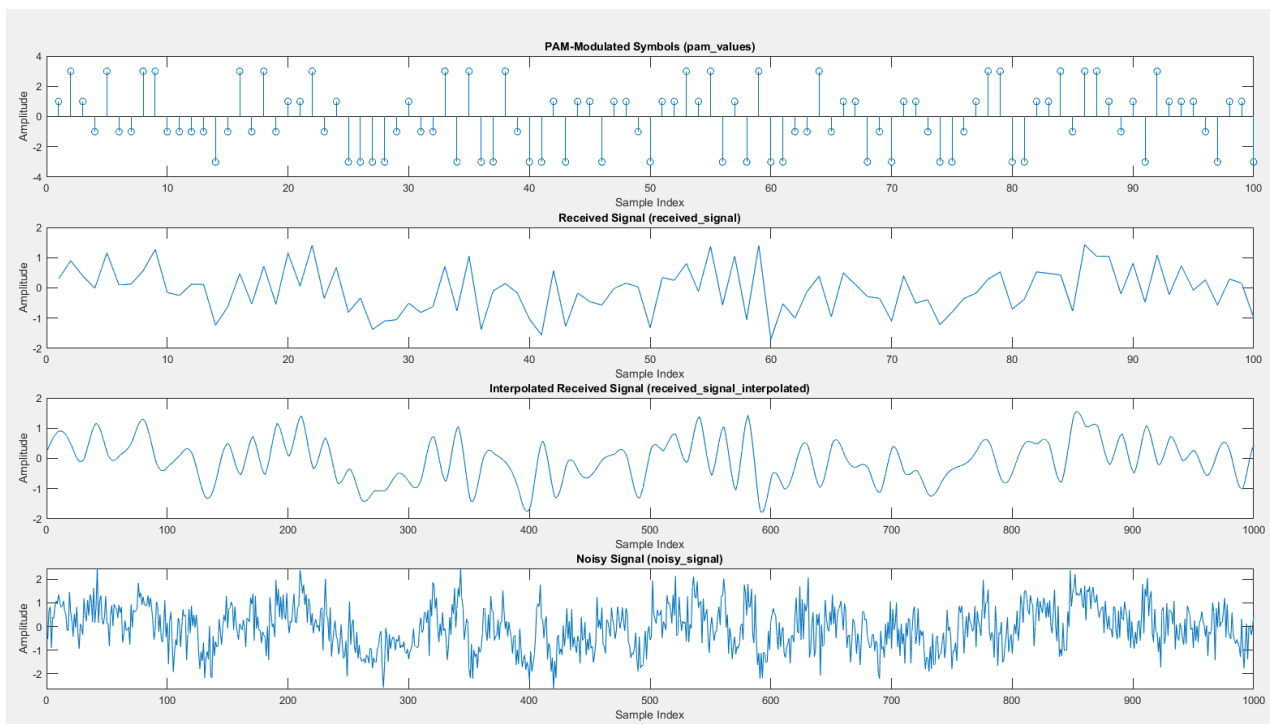
4.2.4 Προσθήκη ψηφιακής μνήμης στο ESN για βελτίωση της πρόβλεψης

Αρχικά εφαρμόστηκε η συνάρτηση `interp()` (interpolation) στην λαμβανόμενη ακολουθία ως έξοδος από την `Rayleigh()` με τον εξής τρόπο:

```
% Interpolation
interp_factor = 10;
received_signal_interpolated = interp(received_signal, interp_factor);
```

Εικόνα 50: Interpolation

Έτσι τώρα για κάθε 1 σημείο της ακολουθίας θέτονται άλλα 10 στη περιοχή τιμών του. Η διαδικασία είναι αρκετά σημαντική, καθώς βοηθά στη βελτίωση της ποιότητας και της ακρίβειας των σημάτων. Παρέχει ένα πιο πλούσιο και καθαρό σήμα, το οποίο μπορεί να επεξεργαστεί πιο αποτελεσματικά.



Εικόνα 51: Plot με τα στάδια του Σήματος

Στη συνέχεια η ακολουθία *inputSequence* προετοιμάστηκε για περαιτέρω επεξεργασία ώστε να είναι κατάλληλη για ανάλυση σε επίπεδο συμβόλου. Η συνάρτηση *pre_processing* είναι υπεύθυνη για τη μετατροπή του λαμβανόμενου σήματος σε μορφή που είναι έτοιμη για περαιτέρω ανάλυση ή επεξεργασία βάσει χαρακτηριστικών (*feature extraction*). Τα δεδομένα κατηγοριοποιούνται σε έναν πίνακα του οποίου κάθε γραμμή περιέχει όλα τα σχετιζόμενα σύμβολα. Η οργανωμένη μορφή των δειγμάτων ανά σύμβολο διευκολύνει τη συνέχεια της επεξεργασίας, συμπεριλαμβανομένου της εφαρμογής διακριτών συντελεστών καθυστέρησης (*taps*) στα δεδομένα, που υλοποιούνται στην *compute_statematrix*, για την καλύτερη ακρίβεια και την απόδοση του ESN.

Ορίζεται μεταβλητή *taps*, η οποία καθορίζει πόσες προηγούμενες χρονικές καταστάσεις θα συμπεριληφθούν κατά τη διαδικασία συλλογής των καταστάσεων. Ουσιαστικά, επεκτείνει τη διάσταση των καταστάσεων για να περιλαμβάνει μια χρονική σειρά δεδομένων. Αυτό το επιτυγχάνεται με δύο συναρτήσεις, την *tap_state()* και την *tap()*, οι οποίες δημιουργούν ένα νέο πίνακα καταστάσεων *stateCollectMat*, ο οποίος περιλαμβάνει τις τρέχουσες καταστάσεις μαζί με τις καταστάσεις των προηγούμενων χρονικών βημάτων. Αυτό επιτρέπει στο μοντέλο να έχει πρόσβαση σε πληροφορίες από το παρελθόν, χαρακτηριστικό πολύ χρήσιμο για την πρόβλεψη.

Η συνάρτησεις *tap()* και *tap_state()* εφαρμόζουν μια λειτουργία που αντιστοιχίζει αυτή ενός γραμμικού φίλτρου στην αρχή του δικτύου, δημιουργούν ένα νέο πίνακα όπου κάθε γραμμή περιλαμβάνει την τρέχουσα κατάσταση και τις γειτονικές καταστάσεις σύμφωνα με τον αριθμό των διακριτών συντελεστών καθυστέρησης. Η *tap()* αναλαμβάνει τις γραμμές και η *tap_state()* τις στήλες.

Με τη συλλογή των καταστάσεων που περιλαμβάνουν τα *taps*, η δομή των δεδομένων γίνεται πιο πλήρης και χρήσιμη για πρόβλεψη χρονοσειρών ή για κατηγοριοποίηση με βάση το ιστορικό των καταστάσεων, και προστίθεται χρήσιμη μνήμη στα δεδομένα προς επεξεργασία.

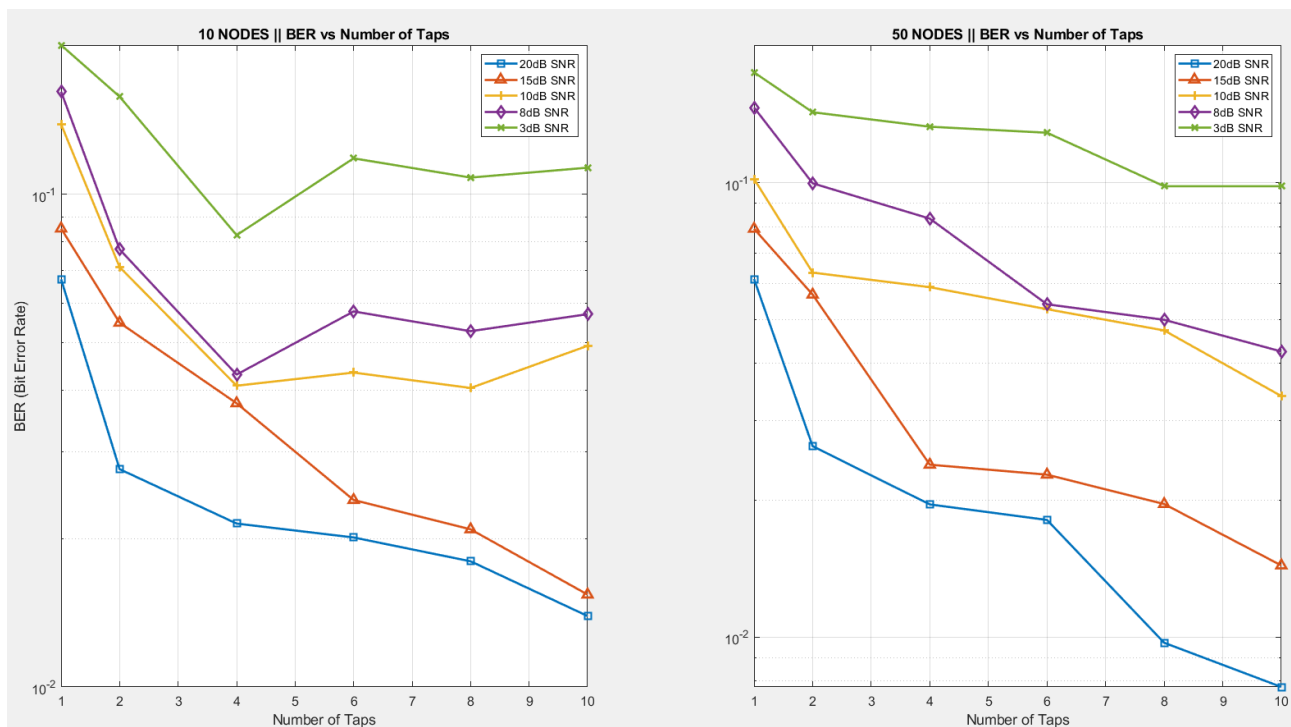
4.2.5 Βελτιωμένη δοκιμή και αποτελέσματα του ESN

Με αιτία τον εμπλουτισμό των δεδομένων, ο αριθμός των εσωτερικών μονάδων μειώθηκε καθώς πλέον τα μεγέθη είναι υπερπολλαπλάσια των προηγούμενων και ο χρόνος εκτέλεσης αυξάνεται σημαντικά.

Συγκεκριμένα πραγματοποιήθηκαν:

50 δοκιμές για 10 εσωτερικές μονάδες και 50 δοκιμές για 50 εσωτερικές μονάδες.

Ο μέσος όρος των BER που λαμβάνεται έχει βελτιωθεί κατά πολύ σε σχέση με πριν, και εξασφαλίζεται μέσο $BER = 10^{-2}$ με τις κατάλληλες συνθήκες.



Εικόνα 52: Αποτελέσματα τελικής δοκιμής του ESN

Παρατηρείται ότι όσο αυξάνεται ο αριθμός των taps τόσο καλύτερη είναι η πρόβλεψη. Επίσης φαίνεται πόσο ο AWGN επηρεάζει τη πρόβλεψη, βάσει της τιμής του SNR(dB) που θέτουμε. Από την αύξηση των 10 εσωτερικών μονάδων στους 50 εσωτερικές μονάδες παρατηρείται επίσης βελτίωση.

[23][24][27][28]

5

Συμπέρασμα.

Στα πλαίσια της εργασίας εκτέλεθηκε ψηφιακό σήμα 4-PAM σε κανάλι διαλείψεων Rayleigh, με ζητούμενο να αντιμετωπιστούν αποδοτικά οι επιπτώσεις του, στο εκπεμπόμενο σήμα.

Για το σκοπό αυτό, αρχικά μελετήθηκαν διαφορετικές μέθοδοι ανάκτησης σήματος όπως, οι ισοσταθμιστές (Equalizers) και τα τεχνητά νευρωνικά δίκτυα (ANN).

Οι διατάξεις ισοσταθμιστών που μελετήθηκαν ήταν:

- Οι Γραμμικοί Ισοσταθμιστές FFE, οι οποίοι δε μπορούν να ισοσταθμίσουν μη γραμμικά φαινόμενα.
- Οι Μη Γραμμικοί Ισοσταθμιστές DFE, οι οποίοι είναι ευάλωτοι στο πρόβλημα διάδοσης σφάλματος (*error propagation*).
- Οι Μη γραμμικοί ισοσταθμιστές Volterra, οι οποίοι μπορούν να αντιμετωπίσουν μη γραμμικά φαινόμενα αλλά γίνονται αρκετά περίπλοκοι στη παραμετροποίηση και αυξάνουν πολύ τη υπολογιστική πολυπλοκότητά του για μεγαλύτερες τάξεις μη γραμμικότητας.

Οι διατάξεις των ANN που μελετήθηκαν ήταν:

- Τα FFN, τα οποία δεν διαθέτουν μνήμη για να διαχειριστούν χρονικές εξαρτήσεις στα δεδομένα.
- Τα CNN, τα οποία ανιχνεύουν τοπικά μοτίβα των δεδομένων και δε μπορούν να διαχειριστούν χρονικές εξαρτήσεις.
- Τα RNN, τα οποία αποτελούν ιδανική λύση για χρονικές εξαρτήσεις σε ακολουθίες, καθώς με τη χρήση των κρυφών καταστάσεων (*hidden states*), διαθέτουν μνήμη.
- Τα LSTM, τα οποία αποτελούν μια πιο εξειδικευμένη δομή RNN, με σκοπό την αντιμετώπιση του προβλήματος *vanishing* και *exploding gradient*, όμως αναπτύσσουν υψηλή υπολογιστική πολυπλοκότητα και καθυστερούν πολύ στη διαδικασία της εκπαίδευσης.
- Τα GRU, τα οποία αποτελούν μια πιο απλοποιημένη εκδοχή των LSTM, ιδανικά για προβλήματα που απαιτούν πιο περιορισμένης μνήμης.

Σε τελικό στάδιο μελετήθηκαν και έπειτα δοκιμάστηκαν οι δυνατότητες των ESN, ως εναλλακτική λύση στο πρόβλημα. Τα ESN μειώνουν τη υπολογιστική πολυπλοκότητα της διαδικασίας της εκπαίδευσης, καθώς εκπαιδεύουν μόνο τα βάρη στο στρώμα εξόδου τους, αντιστέκονται στο πρόβλημα μηδενισμού και απειρισμού των μερικών παραγώγων κατά τη διαδικασία αντίστροφης διάδοσης σφάλματος και τέλος διαχειρίζονται ικανοποιητικά τα μη γραμμικά φαινόμενα του καναλιού.

Βιβλιογραφία & Πηγές Πληροφορίας:

1. <https://humanorigins.si.edu/human-characteristics/language-symbols>
2. Johnsen, S., & Lohmann, K. J. (2008). Magnetoreception in animals. *Nature*, 451(7178), 978-980.
3. Γ. Καραγιαννίδης και Κ. Παππή Τηλεπικοινωνιακά Συστήματα, 4^η Έκδοση 2017 ΕΚΔΟΣΕΙΣ Α.ΤΖΙΟΛΑ & ΥΙΟΙ Α.Ε.
4. Σ. ΘΕΟΔΩΡΙΔΗΣ, Κ. ΜΠΕΡΜΠΕΡΙΔΗΣ, Λ. ΚΟΦΙΔΗΣ, Εισαγωγή στη Θεωρία Σημάτων και Συστημάτων, Τυπωθήτω - Γιώργος Δαρδανός, Αθήνα 2003.
5. Κανάτας Αθανάσιος, Εισαγωγή στις Τηλεπικοινωνίες, 2^η Έκδοση 2017 ΕΚΔΟΣΕΙΣ Α.ΤΖΙΟΛΑ & ΥΙΟΙ Α.Ε.
6. Simon R. Sanders & Alejandro Aragon-Zavala, Κεραίες και Διάδοση για Ασύρματα Συστήματα Επικοινωνιών, 2017 Εκδόσεις Πεδίο & Δημοσθένης Βουγιούκας για την ελληνική γλώσσα σ'όλο τον κόσμο.
7. Κανάτας Αθανάσιος, Πάντος Γεώργιος, ΑΣΥΡΜΑΤΕΣ ΕΠΙΚΟΙΝΩΝΙΕΣ, 2^η έκδοση 2017 – Α. Παπασωτηρίου & ΣΙΑ ΙΚΕ.
8. Tse, David and Pramod Viswanath. "Fundamentals of Wireless Communication." (2005).
9. Proakis, J.G. and Salehi, M. (2001) Digital Communications. 5th Edition 2008, McGraw-Hill, New York.
10. Simon S. Haykin Digital Communication Systems Wiley (2013).
11. Sackinger, E.. "Analysis and Design of Transimpedance Amplifiers for Optical Receivers." (2017).
12. Goldsmith A. Equalization. In: *Wireless Communications*. Cambridge University Press; 2005:351-373.
13. A. G. Reza, M. Troncoso-Costas, C. Browning, F. J. Diaz-Otero and L. P. Barry, "Single-Lane 54-Gbit/s PAM-4/8 Signal Transmissions Using 10G-Class Directly Modulated Lasers Enabled by Low-Complexity Nonlinear Digital Equalization," in *IEEE Photonics Journal*, vol. 14, no. 3, pp. 1-9, June 2022, Art no. 7226909, doi: 10.1109/JPHOT.2022.3168683.
14. Nelles, Oliver. (2001). Nonlinear system identification. From classical approaches to neural networks and fuzzy models. 10.1007/978-3-662-04323-3.
15. Nonlinear System Theory: The Volterra/Wiener Approach (1983). By WILSON J. RUGH.
16. Georgouli, A. (2015). *Artificial Intelligence*. Kallipos, Open Academic Editions.
17. Αργυράκης, Π. (2001). Νευρωνικά Δίκτυα και Εφαρμογές. Ελληνικό Ανοικτό Πανεπιστήμιο.
18. Michael A. Nielsen, "Neural Networks and Deep Learning", Determination Press, 2015.
19. Goodfellow, Ian. "Deep learning." (2016).
20. C. C. Aggarwal, Neural Networks and Deep Learning, Springer Nature Switzerland AG 2023, https://doi.org/10.1007/978-3-031-29642-0_8
21. Herbert Jaeger (2007) Echo state network. Scholarpedia, 2(9):2330.
22. Lukoševičius, M., & Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.*, 3, 127-149.
23. Jaeger, Herbert. (2001). The "echo state" approach to analysing and training recurrent neural networks-with an erratum note'. Bonn, Germany: German National Research Center for Information Technology GMD Technical Report. 148.
24. Schrauwen, B., Verstraeten, D., & Campenhout, J.M. (2007). An overview of reservoir computing: theory, applications and implementations. *The European Symposium on Artificial Neural Networks*.

25. Jaeger, Herbert. "A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the " echo state network " approach - Semantic Scholar." (2005).
26. Kumar, Prajnesh & Jin, Mingwei & Bu, Ting & Kumar (Gangwar), Santosh & Huang, Yu-Ping. (2020). Efficient Reservoir Computing using FPGA and Electro-optic Modulation. JM6B.20. 10.1364/FIO.2020.JM6B.20.
27. Tanaka, G., Nakane, R., & Hirose, A. (2019). Echo State Networks Composed of Units with Time-Varying Nonlinearity. *Aust. J. Intell. Inf. Process. Syst.*, 17, 34-39.
28. Jaeger, H. (2002). Adaptive Nonlinear System Identification with Echo State Networks. *Neural Information Processing Systems*.

Σύνδεσμοι Εικόνων:

Εικόνα 1:

https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEgqhsjJ8hCir86ssAhatXiAK1uZ4UTmPa5pnJPmQ_eGA63Fq4MmlEoiyLQm8500PecdGOXWPQbgYcQLq4hiv3wjKDKMZuU1CletIL6HPjmra9TmaP70yU-K3WpQ1hYJzlFNQcjYPrG8IE/s1600/Shannon+weaver+by+Michelle.png

Εικόνα 2:

<https://headendinfo.com/32qam-64qam-128qam-256qam/>

Εικόνα 3:

<https://www.etechnog.com/2022/12/how-antenna-works-transmit-or-receive.html>

Εικόνα 5:

https://blog.maurymw.com/hs-fs/hubfs/Blog/03_gaussian3.png?width=750&name=03_gaussian3.png

Εικόνα 6:

<https://img1.daumcdn.net/thumb/R1280x0/?scode=mtistory2&fname=https%3A%2F%2Fblog.kakaocdn.net%2Fdn%2FAaUpL%2FbtqP12mpINw%2FmnMNIpeKdkhBoF7cdKoNAK%2Fimg.png>

Εικόνα 9:

<https://img1.daumcdn.net/thumb/R1280x0/?scode=mtistory2&fname=https%3A%2F%2Fblog.kakaocdn.net%2Fdn%2FcXeFNW%2FbtqPTDuVmWw%2Ftu2KmLxmAyxHxHLvWDQ2d1%2Fimg.png>

Εικόνα 11:

<https://www.researchgate.net/profile/Babu-Sena-Paul-2/publication/254836151/figure/fig8/AS:668852077740041@1536478170950/The-plot-of-Nakagami-m-distribution-for-different-values-of-m.png>

Εικόνα 16:

<https://www.researchgate.net/publication/360056871/figure/fig1/AS:11431281095933831@1668053688050/Volterra-nonlinear-equalizer-VNLE-with-M-linear-taps-and-L-nonlinear-taps-with-h.jpg>

Εικόνα 19:

<https://www.geeksforgeeks.org/introduction-convolution-neural-network/>

Εικόνα 20:

<https://pub.towardsai.net/whirlwind-tour-of-rnns-a11effb7808f>

Εικόνα 21:

<https://developer.nvidia.com/discover/lstm>

Εικόνα 22:

<https://towardsdatascience.com/a-brief-introduction-to-recurrent-neural-networks-638f64a61ff4>

Εικόνα 23:

https://reservoirpy.readthedocs.io/en/latest/user_guide/quickstart.html

Παράρτημα Κώδικα I:

```
% A sample script for generating training and testing data;
% training and testing an ESN on a NARMA time series prediction task.

% Version 1.0, April 30, 2006
% Copyright: Fraunhofer IAIS 2006 / Patent pending
% Revision 1, H. Jaeger, Feb 23, 2007
% Revision 2, H. Jaeger, Aug 17, 2007
close all;clear all;

sequenceLength = 10000;
disp('Generating data .....');
disp(sprintf('Sequence Length %g', sequenceLength ));

% Sampling rate of the channel
channelRate = 3e9; % 3 GHz
%[inputSequence outputSequence] = generate_NARMA_sequence(sequenceLength , systemOrder) ;
[inputSequence1, outputSequence] = myChannel(sequenceLength,channelRate) ;

samplesPerSymbol = 10;
SamplingRate = channelRate;

[inputSequence2] = pre_processing(samplesPerSymbol,inputSequence1, channelRate,
SamplingRate,sequenceLength);

inputSequence=[ones(length(inputSequence2), 1) inputSequence2];

%%% split the data into train and test

train_fraction = 0.5 ; % use 50% in training and 50% in testing
[trainInputSequence, testInputSequence] = ...
    split_train_test(inputSequence,train_fraction);
[trainOutputSequence,testOutputSequence] = ...
    split_train_test(outputSequence,train_fraction);

%%% generate an esn
nInputUnits = samplesPerSymbol+1;

nInternalUnits = 10;
nOutputUnits = 1;

% Create input scaling vector with (samples_per_symbol + 1) 0.1's
inputScalingVector = 0.1 * ones(nInputUnits, 1);
% Create input shift vector with zeros in the same amount as samples_per_symbol + 1
inputShiftVector = zeros(nInputUnits, 1);

esn = generate_esn(nInputUnits, nInternalUnits, nOutputUnits, ...
    'spectralRadius', 0.9, 'inputScaling', inputScalingVector, 'inputShift', inputShiftVector,
    ...
    'teacherScaling', 0.3, 'teacherShift', -0.2, 'feedbackScaling', 0, ...
    'type', 'plain_esn','methodWeightCompute','pseudoinverse');

disp(esn.methodWeightCompute);
```

```

%%% VARIANTS YOU MAY WISH TO TRY OUT
% (Comment out the above "esn = ...", comment in one of the variants
% below)

% % Use a leaky integrator ESN
% esn = generate_esn(nInputUnits, nInternalUnits, nOutputUnits, ...
%     'spectralRadius',0.5,'inputScaling',[0.1;0.1],'inputShift',[0;0], ...
%     'teacherScaling',[0.3],'teacherShift',[-0.2],'feedbackScaling', 0, ...
%     'type', 'leaky_esn');
%
% % Use a time-warping invariant ESN (makes little sense here, just for
% % demo's sake)
% esn = generate_esn(nInputUnits, nInternalUnits, nOutputUnits, ...
%     'spectralRadius',0.5,'inputScaling',[0.1;0.1],'inputShift',[0;0], ...
%     'teacherScaling',[0.3],'teacherShift',[-0.2],'feedbackScaling', 0, ...
%     'type', 'twi_esn');

% % Do online RLS learning instead of batch learning.
% esn = generate_esn(nInputUnits, nInternalUnits, nOutputUnits, ...
%     'spectralRadius',0.4,'inputScaling',[0.1;0.5],'inputShift',[0;1], ...
%     'teacherScaling',[0.3],'teacherShift',[-0.2],'feedbackScaling',0, ...
%     'learningMode', 'online', 'RLS_lambda',0.9999995, 'RLS_delta',0.000001, ...
%     'noiseLevel', 0.00000000) ;

esn.internalWeights = esn.spectralRadius * esn.internalWeights_UnitSR;

%%% train the ESN
nForgetPoints = 0;
[trainedEsn stateMatrix] = ...
    train_esn(trainInputSequence, trainOutputSequence, esn, nForgetPoints) ;

%%% save the trained ESN
% save_esn(trainedEsn, 'esn_narma_demo_1');

%%% plot the internal states of 4 units
% nPoints = 200 ;
% plot_states(stateMatrix,[1 2 3 4], nPoints, 1, 'traces of first 4 reservoir units') ;

% compute the output of the trained ESN on the training and testing data,
% discarding the first nForgetPoints of each
%nForgetPoints = 0 ;
predictedTrainOutput = test_esn(trainInputSequence, trainedEsn, nForgetPoints);
predictedTestOutput =test_esn(testInputSequence, trainedEsn, nForgetPoints);

%create input-output plots
nPlotPoints = 200 ;
plot_sequence(trainOutputSequence(nForgetPoints+1:end,:), predictedTrainOutput, nPlotPoints,...
    'training: teacher sequence (red) vs predicted sequence (blue)');
plot_sequence(testOutputSequence(nForgetPoints+1:end,:), predictedTestOutput, nPlotPoints, ...
    'testing: teacher sequence (red) vs predicted sequence (blue)') ;

%%%compute NRMSE training error
trainError = compute_NRMSE(predictedTrainOutput, trainOutputSequence);
disp(sprintf('train NRMSE = %s', num2str(trainError)))

%%%compute NRMSE testing error
testError = compute_NRMSE(predictedTestOutput, testOutputSequence);
disp(sprintf('test NRMSE = %s', num2str(testError)))

M=4;
% Trim the testOutputSequence to match the length of predictedTestOutput
testOutputSequence = testOutputSequence(nForgetPoints+1:end,:);
testOutputSequence = testOutputSequence(end-length(predictedTestOutput)+1:end);

```

```
% Now demodulate and calculate BER
[a] = pamdemod(predictedTestOutput, 4);
[b] = pamdemod(testOutputSequence, 4);

% Ensure a and b are the same length
minLength = min(length(a), length(b));
a = a(1:minLength);
b = b(1:minLength);

[~, berTest] = biterr(a, b);

% Similar fix for the training sequence
trainOutputSequence = trainOutputSequence(nForgetPoints+1:end,:);
trainOutputSequence = trainOutputSequence(end-length(predictedTrainOutput)+1:end);

[c] = pamdemod(predictedTrainOutput, 4);
[d] = pamdemod(trainOutputSequence, 4);

% Ensure c and d are the same length
minLengthTrain = min(length(c), length(d));
c = c(1:minLengthTrain);
d = d(1:minLengthTrain);

[~, berTrain] = biterr(c, d);

% Display BER
fprintf('Bit Error Rate (BER) for Training: %.5f\n', berTrain);
fprintf('Bit Error Rate (BER) for Testing: %.5f\n', berTest);

% Plotting remains the same
testInputSequence = testInputSequence(nForgetPoints+1:end,:);
figure;

subplot(4, 1, 4);
plot(predictedTestOutput, 'Color', 'b'); % Plot in blue
title('predictedTestOutput');
xlabel('Sample Index');
ylabel('Amplitude');

subplot(4, 1, 3);
plot(testOutputSequence, 'Color', 'b'); % Plot in blue
title('testOutputSequence');
xlabel('Sample Index');
ylabel('Amplitude');

subplot(4, 1, 2);
plot(predictedTrainOutput, 'Color', 'r'); % Plot in red
title('predictedTrainOutput');
xlabel('Sample Index');
ylabel('Amplitude');

subplot(4, 1, 1);
plot(trainOutputSequence, 'Color', 'r'); % Plot in red
title('trainOutputSequence');
xlabel('Sample Index');
ylabel('Amplitude');
```

Παράρτημα Κώδικα II:

```
function [inputSequence1, outputSequence] = myChannel(sequenceLength, channelRate)
% Number of paths
numPaths = 4;

% Initialize pathDelays array
pathDelays = zeros(1, numPaths, 'double');

% Populate pathDelays array using a for loop
for i = 0:numPaths-1
    pathDelays(i+1) = i * (1/channelRate); % Indexing starts from 1 in MATLAB
end

averagePathGains = [-1 -5 -10 -20]; % Path gains

% Creating a Rayleigh channel object
Rayleigh = comm.RayleighChannel(...
    'SampleRate', 2*channelRate, ...
    'PathDelays', pathDelays, ...
    'AveragePathGains', averagePathGains, ...
    'NormalizePathGains', true, ...
    'PathGainsOutputPort', true);
% 'RandomStream', 'mt19937ar with seed', ...
% 'Seed', seed, ...

% Modulation order for PAM
M = 4;
% Generating random PAM-modulated sequence
random_values = randi([0 M-1], sequenceLength, 1);
pam_values = pammod(random_values, M, 0, 'gray'); % PAM modulation

% From BaseBand to Bandpass
t = (0:length(pam_values)-1).' / channelRate;
carrier = cos(2*pi*channelRate*t);
bandpass_signal = real(pam_values) .* carrier;

% Passing the signal through the Rayleigh channel
[received_signal, ~] = Rayleigh(bandpass_signal);

% Interpolation
interpol_factor = 10;
received_signal_interpolated = interp(received_signal, interpol_factor);
% % Adding AWGN
SNR_dB = 20; % Signal-to-Noise Ratio (SNR) in dB
noisy_signal = awgn(received_signal_interpolated, SNR_dB, 'measured');

% Creating inputSequence (real part of noisy signal after interpolation)
inputSequence1 = real(noisy_signal);

% Creating outputSequence (original PAM-modulated symbols)
outputSequence = pam_values;

% Plotting the results
figure;

% Plot PAM-modulated sequence (original symbols)
subplot(4,1,1);
stem(real(pam_values(1:100))); % Plot first 100 samples for clarity
title('PAM-Modulated Symbols (pam_values)');
xlabel('Sample Index');
ylabel('Amplitude');

% Plot received signal after Rayleigh channel
subplot(4,1,2);
plot(real(received_signal(1:100))); % Plot first 100 samples for clarity
title('Received Signal (received_signal)');
xlabel('Sample Index');
ylabel('Amplitude');

% Plot interpolated received signal
subplot(4,1,3);
plot(real(received_signal_interpolated(1:1000))); % Plot first 100 samples for clarity
title('Interpolated Received Signal (received_signal_interpolated)');
xlabel('Sample Index');
ylabel('Amplitude');

% Plot noisy signal after adding AWGN
subplot(4,1,4);
plot(real(noisy_signal(1:1000))); % Plot first 100 samples for clarity
title('Noisy Signal (noisy_signal)');
xlabel('Sample Index');
ylabel('Amplitude');
end
```

Παράρτημα Κώδικα III:

```
function [outputSymbolMatrix] = pre_processing(samplesPerSymbol, receivedSignal, symbolRate, SamplingRate, numSymbols)
% Preparing the dataset for tapping method
% pre_processing - Processes a received signal for further symbol-level analysis.
% This function resamples the received signal to match the desired symbol rate and sampling rate,
% normalizes the signal, and organizes it into a matrix where each row corresponds to a symbol's sample sequence.
%
% Inputs:
% - samplesPerSymbol: Number of samples per symbol in the output.
% - receivedSignal: The input received signal as a 1D array or vector.
% - symbolRate: The rate at which symbols are transmitted (in symbols per second).
% - SamplingRate: The actual sampling rate of the system (in samples per second).
% - numSymbols: The number of symbols in the received signal sequence.
%
% Outputs:
% - outputSymbolMatrix: A matrix where each row corresponds to a symbol, and each column represents samples
%                       from that symbol. The size of the matrix is [numSymbols x samplesPerSymbol].

% Number of resampling points per symbol
% Used to upsample the received signal, creating a finer granularity before resampling.
pointsPerSymbol = 10;

% Resample the received signal to match the target sampling rate.
% This step increases the sampling rate to ensure the signal aligns with the desired rate for further processing.
resampledSignal = resample(receivedSignal', SamplingRate, pointsPerSymbol * symbolRate);

% Calculate the target sampling rate after downsampling.
% The target sampling rate corresponds to the number of samples per symbol multiplied by the symbol rate.
targetSamplingRate = samplesPerSymbol * symbolRate;

% Resample the already resampled signal to match the target sampling rate.
% This ensures that the signal contains the desired number of samples per symbol for further processing.
resampledSignal = resample(resampledSignal, targetSamplingRate, SamplingRate);

% Normalize the resampled signal.
% This adjusts the signal's amplitude levels to a consistent range, making it more suitable for subsequent stages of
processing.
normalizedSignal = normalize(resampledSignal);

% Initialize a matrix to store samples for each symbol.
% The matrix has dimensions [samplesPerSymbol x numSymbols], where each column represents samples from one symbol.
symbolSamplesMatrix = zeros(samplesPerSymbol, numSymbols);

% Loop through the received signal to extract samples for each symbol.
for symbolIndex = 1:numSymbols
% Calculate the start index for the current symbol's samples in the normalized signal.
startIdx = 1 + (symbolIndex - 1) * samplesPerSymbol;
% Calculate the end index for the current symbol's samples.
endIdx = symbolIndex * samplesPerSymbol;
% Extract and store the samples corresponding to the current symbol in the matrix.
symbolSamplesMatrix(:, symbolIndex) = normalizedSignal(startIdx:endIdx);
end

% Transpose the matrix so that each row represents a symbol's sample sequence.
% In the output, rows represent symbols and columns represent samples within those symbols.
outputSymbolMatrix = symbolSamplesMatrix';
end
```