

Πολυτεχνείο Κρήτης  
Τμήμα Ηλεκτρονικών Μηχανικών  
& Μηχανικών Υπολογιστών  
Τομέας Τηλεπικοινωνιών



ΘΕΜΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

---

## **LMS, RLS & CMA Equalization in Wireless FTP Radio link over analog FM**

---

Χρήστος Ν. Κοκκινάκης  
Christos N. Kokkinakis

Ιούλιος 2005



---

## ΣΥΝΟΨΗ

---

Όλα τα ψηφιακά συστήματα επικοινωνίας υποφέρουν από το φαινόμενο της διασυμβολικής παρεμβολής (InterSymbol Interference - ISI), δηλαδή της αλληλοκάλυψης γειτονικών συμβόλων εξαιτίας μη ιδανικοτήτων του καναλιού. Το αντικείμενο του ισοσταθμιστή (equalizer) είναι να εκτιμήσει το κανάλι και να σχεδιάσει ένα φίλτρο που θα εφαρμοστεί στο λαμβανόμενο σήμα και θα αναιρέσει την επίδραση του καναλιού.

Οι αλγόριθμοι LMS και RLS χρησιμοποιούν training sequence για την εκτίμηση του καναλιού, ενώ ο CMA χρησιμοποιεί μόνο κάποιες γνωστές ιδιότητες του λαμβανόμενου σήματος. Οι τρεις αλγόριθμοι εφαρμόζονται πάνω στο πειραματικό σύστημα ασύρματης FTP επικοινωνίας “Homebrew WFTP” και παρουσιάζονται τα θεωρητικά πλεονεκτήματα και μειονεκτήματα καθενός για την συγκεκριμένη εφαρμογή.

Στο τέλος γίνεται μια συνολική αξιολόγηση του συστήματος και των επιδόσεων που πέτυχε για διάφορους συνδυασμούς παραμέτρων και χρησιμοποιούμενων τεχνικών επεξεργασίας, ενώ ιδιαίτερη έμφαση δίνεται στην απόδοση των τριών αλγορίθμων ισοστάθμισης πάνω στο σύστημα.

---

# ΠΕΡΙΕΧΟΜΕΝΑ

---

Σύνοψη.....	iii
Περιεχόμενα.....	iv

## 1 Οργάνωση του Project

1.1 Περιγραφή Σκοπού.....	1
1.2 Καθορισμός Στόχων.....	2
1.3 Milestones.....	3
1.4 Thesis Outline.....	11

## 2 Περιγραφή του Συστήματος

2.1 Το Σύστημα Επικοινωνίας.....	12
2.2 Υλοποίηση σε Κώδικα Matlab.....	16
2.3 Διαχωρισμός Έργου.....	18

## 3 Ισοστάθμιση Καναλιού

3.1 Εισαγωγή.....	19
3.2 Προσαρμοστική Ισοστάθμιση.....	22
3.3 Τεχνικές Ισοστάθμισης.....	26
3.4 Προσαρμοστικοί Αλγόριθμοι Ισοστάθμισης.....	26

#### **4 Ο Αλγόριθμος Least Mean Square**

4.1 Η λύση του Wiener. ....	28
4.2 Η μέθοδος του Steepest Descent. ....	29
4.3 Ο αλγόριθμος LMS. ....	30
4.4 Αξιολόγηση του αλγορίθμου LMS. ....	31
4.5 Επιλογή Παραμέτρων του LMS. ....	36

#### **5 Ο Αλγόριθμος Recursive Least Squares**

5.1 Εισαγωγή. ....	42
5.2 Ο αλγόριθμος RLS. ....	43
5.3 Αξιολόγηση του αλγορίθμου RLS. ....	45
5.4 Επιλογή Παραμέτρων του RLS. ....	48

#### **6 Ο Αλγόριθμος Constant Modulus**

6.1 Blind Ισοσταθμιστές. ....	52
6.2 Ο αλγόριθμος CMA. ....	52
6.3 Αξιολόγηση του αλγορίθμου CMA. ....	53
6.4 Επιλογή Παραμέτρων του CMA. ....	55

#### **7 Αξιολόγηση του Συστήματος “Homebrew WFTP”**

7.1 Εξαρτήσεις του Συστήματος WFTP. ....	60
7.2 Προσομοίωση Λειτουργίας. ....	63
7.3 Αξιολόγηση του Συστήματος. ....	65
7.4 Αξιολόγηση των Ισοσταθμιστών. ....	72
7.5 Επιλεγμένα Συστήματα. ....	76

---

# 1

## ΟΡΓΑΝΩΣΗ ΤΟΥ PROJECT

---

### 1.1 Περιγραφή Σκοπού

Αντικειμενικός σκοπός της παρούσας εργασίας ήταν ο σχεδιασμός, η υλοποίηση και η εξαντλητική δοκιμή ενός συστήματος ασύρματης επικοινωνίας μεταξύ δύο υπολογιστών, με σκοπό την ανταλλαγή αρχείων μεταξύ αυτών (FTP) με όσο το δυνατόν μεγαλύτερη αξιοπιστία και ταχύτητα. Η σύνδεση των υπολογιστών θα γινόταν μέσω των θυρών εισόδου και εξόδου των καρτών ήχου διαμέσου πομποδεκτών FM. Η υλοποίηση του όλου εγχειρήματος έγινε από μια ομάδα 4 ατόμων, της οποίας τα μέλη λειτούργησαν από κοινού για μεγάλο μέρος του project, αλλά είχαν και διακριτούς ρόλους όσον αφορά την ανάπτυξη και δοκιμή διαφόρων υποσυστημάτων.

### 1.2 Καθορισμός Στόχων

Η μετάδοση που επιλέχθηκε για την ασύρματη επικοινωνία των δύο υπολογιστών είναι μέσω ενός πομπού στα FM και ενός αντίστοιχου δέκτη. Ο πομπός συνδέεται με τον ένα υπολογιστή (PC) μέσω της θύρας audio out της κάρτας ήχου ενώ ο δέκτης με την αντίστοιχη θύρα audio in της δικής του κάρτας ήχου. Η επιλογή της κάρτας ήχου για τη διασύνδεση του υπολογιστή με τον πομποδέκτη έγινε προκειμένου να αποφευχθεί το πρόσθετο κόστος του εξοπλισμού ασύρματης δικτύωσης, καθώς κάθε υπολογιστής διαθέτει μια κάρτα ήχου. Ο πομπός διαμορφώνει το φέρον με την ακουστική κυματομορφή που μεταφέρει την πληροφορία και από εκεί εκπέμπεται και λαμβάνεται από τον δέκτη. Ο δέκτης με τη σειρά του αποδιαμορφώνει τα FM και από μια θύρα εξόδου που διαθέτει στέλνει στον δεύτερο υπολογιστή πάλι μια ακολουθία από '0' και '1', η οποία ιδανικά θα ήταν αυτή που εστάλη από τον πομπό.

Το σύστημα που αναπτύχθηκε φαίνεται στην φωτογραφία που ακολουθεί:



Εικόνα 1.1

Οι κύριοι στόχοι για τη λειτουργία αυτού του συστήματος ήταν:

- Αξιόπιστη λειτουργία (bit error rate της τάξεως του  $10^{-6}$ )
- Μεταφορά οποιουδήποτε είδους δεδομένων
- Επίτευξη του μεγαλύτερου δυνατού ρυθμού μετάδοσης.
- Λειτουργία ανοικτού – κλειστού βρόγχου (open-closed loop), δηλαδή τόσο δυνατότητα μετάδοσης ενός αρχείου με αποστολή πακέτων, όπου γίνεται ανίχνευση λαθών και επαναμετάδοση έως την επίτευξη μηδενικού bit error rate (closed loop), όσο και αποστολή των πακέτων ανεξαρτήτως λαθών (open loop).

Σαν επιπλέον γενική κατεύθυνση επιλέξαμε την όσο το δυνατόν απλούστερη σχεδίαση των υποσυστημάτων και τον περιορισμό του κόστους στα απολύτως

αναγκαία. Λόγω της ιδιαίτερης φύσης του υλικού (hardware) που χρησιμοποιήσαμε, το οποίο δεν έχει σχεδιαστεί για τη συγκεκριμένη λειτουργία (βλ κάρτες ήχου και ραδιοφωνικός δέκτης για μετάδοση δεδομένων, Matlab για το στήσιμο real time εφαρμογής κα), αντιμετωπίσαμε από την αρχή πλήθος προβλημάτων που άλλαζαν συχνά τις κατευθύνσεις του project.

### **1.3 Milestones**

Προκειμένου να οργανώσουμε το project και να αξιολογούμε συνεχώς την πρόοδό του, καθορίσαμε milestones, δηλαδή υπο-στόχους που έπρεπε να πετύχουμε σειριακά και σε συγκεκριμένο χρονικό διάστημα προκειμένου σταδιακά να πετύχουμε τον τελικό στόχο που είναι η λειτουργία του συστήματος “Homebrew WFTP” που περιγράφηκε παραπάνω.

Τα 6 milestones στα οποία χωρίστηκε το project τελικά είναι τα παρακάτω:

1. Προετοιμασία και δοκιμαστικές εκπομπές
2. Μετάδοση αδιαμόρφωτης πληροφορίας και λήψη από PC-δέκτη
3. Διαμόρφωση μεταδιδόμενου σήματος και equalization
4. Κωδικοποίηση μεταδιδόμενου σήματος
5. Επαναμετάδοση πακέτων (ARQ)
6. Βελτιστοποίηση – Documentation

Η παραπάνω λίστα προέκυψε βέβαια ύστερα από συνεχείς μεταβολές και ενημερώσεις κατά τη διάρκεια της εκτέλεσης καθώς από τη μελέτη προέκυπταν νέα δεδομένα και προβλήματα.

#### **1.3.1 Προετοιμασία και δοκιμαστικές εκπομπές**

Στο πρώτο στάδιο έγινε η προετοιμασία του συστήματος μετάδοσης και οι πρώτες αναγνωριστικές επαφές με τον εξοπλισμό. Συγκεκριμένα: συλλογή του απαιτούμενου hardware, εγκατάσταση software στους υπολογιστές, συναρμολόγηση πομπού και σύνδεση με τον υπολογιστή και τέλος εκπομπή σήματος από τον ένα υπολογιστή και λήψη από το δέκτη (τρανζίστορ) με τη



μορφή ηχητικού σήματος. Το στάδιο αυτό ήταν και το πιο σύντομο από όλα καθώς μετά τη συλλογή του εξοπλισμού το μόνο που έμενε ήταν μια απλή ραδιοφωνική μετάδοση μέσω υπολογιστή, μια διαδικασία που περιλάμβανε βήματα με τα οποία ήταν εξοικειωμένα τα μέλη της ομάδας (Matlab, συγκώληση πομπού κτλ.) Το στάδιο αυτό περιλάμβανε την ενασχόλησή μας με τα εξής θέματα:

### **A. Συλλογή του απαιτούμενου εξοπλισμού hardware**

Από το σχήμα 1.1 προκύπτουν οι εξής ανάγκες σε hardware οι οποίες συλλέχθηκαν πριν την έναρξη του project:

- 1 desktop PC που χρησιμοποιείται σαν πομπός του αρχείου,
- 1 laptop PC που χρησιμοποιείται σαν δέκτης του αρχείου,
- 1 πομπός FM ισχύος 1 Watt,
- 1 scanner (δέκτης FM),
- 1 διπολική κεραία,
- Βάση στήριξης της κεραίας
- 1 απλό καλώδιο από τον πομπό στην κεραία,
- 2 καλώδια RCA από τον desktop στον πομπό και από τον δέκτη στο laptop,
- 1 καλώδιο LAN,
- Τροφοδοτικό 4,5V για την τροφοδοσία του κυκλώματος του πομπού,
- Μπαταρίες DC1,5V ως εναλλακτική μορφή τροφοδοσίας του πομπού αλλά και σαν κύρια πηγή τροφοδοσίας του δέκτη.

Σαν υπολογιστής-πομπός χρησιμοποιήθηκε ένας desktop υπολογιστής του εργαστηρίου Τηλεπικοινωνιών του Πολυτεχνείου Κρήτης προκειμένου να είναι προσβάσιμος από όλα τα μέλη της ομάδας. Αντίθετα σαν υπολογιστής-δέκτης του αρχείου χρησιμοποιήθηκε ένας laptop, προκειμένου να εκμεταλλευτούμε τη δυνατότητα ελέγχου της λήψης σε διάφορες αποστάσεις και με διαφορετικές συνθήκες. Και οι δύο υπολογιστές ήταν εφοδιασμένοι με την τελευταία έκδοση του Matlab, την 7.0, πάνω στην οποία χτίστηκε όλο το σύστημα και έγινε ο έλεγχος της λειτουργίας του. Αξίζει να σημειωθεί ότι το σύστημα που αναπτύχθηκε μπορεί να λειτουργήσει σε οποιοδήποτε ζεύγος

υπολογιστών, αρκεί αυτό να είναι εφοδιασμένο με την έκδοση 7.0 της Matlab. Η χρήση της τελευταίας έκδοσης προσέφερε το πλεονέκτημα ενός πλούσιου πακέτου συναρτήσεων τηλεπικοινωνιακών συστημάτων (Communications Toolbox) που μας γλίτωσε από χρονοβόρες κατασκευές τετριμμένων συναρτήσεων.

Ο πομπός είναι ισχύος 1 Watt, έτσι ώστε να μην υπόκειται στους περιορισμούς λειτουργίας ραδιοφωνικών σταθμών και να μην απαιτεί άδεια και συνδέεται με μια διπολική κεραία. Η συχνότητα εκπομπής είναι ρυθμιζόμενη επιτρέποντας έτσι την αποφυγή επικάλυψης με άλλη πηγή (π.χ ισχυρός ραδιοφωνικός σταθμός). Για δέκτης χρησιμοποιήθηκε ένα απλό ραδιόφωνο με έξοδο για ακουστικά αλλά αργότερα αντικαταστάθηκε με scanner με line out ήχου για ακριβέστερη λήψη και καλύτερα αποτελέσματα.

Το καλώδιο LAN ενώνει τους δύο υπολογιστές προκειμένου να υπάρχει αξιόπιστη επικοινωνία για το σύστημα επαναμεταδόσεων (ARQ) που απαιτείται. Δηλαδή, κάθε φορά που λαμβάνεται ένα πακέτο σωστά από τον δέκτη, στέλνεται μέσω back-to-back UDP μια ειδοποίηση στον πομπό, ότι μπορεί να συνεχίσει στη μετάδοση του επόμενου πακέτου.

## **B. Εγκατάσταση Software**

Το software για την ανάπτυξη και προσομοίωση της λειτουργίας του συστήματος είναι το Matlab της Mathworks. Αρχικά χρησιμοποιήθηκε η έκδοση 6.5, αργότερα όμως αναβαθμίστηκε στην έκδοση 7.0 καθώς η τελευταία περιέχει πλήθος νέων χαρακτηριστικών που ήταν χρήσιμα για την ανάπτυξη του συστήματος (νέες συναρτήσεις στο Communications Toolbox, νέο interface κτλ.). Απαιτήθηκε εγκατάσταση του προγράμματος αυτού τόσο στον Laptop-πομπό όσο και στον PC-δέκτη μια και οι ρόλοι είναι αντιστρέψιμοι (π.χ για δοκιμή μεγάλης απόστασης πομπού-δέκτη ο laptop χρησιμοποιείται σαν δέκτης προκειμένου να εκμεταλλευτούμε την ευελιξία που προσφέρει η φορητότητα).

## **Γ. Συναρμολόγηση Πομπού – Δέκτη**

Ο πομπός του 1Watt αγοράστηκε σε κομμάτια και συναρμολογήθηκε από τα μέλη της ομάδας στο Εργαστήριο Τηλεπικοινωνιών του Πολυτεχνείου Κρήτης. Τα χαρακτηριστικά λειτουργίας του συνοψίζονται στον παρακάτω πίνακα:

Τάση Τροφοδοσίας	DC 4,5V
Εύρος Συχνοτήτων	88 – 108 Mhz
Ισχύς Εξόδου	1 W
Τύπος Κεραίας	Διπολική (half wave)
Πίνακας 1.1 – Χαρακτηριστικά λειτουργίας πομπού FM	

Το κύκλωμα του πομπού τοποθετήθηκε σε ένα μεταλλικό κουτί με υποδοχές για τα τρία καλώδια (κεραίας, τροφοδοσίας, line-out) που χρειάζονταν, καθώς παρεμβολές στο χώρο επηρέαζαν την λειτουργία του. Ένα καλώδιο σε επαφή με το κουτί αυτό οδηγούνταν στην γείωση της εγκατάστασης ρεύματος του Εργαστηρίου Τηλεπικοινωνιών για να αποφευχθούν φαινόμενα ground loops.

#### **Δ. Τροφοδοσία**

Για την τροφοδοσία του κυκλώματος του πομπού χρησιμοποιήθηκε αρχικά τροφοδοτικό με μετασχηματιστή τάσης στα 4,5V για απεριόριστη παροχή ρεύματος. Τελικά λόγω παρεμβολών της τροφοδοσίας στον πομπό (χαρακτηριστικός βόμβος κατά την μετάδοση μουσικής), ο τύπος τροφοδοσίας άλλαξε σε μπαταρίες (3x1,5V) και το πρόβλημα λύθηκε. Έτσι για όλη τη διάρκεια των δοκιμών ανάπτυξης του συστήματος αλλά και των τελικών δοκιμών χρησιμοποιήθηκαν μπαταρίες.

#### **Ε. Εκπομπή αρχείου και λήψη από τρανζίστορ**

Η δοκιμή λειτουργίας του συστήματος αποτέλεσε και την τελευταία ενέργεια πριν από την ολοκλήρωση του πρώτου milestone. Ο πομπός συνδέθηκε με τον υπολογιστή, όπου εκτελέστηκε ένα μουσικό αρχείο με αποτέλεσμα να ξεκινήσουμε μια ραδιοφωνική εκπομπή μουσικής στην προκαθορισμένη συχνότητα. Μετά από κάποιες ρυθμίσεις στον δέκτη και την κάρτα ήχου του υπολογιστή το κομμάτι ελήφθη σχετικά καθαρά από το ραδιοφωνάκι – δέκτη. Τα χαρακτηριστικά λειτουργίας του δέκτη FM φαίνονται στο πίνακα 1.3:

Τάση Τροφοδοσίας	DC3.6V~DC6V
------------------	-------------

Εύρος Συχνοτήτων	0.1 ~ 1299.955 MHz
Σταθερότητα Συχνότητας	$\pm 5\text{PPM}$ ( $-10^{\circ}\text{C} \sim +60^{\circ}\text{C}$ )
Ευαισθησία FM	30 ~ 550 MHz
Συνθ Αντιστ Κεραίας	50 $\Omega$
Πίνακας 1.3 – Χαρακτηριστικά λειτουργίας δέκτη FM	

### 1.3.2 Μετάδοση πληροφορίας και λήψη από PC-δέκτη

Επόμενος στόχος ήταν η λήψη του σήματος εκπομπής από τον δέκτη και η αποθήκευσή του στο PC προς μελλοντική επεξεργασία, δηλαδή μεταφορά αρχείου από PC σε PC. Για να στέλνονται στον υπολογιστή τα δεδομένα που λαμβάνει το ραδιοφωνάκι πρέπει αφενός μεν να συνδέσουμε την έξοδο του πρώτου στη Line in της κάρτας ήχου του δεύτερου και αφετέρου να γίνεται ηχογράφηση από την κάρτα ήχου την ώρα που φθάνουν τα δεδομένα. Με το πέρας του παραπάνω βήματος ήμασταν σε θέση να συγκρίνουμε το λαμβανόμενο σήμα με το μεταδιδόμενο και να βγάλουμε κάποια πρώτα συμπεράσματα σχετικά με τα επίπεδα θορύβου και την συμπεριφορά του καναλιού. Σημειώνεται ότι στο παρόν στάδιο η μόνη διαμόρφωση που είχαμε εισάγει στο σήμα ήταν η FM που κάνει αυτόματα ο πομπός σε επίπεδο hardware και η αντίστοιχη αποδιαμόρφωση του τρανζίστορ.

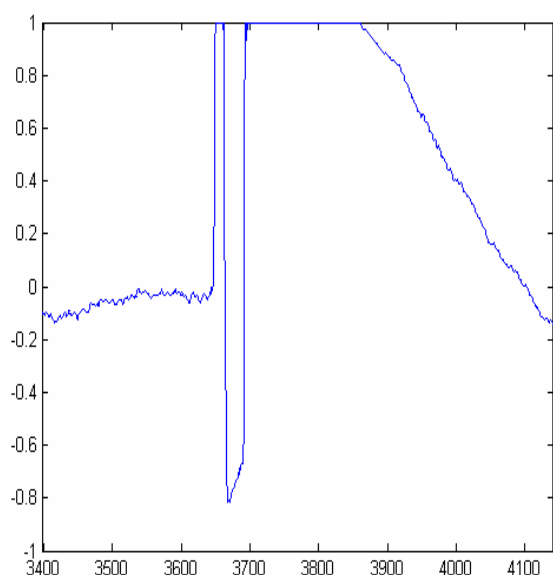
#### A. Οι συναρτήσεις Wavplay() και Wavrecord()

Η συνάρτηση του Matlab που είναι κατάλληλη για την «μεταφορά» των δεδομένων που δέχεται το ραδιοφωνάκι στον υπολογιστή είναι η “wavrecord” η οποία ηχογραφεί την πληροφορία που δέχεται ο υπολογιστής στην κάρτα ήχου με την οποία είναι συνδεδεμένος ο δέκτης (τρανζίστορ). Παίρνει σαν όρισμα το χρόνο και την συχνότητα δειγματοληψίας. Η wavplay() είναι η αντίστοιχη συνάρτηση Matlab που χρησιμοποιείται στον πομπό για την εκτέλεση ενός αρχείου σαν μουσική. Κάθε αρχείο που διαβάζεται στο Matlab, αποθηκεύεται σαν δυαδική πληροφορία σε πίνακα. Η wavrecord() διαβάζει αυτόν τον πίνακα και «παίζει» σαν να είναι μουσική. Έτσι ακόμα και ένα αρχείο κειμένου μπορεί να παιχτεί σαν μουσική και να φτάσει έτσι στην κάρτα

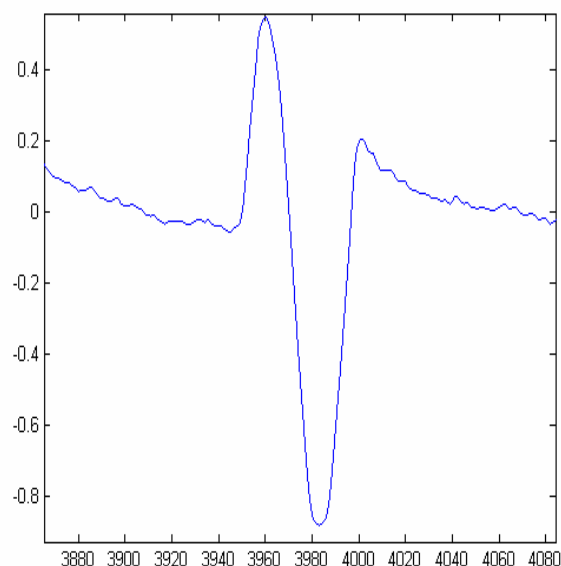
ήχου και από εκεί στον πομπό. Τα ορίσματα που παίρνει είναι τα ίδια με την wavrecord().

## **Β. Ψαλιδισμός Κυματομορφών**

Ένα πρώτο πρόβλημα που αντιμετωπίσαμε στο στάδιο αυτό ήταν η αποκοπή πλάτους που εφαρμόζει η κάρτα ήχου στο σήμα (ψαλιδισμός) που είχε σαν αποτέλεσμα λάθη στη μετάδοση. Ο ψαλιδισμός γίνεται τόσο στον πομπό για το εκπεμπόμενο σήμα όσο και στον δέκτη για το λαμβανόμενο σήμα και οφείλεται στην στάθμη του ήχου στο πάνελ ρυθμίσεων της κάρτας ήχου. Η ρύθμιση αυτής της παραμέτρου είναι ιδιαίτερα σημαντική κατά το setup του συστήματος, ιδιαίτερα όταν οι χρησιμοποιούμενοι υπολογιστές δεν είναι σταθεροί ή όταν αλλάζει η απόσταση μεταξύ πομπού και δέκτη. Η ρύθμιση αυτή γίνεται εμπειρικά και πειραματικά βλέποντας τις κυματομορφές του λαμβανόμενου σήματος. Στις παρακάτω γραφικές φαίνεται μια περίοδος ημιτόνου, όπου η κάρτα ήχου στο μέγιστο της έντασής της ψαλιδίζει το πλάτος (πάνω από 1)(Εικόνα 1.2α) και παραμορφώνει το σήμα και η διόρθωση για το ίδιο σήμα μετά την ρύθμιση του ήχου (Εικόνα 1.2β).



Εικόνα 1.2α



Εικόνα 1.2β

## **Γ. Συγχρονισμός Πομπού-Δέκτη**

Ένα επιπλέον πρόβλημα που αντιμετωπίζουν όλα τα ψηφιακά τηλεπικοινωνιακά συστήματα είναι η ειδοποίηση του δέκτη για την έναρξη της

μετάδοσης. Για την επίλυση αυτού του προβλήματος στο σύστημά μας χρησιμοποιείται *triple hand-shake*. Ο πομπός αφού φορτώσει το αρχείο και είναι έτοιμος προς μετάδοση, στέλνει ένα wake-up signal στον δέκτη, ο οποίος ενεργοποιείται και ξεκινάει την ηχογράφηση από την είσοδο της κάρτας ήχου στέλνοντας παράλληλα μήνυμα στον πομπό ότι ενεργοποιήθηκε και περιμένει πακέτα. Ο πομπός αφού λάβει το μήνυμα αυτό στέλνει διαδοχικά τα πακέτα στα οποία έχει χωρίσει το αρχείο.

### 1.3.3 Διαμόρφωση σήματος και equalization

Πλέον με το link μεταξύ πομπού και δέκτη να έχει επιτευχθεί, είναι δυνατή η εφαρμογή τεχνικών ψηφιακών συστημάτων επικοινωνίας. Πρώτο μέλημα είναι η εξάλειψη της διασυμβολικής παρεμβολής, η οποία αποτελεί και το κυρίως θέμα της παρούσας διπλωματικής και θα αναλυθεί στα επόμενα κεφάλαια, καθώς και η διαμόρφωση του ψηφιακού σήματος με κάποια από τις ευρέως διαδεδομένες τεχνικές (PSK, QAM). Με τις τεχνικές αυτές ο πομπός παύει πλέον να στέλνει μεμονομένα bits, αλλά συνδυάζει blocks από αυτά προκειμένου να στείλει ένα σύμβολο αυξάνοντας έτσι την ανθεκτικότητα του συστήματος επικοινωνίας απέναντι σε τυχαία γεγονότα όπως ο θόρυβος που μπορεί να «χτυπάει» τα μεμονωμένα bit. Ένα θέμα που σχετίζεται με τον συγχρονισμό της προηγούμενης παραγράφου είναι το πως ο αποδιαμορφωτής θα ξέρει ποια είναι η αρχή του κάθε συμβόλου. Αυτή η παράμετρος είναι πολύ σημαντική καθώς λάθος επιλογή της αρχής του συμβόλου, έχει σαν αποτέλεσμα τελείως λανθασμένη αποκωδικοποίηση και μεγάλο bit error rate. Η υλοποίηση αυτού γίνεται με χρήση της συνάρτησης correlation sync η οποία ανιχνεύει το σήμα εισόδου και με βάση την συσχέτιση του με ένα γνωστό τμήμα του σήματος (training sequence) βρίσκει την αρχή του λαμβανόμενου σήματος.

### 1.3.4 Κωδικοποίηση μεταδιδόμενου σήματος

Μετά την επίλυση του συγχρονισμού, το σύστημα μπορεί πλέον να θεωρηθεί αξιόπιστο και πετυχαίνει να στείλει πληροφορία αλάνθαστα για λίγα σύμβολα.

Προκειμένου όμως να επιτευχθούν μεγαλύτερες ταχύτητες πρέπει να μεγαλώσει ο αριθμός των bits/σύμβολο (M), δηλαδή ο αστερισμός να έχει περισσότερα σημεία. Κάτι τέτοιο όμως έχει σαν αποτέλεσμα περισσότερα λάθη λόγω θορύβου, αφού μικρότερες μεταβολές στην τιμή ενός bit μπορούν να οδηγήσουν σε λάθος περιοχή απόφασης. Για να αποφευχθούν τέτοια λάθη, εισάγονται κώδικες διόρθωσης λαθών (Error Correcting Codes), οι οποίοι επισυνάπτουν στο μεταδιδόμενο σήμα πρόσθετα bits με βάση τα οποία στο δέκτη είναι δυνατόν να διορθωθούν κάποια λάθη ανά σύμβολο.

### **1.3.5 Επαναμετάδοση πακέτων (ARQ)**

Μέχρι τώρα θεωρήσαμε ότι το σύστημα δουλεύει μόνο σε ανοικτό βρόγχο (open loop), πράγμα που σημαίνει ότι το αρχείο χωρίζεται σε πακέτα και στέλνεται στον δέκτη με ή χωρίς λάθη. Ένα σύστημα επικοινωνίας WFTP θα πρέπει να λειτουργεί αλάνθαστα, δηλαδή θα πρέπει να μην δέχεται πακέτα που περιέχουν λάθη προκειμένου να δουλεύει και σε closed loop. Στην περίπτωση αυτή χρησιμοποιείται σύστημα επαναμετάδοσης πακέτων (ARQ- Automatic Repeat Request), όπου ο πομπός στέλνει αρνητική επιβεβαίωση (NACK) όταν λάβει πακέτο με λάθη, και ζητάει επαναμετάδοση του συγκεκριμένου πακέτου. Το πλεονέκτημα αυτής της υλοποίησης είναι η εγγυημένη αξιοπιστία που μπορεί να κάνει το σύστημα να πετύχει το στόχο που ετέθη (bit error rate  $10^{-6}$ ), σε κόστος της ταχύτητας μετάδοσης λόγω της επαναμετάδοσης του ίδιου πακέτου. Σημειώνεται ότι υλοποιήσαμε τις απαντήσεις του ARQ μέσω Ethernet χρησιμοποιώντας το πρωτόκολλο UDP.

### **1.3.6 Βελτιστοποίηση – Documentation**

Το τελευταίο στάδιο περιελάμβανε τη συγκέντρωση όλων των απαραίτητων υποσυστημάτων του συστήματος η ταυτόχρονη λειτουργία τους κάτω από όλες τις συνθήκες και ρυθμίσεις και η βελτίωση διαφόρων παραμέτρων του συστήματος. Συγκεκριμένα έγιναν τα εξής:

- Μόνωση πομπού σε μεταλλικό κουτί για αποφυγή παρεμβολών (κλωβός Faraday).
- Βελτίωση κωδίκων που καθυστερούσαν το σύστημα, με τη βοήθεια του Profiler του Matlab.
- Κατασκευή GUI (graphical user interface) τόσο για τη λειτουργία του πομπού όσο και για του δέκτη.
- Κοινό setup πομπού και δέκτη (μέσω GUI) στην αρχή της λειτουργίας του συστήματος προκειμένου και οι δύο να γνωρίζουν τις παραμέτρους της μετάδοσης (π.χ τάξη διαμόρφωσης  $M$ , περίοδος  $T$ , συχνότητα δειγματοληψίας  $F_s$ ).
- Ελαχιστοποίηση χρόνου μετάδοσης και αριθμού λαθών με κατάλληλο συνδυασμό των τεχνικών που αναπτύχθηκαν.
- Σύγκριση όλων των δυνατών συνδυασμών συστημάτων και εξαγωγή συμπερασμάτων ως προς αυτά που λειτουργούν.

## 1.4 Thesis Outline

Η παρούσα διπλωματική αποτελεί τμήμα ενός ομαδικού project, που σαν στόχο έχει τον σχεδιασμό και κατασκευή ενός αξιόπιστου συστήματος μετάδοσης και καταπιάνεται με όλα τα θέματα που θίχτηκαν παραπάνω. Η θεματολογία αυτού του τμήματος περιορίζεται στην ισοστάθμιση καναλιού. Στο Κεφάλαιο 2 αναλύεται η δομή του συστήματος επικοινωνίας, τα διάφορα τμήματα και οι επιλογές που έγιναν από όλα τα μέλη της ομάδας για την σύνθεση αυτού του συστήματος. Στο Κεφάλαιο 3 τοποθετείται το πρόβλημα της ισοστάθμισης τόσο διαισθητικά όσο και από πλευράς μαθηματικής μοντελοποίησης. Στα Κεφάλαια 4 έως 6 αναλύονται οι τρεις αλγόριθμοι που χρησιμοποιήθηκαν για να επιλύσουμε το πρόβλημα της ισοστάθμισης, τα πλεονεκτήματα και τα μειονεκτήματα κάθε αλγόριθμου, οι επιλογές στις παραμέτρους τους. Στο Κεφάλαιο 7 επιχειρείται μια αξιολόγηση του συστήματος συνολικά για διάφορες επιλογές παραμέτρων, αλλά και αξιολόγηση των αλγόριθμων ισοστάθμισης όσον αφορά την απόδοσή τους κάτω από διάφορες επιλογές παραμέτρων με βάση όσα προέβλεπε η θεωρία.



---

## 2 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

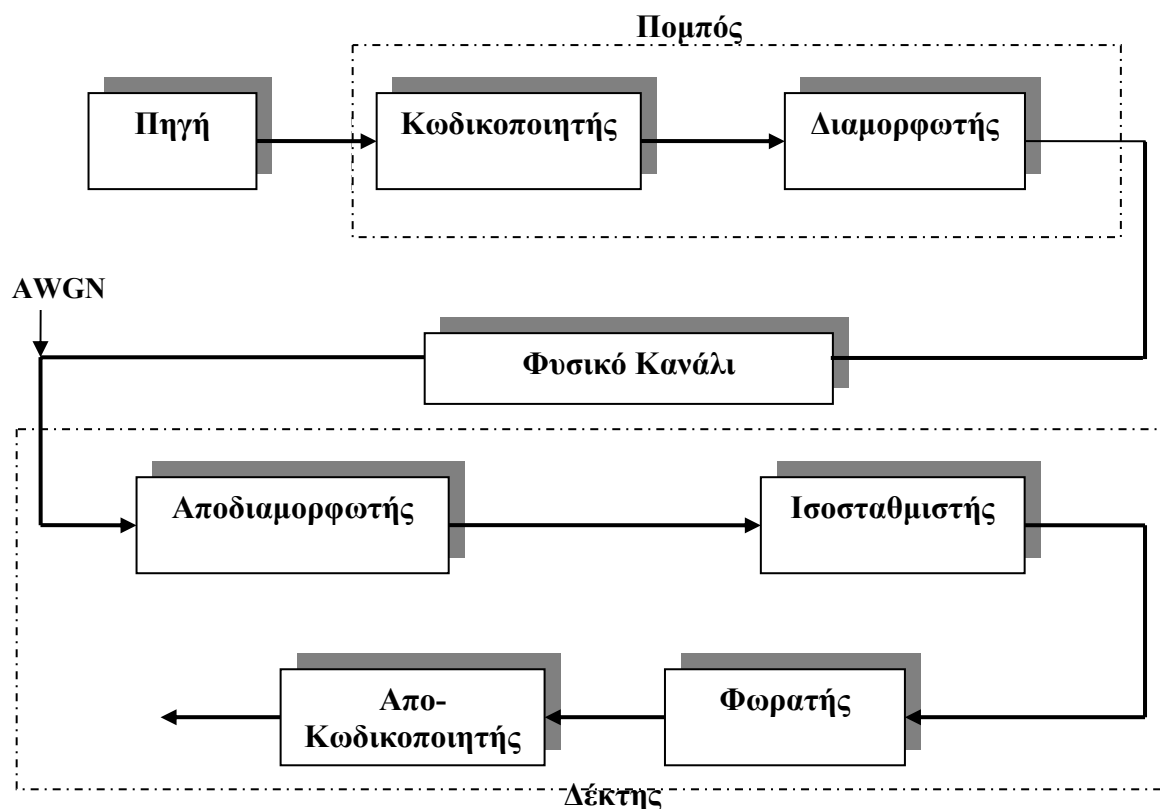
---

Όπως είπαμε, αυτή η εργασία καταπιάνεται με το πρόβλημα της προσαρμοστικής ισοστάθμισης καναλιού (adaptive channel equalization) για ψηφιακά τηλεπικοινωνιακά συστήματα. Προκειμένου να τοποθετηθεί το πρόβλημα αυτό σε σχέση με τη σχεδίαση των ψηφιακών τηλεπικοινωνιακών συστημάτων, είναι αναγκαίο να περιγραφεί το μοντέλο που ακολουθεί το σύστημα που σχεδιάστηκε καθώς και τα επιμέρους τμήματα από τα οποία αποτελείται αυτό.

### 2.1 Το Σύστημα Επικοινωνίας

Τα ψηφιακά τηλεπικοινωνιακά συστήματα, όπως αυτό που αναπτύχθηκε για τις ανάγκες του παρόντος project, σχεδιάζονται προκειμένου να μεταδίδουν πληροφορία, που δημιουργείται σε μια πηγή σε έναν ή περισσότερους προορισμούς σε ψηφιακή μορφή. Για να γίνει αυτό, το προς μετάδοση σήμα περνάει από διάφορα στάδια επεξεργασίας, τόσο στο άκρο του πομπού όσο και σε αυτό του δέκτη. Στο κεφάλαιο αυτό περιγράφεται με συντομία ένα ψηφιακό τηλεπικοινωνιακό σύστημα, καθώς το μοντέλο αυτό χρησιμοποιήθηκε προκειμένου να σχεδιαστεί το παρόν σύστημα ψηφιακής επικοινωνίας. Ταυτόχρονα, μέσα από το μοντέλο αυτό δόθηκε η δυνατότητα να σχεδιαστεί το σύστημα σε αρθρωτή (space-modular) μορφή, επιτρέποντας σε καθένα από τα 4 μέλη της ομάδας να δουλέψει ανεξάρτητα σε κάποιο ξεχωριστό τμήμα, εξειδικεύοντας την εργασία αλλά και επιταχύνοντας το χρόνο ολοκλήρωσής της λόγω παράλληλης εργασίας.

Το Σχήμα 2.1 που ακολουθεί περιγράφει το ψηφιακό τηλεπικοινωνιακό σύστημα που σχεδιάστηκε.



Σχήμα 2.1

### 2.1.1 Παραγωγή αρχείου

Περιλαμβάνει το υποσύστημα δημιουργίας του προς μετάδοση σήματος, που μπορεί γενικά να είναι αναλογικό (όπως σε μια ραδιοφωνική εκπομπή ή τηλεοπτική μετάδοση) ή ψηφιακό. Στην περίπτωση μας, την ψηφιακή, το σήμα είναι μια ακολουθία από bits ή από χαρακτήρες ASCII που δημιουργούνται από τους υπολογιστές. Σε κάθε ψηφιακό σύστημα ένα αναλογικό σήμα πρέπει να μετατραπεί σε ψηφιακό με δειγματοληψία και κβάντιση. Στην περίπτωση της επικοινωνίας μεταξύ υπολογιστών, τα αρχεία είναι ήδη σε ψηφιακή μορφή οπότε μια τέτοιου είδους επεξεργασία δεν είναι αναγκαία, αν και επειδή οι πομποδέκτες είναι αναλογικοί (FM) μια τέτοια διαδικασία θα γίνει στον δέκτη προκειμένου από το αναλογικό σήμα να λάβουμε ψηφιακή πληροφορία την οποία μπορεί να «καταλάβει» ο υπολογιστής. Στο δικό μας σύστημα, το στάδιο αυτό περιλαμβάνει την προετοιμασία (φόρτωση) του προς μετάδοση αρχείου σε μια ακολουθία από bits.

### 2.1.2 Κωδικοποίηση

Για το σύστημα χρησιμοποιείται προαιρετικά κωδικοποίηση, η οποία προσθέτει bits στην πληροφορία με γνωστό τρόπο, και τα οποία χρησιμοποιούνται στο στάδιο του αποκωδικοποιητή του δέκτη για να ανακαλύψουν λάθη στη μετάδοση. Οι κώδικες που χρησιμοποιήθηκαν είναι της οικογένειας Reed Solomon και συνελκτικοί κώδικες. Οι κώδικες Reed Solomon χωρίζουν τα bits του πακέτου σε blocks και από κάθε block μπορούν να διορθώσουν περιορισμένο αριθμό bits. Όμως ο θόρυβος που εμφανίζεται με τη μορφή burst είναι πιθανότερο να επηρεάσει γειτονικά bits που ανήκουν στο ίδιο block, καθιστώντας αδύνατη την διόρθωση από τους κώδικες διόρθωσης λαθών. Γι' αυτό χρησιμοποιείται συνήθως και ένας Interleaver ο οποίος αναλαμβάνει να «ανακατέψει», πάλι με γνωστό τρόπο, τα bits ενός πακέτου, έτσι ώστε bits που ανήκουν στο ίδιο block να μην είναι γειτονικά κατά τη μεταφορά και έτσι να έχουν μικρότερη πιθανότητα να χτυπηθούν από θόρυβο σε ποσοστό τέτοιο που να μην είναι δυνατή η διόρθωσή τους.

### 2.1.3 Διαμόρφωση

Η πληροφορία κωδικοποιημένη από το προηγούμενο στάδιο ή όχι, περνάει στο διαμορφωτή ο οποίος αναλαμβάνει να την μετατρέψει σε κυματομορφές οι οποίες θα «ταξιδέψουν» στο κανάλι. Αυτό γίνεται γιατί στα ψηφιακά τηλεπικοινωνιακά συστήματα υπάρχουν περιορισμοί στο διαθέσιμο εύρος ζώνης που ικανοποιούνται καλύτερα όταν η πληροφορία μεταδίδεται με τη μορφή αυτή. Σημαντικός παράγοντας στη διαμόρφωση, που παίζει ρόλο στην απόδοση αλλά και την ανθεκτικότητα του συστήματος στο θόρυβο, είναι η περίοδος  $T$  που δείχνει πόσα δείγματα απαρτίζουν μια κυματομορφή που μεταφέρει  $M$  bits πληροφορίας. Δοκιμάστηκαν τρεις διαφορετικές διαμορφώσεις: PSK, QAM και PPM. Καθεμιά από τις συναρτήσεις που υλοποιούν τις εν λόγω διαμορφώσεις κατασκευάστηκε από διαφορετικό μέλος της ομάδας οπότε η απόδοσή τους εξαρτάται και από την συγκεκριμένη υλοποίηση, όπως θα φανεί στο Κεφάλαιο 7 της αξιολόγησης του συστήματος. Ο αριθμός των bits/σύμβολο ( $M$ ) που χρησιμοποιήθηκαν ήταν 4, 8, 16 και 32.

### 2.1.4 Κανάλι

Οι διαμορφωμένη πληροφορία (κυματομορφές) ταξιδεύει στο κανάλι και αλλοιώνεται λόγω διασυμβολικής παρεμβολής (ISI – βλ κεφάλαιο 3) και θορύβου. Η διασυμβολική παρεμβολή προκαλείται από την συνέλιξη της απόκρισης του καναλιού με την κυματομορφή του μεταδιδόμενου σήματος, όπως αναλύεται στο Κεφάλαιο 3.2. Ο θόρυβος είναι προσθετικός και μεταβάλλει ανομοιόμορφα την μεταδιδόμενη πληροφορία. Στόχος όλων των τεχνικών που προηγούνται και έπονται αυτού του σταδίου είναι να περιορίσουν τα δύο αυτά φαινόμενα στην πληροφορία.

### 2.1.5 Αποδιαμόρφωση

Στο δέκτη, αυτό που φτάνει είναι οι κυματομορφές που στάλθηκαν από τον πομπό αλλοιωμένες από το κανάλι. Κατά την αποδιαμόρφωση, η συνεχής πληροφορία (κυματομορφές) μετατρέπεται πάλι σε διακριτή (bits). Ο δέκτης ασφαλώς θα πρέπει να γνωρίζει την περίοδο που χρησιμοποιήθηκε αλλά και την αρχή του πρώτου συμβόλου (συγχρονισμός). Μετά από αυτό το στάδιο, η πληροφορία, όντας ψηφιακή, μπορεί να επεξεργαστεί από τα επόμενα στάδια.

### 2.1.6 Ισοστάθμιση

Ο ισοσταθμιστής παίρνει την διακριτή πληροφορία από αποδιαμορφωτή και χρησιμοποιεί έναν από τους αλγορίθμους LMS, RLS και CMA οι οποίοι περιγράφονται στα επόμενα κεφάλαια προκειμένου να ανακατασκευάσει το σήμα περιορίσει την επίδραση του καναλιού. Για να το πετύχει αυτό πρέπει να εκτιμήσει το κανάλι και γι' αυτό χρησιμοποιεί τους παραπάνω αλγόριθμους. Έπειτα δημιουργεί ένα φίλτρο με επίδραση αντίστροφη του καναλιού και «περνάει» σήμα από αυτό. Όσο καλύτερη είναι η εκτίμηση του καναλιού, τόσο καλύτερη είναι η ανακατασκευή του σήματος και τόσα λιγότερα τα λάθη λόγω διασυμβολικής παρεμβολής που θα εμφανιστούν στο τέλος.

### 2.1.7 Φώραση

Ο φωρατής είναι το προτελευταίο στάδιο από το οποίο περνάει η πληροφορία (βλ σχήμα 2.1). Επειδή η επίδραση του θορύβου και του ISI δεν αναιρούνται τελείως από τα προηγούμενα στάδια, είναι αρμοδιότητα του φωρατή να αποφασίσει την αρχική τιμή του κάθε bit. Σχηματικά, ο φωρατής χωρίζει τον δυσδιάστατο αστερισμό (constellation) του σήματος σε  $M$  περιοχές απόφασης και κάθε bit που πέφτει μέσα σε μια περιοχή θεωρείται ότι αντιστοιχεί στον κοντινότερο από τους  $M$  επιτρεπτούς συνδυασμούς bits.

### 2.1.8 Αποκωδικοποίηση

Στο στάδιο της αποκωδικοποίησης, αρχικά ο Interleaver αναλαμβάνει να αναδιατάξει τα bits της εξόδου του προηγούμενου σταδίου στη σωστή σειρά. Έπειτα με βάση την πρόσθετη πληροφορία που εισήχθη από τους κωδικοποιητές εκτιμάται αν έγιναν λάθη κατά τη μετάδοση, τα οποία και διορθώνονται. Πλέον η πληροφορία είναι στη μορφή που δημιουργήθηκε στον πομπό και συγκρίνοντας τα bits του δέκτη με αυτά της αρχικής πληροφορίας, μπορούμε να εξάγουμε το bit error rate με βάση τον τύπο:

$$BER = \frac{\text{Αριθμός\_λάθος\_bits}}{\text{Συνολικό\_αριθμό\_bits\_αρχείου}}$$

## 2.2 Υλοποίηση Κώδικα σε Matlab

Για την υλοποίηση και προσομοίωση του συστήματος χρησιμοποιήθηκε, όπως αναφέρθηκε και στο Κεφάλαιο 1, το πρόγραμμα Matlab 7.0 της Mathworks. Η συγγραφή του κώδικα ακολούθησε την αρθρωτή (modular) δομή που περιγράφηκε στο Σχήμα 2.1. Ο πομπός και ο δέκτης χρησιμοποιούν από μια συνάρτηση η οποία με τη σειρά της καλεί ανάλογα με τα ορίσματά της επιμέρους συναρτήσεις για κάθε τμήμα. Κάθε μέλος της ομάδας ανέλαβε να υλοποιήσει και να δοκιμάσει ένα ξεχωριστό κομμάτι και η σύνθεση του τελικού συστήματος έγινε στο τελευταίο στάδιο του project, οπότε και ξεκίνησε και η προσομοίωση.

Καταρχήν χρησιμοποιείται μια συνάρτηση, η οποία παίρνει μέσω GUI τις παραμέτρους του συστήματος και τις γνωστοποιεί τόσο στον πομπό όσο και στο δέκτη μέσω μιας δομής στο Matlab στην οποία έχουν πρόσβαση και οι δύο. Αυτό είναι απαραίτητο γιατί για ορισμένες παραμέτρους χρειάζεται γνώση από τον πομπό και από το δέκτη (π.χ T για σωστή αποδιαμόρφωση, είδος κωδικοποιητή ώστε να γίνει ανάλογη αποκωδικοποίηση, μέγεθος πακέτου κτλ), ενώ άλλες χρειάζονται μόνο στον δέκτη (π.χ παράμετροι ισοστάθμισης). Μετά από αυτό το στάδιο το σύστημα λειτουργεί με αυτές τις παραμέτρους. Αν χρειαστεί να αλλάξει κάτι από αυτά, πρέπει να ξανατρέξει η συγκεκριμένη συνάρτηση. Παράλληλα ετοιμάζεται και το αρχείο, σπάζοντας σε πακέτα ανάλογα με το μέγεθος που ορίστηκε (από την “packetizer”) ενώ βοηθητικές συναρτήσεις κάνουν padding στα πακέτα. Σημειώνεται ότι ο δέκτης γνωρίζει το padding, που γίνεται με αλγόριθμο, προκειμένου να κάνει κατάλληλο unpadding και να ανακτήσει την αρχική πληροφορία.

Στον δέκτη τρέχει η συνάρτηση “Receiver” η οποία αναλαμβάνει να καθοδηγήσει την λήψη του αρχείου. Από τη στιγμή που ενεργοποιείται η εν λόγω συνάρτηση ο δέκτης ανιχνεύει το UDP για να πιάσει σήμα από τον πομπό για έναρξη μετάδοσης. Μόλις στον πομπό τρέξει η “Transmitter”, η οποία καθοδηγεί την αποστολή, στέλνεται ένα wake-up signal στον δέκτη ο οποίος απαντάει ότι είναι έτοιμος (server mode) και ξεκινάει ηχογράφηση (client mode). Η “Transmitter” μόλις λάβει την απάντηση στέλνει το πρώτο πακέτο (κάνοντας wanplay για χρόνο ανάλογο του μεγέθους του πακέτου συν κάποιο χρόνο ασφαλείας - server mode) και περιμένει απάντηση ότι ο πομπός το έλαβε (client mode). Αν δεν λάβει απάντηση για κάποιο χρονικό διάστημα, το ξαναστέλνει αν είναι ενεργοποιημένο το CRC.

Η “Receiver” κάνει wavrecord και μόλις λάβει όλο το πακέτο έχοντας τρέξει τη συνάρτηση συγχρονισμού, τρέχει διαδοχικά την συνάρτηση που θα κάνει unpad, την συνάρτηση “Demodulator”, την “Amp\_Phase\_Recovery” και “Processor” η οποία ανάλογα με τις επιλογές μας τρέχει άλλες συναρτήσεις (“Equalizer”, “Encoder” και “Interleaver”) και στέλνει θετική απάντηση (server mode) μπαίνοντας σε φάση αναμονής του νέου πακέτου (client mode).

Κάθε μια από τις παραπάνω συναρτήσεις τρέχει με τη σειρά της άλλες συναρτήσεις ανάλογα με τα ορίσματα που αντιστοιχούν στις επιλογές μας. Για παράδειγμα για να τρέξει ο αλγόριθμος του CMA, πρέπει να τρέξει η γενική συνάρτηση “Receiver”, αυτή όταν φτάσει στο στάδιο της επεξεργασίας του πακέτου να καλέσει την “Processor”, η οποία, αν στο setup είναι ανοιχτός ο ισοσταθμιστής, θα καλέσει την “Equalizer” η οποία θα αναγνωρίσει από το setup ότι έχει επιλεγεί ο CMA και θα καλέσει την “cma\_equalizer” με τις επιθυμητές παραμέτρους. Αυτή η αρθρωτή δομή γλιτώνει χρόνο κατά την εκτέλεση του κώδικα, ενώ κάνει πιο εύκολη την αντικατάσταση ή βελτίωση κάποιου υποσυστήματος.

## 2.3 Διαχωρισμός Έργου

Για την καλύτερη αξιοποίηση του διαθέσιμου χρόνου για την ολοκλήρωση του project, το απαιτούμενο έργο χωρίστηκε σε τμήματα και κάθε υποέργο ανατέθηκε σε μεμονωμένα μέλη της ομάδας ή υποομάδες. Στον παρακάτω πίνακα παρατίθεται η συνεισφορά του κάθε μέλους της ομάδας στην υλοποίηση.

Στάδιο	Ηλιάκης	Καρδαράς	Κοκκινάκης	Μπερβανάκης
Συν/γηση Hardware	✓	✓	✓	
Διαμορφωτές (PSK,PPM,QAM)	✓	✓		✓
Συγχρονισμός				✓
Ισοσταθμιστές			✓	
Κώδικες Convolutional	✓			
Κώδικες Reed Solomon		✓		
Interleavers		✓		
Viterbi	✓			✓
ARQ-CRC	✓			
Module Assembly				✓

# 3

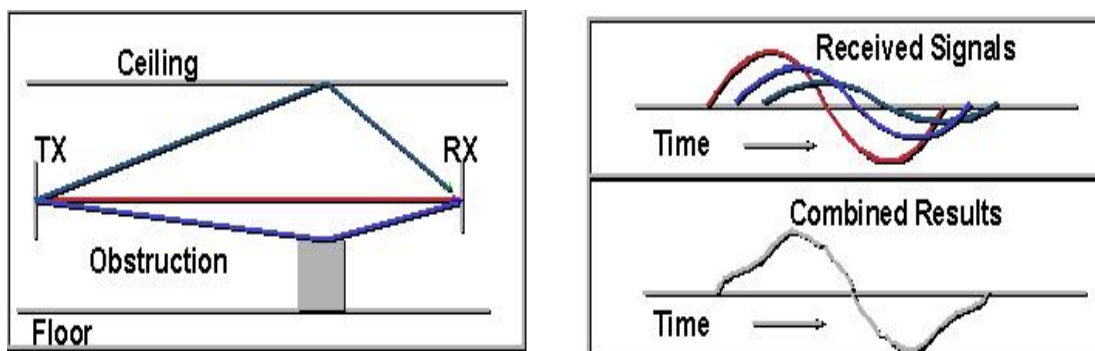
## ΙΣΟΣΤΑΘΜΙΣΗ ΚΑΝΑΛΙΟΥ

### 3.1 Εισαγωγή

Κανάλια επικοινωνίας όπως είναι τα τηλεφωνικά κανάλια, μικροκυμματικά ραδιοκανάλια οπτικής επαφής, δορυφορικά και υποβρύχια ακουστικά κανάλια μπορούν να μοντελοποιηθούν σαν φίλτρα περιορισμένου εύρους ζώνης [9-p.1]. Αυτό σημαίνει ότι παρουσιάζουν επιλεκτικότητα ως προς το συχνοτικό περιεχόμενο της πληροφορίας που επιτρέπουν να περάσει. Αυτό έχει σαν αποτέλεσμα ένα κανάλι περιορισμένου εύρους ζώνης να παραμόρφώνει το μεταδιδόμενο σήμα και να παρουσιάζεται διασυμβολική παρεμβολή (inter-symbol interference - ISI), δηλαδή ένα μεταδιδόμενο σύμβολο να παραμορφώνεται από τα υπόλοιπα μεταδιδόμενα σύμβολα στο ίδιο κανάλι.

Συγκεκριμένα στα ασύρματα τηλεπικοινωνιακά συστήματα, η κεραία εκπέμπει σε διάφορες κατευθύνσεις με αποτέλεσμα το σήμα να φτάνει στο δέκτη ακολουθώντας διαφορετικές διαδρομές (multipath). Έτσι το κυρίως σήμα μπορεί να φτάσει στο δέκτη ακολουθώντας την πιο σύντομη διαδρομή αλλά κάποια «αντίτυπα» του θα φτάσουν με καθυστέρηση εξαιτίας των φαινομένων ανάκλασης, διάθλασης και περίθλασης του σήματος σε διάφορα εμπόδια που συναντάει στη διαδρομή του. Η διασπορά αυτή στο χρόνο που φτάνουν τα διάφορα «αντίτυπα» στο δέκτη οδηγούν σε διασυμβολική παρεμβολή, αφού το κάθε δείγμα του σήματος παραμορφώνεται από εξασθενημένα «αντίτυπα» παλαιότερων δειγμάτων. Το φαινόμενο αυτό παριστάνεται γραφικά στο παρακάτω σχήμα.





Σχήμα 3.1

Όσο υψηλότερη είναι η ταχύτητα μετάδοσης (δηλαδή όσο μικρότερη είναι η χρονική απόσταση μεταξύ των διαφορετικών δειγμάτων), ένα «αντίτυπο» ενός δείγματος μπορεί να καθυστερήσει τόσο ώστε να φτάσει ταυτόχρονα με κάποιο επόμενο δείγμα παραμορφώνοντας το. Πρακτικά το φαινόμενο του multipath εμφανίζεται σαν ηχώ (voice echo) στα τηλεφωνικά συστήματα, φαντάσματα (video ghosts) στα συστήματα εικόνας και φαινόμενο Rayleigh στα κινητά τηλεπικοινωνιακά συστήματα.

Η μέθοδος που χρησιμοποιείται προκειμένου να αναιρέσουμε το φαινόμενο της διασυμβολικής παρεμβολής (ISI) και να αποκαταστήσουμε το σωστό σήμα είναι η ισοστάθμιση καναλιού (channel equalization). Η λειτουργία του ισοσταθμιστή σε ένα σύστημα είναι η εξής:

- i. Εκτίμηση των συντελεστών του καναλιού, δηλαδή ενός ανύσματος με το οποίο μπορεί να πολλαπλασιαστεί οποιοδήποτε σήμα εισόδου και να δώσει το σήμα εξόδου του καναλιού.
- ii. Χρησιμοποίηση αυτής της εκτίμησης προκειμένου να σχεδιάσουμε ένα φίλτρο ισοστάθμισης για να αναιρέσουμε την επίδραση του καναλιού.
- iii. Εφαρμογή του παραπάνω φίλτρου στο μεταδιδόμενο σήμα.

Σημειώνεται ότι κάποιοι ισοσταθμιστές όπως ο LMS το πρώτο στάδιο δεν υπάρχει αλλά γίνεται απευθείας εκτίμηση του ισοσταθμιστή

### 3.2 Προσαρμοστική Ισοστάθμιση

Σε κανάλια επικοινωνίας όπως τα κανάλια ραδιοφωνικών συχνοτήτων, η πιο κοινή μέθοδος που χρησιμοποιείται για να καταπολεμηθεί η διασυμβολική παρεμβολή (ISI) είναι η χρήση γραμμικών φίλτρων με ρυθμιζόμενους συντελεστές. [5] Παρόλο που υπάρχουν και άλλα είδη ισοστάθμισης όπως maximum likelihood sequence detection κτλ, δεν μπορούμε να τα χρησιμοποιήσουμε σε εφαρμογές ασύρματης δικτύωσης, γιατί αυτές οι τεχνικές θεωρούν γνωστό και μη μεταβαλλόμενο στο χρόνο κανάλι επικοινωνίας. Κάθε φορά που χρησιμοποιείται ένας δέκτης, η κατάσταση του καναλιού είναι διαφορετική και εξαρτώμενη από το περιβάλλον όπως εξηγήθηκε και στην παράγραφο 3.1. Έτσι ακόμα και αν το link ανάμεσα σε δύο υπολογιστές είναι σταθερό (π.χ δύο υπολογιστές μέσα στο χώρο του εργαστηρίου), το κανάλι σε κάθε μετάδοση θα είναι διαφορετικό, καθώς κόσμος και αντικείμενα κινούνται μέσα σε αυτό, ακόμα και οι ατμοσφαιρικές συνθήκες μπορούν να μεταβάλουν τα χαρακτηριστικά του καναλιού.

Τα γραμμικά φίλτρα που χρησιμοποιούνται για ισοστάθμιση χωρίζονται σε δύο κατηγορίες τα προρυθμιζόμενα (preset) και τα προσαρμοστικά (adaptive). Στους πρώτους, γίνεται μέτρηση των χαρακτηριστικών του καναλιού και ρυθμίζονται οι παράμετροι του ισοσταθμιστή. Μετά από αυτή τη φάση, κατά τη διάρκεια της μεταφοράς δεδομένων οι παράμετροι θεωρούμε ότι παραμένουν σταθεροί. Αντίθετα στους προσαρμοστικούς οι παράμετροι ρυθμίζονται περιοδικά κατά τη διάρκεια της μετάδοσης προκειμένου να μπορούν να ακολουθούν (tracking) μικρές μεταβολές των χαρακτηριστικών του καναλιού στο χρόνο. Οι πρώτοι είναι κατάλληλοι περισσότερο για ενσύρματα τηλεφωνικά δίκτυα όπου το κανάλι είναι μεν μεταβλητό, καθώς για κάθε κλήση υπάρχουν πολλές διαθέσιμες διαδρομές καλωδίου και κάθε φορά μπορεί να επιλεγεί διαφορετική, αλλά από τη στιγμή που θα εδραιωθεί η κλήση το κανάλι παραμένει σταθερό και άρα είναι δυνατό να μετρηθούν τα χαρακτηριστικά του. Στα ασύρματα δίκτυα όμως το κανάλι μπορεί να μεταβάλλεται συνεχώς εξαιτίας του μεταβαλλόμενου περιβάλλοντος ανάμεσα

σε πομπό και δέκτη, γι' αυτό οι προσαρμοστικοί αλγόριθμοι ισοστάθμισης πετυχαίνουν καλύτερη απόδοση. [2]

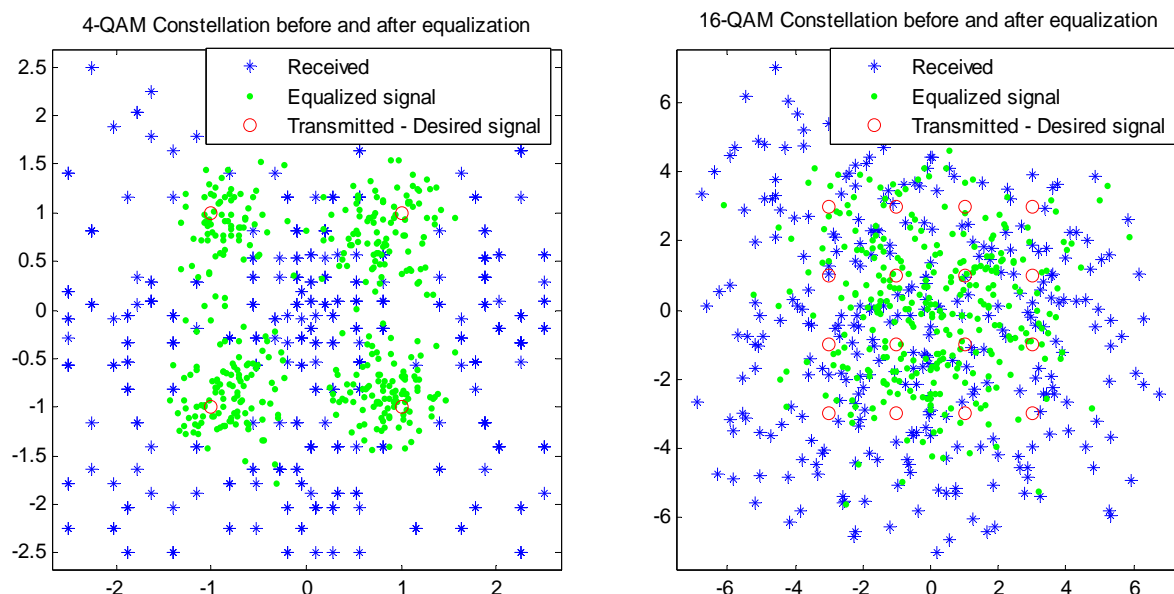
Ένας προσαρμοστικός αλγόριθμος είναι μια διαδικασία ρύθμισης των παραμέτρων ενός προσαρμοστικού φίλτρου προκειμένου να ελαχιστοποιηθεί μια συνάρτηση κόστους επιλεγμένη ανάλογα με την εφαρμογή. Η γενική μορφή ενημέρωσης των συντελεστών ενός προσαρμοστικού FIR φίλτρου είναι

$$W(n+1) = W(n) + \mu * G(e(n), X(n)), \quad (3.2)$$

όπου  $G()$  είναι η συνάρτηση κόστους,  $\mu$  το βήμα (step size),  $e(n)$  το σήμα σφάλματος και  $X(n)$  η ακολουθία εισόδου του φίλτρου.

Το βήμα καθορίζει το πλάτος της μεταβολής που κάνει ο αλγόριθμος προκειμένου να καθοριστεί επαναληπτικά ένα χρήσιμο διάνυμα συντελεστών. Διαλέγουμε το  $\mu$  να είναι ένας μικρός, θετικός αριθμός προκειμένου να πετύχουμε τη σύγκλιση της επαναληπτικής διαδικασίας σε μια βέλτιστη τιμή. Εν γένει μια τέτοια σύγκλιση δεν είναι δυνατή από μια τέτοια επαναληπτική μέθοδο, όμως μπορούμε να φτάσουμε όσο κοντά στη βέλτιστη λύση θέλουμε με μερικές εκατοντάδες ή και λιγότερο επαναλήψεις. Η συνάρτηση κόστους καθορίζει πόσο κοντά βρίσκεται το λαμβανόμενο σήμα από το επιθυμητό.

Στόχος των προσαρμοστικών αλγορίθμων είναι να ελέγχουν, από την πρόβλεψη της συνάρτησης κόστους, τις τιμές της εξόδου του φίλτρου του ισοσταθμιστή με κάποια επιθυμητή τιμή σε κάθε επανάληψη του αλγόριθμου, να υπολογίζουν τη διαφορά και με βάση αυτή να προσαρμόζουν εκ νέου τις την έξοδο του φίλτρου. Η επιθυμητή τιμή μπορεί να εκτιμηθεί από κάποια γνωστή ακολουθία (training sequence) που μεταδίδεται μαζί με το σήμα, ή από κάποια γνωστή ιδιότητα του μεταδιδόμενου σήματος. Το πρόβλημα με αυτή την τεχνική είναι ότι δεν είναι γνωστές οι επιθυμητές τιμές του μεταδιδόμενου σήματος έτσι ώστε να ρυθμιστούν κατάλληλα οι συντελεστές του φίλτρου. Προκειμένου να εκτιμηθούν αυτές οι τιμές, χρησιμοποιούνται δύο προσεγγίσεις για ασύρματες ζεύξεις καναλιών με στοχαστικές ιδιότητες: η χρήση ακολουθίας εκμάθησης (training sequence) και η τυφλή ισοστάθμιση (blind equalization) που περιγράφονται στην παράγραφο 3.4.



Εικόνα 3.1

Το αποτέλεσμα της εφαρμογής ενός προσαρμοστικού ισοσταθμιστή (LMS) για τυχαίο σήμα και τυχαίο κανάλι μηδενικού θορύβου φαίνεται στην εικόνα 3.1.

Στους παραπάνω αστερισμούς φαίνεται η επίδραση του ισοσταθμιστή στο λαμβανόμενο σήμα. Παρατηρούμε μεγάλη απόκλιση ανάμεσα στο μεταδιδόμενο (κόκκινο “ο”) και λαμβανόμενο (μπλε “\*”) σήμα το οποίο οφείλεται αποκλειστικά και μόνο στο κανάλι καθώς δεν έχουμε θεωρήσει θόρυβο στο παρόν simulation. Με εφαρμογή ισοστάθμισης στο λαμβανόμενο σήμα έχουμε καλύτερα αποτελέσματα (πράσινα •), πιο κοντά στο επιθυμητό σήμα. Συγκρίνοντας τις δύο γραφικές, παρατηρούμε ότι η διασυμβολική παρεμβολή γίνεται πιο αισθητή και λιγότερο αναστρέψιμη όσο αυξάνει το  $M$ , δηλαδή ο αριθμός των λαθών μετά την ισοστάθμιση θα αυξάνεται όσο αυξάνεται και το  $M$ . Παράλληλα όμως η αύξηση του αριθμού των bits ανά σύμβολο  $M$  είναι ένας από τους λίγους διαθέσιμους τρόπους για αύξηση της ταχύτητας εκπομπής. Έτσι μια από τις σημαντικότερες επιλογές που πρέπει να γίνουν προκειμένου το σύστημα να επιτύχει γρήγορη μετάδοση με ελάχιστο ποσοστό λαθών ( $\text{bit error rate} < 10^{-6}$ ) είναι η επιλογή του  $M$  ώστε να έχουμε όσο το δυνατόν ταχύτερη επικοινωνία διατηρώντας το BER σε αποδεκτά επίπεδα.

### 3.2.1 Μέτρα απόδοσης προσαρμοστικών φίλτρων

Η απόδοση ενός προσαρμοστικού αλγόριθμου μπορεί να εξεταστεί υπό το πρίσμα διαφόρων κριτηρίων. Ένα σημαντικό κριτήριο είναι ο *ρυθμός σύγκλισης*, δηλαδή πόσο γρήγορα προσαρμόζονται οι παράμετροι του φίλτρου μέχρι να φτάσουν την βέλτιστη λύση. Ένας γρήγορος ρυθμός προσαρμογής επιτρέπει στον αλγόριθμο να παίρνει ακριβείς αποφάσεις σχετικά με την τιμή των bits πιο γρήγορα από έναν αλγόριθμο με χαμηλό ρυθμό σύγκλισης. Επίσης επιτρέπει καλύτερη απόδοση στην παρακολούθηση (tracking) γρήγορα μεταβαλλόμενων καναλιών μετάδοσης από τον ισοσταθμιστή.

Ένα δεύτερο μέτρο απόδοσης είναι το *σφάλμα προσαρμογής* (misadjustment), δηλαδή πόσο κοντά στην θεωρητική βέλτιστη λύση φτάνει το κριτήριο προσαρμογής (adaptation criterion) που χρησιμοποιεί ο αλγόριθμος. Όσο πιο κοντά πλησιάζει, τόσο πιο ακριβείς είναι οι εκτιμήσεις των συμβόλων που καταφέρνει ο αλγόριθμος και άρα πιο κοντά στην επιθυμητές τιμές.

Ακόμα, η υπολογιστική πολυπλοκότητα που χρειάζεται ο κάθε αλγόριθμος, δηλαδή ο αριθμός των απαιτούμενων πράξεων, για να υπολογίσει τους συντελεστές του φίλτρου ισοστάθμισης, παίζει επίσης σημαντικό ρόλο στην απόφαση του είδους του προσαρμοστικού αλγόριθμου που θα χρησιμοποιηθεί σε κάποιο σύστημα για ισοστάθμιση. Ο παράγοντας αυτός για την ακρίβεια είναι ιδιαίτερα κρίσιμος για συστήματα ισοστάθμισης πραγματικού χρόνου όπως αυτό που αναπτύσσουμε, αλλά και για συστήματα περιορισμένων πόρων. Η *ανθεκτικότητα* (robustness) του φίλτρου ισοστάθμισης, δηλαδή η ικανότητά του να μην επηρεάζεται σημαντικά από μικρές διαταραχές του σήματος, η *δομή* του και συγκεκριμένα κάποιες ιδιότητες όπως παραλληλισμός, modularity κτλ καθώς και κάποιες *αριθμητικές ιδιότητες*, όπως αριθμητική ευστάθεια (stability), ακρίβεια (accuracy) ή επίδραση από τον κβαντισμό του σήματος είναι μερικές ακόμα ιδιότητες του φίλτρου που λαμβάνουμε υπόψη στην μελέτη του ισοσταθμιστή.

Οι αλγόριθμοι που θα εξεταστούν στα πλαίσια αυτής της εργασίας, θα αξιολογηθούν με βάση όλα αυτά τα κριτήρια και υπό διάφορες συνθήκες στα κεφάλαια που ακολουθούν.

### 3.3 Τεχνικές Ισοστάθμισης

Στην προσαρμοστική ισοστάθμιση με χρήση *ακολουθίας εκπαίδευσης* (training sequence), ο ισοσταθμιστής «εκπαιδεύεται» με τη μετάδοση από τον πομπό μιας γνωστής ψευδοτυχαίας ακολουθίας μέσα από το κανάλι. Ο ισοσταθμιστής στον αποδιαμορφωτή χρησιμοποιεί τη γνωστή αυτή ακολουθία για την ρύθμιση των συντελεστών του. Μετά την ρύθμιση, ο προσαρμοστικός ισοσταθμιστής αλλάζει την λειτουργία του από την «εκπαίδευση» (training mode) σε «οδηγούμενη από απόφαση» (decision directed mode). Έχοντας υπολογίσει κατά το αμέσως προηγούμενο χρονικό διάστημα την κατάσταση του καναλιού και έχοντας ρυθμιστεί σύμφωνα με αυτή, ο ισοσταθμιστής φιλτράρει την «άγνωστη» πληροφορία που ακολουθεί σύμφωνα με τους συντελεστές αυτούς, που προσεγγίζουν αρκετά την κατάσταση του καναλιού, αν η μεταβολή δεν είναι τόσο γρήγορη. Αν μάλιστα η εκπαίδευση με αποστολή της «γνωστής» ακολουθίας γίνεται αρκετά συχνά, τότε το φίλτρο επανεκπαιδεύεται και ακολουθεί τις μεταβολές του καναλιού.

Στην *τυφλή* ισοστάθμιση, το επιθυμητό σήμα είναι άγνωστο στον δέκτη, εκτός από κάποιες πιθανοτικές (probabilistic), στατιστικές (statistical) ή δομικές (structural) ιδιότητες πάνω σε κάποιο αλφάβητο  $A$ . Έτσι, εφόσον τόσο το σήμα εισόδου  $x(n)$ , όσο και το κανάλι  $h(n)$  είναι άγνωστα, ο μόνος τρόπος για να υπολογιστεί η έξοδος του ισοσταθμιστή είναι να γίνει χρήση των πιθανοτικών ή στατιστικών ιδιοτήτων του.

### 3.4 Προσαρμοστικοί Αλγόριθμοι Ισοστάθμισης

Για την ισοστάθμιση του καναλιού του συστήματος που αναπτύχθηκε χρησιμοποιήθηκαν τρεις αλγόριθμοι: ο least means square (LMS), ο recursive least mean (RLS) και ο constant modulus (CMA). Καθένας από αυτούς παρουσιάζει πλεονεκτήματα αλλά και κάποια μειονεκτήματα, και οι τρεις όμως είναι από τους πιο δημοφιλείς αλγόριθμους για ισοστάθμιση καναλιού. Η απόδοση καθενός από αυτούς εξαρτάται από την εφαρμογή και είναι το αντικείμενο ενδελεχούς μελέτης στα επόμενα τρία κεφάλαια.

Ουσιαστικά οι ισοσταθμιστές όπως η μέθοδος Steepest Descent, ο LMS και ο RLS, που είναι γνωστοί σαν “sample adaptive filters”, είναι επαναληπτικές υλοποιήσεις της ιδέας ελαχιστοποίησης του μέσου τετραγωνικού σφάλματος ανάμεσα στην είσοδο και την έξοδο του ισοσταθμιστή ενώ κάνουν περιοδική χρήση training sequence για την ρύθμιση των συντελεστών τους. Έχουν το πλεονέκτημα της μικρότερης καθυστέρησης επεξεργασίας και μεγαλύτερης ικανότητας μεταβαλλόμενων σημάτων εισόδου σε σχέση με τους προσαρμογείς block δεδομένων (block-adaptive filters). Οι αλγόριθμοι LMS και RLS θα εξεταστούν στα κεφάλαια 4 και 5 που ακολουθούν.

Αντίθετα οι blind αλγόριθμοι όπως ο CMA βελτιώνουν την εκμετάλλευση του καναλιού, καθώς δεν κάνουν χρήση training sequence. Έχουν επίσης το πλεονέκτημα της εύκολης ανάνηψης (recovery) σε περίπτωση σφάλματος, πράγμα διόλου σπάνιο σε συστήματα επεξεργασίας σήματος σε πραγματικό χρόνο. Ο αλγόριθμος CMA θα εξεταστεί στο κεφάλαιο 6.

# 4

## ΑΛΓΟΡΙΘΜΟΣ LEAST MEANS SQUARE

### 4.1 Η λύση του Wiener

Η εξίσωση 3.2 στο προηγούμενο κεφάλαιο μας δίνει τον τύπο που ακολουθούν τα προσαρμοστικά συστήματα για την προσαρμογή των παραμέτρων τους στο περιβάλλον:

$$W(n+1) = W(n) + \mu * G(e(n), X(n)),$$

Η παράμετρος  $G()$  εξαρτάται από την συνάρτηση κόστους (cost function) που θα επιλέξουμε, με βάση την οποία θα κρίνουμε το κατά πόσο το λαμβανόμενο σήμα προσεγγίζει το επιθυμητό και τι ενέργειες θα κάνει ο αλγόριθμος προκειμένου να βελτιώσει ακόμα περισσότερο τη διαδικασία. Μία συνάρτηση που χρησιμοποιήθηκε από πολύ νωρίς σε αλγόριθμους ισοστάθμισης καναλιού είναι η *συνάρτηση μέσου τετραγωνικού σφάλματος* (mean square error function), η οποία ορίζεται παρακάτω:

$$J_{MSE}(n) = \frac{1}{2} \int_{-\infty}^{\infty} e^2(n) p_n(e(n)) de(n) = \frac{1}{2} E\{e^2(n)\} \quad (4.1)$$

όπου η  $p_n(e)$  είναι η συνάρτηση πυκνότητας πιθανότητας του σφάλματος τη στιγμή  $n$ ,  $d$  το επιθυμητό σήμα και  $e(n)$  το σφάλμα τη στιγμή  $n$ .

Η βέλτιστη λύση για την ελαχιστοποίηση της παραπάνω συνάρτησης κόστους στον υπολογισμό των παραμέτρων  $W$  του φίλτρου δίνονται από την εξίσωση του Wiener και σύμφωνα με τον [2-18.6.3] δίνονται από την εξίσωση:

$$W_{MSE}(n) = R_{xx}^{-1}(n) P_{dx}(n) \quad (4.2)$$

όπου  $R_{xx} = E\{X(n)X^T(n)\}$  και  $P_{dx}(n) = E\{d(n)X(n)\}$

Παρόλο που ο τύπος της 4.2 δίνει την βέλτιστη λύση, απαιτεί την αντιστροφή του πίνακα  $R$ , πράξη με μεγάλη υπολογιστική πολυπλοκότητα ( $O(M^3)$ ) και άρα ανάγκη για μεγάλη υπολογιστική ισχύ. Για το σύστημα ανταλλαγής αρχείων που αναπτύχθηκε ο υπολογισμός των παραμέτρων  $W$  με την εξίσωση του Wiener είναι αδύνατος πρακτικά για δύο λόγους:



- Η απαραίτητη υπολογιστική ισχύς για την αντιστροφή του πίνακα δεν είναι διαθέσιμη σε συστήματα πραγματικού χρόνου, καθώς η πληροφορία καταφθάνει συνεχώς και η ταυτόχρονη επεξεργασία τέτοιου όγκου δεδομένων ή η διάθεση των απαραίτητων buffers για ετεροχρονισμένη επεξεργασία δεν όπως το σύστημα επικοινωνίας που αναπτύχθηκε.
- Το κανάλι επικοινωνίας του συστήματος είναι ασύρματο και όχι σταθερό με αποτέλεσμα κάποιες φορές να είναι αναγκαία η χρήση φίλτρου πολλών παραμέτρων γεγονός που επιβαρύνει ακόμα περισσότερο τη διαδικασία.

## 4.2 Method of Steepest Descent

Η μέθοδος *Steepest Descent* είναι ένας εναλλακτικός τρόπος υπολογισμού της λύσης Wiener η οποία είναι απλούστερη υπολογιστικά (κατά μια τάξη). Ουσιαστικά είναι μια διαδικασία ελαχιστοποίησης της τιμής της συνάρτησης κόστους σε ένα σύνολο από προσαρμοζόμενους συντελεστές  $W(n)$ . Η μέθοδος αυτή είναι επαναληπτική (iterative), με την έννοια ότι ξεκινάει από μια αρχική τιμή για τις παραμέτρους  $W$  και με την πάροδο των επαναλήψεων αλλάζει τις τιμές αυτές προσεγγίζοντας τελικά την παραπάνω βέλτιστη λύση του Wiener-Hopf. Η εξίσωση με βάση την οποία γίνεται αυτή η ρύθμιση των παραμέτρων  $W$  δίνεται παρακάτω:

$$W_i(n+1) = W_i(n) - \frac{1}{2} \mu(n) \frac{\partial J(n)}{\partial W_i(n)} \quad (4.3)$$

όπου  $\mu$  το step size της προσαρμογής και  $J(n)$  η συνάρτηση κόστους. Δηλαδή, σύμφωνα με τον τύπο αυτό, κάθε συντελεστής  $i$  προσαρμόζεται ανάλογα με την παράγωγο της συνάρτησης κόστους για τον συντελεστή αυτό. Υπολογίζοντας την παράγωγο αυτή συναρτήσει των συναρτήσεων συσχέτισης  $R$  και αυτοσυσχέτισης  $p$  έχουμε:

$$\frac{\partial J(n)}{\partial W_i(n)} = -2p + 2RW(n) \quad (4.4)$$

οπότε αντικαθιστώντας την εξίσωση 4.4 στην 4.3 έχουμε τον παρακάτω τύπο για την μέθοδο προσαρμογής των παραμέτρων  $W$ :

$$W(n+1) = W(n) - \mu(n)(p(n) - R(n)W(n)) \quad (4.5)$$

### 4.3 Ο αλγόριθμος Least Mean Square

Η μέθοδος του Steepest Descent εμπλέκει τη μεταβολή μέσου τετραγωνικού σφάλματος με στόχο να βρεθούν οι συντελεστές του φίλτρου που ελαχιστοποιούν το τετραγωνικό σφάλμα ανάμεσα στο λαμβανόμενο και το επιθυμητό σήμα. Το πρόβλημα με την παραπάνω διαδικασία είναι ότι ενώ η προσαρμογή (adaptation) των συντελεστών εξαρτάται από τις στατιστικές ιδιότητες  $E\{d(n)x(n-i)\}$  και  $E\{x(n-i)x(n-j)\}$  οι οποίες περιέχονται στα  $P_{dx}(n), R_{xx}(n)$ , στην πράξη μόνο οι τιμές των  $x(n)$  και  $d(n)$  μπορούν να χρησιμοποιηθούν στην διαδικασία προσαρμογής του αλγορίθμου. Είναι βέβαια δυνατόν από τις τιμές αυτές να εκτιμηθούν οι στατιστικές ιδιότητες που απαιτούνται, είναι όμως αρκετά πολύπλοκο υπολογιστικά και μερικές φορές αδύνατο χωρίς γνώση προηγούμενων δειγμάτων. Για το λόγω αυτό χρησιμοποιείται ένας υπολογιστικά απλούστερος αλγόριθμος που αντί για τη μεταβολή του τετραγωνικού σφάλματος χρησιμοποιεί μια εκτίμηση που προέρχεται από το τρέχον (instantaneous) τετραγωνικό σφάλμα. Ο αλγόριθμος αυτός είναι ο Least Mean Squares (LMS) και η μέθοδος προσαρμογής των συντελεστών  $w_i$  του φίλτρου ακολουθεί τον τύπο:

$$W(n+1) = W(n) - \mu \left( \frac{\partial e^2(n)}{\partial W(n)} \right) \quad (4.6)$$

όπου το σφάλμα  $e(n) = x(n) - \hat{x}(n) \Rightarrow e(n) = x(n) - w^T(n)x(n)$  και υπολογίζοντας την παράγωγο του παραπάνω όρου έχουμε την απλούστερη μορφή της παρακάτω εξίσωσης:

$$W(n+1) = W(n) - \mu(n)e(n)X(n) \quad (4.7)$$

Ο υπολογισμός της προσαρμογής που κάνει ο συγκεκριμένος αλγόριθμος με βάση την εξίσωση 4.7 είναι πολύ απλός τόσο από πλευράς χωρικής όσο και από πλευράς χρονικής πολυπλοκότητας (πρακτικά απαιτήσεις σε μνήμη και υπολογιστική ισχύ αντίστοιχα). Μάλιστα ο αριθμός και το είδος των πράξεων που χρειάζεται να εκτελέσει ο παραπάνω αλγόριθμος είναι ίδιος με ένα FIR φίλτρο σταθερών τιμών παραμέτρων, παρόλο που οι τιμές αυτές μεταβάλλονται σε κάθε επανάληψη.

Συγκεκριμένα ο αλγόριθμος υπολογισμού των συντελεστών  $W$  για το φίλτρο ισοστάθμισης εκτελεί διαδοχικά 3 βήματα έως ότου επιτευχθεί η σύγκλιση του αλγορίθμου στην βέλτιστη λύση Wiener-Hopf. Τα βήματα αυτά είναι:

i. Υπολογισμός εξόδου:  $y(n) = \hat{w}^H(n)x(n)$  (4.8)

ii. Εκτίμηση σφάλματος:  $e(n) = d(n) - y(n)$  (4.9)

iii. Προσαρμογή συντελεστών:  $\hat{w}(n+1) = \hat{w}(n) + \mu x(n)e^*(n)$  (4.10)

όπου  $x(n)$  και  $y(n)$  τα σήματα εισόδου και εξόδου του φίλτρου αντίστοιχα,  $d(n)$  το επιθυμητό σήμα,  $e(n)$  το σήμα σφάλματος,  $\hat{w}(n)$  η εκτίμηση των συντελεστών (tap-weights) του φίλτρου και  $\mu$  το step size δηλαδή το βήμα προσαρμογής σε κάθε επανάληψη.

Όπως παρατηρούμε από την παραπάνω ακολουθία ενεργειών οι μόνες τιμές που είναι απαραίτητες σε κάθε επανάληψη για τον υπολογισμό του επόμενου δείγματος είναι τα τρέχοντα στιγμιότυπα των  $x$ ,  $d$  και  $\hat{w}$ . Τα δύο πρώτα βήματα αποτελούν τη διαδικασία φιλτραρίσματος (filtering process), ενώ το τρίτο τη διαδικασία προσαρμογής (adaptive process). Η πρώτη διαδικασία είναι γραμμική ενώ η δεύτερη είναι μη-γραμμική και επαναληπτική.

## 4.4 Αξιολόγηση του αλγόριθμου LMS

### 4.4.1 Πολυπλοκότητα

Μερικά πλεονεκτήματα του αλγόριθμου LMS σε σχέση με τις προηγούμενες μεθόδους που αναλύθηκαν στις παραγράφους 4.1 και 4.2 μπορούν να εξαχθούν κατευθείαν από την εξίσωση 4.10 που υπολογίζει τις νέες τιμές των συντελεστών του  $W$  σε κάθε επανάληψη. Πρώτον, η εξίσωση είναι αρκετά απλή όσον αφορά το είδος των απαιτούμενων πράξεων καθώς δεν χρειάζεται αντιστροφές πινάκων όπως η λύση του Wiener-Hopf παρά μόνο περίπου  $2L$  προσθέσεις και  $2L+1$  πολλαπλασιασμούς σε κάθε βήμα, όπου  $L$  το μήκος του φίλτρου. Επίσης δεν απαιτείται υπολογισμός μέσων όρων ή άλλων στατιστικών ιδιοτήτων όπως τα  $R$  και  $p$  στην MSD και έτσι αποφεύγονται υπολογιστικές και αναλυτικές δυσκολίες που υπεισέρχονται στον υπολογισμό συσχετίσεων. Η πολυπλοκότητά του φαίνεται στον παρακάτω πίνακα:

Βήμα αλγόριθμου	$\mathbf{0x}$	$\mathbf{O+}$
1	L	L-1
2	0	1
3	L+1	L
Πολυπλοκότητα	$\mathbf{Ox(2L+1)}$	$\mathbf{O+(2L)}$

Πίνακας 4.1 – Πολυπλοκότητα LMS

Παρατηρούμε ότι η συνολική πολυπλοκότητα υπολογισμού των tap-weights σε μορφή διανύσματος είναι της τάξης  $\mathbf{O(L)}$ , δηλαδή μια τάξη χαμηλότερη από τη μέθοδο Steepest Descent και δύο τάξεις από τον απευθείας υπολογισμό με την εξίσωση των Wiener-Hopf. Το trade-off για τη μετάβαση από τη λύση του Wiener-Hopf στην μέθοδο Steepest Descent είναι η αναγκαιότητα να συμβιβαστούμε με μια προσέγγιση της βέλτιστης λύσης και όχι τη βέλτιστη λύση αυτή καθαυτή, ενώ για την μετάβαση από την Steepest Descent στον LMS πρέπει επιπρόσθετα να ανεχτούμε και μια αρκετά πιο αργή σύγκλιση τουλάχιστον για μικρά μήκη φίλτρου όπως αυτά που χρησιμοποιήσαμε στο σύστημα (βλ παράγραφο 4.5.2).

#### 4.4.2 Σύγκλιση στο MSE – Ευστάθεια – Learning Curve

Η ευστάθεια του LMS εξαρτάται από την επιλογή της παραμέτρου  $\mu$ , η οποία θα μελετηθεί στο κεφάλαιο 4.5. Για μεγάλες τιμές του  $\mu$ , προκαλεί αστάθεια, δηλαδή αδυναμία του αλγόριθμου να συγκλίνει στις βέλτιστες τιμές των συντελεστών του φίλτρου, αλλά υπάρχει ταλάντωση γύρω από αυτές. Ακολουθώντας την κατάλληλη ανάλυση, αποδεικνύεται [5-p.101] ότι μπορεί να μπει ένα άνω όριο στην τιμή του  $\mu$ , μέχρι το οποίο η σύγκλιση στις βέλτιστες τιμές και συνεπώς η ευστάθεια είναι εξασφαλισμένα. Η σχέση που πρέπει να ικανοποιείται προκειμένου ο LMS να συγκλίνει στο μέσο τετραγωνικό σφάλμα είναι:

$$0 < \mu < \frac{2}{\lambda_{\max}} \quad (4.11)$$

όπου  $\lambda_{\max}$  είναι η μέγιστη ιδιοτιμή του πίνακα συσχέτισης  $\mathbf{R}$ . Αν και ο παραπάνω τύπος θέτει ένα άνω όριο στην τιμή που μπορεί να πάρει το  $\mu$ , παρουσιάζει δύο προβλήματα: η μέγιστη ιδιοτιμή του  $\mathbf{R}$  είναι διαθέσιμη [3-

p.402] και το όριο που θέτει είναι αρκετά υψηλό για να εξασφαλίσει ευστάθεια [6-p.171]. Έτσι μια πιο συντηρητική επιλογή του  $\mu_{\max}$  μπορεί να υπολογιστεί χρησιμοποιώντας το  $\text{trace}(R)$ , οπότε η σχέση 4.11 γίνεται:

$$0 < \mu < \frac{2}{\text{tr}(R)} \quad (4.12)$$

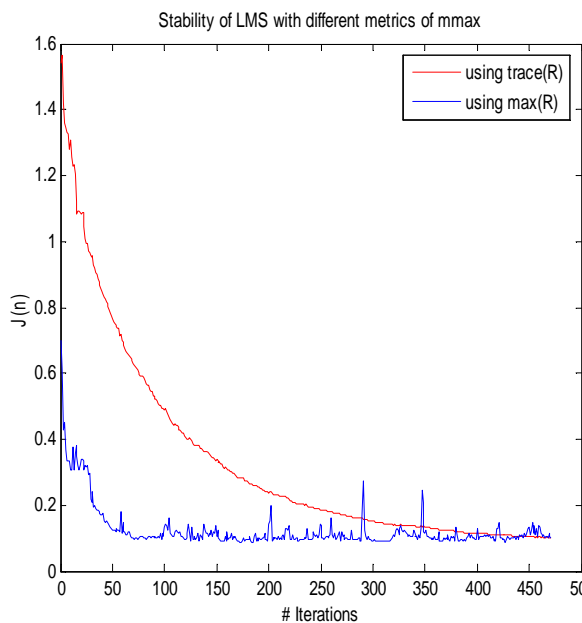
και επειδή ο πίνακας  $R$  είναι εκτός από θετικά ορισμένος (positive definite) και Toeplitz με όλους τα διαγώνια στοιχεία του ίσα με την μέση τετραγωνική τιμή της εισόδου σε κάθενα από τα  $L$  tap-weights του φίλτρου [3-p.403] έχουμε:

$$\text{tr}(R) = \sum_{k=0}^{L-1} E\{|u(n-k)|^2\} \quad (4.13)$$

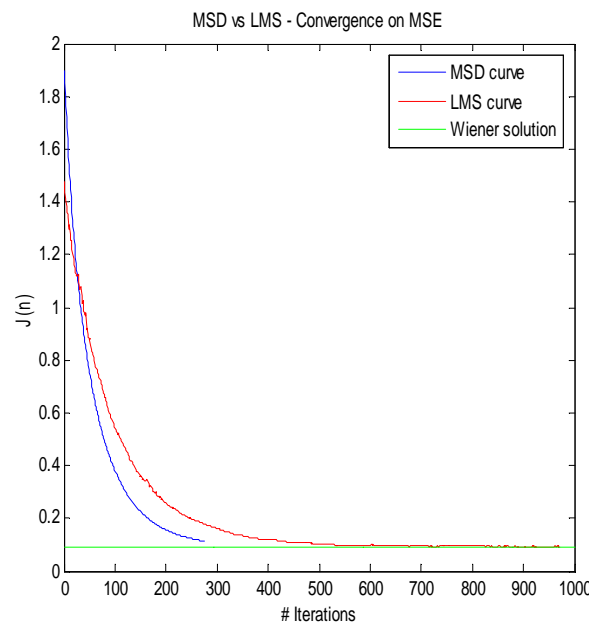
και επειδή το τελευταίο άθροισμα ισούται με την ισχύ των taps εισόδου:

$$0 < \mu < \frac{2}{\text{tap} - \text{input} - \text{power}} \quad (4.14)$$

Η υπεροχή των εξισώσεων 4.12 και 4.14 για τον υπολογισμό του  $\mu_{\max}$  όσον αφορά την ευστάθεια αποδεικνύεται και στην εικόνα 4.1 όπου και οι δύο γραφικές αναπαριστούν την σύγκλιση του LMS, με την πρώτη να υπολογίζει το  $\mu_{\max}$  χρησιμοποιώντας τον τύπο 4.11 και η δεύτερη με τον τύπο 4.12.



Εικόνα 4.1



Εικόνα 4.2

Στην γραφική της εικόνας 4.2 φαίνεται η λύση που βρίσκουν η εξίσωση του Wiener, η μέθοδος του Steepest Descent και ο αλγόριθμος Least Mean Square σε συνάρτηση με τον αριθμό των απαιτούμενων επαναλήψεων έως τη λύση αυτή καθώς και η ακρίβεια σύγκλισης της κάθε μεθόδου στη βέλτιστη λύση.

Η προσομοίωση της λειτουργίας των παραπάνω μεθόδων υπολογισμού των συντελεστών  $W$  του φίλτρου έγινε για το ίδιο σήμα, το ίδιο κανάλι και τις ίδιες ακριβώς παραμέτρους προκειμένου να είναι δυνατή η μεταξύ τους σύγκριση και η εξαγωγή του τελικού συμπεράσματος λαμβάνοντας υπόψη φυσικά και το κόστος που πληρώνουμε σε κάθε περίπτωση σε υπολογιστική πολυπλοκότητα. Για τις ανάγκες της προσομοίωσης χρησιμοποιήθηκε ένα bitstream των 1000 bits που αποτέλεσε την προς μετάδοση πληροφορία, διαμόρφωση 4-QAM στον πομπό, συνελκτικό κανάλι με θόρυβο  $\sigma=0.05$ , φίλτρο μήκους 6 συντελεστών και step size 0.05 τόσο για τον LMS όσο και για την MSD. Το παραπάνω setup θα χρησιμοποιηθεί σε όλο το κεφάλαιο με τις παραπάνω τιμές εκτός αν κατά περίπτωση δηλώνεται κάτι άλλο.

Όπως παρατηρούμε, και οι δύο μέθοδοι πλησιάζουν την βέλτιστη λύση Wiener-Hopf (βλ threshold), η μεν Method of Steepest Descent σε 250 περίπου επαναλήψεις του αλγόριθμου προσαρμογής, ο δε LMS σε περίπου διπλάσιες.

#### 4.4.3 Ταχύτητα Σύγκλισης

Παρόλο που ο LMS κατορθώνει να συγκλίνει πάντα σε μια βέλτιστη λύση (υπό την προϋπόθεση της ανεξαρτησίας των τιμών εισόδου), η ταχύτητα προσαρμογής του είναι αρκετά αργή σε σχέση με άλλους αλγόριθμους όπως ο RLS που θα εξετασθεί στο επόμενο κεφάλαιο. Αυτό συμβαίνει γιατί η σύγκλιση δεν γίνεται με ομοιόμορφο τρόπο (uniformly), αλλά εξαρτάται από την ομοιομορφία των ιδιοτιμών (eigenvalue disparity)  $\lambda_i$  (δηλαδή του λόγου

$\frac{\lambda_{\max}}{\lambda_{\min}}$ ) του πίνακα συσχέτισης  $R$ . Ορίζοντας ένα διάνυσμα  $\gamma_n$  σαν το σφάλμα

ανάμεσα στις αναμενόμενες (expected) τιμές των συντελεστών του φίλτρου και τους αντίστοιχους της βέλτιστης λύσης προκύπτει [6-p.174] ότι  $\gamma_{n+1}(j) = (1 - \mu\lambda_j)\gamma_n(j)$  και λαμβάνοντας υπόψη τον περιορισμό που θέσαμε

για το  $\mu$  στην εξίσωση 4.11  $0 < \mu < \frac{2}{\lambda_{\max}}$  προκύπτει ότι όλοι οι συντελεστές του σφάλματος θα μειώνονται σε μέτρο σε κάθε επανάληψη. Όμως ο ρυθμός της μείωσης για κάθε στοιχείο του πίνακα  $W$  εξαρτάται από τον όρο  $|1 - \mu\lambda_j|$  και επειδή οι τιμές των  $\lambda$  είναι εν γένει διαφορετικές μεταξύ τους, κάποιες από αυτές θα συγκλίνουν γρηγορότερα από κάποιες άλλες. Έτσι η σύγκλιση του LMS στην βέλτιστη λύση γίνεται μη-ομοιόμορφα. Το πρόβλημα αυτό ονομάζεται *πρόβλημα ανομοιοτήτας των ιδιοτιμών* (eigenvalue disparity problem).

Οι τιμές που παίρνουν οι ιδιοτιμές εξαρτώνται με σειρά τους, τόσο από την απόκριση καναλιού, όσο και από την ισχύ του μεταδιδόμενου σήματος. Συγκεκριμένα, όσο μεγαλύτερη είναι η ισχύς ή όσο λιγότερο ISI προκαλεί το κανάλι, τόσο μειώνεται η ανομοιομορφία και άρα αυξάνεται η ταχύτητα σύγκλισης.

#### 4.4.4. Ενίσχυση συνιστωσών Θορύβου

Ένα ακόμα μειονέκτημα του LMS οφείλεται στην αντικατάσταση του πραγματικού gradient  $\nabla J(n)$  της μεθόδου της Steepest Descent (βλ εξίσωση 4.3) με την τοπική εκτίμηση του  $\nabla \hat{J}(n)$  του LMS. Δηλαδή στον LMS δεν υπάρχει time averaging που να εξομαλύνει τις όποιες απότομες διακυμάνσεις που οφείλονται στο θόρυβο, με αποτέλεσμα ο εν λόγω αλγόριθμος να παρουσιάζει ευαισθησία στο θόρυβο. Η εκτίμηση  $\nabla \hat{J}(n)$  μπορεί να χαρακτηριστεί σαν το άθροισμα του πραγματικού  $\nabla J(n)$  συν κάποιο θόρυβο που προέρχεται από τις διακυμάνσεις των δειγμάτων. Όσο πιο μεγάλο είναι το  $\mu$ , τόσο πιο μεγάλο μέρος του θορύβου ενσωματώνεται στους συντελεστές του φίλτρου με αποτέλεσμα την αύξηση του μέσου τετραγωνικού σφάλματος και άρα την πτώση της απόδοσης του φίλτρου ισοστάθμισης. Ακόμα και για τιμές όπου το φίλτρο εξόδου  $y(n)$  προσεγγίζει το επιθυμητό  $d(n)$ , οπότε  $\nabla J(n) \approx 0$ , το  $\nabla \hat{J}(n)$  θα κυμαίνεται γύρω από αυτή την τιμή οπότε το μέσο τετραγωνικό σφάλμα θα κυμαίνεται πάνω από την ελάχιστη τιμή  $J_{\min}$ . [6-p.165]

## 4.5 Επιλογή Παραμέτρων του LMS

Η λειτουργία του LMS απαιτεί την επιλογή τριών παραμέτρων: της τιμής της σταθεράς  $\mu$  που ονομάζεται *σταθερά προσαρμογής* (adaptation constant), του μήκους του φίλτρου  $L$  και της αρχικής τιμής του διανύσματος συντελεστών του φίλτρου  $w(0)$ . Η επιλογή αυτών των παραμέτρων του αλγόριθμου δεν υπακούει κάποιους συγκεκριμένους κανόνες αλλά εξαρτάται από την εφαρμογή. Για το λόγο αυτό, στις επόμενες τρεις παραγράφους, ακολουθεί μια ενδελεχής μελέτη των επιλογών και των επιπτώσεων κάθε επιλογής στη λειτουργία του συστήματος “Homebrew WFTP”.

### 4.5.1 Η σταθερά προσαρμογής $\mu$

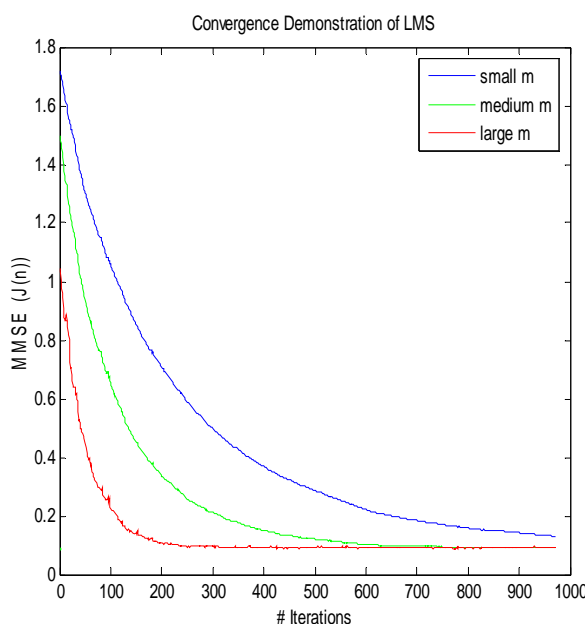
Η σταθερά προσαρμογής είναι η πιο σημαντική από τις παραμέτρους του LMS καθώς επηρεάζει την ταχύτητα σύγκλισης, την ευστάθεια και την ακρίβεια του αλγόριθμου. Όπως φαίνεται από την εξίσωση προσαρμογής 4.10 του LMS

$$\hat{w}(n+1) = \hat{w}(n) + \mu x(n)e^*(n) \quad (4.10)$$

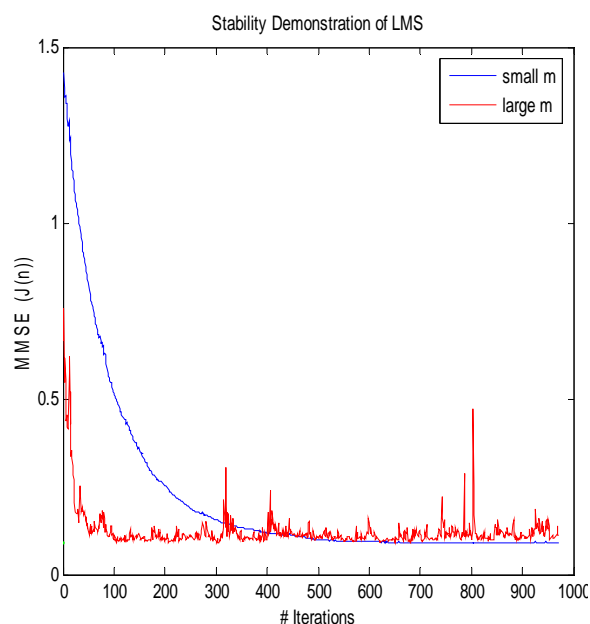
σε κάθε επανάληψη του αλγορίθμου η νέες τιμές των συντελεστών  $w$  του φίλτρου υπολογίζονται από τις αμέσως προηγούμενες τιμές συν κάποια διόρθωση που εξαρτάται από την σταθερά  $\mu$  και τη συνάρτηση κόστους. Έτσι ανάλογα με το πόσο μεγάλη είναι η τιμή της σταθεράς  $\mu$ , η οποία παίρνει τιμές από 0 έως 1, αυξάνεται και ο βαθμός με τον οποίο επηρεάζονται οι νέες τιμές από την συνάρτηση κόστους. Σε κάθε επανάληψη η συνάρτηση κόστους υπολογίζει το μέσο τετραγωνικό σφάλμα ανάμεσα στο λαμβανόμενο σήμα και το (εκτιμώμενο) επιθυμητό και η παράμετρος  $\mu$  καθορίζει το πόσο «γρήγορα» θα επιτρέψει ο αλγόριθμος την διόρθωση αυτή στις νέες τιμές. Έτσι αν  $\mu = 0$  τότε ο αλγόριθμος δεν λαμβάνει καθόλου υπόψη του την υπολογιζόμενη απόκλιση, ενώ για  $\mu=1$  θα εισάγει την διόρθωση αυτή σε ένα βήμα και θα φτάσει ακριβώς στις «επιθυμητές» τιμές, κάτι που στην πράξη δεν λειτουργεί όπως θα εξηγηθεί παρακάτω. Στην πρώτη περίπτωση βέβαια δεν έχουμε καθόλου προσαρμογή, ενώ στη δεύτερη η «προσαρμογή» θα γίνει με μια επανάληψη.



Γενικότερα, όσο μεγαλύτερο είναι το  $\mu$ , τόσο μεγαλύτερος είναι ο βαθμός προσαρμογής του αλγόριθμου και άρα ταχύτερη η σύγκλισή του στη βέλτιστη λύση Wiener-Hopf. Απαιτούνται δηλαδή λιγότερες επαναλήψεις προκειμένου να περάσει τις διορθώσεις που υποδεικνύονται από τη συνάρτηση κόστους στις τιμές των συντελεστών κάνοντας μεγαλύτερα βήματα προς την «σωστή κατεύθυνση». Το trade-off για αυτή την σχετικά γρηγορότερη εύρεση της βέλτιστης λύσης είναι η μεγαλύτερη ροπή προς την αστάθεια, όπως αναλύθηκε στην παράγραφο 4.5.2, και η μεγαλύτερη τάση του αλγόριθμου να επηρεάζεται από αρνητικά γεγονότα, δηλαδή να πολλαπλασιάζει τον θόρυβο στους συντελεστές του φίλτρου του ισοσταθμιστή, όπως αναλύθηκε στην παράγραφο 4.5.3. Στην παρακάτω γραφική φαίνεται η απόδοση του LMS τόσο από πλευράς σύγκλισης στη βέλτιστη λύση όσο και από πλευράς ταχύτητας για διάφορες τυπικές τιμές του  $\mu$ , για τις οποίες το σύστημα παραμένει ευσταθές:



Εικόνα 4.2

Εικόνα 4.3  $m=0.05$  &  $m=0.3$ ,  $L=6$ ,  $N=1000$ ,  $\sigma = 0.05$ 

Η αστάθεια που μελετήθηκε στην παράγραφο 4.4.2 προκαλείται όταν, εξαιτίας του μεγάλου βήματος, ο αλγόριθμος δεν μπορεί να συγκλίνει στην βέλτιστη λύση αλλά ταλαντώνεται γύρω από αυτή. Έτσι όταν το  $\mu$  είναι μεγάλο, ο αλγόριθμος συγκλίνει γρήγορα προς την βέλτιστη λύση που υποδεικνύεται από την συνάρτηση κόστους, αλλά όταν προσεγγίσει τη λύση αυτή, το  $\mu$  ίσως είναι

πολύ μεγάλο προκειμένου να πετύχει την ακρίβεια που απαιτείται για το επόμενο βήμα διόρθωσης. Το συγκεκριμένο πρόβλημα είναι εντονότερο σε αργά χρονικά μεταβαλλόμενα κανάλια όπου η ταλάντωση γίνεται για πολύ περισσότερο χρόνο. Αντίθετα για κανάλια με γρήγορες χρονικές μεταβολές, το μεγάλο  $\mu$  εξασφαλίζει γρήγορη σύγκλιση προς την νέα τιμή και ταυτόχρονα δεν προλαβαίνει να έχει προβλήματα αστάθειας για μεγάλο χρονικό διάστημα αφού το κανάλι θα αλλάξει γρήγορα και θα μπει πάλι σε διαδικασία «κίνησης» προς τη νέα τιμή. Η γραφική της εικόνας 4.3 απεικονίζει την σύγκλιση του LMS για μικρό (0.05) και μεγάλο  $\mu$  (0.3) στο ίδιο κανάλι, όπου είναι φανερό το πρόβλημα αστάθειας στη δεύτερη περίπτωση.

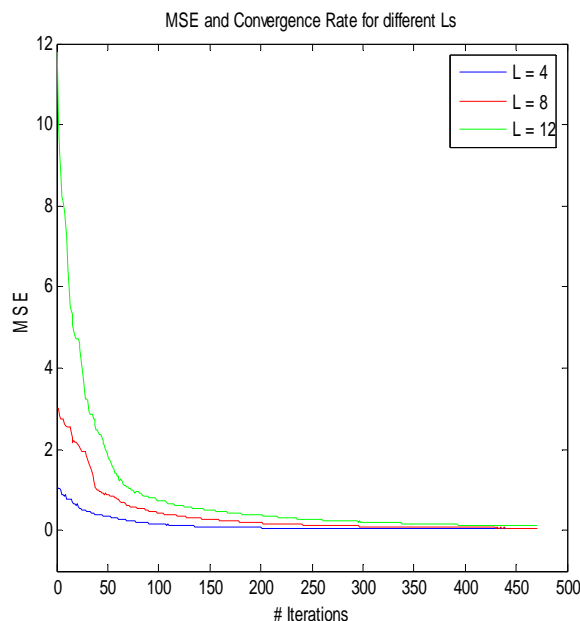
#### 4.5.2 Το μήκος του φίλτρου L

Το μήκος του ισοσταθμιστή είναι ο αριθμός των συντελεστών (coefficients) του φίλτρου, οι οποίοι ρυθμίζουν την λειτουργία του. Εν γένει, όσο μεγαλύτερος είναι ο αριθμός αυτός, τόσο καλύτερη είναι η δυνατότητα ιχνηλάτησης (tracking) των μεταβολών του καναλιού σε συνάρτηση με τον χρόνο. Αυτή η βελτίωση, όμως, έρχεται σε βάρος της ταχύτητας σύγκλισης προς την βέλτιστη λύση. Συγκεκριμένα, όπως αναλύθηκε στην παράγραφο 4.5.3, η ταχύτητα σύγκλισης επηρεάζεται από το λόγο  $\frac{\lambda_{\max}}{\lambda_{\min}}$  του πίνακα R,

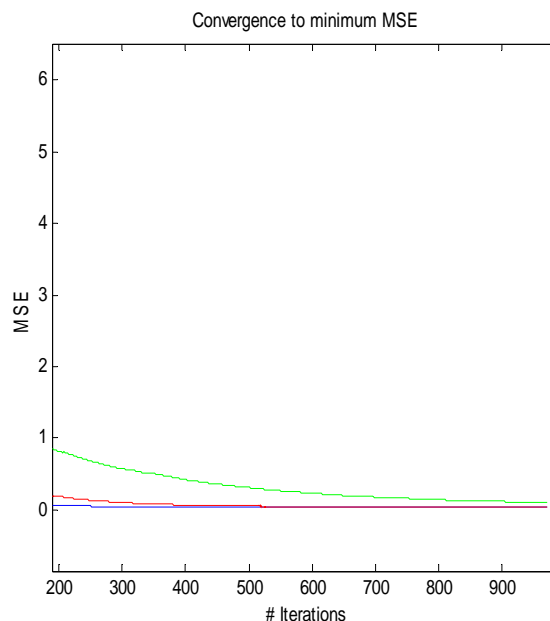
δηλαδή την ανομοιομορφία των ιδιοτιμών. Επειδή ο λόγος αυτός είναι μονότονα αύξουσα συνάρτηση του μήκους του φίλτρου L [6-p.175], δεν είναι δυνατόν να βελτιώσουμε την ανομοιομορφία των ιδιοτιμών αυξάνοντας το μήκος του φίλτρου.

Επίσης όσο μεγαλύτερο είναι το μήκος L του φίλτρου, τόσο μεγαλύτερη υπολογιστική ισχύς χρειάζεται για την ενημέρωση των συντελεστών αυτών σε κάθε επανάληψη (της ίδιας όμως τάξης μεγέθους), γεγονός που μπορεί να κάνει δύσκολη την υλοποίηση φίλτρου μεγάλου L σε συστήματα πραγματικού χρόνου.

Η παρακάτω γραφική δείχνει την επίδοση του LMS για διάφορα μήκη φίλτρου όπου φαίνεται η διαφορά στην ταχύτητα σύγκλισης και η κοινή σύγκλιση στο ίδιο mean square error (λόγω ίδιου  $\mu$ ).



Εικόνα 4.4

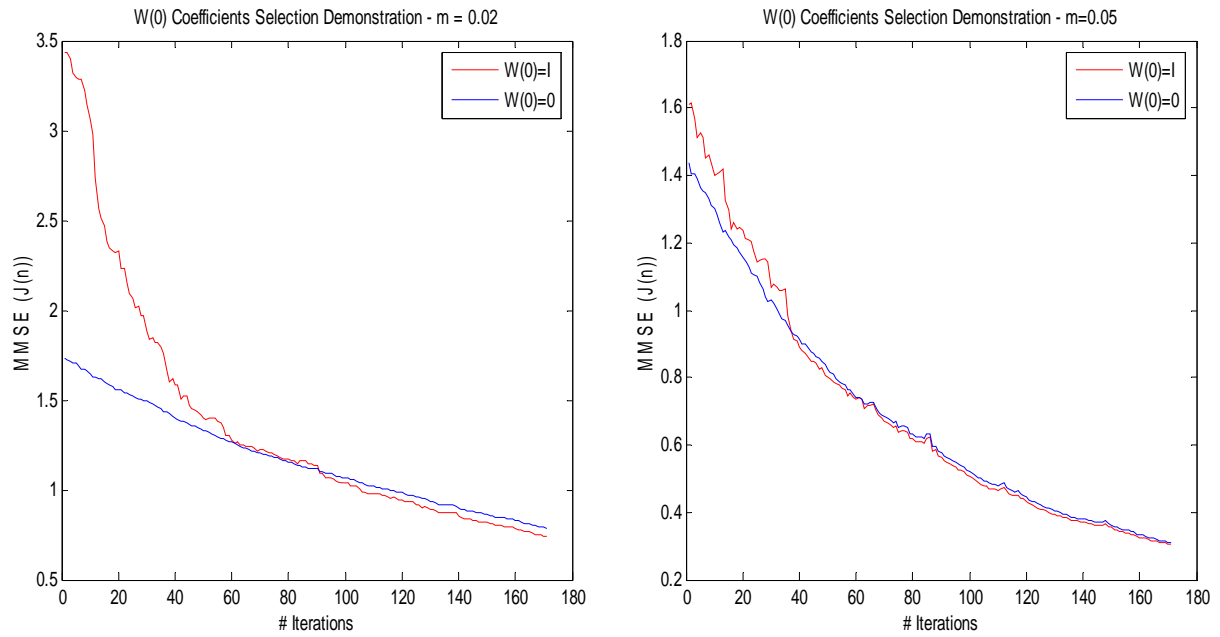


Εικόνα 4.5

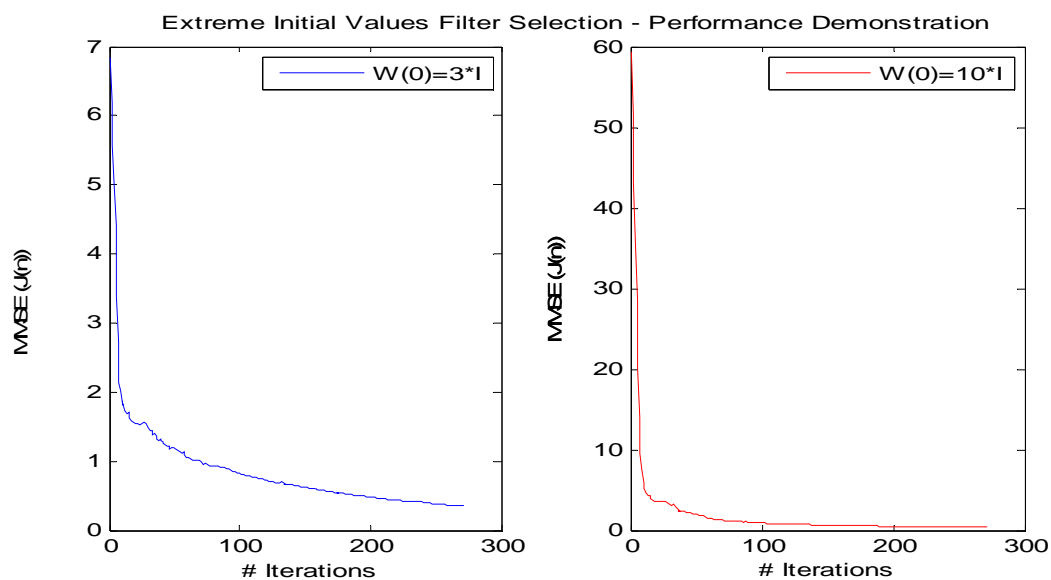
### 4.5.3 Η αρχικοποίηση των συντελεστών του φίλτρου

Η λειτουργία του LMS απαιτεί την αρχικοποίηση των συντελεστών του φίλτρου προκειμένου να έχουμε μια αρχική τιμή  $w(0)$  για την πρώτη επανάληψη του αλγόριθμου. Η επιλογή αυτής της τιμής δεν παίζει ρόλο στην απόδοση του αλγόριθμου, όπως θαδειχθεί στις γραφικές παρακάτω, εξαιτίας του ότι η συνάρτηση  $J(n)$  που ελαχιστοποιείται έχει μόνο ένα μέγιστο (unimodal) και ο αλγόριθμος είναι επαναληπτικός, άρα η επιρροή της στο τελικό αποτέλεσμα σύντομα θα εξαφανιστεί με την πάροδο των επαναλήψεων. Ο μόνος τρόπος να επηρεαστεί αρνητικά η απόδοση του αλγόριθμου από την επιλογή της αρχικής τιμής είναι το  $\mu$  να είναι πάρα πολύ μικρό και ο αριθμός των δειγμάτων επίσης μικρός με αποτέλεσμα η επιρροή της αρχικής τιμής να μην προλάβει να εξαφανιστεί και ο αλγόριθμος να μην συγκλίνει. Για τον αριθμό δειγμάτων που χρησιμοποιούμε στα πακέτα μας όμως ( $>500$ ), τέτοια περίπτωση δεν υπάρχει και άρα το πρόβλημα αυτό δεν μας απασχολεί.

Οι γραφικές που ακολουθούν δείχνουν την επίδραση της παραμέτρου αυτής για μικρά  $\mu$  (0.02 και 0.05 αντίστοιχα) και για μια συντηρητική επιλογή αριθμού δειγμάτων (1000). Ακόμα και έτσι, αρχικοποιώντας τους συντελεστές σε 0 και 1 αντίστοιχα, έχουμε ακριβώς την ίδια σύγκλιση με την δεύτερη να «πιάνει» την πρώτη σε μόλις 40 περίπου επαναλήψεις.

Εικόνα 4.6  $m=0.05$ ,  $L=6$ ,  $N=1000$ ,  $\sigma = 0.05$ 

Ακόμα και αν η αρχικοποίηση αυτή γίνει για τιμές  $3 \cdot I$  ή  $10 \cdot I$ , που θεωρούνται αρκετά υψηλές για συντελεστές του φίλτρου, ο αλγόριθμος καταφέρνει να εξισορροπήσει αυτό το «λάθος» μέσα σε 100 επαναλήψεις όπως φαίνεται στις γραφικές της εικόνας 4.7.

Εικόνα 4.7  $m=0.05$ ,  $L=9$ ,  $N=1000$ ,  $\sigma = 0.05$

Με βάση την παραπάνω ανάλυση εξάγεται το συμπέρασμα ότι η επιλογή της αρχικής τιμής των συντελεστών του φίλτρου δεν επηρεάζει την απόδοση του LMS για τις παραμέτρους ( $\mu$ ,  $L$ ) και τον αριθμό των bits που χρησιμοποιούμε στο σύστημα. Το γεγονός αυτό είναι επιθυμητό καθώς στο περιβάλλον λειτουργίας του συστήματος δεν είναι δυνατόν να έχουμε μια ασφαλή εκτίμηση των βέλτιστων τιμών του φίλτρου ώστε να τις εφαρμόσουμε και να πετύχουμε καλύτερη απόδοση. Βέβαια αν κάτι τέτοιο ήταν εφικτό, θα μας έδινε σίγουρα μια μικρή βελτίωση στην απόδοση καθώς η προσαρμογή θα ξεκινούσε από ένα καλύτερο σημείο. Συνεπώς επιλέγουμε να αρχικοποιούμε πάντα την μεταβλητή αυτή σε 0, που όπως δείχνει και η γραφική 4.6 συνήθως δημιουργεί τα λιγότερα προβλήματα, και έπειτα ο αλγόριθμος θα προσεγγίσει προσαρμοστικά την βέλτιστη λύση όπως εξηγήθηκε παραπάνω.

# 5

## ΑΛΓΟΡΙΘΜΟΣ RECURSIVE LEAST SQUARE

### 5.1 Εισαγωγή

Ο αλγόριθμος *Recursive Least Squares (RLS)* αποτελεί, όπως και ο LMS, μια μέθοδο υπολογισμού των συντελεστών  $W$  του προσαρμοστικού φίλτρου ισοστάθμισης. Σε αντίθεση όμως με τον LMS, χρησιμοποιεί πληροφορία και από προηγούμενα δείγματα που έχουν περάσει από το φίλτρο και όχι μόνο τα τρέχοντα δείγματα (tap weights) για την εκτίμηση του αντίστροφου του πίνακα αυτοσυσχέτισης. Για να μειώσει την συμβολή των παλαιότερων δειγμάτων στην εκτίμηση του πίνακα, εισάγει ένα συντελεστή  $\lambda$  που ονομάζεται «*forgetting factor*», και ο οποίος παίρνει τιμές από 0 έως 1. Για τις ανάγκες του συστήματος χρησιμοποιείται ο RLS με εκθετικό συντελεστή βάρους.

Ο ρόλος του συντελεστή  $\lambda$  φαίνεται στην παρακάτω συνάρτηση κόστους:

$$J(n) = \sum_{i=1}^n \lambda^{n-i} |e(i, n)|^2 \quad (5.1)$$

όπου  $e(i) = d(i) - y(i) = d(i) - w^H(n)x(i)$  δηλαδή η διαφορά ανάμεσα στο σήμα εξόδου του φίλτρου του ισοσταθμιστή  $y$  και στο επιθυμητό σήμα  $d$ , ενώ το  $i$  δηλώνει ότι αυτό γίνεται για όλα τα δείγματα  $0 < i < n$ , δηλαδή από την αρχή ως το τρέχον.

Παρατηρούμε ότι όταν το  $\lambda = 1$ , τότε όλα τα δείγματα  $i$  από  $i=1 \rightarrow n$  μετρώνε ισομερώς στον υπολογισμό του  $w$  οπότε έχουμε την απλή μέθοδο των ελαχίστων τετραγώνων (least squares). Αντίθετα αν το  $0 < \lambda < 1$  τότε η συμβολή των παλαιότερων δειγμάτων εξασθενεί με την πάροδο των επαναλήψεων, και μάλιστα αυτή η εξασθένιση είναι εκθετική, γι' αυτό και ο συντελεστής  $\lambda$  καλείται και *εκθετικός συντελεστής βάρους* (exponential weighting factor). Συνήθως το  $\lambda$  παίρνει τιμές κοντά στο 1, όπως θα δούμε στην παράγραφο 5.4 όπου αναλύονται οι επιλογές στη ρύθμιση των παραμέτρων του αλγορίθμου, οπότε το  $1-\lambda$  είναι η μνήμη του αλγορίθμου.

Ακολουθώντας την ίδια μεθοδολογία με τον LMS, υπολογίζουμε την παράγωγο της συνάρτησης κόστους της εξίσωσης 5.1 ως προς τους συντελεστές  $w$  (gradient).

$$\frac{\partial J(n)}{\partial W_i(n)} = \sum_{i=1}^n \lambda^{n-i} (-2E\{d(i)x(i)\} + 2E\{x(i)x^H(i)w(n)\}) \quad (5.2)$$

Σε αντίθεση τώρα όμως με τη μεθοδολογία του LMS, δεν θα χρησιμοποιήσουμε την μέθοδο gradient descent, αλλά θα υπολογίσουμε τις εκτιμήσεις του τύπου 5.2, δηλαδή την εκτίμηση  $E\{dx\}$  και  $E\{xx^H\}$  καθώς υπάρχει γνώση για τα όλα τα προηγούμενα δείγματα  $w(i)$ . Έτσι θέτουμε κατευθείαν την παράγωγο  $\frac{\partial J(n)}{\partial W_i(n)} = 0$  οπότε προκύπτει:

$$\Phi(n)\hat{w}(n) = z(n) \quad (5.3)$$

όπου  $\Phi(n)$  είναι ο  $M \times M$  πίνακας συσχέτισης (correlation):

$$\Phi(n) = \sum_{i=1}^n \lambda^{n-i} x(i)x^H(i) \quad (5.4)$$

και  $z(n)$  ο  $M \times 1$  πίνακας αυτοσυσχέτισης

$$z(n) = \sum_{i=1}^n \lambda^{n-i} x(i)d^*(i) \quad (5.5)$$

## 5.2 Ο αλγόριθμος Recursive Least Squares

Ο αλγόριθμος RLS υπολογίζει τους τύπους  $\Phi(n)$  και  $z(n)$  επαναληπτικά ακολουθώντας τους παρακάτω τύπους:

$$\Phi(n) = \lambda\Phi(n-1) + x(n)x^H(n) \quad (5.6)$$

$$z(n) = \lambda z(n-1) + d^*(n)x(n)$$

Για να υπολογίσουμε όμως τον πίνακα συντελεστών  $w(n)$  χρειάζεται να υπολογίσουμε τον αντίστροφο του  $\Phi$ , δηλαδή τον  $\Phi^{-1}(n)$ . Για να αποφύγουμε την απευθείας αντιστροφή που όπως έχει αναφερθεί κοστίζει υπολογιστικά, εφαρμόζουμε το Matrix Inversion Lemma [3-p.565] στον πίνακα συσχέτισης  $\Phi(n)$  της επαναληπτικής εξίσωσης 5.6 και θέτουμε  $P(n) = \Phi^{-1}(n)$ . οπότε υπολογίζουμε:

$$P(n) = \lambda^{-1}P(n-1) + \lambda^{-1}k(n)x(n) \quad (5.7)$$

$$\text{όπου} \quad k(n) = \frac{\lambda^{-1}P(n-1)x(n)}{1 + \lambda^{-1}x^H(n)P(n-1)x(n)} \quad (5.8)$$

$$\text{Έτσι} \quad w(n) = \hat{w}(n-1) + k(n)(d^*(n) - x^H(n)\hat{w}(n-1)) = \hat{w}(n-1) + k(n)\xi^*(n) \quad (5.9)$$

$$\text{Όπου} \quad \xi(n) = d(n) - \hat{w}^H(n-1)x(n) \quad (5.10)$$

Πριν όμως ξεκινήσει η παραπάνω επαναληπτική διαδικασία, είναι αναγκαία η αρχικοποίηση κάποιων παραμέτρων, καθώς στον RLS η αρχικοποίηση παίζει σημαντικό ρόλο. Συγκεκριμένα οι τιμές που πρέπει να αρχικοποιηθούν είναι το  $P(0)$  της εξίσωσης 5.7 και το διάνυσμα των συντελεστών του φίλτρου  $w(0)$  της εξίσωσης 5.9. Οι αρχικές τιμές που δίνουμε [3-p.570] είναι:

$$P(0) = \delta^{-1}I \quad (5.11)$$

όπου  $\delta$  ένας θετικός αριθμός με μικρή τιμή, η επιλογή του οποίου αναλύεται στην παράγραφο 5.4. Η επιλογή του  $w(0)$  γίνεται έχοντας υπόψη ότι, όπως και στον LMS η προσαρμογή δεν ξεκινάει κατευθείαν από το πρώτο δείγμα, αλλά πρέπει πρώτα να «γεμίσει» το φίλτρο, δηλαδή αν το μήκος του ισοσταθμιστή είναι  $L$ , η προσαρμογή θα ξεκινήσει μετά το  $L^o$  δείγμα. Έτσι επιλέγουμε:

$$\hat{w}(0) = 0 \quad (5.12)$$

Μετά την απαραίτητη αυτή αρχικοποίηση, που ονομάζεται *soft-constrained initialization* ακολουθεί η επαναληπτική διαδικασία του RLS η οποία περιλαμβάνει τις εξισώσεις 5.7-5.10 με την παρακάτω σειρά:

$$k(n) = \frac{\lambda^{-1}P(n-1)x(n)}{1 + \lambda^{-1}x^H(n)P(n-1)x(n)} \quad (5.13)$$

$$\xi(n) = d(n) - \hat{w}^H(n-1)x(n) \quad (5.14)$$

$$w(n) = \hat{w}(n-1) + k(n)(d^*(n) - x^H(n)\hat{w}(n-1)) = \hat{w}(n-1) + k(n)\xi^*(n) \quad (5.15)$$

$$P(n) = \lambda^{-1}P(n-1) + \lambda^{-1}k(n)x(n) \quad (5.16)$$

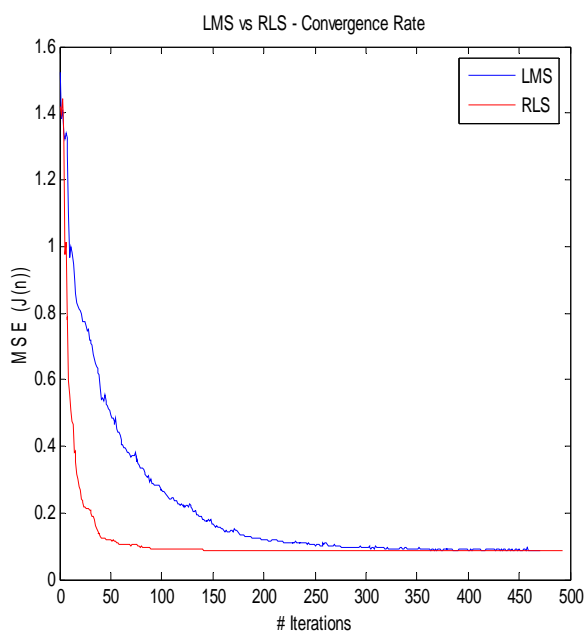
Οι παραπάνω τέσσερις εξισώσεις 5.13-5.16 αποτελούν την επαναληπτική διαδικασία που εφαρμόζει ο RLS αλγόριθμος για τον προσαρμοστικό υπολογισμό των συντελεστών  $w$  του φίλτρου ισοστάθμισης.



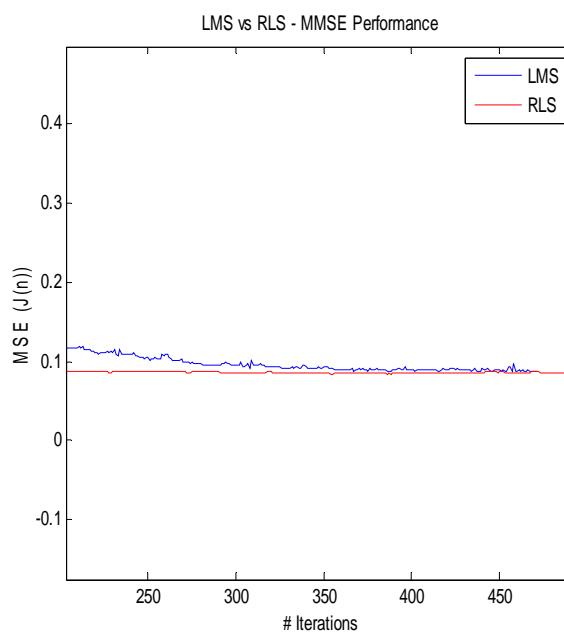
## 5.3 Αξιολόγηση του αλγόριθμου RLS

### 5.3.1 Ρυθμός Σύγκλισης

Ο κύριος λόγος που επελέγη ο RLS ως εναλλακτικός του LMS αλγόριθμος είναι ο γρηγορότερος ρυθμός σύγκλισης, δηλαδή ο μικρότερος αριθμός επαναλήψεων που απαιτούνται για να βρεθεί η βέλτιστη λύση. Η σύγκλιση του RLS στην βέλτιστη λύση γίνεται σε περίπου  $2L$  επαναλήψεις [3-p.580], όπου  $L$  ο αριθμός των χρησιμοποιούμενων tap-weights, απόδοση κατά μία τάξη καλύτερη αυτής του LMS [3-p.587] πράγμα που φαίνεται και στις γραφικές της εικόνας 5.1. Η προσομοίωση έγινε και για τους δύο αλγόριθμους για το ίδιο σήμα και ο αριθμός των taps που χρησιμοποιήθηκε είναι 6. Παρατηρούμε ότι ο RLS έπεσε κάτω από το 0.2 σε περίπου 10 επαναλήψεις, ενώ ο LMS πέτυχε την ίδια απόδοση σε πάνω από 100 επαναλήψεις.



Εικόνα 5.1



Εικόνα 5.2

Η περιστροφή (rotation) που δημιουργείται από τον πολλαπλασιασμό με τον αντίστροφο  $\Phi^{-1}(n)$ , αφαιρεί την εξάρτηση του αποτελέσματος από της ιδιοτιμές (eigenvalues), οπότε και την εξάρτηση του ρυθμού σύγκλισης του αλγόριθμου από το κανάλι και την ισχύ μετάδοσης, όπως εξηγήθηκε και στην παράγραφο 4.4 του προηγούμενου κεφαλαίου για τον LMS. Στην

πραγματικότητα, ξεκινάμε με μια υποθετική τιμή  $\Phi = w^2 I$ . Αντιστρέφοντας τον πίνακα  $w^2 I$ , δεν έχουμε περιστροφή (λόγω του  $I$ ) και έτσι δεν μειώνεται η ανακολουθία (disparity) των ιδιοτιμών. Κατά την σύγκλιση, αν η εκτίμησή μας για τον αντίστροφο πίνακα συσχέτισης συγκλίνει στην πραγματική τιμή, τότε η εξάρτηση αυτή εξαφανίζεται. Έτσι ξεκινώντας με την αρχική τιμή  $w^2 I$  και υπολογίζοντας επαναληπτικά μια εκτίμηση του  $\Phi$  έχει σαν αποτέλεσμα να υπολογίζουμε τον παρακάτω τύπο για την συνάρτηση αυτοσυσχέτισης:

$$\Phi'(n) = \sum_{i=1}^n \lambda^{n-i} x(i)x^H(i) + w^2 I, n = 0, 1, \dots$$

στην οποία είναι  $E\{\Phi'(n)\} \neq E\{\Phi(n)\}$  οπότε η αρχικοποίηση που εισήχθη αποκρίνει την λύση από την τιμή  $\Phi^{-1}(n)z(n)$  που όπως έχουμε πει είναι πολύπλοκη υπολογιστικά. Για μεγάλο  $n$  στον τύπο παραπάνω το αποτέλεσμα της αρχικοποίησης αυτής ουσιαστικά δεν παίζει κανένα ρόλο και η  $\Phi^{-1}(n)$  συγκλίνει στην βέλτιστη λύση. [6-p.320].

### 5.3.2 Σύγκλιση

Για τις ανάγκες της ανάλυσης [3-p.575] που θα ακολουθήσει στις επόμενες δύο παραγράφους θεωρούμε ότι η επιθυμητή απόκριση:

$$d(n) = e(n) + w^H x(n) \quad (5.17)$$

Αντικαθιστούμε την 5.17 στην έκφραση του  $z(n)$  της εξίσωσης 5.5 και έχουμε:

$$\begin{aligned} z(n) &= \sum_{i=1}^n x(i)x^H(i)w_o + \sum_{i=1}^n x(i)e_o^*(i) \\ &= \Phi(n)w_o + \sum_{i=1}^n x(i)e_o^*(i) \end{aligned} \quad (5.18)$$

Έτσι η εξίσωση 5.3  $\Phi(n)\hat{w}(n) = z(n) \Leftrightarrow \hat{w}(n) = \Phi^{-1}(n)z(n), n \geq L$  γράφεται:

$$\begin{aligned} \hat{w}(n) &= \Phi^{-1}(n)\Phi(n)w_o + \Phi^{-1}(n)\sum_{i=1}^n x(i)e_o^*(i) \\ &= w_o + \Phi^{-1}(n)\sum_{i=1}^n x(i)e_o^*(i) \end{aligned} \quad (5.19)$$

Χρησιμοποιώντας την ιδιότητα των τυχαίων μεταβλητών  $E[x]=E[E[x|y]]$ , όπου  $E[x|y]$  η δεσμευμένη πιθανότητα του  $x$  σε μια άλλη μεταβλητή  $y$ , προκύπτει για την εκτίμηση των συντελεστών του φίλτρου  $w$ :

$$\begin{aligned} E[\hat{w}(n)] &= w_o + E[\Phi^{-1}(n) \sum_{i=1}^n x(i)e_o^*(i)] \\ &= w_o + E[E[\Phi^{-1}(n) \sum_{i=1}^n x(i)e_o^*(i) | x(i), i=1,2,\dots,n]]] \end{aligned} \quad (5.20)$$

Αν λάβουμε υπόψη ότι i) ο πίνακας  $\Phi(n)$  ορίζεται μοναδικά από την ακολουθία εισόδου  $x(1), x(2), \dots, x(n)$ , ii) το σφάλμα  $e(n)$  είναι ανεξάρτητο από το σήμα εισόδου  $x(i)$  και έχει μέση τιμή 0, η σχέση 5.20 απλοποιείται στην:

$$E[\hat{w}(n)] = w_o, n \geq L \quad (5.21)$$

η οποία αποδεικνύει ότι εκτίμηση των συντελεστών του RLS συγκλίνει πάντα στην μέση τιμή, αρκεί ο αριθμός των επαναλήψεων  $n$  να είναι μεγαλύτερος από το μέγεθος του φίλτρου, αντίθετα με τον LMS που πρέπει να φτάσει  $n \rightarrow \infty$ . Επίσης ο RLS κατορθώνει να επιτύχει μικρότερο μέσο τετραγωνικό σφάλμα [3-p.580] στο τέλος της προσαρμογής σε σχέση με τον LMS (βλ εικόνα 5.2) ενώ θεωρητικά αν ο αριθμός των επαναλήψεων τείνει στο άπειρο, μπορεί να πετύχει μηδενικό μέσο τετραγωνικό σφάλμα, κάτι που δεν κατορθώνει ο LMS.

### 5.3.3 Πολυπλοκότητα

Τα παραπάνω δύο πλεονεκτήματα του RLS δεν έρχονται βέβαια χωρίς κόστος για το σύστημα. Η υπολογιστική πολυπλοκότητα του αλγορίθμου είναι αρκετά μεγαλύτερη συγκριτικά με τον LMS, παρόλο που η εφαρμογή του Inversion Matrix Lemma την περιορίζει σημαντικά. Συγκεκριμένα η συνολική πολυπλοκότητα είναι της τάξης  $O(L^2)$  σε σχέση με το  $O(L)$  του LMS. Ο αριθμός των πράξεων που απαιτούνται για τον υπολογισμό των βημάτων του αλγορίθμου (εξισώσεις 5.13 έως 5.16) είναι  $4L^2+4L$  πολλαπλασιασμοί και  $3L^2+L$  προσθέσεις [6-p.319] όπως φαίνεται αναλυτικά στον πίνακα 5.1. Είναι δυνατόν, με εκμετάλλευση της συμμετρίας του  $\Phi^{-1}(n)$  να μειωθούν περαιτέρω οι απαιτούμενες πράξεις [10], όμως ακόμα και έτσι η τάξη μεγέθους παραμένει στο  $O(L^2)$ .

Βήμα αλγόριθμου	$\mathbf{O}_x$	$\mathbf{O}_+$
1α) αριθμητής	$L^2$	$L(L-1)$
β) παρανομαστής	$L^2+L$	$L(L-1)+L$
γ) διαίρεση	$L$	$0$
2	$L$	$L$
3	$L$	$L$
4	$L^2+L^2$	$L(L-1)+L^2$
Πολυπλοκότητα	$O_x(4L^2+4L)$	$O_+(3L^2+L)$
Πίνακας 5.1 – Πολυπλοκότητα RLS [6]		

### 5.3.4 Ευστάθεια

Το δεύτερο μειονέκτημα του RLS είναι η ροπή προς την αστάθεια που παρουσιάζει, παρόλη την εξαφάνιση της εξάρτησης από τις ιδιοτιμές που παρουσιαζόταν στον LMS. Η αστάθεια του RLS εξαρτάται από την επιλογή της παραμέτρου  $\lambda$ , και γι' αυτό θα εξετασθεί στην επόμενη παράγραφο.

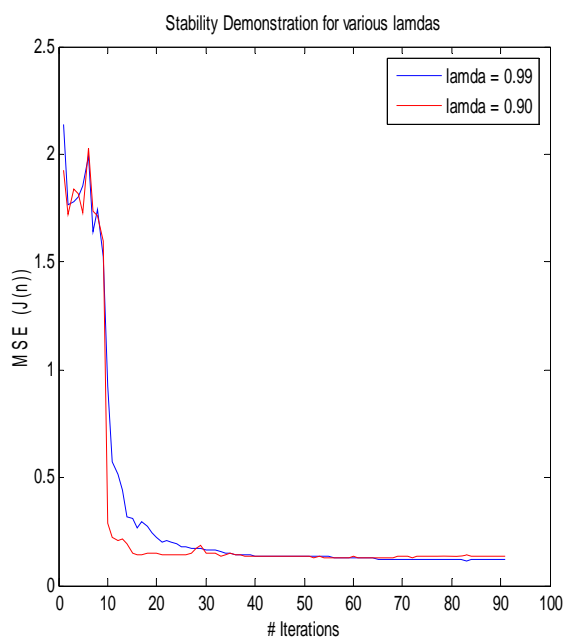
## 5.4 Επιλογή Παραμέτρων του RLS

### 5.4.1. Επιλογή του $\lambda$

Η παράμετρος  $\lambda$  περιέχεται στους τύπους 5.13 και 5.16 του αλγόριθμου RLS και ονομάζεται *forgetting factor*. Καθορίζει τον βαθμό με τον οποίο οι παλαιότερες τιμές των συντελεστών του φίλτρου ισοστάθμισης επηρεάζουν τη διαμόρφωση των νέων, δηλαδή επηρεάζει την απόδοσή του ως προς το tracking ενώ, όπως θα δούμε και από τις γραφικές που ακολουθούν, δεν επηρεάζει την ταχύτητα σύγκλισης του αλγόριθμου. Έχει τον αντίστοιχο ρόλο της παραμέτρου  $\mu$  στον LMS και παίρνει και αυτό τιμές  $0 < \lambda \leq 1$ . Για  $\lambda=1$ , οι παλαιότερες τιμές δεν παίζουν κανένα ρόλο στην εξαγωγή των νέων τιμών των συντελεστών επιλογή που χρησιμοποιείται μόνο σε χρονικά αμετάβλητα κανάλια. Όσο πιο μικρή είναι η τιμή του  $\lambda$ , τόσο καλύτερη είναι η ικανότητα παρακολούθησης (tracking) του αλγόριθμου, καθώς λιγότερες προηγούμενες τιμές  $w(n-i)$  λαμβάνονται υπόψη προκειμένου να εκτιμηθούν οι νέες τιμές  $w(n)$  του φίλτρου στον τύπο 5.10 και άρα τόσο πιο ευέλικτη είναι η προσαρμογή σε

μεγάλες αποκλίσεις. Παράλληλα όμως, μικρό  $\lambda$  μπορεί να οδηγήσει σε αστάθεια, η οποία είναι μαζί με την υψηλή πολυπλοκότητα τα δύο σημαντικότερα μειονεκτήματα του RLS. Για να αποφύγουμε την αστάθεια, πρακτικά επιλέγουμε την τιμή του  $\lambda$  σίγουρα μεγαλύτερη από 0.8 και αν το κανάλι δεν είναι γρήγορα χρονικά μεταβαλλόμενο μια τιμή πολύ κοντά στο 1 (ακόμα και 0.999).

Στις γραφικές της εικόνας 5.3α και 5.3β βλέπουμε την σύγκλιση του αλγόριθμου για διάφορες τιμές του  $\lambda$  όπου φαίνεται ότι η ταχύτητα σύγκλισης (δηλαδή ο αριθμός των επαναλήψεων έως την σύγκλιση) είναι ανεξάρτητη από την τιμή του  $\lambda$  καθώς επίσης και η αστάθεια που προκαλείται για τιμές μικρότερες του 0.9 (όπως φαίνεται στις γραφικές της εικόνας 5.3β), η οποία μάλιστα μεγαλώνει όσο μικραίνει το  $\lambda$ .



Εικόνα 5.3α

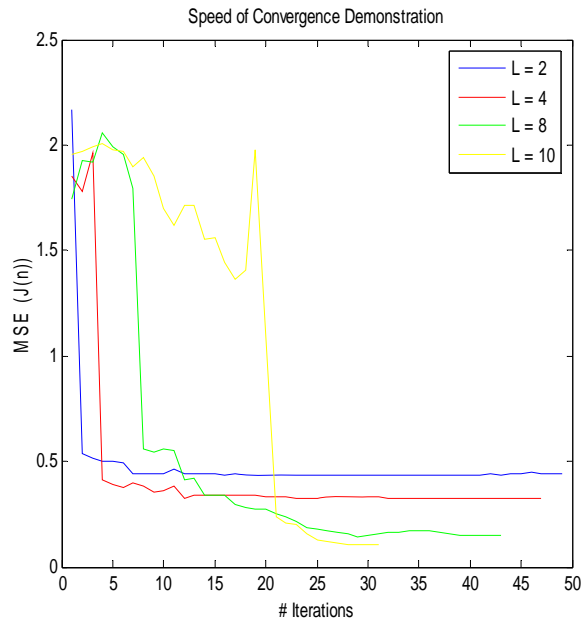
#### 5.4.2. Επιλογή του $\delta$

Η δεύτερη παράμετρος του RLS που ρυθμίζει την λειτουργία του είναι ο παράγοντας  $\delta$  που χρησιμοποιείται στην εξίσωση 5.11 για την αρχικοποίηση του αντίστροφου πίνακα συσχέτισης  $P(0)$ . Όπως αναφέρθηκε και στην παράγραφο 5.2, η τιμή του  $\delta$  πρέπει να είναι πολύ μικρή και πρακτικά να είναι

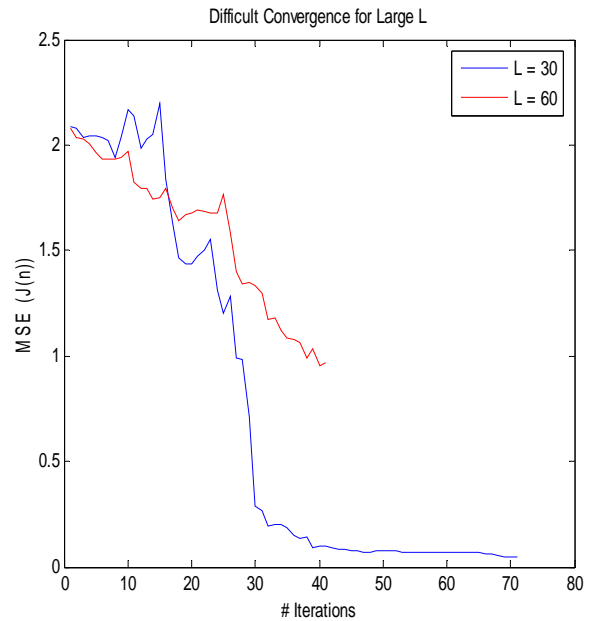
μικρή σε σχέση με το  $0.01 \sigma_u^2$ , όπου  $\sigma_u^2$  είναι η διακύμανση του δείγματος  $u(n)$ . Το αποτέλεσμα αυτό δεν προκύπτει από κάποιο μαθηματικό μοντέλο αλλά από πρακτική εμπειρία και στατιστική ανάλυση της soft-constrained αρχικοποίησης (δηλ  $P(0)$  και  $w(0)$ ) των εξισώσεων 5.11, 5.12) του αλγόριθμου που έγινε από τους Hubing και Alexander. Η παραπάνω επιλογή του λειτουργεί καλά για σήμα με υψηλό SNR. Για χαμηλό SNR, η σταθερά  $\delta$  πρέπει να είναι υψηλή προκειμένου να λειτουργήσει ικανοποιητικά ο αλγόριθμος. Πάντως όσο αυξάνει το μήκος  $N$  της ακολουθίας εισόδου, δηλαδή το μήκος του πακέτου για το σύστημά μας, η αρχικοποίηση του  $P(0)$  και άρα η επιλογή του  $\delta$  δεν παίζουν σημαντικό ρόλο στην επίδοση του αλγόριθμου. [3-p 570]

### 5.4.3 Επιλογή του μήκους του ισοσταθμιστή $L$

Το μήκος του ισοσταθμιστή RLS παίζει σημαντικό ρόλο στην ικανότητα παρακολούθησης του αλγόριθμου όπως αναφέρθηκε και στην εισαγωγή για τα προσαρμοστικά φίλτρα. Όσο πιο πολλούς συντελεστές (tap-weights) έχει ένα φίλτρο, τόσο καλύτερα μπορεί να παρακολουθήσει τις χρονικές μεταβολές του καναλιού. Αν το κανάλι δεν μεταβάλλεται χρονικά σε σημαντικό βαθμό τότε μια μικρή τιμή για το  $L$  (π.χ 2 ή 4) είναι αρκετή. Η αύξηση του  $L$ , όμως επηρεάζει την απόδοση του RLS και συγκεκριμένα την ταχύτητα σύγκλισής του. Όπως φαίνεται και στις γραφικές της εικόνας 5.5, όπου παρουσιάζεται η σύγκλιση για τέσσερα διαφορετικά  $L$ , ο αλγόριθμος συγκλίνει σε περίπου  $2L$  επαναλήψεις. Παράλληλα όμως, η αύξηση του  $L$  επιφέρει καλύτερη ισοστάθμιση καθώς το μέσο τετραγωνικό σφάλμα πέφτει σε πολύ μικρότερες τιμές. Το συμπέρασμα είναι ότι σε περιπτώσεις όπου το προς επεξεργασία σήμα δεν είναι πολύ αλλοιωμένο από το κανάλι, μας συμφέρει η χρήση φίλτρου με λίγους συντελεστές προκειμένου να εκμεταλλευτούμε την γρήγορη σύγκλιση, ενώ σε περιπτώσεις όπου το σήμα είναι αλλοιωμένο και χρειαζόμαστε την καλύτερη δυνατή ισοστάθμιση, χρησιμοποιούμε φίλτρο περισσότερων συντελεστών.



Εικόνα 5.5



Εικόνα 5.6

Για επιλογές του  $L$  με μεγάλες τιμές, παρατηρούμε στην εικόνα 5.6 ότι η ισοστάθμιση αργεί πάρα πολύ αλλά φτάνει μια τιμή για το μέσο τετραγωνικό σφάλμα που πλησιάζει το μηδέν. Για πολύ μεγάλα  $L$ , όμως, ακόμα και αυτό το πλεονέκτημα χάνεται καθώς ο αλγόριθμος συγκλίνει τόσο αργά που δεν καταφέρνει να φτάσει σε μια ικανοποιητική τιμή για το μέσο τετραγωνικό σφάλμα.

# 6

## ΑΛΓΟΡΙΘΜΟΣ CONSTANT MODULUS

### 6.1 Blind Ισοσταθμιστές

Ο αλγόριθμος Constant Modulus ανήκει στην κατηγορία των blind ισοσταθμιστών, δηλαδή δεν χρησιμοποιεί ακολουθία εκμάθησης για την εκπαίδευση του ισοσταθμιστή στο κανάλι. Οι blind ισοσταθμιστές χρησιμοποιούνται, όπως αναφέρθηκε και στο κεφάλαιο 3, όταν το σήμα εισόδου δεν είναι γνωστό και άρα είναι αδύνατο να βρούμε τη συνάρτηση κόστους όπως έγινε στους αλγόριθμους LMS και RLS. Στην περίπτωση αυτή χρησιμοποιούνται υποθέσεις σχετικά με κάποιες ιδιότητες του σήματος εισόδου με σκοπό να ελαχιστοποιηθεί η συνάρτηση κόστους. Συγκεκριμένα η συνάρτηση κόστους του Constant Modulus «ποινικοποιεί» παρεκκλίσεις του modulus (π.χ πλάτος) του σήματος ισοστάθμισης από μια καθορισμένη τιμή [12-p.1928]. Όπως θα φανεί παρακάτω, κάτω από ιδανικές συνθήκες, είναι δυνατόν να ελαχιστοποιηθεί η συνάρτηση και να έχουμε τέλεια ισοστάθμιση.

### 6.2 Ο αλγόριθμος Constant Modulus

Στον CMA η ιδιότητα που χρησιμοποιείται είναι η constant modularity του σήματος εισόδου σε σχέση με το σήμα εξόδου, με βάση την οποία προσαρμόζονται οι συντελεστές του φίλτρου. Η συνάρτηση κόστους που προσπαθεί να ελαχιστοποιήσει ο CMA είναι:

$$J_{CM} = E\{|e(n)|^2\} = E\{|R_2 - y^2(n)|^2\} \quad (6.1)$$

όπου  $R_2 = \frac{E\{|x(n)|^4\}}{E\{|x(n)|^2\}}$  είναι η επιθυμητή τιμή του πλάτους. Ο δείκτης 2 στον

τύπο του  $R_2$  προέρχεται από την καταγωγή του CMA από μια ευρύτερη οικογένεια blind αλγορίθμων, των Bussgang αλγορίθμων, εκ των οποίων ο πιο αποτελεσματικός είναι ο Godard, με συνάρτηση:



$$J_{(n)} = E\{|e(n)|^2\} = E\{|R_p - y^p(n)|^2\} \text{ όπου } R_p = \frac{E\{|x(n)|^{2p}\}}{E\{|x(n)|^2\}}.$$

Με βάση την εξίσωση 6.1, ο αλγόριθμος έχει σχεδιαστεί να «τιμωρεί» παρεκκλίσεις της εξόδου  $y(n)$  από τα χαρακτηριστικά του constant modulus. Η τέλεια ισοστάθμιση επιτυγχάνεται όταν η παράμετρος  $R_2$  επιλεγθεί έτσι ώστε η τιμή του  $J(n)$  να είναι μηδενική. Έτσι η προσαρμογή των συντελεστών του φίλτρου του ισοσταθμιστή γίνεται με βάση την παρακάτω εξίσωση:

$$\hat{w}(n+1) = \hat{w}(n) + \mu x(n)e^*(n) \quad (6.2)$$

όπου  $\mu$  το γνωστό step size που χρησιμοποιείται σε όλους τους προσαρμοστικούς αλγόριθμους,  $x(n)$  το τμήμα του σήματος εισόδου στο φίλτρο του ισοσταθμιστή και  $e(n)$  το σφάλμα που δίνεται από τον τύπο:

$$e(n) = y(n)(R_2 - |y(n)|^2) \quad (6.3)$$

## 6.3 Αξιολόγηση του αλγόριθμου

### 6.3.1 Αξιοποίηση Καναλιού

Στους αλγόριθμους LMS και RLS χρησιμοποιείται ακολουθία εκπαίδευσης προκειμένου να εκτιμηθεί το κανάλι και να ρυθμιστούν ανάλογα οι συντελεστές του φίλτρου, έτσι ώστε από αυτό το φίλτρο να περάσει η πληροφορία και να ανακατασκευαστεί το σήμα που εστάλη. Αυτή η πληροφορία αποστέλλεται σε κάθε πακέτο, με αποτέλεσμα αυτό να μεγαλώνει και άρα να πέφτει ο ρυθμός μετάδοσης. Αν για παράδειγμα το μέγεθος του χρησιμοποιούμενου πακέτου είναι 10000bits και προσθέτουμε 1000 επιπλέον bits για training sequence, ρίχνουμε κατευθείαν το bit error rate κατά το ένα δέκατο και στην πράξη λόγω της επεξεργασίας για την ρύθμιση του φίλτρου και το φιλτράρισμα της πληροφορίας κατά το διπλάσιο. Αν και σε πραγματικά ασύρματα συστήματα επικεφαλίδες στο πακέτο χρησιμοποιούνται για διάφορους σκοπούς, στο σύστημα WFTP που αναπτύξαμε, η επικεφαλίδα περιλαμβάνει μόνο την ακολουθία εκμάθησης για την ισοστάθμιση. Για το λόγω αυτό δοκιμάστηκε και ο CMA προκειμένου να γλιτώσουμε αυτές τις απώλειες σε bit error rate.

### 6.3.2 Σύγκλιση

Επειδή με τον CMA δεν χρησιμοποιείται ακολουθία εκπαίδευσης, οποιαδήποτε ακολουθία του σήματος εξόδου που παρουσιάζει σταθερή μετατόπιση φάσης σε σχέση με το σήμα εισόδου, θεωρείται σωστή στον δέκτη. Έτσι η συνάρτηση κόστους του αλγόριθμου αυτού δεν θα έχει μοναδικό ελάχιστο, αλλά από ένα τοπικό ελάχιστο για κάθε μετατόπιση φάσης. Συνεπώς για έναν ισοσταθμιστή μήκους  $L$ , θα έχουμε  $L^2$  ελάχιστα, καθώς υπάρχουν  $L^2$  μετατοπίσεις φάσης για τις οποίες υπάρχει λύση στην συνάρτηση κόστους. Πρακτικά, όμως, δεν είναι όλες οι  $L^2$  λύσεις αποδεκτές. Για παράδειγμα αν η ακολουθία εισόδου είναι διαμορφωμένη με διαφορικό PSK  $M$  συμβόλων, μόνο  $M$  διαφορετικές ακολουθίες θα είναι αποδεκτές στον δέκτη. Έτσι μόνο  $M$  από τις  $L^2$  ελάχιστα είναι αποδεκτά ως λύσεις, τα οποία και ονομάζονται *γενικά ελάχιστα* (global minima) ενώ τα υπόλοιπα  $L^2-M$  ονομάζονται *τοπικά ελάχιστα* (local minima). Η σύγκλιση σε γενικά ελάχιστα εξασφαλίζεται από την συνάρτηση κόστους του CMA μόνο απουσία θορύβου. Όταν υπάρχει θόρυβος καναλιού είναι πιθανή η σύγκλιση σε τοπικό ελάχιστο, που είναι σαφώς μια ανεπιθύμητη εξέλιξη. Το αν ο αλγόριθμος θα συγκλίνει σε κάποιο γενικό ή τοπικό ελάχιστο εξαρτάται, σε μεγάλο βαθμό, από την αρχικοποίηση του αλγόριθμου, όπως θα δειχθεί στην παράγραφο 6.4, πράγμα που κάνει την εν λόγω παράμετρο την πλέον σημαντική για την απόδοση του CMA.

### 6.3.3 Δυνατότητα Ανάνηψης

Ένα ακόμα πλεονέκτημα του CMA, πέρα από την έλλειψη επιπρόσθετης πληροφορίας για training sequence, είναι η ικανότητα προσαρμογής του ισοσταθμιστή ανεξάρτητα από την διαδικασία ανάκτησης του φέροντος. Μια μετατόπιση της συχνότητας του φέροντος κατά  $\Delta f$  προκαλεί μετατόπιση της συχνότητας του σήματος στην έξοδο του ισοσταθμιστή σύμφωνα με τον τύπο:

$$y(n) = |y(n)| e^{j2\pi\Delta f n + \phi(n)} \quad (6.5)$$

Επειδή όμως η συνάρτηση κόστους του αλγόριθμου δεν εξαρτάται από την πιθανή μετατόπιση φάσης του φέροντος (βλ εξίσωση 6.1), ο υπολογισμός των παραμέτρων προσαρμογής γίνεται ανεξάρτητα από την μετατόπιση αυτή.

### 6.3.4 Ρυθμός Σύγκλισης

Ο ρυθμός σύγκλισης του LMS, καθορίζεται από την επιλογή της παραμέτρου  $\mu$ , η οποία περιορίζεται τόσο από άνω όσο και από κάτω όρια σύμφωνα με τις εξισώσεις 4.11 και 4.12. Αυτό συμβαίνει γιατί η καμπυλότητα της επιφάνειας της συνάρτησης κόστους έχει μοναδική καμπυλότητα. Κάτι τέτοιο δεν μπορεί να ειπωθεί για τον CMA καθώς πάντα υπάρχει η πιθανότητα σύγκλισης σε τοπικό ελάχιστο που όπως είπαμε θα οδηγήσει σε πολύ αργή σύγκλιση. Ο ρυθμός σύγκλισης λοιπόν του CMA παρουσιάζει διακυμάνσεις και η μελέτη της σύγκλισης γίνεται ανά περιοχές, κοντά και μακριά από τα ελάχιστα [12-1938].

## 6.4 Επιλογή Παραμέτρων του CMA

### 6.4.1 Ο ρόλος της αρχικοποίησης

Επειδή η συνάρτηση κόστους του CMA έχει πολλαπλά ελάχιστα, όπως αναφέρθηκε στην παράγραφο 6.3.2, η αρχικοποίηση παίζει σημαντικό ρόλο για την απόδοση του αλγορίθμου αλλά και για το χρόνο σύγκλισης. Συγκεκριμένα η αρχικοποίηση καθορίζει αν ο αλγόριθμος θα συγκλίνει σε κάποιο γενικό (global) ή τοπικό (local) ελάχιστο. Σύγκλιση των τιμών των συντελεστών του φίλτρου σε τοπικό ελάχιστο οδηγεί σε πολύ αργή σύγκλιση του αλγορίθμου και καταστρέφει την απόδοσή του [12.p-1932].

Ο CMA μπορεί να συγκλίνει πάντα σε γενικό ελάχιστο [11-p.18], που είναι και το επιθυμητό, αρκεί να τηρούνται κάποιες προϋποθέσεις. Για τις ανάγκες της ανάλυσης που ακολουθεί [10-p.15], προηγούνται κάποιοι χρήσιμοι ορισμοί:

#### Συνολική Απόκριση Συστήματος (Overall System Response)

Αν  $\{c(n)\}$  είναι η κρουστική απόκριση του καναλιού και  $\{f(n): N1 \leq n \leq N2\}$  η κρουστική απόκριση του ισοσταθμιστή, η συνολική απόκριση του συστήματος είναι η συνέλιξη των δύο, δηλαδή:

$$g(n) = c(n) * f(n) \quad \text{ή} \quad g(n) = \sum_{k=N1}^{N2} c(n-k)f(k) \quad (6.6)$$

Unique Global Minimum Cones

$$\begin{aligned}
G_n^+ &= \{g \in l^1(\mathfrak{R}) : g(n) > 0\} \text{ και } |g(n)| > |g(k)| \text{ για κάθε } k \neq n \\
G_n^- &= \{g \in l^1(\mathfrak{R}) : g(n) < 0\} \text{ και } |g(n)| > |g(k)| \text{ για κάθε } k \neq n
\end{aligned} \quad (6.7)$$

Εφικτό Σύνολο (Attainable Set)

Το σύνολο  $T$  που περιέχει όλα τα πιθανά  $g \in l^1(\mathfrak{R})$  για πεπερασμένο μήκος ισοσταθμιστή:

$$T = \{g : g(n) = \sum_{k=N1}^{N2} c(n-k)f(k), f(k) \in \mathfrak{R}\} \quad (6.8)$$

Κύρτωση της ακολουθίας εισόδου  $s(n)$ 

Η κύρτωση τετάρτου βαθμού της ακολουθίας εισόδου ορίζεται:

$$K(s) = E\{s^4\} - 3E^2\{s^2\} \quad (6.9)$$

Με βάση αυτούς τους ορισμούς θέτουμε τους παρακάτω περιορισμούς που πρέπει να πληρούνται κατά την αρχικοποίηση του CMA προκειμένου να έχουμε σύγκλιση σε γενικό ελάχιστο.

- [1]. Η αρχική κρουστική απόκριση του συστήματος πρέπει να κυμαίνεται εντός ενός Unique Global Minimum Cone  $G_n$
- [2]. Η αρχική κατάσταση του ισοσταθμιστή θα πρέπει να ικανοποιεί την παρακάτω συνθήκη:

$$\frac{kurt(y)}{kurt(s)} > 0.5 \quad (6.10)$$

όπου  $y$  η έξοδος του ισοσταθμιστή και  $s$  η είσοδος του καναλιού, με  $kurt(s) = \frac{K(s)}{\sigma_s^4}$  και  $\sigma_s^4$  η διασπορά του  $s$ .

- [3]. Αν η θέση του γενικού ελάχιστου που αντιστοιχεί στο  $G_n$  καθορίζεται από το διάνυσμα  $e_n$ , τότε το  $e_n$  θα πρέπει να είναι μέσα ή κοντά στο σύνολο  $T \cap G_n$

Η συνθήκη [1] μπορεί να ικανοποιηθεί αρχικοποιώντας ως εξής

$$\begin{aligned}
f &= [f(-N) \quad \dots \quad f(-1) \quad f(0) \quad f(1) \quad \dots \quad f(N)]^T \\
&= [0 \quad \dots \quad 0 \quad 1 \quad 0 \quad \dots \quad 0]^T
\end{aligned} \quad (6.11)$$

Για την παραπάνω τιμή, η απόκριση συχνότητας του συστήματος θα είναι  $g(n) = c(n)$  και επειδή το  $\{c(n)\}$  έχει μοναδικό ακρότατο (peak), το  $g$  θα είναι μέσα στο global cone.

Η συνθήκη [3] μπορεί να εξασφαλιστεί αν το μήκος του ισοσταθμιστή είναι αρκετά μεγάλο.

Η συνθήκη [2] δεν μπορεί να εξασφαλιστεί κατά την αρχικοποίηση χωρίς γνώση του πραγματικού καναλιού. Το πρόβλημα αυτό αντιμετωπίζεται στο στάδιο της ενημέρωσης (updating) του αλγορίθμου με τον ακόλουθο τρόπο:

- Με κατάλληλη αρχικοποίηση του μεσαίου tap του φίλτρου όπως στην εξίσωση 6.11
- Tap centering: Κεντράρισμα του ειδικού βάρους του πίνακα των συντελεστών του φίλτρου για κάθε επανάληψη.
- Επιμήκυνση Ισοσταθμιστή: Εφόσον το  $g$  παρουσιάζει απότομες «ουρές» μετά από ένα εύλογο αριθμό επαναλήψεων, ο ισοσταθμιστής επιμηκώνεται προς την κατεύθυνση των ουρών.

### 6.4.2 Η παράμετρος $\mu$

Ακολουθώντας την παρακάτω ανάλυση [13-p176] είναι δυνατόν να εξάγουμε έναν τύπο που να δίνει τη βέλτιστη επιλογή του  $\mu$  για τον αλγόριθμο CMA. Εισάγουμε τις έννοιες του a-priori και a-posteriori σφάλματος αντίστοιχα:

$$\begin{aligned} e_a(i) &= s(i - D)e^{j\theta} - y(i) = x_i w_i^0 - x_i w_i \cong x_i \tilde{w}_i \\ e_p(i) &= x_i (\tilde{w}_{i+1} - q_i) \end{aligned} \quad (6.12)$$

Με βάση τους παραπάνω ορισμούς η σχέση 6.2 δίνει τη σχέση μεταξύ των δύο σφαλμάτων:

$$e_p(i) = e_a(i) - \frac{\mu}{\tilde{\mu}(i)} e(i) \quad (6.13)$$

όπου  $\tilde{\mu}(i) = 1 / \|x_i\|^2$ . Αντικαθιστώντας στην 6.2 βρίσκουμε:

$$\tilde{w}_{i+1} = \tilde{w}_i - \tilde{\mu}(i) x_i^* [e_a(i) - e_p(i)] + q_i \quad (6.14)$$

όπου  $q$  είναι ένας όρος που εκφράζει την μεταβλητή καθυστέρηση. Παίρνοντας τις ενέργειες και των δύο προηγούμενων εξισώσεων έχουμε:

$$\| \tilde{w}_{i+1} + q_i \|^2 + \tilde{\mu}(i) |e_a(i)|^2 = \| \tilde{w}_i \|^2 + \tilde{\mu}(i) |e_p(i)|^2 \quad (6.15)$$

η οποία δείχνει ότι οι ενέργειες δύο διανυσμάτων συντελεστών  $w$  σε δύο διαδοχικές χρονικές στιγμές σχετίζονται με τις ενέργειες των του a-priori και a-posteriori σφαλμάτων.

Θέτοντας την ισότητα  $E \| \tilde{w}_{i+1} \|^2 = E \| \tilde{w}_i \|^2$  και χρησιμοποιώντας την σχέση 6.13 έχουμε την ακόλουθη ισότητα:

$$E \tilde{\mu}(i) |e_a(i)|^2 = \text{Tr}(Q) + E \tilde{\mu}(i) |e_a(i) - \frac{\mu}{\tilde{\mu}(i)} e(i)|^2 \quad (6.16)$$

όπου  $Q = E q_i q_i^*$ , η οποία μπορεί να λυθεί για steady state μέσω τετραγωνικό σφάλμα. Για να γίνει αυτό θέτουμε δύο λογικές για steady-state υποθέσεις (assumptions):

**L1.** Το μεταδιδόμενο σύμβολο  $s(i-D)$  και το σφάλμα  $e_a(i)$  είναι ανεξάρτητα στο steady state έτσι ώστε  $E s^*(i-D) e_a(i) = 0$  αφού το  $s(i-D)$  έχει μηδενική μέση τιμή.

**L2.** Ο όρος  $\mu^2 \| u_i \|^2$  είναι ανεξάρτητος του  $y$  στο steady-state.

Έτσι η σχέση 6.16 γίνεται:

$$E \tilde{\mu}(i) |e_a(i)|^2 = \text{Tr}(Q) + E \tilde{\mu}(i) |e_a(i) - \frac{\mu}{\tilde{\mu}(i)} y(i)(R_2 - |y(i)|^2)|^2 \quad (6.17)$$

η οποία για  $i \rightarrow \infty$  γίνεται:

$$2\mu(i) E e_a(i) y(i) (R_2 - y^2(i)) = \text{Tr}(Q) + \mu^2 E \| u \|^2 y^2(i) (R_2 - y^2(i))^2 \quad (6.18)$$

Για να εξάγουμε το steady-state μέσω τετραγωνικό σφάλμα  $E |e_a(i)|^2$  αντικαθιστούμε στην παραπάνω εξίσωση  $y(i) = s(i-D) e_a$  και χρησιμοποιούμε τα L1, L2 οπότε:

$$\zeta_{CMA} \approx \frac{\text{Tr}(Q) / \mu + m E (s^2(i-D) R_2^2 - 2 R_2 s^4(i-D) + s^6(i-D)) E \| x(i) \|^2}{2 E (3 s^2(i-D) - R_2)} \quad (6.19)$$

Η συνάρτηση αυτή αποτελείται από δύο όρους: ο πρώτος μειώνεται με την αύξηση του  $\mu$  αυξάνεται με την αύξηση της μεταβλητότητας του συστήματος  $\text{tr}(Q)$ , ενώ ο δεύτερος αυξάνει με την αύξηση του  $\mu$ .

Μπορεί, λοιπόν, να εξαχθεί μια τιμή του  $\mu$ , που να ελαχιστοποιεί την 6.19 και η οποία είναι η βέλτιστη επιλογή του step size που μπορούμε να κάνουμε. Η τιμή αυτή δίνεται από τον τύπο:

$$\mu_o = \sqrt{\text{Tr}(Q) / [E(s^2(i-D)R_2 - 2R_2s^4(i-D) + s^6(i-D)E \|u\|^2)]} \quad (6.20)$$

# 7

## ΑΞΙΟΛΟΓΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Σκοπός του παρόντος κεφαλαίου είναι να μελετηθεί συνολικά η απόδοση του συστήματος “*Homebrew WFTP*” που κατασκευάστηκε, με ιδιαίτερη έμφαση και λεπτομέρεια στην ισοστάθμιση που είναι και το αντικείμενο αυτής της εργασίας. Αρχικά θα μελετηθούν οι εξαρτήσεις του συστήματος από διάφορους εξωτερικούς και εσωτερικούς παράγοντες καθώς και από σχεδιαστικές επιλογές που έγιναν κατά την εξέλιξη του project. Στη συνέχεια, με παράθεση πειραματικών αποτελεσμάτων μιας σειράς προσομοιώσεων που ακολούθησε την ανάπτυξη του συστήματος, θα επιχειρηθεί μια μελέτη των trade-offs που διέπουν τη λειτουργία του συστήματος και θα προταθεί η βέλτιστη επιλογή ρυθμίσεων για την οποία επιτυγχάνονται οι κύριοι στόχοι που αναφέρθηκαν στο κεφάλαιο 1.2.

### 7.1 Εξαρτήσεις του Συστήματος WFTP

#### A. Το Κανάλι

Μια κύρια πηγή εξάρτησης της λειτουργίας του συστήματος είναι το κανάλι ανάμεσα στον πομπό και στον δέκτη. Όπως αναφέρθηκε και στο Κεφάλαιο 2, φαινόμενα multipath που οφείλονται στα διάφορα εμπόδια που συναντάει το σήμα στη διαδρομή του και το αναγκάζουν να φτάνει στο δέκτη σε πολλά εξασθετισμένα αντίτυπα οδηγώντας σε παραμόρφωση (ISI). Το φαινόμενο αυτό, ενισχύεται όσο η απόσταση ανάμεσα σε πομπό και δέκτη μεγαλώνει αλλά και λόγω κίνησης μεγάλων αντικειμένων μέσα στο κανάλι. Στην πράξη δεν συναντήσαμε έντονα φαινόμενα χρονικής μεταβολής του καναλιού, καθώς οι συνθήκες μέσα σε εσωτερικούς χώρους όπου δούλεψε το σύστημα δεν είναι ευμετάβλητες και πάντοτε υπήρχε οπτική επαφή πομπού-δέκτη. Επίσης η απόσταση ανάμεσα στον πομπό και το δέκτη κατά τις πειραματικές μεταδόσεις δεν ξεπερνούσε τα 5 μέτρα.



Η επιλογή της συχνότητας λειτουργίας του συστήματος αποδείχτηκε καθοριστικός παράγοντας για την επίτευξη καλής επικοινωνίας ανάμεσα στους δύο υπολογιστές. Λόγω της εκπομπής στο ραδιοφωνικό φάσμα συχνοτήτων, ο δέκτης αντιμετώπιζε συχνά παρεμβολές από σταθμούς που εξέπεμπαν στη συγκεκριμένη ή σε γειτονικές συχνότητες. Παρόλο που ο πομπός διέθετε επιλογή για τη συχνότητα εκπομπής και ο δέκτης ήταν αρκετά ακριβής για να το λάβει, στην πράξη η συχνότητα του πομπού άλλαζε ανάλογα με την πηγή τροφοδοσίας ή την (μεταβαλλόμενη με το χρόνο) τάση που έδιναν οι μπαταρίες. Επίσης οι παρεμβολές από άλλους ραδιοφωνικούς σταθμούς δεν γίνονταν στις ίδιες συχνότητες καθόλη τη διάρκεια της ημέρας και έτσι δεν ήταν δυνατό να βρεθεί συχνότητα στην οποία να λειτουργεί το σύστημα κάθε φορά και χωρίς προβλήματα.

## **B. Το Υλικό**

Το σύστημα σχεδιάστηκε έτσι ώστε να λειτουργεί με τις κάρτες ήχου του υπολογιστή. Αν και αυτό το χαρακτηριστικό δίνει ευελιξία ως προς την εγκατάσταση του συστήματος σε οποιοδήποτε υπολογιστή, στην πράξη αποτέλεσε τροχοπέδη για την επίτευξη υψηλότερων ρυθμών μετάδοσης. Καταρχήν, όλες οι κάρτες δεν έχουν την δυνατότητα να παίζουν το ίδιο καλά, και χρειάζεται μια δοκιμαστική μετάδοση προκειμένου να ανιχνευτούν πιθανά προβλήματα (εσωτερικός θόρυβος της κάρτας, παρεμβολή από άλλη κάρτα κοντά στην κάρτα ήχου στον υπολογιστή κτλ). Η πλειονότητα όμως αυτών, μπορεί ύστερα από κάποιες ρυθμίσεις να παίζει ικανοποιητικά. Η ένταση του ήχου, τόσο στον πομπό όσο και στο «recording» του δέκτη, παίζουν ρόλο και είναι δυνατόν να «ψαλιδίσουν» τις κυματομορφές καταστρέφοντας την πληροφορία (βλ Κεφάλαιο 1.3.2B). Τέλος φαίνεται πως υπάρχει ένα άνω όριο στην συχνότητα δειγματοληψίας των περισσότερων καρτών ήχου, ξεπερνώντας αυτό το όριο και ανάλογα με τις άλλες παραμέτρους της εκπομπής μπορεί να είναι αδύνατον να ανακατασκευαστεί το μεταδιδόμενο σήμα. Η συχνότητα των 44100 αποδείχτηκε ότι λειτουργεί πάντα, ενώ για μικρές διαμορφώσεις (4-QAM, 4-PSK) είναι δυνατόν να ανέβει στα 88200.

## **Γ. Περιβάλλον Ανάπτυξης**

Η ανάπτυξη και χρήση του συστήματος επικοινωνίας WFTP έγινε στο Matlab, στην έκδοση 7.0 και το λειτουργικό σύστημα στο οποίο έτρεχε η εφαρμογή αυτή είναι τα Microsoft Windows XP. Η επιλογή αυτή έγινε λόγω της εξοικείωσης των μελών της ομάδας με τα συγκεκριμένα προγράμματα, αλλά και της υποστήριξης (Communication & Signal Processing Toolbox, συνεργασία με το Excel για την επεξεργασία των αποτελεσμάτων κ.α) που παρείχαν για την διεκπεραίωση του Project. Έτσι ο χρόνος ολοκλήρωσης του project ήταν πολύ μικρός σε σχέση με εναλλακτικές επιλογές που είχαμε (π.χ C, assembly).

Το κόστος αυτής της επιλογής ήταν η μεγάλη απαιτούμενη υπολογιστική ισχύς και η εξάρτηση της λειτουργίας του συστήματος από το λειτουργικό. Το Matlab είναι ένα αρκετά «βαρύ» εργαλείο και για τη λειτουργία του χρησιμοποιεί αποκλειστικά πίνακες, οι οποίοι στην περίπτωση μας ήταν αρκετά μεγάλοι καθώς αποθήκευαν δεδομένα που έφταναν σε πραγματικό χρόνο. Έτσι κάποιες απαιτητικές λειτουργίες (π.χ ισοστάθμιση) χρειάστηκαν βελτίωση και κάποιες παραχωρήσεις προκειμένου να λειτουργήσουν σε ικανοποιητικό χρόνο.

## **Δ. Χρήση UDP για ARQ**

Το πρωτόκολλο επικοινωνίας UDP χρησιμοποιείται για το την αξιόπιστη επικοινωνία πομπού και δέκτη για αποστολή wake-up signal από τον πομπό στο δέκτη για έναρξη ασύρματης μετάδοσης και για επιβεβαίωση (θετική ή αρνητική) λήψης πακέτου από το δέκτη σε closed loop. Το συγκεκριμένο πρωτόκολλο, επιβάλλει διάφορες μεταβλητές καθυστερήσεις στο σύστημα, τουλάχιστον όπως έχει υλοποιηθεί η διασύνδεσή του με το Matlab. Η κατάσταση γίνεται χειρότερη αν οι δύο υπολογιστές είναι συνδεδεμένοι σε μεγαλύτερο δίκτυο υπολογιστών (π.χ Δίκτυο Πολυτεχνείου), καθώς το UDP/IP είναι broadcast πρωτόκολλο και άρα όλα τα πακέτα φτάνουν σε όλους του υπολογιστές του τοπικού υποδικτύου (π.χ Telecom Lab).

## 7.2 Προσομοίωση Λειτουργίας

Στα πλαίσια της αξιολόγησης – αποσφαλμάτωσης του συστήματος πραγματοποιήθηκε μια μακρά σειρά δοκιμών (το πιο χρονοβόρο στάδιο της ανάπτυξης του Project). Αυτές περιελάμβαναν την μετάδοση πληροφορίας με όλες τις δυνατές παραμέτρους και την αποθήκευση των αποτελεσμάτων κάθε μετάδοσης για μελλοντική επεξεργασία. Η διαδικασία αυτή απέφερε τα αποτελέσματα που θα παρατεθούν στην συνέχεια στους αντίστοιχους πίνακες. Σημειώνεται ότι επειδή μεταδόσεις με το ίδιο setup έγιναν πολλές φορές προκειμένου να εξαχθούν έγκυρα συμπεράσματα, οι τιμές που αναγράφονται στους πίνακες σχετικά με την απόδοση (π.χ χρόνος, BER κτλ) είναι ο μέσος όρος των μεταδόσεων. Συνεπάγεται λοιπόν ότι η ακρίβεια των παρακάτω αποτελεσμάτων περιλαμβάνει στατιστικό σφάλμα.

Οι διάφοροι παράμετροι που αλλάζαμε συνεχώς προκειμένου να αξιολογήσουμε την απόδοση του συστήματος ήταν οι εξής:

- Το είδος της διαμόρφωσης (PSK, PPM, QAM) και το  $M$  (4, 8, 16, 32 κτλ)
- Το μέγεθος του πακέτου
- Το μέγεθος της ακολουθίας εκμάθησης
- Η περίοδος του συμβόλου  $T$
- Η συχνότητα δειγματοληψίας  $F_s$
- Η χρήση ή όχι Amplitude/Phase Recovery
- Η χρήση ή όχι Ισοσταθμιστή, το είδος του (LMS, RLS, CMA), το μήκος του φίλτρου και οι παράμετροί του
- Η χρήση ή όχι Convolutional Coder και το είδος του
- Η χρήση ή όχι Encoder/ Interleaver, το είδος του
- Η χρήση ή όχι Interleaver
- Ο χρόνος μετάδοσης και το transfer rate για όλο το αρχείο
- Το bit error rate για κάθε πακέτο

Λόγω του πλήθους των παραμέτρων, και του μεγάλου αριθμού των μεταξύ τους συνδυασμών, είναι επιθυμητό να καταλήξουμε σε ορισμένες βέλτιστες τιμές για όσο το δυνατόν περισσότερες παραμέτρους.

Πακέτο	train	T	Fs	Tr.Rate	Time
10.000	500	10	44100	4,228	212,8
50.000	500	10	44100	5,245	171,5
80.000	500	10	44100	5,431	165,7
100.000	500	10	44100	5,548	162,2
150.000	500	10	44100	5,641	159,5
Πίνακας 7.1					

Το μέγεθος του πακέτου είναι ο πρώτος υπονήφιος επειδή επηρεάζει το χρόνο αποστολής του αρχείου, καθώς κάθε αποστολή συνοδεύεται από θετική ή αρνητική επιβεβαίωση της λήψης από το δέκτη. Λόγω του πρωτόκολλου stop and wait που χρησιμοποιήθηκε, μέχρι να λάβει απάντηση, ο πομπός μένει αδρανής. Όσο πιο μικρό είναι το μέγεθος του πακέτου λοιπόν, τόσο περισσότερα πακέτα πρέπει να σταλούν και άρα μεγαλύτερος ο χρόνος αναμονής λόγω του ARQ. Από την άλλη, αν γίνει λάθος και ο πομπός χρειαστεί να ξαναστείλει το πακέτο, ένα μεγαλύτερο πακέτο θα κάνει περισσότερο χρόνο να ξανασταλθεί, ρίχνοντας το transfer rate. Επίσης σε closed loop, μεγάλο πακέτο έχει μεγαλύτερη πιθανότητα να έχει λάθος bits και άρα να χρειαστεί να ξανασταλθεί. Στον πίνακα 7.1 παρουσιάζονται ενδεικτικά οι χρόνοι που απαιτούνται για την αποστολή ενός αρχείου 108kbytes για διάφορα μεγέθη πακέτου με διαμόρφωση 4PSK μέσα από το σύστημα “Homebrew WFTP”. Οι υπόλοιπες παράμετροι διατηρήθηκαν σταθερές για να είναι τα αποτελέσματα συγκρίσιμα. Το αρχείο επιλέχθηκε να είναι της τάξεως των 100kbytes  $\approx$  900kbits. Από τον πίνακα προκύπτει ότι η πιο συμφέρουσα επιλογή που συμβιβάζει τα παραπάνω trade-offs είναι το πακέτο των 100.000 bits. Στο εξής, αν δεν αναφέρεται κάτι άλλο, το μέγεθος του πακέτου θα θεωρείται 100kbits. Το μέγεθος του αρχείου ( $<10^6$ ) έχει σαν αποτέλεσμα να μην είμαστε σε θέση να προσδιορίσουμε επαρκώς σωστά το μέσο BER κάτω του  $10^{-5}$ . Έτσι στις περιπτώσεις όπου το BER βγήκε μηδέν σε κάθε επανάληψη του πειράματος θα σημειώνεται σαν «μη επαρκώς καθορισμένο (μ.ε.κ.)» και όχι π.χ 0. Ακόμα, η συνάρτηση Amplitude/Phase Recovery, χρησιμοποιείται πάντα καθώς ουσιαστικά δουλεύει για μικρό training sequence (π.χ 100 bits) και επιφέρει μικρή πτώση στο ρυθμό μετάδοσης (μικρότερο του 0,1bit/sec). Έτσι εξασφαλίζουμε με μεγαλύτερη σιγουριά το αλάνθαστο των μεταδόσεων.

## 7.3 Αξιολόγηση του Συστήματος

### 7.3.1 Το PSK

α/α	T	Fs	Tr.Rate	Av.BER	(+)
1	10	44100	5,518	<i>μ.ε.κ.</i>	-
2	5	44100	11,52	<i>μ.ε.κ.</i>	-
3	5	88200	16,71	0,16	-
4	10	88200	8,281	<i>μ.ε.κ.</i>	-
5	10	44100	5,497	<i>μ.ε.κ.</i>	Equalizers
6	10	88200	7,808	<i>μ.ε.κ.</i>	Equalizers
7	10	44100	4,880	<i>μ.ε.κ.</i>	CRC
8	10	88200	6,921	<i>μ.ε.κ.</i>	CRC
9	10	44100	4,851	<i>μ.ε.κ.</i>	Equalizers+CRC
10	10	88200	6,587	<i>μ.ε.κ.</i>	Equalizers+CRC
<b>Πίνακας 7.2 – 4PSK</b>					

Η πρώτη διαμόρφωση που επιλέχτηκε ήταν αυτή του 4PSK. Δοκιμάσαμε αρχικά να το στείλουμε επιλέγοντας  $T=10$  σύμβολα και  $F_s=44100\text{Hz}$ . Μειώνοντας την περίοδο  $T$  στο μισό ή αυξάνοντας την συχνότητα δειγματοληψίας στο διπλάσιο πετυχαίνουμε θεωρητικά διπλασιασμό του ρυθμού μετάδοσης. Στην πρώτη περίπτωση πράγματι ο ρυθμός μετάδοσης διπλασιάστηκε, ενώ στην δεύτερη είχαμε μια αύξηση του ρυθμού της τάξης του 50%, σε κάθε περίπτωση με αποδεκτό BER ( $<10^{-5}$ ). Επιχειρώντας και τα δύο μαζί προκύπτουν λάθη στο 16% των bits. Εισάγωντας ισοσταθμιστές παρατηρούμε μια αμελητέα πτώση του ρυθμού μετάδοσης, που οφείλεται στο χρόνο επεξεργασίας που εισάγει η μονάδα του ισοσταθμιστή. Ο στόχος για λάθη της τάξεως του  $10^{-6}$  που ετέθη σε κάθε μετάδοση επιτυγχάνεται υπό κανονικές συνθήκες, άρα δεν υπάρχει λόγος να χρησιμοποιούμε coders ή επαναμετάδοση πακέτων (CRC) καθώς το μόνο που θα καταφέρουμε σε μια τέτοια περίπτωση είναι να ρίξουμε το ρυθμό μετάδοσης. Επιπλέον στον πίνακα 7.2, στις τέσσερις τελευταίες μετρήσεις αναφέρονται οι χρόνοι με CRC για λειτουργία του συστήματος σε closed-loop και φαίνεται η αναπόφευκτη πτώση του ρυθμού μετάδοσης που οφείλεται στις καθυστερήσεις του UDP καθώς και το χρόνο αποθήκευσης των μεταβλητών και των αποτελεσμάτων των προσομοιώσεων. Η εγγυημένα αλάνθαστη μετάδοση που προσφέρει η χρήση του CRC όμως ανταμείβει τις όποιες απώλειες σε ταχύτητα.

Επόμενο λογικό βήμα ήταν να αυξήσουμε τον αριθμό των bits ανά σύμβολο Μ σε 8. Δοκιμάσαμε όλους τους συνδυασμούς παραμέτρων που δούλεψαν και για το 4PSK, δηλαδή  $T=5$  και 10 και  $F_s=44100$  και 88200. Τα αποτελέσματα φαίνονται στον παρακάτω πίνακα:

α/α	T	Fs	Tr.Rate	Av.BER	(+)
1	10	44100	8,635	<b>μ.ε.κ.</b>	-
2	10	44100	8,378	<b>μ.ε.κ.</b>	Equalizer
3	5	44100	16,42	0,001	-
4	5	44100	15,04	<b>μ.ε.κ.</b>	Equalizer
5	5	88200	23,2	0,28	-
6	10	88200	12,52	<b>μ.ε.κ.</b>	-
7	10	88200	12,40	<b>μ.ε.κ.</b>	Equalizer
8	10	44100	7,168	<b>μ.ε.κ.</b>	CRC
9	10	88200	9,656	<b>μ.ε.κ.</b>	CRC
10	10	44100	6,977	<b>μ.ε.κ.</b>	Equalizer+CRC
11	5	44100	10,02	<b>μ.ε.κ.</b>	Equalizer
12	10	88200	9,574	<b>μ.ε.κ.</b>	Equalizer+CRC
<b>Πίνακας 7.3 – 8PSK</b>					

Με  $T=10$  και  $F_s=44100$ , το σύστημα δούλεψε ικανοποιητικά, κάθε φορά, πετυχαίνοντας μέσο ρυθμό μετάδοσης 8,635bits/sec. Με χρήση ισοσταθμιστή για ακόμα καλύτερα αποτελέσματα ο ρυθμός αυτός έπεσε κατά 0,3bits/sec. Σε κάθε περίπτωση ο ρυθμός αυτός είναι χαμηλότερος του ταχύτερου 4PSK το οποίο δουλεύει σωστά με τη μισή περίοδο T. Ρίχνοντας λοιπόν το T στο μισό και για το 8PSK, διπλασιάζουμε το ρυθμό μετάδοσης εμφανίζοντας όμως λίγα λάθη της τάξεως του  $10^{-4}$ . Αυτά μπορούμε να τα διορθώσουμε με τη χρήση ισοσταθμιστή θυσιάζοντας κάτι από το ρυθμό μετάδοσης. Αν παράλληλα δοκιμάσουμε να διπλασιάσουμε την συχνότητα  $F_s$  στα 88200, τα λάθη εκτοξεύονται στα 28%, με αποτέλεσμα να μην είναι δυνατή η διόρθωσή τους ούτε με ισοσταθμιστή ούτε με κάποιο coder (ο οποίος θα έριχνε έτσι και αλλιώς τον ρυθμό μετάδοσης). Για να έχουμε συχνότητα δειγματοληψίας  $F_s=88200$ , η περίοδος T πρέπει να είναι τουλάχιστον 10 οπότε πετυχαίνουμε μια μέση απόδοση της τάξεως των 12,5kbps με μηδενικό bit error rate και με χρήση ισοσταθμιστή για μεγαλύτερη ασφάλεια της τάξεως των 12,3kbps. Τελικά η καλύτερη απόδοση του 8PSK επιτυγχάνεται με μικρή περίοδο ( $T=5$ ), συντηρητική  $F_s$  (44100) και χρήση ισοσταθμιστή (LMS ή RLS) και είναι πάνω από 15kbps (βλ μετάδοση 4).

Στις μετρήσεις 8-12 φαίνονται οι χρόνοι για το CRC για όσους συνδυασμούς έδωσαν μηδενικά λάθη.

Οι πειραματικές μετρήσεις που φαίνονται στον πίνακα 7.4 έγιναν για το 16PSK και δείχνουν την απόδοσή του για  $T=10$ ,  $F_s=44100$  και  $88200$ , με και χωρίς ισοσταθμιστές και coders:

$\alpha/\alpha$	T	Fs	Tr.Rate	Av.BER	(+)
1	10	44100	11,37	$2,2 \cdot 10^{-3}$	-
2	10	44100	11,29	<b>μ.ε.κ.</b>	Equalizer
3	10	44100	9,94	<b>μ.ε.κ.</b>	Equalizer+Reed Solomon
4	10	88200	16,63	$5,6 \cdot 10^{-2}$	-
5	10	88200	15,9	$2 \cdot 10^{-2}$	Equalizer
6	10	88200	7,00	$1,1 \cdot 10^{-2}$	Equalizer+Reed Solomon
7	10	44100	7,3	<b>μ.ε.κ.</b>	Equalizer+Conv.Coder
8	10	88200	10,8	$1,6 \cdot 10^{-2}$	Equalizer+Conv.Coder
<b>Πίνακας 7.4α – 16PSK</b>					

Ξεκινώντας πάλι από τη βάση, για  $T=10$  και  $F_s=44100$ , το σύστημα βγάζει λάθη γύρω στο 0,2% χωρίς καμιά επεξεργασία πλην της συνάρτησης Amplitude/Phase Recovery που χρησιμοποιείται σε κάθε μετάδοση. Ο ρυθμός μετάδοσης είναι 11,37, δηλαδή χαμηλότερος του 8PSK με  $T=5$  και  $F_s=44100$ . Για να διορθωθούν τα λάθη αυτά πρέπει εκτός από τον ισοσταθμιστή να μπει και κάποιος encoder (Reed Solomon ή Convolutional), οι οποίοι ρίχνουν όμως το ρυθμό κάτω από το 10. Αξίζει να σημειωθεί ότι ο ισοσταθμιστής μόνο (ειδικά ο RLS) καταφέρνει τις περισσότερες φορές να διορθώνει τα λάθη, αλλά επειδή δουλεύει οριακά, κάποιες φορές αφήνει μερικά. Έτσι συμφέρει να μπει μαζί και ένας «γρήγορος» Reed Solomon, ο οποίος όταν δεν μένουν λάθη από τον ισοσταθμιστή επιβαρύνει πολύ λίγο την ταχύτητα (απλά για τον έλεγχο) ενώ καθυστερεί σημαντικά μόνο όταν χρειαστεί να διορθώσει.

Συνεχίζοντας την προσπάθεια βελτίωσης της ταχύτητας διατηρώντας παράλληλα χαμηλό BER, αυξάνουμε την συχνότητα στα 88200 οπότε τα λάθη φτάνουν το 5,6%. Με χρήση ισοσταθμιστή, υποτριπλασιάζονται, ενώ αν μπει και Reed Solomon πέφτουν στο 1,1% και με Convolutional Encoder λίγο χειρότερα. Πάντως δεν είναι δυνατόν να διορθωθούν τελείως.

Στον πίνακα 7.4β φαίνονται οι χρόνοι για closed loop, δηλαδή με το CRC.

α/α	T	Fs	Tr.Rate	Av.BER	(+)
1	10	44100		$2,2 \cdot 10^{-3}$	CRC
2	10	88200		$5,6 \cdot 10^{-2}$	CRC
3	10	44100		$10^{-3}$	Equalizer + CRC
4	10	88200		$2 \cdot 10^{-2}$	Equalizer + CRC
5	10	44100		<b>μ.ε.κ.</b>	Equalizer+Reed Solomon+ CRC
6	10	88200		$1,1 \cdot 10^{-2}$	Equalizer+Reed Solomon+CRC
7	10	44100		<b>μ.ε.κ.</b>	Equalizer+Conv.Coder +CRC
8	10	88200		$1,6 \cdot 10^{-2}$	Equalizer+Conv.Coder+CRC
<b>Πίνακας 7.4α – 16PSK</b>					

Ανεβάζοντας το  $M=32$ , και για τους ίδιους συνδυασμούς τιμών με το 16PSK πήραμε τα παρακάτω αποτελέσματα:

α/α	T	Fs	Tr.Rate	Av.BER	(+)
1	10	44100	13,72	$1,5 \cdot 10^{-3}$	-
2	10	44100	13,47	$1,2 \cdot 10^{-3}$	Equalizer
3	10	44100	3,467	$1,1 \cdot 10^{-3}$	Equalizer+Reed Solomon
4	10	44100	4,973	$1 \cdot 10^{-3}$	Equalizer+Conv.Coder
<b>Πίνακας 7.5 – 32PSK</b>					

Για  $T=10$  και  $F_s=44100$ , προκύπτουν λάθη της τάξεως του  $10^{-3}$ , και τόσο οι ισοσταθμιστές όσο και οι encoders δεν κατόρθωσαν να ρίξουν τα λάθη αυτά. Για μεγαλύτερη συχνότητα δειγματοληψίας, τα λάθη ανέβαιναν δύο τάξεις ψηλότερα και δεν ήταν δυνατό να διορθωθούν με καμιά τεχνική. Δεν δοκιμάσαμε το 32-PSK σε closed loop, αφού στο σύστημά μας δεν είναι δυνατή η αποστολή αλάνθαστων πακέτων με τόσο υψηλό  $M$ .

Από τις παραπάνω αναλύσεις για το MPSK προκύπτει το γενικό συμπέρασμα ότι για  $M=4$  και  $M=8$ , το μηδενικό BER επιτυγχάνεται τόσο σε open όσο και σε closed loop, ακόμα και χωρίς ισοσταθμιστές. Στο 16PSK χρειάζονται equalizers ή/και encoders. Η υψηλότερη ταχύτητα επιτυγχάνεται στο 8PSK με συχνότητα δειγματοληψίας 44100 και κυμαίνεται στα 15kbps για  $T=5$ . Η ταχύτητα αυτή μπορεί να αυξηθεί και άλλο μειώνοντας το training sequence, αλλά τότε βάζουμε σε ρίσκο την αξιοπιστία του συστήματος τόσο για σφάλματα συγχρονισμού όσο και για κακή εκτίμηση του καναλιού από τον ισοσταθμιστή. Η επόμενη πιο «ασφαλής» ταχύτητα είναι τα 12,5kbps του 8PSK με  $T=10$  και  $F_s=88200$  τα οποία σηκώνουν και μείωση του training sequence στο μισό, επιλογή που δίνει περίπου 1kbps ακόμα.



### 7.3.2 Το QAM

$\alpha/\alpha$	T	Fs	Tr.Rate	Av.BER	(+)
1	10	44100	5,482	<b>μ.ε.κ.</b>	Equalizers
2	10	88200	7,343	<b>μ.ε.κ.</b>	Equalizers
3	10	44100	4,853	<b>μ.ε.κ.</b>	Equalizers+CRC
4	10	88200	6,250	<b>μ.ε.κ.</b>	Equalizers+CRC
<b>Πίνακας 7.6 – 4QAM</b>					

Ακολουθώντας την ίδια σειρά ενεργειών με το PSK, θα ξεκινήσουμε την αξιολόγηση του QAM. Αρχικά δοκιμάζουμε για T=10 και Fs=44100, το οποίο έρχεται με BER της τάξεως του  $10^{-6}$  με μέσο ρυθμό 5,48, απόδοση παρόμοια με του 4PSK με τις ίδιες ρυθμίσεις. Ανεβάζουμε τη συχνότητα δειγματοληψίας στα 88200 οπότε ο ρυθμός ανεβαίνει στα 7,34 και το bit error rate παραμένει το χαμηλό. Η απόδοση αυτή είναι ελαφρώς χειρότερη από το αντίστοιχο PSK, αλλά η διαφορά είναι στα όρια του στατιστικού σφάλματος (εφόσον οι τιμές στους πίνακες είναι μέσοι όροι) οπότε το συμπέρασμα αυτό δεν γενικεύεται. Οι ίδιες μετρήσεις σε closed loop εμφανίζονται στις γραμμές 3 και 4 του πίνακα 7.6.

Για M=8, και για τις ίδιες παραμέτρους που τρέξαμε και στο 8PSK, το 8QAM έδωσε τα παρακάτω αποτελέσματα:

$\alpha/\alpha$	T	Fs	Tr.Rate	Av.BER	(+)
1	10	44100	8,655	<b>μ.ε.κ.</b>	-
2	10	88200	12,45	$8 \cdot 10^{-1}$	-
3	10	44100	8,569	<b>μ.ε.κ.</b>	Equalizer
4	10	88200	12,30	$1,2 \cdot 10^{-2}$	Equalizer
5	10	88200	5,314	$2,7 \cdot 10^{-4}$	Equalizer+Reed Solomon
6	10	88200	8,753	$1,6 \cdot 10^{-4}$	Equalizer+Conv.Coder
<b>Πίνακας 7.7α – 8QAM</b>					

Για T=10 και Fs 44100, το 8QAM συνήθως δεν βγάζει λάθη παραπάνω από το στόχο που ετέθη. Για μεγαλύτερη σιγουριά, εισάγουμε έναν ισοσταθμιστή που όπως έχουμε ξανασχολιάσει ρίχνει ελάχιστα τον ρυθμό, εξασφαλίζει όμως το επιθυμητό BER. Μπορούμε να προσθέσουμε και coders, αλλά όπως έχουμε δει και παραπάνω, το μόνο που καταφέρνουμε είναι να ρίξουμε το ρυθμό μετάδοσης. Ανεβάζοντας την συχνότητα στα 88200 προκύπτουν λάθη της τάξης του  $10^{-2}$ , τα οποία με τη βοήθεια των coders πέφτουν στην τάξη του

$10^{-4}$ . Δηλαδή το όριο για την σωστή λειτουργία του QAM με συχνότητα 88200 που προσφέρει σημαντική αύξηση των επιδόσεων είναι για  $M=4$  σε αντίθεση με το PSK που είναι για  $M=8$ .

Στον πίνακα 7.7β, φαίνονται οι χρόνοι σε closed loop για όσους από τους παραπάνω συνδυασμούς είναι δυνατόν να δουλέψει το CRC, δηλαδή όσα σε open loop έχουν BER κοντά στο 0:

$\alpha/\alpha$	T	Fs	Tr.Rate	Av.BER	(+)
1	10	44100	7,234	<b>μ.ε.κ.</b>	CRC
3	10	44100	7,143	<b>μ.ε.κ.</b>	Equalizer + CRC
4	10	88200	4,727	<b>μ.ε.κ.</b>	Equalizer+Reed Solomon+CRC
5	10	88200	4,580	<b>μ.ε.κ.</b>	Equalizer+Conv.Coder+CRC
<b>Πίνακας 7.7β – 8QAM-CRC</b>					

Για  $M=16$ , τα πειραματικά αποτελέσματα για διαμόρφωση QAM βρίσκονται στον παρακάτω πίνακα:

$\alpha/\alpha$	T	Fs	Tr.Rate	Av.BER	(+)
1	10	44100	9,823	$1.4 \cdot 10^{-3}$	Equalizer
2	10	88200	13,868	$3.7 \cdot 10^{-2}$	Equalizer
3	10	44100	7,212	<b>μ.ε.κ.</b>	Equalizer+Reed Solomon
4	10	88200	6,996	$1.8 \cdot 10^{-2}$	Equalizer+Reed Solomon
5	10	44100	7,462	<b>μ.ε.κ.</b>	Equalizer+Conv.Coder
6	10	88200	8,318	$0.7 \cdot 10^{-2}$	Equalizer+Conv.Coder
<b>Πίνακας 7.8α – 16QAM</b>					

Τα λάθη χωρίς χρήση ισοσταθμιστή ήταν πάρα πολλά, γι' αυτό ξεκινάμε κατευθείαν με τον ισοσταθμιστή ενεργό. Όπως φαίνεται από τον πίνακα 7.8α, τα λάθη είναι πολλά ακόμα και με ισοσταθμιστή. Μπορούν όμως με τη βοήθεια των coders να διορθωθούν τουλάχιστον για την περίπτωση της συχνότητας 44100. Η πτώση σε ρυθμό είναι της τάξεως του 20%, όμως εξασφαλίζεται αξιόπιστη λειτουργία και του 16QAM

Για closed loop, οι αντίστοιχοι μέσοι χρόνοι, μόνο για  $F_s=44100$  που έχει νόημα, συνοψίζονται παρακάτω:

$\alpha/\alpha$	T	Fs	Tr.Rate	Av.BER	(+)
1	10	44100	7,965	<b>μ.ε.κ.</b>	Equalizer + CRC
2	10	44100	6,157	<b>μ.ε.κ.</b>	Equalizer+Reed Solomon+ CRC
3	10	44100	8,438	<b>μ.ε.κ.</b>	Equalizer+Conv.Coder +CRC
<b>Πίνακας 7.8β – 16QAM - CRC</b>					

### 7.3.3 Το PPM

Τελευταία διαμόρφωση που μελετήσαμε είναι το PPM. Το χαρακτηριστικό της συγκεκριμένης διαμόρφωσης είναι ότι σε αντίθεση με τα PSK και QAM, δεν έχει αστερισμό και οι περίοδοι  $T$  που μπορούν να χρησιμοποιηθούν είναι πολλαπλάσια του  $M$ . Εξαιτίας του πρώτου, δεν μπορούμε να εφαρμόσουμε ισοστάθμιση, καθώς αυτή εφαρμόζεται στα σύμβολα (έξοδος των συσχετιστών). Υπάρχει βέβαια η επιλογή να την εφαρμόσουμε στις κυματομορφές, πράγμα που πολλαπλασιάζει τον αριθμό των προς επεξεργασία bits με  $T$ . Το  $T$  που χρησιμοποιούμε συνήθως στο PPM είναι 8 ή 16, πράγμα που για πακέτο 100.000bits σημαίνει πάνω από 800.000 ή 1.600.000 δείγματα. Με μια τάξη  $10^6$  δειγμάτων ανά πακέτο, ο χρόνος επεξεργασίας αυξάνεται και μάλιστα εκθετικά ρίχνοντας τον ρυθμό μετάδοσης στο 1/10. Συνεπώς δεν έχει νόημα η εφαρμογή ισοσταθμίσης στο κανάλι, όταν χρησιμοποιείται PPM.

Η μεγάλη σχετικά με τα QAM και PSK περίοδος δειγματοληψίας του PPM μας αποθαρρύνει και από τη χρήση coders. Είδαμε ότι ένας καλός coder στις παραπάνω διαμορφώσεις ρίχνει τον ρυθμό μετάδοσης κατά 20-40%. Η μεγαλύτερη περίοδος που χρησιμοποιεί το PPM μεγαλώνει και την επιβάρυνση από τους coders κατά ένα ποσοστό ανάλογο του λόγου των περιόδων αυτών. Δεδομένου μάλιστα ότι το 4PPM που προσφέρει μεγαλύτερη ταχύτητα από τα αντίστοιχα PSK, QAM δεν εμφανίζει λάθη, θα περιοριστούμε στη δοκιμή του μέχρι εκεί που φτάνει με λάθη λιγότερα από την τάξη του  $10^{-6}$ . Παρακάτω λοιπόν θα αναλύσουμε την απόδοσή του μόνο για μικρά  $M$ .

Για το 4PPM οι μετρήσεις φαίνονται στον πίνακα 7.9:

$\alpha/\alpha$	$T$	$F_s$	Tr.Rate	Av.BER	CRC
1	8	44100	5,659	<b>μ.ε.κ.</b>	-
2	8	88200	11,399	<b>μ.ε.κ.</b>	-
3	8	44100	4,988	<b>μ.ε.κ.</b>	✓
4	8	88200	9,004	<b>μ.ε.κ.</b>	✓
<b>Πίνακας 7.9 – 4PPM</b>					

Όπως βλέπουμε, αν και χρησιμοποιεί υψηλή σε σχέση με τα 4QAM και 4PSK περίοδο  $T=8$ , οι χρόνοι του PPM είναι σαφώς καλύτεροι και από τα δύο, με

μηδενικά λάθη. Ακόμα και με το CRC (γραμμές 3-4), το PPM εξακολουθεί να συμφέρει για τόσο μικρό M.

Για 8PPM, οι μετρήσεις συνοψίζονται στον πίνακα 7.10

α/α	T	Fs	Tr.Rate	Av.BER	CRC
1	16	44100	4,487	μ.ε.κ.	-
2	16	88200	5,930	μ.ε.κ.	-
3	16	44100	4,034	μ.ε.κ.	✓
4	16	88200	5,193	μ.ε.κ.	✓
<b>Πίνακας 7.10 – 8PPM</b>					

Ήδη βλέπουμε ότι η αύξηση του T για να έχουμε χαμηλό BER, προκάλεσε μια πτώση στον ρυθμό μετάδοσης του πακέτου που δεν καλύπτεται από την παράλληλη αύξηση του M. Η κατάσταση αυτή χειροτερεύει για 16PPM όπου η περίοδος για μηδενικό BER είναι 32 οπότε οι χρόνοι πέφτουν σε απαράδεκτα επίπεδα (βλ πίνακα 7.11):

α/α	T	Fs	Tr.Rate	Av.BER	CRC
1	32	44100	3,097	μ.ε.κ.	-
2	32	88200	4,189	μ.ε.κ.	-
3	32	44100	2,884	μ.ε.κ.	✓
4	32	88200	3,800	μ.ε.κ.	✓
<b>Πίνακας 7.11 – 16PPM</b>					

## 7.4 Αξιολόγηση των Ισοσταθμιστών

Στην παράγραφο αυτή θα αξιολογηθούν οι τρεις αλγόριθμοι ισοστάθμισης LMS, RLS και CMA πάνω στο σύστημα τόσο από πλευράς bit error rate όσο και από πλευράς ρυθμού μετάδοσης για διάφορους συνδυασμούς των παραμέτρων τους. Σημειώνουμε ότι οι αλγόριθμοι ισοστάθμισης χρησιμοποιούνται μόνο για εκτίμηση του καναλιού (υπολογισμός των taps). Για την διόρθωση των σφαλμάτων του καναλιού χρησιμοποιείται η συνάρτηση filter που παίρνει σαν όρισμα τα taps αυτά. Με τον τρόπο αυτό πετυχαίνουμε την ίδια επίδοση από πλευράς εμφάνισης λαθών, με σημαντική μείωση του χρόνου επεξεργασίας του πακέτου (η συνάρτηση “filter” είναι πιο γρήγορη από την “equalize”).

### 7.4.1 LMS

Ο αλγόριθμος θα αξιολογηθεί βάσει της απόδοσής του σε BER και ρυθμό μετάδοσης για το 8-PSK, με  $T=10$  και  $F_s=88200$  και μέγεθος πακέτου 100kbits. Επιλέχθηκε το 8-PSK, γιατί το σύστημα αυτό επιτυγχάνει χαμηλό BER με μια καλή επιλογή ισοσταθμιστή, οπότε θα φανεί η επιρροή «κακών» επιλογών των παραμέτρων καλύτερα από π.χ ένα 4PSK. Οι παράμετροι που θα εξεταστούν είναι το μήκος του  $L$ , το μέγεθος του training sequence και το step size  $\mu$  προκειμένου να επιβεβαιωθούν οι θεωρητικές προβλέψεις του κεφαλαίου 4.

$\alpha/\alpha$	Train	$L$	$\mu/\mu_{\max}$	Tr.Rate	Av.BER
1	500	2	0,01	11,31	<b><math>\mu.ε.κ.</math></b>
2	500	4	0,01	11,33	<b><math>\mu.ε.κ.</math></b>
3	500	20	0,01	11,38	<b><math>\mu.ε.κ.</math></b>
4	200	2	0,01	12,41	<b><math>\mu.ε.κ.</math></b>
5	200	20	0,01	12,31	<b><math>\mu.ε.κ.</math></b>
6	100	4	0,01	12,85	<b><math>\mu.ε.κ.</math></b>
7	1000	4	0,01	9,762	<b><math>\mu.ε.κ.</math></b>
8	500	4	0,001	11,32	<b><math>\mu.ε.κ.</math></b>
9	500	4	0,01	11,30	<b><math>\mu.ε.κ.</math></b>
10	500	4	0,1	11,40	<b><math>\mu.ε.κ.</math></b>
11	500	4	0,5	11,40	$10^{-5}$
12	500	4	0,9	11,42	$5 \cdot 10^{-5}$
13	500	4	1,1	14,1	$3 \cdot 10^{-3}$
<b>Πίνακας 7.12 – LMS,</b>					

Αρχικά εξετάζεται η απόδοσή του LMS με βάση το μήκος  $L$ . Κρατάμε το μέγεθος του training sequence σταθερό και αλλάζουμε το  $L$  από 2 έως 20. Παρατηρούμε ότι ο ρυθμός μετάδοσης του αρχείου δεν επηρεάζεται από αυτή την παράμετρο, μάλλον γιατί το κανάλι μας δεν χρειάζεται πολλά taps για να μοντελοποιηθεί και η εκτίμησή του είναι γρήγορη ακόμα και για μεγάλο αριθμό taps. Ακόμα το BER δεν αυξάνεται σε μετρήσιμα μεγέθη (πάνω από  $10^{-5}$ ), πράγμα λογικό, αφού περισσότερα taps σημαίνει καλύτερη εκτίμηση του καναλιού και άρα λιγότερα λάθη. Εφόσον όμως ο ισοσταθμιστής δουλεύει καλά και για 2 taps, θα έπρεπε να δουλεύει και για 20, όπως και κάνει.

Έπειτα αλλάζουμε το μέγεθος του training sequence για τις ακραίες τιμές του  $L$  (2 και 20). Παρατηρούμε μια σαφή βελτίωση των χρόνων η οποία οφείλεται και στην επιτάχυνση της Amplitude/Phase Recovery η οποία επίσης

χρησιμοποιεί το training sequence. Για μέγεθος 100 ο χρόνος μειώνεται ακόμα περισσότερο και φτάνει κοντά στα 13kbps χωρίς να εμφανίζονται λάθη. Δημιουργείται όμως μεγαλύτερη πιθανότητα αστοχίας της συνάρτησης συγχρονισμού να ανιχνεύσει το σήμα για τόσο μικρό training.

Τέλος, αλλάζουμε τις τιμές του step size  $\mu$  από πολύ μικρές έως σχετικά μεγάλες. Όπως αναλύθηκε και θεωρητικά στο Κεφάλαιο 4, αύξηση του  $\mu$  επιταχύνει τη σύγκλιση, αλλά χειροτερεύει την επίδοση του συστήματος (βλ πίνακα 7.14 - μεταδόσεις 11-13). Όσο πλησιάζουμε το θεωρητικό μέγιστο του  $\mu$ , τόσο αρχίζουν να εμφανίζονται λάθη. Όταν το ξεπερνάμε, παρατηρούμε μια σαφή αύξηση του ρυθμού μετάδοσης, με παράλληλη αύξηση όμως των λαθών στην μετάδοση και ασταθή λειτουργία του αλγόριθμου (χαρακτηριστικό warning από Matlab). Επιβεβαιώνεται λοιπόν η θεωρία που θέλει το  $\mu$  να έχει πάντα τιμές κάτω από το μισό του μέγιστου θεωρητικού  $\mu$ .

### 7.4.2 RLS

Ομοίως με τον LMS, θα αξιολογήθηκε και ο RLS, για τις ίδιες παραμέτρους και πάνω στο ίδιο σύστημα (8PSK) και έτσι τα δεδομένα που προέκυψαν είναι συγκρίσιμα.

$\alpha/\alpha$	Train	L	$\lambda$	Tr.Rate	Av.BER
1	500	2	0,999	11,24	<b><math>\mu.ε.κ.</math></b>
2	500	4	0,999	11,28	<b><math>\mu.ε.κ.</math></b>
3	500	20	0,999	11,36	<b><math>\mu.ε.κ.</math></b>
4	500	40	0,999	11,13	<b><math>\mu.ε.κ.</math></b>
5	1000	4	0,999	9,650	<b><math>\mu.ε.κ.</math></b>
6	1000	20	0,999	9,642	<b><math>\mu.ε.κ.</math></b>
7	200	4	0,999	12,31	<b><math>\mu.ε.κ.</math></b>
8	200	20	0,999	12,30	<b><math>\mu.ε.κ.</math></b>
9	100	4	0,999	12,73	<b><math>\mu.ε.κ.</math></b>
10	50	10	0,999	12,81	<b><math>\mu.ε.κ.</math></b>
11	500	4	0,99	11,12	$10^{-2}$
12	500	4	0,9	11,12	$6 \cdot 10^{-3}$
13	500	4	0,8	11,27	$7 \cdot 10^{-3}$
<b>Πίνακας 7.13 – RLS,</b>					

Στις μετρήσεις 1-4 φαίνεται η επίδραση του L, τόσο στο ρυθμό μετάδοσης όσο και στο σφάλμα. Όπως και στον LMS, αυξάνοντας το μέγεθος του φίλτρου δεν βλέπουμε να επηρεάζεται ο ρυθμός μετάδοσης, αν και η εκτίμηση του

καναλιού στον RLS είναι πιο περίπλοκη και άρα θεωρητικά πιο χρονοβόρα. Οι καθυστερήσεις που μπαίνουν σε κάθε μετάδοση είναι πολύ περισσότερο μεταβλητές, τόσο που υπερκαλύπτουν την όποια διαφορά.

Αυξάνοντας το μέγεθος του training sequence (βλ μετρήσεις 5 και 6) πέφτει αισθητά ο ρυθμός μετάδοσης καθώς ο αλγόριθμος πρέπει να επεξεργαστεί περισσότερη πληροφορία για να εκτιμήσει το κανάλι. Ανάλογα, μειώνοντας το ίδιο μέγεθος μεγαλώνει ο ρυθμός μετάδοσης σε σημαντικό μάλιστα βαθμό. Στην μετάδοση μάλιστα 10, το σύστημα καταφέρνει να λειτουργήσει με ικανοποιητικά με training sequence μόλις 50bits σε πακέτο 100.000 bits με θετικό αντίκτυπο στο ρυθμό μετάδοσης. Κάτι τέτοιο θέτει σε κίνδυνο την αξιοπιστία του συστήματος καθώς δεν ήταν λίγες οι φορές που είχαμε αστοχία της συνάρτησης συγχρονισμού για τόσο μικρή ακολουθία.

Τέλος όπως φαίνεται στις μετρήσεις 10-13 του πίνακα 7.15 η μεταβολή του  $\lambda$ , δεν επηρεάζει τον χρόνο αποστολής όπως προβλέπεται και στη θεωρία στο Κεφάλαιο 5.4.1. Αντίθετα επηρεάζει το BER, το οποίο ανεβαίνει όσο το  $\lambda$  απομακρύνεται από το 1, πράγμα που τονίστηκε στην θεωρία ως ένα από τα δύο σημαντικότερα μειονεκτήματα του RLS (αστάθεια).

### 7.4.3 CMA

Ομοίως με τους άλλους δύο αλγόριθμους, αξιολογήσουμε και την απόδοση του CMA. Επιλέξαμε και πάλι το 16PSK, αυτή τη φορά γιατί το QAM δεν λειτούργησε ικανοποιητικά με τον CMA, τουλάχιστον για  $M > 8$ . Οι μετρήσεις για διάφορα μήκη  $L$  και step sizes φαίνονται στον παρακάτω πίνακα:

$\alpha/\alpha$	L	Step size	Tr.Rate	Av.BER
2	4	0,05	12,42	$<10^{-5}$
3	20	0,05	12,49	<b>μ.ε.κ.</b>
4	40	0,05	12,59	<b>μ.ε.κ.</b>
5	8	0,0001	12,52	<b>μ.ε.κ.</b>
6	8	0,001	12,32	<b>μ.ε.κ.</b>
7	8	0,01	12,43	<b>μ.ε.κ.</b>
8	8	0,1	12,44	$\approx 5 \cdot 10^{-5}$
9	8	0,2	12,78	$\approx 10^{-5}$
10	8	0,9	12,39	$\approx 10^{-3}$
11	8	1,1	12,98	$\approx 10^{-2}$
<b>Πίνακας 7.14 – CMA,</b>				

Για το μέγεθος του φίλτρου, ισχύουν αυτά που ισχύουν και για τους LMS και RLS, δηλαδή δεν επηρεάζει τον ρυθμό μετάδοσης, τουλάχιστον όχι σε μέγεθος μεγαλύτερο του στατιστικού σφάλματος των παραπάνω μετρήσεων. Πάντως παρατηρούμε ότι η ταχύτητα του αλγόριθμου αυτού είναι αρκετά μεγαλύτερη των άλλων δύο, το οποίο οφείλεται στην έλλειψη του training sequence που επιβάρυνε τους LMS και RLS τόσο σαν πρόσθετη πληροφορία που μεγάλωνε το πακέτο όσο και σαν χρόνος επεξεργασίας της ακολουθίας αυτής.

Αν και η γενική εικόνα του CMA είναι αυτή που παρουσιάζεται στον πίνακα 7.14, η αστάθεια στην απόδοσή του ήταν αρκετά μεγαλύτερη των άλλων δύο αλγόριθμων που υλοποιήθηκαν. Ακόμα και για μικρά step sizes και μεγάλα L, κάποιες φορές εμφανίστηκαν πολλά λάθη. Δεδομένου ότι οι μετρήσεις έγιναν σε διαφορετικές μέρες και σε δύο διαφορετικούς χώρους, είναι πιθανόν η αρχικοποίηση του αλγόριθμου, η οποία παίζει σημαντικό ρόλο στον CMA, να ήταν διαφορετική (λόγω καναλιού) και κάποιες φορές να μην έφτανε σε λύση. Για μικρότερα M πάντως, όπου το σύστημα δουλεύει συνήθως και χωρίς ισοσταθμιστή, δεν παρατηρήθηκε πρόβλημα με λάθη ενώ τα κέρδη από την αύξηση της ταχύτητας ήταν σημαντικότερα.

Αντίθετα με QAM, ο CMA δεν κατάφερε να πετύχει την απόδοση του PSK. Συγκεκριμένα για 16-QAM έβγαζε λάθη, ενώ κάποιες φορές δεν διόρθωσε λάθη και για 8QAM με συχνότητα 88200.

## 7.5 Επιλεγμένα Συστήματα

Από τα συμπεράσματα που παρατέθηκαν παραπάνω, προκύπτει ότι όλα τα συστήματα (δηλαδή συνδυασμοί παραμέτρων) δεν κατάφεραν να λειτουργήσουν ικανοποιητικά. Στην παράγραφο αυτή θα παρατεθούν τα συστήματα εκείνα που λειτούργησαν στην πράξη με χαμηλό ρυθμό λαθών και στο τέλος θα παρατεθούν εκείνα που πέτυχαν την καλύτερη απόδοση από πλευράς ταχύτητας μετάδοσης.



---

## ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ

---

- [1]. *Proakis J.G., Salehi M.* “Συστήματα Τηλεπικοινωνιών” Κεφάλαιο 8, Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών 2002.
- [2]. *Proakis J.G.* “Mobile Communications Hanbook – Channel Equalization”, CRC Press 1999.
- [3]. *Haykin S.*, “Adaptive Filter Theory”, Chapters 9,13,16,18, Prentice Hall.
- [4]. *Dollas A.*, “The art of Microelectronic Circuits”, Chapters 2&3, Technical Univeristy of Crete 2003.
- [5]. *Widrow B. Stearns S.*, “Adaptive Signal Processing”, Chapter 6, Prentice Hall 1985.
- [6]. *Clarkson P.M.*, “Optimal and Adaptive Signal Processing”, Chapter 4, CRC Press 1993.
- [7]. *Vaseghi S.V.*, “Advanced Digital Signal Processing and Noise Reduction” 2<sup>nd</sup> Edition, Chapters 6&7, Wiley 2000.
- [8]. *Alexander S.T.*, “Adaptive Signal Processing, Theory and Applications”, Springer-Verlang 1986.
- [9]. *Smalley D.* “Equalization Concepts” σελ 1-6, Atalanta Regional Technology Center, Application Report, 1994
- [10]. *Friedlander B.*, “Adaptive algorithms for finite impulse response filters”, Chapter 2, Prentice Hall, 1985.
- [11]. *Li Y., Ding Z.*, “Convergence of Finite Length Blind Adaptive Equalizers”, Chapter 2, IEEE Transactions on Signal Processing, vol.43, 1995
- [12]. *Johnson C.R.*, Blind Equalization Using the Constant Modulus Criterion: A Review, Proceedings of the IEEE, vol 77, No 10, 1989.
- [13]. *Yousef N.R.*, “A feedback Analysis of the Tracking Performance of Blind Adaptive Equalization Algorithms”, Proceedings of the 38<sup>th</sup> Conference on Decision & Control, Phoenix, Arizona, USA, December 1999.