An LMS-based Equalizer for a PMF-based Telecommunications Application

National Technical University of Athens

School of Electrical and Computer Engineering

Division of Communication, Electronic and Information Engineering

Panagiotis Grafakos

Advisor: Prof. Yannis Papananos

Athens, October 2023

An LMS-based Equalizer for a PMF-based Telecommunications Application

National Technical University of Athens

School of Electrical and Computer Engineering
Division of Communication, Electronic and Information Engineering

Panagiotis Grafakos
Advisor: Prof. Yannis Papananos

Εγκρίθηκε από την τριμελή επιτροπή την 5η Οκτωβρίου 2023.

....	....	....
Ι. Παπανάνος	Δ. Κακλαμάνη	Ε. Χριστοφόρου
Καθηγητής Ε.Μ.Π.	Καθηγήτρια Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

....
Παναγιώτης Γραφάκος
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών
Ε.Μ.Π.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Στην παρούσα διπλωματική εργασία παρουσιάζεται η ανάπτυξη ενός εξισορροπιστή Ελάχιστων Μέσων Τετραγώνων (LMS) για μια ζεύξη δεδομένων υψηλής ταχύτητας που χρησιμοποιεί Πολυμερικές Μικροκυματικές Ίνες (PMF) ως μέσο μετάδοσης. Παρουσιάζονται τόσο η ανάπτυξη του προσαρμοστικού αλγορίθμου, όσο και η σχεδίαση και πραγματοποίηση των βοηθητικών ολοκληρωμένων κυκλωμάτων που απαιτούνται για τη λειτουργία του εξισορροπιστή.

Αρχικά παρουσιάζεται η ζεύξη δεδομένων και εξηγείται η επιλογή του PMF για τη συγκεκριμένη εφαρμογή. Κατόπιν θίγεται η ανάγκη εισαγωγής ενός εξισορροπιστή, και παρατίθενται οι προσεγγίσεις που μπορούν να ακολουθηθούν για να πραγματοποιηθεί η εξισορρόπηση. Αφού αξιολογηθούν και συγκριθούν, επιλέγεται να υλοποιηθεί ένας εξισορροπιστής με δυνατότητα προσαρμογής στην πλευρά του δέκτη.

Στη συνέχεια γίνεται μια εισαγωγή στη θεωρία προσαρμοστικού ελέγχου, και παρουσιάζονται κάποιες συνήθεις προσεγγίσεις. Μετά από μια διαδικασία αξιολόγησης και σύγκρισης, καταλήγουμε στο συμπέρασμα ότι η προτιμότερη υλοποίηση του προσαρμοστικού αλγορίθμου για την εφαρμογή μας είναι ο αλγόριθμος LMS.

Κατόπιν αναπτύσσονται τα ολοκληρωμένα κυκλώματα του αναλογικού μέρους του εξισορροπιστή, συγκεκριμένα ένα φίλτρο Butterworth, ένας αναλογικός πολλαπλασιαστής και ένας μετατροπέας ψηφιακού σε αναλογικό σήμα. Εξηγείται η λειτουργία αυτών των κυκλωμάτων και παρουσιάζονται τα σχηματικά τους και τα κυκλώματα που χρησιμοποιήθηκαν για την πόλωσή τους.

Εν τέλει προσομοιώνεται το σύστημα που αναπτύχθηκε στο πλαίσιο αυτής της εργασίας, και επαληθεύεται και αξιολογείται η λειτουργία του.

Λέξεις-Κλειδιά: Προσαρμοστικός Εξισορροπιστής, Ελάχιστα Μέσα Τετράγωνα, Αλγόριθμος Σταθερού Πλάτους, Πολυμερική Μικροκυματική Ίνα, Φίλτρο Butterworth, Αναλογικός Πολλαπλασιαστής

Abstract

In the present thesis, the development of a Least Mean Squares equalizer for a high speed data link using Polymer Microwave Fibers as transmission medium is presented. Both the development of the adaptive algorithm, as well as the design and implementation of the auxiliary Integrated Circuits necessary for the operation of the equalizer is shown.

We initially present the data link and explain the choice of transmission medium for this application. Next, we discuss the need for the inclusion of an equalizer in the data link, and proceed to present the different approaches that can be taken to perform equalization. After evaluating and comparing them, we reach the conclusion that the use of a receiver-side equalizer capable of adaptation is the optimal implementation for our application.

We then discuss the theory behind adaptive control engineering, and present some common approaches. Through an evaluation and comparison process similar to that mentioned above, we reach the conclusion that the preferred implementation of the adaptive algorithm for this process is the Least Mean Squares algorithm.

In the next steps we develop the Integrated Circuits that will make up the analog parts of the equalizer, namely a Butterworth Filter, an analog multiplier, and a Digital to Analog Converter. We explain the function of these circuits, present the schematics used for their realization, and also the biasing circuits used to ensure their proper operation.

Finally, we perform a simulation of the complete system developed within this thesis, verify its functionality and evaluate its performance.

Keywords: Adaptive Equalizer, Least Mean Squares, Constant Modulus Algorithm, Polymer Microwave Fiber, Butterworth Filter, Analog Multiplier

Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κύριο Ιωάννη Παπανάνο, για την επίβλεψη και την πολύτιμη γνώση που μου παρείχε καθ΄ όλη τη διαδικασία ολοκλήρωσης αυτής της εργασίας. Θα ήθελα επίσης να ευχαριστήσω τον κ. Παπανάνο για το γεγονός ότι μου παρείχε τη δυνατότητα να εκπονήσω αυτή τη διπλωματική ως πρακτικάριος της Infineon Technologies AG, όπου μπόρεσα να αναπτύξω μια καλύτερη ιδέα του πώς λειτουργεί μια εταιρεία αυτής της κλίμακας, αλλά και του πώς εργάζεται ένας μηχανικός.

Θα ήθελα επίσης να εκφράσω τις ευχαριστίες μου προς τον Δρ. Siegfried Krainer και τη Δρ. Parastuty Zulaicha για την υποστήριξη και τη συνεργασία τους κατά τη διάρκεια της πρακτικής μου, καθώς και το εξαιρετικό εργατικό κλίμα που παρέχουν σε όλα τα μέλη της ομάδας μας.

Μια ιδιαίτερη ευχαριστία οφείλεται και στους φίλους και συνεργάτες, υποψήφιους διδάκτορες υπό την επίβλεψη του κ. Παπανάνου, Βασίλη Λιακώνη και Βασίλη Μανουρά, που με υποστήριζαν και ήταν πάντα διαθέσιμοι για να παρέχουν πραγματικά πολύτιμη βοήθεια.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου, τους γονείς μου Στάθη και Γιάννα, και την αδερφή μου Μαρίζα, καθώς και τους φίλους μου που, με τη φροντίδα και την υποστήριξή τους, μου επέτρεψαν να εκμεταλλευτώ στο έπακρο τις ευκαιρίες που μου προσφέρθηκαν.

Acknowledgements

Firstly, I would like to thank my supervising professor, Pr. Yannis Papananos, for his oversight and valuable insight throughout the process of completing this work. I would also like to thank Professor Papananos for giving me the opportunity to do this thesis as an intern within Infineon Technologies AG, where I developed a better understanding of how a company of that scale operates, and the working life of an engineer in general.

I would also like to extend my thanks to Dr. Siegfried Krainer and Dr. Parastuty Zulaicha for their support and cooperation throughout my internship, as well as providing me and every other member of our team with an excellent working environment.

A special thanks is also due to my friends and co-workers, and doctorate students under Pr. Papananos, Vasilis Liakonis and Vasilis Manouras, for supporting me and always being available with valuable help.

Finally, I would like to thank my family, my parents Stathis and Yanna, and my sister Mariza, as well as all the friends who, with their care and support, allowed me to make the best of all the opportunities offered to me.

Εκτεταμένη Περίληψη

Με την ανάπτυξη τόσο των τεχνολογιών που χρησιμοποιούνται στα τηλεπικοινωνιακά συστήματα όσο και των απαιτήσεων των χρηστών, οι ταχύτητες που επιτυγχάνονται αυξάνονται διαρκώς, ωθώντας τα συστήματα τηλεπικοινωνιών στα όρια της λειτουργίας τους. Το γεγονός αυτό καθιστά τη διόρθωση των τηλεπικοινωνιακών σημάτων ακόμα πιο αναγκαία απ΄ ό,τι στα συστήματα παλαιότερων γενεών, και είναι αυτή η ανάγκη στην οποία επιχειρούμε να ανταποκριθούμε με αυτή τη διπλωματική εργασία.

Στο πλαίσιο αυτής της διπλωματικής αναπτύξαμε έναν εξισορροπιστή για μια ενσύρματη ζεύξη δεδομένων υψηλής ταχύτητας που λειτουργεί στο εύρος συχνοτήτων sub-THz, ο οποίος προσπαθεί να διορθώσει τα μεταδιδόμενα δεδομένα μέσω μιας προσαρμοστικής διαδικασίας, έχοντας τη δυνατότητα να αντισταθμίσει αναπάντεχες μεταβολές στη συμπεριφορά του καναλιού κατά τη διάρκεια της μετάδοσης, βελτιώνοντας τη συμπεριφορά όλου του συστήματος.

Εισαγωγή στην Εξισορρόπηση

Ο όρος εξισορρόπηση αναφέρεται σε μια διαδικασία ελέγχου η οποία επιχειρεί να ανακατασκευάσει ένα παραμορφωμένο σήμα που έχει μεταδοθεί μέσω ενός καναλιού που εισάγει αυτή την παραμόρφωση, αντισταθμίζοντας την επίδραση του καναλιού στο μεταδιδόμενο σήμα. Μπορεί επομένως να ειδωθεί ως ένα κλασικό πρόβλημα αποσυνέλιξης, με τον εξισορροπιστή να προσπαθεί να απομονώσει το σήμα $s(t)$ από το ληφθέν

$$r(t) = \int_{\tau=-\infty}^{\infty} s(t)h(\tau - t)d\tau \tag{1}$$

όπου με $h(t)$ συμβολίζεται η κρουστική απόκριση του καναλιού. Αν η έξοδος του εξισορροπιστή συμβολιστεί με $y(t)$ και η απόκριση συχνότητάς του με $H_{\text{eq}}(\omega)$, καταλήγουμε στην εξίσωση

$$Y(\omega) = H_{\mathrm{eq}}(\omega)R(\omega) \tag{2}$$

όπου $Y(\omega)$ και $R(\omega)$ είναι τα φάσματα συχνότητας των σημάτων εξόδου και εισόδου του εξισορροπιστή, αντίστοιχα. Αν η απόκριση συχνότητας του καναλιού είναι η $H(\omega)$, προκύπτει η εξίσωση

$$Y(\omega) = H_{\mathrm{eq}}(\omega)H(\omega)S(\omega) \tag{3}$$

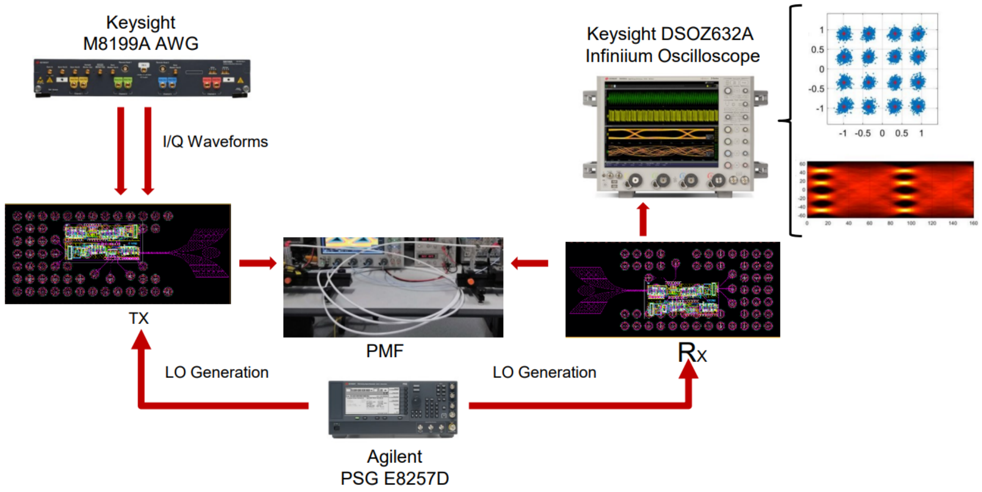
Επομένως, στην ιδανική περίπτωση, όπου $Y(\omega) = S(\omega)$, προκύπτει ότι

$$H^{*}_{\mathrm{eq}}(\omega) = \frac{1}{H(\omega)} \tag{4}$$

και άρα ο εξισορροπιστής έχει συμπεριφορά αντίστροφη αυτής του καναλιού.

Σύστημα προς Εξισορρόπηση

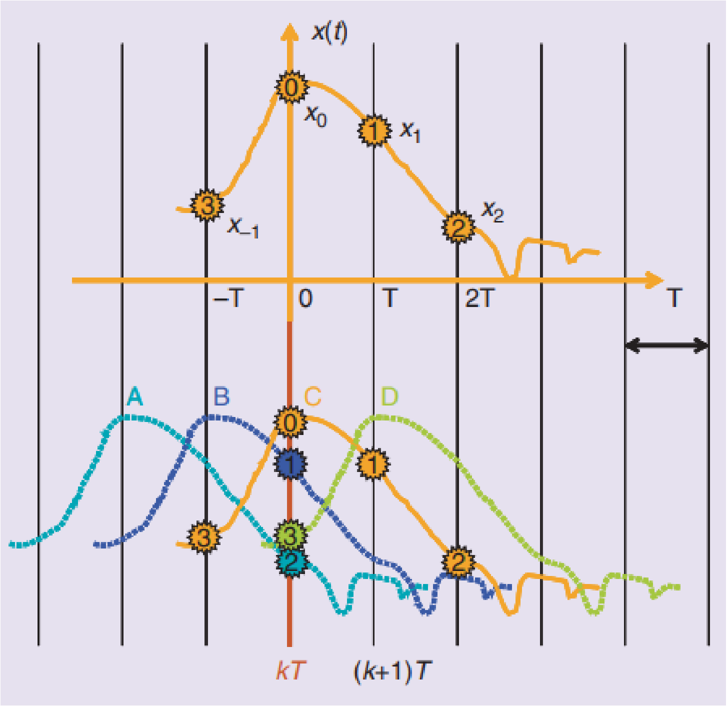
Το τηλεπικοινωνιακό σύστημα που καλείται να αντισταθμίσει ο εξισορροπιστής που παρουσιάζεται εδώ είναι ένα ενσύρματο σύστημα επικοινωνίας που μεταδίδει τα σήματα μέσω μιας Πολυμερικής Μικροκυματικής Ίνας (PMF). Απαρτίζεται από ένα ζεύγος πομπού/δέκτη που χρησιμοποιούν κεραίες Vivaldi και μίκτες, ενισχυτές ισχύος και πολλαπλασιαστές συχνοτήτων που έχουν περιγραφεί σε προηγούμενες εργασίες[1, 2, 3]. Ο πομπός και ο δέκτης δέχονται σήματα τοπικού ταλαντωτή (LO) και το σήμα προς μετάδοση διαμορφωμένο με τη μέθοδο QPSK από εξωτερικές γεννήτριες σήματος, και η μετάδοση γίνεται μέσω ενός PMF. Μια υλοποίηση του συστήματος με στόχο τη μέτρησή του φαίνεται στο ακόλουθο Σχήμα 1.



Σχήμα 1: Σύστημα προς εξισορρόπηση

Πηγές Παραμόρφωσης

Τα τηλεπικοινωνιακά συστήματα επηρεάζονται από πολλά φαινόμενα που προκαλούν παραμόρφωση στο μεταδιδόμενο σήμα. Στην περίπτωση των ενσύρματων τηλεπικοινωνιών τα πιο συνήθη από αυτά είναι το επιδερμικό φαινόμενο στα μεταλλικά μέσα μετάδοσης, οι ηλεκτρομαγνητικές παρεμβολές τύπου διασταυρούμενης συνομιλίας από σήματα που μεταδίδονται μέσω γειτονικών διαδρομών μετάδοσης, η καμπή του μέσου, και η παρεμβολή μεταξύ συμβόλων, που οφείλεται στην εξάρτηση της ταχύτητας ομάδας από τη συχνότητα. Στην πιο συγκεκριμένη περίπτωση του PMF, καθώς το καλώδιο μετάδοσης είναι κατασκευασμένο από πολυμερή, δεν υπάρχουν μεταλλικές απώλειες όπως το επιδερμικό φαινόμενο. Επιπλέον, τα υλικά αυτά είναι πιο ανθεκτικά σε ηλεκτρομαγνητικές παρεμβολές, και έχει ήδη δειχθεί ότι η καμπή του υλικού προκαλεί περιορισμένο πρόβλημα στην ποιότητα του σήματος. Το πλέον σημαντικό από τα φαινόμενα που συνεισφέρουν στην παραμόρφωση του σήματος, επομένως, είναι η παρεμβολή συμβόλων, συνήθως συμβολιζόμενη με το ακρωνύμιο ISI. Η παρεμβολή που προκαλείται από την επικάλυψη γειτονικών συμβόλων σκιαγραφείται στο Σχήμα 2. Με την παραμόρφωση του συμβόλου έτσι ώστε να διαρκεί για περισσότερες από μία περιόδους συμβόλου, τα γειτονικά σύμβολα επικαλύπτονται, έτσι ώστε το λαμβανόμενο σύμβολο να περιέχει πληροφορίες από περισσότερα από ένα μεταδιδόμενα σύμβολα. Η μαθηματική έκφραση που περιγράφει αυτή



Σχήμα 2: Επικάλυψη Συμβόλων

την παραμόρφωση παρατίθεται παρακάτω.

$$y(t) = \sum_{k=-L}^{L} p(t) * a(t - kT_b) \quad (5)$$

Τύποι Εξισορροπιστών

Στη βιβλιογραφία έχουν κατά καιρούς προταθεί πολλές εναλλακτικές υλοποιήσεις εξισορροπιστών, οι οποίες μπορούν να καταταχθούν σε κατηγορίες, η κάθε μία από τις οποίες προσφέρει πλεονεκτήματα και μειονεκτήματα [4, 5]. Μία πρώτη βασική διάκριση που μπορεί να γίνει μεταξύ των διαθέσιμων λύσεων είναι σχετικά με την πλευρά της ζεύξης στην οποία επιχειρείται η εξισορρόπηση, αυτή του πομπού ή αυτή του δέκτη. Στην πρώτη περίπτωση, έχοντας μια a priori γνώση των χαρακτηριστικών του καναλιού, μπορεί να εισαχθεί τεχνητή απόσβεση στις περιοχές του φάσματος που το κανάλι δεν θα αποσβέσει τόσο έντονα, ή να ενισχυθούν προκαταβολικά οι περιοχές που θα αντιμετωπίσουν πιο ισχυρή απόσβεση κατά τη μετάδοσή τους, με στόχο την ομοιόμορφη απόσβεση του σήματος σε όλο το εύρος φάσματός του.

Μία ακόμα διάκριση που μπορεί να γίνει βασίζεται στα στοιχεία που αποτελούν τον εξισορροπιστή, και συγκεκριμένα στο κατά πόσο αυτά είναι παθητικά ή ενεργητικά. Στην πρώτη περίπτωση ο εξισορροπιστής καταναλώνει σημαντικά λιγότερη ισχύ και παρέχει ευκολία στην αρχική ρύθμιση της απόκρισής του, αλλά αδυναμεί να ανταπεξέλθει σε παρεκκλίσεις από την αναμενόμενη συμπεριφορά του καναλιού, καθώς δεν μπορεί να προσαρμοστεί εκ των υστέρων.

Μια επιπλέον διάκριση μπορεί να γίνει μνεταξύ γραμμικών και μη γραμμικών υλοποιήσεων, με τις υλοποιήσεις που περιέχουν βρόχους ανάδρασης να ανοίκουν στη δεύτερη κατηγορία. Οι υλοποιήσεις αυτές μπορούν να αντιμετωπίσουν καλύτερα το θόρυβο που εισάγεται στην είσοδο του εξισορροπιστή, αλλά λειτουργούν βάσει της προϋπόθεσης ότι οι προηγούμενες αποφάσεις που έχουν ληφθεί κατά τη δαδικασία της εξισορρόπησης ήταν σωστές, και επομένως πάσχουν συχνά από το φαινόμενο διάδοσης σφάλματος.

Τέλος, μπορούμε να διαχωρίσουμε τους εξισορροπιστές σε προσαρμοστικούς και μη. Οι πρώτοι παρέχουν τη δυνατότητα αντιμετώπισης φαινομένων που δεν είχαν ληφθεί υπ΄ όψιν κατά τη σχεδίαση του εξισορροπιστή, άγνωστων χαρακτηριστικών του καναλιού μετάδοσης, ή ακόμα και αναπάντεχων μεταβολών στο κανάλι. Έχουν ωστόσο το σημαντικό μειονέκτημα της σημαντικής κατανάλωσης ισχύος, και της υποχρεωτικής χρήσης ψηφιακών κυκλωμάτων σε πολλές υλοποιήσεις [4].

Για το σκοπό αυτής της εργασίας, λαμβάνοντας υπ΄ όψιν τα πλεονεκτήματα και μειονεκτήματα των διάφορων προσεγγίσεων, αποφασίσαμε να υλοποιήσουμε τον εξισορροπιστή ως ένα προσαρμοστικό φίλτρο στην πλευρά του δέκτη.

Προσαρμοστικοί Αλγόριθμοι

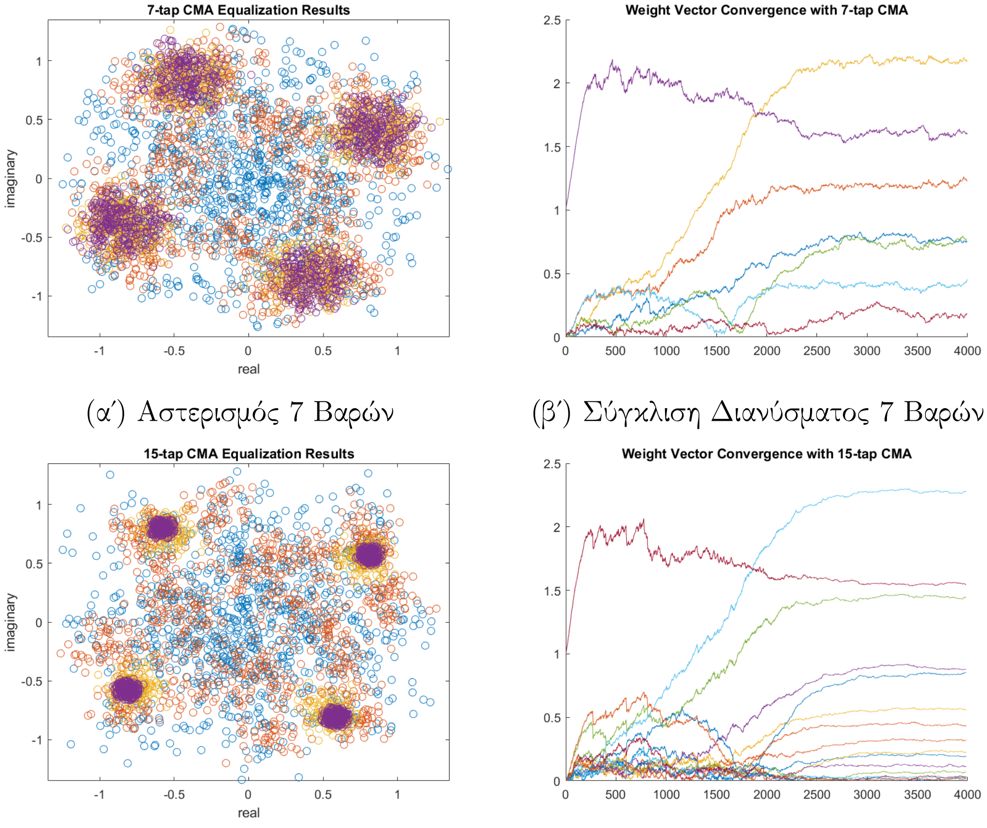
Στη βιβλιογραφία έχουν αναπτυχθεί πολλές μέθοδοι υλοποίησεις προσαρμοστικών αλγορίθμων. Στο πλαίσιο της εργασίας αυτής θεωρήσαμε δύο προσεγγίσεις, αυτή της μεθόδου Ελάχιστων Μέσων Τετραγώνων (LMS) και αυτή του Αλγόριθμου Σταθερού Πλάτους (CMA). Και οι δύο μέθοδοι, ώντας προσαρμοστικοί αλγόριθμοι, επιχειρούν να ενημερώσουν ένα διάνυσμα βαρών ούτως ώστε να ελαχιστοποιήσουν κάποια συνάρτηση-στόχο. Η συνάρτηση αυτή επιλέγεται έτσι ώστε να ελαχιστοποιείται όταν η έξοδος του εξισορροπιστή ταυτιστεί με κάποιο επιθυμητό σήμα αναφοράς. Η διαφορά μεταξύ των δύο αλγορίθμων έγκειται στην επιλογή της συνάρτησης στόχου αλλά και στη μέθοδο εξαγωγής του σήματος σφάλματος.

Στην περίπτωση του LMS το σήμα εξόδου του εξισορροπιστή συγκρίνεται συνήθως με μια ακολουθία εκπαίδευσης, μια ακολουθία συμβόλων που μεταδίδεται από τον πομπό αλλά είναι γνωστή στο δέκτη, ούτως ώστε ο δεύτερος να συγκρίνει το ληφθέν σήμα με το μεταδιδόμενο. Μια εναλλακτική προσέγγιση είναι η λεγόμενη τυφλή εξισορρόπιση, κατά την οποία δεν υπάρχει γνώση της μεταδιδόμενης ακολουθίας συμβόλων στην πλευρά του δέκτη, ο οποίος καλείται να παράξει μόνος του το σήμα αναφοράς. Μια συνηθισμένη μέθοδος για να επιτευχθεί αυτό είναι η χρήση κάποιου συστήματος απόφασης που ¨αποκωδικοποιεί’ την έξοδο του εξισορροπιστή και την ωθεί στον ιδανικό αστερισμό. Αφού πραγματοποιηθεί η σύγκριση του σήματος εξόδου με το ιδανικό, ο αλγόριθμος LMS επιχειρεί να ελαχιστοποιήσει το μέσο τετράγωνο του σφάλματος.

Στην περίπτωση του CMA η εξισορρόπιση είναι πάντα τυφλή, καθώς ο αλγόριθμος αυτός εκμεταλλεύεται την ιδιότητα κάποιων τεχνικών διαμόρφωσης να παράγουν σύμβολα με ίδιο πλάτος, και επομένως συγκρίνει τα εξισορροπημένα σύμβολα με ένα σταθερό σήμα που αντιπροσωπεύει το ιδανικό πλάτος που θα έπρεπε να είχαν τα σύμβολα αν δεν είχαν παραμορφωθεί.

Αφού καταλήξαμε στις εξισώσεις που περιγράφουν τη διαδικασία ενημέρωσης του διανύσματος βαρών που περιγράφει το φίλτρο, πραγματοποιήσαμε προσομοιώσεις των δύο αυτών λύσεων στο περιβάλλον της Matlab ούτως ώστε να συγκρίνουμε την αποτελεσματικότητά τους. Αρχικά συγκρίναμε δύο υλοποιήσεις του αλγόριθμου CMA με διανύσματα βαρών διαφορετικού μήκους (7 και 15 βάρη), καταλήγοντας στο συμπέρασμα ότι η αύξηση του αριθμού των βαρών βελτιώνει τη σύγκλιση και το τελικό σφάλμα, αλλά δεν λύνει το βασικό πρόβλημα αυτού του αλγορίθμου, την αδυναμία του να αντισταθμίσει την καθυστέρηση φάσης που εισάγεται από το κανάλι. Οι γραφικές παραστάσεις που προέκυψαν από τις προσομοιώσεις φαίνονται στο Σχήμα 3.

Στη συνέχεια υλοποιήσαμε τον αλγόριθμο LMS με τρεις διαφορετικούς τρόπους: αρχικά με τη χρήση μιας ακολουθίας εκπαίδευσης, στη συνέχεια ως έναν τυφλό αλγόριθμο που τερματίζει όταν το σφάλμα αποκτήσει μηδενική τιμή, και τέλος ως έναν τυφλό αλγόριθμο με ανοχή σφάλματος, ο οποίος τερματίζει όταν το σφάλμα πέσει κάτω από ένα αποδεκτό κατώφλι. Από τη σύγκριση αυτή καταλήξαμε στο συμπέρασμα ότι η ακολουθία εκπαίδευσης δεν είναι απαραίτητη, και ότι η εισαγωγή ενός αποδεκτού επιπέδου σφάλματος καθιστά τη διαδικασία εξαιρετικά γρήγορη. Τα αποτελέσματα των προσομοιώσεων παρουσιάζονται στα Σχήματα 4 και 5.



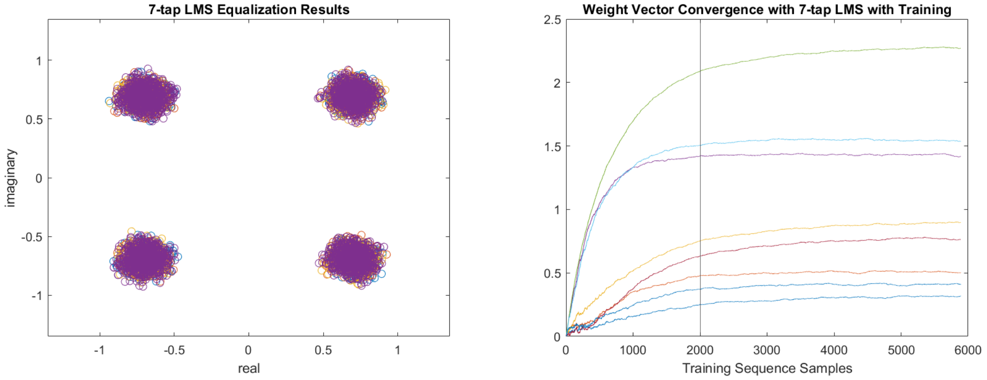
(α΄) Αστερισμός 7 Βαρών

(β΄) Σύγκλιση Διανύσματος 7 Βαρών

(γ΄) Αστερισμός 15 Βαρών

(δ΄) Σύγκλιση Διανύσματος 15 Βαρών

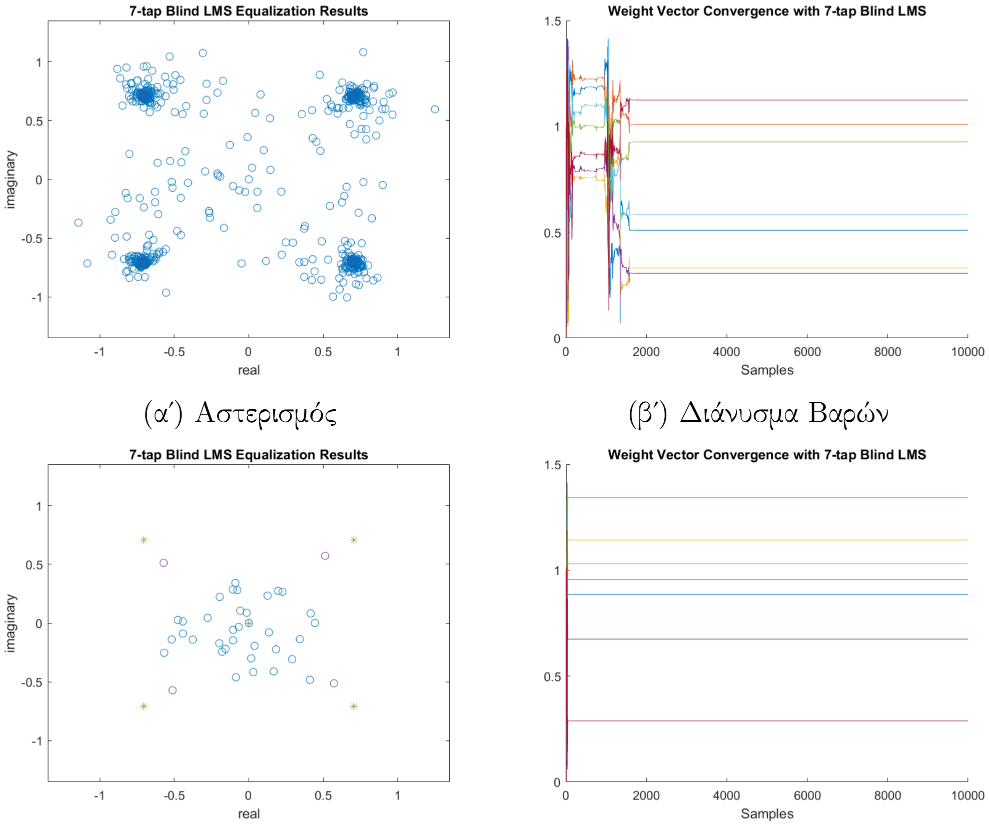
Σχήμα 3: Σύγκριση Συμπεριφοράς CMA: τα (α) και (β) δείχνουν τη συμπεριφορά του συστήματος 7 βαρών, τα (γ) και (δ) αυτή του συστήματος 15 βαρών



(α΄) Αστερισμός

(β΄) Διάνυσμα Βαρών

Σχήμα 4: Αλγόριθμος LMS με Ακολουθία Εκπαίδευσης



(α′) Αστερισμός

(β′) Διάνυσμα Βαρών

(γ′) Αστερισμός

(δ′) Διάνυσμα Βαρών

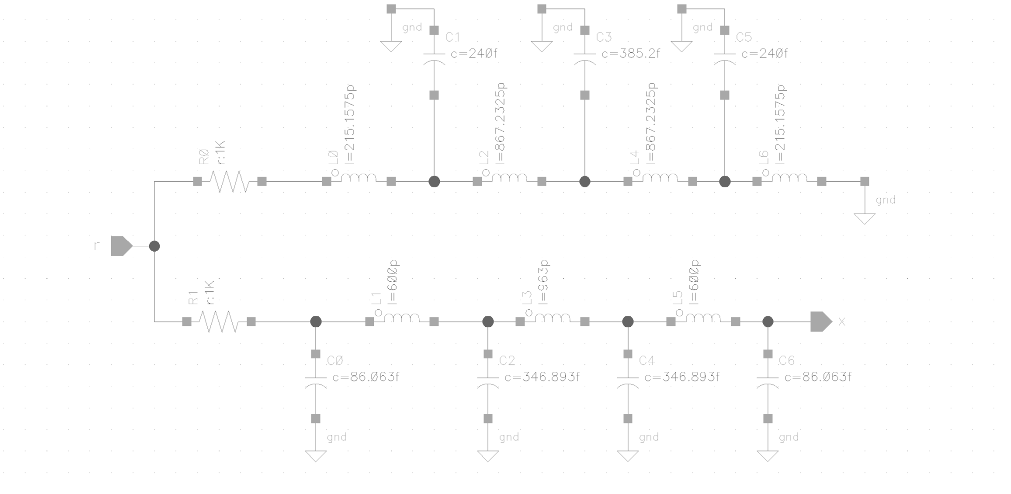
Σχήμα 5: Αλγόριθμος LMS χωρίς Ακολουθία Εκπαίδευσης: τα (α′) και (β′) δείχνουν τη συμπεριφορά χωρίς κατώφλι αποδεκτού σφάλματος, τα (γ′) και (δ′) με κατώφλι στο 0.3

Από τις προσομοιώσεις αυτές και τη σύγκριση των αποτελεσμάτων τους καταλήξαμε στο συμπέρασμα ότι η βέλτιστη υλοποίηση του προσαρμοστικού φίλτρου είναι αυτή με τη χρήση του αλγόριθμου LMS με την προσθήκη του κατωφλιού αποδεκτού σφάλματος.

Υλοποίηση Συστήματος

Για την υλοποίηση των αναλογικών ολοκληρωμένων κυκλωμάτων που θα αποτελέσουν το σύστημα του εξισορροπιστή χρησιμοποιήθηκε η τεχνολογία B11HFC της Infineon Technologies AG η οποία χρησιμοποιεί το υλικό SiGe για την κατασκευή διπολικών τρανζίστορ υψηλής ταχύτητας. Τα κυκλώματα που σχεδιάστηκαν ήταν ένα αναλογικό βαθυπερατό φίλτρο τύπου Butterworth για την εισαγωγή καθυστέρησης στο σήμα που λαμβάνει ο δέκτης, ένας διαγωγός ρυθμιζόμενου κέρδους ο οποίος λειτουργεί ως αναλογικός πολλαπλασιαστής, και ένας μετατροπέας ψηφιακού σε αναλογικό σήμα για την μετατροπή του διανύσματος βαρών που προκύπτει από την ψηφιακή λογική, και άρα είναι ψηφιακής μορφής, σε αναλογικό σήμα, ούτως ώστε να μπορέσει να πολλαπλασιαστεί με το σήμα που λαμβάνει ο δέκτης.

Το σχηματικό του φίλτρου παρουσιάζεται στο Σχήμα 6. Όπως φαίνεται από το σχηματικό αυτό, το φίλτρο είναι δυαδικό, ούτως ώστε κάθε στοιχείο του διανύσματος κατάστασής του να είναι διαθέσιμο ως τάση, και άρα να μπορεί να χρησιμοποιηθεί ως είσοδος του πολλαπλασιαστή. Οι τιμές των στοιχείων του φαίνονται στους Πίνακες 1 και 2. Η απόκριση συχνότητάς του φαίνεται στο Σχήμα 7.



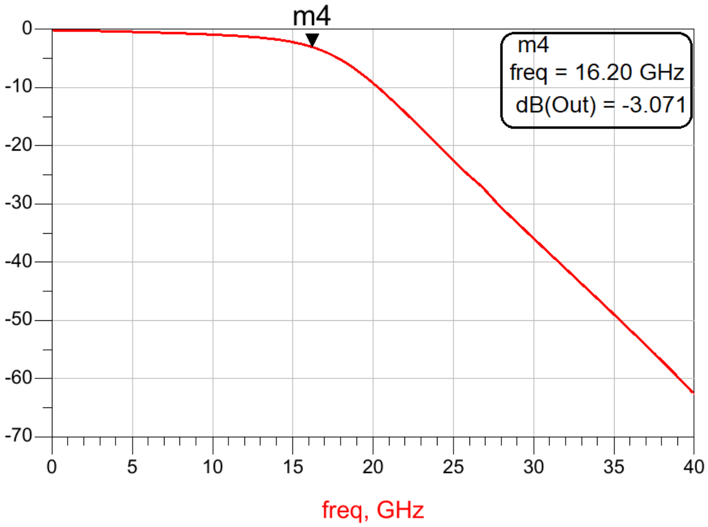
Σχήμα 6: Βαθυπερατό Φίλτρο

Στοιχείο	Τιμή
C_1	70 fF
C_2	270 fF
C_3	270 fF
C_4	70 fF
L_1	600 pH
L_2	963 pH
L_3	600 pH

Πίνακας 1: Τιμές Στοιχείων

Στοιχείο	Τιμή
C_1	240 fF
C_2	385 fF
C_3	240 fF
L_1	215 pH
L_2	867 pH
L_3	867 pH
L_4	215 pH

Πίνακας 2: Τιμές Δυαδικών Στοιχείων



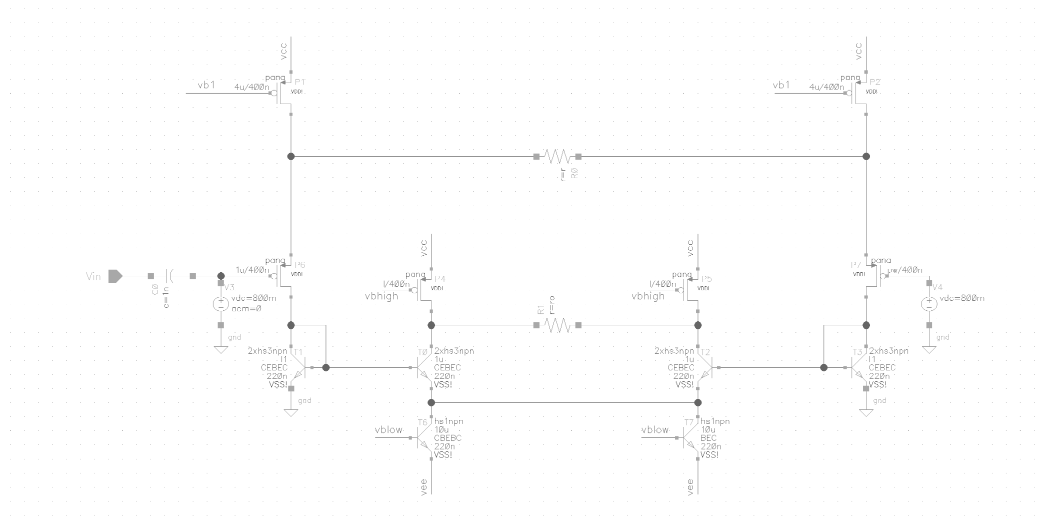
Σχήμα 7: Απόκριση Συχνότητας Φίλτρου

Το σχηματικό του διαγωγού ρυθμιζόμενου κέρδους φαίνεται στο Σχήμα 8. Η εξίσωση που περιγράφει τη λειτουργία του είναι η εξής:

$$I_O = \frac{I_B}{I_A}\frac{V_{in}}{R_0} \tag{6}$$

Ο διαγωγός μπορεί επομένως να ειδωθεί ως ένας πολλαπλασιαστής, στην έξοδο του οποίου λαμβάνουμε ένα ρεύμα έντασης ανάλογης με το γινόμενο της τάσης εισόδου επί ένα ρεύμα πόλωσης. Αν το ρεύμα πόλωσης I_B είναι με τη σειρά του ανάλογο του βάρους, ο διαγωγός πολλαπλασιάζει στην ουσία το βάρος αυτό με την τάση εισόδου, η οποία είναι το λαμβανόμενο σήμα στο οποίο έχουμε εισάγει καθυστέρηση μέσω του φίλτρου που περιγράφηκε παραπάνω. Προκειμένου ο διαγωγός να βρίσκεται στη γραμμική (ή σχεδόν γραμμική) περιοχή λειτουργίας του, το ρεύμα πόλωσης I_B πρέπει να περιέχεται εντός του διαστήματος 2.7 - 5.3 mA, όπως φαίνεται και στο Σχήμα 9. Η περιοχή αυτή καθορίζει και το εύρος τιμών που μπορεί να παίρνει το ρεύμα εξόδου του μετατροπέα ψηφιακού σε αναλογικό σήμα.

Η αρχιτεκτονική που επιλέχθηκε για την υλοποίηση του μετατροπέα είναι αυτή ενός δυαδικού μετατροπέα 8-bit βασιζόμενου σε πηγές ρεύματος. Καθώς ο μετατροπέας λαμβάνει ψηφιακές λέξεις 8 bit ως είσοδο, αποτελείται από 8 κελιά, το καθένα από τα οποία δίνει στην έξοδό του ρεύμα διπλάσιο από το ρεύμα εξόδου του προηγούμενου κελιού. Το πρώτο κελί δίνει ρεύμα συμβολιζόμενο



Σχήμα 8: Αναλογικός Πολλαπλασιαστής

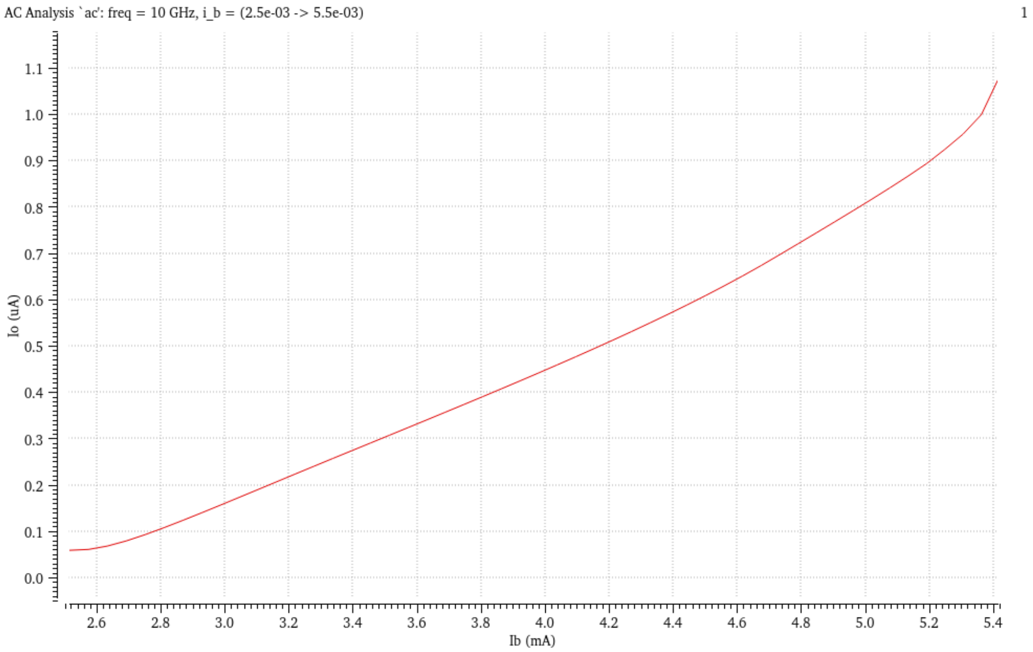
ως I_{LSB} το οποίο προκύπτει από τη σχέση

$$I_{LSB} = \frac{2.6\text{mA}}{2^{B-1}} = 20\mu\text{A} \tag{7}$$

Το σχηματικό του κελιού του μετατροπέα φαίνεται στο Σχήμα 10. Τα διπολικά τρανζίστορ N_{12} και N_{13} αποτελούν μια κασκοδική πηγή ρεύματος, η οποία παρέχει το κάθε κελί με το ρεύμα που πρέπει να εμφανίζεται στην έξοδό του. Τα τρανζίστορ N_{14} και N_{15} ελέγχονται από συμπληρωματικές τάσεις, έτσι ώστε πάντα ένα από τα δύο να άγει. Όταν το bit που αντιστοιχεί στο κελί έχει τιμή 1, άγει το τρανζίστορ N_{14}. Αλλιώς, αυτό που άγει είναι το N_{15}. Το γεγονός αυτό εξασφαλίζει ότι η πηγή ρεύματος θα άγει καθ' όλη τη λειτουργία του συστήματος, και επομένως οι παρασιτικές χωρητικότητες που εμφανίζονται στην είσοδό της θα είναι πάντα φορτισμένες. Τα τρανζίστορ N_{16} και N_{17} αυξάνουν την αντίσταση εξόδου του κελιού, ενώ τα τρανζίστορ N_{18} - N_{23} άγουν ένα μικρό ποσοστό του ρεύματος που παρέχεται από τα N_{12} και N_{13} έτσι ώστε τα τρανζίστορ N_{16} και N_{17} να μην βρίσκονται στην περιοχή αποκοπής.

Προσομοίωση Συστήματος

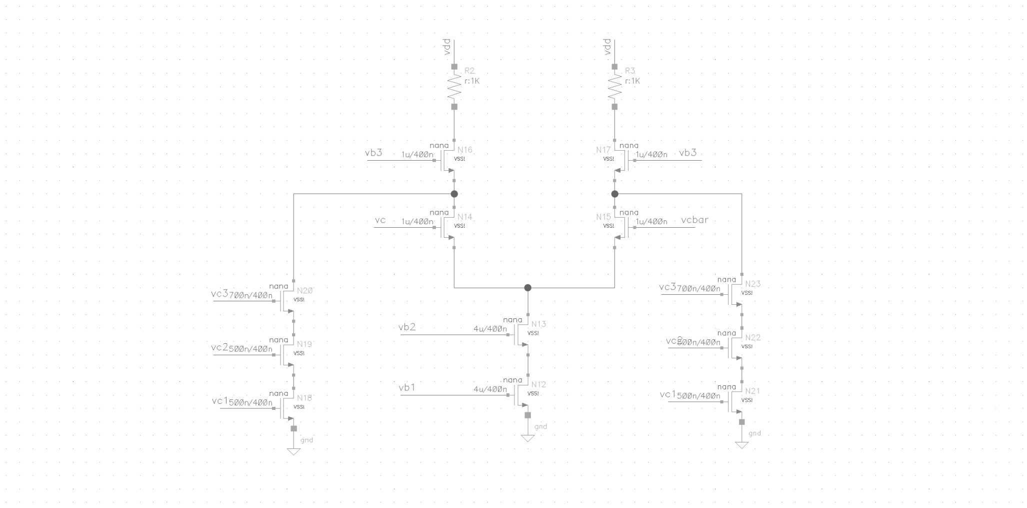
Για την τελική αξιολόγηση του συστήματος που παρουσιάστηκε μέχρι στιγμής πραγματοποιήσαμε την προσομοίωσή του. Για να προσομοιώσουμε το σύστη-



Σχήμα 9: Σχεδόν Γραμμική Περιοχή Λειτουργίας Διαγωγού

μα μεικτού σήματος, το διαχωρίσαμε αρχικά στο αναλογικό και ψηφιακό του μέρος. Το πρώτο, το οποίο συμπεριλαμβάνει την παραγωγή του σήματος, τη διαμόρφωσή του, τη μετάδοσή του μέσω ενός PMF, αλλά και το φίλτρο και τους αναλογικούς πολλαπλασιαστές, προσομοιώθηκε στο περιβάλλον SystemVue της Keysight. Τα σήματα που προέκυψαν από αυτή την προσομοίωση τροφοδοτήθηκαν σε μια υλοποίηση της ψηφιακής λογικής στο περιβάλλον της Matlab, όπου και πραγματοποιήθηκε μια "offline" εξισορρόπηση του σήματος. Τα διαγράμματα βαθμίδας που χρησιμοποιήθηκαν για την προσομοίωση του αναλογικού μέρους του συστήματος φαίνονται στα Σχήματα 11 και 12. Τα χαρακτηριστικά του PMF μέσω του οποίου μεταδόθηκε το σήμα προέκυψαν από την προσομοίωση ενός τέτοιου καλωδίου που παρείχε η εταιρεία Huber & Suhner και φαίνονται στο Σχήμα 13.

Το σήμα που μεταδόθηκε, καθώς και το σήμα που έλαβε ο δέκτης μετά την μετάδοση μέσω του PMF, φαίνονται στο Σχήμα 14. Τα σήματα που συλλέχθηκαν από τους καταχωρητές του Σχήματος 12 λειτουργούν με τη σειρά τους ως είσοδοι για την ψηφιακή λογική που υλοποιεί τον αλγόριθμο LMS. Το αποτέλεσμα της προσομοίωσης του αλγορίθμου φαίνεται στο Σχήμα 15. Για

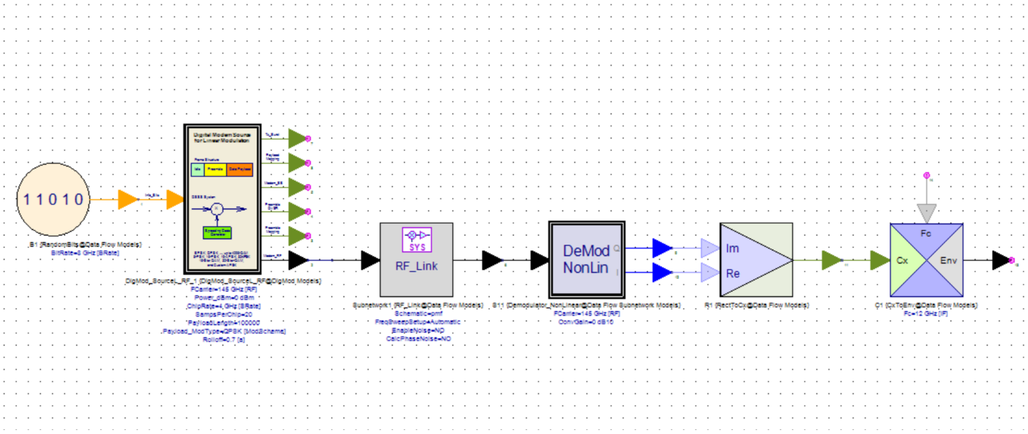


Σχήμα 10: Κελί Μετατροπέα Ψηφιακού σε Αναλογικό Σήμα

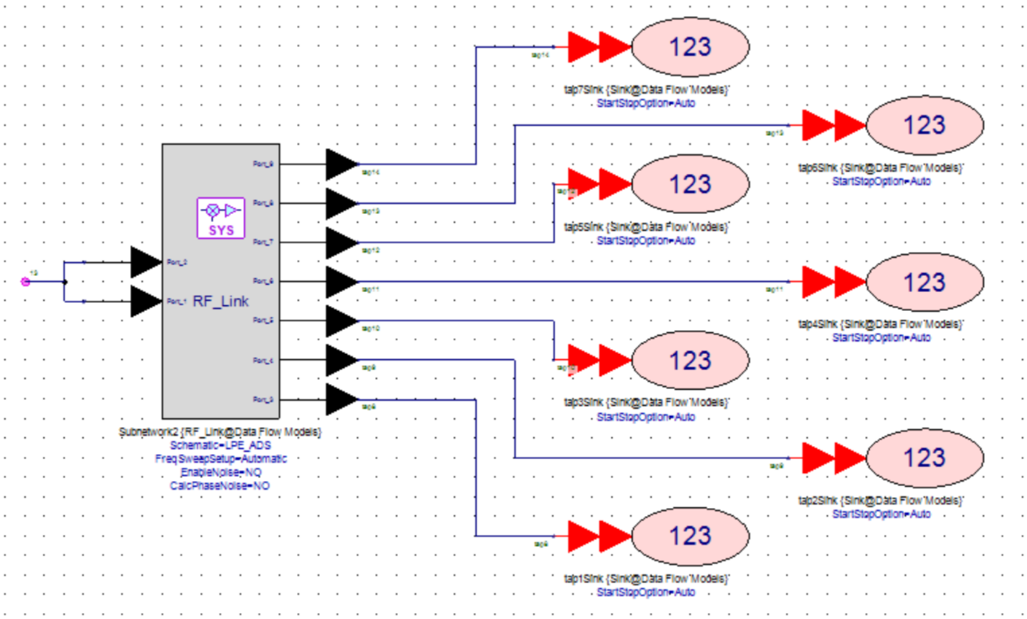
να πραγματοποιηθεί σύγκριση του ληφθέντος με το εξισορροπημένο σήμα, τα παραθέτουμε σε ένα καινούργιο διάγραμμα στο Σχήμα 16. Για να συγκριθεί και ποσοτικά η ποιότητα των δύο σημάτων, χρησιμοποιήθηκε η μετρική του Πλάτους Διανύσματος Σφάλματος (EVM), η οποία ορίζεται βάσει της εξίσωσης

$$\mathrm{EVM} = \frac{|e|}{|\mathrm{ref}|} * 100 \tag{8}$$

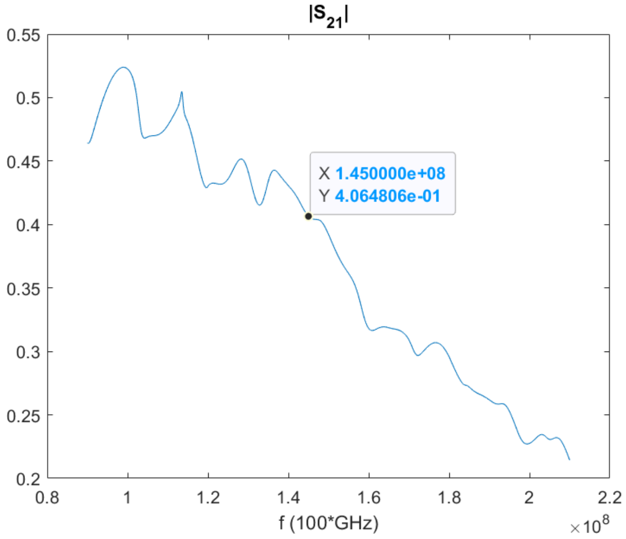
Χρησιμοποιώντας την εξίσωση αυτή προέκυψε ότι το ληφθέν σήμα χαρακτηρίζεται από EVM της τάξης του 64.62%, ενώ το εξισορροπημένο σήμα έχει EVM 16.99%. Επιτεύχθηκε, επομένως, βελτίωση του EVM του σήματος κατά σχεδόν 50%.


1 1 0 1 0
B1 {RandomBits@Data Flow Models}
BitRate=8 GHz [BRate]
Digital Modem Source for Linear Modulation
DigMod_SourceL_RF_1 {DigMod_SourceL_RF@DigMod Models}
FCarrier=145 GHz [RF]
Power_dBm=0 dBm
ChipRate=4 GHz [BRate]
SampsPerChip=20
PayloadLength=100000
Payload_ModType=QPSK [ModScheme]
Rolloff=0.7 [a]
SYS
RF_Link
Subnetwork1 {RF_Link@Data Flow Models}
Schematic=pmf
FreqSweepSetup=Automatic
EnableNoise=NO
CalcPhaseNoise=NO
DeMod
NonLin
Q
I
S11 {Demodulator_NonLinear@Data Flow Subnetwork Models}
FCarrier=145 GHz [RF]
ConvGain=0 dB10
Im
Re
R1 {RectToCx@Data Flow Models}
Fc
Cx
Env
C1 {CxToEnv@Data Flow Models}
Fc=12 GHz [IF]

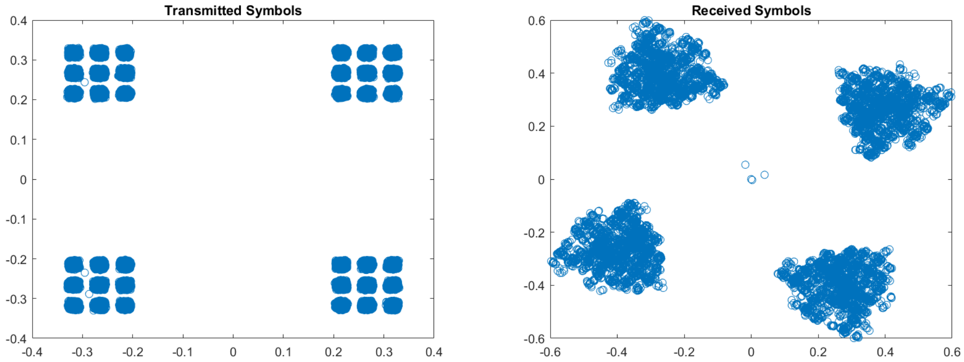
Σχήμα 11: Διάγραμμα Βαθμίδας Συστήματος Μετάδοσης



Σχήμα 12: Βαθμίδες Φίλτρου και Συλλογής Σήματος



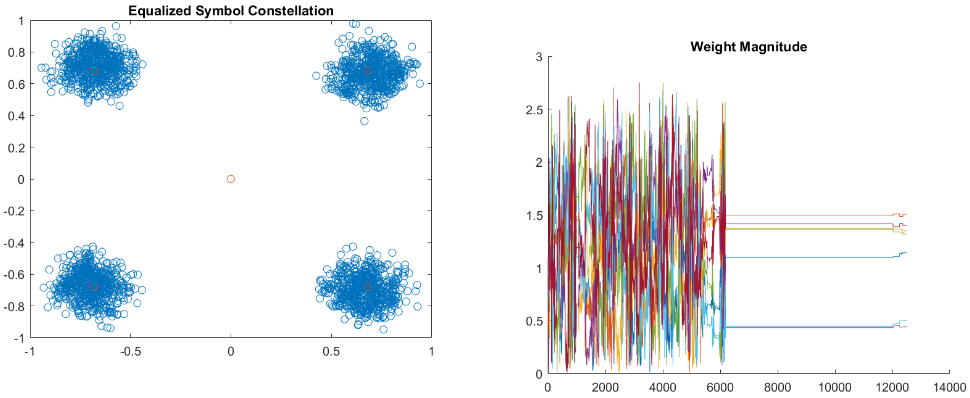
Σχήμα 13: S_{21} καλωδίου PMF μήκους 1m



(α΄) Μεταδιδόμενο Σήμα

(β΄) Ληφθέν Σήμα

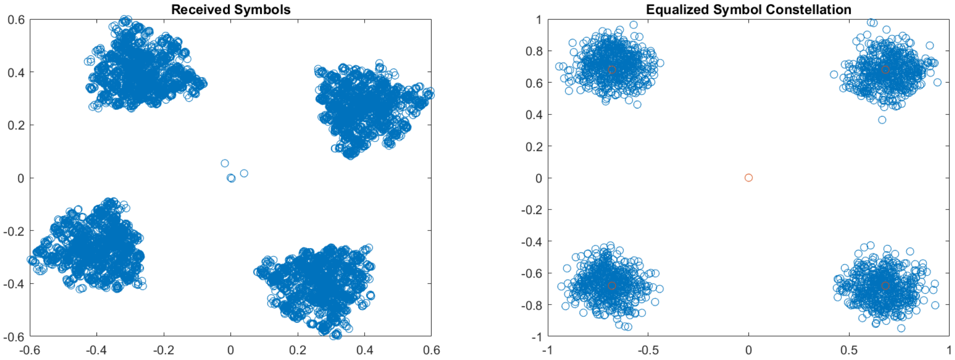
Σχήμα 14: Σήματα Πομπού και Δέκτη



(α′) Αστερισμός Εξισορροπημένου Σήματος

(β′) Διάνυσμα Βαρών

Σχήμα 15: Αποτελέσματα Εξισορρόπισης



(α′) Ληφθέν Σήμα

(β′) Εξισορροπημένο Σήμα

Σχήμα 16: Σύγκριση Αστερισμών

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

With the prevalence of telecommunication systems in our everyday life, both the technologies used in them and the needs of their users have developed accordingly. With the dramatic increase in data rates, the systems tasked with providing information are pushed to their limits, and their performance struggles to meet expectations. This has made signal correction an even bigger necessity than in previous generations of systems, and it is to this necessity that this work aims to respond.

In the context of this work, we have developed an equalizer solution for a high rate wireline data link operating in the sub-THz band of frequencies, that attempts to correct the transmitted data in an adaptive fashion, with the ability to properly compensate for unexpected channel behaviour while transmission is in process, increasing the performance of the whole system. Through the analysis and evaluation of the already available methods and the development of a new solution, this work attempts to provide a comprehensive answer to the issue of equalization in modern, high data rate systems.

1.2 Introduction to Equalization

The term equalization refers to a control process that attempts to reconstruct a distorted, transmitted signal to its original form, compensating for the effect the channel has had to the signal. It can be seen as a classic deconvolution problem, where the equalizer attempts to extract the signal $s(t)$ from the received signal

$$r(t) = \int_{\tau=-\infty}^{\infty} s(t)h(\tau - t)d\tau \tag{1.1}$$

where $h(t)$ is the channel impulse response. If $y(t)$ is the equalizer output, and $H_{\text{eq}}(\omega)$ is the equalizer frequency response, then we can arrive at the expression

$$Y(\omega) = H_{\text{eq}}(\omega)R(\omega) \tag{1.2}$$

where $Y(\omega)$ and $R(\omega)$ are the frequency domain expressions of the equalized and received signals respectively. If we denote the channel frequency response as $H(\omega)$, we arrive at the expression

$$Y(\omega) = H_{\text{eq}}(\omega)H(\omega)S(\omega) \tag{1.3}$$

In the ideal case, then, where $Y(\omega) = S(\omega)$, it would appear that

$$H^*_{\text{eq}}(\omega) = \frac{1}{H(\omega)} \tag{1.4}$$

and the equalizer is the inverse model of the channel. Of course, in most cases, the equalizer only achieves an estimate of this inverse function. The many methods employed to realize this inverse behaviour will be presented in more detail later on in this work.

1.3 The System to be Equalized

As mentioned above, while this work aims to provide a comprehensive overview of the subject of equalization in general, its starting point is the equalization of a real system, which we will be presenting in this Section. This system is based on the use of a Plastic Microwave Fiber (PMF) as a wireline channel for the achievement of high data rate transmission with efficient cost. The choice of PMF over the more conventional copper or optical fiber solutions lies in the fact that the PMF is an "in between" material that combines the high data rates achievable by optical fibers with the cost efficiency of copper wires. [6]

In the case of copper wires, we are mainly faced with issues regarding performance. These solutions are often plagued by issues like electromagnetic influence (EMI) and crosstalk susceptibility, dispersion in the transmitted signal, and losses highly dependent on carrier frequency, mostly due to the skin effect. However they present us with the significant advantage of cost efficient transmission.

With optical fibers, on the other hand, we are provided with a much less lossy channel, but are faced with many issues regarding efficiency and overall system costs. Optical systems require one or even multiple integrated light sources, and expensive interconnects for the silicon to optical interface. They are also generally more power hungry than their copper counterparts, and much more sensitive to misalignment and bending, with optical components having to be aligned with μm-range accuracy.

These aforementioned characteristics have left a gap in the 1-25 meter range that can be covered by PMF systems. This is illustrated in Figure 1.1, where the results of recent publications in the field of PMF systems are compared to those reported for copper and optical fiber systems. Besides the increased link efficiency in that distance range, the PMF also offers benefits such as higher EMI tolerance, due to its dielectric nature. It also has the significant advantages of more robust interconnections than its optical counterparts, as well as large available bandwidths due to the fact that it allows for mm-wave frequency range operations[6].

The proposed PMF system can be seen in Figure 1.2. It is comprised of a transmitter and receiver featuring Vivaldi antennas, and receiving Local Oscillator (LO) and In-Phase and Quadrature (I/Q) signals from external waveform generators. The PMF cable is used to connect the two IC's and the received signal is adepicted on an oscilloscope. This setup was used to perform some measurements to evaluate the performance of the system in a previous work (Papananos, Liakonis, Manouras, 2023). The transceiver architecture does not lie within the objects of this work, and will not be extensively discussed, since it has already been presented in previous works [1, 2, 3].

1.4 Structure of this Work

Chapter 1 functions as an introduction to the subject of this thesis. In Chapter 2, the theory behind the transmission of electromagnetic signals, the distortion it causes and the process of equalization will be presented in short, followed by an overview of the main approaches to equalization. Chapter 3 contains an introduction to adaptive control, as well as an evaluation

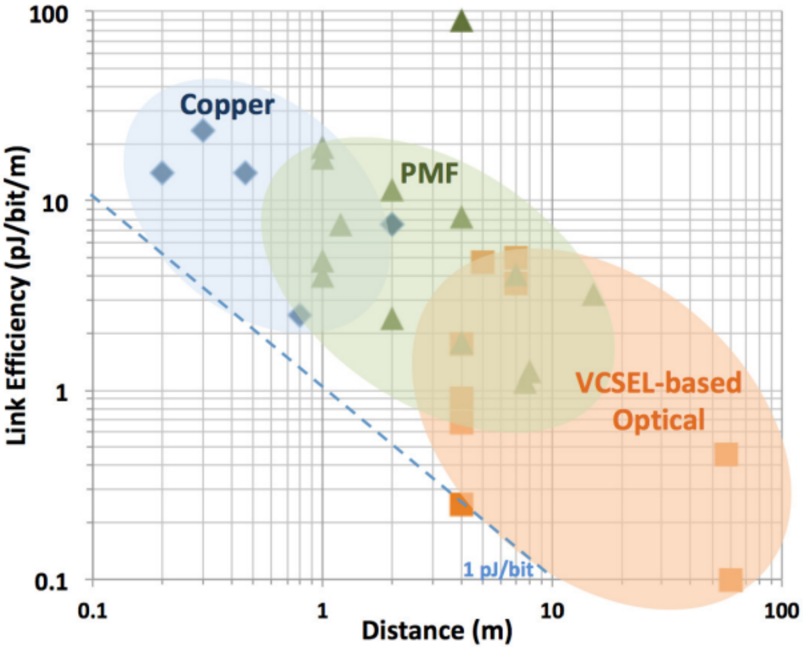


Figure 1.1: Copper, Optical Fiber and PMF Efficiency Comparison

and comparison of two commonly used adaptive algorithms, the Least Mean Squares (LMS) algorithm and the Constant Modulus Algorithm (CMA). In Chapter 4 we discuss the implementation of the integrated circuits (ICs) that comprise the system, and present their behaviour and the process of their design. Finally, the results of the simulations of the complete equalizer system are presented in Chapter 5.

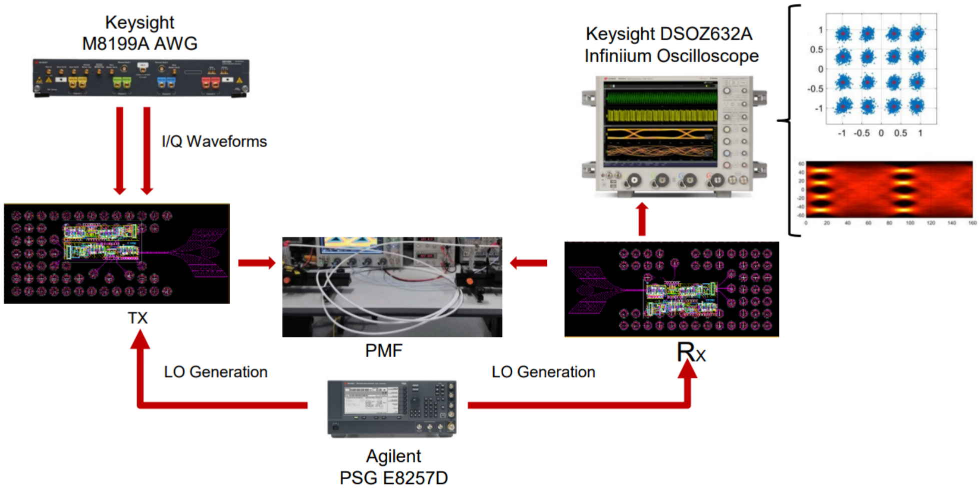


Figure 1.2: System Setup

Chapter 2

Equalization Theory and Available Solutions

In this chapter we present the basics of equalization theory, as well as an overview of the solutions so far available in the literature. We will begin by presenting the effects of transmission on the transmitted signal and its quality, and then present the main categories of equalizer systems.

2.1 Transmission Effects

Any signal transmitted through a non-ideal (lossy) channel will deteriorate. Depending on the transmission mode employed in the telecommunications scheme, as well as the physical properties of the channel itself, the way the signal deteriorates will vary. Effects that may be present in a transmission system include but are not limited to metallic channel losses, inter-symbol interference and crosstalk, the effects of which will be discussed shortly.

2.1.1 Metallic Losses

In most metallic channels, like PCB channels in chip-to-chip communication, or metallic cables (coaxial, shielded twisted pair cables and so on) a frequency dependent loss characteristic can be observed. These high-frequency losses are mostly due to the skin effect and dielectric losses. The term "skin effect" refers to the phenomenon of current flowing only on the surface of the conductor in the case of high frequency signals due to the induced magnetic field. The dielectric losses are caused by the non-ideal characteristics of the isolating material that is used as cladding of the conducting medium. When both of these factors are taken into consideration, the channel loss can be described by Equation 2.1:

$$C(f) = e^{-[h_s(1+j)\sqrt{f}+h_d f]l} \tag{2.1}$$

where h_s is the skin-effect loss coefficient, h_d is the dielectric loss coefficient, l refers to channel length and f to frequency [4]. This exponential dependence of loss on frequency that is evident in Equation 2.1 explains the limited bandwidth of metallic transmission media. The dependency on channel length also shown in Equation 2.1 explains the limits of distance in wireline transmissions.

2.1.2 Inter-Symbol Interference

In the case of a channel with frequency dependent loss characteristics, it is obvious that the different frequency components of the signal will be differently attenuated and delayed. This variation in magnitude attenuation and phase delay causes dispersion to the different frequency components contained within the transmitted signal, which in turn causes neighboring symbols to overlap. This phenomenon is referred to as inter-symbol interference (ISI).

In the case of ISI being present in a transmission scheme, the received signal $y(t)$ can be described by the following equation:

$$y(t) = \sum_{k=-\infty}^{\infty} p(t) * a(t - kT_b) \tag{2.2}$$

where $p(t)$ is the channel response and $a(t)$ is the transmitted signal. T_b refers to the bit period, and the symbol $*$ denotes convolution.

Assuming a causal channel response, that is, a transmission channel that allows interference from preceding symbols, but not succeeding symbols, the range of k can be limited to positive integers only. If we make the additional assumption that the effects of preceding symbols on following symbols die down after a certain number of symbols, L, this range can be further limited until Equation 2.2 takes the form

$$y(t) = \sum_{k=0}^{L} p(t) * a(t - kT_b) \tag{2.3}$$

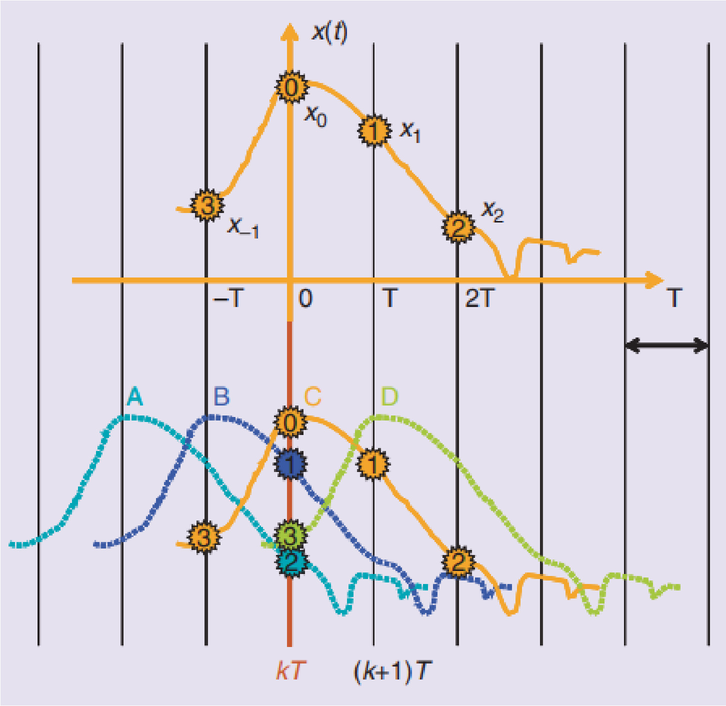


Figure 2.1: ISI Illustration

Inter Symbol Interference is mostly due to channels not having constant group delay. Group delay refers to the time delay faced by signals of different frequencies when transmitted through the same medium. In the case of a QPSK signal, for example, whose frequency information can be approximated by a *sinc* function, there might be a significant difference between the delay faced by different parts of its spectrum, and it is this variable delay that causes the overlap between succeeding symbols.

This effect is illustrated in Figure 2.1, which shows how overlapping might happen in the case of a symbol that has been spread out to longer than a symbol period. [4]

2.1.3 Crosstalk

Crosstalk is the electromagnetic interferenece caused by coupling between signal lines due to mutual capacitance and inductance. This phenomenon appears in two different forms, near end crosstalk (NEXT) and far end crosstalk

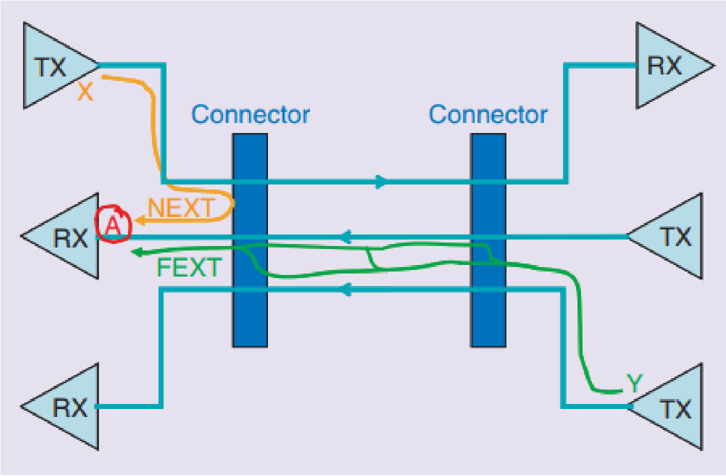


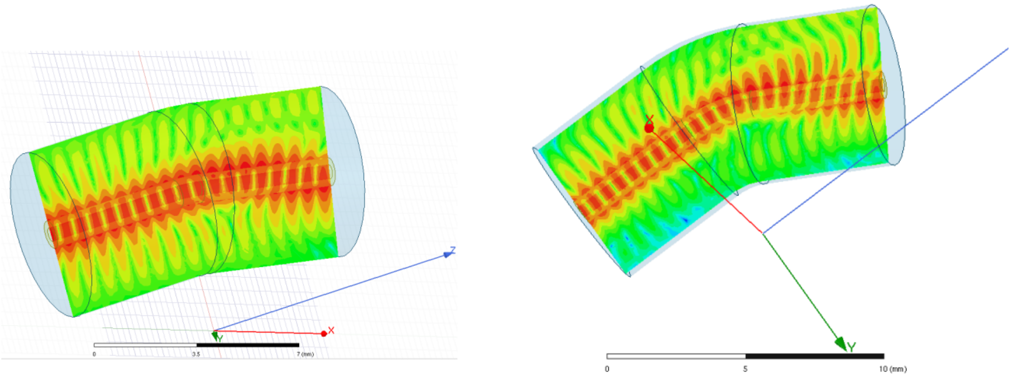
Figure 2.2: Crosstalk Paths

(FEXT). NEXT refers to interference caused by transmitters close to the receiver, while FEXT refers to that caused by far away signal sources. Obviously, since the signal has to travel further in the case of FEXT, it is more severely attenuated than in the case of NEXT, which renders NEXT the source of more significant problems. It has been shown that NEXT has a more severe effect the higher the frequency. This has made it so that NEXT can be a critical issue in a system. The two different crosstalk phenomena are illustrated graphically in Figure 2.2 [4].

2.1.4 Other Effects

Other effects that might cause signal degradation include power supply noise and reflections. Power supply noise is caused by switching large currents in short durations over the parasitic inductance of the power distribution network, exacerbated by higher switching frequencies, larger output swing and bigger number of switching drivers operating at the same time. Reflections are caused by discontinuities in impedance, which often appear in card-to-board connectors, long vias with their respective end pads, solder balls and so on.

In the case of wireline systems utilizing waveguides or optical fibers, another matter of concern is the effect of fiber bending on the transmission. It is generally understood that a large bend in the fiber introduces additional losses to the transmission path. The effect of different sized bends on the propagation of a wave through a PMF can be seen in Figure 2.3. It is evident that the field is disturbed after the bend, with this disturbance persisting due to the introduction of a z-dependency into the wave propagation.



(a) 10° Bend

(b) 30° Bend

Figure 2.3: Bending Effect on Propagation

2.2 Equalizer Types

The process of equalization can be performed following many different approaches. Distinctions between different systems can be made with regards to whether the equalization is performed on the transmitter or the receiver side, with the first case sometimes referred to as transmitter pre-emphasis, whether the equalizer is an active or passive, linear or nonlinear, adaptive or nonadaptive system. Each approach provides its own strengths, and can be more or less suited for a specific application. In the following paragraphs these different approaches will be presented in some more detail.

2.2.1 Transmitter and Receiver Side Equalizers

Since in most cases, it is the high frequency part of the spectrum that is more severely attenuated through the channel, a strategy that can be employed to counteract this effect is to pre-emphasize the part of the signal spectrum that is expected to face this more extensive attenuation, by amplifying it before transmission. While this process can compensate for a priori known bandwidth limitations, it may cause EMI and more severe crosstalk problems. Another approach to transmitter-side equalization is to instead de-emphasize the parts of the signal that will be less affected by the transmission through the channel, and so receive a signal that has been more or less equally attenuated throughout its spectrum.

Transmitter pre-emphasis is most usually realized through the use of FIR filters. A simplified version of this approach is to use two differential amplifiers, the first one controlled by the original code to be transmitted, and the second one by the original code after it has been inverted and delayed by one symbol period. The second code is often referred to as "emphasis code". The general, non-simplified approach is to implement the equalizer as a group of differently weighted digital delay units, thus forming a digital FIR filter. The delayed symbols compensate for post-cursor ISI.

Transmitter de-emphasis, that is reducing the power contained within the low frequency components of the signal before transmission, can be most simply realized by increasing signal amplitude at the transition edges and decreasing it when there are no transitions. By inverting the previously transmitted bit and using the generated sequence as an emphasis code, we can amplify the signal whenever the previous bit is opposite to the current bit, and attenuate it when they are the same. This is achieved by the equalizer shown in Figure 2.4 [7, 8].

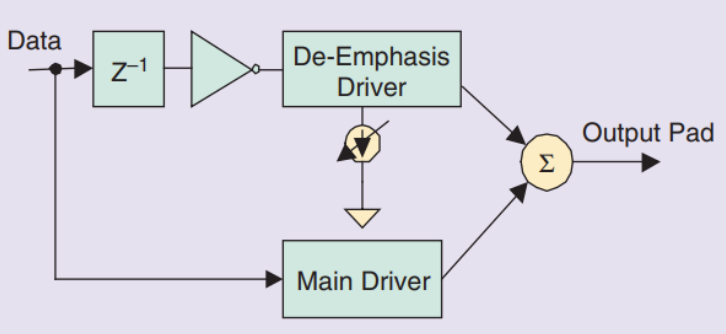


Figure 2.4: Block Diagram of a De-emphasis Equalizer

In both cases (pre-emphasis and de-emphasis) the equalizer has the transfer function of a high-pass FIR filter. The implementation of transmitter-side equalizers is generally easier than that of receiver-side equalizers, since the bit sequence to be transmitted is readily available in a parallel data bus, and can be naturally fed into the FIR filter as an input. However, de-emphasizing the signal obviously reduces the signal power, and since no channel information is available at the transmitter side, this information must be sent back to it from the receiver if we want dynamic equalization. Considering that the task of adapting to varying channel conditions can usually be left to the receiver, the second issue is not as significant.

2.2.2 Active and Passive Equalizers

Depending on the devices used to implement the equalizer, it can be deemed as either passive or active. In the case of passive receiver side equalizers, the designer has the significant advantage of being able to determine the frequency response of the equalizer at different bands independently, since each frequency band's behavior is defined by different elements. This can be seen in the example implementation shown in Figure 2.5, where the low frequency response is determined by R_2 and C_2, the mid-band frequency response by L_2 and the high frequency response by L_1 and C_1. The advantages of easy implementation and low power consumption should also be taken into consideration when deciding on which approach to equalization should be taken. However, this approach comes with the equally significant disadvantage of being incapable of realizing adaptive filters, and is therefore unsuitable for many applications [4].

2.2.3 Linear and Nonlinear Equalizers

Whether digital or analog, discrete or continuous time, most receiver side equalizer implementations comprise a linear FIR filter. In most if not all of these cases, the FIR filter amplifies the high-frequency components of the signal, thus also amplifying high-frequency noise, which can potentially have an adverse effect on the system's noise margin. This difficulty can be overcome by following a nonlinear approach, such as using a Decision Feedback Equalizer (DFE). The block diagram of a DFE can be seen in Figure 2.6, where the filter non-linearity is specifically designed to avoid noise amplification. This is made possible by using the decisions made previously to estimate and compensate for pre-cursor ISI. This approach is obviously unable to compensate for post-cursor ISI, and its effectiveness depends on the assumption that previous decisions are correct. This assumption can lead to error propagation in the case that it is not actually correct [5]. These

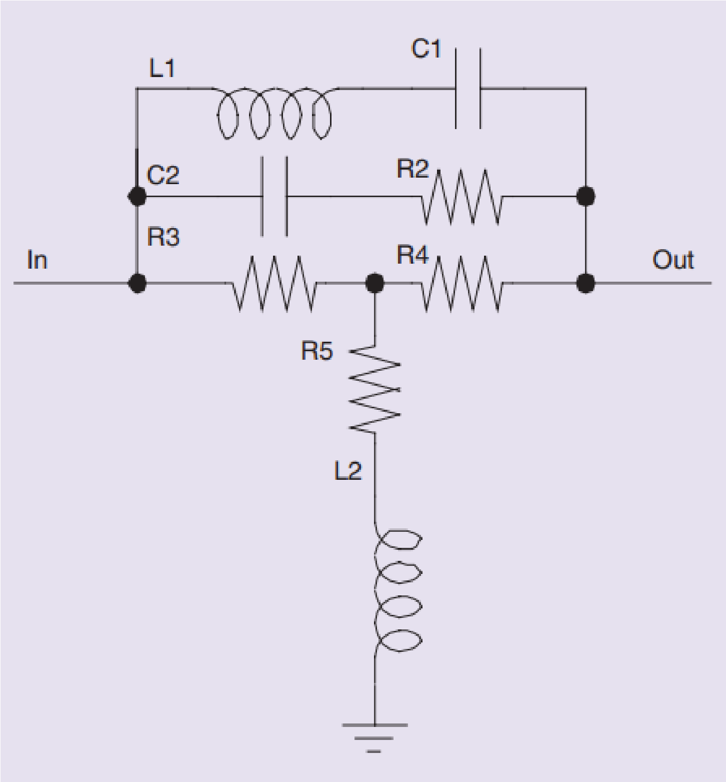


Figure 2.5: Passive T-bridge Equalizer

difficulties make it so that the use of DFEs is only worthwhile in cases where the main limit of the equalizer's effectiveness is noise. In the case that a DFE is necessary, its behavior can be improved to allow for post-cursor ISI compensation by using a feedforward (linear) equalizer in conjunction.

2.2.4 Adaptive and Non-adaptive Equalizers

In a practical transmission system, the effects the channel will have on the transmitted can not be known a priori. Even in the case that there is some a priori knowledge of these effects, variations are to be expected, since they often depend on uncontrollable external conditions, such as weather conditions, channel length or additional connectors in wireless transmission, or temperature, bends or channel material properties in wireline transmission. All of these factors can significantly alter the channel's attenuation and band-

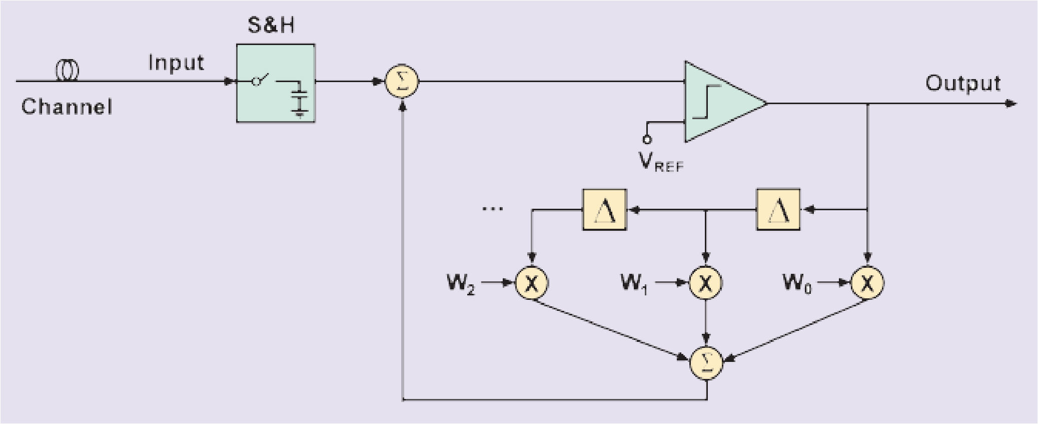


Figure 2.6: Block Diagram of a Decision Feedback Equalizer (DFE)

width. It is therefore very often necessary to adopt an adaptive solution that can follow changes in the transmission conditions and avoid under- or overcompensation. There are many different approaches to adaptive equalization that a designer can take, mostly depending on whether a training sequence is available or not at the receiver side (if not, the equalizer may be referred to as a blind one), and also on which algorithm is chosen to implement the adaptation process. These will be discussed in detail in the following chapter.

A generic block diagram presenting the operation of an adaptive equalizer can be seen in Figures 2.7 and 2.8. In the case of blind equalizers, the reference signal is usually generated by a decision system. The parameter of interest that determines the control signal and therefore the next state of the equalizer is usually an error signal. In the case that a training sequence is available, the equalizer switches to a reference signal generated by a decision system after the training sequence is over, and from that point on operates as a blind equalizer would [5].

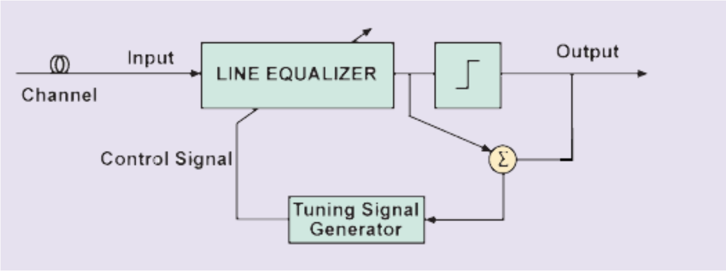


Figure 2.7: Block Diagram of a Blind Adaptive Equalizer

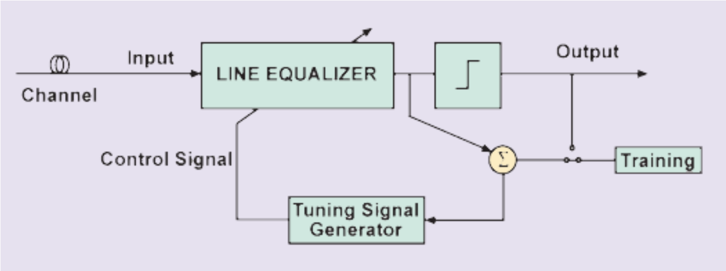


Figure 2.8: Block Diagram of an Adaptive Equalizer with Training Sequence

Chapter 3

Adaptive Algorithm Analysis

In this chapter we will give an overview of adaptive control schemes, and the algorithms most commonly employed to implement this control method. We will focus on two specific algorithms, the Least Mean Squares and the Common Modulus Algorithm, and compare them in terms of their effectiveness, ease of implementation and cost, as well as their suitability to the problem at hand itself.

3.1 Introduction to Adaptive Algorithms

In the context of control engineering, an adaptive filter is a control mechanism that updates its coefficients online, in order to satisfy a specified performance criterion. This performance criterion is expressed through an objective function, F or ξ, which is usually a function of the filter's input, a reference signal and the output signals of the adaptive filter, or, as a mathematical expression, $F = F[x(k), d(k), y(k)]$ [9]. For the objective function to be meaningful, two conditions must be satisfied:

- Non-negativity: $F[x(k), d(k), y(k)] \geq 0, \ \forall \ x(k), \ y(k)$ and $d(k)$
- Optimality: $F[x(k), d(k), d(k)] = 0$.

These two conditions show what exactly it is that an adaptive algorithm attempts to do: minimize an objective function in such a way that $y(k)$ approximates $d(k)$, and the filter coefficients consequently converge to the optimal coefficient vector that minimizes F. Another approach to understanding this objective function would be to treat it as a function of an error signal, $e(k)$, which in turn is a function of $x(k)$, $y(k)$ and $d(k)$, $e(k) = e(x(k), d(k), y(k))$. It can be therefore seen that the definition of the adaptive algorithm consists of three separate definitions: a definition of the minimization algorithm, a definition of the form of the objective function, and a definition of the error signal.

By approximating the objective function F around a point $\boldsymbol{\theta}(k)$ through a Taylor series we get the following expression:

$$\begin{aligned} F[\boldsymbol{\theta}(k) + \Delta\boldsymbol{\theta}(k)] \approx{} & F[\boldsymbol{\theta}(k)] + \nabla_{\boldsymbol{\theta}}\{F[\boldsymbol{\theta}(k)]\}\Delta\boldsymbol{\theta}(k) \\ & + \frac{1}{2}\Delta\boldsymbol{\theta}^T(k)\boldsymbol{H}_{\boldsymbol{\theta}}\{F[\boldsymbol{\theta}(k)]\}\Delta\boldsymbol{\theta}(k) \end{aligned} \tag{3.1}$$

where $\boldsymbol{\theta}(k)$ is the value of the coefficient vector at moment k, $\nabla_{\boldsymbol{\theta}}\{F[\boldsymbol{\theta}(k)]\}$ is the gradient vector and $\boldsymbol{H}_{\boldsymbol{\theta}}\{F[\boldsymbol{\theta}(k)]\}$ is the Hessian matrix of the objective function [9]. The aim of the minimization algorithm is to minimize the objective function with respect to the set of parameters by iterating

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) + \Delta\boldsymbol{\theta}(k) \tag{3.2}$$

where the correction term $\Delta\boldsymbol{\theta}(k)$ is supposed to minimize the Taylor series approximation of the objective function in Equation 3.1. The definition of the minimization algorithm consists of choosing a method to calculate this correction term. The most often employed one is the steepest descent, or gradient method, which searches for the minimum of the objective function by following the opposite direction of its gradient vector. In this case, the updating equation becomes

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) - \mu\nabla_{\boldsymbol{\theta}}\{F[e(k)]\} \tag{3.3}$$

where μ is the algorithm's step size. The Least Mean Squares algorithm, which will be explained in more detail later on, belongs to this group of algorithms.

The objective function can be defined in any way that satisfies the two conditions of non-negativity and optimality mentioned above. This choice has a great effect on the complexity of the calculation of both the gradient vector and the Hessian matrix. A definition that is very commonly used to derive an adaptive algorithm is the Mean-Square Error (MSE) one:

$$F[e(k)] = E[|e(k)|^2] \tag{3.4}$$

This definition is used by both the LMS and CMA algorithms.

Finally, the third definition, that of the error signal, varies considerably from algorithm to algorithm, and is crucial in determining many of the algorithm's characteristics, including computational complexity, convergence speed, robustness and so on.

In the following sections we will apply this threefold definition to the two algorithms considered in this work, the LMS and the CMA algorithm, to obtain a thorough theoretical understanding of their operation. We will then simulate their operation in a Matlab environment to judge their effectiveness and overall behavior.

3.2 Least Mean Squares Algorithm

The Mean-Square Error objective function can be written as

$$F[e(k)] = \xi(k) = E[e^2(k)] = E[d^2(k) - 2d(k)y(k) + y^2(k)] \tag{3.5}$$

Supposing that the adaptive filter consists of a linear combiner, so that the filter output is a linear combination of the input signals, its behavior can be mathematically expressed as:

$$y(k) = \sum_{i=0}^{N} w_i(k)x_i(k) = \mathbf{w}^T(k)\mathbf{x}(k) \tag{3.6}$$

where $\mathbf{x}(k) = [x_0(k)\ x_1(k)\ ...\ x_N(k)]^T$ and $\mathbf{w}(k) = [w_0(k)\ w_1(k)\ ...\ w_N(k)]^T$ are the input signals and the vector of the coefficients to be adapted, respectively. In many cases, the elements of the input signal vector are a delayed version of the same signal, so that $x_0(k) = x(k)$, $x_1(k) = x(k-1)$ and so on. In this case, $y(k)$ is the result of applying a FIR filter to the input signal $x(k)$. Equation 3.6 then becomes

$$y(k) = \sum_{i=0}^{N} w_i(k)x(k-i) = \mathbf{w}^T(k)\mathbf{x}(k) \tag{3.7}$$

Consequently, the expression for the MSE in Equation 3.5 becomes

$$\begin{aligned} E[e^2(k)] &= \xi(k) \\ &= E[d^2(k) - 2d(k)\mathbf{w}^T(k)\mathbf{x}(k) + \mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}(k)] \\ &= E[d^2(k)] - 2E[d(k)\mathbf{w}^T(k)\mathbf{x}(k)] + E[\mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}(k)] \end{aligned} \tag{3.8}$$

Assuming fixed filter coefficients, Equation 3.8 in turn becomes

$$\begin{aligned}\xi &= E[d^2(k)] - 2\mathbf{w}^T E[d(k)\mathbf{x}(k)] + \mathbf{w}^T E[\mathbf{x}(k)\mathbf{x}^T(k)]\mathbf{w} \\ &= E[d^2(k)] - 2\mathbf{w}^T\mathbf{p} + \mathbf{w}^T\mathbf{R}\mathbf{w} \qquad (3.9)\end{aligned}$$

where $\mathbf{p} = E[d(k)\mathbf{x}(k)]$ is the cross-correlation vector of the desired and input signals, and $\mathbf{R} = E[\mathbf{x}(k)\mathbf{x}^T(k)]$ is the input signal autocorrelation matrix. It can be seen that Equation 3.9 describes ξ as a quadratic function of the coefficient vector that could be easily solved for a $\mathbf{w}$ that minimizes ξ, as long as $\mathbf{p}$ and $\mathbf{R}$ are known. If this condition is satisfied, the gradient vector of the MSE becomes

$$\nabla_{\mathbf{w}}\xi = \frac{\partial \xi}{\partial \mathbf{w}} = -2\mathbf{p} + 2\mathbf{R}\mathbf{w} \qquad (3.10)$$

which is the Wiener solution. However, accurate estimations of $\mathbf{p}$ and $\mathbf{R}$ are not available in most practical situations. The adaptive algorithm must therefore be able to provide these estimates, which is often performed by using time averages of the input and desired signals. If these estimates, $\hat{\mathbf{p}}(k)$ and $\hat{\mathbf{R}}(k)$, are good enough, the Wiener solution in the form of Equation 3.3 becomes

$$\begin{aligned}\mathbf{w}(k+1) &= \mathbf{w}(k) - \mu\hat{\nabla}_{\mathbf{w}}\xi(k) \\ &= \mathbf{w}(k) + 2\mu(\hat{\mathbf{p}}(k) - \hat{\mathbf{R}}(k)\mathbf{w}(k)) \qquad (3.11)\end{aligned}$$

where $\hat{\nabla}_{\mathbf{w}}\xi(k)$ is an estimate of the gradient vector of the objective function with respect to the filter coefficients.

One possibility for the estimation of the gradient vector of the objective function is to provide instantaneous estimates of $\mathbf{p}$ and $\mathbf{R}$ as shown in the following:

$$\begin{aligned}\hat{\mathbf{p}}(k) &= d(k)\mathbf{x}(k) \\ \hat{\mathbf{R}}(k) &= \mathbf{x}(k)\mathbf{x}^T(k) \qquad (3.12)\end{aligned}$$

This would result in the following expression for the gradient vector estimate:

$$\begin{aligned}\hat{\nabla}_{\mathbf{w}}\xi(k) &= -2d(k)\mathbf{x}(k) + 2\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}(k) \\ &= 2\mathbf{x}(k)(-d(k) + \mathbf{x}^T(k)\mathbf{w}(k)) \\ &= -2e(k)\mathbf{x}(k) \qquad (3.13)\end{aligned}$$

This leads to the updating equation of the Least Mean Squares algorithm,

$$\mathbf{w}(k+1) = \mathbf{w}(k) + 2\mu e(k)\mathbf{x}(k) \tag{3.14}$$

where μ is chosen to ensure the convergence of the algorithm. Now that an expression for the LMS algorithm has been derived, it can be used to determine some of its properties.

3.2.1 Gradient Behavior

The instantaneous estimates of $\mathbf{p}$ and $\mathbf{R}$ used by the LMS algorithm as described in Equation 3.12, can be used in the expression of the MSE gradient provided by Equation 3.10 to give an expression for the search direction of the LMS algorithm,

$$\hat{\nabla}_{\mathbf{w}}\xi(k) = 2[\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}(k) - d(k)\mathbf{x}(k)] \tag{3.15}$$

It can be expected that the direction provided by Equation 3.15 might differ significantly from that provided by Equation 3.10. When using the LMS algorithm we accept the possibility that the gradient will behave differently from that of the steepest descent algorithm with regards to convergence, in exchange for improved computational ease and simplicity. For a fixed $\mathbf{w}$ vector we derive the expression

$$\begin{aligned} E[\hat{\nabla}_{\mathbf{w}}\xi(k)] &= 2\{E[\mathbf{x}(k)\mathbf{x}^T(k)]\mathbf{w}(k) - E[d(k)\mathbf{x}(k)]\} \\ &= \nabla_{\mathbf{w}}\xi(k) \end{aligned} \tag{3.16}$$

It can be seen that on average the LMS gradient direction tends to coincide with the steepest-descent gradient direction, and that $\hat{\nabla}_{\mathbf{w}}\xi(k)$ is an unbiased instantaneous estimate of $\nabla_{\mathbf{w}}\xi(k)$.

3.2.2 Coefficient Vector Convergence

Making the assumption that the ideal state of the adaptive FIR filter is described by the coefficient vector $\mathbf{w}_o$ (which would correspond to a non-adaptive FIR filter with the same coefficient vector), and that the adaptation is being performed via an LMS algorithm, and that measurement zero-mean white noise $n(k)$ with a variance σ_n^2 is added to the output of the unknown system, we can define the adaptive filter coefficient error with regards to the ideal coefficient vector $\mathbf{w}_o$ in each iteration as follows:

$$\Delta\mathbf{w}(k) = \mathbf{w}(k) - \mathbf{w}_o \tag{3.17}$$

which in turn allows us to derive the following alternative description of the LMS algorithm:

$$\begin{aligned}\Delta\mathbf{w}(k+1) &= \Delta\mathbf{w}(k) + 2\mu e(k)\mathbf{x}(k)\\ &= \Delta\mathbf{w}(k) + 2\mu\mathbf{x}(k)[\mathbf{x}^T(k)\mathbf{w}_o + n(k) - \mathbf{x}^T(k)\mathbf{w}(k)]\\ &= \Delta\mathbf{w}(k) + 2\mu\mathbf{x}(k)[e_o(k) - \mathbf{x}^T(k)\Delta\mathbf{w}(k)]\\ &= [\mathbf{I} - 2\mu\mathbf{x}(k)\mathbf{x}^T(k)]\Delta\mathbf{w}(k) + 2\mu e_o(k)\mathbf{x}(k)\end{aligned} \tag{3.18}$$

where $e_o(k)$ is the optimum output error defined as

$$\begin{aligned}e_o(k) &= d(k) - \mathbf{w}_o^T\mathbf{x}(k)\\ &= \mathbf{w}_o^T\mathbf{x}(k) + n(k) - \mathbf{w}_o^T\mathbf{x}(k)\\ &= n(k)\end{aligned} \tag{3.19}$$

The expected error of the coefficient vector is therefore

$$E[\Delta\mathbf{w}(k+1)] = E\{[\mathbf{I} - 2\mu\mathbf{x}(k)\mathbf{x}^T(k)]\Delta\mathbf{w}(k)\} + 2\mu E[e_o(k)\mathbf{x}(k)] \tag{3.20}$$

Assuming that $\mathbf{x}(k)$ is statistically independent of $\Delta\mathbf{w}(k)$ and $e_o(k)$, the above equation can be simplified to

$$\begin{aligned}E[\Delta\mathbf{w}(k+1)] &= \{\mathbf{I} - 2\mu E[\mathbf{x}(k)\mathbf{x}^T(k)]\}E[\Delta\mathbf{w}(k)]\\ &= (\mathbf{I} - 2\mu\mathbf{R})E[\Delta\mathbf{w}(k)]\end{aligned} \tag{3.21}$$

This in turn leads to

$$E[\Delta\mathbf{w}(k+1)] = (\mathbf{I} - 2\mu\mathbf{R})^{k+1}E[\Delta\mathbf{w}(0)] \tag{3.22}$$

If $\mathbf{Q}$ is the unitary matrix that, through a similarity transformation, diagonalises $\mathbf{R}$, we can use Equation 3.21 to derive the expression

$$\begin{aligned} E[\mathbf{Q}^T\Delta\mathbf{w}(k+1)] &= (\mathbf{I}-2\mu\mathbf{Q}^T\mathbf{R}\mathbf{Q})E[\mathbf{Q}^T\Delta\mathbf{w}(k)] \\ &= E[\Delta\mathbf{w}'(k+1)] \\ &= (\mathbf{I}-2\mu\mathbf{\Lambda})E[\Delta\mathbf{w}'(k)] \\ &= \begin{bmatrix} 1-2\mu\lambda_0 & 0 & \cdots & 0 \\ 0 & 1-2\mu\lambda_1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1-2\mu\lambda_N \end{bmatrix} E[\Delta\mathbf{w}'(k)] \end{aligned} \tag{3.23}$$

where $\Delta\mathbf{w}'(k+1) = \mathbf{Q}^T\Delta\mathbf{w}(k+1)$. Using Equations 3.22 and 3.23 we can arrive at the following expression:

$$\begin{aligned} E[\Delta\mathbf{w}'(k+1)] &= (\mathbf{I}-2\mu\mathbf{\Lambda})^{k+1}E[\Delta\mathbf{w}'(0)] \\ &= \begin{bmatrix} (1-2\mu\lambda_0)^{k+1} & 0 & \cdots & 0 \\ 0 & (1-2\mu\lambda_1)^{k+1} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & (1-2\mu\lambda_N)^{k+1} \end{bmatrix} E[\Delta\mathbf{w}'(0)] \end{aligned} \tag{3.24}$$

In order to guarantee that the coefficients converge, we must choose a μ within the range

$$0 < \mu < \frac{1}{\lambda_{\max}} \tag{3.25}$$

where $\lambda_{\max}$ is the largest eigenvalue of $\mathbf{R}$. This choice guarantees that all the elements of the diagonal matrix in the above equation will tend to zero as $k \to \infty$, and as a result so will $E[\Delta\mathbf{w}'(k+1)]$. This analysis highlights the importance of making a correct choice for the algorithm step size, as it can make the difference between divergence and convergence.

3.3 Constant Modulus Algorithm

The Constant Modulus Algorithm (CMA) refers to a family of blind (that is, without using a reference signal) algorithms, that attempt to minimize the distance between the modulus of an equalizer output and some predetermined constant value. This algorithm is a specific case of the general Godard algorithm. We will begin by describing the general case, and then proceed to derive expressions and behavior characteristics pertaining to the specialized case of the CMA.

3.3.1 The General Case: The Godard Algorithm

Let us suppose that a set of symbols, $s(k)$, belonging to a certain constellation $\mathcal{C}$, that is $s(k) \in \mathcal{C}$, is being transmitted through a channel with the impulse response $h(k)$. The received signal would then be

$$x(k+J) = s(k)h(J) + \left(\sum_{l=-\infty, l \neq k}^{k+J} s(l)h(k+J-l) \right) + n(k+J) \tag{3.26}$$

where J denotes channel delay, which will henceforth be considered to be zero without any loss of generality. We also assume a uniform distribution within the constellation space for symbol occurrence. The general objective of the Godard algorithm is to minimize the objective function

$$\begin{aligned} \xi_{Godard} &= E[(|\mathbf{w}^H(k)\mathbf{x}(k)|^q - r_q)^p] \\ &= E[(|y(k)|^q - r_q)^p] \\ &= E[e^p_{Godard}(k)] \end{aligned} \tag{3.27}$$

where

$$r_q = \frac{E[|s(k)|^{2q}]}{E[|s(k)|^q]} \tag{3.28}$$

where p and q are positive integers. The value of r_q represents the level $|y(k)|^q$ should reach, with the error being raised to the power of p.

To derive an analytic expression for the update of the weight vector, we simply differentiate Equation 3.27 with respect to $\mathbf{w}^*(k)$ and obtain the following expression [10]:

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) - \frac{1}{2}\,\mu\,p\,q\,(|y(k)|^q - r_q)^{p-1}|y(k)|^{q-2}y^*(k)\mathbf{x}(k) \\ &= \mathbf{w}(k) - \frac{1}{2}\,\mu\,p\,q\,e^{p-1}_{Godard}(k)|y(k)|^{q-2}y^*(k)\mathbf{x}(k) \end{aligned} \tag{3.29}$$

3.3.2 The Constant Modulus Algorithm Case

If we choose $p = q = 2$ for the Godard algorithm, the objective function of Equation 3.27 becomes

$$\begin{aligned} E[e^2_{CMA}(k)] &= E[(|\mathbf{w}^H(k)\mathbf{x}(k)|^2 - r_2)^2] \\ &= E[(|y(k)|^2 - r_2)^2] \end{aligned} \tag{3.30}$$

which satisfies the two conditions set for objectivity functions in Section 3.1 of this chapter [11, 12]. In this case,

$$r_2 = \frac{E[|s(k)|^4]}{E[|s(k)|^2]} \tag{3.31}$$

and

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) - 2\,\mu\,(|y(k)|^2 - r_2)\,y^*(k)\mathbf{x}(k) \\ &= \mathbf{w}(k) - 2\,\mu\,e_{CMA}(k)\,y^*(k)\mathbf{x}(k) \end{aligned} \tag{3.32}$$

3.3.3 CMA Error Surface

To better illustrate the behavior of the CMA we will attempt to derive an expression for its error surface in the case of a very simple transmission channel. We begin by assuming a channel described by the transfer function

$$H(z) = \frac{\kappa z}{z + a} \tag{3.33}$$

The ideal equalizer for this channel is given by the Wiener solution

$$\begin{aligned} W(z) &= \pm z^{-i}(w_0 + w_1 z^{-1}) \\ &= \pm \frac{z^{-1}}{\kappa}[1 + a z^{-1}] \end{aligned} \tag{3.34}$$

where i is a non-negative integer. If we substitute $i = 0$, we get an equalized signal with zero delay. The objective function can be written as

$$\begin{aligned} \xi_{CMA} &= E[|y(k)|^2 - r_2]^2 \\ &= E[|y(k)|^4] - 2E[|y(k)|^2]r_2 + r_2^2 \end{aligned} \tag{3.35}$$

The expected values in the above equation are given by

$$E[|y(k)|^2] =(w_0^2 + w_1^2)\frac{\kappa^2 E[|s(k)|^2]}{1-a^2} - 2w_0 w_1 \frac{a\kappa^2 E[|s(k)|^2]}{1-a^2} \tag{3.36}$$

$$\begin{aligned}
E[|y(k)|^4] =&(w_0^4 + w_1^4)\left[\frac{\kappa^4 E[|s(k)|^4]}{1-a^4} + \frac{6a^2\kappa^4\{E[|s(k)|^2]\}^2}{(1-a^4)(1-a^2)}\right] \\
&+ 6w_0^2 w_1^2 \left\{a^2\left[\frac{\kappa^4 E[|s(k)|^4]}{1-a^4} + \frac{6a^2\kappa^4\{E[|s(k)|^2]\}^2}{(1-a^4)(1-a^2)}\right]\right. \\
&\qquad\qquad \left. + \frac{\kappa^2\{E[|s(k)|^2]\}^2}{1-a^2}\right\} \\
&- 4w_0 w_1^3 a \left\{\left[\frac{\kappa^4 E[|s(k)|^4]}{1-a^4}\right] + \frac{6a^2\kappa^4\{E[|s(k)|^2]\}^2}{(1-a^4)(1-a^2)}\right\} \\
&- 4w_0^3 w_1 \left\{a^3\left[\frac{\kappa^4 E[|s(k)|^4]}{1-a^4} + \frac{6a^2\kappa^4\{E[|s(k)|^2]\}^2}{(1-a^4)(1-a^2)}\right]\right. \\
&\qquad\qquad \left. + \frac{3a\kappa^4\{E[|s(k)|^2]\}^2}{(1-a^2)}\right\}
\end{aligned} \tag{3.37}$$

For the purpose of obtaining a plot of the error surface, we arbitrarily assume that $a = 0.8$ and $\kappa = 0.36$. Utilizing Equations 3.36 and 3.37 we obtain the graph shown in Figure 3.1. It highlights one of the key weaknesses of the CMA algorithm, the existence of multiple minima towards which the algorithm may converge, and thus fail to reach the optimal solution [9].

3.4 Weight Vector Initialization

An issue that we have referred to but not developed a solution for, is the matter of weight vector initialization. It has been shown that in the case of the CMA algorithm in specific, the error surface presents us with multiple local minima, and therefore the possibility that the algorithm will converge towards a non-global minimum if it has not been properly initialized. While the LMS algorithm does not face the same difficulties, it's behaviour can be drastically improved and sped up if it has been initialized in a manner that ensures the weight vector needs to be adapted less times to reach the minimum MSE.

To achieve this improvement in behaviour, we have taken the following approach: In the case of a wire-line transmission system, some of the properties of the channel are known a priori. We can therefore estimate a weight

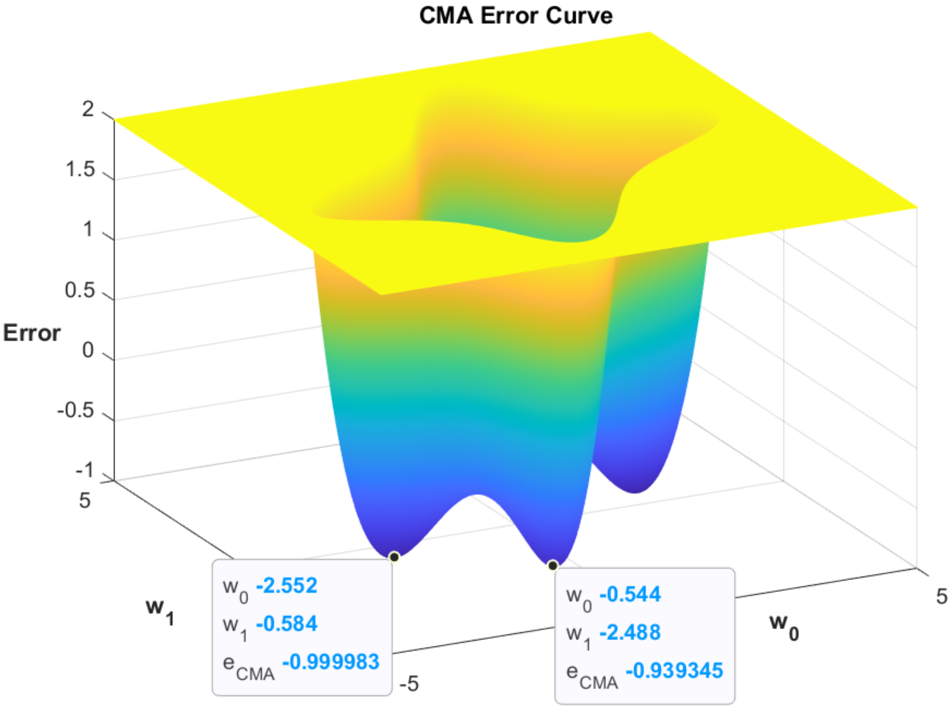


Figure 3.1: CMA Error Curve

vector that could compensate for this known behaviour, and use it as the initial condition of the equalizer. In reality, the actual behaviour of the channel will differ from the expected one, but it should be similar enough to its nominal response that the final, desirable condition of the equalizer will not be too far removed from the one we arrive at when attempting to compensate for the nominal case. In this section we will be deriving a method to do just that.

Following the symbolism we used earlier in this chapter, we denote the transmitted signal with $s(k)$, the channel impulse response with $h(t)$ and the equalized signal with $y(k)$. We also introduce the symbol $r_i(t)$ to refer to the impulse response of the i-th tap of the filter that realizes the delay line which provides us with the state vector. Ideally,

$$r_i(t) = \delta(t - iT_b)$$

where T_b is the symbol period and $\delta(t)$ is the Dirac function. We also introduce the collective impulse response of the channel and each filter tap, $c_i(t)$, which we define as

$$c_i(t) = h(t) * r_i(t) \tag{3.38}$$

If we assume that noise $n(t)$ is introduced at the input of the filter, the equalized signal would be

$$y(n) = \sum_{i=1}^{N} \sum_{k=-\infty}^{+\infty} w_i s(k) c_i(t - kT_b) + \sum_{i=1}^{N} w_i(t) n(t) * r_i(t) \tag{3.39}$$

where N is the number of filter taps. If we also assume that the ISI lasts for L samples, and therefore the cumulative response $c_i(t)$ becomes zero after L samples, we arrive at the expression

$$y(n) = \sum_{i=1}^{N} \sum_{k=n-L}^{n} w_i s(k) c_i(nT_b + t_0 - kT_b) + \sum_{i=1}^{N} w_i (n(t) * r_i(t))|_{t=nT_b+t_0} \tag{3.40}$$

where t_0 is the channel delay. We can then define the matrix $\mathbf{C}$ such that it has elements $C_{rs} = c_s((r-1)T_b + t_0)$. In this case, s would run from 1 to N and r from 1 to $L+1$. $\mathbf{C}$ would therefore be a $(L+1) \times N$ sized matrix. We also define the vectors $\mathbf{s}^T(n) = [s(n)\ s(n-1)\ \ldots\ s(n-L)]$ and $\boldsymbol{\eta}^T(n) = [n(t) * r_1(t))|_{t=nT_b+t_0}\ \ldots\ n(t) * r_N(t))|_{t=nT_b+t_0}]$, which leads us to the expression

$$y(n) = \mathbf{s}^T(n)\mathbf{C}\mathbf{w} + \boldsymbol{\eta}^T(n)\mathbf{w} \tag{3.41}$$

If the equalizer introduces a delay of δ symbols, we can also define the vector $\mathbf{h}_\delta^T = [0\ \ldots\ 0\ 1\ 0\ \ldots\ 0]$ where δ zeros are followed by a 1, which is then followed by $L - \delta$ zeros. Then the desired equalizer output would be $s(n-\delta) = \mathbf{s}^T(n)\mathbf{h}_\delta$ and the equalizer error would be expressed by

$$\begin{aligned} e(n) &= y(n) - \mathbf{s}^T(n)\mathbf{h}_\delta \\ &= \mathbf{s}^T(n)[\mathbf{C}\mathbf{w} - \mathbf{h}_\delta] + \boldsymbol{\eta}^T(n)\mathbf{w} \end{aligned} \tag{3.42}$$

The expected value of the square error would then be

$$\mathrm{E}[|e(n)|^2] = (\mathbf{C}\mathbf{w} - \mathbf{h}_\delta)^T \mathrm{E}[\mathbf{s}(n)\mathbf{s}^T(n)](\mathbf{C}\mathbf{w} - \mathbf{h}_\delta) + \mathbf{w}^T \mathrm{E}[\boldsymbol{\eta}(n)\boldsymbol{\eta}^T(n)]\mathbf{w} \tag{3.43}$$

The expected value $\mathrm{E}[\mathbf{s}(n)\mathbf{s}^T(n)]$ is given by $\sigma_\alpha^2 \mathbf{I}$ where $\sigma_\alpha^2 = \mathrm{E}[|s(n)|^2]$ and

$\mathrm{E}[\boldsymbol{\eta}(n)\boldsymbol{\eta}^T(n)]$ is given by

$$\mathbf{M} = \frac{N_0}{2}\begin{bmatrix} \int_0^\infty r_1(t)r_1(t)dt & \dots & \int_0^\infty r_1(t)r_N(t)dt \\ \int_0^\infty r_2(t)r_1(t)dt & \dots & \int_0^\infty r_2(t)r_N(t)dt \\ \vdots & & \vdots \\ \int_0^\infty r_N(t)r_1(t)dt & \dots & \int_0^\infty r_N(t)r_N(t)dt \end{bmatrix} \tag{3.44}$$

so that

$$\mathrm{E}[|e(n)|^2] = \sigma_\alpha^2(\mathbf{Cw} - \mathbf{h}_\delta)^T(\mathbf{Cw} - \mathbf{h}_\delta) + \mathbf{w}^T\mathbf{Mw} \tag{3.45}$$

In the ideal case, and in most filter implementations, the matrix $\mathbf{M}$ would be diagonal, since the tap impulse responses are orthogonal. From this expression we arrive at the optimal solution for the weight vector

$$\mathbf{w}_{\mathrm{opt}} = \mathbf{A}^{-1}\mathbf{C}^T\mathbf{h}_\delta \tag{3.46}$$

where $\mathbf{A} = \mathbf{C}^T\mathbf{C} + \lambda\mathbf{M}$ and $\lambda = {}^1\!/\!{}_{\sigma_\alpha^2}$ [13].

3.5 Comparison of Solutions

For the purpose of comparing the two algorithms presented above, we transmitted a QPSK signal through a simulated transmission channel and attempted to equalize the signal received at its output, first with a CMA-based equalizer, and then with an LMS-based equalizer.

3.5.1 A CMA Equalizer

Using the update equation developed in 3.32 in a Matlab script, we have simulated the equalization of a QPSK-based telecommunication system. The transmission channel was modeled after a Plastic Microwave Fiber (PMF) with a transfer characteristic generally described by the expression

$$H(f) = e^{-k_1 f L - k_2 f^{k_3} L} \tag{3.47}$$

where k_1, k_2 and k_3 are coefficients that, through an optimization process to fit the results received from measuring the s-parameters of a PMF, where chosen to be 0.0068, 19.267 and 1.0775 respectively. L represents the length of the channel, which was chosen to be 1m, and f the carrier frequency. The resulting transfer function magnitude can be seen in Figure 3.2.

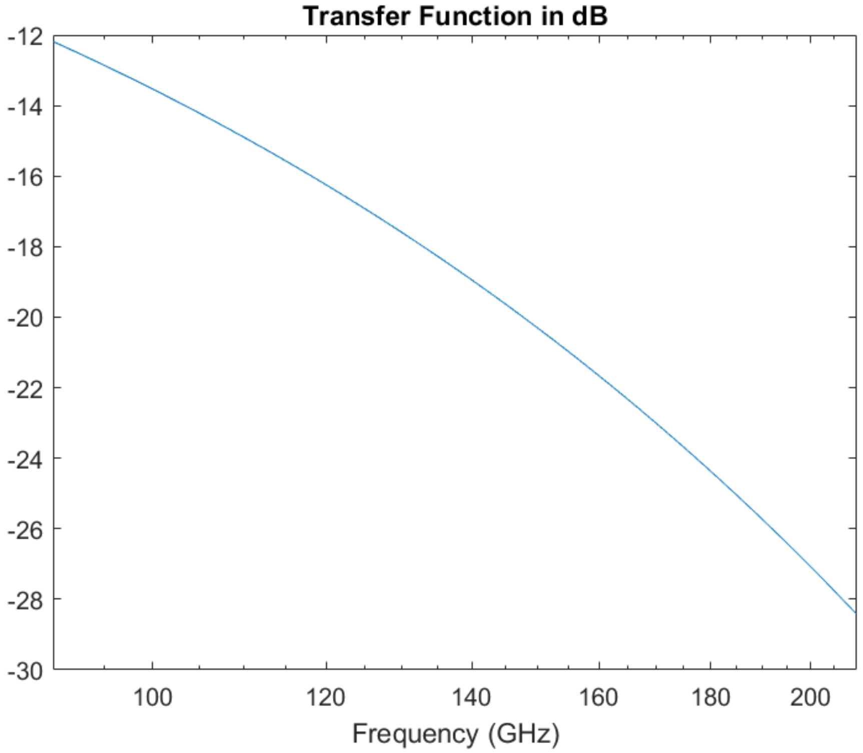


Figure 3.2: Channel Transfer Function as Modeled by Equation 3.47

To show the effect the length of the coefficients vector has on the convergence of the algorithm, we performed a comparison between two CMA implementations, one where the vector length was 7, and another where it was 15. The code implementing this algorithm is shown in Listings 3.1 and 3.2, with the first one containing the lines dealing with the initialization of the system and the signals, and the modulation, transmission and demodulation of the signal to be equalized. Listing 3.2, on the other hand, deals with the actual equalization of the simulated signal.

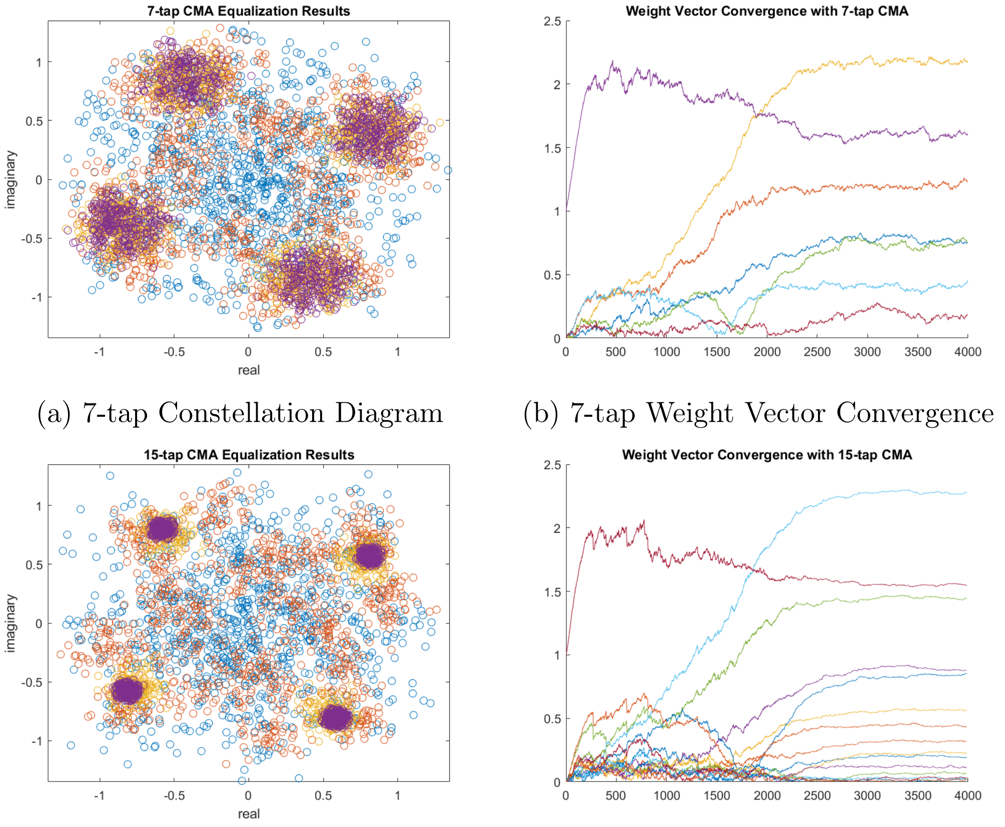
The results of this comparison can be seen in Figure 3.3, where the equalized symbols at the beginning of the process are shown in blue, the ones from intermediate states in yellow and orange, and the symbols corresponding to the final, stable state of the equalizer are shown in purple. This shows the evolution of the equalized constellation as the adaptation process is being performed. It is evident that though the process has not been sped up by increasing the number of equalizer taps, the convergence is much better with regards to the steady state error. Another issue of the Godard family of algorithms becomes obvious through this figure: the algorithms are phase blind. Though they may push the symbols towards the unit cycle, they have no way of correcting any phase shift that might have been added to the symbol constellation through the transmission of the signal.

```
N = 4000;     % number of sample data
dB = 25;      % Signal to noise ratio(dB)
L = 7; % number of taps, smoothing length L+1
ChL = 2100;   % length of the channel= ChL+1
EqD = 7;  %  channel equalization delay

%channel
f = linspace(0,210,2100);
Ch = exp(-0.0068*f-1i*19.267*f.^1.0775);
Ch_t = ifft(Ch);

% QPSK symbols
TxS = (round(rand(1,N))*2-1)/sqrt(2);
TxS = TxS + (1i*(round(rand(1,N))*2-1))/sqrt(2);

%channel distortion and time domain signal
bitrate = 8*10^9;
f_c = 145*10^9;
Tb = 1/bitrate;
t = 0:(Tb/999):Tb; %1000 samples per symbol period
y = []; % to store time domain signal
for k = 1:N
    y_temp = real(TxS(k))*cos(2*pi*f_c*t) + imag(TxS(k))*sin(2*pi*f_c*t);
    y = [y y_temp];
end

Ch_t = ifft(Ch);
r = conv(y, Ch_t);


%demodulation
demod_i = cos(2*pi*f_c*t);
demod_q = sin(2*pi*f_c*t);
r_const = [];
for i = 1:length(TxS)
    temp = r(1000*(i-1)+1:1000*i);
    r_i = cumtrapz(temp.*demod_i)/pi;
    r_q = cumtrapz(temp.*demod_q)/pi;
    r_const = [r_const r_i(1000) + 1i*r_q(1000)];
end
r_const = 0.01*transpose(r_const);
```

Listing 3.1: Blind CMA Matlab Implementation - Initialization, Modulation, Transmission and Demodulation

```
%estimation using CMA
K = N - L;    % Discard initial samples for avoiding 0's and negative
X = zeros(L+1, K);  % each vector
for i=1:K
    X(:,i) = r_const(i+L:-1:i).';
end
X = 10*X; % pre-equalization amplification
e = zeros(1,K);  % to store the error signal
sym = zeros(1,K); % to store equalized symbols
c = zeros(L+1,1); c(EqD)=1; % initial condition
R2 = 1; % constant modulous of QPSK symbols
mu = 0.03; % step size

coeff_sink = [];
coeff_sink = [coeff_sink c];
for i=1:K
   sym(i) = c'*X(:,i);
   e(i) = norm(sym(i)).^2-R2 ;  % initial error
   c = c - mu*2*e(i)*X(:,i)*X(:,i)'*c; % update co-efficients
   coeff_sink = [coeff_sink c];
end
```

Listing 3.2: Blind CMA Matlab Implementation - Equalization



(a) 7-tap Constellation Diagram

(b) 7-tap Weight Vector Convergence

(c) 15-tap Constellation Diagram

(d) 15-tap Weight Vector Convergence

Figure 3.3: CMA Performance Comparison: (a) and (b) show the behaviour of a 7-tap system, (c) and (d) of a 15-tap system

3.5.2 A LMS Equalizer

Using the same setup as in the previous simulation, we also simulated two different implementation of the LMS algorithm-based equalizer. In the first implementation, the equalizer was provided with a training sequence, a known string of symbols that was transmitted through the channel, that functions as a reference signal for the equalizer to recognise the effects the channel has on the transmitted signal and adapt to its behavior. The code performing this implementation is provided in Listings 3.3 and 3.4. The algorithm initialization is the same as that of Listing 3.1, and so it has been omitted. In the first provided code abstract, the system equalizes the symbols of the training sequence, calculating the error signal by comparing the equalizer output to the known symbols of the training sequence. After the training sequence has been transmitted and equalizer, the equalizer should have reached a state that can compensate for the transmission effects, and attempts to equalize the unknown transmitted symbol sequence blindly. This operation is performed by the code of Listing 3.4.

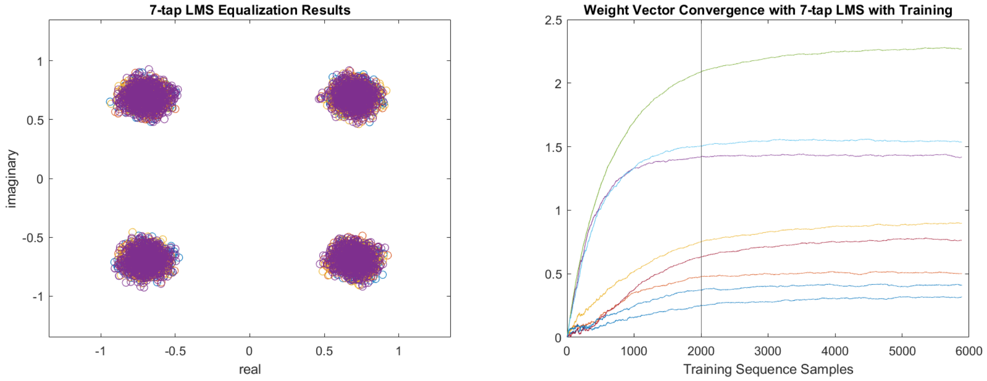
The results of this process are shown in Figure 3.4. The equalized symbols are presented on the constellation diagram following the logic of Figure 3.3, with purple denoting the symbols received after the weight vector has converged. The convergence behavior of the weight vector is clearly illustrated and exactly as expected. The behavior of the error signal can also be seen in Figure 3.5. It can be seen that it quickly approaches zero, but never reaches it. There is a constant, slight adjustment of the weight vector, that causes this "instability" in the error signal.

```
for i=1:N-1
    w(:,i) = w_init;
end

sb = zeros(sig_length,1);
e = sb;
for i=1:N-1
    sb(i) = sign(real(y_const(i)))/sqrt(2)
        +1i*sign(imag(y_const(i)))/sqrt(2); % decision system
    e(i) = y_const(i) - sb(i); %error
end
for i=N:train_length
    sb(i) = sign(real(y_const(i)))/sqrt(2)
    +1i*sign(imag(y_const(i)))/sqrt(2); % decision system
    e(i) = y_const(i) - sb(i); %error with training sequence
    corr(:,i) = 2*mu*e(i)*x(:,i);
    if real(sb(i)) == 0 || imag(sb(i)) == 0
        w(:,i) = w_init;
    elseif abs(e(i))>tol
        w(:,i) = w(:,i-1) - corr(:,i);
        % assume weight saturation to avoid divergence
        for j=1:N
            if abs(real(w(j,i))) > 1
                w(j,i) = sign(real(w(j,i)))+1i*imag(w(j,i));
            end
            if abs(imag(w(j,i))) > 1
                w(j,i) = real(w(j,i))+1i*sign(imag(w(j,i)));
            end
        end
    else
        w(:,i) = w(:,i-1);
    end
end
```

Listing 3.3: Training Sequence LMS Matlab Implementation - Training Sequence Equalization

```
for i=train_length+1:sig_length
    sb(i) = sign(real(y_const(i)))/sqrt(2)
        +1i*sign(imag(y_const(i)))/sqrt(2); % decision system
    e(i) = y_const(i) - sb(i); %error
    corr(:,i) = 2*mu*e(i)*x(:,i);
    if real(sb(i)) == 0 || imag(sb(i)) == 0
        w(:,i) = w_init;
    elseif abs(e(i))>tol
        w(:,i) = w(:,i-1) - corr(:,i);
        % assume weight saturation to avoid devergence
        for j=1:N
            if abs(real(w(j,i))) > 1
                w(j,i) = sign(real(w(j,i)))+1i*imag(w(j,i));
            end
            if abs(imag(w(j,i))) > 1
                w(j,i) = real(w(j,i))+1i*sign(imag(w(j,i)));
            end
        end
    else
        w(:,i) = w(:,i-1);
    end
end
```

Listing 3.4: Training Sequence LMS Matlab Implementation - Signal Equalization



(a) Constellation Diagram

(b) Weight Vector

Figure 3.4: LMS Algorithm with Training Sequence

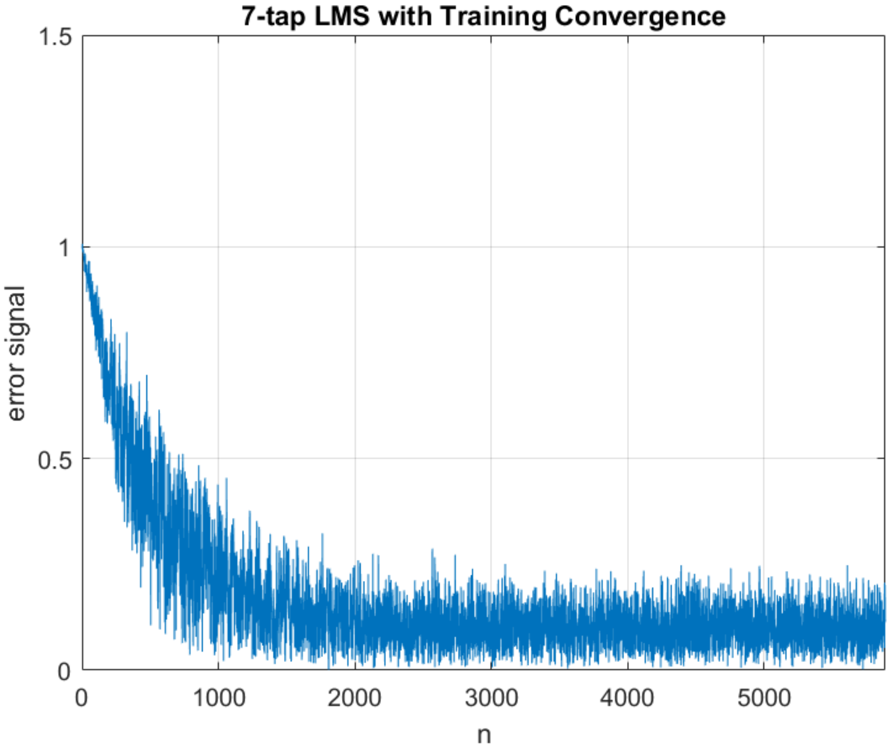
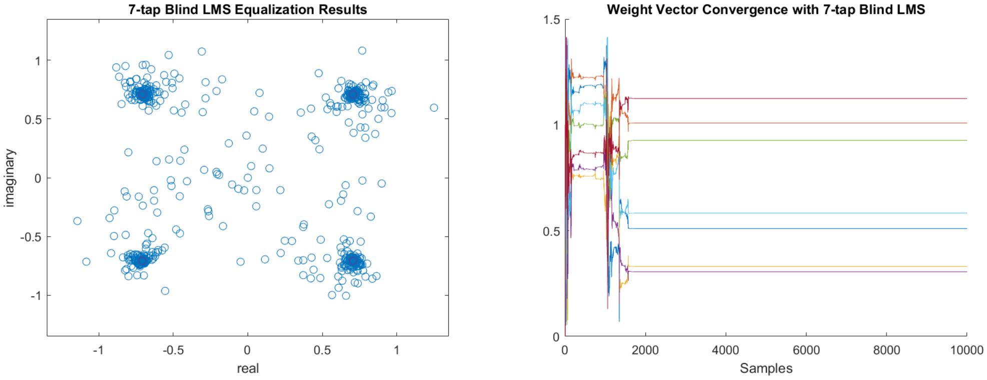


Figure 3.5: Error Signal of LMS Equalizer with Training Sequence

In the second implementation, we adopted a "blind" approach to the implementation of the LMS algorithm, in the sense that the reference signal is not externally provided to the equalizer, but is rather generated within it by a decision system, as in the block diagram of Figure 2.7, that is implemented as a quantizer. To avoid the situation where the equalizer reaches a state where the error signal becomes near zero, but diverges from it, we have included a break clause in the adaptation process: once the error signal becomes zero, even if for a single sample, stop updating the weigh vector. The behavior of the weight vector, as seen in Figure 3.6, highlights the importance of this clause: it can be seen that such a situation occurs, with the error signal approaching but not reaching zero and the vector temporarily reaching a stable state, but then diverging from it, at around the 1000th sample. Later on, it reaches zero, and the process freezes, completely halting the adaptation of the weights. This implementation appears to give much better results than the previous one, as attested by the constellation diagram of Figure

3.6. Figure 3.7 shows the behavior of the error signal.



(a) Constellation Diagram

(b) Weight Vector

Figure 3.6: LMS Algorithm without Training Sequence

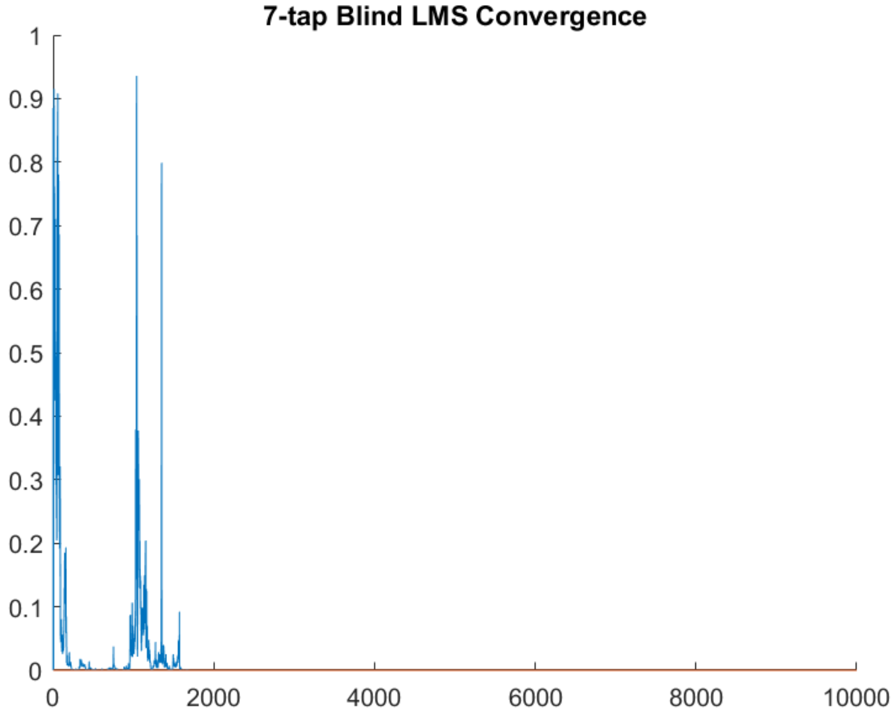


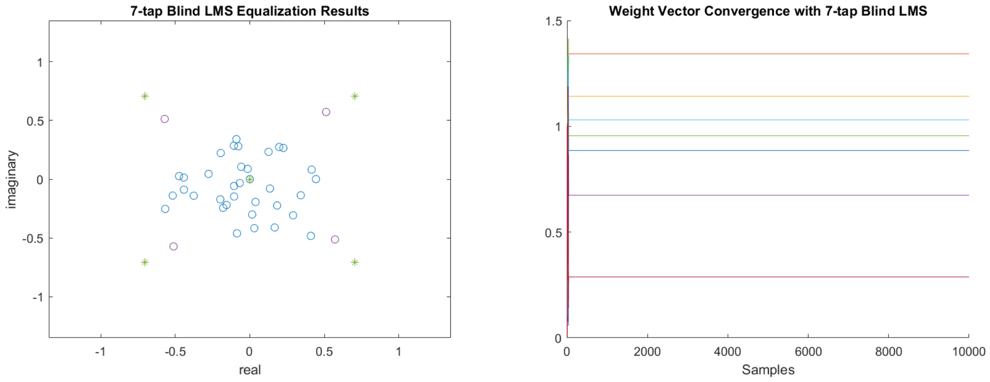
Figure 3.7: Error Signal of LMS Equalizer without Training Sequence

In the case that a certain level of remaining error can be tolerated, the adaptation process can be stopped even earlier. While this approach provides us with a less than ideal constellation diagram at the output of the equalizer, it can save a lot of resources, by turning off the energy-heavy digital logic necessary for the implementation of the LMS algorithm earlier. The code implementing this algorithm is provided in Listing 3.5. The results of this approach can be seen in Figures 3.8 and 3.9. The error threshold was chosen to be at $c_{\mathbf{tol}} = 0.3$, and it can be seen that the error signal reaches it almost immediately, after less than 50 samples. To evaluate the quality of the equalized constellation diagram, the ideal points of the constellation are shown in Figure 3.8a with green asterisks. The purple symbols, which are the result of the stable state of the equalizer, do not perfectly coincide with the ideal points, but are close and consistent enough that this remaining error can be considered inconsequential.

The code that implements both of the aforementioned systems is presented in Listing 3.5. The variable `tol` represents the error tolerance we have imposed on the algorithm, and can be set to zero for the first implementation, or another value representing acceptable error for the second implementation. As before, the initialization is performed the same way it was in Listing 3.1 and has not been repeated for the sake of brevity.

```
%% equalization
w(:,1) = w_init;
sb = zeros(sig_length,1);
y = zeros(sig_length,1);
e = zeros(sig_length,1);
for i=1:N-1
    y(i) = w(:,i)'*r(:,i);
    sb(i) = sign(real(y(i)))/sqrt(2)
       + 1i*sign(imag(y(i)))/sqrt(2); % decision system
    e(i) = y(i) - sb(i); %error
end
for i=N:sig_length
    y(i) = w(:,i-1)'*r(:,i); % eq filter
    sb(i) = sign(real(y(i)))/sqrt(2) +1i*sign(imag(y(i)))/sqrt(2); % decision system
    e(i) = y(i) - sb(i); %error
    corr(:,i) = 2*mu*e(i)*R(:,i-N+1);
    if real(sb(i)) == 0 || imag(sb(i)) == 0
        w(:,i) = w_init;
    elseif abs(e(i))>tol
        w(:,i) = w(:,i-1) - 2*mu*e(i)*R(:,i-N+1);
        % assume weight saturation to avoid divergence
        for j=1:N
            if abs(real(w(j,i))) > 1
                w(j,i) = sign(real(w(j,i)))+1i*imag(w(j,i));
            end
            if abs(imag(w(j,i))) > 1
                w(j,i) = real(w(j,i))+1i*sign(imag(w(j,i)));
            end
        end
    else
        w(:,i) = w(:,i-1);
    end
end
```

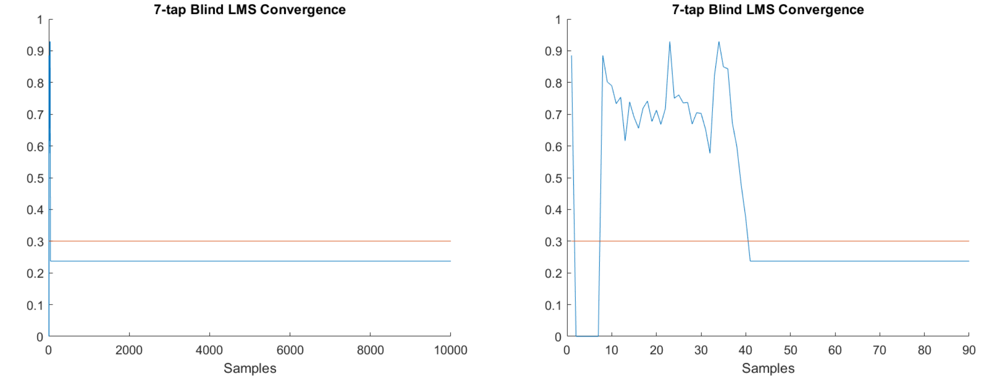
Listing 3.5: Blind LMS Algorithm Matlab Implementation



(a) Constellation Diagram

(b) Weight Vector

Figure 3.8: LMS Algorithm with Error Tolerance



(a) Full Vector

(b) Close-up

Figure 3.9: Error Signal

From this analysis we reach the conclusion that the LMS algorithm is by far more suitable to our needs, and its perhaps most significant drawback, the energy consumption of the digital logic circuit, can be minimized by introducing an acceptable error threshold. From this conclusion, we decided to proceed with the implementation of the adaptive equalizer utilizing the LMS algorithm.

Chapter 4

System Implementation

In this chapter we will present an implementation of the complete equalizer system. As mentioned previously, the comparison of the two adaptive algorithms we performed lead us to the decision to design an equalizer based on the LMS algorithm. For the implementation of this design, there are several IC's necessary. These IC's were all realized using Infineon Technologies' B11HFC technology, which we will shortly introduce before proceeding to explain the IC's utilized for this design.

4.1 Infineon Technologies' B11HFC Technology

The B11HFC technology has been developed for the purpose of designing IC's operating in the micro- and mm-wave frequency bands, and it provides designers with a highly dependable and extensive set of basic building blocks to allow them to realize their designs. It is a 400GHz/130nm SiGe BiCMOS technology with copper plating suitable for analog and mixed signal applications. It is employed in the fields of Automotive Radar MMIC's, RF ASIC's and other applications such as high speed wireless links. It includes three types of NPN HBT's of customizable size and multiplicity, namely high speed, medium speed and high voltage bipolar junction NPN transistors. In the context of this work, we mostly utilized the high speed NPN HBT, whose characteristics we will shortly introduce.

The high speed NPN HBT is described (and customized) by two parameters, emitter length and width, which are allowed to take values within the ranges of 0.7 to 10 μm and 0.22 to 0.34 μm respectively. The effective area of

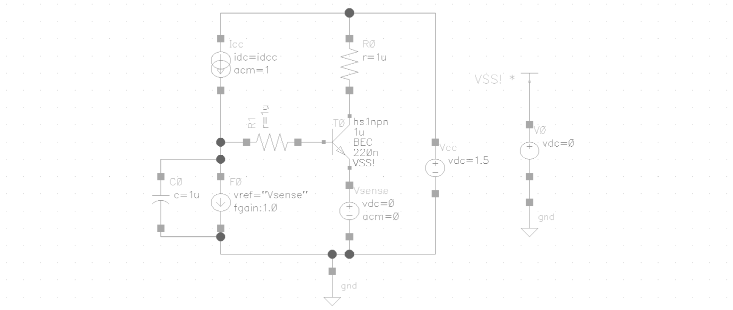


Figure 4.1: Schematic for Device Characteristics Determination

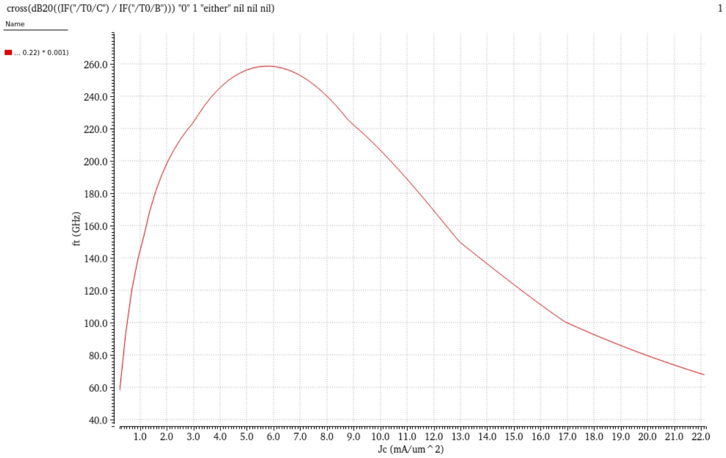


Figure 4.2: Transistor f_T

the emitter, A_{eff}, is not exactly equal to the product of the emitter length and width, as the masking process adds an extra 0.09 μm both length and width wise, resulting in an extra area that must be subtracted from the emitter area. To derive the device characteristics we utilized the schematic seen in Figure 4.1. The simulation results are presented in Figure 4.2. This graph shows that the device achieves its peak maximum frequency of 260 GHz when biased with a current density of 5.6 mA/μm^2.

4.2 Low Pass Filter Implementation

The basic blocks of the analog FIR filter architecture are the elements that introduce delay to the received signal. Though there are many solutions available in the bibliography, including Gm-C or switched capacitor delay cells, these may face power dissipation issues when the operating frequency reaches high enough levels. With operating frequencies reaching the microwave domain, designers have adopted microwave techniques to realise these FIR filters, mostly by implementing the required delay units for a tapped delay line architecture through the use of on-chip transmission lines, thus forming a transversal or travelling wave filter architecture.

Even this approach, however, faces several issues, that can make the designing process difficult. Firstly, the low impedance presented by the ideal doubly terminated delay line to the driver circuit causes a significant reduction to the tap voltages (by a factor of 2 for a transversal filter, and a factor of 4 for a travelling wave filter), which in turn makes a higher gain necessary. Unfortunately, increasing the gain of the gain cells also increases their input capacitance, resulting in a lower cutoff frequency for the delay line, which obviously degrades equalizer performance (increased "ringing" in the tap responses).

Another significant issue is presented by the transmission line nonidealities, which affect transversal filters more severely than travelling wave architectures, by introducing losses that have to be compensated by the tap weights, increasing overall power dissipation. While methods to minimize these losses do exist, they result in large, unwieldy delay lines.

An important question that needs to be addressed during the design of a FIR filter equalizer is the timing sensitivity. While the timing issues inherent to symbol-spaced architectures are well known and usually combated by the adoption of a fractionally spaced implementation instead, it can be shown that delay line nonidealities introduce issues of phase-timing sensitivity even in these improved architectures. Traditionally, this would be solved through the introduction of a Clock Phase Recovery (CPR) system. In the application specific data rates, it is preferred to implement this system deductively, by using the output of the Transimpedance Amplifier/Variable Gain Amplifier (TIA/VGA) block that usually precedes the equalizer in a receiver chain, which should only be affected by the ISI introduced by the transmission channel. Unfortunately, this technique is only applicable to transmission channels with relatively good behavior, and can be difficult to use in certain applications. It is therefore preferable to adopt an implementation that renders a CPR system unnecessary.

We have already discussed the LMS algorithm in length, and seen many of its advantages and disadvantages. An aspect of the algorithm's behavior that we have not yet discussed, and that might introduce an additional difficulty in its implementation, is the fact that it requires that the state signals linearly combined by the equalizer be readily available in the filter, which is not the case in traveling wave architectures. This dependence is made obvious by Equation 3.14, which we derived in Chapter 3.

Another important issue is presented by the periodic nature of the frequency response of the tapped delay line filter. Due to this property, out of band components of the receiver input noise can have a severe effect on overall equalizer performance. These components will be somewhat attenuated by the finite bandwidth of the TIA, but the lack of reliability of the TIA stop-band rejection (TIA bandwidth may vary by as much as 20% depending on environmental conditions and between different lots) renders the inclusion of an anti-alias filter preferable if not necessary. This does not come at a negligible cost, since the high operating frequencies would result in an additional large area consumption by this filter, for a system that would already occupy a large area on chip.

The majority of the issues discussed above can be traced back to the chosen architecture, the tapped delay line implementation. Therefore, in our work we have pursued the design of a FIR filter that does not rely on delay lines. For this purpose, a singly terminated ladder filter with a Butterworth response was designed to function as the core of the proposed adaptive equalizer, providing the tap responses that, combined by tunable transconductors in a weighted sum with weights decided by the LMS algorithm, would reconstruct the originally transmitted signal from the noisy received one [13].

4.2.1 Butterworth Filter Design

The Butterworth filter has a maximally flat frequency response in the pass band, and rolls off towards zero in the stop band. When depicted on a Bode plot, this response should decrease linearly towards negative infinity. The filter order affects the slope of the decrease of its response by -10dB per decade per order (a first-order filter response decreases by -10 dB per decade, a second-order one by -20 dB per decade and so on). Since the Butterworth filter response is maximally flat in the pass band, and presents no ripples in the stop band, its magnitude should be a monotonic function of the frequency ω. Compared to other popular filter architectures, like the Chebyshev filters of both types, or the elliptic filter, the Butterworth response presents a slower roll-off, and must therefore be of a higher order so as to meet strict cutoff specifications. This comparison, on the other hand, also leads us to the conclusion that the Butterworth phase response is much more linear in the pass band than what is achievable by these other architectures [14].

The gain of an n-th order Butterworth filter as a function of frequency, $G(\omega$, is given in terms of its transfer function, $H(f)$, by the following equation:

$$G^2(\omega) = |H(j\omega)|^2 = \frac{G_0^2}{1 + \left(\frac{j\omega}{j\omega_c}\right)^{2n}} \tag{4.1}$$

where ω_c is the cutoff frequency, which is approximately equal to the -3 dB frequency, and G_0 is the gain at $\omega = 0$, usually referred to as the DC gain.

To implement a filter with this desired frequency response, we may choose among many different topologies. In our case, we have adopted the Cauer topology, which uses passive elements (shunt capacitors and series inductors) to implement an analog filter. When utilizing this topology, the k-th element of the filter is given by

$$C_k = 2\sin\left[\frac{2k-1}{2n}\pi\right], \quad k = \text{odd} \tag{4.2}$$

$$L_k = 2\sin\left[\frac{2k-1}{2n}\pi\right], \quad k = \text{even} \tag{4.3}$$

In the case of a voltage driven filter, as is the one in this implementation, the first element must be a series inductor, and therefore all series elements should correspond to odd k values, while shunt elements should correspond to even k values. Equations 4.2 and 4.3 apply to doubly terminated filters with unity source and load impedance, and cutoff frequency $\omega_c = 1$, but they can be adjusted for different values of impedance and cutoff frequency. For the case of a singly terminated voltage driven filter, which is the one that concerns us, the element values are determined by:

$$g_j = \frac{a_j a_{j-1}}{c_{j-1} g_{j-1}}, \qquad j = 2, 3, ..., n \tag{4.4}$$

with

$$g_1 = a_1$$

where

$$a_j = \sin\left[\frac{2j-1}{2n}\pi\right], \qquad j = 1, 2, ..., n$$
$$c_j = \cos^2\left[\frac{j}{2n}\pi\right], \qquad j = 1, 2, ..., n \tag{4.5}$$

with g_j denoting the immittance of element j. The filter architecture can be seen in Figure 4.3. The element values are also presented in the following Table:

Element	Value
C_1	70 fF
C_2	270 fF
C_3	270 fF
C_4	70 fF
L_1	600 pH
L_2	963 pH
L_3	600 pH

Table 4.1: Ladder Element Values

As mentioned above, the choice of utilizing the LMS algorithm in our adaptive schema makes the presence of all state signals necessary. To achieve this, a dual ladder was included in the filter architecture. The values of the elements of this dual ladder are presented in the following Table:

Element	Value
C_1	240 fF
C_2	385 fF
C_3	240 fF
L_1	215 pH
L_2	867 pH
L_3	867 pH
L_4	215 pH

Table 4.2: Dual Ladder Element Values

The above Tables show the fact that both ladders are symmetrical. The frequency response of the filter can be seen in Figure 4.4. The desired maxi-

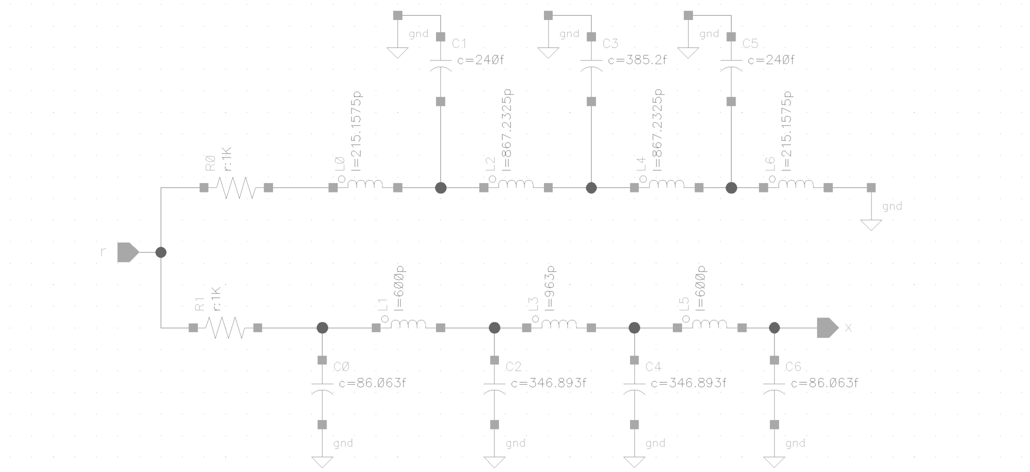


Figure 4.3: Low Pass Filter

mally flat pass band behavior has been achieved, as well as a cutoff frequency of 16.2 GHz.

4.3 Multiplier Implementation

The output of the different filter taps needs to be multiplied by the weight vector to realize the FIR filter described by the equation

$$y(k) = \mathbf{w}^T(k)\mathbf{x}(k) \tag{4.6}$$

where $y(k)$ is the equalizer output and $\mathbf{x}(k)$ is the vector of filter tap outputs. To maximize speed and minimize power dissipation, we chose to implement this multiplication process with a differential analog multiplier.

4.3.1 Analog Multiplier

The multiplier schematic can be seen in Figure 4.5. This circuit is a variable gain transconductor, that provides an output current proportional to a ratio of bias currents and the input differential voltage, and is as such equivalent to a multiplier. The input voltage seen at the gate of transistor P_6, V_{in}, determines the current that passes through resistor R_0, I_R, based on the fact

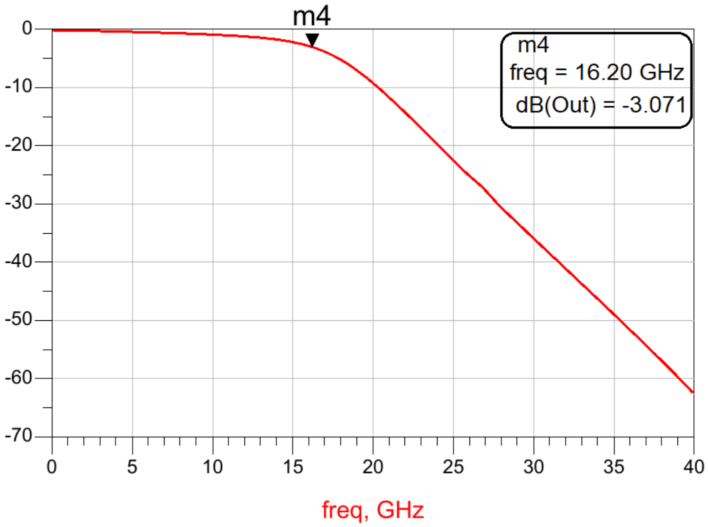


Figure 4.4: Filter Frequency Response

that

$$
\begin{aligned}
&V_{GS6} \approx V_{GS7} \\
&V_R = V_{G6} + V_{GS6} - (V_{G7} + V_{GS7}) \approx V_{G6} - V_{G7} = V_{in} \\
&I_R = \frac{V_R}{R_0} = \frac{V_{in}}{R_0} \qquad (4.7)
\end{aligned}
$$

The current that passes through the drain of transistors P_6 and P_7 is $I_1 = I_A - I_R$ and $I_2 = I_A + I_R$, respectively, where I_A is the bias current running through transistors P_1 and P_2, which operate as current sources. If V_x and V_y are the base voltages of transistors T_0 (and T_1) and T_2 (and T_3) respectively,

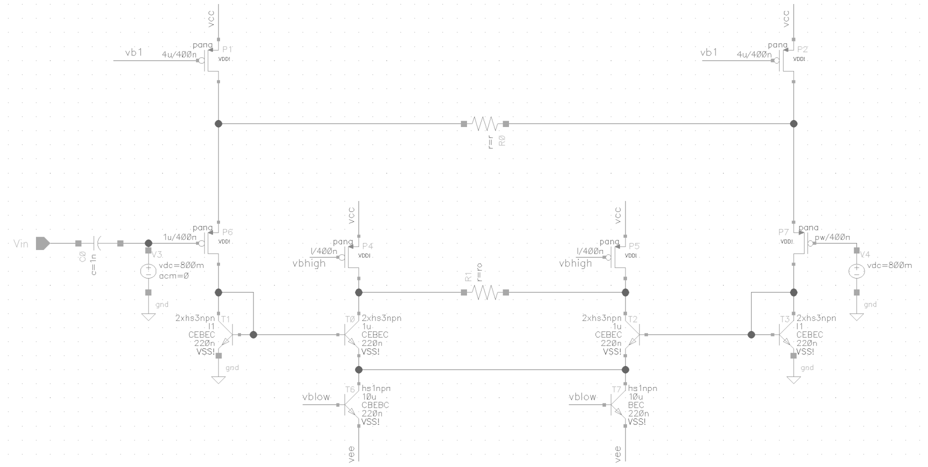


Figure 4.5: Analog Multiplier

we can arrive at the following equation:

$$V_x - V_y = V_{BE1} - V_{BE3} = V_{BE0} - V_{BE2} \tag{4.8}$$

$$I_C = I_S \exp\frac{V_{BE}}{V_T} \Rightarrow V_{BE} = V_T \ln\frac{I_C}{I_S}$$

$$(4.8) \Rightarrow V_T \ln\frac{I_{C1}}{I_S} - V_T \ln\frac{I_{C3}}{I_S} = V_T \ln\frac{I_{C0}}{I_S} - V_T \ln\frac{I_{C2}}{I_S} \Rightarrow$$

$$V_T \ln\frac{I_{C1}}{I_{C3}} = V_T \ln\frac{I_{C0}}{I_{C2}} \Rightarrow \frac{I_{C1}}{I_{C3}} = \frac{I_{C0}}{I_{C2}} \Rightarrow$$

$$\frac{I_A - I_R}{I_A + I_R} = \frac{I_B - I_O}{I_B + I_O} \Rightarrow I_O = \frac{I_B}{I_A} I_R \Rightarrow$$

$$I_O = \frac{I_B}{I_A}\frac{V_{in}}{R_0} \tag{4.9}$$

where I_B is the biasing current running through transistors P_4 and P_5, operating as current sources, as well as transistors T_6 and T_7, also operating as current sources. The biasing voltages seen at the gates of P_4 and P_5 and the bases of T_6 and T_7 are different, but they bias all four of the transistors with the same current.

Equation 4.9 confirms our original claim about the output current being proportional to the input differential voltage and the ratio of biasing currents. It also confirms that the gain of the transconductor in Figure 4.5 is variable,

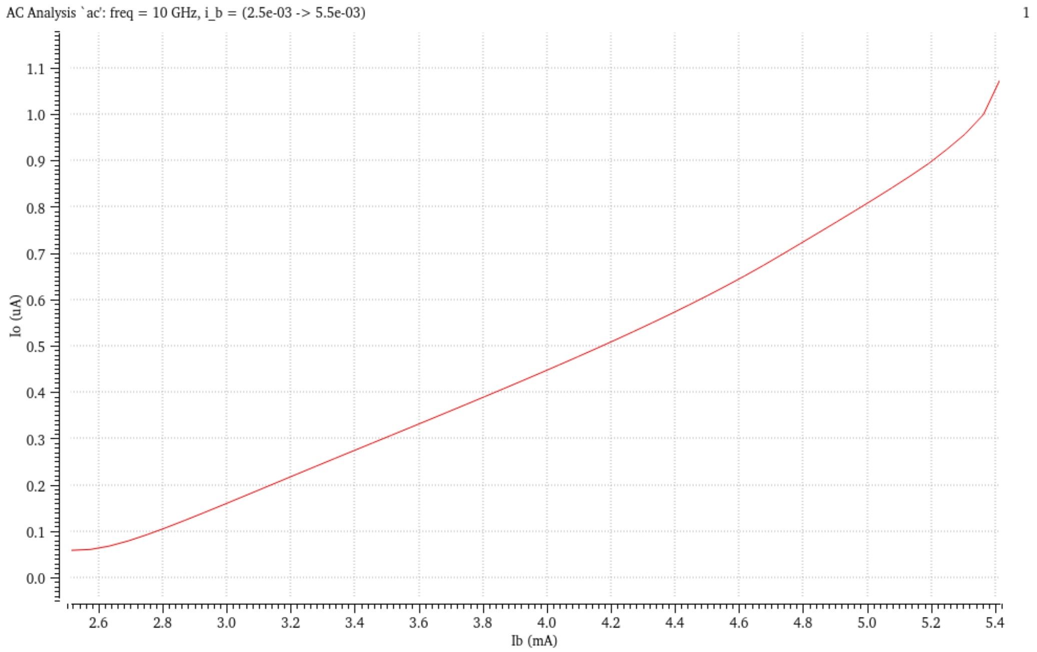


Figure 4.6: Near-Linear Area of Transconductor Operation

and can be controlled by the biasing currents of the circuit. Since the ratio of bias currents is proportional to its numerator, and we utilize the controllable transconductor as a multiplier, we can see its output as

$$I_O = G * I_B V_{in} \tag{4.10}$$

where $G = {}^{1}\!/_{R_0 I_A}$. This circuit therefore performs a multiplication of current I_B and the input voltage V_{in}. Since the transconductor must operate as a multiplier, we need it to operate linearly with regards to the input voltage for the whole range of values the biasing current I_B might take, which for our needs should correspond to the range of values the elements of the weight vector can take. A suitable area of operation is shown in Figure 4.6. It can be seen that for a biasing current within the range of 2.7 to 5.2 mA we are provided with an output linearly dependant on that current.

The only issue with this circuit is the fact that it operates on analog input signals, and the weight vector provided by the LMS algorithm would by necessity be in digital form. It is thus necessary to include a Digital-to-Analog Converter in the signal path from the digital logic circuit that realizes the LMS algorithm to the multipliers that perform the operations of Equation 4.6.

4.3.2 Digital-to-Analog Converter

A Digital-to-Analog Converter is a circuit that produces an analog signal proportional to a digital input as its output. If d is a B-bit long digital word made up of the bits b_i, it can be written as

$$d = \sum_{i=0}^{B-1} 2^i b_i, \qquad b_i \in [0, 1] \tag{4.11}$$

If this digital word is the input of a DAC, its output should be

$$u = \Delta * \sum_{i=0}^{B-1} 2^i b_i \tag{4.12}$$

where Δ is the step size of the DAC and corresponds to the minimum output, in the case of the input word being $d = 1$. It is evident from the above equation that the DAC's output can only be an integer multiple of the step Δ.

There are many available design philosophies for realizing this operation. They can generally be divided in three categories: resistor based, capacitor based and current based architectures. We will not describe the topologies belonging to these three groups in detail, but rather focus on the architecture chosen to be implemented in this work: a current-based binary 8-bit DAC [15].

This circuit is made up of 8 segmented current steering cells, the schematic of which can be seen in Figure 4.7. If I_{LSB} is the current corresponding to the least significant bit of the input digital word, that is $I_{LSB} = \Delta$, each of the eight current steering cells should be able to provide a current of

$$I_i = 2^i I_{LSB}, \qquad i \in [0, B-1] \tag{4.13}$$

If the i-th bit is zero, the corresponding current steering cell should give no output current ($I_i = 0$). In the chosen architecture, the output is differential, so in the case of an input value of k, the required output, that is $k * I_{LSB}$, is provided by the positive output node, and the negative output node provides us with a current of $I_{total} - k * I_{LSB}$, where I_{total} is the current we would see at the output of the DAC for the maximum 8-bit input word. Each cell making up the DAC also has a positive and negative output node. The positive outputs of every cell are added and fed to the DAC load. Likewise for the negative outputs.

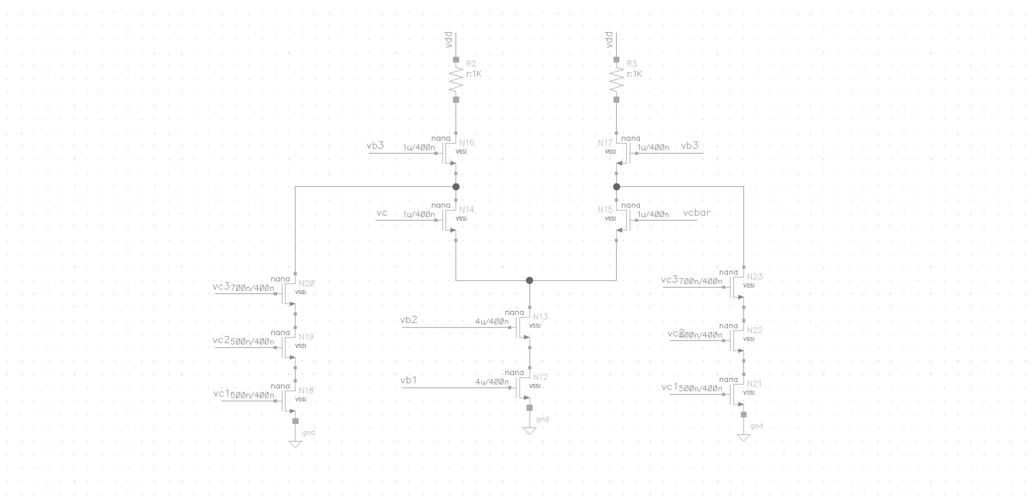


Figure 4.7: Current Steering Cell

In the cell presented in Figure 4.7, transistors N_{12} and N_{13} make up a cascode current source and should always be on. Transistors N_{14} and N_{15} are controlled by complementary gate voltages, so one of them is always on. In the case that the bit of the input word corresponding to the current cell is one, N_{14}, controlled by V_c, is on. In the case that the corresponding bit is zero, N_{15}, controlled by $\overline{V_c}$, is the one that is on. These two transistors function as the switches controlling the cell. The presence of two complementary switches that ensure the current source is always on significantly speeds up the operation of the circuit, since the large parasitic capacitors caused by the current source are always charged.

Transistors N_{16} and N_{17} are included in order to increase the cell's output resistance and also decrease the parasitic capacitance seen by the load towards the cell. The two groups of three transistors, $N_{18} - N_{20}$ and $N_{21} - N_{23}$, are current bleeding transistors. They are always on and bleeding off a small percentage of the current provided by N_{12} and N_{13} so that N_{16} or N_{17}, depending on which switch is open, N_{14} or N_{15} respectively, are not in the cutoff region, and therefore produce similar if not equal parasitic capacitances.

The output of the DAC is used as the biasing current I_B of the controllable gain transconductor presented in the previous subsection, and should be within its linear (or near-linear) area of operation. This area provides us with a range of $5.2 - 2.6 = 2.6$mA, and so the least significant bit current provided by the smallest current steering cell should be

$$I_{LSB} = \frac{2.6\text{mA}}{2^{B-1}} = 20\mu\text{A} \tag{4.14}$$

for a DAC accepting input words of length $B = 8$ bits.

When choosing this operating area, however, we introduce a certain difficulty in our design: the operating area does not start from zero, and so the output of the DAC should be added to an offset current of 2.6 mA. We must therefore introduce a current source to provide us with this offset current, and place it in parallel with the DAC, so that its output current is added to that provided by the DAC to its load, as seen in Figure 4.8. We are thus provided with a biasing current of

$$I_B = I_{offset} + w * I_{LSB} \tag{4.15}$$

and the transconductor performs the operation

$$I_O = G * I_B V_{in} = G * V_{in} \left(I_{offset} + w * I_{LSB} \right) \tag{4.16}$$

which provides us with an output current that is obviously not proportional to the product of w and V_{in}, but is offset by a factor of $G * V_{in} * I_{offset}$. This can be corrected by the use of an additional equalizer, provided with the same input voltage V_{in} and biased by the offset current I_{offset}, whose output current is subtracted from the output current of the original transconductor. The block diagram of the circuit performing this subtraction can be seen in Figure 4.9.

4.3.3 Biasing Circuits

In this subsection we will present the current mirrors and other circuits used to bias the circuits already shown above. We will also present the circuits providing the system with the necessary offset currents previously discussed.

In the case of the controllable gain transconductor, there are three reference voltages that need to be generated:

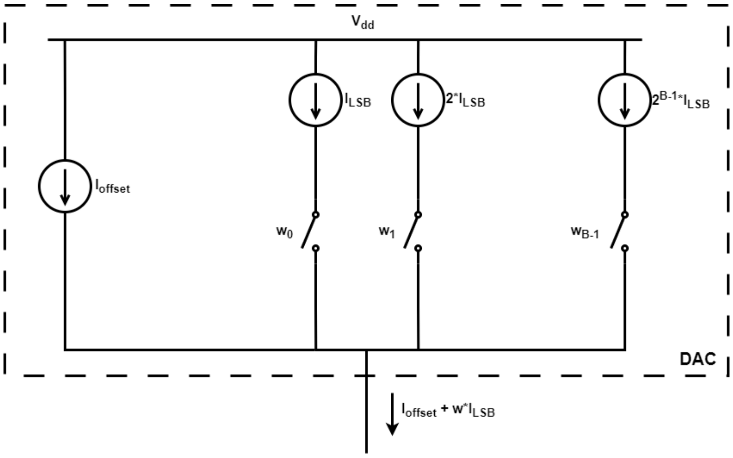


Figure 4.8: DAC with Offset Current

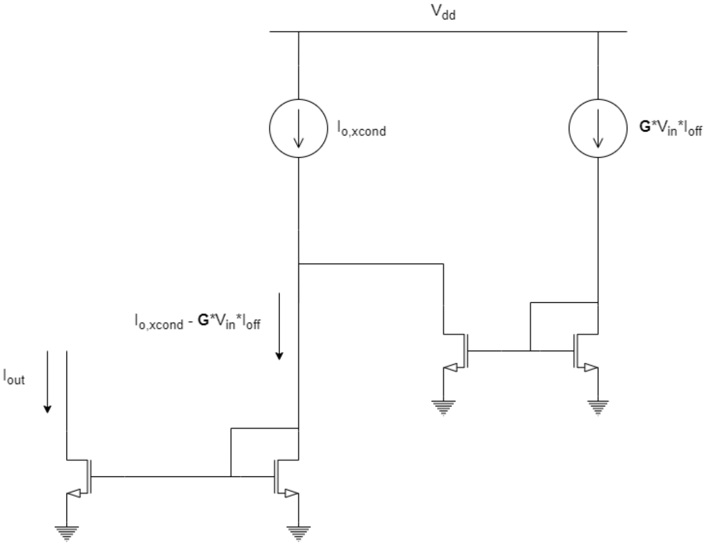


Figure 4.9: Circuit performing subtraction

- the one biasing transistors P_1 and P_2 so that they provide the rest of the circuit with biasing current I_A, noted as V_{B1} in Figure 4.5

- the one biasing transistors P_4 and P_5 so that they may mirror the current provided at the output of the DAC, noted as $V_{B,\text{high}}$ in Figure 4.5

- the one biasing transistors T_6 and T_7 so that they provide the same current as transistors P_4 and P_5, noted as $V_{B,\text{low}}$ in Figure 4.5

The latter two reference voltages can be provided by the same current mirror, as long as it has been biased by the output current of the DAC.

In the case of the DAC, on the other hand, there are six reference voltages that need to be generated:

- the one biasing transistor N_{12}, providing it with the current corresponding to the least significant bit, I_{LSB}, noted as V_{B1} in Figure 4.7

- the one biasing transistor N_{13}, noted as V_{B2} in Figure 4.7

- the one biasing transistors N_{16} and N_{17}, noted as V_{B3} in Figure 4.7

- the ones biasing the bleed-off transistors $N_{18} - N_{23}$, noted as V_{C1}, V_{C2} and V_{C3} in Figure 4.7

The biasing voltage V_{B1} determines the least significant bit current, and is crucial for the correct functioning of the whole system. The current dependance on this voltage can be seen in Figure 4.10. On the other hand, the voltage V_{B2} has much less of an impact on the behaviour of the DAC cells, as certified by the graph of Figure 4.11. The second trace in this graph shows the dependence of the second cell's output current on V_{B2}. This current should be double that of the first cell's, as the DAC is realized with a binary architecture, and each cell should be able to provide us with an output current double that of the previous cell. It can be seen that the two currents are generated by V_{B2} values within the same range. This allows for much more leeway in the design of the biasing circuit providing the system with this voltage. The schematic of the biasing circuit generating voltages $V_{B1} - V_{B3}$ and $V_{C1} - V_{C3}$ and biasing the DAC can be seen in Figure 4.12.

The biasing circuit providing the analog multiplier with biasing voltage V_{B1} can be seen in Figure 4.13. The input current I_B is provided to the multiplier via a current mirror placed at the output of the DAC.

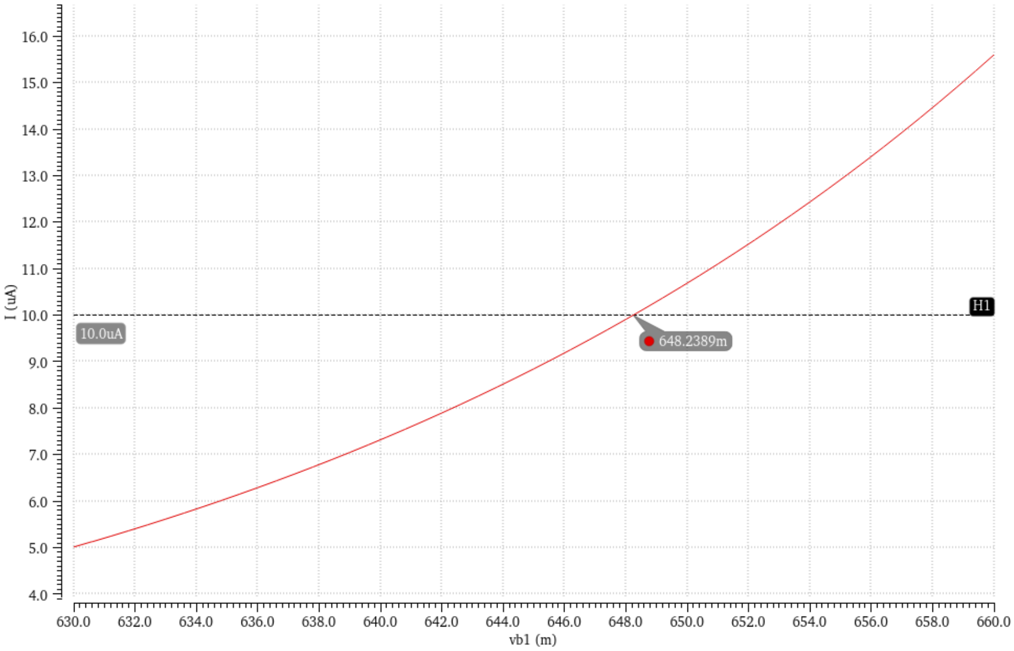


Figure 4.10: Output current dependence on Vb1

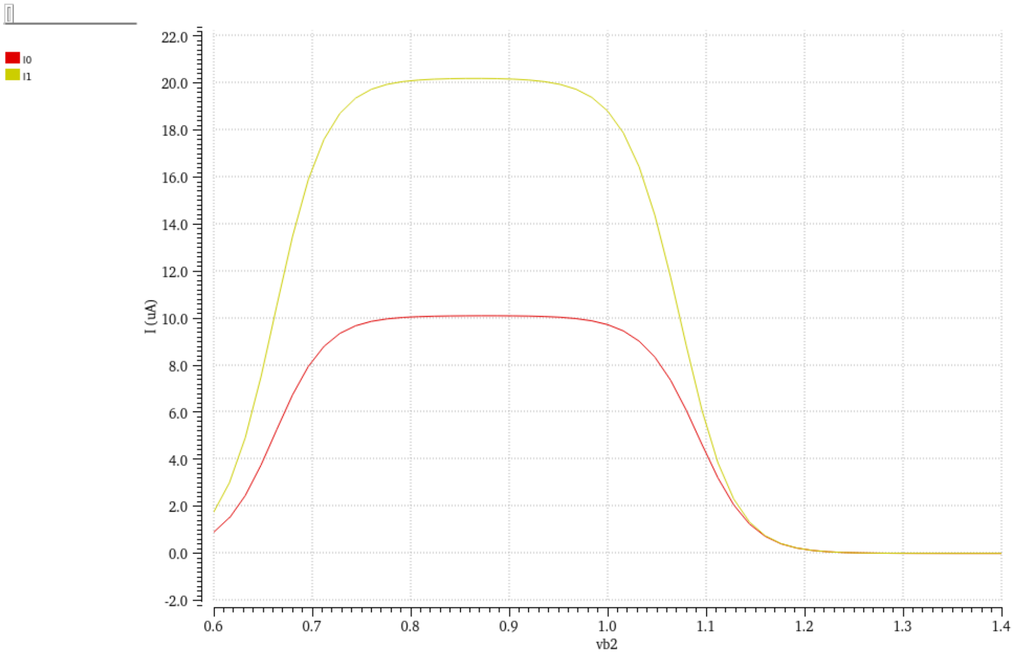


Figure 4.11: Output current dependence on Vb2

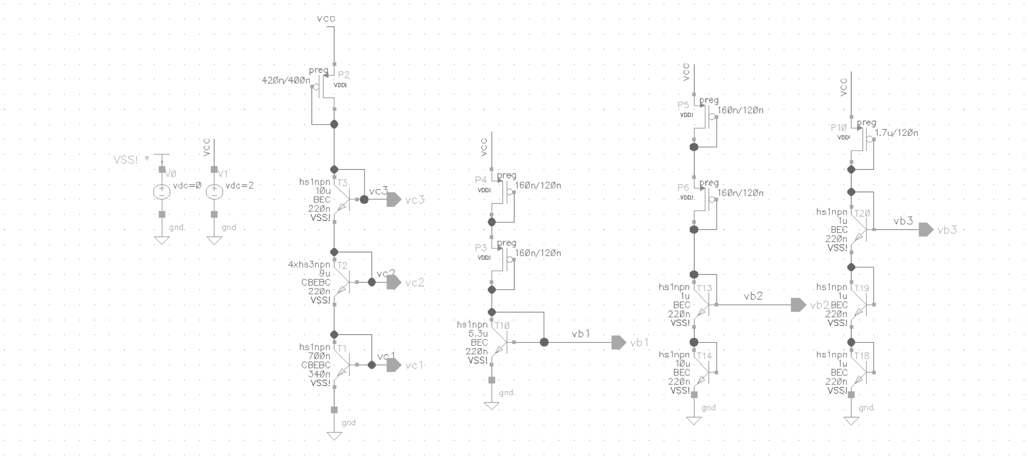


Figure 4.12: DAC Biasing Circuit

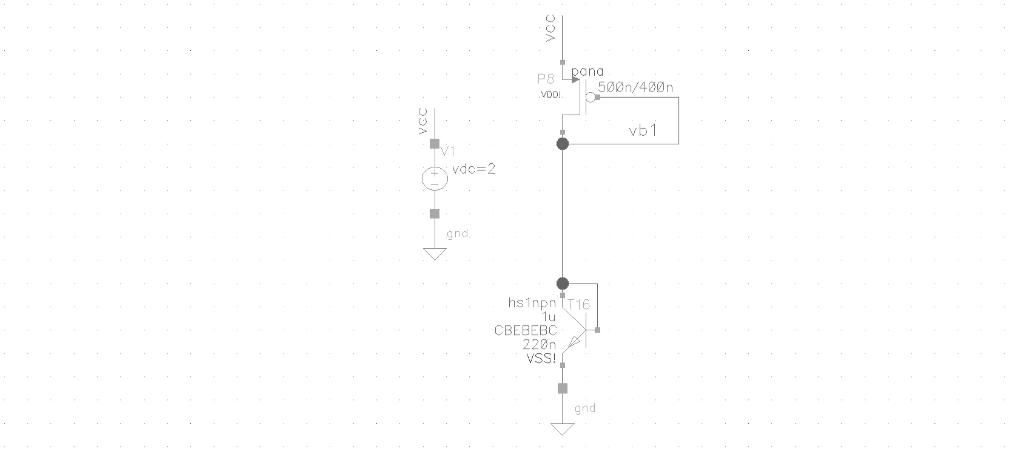


Figure 4.13: Multiplier Biasing Circuit

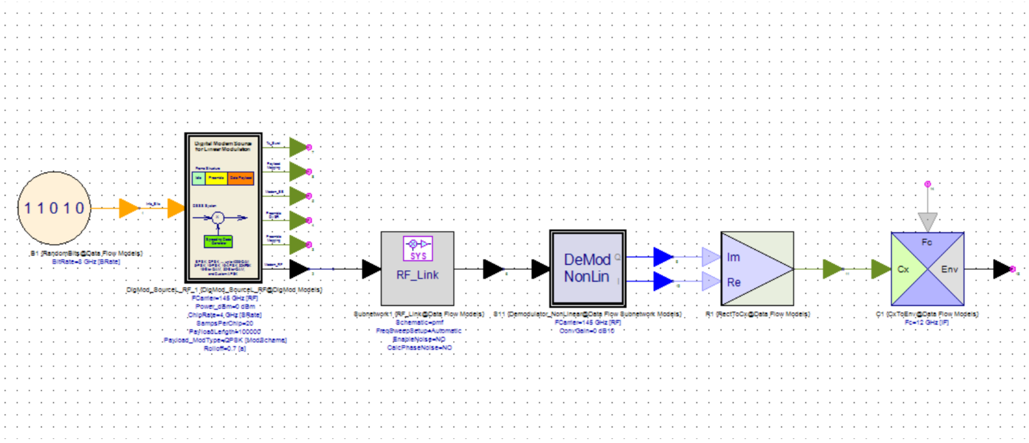
Chapter 5

Simulation Results

In this Chapter we will present the results obtained by simulating the whole equalizer system. In setting up the mixed signal system, we split it into the analog and digital parts making it up. The analog part, including signal generation, modulation and transmission, as well as the filter and analog multipliers, was simulated separately using Keysight's SystemVue tool. The results of this simulation were then used to perform an "offline" adaptation of the system through a digitally implemented LMS adaptive algorithm, which was realized using Matlab. The results obtained by this implementation of the digital part of the system were then validated by simulating a non-adaptive equalizer with the state that resulted from the adaptation process. First we will present the block diagrams used in the system simulations, the results of the simulated transmission, the Matlab scripts used to implement the digital part of the system, and finally the simulated results of the equalization process itself.

5.1 System Setup

The transmission and the analog part of the system were simulated using Keysight's SystemVue tool. This tool can be used for block diagram level system simulations, and provides high accuracy S-parameter simulation. The transmission system can be seen in Figure 5.1.



B1 {RandomBits@Data Flow Models}
BitRate=8 GHz [BRate]

DigMod_SourceL_RF_1 {DigMod_SourceL_RF@DigMod Models}
FCarrier=145 GHz [RF]
Power_dBm=0 dBm
ChipRate=4 GHz [BRate]
SampsPerChip=20
PayloadLength=100000
Payload_ModType=QPSK [ModScheme]
Rolloff=0.7 [a]

Subnetwork1 {RF_Link@Data Flow Models}
Schematic=pmf
FreqSweepSetup=Automatic
EnableNoise=NO
CalcPhaseNoise=NO

S11 {Demodulator_NonLinear@Data Flow Subnetwork Models}
FCarrier=145 GHz [RF]
ConvGain=0 dB10

R1 {RectToCx@Data Flow Models}

C1 {CxToEnv@Data Flow Models}
Fc=12 GHz [IF]

Figure 5.1: SystemVue Transmission System Block Diagram

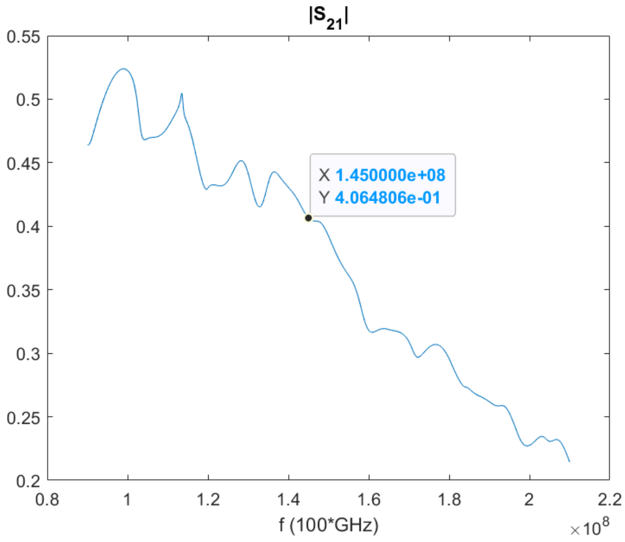


Figure 5.2: S_{21} of 1m long PMF cable

The pseudo random bit sequence generator provides us with the unmodulated bit stream that forms the signal to be transmitted. This bit stream is in turn fed into a Radio Frequency Quadrature Phase Shift Keying (QPSK) modulator, which provides us with a QPSK signal with a carrier frequency of 140 GHz. This signal, which can be seen in Figure 5.3, is then transmitted through a PMF cable, represented in Figure 5.1 by the `RF_Link` block. The cable's S-parameters were obtained by simulating a PMF cable provided by Huber & Suhner. The magnitude of parameter S_{21}, which represents the transferring behaviour of the cable, is shown in Figure 5.2. The three blocks following `RF_Link` perform the downmixing of the signal by demodulating it into a complex number that is then converted to an envelope signal with a carrier frequency of 10 GHz. The constellation received after this downmixing operation is shown in Figure 5.4. Note the symbols at the plane origin, corresponding to zero magnitude both in the in-phase and the quadrature parts of the signal. This is caused by initial system delay, and as such these received symbols do not correspond to actually transmitted symbols. They can therefore not be equalized, and will remain zero after the equalization process has been completed.

Comparing Figures 5.3 and 5.4 highlights the effect the transmission has on the symbol's quality. The constellation is clearly distorted, with phase offset having been added to the symbols resulting in a more spread out, indistinct constellation at the receiver input point.

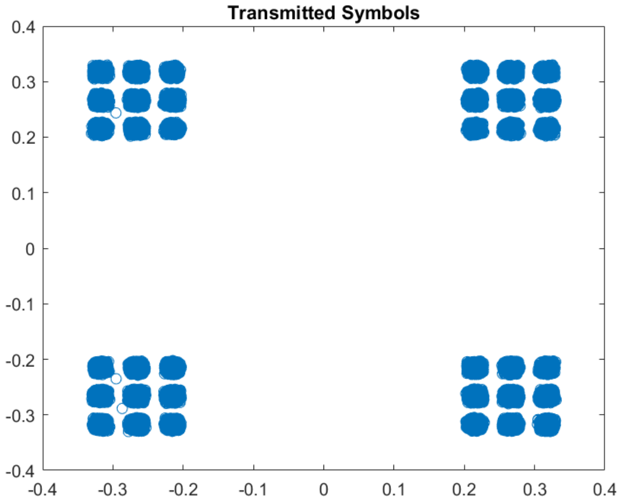


Figure 5.3: Transmitted Constellation

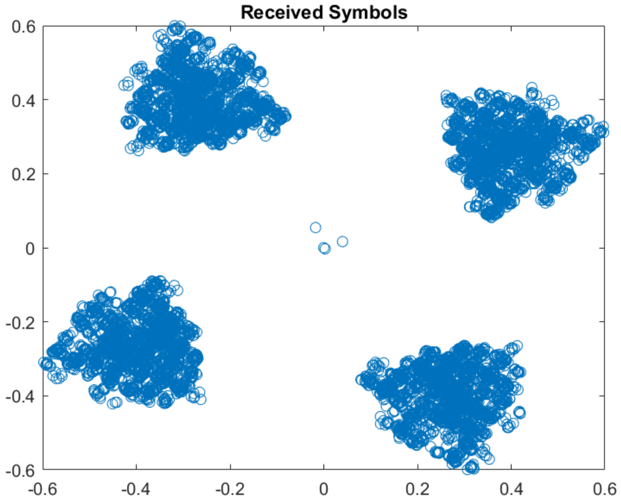


Figure 5.4: Received Constellation

This signal is in turn fed into a Low Pass Lumped Parameter Filter like the one described in Section 4.1. The SystemVue block diagram representing this filter can be seen in Figure 5.5. This block has seven outputs, each one corresponding to a different tap, and providing the system with a voltage representing one of the seven states making up the state vector $\mathbf{x}$. Each element of the state vector is fed into a `Data_Sink` block to be saved and processed offline, in the Matlab implementation of the LMS algorithm. It also has two inputs, each one corresponding to one of the two ladders of shunt capacitors and series inductors that make up the filter, which has already been shown in Figure 4.3. The S-parameters for this filter have been obtained by simulating it in Keysight's Advanced Design System environment, and then used by SystemVue to generate an S-parameter block.

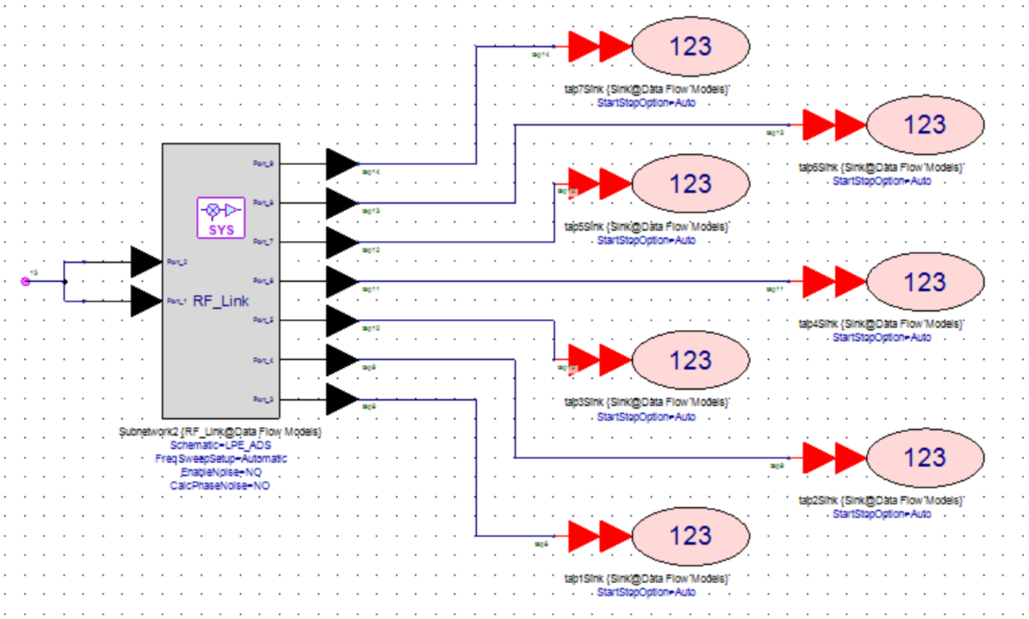

123
tap7Sink {Sink@Data Flow Models}
StartStopOption=Auto
123
tap6Sink {Sink@Data Flow Models}
StartStopOption=Auto
123
tap5Sink {Sink@Data Flow Models}
StartStopOption=Auto
SYS
RF_Link
123
tap4Sink {Sink@Data Flow Models}
StartStopOption=Auto
123
tap3Sink {Sink@Data Flow Models}
StartStopOption=Auto
Subnetwork2 {RF_Link@Data Flow Models}
Schematic=LPE_ADS
FreqSweepSetup=Automatic
EnableNoise=NO
CalcPhaseNoise=NO
123
tap2Sink {Sink@Data Flow Models}
StartStopOption=Auto
123
tap1Sink {Sink@Data Flow Models}
StartStopOption=Auto

Figure 5.5: SystemVue Filter and Data Sink Blocks

5.2 Algorithm Implementation

The adaptive algorithm utilized for the realization of the equalization process is implemented through a Matlab script that calculates the original weight vector and then updates it in a fashion that minimizes (or at least attempts to minimize) the equalization error. As previously discussed, this error is calculated by comparing the equalized symbol with the ideal equalizer output. This reference signal is generated by a decision system that behaves as a quantizer: it quantizes the in-phase and quadrature signals separately, towards two output levels, with the deciding factor being the sign of these signals. In other words, it is a two-bit quantizer with a decision threshold at zero. The simple function that implements this decision system is shown in Listing 5.1.

```
    function dec = decision(x,scale)
        dec = (sign(real(x))+1i*sign(imag(x)))*scale;
    end
```

Listing 5.1: Decision System Code

The first step in the realization of the algorithm is the initialization of its parameters. In our case, this includes the declaration of variables such as number of taps, algorithm step size and error tolerance, as well as reading the file of the received signal and sampling it at the frequency declared in the SystemVue simulations. For the easier calculation of the error signal and the implementation of the decision system shown in the above code, we have included a `scale` parameter which represents the magnitude of the received signal in the code.

```
    L = 7; %number of taps
    mu = 0.1; %algorithm step size
    tol = 0.1; %error tolerance

    load('receivedSink.mat')
    r = receivedSink;
    r_sym = r(1:20:length(r));
    scale = max(abs(r_sym));
    N = length(r_sym); %signal length in number of symbols
    K = N - L + 1;
    R = zeros(L,K);
    for i = 1:K
        R(:,i) = r_sym(i+L-1:-1:i).';
    end
```

Listing 5.2: LMS Initialization

The next step in the process is the initialization of the state vector. To do this, first we must read the files where the results of the SystemVue simulation previously discussed are saved, and sample the signals provided therein in the same manner as with the received signal. We then perform an initial amplification of the state vector, as seen in line 24. This is done in the Matlab code and not the SystemVue simulation to speed up the simulation process as much as possible, even if by avoiding the negligible delay caused by a single block in the signal path. The code that performs these operations is presented below, in Listing 5.3.

```
    load("tap1Sink.mat")
    t1_sym = tap1Sink(1:20:length(tap1Sink));

    load("tap2Sink.mat")
    t2_sym = tap2Sink(1:20:length(tap2Sink));

    load("tap3Sink.mat")
    t3_sym = tap3Sink(1:20:length(tap3Sink));

    load("tap4Sink.mat")
    t4_sym = tap4Sink(1:20:length(tap4Sink));

    load("tap5Sink.mat")
    t5_sym = tap5Sink(1:20:length(tap5Sink));

    load("tap6Sink.mat")
    t6_sym = tap6Sink(1:20:length(tap6Sink));

    load("tap7Sink.mat")
    t7_sym = tap7Sink(1:20:length(tap7Sink));

    x = [t1_sym.'; t2_sym.'; t3_sym.'; t4_sym.'; t5_sym.'; t6_sym.'; t7_sym.'];

    x = 10*x;
```

Listing 5.3: State Vector Initialization

The next step to be considered is the initialization of the weight vector. As discussed earlier in Chapter 3, it was deemed preferable to estimate a weight vector that compensates for the nominal channel behaviour and use that as the initial equalizer state, than to arbitrarily choose a random initial weight vector. Following the analysis presented in that Chapter we arrived at Equation 3.46, which we state again here for the convenience of the reader:

$$\mathbf{w}_{\text{opt}} = \mathbf{A}^{-1}\mathbf{C}^{T}\mathbf{h}_{\delta} \tag{5.1}$$

We use the S-parameters of the PMF and the impulse responses of the filter that we received from simulating it in Keysight's Advanced Design System to calculate this initial weight vector. By performing the inverse FFT on the PMF frequency response described by its S_{21} parameter, we receive the time domain impulse response `Ch_t`. `SNR` refers to the signal to noise ratio, which we arbitrarily place at 20dB. The parameters `tap_length` and `sig_length` are self explanatory. Following the symbolism used in Chapter 3, we denote the impulse response of the i-th filter tap with `r(i,:)`. The code performing this calculation can be seen below.

```
    % noise expected value calc
    for i=1:N
        for j=1:N
            temp = cumtrapz(r(i,:).*r(j,:));
            M(i,j) = 0.5*10^(-0.1*SNR)*temp(tap_length);
        end
    end

    % eq response
    for i=1:N
        c(i,:) = conv(Ch_t,r(i,:));
        for j=1:L+1
            if j*T_b + t_0 <= sig_length
                C(j,i) = c(i, j*T_b + t_0);
            else
                C(j,i) = 0;
            end
        end
    end

    A = C'*C + M;
    w_init = (A\(C'))*h;
```

Listing 5.4: Initial Weight Vector Calculation

The last step is the algorithmic implementation of the adaptation process itself, which can be seen in the following code excerpt.

```
w = zeros(L,N);
w(:,1) = w_init;
e = zeros(1,N);
y = zeros(1,N);
y_dec = zeros(1,N);
corr = zeros(L,N);

for i = 1:L
    y(i) = w(:,i).'*x(:,i);
    y_dec(i) = decision(y(i),scale);
    e(i) = y(i) - y_dec(i);
end

for i = L+1:N
    y(i) = w(:,i-1).'*x(:,i);
    y_dec(i) = decision(y(i),scale);
    e(i) = y(i) - y_dec(i);
    corr(:,i) = 2*mu*e(i)*R(:,i-L+1);
    if real(y_dec(i)) == 0 || imag(y_dec(i)) == 0
        w(:,i) = w_init;
    elseif abs(e(i)) > tol
        w(:,i) = w(:,i-1) - corr(:,i);
        for j=1:L
            if abs(real(w(j,i))) > 2
                w(j,i) = sign(real(w(j,i)))+1i*imag(w(j,i));
            end
            if abs(imag(w(j,i))) > 2
                w(j,i) = real(w(j,i))+1i*sign(imag(w(j,i)));
            end
        end
    else
        w(:,i) = w(:,i-1);
    end
end
```

Listing 5.5: Equalization

It can be seen that we have included several clauses in the adaptation loop. Firstly, we have included a condition that activates when the real or the imaginary part of the equalized symbol is zero. This case corresponds to the initial samples that, due to the channel delay, are zero. While the current sample falls within this category, the equalizer basically maintains its initial state by keeping the weight vector to the value initially calculated for the compensation of the nominal channel response.

The second clause corresponds to the case where the equalization error is high enough that it can't be tolerated, and as such the weight vector needs to be adapted. To avoid the possibility of the error and weight entering a positive feedback loop, where high error values lead to higher weight vector element values, which in turn lead to even higher error values, pushing the algorithm towards divergence, we have included a clause that causes the weight vector element values to be saturated past a certain point. This is not an arbitrary measure that doesn't correspond to real operating conditions, as the weight that can be provided by the DAC discussed previously in this Chapter is also bounded. Since the weight vector values are complex numbers, with the real and imaginary part being provided by different DAC's, we have included this clause for both parts of the weight.

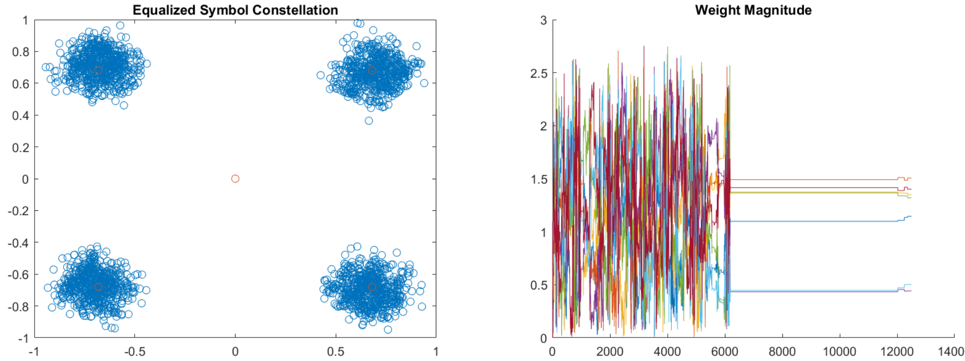
The results of the equalization process can be seen in the following figures. Figure 5.6 presents the constellation diagram that results from the process previously described, as well as the behaviour of the weight vector. For the convenience of the reader, as well as the sake of clarity when making comparisons between the received and equalized symbol constellation, we have included a comparison of the two diagrams in Figure 5.8. It is plainly visible that the equalizer has corrected at least part of the undesirable effects evident in the received constellation. The phase offset has been almost completely compensated, and the spread in symbols has been reduced.

This intuitive understanding of the constellation's improvement can be quantized through the use of the Error Vector Magnitude (EVM) metric. This metric evaluates a constellation's quality by comparing the magnitude of the error vector to the magnitude of the reference symbol through a percentage. We have calculated this percentage by first calculating the Root Mean Square (RMS) value of the error vector and the reference symbols, and then calculating their ratios, as shown in the following equation:

$$\mathrm{EVM} = \frac{|e|}{|\mathit{ref}|} * 100 \tag{5.2}$$

Using this expression, we calculated the EVM of the constellation before and after equalization. The EVM of the received signal is at 64.62%, while that of the equalizer output is at 16.99%. We can see that we have achieved a quality improvement of almost 50%.

From Figure 5.6 it can be seen that once the error threshold is reached, the adaptation process freezes, and the weight vector stabilises in a state that corresponds to a symbol constellation whose improvement over the originally received one we have just discussed. The behaviour of the error signal is shown in Figure 5.7. This plot highlights the reason that makes the inclusion of the acceptable error threshold and maximum weight value not just an improvement to the equalizer's behaviour, but an actual necessity. It can be seen that in the early samples the error signal sometimes approaches a low, almost acceptable value, but quickly diverts from the state where the equalization error would be minimal, presenting significant spikes in value. Once the algorithm has achieved a low error state, however, and the adaptation is not performed for every sample received, this behaviour completely stops, and we face no further danger of going into a divergent loop, where high error values lead to higher weight values, which in turn lead to even higher error values.



(a) Equalized Constellation Diagram

(b) Weight Vector

Figure 5.6: Equalization Results

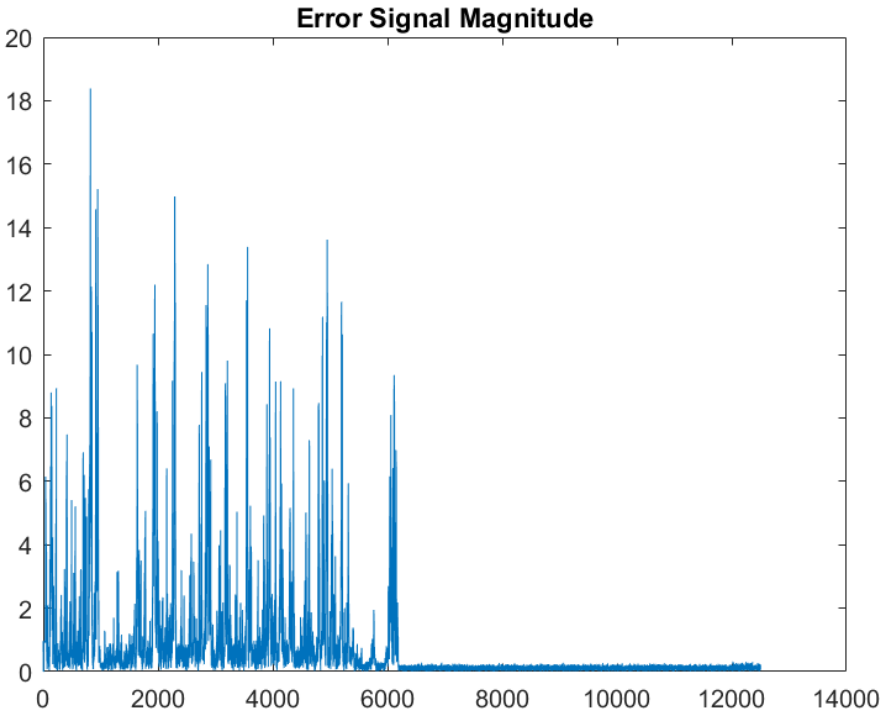
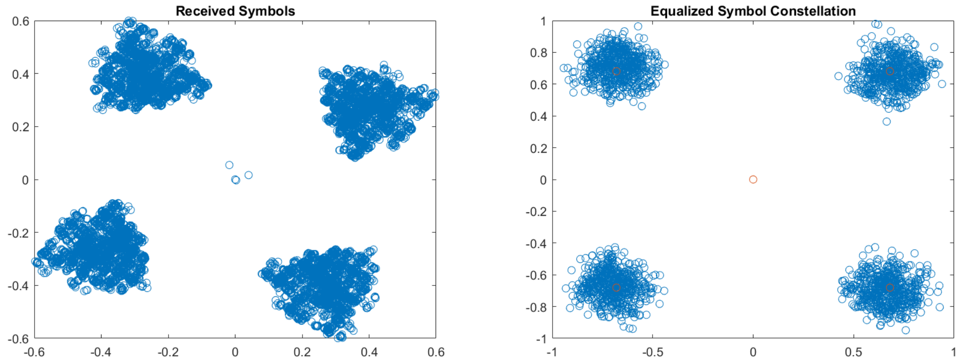


Figure 5.7: Equalization Error



(a) Received Constellation Diagram (b) Equalized Constellation Diagram

Figure 5.8: Constellation Diagram Comparison

Chapter 6

Future Work

While the purpose of developing a complete equalizer system, which was the aim of this work, has been completed, there are still avenues for further development. One task that was not accomplished, for example, was the physical implementation of the digital logic of the LMS algorithm. This could be achieved through the use of a Field Programmable Gate Array (FPGA), a customizable digital system that can realize most digital logic circuits. This would

Another idea that could still be pursued would be the integration of the two algorithms we provided as solutions for the problem of equalization, the CMA and the LMS algorithm. A possible implementation would be to compensate the modulus of the constellation through an analog system realizing the Constant Modulus Algorithm, and then compensating for the phase offset through a digital implementation of the LMS algorithm. This combination of solutions might provide us with the advantage of lessening the load for the digital part of the system, since it would only have to compensate for one of the effects seen on the received constellation. This, in turn, could result in lower power dissipation for the system.

A third and final possible step to be taken after the work presented here would be the completion of layouts for the IC's developed within the context of this work, so that these circuits could be printed on silicon chips and measured in a lab setting. This would allow for a final, comprehensive evaluation of the system, providing realistic data with no margin for simulation errors.

Bibliography

[1] Panteleimon Gavalas, Vasileios Manouras, and Ioannis Papananos. "A 130–160 GHz Frequency Tripler in a 130nm SiGe BiCMOS Technology". In: *2022 Austrochip Workshop on Microelectronics (Austrochip)*. 2022, pp. 17–20. DOI: 10.1109/Austrochip56145.2022.9940746.

[2] Filippa Soumpasakou, Vasileios Manouras, and Ioannis Papananos. "Design and Implementation of a D-Band I/Q Modulator in a 130 nm SiGe BiCMOS Technology". In: *2022 Panhellenic Conference on Electronics & Telecommunications (PACET)*. 2022, pp. 1–4. DOI: 10.1109/PACET56979.2022.9976371.

[3] Vasileios Manouras and Ioannis Papananos. "A Ka-Band Quasi-F-1 Power Amplifier in a 130 nm SiGe BiCMOS Technology". In: *2022 Panhellenic Conference on Electronics & Telecommunications (PACET)*. 2022, pp. 1–5. DOI: 10.1109/PACET56979.2022.9976366.

[4] Jin Liu and Xiaofeng Lin. "Equalization in high-speed communication systems". In: *IEEE Circuits and Systems Magazine* 4.2 (2004), pp. 4–17. DOI: 10.1109/MCAS.2004.1330746.

[5] Cecilia Gimeno Gasca, Santiago Celma Pueyo, and Concepcion Aldea Chagoyen. *CMOS Continuous-Time Adaptive Equalizers for High-Speed Serial Links*. Springer, 2015.

[6] Patrick Reynaert et al. "Polymer Microwave Fiber: a New Communication Concept That Blends Wireless, Wireline and Optical Communication". In: *2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. 2019, pp. 755–758. DOI: 10.1109/ICECS46596.2019.8964776.

[7] A. Fiedler et al. “A 1.0625 Gbps transceiver with 2x-oversampling and transmit signal pre-emphasis”. In: *1997 IEEE International Solids-State Circuits Conference. Digest of Technical Papers*. 1997, pp. 238–239. DOI: 10.1109/ISSCC.1997.585369.

[8] D.J. Foley and M.P. Flynn. “A low-power 8-PAM serial transceiver in 0.5-/spl mu/m digital CMOS”. In: *IEEE Journal of Solid-State Circuits* 37.3 (2002), pp. 310–316. DOI: 10.1109/4.987082.

[9] Paulo R. S. Diniz. *Adaptive Filtering*. Springer, 2013.

[10] D. Godard. “Self-Recovering Equalization and Carrier Tracking in Two-Dimensional Data Communication Systems”. In: *IEEE Transactions on Communications* 28.11 (1980), pp. 1867–1875. DOI: 10.1109/TCOM.1980.1094608.

[11] Min Wu and F. Cornett. “Discrete-time and continuous-time constant modulus algorithm analysis”. In: *Proceedings of the Twenty-Seventh Southeastern Symposium on System Theory*. 1995, pp. 504–508. DOI: 10.1109/SSST.1995.390530.

[12] Ye Li and Zhi Ding. “Global convergence of fractionally spaced Godard (CMA) adaptive equalizers”. In: *IEEE Transactions on Signal Processing* 44.4 (1996), pp. 818–826. DOI: 10.1109/78.492535.

[13] Shanthi Pavan. “Power and area efficient high speed analog adaptive equalization”. In: *2008 IEEE International Symposium on Circuits and Systems*. 2008, pp. 3126–3129. DOI: 10.1109/ISCAS.2008.4542120.

[14] K. L. Su. *Analog Filters. Second Edition*. Kluwer Academic Publishers, 2003.

[15] Behzad Razavi. “The Current-Steering DAC [A Circuit for All Seasons]”. In: *IEEE Solid-State Circuits Magazine* 10.1 (2018), pp. 11–15. DOI: 10.1109/MSSC.2017.2771102.