



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ενεργός Έλεγχος Θορύβου (Active Noise Control)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μανώλης Χ. Τσακίρης

Επιβλέπων : Γεώργιος Ε. Καμπουράκης

Λέκτορας

Αθήνα, Νοέμβριος 2007



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ενεργός Έλεγχος Θορύβου (Active Noise Control)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μανώλης Χ. Τσακίρης

Επιβλέπων : Γεώργιος Ε. Καμπουράκης

Λέκτορας

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 12^η Νοεμβρίου 2007.

.....

Γεώργιος Καμπουράκης
Μέλος Δ.Ε.Π

.....

Βασίλειος Λούμος
Μέλος Δ.Ε.Π

.....

Ελευθέριος Καγιάφας
Μέλος Δ.Ε.Π

Αθήνα, Νοέμβριος 2007

.....

Μανώλης Χ. Τσακίρης

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

Copyright © Μανώλης Τσακίρης

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

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 εργασία αυτή μελετάται με μεγάλη λεπτομέρεια ένα σύστημα ενεργού ελέγχου θορύβου με ανάδραση. Η ανάπτυξη του θέματος αρχίζει με την αρχή λειτουργίας και την αναγκαιότητα του ενεργού ελέγχου θορύβου. Στη συνέχεια εξετάζεται θεωρητικά η βασική διάταξη ενεργού ελέγχου θορύβου, στην οποία επιχειρείται η εξασθένιση του θορύβου, τον οποίο παράγει μία πηγή κύριου θορύβου, με τη χρήση ενός μικροφώνου, ενός ψηφιακού επεξεργαστή σήματος, ενός ενισχυτή ισχύος και ενός ηχείου. Η όλη μελέτη επικεντρώνεται στις λεπτομέρειες του αλγορίθμου παραγωγής αντιθορύβου. Ο αλγόριθμος παραγωγής αντιθορύβου θεμελιώνεται από βασικές αρχές, ξεκινώντας από τη θεωρία βέλτιστης γραμμικής πρόβλεψης και καταλήγοντας στα LMS προσαρμοζόμενα φίλτρα. Ιδιαίτερη έμφαση δίνεται στην ύπαρξη του λεγόμενου Secondary Path και προτείνεται μία τεχνική αντιστάθμισής του. Ο αλγόριθμος, ο οποίος προκύπτει είναι ο *Leaky FULMS Feedback Active Noise Control Algorithm with Off-Line Secondary Path Modelling*. Αυτός είναι ο θεμελιώδης αλγόριθμος των συστημάτων ενεργού ελέγχου θορύβου. Τέλος ο αλγόριθμος υλοποιείται στον επεξεργαστή ψηφιακού σήματος ADSP-21065L της Analog Devices και πραγματοποιούνται πειράματα πραγματικού χρόνου με τη διάταξη, η οποία προαναφέρθηκε. Η εξασθένιση σε πραγματικό χρόνο φθάνει τα 20dB @ 200Hz. Η παρουσίαση συνοδεύεται από αρχεία ήχου, γραφικές παραστάσεις και αρχεία μοντέλων Simulink/MATLAB.

Λέξεις κλειδιά: ενεργός έλεγχος θορύβου, κύριος θόρυβος, αντιθόρυβος, εναπομένων θόρυβος, προβλέπτες, προσαρμοζόμενα φίλτρα, αλγόριθμος LMS, αλγόριθμος FUXLMS/FXLMS, αλγόριθμος Leaky-LMS, αλγόριθμος Leaky-FULMS/FXLMS, Secondary Path, chirp σήματα, προσαρμοζόμενη αναγνώριση συστήματος, ADSP-21065L.

Summary

In the present work we study in detail a feedback active noise control system. The subject is treated starting from the principles of operation and the necessity of active noise control. Next, the basic active noise control set-up is examined, where an attempt is made to attenuate the noise, coming from a primary noise source, using a single microphone, a digital signal processor, a power amplifier and a single loudspeaker. The treatment focuses on the details of the antinoise generation algorithm. The antinoise generation algorithm is studied from first principles, starting from the theory of optimal linear prediction up to the development of the LMS adaptive filters. Special attention is drawn to the so-called Secondary Path and a technique for its compensation is proposed. The final algorithm is the *Leaky FULMS Feedback Active Noise Control Algorithm with Off-Line Secondary Path Modelling*. This is the fundamental algorithm of active noise control systems. Finally, this algorithm is implemented on the digital signal processor ADSP-21065L of Analog Devices and real time experiments are conducted using the beforementioned set-up. The real time noise attenuation reaches a maximum of 20dB @ 200Hz. The presentation is accompanied by audio files, graphical shapes and files of models Simulink/MATLAB.

Keywords: *active noise control, primary noise, antinoise, residual noise, predictors, adaptive filters, LMS algorithm, FULMS/FXLMS algorithm, Leaky-LMS algorithm, Leaky-FULMS/FXLMS algorithm, Secondary Path, chirp signals, adaptive system identification, ADSP-21065L.*

Ενεργός Έλεγχος Θορύβου

Περιεχόμενα

1 Εισαγωγή.....	8
1.1 Γενική εισαγωγή.....	9
1.2 Σκοπός της παρούσας εργασίας	10
1.3 Τεχνική εισαγωγή.....	10
2 Συστήματα Ενεργού Ελέγχου Θορύβου.....	12
2.1 Θόρυβος Ευρείας Ζώνης και Θόρυβος Στενής Ζώνης	12
2.2 Feedforward και Feedback Ενεργός Έλεγχος Θορύβου	14
2.3 Ο απλοποιημένος LMS feedback ANC ελεγκτής ενεργού θορύβου	15
2.4 Ο LMS Feedback Ελεγκτής Ενεργού Θορύβου (με Secondary Path)	24
2.5 Ο FULMS Feedback Ελεγκτής Ενεργού Θορύβου	31
2.6 Ο βελτιωμένος FULMS Feedback Ελεγκτής Ενεργού Θορύβου (με Secondary Path)	37
2.7 Προσαρμοζόμενη Αναγνώριση του Secondary Path	39
2.8 Διέγερση του Secondary Path χρησιμοποιώντας Chirp Σήματα	43
2.9 Ο FULMS Feedback Ελεγκτής Ενεργού Θορύβου με off-line Αναγνώριση του Secondary Path	50
2.10 Ο Leaky LMS αλγόριθμος	53
2.11 Ο Leaky FULMS Feedback Ελεγκτής Ενεργού Θορύβου με Off-Line Αναγνώριση του Secondary Path.....	56
3 Υλοποίηση του Ελεγκτή Ενεργού Θορύβου με τον Ψηφιακό Επεξεργαστή Σήματος 21065L της Analog Devices.....	57
3.1 Εισαγωγή.....	57
3.2 Ο Ελεγκτής Ενεργού Θορύβου Πραγματικού Χρόνου: Πειράματα και Αποτελέσματα	57
3.2.1 Η Ηλεκτροακουστική Πειραματική Διάταξη	57
3.2.2 Κύριος θόρυβος 200Hz	58
3.2.3 Κύριος θόρυβος 250Hz	58
3.2.4 Κύριος θόρυβος 300Hz	59
3.2.5 Κύριος θόρυβος 350Hz	59
3.2.6 Κύριος θόρυβος 400Hz	60
3.2.7 Κύριος θόρυβος “s.o.t.w.”	60
4 Παράρτημα	62
4.1 Αβεβαιότητα ως προς τη συχνότητα κατά την μετατροπή ηλεκτρονικών σημάτων σε ακουστικά (πείραμα πραγματικού χρόνου)	62
4.2 Προσαρμοζόμενα Φίλτρα	67
4.2.1 Εισαγωγή στα Προσαρμοζόμενα Φίλτρα	67
4.2.2 Εισαγωγή στη Θεωρία Πρόβλεψης.....	67
4.2.3 Στοιχεία Βέλτιστης Γραμμικής Πρόβλεψης.....	68
4.2.4 Επιλύση των Κανονικών Εξισώσεων	71
4.2.5 Ανάπτυξη του Steepest-Descent αλγορίθμου	73

4.2.6 Σύγκλιση του Steepest-Descent αλγορίθμου	76
4.2.7 Ανάπτυξη του προσαρμοζόμενου αλγορίθμου LMS	79
4.2.8 Η ιδιότητα της ελάχιστης διαταραχής του LMS αλγορίθμου	81
4.2.9 Αναθεωρημένη συνθήκη ευστάθειας για πρακτικές εφαρμογές	83
4.2.10 Σταθερά χρόνου της προσαρμοστικής διαδικασίας	84
4.2.11 Ανάπτυξη του LMS προσαρμοζόμενου φίλτρου	86
4.3 Ο κώδικας του Ελεγκτή Ενεργού Θορύβου σε Assembly του Επεξεργαστή 21065L	88
4.3.1 "AD1819a_Initialization"	88
4.3.2 "Declare_and_Initialize_Data"	99
4.3.3 "Clear_SPT1_regs"	104
4.3.4 "Codec_Processing_ISR"	105
4.3.5 "Init_065L_EZLAB"	111
4.3.6 "SDRAM_Init"	114
4.3.7 "ISR_table"	116
4.3.8 "ANC"	121
4.3.9 "Master_Routine"	126
4.3.10 "Router_Routine"	128
4.3.11 "SPM"	129
4.3.12 "new65Ldefs"	132
4.3.13 "def21065L"	134
5 Βιβλιογραφία	149

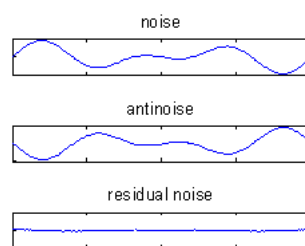
1 Εισαγωγή

1.1 Γενική εισαγωγή

Έχει διαπιστωθεί σε μελέτες ότι η τεχνολογική πρόοδος ακολουθείται από μία σημαντική αύξηση του περιβαλλοντικού θορύβου. Αυτή η αύξηση οφείλεται κυρίως στην ευρεία χρήση βιομηχανικού εξοπλισμού όπως μηχανές, ανεμιστήρες, μετασχηματιστές και συμπιεστές, τόσο στην παραγωγή και τις βιομηχανικές διεργασίες όσο και στα καταναλωτικά προϊόντα. Ο περιβαλλοντικός θόρυβος θεωρείται σημαντική απειλή για την ανθρώπινη υγεία, καθώς είναι υπεύθυνος για την πρόκληση κόπωσης, την έλλειψη συγκέντρωσης, την υποβάθμιση της απόδοσης στην εργασία και γενικά την υποβάθμιση της ποιότητας της ζωής.

Οι παραδοσιακές μέθοδοι αντιμετώπισης του θορύβου βασίζονται στη χρήση παθητικών μέσων, όπως ηχομονωμένων χώρων, σιγαστήρων, ηχοπαγίδων και ηχητικών φραγμάτων, με σκοπό την εξασθένιση της ισχύος του θορύβου. Η απόδοση αυτών των παθητικών τεχνικών εξαρτάται σε μεγάλο βαθμό από το μήκος κύματος του ανεπιθύμητου θορύβου, συγκρινόμενο με τις διαστάσεις του παθητικού μέσου που χρησιμοποιείται. Όταν το μήκος κύματος είναι μικρό, τότε η απόδοση του συστήματος μπορεί να γίνει πολύ υψηλή καθώς το κύμα του θορύβου διαδίδεται κατά μία μεγάλη απόσταση μέσα στο ηχοαπορροφητικό υλικό και τελικά εξασθενεί ικανοποιητικά. Από την άλλη πλευρά, όταν το μήκος κύματος είναι μεγάλο, το κύμα του θορύβου διαδίδεται κατά μία σύντομη απόσταση μέσα στο ηχοαπορροφητικό υλικό, και τελικά μικρή αλλαγή συμβαίνει στο πλάτος του. Σε αυτή την περίπτωση των μεγάλων μηκών κύματος, ή ισοδύναμα των χαμηλών συχνοτήτων, τεράστια ηχομονωτικά ή ηχοαπορροφητικά σώματα πρέπει να χρησιμοποιηθούν, τα οποία είναι κοστοβόρα στην κατασκευή και των οποίων το μέγεθος και το βάρος τα καθιστούν μη πρακτικά σε εφαρμογές, στις οποίες χρειάζονται.

Ο ενεργός έλεγχος θορύβου (Active Noise Control -ANC) είναι μία πρόσφατα αναπτυσσόμενη τεχνολογία, η οποία αντιμετωπίζει αποτελεσματικά το πρόβλημα του χαμηλόσυχνου θορύβου, στις περιπτώσεις όπου τα παθητικά μέσα είτε είναι αναποτελεσματικά είτε είναι πολύ ακριβά ή μη πρακτικά. Ο ANC υλοποιείται μέσω ενός ηλεκτροακουστικού συστήματος το οποίο εξασθενεί το θόρυβο και η φιλοσοφία του βασίζεται στην έννοια της υπέρθεσης κυμάτων και στο φαινόμενο της καταστροφικής συμβολής. Σε ένα ANC σύστημα ένας αριθμός μικροφώνων (αισθητήρες) ανιχνεύουν το θόρυβο του περιβάλλοντος (κύριος θόρυβος). Τα σήματα αυτά οδηγούνται σε έναν μικροεπεξεργαστή, ο οποίος παράγει έναν αντι-θόρυβο και οδηγεί έναν αριθμό ηχείων (ενεργοποιητές). Ο αντι-θόρυβος αποτελείται από συχνότητες ίσου πλάτους και αντίθετης φάσης σε σχέση με τις συχνότητες του κύριου θορύβου. Αυτό που συμβαίνει είναι η υπέρθεση του αντι-θορύβου στον κύριο θόρυβο και στη συνέχεια καταστροφική συμβολή μεταξύ τους. Το αποτέλεσμα είναι ένας εναπομένον θόρυβος σημαντικότερα ασθενέστερης ισχύος, όπως φαίνεται στο σχήμα :



Εικόνα 1.1.1 η ιδέα του ενεργού ελέγχου θορύβου

Φυσικά, η απόδοση ενός ANC συστήματος εξαρτάται από την ακρίβεια του αντι-θορύβου, η οποία με τη σειρά της εξαρτάται από την ποιότητα του αλγορίθμου, τον οποίο τρέχει ο μικροεπεξεργαστής. Επιπλέον καίριας σημασίας είναι οι ακουστικές ρυθμίσεις του συστήματος, όπως για παράδειγμα ο αριθμός των μικροφώνων και των ηχείων που θα χρησιμοποιηθούν, καθώς και η επιλογή τη θέσης τους στο χώρο.

Ο ANC έχει γίνει αντικείμενο μεγάλης προσοχής και εκτεταμένης έρευνας από επιστήμονες, μηχανικούς και τη βιομηχανία. Εμπορικές εφαρμογές έχουν ήδη λάβει χώρα, όπως μείωση του θορύβου των τουρμπινών στο εσωτερικό των αεροπλάνων, ανάπτυξη ακουστικών ήσυχης ζώνης και επιλεκτική μείωση του θορύβου της λεωφόρου στο εσωτερικό των αυτοκινήτων. Τέλος, είναι σημαντικό να υπογραμμισθεί ότι ο ANC είναι μια τεχνολογία, η οποία συμπληρώνει τις παθητικές τεχνικές και δεν τις αντικαθιστά. Αυτό φαίνεται σε πολλές εφαρμογές, στις οποίες χρησιμοποιούνται υβριδικά ενεργά-παθητικά συστήματα απόσβεσης θορύβου, τα οποία ικανοποιούν υψηλές προδιαγραφές απόδοσης, τόσο σε ποσοστό απόσβεσης όσο και σε εύρος συχνοτικού φάσματος.

1.2 Σκοπός της παρούσας εργασίας

Ο σκοπός μας σε αυτήν την εργασία είναι ο εξής:

1) να μελετήσουμε έναν απλό ANC ελεγκτή από το μηδέν

2) να διαφωτίσουμε ορισμένα σημεία, τα οποία είναι σκοτεινά στη βιβλιογραφία, αλλά τεράστιας σημασίας κατά την υλοποίηση του ελεγκτή

3) να υλοποιήσουμε τον ANC ελεγκτή

Κάθε βήμα που θα κάνουμε θα παρουσιάζεται ως η φυσική συνέχεια των προηγούμενων βημάτων. Τα αποτελέσματα των προσομοιώσεων και των πειραμάτων πραγματικού χρόνου θα αναλύονται με τη βοήθεια γραφικών παραστάσεων, αρχείων ήχου και αρχείων MATLAB/SIMULINK έτσι ώστε από την πλευρά του αναγνώστη να επιτυγχάνεται ο μέγιστος βαθμός κατανόησης.

1.3 Τεχνική εισαγωγή

Τα ANC συστήματα, ακόμα και στις πιο απλές τους εκφάνσεις, παρουσιάζουν υψηλή περιπλοκότητα, με αποτέλεσμα οι έννοιες και τα εργαλεία που χρησιμοποιούνται κατά την ανάπτυξή τους να φαίνονται αχρείαστα σύνθετα σε κάποιον μηχανικό, ο οποίος είναι αμήτοος στον ενεργό έλεγχο θορύβου. Προτού λοιπόν εστιάσουμε στις θεμελιώδεις ιδέες και δομές που πλήρως περιγράφουν έναν ANC ελεγκτή, θα υπογραμμίσουμε δύο εγγενή χαρακτηριστικά των ANC συστημάτων, τα οποία εν μέρει δικαιολογούν την ανάγκη μίας προχωρημένης προσέγγισης επί του θέματος.

Πρώτον, για να είναι ένα ANC σύστημα αποδοτικό, οφείλει να είναι **σύστημα πραγματικού χρόνου**. Ο θόρυβος πρέπει να δειγματοληπτείται σε πραγματικό χρόνο (με συχνότητα δειγματοληψίας που ικανοποιεί το θεώρημα του Shannon) και για κάθε αποκτηθέν δείγμα, ολόκληρος ο ANC αλγόριθμος πρέπει να εκτελεστεί προκειμένου να παράγει ένα δείγμα αντι-θορύβου, με το οποίο θα οδηγήσει τα ηχεία, προτού το επόμενο δείγμα θορύβου αφιχθεί. Αυτή η διαδικασία επαναλαμβάνεται έως ότου τερματίσουμε τη λειτουργία του συστήματος. Τα παραπάνω συνεπάγονται ότι hardware πραγματικού χρόνου πρέπει να χρησιμοποιηθεί και συνεπώς η ANC υλοποίηση δεν είναι εφικτή με χρήση συμβατικών μέσων όπως ένα PC που τρέχει το MATLAB ή κάποιο κώδικα C++. Το λειτουργικό σύστημα, οι οδηγοί της κάρτας ήχου και τα διάφορα στρώματα λογισμικού που χωρίζουν την αναλογική είσοδο από τον ANC αλγόριθμο συνήθως καθιστούν ένα συμβατικό PC πολύ αργό για τις απαιτήσεις της ANC λειτουργίας. Έτσι, ειδικοί επεξεργαστές ψηφιακού σήματος πρέπει να χρησιμοποιηθούν, στους οποίους ο ANC αλγόριθμος «βλέπει» κατευθείαν ολόκληρο το αναλογικό interface χωρίς την ανάγκη διαμεσολάβησης κάποιου άλλου στρώματος λογισμικού.

Δεύτερον, ένα ANC σύστημα οφείλει να είναι ένα **στοχαστικό σύστημα**. Για να καταδείξουμε την ανάγκη αυτή, θεωρούμε την ακόλουθη περίπτωση:

Υποθέστε ότι θέλουμε να αποσβέσουμε ένα θόρυβο συχνότητας **200Hz**. Έστω ότι ο κύριος αυτός θόρυβος είναι πάντοτε σταθερός σε πλάτος και συχνότητα. Επιδιώκουμε να τον εξασθενήσουμε με τη βοήθεια ενός ANC ελεγκτή, ο οποίος τρέχει έναν **ντετερμινιστικό** αλγόριθμο. Ο αλγόριθμος αυτός θα μπορούσε να δομείται ως εξής:

1. Εκτέλεση Fourier ανάλυσης στο σήμα που προέρχεται από το μικρόφωνο και ανίχνευση της συχνότητας **f** του κύριου θορύβου και του πλάτους του **A**.

2. Έναρξη αναπαραγωγής δειγμάτων συχνότητας **f** και πλάτους **A**. Αυτός είναι ο αντιθόρυβος και έχει μία τυχαία διαφορά φάσης με τον κύριο θόρυβο.

3. Κατάλληλη καθυστέρηση στην εκπομπή των δειγμάτων του αντι-θορύβου έως ότου η ενέργεια του σήματος από το μικρόφωνο ελαχιστοποιηθεί. Σε αυτό το στάδιο ο αντι-θόρυβος θα έχει μία διαφορά φάσης με τον κύριο θόρυβο περίπου ίση με 180 μοίρες, με αποτέλεσμα η καταστροφική συμβολή να μειώνει σημαντικά την αρχική ισχύ του θορύβου.

4. Συνεχής αναπαραγωγή και εκπομπή των δειγμάτων που προσδιορίστηκαν στο βήμα 2, λαμβάνοντας υπόψιν την καθυστέρηση που προσδιορίστηκε στο βήμα 3.

Εκ πρώτης όψεως αναμένουμε ότι ο αλγόριθμος αυτός θα λειτουργήσει αρκετά ικανοποιητικά. Δυστυχώς στην πράξη ο αλγόριθμος αυτός θα αποτύχει πλήρως. Δίνουμε δύο καλούς λόγους για αυτό το επιχείρημά μας:

Πρώτος Λόγος: Στο βήμα 1 πραγματοποιούμε Fourier ανάλυση. Μία συχνότητα $f_{detected}$ θα ανιχνευθεί. Αλλά είναι γνωστό ότι η διακριτή Fourier ανάλυση έχει περιορισμένη διακριτική δυνατότητα, το οποίο σημαίνει ότι δεν είναι δυνατόν να ανιχνευθούν όλες οι συχνότητες. Δηλαδή, όταν εκτελούμε DFT ενός σήματος N δειγμάτων, δειγματοληπτημένου με συχνότητα F_s , το προκύπτον διακριτό φάσμα θα αποτελείται μόνο από τις συχνότητες

$$f_{DFT} = m * \frac{F_s}{N}, m = 0, 1, 2, \dots, N - 1, \quad (1.3.1)$$

Η εξίσωση (1.3.1) είναι η ανάλυση (ή η διακριτικότητα) του DFT.

Εάν αυξήσουμε τον αριθμό των δειγμάτων N , τα οποία χρησιμοποιούνται στον DFT μετασχηματισμό, η ανάλυση βελτιώνεται, αλλά και πάλι δεν μπορούν να ανιχνευθούν όλες οι συχνότητες. Τώρα, εκτός εάν η συχνότητα του κύριου θορύβου ταυτίζεται με κάποιο DFT bean, θα συμβεί DFT διαρροή στα beans που βρίσκονται κοντά στη συχνότητα του κυρίου θορύβου. Το αποτέλεσμα είναι ότι ο αλγόριθμός μας θα ανιχνεύσει περισσότερες της μιας συχνότητες κύριου θορύβου, γεγονός, το οποίο δεν ισχύει, και στη συνέχεια θα επιχειρήσει να τις αποσβέσει! Συνεπώς, η μόνη περίπτωση κατά την οποία ο DFT μετασχηματισμός ανιχνεύει με 100% ακρίβεια τη συχνότητα του θορύβου είναι όταν

$$\exists m \in \mathbb{N}: 0 < m < N - 1 \text{ και } f_{primary\ noise} = m * \frac{F_s}{N}, \quad (1.3.2)$$

Σε κάθε άλλη περίπτωση

$$f_{detected} \neq f_{primary\ noise}, \quad (1.3.3)$$

Δεύτερος Λόγος: Ακόμα και εάν η (1.3.2) ισχύει, ενδεχόμενο πολύ σπάνιο να συμβεί, δηλαδή ακόμα και εάν ο DFT ανιχνεύσει με 100% ακρίβεια τη συχνότητα του κυρίου θορύβου, ο μικροεπεξεργαστής θα δημιουργήσει δείγματα της σωστής συχνότητας, αλλά το ηχείο θα παραμορφώσει ελαφρώς τη συχνότητα αυτή, όπως καταδεικνύεται στο **κεφάλαιο 4.1 του παραρτήματος**. Συνεπώς

$$f_{antinoise} \neq f_{primary\ noise}, \quad (1.3.4)$$

Ως αποτέλεσμα των δύο παραπάνω λόγων, η (1.3.4) θα είναι κατά κανόνα πάντοτε αληθής και έτσι ακόμα και εάν σε κάποιο στάδιο εκτέλεσης του αλγορίθμου η διαφορά φάσης μεταξύ κύριου θορύβου και αντι-θορύβου είναι ακριβώς 180 μοίρες, ενώ τα πλάτη τους είναι ίσα, ο εναπομένον θόρυβος προκύπτει από την (4.1.9) με $k = 1$ και $\varphi = 180^\circ$ ίσος με $2\sin(\frac{\omega_1 - \omega_2}{2}t)$! Το αποτέλεσμα αυτό είναι φυσικά μη αποδεκτό, καθώς ανάμεσα στα άλλα το πλάτος του εναπομένοντος θορύβου είναι διπλάσιο από το πλάτος του αρχικού και συνεπώς η κατάσταση χειρότερη!

Οι παραπάνω αβεβαιότητες, ειδικά η αβεβαιότητα που εισάγεται από τον DFT, μπορούν να αποφευχθούν εάν ο ANC ελεγκτής είναι καθολικά αναλογικός. Ένας τέτοιος ελεγκτής μπορεί να είναι ένας προσεκτικά σχεδιασμένος ενισχυτής με αναστρέφουσα απόκριση φάσης. Τέτοια συστήματα έχουν προταθεί και έχουν υλοποιηθεί με σχετική επιτυχία. Παρόλα αυτά, η κατασκευή τους είναι πολύ ακριβή ενώ υποφέρουν από αστάθειες. Για παράδειγμα, εάν η απόσταση μεταξύ του μικροφώνου και του ηχείου αυξηθεί, είναι δυνατόν να συμβεί θετική ανάδραση, με αποτέλεσμα το σύστημα να ταλαντώνεται και ο εναπομένοντας θόρυβος να αυξάνεται.

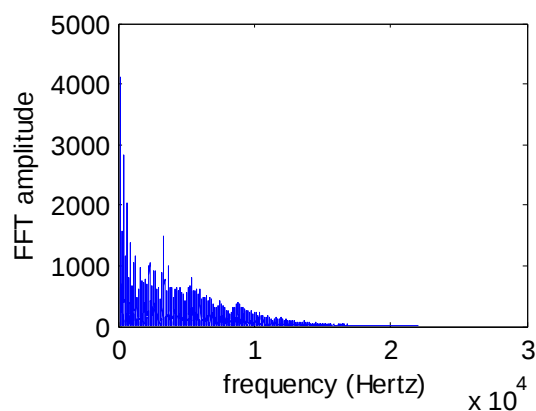
Μέχρι στιγμής είδαμε ότι η διαδικασία ανίχνευσης μίας συχνότητας και η αναπαραγωγή της ως ακουστικό σήμα εισάγει μία αβεβαιότητα ((1.3.3) και (1.3.4)), η οποία δεν πρέπει να αγνοηθεί, ανεξαρτήτως του μεγέθους της. Με απλά λόγια, δεν είμαστε σε θέση να προσδιορίσουμε με 100% ακρίβεια τη συχνότητα του κύριου θορύβου και επίσης δεν είμαστε σε θέση να προσδιορίσουμε με 100% ακρίβεια τη συχνότητα του εκπεμπόμενου αντι-θορύβου. Όσον αφορά την εξασθένηση θορύβου, τέτοιου είδους αβεβαιότητες καθιστούν οποιονδήποτε ντετερμινιστικό αλγόριθμο άχρηστο. Η αβεβαιότητα αυτή στην είσοδο και στην έξοδο ενός ANC ελεγκτή τον καθιστά στοχαστικό σύστημα. Συνεπώς, προκειμένου να αντιμετωπισθούν οι συνέπειες αυτής της αβεβαιότητας θα πρέπει η ίδια η συμπεριφορά του ANC ελεγκτή να είναι στοχαστική. Μόνο με αυτόν τον τρόπο είναι δυνατόν να αντισταθμιστούν οι συνέπειες των αβεβαιοτήτων που αναλύθηκαν. Συνεπώς οι συμβατικές τεχνικές επεξεργασίας σημάτων δεν επαρκούν και περισσότερο προχωρημένα εργαλεία απαιτούνται. Τα προχωρημένα αυτά εργαλεία που χρησιμοποιούνται κατά κόρον στον ενεργό έλεγχο θορύβου είναι τα προσαρμοζόμενα φίλτρα (adaptive filters). Σε αυτό το σημείο συνιστάται ο αναγνώστης να μελετήσει το κεφάλαιο 4.2 του παραρτήματος, πρώτου συνεχίσει με τη μελέτη του επόμενου κεφαλαίου. Στο κεφάλαιο 4.2 αναλύεται η έννοια των προσαρμοζόμενων φίλτρων και αναπτύσσεται ο θεμελιώδης προσαρμοστικός αλγόριθμος LMS.

2 Συστήματα Ενεργού Ελέγχου Θορύβου

2.1 Θόρυβος Ευρείας Ζώνης και Θόρυβος Στενής Ζώνης

Δύο είδη θορύβου υπάρχουν στο περιβάλλον: ευρείας ζώνης και στενής ζώνης. Ο θόρυβος ευρείας ζώνης έχει κατανομημένη την ενέργειά του σε ένα ευρύ φάσμα συχνοτήτων. Για παράδειγμα, ο θόρυβος, ο οποίος δημιουργείται από την κυκλοφοριακή κίνηση στους δρόμους μίας μεγάλης πόλης, είναι ευρείας ζώνης. Ένα άλλο παράδειγμα είναι ο θόρυβος, ο οποίος δημιουργείται από την τυχαία ροή αέρα σε αγωγούς. Συνήθως, ο θόρυβος αυτού του είδους είναι εξαιρετικά μη προβλέψιμος, καθώς δημιουργείται από τυχαίες διαδικασίες. Σε αντίθεση, ο θόρυβος στενής ζώνης συγκεντρώνει την ενέργειά του σε συγκεκριμένες συχνότητες. Ο θόρυβος αυτού του είδους σχετίζεται κατά ένα μεγάλο ποσοστό με περιστρεφόμενες ηλεκτρικές μηχανές. Για παράδειγμα, ο εξοπλισμός, ο οποίος βασίζεται σε μηχανές εσωτερικής καύσης, όπως τουρμπίνες, βοηθητικές πηγές ενέργειας και συμπιεστές σε συστήματα ψύξης, περιέχει περιστρεφόμενα ηλεκτρομηχανολογικά εξαρτήματα, τα οποία δημιουργούν θόρυβο σε αρμονικές και υπαρμονικές της συχνότητας περιστροφής του περιστρεφόμενου μέρους. Ως αποτέλεσμα, ο θόρυβος στενής ζώνης είναι περιοδικός ή σχεδόν περιοδικός και συνεπώς σε μεγάλο βαθμό προβλέψιμος.

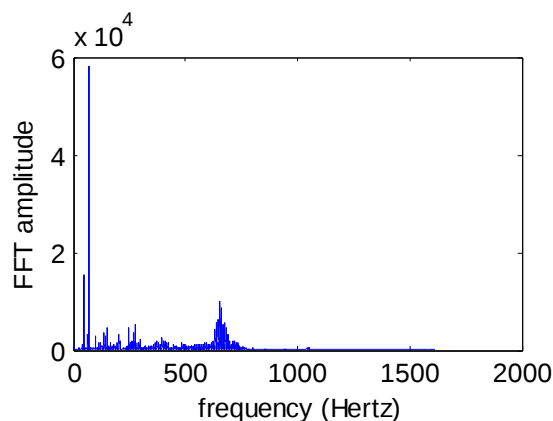
Ως ένα περισσότερο συγκεκριμένο παράδειγμα θορύβου ευρείας ζώνης θεωρούμε το θόρυβο, ο οποίος δημιουργείται από την μηχανή ενός αυτοκινήτου: [Audio Files\car engine noise.wav](#). Το φάσμα του είναι



Εικόνα 2.1 φάσμα θορύβου μηχανής αυτοκινήτου

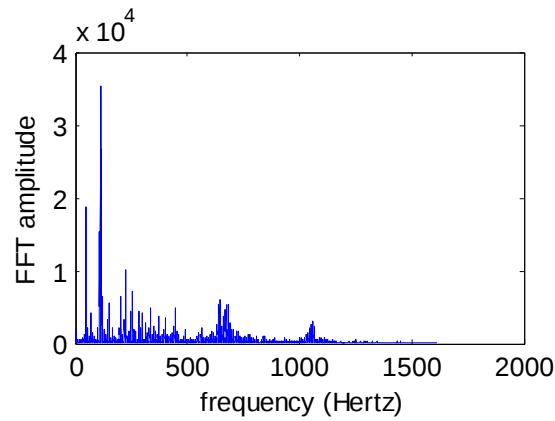
Βλέπουμε ότι το φάσμα αυτού του θορύβου περιέχει ένα συνεχές εύρος συχνοτήτων από χαμηλές έως πολύ υψηλές συχνότητες.

Ως ένα πιο συγκεκριμένο παράδειγμα θορύβου στενής ζώνης θεωρούμε το θόρυβο, ο οποίος παράγεται κατά τη λειτουργία ενός ηλεκτρικού διαδρόμου γυμναστηρίου: [Audio Files\Run_0.5.wav](#). Το φάσμα του είναι



Εικόνα 2.2 θόρυβος που παράγεται κατά τη λειτουργία ηλεκτρικού διαδρόμου σε ταχύτητα 0.5km/h

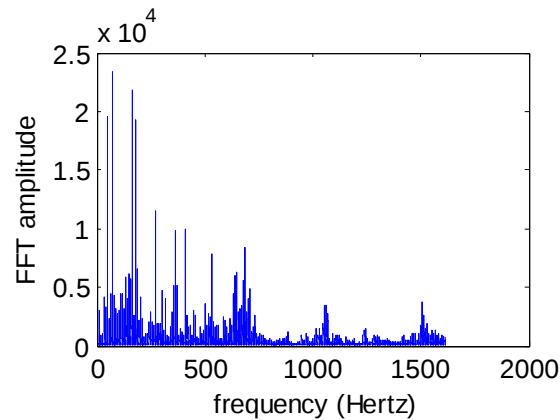
Εάν αυξήσουμε την ταχύτητα του διαδρόμου σε 1.0km/h ο θόρυβος ακούγεται ως εξής: [Audio Files\Run_1.0.wav](#). Το φάσμα του είναι



Εικόνα 2.3 θόρυβος που παράγεται κατά τη λειτουργία ηλεκτρικού διαδρόμου σε ταχύτητα 1km/h

Παρατηρούμε ότι το νέο φάσμα είναι περισσότερο πυκνό και τείνει να γίνει ευρείας ζώνης. Ακόμα όμως διαφέρει από το φάσμα της εικόνας 2.1, διότι μπορούμε να ξεχωρίσουμε ορισμένες συχνότητες, οι οποίες είναι σημαντικά πιο ισχυρές από τις υπόλοιπες.

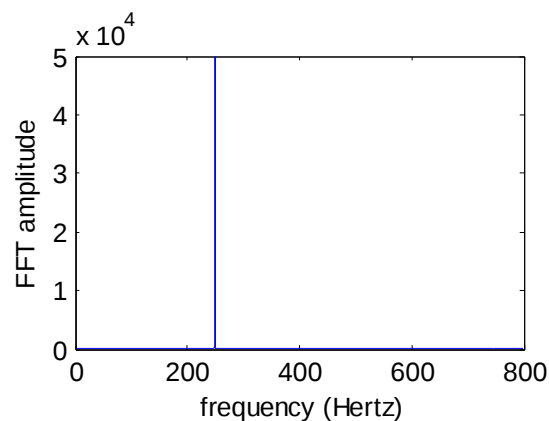
Εάν αυξήσουμε περαιτέρω την ταχύτητα σε 1.5km/h, ο θόρυβος ακούγεται ως εξής: [Audio Files\Run 1.5.wav](#), και το φάσμα του είναι



Εικόνα 2.4 θόρυβος που παράγεται κατά τη λειτουργία ηλεκτρικού διαδρόμου σε ταχύτητα 1.5km/h

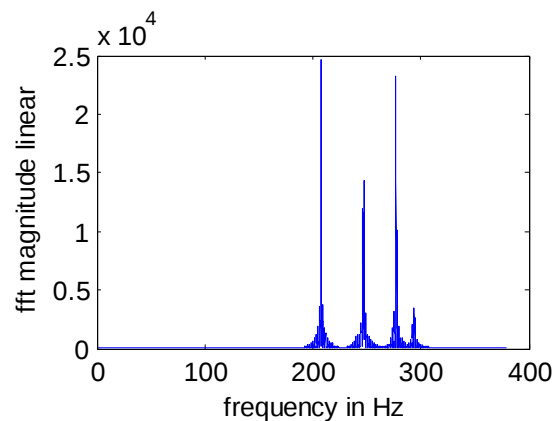
Ο θόρυβος αυτός εξακολουθεί να είναι στενής ζώνης, διότι και πάλι μπορούμε να ξεχωρίσουμε ορισμένες συχνότητες με αισθητά ισχυρότερη ενέργεια από τις υπόλοιπες. Παρ'όλα αυτά, το φάσμα του είναι αρκετά πιο περίπλοκο από αυτό των εικόνων 2.2 και 2.3, καθώς ένα σχεδόν συνεχές φάσμα συχνοτήτων έχει διεγερθεί.

Το πιο απλό παράδειγμα θορύβου στενής ζώνης είναι ένας θόρυβος αποτελούμενος από μία μοναδική συχνότητα, έστω 250Hz: [Audio Files\monochromatic noise of 250Hz.wav](#). Το φάσμα του είναι



Εικόνα 2.5 φάσμα μονοχρωματικού θορύβου στα 250Hz

Ως θόρυβος στενής ζώνης, μπορεί επίσης να θεωρηθεί και το εξής σήμα: [Audio Files\s.o.t.w. 1sec fundamental duration.wav](#) (Ίσως αναγνωρίσατε τη μελωδία, είναι το πασίγνωστο “smoke on water” των Deep Purple του 1973. Θα αναφερόμαστε στο σήμα αυτό ως το “s.o.t.w.” σήμα.). Το φάσμα του είναι



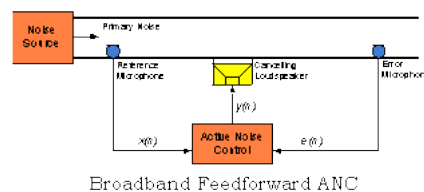
Εικόνα 2.6 φάσμα του σήματος “s.o.t.w.”

Είναι σημαντικό να σημειώσουμε, ότι αυτό είναι το φάσμα, το οποίο δίνει ο DFT. Αυτό σημαίνει ότι η συχνοτική ανάλυση γίνεται σε όλα τα δείγματα του σήματος. Όμως κάθε χρονική στιγμή μία μόνο συχνότητα είναι ενεργή, δηλαδή ένας *short time DFT* θα παρήγαγε μία μόνο συχνότητα. Αυτό επιβεβαιώνεται εύκολα, απλώς και μόνο ακούγοντας το σήμα.

2.2 Feedforward και Feedback Ενεργός Έλεγχος Θορύβου

Τα συστήματα ενεργού ελέγχου θορύβου διακρίνονται σε *feedforward* έλεγχο και *feedback* έλεγχο.

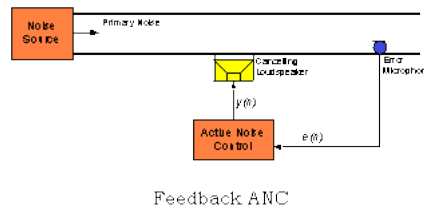
Ο *feedforward* ενεργός έλεγχος θορύβου χρησιμοποιεί κατά κανόνα δύο μικρόφωνα. Το πρώτο μικρόφωνο τοποθετείται κοντά στην πηγή του κυρίου θορύβου και ονομάζεται *μικρόφωνο αναφοράς*. Το δεύτερο μικρόφωνο τοποθετείται σε ένα κεντρικό σημείο της περιοχής, στην οποία θέλουμε να δημιουργήσουμε μία ήσυχη ζώνη και ονομάζεται *μικρόφωνο σφάλματος*. Αυτή η ιδέα απεικονίζεται στο εξής σχήμα:



Εικόνα 2.7 διάταξη *feedforward* ANC

Ο ρόλος του μικροφώνου αναφοράς, το οποίο τροφοδοτεί στον ANC ελεγκτή το *σήμα αναφοράς*, είναι να παρέχει πληροφορία στον ελεγκτή σχετικά με τον επερχόμενο θόρυβο. Με αυτόν τον τρόπο ο ελεγκτής δύναται να προβλέψει το θόρυβο, ο οποίος θα αφιχθεί στην επιθυμητή ήσυχη ζώνη, και να τον εξασθενίσει. Ο ρόλος του μικροφώνου σφάλματος είναι να μετράει τον εναπομένοντα θόρυβο και να τροφοδοτεί στον ANC ελεγκτή το λεγόμενο *σήμα σφάλματος*. Ο ελεγκτής οδηγεί το *ηχείο εξασθένισης* με ένα κατάλληλο σήμα, ονομαζόμενο αντι-θόρυβος, τέτοιο ώστε το σήμα σφάλματος να μειώνεται και συνεπώς και ο εναπομένοντα θόρυβος. Οι διατάξεις του *feedforward* ενεργού ελέγχου θορύβου, λειτουργούν αποτελεσματικά τόσο για θόρυβο στενής ζώνης όσο και για θόρυβο ευρείας ζώνης. Αξίζει να σημειώσουμε, ότι ακόμα και όταν ο κύριος θόρυβος είναι σε υψηλό ποσοστό τυχαίος, εάν το σήμα αναφοράς αφιχθεί στον ANC ελεγκτή πιο γρήγορα από το διαδιδόμενο θόρυβο από την πηγή του στην περιοχή της ήσυχης ζώνης, τότε ο ελεγκτής έχει τη δυνατότητα να προβλέψει την κυματομορφή του θορύβου όταν αυτός προσεγγίζει την ήσυχη ζώνη, και συνεπώς εγκαίρως να συνθέσει το κατάλληλο σήμα αντιθορύβου. Αυτή η λειτουργία προϋποθέτει μία γνώση της ακουστικής συνάρτησης μεταφοράς από την πηγή του κυρίου θορύβου έως τη θέση του μικροφώνου σφάλματος.

Από την άλλη πλευρά, ο *feedback* ενεργός έλεγχος θορύβου χρησιμοποιεί κατά κανόνα μόνο ένα μικρόφωνο. Μία τέτοια διάταξη έχει ως εξής:



Εικόνα 2.8 διάταξη feedback ANC

Σε αυτή τη διάταξη δεν υπάρχει μικρόφωνο αναφοράς. Αυτό συνεπάγεται ότι ο ANC ελεγκτής δε δύναται να έχει κάποια εκ των προτέρων πληροφορία σχετικά με το θόρυβο, ο οποίος πλησιάζει την ήσυχη ζώνη. Παρ'όλα αυτά, αυτό, το οποίο δύναται να κάνει ο ελεγκτής, είναι αφού αφιχθεί ο θόρυβος, να αρχίσει να τον προβλέπει βασιζόμενος στα προηγούμενα δείγματά του. Συνεπώς, εάν ο κύριος θόρυβος είναι ευρείας ζώνης και συνεπώς μη προβλέψιμος, τότε ο ελεγκτής δε θα λειτουργήσει ικανοποιητικά, καθώς δεν υπάρχει κανένας τρόπος να προβλεφθεί ένα τυχαίο σήμα από τις προηγούμενες τιμές του. Έτσι, ο feedback ενεργός έλεγχος θορύβου μπορεί να χρησιμοποιηθεί μόνο όταν ο θόρυβος προς εξασθένηση αποτελείται στο μεγαλύτερο ποσοστό του από περιοδικότητες. Η χρησιμότητα του feedback ANC έγγυται στις περιπτώσεις εφαρμογών, όπου δεν είναι δυνατόν ή πρακτικό να χρησιμοποιηθεί μικρόφωνο αναφοράς.

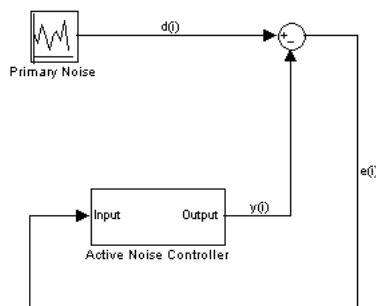
Γενικά, και ο feedback και ο feedforward ενεργός έλεγχος θορύβου έχει αναλυθεί εκτενώς στη βιβλιογραφία. Βεβαίως, το πεδίο του ANC είναι ακόμα υπό εξέλιξη και καινοτομίες καθώς και εντατική έρευνα συνεχώς λαμβάνουν χώρα. Παρ'όλα αυτά, η γενική προτίμηση της ANC κοινότητας είναι ο πολυκαναλικός feedforward ANC. Αυτό συμβαίνει, διότι τα feedforward συστήματα είναι περισσότερο σθεναρά και μπορούν να χρησιμοποιηθούν για την επίλυση περισσότερο πολύπλοκων συστημάτων. Επιπρόσθετα, ANC διατάξεις με μόνο δύο μικρόφωνα και ένα ηχείο εξασθένησης επιτυγχάνουν περιορισμένη απόδοση και συνεπώς σπάνια χρησιμοποιούνται σε πραγματικές εφαρμογές. Ο λόγος για την ανάγκη χρήσης πολλαπλών ακουστικών αισθητήρων (μικρόφωνα) και πολλαπλών ακουστικών ενεργοποιητών (ηχεία), είναι ότι, όταν ο θόρυβος περιέχει συνιστώσες υψηλής συχνότητας, τότε εξαιτίας των μικρών μηκών κύματος, το ακουστικό πεδίο θορύβου είναι και χωρικά μεταβαλλόμενο εκτός από χρονικά μεταβαλλόμενο. Εκτός από διάφορες επιλογές σχετικά με την ηλεκτροακουστική διάταξη, υπάρχει και μία μεγάλη ποικιλία αλγορίθμων παραγωγής αντι-θορύβου, ως συνάρτησης του σημάτων αναφοράς και σφάλματος. Οι αλγόριθμοι ποικίλουν από σχετικά απλούς (για παράδειγμα LMS αλγόριθμοι) έως εξαιρετικά πολύπλοκους (για παράδειγμα υβριδικοί Kalman-Laguerre αλγόριθμοι). Ο σκοπός αυτής της εργασίας δεν είναι να καλύψει όλο το πεδίο του ενεργού ελέγχου θορύβου, το οποίο γίνεται πολύ όμορφα στη βιβλιογραφία, αλλά να αναλυθούν σε βάθος κάποια από τα θεμελιώδη και εγγενή χαρακτηριστικά του, με έναν τρόπο αυτόνομο και διαφωτιστικό. Για αυτόν τον σκοπό, θα εξετάσουμε μόνο τη μονοκαναλική feedback διάταξη και θα αναδείξουμε τις ικανότητές της και τις ανικανότητές της, χρησιμοποιώντας ως κύριους θορύβους απλές κυματομορφές.

Ειδικότερα δύο σήματα θα χρησιμοποιηθούν στο εξής ως κύριοι θόρυβοι:

1. Ο μονοχρωματικός θόρυβος των 250Hz (βλέπε εικόνα 2.5)
Αυτό το σήμα είναι ιδανικό για να διαπιστώνουμε με σαφήνεια το μέγεθος της εξασθένησης στην αντιπροσωπευτική συχνότητα των 250Hz.
2. Το “s.o.t.w.” σήμα (εικόνα 2.6)
Το σήμα αυτό είναι ιδανικό για να εξετάζουμε τους μηχανισμούς μάθησης και παρακολούθησης των διάφορων ελεγκτών που θα θεωρούμε.

2.3 Ο απλοποιημένος LMS feedback ANC ελεγκτής ενεργού θορύβου

Το διάγραμμα του βασικού, απλοποιημένου feedback ANC συστήματος μπορεί να εξαχθεί απευθείας από την εικόνα 2.8:



Εικόνα 2.9 διάγραμμα του βασικού, απλοποιημένου feedback ANC συστήματος

Στην εικόνα 2.9 το $d(i)$ είναι ο κύριος θόρυβος στην περιοχή της επιθυμητής ήσυχης ζώνης, υποθέτοντας ότι ο κύριος θόρυβος έχει αρκετά χαμηλή συχνότητα ώστε να είναι χωρικά αμετάβλητος. Το $y(i)$ είναι ο αντι-θόρυβος και

$$e(i) = d(i) - y(i), \quad (2.1)$$

είναι το σήμα σφάλματος, το οποίο αντιπροσωπεύει τον εναπομένοντα θόρυβο. Η εξίσωση (2.1) μπορεί επίσης να γραφεί στη μορφή

$$d(i) = y(i) + e(i), \quad (2.2)$$

Όταν ο ελεγκτής ενεργού θορύβου δε λειτουργεί, αυτό που ακούγεται είναι ο κύριος θόρυβος $d(i)$. Αντιθέτως, όταν ο ελεγκτής είναι σε λειτουργία, αυτό το οποίο ακούμε είναι ο εναπομένων θόρυβος $e(i)$. Προκειμένου η εξασθένηση του θορύβου να είναι ικανοποιητική, το σήμα $e(i)$ πρέπει να είναι όσο το δυνατόν μικρότερο, δηλαδή

$$e(i) \approx 0, \quad (2.3)$$

Ή ισοδύναμα

$$d(i) \approx y(i), \quad (2.4)$$

Τώρα εάν η έξοδος του ελεγκτή y είναι μία καλή πρόβλεψη του κύριου θορύβου d , τότε οι (2.3) και (2.4) θα ικανοποιούνται και η εξασθένηση του κυρίου θορύβου θα έχει λάβει χώρα! Συνεπώς, αυτό το οποίο ένας ελεγκτής ενεργού θορύβου οφείλει να κάνει σε πρώτη φάση, είναι να λειτουργεί ως ένας προβλέπτης του κύριου θορύβου. Υψηλότερη ποιότητα πρόβλεψης σημαίνει μικρότερο σφάλμα πρόβλεψης, το οποίο άμεσα συνεπάγεται μικρότερο εναπομένοντα θόρυβο, εφόσον

$$\text{σφάλμα πρόβλεψης} = \text{εναπομένων θόρυβος}, \quad (2.5)$$

Βασιζόμενοι στην παραπάνω ανάλυση και ενθουσιάζοντας τις ικανότητες πρόβλεψης των προσαρμοζόμενων φίλτρων (βλέπε εξισώσεις (4.2.103)-(4.2.107)), συμπεραίνουμε εντελώς αβίαστα ότι το μπλοκ του ελεγκτή στην εικόνα 2.9 δεν πρέπει να είναι τίποτε άλλο από ένα προσαρμοζόμενο φίλτρο! Το φίλτρο πρέπει να είναι απαραίτητως προσαρμοζόμενο, καθώς ο θόρυβος ενδέχεται να μεταβάλλεται με το χρόνο. Μία λογική επιλογή είναι το LMS προσαρμοζόμενο φίλτρο μορφής 2, καθώς αυτό που έχουμε στη διάθεσή μας στο διάγραμμα της εικόνας 2.9 είναι το σφάλμα πρόβλεψης και όχι το επιθυμητό σήμα (όπως θα απαιτούσε η μορφή 1).

Παρ' όλα αυτά, εάν θέλαμε να υλοποιήσουμε κατευθείαν τις εξισώσεις (4.2.103)-(4.2.107), ένα πρόβλημα προκύπτει: Ποιό είναι το σήμα αναφοράς u ? Προκειμένου να απαντήσουμε σε αυτό το ερώτημα, πρέπει να κάνουμε μία μικρή αναδρομή στη σημασία των σημάτων d , y και u στη μορφή 2 του LMS προσαρμοζόμενου φίλτρου: θυμόμαστε ότι το u είναι το σήμα, του οποίου τα δείγματα θα χρησιμοποιήσουμε, προκειμένου να υπολογίσουμε μία πρόβλεψη του σήματος d , δηλαδή

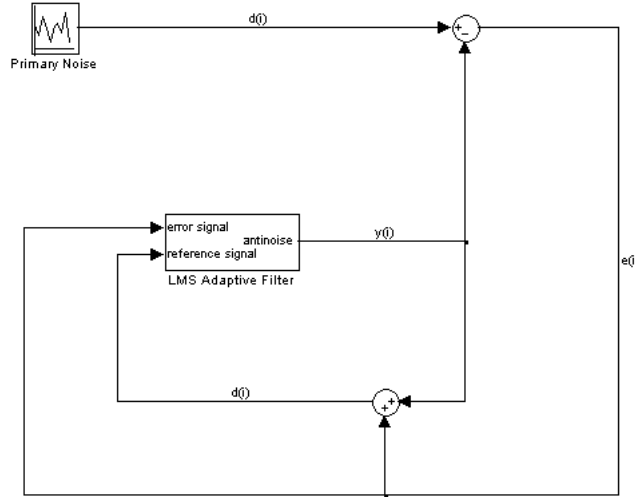
$$y(i) = \hat{d}(i) = \mathbf{w}_{i-1}^T \mathbf{u}(i), \quad (2.6)$$

Αφήνουμε για λίγο κατά μέρος την (2.6). Παρατηρούμε από την εικόνα 2.9 ότι εάν ο ελεγκτής παραγάγει τη χρονική στιγμή i ένα δείγμα εξόδου $y(i)$, τότε αυτό το σήμα θα αφαιρεθεί από το $d(i)$ και το αποτέλεσμα της αφαίρεσης $e(i)$ θα τροφοδοτηθεί πίσω στον ελεγκτή. Τώρα, εάν ο ελεγκτής, ακόμα κατά τη χρονική στιγμή i , προσθέσει στο δείγμα σφάλματος την έξοδο που μόλις παρήγαγε $y(i)$, τότε σύμφωνα με την (2.2) το αποτέλεσμα θα είναι το $d(i)$. Δυστυχώς το $d(i)$ δε μπορεί να

χρησιμοποιηθεί παρά μόνο τη χρονική στιγμή $(i + 1)$, καθώς ο ελεγκτής έχει ήδη στείλει το δείγμα $y(i)$ στο ηχείο. Λαμβάνοντας υπόψιν μας την παραπάνω παρατήρηση μαζί με την (2.6), συμπεραίνουμε ότι μία καλή επιλογή, για την ακρίβεια μία ιδανική επιλογή, για το $\mathbf{u}(i)$ είναι

$$\mathbf{u}(i) = [d(i - 1), d(i - 2), \dots, d(i - L)]^T \equiv \mathbf{d}(i - 1), \quad (2.7)$$

Πράγματι, τι θα ήταν πιο φυσικό, από το να επιχειρήσουμε να προβλέψουμε ένα σήμα από τα προηγούμενα δείγματά του? Ενσωματώνοντας τις παραπάνω ιδέες σε ένα νέο, περισσότερο λεπτομερές διάγραμμα, λαμβάνουμε



Εικόνα 2.10 το LMS προσαρμοζόμενο φίλτρο στον feedback ANC

Όπως υπογραμμίσθηκε και προηγουμένως, το δείγμα $d(i)$ μπορεί να προσδιορισθεί μόνο αφότου έχουμε εκπέμψει το δείγμα $y(i)$, και συνεπώς το $d(i)$ δε μπορεί να χρησιμοποιηθεί νωρίτερα από τη στιγμή $(i + 1)$.

Ο ελεγκτής της εικόνας 2.10 μπορεί να αποκαλείται ως *απλοποιημένος LMS feedback ελεγκτής ενεργού θορύβου*. Οι εξισώσεις, οι οποίες πλήρως περιγράφουν τη λειτουργία του είναι

$$y(i) = \mathbf{w}_{i-1}^T \mathbf{d}(i - 1), \quad (2.8)$$

$$e(i) = d(i) - y(i), \quad (2.9)$$

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{d}(i - 1) e(i), \quad (2.10)$$

$$y(i) = \hat{d}(i), \quad (2.11)$$

Είναι σημαντικό να σημειώσουμε ότι οι παραπάνω εξισώσεις είναι γραμμένες με την ίδια σειρά, με την οποία οι αντίστοιχες λειτουργίες λαμβάνουν χώρα κατά τη χρονική στιγμή i .

Τώρα, η απόδοση του ANC ελεγκτή της εικόνας 2.10 εξαρτάται από δύο παραμέτρους. Η πρώτη παράμετρος είναι το μήκος βήματος μ και η δεύτερη είναι ο αριθμός L των συντελεστών του προσαρμοζόμενου φίλτρου. Η επιλογή αυτών των δύο παραμέτρων είναι εν γένει ένα εξαιρετικά περίπλοκο θέμα και εξαρτάται κυρίως από την εμπειρία του σχεδιαστή και από τις πειραματικές δοκιμές. Παρ'όλα αυτά υπάρχουν κάποιοι γενικοί κανόνες.

Υπενθυμίζουμε ότι το μήκος βήματος θα πρέπει να ικανοποιεί την αναθεωρημένη συνθήκη ευστάθειας (βλέπε κεφάλαιο 4.2.9 του παραρτήματος)

$$\frac{0.01}{LP_u} < \mu < \frac{0.1}{LP_u}$$

Υπενθυμίζουμε επίσης ότι μικρότερο μήκος βήματος συνεπάγεται μεγαλύτερη σθεναρότητα αλλά μικρότερες ικανότητες παρακολούθησης (tracking).

Όσον αφορά τον αριθμό των συντελεστών L , για σήματα στενής ζώνης το w πρέπει να καλύπτει ένα επαρκές ποσοστό της περιόδου του κύριου θορύβου. Συνεπώς, το L πρέπει να επιλεγθεί καταλλήλως ώστε να έχουμε μία επαρκή διακριτική ικανότητα στο χρόνο ώστε να μοντελοποιήσουμε την επιθυμητή απόκριση. Γενικά, όσο πιο πολλοί είναι οι συντελεστές του προσαρμοζόμενου φίλτρου τόσο καλύτερη θα είναι και η σύγκλιση του.

Δημιουργούμε τώρα ένα περιβάλλον προσομοίωσης για το σύστημα της εικονας 2.10.

Ως κύριο θόρυβο χρησιμοποιούμε ένα ημίτονο συχνότητας 250Hz και πλάτους 1. Η ισχύς του σήματος αυτού είναι 0.5.

Το LMS προσαρμοζόμενο φίλτρο θα έχει 10 συντελεστές.

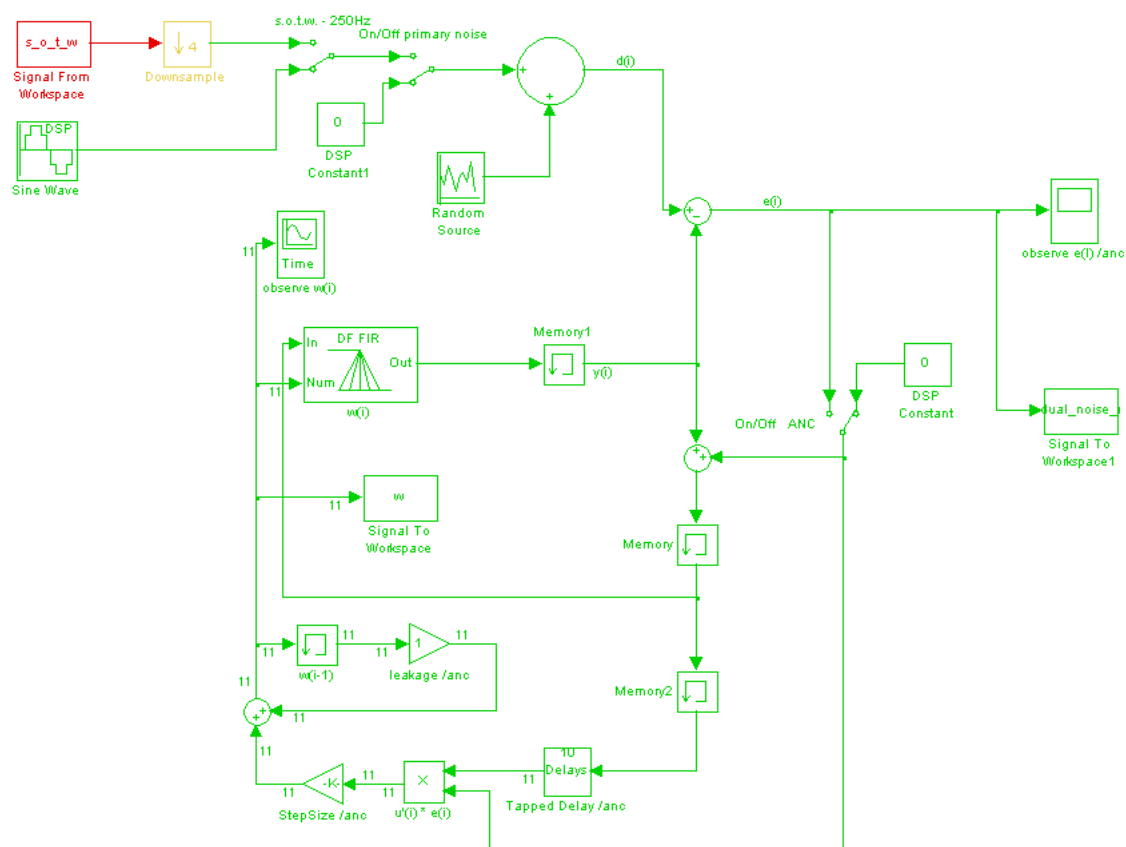
Από την εξίσωση (4.2.88) και λαμβάνοντας υπόψιν ότι το σήμα αναφοράς είναι ο ίδιος ο κύριος θόρυβος, το μήκος βήματος επιλέγεται ως

$$\mu = \frac{0.05}{LP_u} = \frac{0.05}{10 * 0.5} = 0.01, \quad (2.12)$$

Στο ημίτονο του κύριου θορύβου υπερθέτουμε σήμα λευκού θορύβου διασποράς 0.000001.

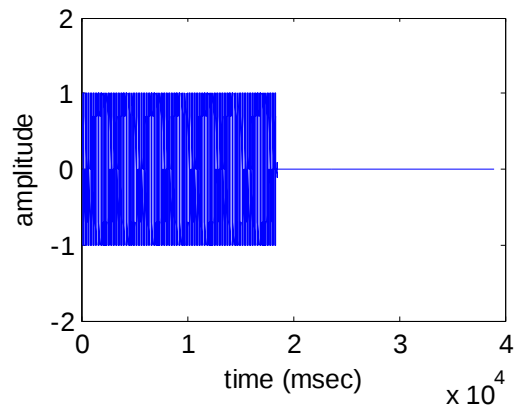
Ο ANC αλγόριθμος θα λειτουργήσει στα 2000Hz.

Το αντίστοιχο μοντέλο Simulink είναι: [Simulink Models\Simplified LMS.mdl](#). Το διάγραμμα του μοντέλου είναι



Εικόνα 2.11 μοντέλο Simulink για τον απλοποιημένο feedback ANC ελεγκτή

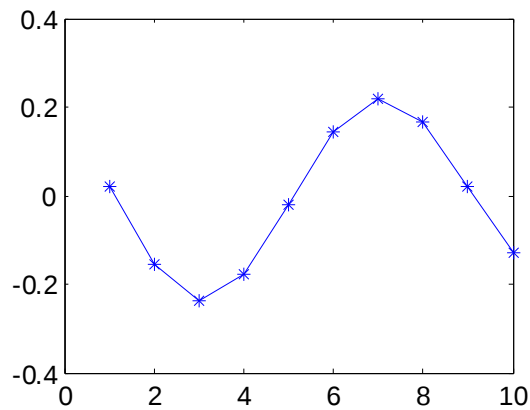
Η απόκριση (εναπομένων θόρυβος) του συστήματος είναι



Εικόνα 2.12 απόκριση του απλοποιημένου feedback ANC ελεγκτή σε κύριο θόρυβο συχνότητας 250Hz ($\mu=0.01$, διασπορά λευκού θορύβου=0.000001)

Παρατηρούμε ότι αρχικά το προσαρμοζόμενο φίλτρο, το οποίο επιτελεί την εξασθένιση θορύβου είναι ανενεργό. Αυτό θα ισχύει σε όλες τις επόμενες αποκρίσεις, έτσι ώστε η σύγκριση μεταξύ κύριου θορύβου και εναπομένοντος θορύβου να είναι εύκολη. Η εξασθένιση του θορύβου είναι 55.9dB.

Οι συντελεστές του προσαρμοζόμενου φίλτρου είναι εντοπισμένοι, δηλαδή συνέκλιναν, στις ακόλουθες θέσεις:

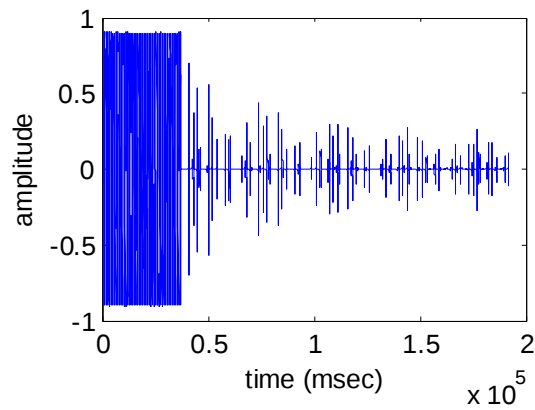


Εικόνα 2.13 w_{∞} στον απλοποιημένο feedback ANC ελεγκτή με κύριο θόρυβο συχνότητας 250Hz ($\mu=0.01$, διασπορά λευκού θορύβου=0.000001)

Ως ένα δεύτερο κύριο θόρυβο θεωρούμε το σήμα “s.o.t.w.”, το οποίο υποδειγματοληπτούμε με ένα παράγοντα 4, προκειμένου ο αλγόριθμός μας να λειτουργήσει στα 2000Hz (και όχι στα 8000Hz, η οποία είναι η συχνότητα δειγματοληψίας του σήματος “s.o.t.w.”). Αυτή η υποδειγματοληψία αφήνει το φάσμα του σήματος ανεπηρέαστο, καθώς αυτό είναι μη μηδενικό μόνο στη ζώνη από 200Hz έως 300Hz, όπως μπορεί να διαπιστωθεί από την εικόνα 2.6.

Η ισχύς του σήματος αυτού είναι 0.379, η οποία είναι μικρότερη από 0.5, αλλά όχι πολύ μικρότερη. Συνεπώς η προηγούμενη επιλογή του μήκους βήματος είναι και ασφαλής και γρήγορη.

Η απόκριση του συστήματος είναι

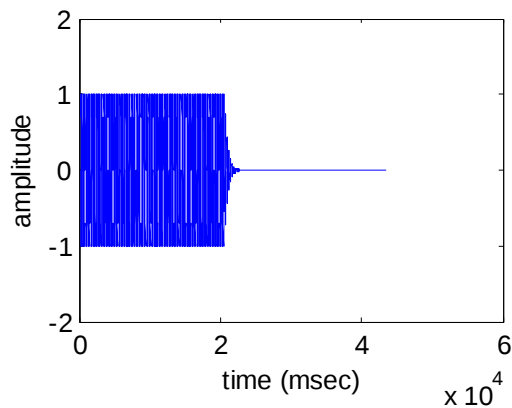


Εικόνα 2.14 απόκριση του απλοποιημένου feedback ANC ελεγκτή σε κύριο θόρυβο “s.o.t.w” ($\mu=0.01$, διασπορά λευκού θορύβου=0.000001)

Η εξασθένιση θορύβου σε ένα παράθυρο χρόνου, το οποίο περιλαμβάνει ολόκληρη την περίοδο του κύριου θορύβου, είναι 28.9dB.

Παρατηρούμε ότι στη μόνιμη κατάσταση, η απόκριση εμφανίζει κάποιες κορυφές. Αυτές οι κορυφές συμβαίνουν κάθε φορά που μία νέα συχνότητα εμφανίζεται στον κύριο θόρυβο. Οι συντελεστές του προσαρμοζόμενου φίλτρου συγκλίνουν κάθε φορά σε κατάλληλες τιμές, ώστε το φίλτρο να προβλέπει την τρέχουσα συχνότητα του κύριου θορύβου και έτσι, όταν η επόμενη συχνότητα εμφανίζεται το στιγμιαίο σφάλμα είναι πολύ μεγάλο. Υπενθυμίζουμε ότι το σφάλμα πρόβλεψης είναι ο εναπομένον θόρυβος. Παρ’όλα αυτά, οι συντελεστές του προσαρμοζόμενου φίλτρου παρακολουθούν τη φασματική αλλαγή στον κύριο θόρυβο και προσαρμόζονται κατάλληλα (παρακολουθήστε αυτήν την προσαρμογή κατά τη διάρκεια της προσομοίωσης, είναι εντυπωσιακή). Για αυτό το λόγο οι κορυφές δεν είναι πολύ υψηλές σε σχέση με το πλάτος του κύριου θορύβου. Σε αυτήν την περίπτωση το w_∞ δεν έχει κάποιο συγκεκριμένο νόημα, όπως στην περίπτωση θορύβου 250Hz, καθώς οι συντελεστές του προσαρμοζόμενου φίλτρου συγκλίνουν σε διαφορετική περιοχή του $\mathbb{R}^L$ κάθε φορά που η επόμενη συχνότητα κάνει την εμφάνισή της. Τώρα εκτελούμε την ίδια προσομοίωση αλλά με μήκος βήματος 0.001.

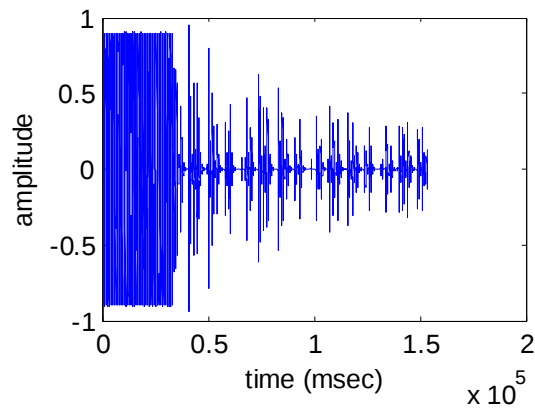
Η απόκριση του συστήματος σε θόρυβο 250Hz είναι



Εικόνα 2.15 απόκριση του απλοποιημένου feedback ANC ελεγκτή σε κύριο θόρυβο συχνότητας 250Hz ($\mu=0.001$, διασπορά λευκού θορύβου=0.000001)

Η εξασθένιση του θορύβου είναι και πάλι 55.9dB.

Η απόκριση του συστήματος σε κύριο θόρυβο “s.o.t.w.” είναι



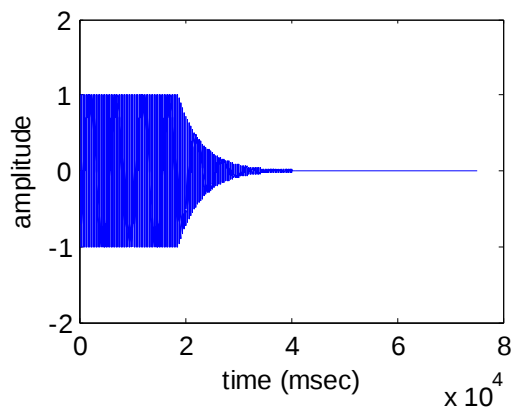
Εικόνα 2.16 απόκριση του απλοποιημένου feedback ANC ελεγκτή σε κύριο θόρυβο “s.o.t.w” ($\mu=0.001$, διασπορά λευκού θορύβου=0.000001)

Η εξασθένιση του θορύβου μειώθηκε από 28.9dB σε 26.9dB.

Το αποτέλεσμα της μείωσης του μήκους βήματος από 0.01 σε 0.001 είναι εμφανές στις εικόνες 2.15 και 2.16. Η μείωση του μήκους βήματος δεν επηρέασε την ικανότητα εκμάθησης του προσαρμοζόμενου φίλτρου. Αυτό φαίνεται ξεκάθαρα στην εικόνα 2.15, όπου η εξασθένιση μόνιμης κατάστασης είναι ακόμα 55.9dB, υπονοώντας ότι επιτεύχθηκε βέλτιστη σύγκλιση. Παρ’όλα αυτά, η μείωση του μήκους βήματος επηρέασε δυσμενώς την ικανότητα παρακολούθησης του προσαρμοζόμενου φίλτρου, με άλλα λόγια την ταχύτητά του. Αυτό φαίνεται στην εικόνα 2.16, όπου το προσαρμοζόμενο φίλτρο χρειάζεται περισσότερες επαναλήψεις προκειμένου να προσαρμόζεται σε κάθε μία από τις διαδοχικά εμφανιζόμενες συχνότητες του κύριου θορύβου. Επειδή ο κύριος θόρυβος είναι μη στάσιμος, το μικρότερο μήκος βήματος επηρεάζει την απόδοση της εξασθένισης του θορύβου, η οποία μειώθηκε από 28.9dB σε 26.9dB.

Τώρα μειώνουμε περαιτέρω το μήκος βήματος στην τιμή 0.0001.

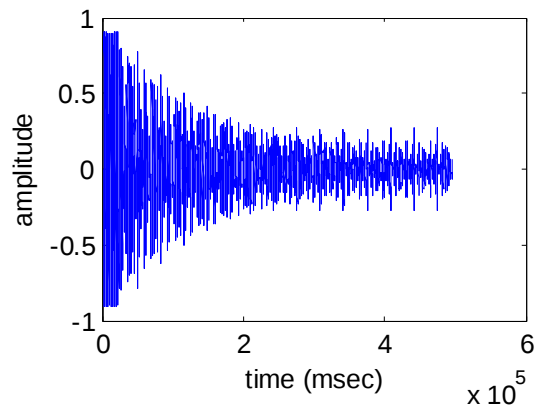
Η απόκριση του συστήματος στον κύριο θόρυβο συχνότητας 250Hz είναι



Εικόνα 2.17 απόκριση του απλοποιημένου feedback ANC ελεγκτή σε κύριο θόρυβο συχνότητας 250Hz ($\mu=0.0001$, διασπορά λευκού θορύβου=0.000001)

Η εξασθένιση του θορύβου στη μόνιμη κατάσταση είναι και τώρα 55.9dB.

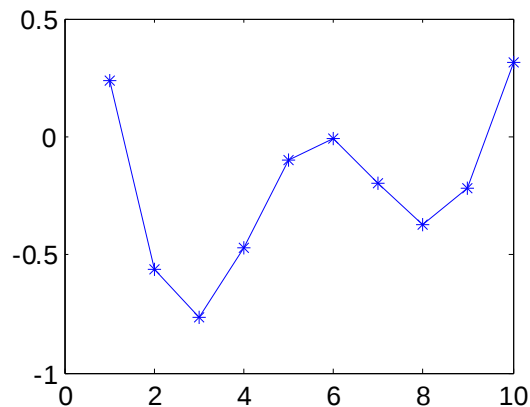
Η απόκριση του συστήματος στον “s.o.t.w.” κύριο θόρυβο είναι



Εικόνα 2.18 απόκριση του απλοποιημένου feedback ANC ελεγκτή σε κύριο θόρυβο “s.o.t.w” ($\mu=0.0001$, διασπορά λευκού θορύβου=0.000001)

Η εξασθένιση του θορύβου μειώθηκε από 26.9dB σε 19dB.

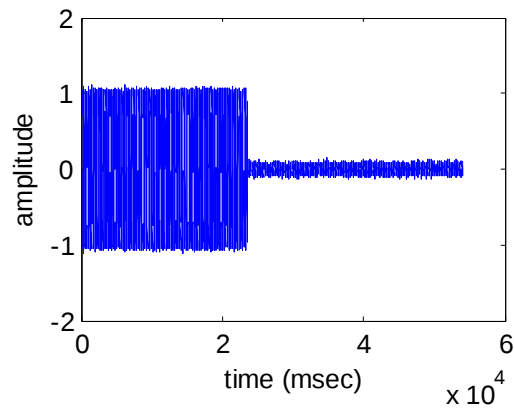
Και πάλι, το ακόμα μικρότερο μήκος βήματος δεν εμποδίζει το προσαρμοζόμενο φίλτρο από το να συγκλίνει στο βέλτιστο σημείο, στην περίπτωση του κύριου θορύβου των 250Hz. Πράγματι, η προσαρμογή είναι πιο αργή, αλλά το προσαρμοζόμενο φίλτρο έχει όλο το χρόνο στη διάθεσή του για να συγκλίνει, και ως αποτέλεσμα η απόδοσή του στη μόνιμη κατάσταση δεν επηρεάζεται από το μικρό μήκος βήματος, εξαιτίας της στασιμότητας του κύριου θορύβου. Από την άλλη πλευρά, το μικρότερο μήκος βήματος υποβαθμίζει επιπλέον την απόδοση του προσαρμοζόμενου φίλτρου, όταν ο κύριος θόρυβος είναι μη στάσιμος, όπως είναι η περίπτωση του “s.o.t.w.” σήματος, όπου ταχεία απόκριση παρακολούθησης απαιτείται. Είναι ενδιαφέρον να σημειώσουμε ότι οι συντελεστές του προσαρμοζόμενου φίλτρου είναι σε μεγάλο βαθμό εντοπισμένοι στη μόνιμη κατάσταση. Αυτό συμβαίνει επειδή οι συχνότητες του κύριου θορύβου διαδέχονται η μία την άλλη γρηγορότερα από το χρόνο που χρειάζεται το φίλτρο για να προσαρμοστεί στην κάθε μία τους. Αυτό που συμβαίνει, είναι ότι οι συντελεστές του φίλτρου καταλαμβάνουν τις βέλτιστες δυνατές θέσεις τους, με δεδομένο το μικρό μήκος βήματος, και ταλαντώνονται με ένα πολύ μικρό πλάτος, προσπαθώντας να προσαρμοστούν όσο το δυνατόν περισσότερο στην τρέχουσα συχνότητα, πρώτου η επόμενη συχνότητα εμφανισθεί και διαταράξει ελαφρώς τη θέση τους. Αυτό δε συνέβη στις περιπτώσεις, στις οποίες το μήκος βήματος είναι 0.01 ή 0.001, επειδή το φίλτρο ήταν αρκετά ταχύ ώστε να μπορεί να προσαρμόζεται σε κάθε συχνότητα, πριν την εμφάνιση της επόμενης. Οι συντελεστές βρέθηκαν εντοπισμένοι στις εξής θέσεις:



Εικόνα 2.19 w_{∞} στον απλοποιημένο feedback ANC ελεγκτή με κύριο θόρυβο “s.o.t.w.” ($\mu=0.0001$, διασπορά λευκού θορύβου=0.000001)

Τέλος, επαναφέρουμε το μήκος βήματος στην τιμή 0.01 και αυξάνουμε τη διασπορά του λευκού θορύβου από 0.000001 σε 0.001.

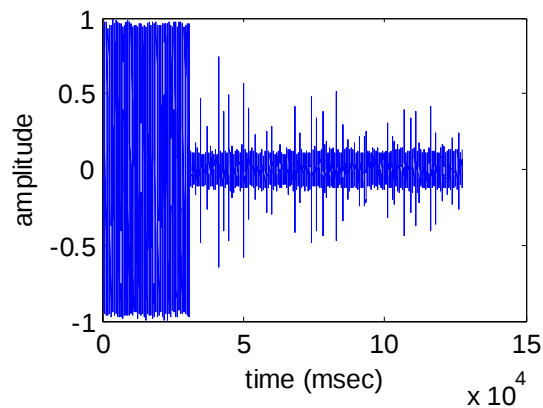
Η απόκριση του συστήματος στον κύριο θόρυβο συχνότητας 250Hz είναι



Εικόνα 2.20 απόκριση του απλοποιημένου feedback ANC ελεγκτή σε κύριο θόρυβο συχνότητας 250Hz ($\mu=0.01$, διασπορά λευκού θορύβου=0.001)

Η εξασθένιση του θορύβου μειώθηκε, σε σχέση με τις προηγούμενες προσομοιώσεις, από 55.9dB σε 25.9dB.

Η απόκριση του συστήματος στον “s.o.t.w.” κύριο θόρυβο είναι



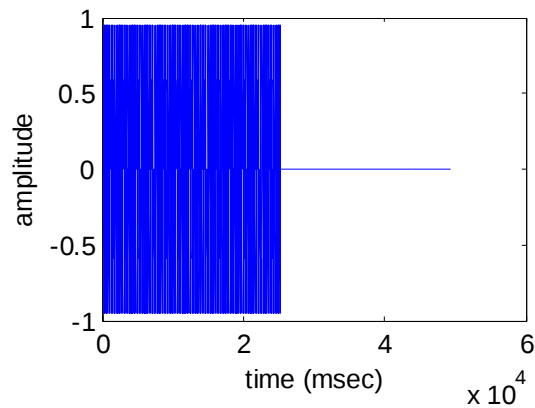
Εικόνα 2.21 απόκριση του απλοποιημένου feedback ANC ελεγκτή σε κύριο θόρυβο “s.o.t.w” ($\mu=0.01$, διασπορά λευκού θορύβου=0.001)

Η εξασθένιση θορύβου μειώθηκε από 28.9dB σε 21.8dB.

Επειδή το προσαρμοζόμενο φίλτρο λειτουργεί σε μία διάταξη ανάδρασης, έχουμε ήδη δείξει ότι επιχειρεί να εξασθενίσει το θόρυβο προβλέποντας την επόμενη τιμή του από τα προηγούμενα δείγματά του. Παρόλα αυτά είναι αδύνατον να προβλέψει το λευκό θόρυβο από τα προηγούμενα δείγματά του, όπως τώρα απαιτείται, και ως αποτέλεσμα ολόκληρος ο λευκός θόρυβος παραμένει ανεπηρέαστος στον εναπομένοντα θόρυβο, υποβαθμίζοντας την απόδοση του ελεγκτή.

Πρωτού αφήσουμε αυτήν την ενότητα, και ξεκινήσουμε να αναπτύσσουμε ένα περισσότερο ρεαλιστικό μοντέλο ενός ANC συστήματος, είναι σημαντικό να τονίσουμε το γεγονός, ότι ο απλοποιημένος feedback ANC ελεγκτής, ο οποίος μέχρι τώρα έχει εξεταστεί, είναι ξεκάθαρα ένας ελεγκτής ευρείας ζώνης. Αυτό σημαίνει ότι τα αποτελέσματα θα είναι ακριβώς τα ίδια για οποιαδήποτε συχνότητα κύριου θορύβου δοκιμάσουμε. Αυτό συμβαίνει επειδή πουθενά στις εξισώσεις (2.8)-(2.11), οι οποίες περιγράφουν τον ελεγκτή, δεν υπάρχει κάποιος περιορισμός σε σχέση με τη συχνότητα. Για να το επιδείξουμε αυτό τρέχουμε μία προσομοίωση με μήκος βήματος 0.01 και λευκό θόρυβο διασποράς 0.000001 όπως στην πρώτη προσομοίωση, με τη διαφορά ότι χρησιμοποιούμε έναν μονοχρωματικό θόρυβο συχνότητας 800Hz.

Η απόκριση του συστήματος είναι



Εικόνα 2.22 απόκριση του απλοποιημένου feedback ANC ελεγκτή σε κύριο θόρυβο συχνότητας 800Hz ($\mu=0.01$, διασπορά λευκού θορύβου=0.000001)

Η εξασθένιση θορύβου είναι, όπως αναμενόταν, 55.9dB.

2.4 Ο LMS Feedback Ελεγκτής Ενεργού Θορύβου (με Secondary Path)

Η εικόνα 2.9 εξυπηρετεί το σκοπό της αναπαράστασης της βασικής αρχής του ενεργού ελέγχου θορύβου. Παρ' όλα αυτά, δεν είναι σε καμία περίπτωση μία αναπαράσταση ενός πραγματικού ANC συστήματος. Για να γίνει ξεκάθαρο αυτό, αρκεί να παρατηρήσουμε, ότι τίποτα δεν ειπώθηκε σχετικά με τη φύση των σημάτων του διαγράμματος της εικόνας 2.9. Για παράδειγμα το $d(i)$, το οποίο αντιπροσωπεύει τον κύριο θόρυβο, είναι ένα ακουστικό ή ένα ηλεκτρονικό σήμα? Και εάν είναι ακουστικό, πώς είναι δυνατόν η έξοδος του προσαρμοζόμενου φίλτρου, η οποία είναι ένα ψηφιακό, ηλεκτρονικό σήμα, να αφαιρείται απευθείας από αυτό? Από την άλλη πλευρά, εάν το $d(i)$ δεν είναι ακουστικό σήμα αλλά ηλεκτρονικό, τότε πού βρίσκεται το σήμα του πραγματικού θορύβου στο διάγραμμα της εικόνας 2.9? Η σωστή απάντηση είναι ότι τα σήματα της εικόνας 2.9 δεν είναι ούτε ακουστικά ούτε ηλεκτρονικά. Είναι απλώς αφηρημένες ποσότητες, οι οποίες αντιπροσωπεύουν έννοιες όπως κύριος θόρυβος και αντι-θόρυβος, αλλά όχι τα αντίστοιχα φυσικά σήματα.

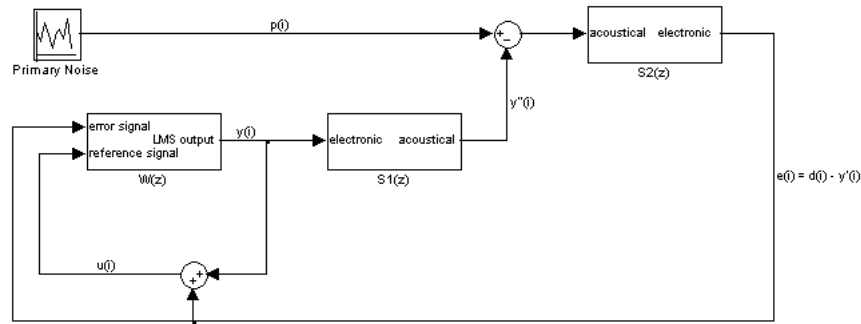
Είναι ανάγκη λοιπόν να θεμελιώσουμε ένα μπλοκ διάγραμμα, αντιπροσωπευτικό ενός πραγματικού ANC ελεγκτή. Το κλειδί προς αυτήν την κατεύθυνση είναι να διαιρέσουμε το διάγραμμα σε δύο χώρους: τον ακουστικό χώρο και τον ηλεκτρονικό χώρο. Με αυτόν τον τρόπο, η έξοδος του προσαρμοζόμενου φίλτρου θα ανήκει στον ηλεκτρονικό χώρο ενώ ο κύριος θόρυβος θα ανήκει στον ακουστικό χώρο. Οι δύο χώροι θα αλληλεπιδρούν μέσω ηλεκτροακουστικών συναρτήσεων μεταφοράς, μία για το μικρόφωνο, το οποίο συνδέει τον ακουστικό χώρο με τον ηλεκτρονικό και μία για το ηχείο, το οποίο συνδέει τον ηλεκτρονικό χώρο με τον ακουστικό.

Επιπλέον αξίζει να σημειώσουμε, ότι ένας πραγματικός ANC ελεγκτής είναι ένα υβριδικό συνεχές/διακριτό σύστημα. Ολόκληρη η επεξεργασία σήματος, την οποία εκτελεί ο μικροεπεξεργαστής είναι μία εξ' ολοκλήρου διακριτή διαδικασία, ενώ η εξασθένιση θορύβου είναι μία καθαρά συνεχής διαδικασία. Για να διευκολύνουμε την ανάπτυξη ενός ρεαλιστικού μπλοκ διαγράμματος, θεωρούμε το ισοδύναμο διακριτό σύστημα του υβριδικού συνεχούς/διακριτού συστήματος. Συνεπώς όλες οι συναρτήσεις μεταφοράς θα είναι συναρτήσεις της μεταβλητής z , υπονοώντας την εφαρμογή του μετασχηματισμού-Z σε εξισώσεις διαφορών και όλα τα σήματα θα είναι συναρτήσεις του κανονικοποιημένου δείκτη χρόνου i .

Συμβολίζουμε με $S_1(z)$ τη συνάρτηση μεταφοράς από την έξοδο του προσαρμοζόμενου φίλτρου έως την ακουστική περιοχή του μικροφώνου σφάλματος. Η $S_1(z)$ περιλαμβάνει τον D/A μετατροπέα, το φίλτρο ανακατασκευής, τον ενισχυτή ισχύος, το ηχείο και τον ακουστικό δρόμο από το ηχείο έως το μικρόφωνο σφάλματος. Επιπλέον συμβολίζουμε με $S_2(z)$ τη συνάρτηση μεταφοράς από την ακουστική περιοχή του μικροφώνου σφάλματος έως «το εσωτερικό» του μικροεπεξεργαστή. Η $S_2(z)$ περιλαμβάνει το μικρόφωνο, τον προενισχυτή, το anti-aliasing φίλτρο και τον A/D μετατροπέα. Τέλος συμβολίζουμε με $W(z)$ τη συνάρτηση μεταφοράς του προσαρμοζόμενου φίλτρου.

Η έξοδος του προσαρμοζόμενου φίλτρου $y(i)$, διέρχεται μέσω της $S_1(z)$ και φθάνει με ακουστική μορφή στην περιοχή του μικροφώνου σφάλματος, όπου υπερτίθεται στο ακουστικό σήμα του κύριου θορύβου. Μπορούμε να υποθέσουμε, ότι είτε τα δύο αυτά ακουστικά σήματα προστίθενται και ο ANC αλγόριθμος αντιστρέφει το πρόσημο της εξόδου του φίλτρου, είτε ότι δεν υπάρχει εσκεμμένη αντιστροφή προσήμου στην έξοδο του προσαρμοζόμενου φίλτρου και τα δύο ακουστικά σήματα αφαιρούνται. Η δεύτερη θεώρηση προϋποθέτει την ύπαρξη ενός μηχανισμού ακουστικής αφαίρεσης. Φυσικά τέτοιος μηχανισμός δεν υπάρχει, πρόκειται περί μίας καθαρής σύμβασης. Η ανάγκη για την πράξη της αφαίρεσης είναι ότι η έξοδος

του προσαρμοζόμενου φίλτρου είναι μία εκτίμηση του κύριου θορύβου και όχι μία εκτίμηση του κύριου θορύβου μετατοπισμένου κατά 180° . Έπειτα από την υπέρθεση του κύριου θορύβου και του αντι-θορύβου, το συνολικό σήμα διέρχεται μέσω της $S_2(z)$, η έξοδος της οποίας δεν είναι τίποτα άλλο από το σήμα σφάλματος $e(i)$. Η παραπάνω διαδικασία μπορεί να αναπαρασταθεί με το ακόλουθο μπλοκ διάγραμμα:



Εικόνα 2.23 ρεαλιστικό μοντέλο ενός feedback ANC ελεγκτή, μορφή 1

Παρατηρούμε ότι τα σήματα $p(i)$ και $y''(i)$ είναι ακουστικά. Επιπλέον το σήμα σφάλματος έχει τόσο μία ακουστική μορφή, την $p(i) - y''(i)$, όσο και μία ηλεκτρονική μορφή, την

$$e(i) = d(i) - y'(i), \quad (2.13)$$

Η ακουστική και η ηλεκτρονική μορφή του σήματος σφάλματος συνδέονται μέσω της εξίσωσης:

$$D(z) - Y'(z) = S_2(z)[P(z) - Y''(z)], \quad (2.14)$$

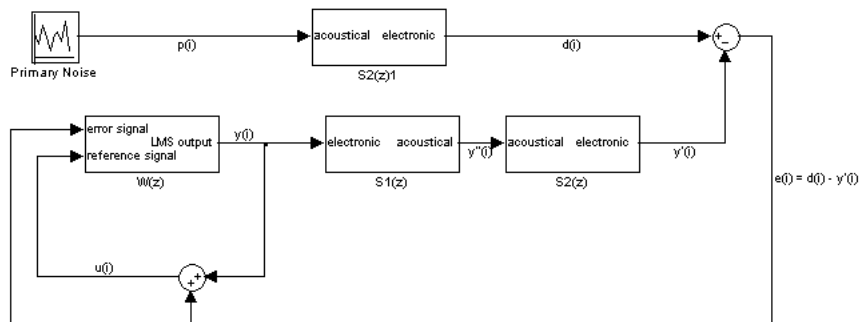
όπου όλες οι ποσότητες είναι εκφρασμένες στον z-χώρο. Από την (2.14) προκύπτει ότι

$$D(z) = S_2(z)P(z), \quad (2.15)$$

και

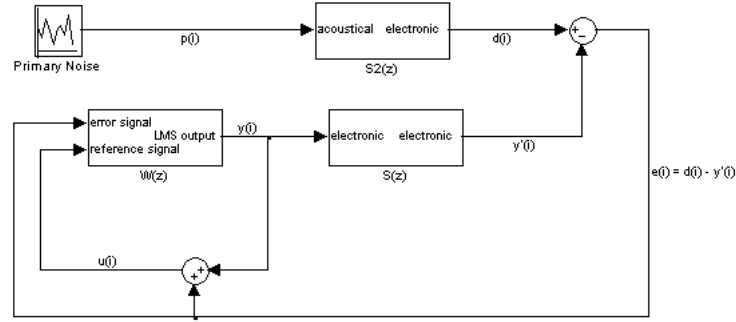
$$Y'(z) = S_2(z)Y''(z), \quad (2.16)$$

Τώρα, εφόσον όλες οι συναρτήσεις μεταφοράς, οι οποίες εξετάζονται, θεωρούνται γραμμικές (ενώ αυτό δε συμβαίνει στην πράξη), μπορούμε να μετακινήσουμε την $S_2(z)$ από τη θέση της στην εικόνα 2.23 και να την τοποθετήσουμε πριν από τον κόμβο αφαίρεσης, τόσο στον κλάδο του κύριου θορύβου όσο και στον κλάδο της εξόδου του προσαρμοζόμενου φίλτρου. Το ισοδύναμο μπλοκ διάγραμμα, το οποίο προκύπτει είναι το εξής:



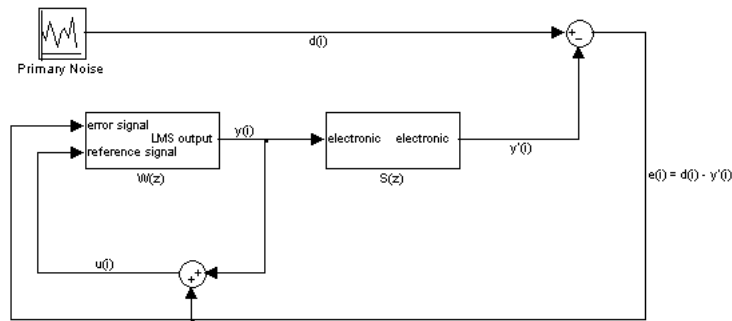
Εικόνα 2.24 ρεαλιστικό μοντέλο ενός feedback ANC ελεγκτή, μορφή 2

Παρατηρούμε ότι τα ονόματα όλων των σημάτων είναι σε συμφωνία με τις (2.15) και (2.16). Παρατηρούμε επιπλέον ότι μπορούμε να αντικαταστήσουμε τον cascade συνδυασμό των $S_1(z)$ και $S_2(z)$, ο οποίος εμφανίζεται μετά από την έξοδο του προσαρμοζόμενου φίλτρου, με μία μοναδική συνάρτηση μεταφοράς $S(z)$. Το νέο μπλοκ διάγραμμα είναι



Εικόνα 2.25 ρεαλιστικό μοντέλο ενός feedback ANC ελεγκτή, μορφή 3

Αξίζει να σημειώσουμε, ότι το ακουστικό σήμα $y''(i)$ έχει απορροφηθεί μέσα στη συνάρτηση μεταφοράς $S(z)$ και ο ακουστικός χώρος του διαγράμματος 2.25 έχει εξαφανισθεί, εκτός από το ακουστικό σήμα $p(i)$ και την ηλεκτροακουστική συνάρτηση μεταφοράς $S_2(z)$. Παρ'όλα αυτά, το σήμα, το οποίο τώρα διεγείρει ουσιαστικά το μοντέλο μας είναι το $d(i)$ και όχι το $p(i)$. Συνεπώς μπορούμε να αγνοήσουμε το $p(i)$ και την $S_2(z)$ και να αποκτήσουμε ένα μπλοκ διάγραμμα του ελεγκτή αποκλειστικά στον ηλεκτρονικό χώρο.



Εικόνα 2.26 ρεαλιστικό μοντέλο ενός feedback ANC ελεγκτή, μορφή 4

Η εικόνα 2.26 είναι ακριβώς ίδια με την 2.9, εκτός από την παρουσία της συνάρτησης μεταφοράς $S(z)$, η οποία ονομάζεται *secondary path transfer function*. Ας θυμηθούμε ότι

$$S(z) = S_1(z)S_2(z), \quad (2.17)$$

το οποίο σημαίνει ότι η $S(z)$ περιγράφει τον εξής βρόχο: μικροεπεξεργαστής $\rightarrow$ D/A $\rightarrow$ ενισχυτής ισχύος $\rightarrow$ ηχείο $\rightarrow$ ακουστικός δρόμος μεταξύ ηχείου και μικροφώνου $\rightarrow$ μικρόφωνο $\rightarrow$ προενισχυτής $\rightarrow$ A/D $\rightarrow$ μικροεπεξεργαστής.

Μεταξύ των διαγραμμάτων 2.9 και 2.26 υπάρχει επίσης μία μικρή, σχεδόν ανεπαίσθητη διαφορά, την οποία μόνο ο πολύ προσεκτικός αναγνώστης ενδέχεται να προσέξει. Ας προσέξουμε ότι το σήμα αναφοράς δεν είναι πλέον το d , όπως στο διάγραμμα 2.9 αλλά το u , το οποίο ισούται με

$$u(i) = e(i) + y(i) \xrightarrow{(2.13)} u(i) = d(i) + y(i) - y'(i), \quad (2.18)$$

Επίσης η έξοδος του προσαρμοζόμενου φίλτρου είναι

$$y(i) = \mathbf{w}_{i-1}^T \mathbf{u}(i-1), \quad (2.19)$$

σε αντιστοιχεία με την (2.7).

Ο μετασχηματισμός Z του $\mathbf{u}(i-1)$ είναι, από την (2.18)

$$z^{-1}U(z) = z^{-1}D(z) + z^{-1}[1 - S(z)]Y(z), \quad (2.20)$$

Παρατηρούμε από την (2.18) ότι τα δείγματα του κύριου θορύβου διαβρώνονται από την εξής διαταραχή:

$$f(i) = y(i) - y'(i), \quad (2.21)$$

ή ισοδύναμα

$$F(z) = Y(z) - S(z)Y(z) \Rightarrow$$

$$F(z) = [1 - S(z)]Y(z), \quad (2.22)$$

Εάν το secondary path αποτελούταν από ιδανικά στοιχεία τότε η $S(z)$ θα ήταν ίση με τη μονάδα και η $F(z)$ θα ήταν ίση με το μηδέν. Το σήμα αναφοράς θα αποτελούταν μόνο από καθαρά παρελθοντικά δείγματα του κύριου θορύβου και η πρόβλεψη του $d(i)$ από το διάνυσμα

$$[d(i-1), d(i-2), \dots, d(i-L)]^T$$

υλοποιούμενη από το LMS προσαρμοζόμενο φίλτρο, δε θα παρουσίαζε κάποια δυσκολία, πάντα με την υπόθεση ότι ο κύριος θόρυβος είναι περιοδικός (για $S(z) = 1$ τα διαγράμματα 2.9 και 2.26 είναι ταυτόσημα).

Δυστυχώς, σε πραγματικές εφαρμογές η $S(z)$ απέχει κατά πολύ από τη μονάδα και το γεγονός αυτό περιπλέκει σημαντικά τα πράγματα. Ας θεωρήσουμε ότι $S(z) \neq 1$ και ότι η $S(z)$ περιγράφεται πλήρως από τους συντελεστές ενός FIR φίλτρου, δηλαδή

$$s = [s_1, s_2, \dots, s_M]^T, \quad (2.23)$$

Ένα πρόβλημα ανακύπτει: Εφόσον το σήμα αναφοράς δεν είναι πλέον το d , όπως ήταν κατά την απουσία της $S(z)$, αλλά $d + y - y'$, με το $y - y'$ να αποτελεί μία διαταραχή, και εφόσον το σήμα σφάλματος δεν είναι το $d - y$, όπως και πάλι ήταν η περίπτωση κατά την απουσία της $S(z)$, αλλά το $d - y'$, τότε δε μπορούμε να ισχυριστούμε ότι το προσαρμοζόμενο φίλτρο παράγει εκτιμήσεις του κύριου θορύβου! Ποιά είναι λοιπόν η λειτουργία, η οποία επιτελείται από το προσαρμοζόμενο φίλτρο του διαγράμματος 2.26?

Παρατηρούμε ότι το σήμα σφάλματος είναι

$$E(z) = D(z) - Y'(z) \Rightarrow$$

$$E(z) = D(z) - S(z)Y(z), \quad (2.24)$$

Το σήμα αυτό χρησιμοποιείται για να τροφοδοτήσει τη θύρα σφάλματος του προσαρμοζόμενου φίλτρου. Συνεπώς θα υπάρξει μία προσπάθεια από την πλευρά του προσαρμοζόμενου φίλτρου να προσαρμόσει τους συντελεστές του έτσι ώστε να ελαχιστοποιηθεί η μέση τετραγωνική τιμή του σήματος σφάλματος. Υποθέτουμε λοιπόν ότι το φίλτρο συγκλίνει με ένα βέλτιστο τρόπο και ως αποτέλεσμα $E(z) = 0$. Από την (2.24) προκύπτει ότι

$$D(z) = S(z)Y(z) \xrightarrow{(4.19)}$$

$$D(z) = S(z)W(z)[z^{-1}U(z)] \Rightarrow$$

$$W(z)[z^{-1}U(z)] = \frac{D(z)}{S(z)} \Rightarrow$$

$$W(z)[z^{-1}U(z)] = S^{-1}(z)D(z), \quad (2.25)$$

Από την (2.25) βλέπουμε ότι το προσαρμοζόμενο φίλτρο επιχειρεί να προβλέψει την ποσότητα $S^{-1}(z)D(z)$ από την $z^{-1}U(z)$. Αυτό ισοδυναμεί με την πρόβλεψη του $D(z)$ και την αντίστροφη μοντελοποίηση της $S(z)$. Αλλά προβλήματα ενδέχεται να προκύψουν καθώς η $S(z)$ είναι φίλτρο με μόνο μηδενικά και συνεπώς η $S^{-1}(z)$ είναι φίλτρο μόνο με πόλους. Εάν ο κύριος θόρυβος αποτελείται από συχνότητες κοντά στα μηδενικά της $S(z)$, τότε το προσαρμοζόμενο φίλτρο θα επιχειρήσει να μοντελοποιήσει την $S^{-1}(z)$ κοντά σε κάποιους από τους πόλους της, όπου όμως λαμβάνει άπειρη τιμή, και συνεπώς η έξοδος του προσαρμοζόμενου φίλτρου θα καταστεί ασταθής. Από μία άλλη πλευρά, εάν η $S(z)$ περιέχει κάποιο μηδενικό στη συχνότητα f_1 , τότε αυτή η συχνότητα είναι είτε μη ελέγξιμη, εάν το μηδενικό ανήκει στην $S_1(z)$, είτε μη παρατηρήσιμη, εάν το μηδενικό ανήκει στην $S_2(z)$. Εάν η f_1 είναι μη παρατηρήσιμη, τότε το προσαρμοζόμενο φίλτρο ποτέ δε θα «ειδοποιηθεί» για

την ύπαρξή της. Από την άλλη πλευρά, εάν η f_1 είναι μη ελέγξιμη, ο πραγματικός, παραγόμενος αντι-θόρυβος σε αυτή τη συχνότητα θα είναι μηδενικός.

Ο ελεγκτής, οποίος αντιστοιχεί στο διάγραμμα 2.26 ονομάζεται *LMS Feedback ελεγκτής ενεργού θορύβου*. Οι εξισώσεις, οι οποίες πλήρως περιγράφουν τη λειτουργία του είναι

$$y(i) = \mathbf{w}_{i-1}^T \mathbf{u}(i-1), \quad (2.26)$$

$$u(i) = d(i) + y(i) - y'(i), \quad (2.27)$$

$$y'(i) = \mathbf{s}^T \mathbf{y}(i), \quad (2.28)$$

$$e(i) = d(i) - y'(i), \quad (2.29)$$

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{u}(i-1)e(i), \quad (2.30)$$

Δημιουργούμε τώρα ένα περιβάλλον προσομοίωσης για το ANC σύστημα του διαγράμματος 2.26.

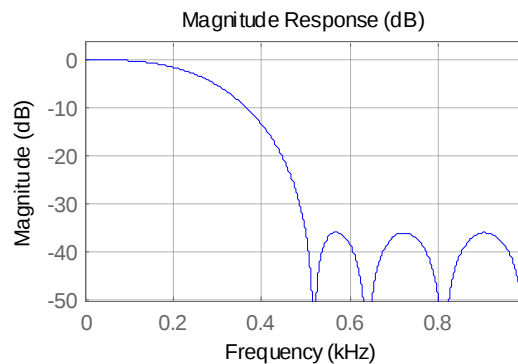
Ως κύριο θόρυβο χρησιμοποιούμε ένα ημίτονο συχνότητας 250Hz και το “s.o.t.w.” σήμα.

Η ισχύς των σημάτων αυτών υπολογίστηκε προηγουμένως. Η επιλοκή είναι ότι το σήμα αναφοράς δεν είναι πλέον ο κύριος θόρυβος, όπως υποδεικνύει η (2.18). Αλλά προκειμένου να αποφασίσουμε για το μήκος βήματος, η φόρμουλα (4.2.88) απαιτεί τη γνώση της ισχύος του σήματος αναφοράς. Όπως είναι προφανές από την (2.18), η ισχύς του σήματος αναφοράς δεν είναι γνωστή έως το χρόνο εκτέλεσης. Αυτό συμβαίνει επειδή η διαταραχή $f(i)$ είναι μία εντελώς στοχαστική ποσότητα. Για αυτόν το λόγο δε μπορούμε να αποφανθούμε με βεβαιότητα για την τιμή του μήκος βήματος. Αυτό θα γίνει σε μία πειραματική βάση. Ως αρχή επιλέγουμε μήκος βήματος ίσο με 0.01.

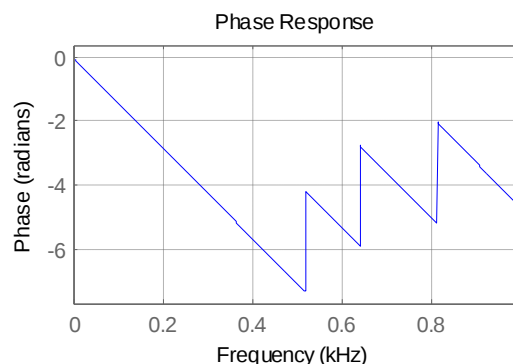
Το μέγεθος του προσαρμοζόμενου φίλτρου επιλέγεται ως 10 συντελεστές.

Η διασπορά του προσθετικού λευκού θορύβου είναι 0.000001.

Ως secondary path θα χρησιμοποιήσουμε ένα FIR φίλτρο με 10 συντελεστές, του οποίου η απόκριση συχνότητας είναι η ακόλουθη:



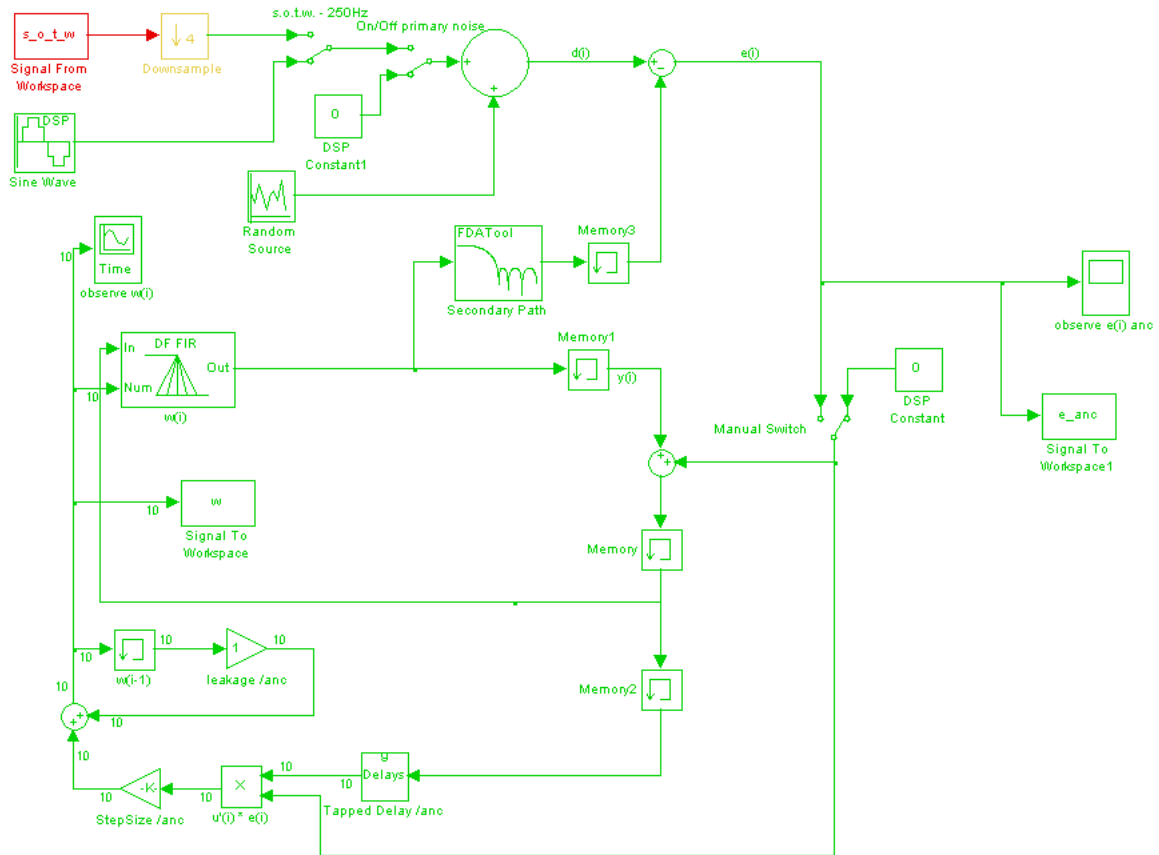
Εικόνα 2.27 απόκριση πλάτους του secondary path



Εικόνα 2.28 απόκριση φάσης του secondary path

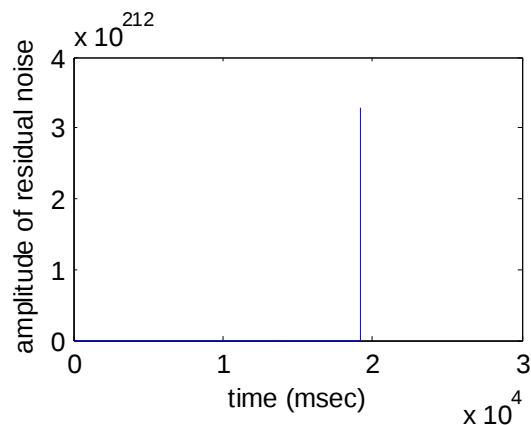
Το αντίστοιχο μοντέλο Simulink είναι το εξής: [Simulink Models\LMS_with_Secondary_Path.mdl](#)

Το διάγραμμα αυτού του μοντέλου είναι



Εικόνα 2.29 μπλοκ διάγραμμα του μοντέλου Simulink του feedback LMS ANC συστήματος με secondary path

Η απόκριση του συστήματος στον κύριο θόρυβο των 250Hz είναι

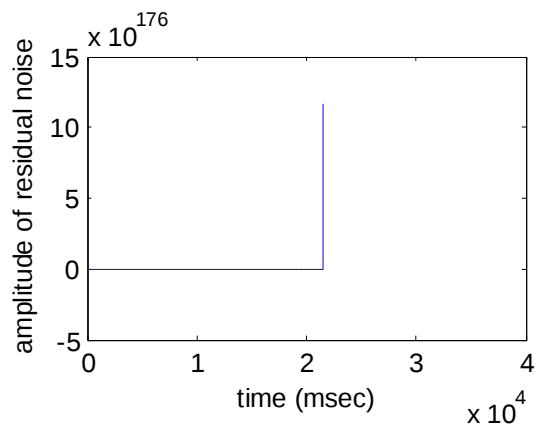


Εικόνα 2.30 απόκριση του feedback ANC συστήματος με secondary path σε κύριο θόρυβο 250Hz ($\mu=0.01$, $L=10$)

Τη στιγμή ακριβώς όπου ενεργοποιούμε το προσαρμοζόμενο φίλτρο, ο αλγόριθμος γίνεται ασταθής. Ο αντι-θόρυβος αυξάνεται χωρίς όριο και οι συντελεστές του προσαρμοζόμενου φίλτρου υπερχειλίζουν σχεδόν ακαριαία.

Εφόσον η αστάθεια είναι τόσο σοβαρή, δεν υπάρχει κάποιος λόγος να τρέξουμε την προσομοίωση με τον “s.o.t.w.” θόρυβο. Η απόκριση θα είναι επίσης ασταθής. Αντιθέτως, τρέχουμε ξανά την προσομοίωση, όμως με μήκος βήματος πολύ μικρότερο, ίσο με 0.0001, ελπίζοντας ότι η μείωση του μήκους βήματος θα οδηγήσει σε ευστάθεια.

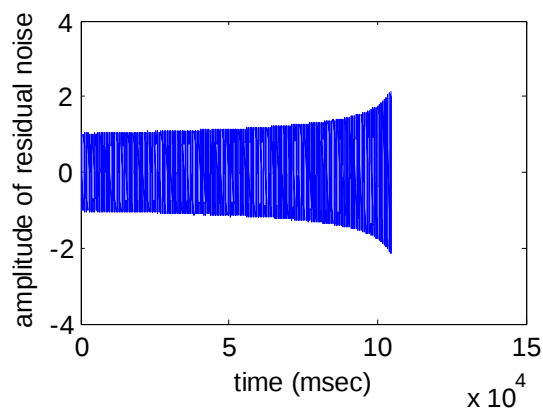
Η απόκριση του συστήματος είναι



Εικόνα 2.31 απόκριση του feedback ANC συστήματος με secondary path σε κύριο θόρυβο 250Hz ($\mu=0.0001$, $L=10$)

Η αστάθεια δεν αίρεται.

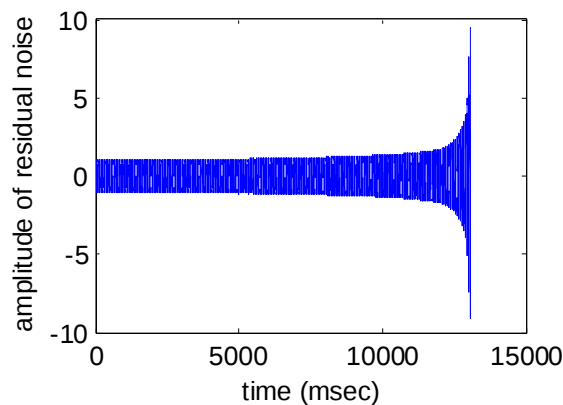
Μειώνουμε περαιτέρω το μήκος βήματος σε 0.000001. Η απόκριση του συστήματος είναι



Εικόνα 2.32 απόκριση του feedback ANC συστήματος με secondary path σε κύριο θόρυβο 250Hz ($\mu=0.000001$, $L=10$)

Η αστάθεια εξακολουθεί να εμφανίζεται, με τη διαφορά ότι εξαιτίας της μείωσης του μήκους βήματος, η εξέλιξή της στο χρόνο είναι πιο αργή.

Ως ένα τελευταίο πείραμα, θέτουμε το μήκος βήματος ίσο με 0.000001 και το μέγεθος του προσαρμοζόμενου φίλτρου ίσο προς 80 συντελεστές. Η απόκριση του συστήματος είναι



Εικόνα 2.33 απόκριση του feedback ANC συστήματος με secondary path σε κύριο θόρυβο 250Hz ($\mu=0.000001$, $L=80$)

Από τις παραπάνω προσομοιώσεις βλέπουμε ότι η παρουσία secondary path επιδρά δραματικά στην απόδοση και τη σθεναρότητα του ANC ελεγκτή, ο οποίος περιγράφεται από τις εξισώσεις (2.26)-(2.30) και από το διάγραμμα της εικόνας 2.26.

Επιπλέον δείξαμε ότι το προσαρμοζόμενο φίλτρο αδυνατεί να αντισταθμίσει την ύπαρξη του secondary path, είτε με τη μείωση του μήκους βήματος είτε με την αύξηση του μεγέθους του, όσο το σήμα e οδηγείται στη θύρα σφάλματος και το σήμα $d + y - y'$ χρησιμοποιείται ως είσοδος και ως αναφορά και την αναβάθμιση του διανύσματος βάρους. Προκειμένου να θεραπεύσουμε τις αστάθειες, τις οποίες εισάγει το secondary path, μία περισσότερο ενδελεχής εξέταση του συστήματος της εικόνας 2.26 απαιτείται.

2.5 Ο FULMS Feedback Ελεγκτής Ενεργού Θορύβου

Δείξαμε στο προηγούμενο εδάφιο ότι το προσαρμοζόμενο φίλτρο του διαγράμματος 2.26 επιχειρεί να προβλέψει την ποσότητα $S^{-1}(z)D(z)$ από την ποσότητα $z^{-1}U(z)$. Παρ'όλα αυτά, η διαδρομή προσαρμογής του $\mathbf{w}$ είναι άγνωστη και καθορίζεται σε μεγάλο βαθμό από τα χαρακτηριστικά της $S(z)$. Ένα ερώτημα ανακύπτει: μπορούμε να τροποποιήσουμε το διάγραμμα 2.26 ή ισοδύναμα τις εξισώσεις (2.26)-(2.30), με ένα τρόπο ώστε το $\mathbf{w}$ να ακολουθεί μία διαδρομή στοχαστικής κλίσης πάνω στην MSE επιφάνεια και ταυτόχρονα η επίδραση της $S(z)$ στη διαδικασία προσαρμογής του προσαρμοζόμενου φίλτρου να ελαχιστοποιείται? Για να διερευνήσουμε αυτή τη δυνατότητα, ακολουθούμε μία διαδικασία παρόμοια με τη διαδικασία εξαγωγής του LMS αλγορίθμου.

Το διαθέσιμο σήμα σφάλματος είναι

$$\begin{aligned} e(i) &= d(i) - y'(i) \\ &= d(i) - \mathbf{s}^T \mathbf{y}(i), \quad (2.31) \end{aligned}$$

όπου το $\mathbf{s}$ δίνεται από την (2.23) και $\mathbf{y}(i)$ είναι το διάνυσμα, το οποίο περιέχει τις L πιο πρόσφατες εξόδους του προσαρμοζόμενου φίλτρου

$$\mathbf{y}(i) = [y(i), y(i-1), \dots, y(i-L+1)]^T, \quad (2.32)$$

Από τις (2.31) και (2.26) έχουμε

$$e(i) = d(i) - \mathbf{s}^T \text{diag}\{\mathbf{W}_{i-1} \mathbf{U}(i-1)\}, \quad (2.33)$$

όπου

$$\mathbf{W}_{i-1} = [\mathbf{w}_{i-1}, \mathbf{w}_{i-2}, \dots, \mathbf{w}_{i-L}]^T, \quad (2.34)$$

$$\mathbf{U}(i-1) = [\mathbf{u}(i-1), \mathbf{u}(i-2), \dots, \mathbf{u}(i-L)], \quad (2.35)$$

και $\text{diag}\{\cdot\}$ είναι ο τελεστής πινάκων, ο οποίος επιστρέφει την κύρια διαγώνιο του ορίσματος του σε ένα διάνυσμα στήλη.

Τώρα μπορούμε να απαιτήσουμε η προσαρμογή να είναι της μορφής

$$\mathbf{w}_i = \mathbf{w}_{i-1} - \frac{\mu}{2} \nabla_{\mathbf{w}} \{e(i)^2\}, \quad (2.36)$$

όπου το $e(i)$ δίνεται από την (2.33). Εφόσον η κλίση στην (2.36) είναι ως προς το $\mathbf{w}$ ξαναγράφουμε την (2.33) ως

$$e(i) = d(i) - \mathbf{s}^T \text{diag}\{\mathbf{W} \mathbf{U}(i-1)\}, \quad (2.37)$$

όπου

$$\mathbf{W} = [\underbrace{\mathbf{w}, \mathbf{w}, \dots, \mathbf{w}}_L]^T, \quad (2.38)$$

$$\mathbf{w} = [w(1), w(2), \dots, w(L)]^T, \quad (2.39)$$

Σε αυτήν την περίπτωση η εξίσωση (2.37) απλοποιείται ως εξής:

$$\text{diag}\{\mathbf{W} \mathbf{U}(i-1)\} = \mathbf{U}(i-1)^T \mathbf{w}, \quad (2.40)$$

Από τις (2.40) και (2.37) φθάνουμε σε μία βολική μορφή για το σήμα σφάλματος, δηλαδή

$$e(i) = d(i) - \mathbf{s}^T \mathbf{U}(i-1)^T \mathbf{w}, \quad (2.41)$$

Ο σκοπός μας είναι να βρούμε μία έκφραση για την ποσότητα $\nabla_{\mathbf{w}}\{e(i)^2\}$. Συνεπώς,

$$\begin{aligned} \nabla_{\mathbf{w}}\{e(i)^2\} &= \left[\frac{\partial e(i)^2}{\partial w_1}, \frac{\partial e(i)^2}{\partial w_2}, \dots, \frac{\partial e(i)^2}{\partial w_L} \right]^T = \\ &= \left[2e(i) \frac{\partial e(i)}{\partial w_1}, 2e(i) \frac{\partial e(i)}{\partial w_2}, \dots, 2e(i) \frac{\partial e(i)}{\partial w_L} \right]^T = \\ &= 2e(i) \left[\frac{\partial e(i)}{\partial w_1}, \frac{\partial e(i)}{\partial w_2}, \dots, \frac{\partial e(i)}{\partial w_L} \right]^T = 2e(i) \nabla_{\mathbf{w}}\{e(i)\} \Rightarrow \\ \nabla_{\mathbf{w}}\{e(i)^2\} &= 2e(i) \nabla_{\mathbf{w}}\{e(i)\}, \quad (2.42) \end{aligned}$$

Από την (2.41) έχουμε ότι

$$\begin{aligned} \nabla_{\mathbf{w}}\{e(i)\} &= \nabla_{\mathbf{w}}\{d(i) - \mathbf{s}^T \mathbf{U}(i-1)^T \mathbf{w}\} = \\ &= \nabla_{\mathbf{w}}\{-\mathbf{s}^T \mathbf{U}(i-1)^T \mathbf{w}\} = \\ &= [-\mathbf{s}^T \mathbf{U}(i-1)^T]^T \Rightarrow \\ \nabla_{\mathbf{w}}\{e(i)\} &= -\mathbf{U}(i-1)\mathbf{s}, \quad (2.43) \end{aligned}$$

Από τις (2.42), (2.43) και (2.41) προκύπτει

$$\nabla_{\mathbf{w}}\{e(i)^2\} = -2[d(i) - \mathbf{s}^T \mathbf{U}(i-1)^T \mathbf{w}] \mathbf{U}(i-1)\mathbf{s}, \quad (2.44)$$

Από τις (2.36) και (2.44) έχουμε

$$\begin{aligned} \mathbf{w}_i &= \mathbf{w}_{i-1} + \frac{\mu}{2} 2[d(i) - \mathbf{s}^T \mathbf{U}(i-1)^T \mathbf{w}] \mathbf{U}(i-1)\mathbf{s} \Rightarrow \\ \mathbf{w}_i &= \mathbf{w}_{i-1} + \mu \mathbf{U}(i-1)\mathbf{s}[d(i) - \mathbf{s}^T \mathbf{U}(i-1)^T \mathbf{w}] \Rightarrow \\ \mathbf{w}_i &= \mathbf{w}_{i-1} + \mu \{\mathbf{U}(i-1)\mathbf{s}\}[d(i) - \mathbf{w}^T \{\mathbf{U}(i-1)\mathbf{s}\}], \quad (2.45) \end{aligned}$$

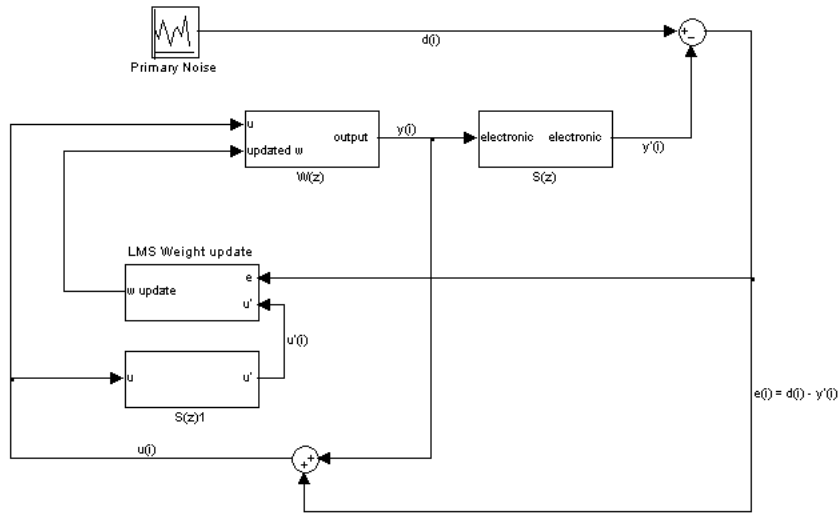
Η εξίσωση (2.38) είναι η εξίσωση αναβάθμισης, την οποία αναζητούσαμε, και υπονοεί, ότι εάν θέλουμε η προσαρμογή να ακολουθεί ένα μονοπάτι στοχαστικής κλίσης πάνω στην MSE επιφάνεια, τότε το $\mathbf{U}(i-1)\mathbf{s}$ πρέπει να χρησιμοποιηθεί για την αναβάθμιση του διανύσματος βάρους, αντί για το $\mathbf{u}(i-1)$. Αλλά το σήμα $\mathbf{U}(i-1)\mathbf{s}$ δεν είναι τίποτα περισσότερο από το συμβολισμό με πίνακες της γραμμικής συνέλιξης μεταξύ του φίλτρου $\mathbf{s}$ και του σήματος $\mathbf{u}$! Συνεπώς, το σήμα $\mathbf{U}(i-1)\mathbf{s}$ είναι ουσιαστικά το σήμα $\mathbf{u}(i-1)$ φιλτραρισμένο μέσα από το φίλτρο $\mathbf{s}$! Συμβολίζουμε το φιλτραρισμένο σήμα με $\mathbf{u}'$, δηλαδή

$$\mathbf{U}(i-1)\mathbf{s} = \mathbf{u}'(i-1), \quad (2.46)$$

και συνεπώς η (2.45) γράφεται

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{u}'(i-1)[d(i) - \mathbf{w}^T \mathbf{u}'(i-1)], \quad (2.47)$$

Εξαιτίας του φιλτραρίσματος, στο οποίο υπόκειται το σήμα $\mathbf{u}$, η φόρμουλα επαναλήψεων (2.47) ονομάζεται FULMS αλγόριθμος (Filtered – U – LMS). Το μπλοκ διάγραμμα αυτού του αλγορίθμου είναι



Εικόνα 2.34 μπλοκ διάγραμμα ενός feedback FULMS ANC συστήματος

Οι εξισώσεις, οι οποίες πλήρως περιγράφουν την λειτουργία του συστήματος του διαγράμματος 2.34, είναι

$$y(i) = \mathbf{w}_{i-1}^T \mathbf{u}(i-1), \quad (2.48)$$

$$u(i) = d(i) + y(i) - y'(i), \quad (2.49)$$

$$y'(i) = \mathbf{s}^T \mathbf{y}(i), \quad (2.50)$$

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{u}'(i-1) e(i), \quad (2.51)$$

$$e(i) = d(i) - y'(i), \quad (2.52)$$

$$\mathbf{u}'(i) = \mathbf{s}^T \mathbf{u}(i), \quad (2.53)$$

Το προσαρμοζόμενο φίλτρο εξακολουθεί να προσπαθεί να προβλέψει την ποσότητα $S^{-1}(z)D(z)$ από την ποσότητα $z^{-1}U(z)$, αλλά αυτήν τη φορά, κατεβαίνοντας την MSE επιφάνεια κατά μήκος ενός μονοπατιού στοχαστικής κλίσης.

Δημιουργούμε τώρα ένα περιβάλλον προσομοίωσης των εξισώσεων (2.49)-(2.53).

Ως κύριους θορύβους χρησιμοποιούμε το ημίτονο των 250Hz και το “s.o.t.w.” σήμα.

Θέτουμε το μήκος βήματος ίσο προς 0.01.

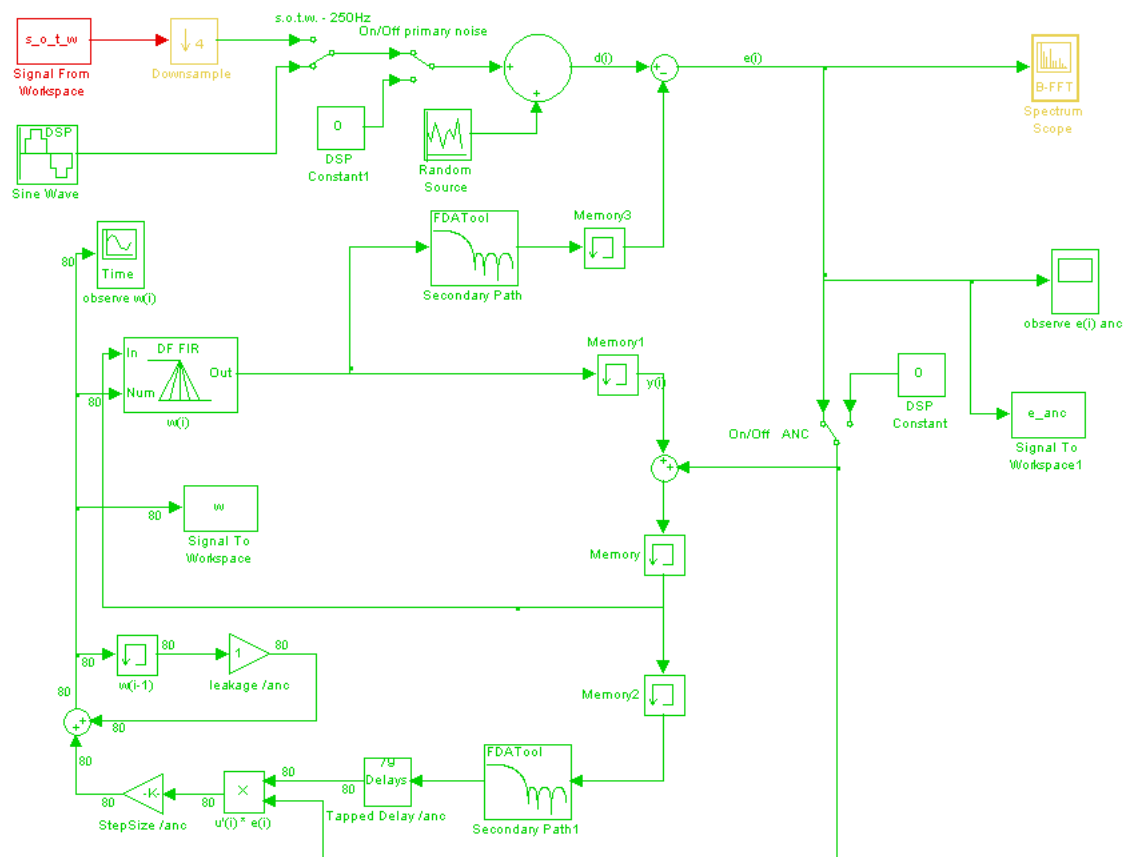
Η διασπορά του προσθετικού λευκού θορύβου είναι 0.000001.

Το προσαρμοζόμενο φίλτρο έχει 10 συντελεστές, όπως και προηγουμένως.

Το secondary path είναι αυτό των εικόνων 2.27 και 2.28.

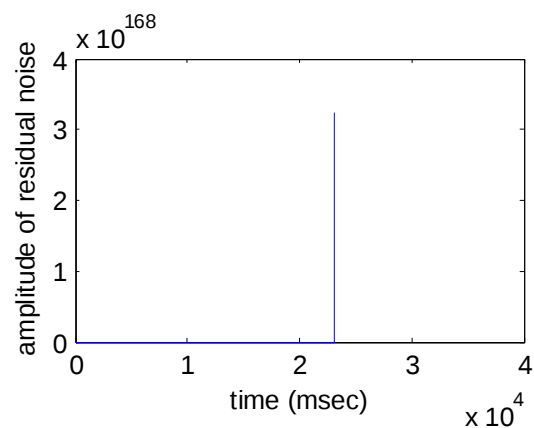
Το αντίστοιχο μοντέλο του Simulink είναι: [Simulink Models\FULMS.mdl](#).

Το διάγραμμα του μοντέλου είναι



Εικόνα 2.35 μπλοκ διάγραμμα του μοντέλου Simulink ενός feedback FULMS ANC συστήματος

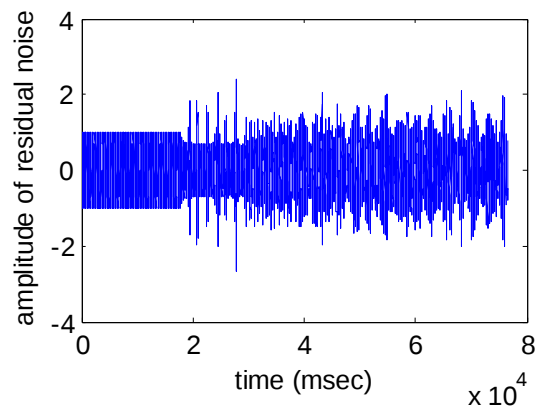
Η απόκριση του συστήματος στον κύριο θόρυβο των 250Hz είναι



Εικόνα 2.36 απόκριση του feedback FULMS ANC συστήματος σε κύριο θόρυβο 250Hz ($\mu=0.01$, $L=10$)

Το σύστημα είναι ασταθές και συνεπώς δοκιμάζουμε ένα μήκος βήματος μικρότερο, για παράδειγμα 0.001.

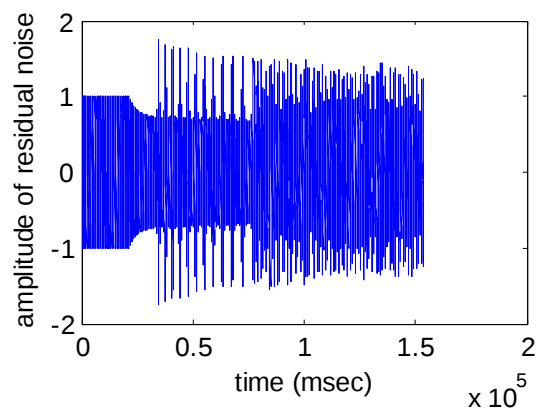
Η απόκριση του συστήματος είναι



Εικόνα 2.37 απόκριση του feedback FULMS ANC συστήματος σε κύριο θόρυβο 250Hz ($\mu=0.001$, $L=10$)

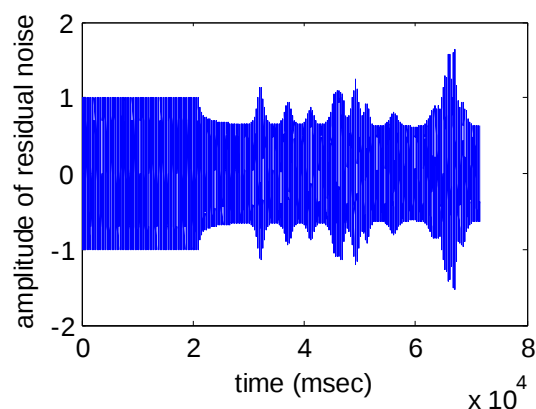
Η αστάθεια αφαιρέθηκε! Φυσικά καμία εξασθένιση δεν έλαβε χώρα.

Μειώνουμε περαιτέρω το μήκος βήματος σε 0.0001. Η απόκριση του συστήματος είναι



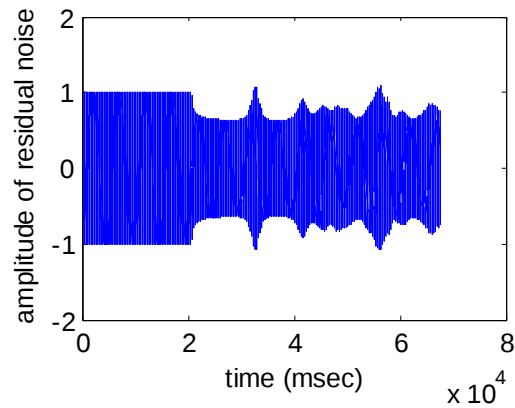
Εικόνα 2.38 απόκριση του feedback FULMS ANC συστήματος σε κύριο θόρυβο 250Hz ($\mu=0.0001$, $L=10$)

Ως ένα τελικό πείραμα, διατηρούμε το μήκος βήματος στην τιμή 0.0001, αλλά αυξάνουμε το μήκος του προσαρμοζόμενου φίλτρου σε 80 συντελεστές. Η απόκριση του συστήματος είναι:



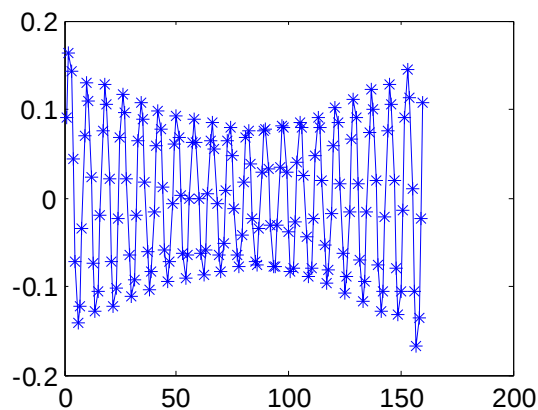
Εικόνα 2.39 απόκριση του feedback FULMS ANC συστήματος σε κύριο θόρυβο 250Hz ($\mu=0.0001$, $L=80$)

Αυξάνουμε και πάλι το μέγεθος του φίλτρου σε 160 συντελεστές. Η απόκριση του συστήματος είναι



Εικόνα 2.40 απόκριση του feedback FULMS ANC συστήματος σε κύριο θόρυβο 250Hz ($\mu=0.0001$, $L=160$)

Η απόκριση είναι ομαλή αλλά οριακά ευσταθής. Είναι πολύ ενδιαφέρον να σημειώσουμε ότι κατά την προσομοίωση η κατανομή των βαρών αλλάζει σχήμα, με αποτέλεσμα τα βάρη να μην είναι εντοπισμένα. Μία τυπική κατανομή είναι η εξής:



Εικόνα 2.41 μία βραχυπρόθεσμη κατανομή των βαρών κατά τη διάρκεια προσομοίωσης ενός feedback FULMS ANC συστήματος με κύριο θόρυβο στα 250Hz ($\mu=0.0001$, $L=160$)

Από την εικόνα 2.41 συμπεραίνουμε ότι ο εναπομένον θόρυβος περιέχει ταλαντώσεις σε περισσότερες από μία συχνότητες, παρά το γεγονός ότι ο κύριος θόρυβος είναι μονοχρωματικός.

Δίνουμε τώρα μία ερμηνεία των κορυφών, οι οποίες εμφανίζονται στον εναπομένοντα θόρυβο των εικόνων 2.37-2.40. Εγγενείς μη κανονικότητες στις εξισώσεις (4.48)-(4.53), οδηγούν σε ταλαντώσεις θετικής ανάδρασης σε συχνότητες διαφορετικές από τη συχνότητα του κύριου θορύβου. Κάθε φορά που μία αυξανόμενη ταλάντωση γεννάται, το στιγμιαίο σφάλμα είναι μεγάλο και μία κορυφή στον εναπομένοντα θόρυβο δημιουργείται. Παρ'όλα αυτά, η έξοδος του προσαρμοζόμενου φίλτρου δεν αυξάνεται χωρίς όριο, διότι ένας μηχανισμός εντοπισμού και παρακολούθησης ανιχνεύει το μεγάλο σφάλμα και προσαρμόζει τα βάρη έτσι ώστε το σφάλμα να ελαχιστοποιηθεί, κατεβαίνοντας ένα μονοπάτι στοχαστικής κλίσης στην MSE επιφάνεια. Αλλά πρώτου μία αυξανόμενη ταλάντωση αποσβεθεί ικανοποιητικά, μία άλλη ταλάντωση διαφορετικής συχνότητας εμφανίζεται οδηγώντας και πάλι σε ένα μεγάλο στιγμιαίο σφάλμα, με αποτέλεσμα τα βάρη να προσαρμόζονται με ένα διαφορετικό τρόπο ώστε να αποσβέσουν τη νέα αυτήν ταλάντωση. Με αυτόν τον τρόπο, το προσαρμοζόμενο φίλτρο είναι απασχολημένο συνεχώς με το να καταστέλλει αυτές τις «εκρήξεις» αστάθειας και εμποδίζεται από το κύριο καθήκον του, το οποίο είναι η εξασθένιση του κύριου θορύβου (αξίζει να παρατηρήσετε αυτό το φαινόμενο στο φασματοσκόπιο του μοντέλου Simulink, είναι εντυπωσιακό).

Από τις παραπάνω προσομοιώσεις εξάγουμε δύο σημαντικά συμπεράσματα. Το πρώτο είναι ότι η υλοποίηση των εξισώσεων (2.48)-(2.53) οδηγεί σε έναν περισσότερο σθεναρό ελεγκτή από τις εξισώσεις (2.26)-(2.30). Ο λόγος είναι ότι στις εξισώσεις (2.48)-(2.53) έχει ληφθεί μέριμνα ώστε η προσαρμογή των βαρών ακολουθεί ένα μονοπάτι στοχαστικής κλίσης πάνω στην MSE επιφάνεια, σε αντίθεση με τις εξισώσεις (2.26)-(2.30). Το δεύτερο συμπέρασμα είναι ότι ο ελεγκτής των εξισώσεων (2.48)-

(2.53) συνεχίζει να είναι ασταθής, παρουσιάζοντας θετική ανάδραση και η απόδοσή του είναι μη αποδεκτή. Το αίτιο είναι η διαταραχή

$$f(i) = y(i) - y'(i)$$

η οποία διαβρώνει το σήμα αναφοράς με τον ακόλουθο τρόπο:

$$u(i) = d(i) + f(i)$$

2.6 Ο βελτιωμένος FULMS Feedback Ελεγκτής Ενεργού Θορύβου (με Secondary Path)

Ο z-μετασχηματισμός του $f(i)$ είναι

$$F(z) = Y(z) - Y'(z) \Rightarrow$$

$$F(z) = Y(z) - S(z)Y(z), \quad (2.54)$$

Τα χαρακτηριστικά του ελεγκτή, ο οποίος περιγράφεται από τις εξισώσεις (2.48)-(2.53), σίγουρα θα βελτιώνονταν, εάν μπορούσαμε να ελαχιστοποιήσουμε τη διαταραχή $f(i)$. Συνεπώς, ενδιαφερόμαστε να καταστήσουμε την ποσότητα $F(z)$ όσο το δυνατόν μικρότερη. Αυτό μπορεί να επιτευχθεί, εάν αποκτήσουμε το σήμα αναφοράς μας όχι προσθέτοντας απευθείας την έξοδο του προσαρμοζόμενου φίλτρου στο σήμα σφάλματος, αλλά φιλτράροντας πρώτα την έξοδο μέσω του s . Η αιτιολόγηση για αυτήν την επιλογή είναι ότι το σήμα αναφοράς θα γίνει The justification for this decision is that the reference signal will be

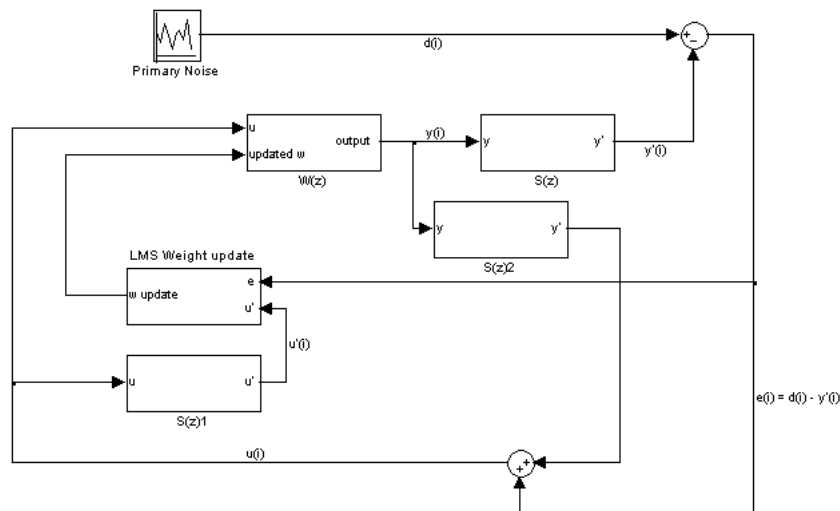
$$u(i) = d(i) + y'(i) - y'(i) = d(i)$$

ενώ η προηγούμενη μορφή του είναι

$$u(i) = d(i) + y(i) - y'(i) = d(i) + f(i)$$

και συνεπώς η διαταραχή εξουδετερώνεται εντελώς!

Το βελτιωμένο μπλοκ διάγραμμα είναι



Εικόνα 2.42 βελτιωμένος FULMS feedback ελεγκτής ενεργού θορύβου

Ο ελεγκτής, ο οποίος αντιστοιχεί στο διάγραμμα της εικόνας 2.42, μπορεί να αποκαλείται ως *βελτιωμένος FULMS Feedback ελεγκτής ενεργού θορύβου*. Οι εξισώσεις, οι οποίες πλήρως περιγράφουν τη λειτουργία του είναι οι εξής:

$$y(i) = \mathbf{w}_{i-1}^T \mathbf{u}(i-1), \quad (2.55)$$

$$u(i) = d(i), \quad (2.56)$$

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{u}'(i-1)e(i), \quad (2.57)$$

$$\mathbf{u}'(i) = \mathbf{s}^T \mathbf{u}(i), \quad (2.58)$$

$$e(i) = d(i) - y'(i), \quad (2.59)$$

$$y'(i) = \mathbf{s}^T \mathbf{y}(i), \quad (2.60)$$

Θα δημιουργήσουμε τώρα, όπως συνήθως, ένα περιβάλλον προσομοίωσης, το οποίο υλοποιεί τις παραπάνω εξισώσεις και συνεπώς τον αντίστοιχο ελεγκτή.

Ως κύριους θορύβους θεωρούμε το ημίτονο των 250Hz και το “s.o.t.w.” σήμα.

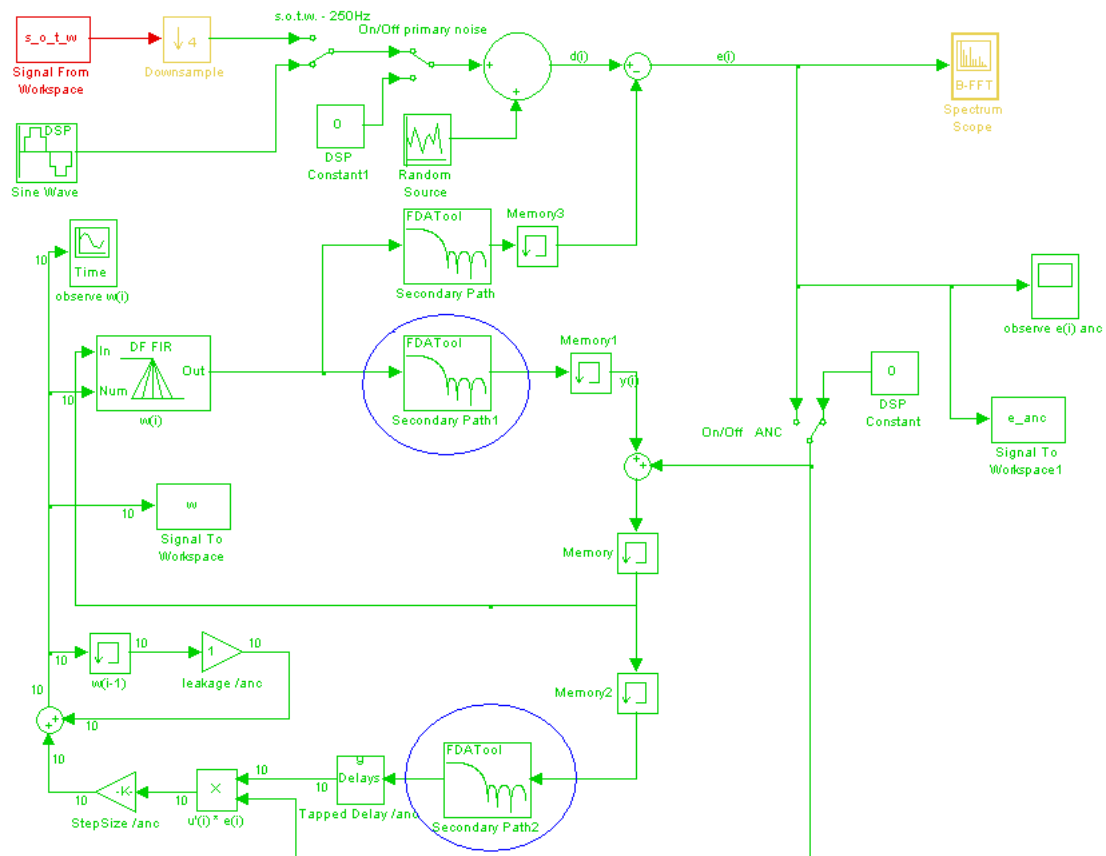
Το μήκος βήματος είναι 0.01.

Το μέγεθος του προσαρμοζόμενου φίλτρου είναι 10 συντελεστές.

Η διασπορά του προσθετικού λευκού θορύβου είναι 0.000001.

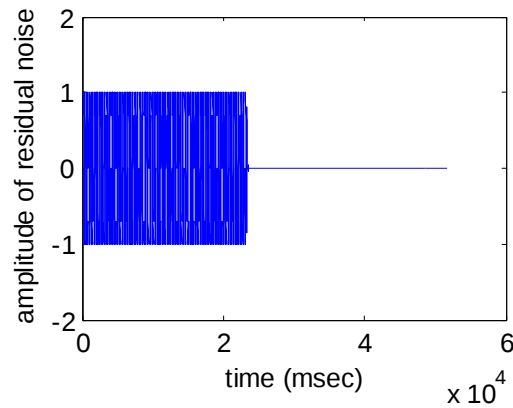
Το secondary path είναι αυτό των εικόνων 2.27 και 2.28.

Το αντίστοιχο μοντέλο Simulink είναι: [Simulink Models\improved Simplified FULMS.mdl](#). Το διάγραμμα αυτού του μοντέλου είναι



Εικόνα 2.43 μπλοκ διάγραμμα του βελτιωμένου FULMS feedback ANC ελεγκτή

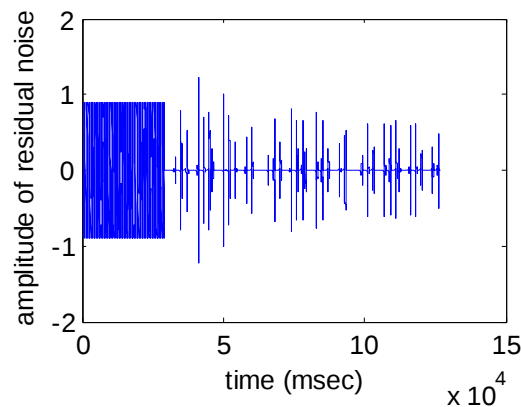
Η απόκριση του συστήματος στον κύριο θόρυβο των 250Hz είναι



Εικόνα 2.44 απόκριση του βελτιωμένου FULMS feedback ANC ελεγκτή σε κύριο θόρυβο 250Hz

Η εξασθένιση θορύβου είναι 55.9dB.

Η απόκριση του συστήματος στον “s.o.t.w.” κύριο θόρυβο είναι



Εικόνα 2.45 απόκριση του βελτιωμένου FULMS feedback ANC ελεγκτή στον “s.o.t.w.” θόρυβο

Η εξασθένιση του θορύβου είναι 28.9dB.

Από τις εικόνες 2.44 και 2.45 βλέπουμε ότι όλες οι αστάθειες έχουν αφαιρεθεί και η απόδοση του ελεγκτή είναι ακριβώς η ίδια, όπως στην ιδανική περίπτωση των εξισώσεων (2.8)-(2.11). Το λόγο δεν είναι δύσκολο να τον φανταστούμε: αυτός ο ελεγκτής είναι επίσης ιδανικός, καθώς ο αλγόριθμος κάνει απευθείας χρήση των συντελεστών του secondary path (δείτε τα ενκυκλωμένα φίλτρα στο διάγραμμα του Simulink μοντέλου)

$$\mathbf{s} = [s_1, s_2, \dots, s_M]^T$$

Φυσικά, κάτι τέτοιο είναι αδύνατο στην πράξη, απλώς επειδή οι παραπάνω συντελεστές είναι άγνωστοι. Παρ’όλα αυτά μπορούμε να χρησιμοποιήσουμε μία προσέγγιση $\hat{\mathbf{s}}$ για τους συντελεστές του secondary path, και τα αποτελέσματα των εικόνων 2.44 και 2.45 εγγυώνται ότι όσο καλύτερη είναι αυτή η προσέγγιση, τόσο η απόδοση θα είναι πιο κοντά στην ιδανική.

2.7 Προσαρμοζόμενη Αναγνώριση του Secondary Path

Η αναγνώριση συστήματος είναι μία πειραματική προσέγγιση στη μοντελοποίηση μίας διαδικασίας. Η βασική ιδέα είναι η διέγερση του συστήματος με κατάλληλα επιλεγμένα σήματα, για παράδειγμα βηματικές εισόδους, ημιτονοειδείς εισόδους, chirp σήματα ή ακόμα και λευκό θόρυβο, ώστε να μετρηθούν οι αντίστοιχες αποκρίσεις του συστήματος και να χρησιμοποιηθούν τα δεδομένα εισόδου/εξόδου για την κατασκευή ενός μοντέλου του συστήματος. Ο όρος αναγνώριση χρησιμοποιείται για να περιγράψει την παραπάνω διαδικασία.

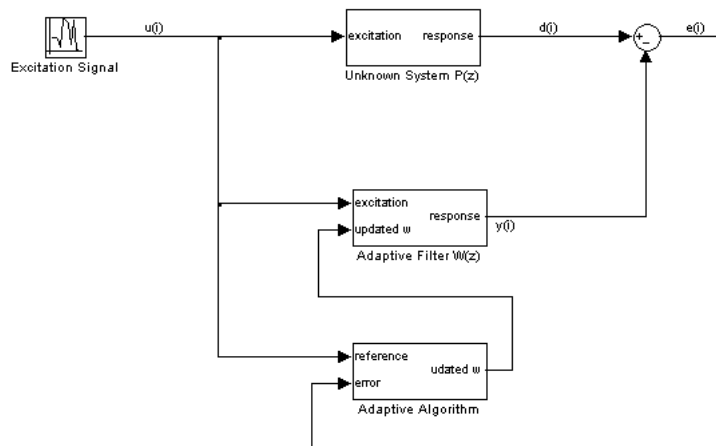
Οι τεχνικές αναγνώρισης μπορούν να κατηγοριοποιηθούν βάσει δύο διαφορετικών χαρακτηριστικών, τα οποία φέρει κάθε τεχνική. Το πρώτο χαρακτηριστικό είναι ο αριθμός των ζευγαριών δειγμάτων εισόδου/εξόδου, τα οποία απαιτούνται για κάθε

ενημέρωση ή αναβάθμιση του υπό κατασκευή μοντέλου. Το δεύτερο χαρακτηριστικό είναι, εάν η αναβάθμιση του μοντέλου συμβαίνει σε πραγματικό χρόνο ή όχι. Εάν ο αλγόριθμος απαιτεί περισσότερα του ενός ζεύγη εισόδου/εξόδου για κάθε ενημέρωση του μοντέλου, τότε η αναγνώριση λειτουργεί με *batch τρόπο*. Αντιθέτως εάν για κάθε ζεύγος εισόδου/εξόδου πραγματοποιείται και μία ενημέρωση του μοντέλου, τότε η αναγνώριση λειτουργεί *δείγμα προς δείγμα*. Τώρα, εάν οι μετρήσεις συλλέγονται από το σύστημα και η επεξεργασία τους γίνεται αργότερα, σε έναν ουδέτερο χρόνο, τότε η αναγνώριση γίνεται *off-line*, δηλαδή όχι σε πραγματικό χρόνο. Από την άλλη πλευρά, εάν οι ενημερώσεις του μοντέλου βασίζονται στις εξόδους του φυσικού συστήματος ακριβώς τη στιγμή, την οποία εκείνο τις παράγει, τότε η αναγνώριση λειτουργεί *on line*.

Συνεπώς μία στρατηγική αναγνώρισης μπορεί να είναι *batch και on-line*, *batch και off-line*, *δείγμα προς δείγμα και off-line* ή *δείγμα προς δείγμα και on-line*. Διαφορετικοί συνδυασμοί εξυπηρετούν διαφορετικές ανάγκες. Γενικά, η on-line αναγνώριση είναι πιο δύσκολη από την off-line και πρέπει να επιλέγεται μόνο εάν είναι απολύτως απαραίτητη, καθώς ανάμεσα στα άλλα, οι αλγόριθμοι που μπορούν να χρησιμοποιηθούν on-line είναι πιο απλοί, και συνεπώς εισάγουν μεγαλύτερο σφάλμα, από τους αλγόριθμους, οι οποίοι μπορούν να χρησιμοποιηθούν off-line.

Εφόσον συζητάμε για αναγνώριση συστήματος στα πλαίσια του ενεργού ελέγχου θορύβου, το οποίο είναι καθαρά σύστημα πραγματικού χρόνου, ενδιαφερόμαστε μόνο για on-line και δείγμα προς δείγμα αναγνώριση. Περισσότερα επιχειρήματα θα δοθούν για αυτήν την επιλογή αργότερα.

Τώρα, βασιζόμενοι στην ανάπτυξη των προσαρμοζόμενων φίλτρων στο παράρτημα, μπορούμε να ισχυριστούμε ότι η on-line και δείγμα προς δείγμα αναγνώριση είναι ισοδύναμη προς μία προσαρμοζόμενη αναγνώριση, όπως απεικονίζεται παρακάτω:



Εικόνα 2.46 προσαρμοζόμενη αναγνώριση συστήματος

Το $P(z)$ είναι ένα άγνωστο σύστημα, το οποίο θέλουμε να αναγνωρίσουμε και $W(z)$ είναι ένα ψηφιακό φίλτρο, το οποίο χρησιμοποιείται για να μοντελοποιήσει το $P(z)$, βάσει ενός προκαθορισμένου αλγορίθμου ελαχιστοποίησης σφάλματος. Διεγείροντας το άγνωστο σύστημα $P(z)$ και το ψηφιακό μοντέλο $W(z)$ με το ίδιο σήμα εισόδου $u(i)$ και μετρώντας τα σήματα εξόδου $d(i)$ και $y(i)$, επιθυμούμε να καθορίσουμε τα χαρακτηριστικά του $P(z)$, προσαρμόζοντας το $W(z)$ ώστε να ελαχιστοποιείται η διαφορά των δύο εξόδων. Τα φίλτρο $W(z)$ μπορεί να είναι μόνο με πόλους, μόνο με μηδενικά ή μπορεί να περιέχει και πόλους και μηδενικά. Επιπλέον ο προσαρμοστικός αλγόριθμος μπορεί να είναι οποιοσδήποτε έγκυρος προσαρμοστικός αλγόριθμος. Για το σκοπό μας, αλλά και για να διατηρήσουμε τη συνοχή με την ανάλυση των προηγούμενων ενοτήτων, θεωρούμε το $W(z)$ ότι είναι ένα FIR LMS προσαρμοζόμενο φίλτρο.

Αφότου ο LMS αλγόριθμος έχει συγκλίνει, το προσαρμοζόμενο φίλτρο θα είναι σε θέση να «μιμηθεί» με ακρίβεια τη συμπεριφορά του φυσικού συστήματος, με την προϋπόθεση ότι η τάξη του προσαρμοζόμενου φίλτρου είναι ίδια ή μεγαλύτερη από την τάξη του φυσικού συστήματος. Συνεπώς, οι συντελεστές του προσαρμοζόμενου φίλτρου θα περιέχουν έπειτα από τη σύγκλισή τους ικανοποιητικές πληροφορίες για τα χαρακτηριστικά του φυσικού συστήματος και ως αποτέλεσμα αυτό το δεύτερο θα έχει αναγνωρισθεί.

Δημιουργούμε τώρα ένα περιβάλλον προσομοίωσης για τη διάταξη της προσαρμοζόμενης αναγνώρισης συστήματος της εικόνας 2.46, χρησιμοποιώντας ένα LMS προσαρμοζόμενο φίλτρο.

Το FIR φίλτρο των εικόνων 2.27 και 2.28 θα παίζει το ρόλο του $P(z)$, το οποίο θέλουμε να αναγνωρίσουμε.

Η αναγνώριση θα γίνει από ένα LMS προσαρμοζόμενο φίλτρο με 10 συντελεστές (ίδια τάξη με το $P(z)$).

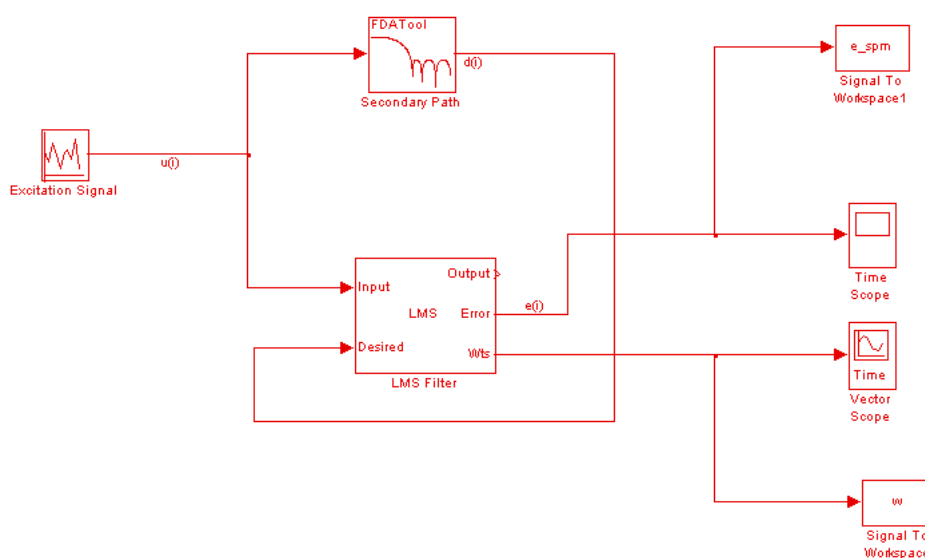
Ως σήμα διέγερσης θα χρησιμοποιήσουμε λευκό θόρυβο διασποράς 1, προκειμένου να αναγνωρίσουμε το σύστημα σε όλο το φάσμα του. Ο λευκός θόρυβος θα έχει μέση τιμή μηδέν και συνεπώς η ισχύς του θα είναι ίση με τη διασπορά το, εξαιτίας της πολύ γνωστής σχέσης

$$\text{Διασπορά}_x = E(x - \bar{x})^2 = E x^2 - \bar{x}^2, \quad (2.61)$$

Από την αναθεωρημένη συνθήκη ευστάθειας, το μήκος βήματος επιλέγεται ως εξής:

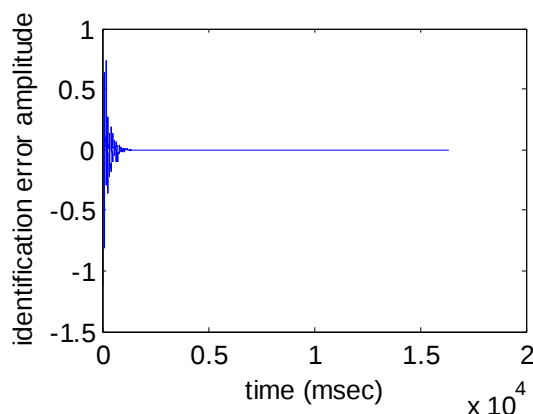
$$\frac{0.05}{LP_u} = \frac{0.05}{10 * 1} = 0.005, \quad (2.62)$$

Το αντίστοιχο μοντέλο Simulink είναι: [Simulink Models\adaptive_system_identification.mdl](#). Το μπλοκ διάγραμμα του είναι



Εικόνα 2.47 μπλοκ διάγραμμα του Simulink μοντέλου προσαρμοζόμενης αναγνώρισης συστήματος

Το σφάλμα αναγνώρισης, το οποίο είναι το σφάλμα του προσαρμοζόμενου φίλτρου, είναι

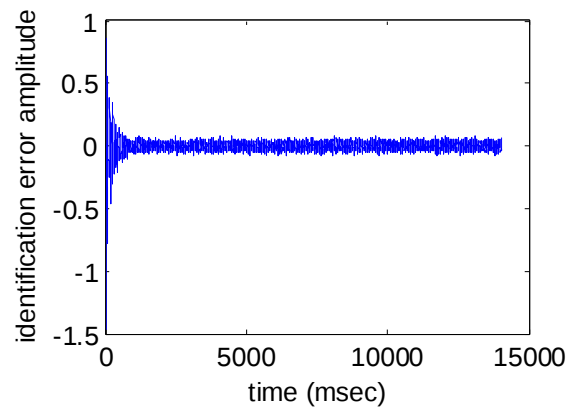


Εικόνα 2.48 σφάλμα αναγνώρισης (τάξη προσαρμοζόμενου φίλτρου=τάξη φυσικού συστήματος)

Από την εικόνα 2.48 βλέπουμε ότι τελικά η αναγνώριση είναι τέλεια. Εξέταση των βαρών του προσαρμοζόμενου φίλτρου έπειτα από τη σύγκλιση τους, δείχνει ότι είναι ακριβώς οι ίδιοι με τους συντελεστές του «φυσικού συστήματος».

Για λόγους κατανόησης τρέχουμε την ίδια προσομοίωση, αλλά έχοντας μειώσει την τάξη του προσαρμοζόμενου φίλτρου σε 9.

Το σφάλμα αναγνώρισης είναι

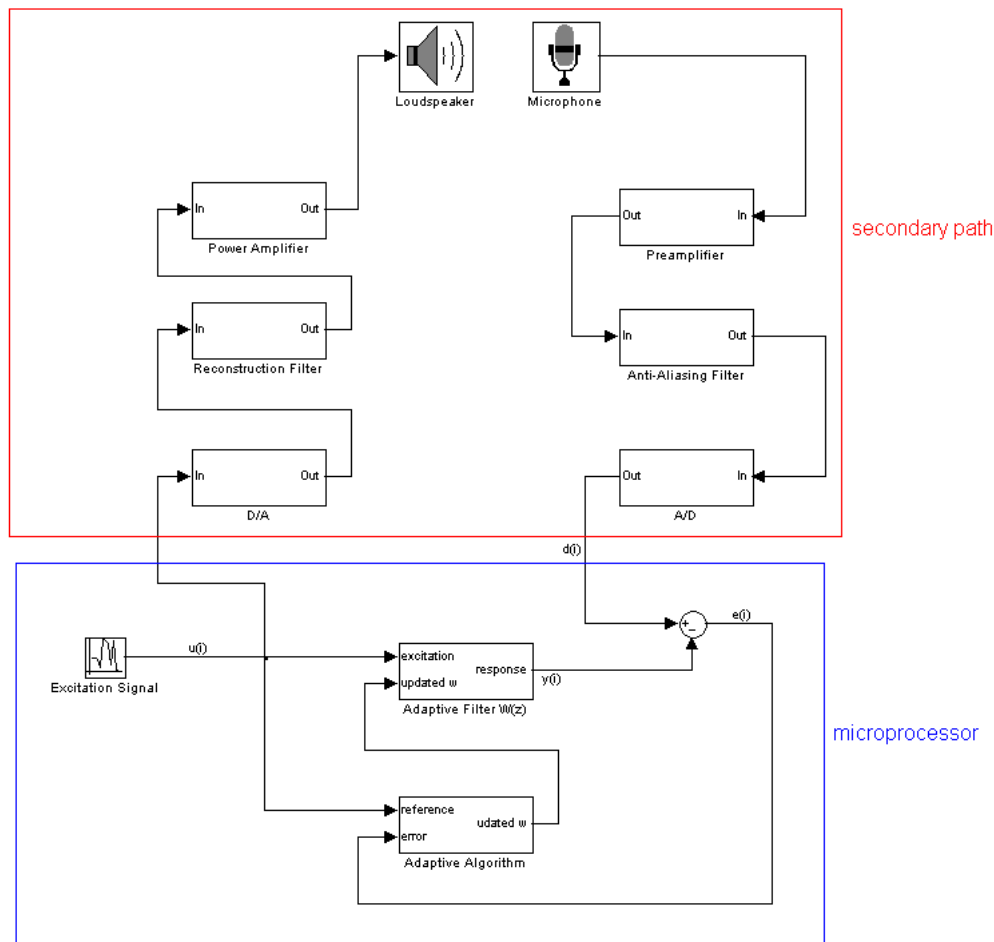


Εικόνα 2.49 σφάλμα αναγνώρισης (τάξη του προσαρμοζόμενου φίλτρου=τάξη του φυσικού συστήματος-1)

Όπως ήταν αναμενόμενο, η αναγνώριση δεν είναι τέλεια. Φυσικά, ο λόγος είναι ότι η τάξη του προσαρμοζόμενου φίλτρου (9) είναι μικρότερη από την τάξη του φυσικού συστήματος (10) σε συνδυασμό με το γεγονός ότι το σήμα διέγερσης είναι ο λευκός θόρυβος, το οποίο σημαίνει ότι αναγνώριση επί όλου του φάσματος επιχειρήθηκε.

Σε αυτό το σημείο είναι πολύ ενδιαφέρον να θυμηθούμε τη συζήτηση, η οποία ακολούθησε τις εξισώσεις (4.2.98)-(4.2.102), σχετικά με το πού βρίσκεται το νόημα στο να προσπαθούμε να προβλέψουμε το $d(i)$ χρησιμοποιώντας ένα προσαρμοζόμενο φίλτρο, εάν ήδη γνωρίζουμε το $d(i)$. Αναφέραμε σύντομα, αλλά χωρίς καμία επιπρόσθετη εξήγηση, εφόσον το υπόβαθρο ήταν ημιτελές, ότι μπορεί να μην ενδιαφερόμαστε κατά κύριο λόγο στην έξοδο του προσαρμοζόμενου φίλτρου, η οποία είναι όντως μία πρόβλεψη του $d(i)$, αλλά μπορεί να ενδιαφερόμαστε στην πληροφορία που έχει αποθηκευτεί στο w έως τη στιγμή που η έξοδος του προσαρμοζόμενου φίλτρου είναι μία ικανοποιητική πρόβλεψη του $d(i)$! Λοιπόν, η προσαρμοζόμενη αναγνώριση συστήματος είναι ένα τέτοιο παράδειγμα! Πράγματι, δεν ενδιαφερόμαστε για το ίδιο το $d(i)$, αυτό, στο οποίο ενδιαφερόμαστε είναι ένα μοντέλο του φυσικού συστήματος, το οποίο παράγει το $d(i)$. Αλλά εάν χρησιμοποιήσουμε ένα προσαρμοζόμενο φίλτρο όπως στην εικόνα 2.46, τότε η έξοδός του θα υποχρεωθεί να είναι μία πρόβλεψη του $d(i)$, όπως ήδη έχουμε δείξει. Παρ'όλα αυτά, η πληροφορία, την οποία αναζητούμε, βρίσκεται απλώς στους συντελεστές του προσαρμοζόμενου φίλτρου, τους οποίους μπορούμε να χρησιμοποιήσουμε ώστε να προσεγγίσουμε την απόκριση συχνότητας του φυσικού συστήματος.

Εφαρμόζοντας την ιδέα της εικόνας 2.46 για την αναγνώριση ενός πραγματικού secondary path, το διάγραμμα της πειραματικής διάταξης θα έχει ως εξής:



Εικόνα 2.50 προσαρμοζόμενη αναγνώριση secondary path σε πραγματικό χρόνο

2.8 Διέγερση του Secondary Path χρησιμοποιώντας Chirp Σήματα

Η ύπαρξη του secondary path επιδρά δραματικά στην απόδοση ενός ANC ελεγκτή. Εάν το secondary path δεν υπήρχε, η εξασθένιση του θορύβου θα ήταν ιδανική και πολλά προβλήματα ευστάθειας, τα οποία είναι συχνά αξεπέραστα, δε θα εμφανίζονταν ποτέ. Συνεπώς η επιτυχής αναγνώριση του secondary path είναι ζωτικής σημασίας. Αλλά ας ξεκαθαρίσουμε πρώτα τι ακριβώς σημαίνει επιτυχής αναγνώριση, όσον αφορά την αναγνώριση του secondary path.

Το πρώτο βήμα στο σχεδιασμό της αναγνώρισης του secondary path, είναι ο καθορισμός της ζώνης συχνοτήτων ενδιαφέροντος. Για παράδειγμα, εάν ήδη γνωρίζουμε ότι ο κύριος θόρυβος προς εξασθένιση έχει την ισχύ του συγκεντρωμένη σε μία συγκεκριμένη ζώνη συχνοτήτων, τότε απαιτείται να αναγνωρίσουμε το secondary path μόνο σε αυτή τη ζώνη. Έτσι, εάν γνωρίζουμε ότι το φάσμα του κύριου θορύβου είναι μη μηδενικό από 200Hz έως 300Hz, τότε πρέπει να διεγείρουμε το secondary path με συχνότητες από 100Hz έως 400Hz. Οι επιπλέον συχνότητες διέγερσης (100Hz-200Hz και 300Hz-400Hz) απαιτούνται για λόγους ασφάλειας. Για να το δούμε αυτό, υποθέτουμε ότι η $S(z)$ έχει αναγνωριστεί μόνο στη ζώνη 200Hz-300Hz, η οποία συμπίπτει με τη ζώνη ενδιαφέροντος. Τώρα, εάν μία μικρή διαταραχή εμφανιστεί στα 310Hz, το οποίο είναι αρκετά πιθανό να συμβεί εξαιτίας των μη γραμμικοτήτων του συστήματος, τότε η μη αναγνώριση της $S(z)$ στα 310Hz ενδέχεται να οδηγήσει σε θετική ανάδραση σε αυτήν τη συχνότητα και συνεπώς σε έναν συνεχώς αυξανόμενο θόρυβο, ο οποίος μπορεί να τραυματίσει τυχόν παριστάμενους ανθρώπους και να καταστρέψει τον εξοπλισμό.

Έπειτα από την επιλογή της ζώνης συχνοτήτων, στην οποία θα διεγερθεί η $S(z)$, το επόμενο βήμα είναι να επιλέξουμε ένα κατάλληλο σήμα διέγερσης, το οποίο θα περιέχει αυτήν τη ζώνη. Σε αυτήν την εργασία προτιμούμε τη χρήση chirp σημάτων για λόγους που θα εξηγηθούν αμέσως παρακάτω.

Ένα γραμμικό chirp είναι μία κυματομορφή, της οποίας η στιγμιαία συχνότητα αυξάνεται γραμμικά με το χρόνο μεταξύ δύο συγκεκριμένων συχνοτήτων. Μπορείτε να ακούσετε ένα γραμμικό chirp ακριβώς εδώ: [Audio Files\example_chirp_100Hz_to_500Hz.wav](#). Τα chirp σήματα χρησιμοποιούνται σε ηχητικές και ραντάρ εφαρμογές, ενώ φάλαινες, νυχτερίδες και άλλα πτηνά είναι γνωστό ότι χρησιμοποιούν chirp σήματα ως σήματα επικοινωνίας. Τα choirs είναι επαναλαμβανόμενα γραμμικά chirp σήματα. Μπορείτε να ακούσετε ένα choir σήμα ακριβώς εδώ: [Audio Files\example_choir_100Hz_to_500Hz.wav](#)

Τα chirp σήματα είναι ευρέως γνωστά εξαιτίας δύο σημαντικών πλεονεκτημάτων όσον αφορά την αναγνώριση συστήματος έναντι άλλων κυματομορφών, όπως λευκού θορύβου, κρουστικών σημάτων, ημιτόνων και μέγιστου-μήκους-ακολουθιών. Πρώτον, τα chirps έχουν τον ελάχιστο δυνατό *peak factor* ή αλλιώς *crest factor*. Η ελάχιστη αυτή τιμή είναι το $\sqrt{2}$. Ο peak factor μίας κυματομορφής είναι

$$peak\ factor = \frac{peak\ value}{rms}, \quad (2.63)$$

Οι κυματομορφές με χαμηλό peak factor έχουν την ιδιότητα να επιτυγχάνουν το μέγιστο δυνατό SNR. Το δεύτερο πλεονέκτημα των chirps είναι ότι η ζώνη ισχύος τους, δηλαδή η ζώνη συχνοτήτων, στην οποία είναι συγκεντρωμένη όλη η ισχύς τους, δύναται να καθοριστεί πλήρως. Όταν μία τέτοια κυματομορφή χρησιμοποιείται για αναγνώριση συστήματος, ο απαιτούμενος χρόνος μέτρησης μπορεί να είναι πολύ μικρότερος από το χρόνο, ο οποίος θα απαιτούταν για επαναλαμβανόμενες μετρήσεις χρησιμοποιώντας άλλες κυματομορφές. Για παράδειγμα, η αναγνώριση του secondary path με χρήση λευκού θορύβου θα απαιτούσε πολλαπλές επαναλήψεις έτσι ώστε να επιτευχθεί στατιστική αξιοπιστία, ενώ το φάσμα διέγερσης θα περιείχε αναγκαστικά όλες τις συχνότητες.

Το πλάτος ενός chirp διακριτού χρόνου είναι

$$Acos[\phi(i)]$$

όπου $\phi(i)$ είναι η στιγμιαία φάση του chirp και ισούται με

$$\phi(i) = a + bi + ci^2, \quad (2.64)$$

Η τετραγωνική φάση $\phi(i)$ μπορεί να εκφραστεί σε μία μορφή χρήσιμη για σχεδιασμό ενός chirp, δηλαδή

$$\phi(i) = 2\pi \left\{ f_{ln}i + \frac{(f_{hn} - f_{ln})}{2N} i^2 \right\} + a, \quad (2.65)$$

όπου a είναι ένας σταθερός όρος φάσης και f_{ln} , f_{hn} είναι η κανονικοποιημένη χαμηλότερη και υψηλότερη αντίστοιχα συχνότητα της επιθυμητής ζώνης συχνοτήτων, δηλαδή

$$f_{ln} = \frac{f_l}{F_s} \quad \text{and} \quad f_{hn} = \frac{f_h}{F_s}, \quad (2.66)$$

όπου F_s είναι η συχνότητα δειγματοληψίας του chirp διακριτού χρόνου. Η στιγμιαία συχνότητα του chirp είναι

$$f(i) = f_{ln} + \frac{(f_{hn} - f_{ln})}{N} i, \quad (2.67)$$

σε αντιστοιχία με τη σχέση συνεχούς χρόνου

$$f(t) = \frac{1}{2\pi} \frac{d\phi(t)}{dt}, \quad (2.68)$$

Από την (2.67) βλέπουμε ότι

$$f(0) = f_{ln} \quad \text{και} \quad f(N) = f_{hn}, \quad (2.69)$$

Η εξίσωση (2.69) δείχνει ότι ένα chirp σαρώνει με το χρόνο μία προκαθορισμένη ζώνη συχνοτήτων $[f_{ln}, f_{hn}]$. Αλλά εφόσον το chirp που εξετάζουμε είναι ένα chirp διακριτού χρόνου, αυτή η ζώνη συχνοτήτων σαρώνεται αναγκαστικά με έναν ασυνεχή

τρόπο, όπως εύκολα φαίνεται από την (2.67). Συνεπώς, το ερώτημα που ανακύπτει είναι «ποιά είναι η ανάλυση αυτής της σάρωσης»?

Με μία πρώτη ματιά, από την (2.67) η ανάλυση της συχνότητας είναι

$$\Delta f_1 = \frac{(f_h - f_l)}{N}, \quad (2.70)$$

Παρόλα αυτά, εάν η συχνότητα δειγματοληψίας είναι F_s , τότε το chirp δε δύναται να περιέχει συχνότητες, οι οποίες είναι πλησιέστερα η μία στην άλλη από

$$\Delta f_2 = \frac{F_s}{N}, \quad (2.71)$$

Συγκρίνοντας τις (2.70) και (2.71), βλέπουμε ότι για να έχει ένα chirp νόημα πρέπει

$$\Delta f_1 < \Delta f_2, \quad (2.72)$$

Η συνέπεια της (2.72) είναι ότι οι συχνότητες, οι οποίες δίνονται από την (2.67) δεν είναι όλες τους διακριτές η μία από την άλλη. Από τις (2.70) και (2.71) μπορούμε να ορίσουμε το Q ως το πηλίκο

$$Q = \frac{\Delta f_2}{\Delta f_1} \xrightarrow{(2.70),(2.71)}$$

$$Q = \frac{F_s}{(f_h - f_l)} \gg 1, \quad (2.73)$$

το οποίο υπονοεί ότι το chirp θα περιέχει μόνο $\frac{N}{Q}$ συχνότητες (και όχι N , όπως η (2.67) υπονοεί), δηλαδή

$$\text{αριθμός αρμονικών σε ένα chirp} = N(f_{hn} - f_{ln}), \quad (2.74)$$

Οι διακριτές, κανονικοποιημένες συχνότητες δίνονται από μία τροποποίηση της (2.67), δηλαδή

$$f(\lambda) = f_{ln} + \frac{(f_{hn} - f_{ln})}{N} \lambda,$$

$$\lambda = \text{πηλίκο της Ευκλείδειας διαίρεσης } i:Q,$$

$$i = 0, 1, \dots, N, \quad (2.75)$$

με Q συχνότητες της (2.67) να αντιστοιχούν σε μία μοναδική συχνότητα της (2.75). Όσον αφορά την ανάλυση συχνότητας οποιουδήποτε chirp, αυτή είναι

$$\text{ανάλυση συχνότητας οποιουδήποτε chirp} = \frac{F_s}{N}, \quad (2.76)$$

Επιτέον κάθε μία από τις συχνότητες, οι οποίες εμπεριέχονται σε ένα chirp, και δίνονται από την (2.75), θα έχει διάρκεια ίση με

$$\text{διάρκεια οποιασδήποτε συχνότητας ενός οποιουδήποτε γραμμικού chirp} = QT_s = \frac{1}{(f_h - f_l)}, \quad (2.77)$$

Ας σχεδιάσουμε τώρα ένα σήμα choir, το οποίο θα χρησιμοποιήσουμε για να διεγείρουμε το secondary path των εικόνων 2.27 και 2.28, ενώ παράλληλα θα αναδείξουμε τη σημασία των παραπάνω εξισώσεων.

Υποθέτουμε ότι ο κύριος θόρυβος είναι το “s.o.t.w.” σήμα, του οποίου το φάσμα φαίνεται στην εικόνα 2.6. Βλέπουμε ότι η ζώνη ενδιαφέροντος είναι από 200Hz έως 300Hz, και συνεπώς το secondary path πρέπει να αναγνωρισθεί στη ζώνη συχνοτήτων από 100Hz έως 400Hz. Σήμα διέγερσης θα είναι ένα επαναλαμβανόμενο chirp με ζώνη ισχύος από 100Hz έως 400Hz. Συνεπώς

$$f_l = 100\text{Hz} \quad \text{και} \quad f_h = 400\text{Hz}, \quad (2.78)$$

Μία βολική επιλογή για τη συχνότητα δειγματοληψίας είναι

$$F_s = 2000\text{Hz}, \quad (2.79)$$

Από τις (2.78) και (2.79) έχουμε

$$f_{ln} = 0.05 \quad \text{και} \quad f_{hn} = 0.2, \quad (2.80)$$

Στη συνέχεια, θέλουμε το chirp μας να έχει ανάλυση συχνότητας 2Hz ή ισοδύναμα να περιέχει $\frac{400-100}{2} = 150$ αρμονικές, δηλαδή

$$\text{επιθυμητός αριθμός αρμονικών} = 150, \quad (2.81)$$

$\Leftrightarrow$

$$\text{ανάλυση συχνότητας} = 2\text{Hz}, \quad (2.82)$$

Από τις (2.74) και (2.80)

$$N = \frac{150}{0.2 - 0.05} = 1000, \quad (2.83)$$

δηλαδή 1000 δείγματα από το chirp απαιτούνται. Προκειμένου να δείξουμε τη συνοχή της ανάλυσής μας, σημειώνουμε ότι σύμφωνα με την (2.76) η ανάλυση συχνότητας είναι

$$\frac{F_s}{N} = \frac{2000\text{Hz}}{1000} = 2\text{Hz}, \quad (2.84)$$

το οποίο συμφωνεί με την (2.82).

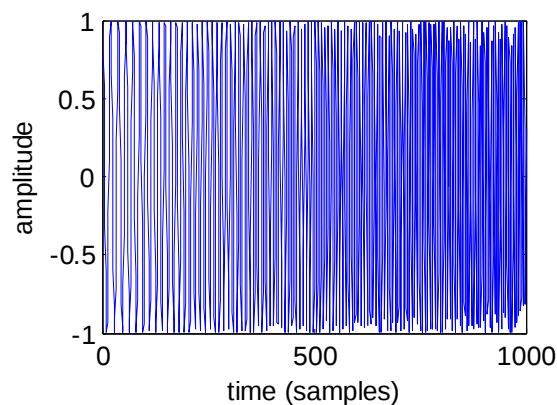
Από την (2.65), η φάση του chirp θα είναι

$$\varphi(i) = 2\pi\{0.05i + 0.000075i^2\}, \quad (2.85)$$

όπου η σταθερά φάσης a επιλέχθηκε ως μηδέν για διευκόλυνση. Το chirp σήμα μας είναι τελικά

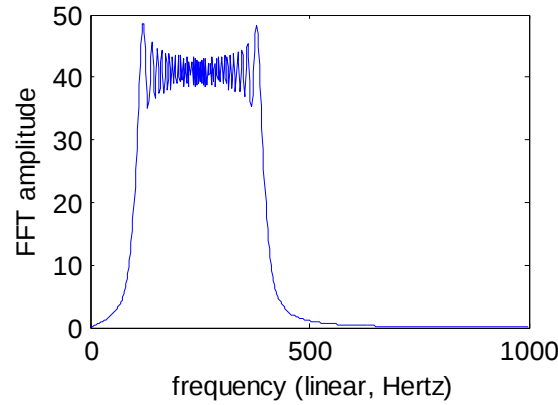
$$x(i) = \cos[\varphi(i)], \quad (2.86)$$

και η κυματομορφή του είναι



Εικόνα 2.51 κυματομορφή του chirp σήματος $\cos[2\pi\{0.05i + 0.000075i^2\}]$

Το φάσμα του chirp είναι



Εικόνα 2.52 φάσμα του chirp σήματος $\cos[2\pi\{0.05i + 0.000075i^2\}]$

Η διάρκεια της κάθε αρμονικής του chirp μας είναι από τις (2.77) και (2.78)

$$\text{διάρκεια της κάθε αρμονικής} = \frac{1}{400 - 100} \approx 0.0033\text{sec}, \quad (2.87)$$

Τώρα, αξίζει να δώσουμε προσοχή σε μία εξαιρετικά σημαντική λεπτομέρεια, η οποία πρέπει να ληφθεί σοβαρά υπόψιν, προκειμένου η προσαρμοζόμενη αναγνώριση του secondary path να είναι επιτυχής: Εφόσον η περίοδος δειγματοληψίας είναι $\frac{1}{2000\text{Hz}} = 0.0005\text{sec}$, η διάρκεια κάθε αρμονικής είναι $\frac{0.0033}{0.0005} \approx 6$ δείγματα. Αλλά θυμόμαστε ότι οι συντελεστές $\hat{s}$ του προσαρμοζόμενου φίλτρου $\hat{S}(z)$, το οποίο υλοποιεί την αναγνώριση του secondary path $S(z)$, αναβαθμίζονται δείγμα προς δείγμα, δηλαδή μία φορά για κάθε περίοδο δειγματοληψίας. Αυτό σημαίνει ότι εάν χρησιμοποιήσουμε το παραπάνω chirp για να αναγνωρίσουμε το $S(z)$, τότε το προσαρμοζόμενο φίλτρο θα έχει στη διάθεσή του 6 βήματα προκειμένου να προσαρμοστεί στην τρέχουσα αρμονική f_λ , πρώτου η επόμενη αρμονική του chirp $f_{\lambda+1}$ αφιχθεί. Πιο συγκεκριμένα, κατά τη διάρκεια αυτών των 6 βημάτων, οι συντελεστές του προσαρμοζόμενου φίλτρου θα πλησιάσουν στον M-διάστατο διανυσματικό χώρο (θυμηθείτε ότι M είναι η τάξη του προσαρμοζόμενου φίλτρου $\hat{S}(z)$) τον βέλτιστο αναγνωριστή στη συχνότητα f_λ του $S(z)$, δηλαδή τον $\widehat{s}_\lambda^o$, και στο τέλος του έκτου βήματος η τιμή τους θα είναι $\widehat{s}_\lambda$. Όταν η επόμενη αρμονική του chirp εμφανισθεί, οι συντελεστές του προσαρμοζόμενου φίλτρου θα κινηθούν προς το βέλτιστο αναγνωριστή του secondary path στη συχνότητα όμως $f_{\lambda+1}$, δηλαδή προς τον $\widehat{s}_{\lambda+1}^o$, και στο τέλος του έκτου βήματος η τιμή τους θα είναι $\widehat{s}_{\lambda+1}$. Τώρα, εξαιτίας της ιδιότητας ελάχιστης διαταραχής του LMS αλγορίθμου και υπό την προϋπόθεση ότι το μήκος βήματος είναι μικρό, η συχνότητα $f_{\lambda+1}$ θα διαταράξει στον ελάχιστο δυνατό βαθμό την τιμή $\widehat{s}_\lambda$, και ως αποτέλεσμα το $\hat{s}$ θα κινηθεί προς το $\widehat{s}_{\lambda+1}^o$, ενώ ταυτόχρονα παραμένει όσο πιο κοντά γίνεται στην τιμή $\widehat{s}_\lambda$. Αυτό, το οποίο θέλουμε, είναι μετά την πλήρη σάρωση της ζώνης συχνοτήτων από το chirp, το $\hat{s}$ να είναι όσο το δυνατόν πιο κοντά, με την έννοια της Ευκλείδειας νόρμας, σε όλα τα $\widehat{s}_\lambda^o \forall \lambda$. Παρατηρούμε ότι εάν το μήκος βήματος είναι πολύ μεγάλο, το $\hat{s}$ θα ταλαντώνεται μεταξύ όλων των $\widehat{s}_\lambda^o \forall \lambda$, καθώς κατά τη διάρκεια των 6 διαθέσιμων βημάτων το $\hat{s}$ θα συγκλίνει στην κοντινή περιοχή του αντίστοιχου βέλτιστου αναγνωριστή της $S(z)$, και ως αποτέλεσμα η $S(z)$ θα είναι αδύνατο να αναγνωριστεί σε όλες τις επιθυμητές συχνότητες $\{f_\lambda\}$.

Προκειμένου να αποφύγουμε αυτό το πρόβλημα, η τάξη του προσαρμοζόμενου φίλτρου και το μήκος βήματός του θα πρέπει να επιλεγούν καταλλήλως έπειτα από μία trial-and-error διαδικασία. Ένας καλός τρόπος για να επιβεβαιώσουμε ότι το προσαρμοζόμενο φίλτρο έχει αναγνωρίσει το $S(z)$ σε όλες τις συχνότητες, οι οποίες εμπεριέχονται στο chirp, είναι να παρατηρήσουμε τους συντελεστές του, καθώς αυτοί προσαρμόζονται για να ελαχιστοποιήσουν το σφάλμα αναγνώρισης και να παρατηρήσουμε στη μόνιμη κατάσταση εάν αυτοί ταλαντώνονται ή όχι. Εάν οι συντελεστές ταλαντώνονται και στη μόνιμη κατάσταση ισορροπίας, πρέπει να μειώσουμε το μήκος βήματος και να ελέγξουμε και πάλι για ταλαντώσεις. Συνήθως, όσο μικρότερο είναι το μήκος βήματος, τόσο καλύτερη αλλά και τόσο πιο αργή είναι η διαδικασία της αναγνώρισης.

Είναι χρήσιμο να παρατηρήσουμε σε αυτό το σημείο τη λειτουργική διαφορά μεταξύ του μήκους βήματος του προσαρμοζόμενου φίλτρου, το οποίο εκτελεί την εξασθένιση θορύβου, και του προσαρμοζόμενου φίλτρου, το οποίο εκτελεί την αναγνώριση του secondary path. Από τη μία πλευρά επιλέγουμε το μήκος βήματος του ANC ελεγκτή όσο πιο μεγάλο γίνεται, προκειμένου να επιτύχουμε γρήγορη σύγκλιση και ταχεία ικανότητα παρακολούθησης. Από την άλλη πλευρά, επιλέγουμε το μήκος βήματος του αναγνωριστή να είναι αρκετά μικρό έτσι ώστε να επιτύχουμε μεγάλη ικανότητα καθολικής εκμάθησης (δηλαδή αναγνώριση σε όλες τις συχνότητες).

Μία τελική παρατήρηση αφορά τη συχνότητα δειγματοληψίας. Είναι ένας γενικός κανόνας της ψηφιακής επεξεργασίας σημάτων, ότι η συχνότητα δειγματοληψίας πρέπει να είναι όσο το δυνατόν μικρότερη. Για παράδειγμα, η τάξη ενός ψηφιακού φίλτρου, το οποίο εκτελεί ένα συγκεκριμένο φιλτράρισμα, αυξάνεται όταν η συχνότητα δειγματοληψίας αυξάνεται. Αυτό βεβαίως ισχύει και για τα προσαρμοζόμενα φίλτρα, τα οποία είναι εξίσου ψηφιακά φίλτρα. Υπενθυμίζουμε ότι στην περίπτωση του LMS προσαρμοζόμενου φίλτρου, το οποίο εμείς χρησιμοποιούμε τόσο για εξασθένιση θορύβου όσο και για αναγνώριση του secondary path, το μήκος του, δηλαδή ο αριθμός των συντελεστών του, πρέπει να είναι ένα ικανοποιητικό ποσοστό της περιόδου του σήματος αναφοράς. Έτσι, εάν ο θόρυβος προς εξασθένιση έχει συγκεντρωμένη την ισχύ του στη ζώνη 200Hz έως 300Hz, και εάν η ελάχιστη τάξη φίλτρου, το οποίο τον εξασθενεί ικανοποιητικά, είναι 10 συντελεστές στα 2000Hz, τότε εάν εμείς αυξήσουμε τη συχνότητα δειγματοληψίας στα 8000Hz, η ελάχιστη τάξη προσαρμοζόμενου φίλτρου, το οποίο επιτυγχάνει την ίδια αποδόση θα είναι $10 * \frac{8000}{2000} = 40$ συντελεστές. Σε ένα περιβάλλον προσομοίωσης, όπως είναι το Simulink, η συχνότητα δειγματοληψίας μπορεί να επιλεγεί ελεύθερα. Αντιθέτως, σε εφαρμογές πραγματικού χρόνου η ελάχιστη συχνότητα δειγματοληψίας καθορίζεται από τις δυνατότητες του υλικού, το οποίο χρησιμοποιούμε.

Δημιουργούμε τώρα ένα περιβάλλον προσομοίωσης για να αναγνωρίσουμε το secondary path των εικόνων 2.27 και 2.28 στη ζώνη [100Hz, 400Hz], χρησιμοποιώντας ως σήμα διέγερσης κυκλικές επαναλήψεις του chirp σήματος, το οποίο σχεδιάσαμε στο προηγούμενο εδάφιο, δηλαδή

$$\text{σήμα διέγερσης} = \text{κυκλικές επαναλήψεις του } \cos[2\pi\{0.05i + 0.000075i^2\}], \quad (2.88)$$

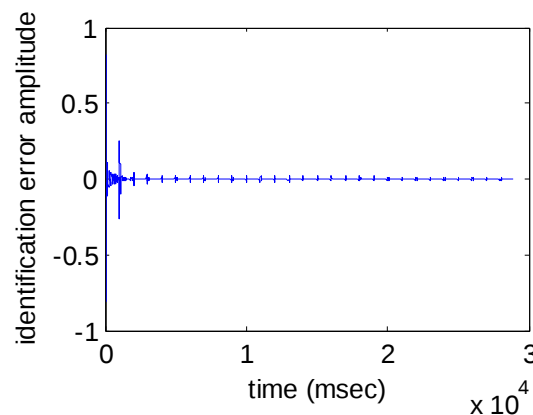
Η ισχύς του σήματος αυτού είναι 0.5.

Η τάξη του αναγνωριστή είναι 10.

Συνεπώς μία καλή επιλογή για το μήκος βήματος του αναγνωριστή είναι 0.01.

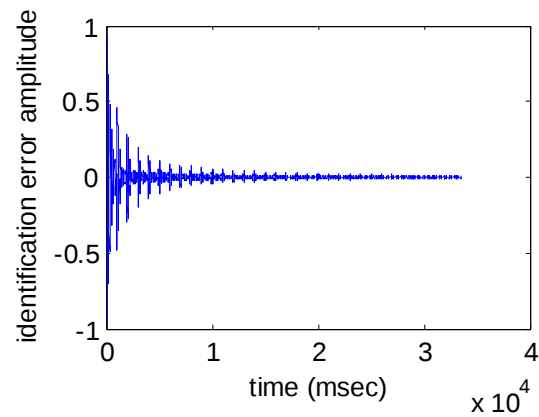
Το αντίστοιχο μοντέλο Simulink είναι: [Simulink Models\adaptive_system_identification.mdl](#).

Το σφάλμα αναγνώρισης είναι



Εικόνα 2.53 σφάλμα αναγνώρισης του secondary path χρησιμοποιώντας κυκλικές επαναλήψεις του $\cos[2\pi\{0.05i + 0.000075i^2\}]$ ($\mu=0.01$, τάξη φίλτρου=10)

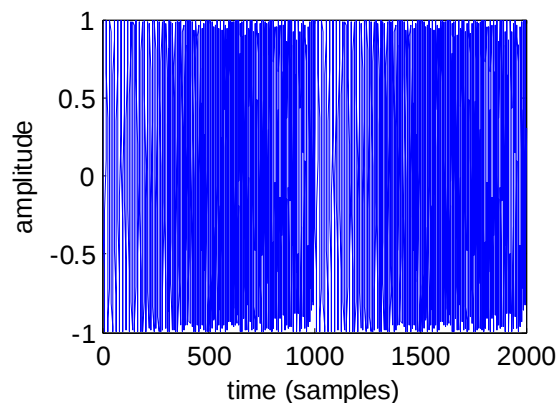
Εάν το μήκος βήματος μειωθεί σε 0.001 το σφάλμα αναγνώρισης είναι



Εικόνα 2.54 σφάλμα αναγνώρισης του *secondary path* χρησιμοποιώντας κυκλικές επαναλήψεις του $\cos[2\pi\{0.05i + 0.000075i^2\}]$ ($\mu=0.001$, τάξη φίλτρου=10)

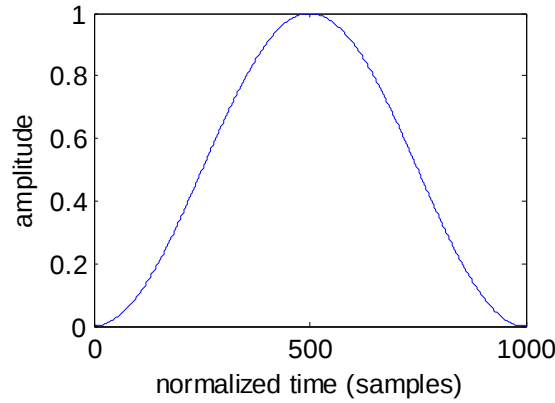
Παρατηρούμε ότι το σφάλμα αναγνώρισης της εικόνας 2.54 είναι αφενός μεγαλύτερο από αυτό της 2.53 αλλά αφεαίρου ομαλότερο. Το γεγονός ότι είναι ομαλότερο, δηλαδή ότι δεν παρουσιάζονται κορυφές σφάλματος αναγνώρισης, μεταφράζεται στο γεγονός ότι οι συντελεστές του αναγνωριστή είναι περισσότερο εντοπισμένοι από τους συντελεστές του αναγνωριστή της εικόνας 2.53, και συνεπώς η αναγνώριση είναι περισσότερο καθολική, δηλαδή εκτείνεται σε ένα ευρύτερο φάσμα συχνοτήτων.

Παρ' όλα αυτά, στην περίπτωση της αναγνώρισης του *secondary path* σε πραγματικό χρόνο, η ασυνέχεια στη σάρωση συχνοτήτων, η οποία συμβαίνει στο τέλος κάθε περιόδου του *choir*, ενδέχεται να καταστρέψει το ηχείο. Αυτή η ασυνέχεια φαίνεται στην παρακάτω εικόνα:

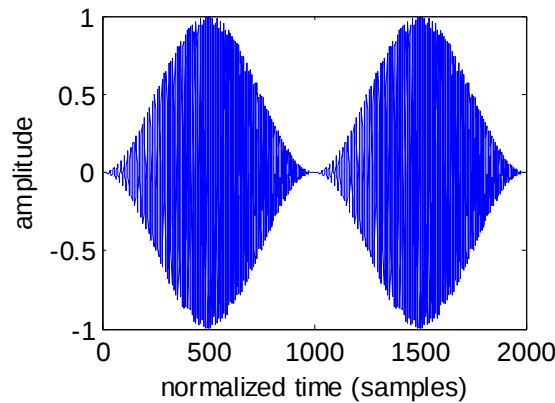


Εικόνα 2.55 δύο περίοδοι του $\text{chirp } \cos[2\pi\{0.05i + 0.000075i^2\}]$

Ένας τρόπος να αντιμετωπισθεί αυτό το πρόβλημα είναι να παραθυρώσουμε το chirp της εξίσωσης (2.88) μέσω ενός παραθύρου Hanning. Το αποτέλεσμα φαίνεται στις παρακάτω δύο εικόνες:



Εικόνα 2.56 παράθυρο Hanning 1000 δειγμάτων



Εικόνα 2.57 δύο περίοδοι του Hanning παραθυρωμένου chirp $\{\cos[2\pi\{0.05i + 0.000075i^2\}]\}$

Το choir της εικόνας 2.57 είναι πολύ ομαλότερο από το choir της εικόνας 2.55.

2.9 Ο FULMS Feedback Ελεγκτής Ενεργού Θορύβου με off-line Αναγνώριση του Secondary Path

Είμαστε τώρα έτοιμοι να αναπτύξουμε ένα ακόμα περισσότερο ρεαλιστικό μοντέλο ενός ελεγκτή ενεργού θορύβου. Ο ελεγκτής θα αναγνωρίζει με προσαρμοζόμενο τρόπο το secondary path, παράγοντας μία προσέγγιση $\hat{s}$ για αυτό, και θα χρησιμοποιεί αυτήν την προσέγγιση για να φιλτράρει την έξοδο του $W(z)$ αλλά και το σήμα αναφοράς, ακριβώς όπως απαιτείται από το βελτιωμένο FULMS αλγόριθμο. Αυτός ο ελεγκτής είναι εύστοχο να αποκαλείται *FULMS Feedback Ελεγκτής Ενεργού Θορύβου με off-line αναγνώριση του Secondary Path* (η αναγνώριση του secondary path είναι πράγματι off-line για λόγους που θα αναφερθούν λείαν συντόμως) και οι εξισώσεις, οι οποίες πλήρως περιγράφουν τη λειτουργία του είναι

$$y(i) = \mathbf{w}_{i-1}^T \mathbf{u}(i-1), \quad (2.88)$$

$$u(i) = d(i) + \hat{s}^T \mathbf{y}(i) - \mathbf{s}^T \mathbf{y}(i), \quad (2.89)$$

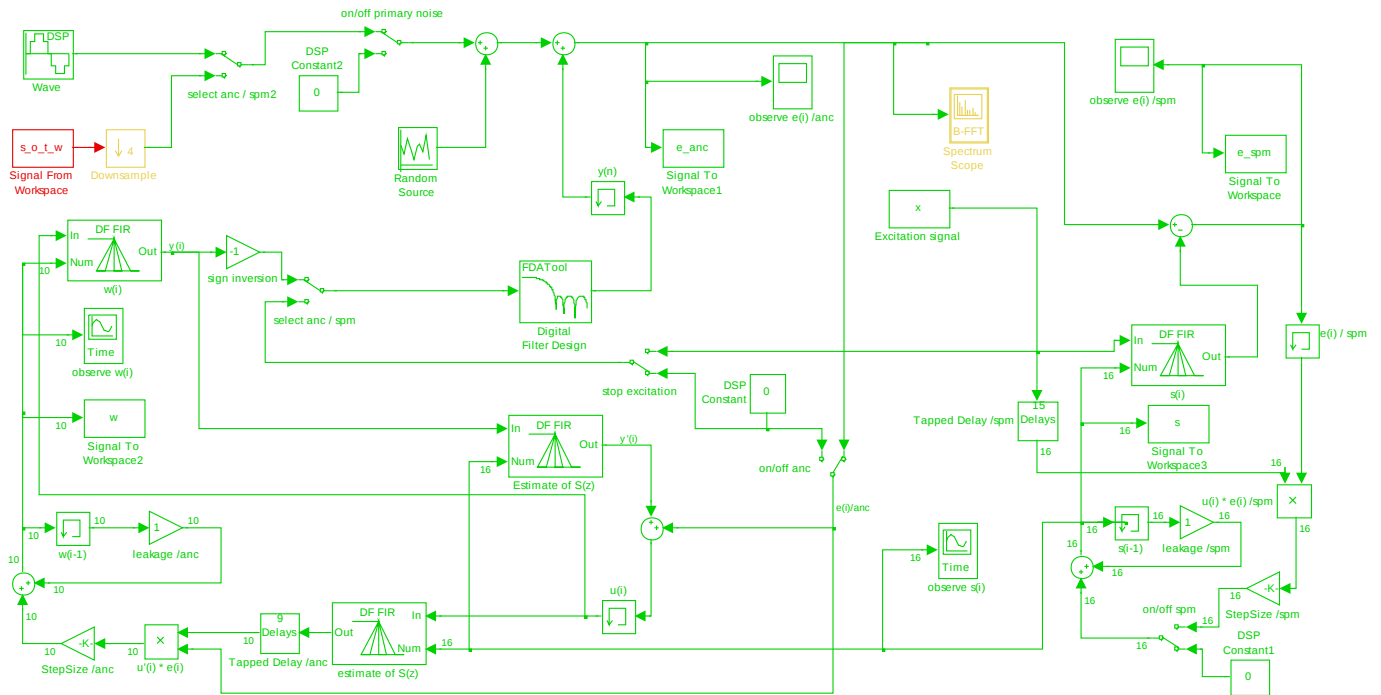
$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{u}'(i-1) e(i), \quad (2.90)$$

$$e(i) = d(i) - \mathbf{s}^T \mathbf{y}(i), \quad (2.91)$$

$$\mathbf{u}'(i) = \hat{s}^T \mathbf{u}(i), \quad (2.92)$$

όπου $\hat{s}$ είναι μία προσέγγιση για το secondary path s .

Δημιουργούμε τώρα ένα περιβάλλον προσομοίωσης για τον παραπάνω αλγόριθμο. Το αντίστοιχο μοντέλο Simulink είναι το εξής: [Simulink Models\FULMS Feedback ANC Off Line Sec Path Iden.mdl](#). Το μπλοκ διάγραμμά του είναι



Εικόνα 2.58 ο FULMS feedback ελεγκτής ενεργού θορύβου με off-line αναγνώριση του secondary

Ως κύριους θορύβους χρησιμοποιούμε το ημίτονο 250Hz και το σήμα “s.o.t.w.”.

Η διασπορά του προσθετικού λευκού θορύβου είναι 0.000001.

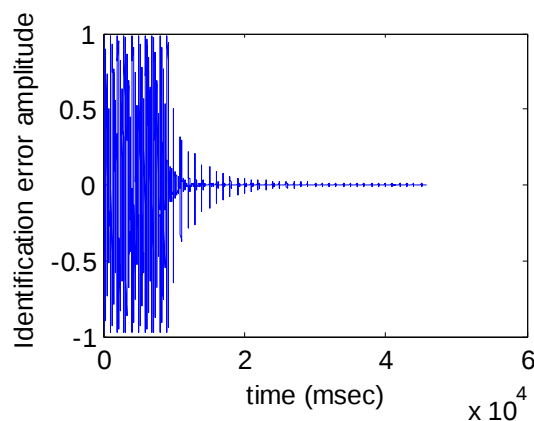
Το μέγεθος του προσαρμοζόμενου φίλτρου, το οποίο εκτελεί την εξασθένιση του θορύβου, είναι 10 συντελεστές και το μήκος βήματός του είναι 0.01.

Ως secondary path θεωρούμε το φίλτρο των εικόνων 2.27 και 2.28.

Το μέγεθος του προσαρμοζόμενου φίλτρου, το οποίο εκτελεί την αναγνώριση του secondary path, είναι 16 συντελεστές και το μήκος βήματός του είναι 0.006.

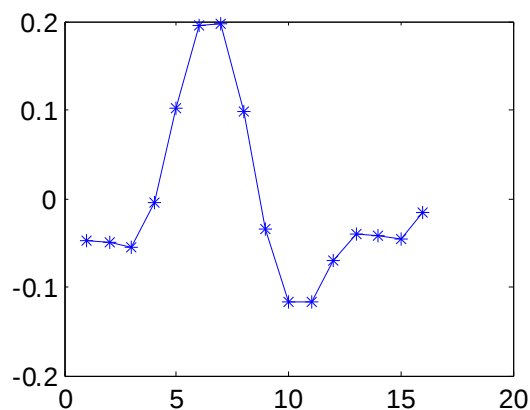
Το σήμα διέγερσης του secondary path είναι αυτό της εικόνας 2.57.

Αρχικά, ο ελεγκτής «εκπαιδεύεται» με την αναγνώριση του secondary path. Το σφάλμα αναγνώρισης είναι



Εικόνα 2.59 σφάλμα αναγνώρισης του secondary path

Το προσαρμοζόμενο φίλτρο, το οποίο εκτελεί την αναγνώριση είναι αρχικά ανενεργό για 10 περιόδους choir. Στη συνέχεια, γίνεται ενεργό και το σφάλμα αναγνώρισης μειώνεται και γρήγορα φθάνει στην μόνιμη κατάστασή του. Αυτό συμβαίνει επειδή το μήκος βήματος είναι μεγάλο (0.006). Παρόλα αυτά, το μεγάλο μήκος βήματος δεν εμποδίζει τους συντελεστές του αναγνωριστή από το να εντοπιστούν στις ακόλουθες θέσεις

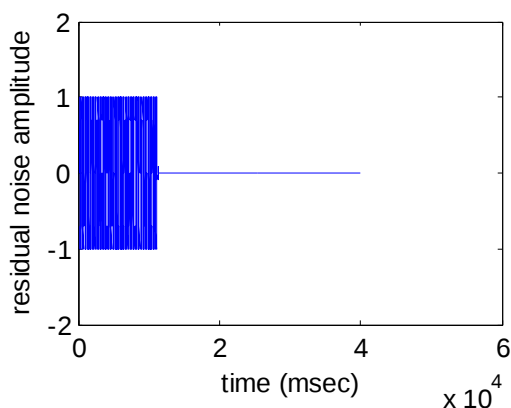


Εικόνα 2.60 $\hat{s}$

Στην πραγματικότητα, οι συντελεστές δεν είναι ποτέ σταθεροί, όσο τουλάχιστον ο παράγοντας οδήγησης της διαδικασίας προσαρμογής δεν είναι μηδέν. Αυτό βεβαίως δεν τους εμποδίζει από το να είναι σε μεγάλο βαθμό εντοπισμένοι. Η διαδικασία αναγνώρισης διακόπτεται, όταν το σφάλμα αναγνώρισης φθάσει στην μόνιμη κατάστασή του.

Έπειτα από το τέλος της φάσης εκπαίδευσης, το προσαρμοζόμενο φίλτρο, το οποίο εκτελεί την εξασθένιση θορύβου, γίνεται ενεργό. Στη συνέχεια εφαρμόζουμε τον κύριο θόρυβο (εάν το φίλτρο γίνει ενεργό αφότου έχει εμφανιστεί ο κύριος θόρυβος, περιπλοκές δημιουργούνται και τα βάρη του υπερχειλίζουν) .

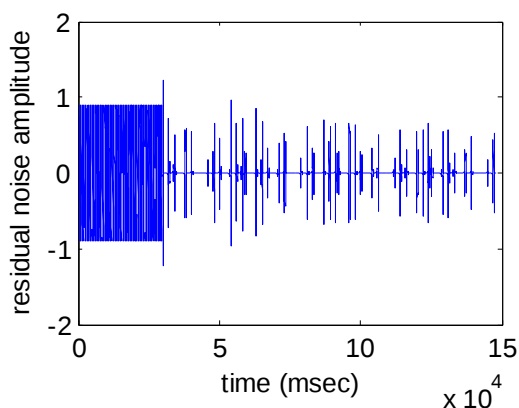
Η απόκριση του συστήματος στον κύριο θόρυβο των 250Hz είναι



Εικόνα 2.61 απόκριση του FULMS feedback ANC ελεγκτή με off-line secondary path αναγνώριση σε κύριο θόρυβο 250Hz

Η εξασθένιση θορύβου είναι 55.9dB. Συγκρίνοντας τις εικόνες 2.61 και 2.12 βλέπουμε ότι οι αποκρίσεις είναι ίδιες.

Η απόκριση του συστήματος στον “s.o.t.w.” θόρυβο είναι



Εικόνα 2.62 απόκριση του FULMS feedback ANC ελεγκτή με off-line secondary path αναγνώριση σε κύριο θόρυβο “s.o.t.w.”

Η εξασθένιση θορύβου είναι 19.8dB. Στην ιδανική περίπτωση της εικόνας 2.14 η εξασθένιση θορύβου ήταν 28.9dB. Αυτή η διαφορά των 28.9dB-19.8dB=9.1dB οφείλεται σε σφάλματα αναγνώρισης του secondary path.

Χρωστάμε τώρα μία καλή εξήγηση στον αναγνώστη, γιατί το σύστημα της εικόνας 2.58 δουλεύει με off-line αναγνώριση και όχι με on-line, όπως θα ήταν αναμενόμενο. Αυστηρά μιλώντας, κατά τη διάρκεια της αρχικής εκπαίδευσης του συστήματος, η διαδικασία αναγνώρισης ήταν πραγματικά on-line, καθώς οι αναβαθμίσεις του μοντέλου συμβαίνουν ταυτόχρονα με τις αποκρίσεις του secondary path. Παρ' όλα αυτά, έπειτα από τη φάση εκπαίδευσης και κατά την έναρξη της εξασθένισης του θορύβου, ο ελεγκτής χρησιμοποιεί την προσέγγιση του secondary path $\hat{s}$, η οποία είναι τώρα συγκεκριμένη και σταθερή. Εάν όμως το s μεταβληθεί, ως πούμε εξ' αιτίας μιας διαταραχής, ο ελεγκτής δε θα μπορεί να αντισταθμίσει αυτήν την αλλαγή, αφού το $\hat{s}$ θα παραμένει σταθερό, με αποτέλεσμα η απόδοσή του να υποβαθμίζεται. Μία πραγματική on-line αναγνώριση θα έπρεπε να είναι σε θέση να ανιχνεύσει αυτήν την αλλαγή στο s και να προσαρμόσει κατάλληλα το $\hat{s}$, χωρίς όμως να διακοπεί η εξασθένιση του θορύβου. Από αυτήν τη σκοπιά, η αναγνώριση που εκτελείται στο διάγραμμα 2.58 είναι πράγματι off-line. Είναι πολύ σημαντικό να σημειώσουμε, ότι σε πραγματικές εφαρμογές, το secondary path είναι σχεδόν πάντοτε χρονομεταβλητό και συνεπώς οι σθεναροί και υψηλής απόδοσης ελεγκτές το αναγνωρίζουν on-line. Αυτό είναι απολύτως απαραίτητο, καθώς σε μία ANC ηλεκτροακουστική διάταξη κάθε μεταφορική ή περιστροφική μετατόπιση ενός μικροφώνου ή ενός ηχείου επηρεάζει τη συνάρτηση μεταφοράς $S(z)$. Επιπλέον, η απόκριση του ενισχυτή ισχύος μεταβάλλεται με τη θερμοκρασία αλλά και με την αύξηση των θερμικών του κύκλων. Επίσης, το secondary path είναι πιθανό να εξαρτάται από χαρακτηριστικά του αέρα, όπως για παράδειγμα την ταχύτητά του ή την πυκνότητά του. Τέλος, ενδέχεται να υπάρχουν χρονομεταβλητές παρεμβολές, για παράδειγμα κινούμενα αντικείμενα όπως οχήματα, ή και άνθρωποι, μεταξύ του ηχείου εξασθένισης και του μικροφώνου σφάλματος, καθιστώντας έτσι το secondary path εντελώς μη στάσιμο.

Στην παρούσα εργασία, το secondary path θεωρείται στάσιμο. Φυσικά, στις περισσότερες πρακτικές εφαρμογές αυτή η υπόθεση δεν είναι ρεαλιστική για λόγους που ήδη αναφέρθηκαν. Παρ' όλα αυτά, όταν το περιβάλλον λειτουργίας του ANC ελεγκτή είναι σταθερό, ή σχεδόν σταθερό, αυτή η υπόθεση δεν εισαγάγει μεγάλο σφάλμα.

2.10 Ο Leaky LMS αλγόριθμος

Η ψηφιακή υλοποίηση του LMS αλγορίθμου εισαγάγει περιορισμένη αριθμητική ακρίβεια τόσο στην είσοδο του προσαρμοζόμενου φίλτρου όσο και σε όλες τις αλγοριθμικές ποσότητες. Αυτός ο περιορισμός οφείλεται σε σφάλματα κβαντοποίησης στον A/D μετατροπέα αλλά και σε σφάλματα στρογγυλοποίησης και λειτουργίες αποκοπής, κυρίως στους πολλαπλασιασμούς. Συνεπώς η αναβάθμιση βαρών του LMS δεν είναι ακριβώς αυτή της εξίσωσης (4.2.106), δηλαδή η

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{u}(i)e(i)$$

αλλά η

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{u}(i)e(i) + \mathbf{b}(i), \quad (2.93)$$

όπου ο όρος $\mathbf{b}(i)$ είναι ένα διάνυσμα σφάλματος, το οποίο αντιπροσωπεύει ανακρίβειες λόγω περιορισμένης αριθμητικής ακρίβειας. Θεωρώντας τις αναμενόμενες τιμές των δύο πλευρών της εξίσωσης (2.93) και επιλύοντας ως προς το $\mathbf{w}_{steady-state}$, έχουμε

$$\mathbf{w}_{steady-state} = R_u^{-1} \mathbf{r}_{ud} + \frac{1}{\mu} R_u^{-1} \mathbf{b}, \quad (2.94)$$

Βλέπουμε ότι το $\mathbf{b}$ διαταράσσει τη βέλτιστη λύση των κανονικών εξισώσεων και μάλιστα όσο πιο μικρό είναι το μήκος βήματος τόσο πιο μεγάλη θα είναι αυτή η διαταραχή. Αυτό είναι ένα σημαντικό πρόβλημα καθώς μία αύξηση στην αλγοριθμική ευστάθεια, μέσω μείωσης του μήκους βήματος, οδηγεί σε αύξηση της αριθμητικής αστάθειας!

Επιπρόσθετα στο πρόβλημα της αριθμητικής ακρίβειας, ανεπαρκής φασματική διέγερση του LMS αλγορίθμου ενδέχεται να οδηγήσει σε απόκλιση των βαρών. Για παράδειγμα ένα αθόρυβο ημίτονο θα επιδείξει μία τέτοια συμπεριφορά. Σε αυτήν την περίπτωση ο πίνακας αυτοσυσχέτισης είναι ill-conditioned και η λύση στην ελαχιστοποίηση της συνάρτησης κόστους δεν είναι μοναδική. Ως συνέπεια, είναι πιθανό, μαζί με την επίδραση των αριθμητικών σφαλμάτων, τα βάρη να αρχίσουν να αυξάνονται χωρίς περιορισμό και τελικά να υπερχειλίσουν κατά τη διαδικασία αναβάθμισής τους.

Αυτό μπορούμε να το δούμε ακόμα καλύτερα στο χώρο των συχνοτήτων. Ας υποθέσουμε ότι το σήμα αναφοράς του προσαρμοζόμενου φίλτρου είναι στενής ζώνης, δηλαδή

$$U(\omega) = \begin{cases} \text{όχι μηδέν,} & |\omega - \omega_0| < \omega_1 \\ 0, & |\omega - \omega_0| > \omega_1 \end{cases}, \quad (2.95)$$

όπου $U(\omega)$ είναι ο DFT του σήματος u . Τώρα, η έξοδος του φίλτρου θα είναι επίσης στενής ζώνης, καθώς

$$Y(\omega) = W(\omega)U(\omega) = \begin{cases} \text{όχι μηδέν,} & |\omega - \omega_0| < \omega_1 \\ 0, & |\omega - \omega_0| > \omega_1 \end{cases}, \quad (2.96)$$

Από την (2.96) βλέπουμε ότι το $W(\omega)$ θα μπορούσε να αυξάνεται χωρίς όριο έξω από το διάστημα $|\omega - \omega_0| < \omega_1$, χωρίς να επηρεάζει το σφάλμα προσαρμογής, με αποτέλεσμα ο προσαρμοστικός μηχανισμός να μην αντιλαμβάνονταν ποτέ αυτήν την αστάθεια!

Τέλος, στα πλαίσια του ενεργού ελέγχου θορύβου, ένα άλλο πρόβλημα ανακύπτει: υψηλά επίπεδα θορύβου συσχετισμένα με χαμηλόσυχνους συντονισμούς, μπορεί να οδηγήσουν σε μη γραμμική παραμόρφωση και υπερφόρτωση του ηχείου.

Από τα παραπάνω επιχειρήματα, είναι περισσότερο από προφανές ότι κάποιος περιορισμός ισχύος πρέπει να εισαχθεί στην έξοδο του προσαρμοζόμενου φίλτρου. Αυτός ο περιορισμός μπορεί να υλοποιηθεί τροποποιώντας τη συνάρτηση κόστους ως εξής:

$$J(i) = e^2(i) + \alpha \mathbf{w}_i^T \mathbf{w}_i, \quad (2.97)$$

όπου α είναι ένας θετικός παράγοντας βάρους.

Και πάλι θέλουμε να βρούμε μία αναβάθμιση στοχαστικής κλίσης της μορφής

$$\mathbf{w}_i = \mathbf{w}_{i-1} - \frac{\mu}{2} \nabla_{\mathbf{w}} \{J(i)\}, \quad (2.98)$$

Το πρόβλημα ανάγεται στον προσδιορισμό της κλίσης $\nabla_{\mathbf{w}} \{J(i)\}$.

Αντικαθιστώντας την (2.97) στο όρισμα της κλίσης έχουμε

$$\begin{aligned} \nabla_{\mathbf{w}} \{J(i)\} &= \nabla_{\mathbf{w}} \{e^2(i) + \alpha \mathbf{w}_{i-1}^T \mathbf{w}_{i-1}\} = \\ &= \nabla_{\mathbf{w}} \{e^2(i)\} + \nabla_{\mathbf{w}} \{\alpha \mathbf{w}_{i-1}^T \mathbf{w}_{i-1}\} = \\ &= -2\mathbf{u}(i)d(i) + 2\mathbf{u}(i)\mathbf{u}(i)^T \mathbf{w}_{i-1} + 2\alpha \mathbf{w}_{i-1} \Rightarrow \\ \nabla_{\mathbf{w}} \{J(i)\} &= -2\mathbf{u}(i)d(i) + 2\mathbf{u}(i)\mathbf{u}(i)^T \mathbf{w}_{i-1} + 2\alpha \mathbf{w}_{i-1}, \quad (2.99) \end{aligned}$$

από τις (2.98) και (2.99) η αναβάθμιση των βαρών εκφράζεται ως

$$\begin{aligned} \mathbf{w}_i &= \mathbf{w}_{i-1} + \mu \mathbf{u}(i)d(i) - \mu \mathbf{u}(i)\mathbf{u}(i)^T \mathbf{w}_{i-1} - \mu \alpha \mathbf{w}_{i-1} \Rightarrow \\ \mathbf{w}_i &= (1 - \mu \alpha) \mathbf{w}_{i-1} + \mu \mathbf{u}(i)[d(i) - \mathbf{u}(i)^T \mathbf{w}_{i-1}], \quad (2.100) \end{aligned}$$

Η εξίσωση (2.100) ονομάζεται *Leaky LMS αλγόριθμος*. Η ποσότητα

$$v = (1 - \mu \alpha), \quad (2.101)$$

ονομάζεται *leakage factor*. Στην περίπτωση ύπαρξης ενός secondary path, ο αντίστοιχος αλγόριθμος ονομάζεται *Leaky FULMS αλγόριθμος*, με αναβάθμιση βαρών

$$\mathbf{w}_i = (1 - \mu \alpha) \mathbf{w}_{i-1} + \mu \mathbf{u}'(i)[d(i) - \mathbf{u}'(i)^T \mathbf{w}_{i-1}], \quad (2.102)$$

Είναι πολύ ενδιαφέρον να σημειώσουμε ότι, όταν ο παράγοντας οδήγησης του προσαρμοζόμενου φίλτρου

$$\text{παράγοντας οδήγησης} = \mathbf{u}(i)[d(i) - \mathbf{u}(i)^T \mathbf{w}_{i-1}] = \mathbf{u}(i)e(i), \quad (2.103)$$

είναι μηδέν για $i \geq N$, τότε

$$\mathbf{w}_{N+k} = v^k \mathbf{w}_N, \quad (2.104)$$

Συνεπώς

$$\lim_{k \rightarrow \infty} \mathbf{w}_{N+k} = 0, \quad (2.105)$$

υποθέτοντας ότι

$$v < 1, \quad (2.106)$$

Έτσι, όταν δεν υπάρχει παράγοντας οδήγησης, τα βάρη του προσαρμοζόμενου φίλτρου *διαρρέουν* προς το μηδέν.

Σε πρακτικές εφαρμογές ο LMS αλγόριθμος, όπως και οι διάφορες τροποποιημένες εκδοχές του, πάντοτε χρησιμοποιούν παράγοντα διαρροής. Αυτό συμβαίνει, επειδή η παρουσία του παράγοντα διαρροής όχι μόνο μειώνει τα αριθμητικά λάθη σε πεπερασμένης ακρίβειας υλοποιήσεις, αλλά περιορίζει και την έξοδο του προσαρμοζόμενου φίλτρου και γενικά έχει μία ευεργετική, σταθεροποιητική επίδραση σε ολόκληρη τη διαδικασία προσαρμογής.

Για να δούμε γιατί ο παράγοντας διαρροής συνεισφέρει θετικά στην ολική σθεναρότητα του προσαρμοστικού αλγορίθμου, θεωρούμε την αναμενόμενη τιμή των δύο μερών της (2.100):

$$\begin{aligned} \bar{\mathbf{w}}_i &= (1 - \mu\alpha)\bar{\mathbf{w}}_{i-1} - \mu R_u \bar{\mathbf{w}}_{i-1} + \mu \mathbf{r}_{ud} \Rightarrow \\ \bar{\mathbf{w}}_i &= [I - \mu(\alpha I + R_u)]\bar{\mathbf{w}}_{i-1} + \mu \mathbf{r}_{ud}, \quad (2.107) \end{aligned}$$

Η εξίσωση (4.2.51) είναι η

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu(\mathbf{r}_{ud} - R_u \mathbf{w}_{i-1})$$

και περιγράφει την εξίσωση αναβάθμισης της αναμενόμενης τιμής του κλασικού LMS αλγορίθμου. Αυτή η εξίσωση μπορεί να αναδιαταχθεί ως εξής:

$$\mathbf{w}_i = [I - \mu R_u]\mathbf{w}_{i-1} + \mu \mathbf{r}_{ud}, \quad (2.108)$$

Συγκρίνοντας τις (2.107) και (2.108), βλέπουμε ότι το αποτέλεσμα του παράγοντα διαρροής είναι η τροποποίηση του πίνακα αυτοσυσχέτισης του σήματος αναφοράς. Συνεπώς εφαρμόζοντας τον Leaky LMS με σήμα αναφοράς, το οποίο έχει πίνακα αυτοσυσχέτισης R_u , είναι ισοδύναμο με την εφαρμογή του κλασικού LMS σε σήμα αναφοράς με πίνακα αυτοσυσχέτισης

$$R'_u = \alpha I + R_u, \quad (2.109)$$

Τώρα, εφαρμόζοντας φασματική αποσύνθεση στον R_u (εξίσωση (4.2.57)), η (2.109) γίνεται

$$\begin{aligned} R'_u &= \alpha U U^T + U \Lambda U^T \Rightarrow \\ R'_u &= U[\alpha I + \Lambda]U^T, \quad (2.110) \end{aligned}$$

Η εξίσωση (2.110) αποκαλύπτει ότι, αυτό το οποίο ο παράγοντας διαρροής κάνει, είναι να τροποποιεί τις ιδιοτιμές του πίνακα αυτοσυσχέτισης. Εφόσον ο α είναι πάντοτε θετικός, οι νέες ιδιοτιμές είναι πάντα θετικές, ακόμα και εάν κάποιες από τις προηγούμενες ιδιοτιμές ήταν μηδέν (υπενθυμίζουμε ότι R_u στη χειρότερη περίπτωση είναι θετικά ημιορισμένος) και συνεπώς οι ill-conditioned πίνακες τείνουν να γίνουν well-conditioned!

Επίσης, αφού οι ιδιοτιμές του πίνακα αυτοσυσχέτισης είναι οι συντελεστές του φάσματος ισχύος του αντίστοιχου σήματος, η εισαγωγή διαρροής είναι ισοδύναμη με την προσθήκη λευκού θορύβου.

Παρ'όλα αυτά, το τίμημα της διαρροής είναι η εισαγωγή πόλωσης στη μόνιμη κατάσταση, όπως φαίνεται από την (2.107) και σε αντιστοιχία με τις κανονικές εξισώσεις:

$$\mathbf{w}_{\infty, leaky} = [\alpha I + R_u]^{-1} \mathbf{r}_{ud}, \quad (2.111)$$

οι οποίες είναι φυσικά διαφορετικές από τις κανονικές εξισώσεις όταν η διαρροή είναι μηδέν

$$\mathbf{w}_{\infty, optimal} = \mathbf{R}_u^{-1} \mathbf{r}_{ud}$$

Η επιλογή του α συνεπώς αντιπροσωπεύει ένα συμβιβασμό μεταξύ σθεναρότητας και απόδοσης. Η τιμή του α γενικώς καθορίζεται σε μία πειραματική βάση. Επίσης, μπορεί να αποδειχθεί ότι ισχύει η παρακάτω σχέση

$$J_{excess-leakage} \sim \alpha^2, \quad (2.112)$$

και συνεπώς το α θα πρέπει να κρατηθεί αυστηρώς μικρότερο από τη μονάδα

$$\alpha < 1, \quad (2.113)$$

2.11 O Leaky FULMS Feedback Ελεγκτής Ενεργού Θορύβου με Off-Line Αναγνώριση του Secondary Path

Η ολοκληρωμένη και τελική μορφή του ελεγκτή ενεργού θορύβου, τον οποίο μελετούμε, είναι ο *Leaky FULMS Feedback Ελεγκτής Ενεργού Θορύβου με Off-Line Αναγνώριση του Secondary Path*, του οποίου η λειτουργία περιγράφεται πλήρως από τις παρακάτω εξισώσεις

$$\mathbf{y}(i) = \mathbf{w}_{i-1}^T \mathbf{u}(i-1), \quad (2.114)$$

$$u(i) = d(i) + \hat{\mathbf{s}}^T \mathbf{y}(i) - \mathbf{s}^T \mathbf{y}(i), \quad (2.115)$$

$$\mathbf{w}_i = (1 - \mu\alpha)\mathbf{w}_{i-1} + \mu\mathbf{u}'(i-1)\mathbf{e}(i), \quad (2.116)$$

$$\mathbf{e}(i) = d(i) - \mathbf{s}^T \mathbf{y}(i), \quad (2.117)$$

$$\mathbf{u}'(i) = \hat{\mathbf{s}}^T \mathbf{u}(i), \quad (2.118)$$

Δημιουργούμε τώρα ένα περιβάλλον προσομοίωσης των εξισώσεων (2.114)-(2.118) ως εξής:

Ως κύριους θορύβους χρησιμοποιούμε ένα ημίτονο 250Hz και το σήμα “s.o.t.w.”.

Το μέγεθος του προσαρμοζόμενου φίλτρου, το οποίο εκτελεί την εξασθένιση θορύβου είναι 10 συντελεστές, το μήκος βήματός του είναι 0.01 και η παράμετρος διαρροής α είναι 0.1.

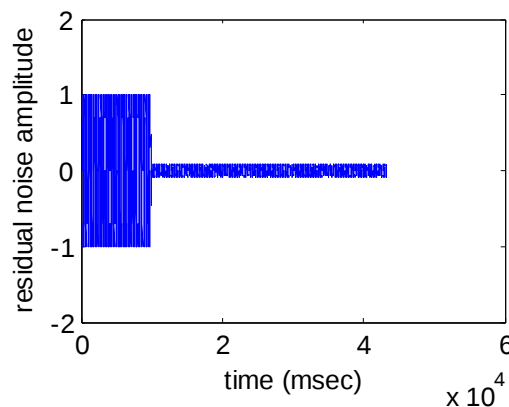
Το secondary path είναι το φίλτρο των εικόνων 2.27 και 2.28.

Το μέγεθος του προσαρμοζόμενου φίλτρου, το οποίο εκτελεί την αναγνώριση του secondary path, είναι 16 συντελεστές, το μήκος βήματός του είναι 0.006 και η παράμετρος διαρροής του α είναι 0.

Διασπορά του προσθετικού λευκού θορύβου είναι 0.000001.

Το αντίστοιχο μοντέλο Simulink είναι: [Simulink Models\FULMS Feedback ANC Off Line Sec Path Idem.mdl](#).

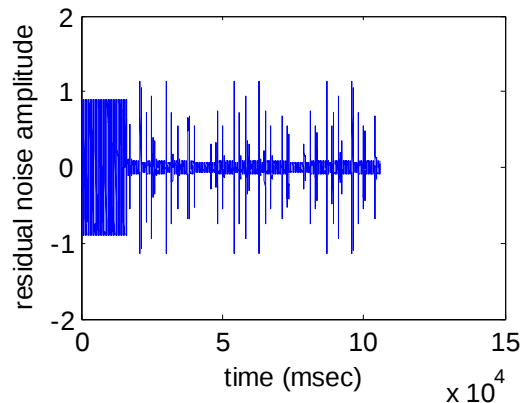
Η απόκριση του συστήματος στο ημίτονο 250Hz είναι



Εικόνα 2.63 απόκριση του leaky FULMS feedback ANC ελεγκτή σε κύριο θόρυβο 250Hz ($\mu=0.01$, $\alpha=0.1$)

Η εξασθένιση θορύβου μειώνεται από 55.9dB ($\alpha = 0$) σε 22.3dB ($\alpha = 0.1$).

Η απόκριση του συστήματος στον κύριο θόρυβο “s.o.t.w.” είναι



Εικόνα 2.64 απόκριση του leaky FULMS feedback ANC ελεγκτή σε κύριο θόρυβο “s.o.t.w.” ($\mu=0.01$, $\alpha=0.1$)

Η εξασθένιση θορύβου μειώνεται από 19.8dB ($\alpha = 0$) σε 15.3dB ($\alpha = 1$).

Από τις παραπάνω προσομοιώσεις επιβεβαιώνουμε ότι η ύπαρξη διαρροής οδηγεί σε υποβάθμιση της απόδοσης. Όσο μεγαλύτερη είναι η τιμή του α τόσο πιο φτωχή θα είναι η απόδοση.

3 Υλοποίηση του Ελεγκτή Ενεργού Θορύβου με τον Ψηφιακό Επεξεργαστή Σήματος 21065L της Analog Devices

3.1 Εισαγωγή

Σκοπός μας σε αυτήν την ενότητα είναι η υλοποίηση του αλγορίθμου, ο οποίος περιγράφεται από τις εξισώσεις (2.114)-(2.118). Η υλοποίηση θα γίνει με χρήση του διαδεδομένου επεξεργαστή ψηφιακού σήματος 21065L της εταιρείας Analog Devices και συγκεκριμένα με το EZ-LITE 21065L, το οποίο είναι μία κάρτα με πυρήνα τον επεξεργαστή 21065L, εξωτερική μνήμη, και έτοιμο αναλογικό interface. Στο παρόν κείμενο δε θα αναφερθούμε στον τρόπο λειτουργίας του επεξεργαστή, ούτε στον τρόπο επικοινωνίας του με το αναλογικό interface. Οι πληροφορίες αυτές βρίσκονται στα αντίστοιχα user manuals (βλέπε βιβλιογραφία). Συνοπτικά αναφέρουμε τα βασικά χαρακτηριστικά του επεξεργαστή 21065L:

- 32 bits μήκος λέξης
- Fixed και Floating Point αριθμητική on chip
- 66 MIPS/198 MFLOPS
- 18K × 32bit εσωτερική μνήμη ρυθμιζόμενη από το χρήστη
- 240 Mbits/sec I/O
- 10 DMA κανάλια
- 2 TDM σειριακές θύρες

Στο επόμενο εδάφιο περνάμε κατευθείαν στα αποτελέσματα των πειραμάτων πραγματικού χρόνου. Ο πλήρης κώδικας με σχόλια του επεξεργαστή μπορεί να ευρεθεί στο κεφάλαιο 4.3 του παραρτήματος.

3.2 Ο Ελεγκτής Ενεργού Θορύβου Πραγματικού Χρόνου: Πειράματα και Αποτελέσματα

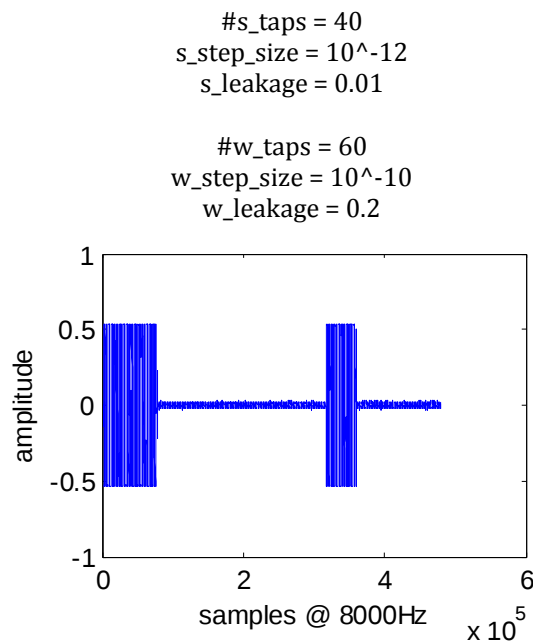
3.2.1 Η Ηλεκτροακουστική Πειραματική Διάταξη

Τα εξαρτήματα, τα οποία χρησιμοποιούνται στην πειραματική διάταξη, είναι τα εξής: ένα συμβατικό Sony cd-player (CMT-NEZ50) με δύο ηχεία (SS-CNEZ50), ένας NAD ενισχυτής ισχύος με προενισχυτή (AV-716), μία εξωτερική κάρτα ήχου ALESIS με

δυνατότητες προενίσχυσης σήματος μικροφώνου (io|2), ένα SHURE μικρόφωνο SM57 και τέλος η κάρτα EZLAB-21065L της Analog Devices. Ο εξοπλισμός αυτός χρησιμοποιείται ως εξής: ως κύρια πηγή θορύβου χρησιμοποιούμε το Sony cd-player με το ένα του ηχείο. Ως μικρόφωνο σφάλματος χρησιμοποιούμε το μικρόφωνο SHURE. Το σήμα του μικροφώνου προενισχύεται και γίνεται line σήμα από την εξωτερική κάρτα ήχου Alesis. Η έξοδος της κάρτας ήχου, η οποία είναι το σήμα σφάλματος σε line μορφή, οδηγείται στην αναλογική είσοδο του EZLAB-21065L. Ο αντι-θόρυβος, ο οποίος παράγεται από τον επεξεργαστή 21065L, εξάγεται από την line-out θύρα του EZLAB-21065L. Στη συνέχεια το σήμα αυτό οδηγείται στη line-in είσοδο του NAD ενισχυτή. Η έξοδος του ενισχυτή οδηγεί το δεύτερο ηχείο, το οποίο είναι και το ηχείο εξασθένισης. Η γεωμετρία της διάταξης έχει ως εξής: τα δύο ηχεία βρίσκονται τοποθετημένα το ένα αντικρουστά στο άλλο, σε απόσταση 30 εκατοστών. Ακριβώς στη μέση της απόστασης βρίσκεται οριζόντια τοποθετημένο το μικρόφωνο, κάθετα προς την ευθεία που νοερά ενώνει τα δύο ηχεία. Σημαντική λεπτομέρεια: το μικρόφωνο είναι ανυψωμένο κατά τα 3/5 του ύψους των ηχείων. Όλες οι μετρήσεις έγιναν σε περιβάλλον χαμηλού θορύβου.

3.2.2 Κύριος θόρυβος 200Hz

Το secondary path διεγέρθηκε από ένα chirp σήμα 150Hz-500Hz, το οποίο πρώτα παραθυρώθηκε με ένα παράθυρο Tukey ($r=0.2$). Τα χαρακτηριστικά των δύο προσαρμοζόμενων φίλτρων είναι τα εξής:



Εικόνα 3.1 κυματομορφή απόκρισης του ελεγκτή ενεργού θορύβου πραγματικού χρόνου σε κύριο θόρυβο 200Hz.

Η εξασθένιση είναι 34.6 dB.

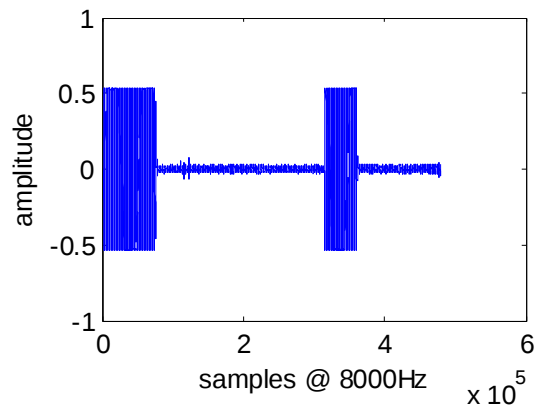
Ακούστε την απόκριση ακριβώς εδώ: Audio Files\real_time_response_200Hz.wav

3.2.3 Κύριος θόρυβος 250Hz

Το secondary path διεγέρθηκε από ένα chirp σήμα 150Hz-500Hz, το οποίο πρώτα παραθυρώθηκε με ένα παράθυρο Tukey ($r=0.2$). Τα χαρακτηριστικά των δύο προσαρμοζόμενων φίλτρων είναι τα εξής:

```
#s_taps = 40
s_step_size = 10^-12
s_leakage = 0.01

#w_taps = 60
w_step_size = 10^-10
w_leakage = 0.2
```



Εικόνα 3.2 κυματομορφή απόκρισης του ελεγκτή ενεργού θορύβου πραγματικού χρόνου σε κύριο θόρυβο 250Hz.

Η εξασθένιση είναι 34.15 dB.

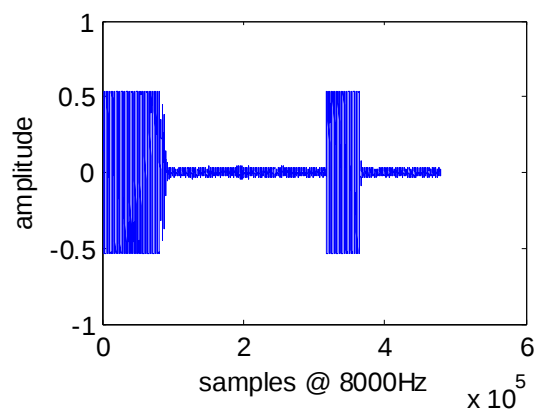
Ακούστε την απόκριση ακριβώς εδώ: [Audio Files\real time response 250Hz.wav](#)

3.2.4 Κύριος θόρυβος 300Hz

Το secondary path διεγέρθηκε από ένα chirp σήμα 150Hz-500Hz, το οποίο πρώτα παραθυρώθηκε με ένα παράθυρο Tukey ($r=0.2$). Τα χαρακτηριστικά των δύο προσαρμοζόμενων φίλτρων είναι τα εξής:

```
#s_taps = 40
s_step_size = 10^-12
s_leakage = 0.01
```

```
#w_taps = 60
w_step_size = 10^-10
w_leakage = 0.2
```



Εικόνα 3.3 κυματομορφή απόκρισης του ελεγκτή ενεργού θορύβου πραγματικού χρόνου σε κύριο θόρυβο 300Hz.

Η εξασθένιση είναι 31.95 dB.

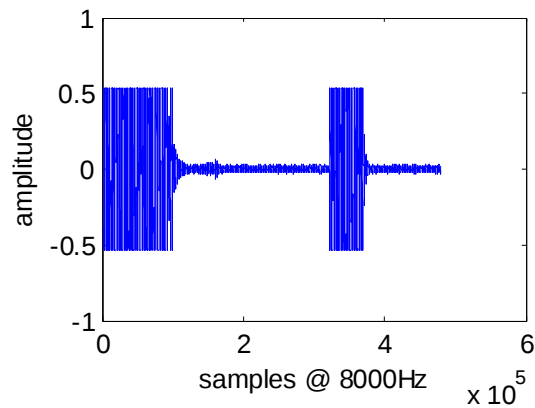
Ακούστε την απόκριση ακριβώς εδώ: [Audio Files\real time response 300Hz.wav](#)

3.2.5 Κύριος θόρυβος 350Hz

Το secondary path διεγέρθηκε από ένα chirp σήμα 150Hz-500Hz, το οποίο πρώτα παραθυρώθηκε με ένα παράθυρο Tukey ($r=0.2$). Τα χαρακτηριστικά των δύο προσαρμοζόμενων φίλτρων είναι τα εξής:

```
#s_taps = 40
s_step_size = 10^-12
s_leakage = 0.01
```

```
#w_taps = 60
w_step_size = 10^-10
w_leakage = 0.2
```



Εικόνα 3.4 κυματομορφή απόκρισης του ελεγκτή ενεργού θορύβου πραγματικού χρόνου σε κύριο θόρυβο 350Hz.

Η εξασθένιση είναι 33.74 dB.

Ακούστε την απόκριση ακριβώς εδώ: [Audio Files\real_time_response_350Hz.wav](#)

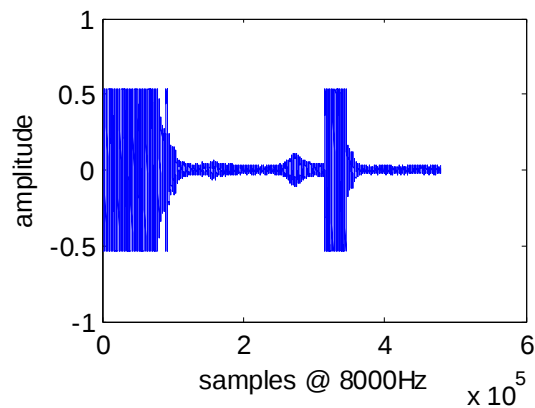
Έχει μεγάλο ενδιαφέρον να παρατηρήσουμε, αφότου ακούσουμε την απόκριση, ότι ο εναπομένων θόρυβος αρχίζει και ταλαντώνεται καθώς θέτουμε ενεργό το προσαρμοζόμενο φίλτρο, το οποίο εκτελεί την εξασθένιση.

3.2.6 Κύριος θόρυβος 400Hz

Το secondary path διεγέρθηκε από ένα chirp σήμα 150Hz-500Hz, το οποίο πρώτα παραθυρώθηκε με ένα παράθυρο Tukey ($r=0.2$). Τα χαρακτηριστικά των δύο προσαρμοζόμενων φίλτρων είναι τα εξής:

```
#s_taps = 40
s_step_size = 10^-12
s_leakage = 0.01
```

```
#w_taps = 60
w_step_size = 10^-10
w_leakage = 0.2
```



Εικόνα 3.5 κυματομορφή απόκρισης του ελεγκτή ενεργού θορύβου πραγματικού χρόνου σε κύριο θόρυβο 400Hz.

Η εξασθένιση είναι 24.1 dB.

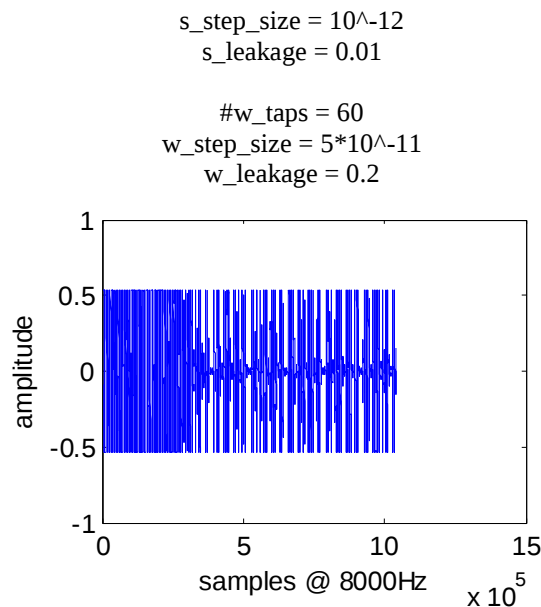
Ακούστε την απόκριση ακριβώς εδώ: [Audio Files\real_time_response_400Hz.wav](#)

Σε αυτήν τη συχνότητα ο εναπομένων θόρυβος αρχίζει και παρουσιάζει αστάθειες και συνεπώς το μέγεθος της εξασθένισης (24.1dB) δεν είναι ενδεικτικό, ούτε έχει σημασία.

3.2.7 Κύριος θόρυβος “s.o.t.w.”

Το secondary path διεγέρθηκε από ένα chirp σήμα 150Hz-350Hz παραθυρωμένο κατά Tukey με $r=0.1$. Τα χαρακτηριστικά των δύο προσαρμοζόμενων φίλτρων έχουν ως εξής:

```
#s_taps = 40
```



Εικόνα 3.6 κυματομορφή απόκρισης του ελεγκτή ενεργού θορύβου πραγματικού χρόνου σε κύριο θόρυβο “s.o.t.w.”.

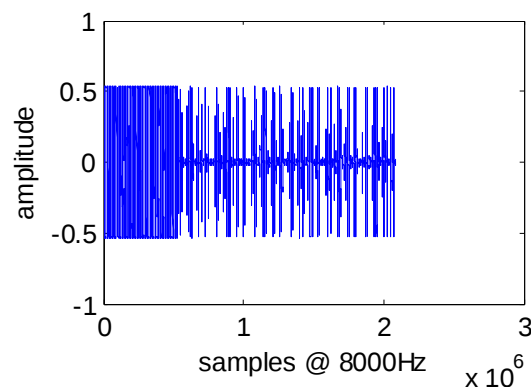
Μπορείτε να ακούσετε την απόκριση ακριβώς εδώ: [Audio Files\real_time_response_sotw.wav](#)

Παρατηρούμε, ακούγοντας την απόκριση, ότι πρώτον κάποιες συχνότητες οδηγούν τον εναπομένοντα θόρυβο σε μικρές ταλαντώσεις, δεύτερον ότι η ταχύτητα παρακολούθησης του προσαρμοζόμενου φίλτρου δεν είναι μεγάλη και τρίτον ότι η απόδοση του φίλτρου είναι διαφορετική για διαφορετικές συχνότητες.

Για να έχει νόημα η μέτρηση της εξασθένισης, δημιουργούμε ένα σήμα ακριβώς ίδιο με το “s.o.t.w.”, με τη διαφορά ότι κάθε αρμονική του διαρκεί διπλάσιο χρόνο. Αυτό το κάνουμε ώστε να προλαβαίνει το προσαρμοζόμενο φίλτρο να προσαρμοστεί όσο το δυνατόν καλύτερα μπορεί. Θα μπορούσαμε βέβαια μην αλλάξουμε τον κύριο θόρυβο αλλά να αυξήσουμε το μήκος βήματος του φίλτρου. Αυτό δοκιμάστηκε και κρίθηκε ως μη συνετή επιλογή, λόγω προβλημάτων αστάθειας.

Το νέο σήμα “slow s.o.t.w.” είναι το εξής: [Audio Files\smoke on the water 2sec fundamental duration.wav](#)

Η απόκριση του συστήματος στον “slow s.o.t.w.” έχει ως εξής:

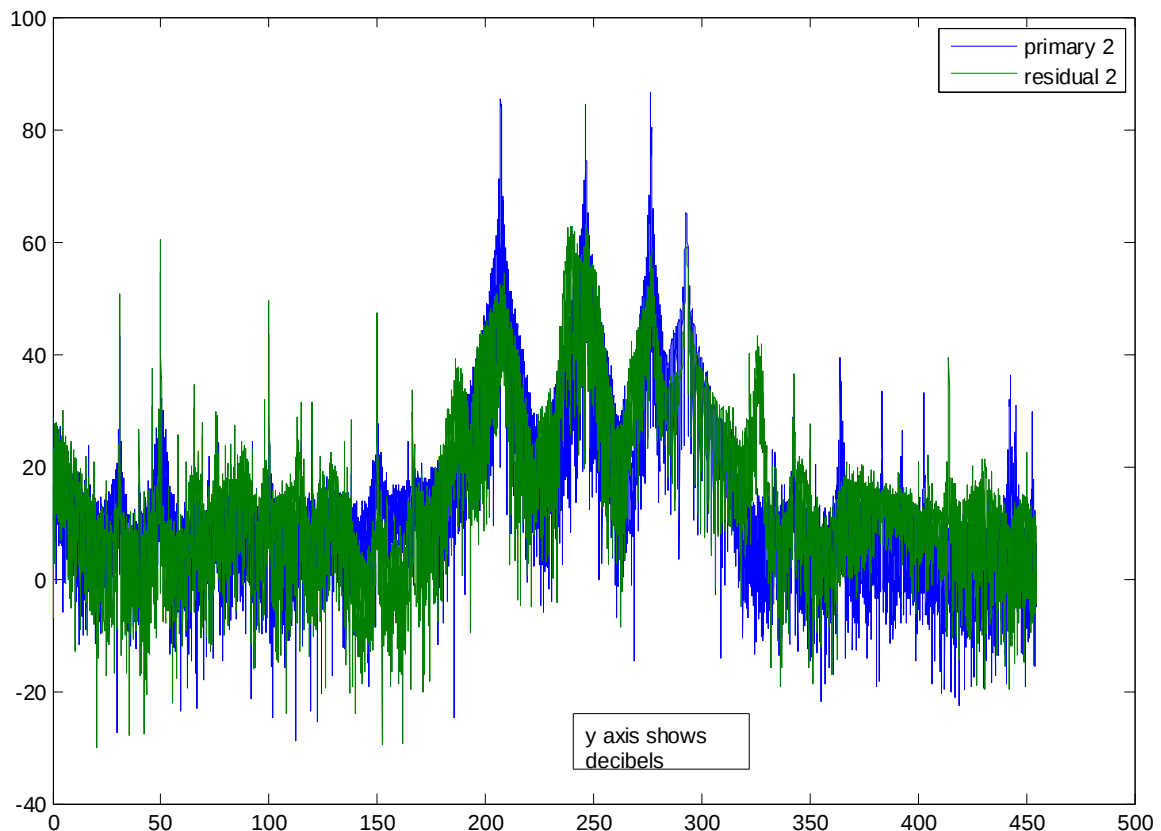


Εικόνα 3.7 κυματομορφή απόκρισης του ελεγκτή ενεργού θορύβου πραγματικού χρόνου σε κύριο θόρυβο “slow s.o.t.w.”.

Μπορείτε να ακούσετε την απόκριση ακριβώς εδώ: [Audio Files\real_time_response_slow_sotw.wav](#)

Η απόκριση είναι πολύ ικανοποιητική και εντυπωσιακή ταυτόχρονα.

Το φάσμα του εναπομένοντος θορύβου σε σχέση με το φάσμα του κύριου θορύβου έχει ως εξής:



Εικόνα 3.8 φάσμα του εναπομένοντος θορύβου (πράσινο) σε σύγκριση με το φάσμα του κύριου θορύβου (μπλε) “slow s.o.t.w.”

Η εξασθένιση σε ολόκληρο το χρονικό παράθυρο, στο οποίο ο ενεργός ελεγκτής θορύβου πραγματικού χρόνου είναι ON, είναι 20 dB περίπου σε κάθε συχνότητα.

4 Παράρτημα

4.1 Αβεβαιότητα ως προς τη συχνότητα κατά την μετατροπή ηλεκτρονικών σημάτων σε ακουστικά (πείραμα πραγματικού χρόνου)

Το πείραμα που ακολουθεί έχει ως σκοπό, να καταδείξει την ελαφριά παραμόρφωση στη συχνότητα κατά τη μετατροπή ηλεκτρονικών σημάτων σε ακουστικά σήματα. Η παραμόρφωση αυτή ενδέχεται να είναι πολύ μικρής τάξης, αλλά όντας τυχαία, έχει ως αποτέλεσμα την εισαγωγή μιας αβεβαιότητας ως προς τη συχνότητα, η οποία θέτει καθοριστικούς περιορισμούς κατά την ανάπτυξη ενός συστήματος ενεργού θορύβου.

Θεωρήστε ότι προσθέτουμε δύο ημίτονα ακριβώς της ίδιας συχνότητας ω_1 αλλά με διαφορετικά πλάτη και φάσεις. Χωρίς βλάβη της γενικότητας ένα από αυτά τα δύο ημίτονα μπορεί να παίζει το ρόλο του σήματος αναφοράς και συνεπώς να έχει μοναδιαίο πλάτος και μηδενική φάση. Το προκύπτον σήμα είναι

$$\begin{aligned}
 x &= x_1 + x_2 = \\
 &\sin(2\pi\omega_1 t) + k \sin(2\pi\omega_1 t + \varphi) = \\
 &\sin(2\pi\omega_1 t) + k \cos\varphi \sin(2\pi\omega_1 t) + k \sin\varphi \cos(2\pi\omega_1 t) = \\
 &(1 + k \cos\varphi) \sin(2\pi\omega_1 t) + k \sin\varphi \cos(2\pi\omega_1 t) = \\
 &\sqrt{(1 + k \cos\varphi)^2 + (k \sin\varphi)^2} * \sin\left(2\pi t \omega_1 + \arctan \frac{k \sin\varphi}{1 + k \cos\varphi}\right), \quad (4.1.1)
 \end{aligned}$$

Η τελευταία ισότητα προέκυψε απαιτώντας η ποσότητα

$$(1 + k \cos\varphi) \sin(2\pi\omega_1 t) + k \sin\varphi \cos(2\pi\omega_1 t)$$

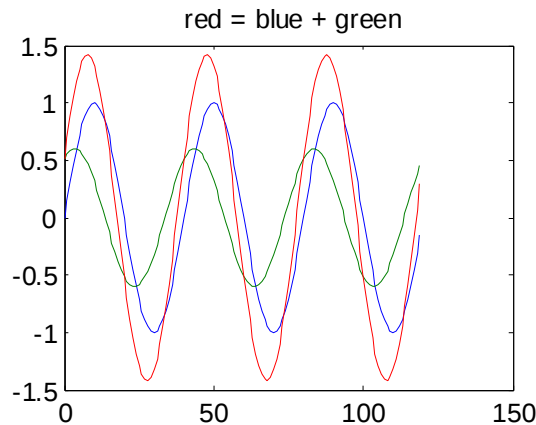
να ισούται με μία άλλη ποσότητα της μορφής

$$C * \sin(2\pi\omega_1 t + \theta)$$

Αυτό είναι ένα θεμελιώδες αποτέλεσμα της τριγωνομετρίας, του οποίου θα κάνουμε χρήση στην ακόλουθη ανάλυση, και έτσι το παραθέτουμε εδώ για ευκολία της αναφοράς:

$$A \sin(\omega t) + B \cos(\omega t) = \begin{cases} \sqrt{A^2 + B^2} \sin\left[\omega t + \arctan\left(\frac{B}{A}\right)\right], & (4.1.2) \\ \sqrt{A^2 + B^2} \cos\left[\omega t + \arctan\left(-\frac{A}{B}\right)\right], & (4.1.3) \end{cases}$$

Έτσι με βάση την (4.1.1), όταν δύο ημίτονα της ίδιας συχνότητας υπερτίθενται, ένα άλλο ημίτονο προκύπτει ακριβώς της ίδιας συχνότητας με τα αρχικά, ανεξάρτητα από τη σχέση πλάτους και φάσης των δύο αρχικών ημιτόνων. Αυτό μπορεί να οπτικοποιηθεί με τη βοήθεια της ακόλουθης εικονας:



Εικόνα 4.1.1 υπέρθεση δύο ημιτόνων ίδιας συχνότητας

Τώρα θεωρήστε ότι προσθέτουμε δύο ημίτονα με διαφορετικά πλάτη και διαφορετικές συχνότητες και φάσεις. Έστω οι δύο συχνότητες ω_1 και ω_2 . Πάλι ένα από τα ημίτονα παίζει το ρόλο του σήματος αναφοράς. Το προκύπτον σήμα είναι:

$$x = x_1 + x_2 = \sin(\omega_1 t) + K \sin(\omega_2 t + \varphi)$$

Τώρα δουλεύουμε λίγο πάνω στο x ώστε να το φέρουμε σε μία μορφή, η οποία θα μας βοηθήσει περαιτέρω στην κατανόηση:

$$\begin{aligned} x &= \sin(\omega_1 t) + K \cos \varphi \sin(\omega_2 t) + K \sin \varphi \cos(\omega_2 t) = \\ &= \sin(\omega_1 t) + A \sin(\omega_2 t) + B \cos(\omega_2 t) = (1) \\ &\text{όπου } A = K \cos \varphi \text{ και } B = K \sin \varphi, \quad (4.1.4) \end{aligned}$$

$$\text{έστω } \omega_a = \frac{1}{2} * (\omega_1 + \omega_2) \text{ και } \omega_b = \frac{1}{2} * (\omega_1 - \omega_2), \quad (4.1.5)$$

όπου υποθέτουμε ότι $\omega_1 > \omega_2$ χωρίς βλάβη της γενικότητας, και έτσι προκύπτει $\omega_1 = \omega_a + \omega_b$ και $\omega_2 = \omega_a - \omega_b$, (4.1.6)

$$\begin{aligned} (1) &= \sin(\omega_a t + \omega_b t) + A \sin(\omega_a t - \omega_b t) + B \cos(\omega_a t - \omega_b t) = \\ &= \sin(\omega_a t) \cos(\omega_b t) + \cos(\omega_a t) \sin(\omega_b t) + A \sin(\omega_a t) \cos(\omega_b t) - \\ &= A \cos(\omega_a t) \sin(\omega_b t) + B \cos(\omega_a t) \cos(\omega_b t) + B \sin(\omega_a t) \sin(\omega_b t) = \\ &= [\sin(\omega_a t) \cos(\omega_b t) + A \sin(\omega_a t) \cos(\omega_b t) + B \sin(\omega_a t) \sin(\omega_b t)] + \\ &= [\cos(\omega_a t) \sin(\omega_b t) - A \cos(\omega_a t) \sin(\omega_b t) + B \cos(\omega_a t) \cos(\omega_b t)] = \\ &= \sin(\omega_a t) [\cos(\omega_b t) + A \cos(\omega_b t) + B \sin(\omega_b t)] + \\ &= \cos(\omega_a t) [\sin(\omega_b t) - A \sin(\omega_b t) + B \cos(\omega_b t)] = \\ &= \sin(\omega_a t) [B \sin(\omega_b t) + (1 + A) \cos(\omega_b t)] + \\ &= \cos(\omega_a t) [(1 - A) \sin(\omega_b t) + B \cos(\omega_b t)] = \\ &= C \sin(\omega_a t) \cos(\omega_b t + \theta) + D \cos(\omega_a t) \sin(\omega_b t + \psi) \end{aligned}$$

όπου, βλέπε (4.1.2),

$$C = \sqrt{(1 + A)^2 + B^2} \text{ και } \theta = \arctan\left(-\frac{B}{1 + A}\right), \quad (4.1.7)$$

και, βλέπε (4.1.3),

$$D = \sqrt{(1 - A)^2 + B^2} \text{ and } \psi = \arctan\left(\frac{B}{1 - A}\right), \quad (4.1.8)$$

with $A \neq \pm 1$

Από τις (4.1.4), (4.1.7) η (4.1.8) λαμβάνει την ακόλουθη μορφή:

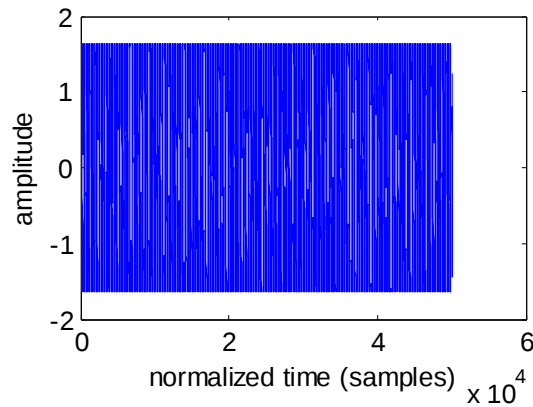
$$C = \sqrt{1 + K^2 + 2K\cos\varphi}, \quad \theta = \arctan\left(-\frac{K\sin\varphi}{1 + K\cos\varphi}\right), \quad (4.1.7)'$$

$$D = \sqrt{1 + K^2 - 2K\cos\varphi}, \quad \psi = \arctan\left(\frac{K\sin\varphi}{1 - K\cos\varphi}\right), \quad (4.1.8)'$$

Τέλος, χρησιμοποιώντας τις (4.1.5), (4.1.7)' και την (4.1.8)', έχουμε ότι

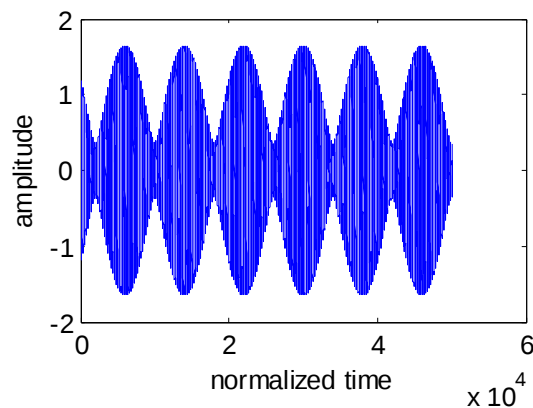
$$\begin{aligned} & \sin(\omega_1 t) + K\sin(\omega_2 t + \varphi) = \\ & \sqrt{1 + K^2 + 2K\cos\varphi} * \cos\left[\frac{\omega_1 - \omega_2}{2}t + \arctan\left(-\frac{K\sin\varphi}{1 + K\cos\varphi}\right)\right] * \sin\left(\frac{\omega_1 + \omega_2}{2}t\right) + \\ & \sqrt{1 + K^2 - 2K\cos\varphi} * \sin\left[\frac{\omega_1 - \omega_2}{2}t + \arctan\left(\frac{K\sin\varphi}{1 - K\cos\varphi}\right)\right] * \cos\left(\frac{\omega_1 + \omega_2}{2}t\right), \quad (4.1.9) \end{aligned}$$

Προκειμένου να οπτικοποιήσουμε την κυματομορφή της (4.1.9) απεικονίζουμε 50000 δείγματα του σήματος $\sin(2\pi 200t) + 0.65\sin(2\pi 550t + 1.57)$ με συχνότητα δειγματοληψίας τα 8000Hz:



Εικόνα 4.1.2 50000 του $\sin(2\pi 200t) + 0.65\sin(2\pi 550t + 1.57)$ με $F_s = 8000\text{Hz}$

Παρόλα αυτά, πολύ ενδιαφέρουσα είναι η κυματομορφή της (4.1.9), όταν οι δύο συχνότητες είναι πολύ κοντά η μία στην άλλη. Επιδεικνύουμε αυτό απεικονίζοντας 50000 δείγματα του σήματος $\sin(2\pi 200t) + 0.65\sin(2\pi 201t + 1.57)$ συχνότητα δειγματοληψίας τα 8000Hz όπως προηγουμένως:



Εικόνα 4.1.3 50000 δείγματα του $\sin(2\pi 200t) + 0.65\sin(2\pi 201t + 1.57)$ με $F_s = 8000\text{Hz}$

Το σήμα αυτό είναι το γνωστό διακρότημα, το οποίο απαντάται πολύ συχνά στην κυματική φυσική και στις τηλεπικοινωνίες (π.χ. διαμόρφωση πλάτους). Πράγματι, παρατηρώντας τους όρους της (4.1.9), βλέπουμε ότι το σήμα αποτελείται από το άθροισμα δύο ημιτόνων, των οποίων το πλάτος διαμορφώνεται από ένα άλλο ημίτονο πολύ μικρότερης συχνότητας. Συνεπώς κάθε φορά που υπερτίθενται δύο ημίτονα με συχνότητες πολύ κοντά η μία στην άλλη, το αποτέλεσμα είναι ένα διακρότημα, με κυματομορφή όπως στην εικόνα 4.1.3. (σημείωση: η απόδειξη της (4.1.9) είναι εξαιρετικά δυσεύρετη στη βιβλιογραφία. Η απόδειξη που δόθηκε είναι του υπογράφοντος)

Είμαστε πολύ κοντά στο να κάνουμε μία σημαντικότερη παρατήρηση. Για το σκοπό αυτό, ας κάνουμε ένα απλό πείραμα πραγματικού χρόνου. Χρησιμοποιούμε δύο συμβατικά cd-players. Κάθε cd-player οδηγεί ένα ηχείο. Τα δύο ηχεία είναι τοποθετημένα το ένα απέναντι στο άλλο, σε απόσταση 30cm. Τώρα, δημιουργούμε από το MATLAB δείγματα ενός ημιτόνου συχνότητας 200Hz δειγματοληπτημένο στα 8000Hz. Το ημίτονο αυτό είναι περιοδικό με περίοδο 40 δειγμάτων. Αυτά τα 40 δείγματα είναι τα ακόλουθα:

0
0.1564
0.3090
0.4540
0.5878
0.7071
0.8090
0.8910
0.9511
0.9877
1.0000
0.9877
0.9511
0.8910
0.8090
0.7071
0.5878
0.4540
0.3090
0.1564
-0.0000
-0.1564
-0.3090
-0.4540
-0.5878
-0.7071
-0.8090
-0.8910
-0.9511
-0.9877
-1.0000
-0.9877
-0.9511
-0.8910
-0.8090
-0.7071
-0.5878
-0.4540
-0.3090
-0.1564

Χρησιμοποιώντας αυτά τα δείγματα, δημιουργούμε ένα wav αρχείο το οποίο περιέχει το παραπάνω ημίτονο σε διάρκεια 2 λεπτών. Στη συνέχεια δημιουργούμε ένα cd με αυτό το σήμα και φτιάχνουμε ένα ακριβές αντίγραφο αυτού. Τοποθετούμε τα δύο cd, ένα σε κάθε cd-player. Πατάμε το “play” του πρώτου cd-player και στη συνέχεια κάνουμε το ίδιο για το δεύτερο cd-player. Η έξοδος κάθε ηχείου είναι πίεση αέρα συχνότητας 200Hz. Επειδή οι ρυθμίσεις της έντασης είναι διαφορετικές για τα δύο cd-player, τα πλάτη των δύο εκπεμπόμενων ημιτόνων είναι διαφορετικά. Και επειδή το “play” του δεύτερου cd-player πατήθηκε έπειτα από ένα τυχαίο χρονικό διάστημα από το πάτημα του “play” του πρώτου cd-player, τα δύο εκπεμπόμενα ημίτονα θα έχουν μία τυχαία διαφορά φάσης. Τα δύο σήματα πίεσης αέρα υπερτίθενται και σύμφωνα με την (4.1.1) το

4.2 Προσαρμοζόμενα Φίλτρα

4.2.1 Εισαγωγή στα Προσαρμοζόμενα Φίλτρα

Οι αλγόριθμοι ενεργού θορύβου χρησιμοποιούν εκτενώς προσαρμοζόμενα φίλτρα. Τα προσαρμοζόμενα φίλτρα είναι ψηφιακά φίλτρα. Τα συμβατικά ψηφιακά φίλτρα είναι γραμμικά και χρονικά αμετάβλητα. Αντιθέτως, τα προσαρμοζόμενα φίλτρα είναι χρονικά μεταβαλλόμενα, δηλαδή παράμετροι όπως η ζώνη διέλευσης και η συχνότητα συντονισμού τους, αλλάζουν με το χρόνο. Επιπλέον τα προσαρμοζόμενα φίλτρα δύνανται να είναι είτε γραμμικά, είτε μη γραμμικά. Αυτό επιτυγχάνεται επιτρέποντας στους συντελεστές του προσαρμοζόμενου φίλτρου να μεταβάλλονται χρονικά, ρυθμιζόμενοι αυτόματα από κάποιον προσαρμοστικό αλγόριθμο. Ως αποτέλεσμα, τα προσαρμοζόμενα φίλτρα μπορούν να χρησιμοποιηθούν επιτυχώς, όταν η ακριβής προδιαγραφή φιλτραρίσματος είναι άγνωστη ή μη στάσιμη. Όπως γίνεται κατανοητό στο κεφάλαιο 2, η ακριβής επεξεργασία σήματος, η οποία πρέπει να εφαρμοστεί στα σήματα τα προερχόμενα από τα μικρόφωνα, προκειμένου ο ANC ελεγκτής να λειτουργήσει σωστά, είναι και άγνωστη και χρονικά μεταβαλλόμενη. Από αυτή και μόνη την παρατήρηση, γίνεται σαφές η τεράστια σημασία των προσαρμοζόμενων φίλτρων στην ANC τεχνολογία. Βεβαίως, τα προσαρμοζόμενα φίλτρα έχουν χρησιμοποιηθεί και χρησιμοποιούνται σε ένα ευρύ φάσμα εφαρμογών, όπως αναγνώριση συστήματος και αντιστάθμιση διαύλου. Η έννοια των προσαρμοζόμενων και προσαρμοστικών μηχανισμών εξετάζονταν ήδη από το 1950, αλλά δεν εισέπραξε μεγάλη προσοχή από τους μηχανικούς, έως την εργασία των Widrow και Hoff, το 1960. Οι Widrow και Hoff ανέπτυξαν την αυθεντική μορφή του LMS αλγορίθμου, ο οποίος σήμερα θεωρείται ο ακρογωνιαίος λίθος της σύγχρονης θεωρίας προσαρμοζόμενων φίλτρων.

Ως ψηφιακά φίλτρα, τα προσαρμοζόμενα φίλτρα μπορούν να υλοποιηθούν ως FIR και IIR δομές ή ακόμα και ως δομές πλέγματος. Σε αυτή την εργασία εξετάζουμε μόνο την FIR δομή, εξαιτίας της απλότητάς της και των ευεργετικών ιδιοτήτων της όσον αφορά την ευστάθεια (τα FIR φίλτρα είναι φίλτρα χωρίς πόλους).

Θα συμβολίζουμε την έξοδο ενός FIR προσαρμοζόμενου φίλτρου με L συντελεστές ως $y(i)$, την είσοδό του ως

$$\mathbf{u}(i) = [u(i), u(i-1), u(i-2), \dots, u(i-L+1)]^T, \quad (4.2.1)$$

και τους συντελεστές του ως

$$\mathbf{w}_i = [w_0(i), w_1(i), w_2(i), \dots, w_{L-1}(i)]^T, \quad (4.2.2)$$

Το διάνυσμα $\mathbf{w}_i$ θα ονομάζεται εφεξής διάνυσμα βάρους. Παρατηρείστε ότι στην (4.2.2) οι συντελεστές του διανύσματος βάρους $\mathbf{w}_i$ είναι συνάρτηση του δείκτη κανονικοποιημένου χρόνου i , το οποίο υπονοεί ότι μεταβάλλονται με το χρόνο. Η έξοδος του προσαρμοζόμενου φίλτρου δίνεται από την εξίσωση

$$y(i) = \mathbf{w}_i^T \mathbf{u}(i), \quad (4.2.3)$$

Αυτό, το οποίο απομένει για την πλήρη περιγραφή του προσαρμοζόμενου φίλτρου των εξισώσεων (4.2.1)-(4.2.3), είναι ο καθορισμός του αλγορίθμου, ο οποίος ρυθμίζει ή καλύτερα προσαρμόζει το διάνυσμα βάρους $\mathbf{w}_i$. Το θέμα αυτό είναι αρκετά περίπλοκο και αναλύεται στο υπόλοιπο αυτού του κεφαλαίου.

Από την (4.2.3) βλέπουμε ότι η επεξεργασία που εφαρμόζεται στο σήμα εισόδου $u(i)$ είναι χρονικά μεταβαλλόμενη και γραμμική. Τα προσαρμοζόμενα φίλτρα, τα οποία λειτουργούν γραμμικά, όπως στην (4.2.3), έχουν τις ρίζες τους στη θεωρία βέλτιστης γραμμικής πρόβλεψης, και από εκεί οφείλουμε να ξεκινήσουμε την ανάλυσή μας, προκειμένου να κατανοήσουμε τι είναι αυτό, το οποίο ένα προσαρμοζόμενο φίλτρο πραγματικά κάνει.

4.2.2 Εισαγωγή στη Θεωρία Πρόβλεψης

Η θεωρία της πρόβλεψης επιχειρεί να δώσει απάντηση στο ερώτημα «πώς μπορούμε να προβλέψουμε την τιμή μίας τυχαίας μεταβλητής y βασιζόμενοι στις τιμές μίας άλλης τυχαίας μεταβλητής x ». Σε αυτήν την περίπτωση η συνάρτηση $h(x)$ ονομάζεται προβλέπτης της τυχαίας μεταβλητής y από την τυχαία μεταβλητή x , εάν

$$\hat{y} = h(x), \quad (4.2.4)$$

είναι μία πρόβλεψη για την τυχαία μεταβλητή y . Η ποσότητα

$$e = y - \hat{y} = y - h(x), \quad (4.2.5)$$

ονομάζεται σφάλμα πρόβλεψης. Όπως θα ήταν διαισθητικά αναμενόμενο, ένας καλός προβλέπτης (4.2.4) είναι ένας προβλέπτης, του οποίου το σφάλμα πρόβλεψης (4.2.5) είναι ικανοποιητικά μικρό. Αλλά πώς βρίσκουμε μία φόρμουλα για το $\hat{y}$? Και ειδικότερα, πώς βρίσκουμε τη βέλτιστη φόρμουλα για το $\hat{y}$? Και εάν η φόρμουλα είναι βέλτιστη, τότε υπό ποια έννοια είναι βέλτιστη? Αυτά τα ερωτήματα είναι εύλογα και ανήκουν στην καρδιά κάθε προβλήματος πρόβλεψης. Για να απαντήσουμε αυτά τα ερωτήματα, χρειάζεται πρώτα να επιλέξουμε μία συνάρτηση κόστους, η οποία να δίνει ένα μέγεθος του σφάλματος πρόβλεψης. Ο προβλέπτης $\hat{y}$, ο οποίος θα προκύψει, θα είναι βέλτιστος μόνο υπό την έννοια ότι θα ελαχιστοποιεί τη δεδομένη συνάρτηση κόστους. Διαφορετικές συναρτήσεις κόστους οδηγούν εν γένει σε διαφορετικούς προβλέπτες, κάθε ένας από τους οποίους είναι βέλτιστος με τον τρόπο του. Το περισσότερο δημοφιλές κριτήριο σχεδίασης ενός προβλέπτη είναι το μέσο τετραγωνικό σφάλμα (Mean Square Error-(MSE)), το οποίο δίνεται από την εξίσωση

$$J(e) = Ee^2, \quad (4.2.6)$$

όπου ο τελεστής E δηλώνει την αναμενόμενη τιμή του ορίσματος του και το e δίνεται από την (4.2.5). Πρέπει να σημειώσουμε ότι η J είναι μία βαθμωτή συνάρτηση του e . Ο προβλέπτης $\hat{y}_{optimal}$, ο οποίος προκύπτει, ονομάζεται Προβλέπτης Least-Mean-Squares (LMS) του y και ικανοποιεί την ακόλουθη συνθήκη:

$$E(y - \hat{y}_{optimal})^2 = \min_{\hat{y}} E(y - \hat{y})^2, \quad (4.2.7)$$

Θεώρημα 4.2.1 Ο LMS προβλέπτης του y δοθέντος του x είναι η υπό συνθήκη αναμενόμενη τιμή του y δοθέντος του x , δηλαδή

$$\hat{y}_{optimal} = E(y|x) = \int y f_{y|x}(y|x) dy, \quad (4.2.8)$$

Ο προβλέπτης αυτός δεν είναι πολωμένος, δηλαδή,

$$E\hat{y}_{optimal} = Ey, \quad (4.2.9)$$

και το προκύπτον ελάχιστο κόστος είναι

$$E(y - \hat{y}_{optimal})^2 = \sigma_y^2 - \sigma_{\hat{y}_{optimal}}^2, \quad (4.2.10)$$

Παρόλα αυτά, σε πρακτικές εφαρμογές είναι σχεδόν αδύνατον να χρησιμοποιήσουμε το παραπάνω θεώρημα. Ο λόγος είναι ότι απαιτείται πλήρης γνώση των στατιστικών ιδιοτήτων της προς πρόβλεψη μεταβλητής y . Αυτό μπορούμε να το δούμε από την (4.2.8) όπου η υπό συνθήκη πυκνότητα πιθανότητας $f_{y|x}(y|x)$ εμφανίζεται. Τέτοια γνώση δεν είναι συνήθως διαθέσιμη στην πράξη, εκτός από πολύ ειδικές περιπτώσεις, όπως είναι τα Gaussian δεδομένα. Εξαιτίας αυτού, στη συνέχεια εξετάζουμε μόνο την υποκλάση των γραμμικών προβλεπτών. Όπως θα δείξουμε στη συνέχεια, το μόνο που χρειαζόμαστε για να σχεδιάσουμε έναν βέλτιστο γραμμικό προβλέπτη υπό την LMS έννοια, είναι οι στατιστικές ροπές δεύτερης τάξης των τυχαίων μεταβλητών x και y , δηλαδή, τα Ex , Ey , Ex^2 , Ey^2 και Exy . Αυτές οι στατιστικές ροπές μπορούν εύκολα να εξαχθούν έπειτα από στατιστική επεξεργασία των δεδομένων.

4.2.3 Στοιχεία Βέλτιστης Γραμμικής Πρόβλεψης

Η μορφή ενός γραμμικού προβλέπτη είναι

$$\hat{y} = h(x) = \mathbf{k}x + b, \quad (4.2.11)$$

Εάν θέλουμε να είμαστε περισσότερο ακριβείς, αυτός είναι ένας *affine* προβλέπτης, εξαιτίας της παρουσίας του όρου b . Εάν ο b είναι μηδέν τότε η (4.2.11) περιγράφει έναν καθαρά γραμμικό προβλέπτη. Επιπλέον το x είναι ένα διάνυσμα στήλη, το οποίο περιέχει τις L πιο πρόσφατες παρατηρήσεις του x . Το $\mathbf{k}$ είναι ένα διάνυσμα γραμμή με τον ίδιο αριθμό στοιχείων με το x . Χωρίς βλάβη της γενικότητας θα υποθέσουμε ότι οι μεταβλητές x και y έχουν μέση τιμή μηδέν, δηλαδή

$$Ex = 0, \quad Ey = 0, \quad (4.2.12)$$

Αργότερα, θα κάνουμε μία μικρή διόρθωση για μεταβλητές μη μηδενικής μέσης τιμής. Αφού η (4.2.12) ισχύει, βλέπουμε κατευθείαν από την (4.2.11) ότι $b = 0$. Έτσι, για κάθε τυχαία μεταβλητή μηδενικής μέσης τιμής, κάθε affine προβλέπτης της οφείλει να είναι γραμμικός, δηλαδή

$$\hat{y} = \mathbf{k}x, \quad (4.2.13)$$

Με βάση το LMS κριτήριο, ενδιαφερόμαστε για το βέλτιστο διάνυσμα $\mathbf{k}_{optimal}$, το οποίο ελαχιστοποιεί τη συνάρτηση κόστους (4.2.6). Ας αναπτύξουμε την (4.2.6) χρησιμοποιώντας την (4.2.13):

$$\begin{aligned} J &= E(y - \hat{y})^2 = E(y - \mathbf{k}x)^2 = E(y - \mathbf{k}x)(y - \mathbf{k}x) = \\ &E(y^2 - y\mathbf{k}x - \mathbf{k}xy + \mathbf{k}x\mathbf{k}x) = E(y^2 - 2\mathbf{k}xy + \mathbf{k}x\mathbf{x}^T\mathbf{k}^T) = \\ &E(y^2) - 2\mathbf{k}E(xy) + \mathbf{k}E(\mathbf{x}\mathbf{x}^T)\mathbf{k}^T \Rightarrow \\ J &= \sigma_y^2 - 2\mathbf{k}r_{xy} + \mathbf{k}R_x\mathbf{k}^T, \quad (4.2.14) \end{aligned}$$

όπου σ_y^2 είναι η διασπορά του y , $E(y^2)$, r_{xy} είναι το διάνυσμα ετεροσυσχέτισης μεταξύ του x και του y , $E(xy)$ και R_x είναι ο πίνακας αυτοσυσχέτισης του x , $E(\mathbf{x}\mathbf{x}^T)$. Εάν οι τυχαίες μεταβλητές x και y είναι στάσιμες με την ευρεία έννοια τότε τα r_{xy} και R_x είναι σταθερά ως προς το χρόνο.

Συνεπώς για ένα δεδομένο πρόβλημα πρόβλεψης τα r_{xy} , R_x είναι σ_y^2 είναι σταθερές, δηλαδή είναι παράμετροι του προβλήματος. Σε αντίθεση, το $\mathbf{k}$ είναι μία μεταβλητή, της οποίας την τιμή εμείς την αποφασίζουμε. Ως αποτέλεσμα, μπορούμε να θεωρήσουμε ότι η συνάρτηση κόστους J της (4.2.14) είναι μόνο συνάρτηση του $\mathbf{k}$. Αυτό δηλώνεται emphaticά ως εξής:

$$J(\mathbf{k}) = \sigma_y^2 - 2\mathbf{k}r_{xy} + \mathbf{k}R_x\mathbf{k}^T, \quad (4.2.15)$$

Για κάθε τιμή του διανύσματος $\mathbf{k}$, η (4.2.15) επιστρέφει μία βαθμωτή τιμή, η οποία είναι το μέσο τετραγωνικό σφάλμα της πρόβλεψης (4.2.13). Προφανώς, ενδιαφερόμαστε για το $\mathbf{k}$, το οποίο ελαχιστοποιεί την $J(\mathbf{k})$. Το επόμενο θεώρημα υποδεικνύει τη διαδικασία, την οποία πρέπει να ακολουθήσουμε προκειμένου να προσδιορίσουμε από την (4.2.15) το βέλτιστο διάνυσμα $\mathbf{k}_{optimal}$.

Θεώρημα 4.2.2 Έστω ότι η $J(\mathbf{k}) = J(k_1, k_2, \dots, k_L)$ έχει μερικές παραγώγους έως και τρίτης τάξης, οι οποίες είναι συνεχείς σε μία περιοχή του $\mathbf{k} = \mathbf{k}_{candidate}$. Έστω επίσης ότι οι πρώτες μερικές παράγωγοι του $J(\mathbf{k})$ είναι μηδέν στο σημείο $\mathbf{k} = \mathbf{k}_{candidate}$, και έστω

$$J_{ij} = \frac{\partial^2 J(\mathbf{k})}{\partial k_i \partial k_j}$$

υπολογισμένο στο σημείο $\mathbf{k} = \mathbf{k}_{candidate}$. Τότε η $J(\mathbf{k})$ έχει τοπικό ελάχιστο (ή τοπικό μέγιστο) στο σημείο $\mathbf{k} = \mathbf{k}_{candidate}$, εάν ο συμμετρικός πίνακας $H = (J_{ij})$ είναι θετικά ορισμένος (ή αρνητικά ορισμένος).

Ένας συμμετρικός πίνακας H είναι θετικά ορισμένος εάν $\mathbf{x}^T H \mathbf{x} > 0$ για όλα τα διανύσματα στήλης $\mathbf{x}$, εκτός όταν $\mathbf{x} = \mathbf{0}$.

Ως επιπλέον παρατήρηση αναφέρουμε ότι το θεώρημα 4.2.2 είναι ουσιαστικά μία ανάλογη επέκταση του κλασικού θεωρήματος του Fermat, σύμφωνα με το οποίο εάν $\frac{df(x)}{dx}|_{x_0} = 0$, τότε η $f(x)$ έχει τοπικό ελάχιστο (μέγιστο) στο x_0 αν και μόνο αν $\frac{d^2 f(x)}{dx^2}|_{x_0} > 0$ ($\frac{d^2 f(x)}{dx^2}|_{x_0} < 0$).

Σύμφωνα με το θεώρημα 4.2.2, εάν βρούμε ένα υποψήφιο διάνυσμα $\mathbf{k}_{candidate}$, στο οποίο να μηδενίζονται οι πρώτες μερικές παράγωγοι της $J(\mathbf{k})$, τότε αρκεί να ελέγξουμε εάν ο πίνακας των δεύτερων μερικών παραγώγων είναι θετικά ορισμένος στο $\mathbf{k}_{candidate}$. Για να να ξεκινήσουμε με τον προσδιορισμό του $\mathbf{k}_{candidate}$, διαφορίζουμε την $J(\mathbf{k})$ ως προς το $\mathbf{k}$ και στη συνέχεια θέτουμε την παράγωγο ίση με το μηδέν. Συνεπώς έχουμε

$$\frac{\partial J(\mathbf{k})}{\partial \mathbf{k}} \equiv \nabla_{\mathbf{k}} J(\mathbf{k}) = \left[\frac{\partial J(\mathbf{k})}{\partial k_1}, \frac{\partial J(\mathbf{k})}{\partial k_2}, \dots, \frac{\partial J(\mathbf{k})}{\partial k_L} \right]^T, \quad (4.2.16)$$

και η παράγωγος είναι:

$$\frac{\partial J(\mathbf{k})}{\partial \mathbf{k}} \equiv \nabla_{\mathbf{k}} J(\mathbf{k}) = -2\mathbf{r}_{xy} + 2R_x \mathbf{k}^T, \quad (4.2.17)$$

Θέτοντας τώρα την (4.2.17) ίση με το μηδέν έχουμε:

$$R_x \mathbf{k}_{candidate}^T = \mathbf{r}_{xy}, \quad (4.2.18)$$

και εάν ο R_x είναι μη ιδιάζων

$$\mathbf{k}_{candidate}^T = R_x^{-1} \mathbf{r}_{xy}, \quad (4.2.19)$$

Όπως φαίνεται, το διάνυσμα, το οποίο μηδενίζει την παράγωγο της συνάρτησης κόστους και δίνεται από την (4.2.19), συμβολίζεται ως $\mathbf{k}_{candidate}$ και όχι ως $\mathbf{k}_{optimal}$. Αυτό συμβαίνει, επειδή μέχρι στιγμής έχουμε δείξει ότι το διάνυσμα $R_x^{-1} \mathbf{r}_{xy}$ είναι απλώς υποψήφιο για τη συνθήκη

$$J(\mathbf{k}_{optimal}) = \min_{\mathbf{k}} J(\mathbf{k}), \quad (4.2.20)$$

δηλαδή,

$$J(\mathbf{k}_{candidate}) = \min_{\mathbf{k}} J(\mathbf{k})?, \quad (4.2.21)$$

Αξίζει να αναφέρουμε ότι το σημείο αυτό πολύ συχνά αποσιωπάται στη βιβλιογραφία της επεξεργασίας σημάτων, δηλαδή το διάνυσμα $R_x^{-1} \mathbf{r}_{xy}$ θεωρείται χωρίς περαιτέρω ελέγχους, ως το βέλτιστο διάνυσμα. Αυτό βεβαίως συμβαίνει ώστε να αποφευχθεί επιπλέον μαθηματική ανάλυση και τεκμηρίωση. Προχωρούμε λοιπόν στον έλεγχο της ισχύος της δεύτερης συνθήκης του θεωρήματος 4.2.2. Η ύπαρξη των μερικών παραγώγων έως και τρίτης τάξης εύκολα μπορεί να διαπιστωθεί από την (4.2.15). Απομένει να βρούμε τον πίνακα H και να αποφανθούμε εάν είναι θετικά ορισμένος ή όχι.

Είδαμε προηγουμένως, ότι όταν ο τελεστής gradient $\frac{\partial}{\partial \mathbf{k}} \equiv \nabla_{\mathbf{k}}$ δεχθεί ως όρισμα μία βαθμωτή συνάρτηση διανυσματικού ορίσματος $J(\mathbf{k})$, παραγάγει ένα διάνυσμα στήλη, του οποίου τα στοιχεία είναι βαθμωτές συναρτήσεις του διανυσματικού ορίσματος (4.2.16). Μπορούμε με συνέπεια να επεκτείνουμε αυτή την ιδέα και να ορίσουμε την έξοδο του gradient τελεστή ως ένα πίνακα βαθμωτών συναρτήσεων διανυσματικού ορίσματος, όταν η είσοδός του είναι ένα διάνυσμα γραμμή βαθμωτών συναρτήσεων διανυσματικού ορίσματος. Περισσότερο επίσημα:

$$\nabla_{\mathbf{k}} [J_1(\mathbf{k}), J_2(\mathbf{k}), \dots, J_N(\mathbf{k})] =$$

$$\begin{bmatrix} \frac{\partial J_1(\mathbf{k})}{\partial k_1} & \frac{\partial J_2(\mathbf{k})}{\partial k_1} & \dots & \frac{\partial J_N(\mathbf{k})}{\partial k_1} \\ \frac{\partial J_1(\mathbf{k})}{\partial k_2} & \frac{\partial J_2(\mathbf{k})}{\partial k_2} & \dots & \frac{\partial J_N(\mathbf{k})}{\partial k_2} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial J_1(\mathbf{k})}{\partial k_L} & \frac{\partial J_2(\mathbf{k})}{\partial k_L} & \dots & \frac{\partial J_N(\mathbf{k})}{\partial k_L} \end{bmatrix}, \quad (4.2.23)$$

Ας κάνουμε τώρα την αντικατάσταση $J_i(\mathbf{k}) = \frac{\partial J(\mathbf{k})}{\partial k_i}$, $i = 1, 2, \dots, L$ στην (4.2.23) :

$$\nabla_{\mathbf{k}} \left[\frac{\partial J(\mathbf{k})}{\partial k_1}, \frac{\partial J(\mathbf{k})}{\partial k_2}, \dots, \frac{\partial J(\mathbf{k})}{\partial k_L} \right] =$$

$$\begin{bmatrix} \frac{\partial^2 J(\mathbf{k})}{\partial k_1^2} & \frac{\partial^2 J(\mathbf{k})}{\partial k_1 \partial k_2} & \cdots & \frac{\partial^2 J(\mathbf{k})}{\partial k_1 \partial k_L} \\ \frac{\partial^2 J(\mathbf{k})}{\partial k_2 \partial k_1} & \frac{\partial^2 J(\mathbf{k})}{\partial k_2^2} & & \frac{\partial^2 J(\mathbf{k})}{\partial k_2 \partial k_L} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 J(\mathbf{k})}{\partial k_L \partial k_1} & \frac{\partial^2 J(\mathbf{k})}{\partial k_L \partial k_2} & \cdots & \frac{\partial^2 J(\mathbf{k})}{\partial k_L^2} \end{bmatrix} =$$

$$\nabla_{\mathbf{k}}\{(\nabla_{\mathbf{k}}J(\mathbf{k}))^T\}, \quad (4.2.24)$$

Παρατηρούμε ότι οι συντελεστές του πίνακα $\nabla_{\mathbf{k}}\{(\nabla_{\mathbf{k}}J(\mathbf{k}))^T\}$ είναι οι ίδιοι με τους συντελεστές του πίνακα του θεωρήματος 4.2.2. Ειδικότερα ο πίνακας $\nabla_{\mathbf{k}}\{(\nabla_{\mathbf{k}}J(\mathbf{k}))^T\}$ είναι ένας αρκετά γνωστός πίνακας, ο λεγόμενος Hessian matrix της $J(\mathbf{k})$. Σύμφωνα με το θεώρημα 4.2.2, εάν αυτός ο πίνακας είναι θετικά ορισμένος τότε η (4.2.21) ισχύει και το διάνυσμα $R_x^{-1}\mathbf{r}_{xy}$ ελαχιστοποιεί τη συνάρτηση κόστους. Από την (4.2.17) εύκολα προκύπτει ότι

$$\frac{\partial^2 J(\mathbf{k})}{\partial \mathbf{k}^2} \equiv \nabla_{\mathbf{k}}\{(\nabla_{\mathbf{k}}J(\mathbf{k}))^T\} = R_x, \quad (4.2.25)$$

Αλλά ο R_x είναι θετικά ορισμένος, εφόσον για κάθε μη μηδενικό διάνυσμα στήλη $\mathbf{v}$ ισχύει ότι

$$\begin{aligned} \mathbf{v}^T R_x \mathbf{v} &= \mathbf{v}^T E(\mathbf{x}\mathbf{x}^T) \mathbf{v} = E(\mathbf{v}^T \mathbf{x}\mathbf{x}^T \mathbf{v}) = E(\mathbf{v}^T \mathbf{x}(\mathbf{v}^T \mathbf{x})^T) = \\ &= E\|\mathbf{v}^T \mathbf{x}\|^2 > 0 \Leftrightarrow \mathbf{v}^T R_x \mathbf{v} > 0, \end{aligned} \quad (4.2.26)$$

Αποδείξαμε συνεπώς ότι πράγματι

$$R_x \mathbf{k}_{optimal}^T = \mathbf{r}_{xy}, \quad (4.2.27)$$

Οι εξισώσεις (4.2.27) έχουν μελετηθεί εκτεταμένα στη βιβλιογραφία. Οι εξισώσεις αυτές είναι οι διάσημες **κανονικές εξισώσεις**. Τώρα, το $\mathbf{k}_{optimal}$ μπορεί να θεωρηθεί ως οι συντελεστές ενός κλασικού FIR φίλτρου με L συντελεστές, το οποίο προβλέπει με βέλτιστο τρόπο, υπό την έννοια των μέσων ελάχιστων τετραγώνων, τις τιμές της μεταβλητής y από τα διακριτά δείγματα της μεταβλητής x . Το μέσο τετραγωνικό σφάλμα (MSE) του βέλτιστου αυτού γραμμικού προβλέπτη μπορεί να ευρεθεί αντικαθιστώντας την (4.2.27) στην (4.2.15). Επειδή ο προβλέπτης είναι βέλτιστος, αυτό είναι και το ελάχιστο μέσο τετραγωνικό σφάλμα της γραμμικής πρόβλεψης του y από το x :

$$J_{min} = \sigma_y^2 - 2\mathbf{k}\mathbf{r}_{xy} + \mathbf{k}\mathbf{r}_{xy} = \sigma_y^2 - \mathbf{k}\mathbf{r}_{xy}, \quad (4.2.28)$$

Μία πολύ ενδιαφέρουσα παρατήρηση είναι ότι εάν ο R_x είναι μη ιδιάζων και $\mathbf{r}_{xy} = 0$, βλέπουμε από την (4.2.27) ότι $\mathbf{k}_{optimal} = 0$, το οποίο σημαίνει ότι δεν υφίσταται προβλέπτης σε αυτήν την περίπτωση για το y . Αυτό βεβαίως, είναι διαισθητικά αναμενόμενο, αφού εάν $\mathbf{r}_{xy} = 0$, τότε δεν υπάρχει απολύτως καμία συσχέτιση μεταξύ y και x , και συνεπώς δεν υπάρχει κανένας εφικτός τρόπος εξαγωγής οποιασδήποτε πληροφορίας για το y από το x .

Τέλος, εάν οι τυχαίες μεταβλητές x και y είναι πολωμένες, δηλαδή $E x, E y \neq 0$, τότε οι κεντροποιημένες τυχαίες μεταβλητές $(x - \bar{x})$ και $(y - \bar{y})$, με $\bar{x} = E x$ και $\bar{y} = E y$, είναι μη πολωμένες. Ο βέλτιστος γραμμικός προβλέπτης σε αυτήν την περίπτωση είναι

$$\begin{aligned} \hat{y} &= \bar{y} + ([E\{(x - \bar{x})(x - \bar{x})^T\}]^{-1} E\{(x - \bar{x})(y - \bar{y})\})^T (x - \bar{x}) \Rightarrow \\ &= \bar{y} + (R_{(x-\bar{x})}^{-1} r_{(x-\bar{x})(y-\bar{y})})^T (x - \bar{x}), \end{aligned} \quad (4.2.29)$$

4.2.4 Επιλύση των Κανονικών Εξισώσεων

Στο προηγούμενο εδάφιο είδαμε ότι το διάνυσμα, το οποίο ελαχιστοποιεί το μέσο τετραγωνικό σφάλμα της γραμμικής πρόβλεψης του y από το x , δίνεται από την εξίσωση:

$$\mathbf{k}_{optimal} = R_x^{-1} \mathbf{r}_{xy}, \quad (4.2.30)$$

Η άμεση επίλυση της (4.2.30) ως προς το $\mathbf{k}_{optimal}$ παρουσιάζει δύο σημαντικά προβλήματα. Το πρώτο πρόβλημα αφορά την αυξημένη υπολογιστική πολυπλοκότητα της αντιστροφής του πίνακα R_x . Υπάρχουν διάφοροι αλγόριθμοι αντιστροφής πίνακα,

με τον πιο δημοφιλή να είναι η *Gaussian Elimination* πολυπλοκότητας $O(N^3)$ για έναν $N \times N$ πίνακα. Αυτή η πολυπλοκότητα δεν είναι η βέλτιστη. Για παράδειγμα η μέθοδος του *Strassen* επιτυγχάνει πολυπλοκότητα $O(N^{2.807})$, ενώ το προσφατό κάτω όριο πολυπλοκότητας είναι περίπου $O(N^{2.376})$. Παράλληλα, ο πίνακας αυτοσυσχέτισης μίας στάσιμης στοχαστικής ανέλιξης είναι συμμετρικός και έχει δομή Toeplitz, δηλαδή κάθε διαγώνιος του πίνακα αποτελείται από ίσα στοιχεία. Συνεπώς υπάρχουν αποδοτικοί αλγόριθμοι προς αντιστροφή συμμετρικών Toeplitz πινάκων, με τους οποίους επιτυγχάνεται επιπρόσθετη μείωση της πολυπλοκότητας. Για παράδειγμα ο αλγόριθμος *Levinson-Durbin* επιλύει τις κανονικές εξισώσεις με πολυπλοκότητα $O(N^2)$. Ακόμα και υπό αυτές τις συνθήκες ελαττωμένης πολυπλοκότητας, η αντιστροφή ενός πίνακα είναι μία αρκετά υπολογιστικά επιβαρύνουσα διαδικασία, ειδικά όταν οι διαστάσεις του πίνακα είναι μεγάλες. Για λόγους κατανόησης, αναφέρουμε ότι σε μία τυπική εφαρμογή ενεργού ελέγχου θορύβου, όπως αυτή την οποία εξετάζει η παρούσα εργασία, ο πίνακας αυτοσυσχέτισης ενδέχεται να έχει διαστάσεις $N = 40$ έως $N = 150$, με αποτέλεσμα ο *Levinson-Durbin* αλγόριθμος να απαιτεί περίπου από $40^2 = 1600$ έως $150^2 = 22500$ FLOPS! Γενικά, η αντιστροφή πίνακα αποφεύγεται, όπου αυτό είναι εφικτό, εφαρμόζοντας τεχνικές ψευδο-αντιστροφής, επιτυγχάνοντας ακόμα περισσότερη μείωση της πολυπλοκότητας. Φυσικά, η ψευδοαντιστροφή του R_x έχει ως συνέπεια μία έκπτωση στην απόδοση, καθώς ο προκύπτων προβλέπτης δεν είναι ακριβώς ο βέλτιστος.

Αλλά ακόμα και εάν η πολυπλοκότητα της αντιστροφής του πίνακα αντιμετωπιζόταν ικανοποιητικά, ο υπολογισμός του $k_{optimal}$ από την (4.2.30) προσκρούει στο δεύτερο σημαντικό πρόβλημα του ευθέως υπολογισμού της (4.2.30), το οποίο είναι το ενδεχόμενο ύπαρξης πινάκων R_x με μεγάλο αριθμό σύνταξης. Ο αριθμός σύνταξης ενός πίνακα A ορίζεται ως

$$\chi(A) = \|A\| * \|A^{-1}\|, \quad A \text{ μη ιδιάζων}, \quad (4.2.31)$$

$$\chi(A) = +\infty, \quad A \text{ ιδιάζων}, \quad (4.2.32)$$

Στον παραπάνω ορισμό η νόρμα $\|\cdot\|$ μπορεί να είναι οποιαδήποτε έγκυρη νόρμα πίνακα, όπως η ορίζουσα του πίνακα, η Frobenious ή Gram-Schmidt νόρμα, η l_q νόρμα και ούτω καθεξής. Ο αριθμός σύνταξης μπορεί να θεωρηθεί ως ένα μέτρο της αντίστροφης απόστασης μεταξύ του πίνακα A και του συνόλου των ιδιζόντων πινάκων που κανονικοποιούνται από την $\|A\|$. Πρακτικά, ο αριθμός σύνταξης μας δίνει πληροφορία σχετικά με την ευαισθησία του πίνακα σε αριθμητικές πράξεις. Η ευαισθησία αυτή έχει άμεση επίδραση στην ευστάθεια της λύσης του γραμμικού συστήματος, το οποίο εκπροσωπείται από τον εν λόγω πίνακα. Πίνακες με αριθμό σύνταξης κοντά στο 1 ονομάζονται well-conditioned, ενώ πίνακες με αριθμό σύνταξης πολύ μεγαλύτερο της μονάδας, για παράδειγμα 10^5 , ονομάζονται ill-conditioned. Τώρα, εάν υπολογίσουμε τον αριθμό σύνταξης ενός ερμιτιανού πίνακα A χρησιμοποιώντας τη φασματική νόρμα στην (4.2.31)

$$\text{φασματική νόρμα: } \|A\| = (\text{μέγιστη ιδιοτιμή του } A^T A)^{1/2}, \quad (4.2.33)$$

τότε ο αριθμός σύνταξης απλοποιείται στη μορφή

$$\chi(A) = \left| \frac{\lambda_{max}}{\lambda_{min}} \right|, \quad (4.2.34)$$

όπου λ_{max} είναι η απολύτως μεγαλύτερη ιδιοτιμή του A και λ_{min} είναι η απολύτως μικρότερη ιδιοτιμή του. Αυτή η ειδική μορφή του αριθμού σύνταξης της (4.2.34) ονομάζεται διασπορά ιδιοτιμών του πίνακα A .

Το πρόβλημα με τους ill-conditioned πίνακες είναι ότι ενδέχεται να μην είναι συνετό να εμπιστευθούμε τα αριθμητικά αποτελέσματα, κατά την εξαγωγή των οποίων οι ill-conditioned πίνακες συμμετείχαν, για παράδειγμα την λύση ενός γραμμικού συστήματος, όπως είναι η (4.2.30). Έτσι, εάν ο πίνακας R_x παρουσιάζει μεγάλη διασπορά ιδιοτιμών, τότε μία μικρή μεταβολή ή ένα μικρό σφάλμα (π.χ. floating point αναπαράσταση) στο σταθερό διάνυσμα r_{xy} , θα έχει ως αποτέλεσμα μία μεγάλη μεταβολή στο $k_{optimal}$, και συνεπώς το πρόβλημα δεν είναι καλά ορισμένο, από αριθμητικής σκοπιάς. Αντιθέτως, εάν η διασπορά ιδιοτιμών είναι μικρή, τότε τυχόν αβεβαιότητες ή round-off σφάλματα ή θόρυβος στο r_{xy} δεν θα επηρεάσουν σημαντικά την τιμή του $k_{optimal}$, και συνεπώς το πρόβλημα είναι καλά ορισμένο από αριθμητικής σκοπιάς.

Από την παραπάνω συζήτηση είναι εμφανές ότι μία απευθείας επίλυση του γραμμικού συστήματος $Ax = b$, όπως είναι οι κανονικές εξισώσεις, εμπεριέχει δυσκολίες, οι οποίες είναι σε πολλές περιπτώσεις αξεπέραστες. Για παράδειγμα, όταν η επεξεργασία σήματος γίνεται on-line και η απόκριση του αλγορίθμου πρέπει να είναι πολύ γρήγορη και πολύ ακριβής, όπως είναι η περίπτωση στον ενεργό έλεγχο θορύβου, τότε η αντιστροφή του πίνακα καταλαμβάνει ένα τεράστιο ποσοστό χρόνου της CPU, συγκρινόμενη με άλλες αλγοριθμικές διεργασίες, οι οποίες πρέπει να τρέξουν (π.χ. φιλτράρισμα). Εάν ο επεξεργαστής

που χρησιμοποιούμε δεν είναι αρκετά γρήγορος ώστε να εκτελέσει έγκαιρα την αντιστροφή, τότε κάποια εναλλακτική τεχνική πρέπει να χρησιμοποιηθεί, προς προσδιορισμό του $\mathbf{k}_{optimal}$. Επιπρόσθετα, όταν η απόδοση ενός συστήματος σε συνδυασμό με την ευστάθεια του είναι μέγιστης σημασίας, όπως ξανά είναι η περίπτωση του ενεργού ελέγχου θορύβου, οι ill conditioned πίνακες αυτοσυσχέτισης είναι πολύ πιθανό να οδηγήσουν σε φτωχή απόδοση και αστάθεια.

Πράγματι, εναλλακτικές τεχνικές υπάρχουν, και συνίστανται στην προσπάθεια επίλυσης του γραμμικού συστήματος επαναληπτικά. Βασιζόμενοι στα παραπάνω επιχειρήματα, θα λέγαμε ότι οι επαναληπτικές μέθοδοι είναι πρακτικά επιβεβλημένες, παρά συνιστούν εναλλακτικές μεθόδους, κυρίως όσον αφορά την επίλυση στοχαστικών γραμμικών συστημάτων σε πραγματικό χρόνο.

4.2.5 Ανάπτυξη του Steepest-Descent αλγορίθμου

Σε αυτό το σημείο ας κάνουμε την εξής εσκεμμένη αλλαγή στο συμβολισμό:

$$\mathbf{k}^T \rightarrow \mathbf{w}$$

$$\mathbf{k}_{optimal}^T \rightarrow \mathbf{w}^o$$

$$\mathbf{x} \rightarrow \mathbf{u}$$

$$y \rightarrow d$$

Ο λόγος για την εισαγωγή αυτού του νέου συμβολισμού, είναι ότι σταδιακά εισερχόμαστε στα συμφραζόμενα των προσαρμοζόμενων φίλτρων. Σε αυτά τα συμφραζόμενα, η μεταβλητή της οποίας τον βέλτιστο γραμμικό προβλέπτη αναζητούμε ονομάζεται *επιθυμητό σήμα* και συμβολίζεται ως d . Ένας προβλέπτης του επιθυμητού σήματος συμβολίζεται ως $\mathbf{w}$ και ονομάζεται *διάνυσμα θάρους*. Τέλος, το σήμα, του οποίου τις παρατηρήσεις χρησιμοποιούμε προκειμένου να αποφασίσουμε για την πρόβλεψη του επιθυμητού σήματος, συμβολίζεται ως $\mathbf{u}$ και ονομάζεται *σήμα αναφοράς*.

Έτσι, το $\mathbf{w}$ είναι ένας γραμμικός προβλέπτης της τυχαίας μεταβλητής d από το τυχαίο διάνυσμα $\mathbf{u}$. Η πρόβλεψη $\hat{d}$ είναι

$$\hat{d} = \mathbf{w}^T \mathbf{u}, \quad (4.2.35)$$

και το σφάλμα πρόβλεψης είναι

$$e = d - \hat{d} = d - \mathbf{w}^T \mathbf{u}, \quad (4.2.36)$$

Το μέσο τετραγωνικό σφάλμα του προβλέπτη είναι (βλέπε εξίσωση (4.2.15))

$$Ee^2 = J(\mathbf{w}) = \sigma_d^2 - 2\mathbf{r}_{ud}^T \mathbf{w} + \mathbf{w}^T R_u \mathbf{w}, \quad (4.2.37)$$

Ο βέλτιστος προβλέπτης $\mathbf{w}^o$ ικανοποιεί τις κανονικές εξισώσεις, οι οποίες τώρα γράφονται ως εξής:

$$R_u \mathbf{w}^o = \mathbf{r}_{ud}, \quad (4.2.38)$$

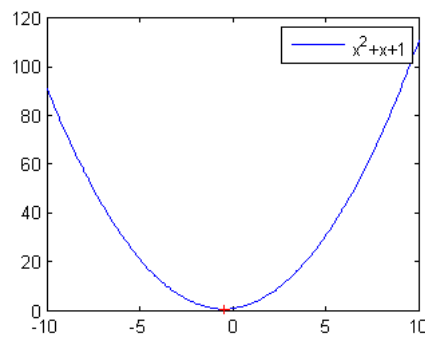
και το ελάχιστο μέσο τετραγωνικό σφάλμα του εν λόγω προβλήματος πρόβλεψης είναι

$$J_{min} = \sigma_d^2 - \mathbf{r}_{ud}^T \mathbf{w}^o, \quad (4.2.39)$$

Όπως ήδη αναλύθηκε, το $\mathbf{w}^o$ είναι ο βέλτιστος προβλέπτης, υπό την έννοια ότι ελαχιστοποιεί την τιμή του μέσου τετραγωνικού σφάλματος $J(\mathbf{w})$. Όπως αποκάλυψε η προηγούμενη ανάλυση, δεν υπάρχει άλλο $\mathbf{w}$, το οποίο ελαχιστοποιεί τη συνάρτηση κόστους. Πιο συγκεκριμένα, το $\mathbf{w}^o$ είναι το μοναδικό σημείο ολικού ακροτάτου της $J(\mathbf{w})$. Αυτό γίνεται σε βάθος κατανοητό, εάν παρατηρήσουμε ότι το μέσο τετραγωνικό σφάλμα της (4.2.37) είναι μία συνάρτηση δευτέρου βαθμού ως προς $\mathbf{w}$. Αυτό συμβαίνει επειδή η αλγεβρική της έκφραση περιλαμβάνει μόνο όρους πρώτης και δεύτερης τάξης των συντελεστών του $\mathbf{w}$. Για παράδειγμα, δεν υπάρχει όρος της μορφής w_i^3 ή $w_j w_i^2$ ή ακόμα $w_i w_j w_k$ στην (4.2.37). Ο όρος $2\mathbf{r}_{ud}^T \mathbf{w}$ συνεισφέρει όρους πρώτης τάξης ενώ ο όρος $\mathbf{w}^T R_u \mathbf{w}$ συνεισφέρει όρους δεύτερης τάξης. Συνεπώς το διάνυσμα $\nabla_{\mathbf{w}} J(\mathbf{w})$ περιέχει μόνο όρους πρώτης τάξης και η εξίσωσή του με το μηδέν δίνει μία μοναδική λύση, την $\mathbf{w}^o = R_u^{-1} \mathbf{r}_{ud}$. Αυτή η περίπτωση είναι ακριβώς παράλληλη με την απλή περίπτωση μιας συνάρτησης μίας μεταβλητής. Έστω για παράδειγμα η συνάρτηση $f(x) = x^2 + x + 1$. Πράγματι, με βάση το θεώρημα του Fermat, η $f(x)$ έχει ένα μοναδικό σημείο ολικού ακροτάτου, και συγκεκριμένα το $-\frac{1}{2}$, καθώς

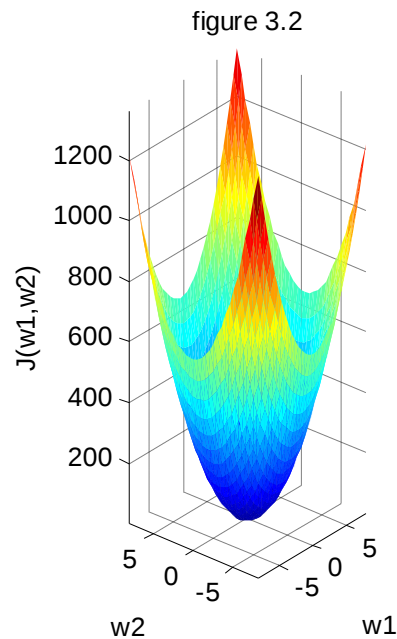
$$f'(x) = 2x + 1 = 0 \Rightarrow x = -\frac{1}{2} \text{ and } f''\left(-\frac{1}{2}\right) = 2 > 0, \quad (4.2.40)$$

Η γραφική παράσταση της $f(x)$ είναι



Εικόνα 4.2.1 γραφική παράσταση της $f(x) = x^2 + x + 1$ και ολικό ελάχιστο στο $-\frac{1}{2}$

Όπως είναι αναμενόμενο, η γραφική παράσταση της $J(\mathbf{w})$ είναι μία επέκταση στον L-διάστατο διανυσματικό χώρο της παραβολής της εικόνας 4.2.1. Αυτή η επέκταση μπορεί να θεωρηθεί ως μία υπερπαραβολική επιφάνεια. Η επιφάνεια αυτή ονομάζεται MSE επιφάνεια και δεν είναι δυνατόν να οπτικοποιηθεί για προβλέπτες με περισσότερους από δύο συντελεστές. Για έναν προβλέπτη της μορφής $\mathbf{w} = [w_1, w_2]^T$ η MSE επιφάνεια έχει την παρακάτω μορφή:



Εικόνα 4.2.2 γραφική παράσταση μίας quadratic συνάρτησης δύο μεταβλητών

Προφανώς ο βέλτιστος προβλέπτης αντιστοιχεί στις συντεταγμένες του w_1 - w_2 επιπέδου, όπου η επιφάνεια παρουσιάζει το ολικό της ελάχιστο, δηλαδή στον πυθμένα της επιφάνειας.

Τώρα, δεδομένης της $J(\mathbf{w})$ ((4.2.37)), και χωρίς καμία εκ των προτέρων πληροφορία σχετικά με τη θέση του ορίσματος που την ελαχιστοποιεί $\mathbf{w}^o$, θέλουμε να επινοήσουμε μία διαδικασία, η οποία ξεκινάει από μία αρχική εκτίμηση για το $\mathbf{w}^o$ και στη συνέχεια βελτιώνεται με έναν επαναληπτικό τρόπο, έως ότου τελικά να συγκλίνει στο $\mathbf{w}^o$. Η διαδικασία, την οποία αναζητούμε, πρέπει να έχει τη εξής μορφή:

$$(\text{αναβαθμισμένη εκτίμηση}) = (\text{προηγούμενη εκτίμηση}) + (\text{ένας όρος διόρθωσης})$$

ή πιο αναλυτικά,

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{p}, \quad i \geq 0, \quad (4.2.41)$$

όπου γράφουμε $\mathbf{w}_{i-1}$ ώστε να δηλώσουμε μία εκτίμηση για το $\mathbf{w}^o$ στην επανάληψη (i-1), και $\mathbf{w}_i$ ώστε να δηλώσουμε την αναμαθισμένη εκτίμηση στην επανάληψη i. Το διάνυσμα $\mathbf{p}$ είναι ένα διάνυσμα αναβάθμισης κατεύθυνσης, το οποίο πρέπει να επιλέξουμε επιπρόσθετα, μαζί με τη θετική, βαθμωτή ποσότητα μ , προκειμένου να επιτύχουμε εγγυημένα τη σύγκλιση των $\{\mathbf{w}_i\}$ στο $\mathbf{w}^o$. Η βαθμωτή ποσότητα μ ονομάζεται *μήκος βήματος* και καθορίζει πόσο μεγάλος ή πόσο μικρός θα είναι ο όρος διόρθωσης. Συνήθως η αρχική εκτίμηση $\mathbf{w}_{i-1}$ θεωρείται ότι είναι το μηδέν.

Το κριτήριο για την επιλογή των μ και $\mathbf{p}$ είναι να επιβληθεί, εάν είναι δυνατόν, η συνθήκη

$$J(\mathbf{w}_i) < J(\mathbf{w}_{i-1}), \quad (4.2.42)$$

Με αυτόν τον τρόπο, η τιμή της συνάρτησης κόστους θα μειώνεται με γνήσιο τρόπο μεταξύ διαδοχικών επαναλήψεων. Για να δείξουμε πώς αυτή η συνθήκη μπορεί να επιβληθεί, ξεκινούμε συσχετίζοντας το $J(\mathbf{w}_i)$ με το $J(\mathbf{w}_{i-1})$. Υπολογίζοντας το $J(\mathbf{w})$ στο $\mathbf{w}_i = \mathbf{w}_{i-1} + \mu\mathbf{p}$ (βλέπε (4.2.37)) και αναπτύσσοντας, έχουμε

$$\begin{aligned} J(\mathbf{w}_i) &= \\ &= \sigma_d^2 - 2\mathbf{r}_{ud}^T(\mathbf{w}_{i-1} + \mu\mathbf{p}) + (\mathbf{w}_{i-1} + \mu\mathbf{p})^T R_u (\mathbf{w}_{i-1} + \mu\mathbf{p}) = \\ &= \sigma_d^2 - 2\mathbf{r}_{ud}^T \mathbf{w}_{i-1} - 2\mathbf{r}_{ud}^T \mu\mathbf{p} + \mathbf{w}_{i-1}^T R_u \mathbf{w}_{i-1} + \mathbf{w}_{i-1}^T R_u \mu\mathbf{p} + \\ &+ \mu\mathbf{p}^T R_u \mathbf{w}_{i-1} + \mu\mathbf{p}^T R_u \mu\mathbf{p} = \{\sigma_d^2 - 2\mathbf{r}_{ud}^T \mathbf{w}_{i-1} + \mathbf{w}_{i-1}^T R_u \mathbf{w}_{i-1}\} \\ &- 2\mathbf{r}_{ud}^T \mu\mathbf{p} + \mathbf{w}_{i-1}^T R_u \mu\mathbf{p} + \mu\mathbf{p}^T R_u \mathbf{w}_{i-1} + \mu\mathbf{p}^T R_u \mu\mathbf{p} = \\ &= J(\mathbf{w}_{i-1}) + \mu(\mathbf{w}_{i-1}^T R_u - \mathbf{r}_{ud}^T)\mathbf{p} + \mu\mathbf{p}^T(R_u \mathbf{w}_{i-1} - \mathbf{r}_{ud}) + \mu^2 \mathbf{p}^T R_u \mathbf{p}, \quad (4.2.43) \end{aligned}$$

Μπορούμε να γράψουμε την (4.2.43) περισσότερο συνοπτικά, παρατηρώντας από την (4.2.17) ότι

$$\nabla_w J(\mathbf{w}) = -2\mathbf{r}_{ud} + 2R_u \mathbf{w}, \quad (4.2.44)$$

Αυτό σημαίνει ότι ο όρος $(\mathbf{w}_{i-1}^T R_u - \mathbf{r}_{ud}^T)$ που εμφανίζεται στην (4.2.43) είναι απλώς το μισό του ανάστροφου του διανύσματος κλίσης της $J(\mathbf{w})$ στο $\mathbf{w} = \mathbf{w}_{i-1}$, δηλαδή

$$(\mathbf{w}_{i-1}^T R_u - \mathbf{r}_{ud}^T) = \frac{1}{2} [\nabla_w J(\mathbf{w}_{i-1})]^T, \quad (4.2.45)$$

Αυτό το συμπέρασμα μας επιτρέπει να γράψουμε την (4.2.43) ως εξής:

$$\begin{aligned} J(\mathbf{w}_i) &= \\ &= J(\mathbf{w}_{i-1}) + \frac{1}{2} \mu [\nabla_w J(\mathbf{w}_{i-1})]^T \mathbf{p} + \frac{1}{2} \mu \mathbf{p}^T \nabla_w J(\mathbf{w}_{i-1}) + \mu^2 \mathbf{p}^T R_u \mathbf{p} \Rightarrow \\ &= J(\mathbf{w}_i) = J(\mathbf{w}_{i-1}) + \mu \mathbf{p}^T \nabla_w J(\mathbf{w}_{i-1}) + \mu^2 \mathbf{p}^T R_u \mathbf{p}, \quad (4.2.46) \end{aligned}$$

Τώρα ο όρος $\mu^2 \mathbf{p}^T R_u \mathbf{p}$ είναι πάντοτε θετικός, καθώς ο R_u αποδείξαμε ότι είναι θετικά ορισμένος ((4.2.26)). Συνεπώς μία απαραίτητη συνθήκη (όχι όμως ικανή) ώστε να ισχύει $J(\mathbf{w}_i) < J(\mathbf{w}_{i-1})$ είναι να απαιτήσουμε η αναβάθμιση $\mathbf{p}$ να ικανοποιεί τη συνθήκη

$$\mathbf{p}^T \nabla_w J(\mathbf{w}_{i-1}) < 0, \quad (4.2.47)$$

Η συνθήκη αυτή δεν είναι ικανή, εξαιτίας του ενδεχομένου να ισχύει η σχέση

$$|\mathbf{p}^T \nabla_w J(\mathbf{w}_{i-1})| < |\mu^2 \mathbf{p}^T R_u \mathbf{p}|, \quad (4.2.48)$$

Η συνθήκη (4.2.47) θα μας εγγυηθεί ότι ο δεύτερος όρος στη δεξιά πλευρά της (4.2.46) είναι αυστηρώς αρνητικός. Τώρα, η επιλογή του $\mathbf{p}$ με βάση την (4.2.47) εξαρτάται από το αν ο παράγοντας $\nabla_w J(\mathbf{w}_{i-1})$ είναι μηδέν ή όχι. Εάν το διάνυσμα κλίσης είναι μηδέν, τότε $R_u \mathbf{w}_{i-1} = \mathbf{r}_{ud}$ και συνεπώς το $\mathbf{w}_{i-1}$ ήδη συμπίπτει με την επιθυμητή λύση $\mathbf{w}^o$. Σε αυτή την περίπτωση, οι επαναλήψεις της (4.2.41) έχουν συγκλίνει στο $\mathbf{w}^o$ και το $\mathbf{p}$ θα έπρεπε να επιλεγεί ως $\mathbf{p} = \mathbf{0}$.

Όταν από την άλλη πλευρά, το διάνυσμα κλίσης υπολογισμένο στο $\mathbf{w}_{i-1}$ δεν είναι μηδέν, τότε υπάρχουν πολλές επιλογές διανυσμάτων $\mathbf{p}$ που ικανοποιούν την (4.2.47). Για παράδειγμα, κάθε $\mathbf{p}$ της μορφής

$$\mathbf{p} = -B\nabla_{\mathbf{w}}J(\mathbf{w}_{i-1}), \quad (4.2.49)$$

για κάθε συμμετρικό, θετικά ορισμένο πίνακα B , είναι κατάλληλο (αυτή η επιλογή είναι συνεπής με την επιλογή $\mathbf{p} = \mathbf{0}$ όταν $\nabla_{\mathbf{w}}J(\mathbf{w}_{i-1}) = \mathbf{0}$), καθώς

$$\begin{aligned} \mathbf{p}^T \nabla_{\mathbf{w}}J(\mathbf{w}_{i-1}) &= (-B\nabla_{\mathbf{w}}J(\mathbf{w}_{i-1}))^T \nabla_{\mathbf{w}}J(\mathbf{w}_{i-1}) = \\ &= -[\nabla_{\mathbf{w}}J(\mathbf{w}_{i-1})]^T B^T \nabla_{\mathbf{w}}J(\mathbf{w}_{i-1}) = -[\nabla_{\mathbf{w}}J(\mathbf{w}_{i-1})]^T B \nabla_{\mathbf{w}}J(\mathbf{w}_{i-1}) < 0 \end{aligned}$$

εν όψει του θετικά ορισμένου του πίνακα B . Η ειδική επιλογή $B = I$ είναι πολύ συνηθής, και αντιστοιχεί στην αναβάθμιση κατεύθυνσης

$$\mathbf{p} = -\nabla_{\mathbf{w}}J(\mathbf{w}_{i-1}) = 2\mathbf{r}_{ud} - 2R_u\mathbf{w}_{i-1}, \quad (4.2.50)$$

Αυτή η επιλογή για το $\mathbf{p}$ απλοποιεί την (4.2.41) στη μορφή

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu(2\mathbf{r}_{ud} - 2R_u\mathbf{w}_{i-1}) = \mathbf{w}_{i-1} + 2\mu(\mathbf{r}_{ud} - R_u\mathbf{w}_{i-1})$$

και εάν απορροφήσουμε τον παράγοντα 2 μέσα στο μήκος βήματος, τότε η εξίσωση αναβάθμισης γίνεται

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu(\mathbf{r}_{ud} - R_u\mathbf{w}_{i-1}), \quad i \geq 0, \quad (4.2.51)$$

Η εξίσωση αναβάθμισης (4.2.51) έχει μία πολύ χρήσιμη και διαισθητική ερμηνεία. Θυμίζουμε ότι το διάνυσμα κλίσης (4.2.16)) σε κάθε σημείο οποιασδήποτε συνάρτησης κόστους, δείχνει προς την κατεύθυνση της μέγιστης αύξησης της συνάρτησης κόστους. Αυτό συμβαίνει επειδή το διάνυσμα κλίσης περιέχει όλες τις κατευθυντικές παραγώγους της συνάρτησης κόστους ως προς κάθε είσοδο του διανύσματος βάρους. Τώρα η (4.2.50) είναι τέτοια ώστε σε κάθε επανάληψη επιλέγει την αναβάθμιση κατεύθυνσης να δείχνει προς την αντίθετη κατεύθυνση, προς την οποία δείχνει το διάνυσμα κλίσης, δηλαδή προς την κατεύθυνση της μέγιστης μείωσης της συνάρτησης κόστους. Για αυτόν τον λόγο η (4.2.51) ονομάζεται Steepest-Descent μέθοδος, δηλαδή η μέθοδος της πιο απότομης κατάβασης. Το όνομα αυτό υπονοεί ότι η ακολουθία των διανυσμάτων βάρους $\{\mathbf{w}_i\}$ αποκτάται κατά την κατάβαση ενός «μονοπατιού» ελαττούμενων τιμών της συνάρτησης κόστους. Είναι επιπλέον διαισθητικά χρήσιμο να θεωρήσουμε τον προβλέπτη $\mathbf{w}$ σαν μία μπάλα, η οποία τη χρονική στιγμή $i - 1$ είναι τοποθετημένη στο σημείο $[\mathbf{w}_{i-1}, J(\mathbf{w}_{i-1})]$ στην MSE επιφάνεια. Η MSE επιφάνεια μπορεί να θεωρηθεί ως η επιφάνεια της δυναμικής ενέργειας της μπάλας λόγω της βαρύτητας. Έτσι, εάν η δυναμική ενέργεια της μπάλας τη χρονική στιγμή $i - 1$ είναι $J(\mathbf{w}_{i-1})$, τότε σε συμφωνία με τη Νευτώνια μηχανική, μία δύναμη ίση με $-\nabla_{\mathbf{w}}J(\mathbf{w}_{i-1})$ θα ασκηθεί στην μπάλα, οδηγώντας τη την αμέσως επόμενη χρονική στιγμή (i), στη θέση

$$[\mathbf{w}_{i-1} - \mu\nabla_{\mathbf{w}}J(\mathbf{w}_{i-1}), J(\mathbf{w}_{i-1} - \mu\nabla_{\mathbf{w}}J(\mathbf{w}_{i-1}))]$$

. Το «μονοπάτι», το οποίο η μπάλα ακολουθεί, είναι τέτοιο ώστε η δυναμική ενέργεια της μπάλας συνεχώς μειώνεται κατά μήκος του. Η μπάλα θα κυλήσει προς τον πυθμένα της επιφάνειας σε σχήμα μπωλ της δυναμικής της ενέργειας (βλέπε εικόνα 4.2.2) και τελικά θα σταματήσει στο σημείο ισορροπίας της δυναμικής της ενέργειας, δηλαδή στο $\mathbf{w}^o$.

Παρόλα αυτά, η σύγκλιση της ακολουθίας $\{\mathbf{w}_i\}$ στο $\mathbf{w}^o$ δεν είναι εγγυημένη. Θυμίζουμε ότι η (4.2.47) είναι μόνο μία απαραίτητη συνθήκη σύγκλισης, όχι όμως και ικανή, διότι υπάρχει το ενδεχόμενο να ισχύει η (4.2.48) σε μερικές περιπτώσεις. Η κρίσιμη παράμετρος, η οποία θα εγγυηθεί και τη σύγκλιση και την ευστάθεια του steepest-descent αλγόριθμου, ή ισοδύναμα θα ελέγχει την (4.2.48) να μην ισχύει, είναι το μήκος βήματος μ .

4.2.6 Σύγκλιση του Steepest-Descent αλγόριθμου

Προκειμένου να διερευνήσουμε με λεπτομέρεια τις συνθήκες, τις οποίες το μ πρέπει να ικανοποιεί για την ευστάθεια και τη βέλτιστη σύγκλιση του steepest-descent αλγόριθμου, εισάγουμε το διάνυσμα σφάλματος βάρους

$$\tilde{\mathbf{w}}_i = \mathbf{w}^o - \mathbf{w}_i, \quad (4.2.52)$$

το οποίο μετράει τη διαφορά μεταξύ της εκτίμησης του βέλτιστου διανύσματος βάρους τη χρονική στιγμή i και του βέλτιστου διανύσματος βάρους $\mathbf{w}^o$, στο οποίο θέλουμε να συγκλίνει ο αλγόριθμος.

Αφαιρώντας και τις δύο πλευρές της (4.2.51) από το $\mathbf{w}^o$, λαμβάνουμε

$$\tilde{\mathbf{w}}_i = \tilde{\mathbf{w}}_{i-1} - \mu(\mathbf{r}_{ud} - R_u \mathbf{w}_{i-1}), \quad (4.2.53)$$

με αρχικό διάνυσμα σφάλματος βάρους το

$$\tilde{\mathbf{w}}_{-1} = \mathbf{w}^o - \mathbf{w}_{-1}, \quad (4.2.54)$$

Χρησιμοποιώντας το γεγονός ότι το $\mathbf{w}^o$ ικανοποιεί τις κανονικές εξισώσεις $R_u \mathbf{w}^o = \mathbf{r}_{ud}$, αντικαθιστούμε το $\mathbf{r}_{ud}$ με $R_u \mathbf{w}^o$ στην (4.2.53) και φτάνουμε στην εξίσωση επαναλήψεων για το διάνυσμα σφάλματος βάρους:

$$\tilde{\mathbf{w}}_i = (I - \mu R_u) \tilde{\mathbf{w}}_{i-1}, \quad i \geq 0, \quad (4.2.55)$$

Η (4.2.55) είναι ένα ομογενές σύστημα εξισώσεων διαφορών με πίνακα συστελεστών τον $(I - \mu R_u)$ και αρχική συνθήκη την (4.2.54). Συνεπώς, μία αναγκαία και ικανή συνθήκη ώστε το διάνυσμα $\tilde{\mathbf{w}}_i$ να τείνει στο μηδέν, ανεξάρτητα από την αρχική συνθήκη $\tilde{\mathbf{w}}_{-1}$, είναι να απαιτήσουμε όλες οι ιδιοτιμές του πίνακα $(I - \mu R_u)$ να είναι απολύτως μικρότερες της μονάδας. Δηλαδή ο $(I - \mu R_u)$ πρέπει να είναι ένας ευσταθής πίνακας. Για να το δούμε αυτό προχωρούμε ως εξής:

Μπορεί να αποδειχθεί ότι οποιοσδήποτε θετικά ορισμένος, συμμετρικός πίνακας έχει πάντοτε ένα πλήρες σύνολο ορθοκανονικών ιδιοδιανυσμάτων. Δηλαδή, εάν ο R_u είναι $L \times L$ συμμετρικός και θετικά ορισμένος, τότε θα υπάρχουν L ορθοκανονικά ιδιοδιανύσματα ξ_i , τα οποία ικανοποιούν τη σχέση

$$R_u \xi_i = \lambda_i \xi_i, \quad \xi_j^T \xi_i = 0 \quad \text{for } i \neq j, \quad (4.2.56)$$

όπου λ_i είναι η ιδιοτιμή του R_u , η οποία αντιστοιχεί στο ιδιοδιάνυσμα ξ_i .

Το σύνολο των L εξισώσεων (4.2.56) μπορεί να γραφεί συνοπτικά σε μορφή πινάκων ως εξής:

$$R_u U = U \Lambda \Rightarrow$$

$$R_u = U \Lambda U^{-1}, \quad (4.2.57)$$

όπου

$$U = [\xi_1, \xi_2, \dots, \xi_L], \quad (4.2.58)$$

$$U^T = U^{-1} \text{ (unitary)}, \quad (4.2.59)$$

και

$$\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_L) \quad (4.2.60)$$

Η εξίσωση (4.2.57) ονομάζεται *φασματική αποσύνθεση* του πίνακα R_u . Τώρα ορίζουμε το μετασχηματισμένο διάνυσμα σφάλματος βάρους ως

$$\boldsymbol{\varphi}_i \equiv U^T \tilde{\mathbf{w}}_i, \quad (4.2.61)$$

Εφόσον ο U είναι unitary και συνεπώς αντιστρέψιμος, τα $\boldsymbol{\varphi}_i$ και $\tilde{\mathbf{w}}_i$ προσδιορίζουν το ένα το άλλο κατά έναν μοναδικό τρόπο. Τα διανύσματα $\boldsymbol{\varphi}_i$ και $\tilde{\mathbf{w}}_i$ έχουν επίσης την ίδια Ευκλείδεια νόρμα, καθώς

$$\|\boldsymbol{\varphi}_i\|^2 = \boldsymbol{\varphi}_i^T \boldsymbol{\varphi}_i = (U^T \tilde{\mathbf{w}}_i)^T U^T \tilde{\mathbf{w}}_i = \tilde{\mathbf{w}}_i^T U U^T \tilde{\mathbf{w}}_i =$$

$$\tilde{\mathbf{w}}_i^T U U^{-1} \tilde{\mathbf{w}}_i = \tilde{\mathbf{w}}_i^T \tilde{\mathbf{w}}_i = \|\tilde{\mathbf{w}}_i\|^2, \quad (4.2.62)$$

Τα παραπάνω σημαίνουν ότι, εάν το $\boldsymbol{\varphi}_i$ τείνει στο μηδέν, τότε και το $\tilde{\mathbf{w}}_i$ τείνει στο μηδέν. Συνεπώς, μπορούμε ισοδύναμα να αναζητήσουμε μία συνθήκη για το μ ώστε το $\boldsymbol{\varphi}_i$ να τείνει στο μηδέν. Είναι περισσότερο βολικό να δουλεύουμε με το $\boldsymbol{\varphi}_i$, επειδή ικανοποιεί ένα σύστημα εξισώσεων διαφορών όμοιο με τις (4.2.55), με τη διαφορά ότι ο πίνακας συντελεστών του είναι διαγώνιος. Για να το δούμε αυτό, πολλαπλασιάζουμε την (4.2.55) από αριστερά με U^T , αντικαθιστούμε τον R_u με την φασματική του αποσύνθεση $U \Lambda U^{-1}$ και ξαναγράφουμε τον ταυτοτικό πίνακα I ως $U U^T$:

$$U^T \tilde{\mathbf{w}}_i = U^T (UU^T - \mu U \Lambda U^T) \tilde{\mathbf{w}}_{i-1}$$

$$\Leftrightarrow$$

$$\boldsymbol{\varphi}_i = (I - \mu \Lambda) \boldsymbol{\varphi}_{i-1}, \quad \boldsymbol{\varphi}_{-1} = U^T \tilde{\mathbf{w}}_{i-1}, \quad (4.2.63)$$

Ο πίνακας συντελεστών για αυτές τις εξισώσεις διαφορών είναι διαγώνιος και ίσος με $(I - \mu \Lambda)$. Συνεπάγεται ότι οι χρονικές εξελίξεις των μεμονωμένων εισόδων του $\boldsymbol{\varphi}_i$ είναι αποσυνζευγμένες. Συγκεκριμένα, εάν συμβολίσουμε αυτές τις μεμονωμένες εισόδους ως

$$\boldsymbol{\varphi}_i = [\varphi_1(i), \varphi_2(i), \dots, \varphi_L(i)]^T, \quad (4.2.64)$$

τότε η (4.2.63) δείχνει ότι η k είσοδος του $\boldsymbol{\varphi}_i$ ικανοποιεί την

$$\varphi_k(i) = (1 - \mu \lambda_k) \varphi_k(i-1) \Rightarrow$$

$$\varphi_k(i) = (1 - \mu \lambda_k)^{i+1} \varphi_k(-1), \quad i \geq 0, \quad (4.2.65)$$

Αναφερόμαστε στην ποσότητα $(1 - \mu \lambda_k)$ ως τον *ρυθμό*, ο οποίος σχετίζεται με το $\varphi_k(i)$. Τώρα, προκειμένου το $\varphi_k(i)$ να τείνει στο μηδέν ανεξάρτητα από το $\varphi_k(-1)$, ο ρυθμός $(1 - \mu \lambda_k)$ πρέπει να είναι απολύτως μικρότερος της μονάδας. Συνεπώς, για να τείνουν όλες οι είσοδοι του διανύσματος $\boldsymbol{\varphi}_i$ στο μηδέν, το μήκος βήματος μ πρέπει να ικανοποιεί τη σχέση

$$|1 - \mu \lambda_k| < 1, \quad \text{για όλα τα } k = 1, 2, \dots, L, \quad (4.2.66)$$

Εφόσον

$$(I - \mu R_u) = (UU^T - \mu U \Lambda U^T) = U(I - \mu \Lambda)U^T$$

είναι η φασματική αποσύνθεση του πίνακα $(I - \mu R_u)$, συμπεραίνουμε ότι οι ρυθμοί $(1 - \mu \lambda_k)$ είναι οι ιδιοτιμές του, και έτσι αποδείξαμε τον αρχικό μας ισχυρισμό, ότι όλες οι ιδιοτιμές αυτού του πίνακα πρέπει να απολύτως μικρότερες της μονάδας, προκειμένου η ακολουθία $\{\tilde{\mathbf{w}}_i\}$ να συγκλίνει στο μηδέν. Εύκολα βλέπουμε ότι η συνθήκη (4.2.66) είναι ισοδύναμη με το να επιλέξουμε το μ έτσι ώστε

$$0 < \mu < \frac{2}{\lambda_{max}}, \quad (4.2.67)$$

όπου το λ_{max} συμβολίζει την απολύτως μεγαλύτερη ιδιοτιμή του R_u .

Είμαστε σε θέση τώρα να διατυπώσουμε το εξής σημαντικό θεώρημα:

Θεώρημα 4.2.3 (steepest-descent αλγόριθμος) Θεωρούμε μία τυχαία μεταβλητή d μηδενικής μέσης τιμής και με διασπορά σ_d^2 και ένα διάνυσμα $\mathbf{u}$ τυχαίων μεταβλητών μηδενικής μέσης τιμής, με $R_u = E \mathbf{u} \mathbf{u}^T$ θετικά ορισμένο. Έστω λ_{max} η απολύτως μεγαλύτερη ιδιοτιμή του R_u . Τότε η λύση των κανονικών εξισώσεων $R_u \mathbf{w}^o = \mathbf{r}_{ud}$ δύναται να αποκτηθεί επαναληπτικά ως εξής: Ξεκινούμε με μία αρχική εκτίμηση $\mathbf{w}_{-1}$ (για παράδειγμα $\mathbf{w}_{-1} = 0$), επιλέγουμε μήκος βήματος μ : $0 < \mu < \frac{2}{\lambda_{max}}$ και εκτελούμε τις επαναλήψεις για $i \geq 0$:

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu (\mathbf{r}_{ud} - R_u \mathbf{w}_{i-1})$$

Τότε $\mathbf{w}_i \rightarrow \mathbf{w}^o$ καθώς $i \rightarrow \infty$.

Με το θεώρημα 4.2.3 επιτύχαμε το σκοπό μας, δηλαδή την ανάπτυξη ενός επαναληπτικού αλγορίθμου, ο οποίος επιλύει το πρόβλημα πρόβλεψης ή βελτιστοποίησης $\min_{\mathbf{w}} E|\mathbf{d} - \mathbf{w}^T \mathbf{u}|^2$ ή ισοδύναμα τις κανονικές εξισώσεις $R_u \mathbf{w}^o = \mathbf{r}_{ud}$. Οι ιδέες, οι οποίες χρησιμοποιήθηκαν για την εξαγωγή του steepest-descent αλγορίθμου, μπορούν επίσης να αφαρμοσθούν σε περισσότερο γενικά προβλήματα βελτιστοποίησης, στα οποία η συνάρτηση κόστους δεν είναι εν γένει δευτέρου βαθμού. Σε κάθε περίπτωση, η εξίσωση αναβάθμισης εξακολουθεί να είναι της μορφής

$$\mathbf{w}_i = \mathbf{w}_{i-1} - \mu \nabla_{\mathbf{w}} J(\mathbf{w}_{i-1}), \quad (4.2.68)$$

Διαφορετικές συναρτήσεις κόστους οδηγούν εν γένει σε διαφορετικές MSE επιφάνειες, οι οποίες στη γενική περίπτωση έχουν πολλαπλά τοπικά ελάχιστα, ένα από τα οποία είναι το ολικό ελάχιστο. Τώρα, στην περίπτωση των πολλαπλών ελαχίστων δεν υπάρχει καμία εγγύηση ότι οι επαναλήψεις της (4.2.68) δε θα συγκλίνουν σε κάποιο τοπικό ελάχιστο, χάνοντας έτσι το ολικό ελάχιστο και συνεπώς μην ελαχιστοποιώντας τη συνάρτηση κόστους. Η συνολική συμπεριφορά σύγκλισης θα εξαρτάται από την τιμή του μήκους βήματος και από την θέση της αρχικής συνθήκης $J(\mathbf{w}_{-1})$ πάνω στην MSE επιφάνεια. Στην προηγούμενη ανάλυσή μας δεν συναντήσαμε καμία τέτοια δυσκολία ή επιπλοκή χρησιμοποιώντας τη συνάρτηση κόστους των μέσων ελαχίστων τετραγώνων της (4.2.6), επειδή όπως δείξαμε η συνάρτηση αυτή δεν έχει τοπικά ελάχιστα, παρά ένα και μοναδικό ολικό ελάχιστο. Αυτός είναι και ο λόγος για τη μεγάλη σπουδαιότητα του steepest-descent αλγορίθμου και της τεράστιας σημασίας του στην ανάπτυξη των προσαρμοζόμενων φίλτρων.

4.2.7 Ανάπτυξη του προσαρμοζόμενου αλγορίθμου LMS

Στην προηγούμενη ενότητα αναπτύξαμε μία επαναληπτική διαδικασία για τον προσδιορισμό του προβλέπτη μιας τυχαίας μεταβλητής d από τις παρατηρήσεις $\mathbf{u}$ μίας άλλης τυχαίας μεταβλητής u . Αυτή η διαδικασία είναι η διάσημη steepest-descent μέθοδος και για τη συνάρτηση κόστους (4.2.6) οι επαναλήψεις έχουν τη μορφή

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu(\mathbf{r}_{ud} - R_u \mathbf{w}_{i-1}), \quad i \geq 0, \quad (4.2.69)$$

Το θεώρημα 4.2.3 εγγυάται ότι οι επαναλήψεις (4.2.69) θα συγκλίνουν στο βέλτιστο γραμμικό προβλέπτη $\mathbf{w}^o$ έπειτα από ένα κατάλληλο αριθμό τους. Η παραπάνω φόρμουλα της steepest-descent μεθόδου βασίζεται ξεκάθαρα στην ακριβή γνώση των ποσοτήτων $\{\mathbf{r}_{ud}, R_u\}$. Το γεγονός αυτό συνιστά έναν πολύ σημαντικό περιορισμό όσον αφορά τις πρακτικές εφαρμογές, και αποτελεί μία θαυμάσια αιτία για την ανάπτυξη προσαρμοστικών αλγορίθμων, για δύο λόγους:

1. **Έλλειψη στατιστικής πληροφορίας.** Καταρχάς, οι ποσότητες $\{\mathbf{r}_{ud}, R_u\}$ είναι σπάνια διαθέσιμες στην πράξη. Ως αποτέλεσμα το πραγματικό διάνυσμα κλίσης
$$\nabla_{\mathbf{w}} J(\mathbf{w}_{i-1}) = -2\mathbf{r}_{ud} + 2R_u \mathbf{w}_{i-1}, \quad (4.2.70)$$
 είναι αδύνατον να υπολογιστεί επακριβώς και η πραγματική steepest-descent υλοποίηση καθίσταται αδύνατη. Για παράδειγμα, στον ενεργό έλεγχο θορύβου, η εκ των προτέρων γνώση του θορύβου, ο οποίος πρόκειται να εξασθενισθεί είναι είτε παντελώς απύσχα είτε υπάρχουν πολύ περιορισμένες πληροφορίες για τον καθορισμό των $\{\mathbf{r}_{ud}, R_u\}$.
2. **Χρονική μεταβολή στην στατιστική πληροφορία.** Δεύτερον, και ίσως περισσότερο σημαντικό, οι ποσότητες $\{\mathbf{r}_{ud}, R_u\}$ εν γένει είναι συναρτήσεις του χρόνου, δηλαδή μεταβάλλονται με την πάροδο του χρόνου. Συνεπώς και το βέλτιστο διάνυσμα βάρους $\mathbf{w}^o$ θα μεταβάλλεται με το χρόνο. Για παράδειγμα, σε εφαρμογές ενεργού ελέγχου θορύβου, ο θόρυβος προς εξασθένιση ενδέχεται να είναι σε δραματικό ποσοστό μη στάσιμος, όσον αφορά το φάσμα του και την ισχύ του.

Οι προσαρμοστικοί αλγόριθμοι λύνουν αυτό το πρόβλημα αντικαθιστώντας τις ποσότητες $\{\mathbf{r}_{ud}, R_u\}$ με κατάλληλες προσεγγίσεις τους. Διαφορετικές προσεγγίσεις οδηγούν σε διαφορετικούς αλγορίθμους με ποικίλους βαθμούς περιπλοκότητας και ιδιοτήτων απόδοσης. Αυτοί οι προσαρμοστικοί αλγόριθμοι, οι οποίοι οδηγούν απευθείας στην υλοποίηση προσαρμοζόμενων φίλτρων, ονομάζονται και αλγόριθμοι στοχαστικής κλίσης, καθώς χρησιμοποιώντας προσεγγίσεις του διανύσματος κλίσης, οι αναβαθμισμένες κατευθύνσεις υπόκεινται σε τυχαίες διακυμάνσεις, οι οποίες αναφέρονται ως θόρυβος κλίσης. Οι προσαρμοστικοί αλγόριθμοι, εξυπηρετούν τουλάχιστον δύο σκοπούς. Πρώτον αποφεύγουν την ανάγκη γνώσης των ακριβών στατιστικών ιδιοτήτων του σήματος (για παράδειγμα μεταβλητότητας και

συμμεταβλητότητας), οι οποίες είναι απαραίτητες για μία επιτυχή steepest-descent υλοποίηση, αλλά παρόλα αυτά σπανίως διαθέσιμες στην πράξη. Οι προσαρμοστικοί αλγόριθμοι επιτυγχάνουν αυτό το χαρακτηριστικό μέσω ενός **μηχανισμού εκμάθησης**, ο οποίος τους επιτρέπει να εκτιμούν τις στατιστικές ιδιότητες του σήματος. Δεύτερον, οι προσαρμοστικοί αλγόριθμοι διαθέτουν έναν **μηχανισμό αναβάθμισης**, ο οποίος τους επιτρέπει να ακολουθούν διάφορες μεταβολές στη στατιστική του σήματος. Αυτές οι δύο συνδυασμένες ικανότητες, της εκμάθησης και της αναβάθμισης, είναι οι κύριοι λόγοι για την ευρεία χρήση των προσαρμοστικών αλγορίθμων στην τεχνολογία (συγκεκριμένα αυτή η χρήση είναι τόσο καθολική που θα ήταν δύσκολο να φανταστούμε πώς θα ήταν η σημερινή τεχνολογία χωρίς την ύπαρξη των προσαρμοστικών αλγορίθμων). Εξαιτίας αυτών των δύο χαρακτηριστικών υπάρχει μία τάση να αναφερόμαστε στα προσαρμοζόμενα φίλτρα ως «έξυπνα συστήματα», έξυπνα υπό την έννοια ότι μπορούν να μάθουν τις στατιστικές ιδιότητες των υπεισερχομένων, στην εκάστοτε εφαρμογή, σημάτων και μπορούν να προσαρμόσουν την συμπεριφορά τους έπειτα από μεταβολές του περιβάλλοντός τους, έτσι ώστε να διατηρούν την απόδοσή τους σε υψηλά όρια.

Μία από τις πιο απλές προσεγγίσεις των $\{\mathbf{r}_{ud}, R_u\}$ είναι οι στιγμιαίες τιμές τους, δηλαδή

$$\hat{\mathbf{r}}_{ud} = \mathbf{u}(i)d(i) \text{ και } \hat{R}_u = \mathbf{u}(i)\mathbf{u}(i)^T, \quad (4.2.71)$$

Αυτή η επιλογή ισοδυναμεί με το να μην λάβουμε υπόψιν μας τον τελεστή της αναμενόμενης τιμής, όπως αυτός συμμετέχει στους πραγματικούς ορισμούς

$$\mathbf{r}_{ud} = E\mathbf{u}d \quad \text{και} \quad R_u = E\mathbf{u}\mathbf{u}^T, \quad (4.2.72)$$

Η (4.2.71) είναι η θρυλική προσέγγιση που έκανε ο Widrow το 1960, θέτοντας τις βάσεις για την ανάπτυξη των προσαρμοστικών αλγορίθμων.

Με αυτόν τον τρόπο το διάνυσμα κλίσης προσεγγίζεται τη χρονική στιγμή i από την ποσότητα $J(\mathbf{w}_{i-1})$, όπως φαίνεται και από την επόμενη εξίσωση:

$$\hat{\nabla}_{\mathbf{w}} J(\mathbf{w}_{i-1}) = -2\mathbf{u}(i)d(i) + 2\mathbf{u}(i)\mathbf{u}(i)^T \mathbf{w}_{i-1}, \quad (4.2.73)$$

Η φόρμουλα των επαναλήψεων από τις (4.2.73), (4.2.71) και (4.2.69), γίνεται

$$\begin{aligned} \mathbf{w}_i &= \mathbf{w}_{i-1} + \mu \mathbf{u}(i)[d(i) - \mathbf{u}(i)^T \mathbf{w}_{i-1}] \\ i &\geq 0, \quad \mathbf{w}_{-1} = \text{αρχική εκτίμηση}, \end{aligned} \quad (4.2.74)$$

Παρατηρούμε ότι η ποσότητα $d(i) - \mathbf{u}(i)^T \mathbf{w}_{i-1}$ είναι το σφάλμα πρόβλεψης στην επανάληψη $(i - 1)$. Φυσικά τα $\mathbf{w}_i$ των (4.2.69) και (4.2.74) είναι αρκετά διαφορετικά. Ο αλγόριθμος στοχαστικής προσέγγισης κλίσης (4.2.74) είναι ο πιο δημοφιλής προσαρμοστικός αλγόριθμος, ακόμα και σήμερα, σε επίπεδο εφαρμογών, εξαιτίας της εντυπωσιακής απλότητάς του. Είναι δε, ευρέως γνωστός ως LMS αλγόριθμος ή μερικές φορές ως Widrow-Hoff αλγόριθμος, προς τιμήν των δημιουργών του.

Θεώρημα 4.2.4 (LMS αλγόριθμος) Θεωρούμε μία τυχαία μεταβλητή d μηδενικής μέσης τιμής, με πραγματοποιήσεις $\{d(0), d(1), \dots\}$, και ένα τυχαίο διάνυσμα $\mathbf{u}$ μηδενικής μέσης τιμής με πραγματοποιήσεις $\{\mathbf{u}(0), \mathbf{u}(1), \dots\}$. Το βέλτιστο διάνυσμα βάρους, το οποίο αποτελεί λύση των κανονικών εξισώσεων $R_u \mathbf{w}^o = \mathbf{r}_{ud}$, μπορεί να **προσεγγισθεί** επαναληπτικά μέσω των επαναλήψεων

$$\begin{aligned} \mathbf{w}_i &= \mathbf{w}_{i-1} + \mu \mathbf{u}(i)[d(i) - \mathbf{u}(i)^T \mathbf{w}_{i-1}] \\ i &\geq 0, \quad \mathbf{w}_{-1} = \text{αρχική εκτίμηση} \end{aligned}$$

όπου το μ είναι ένα θετικό μήκος βήματος, συνήθως μικρό.

Ως απόδειξη, μπορούμε να πούμε, ότι εάν θεωρήσουμε την αναμενόμενη τιμή κάθε μέλους της LMS επανάληψης (4.2.74), τότε καταλήγουμε στην (4.2.69), η οποία είναι η steepest-descent μέθοδος. Όμως τα $\mathbf{w}_i$ της (4.2.69) δείχθηκε προηγουμένως ότι συγκλίνουν στο βέλτιστο διάνυσμα βάρους $\mathbf{w}^o$. Αυτό σημαίνει ότι η αναμενόμενη τιμή των $\mathbf{w}_i$ των LMS επαναλήψεων συγκλίνει στο $\mathbf{w}^o$, δηλαδή

$$E\mathbf{w}_{i,LMS} \xrightarrow{i \rightarrow \infty} \mathbf{w}^o, \quad (4.2.75)$$

Η συμπεριφορά του LMS αλγορίθμου έχει αναλυθεί σε βάθος και αποτελέσματα παρόμοια με την (4.2.75) υπάρχουν και για μεγαλύτερες στατιστικές ροπές του $\mathbf{w}_{i,LMS}$.

Ένας άλλος λόγος, για τον οποίο ο LMS αλγόριθμος είναι τόσο ελκυστικός και δημοφιλής, είναι η μικρή υπολογιστική του πολυπλοκότητα. Μπορεί εύκολα ναδειχθεί ότι για κάθε επανάληψη απαιτούνται $(2L + 1)$ πολλαπλασιασμοί και $(2L)$ προσθέσεις, όπου L είναι ο αριθμός των συντελεστών του προσαρμοζόμενου φίλτρου (βλέπε (4.2.2)).

Είναι προφανές ότι περισσότερο πολύπλοκες προσεγγίσεις των $\{\mathbf{r}_{ud}, R_u\}$ μπορούν να γίνουν, από ότι είναι οι (4.2.71). Κάθε προσέγγιση οδηγεί σε διαφορετική φόρμουλα επανάληψεων, η οποία έχει τις δικές της χαρακτηριστικές ιδιότητες. Γενικά, αυτές οι περισσότερες σύνθετες προσεγγίσεις οδηγούν σε γρηγορότερη σύγκλιση και φυσικά επιλύουν περισσότερο σύνθετα προβλήματα πρόβλεψης από ότι ο LMS αλγόριθμος. Δυστυχώς αυτό συμβαίνει σε βάρος της υπολογιστικής τους πολυπλοκότητας, η οποία πολλές φορές είναι αρκετές τάξεις μεγέθους μεγαλύτερη από αυτήν του LMS. Παραδείγματα τέτοιων αλγορίθμων είναι ο Affine Projection αλγόριθμος και ο RLS αλγόριθμος. Στην παρούσα εργασία επικεντρωνόμαστε μόνο στον LMS αλγόριθμο, εφόσον αυτός είναι το θεμελιώδες εργαλείο ενός ελεγκτή ενεργού θορύβου.

4.2.8 Η ιδιότητα της ελάχιστης διαταραχής του LMS αλγορίθμου

Ο LMS αλγόριθμος αναπτύχθηκε στο προηγούμενο εδάφιο ως μία προσεγγιστική επαναληπτική λύση στο γραμμικό πρόβλημα πρόβλεψης ελαχίστων μέσων τετραγώνων

$$\min_{\mathbf{w}} E|d - \mathbf{w}^T \mathbf{u}|^2$$

υπό την έννοια ότι αποκτήθηκε αντικαθιστώντας το πραγματικό διάνυσμα κλίσης (4.2.70) στη φόρμουλα επανάληψης του steepest-descent αλγορίθμου (4.2.51), με μία στιγμιαία προσέγγισή του, την (4.2.73). Προκύπτει ότι ο LMS αλγόριθμος θα μπορούσε να έχει αποκτηθεί με ένα διαφορετικό τρόπο, ως η ακριβής λύση (όχι προσεγγιστική) ενός καλά ορισμένου προβλήματος πρόβλεψης, το οποίο όμως χαρακτηρίζεται από μία διαφορετική συνάρτηση κόστους.

Από μία άλλη οπτική γωνία, ο LMS αλγόριθμος μπορεί να θεωρηθεί ως η ακριβής λύση ενός **τοπικού** προβλήματος βελτιστοποίησης. Αυτή η ιδιότητα ονομάζεται *ιδιότητα ελάχιστης διαταραχής του LMS αλγορίθμου* και είναι εξαιρετικά χρήσιμη στη διαισθητική κατανόηση της λειτουργίας του LMS αλγορίθμου. Αυτή η ιδιότητα θα μας βοηθήσει αργότερα στο να κατανοήσουμε τις ιδιότητες σύγκλισης των LMS προσαρμοζόμενων φίλτρων και πώς μπορούμε να επιλέξουμε το μήκος βήματος κατάλληλα, ώστε να επιτυγχάνουμε διαφορετικές λειτουργίες κάθε φορά. Για αυτό το λόγο θα αναπτύξουμε πάλι τον LMS αλγόριθμο από το μηδέν, αλλά από μία τελείως διαφορετική σκοπιά.

Υποθέτουμε ότι έχουμε διαθέσιμο κάποιο διάνυσμα βάρους τη χρονική στιγμή $(i - 1)$, έστω $\mathbf{w}_{i-1}$, και έστω $\{d(i), \mathbf{u}_i\}$ τα νέα δεδομένα από μετρήσεις. Έστω $\mathbf{w}_i$ μία νέα εκτίμηση για το βέλτιστο διάνυσμα βάρους, την οποία θέλουμε να προσδιορίσουμε από τα διαθέσιμα δεδομένα μας $\{d(i), \mathbf{u}_i, \mathbf{w}_{i-1}\}$.

Έστω μ ένας δεδομένος θετικός αριθμός. Ορίζουμε δύο σφάλματα πρόβλεψης: το *εκ των προτέρων σφάλμα πρόβλεψης*

$$e(i) \equiv d(i) - \mathbf{u}_i^T \mathbf{w}_{i-1}, \quad (4.2.76)$$

και το *εκ των υστέρων σφάλμα πρόβλεψης*

$$r(i) \equiv d(i) - \mathbf{u}_i^T \mathbf{w}_i, \quad (4.2.77)$$

Το πρώτο μετράει το σφάλμα στην πρόβλεψη του $d(i)$ ως την ποσότητα $\mathbf{u}_i^T \mathbf{w}_{i-1}$, δηλαδή χρησιμοποιώντας το διαθέσιμο διάνυσμα βάρους πριν την αναβάθμιση, ενώ το δεύτερο μετράει το σφάλμα στην πρόβλεψη του $d(i)$ ως την ποσότητα $\mathbf{u}_i^T \mathbf{w}_i$, δηλαδή χρησιμοποιώντας το αναβαθμισμένο διάνυσμα βάρους. Αναζητούμε ένα $\mathbf{w}_i$, το οποίο ικανοποιεί το εξής υπό συνθήκη πρόβλημα βελτιστοποίησης:

$$\min_{\mathbf{w}_i} \|\mathbf{w}_i - \mathbf{w}_{i-1}\|^2 \text{ υπό τη συνθήκη } r(i) = (1 - \mu \|\mathbf{u}_i\|^2) e(i), \quad (4.2.78)$$

Με άλλα λόγια, αναζητούμε ένα $\mathbf{w}_i$, το οποίο είναι πλησιέστερα στο $\mathbf{w}_{i-1}$, υπό την έννοια της Ευκλείδιας νόρμας και ταυτόχρονα πρέπει να ικανοποιεί μία συνθήκη μεταξύ των $r(i)$ και $e(i)$. Η συνθήκη ή ο περιορισμός αυτός έχει νόημα, όταν το μήκος βήματος μ είναι αρκετά μικρό ώστε να ικανοποιεί την εξής σχέση: $|1 - \mu\|\mathbf{u}_i\|^2| < 1$, δηλαδή όταν

$$0 < \mu\|\mathbf{u}_i\|^2 < 2 \quad \text{για όλα τα } i, \quad (4.2.79)$$

Έτσι, όταν η (4.2.79) ισχύει, η νόρμα του εκ των υστέρων σφάλματος $r(i)$ θα είναι πάντοτε ίση ή μικρότερη της νόρμας του εκ των πρωτέρων σφάλματος $e(i)$, δηλαδή $|r(i)| < |e(i)|$, ή ισοδύναμα η ποσότητα $\mathbf{u}_i^T \mathbf{w}_i$ θα είναι μία καλύτερη πρόβλεψη του $d(i)$ από την ποσότητα $\mathbf{u}_i^T \mathbf{w}_{i-1}$ (εκτός της περίπτωσης $e(i) = 0$, στην οποία $r(i) = 0$ επίσης). Θα μπορούσε φυσικά κάποιος να επιβάλλει διαφορετικά είδη περιορισμών στο $r(i)$ από την (4.2.79), απλώς τότε δε θα οδηγούμασταν στον LMS αλγόριθμο, αλλά σε άλλους αλγορίθμους.

Η λύση της (4.2.78) μπορεί να αποκτηθεί από βασικές μαθηματικές αρχές, όπως στη συνέχεια. Έστω

$$\delta \mathbf{w} = \mathbf{w}_i - \mathbf{w}_{i-1}$$

Τότε

$$\begin{aligned} \mathbf{u}_i^T \delta \mathbf{w} &= \\ \mathbf{u}_i^T \mathbf{w}_i - \mathbf{u}_i^T \mathbf{w}_{i-1} &= \\ [\mathbf{u}_i^T \mathbf{w}_i - d(i)] + [d(i) - \mathbf{u}_i^T \mathbf{w}_{i-1}] &= \\ -r(i) + e(i) &= \\ \mu\|\mathbf{u}_i\|^2 e(i) &\Rightarrow \\ \mathbf{u}_i^T \delta \mathbf{w} &= \mu\|\mathbf{u}_i\|^2 e(i), \quad (4.2.80) \end{aligned}$$

Όπου στην τελευταία ισότητα χρησιμοποιήσαμε τον περιορισμό (4.2.78). Αυτός ο υπολογισμός δείχνει ότι το πρόβλημα βελτιστοποίησης (4.2.78) είναι ισοδύναμο με το πρόβλημα επιλογής ενός διανύσματος $\delta \mathbf{w}$ ελάχιστης Ευκλείδιας νόρμας, το οποίο ικανοποιεί την εξίσωση $\mathbf{u}_i^T \delta \mathbf{w} = \mu\|\mathbf{u}_i\|^2 e(i)$, δηλαδή

$$\min_{\delta \mathbf{w}} \|\delta \mathbf{w}\|^2 \quad \text{υπό τον περιορισμό } \mathbf{u}_i^T \delta \mathbf{w} = \mu\|\mathbf{u}_i\|^2 e(i), \quad (4.2.81)$$

Ο περιορισμός $\mathbf{u}_i^T \delta \mathbf{w} = \mu\|\mathbf{u}_i\|^2 e(i)$ είναι μία βαθμωτή εξίσωση με L αγνώστους, τις εισόδους του $\delta \mathbf{w}$. Αυτό σημαίνει, ότι ο περιορισμός αυτός συνιστά ένα υπο-ορισμένο γραμμικό σύστημα εξισώσεων ως προς $\delta \mathbf{w}$, και συνεπώς επιδέχεται άπειρες λύσεις. Ανάμεσα σε όλες τις πιθανές λύσεις $\delta \mathbf{w}$ θέλουμε να προσδιορίσουμε αυτήν, η οποία έχει τη μικρότερη Ευκλείδια νόρμα. Για να το επιτύχουμε αυτό, ξεχωρίζουμε δύο περιπτώσεις:

1. $\|\mathbf{u}_i\|^2 = 0$. Σε αυτήν την περίπτωση $\mathbf{u}_i = \mathbf{0}$ και ο περιορισμός (4.2.81) ικανοποιείται από οποιοδήποτε $\delta \mathbf{w}$. Το $\delta \mathbf{w}$ με την μικρότερη Ευκλείδια νόρμα είναι το $\delta \mathbf{w} = \mathbf{0}$, και συνεπώς $\mathbf{w}_i = \mathbf{w}_{i-1}$.
2. $\|\mathbf{u}_i\|^2 \neq 0$. Σε αυτήν την περίπτωση, είναι εύκολο να βρούμε τουλάχιστον μία λύση της εξίσωσης $\mathbf{u}_i^T \delta \mathbf{w} = \mu\|\mathbf{u}_i\|^2 e(i)$. Πράγματι,

$$\delta \mathbf{w}^o = \mu \mathbf{u}_i e(i), \quad (4.2.82)$$

είναι μία λύση, όπως επιβεβαιώνεται με απλή παρατήρηση.

Όμως, υπάρχουν άπειρες άλλες λύσεις της εξίσωσης (4.2.80). Για να το δούμε αυτό, έστω $\delta \mathbf{w}^o + \mathbf{z}$ κάθε άλλη πιθανή λύση, έτσι ώστε

$$\mathbf{u}_i^T (\delta \mathbf{w} + \mathbf{z}) = \mu\|\mathbf{u}_i\|^2 e(i)$$

Εφόσον όμως το $\delta \mathbf{w}^o$ είναι μία λύση, συμπεραίνουμε ότι το $\mathbf{z}$ πρέπει να ικανοποιεί τη σχέση $\mathbf{u}_i^T \mathbf{z} = 0$. Αυτό όμως σημαίνει ότι όλες οι λύσεις της (4.2.80) μπορούν να αποκτηθούν από το $\delta \mathbf{w}^o$ με πρόσθεση σε αυτό οποιουδήποτε διανύσματος $\mathbf{z}$, το οποίο είναι ορθογώνιο με το $\mathbf{u}_i$. Υπάρχουν άπειρα τέτοια διανύσματα: η εξίσωση $\mathbf{u}_i^T \mathbf{z} = 0$ ορίζει ένα υπερεπίπεδο στον L -διάστατο διανυσματικό χώρο, το οποίο είναι ο γεωμετρικός τόπος όλων των διανυσμάτων που είναι ορθογώνια με το $\mathbf{u}_i$. Ακόμα,

ανάμεσα σε όλες τις λύσεις $\{\delta \mathbf{w}^o + \mathbf{z}\}$, μπορεί να επιβεβαιωθεί ότι το $\delta \mathbf{w}^o$ της (4.2.82) έχει την ελάχιστη Ευκλείδια νόρμα, εφόσον:

$$\begin{aligned} \|\delta \mathbf{w}^o + \mathbf{z}\|^2 &= \\ \|\delta \mathbf{w}^o\|^2 + \|\mathbf{z}\|^2 + 2\mathbf{z}^T \delta \mathbf{w}^o &= \\ \|\delta \mathbf{w}^o\|^2 + \|\mathbf{z}\|^2 + 2\mu(\mathbf{z}^T \mathbf{u}_i)e(i) &= \\ \|\delta \mathbf{w}^o\|^2 + \|\mathbf{z}\|^2 + 0 &\Rightarrow \\ \|\delta \mathbf{w}^o + \mathbf{z}\|^2 &= \|\delta \mathbf{w}^o\|^2 + \|\mathbf{z}\|^2 > \|\delta \mathbf{w}^o\|^2, \quad (4.2.83) \end{aligned}$$

Τέλος, χρησιμοποιώντας τις σχέσεις $\delta \mathbf{w}^o = \mathbf{w}_i - \mathbf{w}_{i-1}$ και $\delta \mathbf{w}^o = \mu \mathbf{u}_i e(i)$, φθάνουμε στην εξής έκφραση για το $\mathbf{w}_i$:

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{u}(i)[d(i) - \mathbf{u}(i)^T \mathbf{w}_{i-1}]$$

η οποία ταυτίζεται με τις LMS επαναλήψεις της (4.2.74). Δείξαμε λοιπόν ότι η λύση στο πρόβλημα τοπικής βελτιστοποίησης (4.2.78) είναι ο LMS αλγόριθμος.

Με βάση όλη την παραπάνω ανάλυση, ο LMS αλγόριθμος παράγει το διάνυσμα βάρους $\mathbf{w}_i$ στην κατεύθυνση του ελαχίστου εκ των υστέρων σφάλματος, ενώ ταυτόχρονα παραμένει όσο πιο κοντά γίνεται στην προηγούμενη αναβάθμιση $\mathbf{w}_{i-1}$. Αυτή η ιδιότητα του LMS, υπονοεί ότι το $\mathbf{w}_i$ διαθέτει ένα είδος μνήμης, υπό την έννοια ότι διατηρεί τη μέγιστη δυνατή γνώση, η οποία αποκτήθηκε από τους «προκατόχους του» $\{\mathbf{w}_{i-1}, \mathbf{w}_{i-2}, \dots\}$. Αυτή η διατήρηση συμβαίνει επειδή σε κάθε αναβάθμιση το ελάχιστο δυνατό βήμα πραγματοποιείται. Προκειμένου να αναλυθεί σε περισσότερο βάθος αυτή η ιδιότητα, υποθέτουμε ότι τα δεδομένα $\{d, \mathbf{u}\}$, τα οποία έρχονται ως εισόδοι στον LMS είναι στάσιμα. Έπειτα από έναν αριθμό επαναλήψεων, το $\mathbf{w}$ έχει συγκλίνει σε μία περιοχή, και παραμένει εκεί εκτελώντας μικρές ταλαντώσεις, ενώ η μέση τιμή του είναι σταθερή και ίση με το $\mathbf{w}^o$, όπως δείχθηκε στην (4.2.75). Τώρα, εάν για κάποιο λόγο οι στατιστικές ιδιότητες των δεδομένων εισόδου αλλάξουν στιγμιαία στη χρονική στιγμή n , τότε η θέση του διανύσματος βάρους στον L -διάστατο διανυσματικό χώρο θα διαταραχθεί στον ελάχιστο δυνατό βαθμό εξαιτίας της αναβάθμισης $\mathbf{w}(n)$. Δηλαδή το $\mathbf{w}(n)$ θα περιέχει ένα μικρό ποσοστό αυτής της στατιστικής διαταραχής. Εάν αυτή η ανωμαλία ή διαταραχή στις στατιστικές ιδιότητες των δεδομένων εισόδου εκλείψει, τότε οι πληροφορίες που έως τη χρονική στιγμή n είχαν αποθηκευθεί στο $\mathbf{w}$ δε θα έχουν διαβρωθεί σημαντικά. Εάν από την άλλη πλευρά, η στατιστική μεταβολή στα δεδομένα εισόδου καθιερωθεί, τότε το $\mathbf{w}$ θα κινηθεί με έναν σταθερό, αν και πάντα στοχαστικό, τρόπο προς τη νέα περιοχή σύγκλισής του, όπως υποδεικνύεται από την (4.2.75), και με αυτόν τον τρόπο προσαρμόζόμενο στις νέες στατιστικές ιδιότητες των σημάτων εισόδου. Τώρα, αν το μήκος βήματος είναι πολύ μικρό, τότε η μνήμη του $\mathbf{w}$ είναι ισχυρή, παρουσιάζοντας αδράνεια σε τυχόν διαταραχές στα δεδομένα εισόδου, οι οποίες θα έτειναν να το μετακινήσουν από την περιοχή σύγκλισής του. Συνεπώς ο αλγόριθμος θα είναι πολύ σθεναρός. Δυστυχώς η σθεναρότητα επιτυγχάνεται σε βάρος των ικανοτήτων παρακολούθησης του αλγορίθμου. Συγκεκριμένα, σθεναρότητα και ικανότητα παρακολούθησης είναι δύο ιδιότητες, οι οποίες τείνουν η μία να ακυρώσει την άλλη, εφόσον, εάν ένας αλγόριθμος έχει πολύ γρήγορη απόκριση παρακολούθησης, τότε αυτό σημαίνει ότι οι συντελεστές του διανύσματος βάρους είναι πολύ ευαίσθητοι σε διαταραχές ή θόρυβο, και έτσι η ανοχή του αλγορίθμου σε αστάθειες είναι μικρή. Σε κάθε περίπτωση, κάποιος συμβιβασμός πρέπει να γίνει ανάμεσα σε σθεναρότητα και ταχύτητα παρακολούθησης, με κατάλληλη έμφαση να δίνεται σε κάθε ένα από αυτά τα χαρακτηριστικά, ανάλογα με τις προδιαγραφές του προς επίλυση προβλήματος.

4.2.9 Αναθεωρημένη συνθήκη ευστάθειας για πρακτικές εφαρμογές

Είδαμε προηγουμένως ότι εάν το μήκος βήματος επιλεγεί σύμφωνα με την (4.2.67), δηλαδή

$$0 < \mu < \frac{2}{\lambda_{max}}$$

τότε η αναμενόμενη τιμή $E\mathbf{w}$ του διανύσματος βάρους $\mathbf{w}$ θα συγκλίνει στο βέλτιστο διάνυσμα βάρους $\mathbf{w}^o$. Όμως, ο παραπάνω περιορισμός στο μ , παρότι θεωρητικά ορθός, δεν είναι πρακτικός, διότι ο υπολογισμός της ιδιοτιμής λ_{max} είναι πολύ δύσκολος, ιδίως όταν το L είναι πολύ μεγάλο. Συνεπώς, σε πρακτικές εφαρμογές, είναι εξαιρετικά επιθυμητή η εκτίμηση του λ_{max} (όχι όμως ο υπολογισμός) μέσω μιας απλής μεθόδου. Σημειώνουμε πρώτα ότι

$$tr(R_u) = Lr_{uu}(0), \quad (4.2.84)$$

όπου $tr(R_u)$ είναι το ίχνος του πίνακα R_u και $r_{uu}(0)$ είναι η αυτοσυσχέτιση του σήματος u υπολογισμένη για μηδενικό χρονικό lag (το όρισμα της συνάρτησης αυτοσυσχέτισης ονομάζεται lag). Εφόσον

$$r_{uu}(k) = Eu(i)u(i-k)$$

συνεπάγεται ότι

$$r_{uu}(0) = Eu(i)^2$$

το οποίο εξ'ορισμού είναι η ισχύς του σήματος u , P_u . Τώρα, εφόσον το ίχνος ενός πίνακα είναι αναλλοίωτο κάτω από ένα μετασχηματισμό ομοιότητας, και θεωρώντας τη φασματική αποσύνθεση του πίνακα R_u , όπως στην (4.2.57), προκύπτει ότι

$$tr(R_u) = tr(U\Lambda U^{-1}) = tr(\Lambda) = \sum_{n=0}^{L-1} \lambda_n, \quad (4.2.85)$$

Συνδυάζοντας τις (4.2.84) και (4.2.85) έχουμε ότι

$$LP_u = \sum_{n=0}^{L-1} \lambda_n \geq \lambda_{max}, \quad (4.2.86)$$

Συνεπώς θέτοντας

$$0 < \mu < \frac{2}{LP_u}, \quad (4.2.87)$$

εξασφαλίζουμε ότι η (4.2.67) ικανοποιείται. Είναι σημαντικό να γίνει κατανοητό, ότι αυτός ο περιορισμός εγγυάται μόνο τη σύγκλιση της ποσότητας $E\mathbf{w}$. Η σύγκλιση της ποσότητας $E\|\mathbf{w}\|^2$ ή υψηλότερων στατιστικών ροπών του $\mathbf{w}$ απαιτεί περισσότερο αυστηρούς περιορισμούς στο μ από την (4.2.67). Η εξίσωση (4.2.87) μας παρέχει κάποιες θεμελιώδεις πληροφορίες, όσον αφορά την επιλογή του μ :

1. Εφόσον το άνω όριο του μ είναι αντιστρόφως ανάλογο του L , μικρό μ χρησιμοποιείται για μεγάλης τάξης προβλέπτες $\mathbf{w}$.
2. Εφόσον το άνω όριο του μ είναι αντιστρόφως ανάλογο της ισχύος του σήματος εισόδου, σε ασθενέστερα σήματα μπορεί να χρησιμοποιηθεί μεγαλύτερο μ και σε ισχυρότερα σήματα πρέπει να χρησιμοποιηθεί μικρότερο μ . Παρ' όλα αυτά αν θεωρήσουμε ένα σενάριο ενεργού ελέγχου θορύβου, όπου θέλουμε να εξασθενίσουμε κύριο θόρυβο ισχύος P_u . Έπειτα από μετρήσεις, επιλέγουμε το μ σύμφωνα με την (4.2.87) και πράγματι ο κύριος θόρυβος εξασθενείται ικανοποιητικά. Ξαφνικά συμβαίνει μία αύξηση στην ισχύ της πηγής του θορύβου, ώστε η τιμή του μ , την οποία αρχικά επιλέξαμε, δεν ικανοποιεί πλέον την (4.2.87). Αυτό που θα συμβεί, είναι ότι ο αλγόριθμος θα γίνει ασταθής και ο αντι-θόρυβος θα αρχίσει να αυξάνεται χωρίς όριο, με κίνδυνο τραυματισμού των παρευρισκόμενων ανθρώπων, και καταστροφής τμημάτων του ηλεκτρονικού εξοπλισμού (κυρίως των ηχείων). Μία χρήσιμη προσέγγιση είναι να κανονικοποιήσουμε το μ ως προς την ισχύ P_u . Ο αλγόριθμος που προκύπτει είναι ο λεγόμενος NLMS αλγόριθμος (Normalized LMS).
3. Ένας τυπικός πρακτικός περιορισμός για το μήκος βήματος είναι ο εξής:

$$\frac{0.01}{LP_u} < \mu < \frac{0.1}{LP_u}, \quad (4.2.88)$$

Αυτή η φόρμουλα είναι μεγάλης πρακτικής χρησιμότητας, αφού προστατεύει το μηχανικό από το να επιλέξει είτε ένα «πολύ μικρό» είτε ένα «πολύ μεγάλο» μ .

4.2.10 Σταθερά χρόνου της προσαρμοστικής διαδικασίας

Μέχρι στιγμής, έχουμε δει ότι ο LMS αλγόριθμος συγκλίνει υπό την έννοια της αναμενόμενης τιμής ($E\mathbf{w}$), εφόσον η συνθήκη ευστάθειας (4.2.67) ικανοποιείται. Αυτό, για το οποίο ακόμα δεν έχουμε μιλήσει είναι η ταχύτητα αυτής της σύγκλισης. Πόσες επαναλήψεις απαιτούνται έως ότου η συνάρτηση κόστους έχει μία ικανοποιητικά μικρή τιμή? Παρ' όλα αυτά, μία καλύτερη ερώτηση προς απάντηση είναι η εξής: ποιά είναι η σταθερά χρόνου σύγκλισης της προσαρμοστικής διαδικασίας? Ως σταθερά χρόνου μίας διαδικασίας ορίζεται ο χρόνος, ο οποίος απαιτείται ώστε κάθε μεταβατική απόκριση να έχει μειωθεί στο $1/e$ της

αρχικής της τιμής. Κάθε μεταβατική απόκριση έχει τη δικιά της σταθερά χρόνου και η σταθερά χρόνου ολόκληρης της διαδικασίας είναι η σταθερά χρόνου της πιο αργά εξελισσόμενης μεταβατικής της απόκρισης.

Τώρα, από την (4.2.65) γνωρίζουμε ότι

$$\varphi_k(i) = (1 - \mu\lambda_k)^{i+1} \varphi_k(-1)$$

Αυτή η έκφραση περιγράφει τη μεταβατική απόκριση του ρυθμού k . Μπορούμε να την γράψουμε ισοδύναμα ως

$$\varphi_k(i) = e^{(i+1)\ln(1-\mu\lambda_k)} \varphi_k(-1), \quad (4.2.89)$$

Η σειρά Taylor της συνάρτησης $\ln(1+x)$ είναι

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots, \quad \text{for } -1 \leq x \leq 1, \quad (4.2.90)$$

Εάν $x \ll 1$ τότε οι μη γραμμικοί όροι μπορούν να αγνοηθούν και η (4.2.90) γίνεται

$$\ln(1+x) \approx x, \quad x \ll 1, \quad (4.2.91)$$

Βασιζόμενοι στην (4.2.91) και εάν το μ είναι τέτοιο ώστε $\mu\lambda_k \ll 1$, η (4.2.89) απλοποιείται στην

$$\varphi_k(i) = e^{-(i+1)(\mu\lambda_k)} \varphi_k(-1), \quad (4.2.92)$$

Έστω τώρα ότι η μεταβατική απόκριση $\varphi_k(i)$ θα μειωθεί στο $1/e$ της αρχικής της τιμής $\varphi_k(-1)$ έπειτα από N επαναλήψεις. Αυτό συνεπάγεται ότι

$$\varphi_k(N) = e^{-(N+1)(\mu\lambda_k)} \varphi_k(-1) = \frac{\varphi_k(-1)}{e} \Rightarrow$$

$$e^{-(N+1)(\mu\lambda_k)} = e^{-1} \Rightarrow (N+1)(\mu\lambda_k) = 1 \Rightarrow$$

$$N = \frac{1 - \mu\lambda_k}{\mu\lambda_k} \xrightarrow{\mu\lambda_k \ll 1} N \approx \frac{1}{\mu\lambda_k} \quad (4.2.93)$$

Εφόσον η (4.2.93) δίνει τη σταθερά χρόνου του ρυθμού k σε επαναλήψεις, η σταθερά χρόνου σε δευτερόλεπτα θα είναι

$$\tau_k \approx \frac{T}{\mu\lambda_k} \quad (4.2.94)$$

όπου T είναι η περίοδος δειγματοληψίας. Όπως προαναφέρθηκε, η ολική σύγκλιση περιορίζεται καθαρά από τον πιο αργό ρυθμό, από αυτούς οι οποίοι διεγείρονται, και συνεπώς η σταθερά χρόνου της LMS προσαρμογής είναι σύμφωνα με την (4.2.94)

$$\tau_{LMS} \approx \frac{T}{\mu\lambda_{\min}} \quad (4.2.95)$$

όπου $\lambda_{\min}$ είναι η απολύτως μικρότερη ιδιοτιμή του πίνακα R_u . Επειδή η τ_{LMS} είναι αντιστρόφως ανάλογη του μ , προκύπτει μεγάλη σταθερά χρόνου όταν το μήκος βήματος είναι μικρό, δηλαδή η ταχύτητα σύγκλισης είναι αργή. Εάν χρησιμοποιήσουμε μεγάλη τιμή για το μήκος βήματος, η σταθερά χρόνου είναι μικρή, το οποίο υπονοεί ότι η ταχύτητα σύγκλισης είναι μεγάλη.

Η φόρμουλα (4.2.95) δεν είναι δυνατόν να χρησιμοποιηθεί στην πράξη, εκτός εάν γνωρίζουμε εκ των προτέρων τα χαρακτηριστικά του θορύβου που θέλουμε να εξασθενίσουμε. Μπορούμε για παράδειγμα να πάρουμε μετρήσεις του θορύβου και να υπολογίσουμε off-line το $\lambda_{\min}$. Με αυτόν τον τρόπο η (4.2.95) μας δίνει μία εκτίμηση για την αναμενόμενη ταχύτητα σύγκλισης.

4.2.11 Ανάπτυξη του LMS προσαρμοζόμενου φίλτρου

Είμαστε έτοιμοι τώρα να αναπτύξουμε το φορμαλισμό του LMS προσαρμοζόμενου φίλτρου. Για αυτόν τον σκοπό θα κάνουμε μία σύντομη αλλά ουσιαστική αναδρομή στις προηγούμενες ενότητες, εστιάζοντας στα σημεία, τα οποία θα μας επιτρέψουν αβίαστα να οδηγηθούμε στην ανάπτυξη του LMS προσαρμοζόμενου φίλτρου, και ταυτόχρονα να απομονώσουμε τις εσωτερικές και εξωτερικές του λειτουργίες.

Αρχικά θέσαμε το πρόβλημα της γραμμικής πρόβλεψης μιας τυχαίας μεταβλητής d από τις L πιο πρόσφατες παρατηρήσεις $\mathbf{u}$ μίας άλλης τυχαίας μεταβλητής u . Εφόσον η διαδικασία πρόβλεψης έπρεπε να είναι γραμμική, για λόγους που ήδη εξηγήθηκαν, η πρόβλεψη του d από το u ήταν απαραίτητο να έχει τη μορφή ενός γραμμικού συνδυασμού των L πιο πρόσφατων δειγμάτων (παρατηρήσεων) του u , δηλαδή

$$\hat{d}(i) = w_0 u(i) + w_1 u(i-1) + \dots + w_{L-1} u(i-L+1), \quad (4.2.96)$$

$$\Leftrightarrow$$

$$\hat{d}(i) = \mathbf{w}^T \mathbf{u}(i), \quad (4.2.97)$$

Αναφερθήκαμε στο διάνυσμα $\mathbf{w}$, το οποίο περιέχει τους συντελεστές της πρόβλεψης, ως έναν προβλέπτη του d από το u . Στη συνέχεια απαιτήσαμε αυτός ο προβλέπτης να είναι βέλτιστος υπό την έννοια ότι το μέσο τετραγωνικό σφάλμα της πρόβλεψης $E\{(d(i) - \mathbf{w}^T \mathbf{u}(i))^2\}$, να είναι το ελάχιστο δυνατό. Έτσι αποδείξαμε ότι ο βέλτιστος προβλέπτης είναι η λύση των κανονικών εξισώσεων $R_u \mathbf{w}^o = \mathbf{r}_{ud}$. Εάν οι συντελεστές ενός FIR φίλτρου θέτονταν ίσοι με $\mathbf{w}^o$ και το σήμα u κατευθυνόταν στην είσοδο του FIR φίλτρου, τότε η έξοδος του φίλτρου $y = (\mathbf{w}^o)^T \mathbf{u}$ θα ήταν μία βέλτιστη γραμμική πρόβλεψη υπό την έννοια των μέσων ελαχίστων τετραγώνων του σήματος d . Ειδικότερα η έξοδος του FIR φίλτρου τη χρονική i θα είναι η βέλτιστη προσέγγιση του δείγματος $d(i)$ του σήματος d , δηλαδή $y(i) = (\mathbf{w}^o)^T \mathbf{u}(i) = \hat{d}(i)$. Παρ' όλα αυτά, επιχειρηματολογήσαμε ότι μία επαναληπτική διαδικασία έπρεπε να επινοηθεί, η οποία να επιλύει το σύστημα των κανονικών εξισώσεων. Αποδείξαμε ότι η steepest-descent μέθοδος, δηλαδή οι επαναλήψεις

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu(\mathbf{r}_{ud} - R_u \mathbf{w}_{i-1}), \quad i \geq 0, \quad \mathbf{w}_{-1} = \text{αρχική εκτίμηση}$$

συγκλίνει στη λύση των κανονικών εξισώσεων και συνεπώς το $\mathbf{w}_\infty$ θα είναι ο βέλτιστος προβλέπτης, τον οποίο αναζητούμε. Και πάλι, εάν το σήμα u οδηγηθεί στην είσοδο ενός FIR φίλτρου, και οι συντελεστές του φίλτρου τεθούν ίσοι με $\mathbf{w}_{i-1}$ τη χρονική i , τότε η έξοδος του φίλτρου θα είναι μία πρόβλεψη του σήματος d από το σήμα u . Πιο συγκεκριμένα, η έξοδος του FIR φίλτρου τη χρονική στιγμή i θα είναι μία πρόβλεψη του δείγματος $d(i)$ του σήματος d , δηλαδή $y(i) = \mathbf{w}_{i-1}^T \mathbf{u}(i) = \hat{d}(i)$. Τώρα, η ποιότητα αυτής της πρόβλεψης θα βελτιώνεται συνεχώς καθώς ο αριθμός των επαναλήψεων i αυξάνεται, έως ότου γίνει βέλτιστη σε χρόνο ∞ , όπου οι συντελεστές του FIR φίλτρου θα είναι πλέον ίσοι με τη λύση των κανονικών εξισώσεων. Παρ' όλα αυτά, ανεπαρκής στατιστική γνώση σε συνδυασμό με στατιστική μη στασιμότητα των σημάτων d και u , καθιστούν τις ποσότητες $\{\mathbf{r}_{ud}, R_u\}$ πρακτικά άγνωστες, επιβάλλοντας με αυτόν τον τρόπο τη χρήση της στιγμιαίας προσέγγισής τους, οδηγώντας έτσι σε έναν αλγόριθμο στοχαστικής κλίσης ή αλλιώς προσαρμοστικό, ευρέως γνωστό ως LMS προσαρμοστικό αλγόριθμο:

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{u}(i) [d(i) - \mathbf{u}(i)^T \mathbf{w}_{i-1}]$$

$$i \geq 0, \quad \mathbf{w}_{-1} = \text{αρχική εκτίμηση}$$

Τώρα, εάν οδηγήσουμε το σήμα u στην είσοδο ενός FIR φίλτρου και σε κάθε χρονική στιγμή i θέτουμε τους συντελεστές του φίλτρου ίσους με $\mathbf{w}_{LMS,i-1}$, τότε η έξοδος του φίλτρου θα είναι μία εκτίμηση του σήματος d από το σήμα u . Πιο συγκεκριμένα, η έξοδος του FIR φίλτρου τη χρονική στιγμή i θα είναι μία πρόβλεψη του δείγματος $d(i)$ του σήματος d , δηλαδή $y(i) = \mathbf{w}_{i-1}^T \mathbf{u}(i) = \hat{d}(i)$. Με την εξέλιξη του χρόνου, η ποιότητα αυτής της πρόβλεψης βελτιώνεται συνεχώς, γεγονός το οποίο σημαίνει ότι το $\mathbf{w}_{LMS,i}$ θα πλησιάζει το $\mathbf{w}^o$. Παρ' όλα αυτά, στο χρόνο ∞ το $\mathbf{w}_{LMS,i}$ δε θα έχει συγκλίνει ακριβώς στο $\mathbf{w}^o$, παρά θα εκτελεί στοχαστικές ταλαντώσεις γύρω από μία περιοχή, της οποίας το στατιστικό κέντρο θα είναι το $\mathbf{w}^o$. Αυτό το FIR φίλτρο, είναι το LMS προσαρμοζόμενο φίλτρο, το οποίο περιγράφεται πλήρως από τις παρακάτω εξισώσεις:

LMS προσαρμοζόμενο φίλτρο μορφή 1

$$y(i) = \mathbf{w}_{i-1}^T \mathbf{u}(i), \quad (4.2.98)$$

$$\mathbf{w}_{i-1} = [w_0(i-1), w_1(i-1), \dots, w_{L-1}(i-1)]^T, \quad (4.2.99)$$

$$\mathbf{u}(i) = [u(i), u(i-1), \dots, u(i-L+1)]^T, \quad (4.2.100)$$

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{u}(i) [d(i) - \mathbf{u}(i)^T \mathbf{w}_{i-1}]$$

$$i \geq 0, \quad \mathbf{w}_{-1} = \text{αρχική εκτίμηση}, \quad (4.2.101)$$

$$y(i) = \hat{d}(i), \quad (4.2.102)$$

Με μία πρώτη επισκόπηση, η ερμηνεία της λειτουργίας αυτού του προσαρμοζόμενου φίλτρου ενδέχεται να φανεί «ανόητη» στον ανυποψίαστο, αλλά προσεκτικό αναγνώστη. Παρατηρούμε ότι σε κάθε χρονική στιγμή i η έξοδος του προσαρμοζόμενου φίλτρου είναι $y(i) = \mathbf{w}_{i-1}^T \mathbf{u}(i) = \hat{d}(i)$. Αλλά προκειμένου να αναβαθμίσουμε κατά τη χρονική στιγμή i το $\mathbf{w}_{i-1}$ σε $\mathbf{w}_i$ το δείγμα $d(i)$ πρέπει να είναι διαθέσιμο, όπως η (4.2.101) υποδεικνύει:

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{u}(i) [d(i) - \mathbf{u}(i)^T \mathbf{w}_{i-1}]$$

Όμως ποιό είναι το νόημα της πρόβλεψης του δείγματος $d(i)$, δηλαδή του υπολογισμού ενός $\hat{d}(i)$, εφόσον έχουμε στη διάθεσή μας το ίδιο το $d(i)$? Η απάντηση, την οποία δίνουμε, είναι ότι το ενδιαφέρον μας ενδέχεται να βρίσκεται στην πληροφορία, η οποία αποθηκεύεται στο διάνυσμα βάρους $\mathbf{w}_i$, καθώς το προσαρμοζόμενο φίλτρο μαθαίνει να προβλέπει το σήμα d ! Στην περίπτωση αυτή η έξοδος του φίλτρου

$$y(i) = \mathbf{w}_i^T \mathbf{u}(i) = \hat{d}(i)$$

είναι ένα δευτερεύον προϊόν.

Ας υποθέσουμε από την άλλη πλευρά, ότι πράγματι ενδιαφερόμαστε για την πρόβλεψη του σήματος d από το σήμα u με τη βοήθεια του LMS προσαρμοζόμενου φίλτρου. Αυτό σημαίνει ότι τη χρονική στιγμή i θέλουμε να υπολογίσουμε μία πρόβλεψη για το δείγμα $d(i)$, χωρίς φυσικά να διαθέτουμε το $d(i)$, όπως η (4.2.101) υπαγορεύει. Οι κανονικές εξισώσεις υπονοούν ότι προκειμένου να είμαστε σε θέση, κατά οποιονδήποτε τρόπο, να προβλέψουμε το d από το u , απαιτείται κάποια πληροφορία, η οποία να συνδέει το d με το u . Στις κανονικές εξισώσεις αυτή η πληροφορία παρέχεται μέσω του διανύσματος ετεροσυσχέτισης $\mathbf{r}_{ud}$. Έτσι, εφόσον τα δείγματα του σήματος d δεν είναι διαθέσιμα και οι LMS επαναλήψεις είναι αυστηρά

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{u}(i) [d(i) - \mathbf{u}(i)^T \mathbf{w}_{i-1}]$$

πώς μπορούμε να τροποποιήσουμε την παραπάνω φόρμουλα ώστε να μην περιέχει το $d(i)$ (επειδή δεν είναι διαθέσιμο και συνεπώς δε μπορούμε να κάνουμε χρήση μίας φόρμουλας, στην οποία υπεισέρχεται) και ταυτόχρονα να εισάγουμε αυτήν την ποσότητα, η οποία μας παρέχει αυτή τη σύνδεση μεταξύ των d και u ? Η ερώτηση απαντιέται εύκολα εάν παρατηρήσουμε ότι η ποσότητα $d(i) - \mathbf{u}(i)^T \mathbf{w}_{i-1}$ δεν είναι τίποτα περισσότερο από το εκ των προτέρων σφάλμα πρόβλεψης, όπως θυμόμαστε από την (4.2.76)

$$e(i) = d(i) - \mathbf{u}_i^T \mathbf{w}_{i-1}$$

Προκύπτει ότι το εκ των προτέρων σφάλμα πρόβλεψης $e(i)$ είναι η ποσότητα, την οποία αναζητούμε, καθώς πρώτον απορροφά το $d(i)$ και δεύτερον παρέχει πληροφορία, η οποία συσχετίζει το d με το u . Φυσικά, προκειμένου να είμαστε σε θέση να χρησιμοποιήσουμε το $e(i)$, είναι απαραίτητο αυτό να μας διατίθεται μέσω κάποιων μετρήσεων. Εάν κανένα από τα d και e δεν είναι διαθέσιμα, τότε η πρόβλεψη του d από το u είναι απλώς αδύνατη. Αντικαθιστώντας το $d(i) - \mathbf{u}_i^T \mathbf{w}_{i-1}$ με το $e(i)$ στην (4.2.101), φθάνουμε στη μορφή 2 του LMS προσαρμοζόμενου φίλτρου:

LMS προσαρμοζόμενο φίλτρο μορφή 2

$$y(i) = \mathbf{w}_{i-1}^T \mathbf{u}(i), \quad (4.2.103)$$

$$\mathbf{w}_{i-1} = [w_0(i-1), w_1(i-1), \dots, w_{L-1}(i-1)]^T, \quad (4.2.104)$$

$$\mathbf{u}(i) = [u(i), u(i-1), \dots, u(i-L+1)]^T, \quad (4.2.105)$$

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{u}(i) e(i)$$

$$i \geq 0, \quad \mathbf{w}_{-1} = \text{αρχική εκτίμηση}, \quad (4.2.106)$$

$$y(i) = \hat{d}(i), \quad (4.2.107)$$

Η χρήση της μορφής 1 και της μορφής 2 του LMS προσαρμοζόμενου φίλτρου γίνεται κατανοητή, στο κεφάλαιο 2, στο πλαίσιο του ενεργού ελέγχου θορύβου.

4.3 Ο κώδικας του Ελεγκτή Ενεργού Θορύβου σε Assembly του Επεξεργαστή 21065L

Παραθέτουμε τον κώδικα του ελεγκτή σε assembly του επεξεργαστή 21065L. Ο κώδικας αποτελείται από 13 αρχεία, καθένα από τα οποία επιτελεί μία συγκεκριμένη λειτουργία. Η κατανόηση του κώδικα που ακολουθεί επαφίεται στον αναγνώστη, έπειτα από μελέτη των user manuals του επεξεργαστή.

4.3.1 “AD1819a_Initialization”

```
/** AD1819a_initialization.ASM *****/
```

```
*****/
```

```
/* ADSP-21060 System Register bit definitions */
```

```
#include "def21065l.h"
```

```
#include "new65Ldefs.h"
```

```
*****/
```

```
/* Activate the proper input selection */
```

```
/* For a microphone input activate MIC_INPUT */
```

```
/* For a line in input activate LINE_INPUT */
```

```
*****/
```

```
//#define MIC_INPUT 0
```

```
#define LINE_INPUT 1
```

```
.EXTERN spt1_svc;
```

```
.GLOBAL Program_SPORT1_Registers;
```

```
.GLOBAL Program_DMA_Controller;
```

```
.GLOBAL      AD1819_Codec_Initialization;

.GLOBAL      tx_buf;

.GLOBAL      rx_buf;

.EXTERN      Clear_All_SPT1_Regs;
```

```
/* AD1819 Codec Register Address Definitions */
```

```
#define      REGS_RESET                0x0000

#define      MASTER_VOLUME             0x0200

#define      RESERVED_REG_1            0x0400

#define      MASTER_VOLUME_MONO        0x0600

#define      RESERVED_REG_2            0x0800

#define      PC_BEEP_Volume             0x0A00

#define      PHONE_Volume               0x0C00

#define      MIC_Volume                 0x0E00

#define      LINE_IN_Volume             0x1000

#define      CD_Volume                  0x1200

#define      VIDEO_Volume               0x1400

#define      AUX_Volume                 0x1600

#define      PCM_OUT_Volume             0x1800

#define      RECORD_SELECT              0x1A00

#define      RECORD_GAIN                0x1C00

#define      RESERVED_REG_3            0x1E00

#define      GENERAL_PURPOSE            0x2000

#define      THREE_D_CONTROL_REG        0x2200

#define      RESERVED_REG_4            0x2400

#define      POWERDOWN_CTRL_STAT        0x2600

#define      SERIAL_CONFIGURATION       0x7400

#define      MISC_CONTROL_BITS          0x7600

#define      SAMPLE_RATE_GENERATE_0     0x7800

#define      SAMPLE_RATE_GENERATE_1     0x7A00

#define      VENDOR_ID_1                0x7C00
```

```
#define      VENDOR_ID_2                0x7E00
```

```
/* Mask bit selections in Serial Configuration Register for
   accessing registers on any of the 3 codecs */
```

```
#define      MASTER_Reg_Mask            0x1000
```

```
#define      SLAVE1_Reg_Mask            0x2000
```

```
#define      SLAVE2_Reg_Mask            0x4000
```

```
#define      MASTER_SLAVE1              0x3000
```

```
#define      MASTER_SLAVE2              0x5000
```

```
#define      MASTER_SLAVE1_SLAVE2 0x7000
```

```
/* Macros for setting Bits 15, 14 and 13 in Slot 0 Tag Phase */
```

```
#define      ENABLE_VFbit_SLOT1_SLOT2  0xE000
```

```
#define      ENABLE_VFbit_SLOT1         0xC000
```

```
/* AD1819 TDM Timeslot Definitions */
```

```
#define      TAG_PHASE                   0
```

```
#define      COMMAND_ADDRESS_SLOT       1
```

```
#define      COMMAND_DATA_SLOT          2
```

```
#define      STATUS_ADDRESS_SLOT         1
```

```
#define      STATUS_DATA_SLOT            2
```

```
#define      LEFT                        3
```

```
#define      RIGHT                       4
```

```
#define      AD1819_RESET_CYCLES 60
```

```
/* ad1819 RESETb spec = 1.0(uS) min */
```

```
/* 60(MIPs) = 16.67 (nS) cycle time, therefore >= 40 cycles */
```

```
#define      AD1819_WARMUP_CYCLES 60000
```

```
/* ad1819 warm-up = 1.0(mS) */
```

```
/* 60(MIPs) = 16.67 (nS) cycle time, therefore >= 40000 cycles */
```

```

/*-----*/

.segment /dm dm_codec;

.var rx_buf[5]; /* receive buffer */

/* transmit buffer */

.var tx_buf[7] = ENABLE_VFbit_SLOT1_SLOT2, /* set valid bits for slot 0, 1, and 2 */

SERIAL_CONFIGURATION, /* serial configuration register address */

0xFF80, /* initially set to 16-bit slot mode for ADI SPORT
compatibility*/

0x0000, /* stuff other slots with zeros for now */

0x0000,

0x0000,

0x0000;

/* slots 5 and 6 are dummy slots, to allow enough time in the TX ISR to go */

/* get rx slots 4 & 5 data in same audio frame as the ADC valid tag bits. */

/* This is critical for slower sample rates, where you may not have valid data */

/* every rx audio frame. So you want to make sure there is valid right */

/* channel data in the same rx DMA buffer fill as the detection of an ADC */

/* valid right bit. These extra slots are required ONLY for fs < 48 kHz. */

.var rcv_tcb[8] = 0, 0, 0, 0, 0, 5, 1, 0; /* receive tcb */

.var xmit_tcb[8] = 0, 0, 0, 0, 0, 7, 1, 0; /* transmit tcb */

/* Codec register initializations */

/* Refer to AD1819 Data Sheet for register bit assignments */

#define Select_LINE_INPUTS 0x0404 /* LINE IN - 0x0404, Mic In - 0x0000 */

#define Select_MIC_INPUT 0x0000

#define Line_Level_Volume 0x0000 /* 0 dB for line inputs */

#define Mic_Level_Volume 0x0F0F

#define Sample_Rate1 23456

```

```

#define Sample_Rate2          48000

#define Sample_Rate_8kHz  8000

.var Init_Codec_Registers[34] =

    MASTER_VOLUME,          0x0000,          /* Master Volume set for no
attenuation */

    MASTER_VOLUME_MONO,     0x8000,          /* Master Mono volume is muted */

    PC_BEEP_Volume,         0xffff,          /* PC volume is muted */

    PHONE_Volume,          0x8008,          /* Phone Volume is muted */

    MIC_Volume,             0x8008,          /* MIC Input analog loopback is muted
*/

    LINE_IN_Volume,         0x8808,          /* Line Input analog loopback is muted */

    CD_Volume,              0x8808,          /* CD Volume is muted */

    VIDEO_Volume,          0x8808,          /* Video Volume is muted */

    AUX_Volume,            0x8808,          /* AUX Volume is muted */

    PCM_OUT_Volume,         0x0000,/* 0x0808,*/ /* PCM out from DACs is 0 db gain for both
channels */

/* This is used to choose between the MIC input and Line in input */

#ifdef MIC_INPUT

    RECORD_SELECT,         Select_MIC_INPUT, /* Record Select on Line Inputs for L/R channels */

    RECORD_GAIN,           Mic_Level_Volume, /* Record Gain set for 0 dB on both L/R channels */

#else

    RECORD_SELECT,         Select_LINE_INPUTS, /* Record Select on Line Inputs for L/R channels */

    RECORD_GAIN,           Line_Level_Volume, /* Record Gain set for 0 dB on both L/R channels */

#endif

    GENERAL_PURPOSE,       0x0000,          /* 0x8000, goes through 3D circuitry */

    THREE_D_CONTROL_REG, 0x0000,          /* no phat stereo */

    MISC_CONTROL_BITS,     0x0000,          /* use SR0 for both Left and Right ADCs and
DACs, repeat sample */

    SAMPLE_RATE_GENERATE_0, 8000,          //Sample_Rate_8kHz,/* user selectable sample rate */

```


Program_SPORT1_Registers:

```
/* This is required for disabling SPORT config for EZLAB RS232 debugger */  
  
CALL Clear_All_SPT1_Regs;      /* Clear and Reset SPORT1 and DMAs */  
  
  
/* sport1 receive control register */  
  
R0 = 0x0F8C40F0;              /* 16 chans, int rfs, ext rclk, slen = 15, sden & schen enabled */  
  
dm(SRCTL1) = R0;              /* sport 0 receive control register */  
  
  
/* sport1 receive frame sync divide register */  
  
R0 = 0x00FF0000;              /* SCKfrq(12.288M)/RFSfrq(48.0K)-1 = 0x00FF */  
  
dm(RDIV1) = R0;  
  
  
/* sport1 transmit control register */  
  
R0 = 0x001C00F0;              /* 1 cyc mfd, data depend, slen = 15, sden & schen enabled */  
  
dm(STCTL1) = R0;              /* sport 0 transmit control register */  
  
  
/* sport1 receive and transmit multichannel word enable registers */  
  
R0 = 0x0000001F;              /* enable receive channels 0-4 */  
  
dm(MRCS1) = R0;  
  
R0 = 0x0000007F;              /* enable transmit channels 0-6 */  
  
dm(MTCS1) = R0;  
  
  
/* sport1 transmit and receive multichannel companding enable registers */  
  
R0 = 0x00000000;              /* no companding */  
  
dm(MRCCS1) = R0;              /* no companding on receive */  
  
dm(MTCCS1) = R0;              /* no companding on transmit */  
  
  
RTS;
```

```
/*-----*/
```

```
/*                      DMA Controller Programming For SPORT1                      */
```

```
/*-----*/
```

Program_DMA_Controller:

```
    r1 = 0x0001FFFF;          /* cpx register mask */

    /* sport1 dma control tx chain pointer register */

    r0 = tx_buf;

    dm(xmit_tcb + 7) = r0;     /* internal dma address used for chaining*/

    r0 = 1;

    dm(xmit_tcb + 6) = r0;     /* DMA internal memory DMA modifier */

    r0 = 7;

    dm(xmit_tcb + 5) = r0;     /* DMA internal memory buffer count */

    r0 = xmit_tcb + 7;         /* get DMA chaining internal mem pointer containing tx_buf address */

    r0 = r1 AND r0;            /* mask the pointer */

    r0 = BSET r0 BY 17;        /* set the pci bit */

    dm(xmit_tcb + 4) = r0;     /* write DMA transmit block chain pointer to TCB buffer */

    dm(CPT1A) = r0;           /* transmit block chain pointer, initiate tx0 DMA transfers */

    /*-----*/

    /* - Note: Tshift0 & TX0 will be automatically loaded with the first 2 values in the - */

    /* - Tx buffer. The Tx buffer pointer ( l13 ) will increment by 2x the modify value - */

    /* - ( IM3 ). - */

    /*-----*/

    /* sport1 dma control rx chain pointer register */

    r0 = rx_buf;

    dm(rcv_tcb + 7) = r0;     /* internal dma address used for chaining */

    r0 = 1;

    dm(rcv_tcb + 6) = r0;     /* DMA internal memory DMA modifier */

    r0 = 5;

    dm(rcv_tcb + 5) = r0;     /* DMA internal memory buffer count */
```

```

r0 = rcv_tcb + 7;

r0 = r1 AND r0;                                /* mask the pointer */

r0 = BSET r0 BY 17;                             /* set the pci bit */

dm(rcv_tcb + 4) = r0;    /* write DMA receive block chain pointer to TCB buffer*/

dm(CPR1A) = r0;        /* receive block chain pointer, initiate rx0 DMA transfers */

```

```

RTS;

```

```

/* ----- */
/*          AD1819A Codec Initialization          */
/* ----- */

```

```

AD1819_Codec_Initialization:

```

```

    bit set imask SPT1I;        /* enable sport0 x-mit interrupt */

```

```

Wait_Codec_Ready:                                /* Wait for CODEC Ready State */

```

```

    R0 = DM(rx_buf + 0);        /* get bit 15 status bit from AD1819 tag phase slot 0 */

    R1 = 0x8000;                /* mask out codec ready bit in tag phase */

    R0 = R0 AND R1;             /* test the codec ready status flag bit */

    IF EQ JUMP Wait_Codec_Ready; /* if flag is lo, continue to wait for a hi */

```

```

    idle;                                /* wait for a couple of TDM audio frames to pass */

    idle;

```

```

Initialize_1819_Registers:

```

```

    i4 = Init_Codec_Registers; /* pointer to codec initialization commands */

    r15 = ENABLE_VFbit_SLOT1_SLOT2; /* enable valid frame bit, and slots 1 and 2 valid data bits */

```

```

    LCNTR = 17, DO Codec_Init UNTIL LCE;

```

```

        dm(tx_buf + TAG_PHASE) = r15;    /* set valid slot bits in tag phase for slots 0, 1, 2 */

        r1 = dm(i4, 1);                  /* fetch next codec register address */

```

```

        dm(tx_buf + COMMAND_ADDRESS_SLOT) = r1;      /* put fetched codec register address into tx slot 1 */

        r1 = dm(i4, 1);                               /* fetch register data contents */

        dm(tx_buf + COMMAND_DATA_SLOT) = r1; /* put fetched codec register data into tx slot 2 */

Codec_Init:    idle;                                   /* wait until TDM frame is transmitted
*/

/*-----*/

/* Verify integrity of AD1819a indexed control register states to see if communication was successful */

/*-----*/

/* This section of codes is for debugging/verification of AD1819 registers. Theses instructions */
/* initiate codec read requests of registers shown in the Init_Codec_Registers buffer. The results */
/* of the read requests are placed in an output buffer called Codec_Init_Results, in which even */
/* DSP memory addresses contain the AD1819A register address, and the DSP's odd address in the buffer */
/* contains the register data for the AD1819A address. The AD1819A registers can then be verified with */
/* a JTAG emulator or debug monitor program. This section of code can be removed after debug. */
/*-----*/

verify_reg_writes:

    i4 = Init_Codec_Registers;

    m4 = 2;

    i5 = Codec_Init_Results;

    r15 = ENABLE_VFbit_SLOT1;                        /* enable valid frame bit, and slots 1 data bits */

    LCNTR = 17, Do ad1819_register_status UNTIL LCE;

        dm(tx_buf + TAG_PHASE) = r15;                /* set valid slot bits in tag phase for slots 0, 1, 2 */

        r1 = dm(i4,2);                               /* get indexed register address that is to be
inspected */

        r2 = 0x8000;                                  /* set bit #15 for read request in command
address word */

        r1 = r1 OR r2;                               /* OR read request with the indirect register
value */

        dm(tx_buf + COMMAND_ADDRESS_SLOT) = r1;      /* send value out of command address timeslot */

```

```

        idle;
latency in getting data */

        idle;

        r3 = dm(rx_buf + STATUS_ADDRESS_SLOT);

        dm(i5,1) = r3;

        r3 = dm(rx_buf + STATUS_DATA_SLOT);    /* fetch value of requested indexed register data */

        dm(i5,1) = r3;                        /* store to results buffer */

ad1819_register_status: nop;

/* For variable sample rate support, you must powerdown and powerback up the ADCs and DACs
so that the incoming ADC data and DAC requests occur in left/right pairs */

PowerDown_DACs_ADCs:

    idle;

    r15 = ENABLE_VFbit_SLOT1_SLOT2;          /* enable valid frame bit, and slots 1 and 2 valid data bits
*/

    dm(tx_buf + TAG_PHASE) = r15;            /* set valid slot bits in tag phase for slots 0, 1 , 2 */

    r0=POWERDOWN_CTRL_STAT;

    dm(tx_buf + COMMAND_ADDRESS_SLOT) = r0;

    r0=0x0300;                               /* power down all DACs/ADCs */

    dm(tx_buf + COMMAND_DATA_SLOT) = r0;

    idle;

    idle;

    LCNTR = AD1819_RESET_CYCLES-2, DO reset_loop UNTIL LCE;

reset_loop:    NOP;                          /* wait for the min RESETb lo spec time */

    idle;

    r15 = ENABLE_VFbit_SLOT1_SLOT2;          /* enable valid frame bit, and slots 1 and 2 valid data bits
*/

    dm(tx_buf + TAG_PHASE) = r15;            /* set valid slot bits in tag phase for slots 0, 1 , 2 */

    r0=POWERDOWN_CTRL_STAT;                  /* address to write to */

    dm(tx_buf + COMMAND_ADDRESS_SLOT) = r0;

```

```

r0=0;                                     /* power up all DACs/ADCs */

dm(tx_buf + COMMAND_DATA_SLOT) = r0;

idle;

idle;


LCNTR = AD1819_WARMUP_CYCLES-2, DO warmup_loop2 UNTIL LCE;
warmup_loop2: NOP;                       /* wait for AD1819 warm-up */


/* ----- */

bit clr imask SPT1I;                     /* disable sport1 xmit */


Install_ISR_SPORT1_Tx_ISR:

/* Use SPORT1 TX interrupt service function call for audio processing */

/* install the transmit interrupt function call to replace the initial RTI instruction */

/* "JUMP Process_AD1819_Audio_Samples" instruction into PX (0x063E 0000 xxxx). */

/* xxxx = address of Process_AD1819_Audio_Samples */

PX2 = 0x063e0000;                         /* Upper 32 bit Opcode for 'JUMP xxxx' instruction */

PX1 = Process_AD1819_Audio_Samples;      /* Lower 16 bits of Opcode contain jump address */

PM(spt1_svc) = PX;                       /* copy to 0x34 - SPORT1 interrupt vect location */


RTS;                                     /* End of AD1819A Initialization */


/* ----- */

.endseg;

```

4.3.2 “Declare_and_Initialize_Data”

```

#define S_LENGTH 40

#define W_LENGTH 60

#define Lowpass_LENGTH 51

```

```

.GLOBAL Declare_And_Initialize_Data;

//Master_Routine

.GLOBAL counter_MR;

.GLOBAL lowpass_length;

//SPM

.GLOBAL s_length;

.GLOBAL d_spm;

.GLOBAL y_spm;

.GLOBAL x_spm;

.GLOBAL e_spm;

.GLOBAL step_size_spm;

.GLOBAL leakage_spm;

//ANC

.GLOBAL w_length;

.GLOBAL y_anc;

.GLOBAL x_anc;

.GLOBAL e_anc;

.GLOBAL net_e_anc;

.GLOBAL power_x_anc;

.GLOBAL power_y_anc;

.GLOBAL power_e_anc;

.GLOBAL switch;

.GLOBAL step_size_anc;

.GLOBAL leakage_anc;

.GLOBAL scaling_factor;

.GLOBAL inverse_scaling_factor;

////////////////////////////////////

.segment/dm dm_data;

    //Master_Routine

        //vectors

```

```

.VAR lowpass_regressor_MR[Lowpass_LENGTH];

//scalars

.VAR lowpass_length = 0x00000033;

.VAR counter_MR = 1;

//SPM

//vectors

.VAR x_regressor_s_spm[S_LENGTH];

.VAR excitation_signal_spm[1000] = "chirp_new_age.dat";

//scalars

.VAR s_length = 0x00000028;

.VAR d_spm = 0.0;

.VAR y_spm = 0.0;

.VAR x_spm = 0.0;

.VAR e_spm = 0.0;

.VAR step_size_spm = 0.000000000001;

.VAR leakage_spm = 0.99999999999999;

//ANC

//vectors

.VAR y_regressor_s_anc[S_LENGTH];

.VAR x_regressor_s_anc[S_LENGTH];

.VAR x_regressor_w_anc[W_LENGTH];

.VAR xs_delay_line_anc[W_LENGTH];

.VAR sent_samples[100];

.VAR received_samples[100];

.VAR test_signal[10] = "sin_200Hz_10samples_2kHz.dat";

//scalars

.VAR w_length = 0x0000003c;

.VAR y_anc = 0.0;

.VAR x_anc = 0.0;

.VAR e_anc = 0.0;

.VAR net_e_anc = 0.0;

```

```

.VAR power_x_anc = 0.0;

.VAR power_y_anc = 0.0;

.VAR power_e_anc = 0.0;

.VAR switch = 0;

.VAR step_size_anc = 0.00000000005;

.VAR leakage_anc = 0.99999999999;

.VAR scaling_factor = 1.0;

.VAR inverse_scaling_factor = 1.0;

.endseg;

////////////////////////////////////

.segment/pm pm_data;

    //Master_Routine

    .VAR lowpass_fir_MR[Lowpass_LENGTH] = "lowpass_fir_51taps_8kHz.dat";

    //SPM

    .VAR s[S_LENGTH];

    //ANC

    .VAR w[W_LENGTH];

.endseg;

////////////////////////////////////

.segment/pm pm_code;

Declare_And_Initialize_Data:

    //Master_Routine

    b2 = lowpass_regressor_MR;

    l2 = @lowpass_regressor_MR;

    m2 = 1;

    b14 = lowpass_fir_MR;

    l14 = @lowpass_fir_MR;

    m14 = 1;

```

```

//SPM

b4 = excitation_signal_spm;

l4 = @excitation_signal_spm;

m4 = 1;


b5 = x_regressor_s_spm;

l5 = @x_regressor_s_spm;

m5 = 1;


b8 = s;

l8 = @s;

m8 = 1;

//ANC

b0 = x_regressor_w_anc;

l0 = @x_regressor_w_anc;

m0 = 1;


b1 = y_regressor_s_anc;

l1 = @y_regressor_s_anc;

m1 = 1;


b3 = x_regressor_s_anc;

l3 = @x_regressor_s_anc;

m3 = 1;


b7 = test_signal;

l7 = @test_signal;

m7 = 1;


b9 = w;

l9 = @w;

```

```
m9 = 1;
```

```
b11 = sent_samples;
```

```
l11 = @sent_samples;
```

```
m11 = 1;
```

```
b12 = received_samples;
```

```
l12 = @received_samples;
```

```
m12 = 1;
```

```
b15 = xs_delay_line_anc;
```

```
l15 = @xs_delay_line_anc;
```

```
m15 = 1;
```

```
rts;
```

```
.endseg;
```

4.3.3 “Clear_SPT1_regs”

```
/* ADSP-21060 System Register bit definitions */
```

```
#include      "def21065l.h"
```

```
#include      "new65Ldefs.h"
```

```
.GLOBAL      Clear_All_SPT1_Regs;
```

```
.SEGMENT /pm  pm_code;
```

```
Clear_All_SPT1_Regs:
```

```
IRPTL = 0x00000000;          /* clear pending interrupts */
```

```
bit clr imask SPT1l;
```

```
R0 = 0x00000000;
```

```

dm(SRCTL1) = R0;          /* sport1 receive control register */

dm(RDIV1) = R0;           /* sport1 receive frame sync divide register */

dm(STCTL1) = R0;         /* sport 0 transmit control register */

dm(MRCS1) = R0;          /* sport1 receive multichannel word enable register */

dm(MTCS1) = R0;          /* sport1 transmit multichannel word enable register */

dm(MRCCS1) = R0;         /* sport1 receive multichannel companding enable register */

dm(MTCCS1) = R0;         /* sport1 transmit multichannel companding enable register */

```

```

/* reset SPORT1 DMA parameters back to the Reset Default State */

```

```

R1 = 0x1FFFF;    dm(IIR1A) = R1;

R1 = 0x0001;     dm(IMR1A) = R1;

R1 = 0xFFFF;     dm(CR1A) = R1;

R1 = 0x1FFFF;    dm(CPR1A) = R1;

R1 = 0x1FFFF;    dm(GPR1A) = R1;

R1 = 0x1FFFF;    dm(IIT1A) = R1;

R1 = 0x0001;     dm(IMT1A) = R1;

R1 = 0xFFFF;     dm(CT1A) = R1;

R1 = 0x1FFFF;    dm(CPT1A) = R1;

R1 = 0x1FFFF;    dm(GPT1A) = R1;

```

```

RTS;

```

```

.ENDSEG;

```

4.3.4 “Codec_Processing_ISR”

```

/* ADSP-21060 System Register bit definitions */

```

```

#include      "def21065l.h"

#include      "new65Ldefs.h"

```

```

/* AD1819 TDM Timeslot Definitions */

```

```

#define      TAG_PHASE      0

```

```

#define      COMMAND_ADDRESS_SLOT      1
#define      COMMAND_DATA_SLOT         2
#define      STATUS_ADDRESS_SLOT       1
#define      STATUS_DATA_SLOT          2
#define      LEFT                       3
#define      RIGHT                      4

```

```

/* Left and Right ADC valid Bits used for testing of valid audio data in current TDM frame */

```

```

#define      M_Left_ADC                12
#define      M_Right_ADC               11
#define      DAC_Req_Left               0x80
#define      DAC_Req_Right              0x40

```

```

.GLOBAL      Process_AD1819_Audio_Samples;

.GLOBAL      Left_Channel_In;

.GLOBAL      Right_Channel_In;

.GLOBAL      Left_Channel_Out;

.GLOBAL      Right_Channel_Out;

.GLOBAL      RX_left_flag;

.GLOBAL      RX_right_flag;

.EXTERN      tx_buf;

.EXTERN      rx_buf;

.EXTERN      Master_Routine;

```

```

.segment /dm  dm_codec;

```

```

/* AD1819a stereo-channel data holders - used for DSP processing of audio data received from codec */

```

```

.VAR      Left_Channel_In;          /* Input values from AD1819A ADCs */

.VAR      Right_Channel_In;

.VAR      Left_Channel_Out;         /* Output values for AD1819A DACs */

.VAR      Right_Channel_Out;

```

```

.VAR          Left_Channel;                /* can use for intermediate results to next filter stage */

.VAR          Right_Channel;               /* can use for intermediate results to next filter stage */

.VAR          RX_left_flag;                /* DSP algorithm only processed when these bits are set */

.VAR          RX_right_flag;

.VAR          ADC_valid_bits;

/* AC'97 audio frame/ISR counter, for debug purposes */

.VAR          audio_frame_timer = 0;

.endseg;

.segment /pm pm_code;

Process_AD1819_Audio_Samples:

    /* Build Transmit Tag Phase Slot Information */

    r0 = 0x8000;                            /* Set Valid Frame bit 15 in slot 0 tag phase */

    dm(tx_buf + TAG_PHASE) = r0;           /* Write tag to tx-buf ASAP before it's shifted out of SPORT! */

    r0 = 0;                                /* Clear AC97 link Audio Output Frame
slots for now */

    dm(tx_buf + COMMAND_ADDRESS_SLOT) = r0;

    dm(tx_buf + COMMAND_DATA_SLOT) = r0;

    dm(tx_buf + LEFT) = r0;

    dm(tx_buf + RIGHT) = r0;

check_ADCs_for_valid_data:

    r0 = dm(rx_buf + TAG_PHASE);           /* Get ADC valid bits from tag phase slot*/

    r1 = 0x1800;                           /* Inspect for valid L/R ADC data */

    r2 = r0 and r1;                        /* Mask other bits in tag */

    dm(ADC_valid_bits) = r2;

```

set_tx_slot_valid_bits:

```
    r1 = dm(tx_buf + TAG_PHASE);          /* set tx valid bits based on ADC valid bits info */

    r3 = r2 or r1;                        /* set left/right channel bits in tag, if required */

    dm(tx_buf + TAG_PHASE) = r3;          /* Write tag to tx-buf ASAP before it's shifted out of SPORT! */
```

check_AD1819_ADC_left:

```
    BTST r2 by M_Left_ADC;                /* Check Master left ADC valid bit */

    IF sz JUMP check_AD1819_ADC_right;     /* If valid data then save ADC sample */

    r6 = dm(rx_buf + LEFT);               /* get Master 1819 left channel input sample */

    r6 = lshift r6 by 16;                 /* shift up to MSBs to preserve sign in 1.31 format */

    dm(Left_Channel_In) = r6;             /* save to data holder for processing */

    r4 = 1;

    dm(RX_left_flag) = r4;                /* if we have a new left sample, let the DSP filter routine know */
    dm(Left_Channel_In) = r6;             /* save to data holder for processing */
```

check_AD1819_ADC_right:

```
    BTST r2 by M_Right_ADC;               /* Check Master right ADC valid bit */

    If sz jump user_applic;               /* If valid data then save ADC sample */

    r6 = dm(rx_buf + RIGHT);              /* get Master 1819 right channel input sample */

    r6 = lshift r6 by 16;                 /* shift up to MSBs to preserve sign in 1.31 format */

    dm(Right_Channel_In) = r6;            /* save to data holder for processing */

    r4 = 1;

    dm(RX_right_flag) = r4;               /* if we have a new right sample, let the DSP filter routine know */
```

```
/* ----- */
```

```
/* user_applic( ) - User Applications Routines */
```

```
/* *** Insert DSP Algorithms Here *** */
```

```
/* */
```

```
/* Input L/R Data Streams - DM(Left_Channel_In) DM(Right_Channel_In) */
```

```
/* Output L/R Results - DM(Left_Channel_Out) DM(Right_Channel_Out) */
```

```
/* */
```

```

/* These left/right data holders are used to pipeline data through multiple modules, and */
/* can be removed if the dsp programmer needs to save instruction cycles */
/* ~~~~~ */
/* Coding TIP: */
/* The samples from the AD1819A are 16-bit and are in the lower 16 bits of the the 32-bit */
/* word. They are shifted to the most significant bit positions in order to preserve the */
/* sign of the samples when they are converted to floating point numbers. The values are */
/* also scaled to the range +/-1.0 with the integer to float conversion */
/* (f0 = float r0 by r1). */
/* */
/* To convert between our assumed 1.31 fractional number and IEEE floating point math, */
/* here are some example assembly instructions ... */
/* */
/* r1 = -31 <-- scale the sample to the range of +/-1.0 */
/* r0 = DM(Left_Channel); */
/* f0 = float r0 by r1; */
/* [Call Floating_Point_Algorithm] */
/* r1 = 31; <-- scale the result back up to MSBs */
/* r8 = fix f8 by r1; */
/* DM(Left_Channel) = r8; */
/* ----- */

user_applic:
    /* do audio processing on input samples */
    r4 = 0x0; /* since we are not using the right flag, always
clear */
    dm(RX_right_flag) = r4; /* since left & right come in pairs at same fs rate, we
only need one flag (because
we powered down ADCs for L/R sync).

Thus, the user can optionally
remove the right flag set/clear

instructions to save cycles */

```

```

    r4 = dm(RX_left_flag);

    r4 = pass r4;

    if eq jump playback_audio_data;          /* if RX_left_flag = 1, then do audio processing */

                                              /* delay routine will clear
RX_left_flag */

do_audio_processing:

    call Master_Routine;

    /* ---- DSP processing is finished, now playback results to AD1819 ---- */

playback_audio_data:

    /* Transmit Left and Right Valid Data every time there the ADCs have valid data */

    r2 = dm(ADC_valid_bits);

tx_AD1819_DAC_left:

    BTST r2 by M_Left_ADC;                  /* Check to see if we need to send DAC right sample */

    IF sz JUMP tx_AD1819_DAC_right;         /* If valid data then transmit DAC sample */

    r15 = dm(Left_Channel_Out);             /* get channel 1 output result */

    r15 = lshift r15 by -16;                /* put back in bits 0..15 for SPORT tx */

    dm(tx_buf + LEFT) = r15;               /* output right result to AD1819a Slot 3 */

tx_AD1819_DAC_right:

    BTST r2 by M_Right_ADC;                 /* Check to see if we need to send DAC right sample */

    If sz jump tx_done;                    /* If valid data then transmit DAC sample */

    r15 = dm(Right_Channel_Out);            /* get channel 2 output result */

    r15 = lshift r15 by -16;                /* put back in bits 0..15 for SPORT tx */

    dm(tx_buf + RIGHT) = r15;              /* output right result to AD1819a Slot 4 */

tx_done:

    r0=dm(audio_frame_timer);              /* get last count */

    rti(db);                               /* return from interrupt, delayed branch */

    r0=r0+1;                               /* increment count */

```

```
dm(audio_frame_timer)=r0;
```

```
/* save updated count */
```

```
/* ////////////////////////////////////// */
```

```
.endseg;
```

4.3.5 “Init_065L_EZLAB”

```
/* ADSP-21060 System Register bit definitions */
```

```
#include "def21065l.h"
```

```
#include "new65Ldefs.h"
```

```
.GLOBAL _main;
```

```
.GLOBAL Init_DSP;
```

```
.EXTERN Init_65L_SDRAM_Controller;
```

```
.EXTERN Blink_LEDs_Test;
```

```
.EXTERN Program_SPORT1_Registers;
```

```
.EXTERN Program_DMA_Controller;
```

```
.EXTERN AD1819_Codec_Initialization;
```

```
.EXTERN Declare_And_Initialize_Data;
```

```
.EXTERN irq2_svc;
```

```
/*-----*/
```

```
.SEGMENT/dm dm_data;
```

```
.endseg;
```

```
/*-----*/
```

```
.segment /pm pm_code;
```

```
_main:
```

```

call Init_65L_SDRAM_Controller;          /* Initialize External Memory */

call Program_SPORT1_Registers;           /* Initialize SPORT1 for codec communications */

call Program_DMA_Controller;             /* Start Serial Port 1 tx and rx DMA Transfers */

call AD1819_Codec_Initialization;        /* Initialize & program AD1819 */

call Declare_And_Initialize_Data;

IRPTL = 0x00000000;                      /* clear pending interrupts */

bit set imask SPT1;                       /* start audio processing, re-
enable SPORT1 tx int */

call Blink_LEDs_Test;                    /* Are We Alive? */

wait_forever:

call wait_flag1;

bit tgl ustat1 0x3F;                     /* toggle flag 4-9 LEDs */

dm(IOSTAT)=ustat1;

jump wait_forever;

/*-----*
*
*
* Subroutines
*
*
*-----*/

wait_flag1:

/* wait for flag 1 button press and release */

if flag1_in jump wait_flag1; /* wait for button press */

release:

if not flag1_in jump release; /* wait for button release */

```

rts;

/*-----*/

/* Note: This routine is first called at the Reset Vector in the Interrupt Vector Table */

/*-----*/

Init_DSP:

/* code to blink flag 4 */

ustat1=0x3F; /* flags 4 thru 9 are outputs for LEDs */

dm(IOCTL)=ustat1;

bit set ustat1 0x3F; /* toggle flag 4-9 LEDs */

dm(IOSTAT)=ustat1; /* turn on all LEDs */

bit clr mode2 FLG30|FLG20|FLG10|FLG00; /* flag 3, 2 1 & 0 inputs */

bit set mode2 IRQ1E | IRQ2E; /* irqx edge sensitive */

bit clr mode2 IRQ0E; /* keep irq1 to level sensitive for UART */

IRPTL = 0x00000000; /* clear pending interrupts */

bit set mode1 IRPTEN | NESTM; /* enable global interrupts & nesting */

bit set imask IRQ0I | IRQ1I | IRQ2I; /* irq1 and irq2 enabled, keep irq0 enabled for UART */

bit set mode1 ALUSAT; /* enable ALU saturation mode */

L0 = 0;

L1 = 0;

L2 = 0;

L3 = 0;

L4 = 0;

L5 = 0;

L6 = 0;

L7 = 0;

L8 = 0;

```

L9 = 0;

L10 = 0;

L11 = 0;

L12 = 0;

L13 = 0;

L14 = 0;

L15 = 0;

```

```

rts;

```

```

.endseg;

```

4.3.6 “SDRAM_Init”

```

/*****

```

```

    ADSP-21065L EZ-LAB Evalutation Board

```

```

    - SDRAM Interface Initialization

```

```

    - Programmable I/O Flag LED Test

```

```

*****/

```

```

#include "def21065L.h"

```

```

#include "new65Ldefs.h"

```

```

#define sdram_size 0xffff

```

```

.SEGMENT/DM segsdram;

```

```

.ENDSEG;

```

```

.SEGMENT/PM pm_code;

```

```

/*****

```

```

    Setup the SDRAM

```

Assumes SDRAM part# MT48LC1M16A1TG-12 S

SDCLK=60MHz

tCK=15ns min @ CL=2 -> SDCL=2

tRAS=72ns min -> SDTRAS=5

tRP=36ns min -> SDTRP=3

tREF=64ms/4K rows -> SDRDIV=(2(30MHz)-CL-tRP-4)64ms/4096=937cycles

2 SDRAMs by 16 bits wide total=1Mbitx32

Mapped to MS3 addresses 0x03000000-0x030fffff

*****/

.GLOBAL Init_65L_SDRAM_Controller;

.GLOBAL Blink_LEDs_Test;

Init_65L_SDRAM_Controller:

ustat1=dm(WAIT);

bit clr ustat1 0x000f8000; /*clear MS3 waitstate and mode*/

dm(WAIT)=ustat1;

ustat1=937; /*refresh rate*/

dm(SDRDIV)=ustat1;

ustat1=dm(IOCTL); /*mask in SDRAM settings*/

bit set ustat1 SDPSS|SDBN2|SDBS3|SDTRP3|SDTRAS5|SDCL2|SDPGS256|SDDCK1;

dm(IOCTL)=ustat1;

rts;

/*-----*/

Blink_LEDs_Test:

/* Setup FLAG outputs */

ustat1=dm(IOCTL);

```

        bit set ustat1 FLG40|FLG50|FLG60|FLG70|FLG80|FLG90;

        dm(IOCTL)=ustat1;                /*flag 5-9 are output*/

        ustat1=0x3f;                    /*clear flags to start*/

        dm(IOSTAT)=ustat1;

/* Blink flags 5 times */

        lcntr=10, do blink_loop until lcn;

                lcntr=15000000;

                do delay until lcn;

delay:                nop;

                ustat1=dm(IOSTAT);

                bit tgl ustat1 FLG4|FLG5|FLG6|FLG7|FLG8|FLG9;

blink_loop:    dm(IOSTAT)=ustat1;

                rts;

.ENDSEG;

```

4.3.7 "ISR_table"

```

.EXTERN        Process_AD1819_Audio_Samples;

.EXTERN        _main;

.EXTERN        Init_DSP;

.GLOBAL        spt1_svc;

.GLOBAL        irq2_svc;

.SEGMENT/PM    isr_tbl;                /* 21065L Interrupt Service Table */

/* 0x00 Reserved Interrupt */

/*          0x00    0x01    0x02    0x03    0x04 */

/* reserved_0:  NOP;    NOP;    NOP;    NOP;    NOP; */

/* *** Reset vector *** */

```

```

/* 0x05 - reset vector starts at location 0x8005 */

rst_svc: call Init_DSP;

        NOP;

        jump _main;


/* 0x08 - Reserved interrupt */

reserved_0x8:  NOP;  NOP;  NOP;  NOP;


/* 0x0C - Vector for status stack/loop stack overflow or PC stack full: */

sovfi_svc:      RTI;      RTI;      RTI;      RTI;


/* 0x10 - Vector for high priority timer interrupt: */

tmzhi_svc:      RTI;      RTI;      RTI;      RTI;


/* 0x14 - Vectors for external interrupts: */

vrpti_svc:      RTI;      RTI;      RTI;      RTI;


/* 0x18 - IRQ2 Interrupt Service Routine (ISR) */

irq2_svc:      RTI;      RTI;      RTI;      RTI;


/* 0x1C - IRQ1 Interrupt Service Routine (ISR) */

irq1_svc:      RTI;      RTI;      RTI;      RTI;


/* IRQ0 INTERRUPT VECTOR */

/* ----- */

/* IRQ0 Interrupt, Used by the UART and Debug Monitor Program */

/* This is separated from rest of the Interrupt Vector Table to prevent the user

from overwriting this location, since it is used by the host monitor program

Do not use unless developing an application which requires it's use without

the host monitor program.

This vector location is written during power-on and protected from over-writing

```

by rewriting it after writing data to memory (so this declaration is redundant, but a good reminder). Over-writing this vector will stop the kernel.

The user may over-write it from his/her code - but will lose the capability to communicate over the RS-232 interface.*/

```
/* *** 0x20 - IRQ0 Interrupt Service Routine (ISR) , 4 locations max *** */
```

```
irq0_svc:
```

```
    /* JUMP KERNEL_UART_ISR; */ RTI;    RTI;    RTI;
```

```
/* 0x24 - Reserved interrupt */
```

```
reserved_0x24:  NOP;    NOP;    NOP;    NOP;
```

```
/* 0x28 - Vectors for Serial Port 0 Receive A & B DMA channels 0/1 */
```

```
spr0_svc:      RTI;    RTI;    RTI;    RTI;
```

```
/* 0x2C - Vectors for Serial Port 1 Receive A & B DMA channels 2/3 */
```

```
spr1_svc:      RTI;    RTI;    RTI;    RTI;
```

```
/* 0x30 - Vectors for Serial Port 0 Transmint A & B DMA channels 4/5 */
```

```
spt0_svc:      RTI;    RTI;    RTI;    RTI;
```

```
/* 0x34 - Vectors for Serial Port 1 Transmit A & B DMA channels 6/7 */
```

```
/* Note: In this example, the host monitor's codec routine ' JUMP KERNEL_CODEC_ISR'
```

is overwritten. This is ok as long as the user does not require host software control of the codec, as in this demo */

```
/* After codec initialization, the following instruction will be
```

installed at this vector for normal codec processing:

```
'Process_AD1819_Audio_Samples;' */
```

```
spt1_svc:      RTI;    RTI;    RTI;    RTI;
```

```
/* 0x38 - Reserved Interrupt */
```

reserved_0x38: RTI; RTI; RTI; RTI;

/* 0x3C - Reserved Interrupt */

reserved_0x3c: RTI; RTI; RTI; RTI;

/* 0x40 - Vector for External Port DMA channel 8 */

ep0_svc:RTI; RTI; RTI; RTI;

/* 0x44 - Vector for External Port DMA channel 9 */

ep1_svc:RTI; RTI; RTI; RTI;

/* 0x48 - Reserved Interrupt */

reserved_0x48: RTI; RTI; RTI; RTI;

/* 0x4C - Reserved Interrupt */

reserved_0x4c: RTI; RTI; RTI; RTI;

/* 0x50 - Reserved Interrupt */

reserved_0x50: RTI; RTI; RTI; RTI;

/* 0x54 - Vector for DAG1 buffer 7 circular buffer overflow */

cb7_svc: RTI; RTI; RTI; RTI;

/* 0x58 - Vector for DAG2 buffer 15 circular buffer overflow */

cb15_svc: RTI; RTI; RTI; RTI;

/* 0x5C - Vector for lower priority timer interrupt */

tmzl_svc: RTI; RTI; RTI; RTI;

/* 0x60 - Vector for fixed-point overflow */

fix_svc: RTI; RTI; RTI; RTI;

```
/* 0x64 - Floating-point overflow exception */
```

```
flt0_svc: RTI;    RTI;    RTI;    RTI;
```

```
/* 0x68 - Floating-point underflow exception */
```

```
flt_u_svc: RTI;    RTI;    RTI;    RTI;
```

```
/* 0x6C - Floating-point invalid exception */
```

```
flt_i_svc: RTI;    RTI;    RTI;    RTI;
```

```
/* 0x70 - User software interrupt 0 */
```

```
sft0_svc: RTI;    RTI;    RTI;    RTI;
```

```
/* 0x74 - User software interrupt 1 */
```

```
sft1_svc: RTI;    RTI;    RTI;    RTI;
```

```
/* 0x78 - User software interrupt 2 */
```

```
sft2_svc: RTI;    RTI;    RTI;    RTI;
```

```
/* 0x7C - User software interrupt 3 */
```

```
/* SW interrupt 3 is used on the ADSP-21065L EZ-LAB Host Software
```

to give the ability to send messages from user code to the debugger.

Unless your are implementing an embedded application, we recommend

not overwriting this location so the monitor has functionality after

downloading of code.

```
*/
```

```
#define KERNEL_SWI3_ISR 0x9002
```

```
sft3_svc: JUMP KERNEL_SWI3_ISR; RTI;    RTI;    RTI;
```

```
.ENDSEG;
```

4.3.8 “ANC”

```
.GLOBAL      ANC;

.EXTERN      w_length;

.EXTERN      s_length;

.EXTERN      e_anc;

.EXTERN      x_anc;

.EXTERN      y_anc;

.EXTERN      net_e_anc;

.EXTERN      power_x_anc;

.EXTERN      power_y_anc;

.EXTERN      power_e_anc;

.EXTERN      step_size_anc;

.EXTERN      leakage_anc;

.EXTERN      inverse_scaling_factor;

.EXTERN      scaling_factor;

.EXTERN      Left_Channel_In;

.EXTERN      Right_Channel_In;

////////////////////////////////////

.segment /pm pm_code;

ANC:

/*error signal e converted to single precision floating
point format. the instruction lshift r0 by -16 does not
preserve the sign extension and so, specific check for
negative numbers must be made */

    r0 = dm(Right_Channel_In);

    r1 = 0x80000000;

    r1 = r0 and r1;

    if EQ jump positive_anc;

negative_anc:

    r0 = lshift r0 by -16;
```

```

r1 = 0xffff0000;

r0 = r1 or r0;

f0 = float r0;

dm(net_e_anc) = f0;

jump filter_through_bandpass_anc;

```

positive_anc:

```

r0 = lshift r0 by -16;

f0 = float r0;

dm(net_e_anc) = f0;

```

```

/*a low pass filter is used with Fc=500Hz in order to enhance
the stability of the adaptive filter by cutting off irrelevant
high frequencies*/

```

filter_through_bandpass_anc:

```

f1 = dm(inverse_scaling_factor);

f0 = f0 * f1;

dm(e_anc) = f0;

pm(i12,m12) = f0;

```

```

/*the output signal is filtered through the approximation of the
secondary path transfer function*/

```

filter_y_through_s:

```

r12=r12 xor r12;

f8=pass f12, f0=dm(i1,m1), f4=pm(i8,m8);

```

```

r6 = dm(s_length);

r6 = r6 - 1;

lcctr=r6, do macloop_anc_1 until lce;

```

macloop_anc_1:f12=f0*f4, f8=f8+f12, f0=dm(i1,m1), f4=pm(i8,m8);

```

f12=f0*f4, f8=f8+f12;

```

```
f8=f8+f12;
```

```
//create an estimation of the true output
```

```
//f15 = dm(scaling_factor);
```

```
//f15 = f15 * f8;
```

```
pm(i11,m11) = f8;
```

```
//reference signal x synthesis:
```

```
f0 = dm(e_anc);
```

```
f8 = f8 + f0;
```

```
dm(i0,m0) = f8;
```

```
dm(i3,m3) = f8;
```

```
dm(x_anc) = f8;
```

```
/*the antinoise is generated as the output of the
```

```
adaptive filter w*/
```

```
filter_x_through_w:
```

```
r12=r12 xor r12;
```

```
f8=pass f12, f0=dm(i0,m0), f4=pm(i9,m9);
```

```
r6 = dm(w_length);
```

```
r6 = r6 - 1;
```

```
lcctr=r6, do macloop_anc_2 until lce;
```

```
macloop_anc_2:f12=f0*f4, f8=f8+f12, f0=dm(i0,m0), f4=pm(i9,m9);
```

```
f12=f0*f4, f8=f8+f12;
```

```
f8=f8+f12;
```

```
dm(y_anc) = f8;
```

```
dm(i1,m1) = f8;
```

```
/*the reference signal is filtered through the approximation
```

```
of the secondary path transfer function*/
```

filter_x_through_s:

r12=r12 xor r12;

f8=pass f12, f0=dm(i3,m3), f4=pm(i8,m8);

r6 = dm(s_length);

r6 = r6 - 1;

lcnter=r6, do macloop_anc_3 until lce;

macloop_anc_3:f12=f0*f4, f8=f8+f12, f0=dm(i3,m3), f4=pm(i8,m8);

f12=f0*f4, f8=f8+f12;

f8=f8+f12;

pm(i15,m15) = f8;

//adapt w

f15 = dm(leakage_anc);

f8 = dm(step_size_anc);

f9 = dm(e_anc);

f9 = f9 * f8;

r7 = dm(w_length);

lcnter = r7;

do adapt_w until lce;

f8 = pm(i15,m15);

f8 = f8 * f9;

f10 = pm(i9,0);

f10 = f10 * f15;

f10 = f10 + f8;

adapt_w:pm(i9,m9) = f10;

//compute power_x in a window of 10000 samples

/*f0 = dm(power_x_anc);

f1 = dm(x_anc);

```

f1 = f1 * f1;

f2 = 0.0001;

f1 = f1 * f2;

f2 = 0.9999;

f0 = f0 * f2;

f0 = f0 + f1;

dm(power_x_anc) = f0;*/

```

//compute power_y in a window of 10000 samples

```

/*      f0 = dm(power_y_anc);

        f1 = dm(y_anc);

        f1 = f1 * f1;

        f2 = 0.0001;

        f1 = f1 * f2;

        f2 = 0.9999;

        f0 = f0 * f2;

        f0 = f0 + f1;

        dm(power_y_anc) = f0;*/

```

//compute power_e in a window of 10000 samples

```

f0 = dm(power_e_anc);

f1 = dm(net_e_anc);

f1 = f1 * f1;

f2 = 0.0001;

f1 = f1 * f2;

f2 = 0.9999;

f0 = f0 * f2;

f0 = f0 + f1;

dm(power_e_anc) = f0;

```

```

//send intinoise to loudspeakers

f8 = dm(y_anc);

f0 = dm(scaling_factor);

f8 = f8 * f0;

f8 = -f8;

dm(i2,m2) = f8;


/*r0 = dm(i7,m7);

f0 = float r0;

dm(i2,m2) = f0,*/

rts;

.endseg;

```

```

////////////////////////////////////

```

4.3.9 “Master_Routine”

```

////////////////////////////////////

```

```

.GLOBAL Master_Routine;

.EXTERN Router_Routine;

.EXTERN RX_left_flag;

.EXTERN Left_Channel_Out;

.EXTERN Right_Channel_Out;

.EXTERN counter_MR;

.EXTERN lowpass_length;

```

```

////////////////////////////////////

```

```

.segment /pm pm_code;

```

```

Master_Routine:

```

```

/*Due to the following instructions data is passed to the user code
at a frequency of 2kHz, although the codec still runs at 8kHz.*/

```

```

r0 = dm(counter_MR);

r1 = 1;

comp(r0,r1);

```

```

if EQ jump user_processing;

r2 = 4;

comp(r0,r2);

if EQ jump next_sample_valid_2kHz;

r0 = r0 + r1;

dm(counter_MR) = r0;

f2 = 0.0;

dm(i2,m2) = f2;

jump do_filtering_MR;

next_sample_valid_2kHz:

dm(counter_MR) = r1;

f2 = 0.0;

dm(i2,m2) = f2;

jump do_filtering_MR;


user_processing:

r0 = r0 + r1;

dm(counter_MR) = r0;

call Router_Routine;


do_filtering_MR:

r12=r12 xor r12;

f8=pass f12, f0=dm(i2,m2), f4=pm(i14,m14);


r6 = dm(lowpass_length);

r6 = r6 - 1;

lcctr=r6, do macloop until lce;

macloop:f12=f0*f4, f8=f8+f12, f0=dm(i2,m2), f4=pm(i14,m14);


f12=f0*f4, f8=f8+f12;

f8=f8+f12;

```

```

f0 = 4.0;

f8 = f8 * f0;

r0 = fix f8;

r0 = lshift r0 by 16;

dm(Right_Channel_Out) = r0;

dm(Left_Channel_Out) = r0;

```

end_of_master_routine:

```

rts(db);

r4 = 0;

dm(RX_left_flag) = r4;

```

.endseg;

4.3.10 “Router_Routine”

```

.EXTERN switch;

.EXTERN SPM;

.EXTERN ANC;

.GLOBAL Router_Routine;

////////////////////////////////////

.segment /pm pm_code;

Router_Routine:

    r0 = dm(switch);

    r1 = 1;

    comp(r0,r1);

    if EQ jump route_anc;

route_spm:call SPM;

    jump end_router_routine;

route_anc:call ANC;

end_router_routine: rts;

```

```
.endseg;
```

```
////////////////////////////////////
```

4.3.11 “SPM”

```
/*This routine models the Secondary Path Transfer Function S(z) off-line
```

```
using the LMS algorithm.*/
```

```
.GLOBAL SPM;
```

```
.EXTERN s_length;
```

```
.EXTERN d_spm;
```

```
.EXTERN y_spm;
```

```
.EXTERN x_spm;
```

```
.EXTERN e_spm;
```

```
.EXTERN step_size_spm;
```

```
.EXTERN leakage_spm;
```

```
.EXTERN Left_Channel_In;
```

```
.EXTERN Right_Channel_In;
```

```
////////////////////////////////////
```

```
.segment /pm pm_code;
```

```
SPM:
```

```
//x sent to output as 1.31 fractional format
```

```
    r0 = dm(i4,m4);
```

```
    f0 = float r0;
```

```
    dm(i2,m2) = f0;
```

```
/*x converted to single precision floating point format
```

```
and stored in x_regressor. x_regressor is stored in
```

```
memory with sign extension.*/
```

```
    dm(i5,m5) = f0;
```

```
    dm(x_spm) = f0;
```

```
/*desired value d converted to single precision floating
```

```
point format. the instruction lshift r0 by -16 does not
```

preserve the sign extension and so, specific check for

negative numbers must be made */

```
r0 = dm(Right_Channel_In);
```

```
r1 = 0x80000000;
```

```
r1 = r0 and r1;
```

```
if EQ jump positive_spm;
```

negative_spm:

```
r0 = lshift r0 by -16;
```

```
//add sign extension for negative numbers
```

```
r1 = 0xffff0000;
```

```
r0 = r1 or r0;
```

```
f0 = float r0;
```

```
jump filter_through_bandpass_spm;
```

positive_spm:

```
r0 = lshift r0 by -16;
```

```
f0 = float r0;
```

/*a low pass filter is used with $F_c=500\text{Hz}$ in order to enhance

the stability of the adaptive filter w by cutting off irrelevant

high frequencies. This filter actually belongs to the secondary

path and s must account for it.*/

filter_through_bandpass_spm:

```
dm(d_spm) = f0;
```

//create output of adaptive filter s

filter_x_spm_through_s:

```
r12=r12 xor r12;
```

```
f8=pass f12, f0=dm(i5,m5), f4=pm(i8,m8);
```

```
r6 = dm(s_length);
```

```

    r6 = r6 - 1;

    lcctr=r6, do macloop_spm_1 until lce;
macloop_spm_1:f12=f0*f4, f8=f8+f12, f0=dm(i5,m5), f4=pm(i8,m8);


    f12=f0*f4, f8=f8+f12;

    f8=f8+f12;

    dm(y_spm) = f8;


//compute error_signal e

    f9 = dm(d_spm);

    f10 = f9 - f8;

    dm(e_spm) = f10;


//adapt s

    f11 = dm(step_size_spm);

    f11 = f11 * f10;

    f15 = dm(leakage_spm);

    r6 = dm(s_length);

    lcctr=r6, do adapt_s until lce;

    f13 = dm(i5,m5);

    f13 = f13 * f11;

    f14 = pm(i8,0);

    f14 = f14 * f15;

    f14 = f14 + f13;

adapt_s:pm(i8,m8) = f14;


exit_SPM:

    rts;


.endseg;

```

[illegible]

/*****
 *****/

```
#define FLG4      0x00000001
```

```
#define FLG6      0x00000004
```

```
#define FLG8      0x00000010
```

```
#define FLG10    0x00000040
```

```
/* IOCTL macro definitions */
```

```
#define FLG50    0x00000002
```

```
#define FLG70    0x00000008
```

```
#define FLG90    0x00000020
```

```
#define FLG11O 0x00000080
```

```
#define DSDCTL 0x00000400
```

```
#define SDDCK1 0x00000800
```

```
#define SDPGS1024      0x00000000
```

```
#define SDPGS512      0x00001000
```

```
#define SDPGS256      0x00002000
```

#define SDSFR 0x00008000

#define SDCL1 0x00010000

#define SDCL2 0x00020000

#define SDCL3 0x00030000

#define SDTRAS0 0x00000000

#define SDTRAS1 0x00040000

#define SDTRAS2 0x00080000

#define SDTRAS3 0x000c0000

#define SDTRAS4 0x00100000

#define SDTRAS5 0x00140000

#define SDTRAS6 0x00180000

#define SDTRAS7 0x001c0000

#define SDTRP0 0x00000000

#define SDTRP1 0x00200000

#define SDTRP2 0x00400000

#define SDTRP3 0x00600000

#define SDTRP4 0x00800000

#define SDTRP5 0x00a00000

#define SDTRP6 0x00c00000

#define SDTRP7 0x00e00000

#define SDPM 0x01000000

#define SDBS0 0x08000000

#define SDBS1 0x0a000000

#define SDBS2 0x0c000000

#define SDBS3 0x0e000000

```
#define SDBUF 0x10000000
```

```
#define SDBN2 0x00000000
```

```
#define SDBN4 0x20000000
```

```
#define SDPSS 0x80000000
```

```
/* **** */
```

4.3.13 “def21065L”

```
/* ----- */
```

def21065L.h - SYSTEM AND IOP REGISTER BIT AND ADDRESS DEFINITIONS FOR ADSP-21065L

Last Modification on: Feb-15-95

This include file contains a list of macro defines to enable the programmer to

use symbolic names for all of the system register bits for the ADSP-21065L.

It also contains macros for the IOP register addresses and some bit fields.

```
----- */
```

```
/* MODE1 register */
```

```
#define BR8 0x00000001 /* Bit 0: Bit-reverse for I8 */
```

```
#define BR0 0x00000002 /* Bit 1: Bit-reverse for IO (uses DMS0- only) */
```

```
#define SRCU 0x00000004 /* Bit 2: Alt. register select for comp. units */
```

```
#define SRD1H 0x00000008 /* Bit 3: DAG1 alt. register select (7-4) */
```

```
#define SRD1L 0x00000010 /* Bit 4: DAG1 alt. register select (3-0) */
```

```
#define SRD2H 0x00000020 /* Bit 5: DAG2 alt. register select (15-12) */
```

```
#define SRD2L 0x00000040 /* Bit 6: DAG2 alt. register select (11-8) */
```

```
#define SRRFH 0x00000080 /* Bit 7: Register file alt. select for R(15-8) */
```

```
#define SRRFL 0x00000400 /* Bit 10: Register file alt. select for R(7-0) */
```

```
#define NESTM 0x00000800 /* Bit 11: Interrupt nesting enable */
```

```
#define IRPTEN 0x00001000 /* Bit 12: Global interrupt enable */
```

```
#define ALUSAT 0x00002000 /* Bit 13: Enable ALU fixed-pt. saturation */
```

```

#define SSE    0x00004000 /* Bit 14: Enable short word sign extension    */

#define TRUNC  0x00008000 /* Bit 15: 1=fltg-pt. truncation 0=Rnd to nearest */

#define RND32   0x00010000 /* Bit 16: 1=32-bit fltg-pt.rounding 0=40-bit rnd */

#define CSEL    0x00060000 /* Bit 17-18: CSelect: Bus Mastership    */


/* MODE2 register */

#define IRQ0E    0x00000001 /* Bit 0: IRQ0- 1=edge sens. 0=level sens.    */

#define IRQ1E    0x00000002 /* Bit 1: IRQ1- 1=edge sens. 0=level sens.    */

#define IRQ2E    0x00000004 /* Bit 2: IRQ2- 1=edge sens. 0=level sens.    */

#define PERIOD_CNT0    0x00000008 /* Bit 3: Enable Period Count    */

#define CADIS    0x00000010 /* Bit 4: Cache disable    */

#define TIMEN0    0x00000020 /* Bit 5: Timer0 enable    */

#define BUSLK    0x00000040 /* Bit 6: External bus lock    */

#define PWMOUT0    0x00000080 /* Bit 7: PWMOUT/WIDTH_CNT control-Timer0    */

#define INT_HI0    0x00000100 /* Bit 8: Interrupt Vector location    */

#define PULSE_HI0    0x00000200 /* Bit 9: Pulse transition edge select    */

#define PERIOD_CNT1    0x00000400 /* Bit 10: Enable Period Count    */

#define TIMEN1    0x00000800 /* Bit 11: Timer0 enable    */

#define PWMOUT1    0x00001000 /* Bit 12: PWMOUT/WIDTH_CNT control-Timer1    */

#define INT_HI1    0x00002000 /* Bit 13: Interrupt Vector location    */

#define PULSE_HI1    0x00004000 /* Bit 14: Pulse transition edge select    */

#define FLG00    0x00008000 /* Bit 15: FLAG0 1=output 0=input    */

#define FLG10    0x00010000 /* Bit 16: FLAG1 1=output 0=input    */

#define FLG20    0x00020000 /* Bit 17: FLAG2 1=output 0=input    */

#define FLG30    0x00040000 /* Bit 18: FLAG3 1=output 0=input    */

#define CAFRZ    0x00080000 /* Bit 19: Cache freeze    */


/* ASTAT register */

#define AZ    0x00000001 /* Bit 0: ALU result zero or fltg-pt. underflow */

#define AV    0x00000002 /* Bit 1: ALU overflow    */

#define AN    0x00000004 /* Bit 2: ALU result negative    */

```

```

#define AC    0x00000008 /* Bit 3: ALU fixed-pt. carry          */
#define AS    0x00000010 /* Bit 4: ALU X input sign (ABS and MANT ops)  */
#define AI    0x00000020 /* Bit 5: ALU fltg-pt. invalid operation      */
#define MN    0x00000040 /* Bit 6: Multiplier result negative          */
#define MV    0x00000080 /* Bit 7: Multiplier overflow                 */
#define MU    0x00000100 /* Bit 8: Multiplier fltg-pt. underflow       */
#define MI    0x00000200 /* Bit 9: Multiplier fltg-pt. invalid operation */
#define AF    0x00000400 /* Bit 10: ALU fltg-pt. operation             */
#define SV    0x00000800 /* Bit 11: Shifter overflow                  */
#define SZ    0x00001000 /* Bit 12: Shifter result zero                */
#define SS    0x00002000 /* Bit 13: Shifter input sign                 */
#define BTF   0x00040000 /* Bit 18: Bit test flag for system registers */
#define FLG0   0x00080000 /* Bit 19: FLAG0 value                       */
#define FLG1   0x00100000 /* Bit 20: FLAG1 value                       */
#define FLG2   0x00200000 /* Bit 21: FLAG2 value                       */
#define FLG3   0x00400000 /* Bit 22: FLAG3 value                       */
#define CACC0  0x01000000 /* Bit 24: Compare Accumulation Bit 0         */
#define CACC1  0x02000000 /* Bit 25: Compare Accumulation Bit 1         */
#define CACC2  0x04000000 /* Bit 26: Compare Accumulation Bit 2         */
#define CACC3  0x08000000 /* Bit 27: Compare Accumulation Bit 3         */
#define CACC4  0x10000000 /* Bit 28: Compare Accumulation Bit 4         */
#define CACC5  0x20000000 /* Bit 29: Compare Accumulation Bit 5         */
#define CACC6  0x40000000 /* Bit 30: Compare Accumulation Bit 6         */
#define CACC7  0x80000000 /* Bit 31: Compare Accumulation Bit 7         */

/* STKY register */
#define AUS    0x00000001 /* Bit 0: ALU fltg-pt. underflow             */
#define AVS    0x00000002 /* Bit 1: ALU fltg-pt. overflow               */
#define AOS    0x00000004 /* Bit 2: ALU fixed-pt. overflow              */
#define AIS    0x00000020 /* Bit 5: ALU fltg-pt. invalid operation      */
#define MOS    0x00000040 /* Bit 6: Multiplier fixed-pt. overflow       */

```

```

#define MVS    0x00000080 /* Bit 7: Multiplier fltg-pt. overflow */
#define MUS    0x00000100 /* Bit 8: Multiplier fltg-pt. underflow */
#define MIS    0x00000200 /* Bit 9: Multiplier fltg-pt. invalid operation */
#define CB7S   0x00020000 /* Bit 17: DAG1 circular buffer 7 overflow */
#define CB15S  0x00040000 /* Bit 18: DAG2 circular buffer 15 overflow */
#define PCFL   0x00200000 /* Bit 21: PC stack full */
#define PCEM   0x00400000 /* Bit 22: PC stack empty */
#define SSOV   0x00800000 /* Bit 23: Status stack overflow (MODE1 and ASTAT) */
#define SSEM   0x01000000 /* Bit 24: Status stack empty */
#define LSOV   0x02000000 /* Bit 25: Loop stack overflow */
#define LSEM   0x04000000 /* Bit 26: Loop stack empty */

/* IRPTL and IMASK and IMASKP registers */
#define RSTI   0x00000002 /* Bit 1: Offset: 04: Reset */
#define SOVFI   0x00000008 /* Bit 3: Offset: 0c: Stack overflow */
#define TMZHI  0x00000010 /* Bit 4: Offset: 10: Timer = 0 (high priority) */
#define VIRPTI  0x00000020 /* Bit 5: Offset: 14: Vector interrupt */
#define IRQ2I  0x00000040 /* Bit 6: Offset: 18: IRQ2- asserted */
#define IRQ1I  0x00000080 /* Bit 7: Offset: 1c: IRQ1- asserted */
#define IRQ0I  0x00000100 /* Bit 8: Offset: 20: IRQ0- asserted */
#define SPR0I   0x00000400 /* Bit 10: Offset: 28: SPORT0 receive */
#define SPR1I   0x00000800 /* Bit 11: Offset: 2c: SPORT1 receive */
#define SPT0I   0x00001000 /* Bit 12: Offset: 30: SPORT0 transmit */
#define SPT1I   0x00002000 /* Bit 13: Offset: 34: SPORT1 transmit */
#define EPOI    0x00010000 /* Bit 16: Offset: 40: External port channel 0 DMA */
#define EP1I    0x00020000 /* Bit 17: Offset: 44: External port channel 1 DMA */
#define CB7I   0x00200000 /* Bit 21: Offset: 54: Circ. buffer 7 overflow */
#define CB15I  0x00400000 /* Bit 22: Offset: 58: Circ. buffer 15 overflow */
#define TMZLI  0x00800000 /* Bit 23: Offset: 5c: Timer = 0 (low priority) */
#define FIXI   0x01000000 /* Bit 24: Offset: 60: Fixed-pt. overflow */
#define FLTOI  0x02000000 /* Bit 25: Offset: 64: fltg-pt. overflow */

```

```

#define FLTUI 0x04000000 /* Bit 26: Offset: 68: fltg-pt. underflow */
#define FLTII 0x08000000 /* Bit 27: Offset: 6c: fltg-pt. invalid */
#define SFTOI 0x10000000 /* Bit 28: Offset: 70: user software int 0 */
#define SFTI1 0x20000000 /* Bit 29: Offset: 74: user software int 1 */
#define SFTI2 0x40000000 /* Bit 30: Offset: 78: user software int 2 */
#define SFTI3 0x80000000 /* Bit 31: Offset: 7c: user software int 3 */

/* SYSCON Register */
#define SYSCON 0x00 /* Memory mapped System Configuration Register */
#define SRST 0x00000001 /* Soft Reset */
#define BSO 0x00000002 /* Boot Select Override */
#define IIVT 0x00000004 /* Internal Interrupt Vector Table */
#define HBW00 0x00000000 /* Host Bus Width: 32bit */
#define HBW01 0x00000010 /* Host Bus Width: 16bit */
#define HBW10 0x00000020 /* Host Bus Width: 8bit */
#define HMSWF 0x00000040 /* Host packing order (0 = LSW first, 1 = MSW) */
#define HPFLSH 0x00000080 /* Host pack flush */
#define IMDW0X 0x00000100 /* Internal memory block 0, extended data (40 bit) */
#define IMDW1X 0x00000200 /* Internal memory block 1, extended data (40 bit) */
#define EBPR00 0x00000000 /* External bus priority: Even */
#define EBPR01 0x00010000 /* External bus priority: Core has priority */
#define EBPR10 0x00020000 /* External bus priority: IO has priority */
#define DCPR 0x00040000 /* Select rotating access priority on DMA6 - DMA9 */

/* SYSTAT Register */
#define SYSTAT 0x03 /* Memory mapped System Status Register */
#define HSTM 0x00000001 /* Host is the Bus Master */
#define BSYN 0x00000002 /* Bus arbitration logic is synchronized */
#define CRBM 0x00000030 /* Current ADSP21065L Bus Master */
#define IDC 0x00000300 /* ADSP21065L ID Code */
#define VIPD 0x00002000 /* Vector interrupt pending (1 = pending) */

```

```
#define HPS 0x00004000 /* Host pack status */
```

```
/*----- SYSTEM registers -----*/
```

```
#define SYSCON 0x00
```

```
#define VIRPT 0x01
```

```
#define WAIT 0x02
```

```
#define SYSTAT 0x03
```

```
/*----- DMA BUFFER registers -----*/
```

```
/* These are old register names */
```

```
#define DMAB0 0x04
```

```
#define DMAB1 0x05
```

```
/* These are the new register names */
```

```
#define EPB0 0x04
```

```
#define EPB1 0x05
```

```
/*----- MESSAGE registers -----*/
```

```
#define MSGR_0 0x08
```

```
#define MSGR_1 0x09
```

```
#define MSGR_2 0x0a
```

```
#define MSGR_3 0x0b
```

```
#define MSGR_4 0x0c
```

```
#define MSGR_5 0x0d
```

```
#define MSGR_6 0x0e
```

```
#define MSGR_7 0x0f
```

```
/*----- MISCELLANEOUS registers -----*/
```

```
#define      BMAX      0x18
```

```
#define      BCNT      0x19
```

```
#define      ELAST      0x1b
```

```
/*----- DMAC registers -----*/
```

```
/* These are the old register names */
```

```
#define      DMAC6      0x1c
```

```
#define      DMAC7      0x1d
```

```
/* These are the new register names */
```

```
#define      DMAC0      0x1c
```

```
#define      DMAC1      0x1d
```

```
/*----- 21065L registers -----*/
```

```
#define      SDRDIV      0x20
```

```
#define      SDRCNT      0x21
```

```
#define      SDADDLAT      0x22
```

```
#define      TPERIOD0      0x28
```

```
#define      TPWIDTH0      0x29
```

```
#define      TCOUNT0      0x2a
```

```
#define      TPERIOD1      0x2b
```

```
#define      TPWIDTH1      0x2c
```

```
#define      TCOUNT1      0x2d
```

```
#define   IOCTL      0x2e

#define   IOSTAT     0x2f
```

```
/*----- DMA ADDRESS registers -----*/
```

```
#define   II0        0x60
#define   IM0        0x61
#define   C0         0x62
#define   CP0        0x63
#define   GP0        0x64
#define   DB0        0x65
#define   DA0        0x66
```

```
#define   II1        0x68
#define   IM1        0x69
#define   C1         0x6A
#define   CP1        0x6B
#define   GP1        0x6C
#define   DB1        0x6D
#define   DA1        0x6E
```

```
#define   II2        0x70
#define   IM2        0x71
#define   C2         0x72
#define   CP2        0x73
#define   GP2        0x74
#define   DB2        0x75
#define   DA2        0x76
```

```
#define   II3        0x78
#define   IM3        0x79
```

```

#define      C3      0x7A

#define      CP3     0x7B

#define      GP3     0x7C

#define      DB3     0x7D

#define      DA3     0x7E


#define      II4     0x30

#define      IM4     0x31

#define      C4      0x32

#define      CP4     0x33

#define      GP4     0x34

#define      DB4     0x35

#define      DA4     0x36

#define      DMASTAT      0x37


#define      II5     0x38

#define      IM5     0x39

#define      C5      0x3A

#define      CP5     0x3B

#define      GP5     0x3C

#define      DB5     0x3D

#define      DA5     0x3E


#define      II6     0x40

#define      IM6     0x41

#define      C6      0x42

#define      CP6     0x43

#define      GP6     0x44

#define      EI6     0x45

#define      EM6     0x46

#define      EC6     0x47

```

```

#define      II7      0x48
#define      IM7      0x49
#define      C7       0x4A
#define      CP7      0x4B
#define      GP7      0x4C
#define      EI7      0x4D
#define      EM7      0x4E
#define      EC7      0x4F

```

```

#define      II8      0x50
#define      IM8      0x51
#define      C8       0x52
#define      CP8      0x53
#define      GP8      0x54
#define      EI8      0x55
#define      EM8      0x56
#define      EC8      0x57

```

```

#define      II9      0x58
#define      IM9      0x59
#define      C9       0x5A
#define      CP9      0x5B
#define      GP9      0x5C
#define      EI9      0x5D
#define      EM9      0x5E
#define      EC9      0x5F

```

```

/*----- DMA ADDRESS registers **** NEW Names **** -----*/

```

```

#define IIR0A      0x60
#define IMR0A      0x61
#define CR0A       0x62
#define CPR0A      0x63
#define GPR0A      0x64

#define IIR0B      0x30
#define IMR0B      0x31
#define CR0B       0x32
#define CPR0B      0x33
#define GPR0B      0x34

#define IIR1A      0x68
#define IMR1A      0x69
#define CR1A       0x6A
#define CPR1A      0x6B
#define GPR1A      0x6C

#define IIR1B      0x38
#define IMR1B      0x39
#define CR1B       0x3A
#define CPR1B      0x3B
#define GPR1B      0x3C

#define IIT0A      0x70
#define IMT0A      0x71
#define CT0A       0x72
#define CPT0A      0x73
#define GPT0A      0x74

#define IIT0B      0x50

```

```
#define IMT0B 0x51
#define CT0B 0x52
#define CPT0B 0x53
#define GPT0B 0x54
```

```
#define IIT1A 0x78
#define IMT1A 0x79
#define CT1A 0x7A
#define CPT1A 0x7B
#define GPT1A 0x7C
```

```
#define IIT1B 0x58
#define IMT1B 0x59
#define CT1B 0x5A
#define CPT1B 0x5B
#define GPT1B 0x5C
```

```
#define IIEP0 0x40
#define IMEP0 0x41
#define CEP0 0x42
#define CPEP0 0x43
#define GPEP0 0x44
#define EIEP0 0x45
#define EMEP0 0x46
#define ECEP0 0x47
```

```
#define IIEP1 0x48
#define IMEP1 0x49
#define CEP1 0x4A
```

```

#define      CPEP1      0x4B
#define      GPEP1      0x4C
#define      EIEP1      0x4D
#define      EMEP1      0x4E
#define      ECEP1      0x4F

```

```

/*----- MSP BOUNDS CHECKING registers -----*/

```

```

#define      DMAUB      0xa0
#define      DMALB      0xa1
#define      PMAUB      0xa2
#define      PMALB      0xa3
#define      IOAUB      0xa4
#define      IOALB      0xa5
#define      EPAUB      0xa6
#define      EPALB      0xa7

```

```

/*----- Emulation BOUNDS CHECKING registers -----*/

```

```

#define      IOAS       0xac
#define      IOAE       0xad
#define      EPAS       0xae
#define      EPAE       0xaf

```

```

/*----- LINK PORT registers -----*/

```

```

#define      LBUF0      0xc0
#define      LBUF1      0xc1
#define      LBUF2      0xc2

```

```

#define      LBUF3      0xc3
#define      LBUF4      0xc4
#define      LBUF5      0xc5
#define      LCTL       0xc6
#define      LCOM       0xc7
#define      LAR        0xc8
#define      LSRQ       0xc9
#define      LPATH1     0xca
#define      LPATH2     0xcb
#define      LPATH3     0xcc
#define      LPCNT      0xcd
#define      CNST1      0xce
#define      CNST2      0xcf

```

```

/*----- SPORT registers -----*/

```

```

#define      STCTL0     0xe0
#define      SRCTL0     0xe1
#define      TX0        0xe2
#define      RX0        0xe3
#define      TDIV0      0xe4
#define      TCNT0      0xe5
#define      RDIV0      0xe6
#define      RCNT0      0xe7
#define      MTCS0      0xe8
#define      MRCS0      0xe9
#define      MTCCS0     0xea
#define      MRCCS0     0xeb
#define      KEYWD0     0xec
#define      IMASK0     0xed

```

```

#define STCTL1      0xf0
#define SRCTL1      0xf1
#define TX1         0xf2
#define RX1         0xf3
#define TDIV1       0xf4
#define TCNT1       0xf5
#define RDIV1       0xf6
#define RCNT1       0xf7
#define MTCS1       0xf8
#define MRCS1       0xf9
#define MTCCS1      0xfa
#define MRCCS1      0xfb
#define KEYWD1      0xfc
#define IMASK1      0xfd

```

```

/* ----- Aliases for TX and Rx ----- */

```

```

#define TX0_A      0xe2
#define RX0_A      0xe3
#define TX1_A      0xf2
#define RX1_A      0xf3

```

```

/* ----- New regs for 21065L ----- */

```

```

#define TX0_B      0xee
#define RX0_B      0xef
#define TX1_B      0xfe
#define RX1_B      0xff

```

5 Βιβλιογραφία

Βιβλία

1. *Active Noise Control Systems, Algorithms and DSP implementations*, Sen Kuo/Dennis Morgan.
2. *Fundamentals of Adaptive Filtering*, Ali Said.
3. *Discrete Time Signal Processing*, Oppenheim/Schafer/Buck.
4. *Understanding Digital Signal Processing*, Lyons.
5. *The Theory of Matrices*, Sam Perlis.
6. *Ψηφιακή Επεξεργασία Σημάτων*, Καραγιάννης

Άρθρα

1. *Chirp design for acoustical system identification*, John Burgess, Acoustical Society of America, October 1991
2. *Leaky FXLMS algorithm: stochastic analysis for Gaussian data and Secondary Path modeling error*, Tobias/Seara , IEEE transactions on speech and audio processing, November 2005
3. *Limited Numerical Precision and the LMS algorithm – the Leaky LMS solution*, Hakansson, Blekinge Institute of Technology, Sweeden, February 2004
4. *Autocorrelation Matrix Eigenvalues and the Power Spectrum*, Messerschmitt, Electrical Engineering and Computer Sciences, University of California Berkeley, June 2006
5. *Linear Prediction : A Tutorial Review*, Makhoul, Proceedings of the IEEE, April 1975

Τεχνικά Εγχειρίδια

1. *ADSP-21065L- EZ-KIT LITE Evaluation System Manual*, Analog Devices
2. *ADSP-21065L SHARC DSP User's Manual*, Analog Devices
3. *ADSP-21065L SHARC DSP Technical Reference*, Analog Devices
4. *Interfacing the ADSP-21065L SHARC DSP to the AD1819a 'AC-97' SoundPort Codec*, Tomarakos/Analog Devices
5. *Using the Low-Cost, High-Performance ADSP-21065L Digital Signal Processor for Digital Audio Applications*, Tomarakos/Ledger

