



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Προσομοίωση ενός ψηφιακού συστήματος ήχου στο Simulink
Όνοματεπώνυμο Φοιτητή	Μπουραντάνης Σπυρίδων
Πατρώνυμο	Κωνσταντίνος
Αριθμός Μητρώου	ΜΠΠΑ/07036
Επιβλέπων	Πικράκης Άγγελος, Λέκτορας

Ημερομηνία Παράδοσης **Νοέμβριος 2011**

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ



Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

Πικράκης Άγγελος
Λέκτορας

(υπογραφή)

Ψαράκης Μιχαήλ
Επίκουρος Καθηγητής

(υπογραφή)

Κωνσταντόπουλος
Χαράλαμπος
Λέκτορας

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΑΙΑ

Περίληψη

Στον 21ο αιώνα, η Ψηφιακή Επεξεργασία Σήματος (DSP) βρίσκεται στον πυρήνα πολλών εφαρμογών, οι οποίες άμεσα ή έμμεσα στηρίζονται στην αναπαράσταση και επεξεργασία ψηφιακών δεδομένων. Ιδιαίτερα στην μουσική βιομηχανία, η ψηφιακή επεξεργασία σήματος συνιστά είδος επεξεργασίας. Όσον αφορά τα ηχητικά εφέ, όπως το chorus, το flanger, το delay, το distortion, το fuzz κλπ, επί δεκαετίες υλοποιούνταν με αναλογικά κυκλώματα. Όμως, εδώ και αρκετά χρόνια οι διαδεδομένες υλοποιήσεις είναι πλέον ψηφιακές. Η παρούσα μεταπτυχιακή διατριβή εμπίπτει στο πεδίο εφαρμογής της ψηφιακής επεξεργασίας σήματος και καλύπτει την επεξεργασία των ηχητικών σημάτων. Ασχολείται με την ανάλυση των μουσικών εφέ και τον σχεδιασμό των αλγορίθμων για την προσομοίωση αυτών.

Επομένως, στόχος της παρούσας διατριβής είναι η ανάπτυξη μιας εφαρμογής λογισμικού για την προσομοίωση ενός ψηφιακού συστήματος ήχου, το οποίο υλοποιείται στο λογισμικό Simulink-Matlab, και αποτελείται από έναν ενισχυτή, ένα γραφικό ισοσταθμιστή 10 περιοχών και τρία ψηφιακά εφέ ήχου, το delay, το reverb και το flanger. Η λειτουργικότητά του περιλαμβάνει τη δυνατότητα φόρτωσης ψηφιακών δειγμάτων ήχου και τον χειρισμό των δεδομένων εισόδου μέσω διαφόρων εφέ σε πραγματικό χρόνο. Τα χαρακτηριστικά του ηχητικού σήματος μπορούν να αλλάξουν προσαρμόζοντας ορισμένες από τις παραμέτρους των εφέ. Το σήμα που έχει περάσει από την επεξεργασία αυτή αποκτά κάποια ειδικά ηχητικά χαρακτηριστικά και είναι σε θέση να οδηγηθεί στο σύστημα ηχείων για την αναπαραγωγή.

Η αλληλεπίδραση μεταξύ του χρήστη και του προγράμματος υλοποιείται μέσα από μια γραφική διεπαφή χρήστη (GUI). Το GUI συνιστά προσπάθεια προσομοίωσης ψηφιακού συστήματος ήχου και επιτρέπει τη ρύθμιση παραμέτρων χρήσης. Για την υλοποίηση, επιλέχθηκε το λογισμικό Simulink-Matlab της MathWorks, Inc γιατί από τη φύση του διευκολύνει την προσομοίωση ποικιλίας συστημάτων, με την μορφή δομικών διαγραμμάτων και απαλλάσσει τον σχεδιαστή του συστήματος από προγραμματιστικές εργασίες χαμηλού επιπέδου.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

Abstract

In the 21st century, Digital Signal Processing (DSP) is at the core of most applications which either directly or indirectly rely on the representation and processing of digital data. In the music industry the Digital Signal Processor has become a very important type of processor. Some time ago, audio effects like chorus, flanger, delay, distortion, fuzz and many other effects used to be implemented on analog circuits, whereas it is now possible to do all the effective processing with DSP. With this kind of processing it is possible to be more flexible than with an analog circuit. This project is within the scope of DSP and covers the processing of audio signals. It deals with the analysis of music effects and the design of algorithms for simulating these effects.

The objective is the development of a software application for the simulation of a digital audio system that is implemented with Simulink-Matlab, consisting of an amplifier, a 10-band graphic equalizer and three digital audio effects, namely delay, reverb and flanger. Its functionality includes the ability to load digital audio samples and manipulate the data by passing the input through various effects in real time. The audio signal's characteristics can be changed by adjusting some of the effects parameters. The signal having passed through this processing acquires some specific sound characteristics and is able to lead the speaker system to reproduce.

The interaction between the user and the program is implemented through a Graphical User Interface (GUI). The GUI attempts to simulate similar professional digital audio systems. It includes features such as parameters which allow the user to perform fine-tuning operations. The development was carried out with Simulink-MATLAB because this particular development platform can be used to model and simulate a variety of systems in a high-level, block diagram format without the necessity of any low-level programming knowledge.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

Ευχαριστίες

Η διαδικασία συγγραφής της παρούσας διπλωματικής εργασίας αποτέλεσε για εμένα ευκαιρία να εμβαθύνω σε ένα νέο γνωστικό πεδίο, προσφέροντας μου ερεθίσματα για περαιτέρω διερεύνηση. Θα ήθελα να ευχαριστήσω όλους όσους συνεισέφεραν στην δημιουργία αυτής της εργασίας. Ιδιαίτερη αναφορά αξίζει στον καθηγητή και επιβλέποντά μου, κ. Άγγελο Πικράκη, για τον χρόνο αλλά και την τεχνογνωσία του, που με όρεξη, μου διέθεσε, αλλά και για τις πολύτιμες συμβουλές του περί ηχητικών θεμάτων. Τέλος, θα ήθελα να ευχαριστήσω τους φίλους και την οικογένεια μου για την υπομονή και τη στήριξη τους, οι οποίοι συνέβαλλαν καθοριστικά στην εκπόνηση της παρούσας εργασίας, καθ' όλη τη διάρκειά της.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΔΑΛΙΑΣ

“Music expresses that which cannot be said and on which it is impossible to be silent.”

Victor Hugo

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΔΑΛΙΑΣ

Πίνακας Περιεχομένων

Περίληψη	5
Abstract	7
Ευχαριστίες	9
Εισαγωγή.....	21
Σύντομη περιγραφή αντικειμένου	21
1 Γενικά για τον ήχο.....	25
1.1. Τι είναι Ήχος.....	25
1.2. Κατηγορίες ήχου.....	27
1.3. Χαρακτηριστικά του ήχου.....	28
1.3.1. Αντικειμενικά χαρακτηριστικά.....	28
1.3.2. Υποκειμενικά χαρακτηριστικά.....	29
1.4. Η μονάδα μέτρησης Decibel (dB).....	30
1.5. Αναλογικό και Ψηφιακό σήμα	31
1.6. Μετατροπή Αναλογικού Σήματος σε Ψηφιακό.....	31
1.6.1. Φιλτράρισμα	32
1.6.2. Δειγματοληψία (Sampling).....	32
1.6.3. Κβάντιση (Quantizing):.....	34
1.6.4. Κωδικοποίηση.....	35
2 Το αυτί και η αντίληψη της ακοής.....	37
2.1 Εισαγωγή.....	37
2.2 Η ανατομία του αυτιού.....	37
2.2.1 Το πτερύγιο: Κατευθυντικός κωδικοποιητής ήχου	37
2.2.2 Το κανάλι του αυτιού.....	38
2.2.3 Το μέσο αυτί.....	38
2.2.4 Το έσω αυτί.....	39
2.3 Περιοχή ακουστότητας	40
2.4 Σχέση χροιάς με το φάσμα	41
2.5 Εντοπισμός ηχητικών πηγών	41
2.6 Το φαινόμενο προβαδίσματος (φαινόμενο Haas).....	42
2.7 Αντίληψη του ήχου από ανάκλαση	43
3 Σειρές καθυστέρησης (Delay Lines).....	45
3.1 Εισαγωγή.....	45
3.2 Απλά Ψηφιακά Φίλτρα	45
3.3 Φίλτρα Κτένας (Comb Filters).....	47

3.3.1	Φίλτρα FIR Comb.....	47
3.3.2	Φίλτρα IIR Comb.....	49
3.4	Allpass Filters (Φίλτρα Γενικής Διέλευσης)	49
3.5	Κύκλωμα Ψηφιακής Σειράς Καθυστέρησης (Digital Delay Line DDL Circuit)	51
3.6	Φαινόμενα Σταθερού Χρόνου Καθυστέρησης.....	53
3.7	Φαινόμενα Μεταβλητού Χρόνου Καθυστέρησης.....	53
4	Η γλώσσα προγραμματισμού Matlab	55
4.1	Γενικά Στοιχεία	55
4.2	Ιστορικά Στοιχεία	55
4.3	Χρήσεις και δυνατότητες	56
4.4	Στοιχεία του Matlab	57
4.5	Η οργάνωση του Matlab.....	57
4.6	Λειτουργία του Matlab	58
4.7	Εισαγωγή στο Simulink	58
4.7.1	Τι είναι το Simulink	58
4.7.2	Ρυθμίσεις Παραμέτρων	60
4.7.3	Τρέχοντας την προσομοίωση.....	61
4.7.4	Οι βασικές βιβλιοθήκες δομικών στοιχείων του Simulink	61
4.8	Εισαγωγή στο GUIDE	63
4.8.1	Τι είναι το Guide	63
4.8.2	Αρχίζοντας το Guide	63
4.8.3	Περίληψη εργαλείων του Guide.....	63
4.8.4	Προσθήκη Components στο GUI	64
5	Ανάλυση μουσικών εφέ και σχεδιασμός αλγορίθμων.....	69
5.1	Εισαγωγή.....	69
5.2	Ενισχυτής ήχου(Amplifier).....	69
5.2.1	Τι είναι ο ενισχυτής ήχου	69
5.2.2	Χαρακτηριστικά μεγέθη ενισχυτών	70
5.2.3	Υλοποίηση στο Simulink	73
5.3	Ισοσταθμιστής (Equalizer).....	77
5.3.1	Τι είναι η ισοστάθμιση συχνοτήτων	77
5.3.2	Σχεδιασμός του ισοσταθμιστή	77
5.3.3	Έλεγχος Εύρους Ζώνης Q (Bandwidth Control Q)	78
5.3.4	Είδη Ισοσταθμιστών	79
5.3.5	Τα Φίλτρα.....	81

5.3.6	Κατηγορίες φίλτρων	82
5.3.7	Υλοποίηση στο Simulink	86
5.4	Delay effect.....	94
5.4.1	Γενικά στοιχεία	94
5.4.2	Σχεδιασμός του Delay effect	97
5.4.3	Υλοποίηση στο Simulink	98
5.5	Reverb effect.....	101
5.5.1	Γενικά στοιχεία	101
5.5.2	Αναλογικοί Μέθοδοι Αντήχησης	102
5.5.3	Ψηφιακοί Αλγόριθμοι Αντήχησης.....	103
5.5.4	Υλοποίηση στο Simulink	109
5.6	Flanger effect.....	111
5.6.1	Γενικά στοιχεία	111
5.6.2	Υλοποίηση στο Simulink	112
5.7	Συνδυασμός των εφέ.....	116
	Συμπεράσματα	121
	Βιβλιογραφία	123
	Παράρτημα A: Εγχειρίδιο χρήσης του προγράμματος	125
	Παράρτημα B: Κώδικας του προγράμματος στο Matlab.....	133

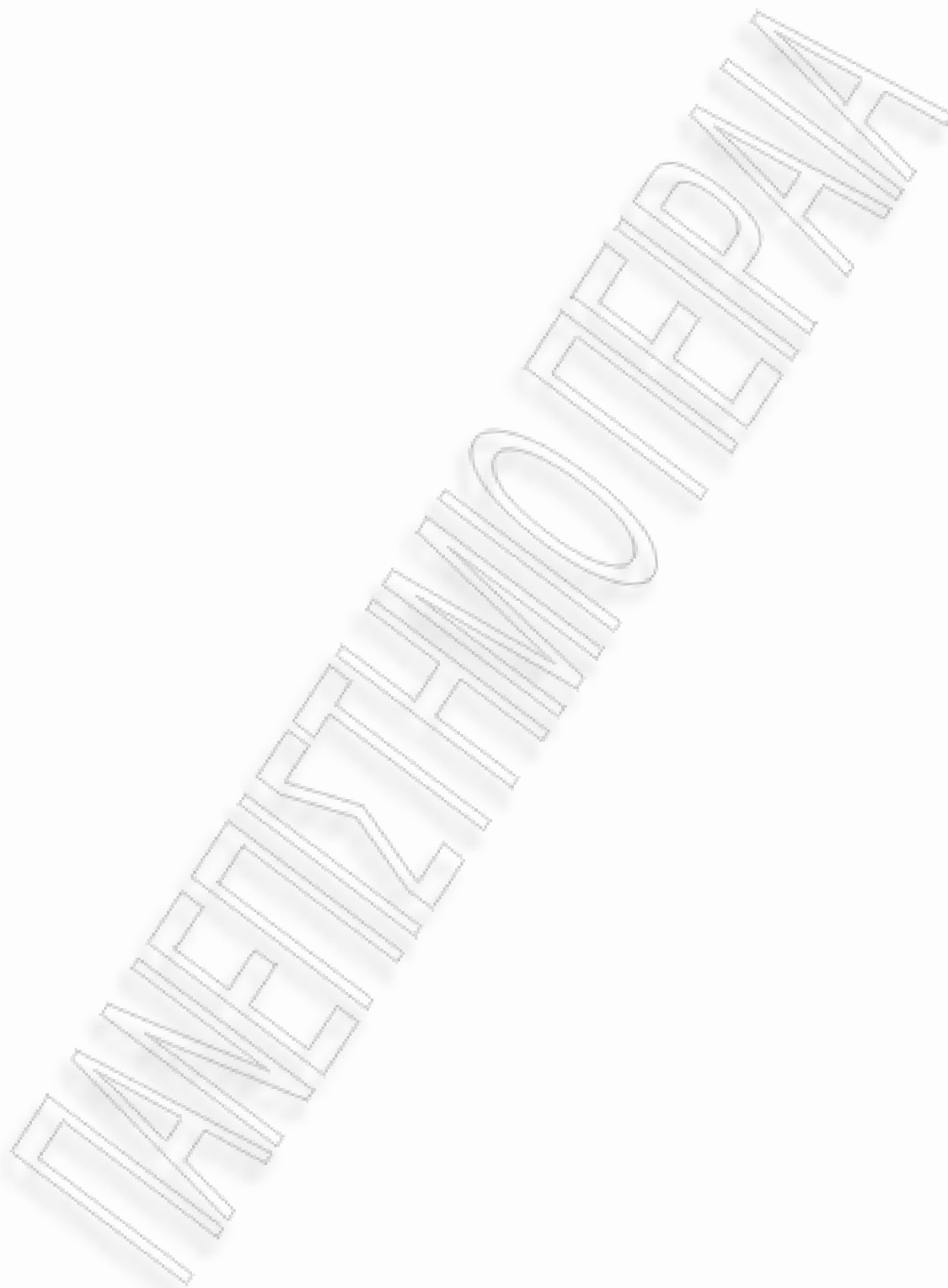
Ευρετήριο Εικόνων

ΕΙΚΟΝΑ 1.1: ΕΓΚΑΡΣΙΑ ΚΑΙ ΔΙΑΜΗΚΗ ΚΥΜΑΤΑ	26
ΕΙΚΟΝΑ 1.2: ΑΡΑΙΩΣΕΙΣ ΚΑΙ ΠΥΚΝΩΣΕΙΣ ΚΑΤΑ ΤΗ ΔΙΑΔΟΣΗ ΕΝΟΣ ΚΥΜΑΤΟΣ	27
ΕΙΚΟΝΑ 1.3: ΜΗΚΟΣ ΚΥΜΑΤΟΣ	27
ΕΙΚΟΝΑ 1.4: ΉΧΟΙ ΜΙΚΡΗΣ ΚΑΙ ΜΕΓΑΛΗΣ ΣΥΧΝΟΤΗΤΑΣ.....	30
ΕΙΚΟΝΑ 1.5: ΑΝΑΛΟΓΙΚΟ ΣΗΜΑ.....	31
ΕΙΚΟΝΑ 1.6: ΨΗΦΙΑΚΟ ΣΗΜΑ.....	31
ΕΙΚΟΝΑ 1.7: ΣΤΑΔΙΑ ΕΠΕΞΕΡΓΑΣΙΑΣ ΑΝΑΛΟΓΙΚΟΥ ΣΗΜΑΤΟΣ ΚΑΤΑ ΤΗ ΜΕΤΑΤΡΟΠΗ ΤΟΥ ΣΕ ΨΗΦΙΑΚΟ	32
ΕΙΚΟΝΑ 1.8: ΦΙΛΤΡΟ ΥΨΗΛΗΣ ΚΑΙ ΧΑΜΗΛΗΣ ΔΙΕΛΕΥΣΗΣ.....	32
ΕΙΚΟΝΑ 1.9: ΤΟ ΑΡΧΙΚΟ ΑΝΑΛΟΓΙΚΟ ΣΗΜΑ	33
ΕΙΚΟΝΑ 1.10: ΔΕΙΓΜΑΤΟΛΗΠΤΗΜΕΝΟ ΣΗΜΑ ΜΕ ΥΨΗΛΟ ΡΥΘΜΟ-ΣΥΧΝΟΤΗΤΑ ΔΕΙΓΜΑΤΟΛΗΨΙΑΣ.....	33
ΕΙΚΟΝΑ 1.11: ΔΕΙΓΜΑΤΟΛΗΠΤΗΜΕΝΟ ΣΗΜΑ ΜΕ ΧΑΜΗΛΟ ΡΥΘΜΟ-ΣΥΧΝΟΤΗΤΑ ΔΕΙΓΜΑΤΟΛΗΨΙΑΣ.....	33
ΕΙΚΟΝΑ 1.12: ΚΒΑΝΤΟΠΟΙΗΣΗ ΑΝΑΛΟΓΙΚΟΥ ΣΗΜΑΤΟΣ	34
ΕΙΚΟΝΑ 1.13: ΚΑΛΗ ΚΑΙ ΚΑΚΗ ΔΕΙΓΜΑΤΟΛΗΨΙΑ.....	35
ΕΙΚΟΝΑ 2.1: ΤΑ ΤΕΣΣΕΡΑ ΚΥΡΙΑ ΤΜΗΜΑΤΑ ΤΟΥ ΑΝΘΡΩΠΙΝΟΥ ΑΥΤΙΟΥ: ΤΟ ΠΤΕΡΥΓΙΟ, ΤΟ ΑΚΟΥΣΤΙΚΟ ΚΑΝΑΛΙ, ΤΟ ΜΕΣΟ ΑΥΤΙ ΚΑΙ ΤΟ ΕΣΩ ΑΥΤΙ.....	37
ΕΙΚΟΝΑ 2.2: ΤΟ ΑΚΟΥΣΤΙΚΟ ΚΑΝΑΛΙ ΤΟΥ ΑΥΤΙΟΥ	38
ΕΙΚΟΝΑ 2.3: ΤΑ ΜΙΚΡΑ ΟΣΤΑ (ΣΦΥΡΑ, ΑΚΜΟΝΑΣ ΚΑΙ ΑΝΑΒΟΛΕΑΣ) ΤΟΥ ΜΕΣΟΥ ΑΥΤΙΟΥ.....	39
ΕΙΚΟΝΑ 2.4: ΣΧΗΜΑΤΙΚΟ ΔΙΑΓΡΑΜΜΑ ΤΟΥ ΑΥΤΙΟΥ.....	39
ΕΙΚΟΝΑ 2.5: Η ΠΕΡΙΟΧΗ ΑΚΟΥΣΤΟΤΗΤΑΣ ΤΟΥ ΑΝΘΡΩΠΙΝΟΥ ΑΥΤΙΟΥ	41
ΕΙΚΟΝΑ 2.6: ΤΟ ΜΕΤΩΠΟ ΚΥΜΑΤΟΣ ΕΝΟΣ ΗΧΟΥ ΓΙΑ ΤΟΝ ΕΝΤΟΠΙΣΜΟ ΗΧΗΤΙΚΩΝ ΠΗΓΩΝ	42
ΕΙΚΟΝΑ 2.7: ΤΟ ΦΑΙΝΟΜΕΝΟ ΠΡΟΒΑΔΙΣΜΑΤΟΣ (ΦΑΙΝΟΜΕΝΟ ΗΑΑΣ) ΣΤΟ ΑΝΘΡΩΠΙΝΟ ΑΚΟΥΣΤΙΚΟ ΣΥΣΤΗΜΑ	43
ΕΙΚΟΝΑ 2.8: ΟΙ ΕΠΙΔΡΑΣΕΙΣ ΤΩΝ ΠΛΕΥΡΙΚΩΝ ΑΝΑΚΛΑΣΕΩΝ ΣΤΗΝ ΑΝΤΙΛΗΨΗ ΤΟΥ ΑΜΕΣΟΥ ΗΧΟΥ ΣΕ ΠΡΟΣΟΜΟΙΩΣΗ ΔΙΑΤΑΞΗΣ ΣΤΕΡΕΟ	44
ΕΙΚΟΝΑ 3.1: ΈΝΑ ΑΠΛΟ ΨΗΦΙΑΚΟ ΦΙΛΤΡΟ ΧΑΜΗΛΗΣ ΔΙΕΛΕΥΣΗΣ (LOW-PASS)	45
ΕΙΚΟΝΑ 3.2: ΣΥΧΝΟΤΙΚΗ ΑΠΟΚΡΙΣΗ ΦΙΛΤΡΟΥ ΧΑΜΗΛΗΣ ΔΙΕΛΕΥΣΗΣ ΓΙΑ $A_0 = A_1 = 0.5$	46
ΕΙΚΟΝΑ 3.3: ΈΝΑ ΑΠΛΟ ΨΗΦΙΑΚΟ ΦΙΛΤΡΟ ΥΨΗΛΗΣ ΔΙΕΛΕΥΣΗΣ (HIGH PASS)	46
ΕΙΚΟΝΑ 3.4: ΣΥΧΝΟΤΙΚΗ ΑΠΟΚΡΙΣΗ ΦΙΛΤΡΟΥ ΥΨΗΛΗΣ ΔΙΕΛΕΥΣΗΣ ΓΙΑ $A_0 = A_1 = 0.546$	
ΕΙΚΟΝΑ 3.5: ΈΝΑ FIR ΦΙΛΤΡΟ ΚΤΕΝΑΣ.....	47
ΕΙΚΟΝΑ 3.6: ΣΥΧΝΟΤΙΚΗ ΑΠΟΚΡΙΣΗ FIR ΦΙΛΤΡΟΥ ΚΤΕΝΑΣ ΜΕ ΧΡΟΝΟ ΚΑΘΥΣΤΕΡΗΣΗΣ = 0.25 MS (F = 4 ΚΗΖ).....	48
ΕΙΚΟΝΑ 3.7: ΈΝΑ IIR ΦΙΛΤΡΟ ΚΤΕΝΑΣ	49
ΕΙΚΟΝΑ 3.8: ΣΥΧΝΟΤΙΚΗ ΑΠΟΚΡΙΣΗ IIR ΦΙΛΤΡΟΥ ΚΤΕΝΑΣ	49
ΕΙΚΟΝΑ 3.9: ΔΙΑΓΡΑΜΜΑ ΦΙΛΤΡΟΥ ΓΕΝΙΚΗΣ ΔΙΕΛΕΥΣΗΣ.....	50
ΕΙΚΟΝΑ 3.10: ΑΠΟΚΡΙΣΗ ΦΑΣΗΣ ΕΝΟΣ ΦΙΛΤΡΟΥ ΓΕΝΙΚΗΣ ΔΙΕΛΕΥΣΗΣ ΣΕ ΣΧΕΣΗ ΜΕ ΤΟΝ ΧΡΟΝΟ ΚΑΘΥΣΤΕΡΗΣΗΣ	50
ΕΙΚΟΝΑ 3.11: ΦΙΛΤΡΟ ΓΕΝΙΚΗΣ ΔΙΕΛΕΥΣΗΣ.....	51
ΕΙΚΟΝΑ 3.12: ΚΥΚΛΩΜΑ ΨΗΦΙΑΚΗΣ ΣΕΙΡΑΣ ΚΑΘΥΣΤΕΡΗΣΗΣ	51
ΕΙΚΟΝΑ 4.1: ΛΙΓΑ ΛΟΓΙΑ ΓΙΑ ΤΟ MATLAB	56
ΕΙΚΟΝΑ 4.2: ΒΙΒΛΙΟΘΗΚΗ ΤΟΥ SIMULINK	59
ΕΙΚΟΝΑ 4.3: ΤΟ ΠΕΡΙΒΑΛΛΟΝ ΤΟΥ SIMULINK	59
ΕΙΚΟΝΑ 4.4: ΕΙΣΑΓΩΓΗ BLOCK ΣΤΟ SIMULINK	60

ΕΙΚΟΝΑ 4.5: ΠΑΡΑΘΥΡΟ ΡΥΘΜΙΣΕΩΝ ΠΑΡΑΜΕΤΡΩΝ ΤΟΥ SIMULINK	60
ΕΙΚΟΝΑ 4.6: ΤΟ ΠΛΑΙΣΙΟ ΔΙΑΛΟΓΟΥ ΕΝΑΡΞΗΣ ΤΟΥ GUIDE.....	63
ΕΙΚΟΝΑ 4.7: ΚΕΝΟ ΠΡΟΤΥΠΟ GUI ΚΑΙ ΠΑΡΟΥΣΙΑΣΗ ΤΩΝ ΕΡΓΑΛΕΙΩΝ ΤΟΥ	64
ΕΙΚΟΝΑ 4.8: ΤΑ COMPONENTS ΠΟΥ ΕΧΕΙ ΤΟ GUI.....	65
ΕΙΚΟΝΑ 4.9: PROPERTY INSPECTOR.....	65
ΕΙΚΟΝΑ 5.1: ΜΠΛΟΚ ΔΙΑΓΡΑΜΜΑ (Α) ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΟ ΚΥΚΛΩΜΑ (Β) ΕΝΟΣ ΕΝΙΣΧΥΤΗ ΗΧΟΥ	70
ΕΙΚΟΝΑ 5.2: ΠΑΡΑΜΟΡΦΩΣΗ ΣΥΧΝΟΤΗΤΩΝ	71
ΕΙΚΟΝΑ 5.3: ΠΑΡΑΜΟΡΦΩΣΗ ΦΑΣΗΣ	71
ΕΙΚΟΝΑ 5.4: ΠΑΡΑΜΟΡΦΩΣΗ ΠΛΑΤΟΥΣ	71
ΕΙΚΟΝΑ 5.5: ΚΑΜΠΥΛΗ ΑΠΟΚΡΙΣΗΣ ΣΥΧΝΟΤΗΤΩΝ ΕΝΟΣ ΕΝΙΣΧΥΤΗ	72
ΕΙΚΟΝΑ 5.6: ΚΑΤΑΜΕΡΙΣΜΟΣ ΘΕΡΜΙΚΩΝ ΑΠΩΛΕΙΩΝ ΣΕ ΕΝΑ ΤΥΠΙΚΟ ΕΝΙΣΧΥΤΗ	72
ΕΙΚΟΝΑ 5.7: ΜΟΝΤΕΛΟ ΤΟΥ ΕΝΙΣΧΥΤΗ.....	73
ΕΙΚΟΝΑ 5.8: ΠΑΡΑΜΕΤΡΟΙ ΤΟΥ ΜΠΛΟΚ FROM WAVE FILE	74
ΕΙΚΟΝΑ 5.9: ΠΑΡΑΜΕΤΡΟΙ ΤΟΥ ΜΠΛΟΚ SLIDER GAIN.....	74
ΕΙΚΟΝΑ 5.10: ΥΠΟΣΥΣΤΗΜΑ ΤΟΥ BALANCE	75
ΕΙΚΟΝΑ 5.11: ΠΑΡΑΜΕΤΡΟΙ ΤΟΥ ΜΠΛΟΚ SUBMATRIX.....	75
ΕΙΚΟΝΑ 5.12: ΠΑΡΑΜΕΤΡΟΙ ΤΟΥ ΜΠΛΟΚ DUPLICATE MONO CHANNEL	76
ΕΙΚΟΝΑ 5.13: ΜΟΝΤΕΛΟ ΤΟΥ SURROUND EFFECT	76
ΕΙΚΟΝΑ 5.14: ΥΠΟΣΥΣΤΗΜΑ ΤΟΥ SURROUND EFFECT.....	77
ΕΙΚΟΝΑ 5.15: Ο ΟΡΙΣΜΟΣ ΤΟΥ Q.....	78
ΕΙΚΟΝΑ 5.16: ΜΕΓΑΛΕΣ ΤΙΜΕΣ ΣΤΟ Q, ΜΙΚΡΟΤΕΡΟ ΕΥΡΟΣ ΣΥΧΝΟΤΗΤΩΝ	78
ΕΙΚΟΝΑ 5.17: ΜΙΚΡΕΣ ΤΙΜΕΣ ΣΤΟ Q, ΜΕΓΑΛΥΤΕΡΟ ΕΥΡΟΣ ΣΥΧΝΟΤΗΤΩΝ.....	79
ΕΙΚΟΝΑ 5.18: ΓΡΑΦΙΚΟΣ ΙΣΟΣΤΑΘΜΙΣΤΗΣ	79
ΕΙΚΟΝΑ 5.19: ΠΑΡΑΜΕΤΡΙΚΟΣ ΙΣΟΣΤΑΘΜΙΣΤΗΣ.....	80
ΕΙΚΟΝΑ 5.20: SOFTWARE LINEAR PHASE ΙΣΟΣΤΑΘΜΙΣΤΗΣ	81
ΕΙΚΟΝΑ 5.21: HARDWARE LINEAR PHASE ΙΣΟΣΤΑΘΜΙΣΤΗΣ.....	81
ΕΙΚΟΝΑ 5.22: ΒΑΘΥΠΕΡΑΤΟ ΦΙΛΤΡΟ, 1ΚΗΖ	83
ΕΙΚΟΝΑ 5.23: ΥΨΙΠΕΡΑΤΟ ΦΙΛΤΡΟ, 1ΚΗΖ.....	83
ΕΙΚΟΝΑ 5.24: ΖΩΝΟΠΕΡΑΤΟ ΦΙΛΤΡΟ	84
ΕΙΚΟΝΑ 5.25: ΦΙΛΤΡΑ ΤΥΠΟΥ SHELVING	85
ΕΙΚΟΝΑ 5.26: HIGH SHELVING FILTER, 1ΚΗΖ.....	85
ΕΙΚΟΝΑ 5.27: LOW SHELVING FILTER, 1ΚΗΖ	85
ΕΙΚΟΝΑ 5.28: PEAKING BAND - ΜΕΓΑΛΟ BANDWIDTH ΜΕ ΧΑΜΗΛΟ Q	86
ΕΙΚΟΝΑ 5.29: PEAKING BAND - ΜΙΚΡΟ BANDWIDTH ΜΕ ΧΑΜΗΛΟ Q	86
ΕΙΚΟΝΑ 5.30: ΜΟΝΤΕΛΟ ΤΟΥ ΙΣΟΣΤΑΘΜΙΣΤΗ.....	86
ΕΙΚΟΝΑ 5.31: ΥΠΟΣΥΣΤΗΜΑ ΤΟΥ EQUALIZER.....	87
ΕΙΚΟΝΑ 5.32: ΠΑΡΑΜΕΤΡΟΙ ΤΟΥ ΜΠΛΟΚ DIGITAL FILTER DESIGN	88
ΕΙΚΟΝΑ 5.33: ΠΡΟΔΙΑΓΡΑΦΕΣ ΤΩΝ ΣΥΧΝΟΤΗΤΩΝ	88
ΕΙΚΟΝΑ 5.34: ΜΕΓΕΘΟΣ ΑΠΟΚΡΙΣΗΣ ΣΕ DB ΚΑΙ ΑΠΟΚΡΙΣΗ ΦΑΣΗΣ ΓΙΑ ΤΗ ΖΩΝΗ ΔΙΕΛΕΥΣΗΣ ΣΥΧΝΟΤΗΤΩΝ 20-32ΗΖ.....	89
ΕΙΚΟΝΑ 5.35: ΜΕΓΕΘΟΣ ΑΠΟΚΡΙΣΗΣ ΣΕ DB ΚΑΙ ΑΠΟΚΡΙΣΗ ΦΑΣΗΣ ΓΙΑ ΤΗ ΖΩΝΗ ΔΙΕΛΕΥΣΗΣ ΣΥΧΝΟΤΗΤΩΝ 64-125ΗΖ.....	89
ΕΙΚΟΝΑ 5.36: ΜΕΓΕΘΟΣ ΑΠΟΚΡΙΣΗΣ ΣΕ DB ΚΑΙ ΑΠΟΚΡΙΣΗ ΦΑΣΗΣ ΓΙΑ ΤΗ ΖΩΝΗ ΔΙΕΛΕΥΣΗΣ ΣΥΧΝΟΤΗΤΩΝ 125-250ΗΖ.....	90
ΕΙΚΟΝΑ 5.37: ΜΕΓΕΘΟΣ ΑΠΟΚΡΙΣΗΣ ΣΕ DB ΚΑΙ ΑΠΟΚΡΙΣΗ ΦΑΣΗΣ ΓΙΑ ΤΗ ΖΩΝΗ ΔΙΕΛΕΥΣΗΣ ΣΥΧΝΟΤΗΤΩΝ 250-500ΗΖ.....	90
ΕΙΚΟΝΑ 5.38: ΜΕΓΕΘΟΣ ΑΠΟΚΡΙΣΗΣ ΣΕ DB ΚΑΙ ΑΠΟΚΡΙΣΗ ΦΑΣΗΣ ΓΙΑ ΤΗ ΖΩΝΗ ΔΙΕΛΕΥΣΗΣ ΣΥΧΝΟΤΗΤΩΝ 500-1000ΗΖ	91

ΕΙΚΟΝΑ 5.39: ΜΕΓΕΘΟΣ ΑΠΟΚΡΙΣΗΣ ΣΕ DB ΚΑΙ ΑΠΟΚΡΙΣΗ ΦΑΣΗΣ ΓΙΑ ΤΗ ΖΩΝΗ ΔΙΕΛΕΥΣΗΣ ΣΥΧΝΟΤΗΤΩΝ 1-2ΚΗΖ.....	91
ΕΙΚΟΝΑ 5.40: ΑΠΟΚΡΙΣΗΣ ΣΕ DB ΚΑΙ ΑΠΟΚΡΙΣΗ ΦΑΣΗΣ ΓΙΑ ΤΗ ΖΩΝΗ ΔΙΕΛΕΥΣΗΣ ΣΥΧΝΟΤΗΤΩΝ 2-4ΚΗΖ.....	92
ΕΙΚΟΝΑ 5.41: ΜΕΓΕΘΟΣ ΑΠΟΚΡΙΣΗΣ ΣΕ DB ΚΑΙ ΑΠΟΚΡΙΣΗ ΦΑΣΗΣ ΓΙΑ ΤΗ ΖΩΝΗ ΔΙΕΛΕΥΣΗΣ ΣΥΧΝΟΤΗΤΩΝ 4-8ΚΗΖ.....	92
ΕΙΚΟΝΑ 5.42: ΜΕΓΕΘΟΣ ΑΠΟΚΡΙΣΗΣ ΣΕ DB ΚΑΙ ΑΠΟΚΡΙΣΗ ΦΑΣΗΣ ΓΙΑ ΤΗ ΖΩΝΗ ΔΙΕΛΕΥΣΗΣ ΣΥΧΝΟΤΗΤΩΝ 8-16ΚΗΖ.....	93
ΕΙΚΟΝΑ 5.43: ΜΕΓΕΘΟΣ ΑΠΟΚΡΙΣΗΣ ΣΕ DB ΚΑΙ ΑΠΟΚΡΙΣΗ ΦΑΣΗΣ ΓΙΑ ΤΗ ΖΩΝΗ ΔΙΕΛΕΥΣΗΣ ΣΥΧΝΟΤΗΤΩΝ 16-20ΚΗΖ.....	93
ΕΙΚΟΝΑ 5.44: ΠΑΡΑΜΕΤΡΟΙ ΤΟΥ ΜΠΛΟΚ DB GAIN.....	94
ΕΙΚΟΝΑ 5.45: ΠΑΡΑΜΕΤΡΟΙ ΤΟΥ ΜΠΛΟΚ SUM.....	94
ΕΙΚΟΝΑ 5.46: ΔΙΑΓΡΑΜΜΑ ΕΝΟΣ FIR ΦΙΛΤΡΟΥ ΚΤΕΝΑΣ.....	95
ΕΙΚΟΝΑ 5.47: ΣΥΝΑΡΤΗΣΗ ΜΕΤΑΦΟΡΑΣ ΤΟΥ FIR ΦΙΛΤΡΟΥ ΚΤΕΝΑΣ ΓΙΑ ΧΡΟΝΙΚΗ ΚΑΘΥΣΤΕΡΗΣΗ 8 ΔΕΙΓΜΑΤΩΝ ΚΑΙ ΓΙΑ MIX = 0,7.....	95
ΕΙΚΟΝΑ 5.48: ΔΙΑΓΡΑΜΜΑ ΕΝΟΣ IIR ΦΙΛΤΡΟΥ ΚΤΕΝΑΣ.....	96
ΕΙΚΟΝΑ 5.49: ΔΙΑΓΡΑΜΜΑ ΕΝΟΣ ΔΙΑΦΟΡΕΤΙΚΟΥ IIR ΦΙΛΤΡΟΥ ΚΤΕΝΑΣ.....	96
ΕΙΚΟΝΑ 5.50: ΔΙΑΓΡΑΜΜΑ ΕΝΟΣ IIR ΦΙΛΤΡΟΥ ΚΤΕΝΑΣ ΓΙΑ ΤΗΝ ΚΑΤΑΣΚΕΥΗ ΤΟΥ DELAY ΕΦΕ.....	97
ΕΙΚΟΝΑ 5.51: ΣΥΝΑΡΤΗΣΗ ΜΕΤΑΦΟΡΑΣ ΤΟΥ IIR ΦΙΛΤΡΟΥ ΚΤΕΝΑΣ ΓΙΑ ΧΡΟΝΙΚΗ ΚΑΘΥΣΤΕΡΗΣΗ 8 ΔΕΙΓΜΑΤΩΝ ΚΑΙ ΓΙΑ G = 0,7.....	98
ΕΙΚΟΝΑ 5.52: ΜΟΝΤΕΛΟ ΤΟΥ DELAY EFFECT.....	99
ΕΙΚΟΝΑ 5.53: ΥΠΟΣΥΣΤΗΜΑ ΤΟΥ DELAY EFFECT.....	99
ΕΙΚΟΝΑ 5.54: ΠΑΡΑΜΕΤΡΟΙ ΤΟΥ ΜΠΛΟΚ INTEGER DELAY.....	100
ΕΙΚΟΝΑ 5.55: ΠΑΡΑΜΕΤΡΟΙ ΤΟΥ ΜΠΛΟΚ GAIN.....	100
ΕΙΚΟΝΑ 5.56: ΑΝΤΗΧΗΣΗ ΜΙΑΣ ΠΗΓΗΣ ΣΕ ΕΝΑΝ ΠΕΡΙΒΑΛΛΟΝΤΑ ΧΩΡΟ.....	101
ΕΙΚΟΝΑ 5.57: ΔΙΑΓΡΑΜΜΑ RT60.....	102
ΕΙΚΟΝΑ 5.58: ΕΠΕΞΕΡΓΑΣΙΑ ΗΧΟΥ ΣΕ ΘΑΛΑΜΟ ΑΝΤΗΧΗΣΗΣ.....	103
ΕΙΚΟΝΑ 5.59: ΑΝΤΗΧΗΣΗ ΜΕ ΕΝΑ IIR ΦΙΛΤΡΟ ΚΤΕΝΑΣ.....	104
ΕΙΚΟΝΑ 5.60: Ο ΧΡΟΝΟΣ ΚΑΘΥΣΤΕΡΗΣΗΣ D.....	104
ΕΙΚΟΝΑ 5.61: ΑΝΤΗΧΗΣΗ ΜΕ ΕΝΑ ΦΙΛΤΡΟ ΓΕΝΙΚΗΣ ΔΙΕΛΕΥΣΗΣ.....	105
ΕΙΚΟΝΑ 5.62: Ο ΧΡΟΝΟΣ ΚΑΘΥΣΤΕΡΗΣΗΣ D.....	106
ΕΙΚΟΝΑ 5.63: ΤΕΣΣΕΡΑ ΠΑΡΑΛΛΗΛΑ ΦΙΛΤΡΑ ΚΤΕΝΑΣ.....	106
ΕΙΚΟΝΑ 5.64: ΤΕΣΣΕΡΑ ΦΙΛΤΡΑ ΓΕΝΙΚΗΣ ΔΙΕΛΕΥΣΗΣ ΕΝ ΣΕΙΡΑ.....	107
ΕΙΚΟΝΑ 5.65: ΔΟΜΗ ΑΝΤΗΧΗΣΗΣ SCHROEDER.....	108
ΕΙΚΟΝΑ 5.66: ΠΡΟΠΟΡΕΥΟΜΕΝΕΣ ΑΝΑΚΛΑΣΕΙΣ ΚΑΙ ΔΙΑΧΕΟΥΣΑ ΑΝΤΗΧΗΣΗ.....	109
ΕΙΚΟΝΑ 5.67: ΜΟΝΤΕΛΟ ΤΟΥ REVERB EFFECT.....	109
ΕΙΚΟΝΑ 5.68: ΥΠΟΣΥΣΤΗΜΑ ΤΟΥ REVERB EFFECT.....	110
ΕΙΚΟΝΑ 5.69: ΥΠΟΣΥΣΤΗΜΑ ΤΩΝ ΦΙΛΤΡΩΝ ΚΤΕΝΑΣ.....	110
ΕΙΚΟΝΑ 5.70: ΥΠΟΣΥΣΤΗΜΑ ΤΩΝ ΦΙΛΤΡΩΝ ΓΕΝΙΚΗΣ ΔΙΕΛΕΥΣΗΣ.....	111
ΕΙΚΟΝΑ 5.71: ΔΙΑΓΡΑΜΜΑ ΤΟΥ FLANGER EFFECT.....	111
ΕΙΚΟΝΑ 5.72: ΜΕΓΕΘΟΣ ΣΥΧΝΟΤΙΚΗΣ ΑΠΟΚΡΙΣΗΣ ΓΙΑ ΔΤ = 1MS.....	112
ΕΙΚΟΝΑ 5.73: ΜΟΝΤΕΛΟ ΤΟΥ FLANGER EFFECT.....	113
ΕΙΚΟΝΑ 5.74: ΥΠΟΣΥΣΤΗΜΑ ΤΟΥ FLANGER EFFECT.....	113
ΕΙΚΟΝΑ 5.75: ΠΑΡΑΜΕΤΡΟΙ ΤΟΥ ΜΠΛΟΚ LOW FREQUENCY OSCILLATOR.....	114
ΕΙΚΟΝΑ 5.76: ΠΑΡΑΜΕΤΡΟΙ ΤΟΥ ΜΠΛΟΚ DSP CONSTANT.....	115
ΕΙΚΟΝΑ 5.77: ΠΑΡΑΜΕΤΡΟΙ ΤΟΥ ΜΠΛΟΚ VARIABLE DELAY UNIT.....	115
ΕΙΚΟΝΑ 5.78: ΔΟΜΙΚΟ ΔΙΑΓΡΑΜΜΑ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΤΟ SIMULINK.....	117
ΕΙΚΟΝΑ 5.79: ΤΟ ΣΧΕΔΙΑΣΤΙΚΟ ΜΕΡΟΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΤΟ GUIDE ΤΟΥ MATLAB.....	118

ΕΙΚΟΝΑ 5.80: ΓΡΑΦΙΚΗ ΔΙΕΠΑΦΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ..... 119



Ευρετήριο Πινάκων

ΠΙΝΑΚΑΣ 1.1: ΤΑΧΥΤΗΤΑ ΔΙΑΔΟΣΗΣ ΤΟΥ ΗΧΟΥ ΣΕ ΔΙΑΦΟΡΑ ΥΛΙΚΑ	26
ΠΙΝΑΚΑΣ 1.2: ΚΩΔΙΚΟΠΟΙΗΣΗ ΤΙΜΩΝ ΔΕΙΓΜΑΤΟΣ	36
ΠΙΝΑΚΑΣ 3.1: ΣΧΕΣΗ ΧΡΟΝΟΥ ΚΑΘΥΣΤΕΡΗΣΗΣ ΚΑΙ ΣΥΧΝΟΤΙΚΗΣ ΑΠΟΚΡΙΣΗΣ ΕΝΟΣ FIR ΦΙΛΤΡΟΥ ΚΤΕΝΑΣ	48
ΠΙΝΑΚΑΣ 4.1: ΟΙ ΒΑΣΙΚΕΣ ΒΙΒΛΙΟΘΗΚΕΣ ΔΟΜΙΚΩΝ ΣΤΟΙΧΕΙΩΝ ΤΟΥ SIMULINK.....	62
ΠΙΝΑΚΑΣ 5.1: ΣΥΝΤΕΛΕΣΤΗΣ ΕΝΤΑΣΗΣ G ΚΑΙ ΧΡΟΝΟΣ ΑΠΟΣΒΕΣΗΣ	104
ΠΙΝΑΚΑΣ 5.2: Η ΣΧΕΤΙΚΗ ΕΝΤΑΣΗ A ΣΕ DB ΣΤΟ ΑΝΤΙΣΤΟΙΧΟ ΕΙΔΩΛΟ N, ΩΣ ΑΚΕΡΑΙΟ ΠΟΛΛΑΠΛΑΣΙΟ ΤΟΥ ΧΡΟΝΟΥ ΚΑΘΥΣΤΕΡΗΣΗΣ D	105

Εισαγωγή

Τα τελευταία χρόνια παρουσιάζεται ραγδαία ανάπτυξη στον τομέα της επιστήμης των ηλεκτρονικών υπολογιστών και γενικά στον κόσμο της μικροηλεκτρονικής. Αυτό είχε καθοριστική επίδραση στην ψηφιακή επεξεργασία σημάτων. Η επεξεργασία σήματος αποτελεί ένα σύγχρονο τομέα των ηλεκτρονικών. Οι τεχνικές της ψηφιακής επεξεργασίας σήματος χρησιμοποιούνται σε πολλούς τομείς της τεχνολογίας και της επιστήμης. Για παράδειγμα εφαρμόζονται στις ψηφιακές τηλεπικοινωνίες, στη βιοϊατρική τεχνολογία, στη σεισμολογία, στη αεροναυτική, στην επεξεργασία ομιλίας και άλλα.

Από τις αρχές της προηγούμενης δεκαετίας, όταν σχεδιάστηκε το πρώτο μουσικό πρόγραμμα για ευρεία χρήση από τους μουσικούς έως σήμερα, η συνεργασία μουσικής και ηλεκτρονικών υπολογιστών επεκτάθηκε σε κάθε τομέα της μουσικής : Σύνθεση, ενορχήστρωση, αυτοσχεδιασμός, παραγωγή και εκπαίδευση. Δημιουργήθηκε έτσι ένας νέος τομέας με τον όρο Μουσική Τεχνολογία, ο οποίος αναφέρεται στη συνεργασία μεταξύ της ηλεκτρονικής τεχνολογίας και της μουσικής. Σήμερα ο όρος Μουσική Τεχνολογία αντιπροσωπεύει όχι απλά μία εξειδίκευση της μουσικής αλλά έναν ειδικό κλάδο, ο οποίος μάλιστα αποτελεί αντικείμενο ειδικού ενδιαφέροντος στο νομοσχέδιο για τη μουσική εκπαίδευση.

Στην καθομιλουμένη, έχουμε συνδέσει τον όρο «εφέ» με την επίδειξη και με τη ψεύτικη εικόνα που προσπαθεί να μας δείξει κάποιος. Επίσης το έχουμε συνδέσει και με το τέχνασμα, ή με την πονηριά, άρα και με μια διάθεση εξαπάτησης. Και επειδή η λέξη δεν είναι ελληνική, όταν τη χρησιμοποιούμε, ίσως να εννοούμε λίγο από όλα αυτά. Η μετάφραση όμως της λέξης effect σημαίνει επίδραση. Ουσιαστικά λοιπόν οι «συσκευές effects» αυτό κάνουν. Επίδρουν στον ήχο για να το διαμορφώσουν, (modulation), να τον παραμορφώσουν, (distortion), να δημιουργήσουν μια άλλη εικόνα, (EQ), άλλη ατμόσφαιρα, (Reverb + Chorus + Phaser + Flanger), να δημιουργήσουν ψευδαισθήσεις, (Stereo Channel, Panning, Preamp/Speaker Simulator), ακουστικά τρυκ, (Delay, Tap Delay, Pitch Shifting, Harmonizer, Wah, Auto-WAH), και πολλά-πολλά ακόμα, που δύσκολα περιγράφονται με λέξεις. Το γεγονός ότι μπορούμε να χρησιμοποιήσουμε το κάθε effect ξεχωριστά ή σε συνδυασμό με πολλά άλλα effects, τότε καταλαβαίνουμε ότι αφ' ενός έχουμε άπειρους ήχους ποσοτικά, αφ' ετέρου βλέπουμε, ότι το ηχητικό αποτέλεσμα παίρνει τέτοιες διαστάσεις, που πρέπει να παραδεχθούμε, ότι δε θα μπορούσαμε να το πετύχουμε με κανένα φυσικό όργανο, αλλά ούτε και με ηλεκτρικό, χωρίς τη βοήθεια αυτών των συσκευών.

Τα effects εντυπωσιάζουν αρχάριους και προχωρημένους και πρέπει να δεχθούμε, ότι μας είναι πλέον απαραίτητα. Είναι απαραίτητα, γιατί ο ήχος πάντα αποτελεί πηγή έμπνευσης και εδώ μπορούμε να έχουμε πολλούς ήχους. Πολλούς πρωτότυπους ήχους, που πέρα από τη δημιουργία της σύνθεσης έχουμε την ικανοποίηση της δημιουργίας πρωτότυπων δικών μας ήχων. Βλέπουμε ότι, στις δυο τελευταίες δεκαετίες έχουμε ήδη κάποια νέα μουσικά στυλ, που με αφετηρία το Hard Rock, στήριξαν τον ήχο τους αποκλειστικά στο Overdrive και στο Distortion τουλάχιστον. Στυλ όπως είναι το Heavy Metal, Thrash, Power metal, Neoclassical, κλπ. το Overdrive και το Distortion είναι πλέον αναπόσπαστο κομμάτι του ήχου τους. Σήμερα με την εξέλιξη της τεχνολογίας έχουμε πληθώρα από effects, που έχουν καταλάβει μια πολύ σημαντική θέση στη μουσική παραγωγή σε όλα τα είδη. Όπως επίσης και στον κινηματογράφο, που τα οπτικά Special effects θα έχαναν ένα μεγάλο μέρος από τη δυναμική και τη μαγεία τους χωρίς τα ηχητικά effects.

Σύντομη περιγραφή αντικειμένου

Αντικείμενο της παρούσας μεταπτυχιακής διατριβής είναι η ψηφιακή επεξεργασία σημάτων και συγκεκριμένα η προσομοίωση ενός ψηφιακού συστήματος ήχου, παρόμοιας διάταξης με τα υπόλοιπα επαγγελματικά ψηφιακά συστήματα ήχου τα οποία προϋπάρχουν. Το λογισμικό που έχουμε αναπτύξει αποτελείται από έναν ενισχυτή ήχου, έναν γραφικό ισοσταθμιστή 10 περιοχών, καθώς και από τρία από τα πιο δημοφιλή εφέ ήχου, το delay, το reverb και το flanger. Η ψηφιακή κατασκευή του συστήματος αυτού μέσα από το λογισμικό πακέτο Matlab, μας κάνει να ξεφεύγουμε από τα αναλογικά κυκλώματα επεξεργασίας ήχου, τα οποία εκτός από το ότι είναι ιδιαίτερα δαπανηρά, δεν είναι και το ίδιο ευέλικτα όσο τα ψηφιακά. Μέσα από αυτή τη διαδικασία το ηχητικό σήμα θα διαμορφωθεί και το ηχητικό αποτέλεσμα θα πάρει τέτοιες διαστάσεις που δεν θα μπορούσαμε να το πετύχουμε με κανένα φυσικό όργανο.

Η παρούσα μεταπτυχιακή εργασία είναι χωρισμένη σε δυο μέρη: το θεωρητικό και το πρακτικό. Στο θεωρητικό μέρος παρουσιάζεται το θεωρητικό υπόβαθρο, του οποίου η κατανόηση θα μας βοηθήσει αργότερα στο πρακτικό μέρος της διατριβής.

Το 1^ο Κεφάλαιο ξεκινάει με αναλυτική περιγραφή του ήχου ως φυσικού φαινομένου. Στη συνέχεια γίνεται αναφορά στον κβαντισμό σημάτων, καθώς και στην μετατροπή αναλογικού σήματος σε ψηφιακό.

Στο 2^ο Κεφάλαιο αναφερόμαστε στη μελέτη κατασκευής του αυτιού καθώς και στην μελέτη της ανθρώπινης αντίληψης του ήχου.

Στο 3^ο Κεφάλαιο γίνεται εκτενής αναφορά στις σειρές καθυστέρησης καθώς και σε κάποια βασικά ψηφιακά φίλτρα, τα οποία από μόνα τους αποτελούν αυτόνομες δομικές μονάδες για την υλοποίηση συγκεκριμένων αλγορίθμων σύνθεσης και επεξεργασίας ήχου.

Στο 4^ο Κεφάλαιο παρουσιάζεται η γλώσσα προγραμματισμού MATLAB, η οποία θα χρησιμοποιηθεί στο πρακτικό μέρος για την ανάπτυξη της εφαρμογής λογισμικού του συστήματός μας.

Το πρακτικό μέρος ασχολείται με την ανάλυση των μουσικών εφέ και τον σχεδιασμό των αλγορίθμων για την προσομοίωση αυτών. Ξεκινά με μια γενική περιγραφή του συστήματος που σχεδιάστηκε και εξειδικεύει ανά κεφάλαιο στα επιμέρους δομικά χαρακτηριστικά του.

Το 5^ο Κεφάλαιο ξεκινά με μια γενική περιγραφή του συστήματος που σχεδιάστηκε και εξειδικεύει στη συνέχεια στα επιμέρους δομικά χαρακτηριστικά του. Αναλύεται η θεωρητική πλευρά καθενός από τα εφέ που χρησιμοποιήθηκαν χωριστά, τα ιδιαίτερα χαρακτηριστικά του, καθώς και ο σχεδιασμός του στο λογισμικό πακέτο Simulink του Matlab.

Μέρος Πρώτο: Θεωρητικό πλαίσιο

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΔΑΛΙΑΣ



СЕРТИФИКАТ

Υλικό

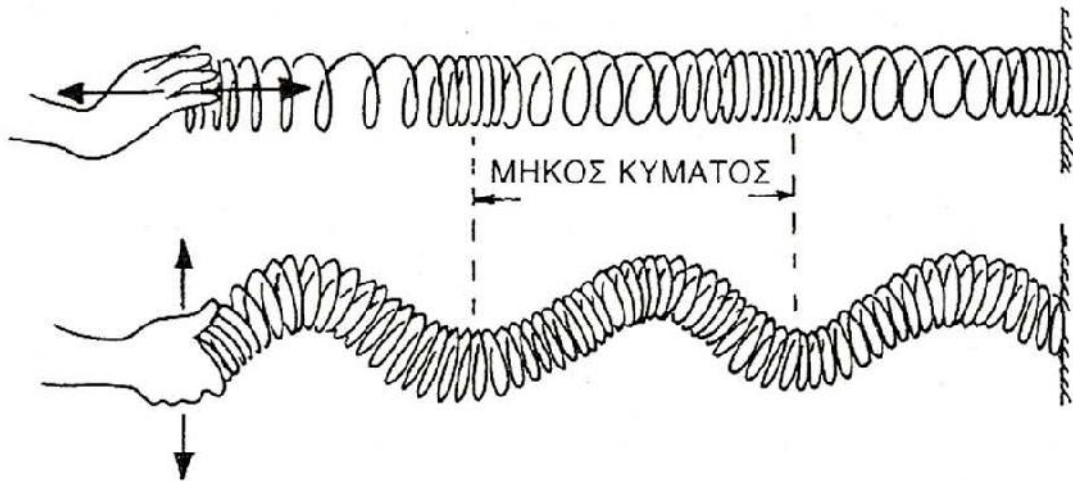
Αέρας
Νερό
Ξύλο
Μάρμαρο
Ατσάλι
Γυαλί

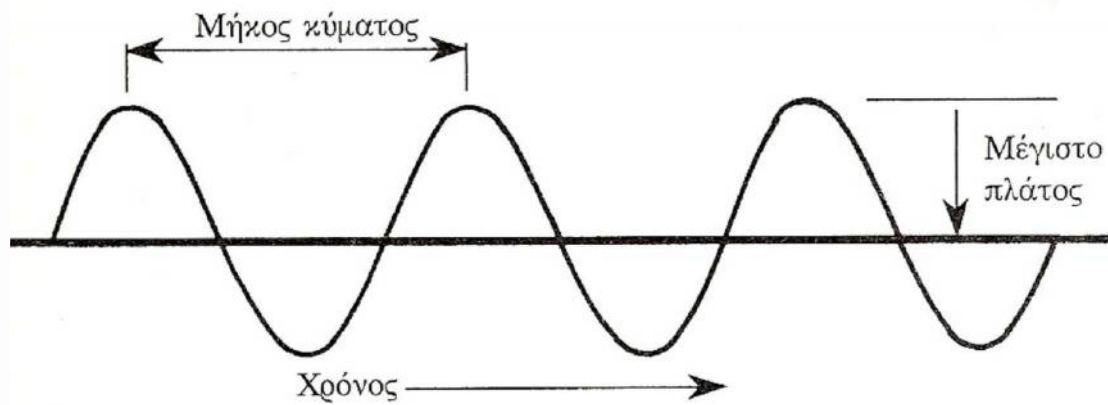
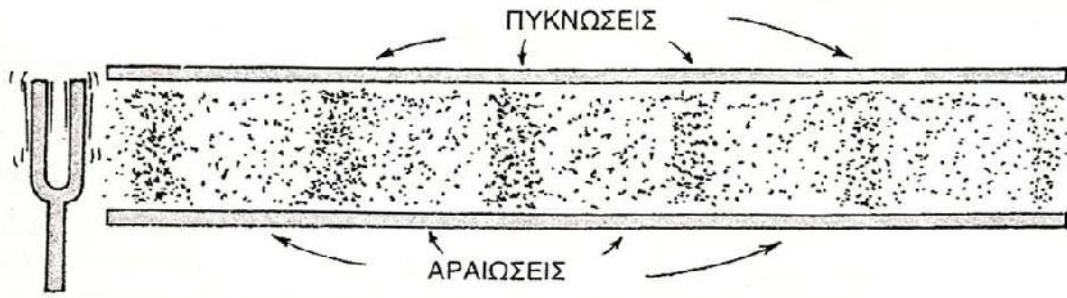
Ταχύτητα ήχου

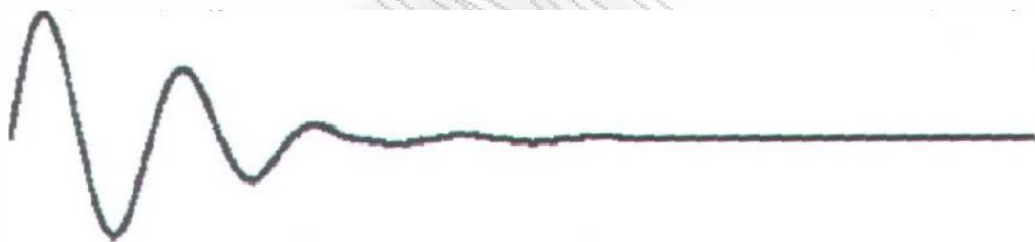
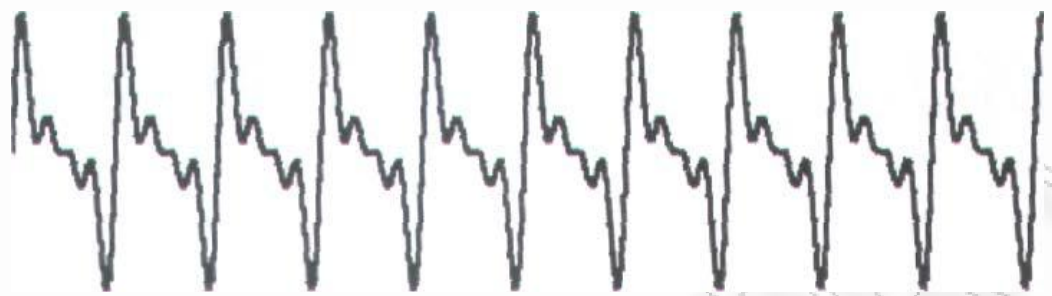
340m/s
1480m/s
3300m/s
3800m/s
5100m/s
5500m/s

f
 λ

T
 c
 f







PAWELIK

- **Συχνότητα (Frequency)** : Είναι ο αριθμός των ταλαντώσεων που εκτελούν τα μόρια του ελαστικού μέσου μέσα στο οποίο διαδίδεται το ηχητικό κύμα στη μονάδα του χρόνου. Η μονάδα μέτρησης της συχνότητας είναι το Hz (κύκλοι ανά δευτερόλεπτο).

Η συντριπτική πλειοψηφία των ήχων δεν αποτελείται από μία μόνο συχνότητα αλλά από ένα σύνολο συχνοτήτων, στο οποίο συνήθως κυριαρχεί μία που τις περισσότερες φορές έχει μεγαλύτερη ένταση από τις υπόλοιπες και καλείται "θεμελιώδης συχνότητα". Οι υπόλοιπες είναι γνωστές ως "παράγωγες συχνότητες" και συνήθως έχουν μικρότερη ένταση και υψηλότερη τιμή.

Η συχνότητα του ήχου με την κυκλική συχνότητα και το μήκος κύματος συνδέονται με τις σχέσεις:

$$f = \frac{\omega}{2\pi}, \lambda = \frac{c}{f}$$

όπου λ το μήκος κύματος.

Ο ήχος ανάλογα από τις συχνότητες που αποτελείται κατατάσσεται στις εξής κατηγορίες:

Ήχοι με συχνότητες από 20 Hz - 20 KHz είναι αντιληπτοί από τον άνθρωπο και ονομάζονται ακουστοί. Αυτές οι τιμές κυμαίνονται και εξαρτώνται από τον συγκεκριμένο άνθρωπο.

Συχνότητες πάνω από 20 KHz ονομάζονται υπερηχητικές (ultrasonic). Αρκετά ζώα μπορούν να λάβουν κύματα ήχου με συχνότητες πάνω από 20 KHz (όπως ο σκύλος μέχρι 50 KHz και η νυχτερίδα μέχρι 100KHz).

Συχνότητες κάτω από 20 HZ ονομάζονται υποηχητικές (infrasonic ή μερικές φορές subsonic). Πηγές ήχων με υποηχητικές συχνότητες είναι οι σεισμοί, κεραυνοί, ηφαίστεια κ.α..

- **Φασματικό περιεχόμενο (Spectrum content)** : Γενικά οι ήχοι στη φύση είναι σύνθετοι. Αυτό σημαίνει ότι είναι αποτέλεσμα της συνύπαρξης πολλών απλών ήχων με διαφορετικά πλάτη και συχνότητες. Το σύνολο των σχετικών εντάσεων των απλών αυτών ήχων συνθέτουν το φασματικό περιεχόμενο του ήχου.

1.3.2. Υποκειμενικά χαρακτηριστικά

Τα υποκειμενικά χαρακτηριστικά του ήχου είναι η ακουστότητα, το ύψος και η χροιά.

- **Ακουστότητα (Loudness)** : Είναι η υποκειμενική απόκριση του αυτιού στη στάθμη του ήχου. Όταν δεν υπάρχει ηχητικό κύμα τότε η ατμοσφαιρική πίεση παραμένει σταθερή και αυτό καλείται σιγή. Καθώς όμως ηχητικά κύματα καταφθάνουν στο αυτί μας, η ατμοσφαιρική πίεση μεταβάλλεται πάνω και κάτω από τη φυσιολογική της τιμή. Η ποσότητα αυτής της μεταβολής εκλαμβάνεται ως ακουστότητα του ήχου. Μπορούμε λοιπόν να πούμε ότι η ακουστότητα είναι εκείνο το χαρακτηριστικό, το οποίο μας δίνει τη δυνατότητα να αντιληφθούμε αν και κατά πόσο δυνατός - ισχυρός είναι ένας ήχος. Είναι η υποκειμενική αντίληψη της έντασης και εξαρτάται από δυο παράγοντες: την ένταση και τη συχνότητα. Μονάδα μέτρησης της ακουστότητας είναι το phon. Ήχος 1phon μόλις που γίνεται αντιληπτός από το ανθρώπινο αυτί, ενώ ήχοι με ακουστότητα κοντά στα 130 phons αρχίζουν να προκαλούν πόνο.

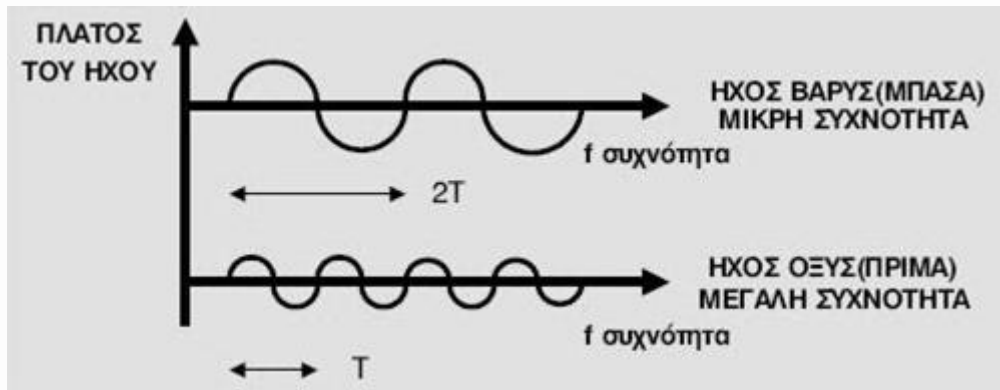
Πολλοί συγχέουν την ακουστότητα με την ένταση, αφού στην καθημερινή μας ζωή οι αναφορές για το πόσο δυνατά ακούγεται ένας ήχος, γίνονται μόνο στην ένταση του. Δύο ήχοι είναι δυνατόν να έχουν την ίδια ένταση αλλά διαφορετική ακουστότητα, δηλαδή ο ένας να ακούγεται πιο δυνατά από τον άλλο λόγω των διαφορετικών τους συχνοτήτων.

Επειδή η ακουστότητα δεν αυξάνεται αναλογικά με την αύξηση των phons, μία πιο πρακτική μονάδα μέτρησης της ακουστότητας που πληροί αυτή την απαίτηση είναι το sone. Ήχος 1 sone ισοδυναμεί με ακουστότητα 40 phones που παράγει τόνος 1 KHz. Συμβολίζοντας με s τα sones και p τα phones μια προσεγγιστική έκφραση μεταξύ τους, δίνει η σχέση :

$$\log s = 0.03(p - 40)$$

- **Ύψος (Pitch)** : Είναι η υποκειμενική απόκριση του αυτιού στη συχνότητα. Ο ρυθμός μεταβολής της ατμοσφαιρικής πίεσης λόγω ενός ηχητικού κύματος προσδιορίζει το χαρακτηριστικό εκείνο που καλούμε ύψος του ήχου. Με βάση το ύψος διακρίνουμε τον ήχο σε οξύ (πρίμα) μέσω και σε βαρύ (μπάσα). Το ύψος του ήχου εξαρτάται κατά κύριο

λόγο από τη θεμελιώδη συχνότητα του ηχητικού κύματος, και κατά δεύτερο λόγο από την ένταση του. Δύο ήχοι είναι δυνατόν να έχουν διαφορετική συχνότητα, αλλά η διαφορά αυτή να μη γίνεται αντιληπτή, λόγω της διαφορετικής τους έντασης.



Εικόνα 1.4: Ήχοι μικρής και μεγάλης συχνότητας

- **Χροιά (Timber)** : Χροιά ονομάζουμε τα υποκειμενικά χαρακτηριστικά του ήχου , τα οποία κάνουν δυνατό το διαχωρισμό δυο τόνων της ίδιας έντασης και θεμελιώδους συχνότητας αλλά διαφορετικών κυματομορφών.

Είναι εκείνο το χαρακτηριστικό που δίνει το λεγόμενο "χρώμα" στον ήχο. Μας δίνει τη δυνατότητα να ξεχωρίζουμε ήχους που έχουν την ίδια ακουστότητα και το ίδιο ύψος, αλλά προέρχονται από διαφορετική πηγή. Η χροιά έχει άμεση σχέση με το φασματικό περιεχόμενο, τον αριθμό των παράγωγων συχνοτήτων του ήχου, τη σχετική θέση του ακροατή ως προς την πηγή, ενώ και η ένταση παίζει καθοριστικό ρόλο. Έτσι, λόγω της διαφορετικής χροιάς είναι δυνατό να ξεχωρίσουμε τον ήχο ενός πιάνου από τον ήχο ενός αρμόνιου ακόμη και αν παίζουν την ίδια νότα με την ίδια ένταση.

1.4. Η μονάδα μέτρησης Decibel (dB)

Το μεγάλο εύρος τιμών που μπορούν να πάρουν τα μεγέθη που συσχετίζονται με τον ήχο, αλλά κυρίως λόγω των ιδιοτήτων που παρουσιάζει το σύστημα ακοής του ανθρώπου το οποίο υπακούει σε λογαριθμικούς κανόνες, μας οδήγησε στη χρήση λογαριθμικής κλίμακας του **Bel**.

Η μονάδα Bel ορίζεται ως ο λογάριθμος μιας αδιάστατης ποσότητας. Αδιάστατες ποσότητες είναι ο λόγος δύο ομοειδών ποσοτήτων π.χ. ισχύων, εντάσεων, πιέσεων κ.λπ. Γενικά η μονάδα Bel ορίζεται από τη σχέση:

$$A(\text{bel}) = \log \frac{k_1}{k_0}$$

όπου k_1 , k_0 δύο τιμές του ίδιου μεγέθους.

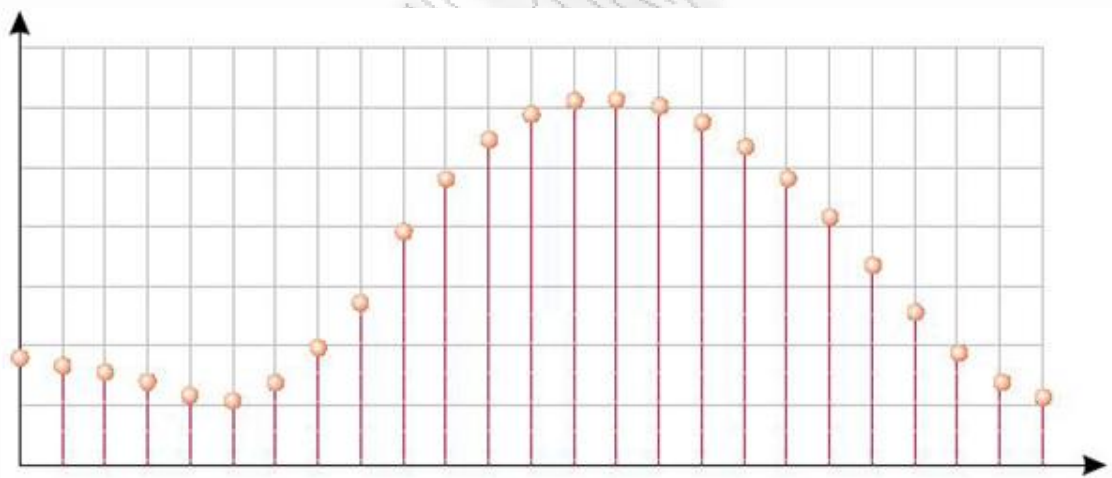
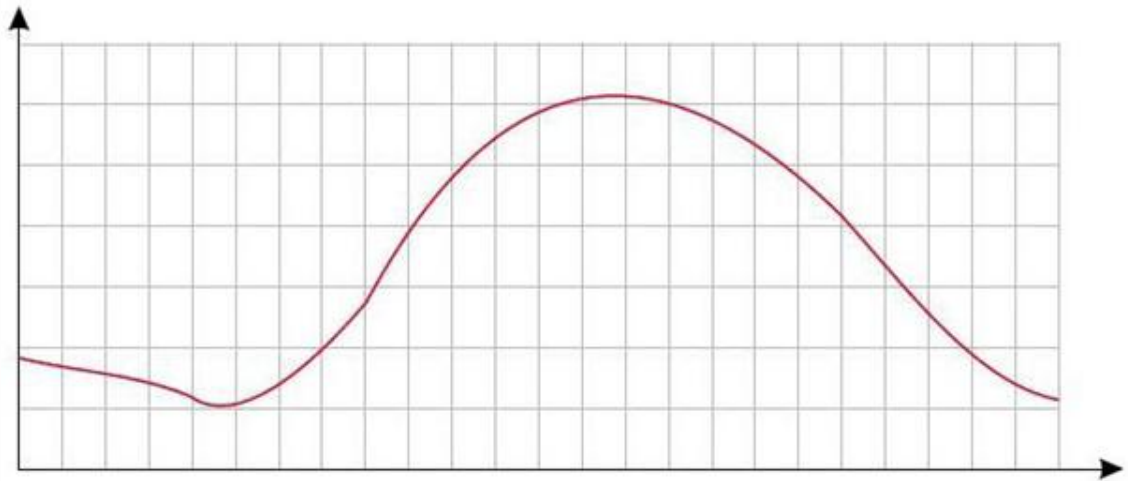
Επειδή το bel από μόνο του είναι μεγάλη μονάδα στην πράξη χρησιμοποιείται υποδιαίρεση του bel ,το **decibel (dB)**. Η μονάδα decibel ορίζεται ως το δέκατο της μονάδας Bel, δηλαδή:

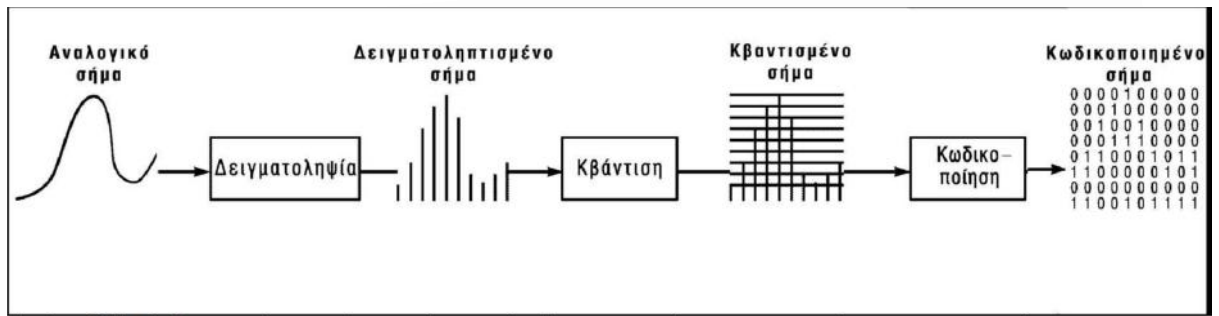
$$A(\text{dB}) = 10 \log \frac{k_1}{k_0}$$

Έτσι, η χρήση του dB για τη σύγκριση της ισχύς W_1 ως προς την W_0 (ποσότητα αναφοράς), απαιτεί τον προσδιορισμό της έκφρασης:

$$10 \log \left(\frac{W_1}{W_0} \right) (\text{dB})$$

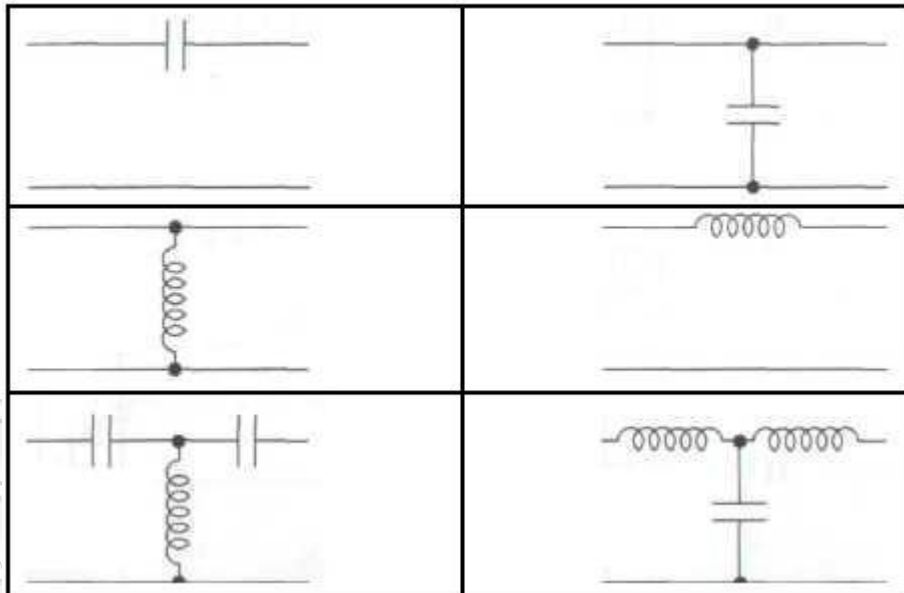
Για παράδειγμα, αν η ισχύς αναφοράς είναι $W_0 = 10 \text{ W}$ ενώ $W_1 = 1000 \text{ W}$, τότε η παραπάνω ποσότητα δίνει 20 dB, το οποίο σημαίνει ότι η ισχύς W_1 είναι κατά 20 dB μεγαλύτερη της W_0 . Εύκολα διαπιστώνουμε πως όταν $W_1 = 2W_0$, τότε η W_1 είναι κατά 3dB μεγαλύτερη της W_0 και αντιστρόφως, Γενικά, διπλασιασμός ισχύος σημαίνει αύξηση κατά 3dB.

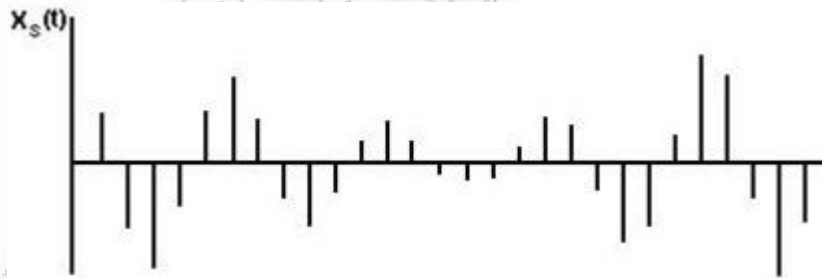
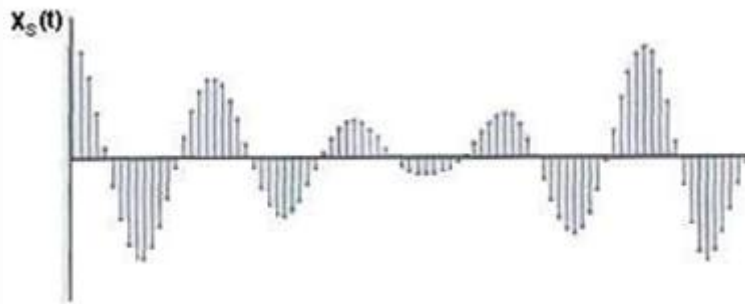
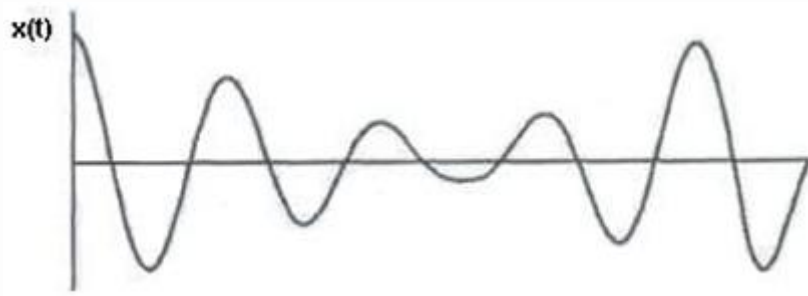




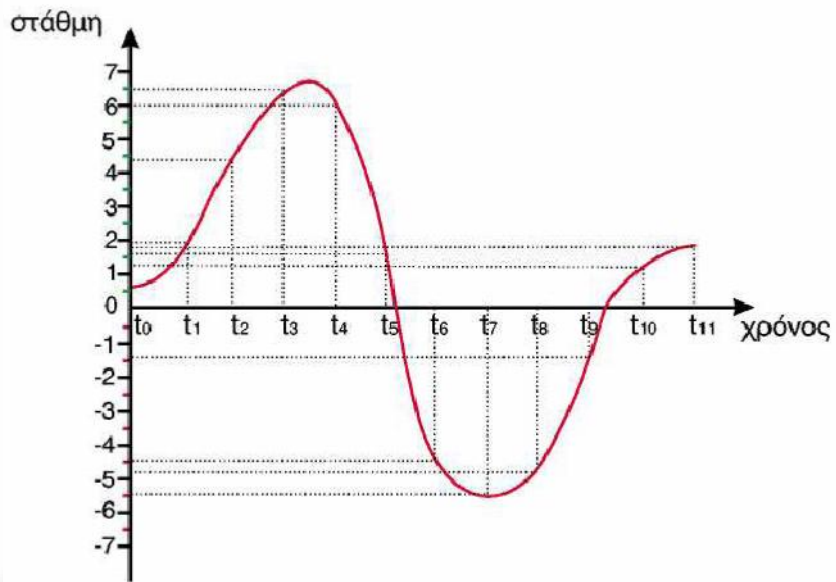
Φίλτρο υψηλής διέλευσης

Φίλτρο χαμηλής διέλευσης

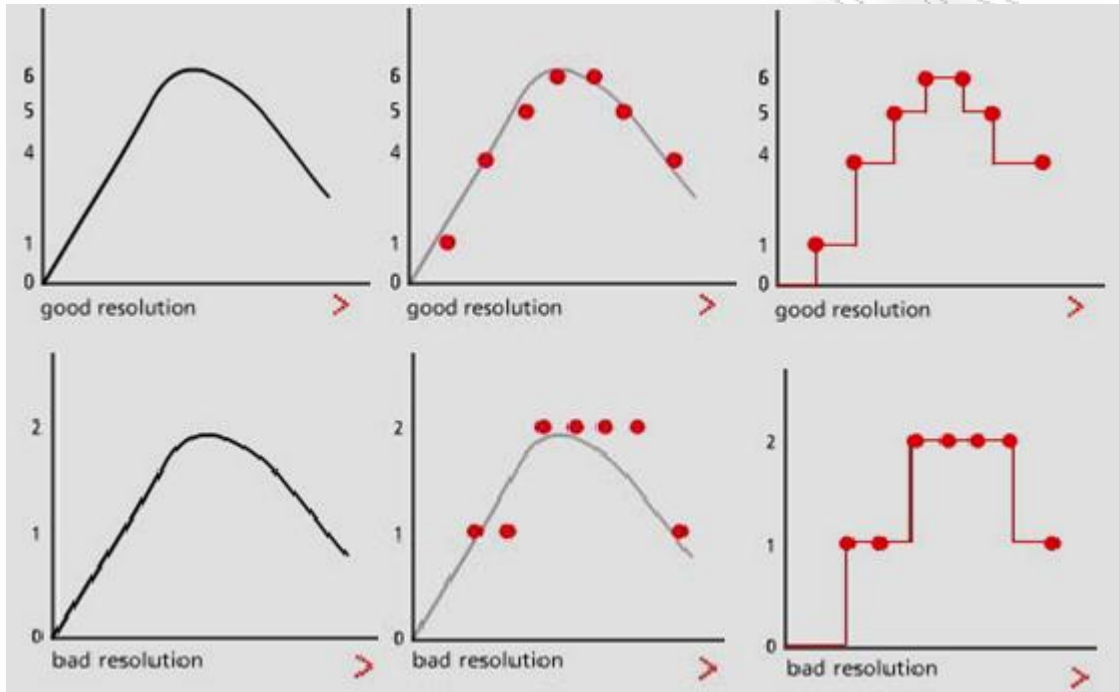




$$f_s = \frac{1}{T_s}$$



$SNR \text{ dB}$ N



Δεκαδικός αριθμός

Συμπλήρωμα ως προς 2

Δεκαδικός αριθμός

Συμπλήρωμα ως προς 2

+7	0111	-1	1111
+6	0110	-2	1110
+5	0101	-3	1101
+4	0100	-4	1100
+3	0011	-5	1011
+2	0010	-6	1010
+1	0001	-7	1001
0	0000	-8	1000

Πίνακας 1.2: Κωδικοποίηση τιμών δείγματος

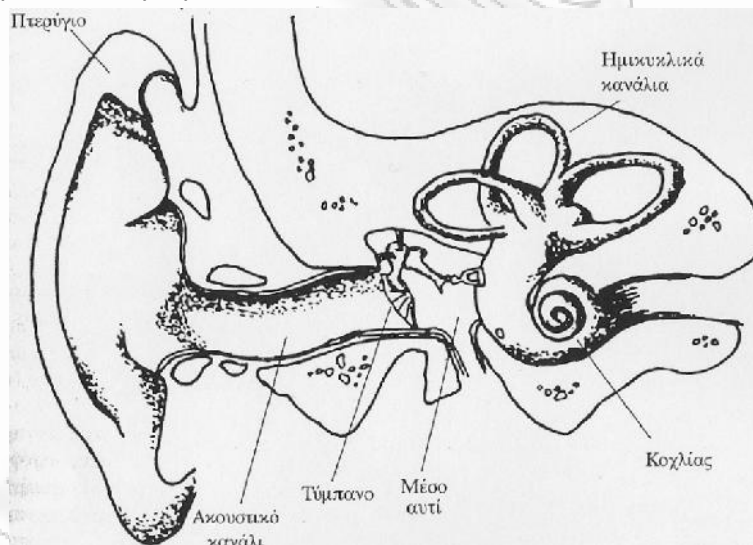
2 Το αυτί και η αντίληψη της ακοής

2.1 Εισαγωγή

Η μελέτη της κατασκευής του αυτιού είναι μια μελέτη φυσιολογίας. Η μελέτη της ανθρώπινης αντίληψης του ήχου βρίσκεται στο γενικό κεφάλαιο της ψυχολογίας. Η ψυχοακουστική είναι ένας συνολικός όρος που περιλαμβάνει την φυσική κατασκευή του αυτιού, τις διαδρομές κίνησης του ήχου, την αντίληψη του ήχου, και τις σχέσεις μεταξύ τους. Το ηχητικό κύμα-ερεθισμός που φτάνει στο αυτί προκαλεί μηχανικές κινήσεις που έχουν σαν αποτέλεσμα λειτουργίες των νεύρων που καταλήγουν στον εγκέφαλο και δημιουργούν μια αίσθηση. Το επόμενο ερώτημα είναι: “Με ποιό τρόπο αναγνωρίζονται και ερμηνεύονται αυτοί οι ήχοι;”

2.2 Η ανατομία του αυτιού

Τα τρία κύρια τμήματα του ανθρώπινου ακουστικού συστήματος, είναι το έξω αυτί, το μέσο αυτί, και το έσω αυτί. Το έξω αυτί αποτελείται από το πτερύγιο (pinna) και από το ακουστικό κανάλι (auditory meatus). Το ακουστικό κανάλι τελειώνει στην ακουστική μεμβράνη ή τύμπανο. Το μέσο αυτί είναι ένας χώρος γεμάτος αέρα που διασχίζεται από τα τρία μικρά οστά που ονομάζονται σφύρα (malleus), άκμονας (incus) και αναβολέας (stapes). Η σφύρα είναι κολλημένη στο τύμπανο και ο αναβολέας κολλημένος στο ελλειψοειδές παράθυρο (oval window) του έσω αυτιού. Όλα μαζί αυτά τα τρία οστά σχηματίζουν μια μηχανική σύνδεση μοχλού μεταξύ του τυμπάνου το οποίο ενεργοποιείται από τον αέρα και του κοχλίου του έσω αυτιού ο οποίος είναι γεμάτος υγρό. Το έσω αυτί τελειώνει στο ακουστικό νεύρο, το οποίο στέλνει ερεθίσματα στον εγκέφαλο.



Εικόνα 2.1: Τα τέσσερα κύρια τμήματα του ανθρώπινου αυτιού: το πτερύγιο, το ακουστικό κανάλι, το μέσο αυτί και το έσω αυτί

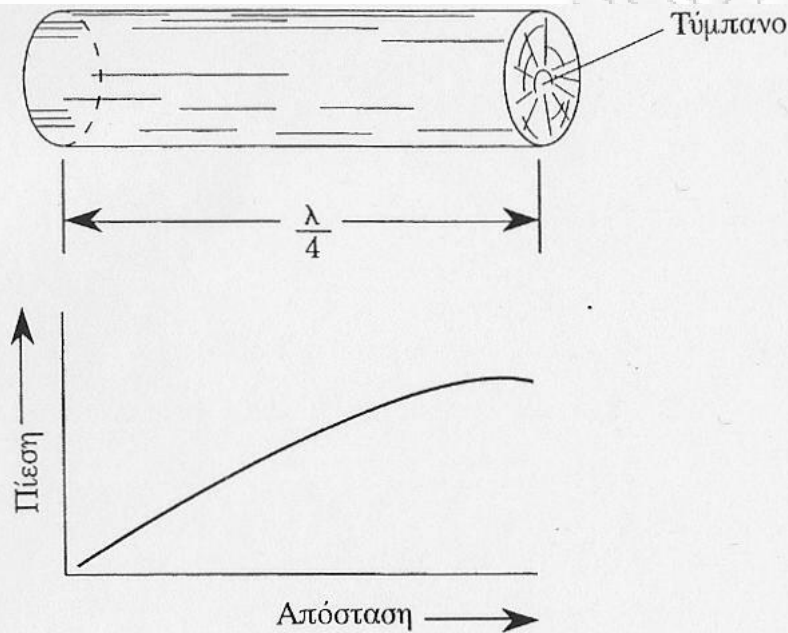
2.2.1 Το πτερύγιο: Κατευθυντικός κωδικοποιητής ήχου

Στην αρχαιότητα, το πτερύγιο θεωρούνταν είτε σαν υποτυπώδες όργανο, είτε σαν απλή συσκευή συλλογής ήχων. Και πράγματι, είναι συσκευή συλλογής ήχων. Το πτερύγιο προσφέρει διαφοροποίηση των ήχων που προέρχονται από εμπρός σε σχέση με τους ήχους που προέρχονται από πίσω. Αυτή η διαφοροποίηση εμπρός/πίσω είναι η ηπιότερη συνεισφορά του πτερυγίου.

Οι πρόσφατες έρευνες έδειξαν ότι το πτερύγιο εκτελεί μια πολύ αποφασιστική λειτουργία χαράσσοντας κατευθυντικές πληροφορίες σε όλους τους ήχους που συλλαμβάνονται από το αυτί. Αυτό σημαίνει ότι πληροφορίες σχετικές με την πηγή του ήχου προστίθενται στο περιεχόμενο του ήχου, έτσι ώστε η τελική ακουστική πίεση στο τύμπανο να δίνει την δυνατότητα στον εγκέφαλο να ερμηνεύει και το περιεχόμενο του ήχου και την κατεύθυνση από την οποία αυτός προέρχεται.

2.2.2 Το κανάλι του αυτιού

Το κανάλι του αυτιού αυξάνει την ηχηρότητα των ήχων που το διαπερνούν. Το κανάλι του αυτιού έχει μέση διάμετρο περίπου 0.7 cm και μήκος περίπου 3 cm και φαίνεται σε σχηματική μορφή σαν ευθύ και με ομογενή διάμετρο σε όλο το μήκος του. Έχουμε ένα αγωγό σε σχήμα σωλήνα που στο μέσα άκρο του κλείνει με το τύμπανο. Η ακουστική ομοιότητα του καναλιού του αυτιού με σωλήνα μουσικού οργάνου δεν διέφυγε από τους πρώτους ερευνητές στον τομέα αυτό. Το φαινόμενο του συντονισμού στο κανάλι του αυτιού αυξάνει την πίεση ήχου στο τύμπανο σε ορισμένες συχνότητες. Το μέγιστο βρίσκεται κοντά στην συχνότητα στην οποία ο σωλήνας των 3 cm έχει ένα τέταρτο μήκους κύματος, περίπου 3000 Hz. Στο παρακάτω σχήμα βλέπουμε το ακουστικό κανάλι, κλειστό στο ένα άκρο από το τύμπανο, το οποίο λειτουργεί σαν “σωλήνας μουσικού οργάνου” ενός τέταρτου μήκους κύματος. Ο συντονισμός δίνει ακουστική ενίσχυση για τις σημαντικές συχνότητες φωνής.

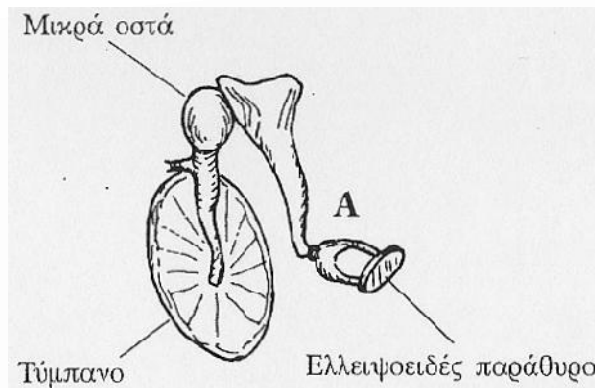


Εικόνα 2.2: Το ακουστικό κανάλι του αυτιού

2.2.3 Το μέσο αυτί

Η μετάδοση της ενέργειας του ήχου από ένα αραιό μέσο όπως είναι ο αέρας, σε πυκνό μέσο όπως είναι το νερό, αποτελεί σοβαρό πρόβλημα. Αν δεν υπάρχει πολύ εξειδικευμένος εξοπλισμός, ο ήχος από τον αέρα ανακλάται στο νερό όπως το φως σε καθρέπτη. Σαν αποτέλεσμα, είναι φανερό ότι δεν πρόκειται να μεταδοθεί αρκετή ισχύς.

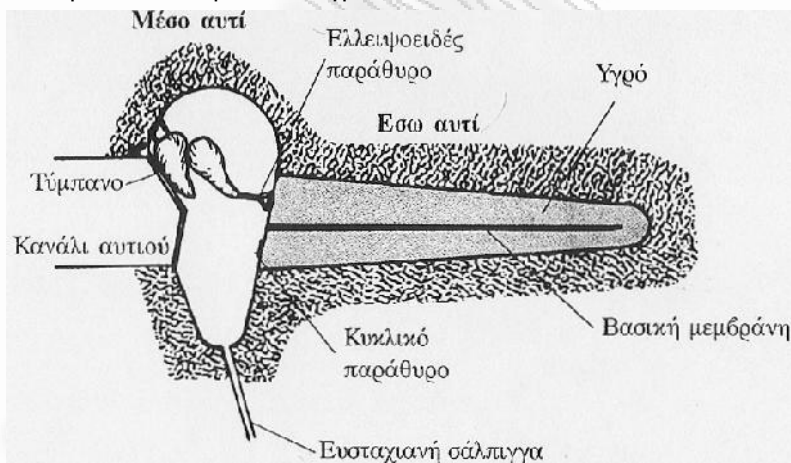
Το θέμα είναι να μεταφερθεί με μέγιστη απόδοση η ασθενική ενέργεια που παριστάνεται από την κίνηση ταλάντωσης ενός μάλλον εύθραυστου διαφράγματος, στο υγρό του έσω αυτιού. Τα τρία μικρά οστά (σφύρα, άκμονας και αναβολέας) σχηματίζουν μια μηχανική σύνδεση μεταξύ του τυμπάνου και του ελλειψοειδούς παραθύρου, το οποίο βρίσκεται σε απαλή επαφή με το υγρό του έσω αυτιού. Το πρώτο από τα τρία οστά, η σφύρα, είναι συνδεδεμένο με το τύμπανο. Το τρίτο, ο αναβολέας, στην πραγματικότητα αποτελεί τμήμα του ελλειψοειδούς παραθύρου. Στη σύνδεση αυτή υπάρχει κίνηση μοχλού με αναλογία από 13:1 μέχρι 3:1. Δηλαδή, η κίνηση του τυμπάνου ελαττώνεται κατά το μέγεθος αυτό στο ελλειψοειδές παράθυρο του έσω αυτιού.



Εικόνα 2.3: Τα μικρά οστά (σφύρα, άκμονας και αναβολέας) του μέσου αυτιού

Η επιφάνεια του τυμπάνου είναι περίπου 80 τετραγωνικά χιλιοστά, και η επιφάνεια του ελλειψοειδούς παραθύρου είναι μόνο 3 τετραγωνικά χιλιοστά. Έτσι, μια δεδομένη δύναμη στο τύμπανο ελαττώνεται κατά ένα λόγο 80/3, ή περίπου 27 φορές. Το πρόβλημα της προσαρμογής ήχου στον αέρα σε ήχο στο υγρό του έσω αυτιού λύνεται όμορφα με την μηχανική του μέσου αυτιού.

Στο παρακάτω σχηματικό διάγραμμα του αυτιού, το κωνικό τύμπανο στο εσωτερικό άκρο του ακουστικού καναλιού σχηματίζει την μία πλευρά του μέσου αυτιού που είναι γεμάτο αέρα. Το μέσο αυτί συνδέεται με το άνω μέρος του λαιμού πίσω από την ρινική κοιλότητα με την ευσταχιακή σάλπιγγα. Το τύμπανο λειτουργεί σαν σύστημα “ακουστικής συνάρτησης”, λειτουργώντας ενάντια στην υποχωρητικότητα του αέρα που είναι παγιδευμένος στο μέσο αυτί. Η ευσταχιακή σάλπιγγα είναι κατάλληλα μικρή και συνεσταλμένη τόσο ώστε να μη καταστρέφει αυτή την υποχωρητικότητα. Το κυκλικό παράθυρο διαχωρίζει το γεμάτο αέρα μέσο αυτί από το πρακτικά ασυμπίεστο υγρό του έσω αυτιού.



Εικόνα 2.4: Σχηματικό διάγραμμα του αυτιού

2.2.4 Το έσω αυτί

Η πολύπλοκη λειτουργία του κοχλία εξακολουθεί ακόμη να βρίσκεται μέσα σε μυστήριο. Ο κοχλίας, που έχει μέγεθος όσο ένα μπιζέλι, βρίσκεται μέσα σε στερεό κόκκαλο, και είναι στριμμένος σαν σαλιγκάρι, απ’ όπου παίρνει και το όνομά του. Το γεμάτο υγρό έσω αυτί χωρίζεται κατά μήκος με δύο μεμβράνες, την μεμβράνη του Reissner και την βασική μεμβράνη.

Η ταλάντωση του τυμπάνου ενεργοποιεί τα μικρά οστά. Η κίνηση του αναβολέα, που είναι κολλημένος στο ελλειψοειδές παράθυρο, προκαλεί την ταλάντωση του υγρού του έσω αυτιού. Μια κίνηση του ελλειψοειδούς παραθύρου προς τα μέσα έχει σαν αποτέλεσμα την ροή του υγρού γύρω από το πέρα άκρο της βασικής μεμβράνης, με αποτέλεσμα κίνηση προς τα έξω του κυκλικού παραθύρου. Ο ήχος που ενεργοποιεί το ελλειψοειδές παράθυρο έχει σαν αποτέλεσμα την δημιουργία στάσιμων κυμάτων στην βασική μεμβράνη. Η θέση των μέγιστων πλατών των στάσιμων κυμάτων στην βασική μεμβράνη μεταβάλλεται καθώς αλλάζει η συχνότητα του ήχου που προκαλεί την διέγερση.

Ο ήχος χαμηλής συχνότητας έχει σαν αποτέλεσμα μέγιστο πλάτος κοντά στο πέρα άκρο της βασικής μεμβράνης. Ο ήχος υψηλής συχνότητας δημιουργεί κορυφές κοντά στο ελλειψοειδές παράθυρο. Σε περίπτωση πολύπλοκου σήματος όπως είναι η ομιλία ή η μουσική, δημιουργούνται πολλές στιγμιαίες κορυφές, οι οποίες μετακινούνται διαρκώς σε πλάτος και θέση κατά μήκος της βασικής μεμβράνης.

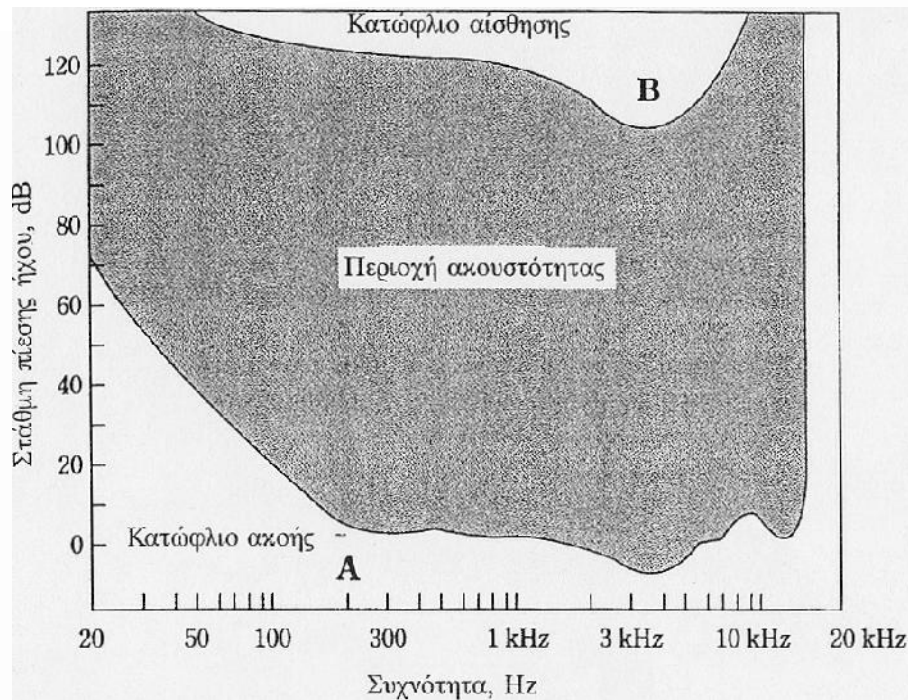
Τα κύματα που δημιουργούνται στην βασική μεμβράνη του γεμάτου υγρό σωλήνα του έσω αυτιού ερεθίζουν άκρες νεύρων ψιλές σαν τρίχες, οι οποίες μεταφέρουν σήματα στον εγκέφαλο με την μορφή εκκένωσης νευρώνων. Περίπου 15000 εξωτερικά κύτταρα με περίπου 140 μικροσκοπικές τρίχες το καθένα, οι οποίες ονομάζονται στερεοβλεφαρίδες (stereocilia) προεξέχουν από το καθένα. Επιπλέον, υπάρχουν περίπου 3500 εσωτερικά κύτταρα με περίπου 40 στερεοβλεφαρίδες το καθένα. Αυτές οι στερεοβλεφαρίδες είναι οι πραγματικοί μετατροπείς ηχητικής ενέργειας σε ηλεκτρική. Καθώς ο ήχος αναγκάζει το υγρό του κοχλία και την βασική μεμβράνη να κινούνται, οι στερεοβλεφαρίδες στα κύτταρα με τις τρίχες λυγίζουν, οπότε ηλεκτρικά κύματα οδηγούνται προς τον ακουστικό φλοιό του εγκεφάλου.

Όταν ο ήχος διεγείρει το υγρό στο έσω αυτί, ερεθίζονται η μεμβράνη και τα κύτταρα με τις τρίχες και στέλνουν ένα ηλεκτρικό κύμα μέσα από τον ιστό που τα περιβάλλει. Αυτά τα επονομαζόμενα ακουστικά δυναμικά (acoustic potentials), που είναι αναλογικά, λαμβάνονται και ενισχύονται, αναπαράγοντας τον ήχο που πέφτει στο αυτί, το οποίο λειτουργεί σαν βιολογικό μικρόφωνο. Τα δυναμικά αυτά είναι ανάλογα με την πίεση ήχου και έχουν γραμμική απόκριση σε μια περιοχή μέχρι 80 dB.

2.3 Περιοχή ακουστότητας

Στο παρακάτω σχήμα, οι δύο καμπύλες A και B παριστάνουν τα άκρα της αντίληψής μας για την ηχηρότητα. Η καμπύλη A, το κατώφλιο ακοής, μας λέει ότι τα ανθρώπινα αυτιά είναι περισσότερο ευαίσθητα στα 3 KHz. Γύρω στα 3 KHz ένας ήχος χαμηλής στάθμης προκαλεί απόκριση κατωφλίου η οποία δεν προκαλείται από μεγαλύτερες ή μικρότερες συχνότητες. Σε αυτή την πολύ ευαίσθητη περιοχή, μία στάθμη πίεσης ήχου 0 dB μόλις μπορεί να ακουστεί από ένα άτομο μέσης ακουστικής οξύτητας. Η καμπύλη B παριστάνει την στάθμη σε κάθε συχνότητα στην οποία γίνεται αισθητό ένα γαργάλημα στα αυτιά. Αυτό συμβαίνει σε στάθμη πίεσης ήχου περίπου 120 ή 130 dB. Επιπλέον αύξηση της στάθμης έχει αποτέλεσμα αύξηση του αισθήματος μέχρι να παραχθεί μια αίσθηση πόνου. Το πρώτο γαργάλημα είναι μία προειδοποίηση ότι ο ήχος γίνεται επικίνδυνα ηχηρός και ότι επίκειται βλάβη στο αυτί ή ότι ήδη έχει γίνει.

Μεταξύ κατωφλίου ακοής (καμπύλη A) και κατωφλίου αίσθησης (καμπύλη B) είναι η περιοχή ακουστότητας. Αυτή είναι μια περιοχή δύο διαστάσεων: της κατακόρυφης διάστασης της στάθμης πίεσης ήχου και της οριζόντιας περιοχής συχνοτήτων που μπορεί να αντιληφθεί το αυτί. Όλοι οι ήχοι που αντιλαμβάνονται οι άνθρωποι πρέπει να έχουν τέτοια συχνότητα και στάθμη ώστε να πέφτουν μέσα στην περιοχή ακουστότητας.



Εικόνα 2.5: Η περιοχή ακουστότητας του ανθρώπινου αυτιού

Η περιοχή ακουστότητας για τους ανθρώπους διαφέρει αρκετά από την περιοχή ακουστότητας για πολλά ζώα. Η νυχτερίδα αντιλαμβάνεται ακουστικές κραυγές που βρίσκονται πολύ πάνω από το ανώτερο όριο συχνότητας των αυτιών μας. Η ακοή των σκύλων εκτείνεται ψηλότερα από την δική μας. Ο ήχος στις υποηχητικές και υπερηχητικές περιοχές, σε σχέση με την ακοή των ανθρώπων, είναι πραγματικός ήχος με την φυσική έννοια, αλλά δεν υποπίπτει στην ανθρώπινη αντίληψη.

2.4 Σχέση χροιάς με το φάσμα

Η χροιά έχει να κάνει με την αντίληψή μας σε σύνθετους ήχους. Η λέξη εφαρμόζεται κύρια στον ήχο διαφόρων μουσικών οργάνων. Ο τόνος κάθε μουσικού οργάνου έχει τη δική του χροιά. Η χροιά καθορίζεται από το πλήθος και την σχετική ένταση των αρμονικών του οργάνου. Η τονική ποιότητα πλησιάζει σαν συνώνυμο στην χροιά.

Η χροιά είναι ένας υποκειμενικός όρος. Ο αντίστοιχος φυσικός όρος είναι το φάσμα. Ένα μουσικό όργανο παράγει ένα θεμελιώδη και μια ομάδα αρμονικών. Το αυτί έχει και αυτό την δική του υποκειμενική ερμηνεία των αρμονικών. Έτσι, η αντίληψη του αυτιού για την συνολική χροιά της νότας του οργάνου μπορεί να είναι αρκετά διαφορετική από το φάσμα.

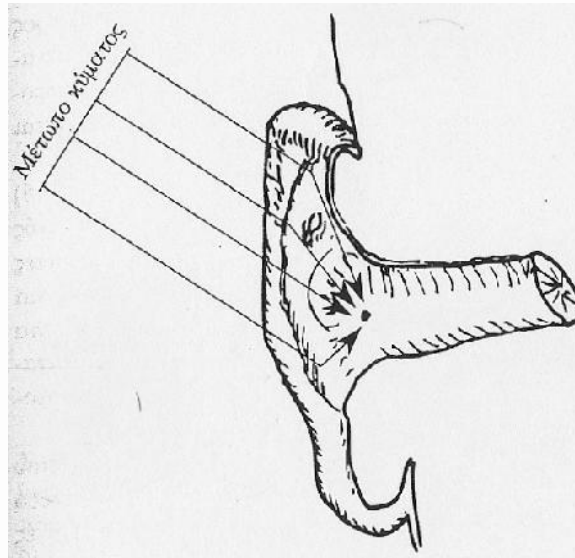
Όταν ακούμε μια ορχήστρα σε αίθουσα συναυλιών, η χροιά που ακούμε είναι διαφορετική για διαφορετικές θέσεις στα καθίσματα. Η μουσική αποτελείται από μεγάλη περιοχή συχνοτήτων, και τα πλάτη και οι φάσεις των διαφόρων συνιστωσών επηρεάζονται από ανακλάσεις από τις διάφορες επιφάνειες της αίθουσας, με αποτέλεσμα να υπάρχει διαφορά μεταξύ χροιάς και συχνότητας σε διαφορετικές θέσεις.

2.5 Εντοπισμός ηχητικών πηγών

Η αντίληψη της κατεύθυνσης της πηγής ενός ήχου είναι εν μέρει αποτέλεσμα της καταπληκτικής λειτουργίας κωδικοποίησης του εξωτερικού αυτιού, δηλαδή του πτερυγίου. Ο ήχος που ανακλάται από τις διάφορες αυλακώσεις, διπλώσεις και επιφάνειες του πτερυγίου προστίθενται με τον (άμεσο) ήχο που δεν ανακλάται στην είσοδο του ακουστικού καναλιού. Αυτή η πρόσθεση, που τώρα είναι κωδικοποιημένη με τις πληροφορίες κατεύθυνσης, περνά μέσα από το ακουστικό κανάλι προς το τύμπανο και από εκεί στο μέσο και στο έσω αυτί και στη συνέχεια στον εγκέφαλο, για ερμηνεία.

Το μέτωπο του κύματος ήχου μπορεί να θεωρηθεί σαν πολλές ακτίνες ήχου που προέρχονται από μια συγκεκριμένη πηγή με μια συγκεκριμένη οριζόντια και κατακόρυφη

γωνία. Καθώς οι ακτίνες αυτές προσπίπτουν στο περύγιο ανακλώνται από τις διάφορες επιφάνειες, και μερικές από τις ανακλάσεις κινούνται προς την είσοδο του ακουστικού καναλιού. Στο σημείο αυτό, αυτές οι ανακλώμενες συνιστώσες προστίθενται με την (άμεση) συνιστώσα που δεν έχει ανακλαστεί.



Εικόνα 2.6: Το μέτωπο κύματος ενός ήχου για τον εντοπισμό ηχητικών πηγών

Οι ανακλάσεις αυτές, που κατευθύνονται προς το άνοιγμα του καναλιού του αυτιού προστίθενται διανυσματικά (σύμφωνα με τα σχετικά πλάτη και τις σχετικές φάσεις). Με τον τρόπο αυτό το περύγιο του αυτιού κωδικοποιεί με πληροφορίες κατεύθυνσης όλους τους ήχους που προσπίπτουν στο αυτί, και τους οποίους ο εγκέφαλος αποκωδικοποιεί σαν αντίληψη κατεύθυνσης.

2.6 Το φαινόμενο προβαδίσματος (φαινόμενο Haas)

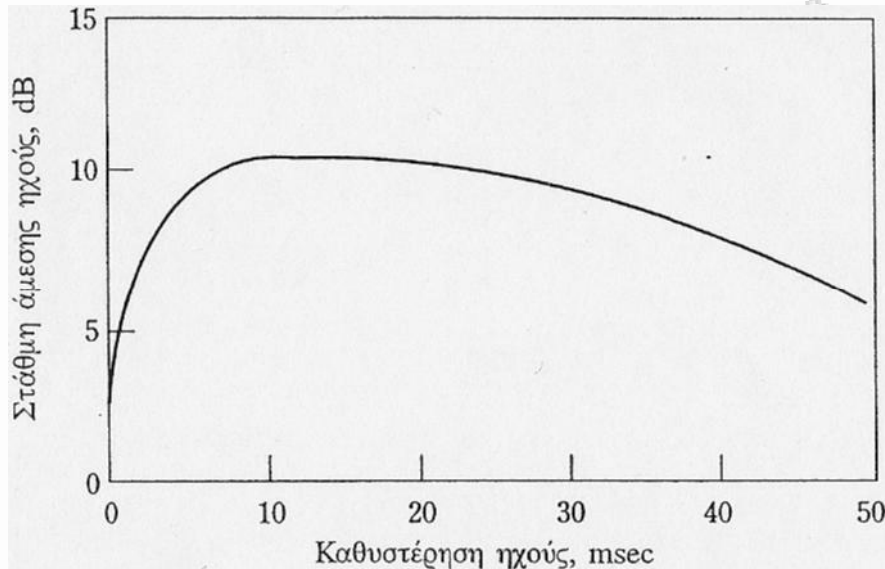
Ο ακουστικός μας μηχανισμός ολοκληρώνει τις εντάσεις ήχου σε μικρά χρονικά διαστήματα και λειτουργεί κάπως σαν βαλλιστικό όργανο μέτρησης. Με απλούστερους όρους, για περίπτωση μέσα σε αίθουσα, το αυτί και ο εγκέφαλος έχουν την αξιοσημείωτη ιδιότητα να συλλέγουν όλες τις ανακλάσεις που φτάνουν μέσα στα περίπου 50 msec μετά τον άμεσο ήχο και να τις προσθέτουν (ολοκληρώνουν) έτσι ώστε να δημιουργείται η εντύπωση ότι όλοι αυτοί οι ήχοι έρχονται από την διεύθυνση της αρχικής πηγής, αν και υπάρχουν ανακλάσεις από άλλες κατευθύνσεις. Η ηχητική ενέργεια που ολοκληρώνεται σε αυτήν την χρονική περίοδο δίνει και μια εντύπωση πρόσθετης ηχηρότητας.

Δεν πρέπει να μας εκπλήττει το γεγονός ότι το ανθρώπινο αυτί συγχωνεύει όλους τους ήχους που φτάνουν κατά την διάρκεια ενός ορισμένου χρονικού παραθύρου. Τα μάτια μας αντίστοιχα συγχωνεύουν μια σειρά από ακίνητες εικόνες στον κινηματογράφο, δίνοντας μας την εντύπωση συνεχούς κίνησης. Η ταχύτητα εμφάνισης των ακίνητων εικόνων είναι σημαντική. Πρέπει να υπάρχουν τουλάχιστον 16 εικόνες ανά δευτερόλεπτο (χρονικό διάστημα 62 msec), έτσι ώστε να μη βλέπουμε μια σειρά ακίνητων εικόνων ή ένα τρεμπόπαιγμα. Η ακουστική συγχώνευση λειτουργεί καλύτερα κατά την διάρκεια των πρώτων 20 ή 30 msec. Πέρα από τα 50 μέχρι 80 msec επικρατούν οι διάκριτες αντηχήσεις.

Ο Haas τοποθέτησε τα άτομά του σε απόσταση 3 μέτρων από δύο μεγάφωνα που ήταν τοποθετημένα έτσι ώστε έτεμναν μια γωνία 45 μοιρών, ενώ η γραμμή συμμετρίας των ακροατών διχοτομούσε την γωνία αυτή. Οι συνθήκες ήταν περίπου ανηχοϊκές. Ζητήθηκε από τους ακροατές να ρυθμίσουν ένα εξασθενητή μέχρις ότου ο ήχος από το “απευθείας” μεγάφωνο εξισωνόταν με τον ήχο από το “καθυστερημένο” μεγάφωνο. Στη συνέχεια, ο Haas προχώρησε στην μελέτη της επίδρασης της μεταβολής της καθυστέρησης.

Ο Haas βρήκε ότι στην περιοχή καθυστέρησης από 5 μέχρι 35 msec, ο ήχος από το καθυστερημένο μεγάφωνο πρέπει να αυξηθεί περισσότερο από 10 dB σε σχέση με τον άμεσο πριν ακουστεί σαν ηχώ. Αυτό είναι το **φαινόμενο προβαδίσματος**, ή **φαινόμενο Haas**. Μέσα σε ένα δωμάτιο, η ενέργεια από ανάκλαση που φτάνει στο αυτί μέσα σε 35 msec ολοκληρώνεται με τον απευθείας ήχο και γίνεται αντιληπτή σαν τμήμα του άμεσου ήχου σε

αντίθεση με τον ήχο αντήχησης. Αυτές οι πρώτες ανακλάσεις αυξάνουν την ηχηρότητα του ήχου και, όπως είπε ο Haas, έχουν σαν αποτέλεσμα "... μια ευχάριστη τροποποίηση της εντύπωσης του ήχου με την έννοια ότι πλαταίνουν την πηγή του πρωτεύοντα ήχου ενώ η πηγή της ηχούς δεν γίνεται αντιληπτή από ακουστικής πλευράς".



Εικόνα 2.7: Το φαινόμενο προβαδίσματος (φαινόμενο Haas) στο ανθρώπινο ακουστικό σύστημα

Στην περιοχή από 5 μέχρι 35 msec, η στάθμη της ηχούς πρέπει να είναι περίπου 10 dB μεγαλύτερη από τον άμεσο ήχο για να ξεχωρίσει σαν ηχώ. Στην περιοχή αυτή, από το αυτί συλλέγονται συριστώσες από ανάκλαση που φτάνουν από πολλές κατευθύνσεις. Ο τελικός ήχος φαίνεται εντονότερος, εξαιτίας των ανακλάσεων και φαίνεται ότι προέρχεται από την άμεση πηγή. Για χρονικές καθυστερήσεις από 50 μέχρι 100 msec και περισσότερο οι ανακλάσεις γίνονται αντιληπτές σαν διάκριτες αντηχήσεις.

Η μεταβατική ζώνη μεταξύ του φαινομένου ολοκλήρωσης για καθυστερήσεις μικρότερες από 35 msec και της αντίληψης του καθυστερημένου ήχου σαν διάκριτης ηχούς είναι βαθμιαία και κατά συνέπεια, κάπως απροσδιόριστη.

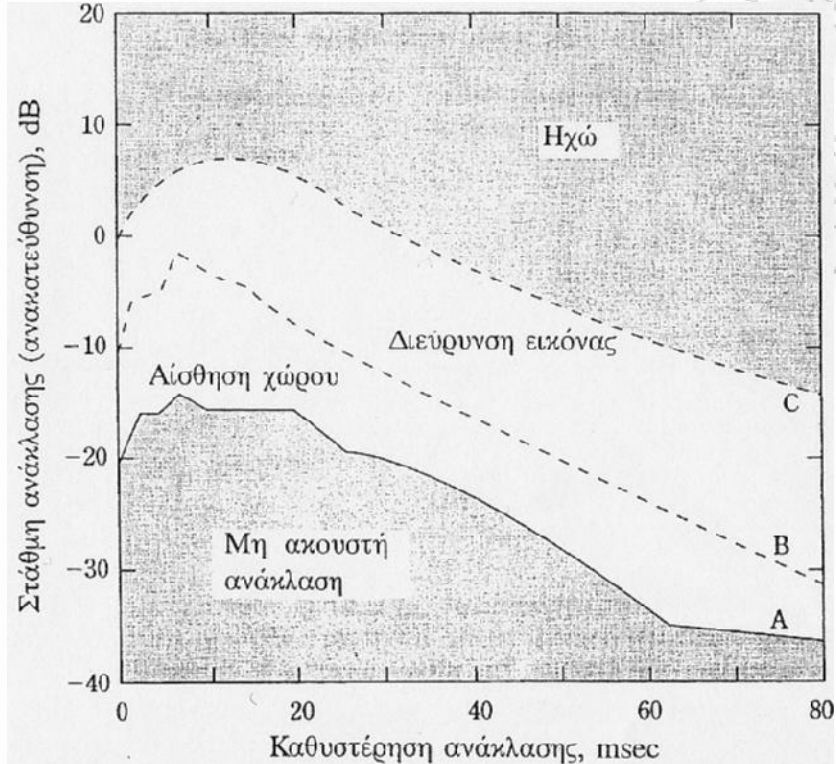
2.7 Αντίληψη του ήχου από ανάκλαση

Είναι ενδιαφέρον το ότι η διάταξη των μεγαφώνων που χρησιμοποίησε ο Haas χρησιμοποιήθηκε και από δεκάδες άλλους ερευνητές και ουσιαστικά πρόκειται για την γνώριμη διάταξη στέρεο, δηλαδή δύο ξεχωριστά μεγάφωνα με τον ακροατή σε συμμετρική θέση μεταξύ των δύο μεγαφώνων. Ο ήχος από το ένα μεγάφωνο ονομάζεται **άμεσος ήχος** και ο ήχος από το άλλο **καθυστερημένος ήχος** (ή **ανάκλαση**). Η καθυστέρηση που εισάγεται μεταξύ των δύο σημάτων και οι σχετικές τους στάθμες είναι ρυθμιζόμενα. Σαν σήμα χρησιμοποιείται ομιλία.

Με τον ήχο του άμεσου μεγαφώνου σε άνετη στάθμη, και με καθυστέρηση, έστω 10 msec, αυξάνεται αργά από μια πολύ χαμηλή τιμή η στάθμη του ανακλώμενου, ή καθυστερημένου, μεγαφώνου. Η στάθμη ήχου της ανάκλασης στην οποία ο ακροατής πρώτα ανιχνεύει μια διαφορά στον ήχο είναι το κατώφλιο ανίχνευσης της ανάκλασης. Για στάθμες μικρότερες από αυτήν, η ανάκλαση δεν είναι ακουστή, ενώ για στάθμες μεγαλύτερες από αυτήν, η ανάκλαση ακούγεται καθαρά.

Καθώς αυξάνεται βαθμιαία η στάθμη ανάκλασης πάνω από την τιμή κατωφλίου, στο άθροισμα του ήχου παρέχεται μια αίσθηση χώρου. Αυτή η αίσθηση χώρου επικρατεί, παρ' όλο που το πείραμα εκτελείται μέσα σε ανηχοϊκό χώρο. Καθώς η στάθμη ανάκλασης αυξάνει περίπου κατά 10 dB πάνω από την τιμή κατωφλίου, στον ήχο παρατηρείται μια άλλη αλλαγή. Στην αυξανόμενη αίσθηση χώρου προστίθεται τώρα μια διεύρυνση της εικόνας του ήχου και πιθανόν μια μετατόπιση της εικόνας προς το άμεσο μεγάφωνο. Καθώς η στάθμη ανάκλασης αυξάνει κατά άλλα 10 dB, ή περίπου τόσο, πάνω από το κατώφλιο διεύρυνσης της εικόνας του ήχου, παρατηρείται μια άλλη μεταβολή: ακούγονται **διάκριτες αντηχήσεις**.

Το παρακάτω σχήμα περιέχει απαντήσεις για την επίδραση του ήχου που ανακλάται από το πάτωμα, την οροφή και από τους τοίχους και προστίθεται στον άμεσο ήχο από τα μεγάφωνα. Οι ανακλάσεις κάτω από το κατώφλιο αντίληψης δεν μπορούν να χρησιμοποιηθούν. Επίσης δεν μπορούν να χρησιμοποιηθούν οι ανακλάσεις που γίνονται αντιληπτές σαν αντηχήσεις. Η χρήσιμη περιοχή είναι η περιοχή χωρίς σκίαση μεταξύ των δύο καμπύλων κατωφλίου, της A και της C. Απλοί υπολογισμοί μπορούν να δώσουν εκτιμήσεις για την στάθμη και για την καθυστέρηση οποιασδήποτε συγκεκριμένης ανάκλασης, εφ' όσον γνωρίζουμε την ταχύτητα του ήχου, την απόσταση που διανύθηκε, και εφαρμόσουμε τον νόμο του αντίστροφου τετραγώνου. Το παρακάτω σχήμα δίνει τις υποκειμενικές αντιδράσεις που πιθανόν θα έχει ο ακροατής στην πρόσθεση οποιασδήποτε ανάκλασης και του άμεσου ήχου.



Εικόνα 2.8: Οι επιδράσεις των πλευρικών ανακλάσεων στην αντίληψη του άμεσου ήχου σε προσομοίωση διάταξης στέρεο

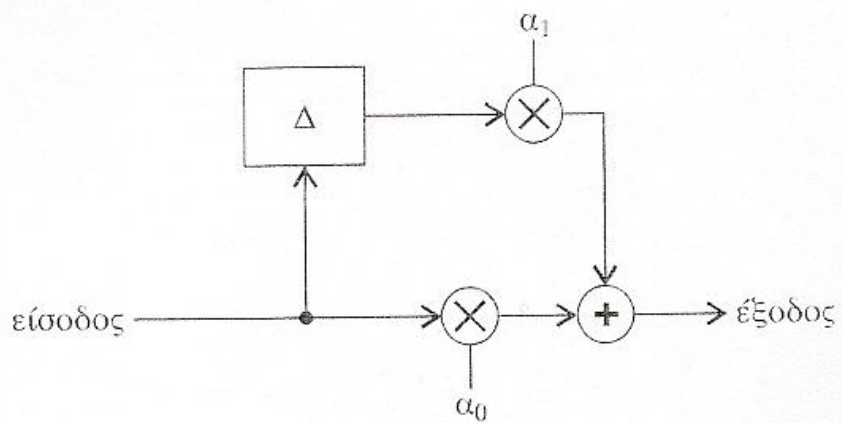
Όπου (A) απόλυτο κατώφλιο ακουστότητας της ανάκλασης, (B) μετατόπιση εικόνας/διεύρυνση κατωφλίου και (C) πλευρική ανάκλαση που γίνεται αντιληπτή σαν διάκριτη ηχώ.

$$\text{Καθυστέρηση ανάκλασης} = \frac{\text{απόσταση ανάκλασης} - \text{άμεση απόσταση}}{\text{ταχύτητα του ήχου}}$$

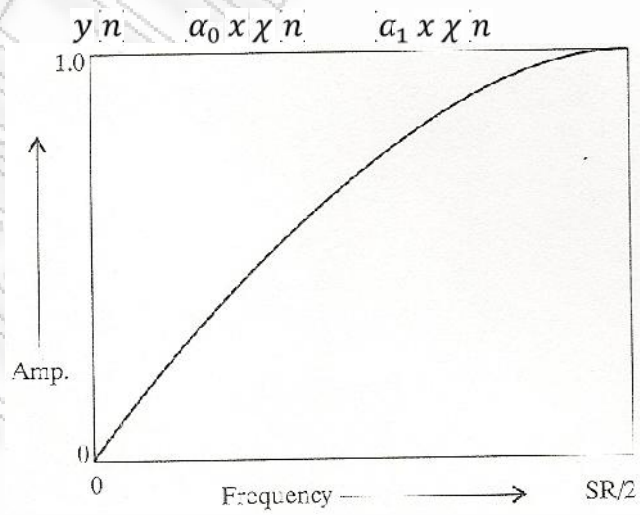
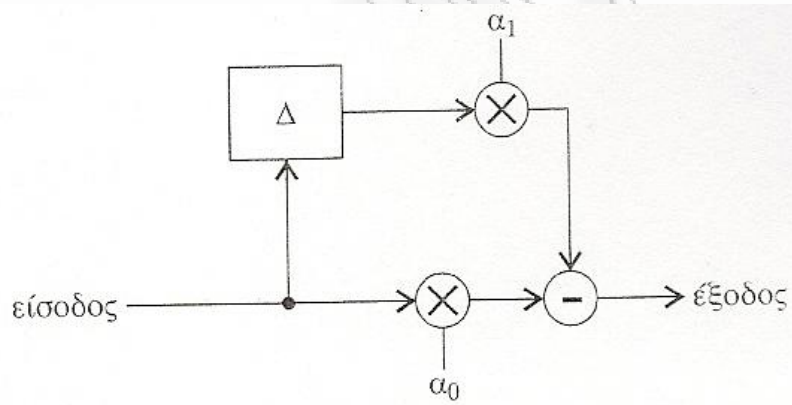
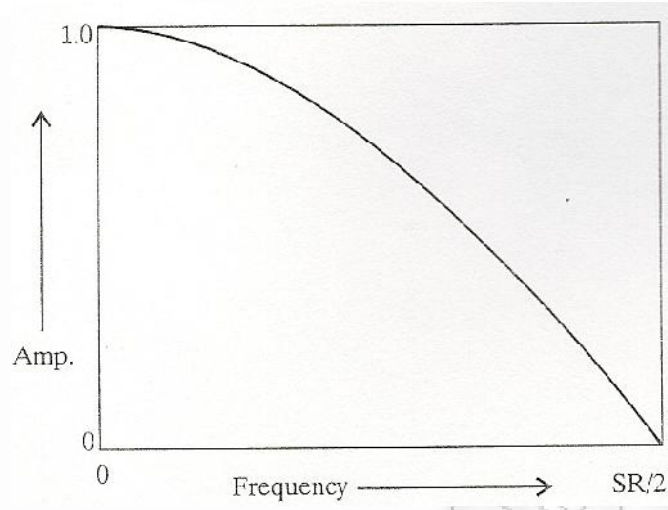
με την προϋπόθεση ανάκλασης κατά 100% στην ανακλώσα επιφάνεια.

$$\text{Στάθμη ανάκλασης στην θέση ακρόασης} = 20 \log \frac{\text{άμεση απόσταση}}{\text{απόσταση ανάκλασης}}$$

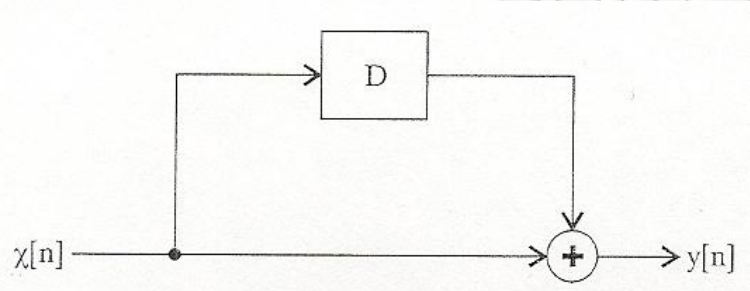
με την προϋπόθεση διάδοσης κατά το αντίστροφο τετράγωνο.



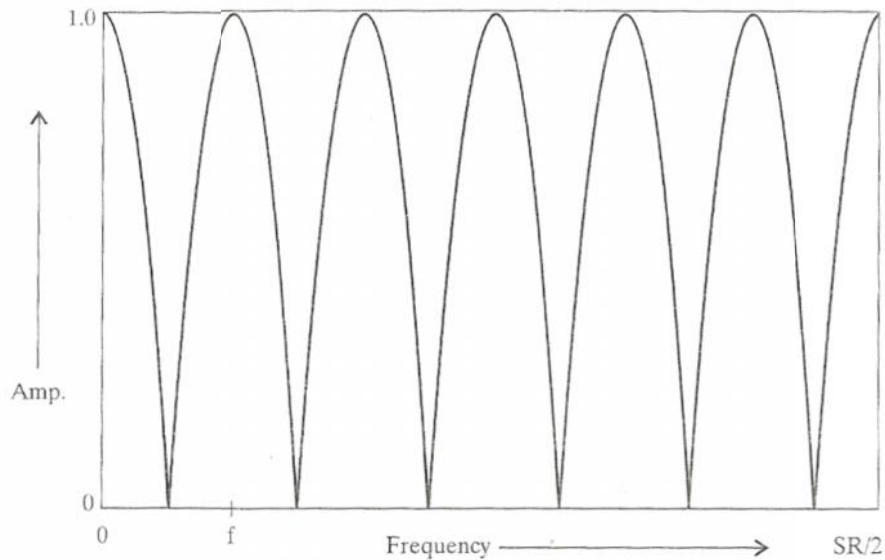
$$y[n] = \alpha_0 x[n] + \alpha_1 x[n-1]$$



$y[n] = x[n] + x[n-1]$



$f_D x sr$



Εικόνα 3.6: Συχνотική απόκριση FIR φίλτρου κτένας με χρόνο καθυστέρησης = 0.25 ms ($f = 4$ KHz)

Το συγκεκριμένο φαινόμενο συχνотικών περιοχών έξαρσης και ακύρωσης οφείλεται στη χρονική διαφοροποίηση που υπάρχει ανάμεσα στα δύο σήματα (εισόδου και καθυστέρησης), προκαλώντας κατ' αυτόν τον τρόπο περιοδικές αποκλίσεις στη φάση τους, με συχνотικές περιοχές μέγιστου πλάτους (διαφορά φάσης 0°) και ακύρωσης (διαφορά φάσης 180°). Ο παρακάτω πίνακας είναι ενδεικτικός της σχέσης χρόνου καθυστέρησης και συχνотικής απόκρισης του φίλτρου (για $sr = 48$ KHz):

Χρόνος Καθυστέρησης (ms)	Πρώτη συχνотική έξαρση – Απόσταση συχνотικών εξάρσεων
20	50 Hz
10	100 Hz
2	500 Hz
1	1 KHz
0.5	2 KHz
0.25	4 KHz
0.125	8 KHz
0.1	10 KHz

Πίνακας 3.1: Σχέση χρόνου καθυστέρησης και συχνотικής απόκρισης ενός FIR φίλτρου κτένας

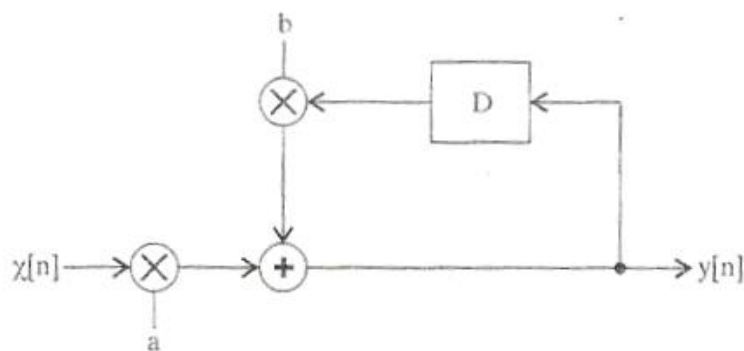
Αντίστοιχα, όταν στο ίδιο κύκλωμα έχουμε αφαίρεση των δειγμάτων:

$$y[n] = x[n] - x[n - D]$$

πρόκειται στην ουσία για την παραπάνω πρόσθεση με διαφορά φάσης 180° . Έτσι τα μηδενικά σημεία παρουσιάζονται πλέον στις συχνотότητες f , $2f$, $3f$ κ.ο.κ. ενώ τα σημεία έξαρσης στις τιμές $f/2$, $3f/2$, $5f/2$ κ.ο.κ.

3.3.2 Φίλτρα IIR Comb

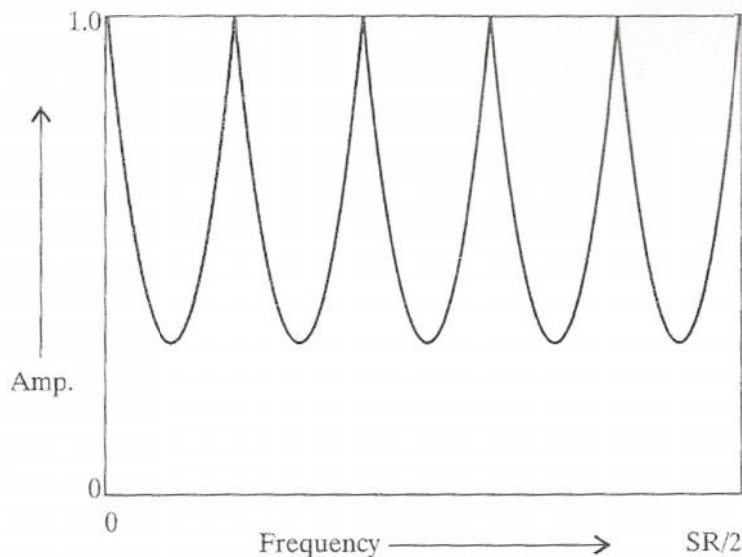
Ένα αναδρομικό φίλτρο κτένας (IIR Comb) ανατροφοδοτεί μέρος της εξόδου του στην είσοδό του:



Εικόνα 3.7: Ένα IIR φίλτρο κτένας

$$y[n] = (a x \chi[n]) + (b x y[n - D])$$

όπου στο τρέχον κάθε φορά δείγμα $\chi[n]$, προσθέτουμε μία προγενέστερη κατά χρόνο D τιμή της εξόδου του $y[n - D]$. Οι πολλαπλασιαστικοί συντελεστές a , b καθορίζουν την απόκριση του φίλτρου (τιμές από 0 έως 1). Για πολύ υψηλές τιμές του b , η ανατροφοδότηση γίνεται ολοένα και μεγαλύτερη με αποτέλεσμα τη δημιουργία παραμόρφωσης. Εάν ο χρόνος καθυστέρησης D είναι μικρότερος από 10 ms (και αναλόγως πάντα τη συχνότητα δειγματοληψίας) δημιουργούνται μία σειρά από διαδοχικές φασματικές εγκοπές (ύψη και βάθη) ίσης συχνοτικής απόστασης, με ανεστραμμένες κοιλότητες:



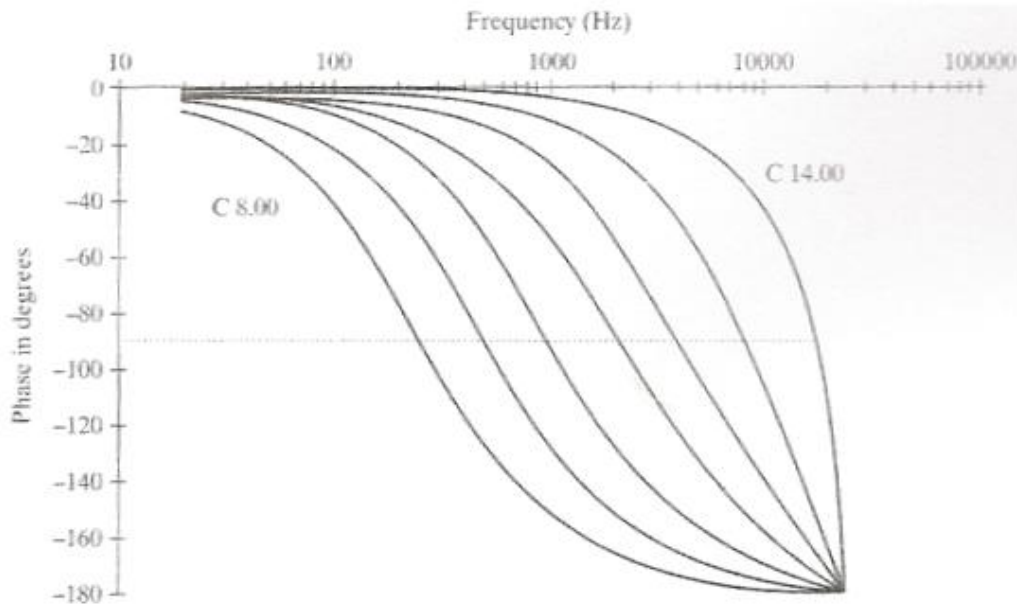
Εικόνα 3.8: Συχνοτική απόκριση IIR φίλτρου κτένας

Τα αναδρομικά φίλτρα λειτουργούν γενικά με αναδρομή ή ανατροφοδότηση προγενέστερων τιμών εξόδου τους. Αυτό έχει σαν αποτέλεσμα να χρησιμοποιούνται λιγότεροι συντελεστές και να χρειάζονται κατά συνέπεια λιγότεροι υπολογισμοί. Επειδή ακριβώς τα φίλτρα αυτά λειτουργούν αναδρομικά, η χρήση προγενέστερων τιμών μπορεί να είναι δυνητικά απεριόριστη (infinite).

3.4 Allpass Filters (Φίλτρα Γενικής Διέλευσης)

Όλα τα φίλτρα που εξετάσαμε μέχρι τώρα επεμβαίνουν σε ένα σήμα μεταβάλλοντας τα πλάτη των επιμέρους συχνοτήτων του, παρουσιάζοντας έτσι συγκεκριμένη συχνοτική απόκριση. Αντίθετα, ένα φίλτρο γενικής διέλευσης επεμβαίνει σε ένα σήμα μεταβάλλοντας τη φάση των

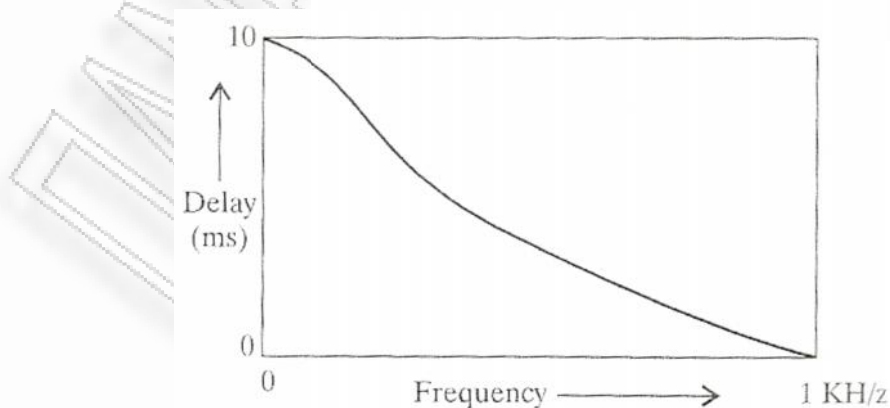
επιμέρους συχνοτήτων από τις οποίες αυτό αποτελείται, παρουσιάζοντας έτσι απόκριση φάσης (phase response). Η διολίσθηση φάσης (phase shift) στη συχνότητα αποκοπής του φίλτρου αντιστοιχεί σε -90° . Για συχνότητες μικρότερες της συχνότητας αποκοπής, η διολίσθηση φάσης σταδιακά ελαττώνεται σε 0° , ενώ για μεγαλύτερες, σταδιακά αυξάνεται σε -180° .



Εικόνα 3.9: Διάγραμμα φίλτρου γενικής διέλευσης

Η παραπάνω εικόνα αποτελεί τυπικό διάγραμμα φίλτρου γενικής διέλευσης με επτά διαφορετικές αποκρίσεις φάσης για επτά αντίστοιχες συχνότητες αποκοπής, ξεκινώντας από το C μεσαίο ($C 8.00 = 262 \text{ Hz}$) και καταλήγοντας στο C έξι οκτάβες υψηλότερα ($C 14.00 = 16768 \text{ Hz}$). Η διακεκομμένη γραμμή αντιστοιχεί σε διολίσθηση φάσης -90° , στη συχνότητα δηλαδή αποκοπής

Η σχεδίαση αυτή υλοποιείται εφαρμόζοντας διαφορετικούς χρόνους καθυστέρησης σε διαφορετικές περιοχές του φάσματος, λαμβάνοντας πάντα υπ' όψη πως, σε γενικές γραμμές, η διολίσθηση φάσης του φίλτρου μεταβάλλεται λογαριθμικά σε σχέση με το χρόνο καθυστέρησης. Έτσι για παράδειγμα, ενώ 0.5 ms καθυστέρησης σε μία χαμηλή συχνότητα αντιστοιχεί σε μικρή διαφορά φάσης, στα 1 KHz ο χρόνος αυτός ισοδυναμεί σε διαφορά φάσης -180° (περίοδος = 1 ms).



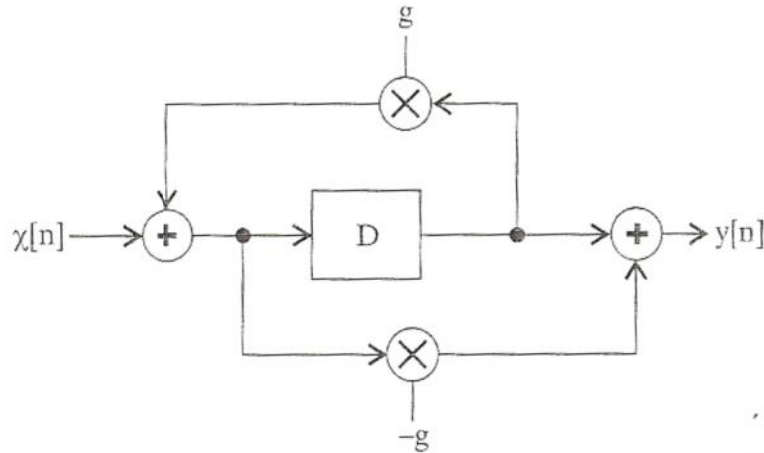
Εικόνα 3.10: Απόκριση φάσης ενός φίλτρου γενικής διέλευσης σε σχέση με τον χρόνο καθυστέρησης

Καθώς αυξάνονται οι τιμές των συχνοτήτων (άξονας χ), παρατηρούμε την λογαριθμική μεταβολή στον χρόνο καθυστέρησης (άξονας y).

Ο γενικός τύπος του φίλτρου αυτού είναι:

$$y[n] = (-g \times \chi[n]) + \chi[n - D] + (g \times y[n - D])$$

με αντίστοιχο διάγραμμα:

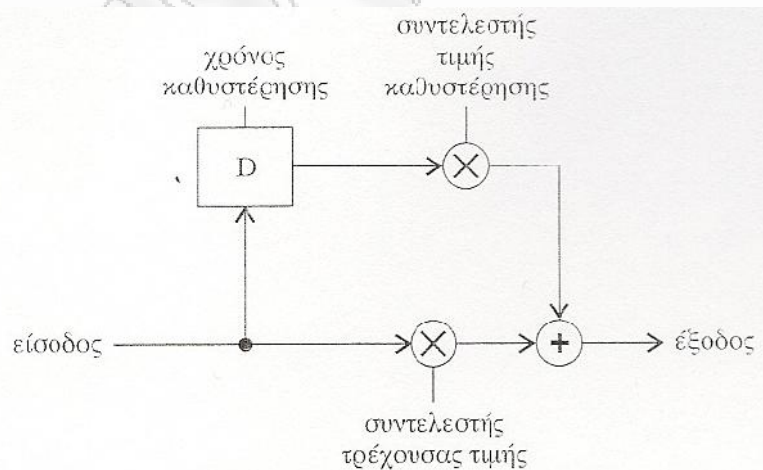


Εικόνα 3.11: Φίλτρο γενικής διέλευσης

Παρατηρούμε πως ένα τέτοιο φίλτρο μοιάζει με το IIR Comb φίλτρο, με τη διαφορά πως κάθε φορά προστίθεται ένα δείγμα εισόδου με καθυστέρηση D ($\chi[n - D]$). Επειδή μάλιστα $a = -b$ (συντελεστές g και $-g$), η αφαίρεση αυτή ακυρώνει το φασματικό αποτέλεσμα ενός comb φίλτρου, διατηρώντας όμως τα χαρακτηριστικά της καθυστέρησης.

3.5 Κύκλωμα Ψηφιακής Σειράς Καθυστέρησης (Digital Delay Line DDL Circuit)

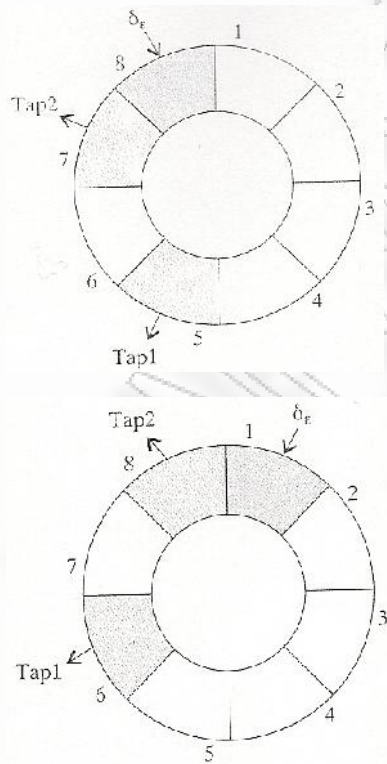
Ένα απλό Κύκλωμα Ψηφιακής Σειράς Καθυστέρησης (Digital Delay Line ή DDL Circuit) έχει την ακόλουθη διάταξη:



Εικόνα 3.12: Κύκλωμα ψηφιακής σειράς καθυστέρησης

Το κύκλωμα αυτό μοιάζει πολύ με αυτό του απλού ψηφιακού φίλτρου χαμηλής διέλευσης, καθώς και του απλού FIR Comb φίλτρου. Η διαφορά μεταξύ τους βρίσκεται στο χρόνο καθυστέρησης. Ένα απλό ψηφιακό φίλτρο χαμηλής διέλευσης καθυστερεί όπως είδαμε κατά ένα δείγμα και ένα απλό FIR Comb από 0.1 έως 1 ms. Γενικά, ένα DDL κύκλωμα έχει συνήθως χρόνους καθυστέρησης μεγαλύτερους από 1 ms. Κατ' αυτόν τον τρόπο εάν στην είσοδο του κυκλώματος θέσουμε ένα ηχητικό σήμα, στην έξοδο θα λάβουμε το αρχικό σήμα μαζί με το είδωλο του καθυστερημένο κατά χρόνο D .

D a 'sr



3.6 Φαινόμενα Σταθερού Χρόνου Καθυστέρησης

Τα φαινόμενα σταθερού χρόνου καθυστέρησης αφορούν φαινόμενα του ήχου (sound effects) τα οποία προκύπτουν, όταν μαζί με ένα αρχικό σήμα αναπαράγεται και η χρονική του καθυστέρηση κατά χρόνο D . Καθ' όλη τη διάρκεια του φαινομένου, ο χρόνος καθυστέρησης D παραμένει σταθερός. Τα φαινόμενα που παρουσιάζονται, προσδιορίζονται ανάλογα τη χρονική καθυστέρηση και χωρίζονται σε τρεις κύριες κατηγορίες:

- $D < 10$ ms
Εάν η καθυστέρηση αφορά από ένα έως μερικά δείγματα, παρουσιάζονται φαινόμενα όμοια με εκείνα που προκαλεί ένα φίλτρο FIR χαμηλής διέλευσης. Για μεγαλύτερες τιμές (από 0.1 ms έως 10 ms) παρουσιάζονται φαινόμενα φίλτρου κτένας.
- $10 \text{ ms} < D < 50$ ms
Στην περίπτωση αυτή το υπό καθυστέρηση σήμα δημιουργεί ένα δεύτερο είδωλο, χρονικά αρκετά κοντινά με το αρχικό, δίνοντας έτσι την αίσθηση ενός πιο γεμάτου σε όγκο ήχου, χωρίς να επηρεάζεται η συνολική ένταση. Το φαινόμενο αυτό χρησιμοποιείται συχνά στην μουσική παραγωγή (πχ. στη φωνή), προκειμένου να εμπλουτίζεται περισσότερο ο αρχικός ήχος.
- $D > 50$ ms
Εδώ το ανθρώπινο αυτί αντιλαμβάνεται ξεκάθαρα τη χρονική διαφοροποίηση των δύο σημάτων, καταλήγοντας σε φαινόμενα επανάληψης του αρχικού σήματος (**αντίλαλος** ή echoes). Έτσι, για καθυστέρηση ίση με 50 ms, έχουμε επανάληψη της πηγής σε μία ανακλώμενη επιφάνεια από ισοδύναμη απόσταση $8.5 = (334 * 0.5) / 2$) περίπου μέτρων.

3.7 Φαινόμενα Μεταβλητού Χρόνου Καθυστέρησης

Τα φαινόμενα μεταβλητού χρόνου καθυστέρησης αφορούν φαινόμενα του ήχου τα οποία προκύπτουν όταν σε μία σειρά καθυστέρησης ο χρόνος καθυστέρησης D μεταβάλλεται διαρκώς. Τα φαινόμενα αυτά ποικίλλουν ανάλογα τον τύπο της σειράς καθυστέρησης, καθώς και τα επιμέρους χαρακτηριστικά της. Φαινόμενα τέτοια αποτελούν το flanging, το phasing, το chorus effect, το vibrato και το pitch-shifter/harmonizer.

i. Flanging

Ο πρώτος που δοκίμασε την τεχνική αυτή ήταν ο Les Paul (1948), όταν για την ίδια πηγή ήχου χρησιμοποίησε ταυτόχρονα δύο συστήματα εγγραφής, το ένα εκ των οποίων με μεταβλητή ταχύτητα. Αργότερα, στα studio ηχογραφήσεων τη δεκαετία του 1960, καθώς ηχογραφούσαν την ίδια πηγή σε δύο διαφορετικά μαγνητόφωνα, μετέβαλαν την ταχύτητα του ενός πιέζοντας ελαφρά τη φλάντζα του καρουλιού της ταινίας (reel flange), γεγονός που καθιέρωσε και τον όρο flanging. Το ίδιο αποτέλεσμα μπορεί να επιτευχθεί εάν σε μία ψηφιακή σειρά καθυστέρησης μεταβάλλουμε διαρκώς το χρόνο καθυστέρησης D , μέσω ενός ταλαντωτή χαμηλών συχνοτήτων (LFO) ημιτονοειδούς ή τριγωνικής κυματομορφής. Το αντίστοιχο κύκλωμα που δημιουργεί μία τέτοια έξοδο είναι ένα FIR comb φίλτρο με μεταβλητό D . Αυτό έχει σαν αποτέλεσμα στο φάσμα εξόδου να έχουμε διαρκώς μεταβαλλόμενες συχνοτικές εξάρσεις και εγκοπές (sweeping comb filter). Άλλοι σχεδιασμοί χρησιμοποιούν IIR comb φίλτρα με μεταβλητό χρόνο καθυστέρησης D .

ii. Phasing

Το phasing υλοποιείται με την χρήση πολλαπλών φίλτρων γενικής διέλευσης (allpass filters), καθένα απ' τα οποία έχει μεταβλητό χρόνο καθυστέρησης D . Με αυτόν τον τρόπο, η φασματική διολίσθηση κάθε φίλτρου γενικής διέλευσης μεταβάλλεται διαρκώς. Οι εξοδοί των φίλτρων προστίθενται με το αρχικό σήμα δημιουργώντας έτσι ένα παρόμοιο αποτέλεσμα με αυτό του flanging (μεταβαλλόμενου φίλτρου κτένας). Σε αντίθεση με το flanging όπου οι εξάρσεις και οι εγκοπές έχουν σταθερή απόσταση μεταξύ τους και το πλήθος τους εξαρτάται από το D , στο phasing το πλήθος των εγκοπών εξαρτάται από το πλήθος των φίλτρων γενικής διέλευσης, ενώ η απόσταση μεταξύ τους μπορεί επίσης να μεταβάλλεται.

iii. Chorus

Το chorus effect πρωτοεμφανίστηκε στη δεκαετία του 1940 και αφορούσε τον εμπλουτισμό ενός αρχικού σήματος με πολλαπλά είδωλά του, προσδίδοντας έτσι μία πιο πλούσια αίσθηση του αρχικού σήματος. Οι τρόποι που μπορεί να επιτευχθεί αυτό

ποικίλλουν (πολλαπλές ελαφρές καθυστερήσεις του αρχικού σήματος, αθροίσεις κοντινών συχνοτήτων με δημιουργία πολλαπλών διακροτημάτων, ασύγχρονο vibrato). Το chorus effect μπορεί να πραγματοποιηθεί εάν στείλουμε το αρχικό σήμα σε μία ψηφιακή σειρά καθυστέρησης πολλαπλών σημείων διαβάσματος (multi-tap delay line), όπου οι χρόνοι καθυστέρησης μεταβάλλονται διαρκώς μέσα από ένα περιορισμένο πεδίο τιμών.

iv. Vibrato

Το vibrato μπορεί να επιτευχθεί με τη χρήση μιας LFO ημιτονοειδούς κυματομορφής, η οποία μεταβάλλει περιοδικά τον χρόνο καθυστέρησης γύρω από έναν σταθερά μετακινούμενο δείκτη διαβάσματος. Το πλάτος ταλάντωσης του ταλαντωτή ορίζει το βάθος του vibrato (depth) και η συχνότητά του, την ταχύτητα του vibrato (rate).

v. Pitch-Shifter / Harmonizer

Τα φαινόμενα αυτά ανήκουν στην κατηγορία τεχνικών μετατόπισης της τονικότητας ενός σήματος, χωρίς να μεταβάλλεται αντίστοιχα η χρονική διάρκεια. Το πρώτο εμπορικό ψηφιακό harmonizer πραγματικού χρόνου (real time) εμφανίστηκε για πρώτη φορά στα μέσα της δεκαετίας του 1970. Η βασική αρχή λειτουργίας του στηρίζεται στο ότι ένα σήμα εισόδου ψηφιοποιείται με μία συχνότητα δειγματοληψίας SR_{in}, ενώ μια δεύτερη συχνότητα δειγματοληψίας SR_{out} εξάγει τα δείγματα του σήματος στην έξοδο. Ο λόγος των δύο συχνοτήτων SR_{in}/SR_{out} καθορίζει την τονική μετατόπιση του σήματος.

4 Η γλώσσα προγραμματισμού Matlab

4.1 Γενικά Στοιχεία

Το **MATLAB** είναι μία γλώσσα υψηλού επιπέδου, τέταρτης γενιάς (fourth-generation programming language), που χρησιμοποιείται για τεχνικούς υπολογισμούς, με ένα περιβάλλον αριθμητικών υπολογισμών (numerical computing environment). Προκύπτει από τα αρχικά των λέξεων **MAT**rix **LAB**oratory και η λειτουργία του βασίζεται κυρίως στη χρήση πινάκων και διανυσμάτων, τα στοιχεία των οποίων μπορεί να είναι πραγματικοί ή μιγαδικοί αριθμοί. Αναπτυγμένη από την εταιρεία MathWorks, το MATLAB επιτρέπει πράξεις πινάκων (matrix manipulations), γραφήματα συναρτήσεων και δεδομένων, υλοποίηση αλγορίθμων, δημιουργία διεπιφάνειας χρηστών (user interfaces), και διεπιφάνειας με προγράμματα γραμμένα σε άλλες γλώσσες, συμπεριλαμβανομένων της C, C++, και Fortran. Το Matlab είναι ένα διαδραστικό σύγχρονο εργαλείο του οποίου βασικό στοιχείο είναι η διάταξη (array). Αυτό μας επιτρέπει τη λύση πολλών υπολογιστικών προβλημάτων, ιδιαίτερα όσων περιλαμβάνουν σχηματισμούς πινάκων και διανυσμάτων σε χρόνο μικρότερο από ότι απαιτούν οι γλώσσες προγραμματισμού Fortran και C.

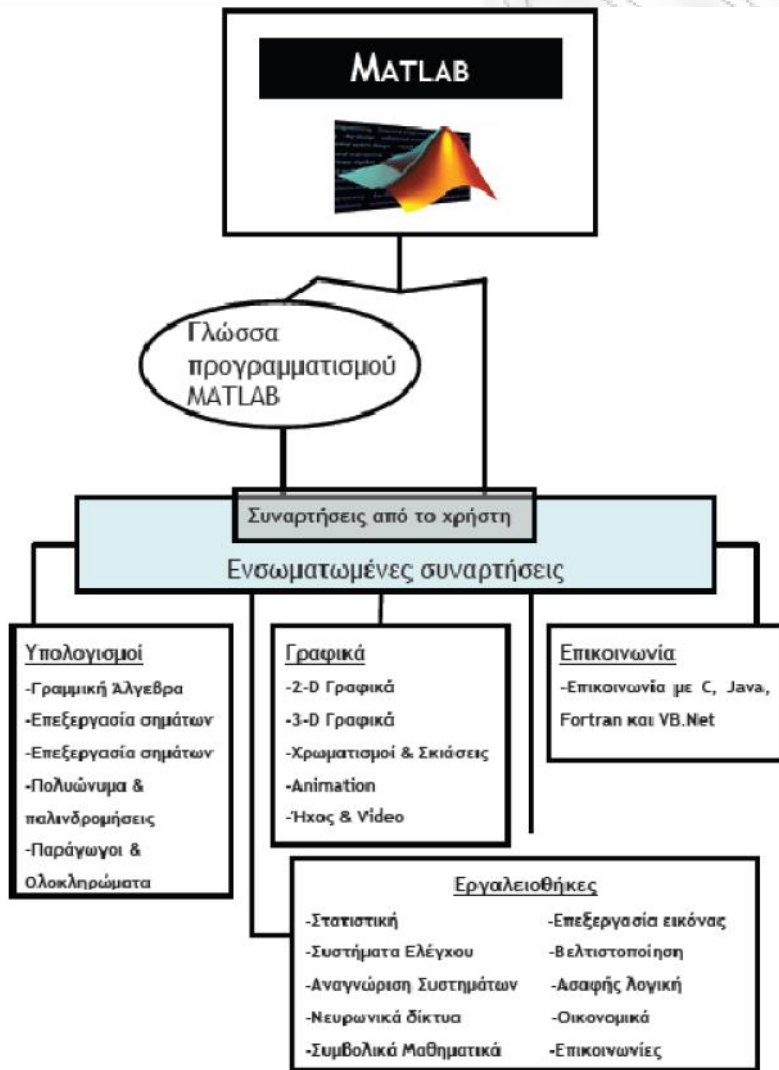
Παρόλο που το MATLAB αποσκοπεί πρωταρχικά στον αριθμητικό υπολογισμό, ένα προαιρετικό toolbox χρησιμοποιεί το MuPAD symbolic engine, που επιτρέπει την πρόσβαση σε δυνατότητες συμβολικού προγραμματισμού (symbolic computing). Ένα επιπρόσθετο πακέτο, το Simulink, προσθέτει graphical multi-domain προσομοίωση και Model-Based Design για δυναμικά και ενσωματωμένα (embedded) συστήματα. Το 2004, το MATLAB είχε περί τους ένα εκατομμύριο χρήστες στη βιομηχανία και τα εκπαιδευτικά ιδρύματα. Οι χρήστες του MATLAB έχουν ποικίλα υπόβαθρα, όπως μηχανικοί, θεωρητικοί επιστήμονες και οικονομολόγοι. Μεταξύ αυτών των χρηστών είναι τα ακαδημαϊκά και ερευνητικά ινστιτούτα όπως το Massachusetts Institute of Technology, NASA, Max Planck Society, και RWTH Aachen University καθώς και βιομηχανικές/εμπορικές επιχειρήσεις όπως ABB Group, Boeing, Caterpillar Inc., Ford Motor, Halliburton, Lockheed Martin, Motorola, Novartis, Pfizer, Philips, Toyota, και UnitCredit Bank.

4.2 Ιστορικά Στοιχεία

Το MATLAB δημιουργήθηκε στα τέλη του 1970 από τον Cleve Moler, τότε Πρόεδρο του Τμήματος Επιστήμης των Υπολογιστών (Computer Science) του Πανεπιστημίου του New Mexico. Το σχεδίασε έτσι ώστε να δώσει στους μαθητές του πρόσβαση στο LINPACK (λογισμικό γραμμικής άλγεβρας: επίλυση γραμμικών συστημάτων, κλπ) και EISPACK (λογισμικό εύρεσης ιδιοτιμών, ιδιοανυσμάτων, κλπ) χωρίς να έχουν την ανάγκη εκμάθησης της γλώσσας Fortran στην οποία είναι γραμμένα τα εν λόγω (ιστορικά) πακέτα. Σύντομα διαδόθηκε σε άλλα πανεπιστήμια και βρήκε πολλούς θιασώτες στην κοινότητα των εφαρμοσμένων μαθηματικών. Ο μηχανικός Jack Little μυήθηκε σε αυτό κατά τη διάρκεια μιας επίσκεψης του Moler στο Πανεπιστήμιο του Stanford το 1983. Αναγνωρίζοντας τον εμπορικό δυναμισμό του λογισμικού, συνενώθηκε με τον Moler και τον Steve Bangert. Αυτοί ξανάγραψαν το MATLAB σε γλώσσα C και ίδρυσαν την εταιρεία MathWorks το 1984 με σκοπό να συνεχίσουν την ανάπτυξη και εμπορευματοποίησή του. Αυτές οι ξαναγραμμένες βιβλιοθήκες είναι γνωστές με το όνομα JACKPAC. Το 2000, το MATLAB ξαναγράφηκε έτσι ώστε να χρησιμοποιεί ένα νεότερο σύνολο βιβλιοθηκών του πακέτου χειρισμού πινάκων, LAPACK (το οποίο σήμερα έχει πλέον αντικαταστήσει το LINPACK, εννοούμε στις βιβλιοθήκες Fortran).

Το MATLAB υιοθετήθηκε αρχικά από τους μηχανικούς που ασχολούνταν με τον Αυτόματο Έλεγχο, που ήταν η ειδικότητα του Jack Little, αλλά σύντομα επεκτάθηκε και σε άλλους τομείς. Σχεδιάστηκε εξ αρχής για μαθηματικούς σκοπούς, για να κάνει δυνατή την υπολογιστική επίλυση απλών και σύνθετων μαθηματικών προβλημάτων. Από τότε έχει αναπτυχθεί αρκετά και χρησιμοποιείται κυρίως για μαθηματικούς υπολογισμούς, ανάπτυξη αλγορίθμων, προσομοίωση, μοντελοποίηση και προτυποποίηση, έτσι ώστε να γίνει ένα ισχυρότατο εργαλείο στην οπτικοποίηση, στον προγραμματισμό, στην έρευνα, στην βιομηχανία και στις επικοινωνίες. Σήμερα χρησιμοποιείται για την εκπαίδευση, ειδικότερα της Γραμμικής Άλγεβρας και της Αριθμητικής Ανάλυσης, και είναι δημοφιλές μεταξύ των επιστημόνων που ασχολούνται με την επεξεργασία εικόνων (π.χ. ακόμη και για την φύλαξη κτιρίων έναντι νυκτερινών εισβολέων) αλλά και την επεξεργασία ήχου.

ΠΕΡΑΙΑ



4.4 Στοιχεία του Matlab

Το σύστημα του Matlab αποτελείται από πέντε βασικά μέρη.

- **Περιβάλλον ανάπτυξης.**
Είναι ένα σύνολο από εργαλεία που βοηθάνε στην χρήση των συναρτήσεων και των αρχείων του Matlab. Τα εργαλεία αυτά συμπεριλαμβάνουν την επιφάνεια εργασίας του Matlab (Matlab desktop), τη γραμμή εργασιών (command window), το ιστορικό εντολών (command history), έναν κειμενογράφο (editor) και έναν διορθωτή (debugger), έναν περιηγητή για τη βοήθεια, το χώρο εργασίας (workspace), τα αρχεία και την αναζήτηση.
- **Βιβλιοθήκη μαθηματικών συναρτήσεων του Matlab.**
Είναι μια τεράστια συλλογή από υπολογιστικούς αλγορίθμους που καλύπτουν ένα ευρύτατο φάσμα που ξεκινά από στοιχειώδεις συναρτήσεις, όπως το άθροισμα, το ημίτονο, και εκτείνεται σε πιο σύνθετες συναρτήσεις, όπως αντιστροφή πίνακα, συναρτήσεις Bessel και μετασχηματισμούς Fourier.
- **Γλώσσα του Matlab.**
Είναι μια υψηλού επιπέδου γλώσσα με δηλώσεις δομής ελέγχου, δομές δεδομένων, συναρτήσεις, εισόδους/εξόδους και χαρακτηριστικά αντικειμενοστραφούς προγραμματισμού.
- **Γραφικά**
Το Matlab μπορεί με ευκολία να αναπαραστήσει γραφικά διανύσματα και πίνακες και δίνει τη δυνατότητα σχολιασμού και εκτύπωσης αυτών των γραφημάτων. Περιλαμβάνει συναρτήσεις υψηλού επιπέδου για απεικόνιση διανυσμάτων σε διαγράμματα δυο και τριών διαστάσεων, επεξεργασία εικόνας, δυναμική κίνηση ομοιωμάτων και γραφικά παρουσίασης. Περιλαμβάνει επιπλέον συναρτήσεις χαμηλού επιπέδου που επιτρέπουν στο χρήστη να προσαρμόσει πλήρως την εμφάνιση των γραφημάτων.
- **Περιβάλλον εφαρμογής προγραμμάτων (Application Program Interface - API)**
Είναι μια βιβλιοθήκη που επιτρέπει στο χρήστη να γράψει κώδικα στις γλώσσες προγραμματισμού C και Fortran που αλληλεπιδρούν με το Matlab. Δίνει τη δυνατότητα κλήσης υπορουτίνων από το Matlab (dynamic linking), κλήση του Matlab ως υπολογιστικής μηχανής και για την ανάγνωση και ανάπτυξη αρχείων τύπου MAT (MAT-files).

4.5 Η οργάνωση του Matlab

Το Matlab είναι ένα «υπολογιστικό εργαλείο» μέσω του οποίου μπορεί να εκτελεστεί μεγάλος αριθμός υπολογισμών με τη χρήση ελάχιστων εντολών. Αυτό επιτυγχάνεται με τη χρήση εντολών-συναρτήσεων οι οποίες είτε βρίσκονται ήδη αποθηκευμένες στη μνήμη του υπολογιστή είτε κατασκευάζονται από τον ίδιο το χρήστη ανάλογα με το πρόβλημα που τον ενδιαφέρει να επιλύσει. Η ευρεία χρήση του MATLAB οφείλεται σε μεγάλο βαθμό στην επεκτασιμότητα της.

Οι βιβλιοθήκες τις οποίες περιέχει το DVD του Matlab είναι οι εξής:

- **Simulink:** Το Simulink είναι ένα λογισμικό πακέτο που επιτρέπει τη μοντελοποίηση, την προσομοίωση και την ανάλυση δυναμικών συστημάτων. Υποστηρίζει γραμμικά και μη γραμμικά συστήματα, μοντελοποιημένα σε συνεχή ή διακριτό χρόνο, ή ακόμη και υβριδικά συστήματα (εν μέρει μοντελοποιημένα σε συνεχή και εν μέρει σε διακριτό χρόνο).
- **Control System Toolbox:** Παρέχει τη δυνατότητα να χρησιμοποιηθεί το Matlab για σχεδίαση και ανάλυση συστημάτων αυτομάτου ελέγχου.
- **Fuzzy Logic Toolbox:** με την εργαλειοθήκη ασαφούς λογικής (*fuzzy toolbox*) δίνεται στους χρήστες του MATLAB η δυνατότητα σχεδιασμού ασαφών συστημάτων.
- **Signal Processing Toolbox:** Η εργαλειοθήκη επεξεργασίας σήματος είναι μια συλλογή εργαλείων που στηρίζεται στο αριθμητικό υπολογιστικό περιβάλλον του MATLAB®. Η εργαλειοθήκη υποστηρίζει ένα ευρύ φάσμα διαδικασιών επεξεργασίας σήματος.

4.6 Λειτουργία του Matlab

Η γλώσσα προγραμματισμού MATLAB λειτουργεί ως διερμηνέας εντολών. Οι εντολές αυτές μπορεί να είναι:

1. ορισμοί μεταβλητών και πράξεις.
2. κλήση ενσωματωμένων συναρτήσεων της MATLAB και των εγκατεστημένων εργαλειοθηκών της (toolboxes).
3. κλήση συναρτήσεων (functions) ή αρχείων εντολών MATLAB (scripts) που κατασκευάζονται από τους χρήστες με τη μορφή m-file. Οι πίνακες (*matrices*) και τα διανύσματα (*vectors*) θεωρούνται ως μονοδιάστατοι πίνακες .

4.7 Εισαγωγή στο Simulink

4.7.1 Τι είναι το Simulink

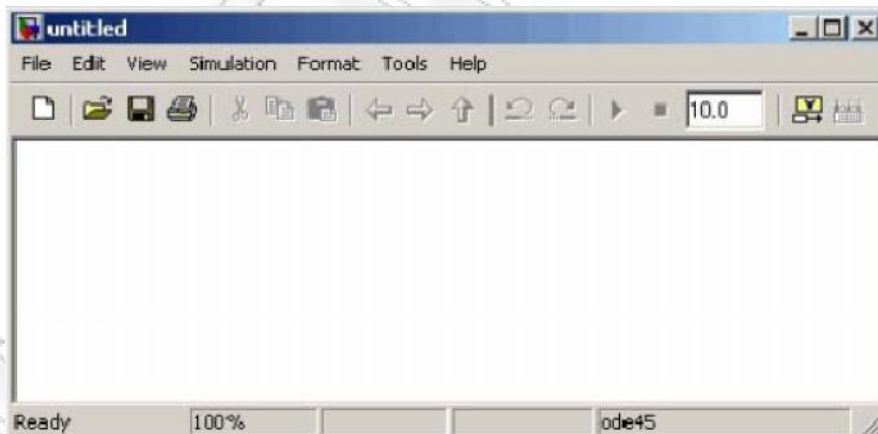
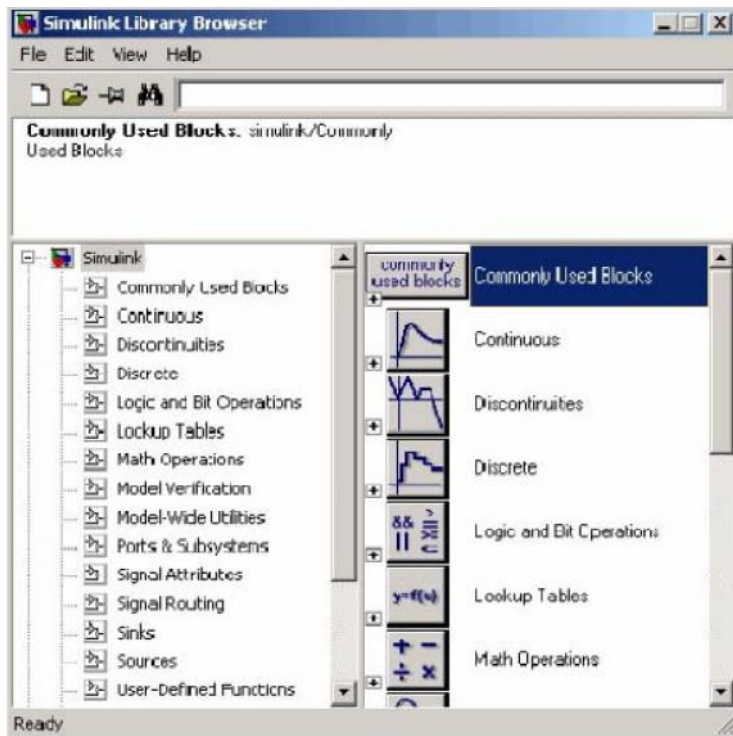
Το Simulink είναι ένα λογισμικό πακέτο για μοντελοποίηση, προσομοίωση και ανάλυση δυναμικών, γραμμικών, μη-γραμμικών, συνεχών, διακριτών, απλών αλλά και πολυμεταβλητών συστημάτων. Με τη βοήθεια του Simulink, μπορούμε εύκολα να χτίσουμε πρότυπα από την αρχή, ή να διαμορφώσουμε ένα ήδη υπάρχον και να το επεξεργαστούμε. Λειτουργεί σαν συνοδευτικό πρόγραμμα στο Matlab.. Το Simulink χρησιμοποιεί ένα διαδραστικό γραφικό περιβάλλον (Graphical User Interface - GUI) και ένα σύνολο βιβλιοθηκών, χρήσιμο για το σχεδιασμό, την προσομοίωση, την εφαρμογή και τον έλεγχο συστημάτων όπως συστήματα επικοινωνιών, επεξεργασίας σήματος και εικόνας, συστημάτων ελέγχου και πολλών άλλων.

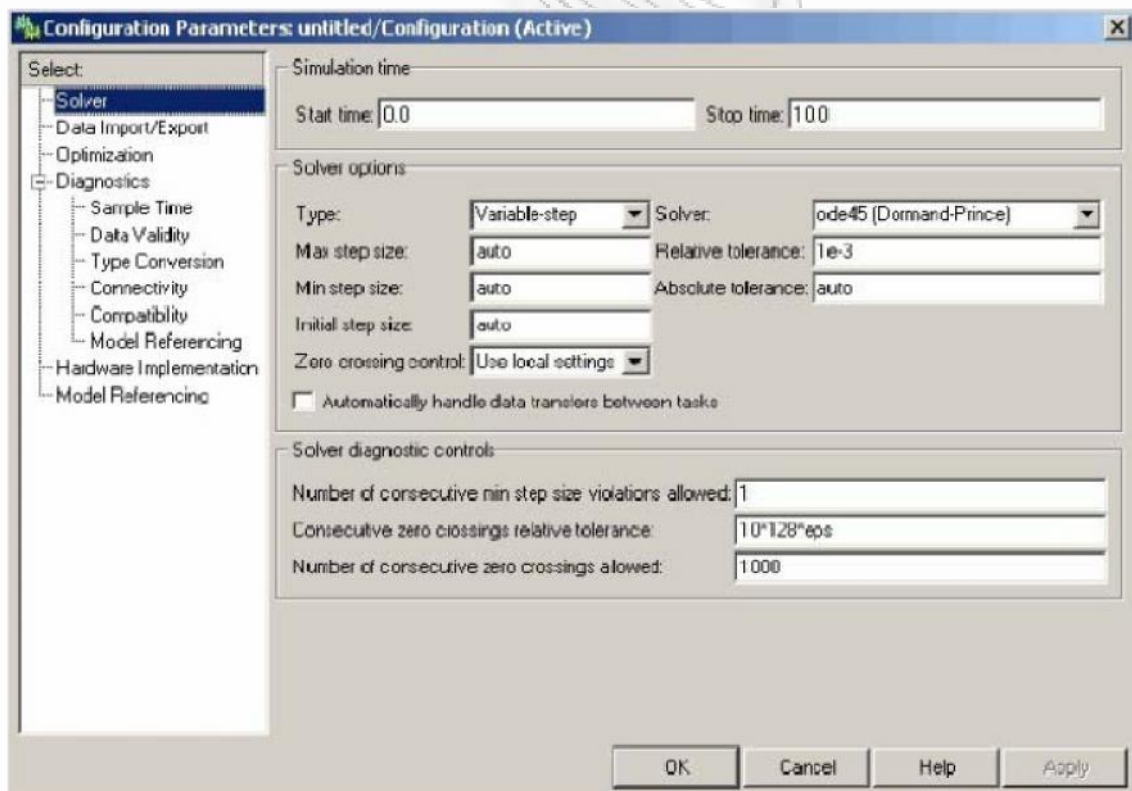
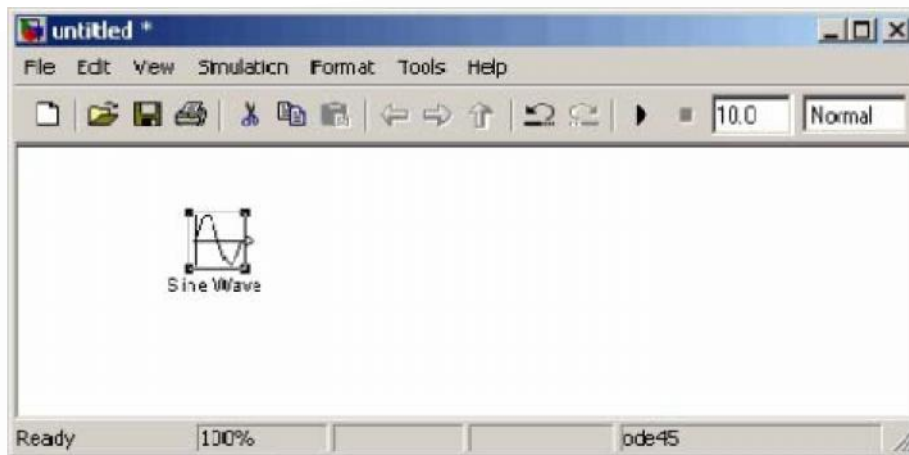
Οι βιβλιοθήκες περιέχουν τα δομικά στοιχεία των συστημάτων (blocks). Με αυτό μπορούμε να σχεδιάσουμε μοντέλα ,όπως θα μπορούσαμε με μολύβι και χαρτί. Τα μοντέλα του Simulink είναι ιεραρχικά δομημένα, δηλαδή κάθε μοντέλο αποτελείται από blocks τα οποία με τη σειρά τους μπορεί να περιέχουν άλλα blocks (υποσυστήματα). Αυτή η προσέγγιση μας βοηθά να κατανοήσουμε το πώς ένα μοντέλο οργανώνεται και πως τα στοιχεία του αλληλεπιδρούν μεταξύ τους. Ένα σύστημα που έχει ιεραρχική δομή μπορεί να ιδωθεί αρχικά σε υψηλό επίπεδο ως ένα σύνολο διασυνδεδεμένων υποσυστημάτων, κάθε ένα από τα οποία μοντελοποιείται ως ένα μπλοκ. Στη συνέχεια, κάνοντας διπλό κλικ με το ποντίκι στα επί μέρους μπλοκ, ο χρήστης μπορεί να κατέβει σε χαμηλότερα επίπεδα ώστε να δει αυξανόμενους βαθμούς λεπτομέρειας.

Μετά τον καθορισμό ενός μοντέλου, μπορούμε να το προσομοιώσουμε χρησιμοποιώντας μία επιλογή από μεθόδους ολοκλήρωσης, είτε από το μενού του Simulink είτε με την εισαγωγή εντολών στο παράθυρο εντολών του Matlab. Οι επιλογές του Simulink είναι ιδιαίτερα κατάλληλες για διαδραστική εργασία. Χρησιμοποιώντας παλμογράφους (scopes) και άλλα blocks μπορούμε να δούμε τα αποτελέσματα της προσομοίωσης, κατά την διάρκειά της. Επιπλέον μπορούμε να αλλάζουμε τις παραμέτρους και αμέσως να δούμε το καινούργιο αποτέλεσμα και την απόκλιση του από το αρχικό. Τα αποτελέσματα της προσομοίωσης μπορούν να τοποθετηθούν στο χώρο εργασίας του Matlab (workspace) για την μετέπειτα επεξεργασία και οπτικοποίηση τους. Επειδή το Matlab και το Simulink είναι ενσωματωμένα μπορούμε να προσομοιώσουμε, να αναλύσουμε και να αναθεωρήσουμε τα μοντέλα σε κάθε περιβάλλον, σε οποιοδήποτε σημείο.

Το Simulink περιλαμβάνει μία κατανοητή και περιεκτική βιβλιοθήκη από πηγές σημάτων (sources), σύνδεσμοι (sinks), γραμμικές (linear), μη γραμμικές (non-linear) συναρτήσεις. Μπορούμε ακόμη να προσαρμόσουμε και να δημιουργήσουμε τα δικά μας blocks.

Η εκκίνηση του Simulink γίνεται ως εξής: Στο παράθυρο του MATLAB, στη γραμμή εντολών εισάγεται η λέξη simulink και πιέζοντας enter εμφανίζεται το παράθυρο βιβλιοθηκών του simulink.







Βιβλιοθήκη

Sources

- περιέχει μπλοκ τα οποία είναι πηγές σημάτων (δεν έχουν είσοδο, παράγουν ως έξοδο ένα σήμα)

Sinks

- περιέχει μπλοκ τα οποία είναι στοιχεία «απορρόφησης» σημάτων (δεν έχουν έξοδο, δέχονται μόνο είσοδο)

Continuous

- περιέχει μπλοκ για τη μοντελοποίηση γραμμικών συστημάτων συνεχούς χρόνου

Δομικό στοιχείο

Constant

Step

Ramp

Pulse generator

Random number

Sine wave

Signal generator

Display

Scope

Stop Simulation

To Workspace

Derivative

Integrator

State space

Transfer function

Zero-pole

Λειτουργία

Σταθερά

Βηματική συνάρτηση

Συνάρτηση αναρρίχησης

Γεννήτρια παλμών

Γεννήτρια τυχαίου σήματος (κανονική κατανομή)

Γεννήτρια ημιτόνου

Γεννήτρια σημάτων (παράγει διάφορες κυματομορφές)

Οθόνη απεικόνισης τιμών

Παλμογράφος

Τερματισμός προσομοίωσης
Αποθήκευση στο χώρο εργασίας

Παραγωγή

Ολοκλήρωση

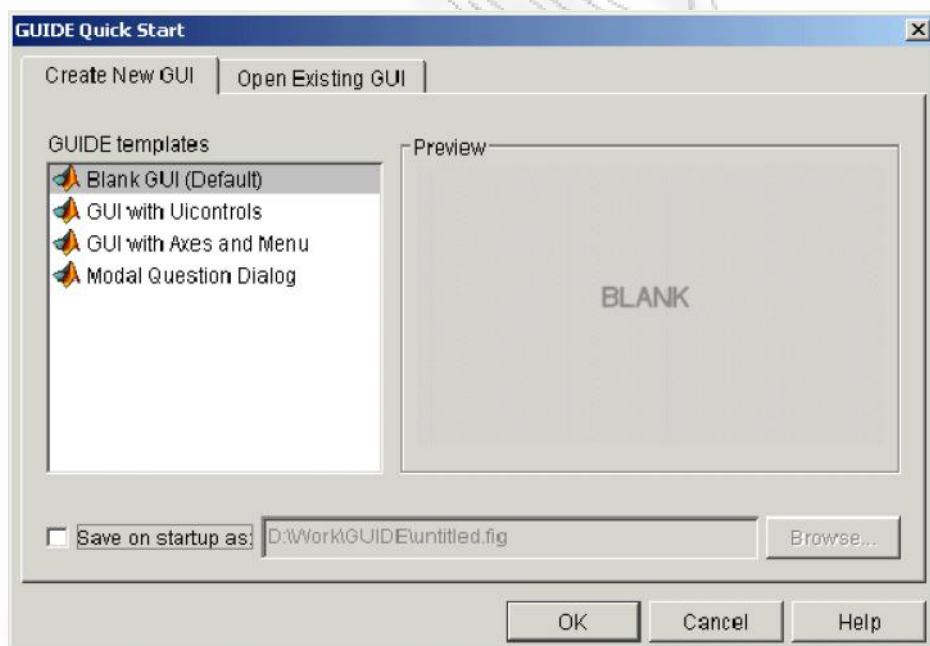
Μοντέλο συστήματος στο χώρο κατάστασης

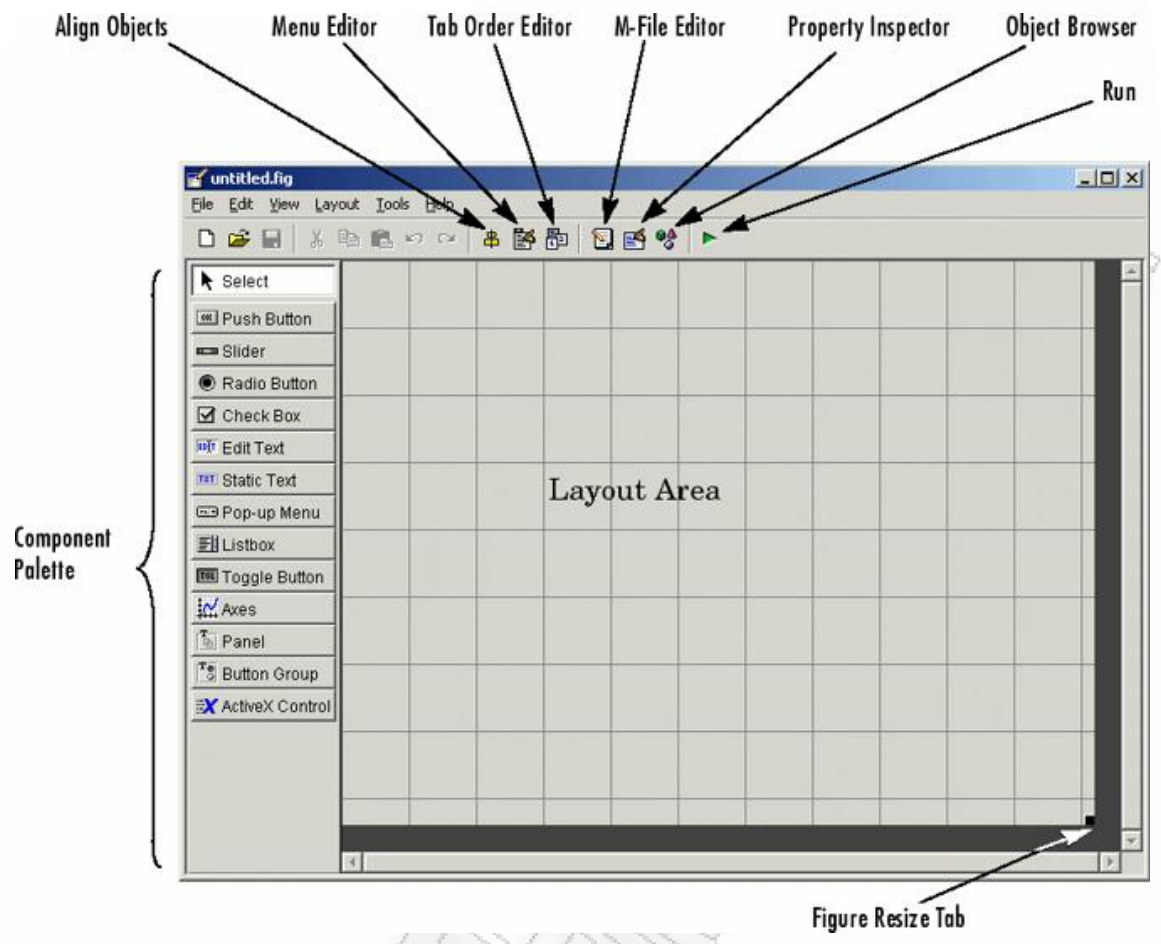
Μοντέλο συστήματος συνάρτησης μεταφοράς

Μοντέλο συστήματος πόλων – μηδενικών

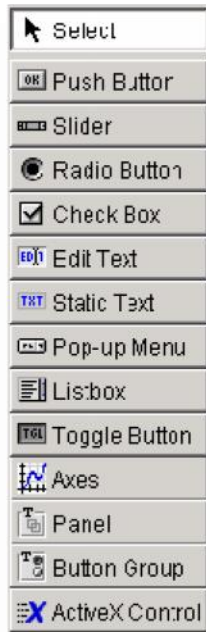
Βιβλιοθήκη	Δομικό στοιχείο	Λειτουργία
Nonlinear - περιέχει μπλοκ που μοντελοποιούν γραμμικά στοιχεία μη	Saturation	Στοιχείο κόρου
	Manual Switch	Χειροκίνητος διακόπτης
	Switch	Διακόπτης
	Quantizer	Κβαντιστής σήματος
Math - Περιέχει μπλοκ που μοντελοποιούν μαθηματικές πράξεις και συναρτήσεις	Abs	Απόλυτη τιμή
	Gain	Κέρδος
	Math function	Διάφορες μαθηματικές συναρτήσεις
	Matrix Gain	Πίνακας κερδών
	MinMax	Ελάχιστο ή μέγιστο
	Product	Πολλαπλασιασμός ή διαίρεση
	Rounding Function	Συνάρτηση στρογγύλευσης
	Sign	Εύρεση προσήμου
	Slider gain	Μεταβλητό κέρδος
	Sum	Άθροιση ή αφαίρεση
	Trigonometric function	Τριγωνομετρικές συναρτήσεις
Signals and Systems - Περιέχει στοιχεία διασύνδεσης σημάτων και συστημάτων	In1	Θύρα εισόδου υποσυστήματος ή μοντέλου
	Demux	Αποπλέκτης σημάτων
	Mux	Πολυπλέκτης σημάτων
	Terminator	Τερματισμός ασύνδετων σημάτων (δέχεται μόνο είσοδο)
	Out1	Θύρα εξόδου υποσυστήματος ή μοντέλου
Control Systems Toolbox - Περιέχει στοιχεία μοντελοποίησης συστημάτων ελέγχου	LTI System	Μοντελοποίηση γραμμικού χρονικά αναλλοίωτου συστήματος με διάφορους τρόπους (μέσω συνάρτησης μεταφοράς, στο χώρο κατάστασης, αναπαράσταση πόλων – μηδενικών)

Πίνακας 4.1: Οι βασικές βιβλιοθήκες δομικών στοιχείων του Simulink





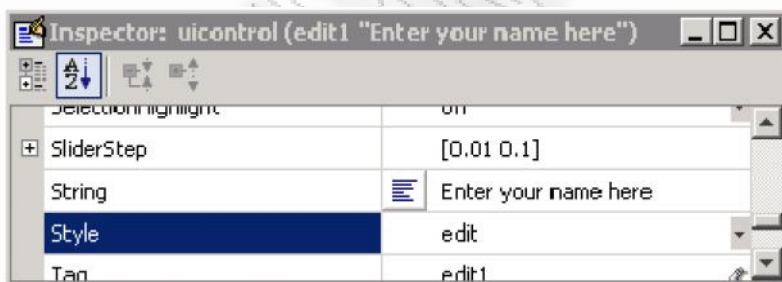
PAWELCZAK



Component palette with names



Component palette without names



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΔΑΛΙΑΣ

Μέρος Δεύτερο: Πρακτικό μέρος

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΔΑΛΙΑΣ

5 Ανάλυση μουσικών εφέ και σχεδιασμός αλγορίθμων

5.1 Εισαγωγή

Το κεφάλαιο αυτό ασχολείται με την ανάλυση των μουσικών εφέ τα οποία θα εφαρμοστούν στο λογισμικό που δημιουργήσαμε, καθώς και στον σχεδιασμό των αλγορίθμων για την προσομοίωση αυτών. Μερικά από τα πιο δημοφιλή εφέ επεξεργασίας ήχου έχουν περιληφθεί:

- Amplifier: Ενισχύει το σήμα το οποίο βρίσκεται σε χαμηλά επίπεδα τάσης προκειμένου να μπορεί να οδηγήσει τα μεγάφωνα ώστε να μπορέσουν με τη σειρά τους, να αποδώσουν ικανοποιητικού επιπέδου ακουστική ισχύ.
- Equalizer: Ενισχύει ή εξασθενεί ένα ή περισσότερα μέρη του ηχητικού φάσματος, μπορεί να αλλάξει το ηχόχρωμα ενός οργάνου και ο ήχος μπορεί να προσαρμοστεί σε συγκεκριμένες συνθήκες δωματίου.
- Delay: Δημιουργεί μία ή περισσότερες καθυστερημένες εκδόσεις του αρχικού σήματος εισόδου οι οποίες προστίθενται στο αρχικό σήμα, εμπλουτίζοντας έναν τυπικό και στεγνό ήχο, δίνοντας την αίσθηση στον ακροατή πως ακούει ανακλώμενο ήχο από τον χώρο στον οποίο βρίσκεται.
- Reverb: Δημιουργεί μια ουρά αντήχησης με βάση το εισερχόμενο σήμα, αποτελούμενη από ένα σύνολο ηχώ οι οποίες εξασθενούν και πυκνώνουν με την πάροδο του χρόνου.
- Flanger: Διαμορφώνει το ύψος και το χρόνο ενός σήματος ώστε να παράγει έναν διαφορετικό ήχο, γνωστό για τον αεροπλανικό τύπου "jet plane" ήχο που παράγει.

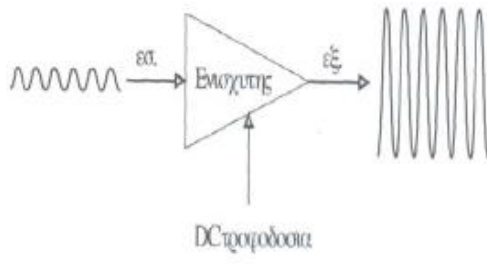
Κάθε αλγόριθμος εισάγεται και αντιμετωπίζεται ανεξάρτητα ως προς το θεωρητικό του μέρος, το σχεδιασμό του και την υλοποίησή του στο λογισμικό πακέτο Simulink του Matlab. Τα εφέ αυτά μπορούν να συνδυαστούν με οποιονδήποτε επιθυμητό τρόπο, επιτρέποντας διαφορετικές διαδρομές του ηχητικού σήματος και επομένως τελείως διαφορετικούς ήχους.

5.2 Ενισχυτής ήχου(Amplifier)

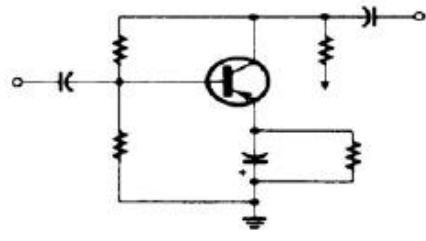
5.2.1 Τι είναι ο ενισχυτής ήχου

Η διάταξη, που στην είσοδό της δέχεται ένα σήμα ακουστικής συχνότητας (20Hz έως 20KHz) και στην έξοδό της παρέχει ένα άλλο σήμα με πολύ μεγαλύτερο κατ' αρχήν πλάτος και ίδια ή ανάλογα τα υπόλοιπα χαρακτηριστικά του προς το σήμα εισόδου, ονομάζεται **ενισχυτής ακουστικών συχνοτήτων ή ενισχυτής ήχου (audio amplifier)**. Αυτό το καταφέρνει με το να πάρει ισχύ από ένα τροφοδοτικό και να ελέγξει την έξοδό του ώστε αυτή να ταιριάζει με την είσοδο.

Ο ενισχυτής είναι το κύκλωμα που αυξάνει το "μέγεθος", δηλαδή την ισχύ ενός σήματος. Ρόλος του είναι να ενισχύσει το εναλλασσόμενο ηλεκτρικό σήμα του ήχου που δέχεται στην είσοδό του, χωρίς όμως να το αλλοιώσει. Η ενίσχυση αυτή είναι απαραίτητη γιατί το ασθενές σήμα δεν είναι ικανό να διεγείρει το ηχείο. Για να λειτουργήσει απορροφά (δηλ. καταναλώνει) ηλεκτρική ενέργεια από κάποια εξωτερική πηγή (π.χ. ΔΕΗ). Κατασκευάζεται από λυχνίες ή τρανζίστορ ή από ολοκληρωμένα κυκλώματα.

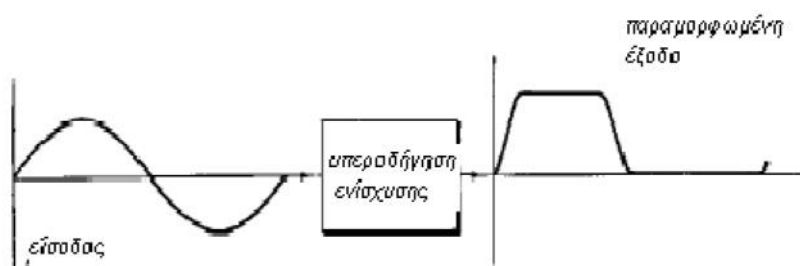
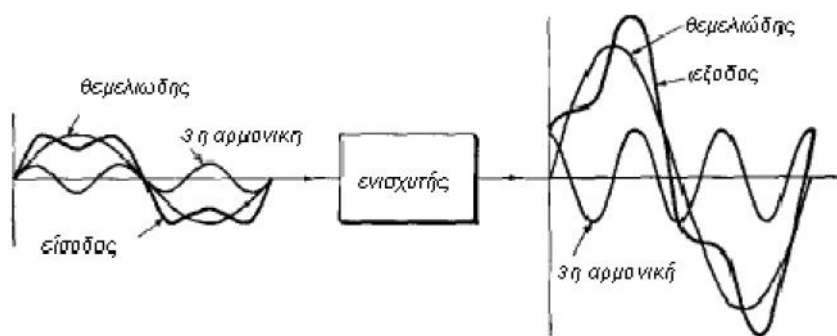
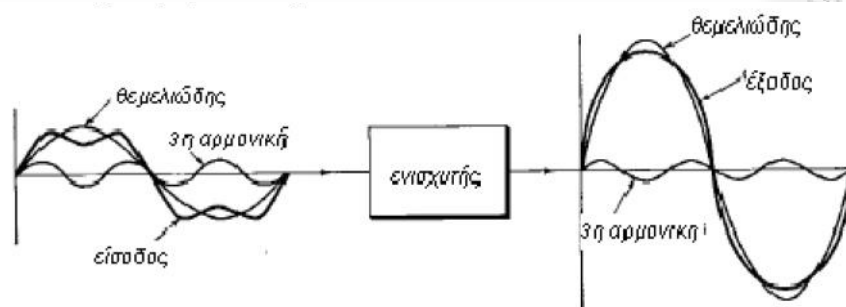


(α)



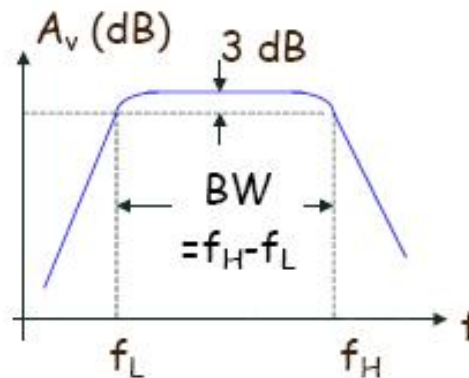
(β)

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΩΝ



iii. Απόκριση συχνότητων (Frequency Response):

Δείχνει το κατά πόσο ο ενισχυτής είναι ικανός να ενισχύσει το ίδιο όλες τις συχνότητες του ακουστικού φάσματος. Συνήθως τη συναντάμε σαν μία καμπύλη, όπου στον οριζόντιο άξονα υπάρχει η συχνότητα και στον κάθετο η ενίσχυση. Στην ιδανική περίπτωση η καμπύλη αυτή πρέπει να είναι επίπεδη για την περιοχή 5Hz έως 25KHz. Στην πραγματικότητα όμως το εύρος συχνότητων (Bandwidth) για το οποίο θεωρούμε ότι η ενίσχυση είναι σταθερή βρίσκεται μεταξύ των τομών της χαρακτηριστικής για απολαβή ίση με το 0,707 της μέγιστης ή $A_{max} - 3dB$. Στο παρακάτω σχήμα φαίνεται η καμπύλη απόκρισης συχνότητων ενός ενισχυτή, όπου f_H : ανώτερη συχνότητα αποκοπής και f_L : κατώτερη συχνότητα αποκοπής



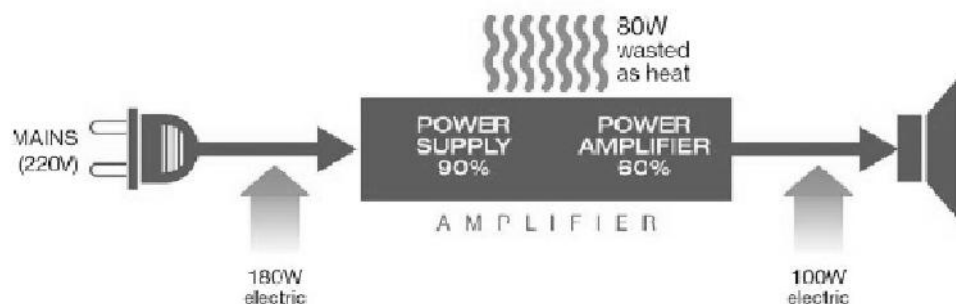
Εικόνα 5.5: Καμπύλη απόκρισης συχνότητων ενός ενισχυτή

iv. Ευαισθησία εισόδου (Input Sensitivity):

Η στάθμη του σήματος που απαιτείται στην είσοδο του ενισχυτή ώστε αυτός να αναπτύξει πλήρη ισχύ στην έξοδό του. Αυτό εξαρτάται από την απολαβή (gain) και την ολική ισχύ του ενισχυτή. Για παράδειγμα, ένας ενισχυτής 10W χρειάζεται να έχει πολύ μικρότερη ενίσχυση από έναν ενισχυτή 200W για να δώσει τη μέγιστη ισχύ του, υπό το ίδιο σήμα εισόδου. Θα ήταν χρήσιμο αν όλοι οι ενισχυτές είχαν την ίδια ενίσχυση ανεξάρτητα από την ισχύ τους αλλά δυστυχώς αυτό δεν συμβαίνει. Για το λόγο αυτό η ευαισθησία εισόδου ποικίλει ευρέως από 0,5 έως 1,5 volt ή περισσότερο.

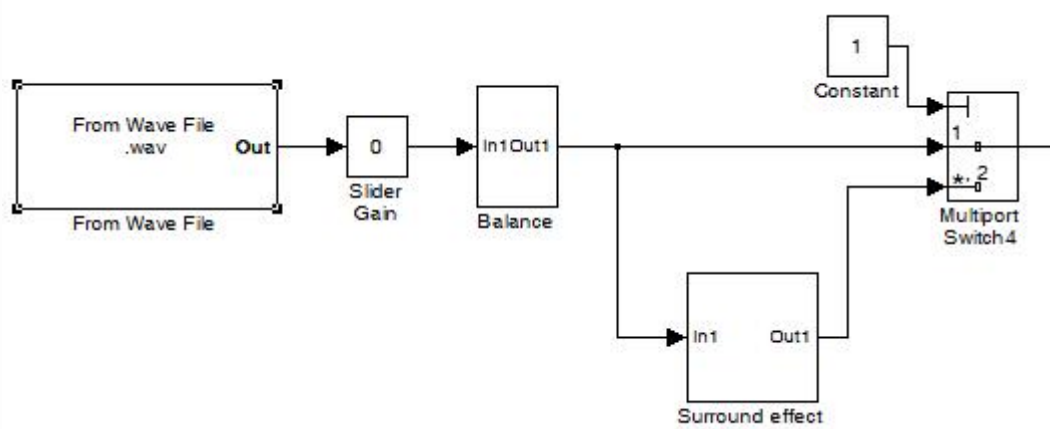
v. Απόδοση ισχύος (power efficiency):

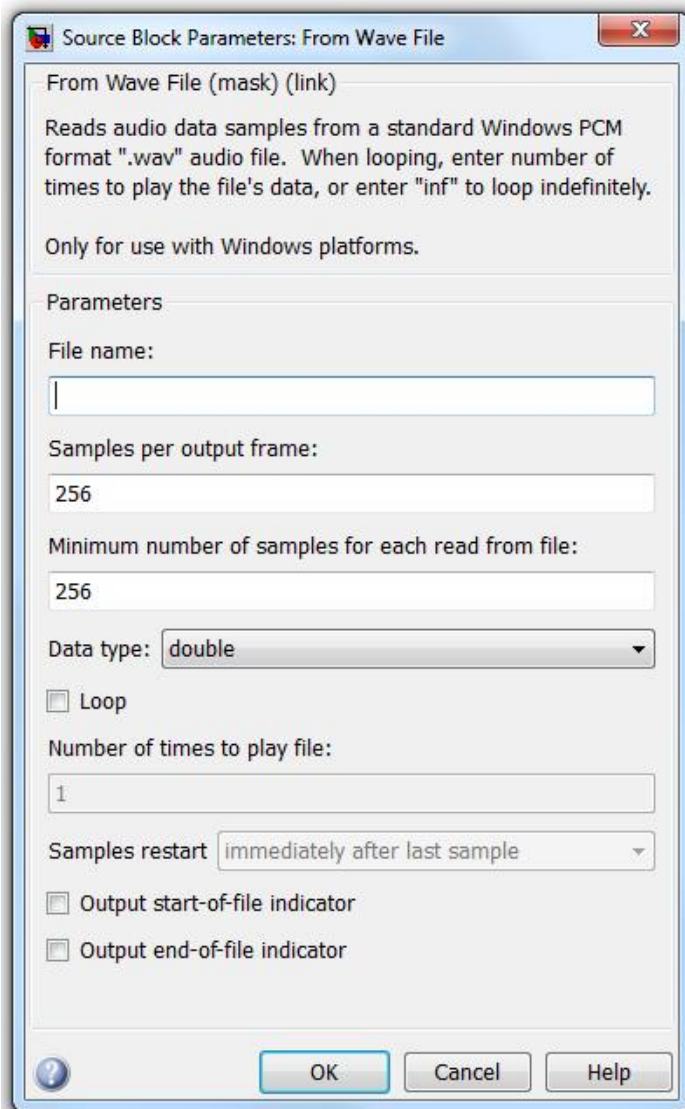
Ορίζεται σαν το λόγο της ηλεκτρικής ισχύος που παρέχεται στο ηχείο από τον ενισχυτή προς την ισχύ που απορροφά ο ενισχυτής από το τροφοδοτικό (%). Είναι ίσως η πιο σημαντική παράμετρος ενός ενισχυτή αφού προδίδει το πόσο δαπανηρός είναι αυτός σε κατανάλωση ισχύος αλλά και σε αρχικό κόστος.

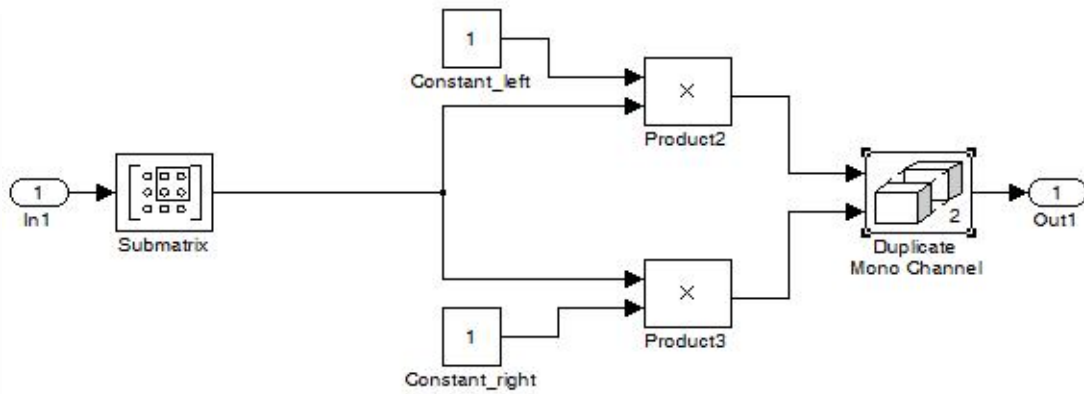


Εικόνα 5.6: Καταμερισμός θερμικών απωλειών σε ένα τυπικό ενισχυτή

Στο παράδειγμα της παραπάνω εικόνας για να τροφοδοτήσουμε το μεγάφωνο με 100 ηλεκτρικά Watt πρέπει να καταναλωθούν εξ' αρχής 180 Watt! Αυτό οφείλεται στον βαθμό απόδοσης του ενισχυτή (60%) σε συνδυασμό με τον βαθμό απόδοσης του τροφοδοτικού του (90%). Ποιες είναι όμως οι συνέπειες από τον χαμηλό αυτό βαθμό απόδοσης; Αν τα πάρουμε από την αρχή, βλέπουμε ότι επιβαρύνεται άσκοπα το δίκτυο ηλεκτροδότησης με δυσάρεστα για την τσέπη μας αποτελέσματα, ειδικά όταν έχουμε να κάνουμε με συναυλιακούς χώρους όπου η συνολική ισχύς των τελικών







Function Block Parameters: Submatrix

Submatrix (mask) (link)

Return selected portion of input matrix. 1-D input signals are treated as 2-D column vectors. The output is always 2-D.

This block performs 1-based indexing.

Parameters

Row span: Range of rows

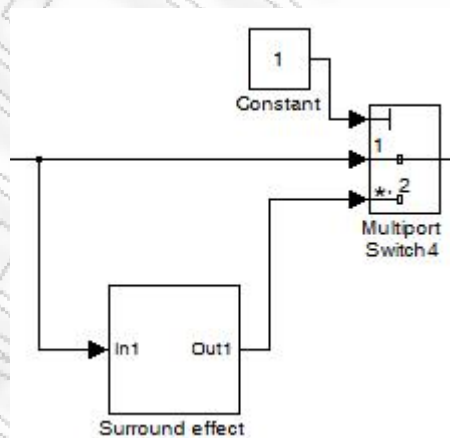
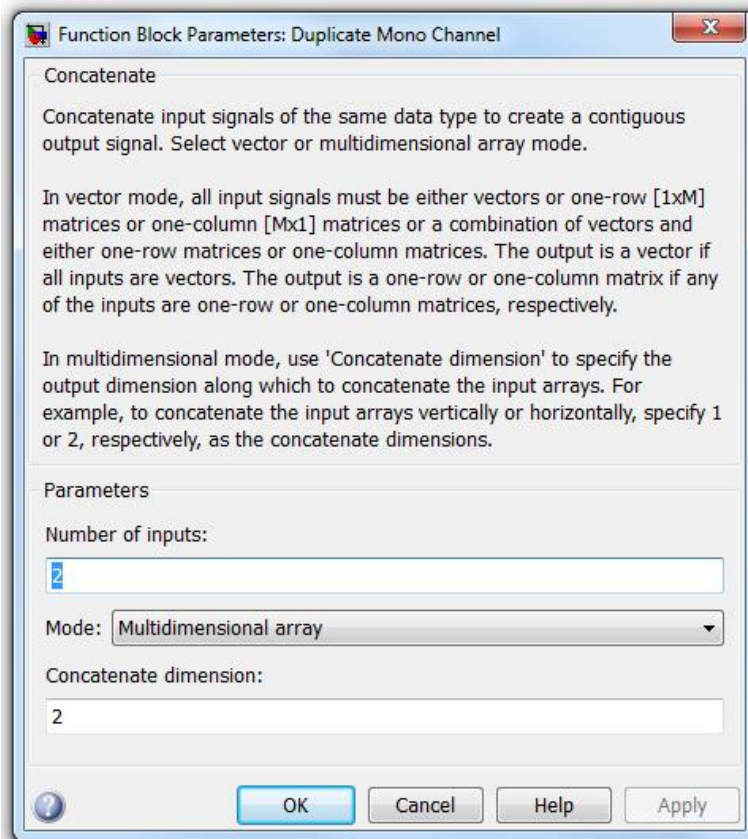
Starting row: First

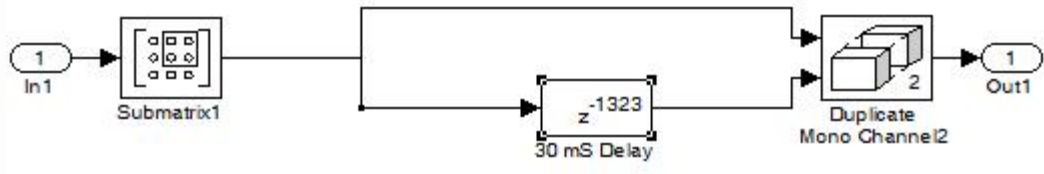
Ending row: Last

Column span: One column

Column: First

OK Cancel Help Apply



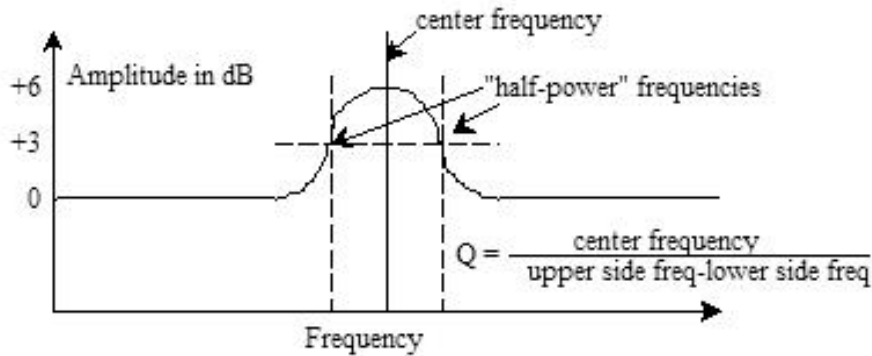


FAKULTÄT FÜR INGENIEURWISSENSCHAFTEN

Οι παραμετρικοί ισοσταθμιστές είναι πιο ευμετάβλητοι αλλά όχι τόσο διαισθητικοί στη χρήση, από ότι οι γραφικοί ισοσταθμιστές. Εκτός από το κέρδος (gain), είναι δυνατόν να προσαρμοστεί και η κεντρική συχνότητα, το εύρος ζώνης και ο συντελεστής Q αντίστοιχα, για κάθε φίλτρο. Με έναν ορισμένο αριθμό των παραμετρικών φίλτρων ως επί το πλείστον κάθε επιθυμητή απόκριση συχνότητας μπορεί να επιτευχθεί, κάνοντας το κατάλληλο για ηχητικές εφαρμογές.

5.3.3 Έλεγχος Εύρους Ζώνης Q (Bandwidth Control Q)

Ένα χαρακτηριστικό των παραμετρικών ισοσταθμιστών, είναι και η επιλογή Q που προέρχεται από την λέξη Quality μιας και ήταν ο κύριος λόγος για να κριθεί αν είναι χαμηλής ή καλής ποιότητας ένας ισοσταθμιστής, τις εποχές των πρώτων EQ.

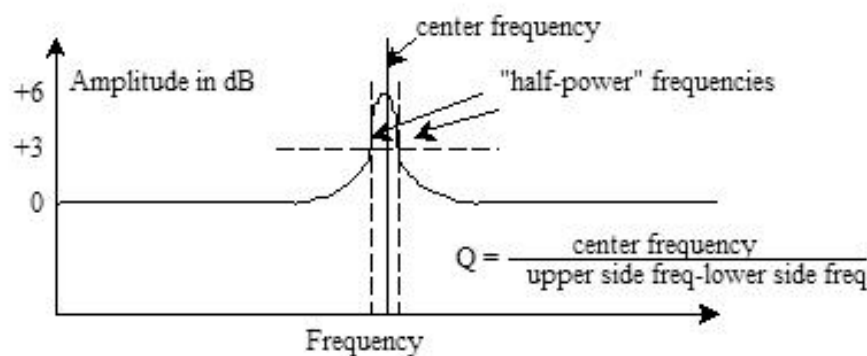


Εικόνα 5.15: Ο ορισμός του Q

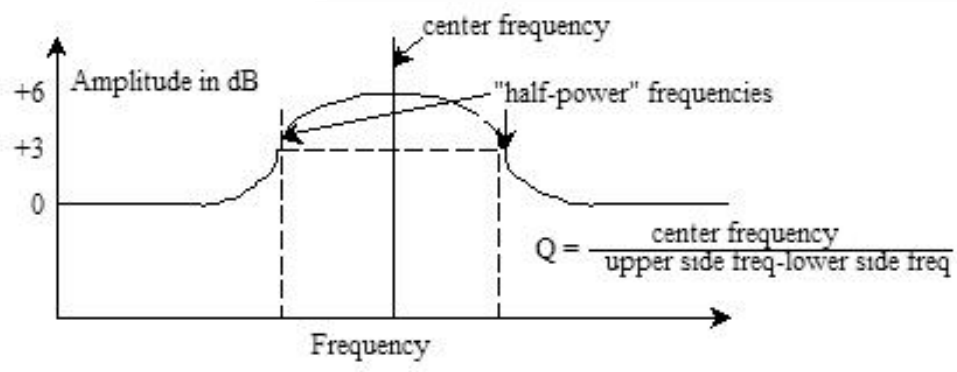
Το Q καθορίζει πόση αυξομείωση θα εφαρμοστεί σε κάθε πλευρά της καμπύλης συχνοτήτων, δηλαδή ελέγχει το πόσο πλατιά ή στενή είναι η περιοχή των συχνοτήτων που επηρεάζει το κάθε φίλτρο. Η τιμή του Q είναι το πηλίκο της διαίρεσης της κεντρικής συχνότητας με το εύρος της ζώνης συχνοτήτων που επηρεάζονται:

$$Q = \frac{\sqrt{2B}}{2B - 1}$$

Μεγάλες τιμές στο Q συνεπάγονται μικρό εύρος συχνοτήτων, εντός του οποίου μπορεί να αυξομειωθεί η ακρίβεια, ενώ χαμηλές τιμές στο Q σημαίνουν ευρύτερες περιοχές στην καμπύλη για ενίσχυση ή κόψιμο.



Εικόνα 5.16: Μεγάλες τιμές στο Q, μικρότερο εύρος συχνοτήτων







δυνατότητα ρύθμισης απολαβής, ενώ παλιότερα χρησιμοποιούνταν κυρίως παθητικά αναλογικά φίλτρα.

Η συμπεριφορά των φίλτρων, τα οποία είναι στην απόλυτη πλειοψηφία γραμμικά, περιγράφεται στο πεδίο του χρόνου από μια διαφορική εξίσωση, στην αναλογική περίπτωση ή από μια εξίσωση διαφορών, για την περίπτωση ψηφιακών φίλτρων, ενώ στο πεδίο των συχνοτήτων περιγράφονται από μια γραμμική αλγεβρική εξίσωση, που προκύπτει από το μετασχηματισμό Laplace στην πρώτη και το μετασχηματισμό Z στη δεύτερη. Η αναπαράσταση, όμως, στο πεδίο των συχνοτήτων θεωρεί μόνιμη ημιτονοειδή κατάσταση και συνεπώς δεν μπορεί να περιγράψει τη μεταβατική συμπεριφορά των φίλτρων, κάτι που είναι δυνατό να εκτιμηθεί έμμεσα, από το λεγόμενο συντελεστή ποιότητας, Q, και να αναπαρασταθεί, είτε με μεταβατική διέγερση των φίλτρων, είτε με κατάλληλες μεθόδους χρονοσυχνοτικής αναπαράστασης, όπως είναι τα διαγράμματα CSD (Cumulative Spectral Decay), ή τα διαγράμματα CAE (Cumulative Attack Envelopes), που τελευταία έχουν αποδειχθεί καλύτερα.

Τα φίλτρα, επίσης, χωρίζονται σε διάφορες κατηγορίες, ανάλογα με την επέμβαση που κάνουν στο φάσμα του σήματος που δέχονται σαν είσοδο. Η επιλεκτικότητα κάθε φίλτρου όσον αφορά την συχνότητα είναι ίσως η πιο συνήθης μέθοδος κατηγοριοποίησης. Έτσι έχουμε τα βαθυπερατά (LP) και υψιπερατά (HP) φίλτρα που επιτρέπουν τη διέλευση των συχνοτήτων που είναι μικρότερες ή μεγαλύτερες, αντίστοιχα, από κάποια συχνότητα, που ονομάζεται συχνότητα αποκοπής. Υπάρχουν ακόμη τα ζωνοπερατά ή ζωνοδιαβατά (BP) και τα ζωνοαπαγορευτικά (BR) φίλτρα, που επιτρέπουν ή απαγορεύουν αντίστοιχα, τη διέλευση μιας ζώνης συχνοτήτων γύρω από μια κεντρική συχνότητα.

Τα φίλτρα χαρακτηρίζονται από διάφορα χαρακτηριστικά τους, όπως η τάξη τους και οι ιδιοσυχνότητες με τους αντίστοιχους συντελεστές ποιότητας ή οι συχνότητες αποκοπής ή η κεντρική συχνότητα και το εύρος ζώνης.

Η τάξη του φίλτρου δείχνει πόσους πόλους έχει το φίλτρο και καθορίζει την κλίση της απόκρισης κατά συχνότητα του πλάτους του. Μεγαλύτερη τάξη αντιστοιχεί σε πιο απότομη κλίση, πιο απότομη αποκοπή. Για κάθε πόλο τώρα ενός φίλτρου, υπάρχει μια αντίστοιχη ιδιοσυχνότητα ω και ένας συντελεστής ποιότητας συντονισμού ή απλά συντελεστής ποιότητας Q, $Q=2\pi$ (μέγιστη αποθηκευμένη ενέργεια / σύνολο της ενέργειας που χάνεται ανά περίοδο), στην κεντρική συχνότητα, που δείχνει την τάση του φίλτρου να ταλαντώσει ή να διατηρεί τη συγκεκριμένη ιδιοσυχνότητα, καθώς επίσης κι αυτό που ονομάζουμε «ταχύτητα» του φίλτρου.

Η διατήρηση κάποιων συχνοτήτων λόγω της τάσης του φίλτρου να ταλαντώσει, οδηγεί τελικά σε μείωση της διαύγειας αναπαραγωγής, ενώ η ταχύτητα του φίλτρου κρίνεται από το ρυθμό με τον οποίο αποσβήνεται το σύνολο των συχνοτήτων στην περιοχή διέλευσης, και συνεπώς ταχύτερο φίλτρο οδηγεί σε οξύτερη αναπαραγωγή μετώπων κύματος. Υπάρχει, μάλιστα, μια αντιστοίχιση συντελεστών ποιότητας, απόσβεσης (damping) ταλαντώσεων και ταχύτητας φίλτρου.

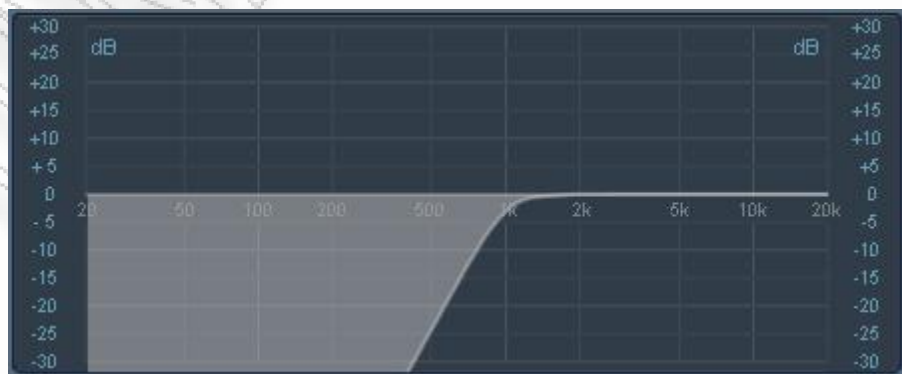
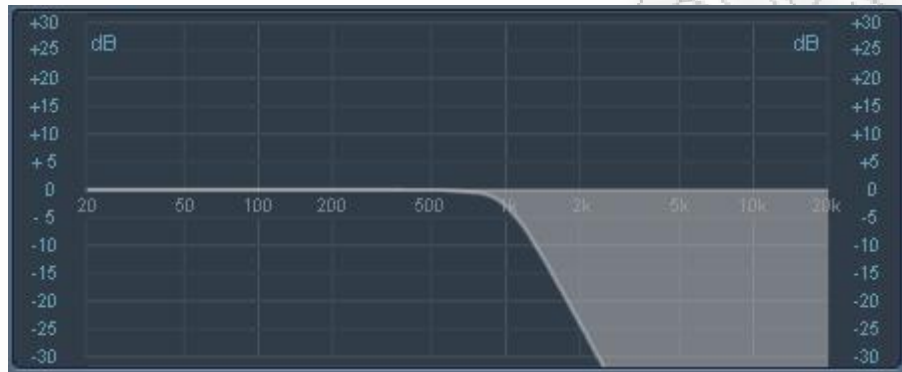
Όταν ο συντελεστής $Q < 0.5$, τότε το φίλτρο δεν ταλαντώνει καθόλου και είναι πολύ αργό (περίπτωση υψηλής απόσβεσης, overdamping). Όταν $Q = 0.5$, τότε το φίλτρο οριακά δεν ταλαντώνει και είναι αρκετά αργό (περίπτωση κρίσιμης απόσβεσης, critical damping). Όταν $Q > 0.5$ και έως $Q < 0.8$, τότε το φίλτρο ταλαντώνει ελάχιστα και είναι καλής ταχύτητας. Όταν $Q < 0.8$ και έως $Q = 1$, τότε το φίλτρο είναι ταχύτατο, αλλά ταλαντώνει έντονα.

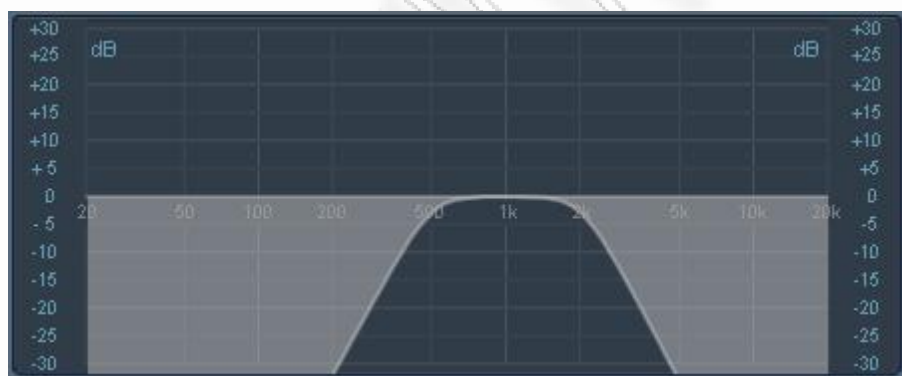
Επιπλέον σε ζωνοπερατά / ζωνοδιαβατά φίλτρα η σχέση του Q με το εύρος ζώνης είναι γνήσια φθίνουσα, δηλαδή αυξανόμενου του Q μειώνεται το εύρος ζώνης, άρα φίλτρα πολύ στενής ζώνης έχουν μεγάλο Q, μικρό συντελεστή απόσβεσης και άρα τάση να συντονίζονται ή να διατηρούν για μεγάλο χρονικό διάστημα μια διέγερση στην ιδιοσυχνότητά τους. Έτσι ενώ μια μεταβατική διέγερση μπορεί να διακοπεί η απόκριση του φίλτρου δεν διακόπτεται ταυτόχρονα, αλλά μετά από κάποιο χρονικό διάστημα, δίνοντας το φαινόμενο κωδωνισμού (ringing), σαν εμμονή του συστήματος στην αναπαραγωγή κάποιων συχνοτήτων, ακόμη και αφού η διέγερσή τους παύσει.

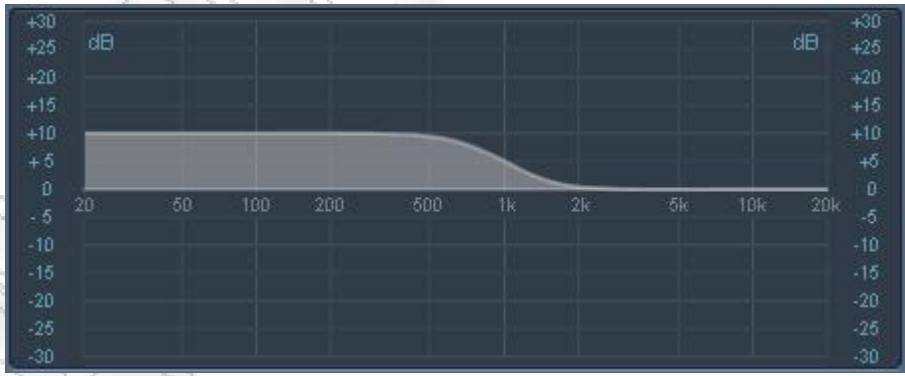
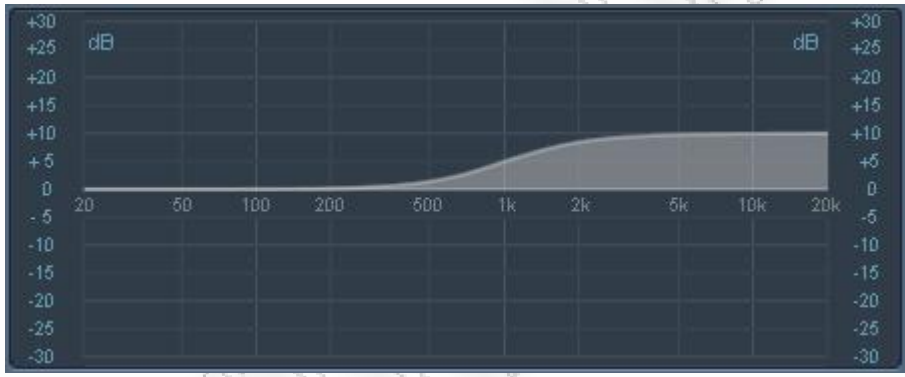
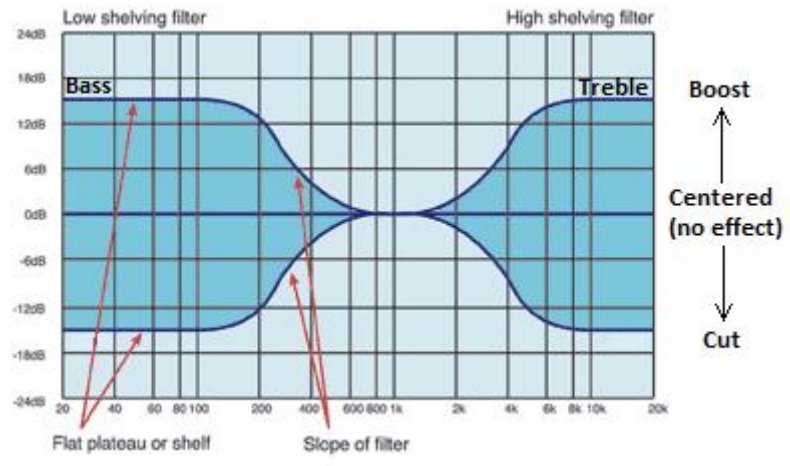
5.3.6 Κατηγορίες φίλτρων

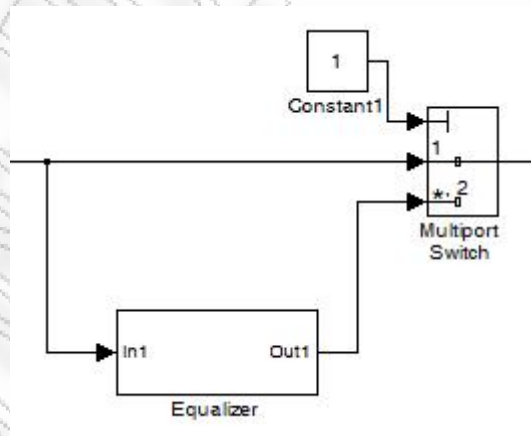
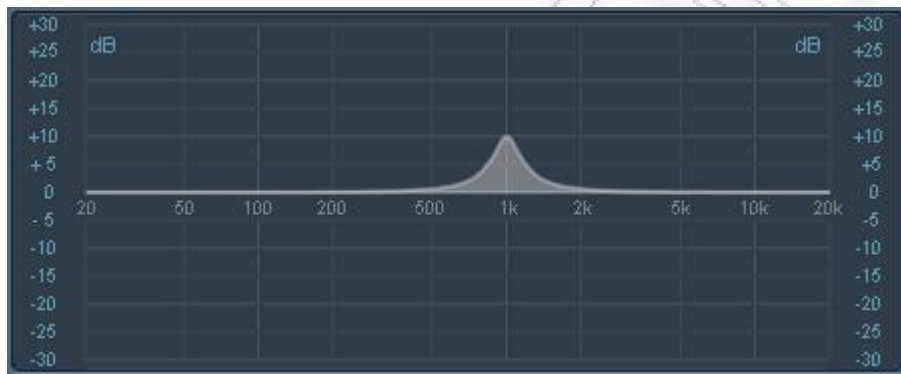
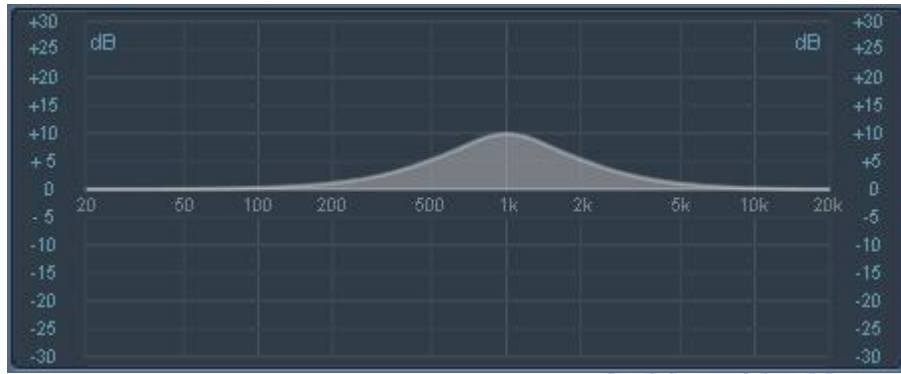
i. Βαθυπερατά φίλτρα (Low Pass Filters)

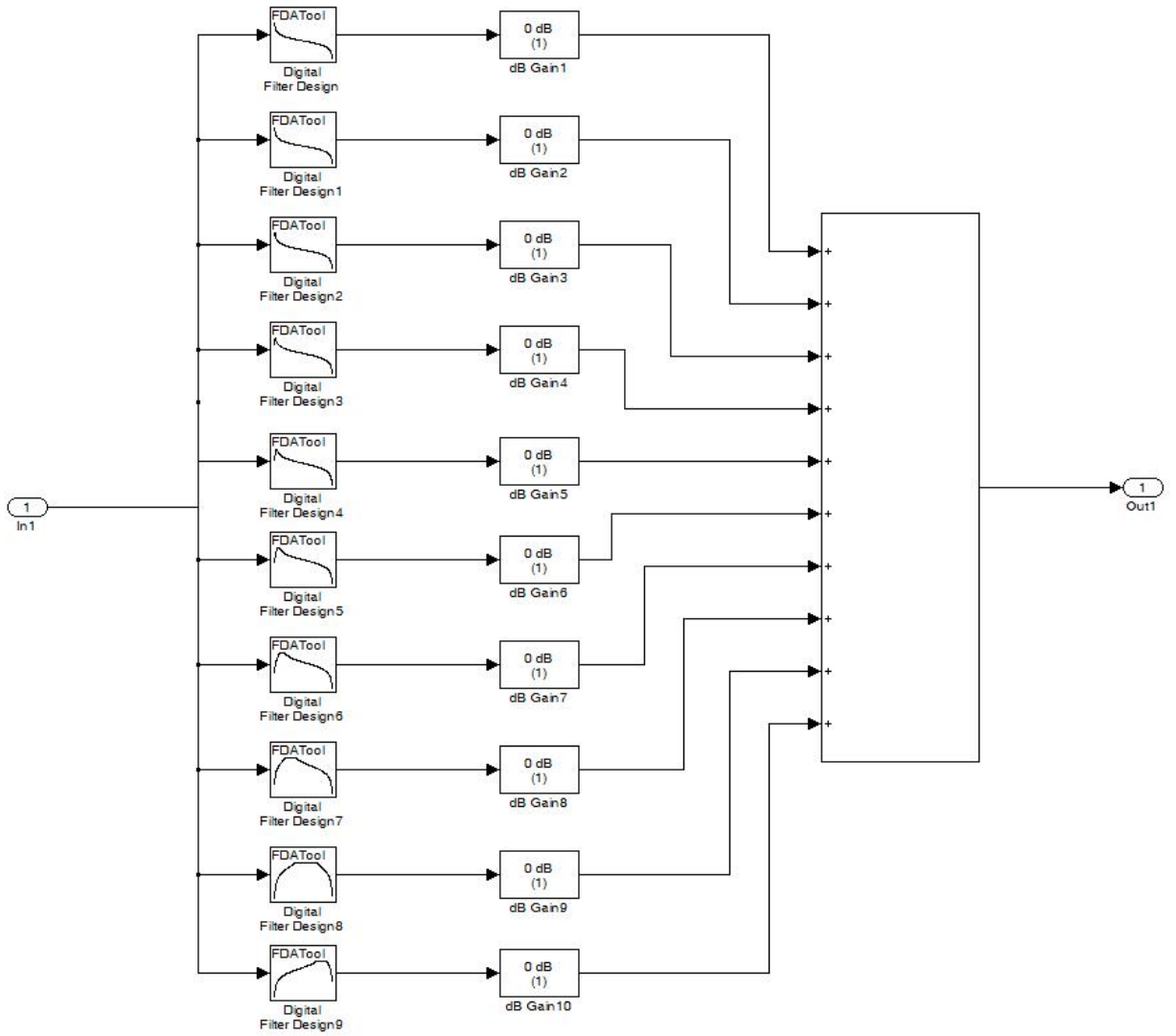
Αυτού του είδους τα φίλτρα είναι χρήσιμα γιατί μας επιτρέπουν να ορίσουμε από ποιο σημείο και μετά θα κοπούν ορισμένες συχνότητες. Αν θέλουμε να κρατήσουμε μόνο τις bass συχνότητες από κάποιον ήχο, τότε χρησιμοποιούμε ένα βαθυπερατό φίλτρο

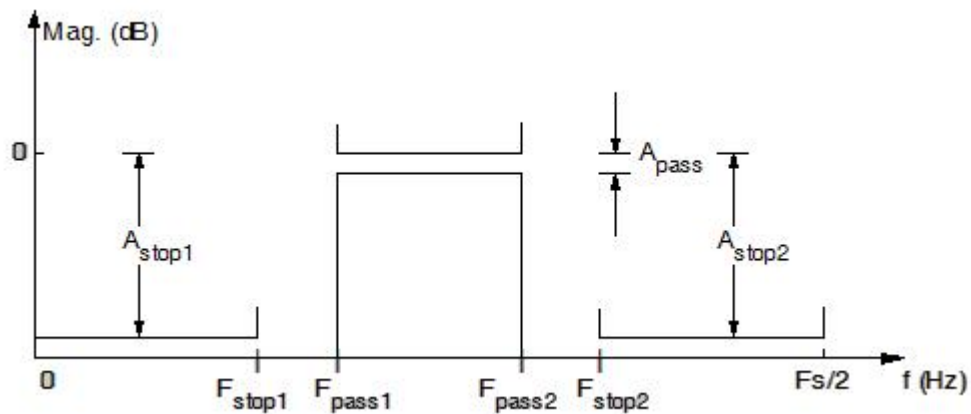
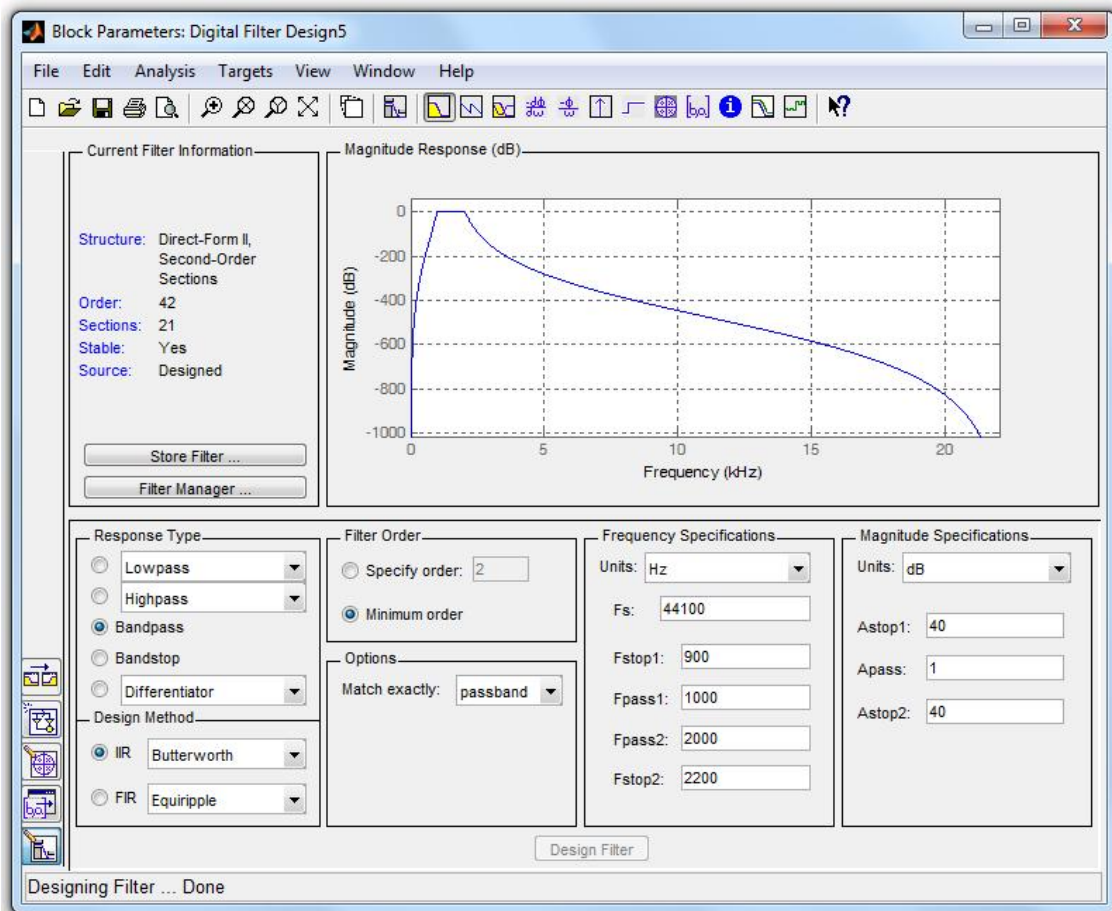


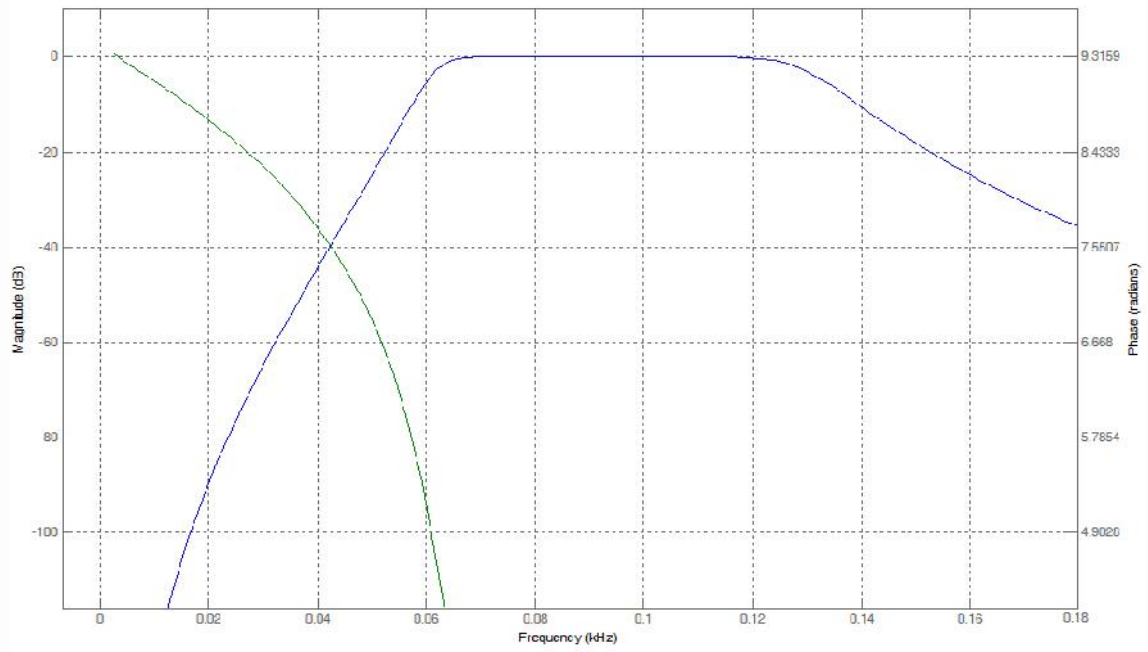
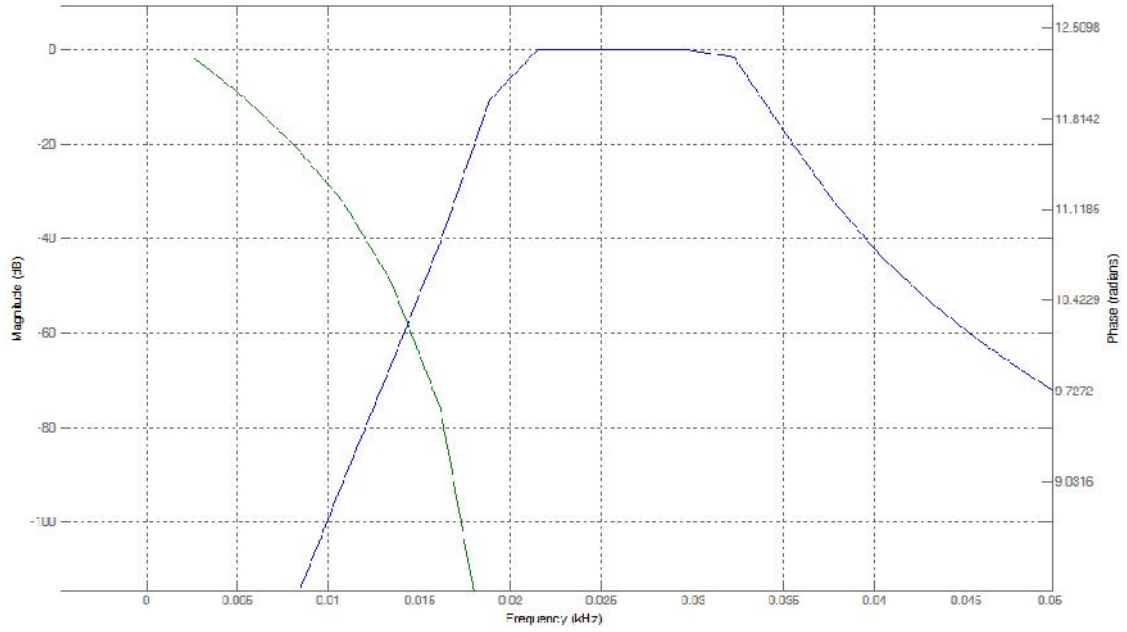


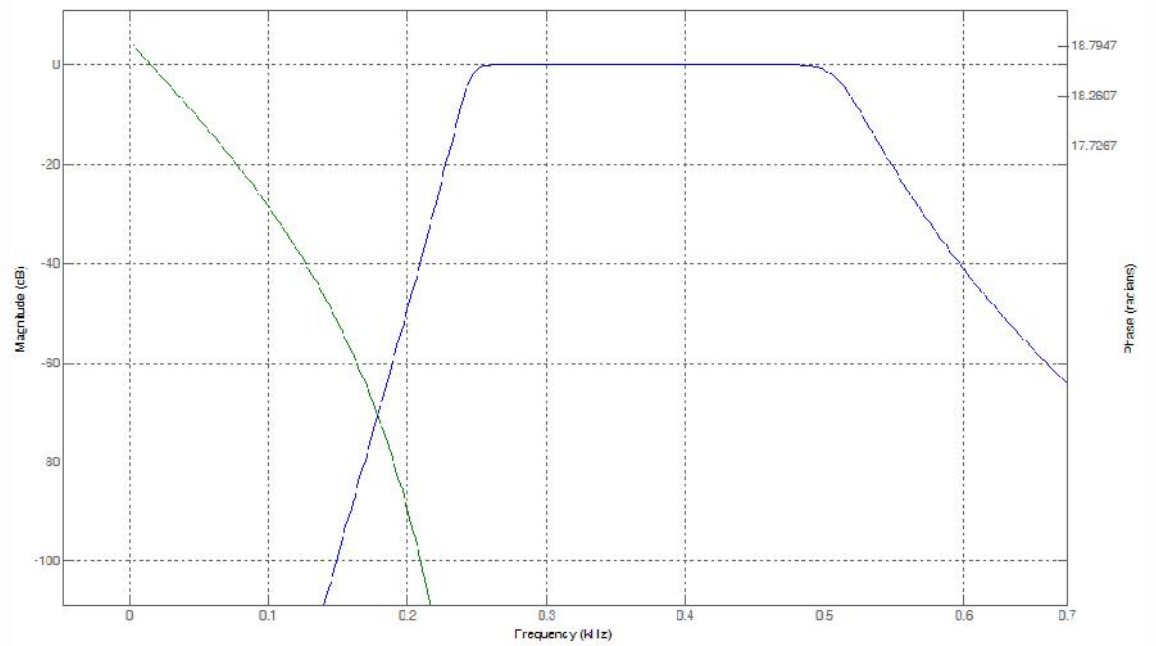
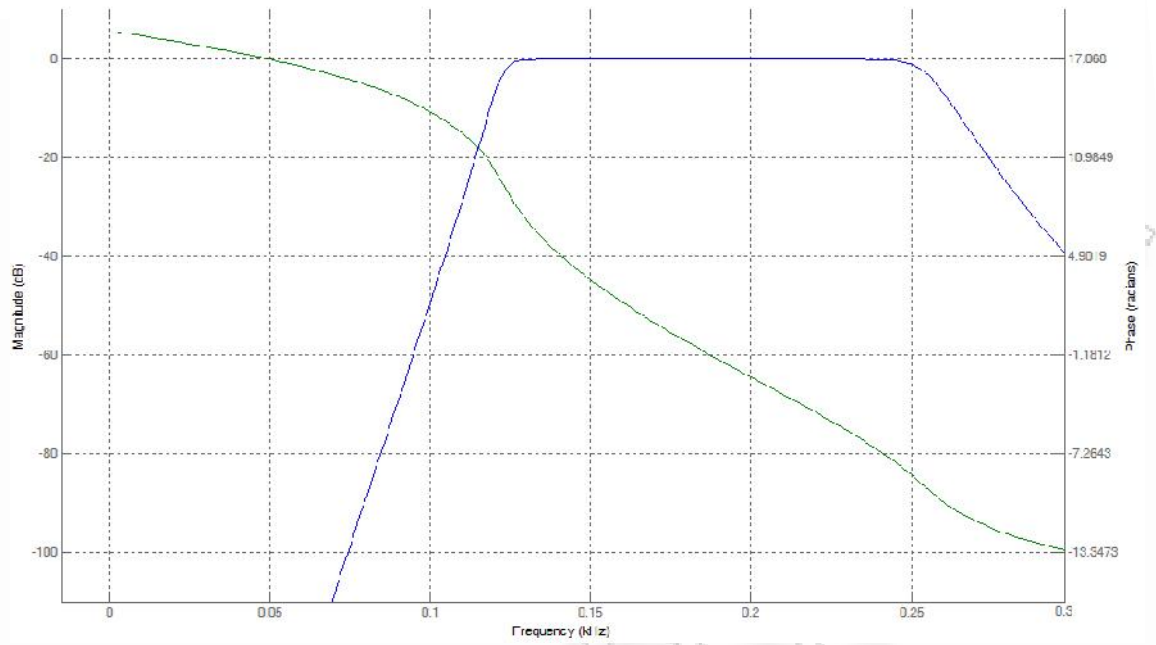


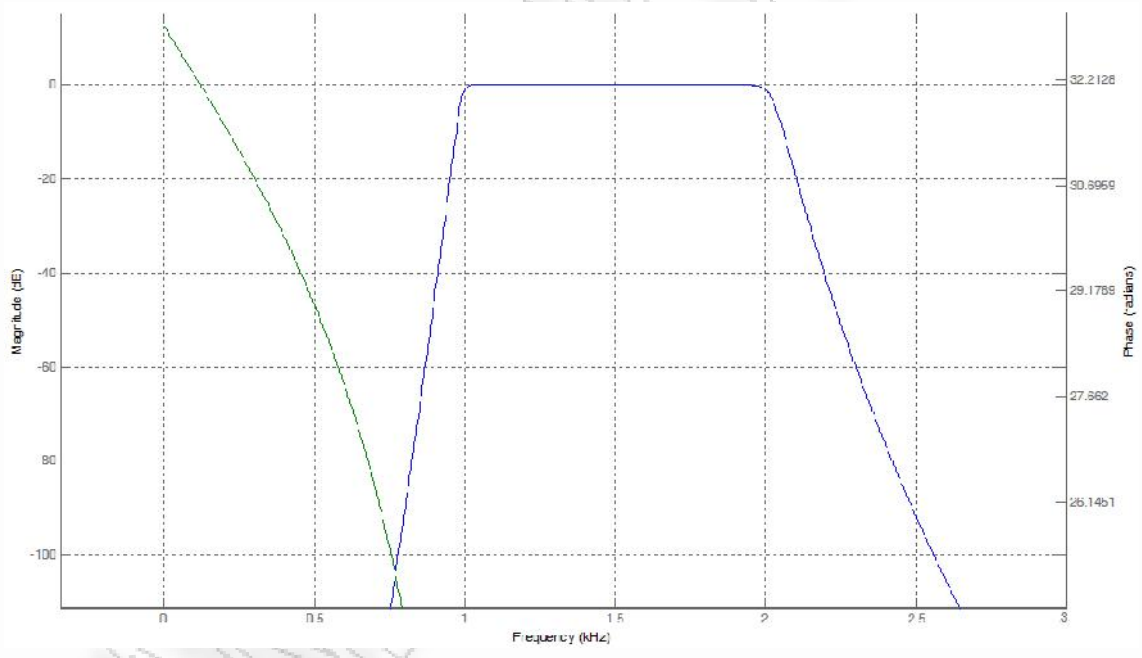
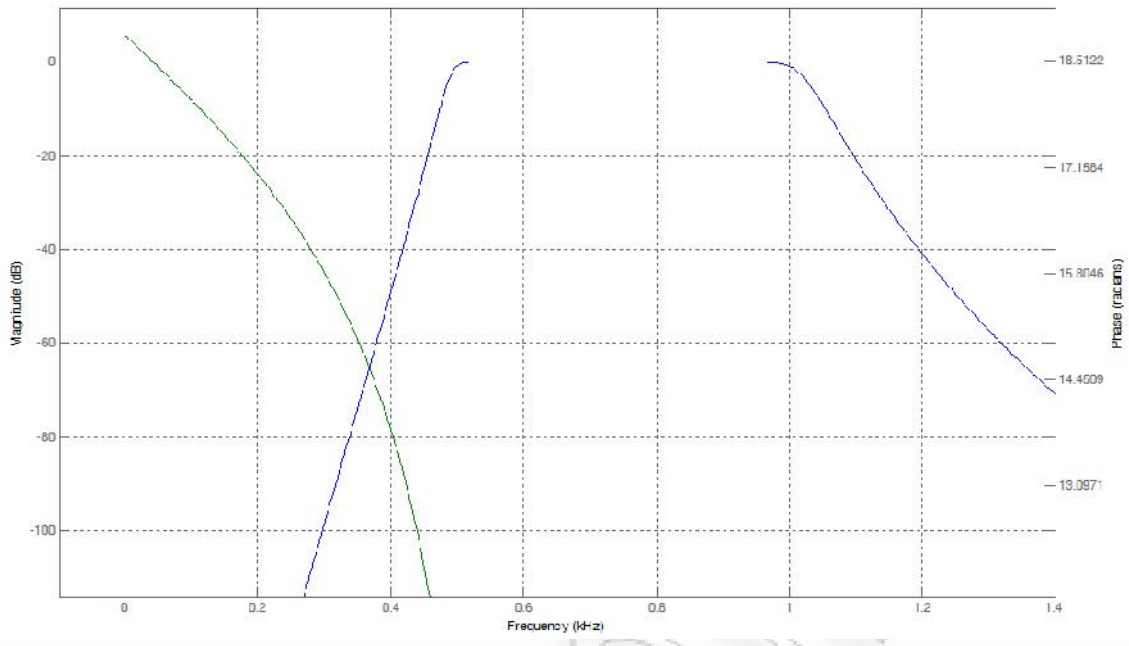


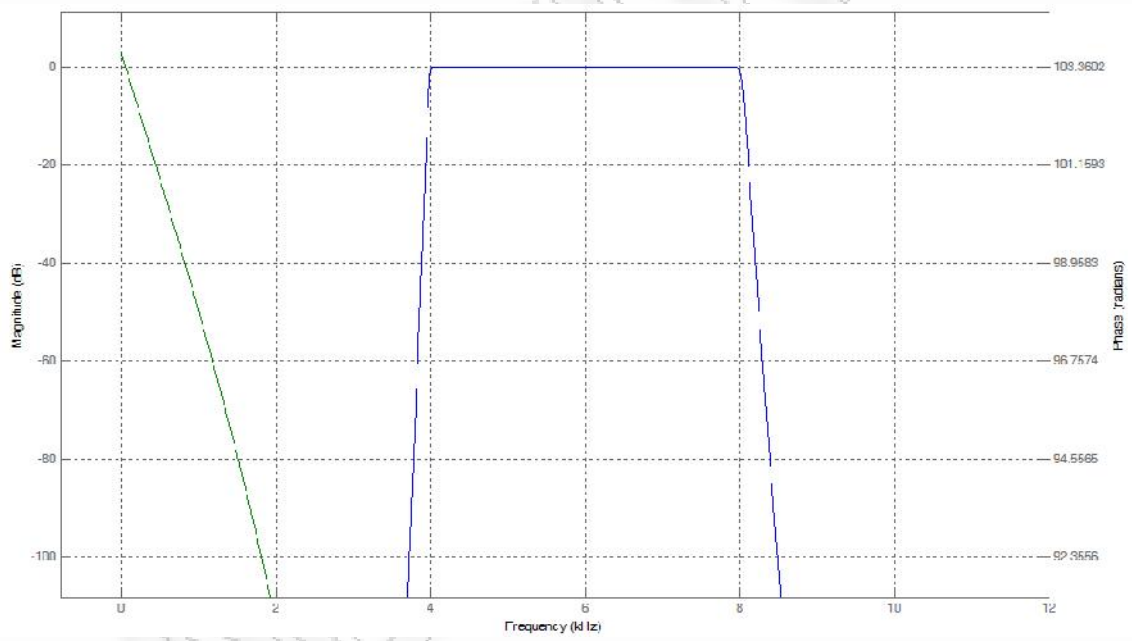
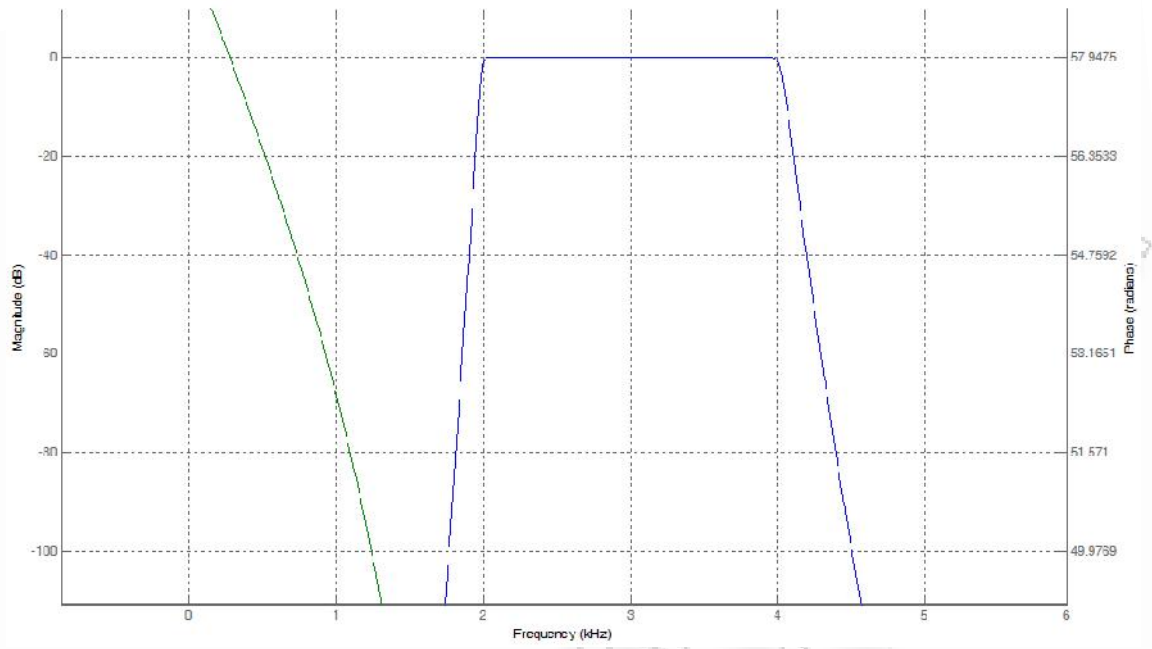




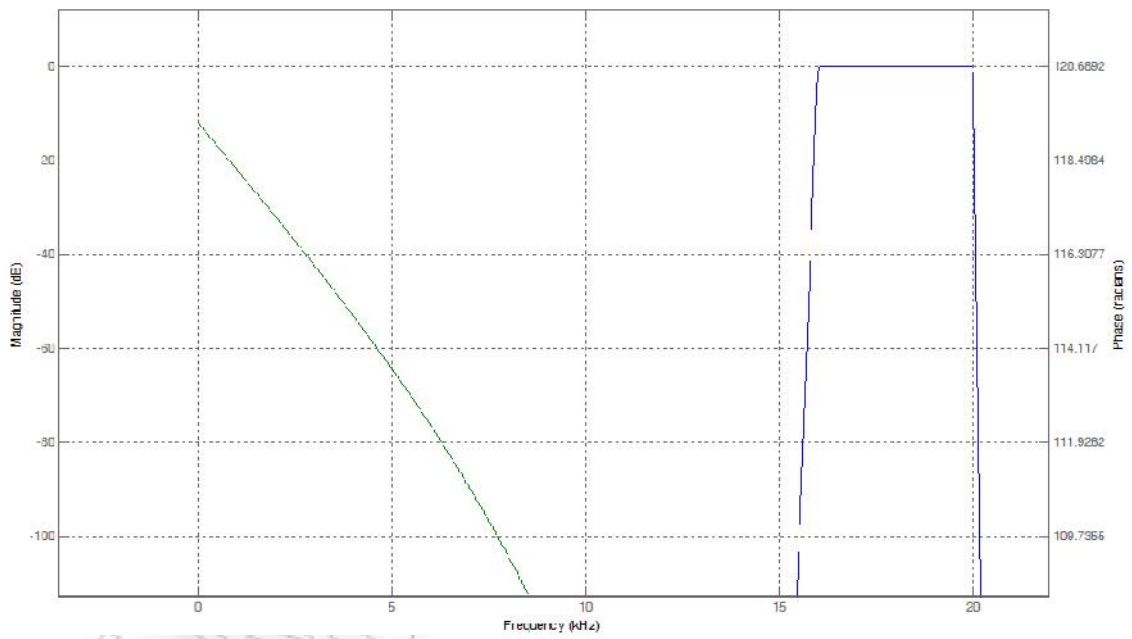
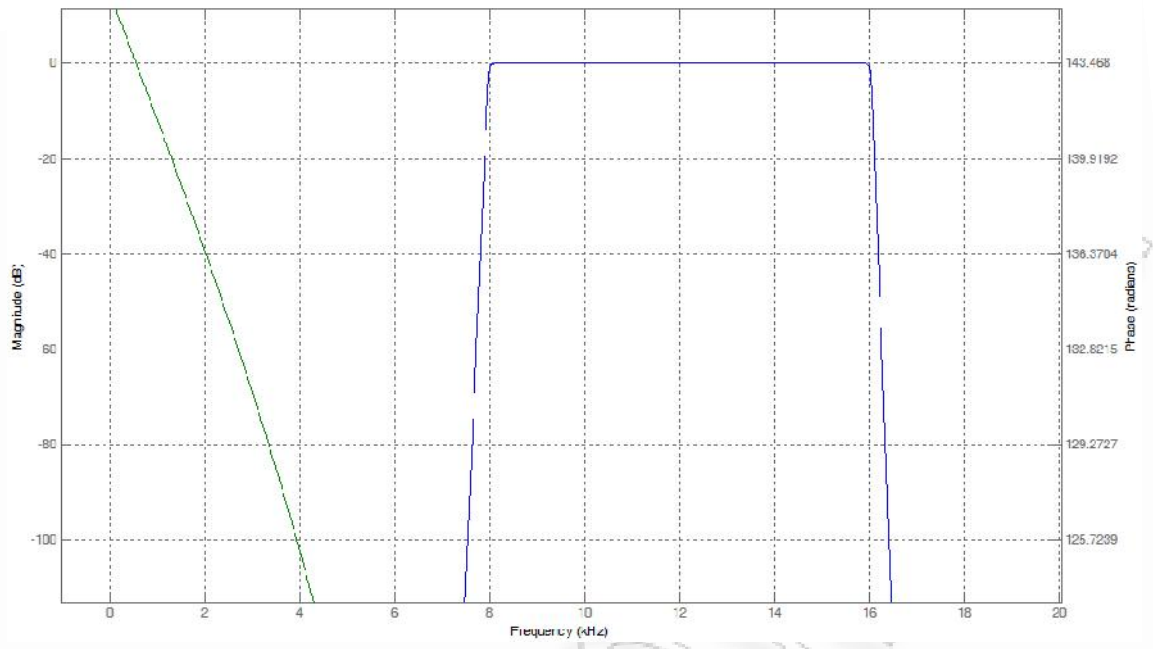


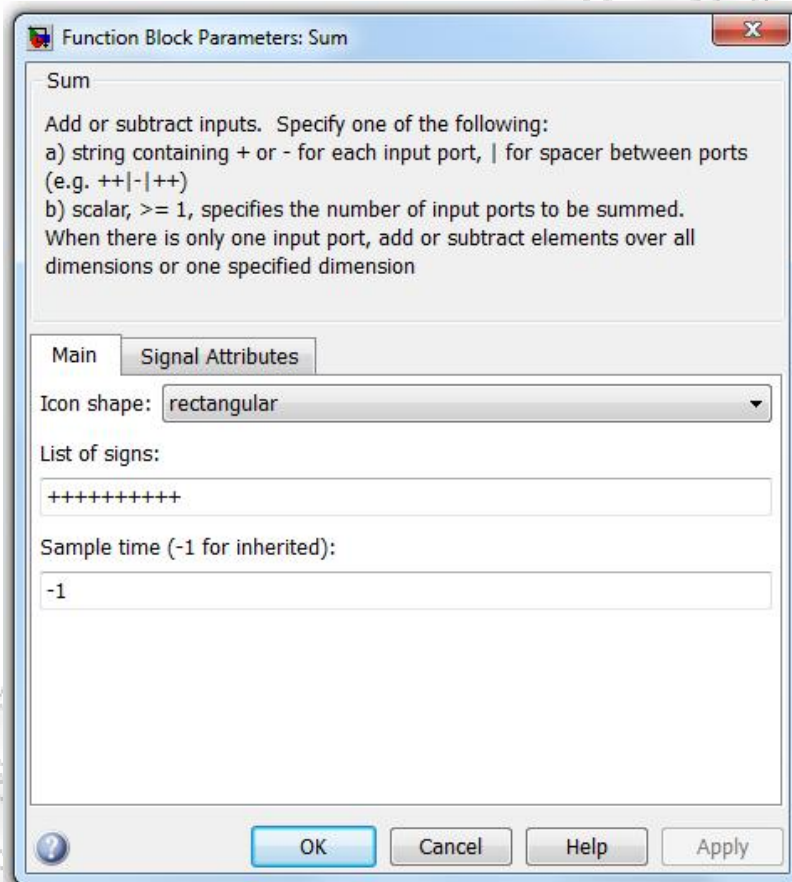
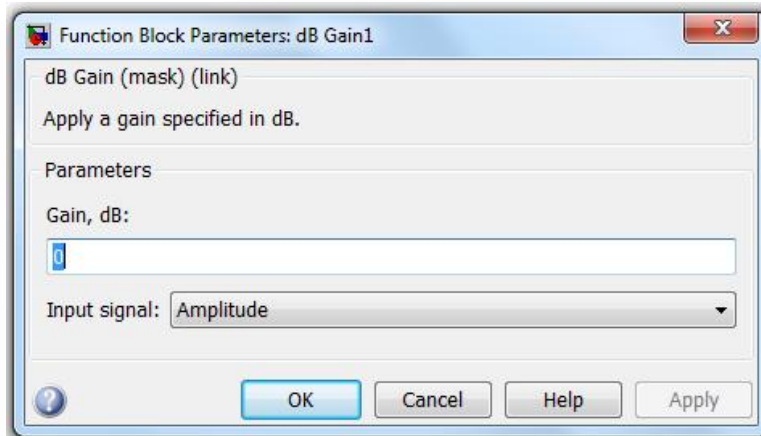




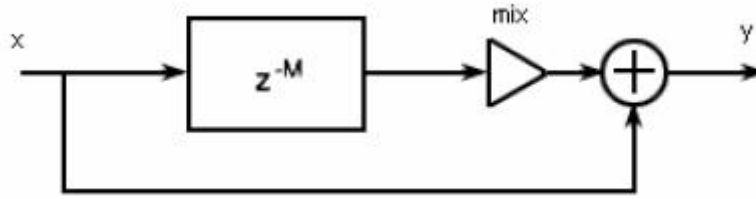


PAWEL





σχεδιάσουμε το delay εφέ είναι χρησιμοποιώντας ένα FIR φίλτρο κτένας. Το διάγραμμα που αντιπροσωπεύει το φίλτρο αυτό φαίνεται παρακάτω.



Εικόνα 5.46: Διάγραμμα ενός FIR φίλτρο κτένας

Η επόμενη διαφορική εξίσωση εκφράζει την απόδοσή του παραπάνω φίλτρου:

$$y(n) = x(n) + \text{mix} x(n - M)$$

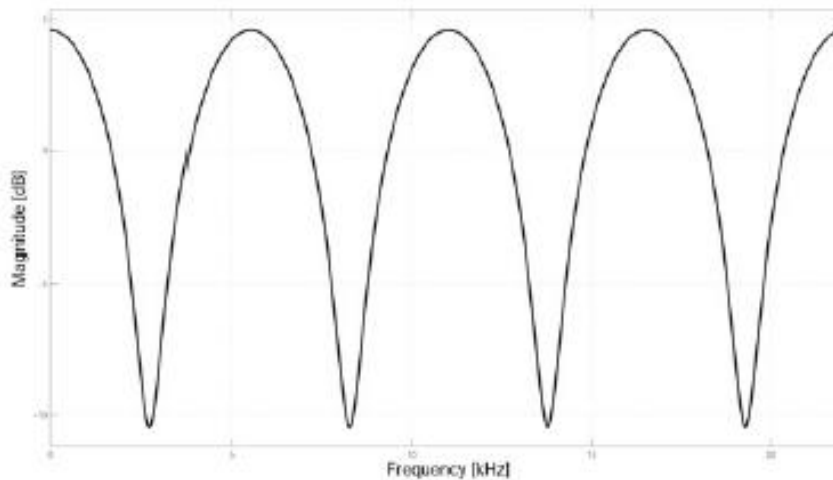
όπου, $M = \tau fs$

και fs η συχνότητα δειγματοληψίας και τ η τιμή της χρονικής καθυστέρησης σε δευτερόλεπτα. Παρατηρούμε ότι η έξοδος του συστήματος βασίζεται στο σήμα εισόδου και σε ένα παρελθοντικό σήμα εισόδου. Αποτελείται επίσης και από ένα βρόγχο επανατροφοδότησης. Για την ακρίβεια, η έξοδος του συστήματος αποτελείται από το σήμα εισόδου συν ένα χρονικά καθυστερημένο αντίγραφο του, εξασθενημένο από το gain mix.

Η συνάρτηση μεταφοράς του φίλτρου δίνεται από τη σχέση:

$$H(z) = 1 + gz^{-M}$$

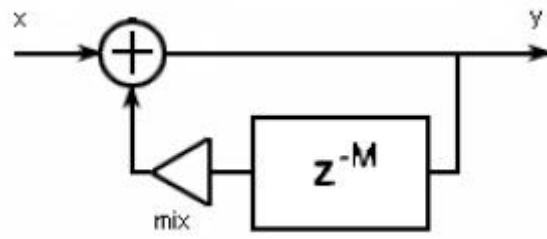
και η γραφική παράσταση του φαίνεται στο παρακάτω σχήμα.



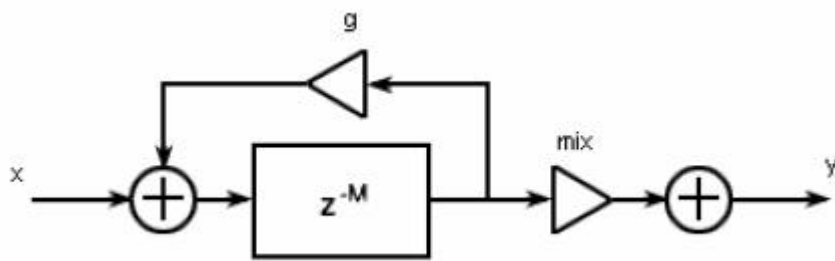
Εικόνα 5.47: Συνάρτηση μεταφοράς του FIR φίλτρο κτένας για χρονική καθυστέρηση 8 δειγμάτων και για mix = 0,7

Είναι φανερό στο παραπάνω σχήμα πώς αυτό το φίλτρο ενισχύει τις συχνότητες οι οποίες είναι πολλαπλάσια της θεμελιώδους συχνότητας fs / M και αμβλύνει τις υπόλοιπες συχνότητες που βρίσκονται στο ενδιάμεσο.

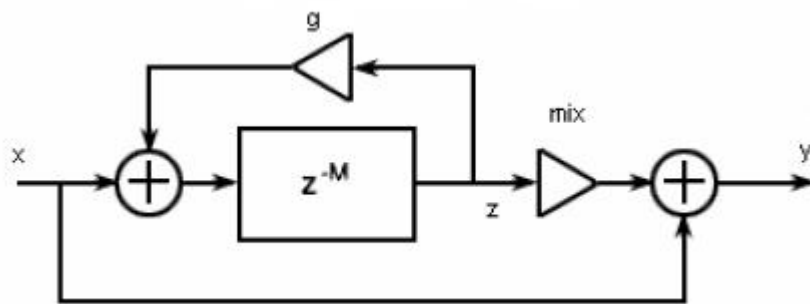
Όσον αφορά την χρονική καθυστέρηση της ανατροφοδότησης, υπάρχουν διάφορες προσεγγίσεις. Το παρακάτω διάγραμμα βασίζεται σε ένα IIR φίλτρο κτένας.



$$y[n] = x[n] + g y[n - M]$$

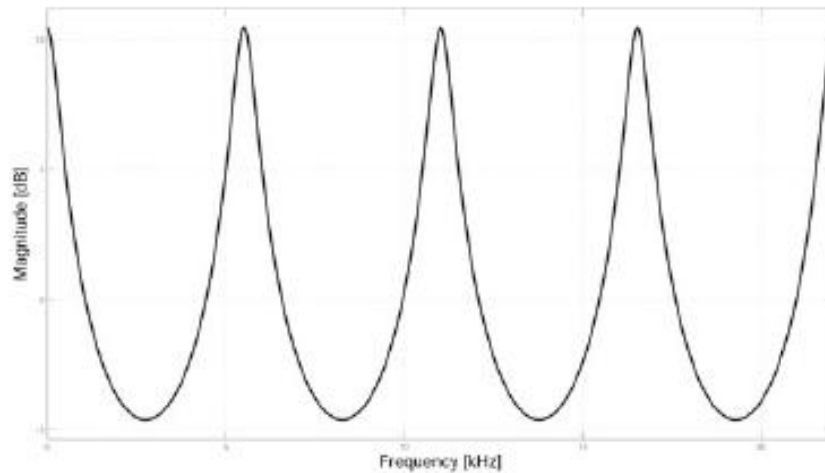


$$y[n] = x[n - M] + g y[n - M]$$



$$y[n] = x[n] + \text{mix} \cdot z[n] - g \cdot z[n]$$

$$H(z) = \frac{z^{-M}}{1 - gz^{-M}}$$



Εικόνα 5.51: Συνάρτηση μεταφοράς του IIR φίλτρου κτένας για χρονική καθυστέρηση 8 δειγμάτων και για $g = 0,7$

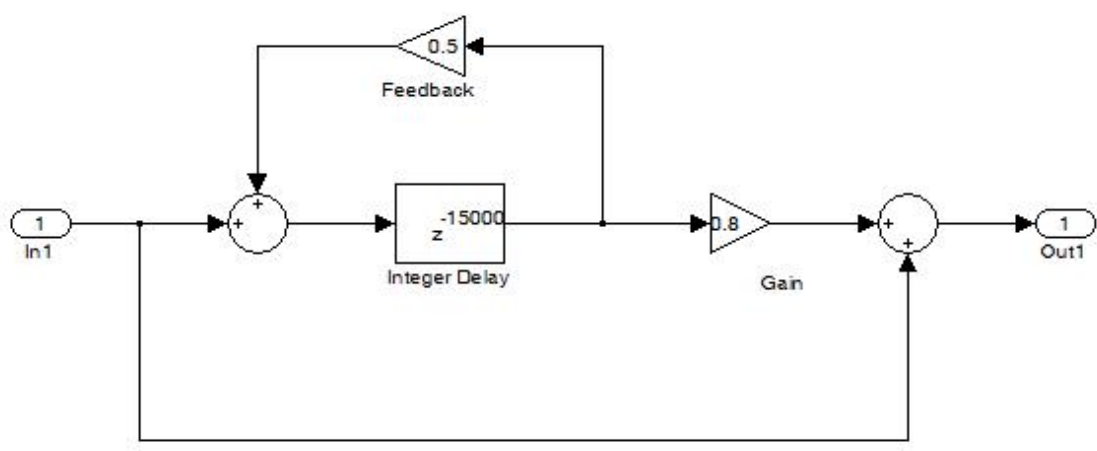
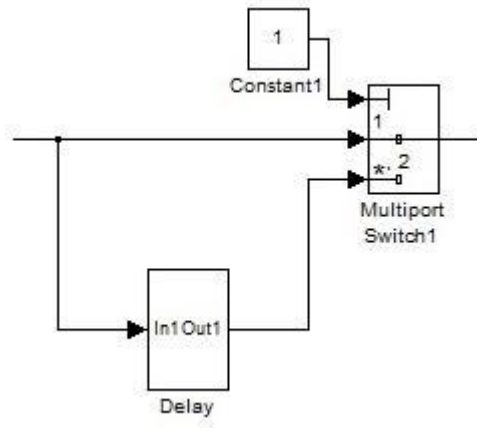
Υπάρχουν τρεις παράμετροι για τη ρύθμιση του delay εφέ. Η χρονική καθυστέρηση M , η σχέση μεταξύ του εύρους του καθυστερημένου σήματος και του σήματος εισόδου το οποίο δίνεται την παράμετρο mix καθώς και από το κέρδος ανάδρασης g .

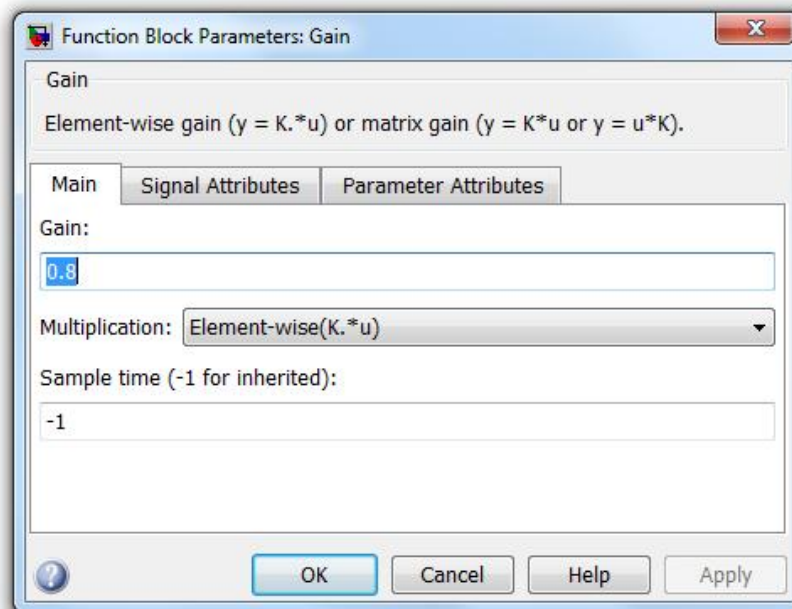
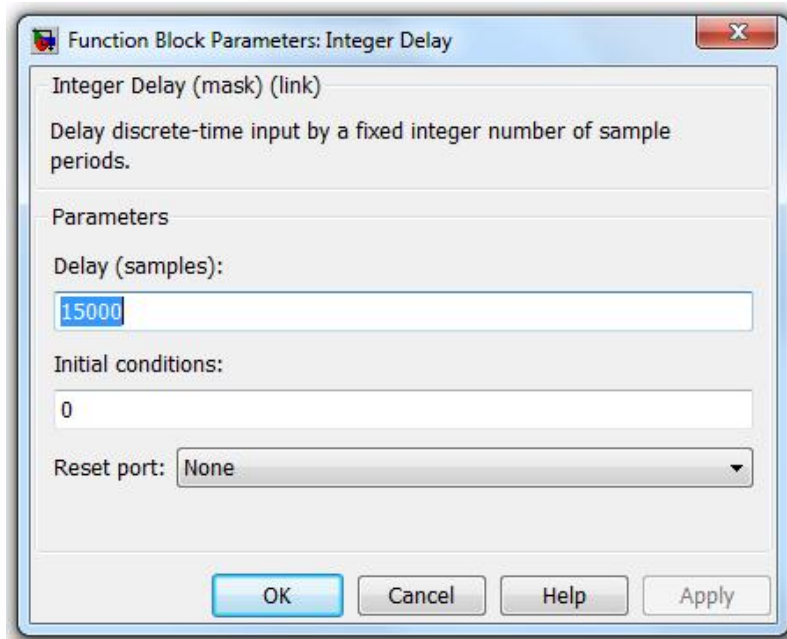
- Η χρονική καθυστέρηση M είναι ο χρόνος μεταξύ των αντιγράφων. Όσον αφορά την μονάδα ηχώ, είναι η καθυστέρηση μεταξύ του στεγνού σήματος και του μοναδικού χρονικά καθυστερημένου αντιγράφου. Η τιμή αυτή μπορεί να κυμαίνεται από μερικά χιλιοστά του δευτερολέπτου μέχρι μερικά δευτερόλεπτα. Στην πρώτη περίπτωση, το αποτέλεσμα θα είναι μια γρήγορη επανάληψη του σήματος εισόδου που παράγει ένα χρωματισμό στον ήχο. Από την άλλη πλευρά, όταν ο χρόνος καθυστέρησης υπερβαίνει το χρονικό διάστημα των 50 ms, μια ενιαία ηχώ γίνεται αντιληπτή χωριστά από το αρχικό σήμα εισόδου. Όσον αφορά το χρόνο καθυστέρησης της ανάδρασης, το M είναι η χρονική διαφορά μεταξύ όλων των αντιγράφων, καθώς και του στεγνού σήματος.
- Ο παράγοντας mix είναι η μίξη μεταξύ του αρχικού στεγνού σήματος και του επεξεργασμένου. Οι τιμές που μπορεί να πάρει κυμαίνονται από 0 (δηλαδή μόνο το στεγνό σήμα) μέχρι 1 (δηλαδή το ίδιο πλάτος τόσο για το στεγνό σήμα όσο και για το επεξεργασμένο).
- Το κέρδος ανάδρασης g είναι ο συντελεστής με τον οποίο κάθε αντίγραφο εξασθενεί κάθε φορά που περνά από το βρόγχο ανάδρασης. Όσο μεγαλύτερη είναι η τιμή του, τόσο πιο πολλά αντίγραφα θα υπάρχουν στο σήμα εξόδου. Αυτή η τιμή δεν πρέπει να υπερβαίνει την μονάδα, διαφορετικά το σήμα θα αυξάνει διαρκώς και το σύστημα θα είναι ασταθές. Ως εκ τούτου, κυμαίνεται από 0 έως 1.

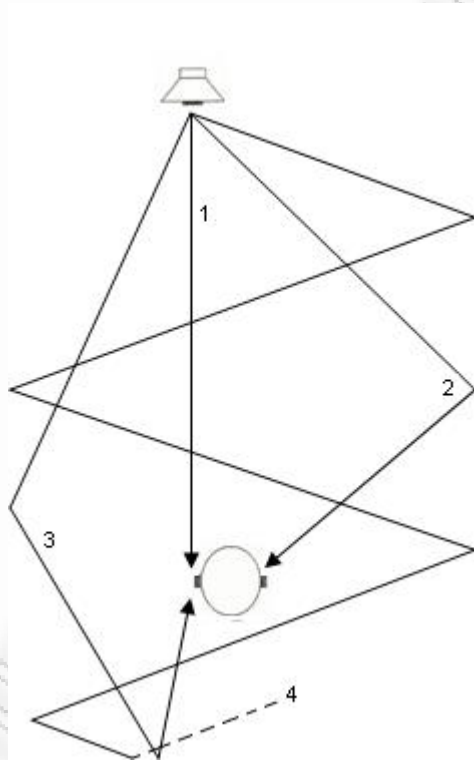
5.4.3 Υλοποίηση στο Simulink

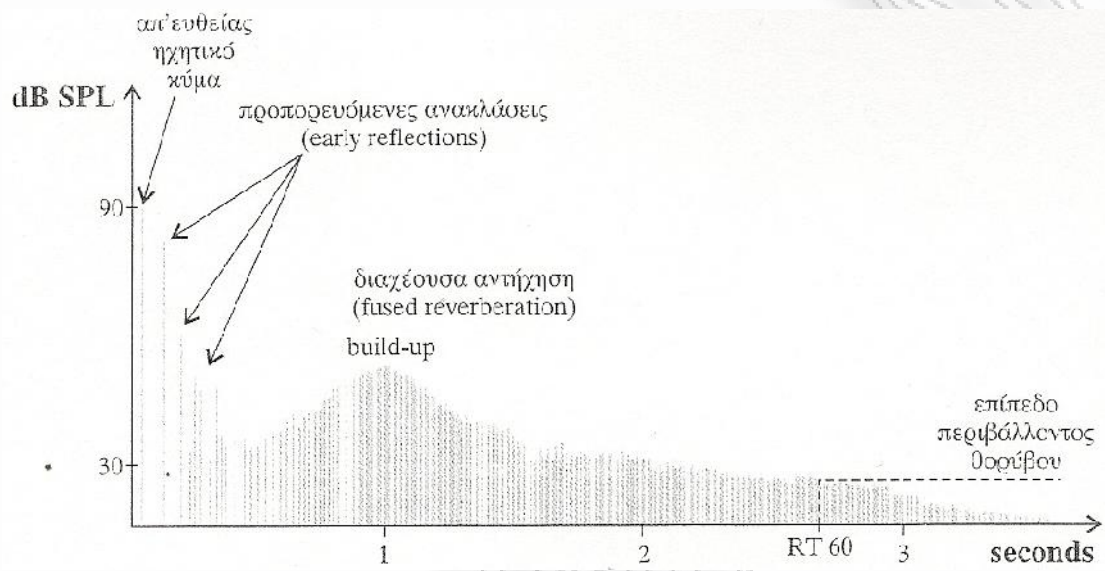
Περιγραφή μοντέλου του Delay effect

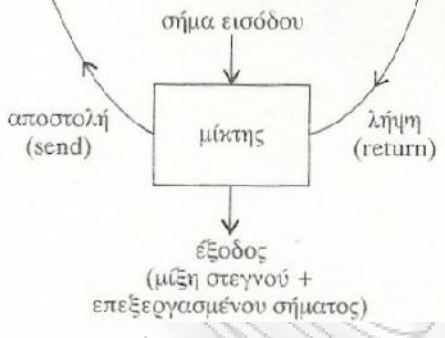
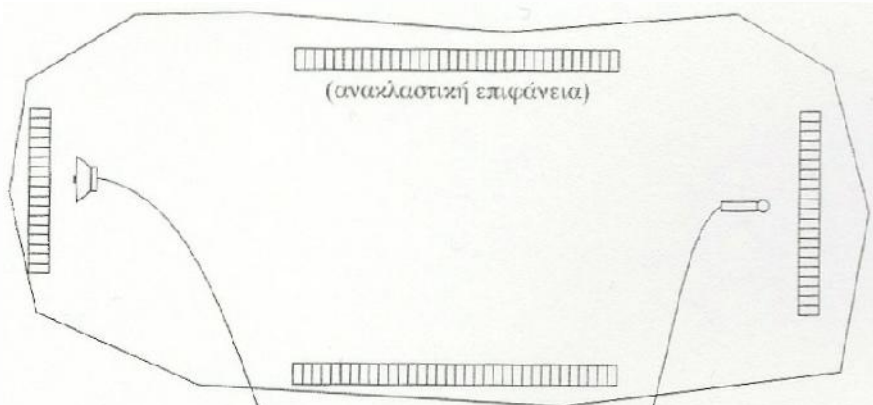
Το μοντέλο του Delay effect έχει την παρακάτω μορφή σε σχέση με το μοντέλο του συνολικού συστήματος.



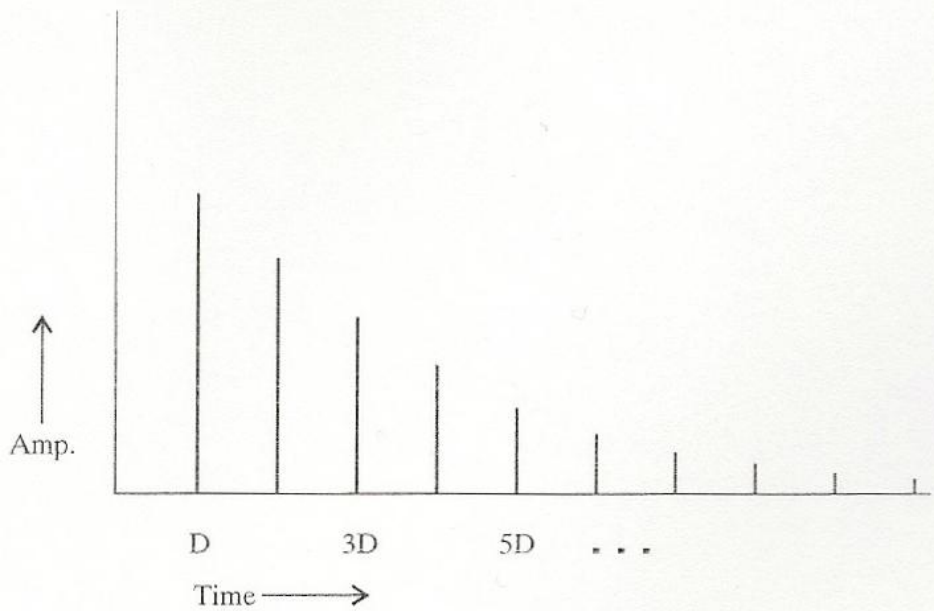
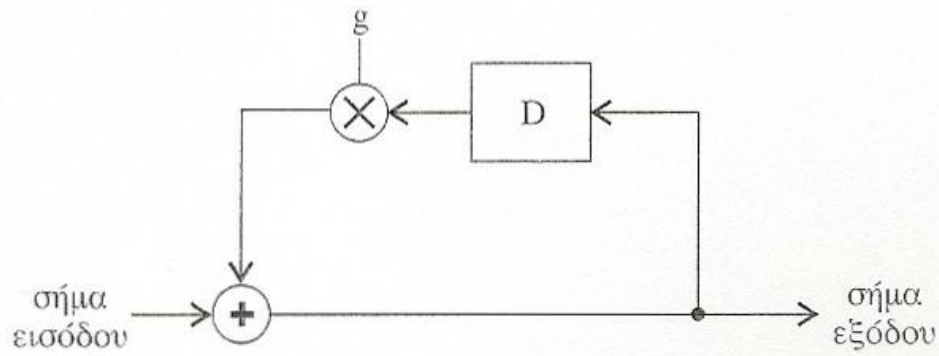








ΠΑΝΕΠΙΣΤΗΜΙΟΝ



g
0.7
0.5
0.1

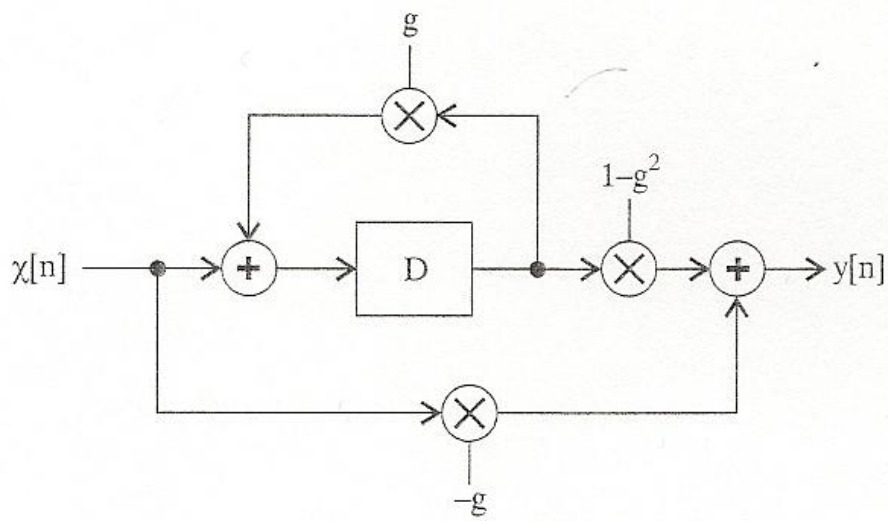
Χρόνος απόσβεσης (-60 dB)
 $19,367 \cdot D$
 $9,96 \cdot D$
 $3 \cdot D$

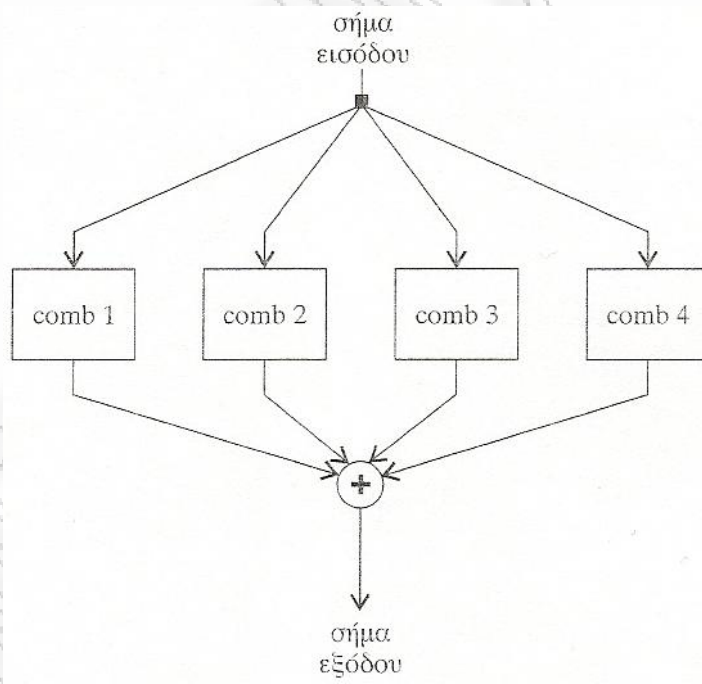
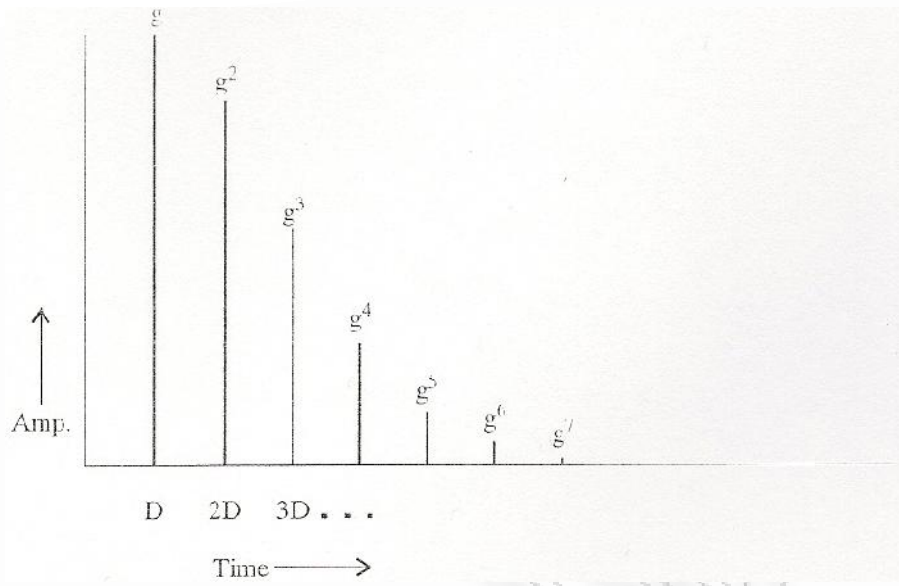
$n(xD)$

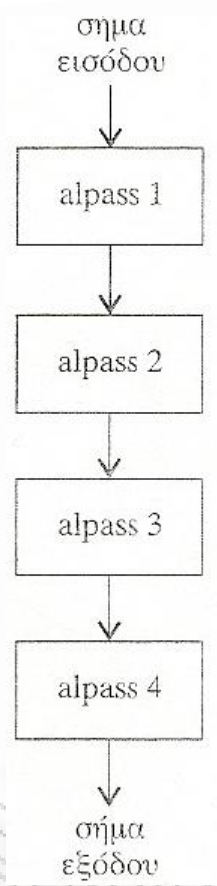
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10

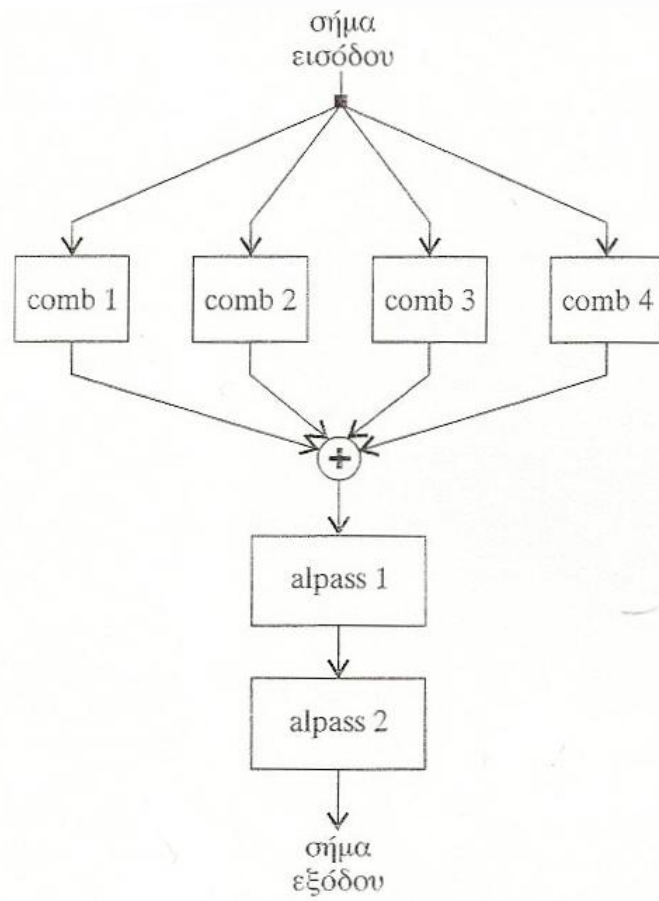
α (dB)

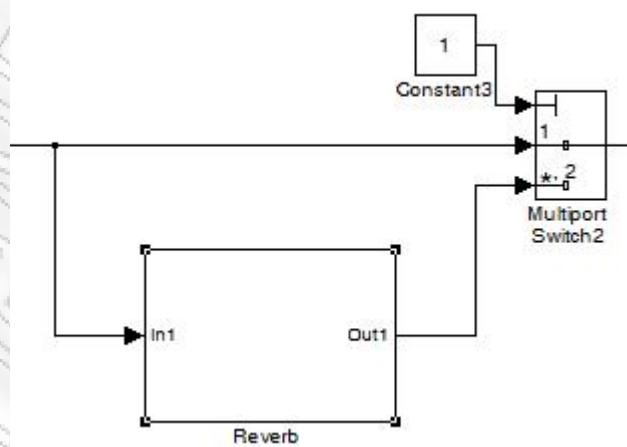
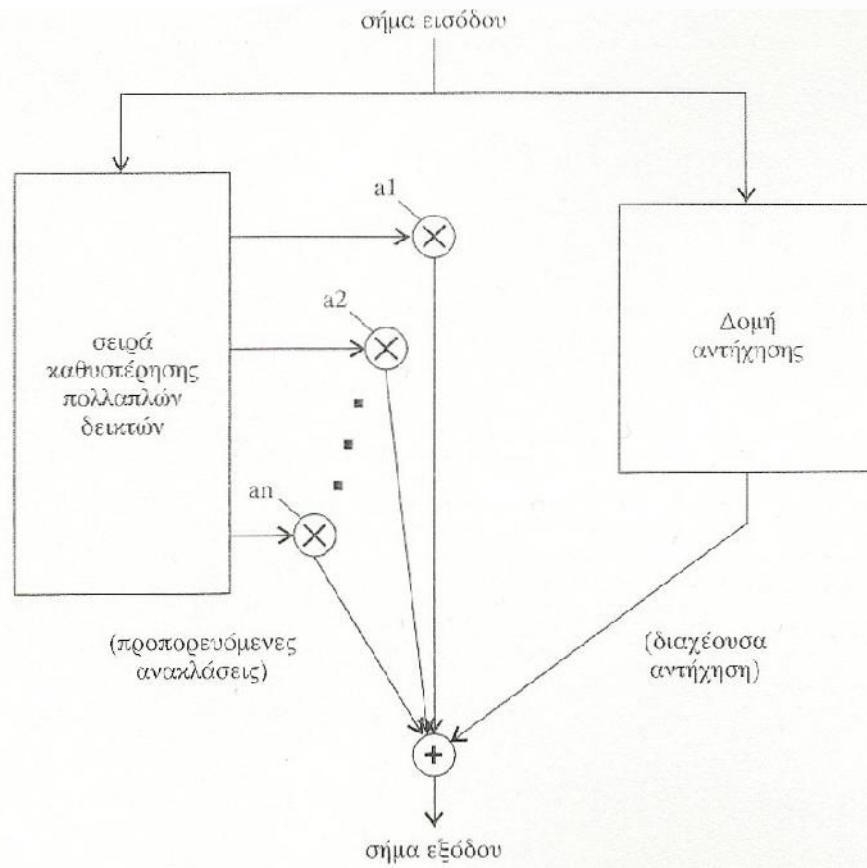
- 6
- 12
- 18
- 24.1
- 30.1
- 36.1
- 42.1
- 48.1
- 54.2
- 60.2

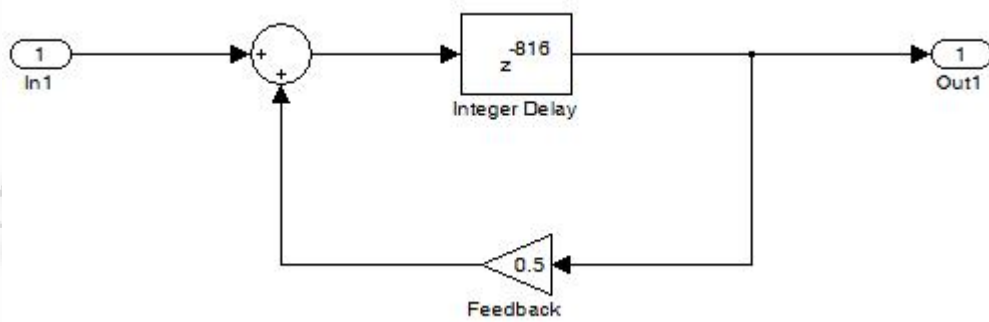
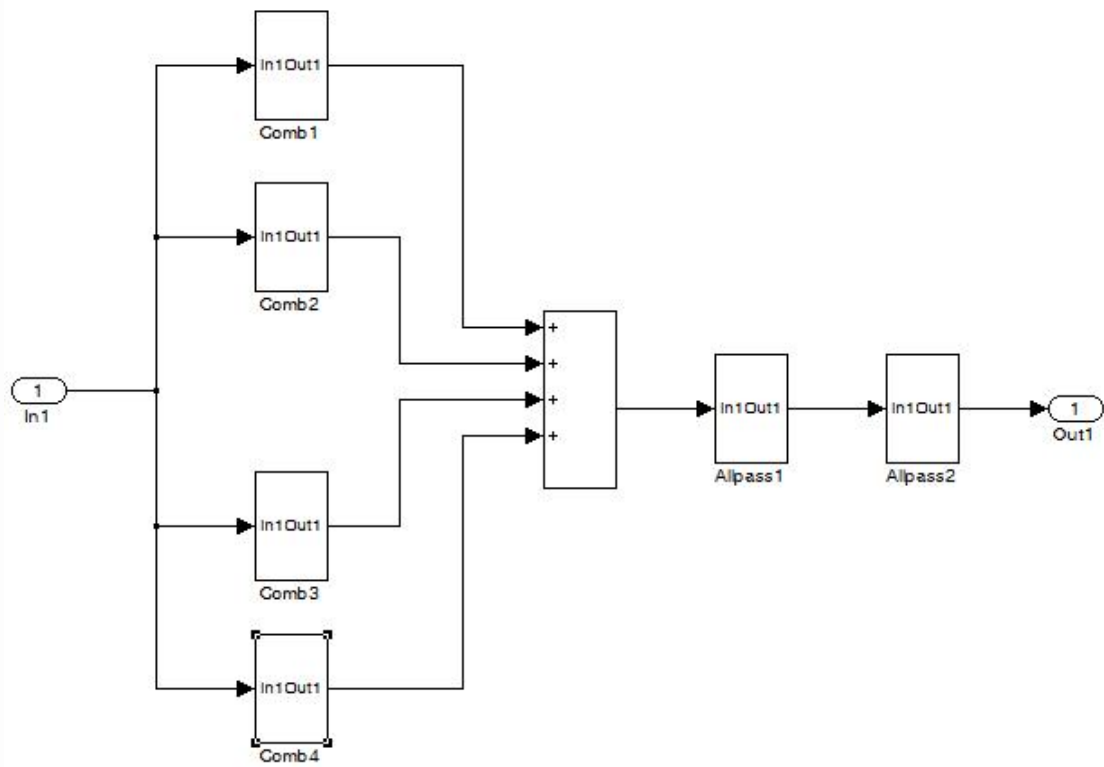


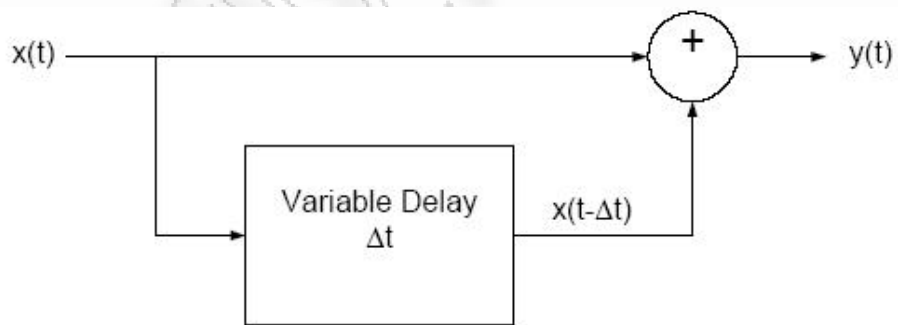
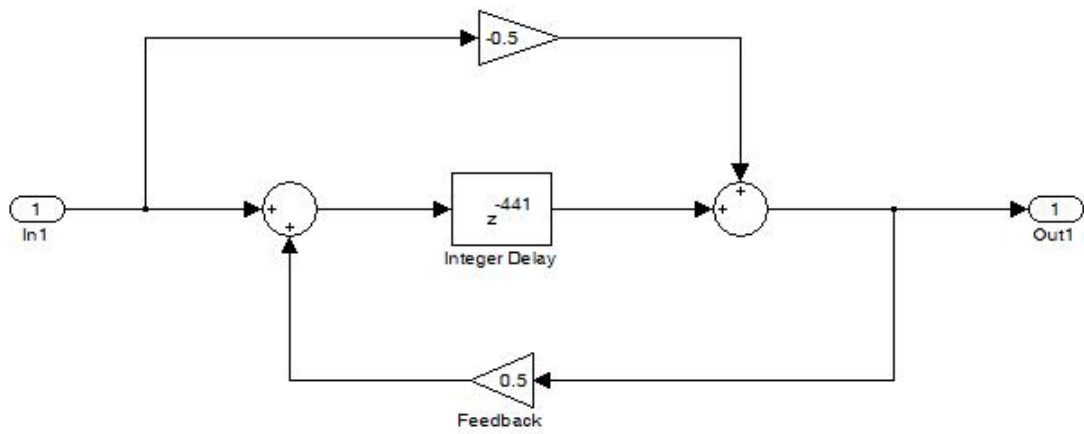








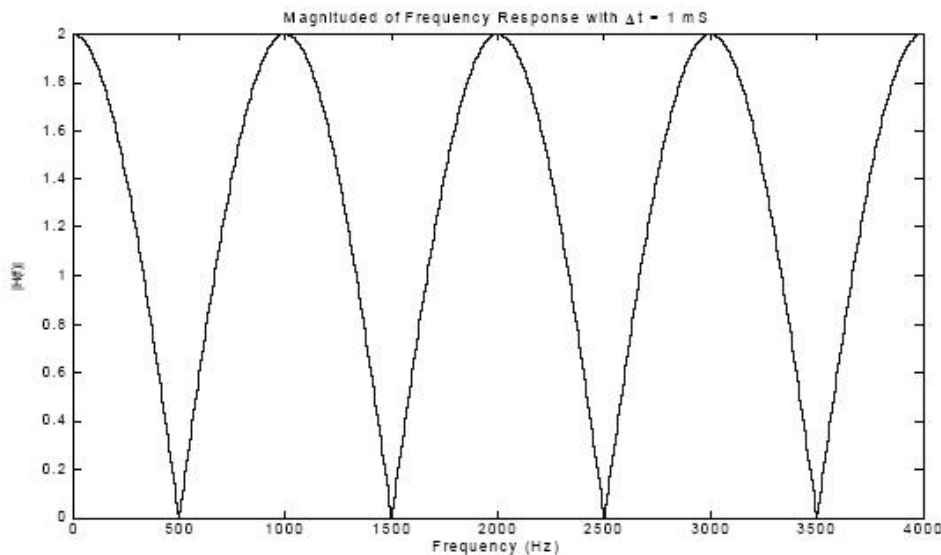




$$y(t) = x(t) + ax(t - \Delta t)$$

$$|H(f)| = 2 \left| \cos\left(\frac{2\pi f \Delta t}{2}\right) \right|$$

όπου Δt η στιγμιαία χρονική καθυστέρηση που καθορίζεται από τον ταλαντωτή χαμηλών συχνοτήτων LFO. Η εξασθένιση συγκεκριμένων συχνοτήτων δημιουργεί εγκοπές στο συχνοτικό φάσμα, κάνοντας έτσι το flanger να μοιάζει με ένα φίλτρο κτένας. Το μέγεθος (magnitude) της συχνοτικής απόκρισης του φίλτρου για $\Delta t = 1$ ms παρουσιάζεται στο παρακάτω σχήμα.



Εικόνα 5.72: Μέγεθος συχνοτικής απόκρισης για $\Delta t = 1$ ms

Είναι σημαντικό να σημειωθεί ότι το flanger διαφέρει από ένα παραδοσιακό φίλτρο κτένας στο γεγονός ότι οι συχνότητες τις οποίες εξασθενεί μεταβάλλονται με το χρόνο. Αυτό είναι το άμεσο αποτέλεσμα του μεταβαλλόμενου χρόνου καθυστέρησης, το οποίο προκαλεί τα «δόντια» του φίλτρου κτένας σε συνεχή κίνηση πάνω και κάτω στο συχνοτικό φάσμα. Οι συχνότητες οι οποίες εξασθενούν (σε Hz) δίνονται από την ακόλουθη εξίσωση:

$$f_{z=0} = \frac{2n+1}{2\Delta t}$$

όπου $n = 0, 1, 2, \dots$. Έτσι, οι συχνότητες που εξασθενούν είναι αρμονικά συνδεδεμένες και εξαρτημένες από την στιγμιαία καθυστέρηση.

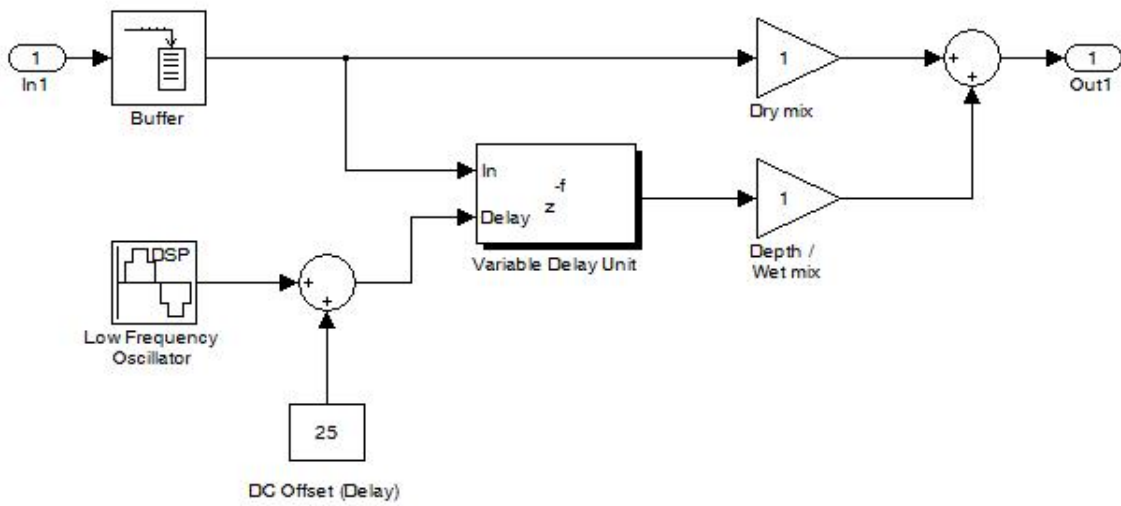
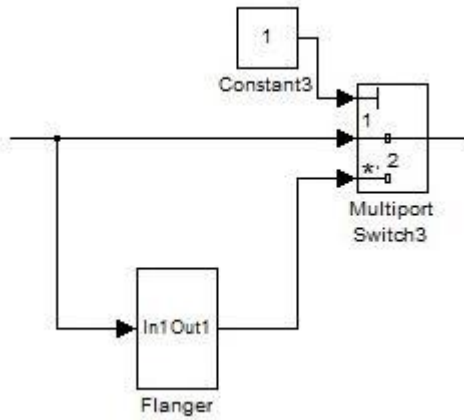
Αυτή η συνεχής αλλαγή των αλλοιωμένων συχνοτήτων και η συνεχής σάρωση των εγκοπών στην συχνοτική απόκριση, γίνεται αντιληπτή από το ανθρώπινο αυτί ως ένας “whooshing” ήχος, ο οποίος είναι χαρακτηριστικός ενός κοινού εφέ flanger. Εάν η στιγμιαία καθυστέρηση είναι μεγαλύτερη από 50 ms, το σήμα το οποίο προκύπτει θα μπορούσε να εκληφθεί ως ηχώ (echo), ωστόσο, σε ένα flanger η χρονική καθυστέρηση είναι συνήθως λιγότερη από 10 ms, καθιστώντας την πολύ γρήγορη για το αυτί να την διακρίνει.

Αρκεί απλά να αντιληφθούμε το flanger σε οποιαδήποτε δεδομένη χρονική στιγμή ως ένα φίλτρο κτένας και να έχουμε κατά νου ότι οι συχνότητες οι οποίες εξαλείφονται από το φίλτρο είναι συνάρτηση της τιμής της στιγμιαίας χρονικής καθυστέρησης Δt , η οποία με τη σειρά της ελέγχεται από έναν ταλαντωτή χαμηλών συχνοτήτων (LFO).

5.6.2 Υλοποίηση στο Simulink

Περιγραφή μοντέλου του Flanger effect

Το μοντέλο του Flanger effect έχει την παρακάτω μορφή σε σχέση με το μοντέλο του συνολικού συστήματος.



Source Block Parameters: Low Frequency Oscillator

Sine Wave (mask) (link)

Output samples of a sinusoid. To generate more than one sinusoid simultaneously, enter a vector of values for the Amplitude, Frequency, and Phase offset parameters.

Main Data Types

Amplitude: 20

Frequency (Hz): 0.7

Phase offset (rad): [-pi/4 pi/4]

Sample mode: Discrete

Output complexity: Real

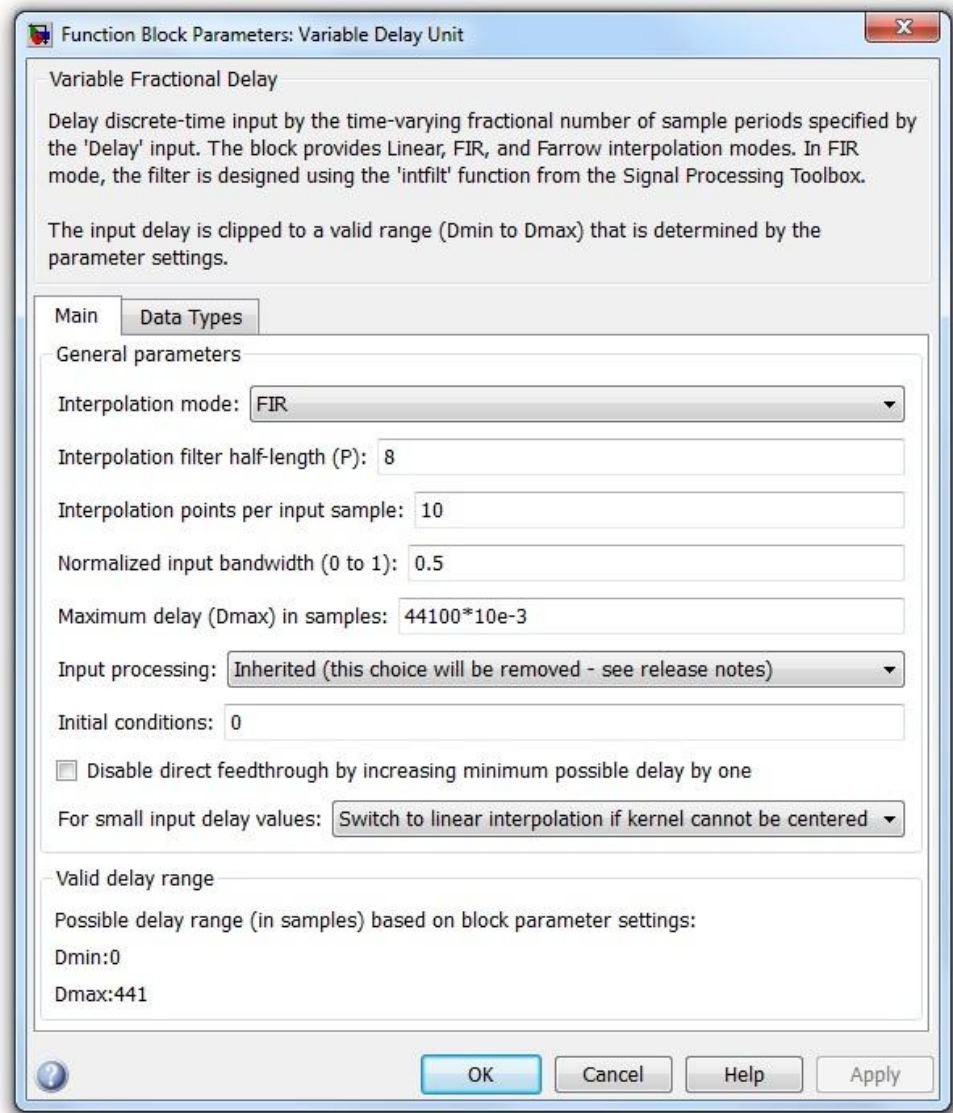
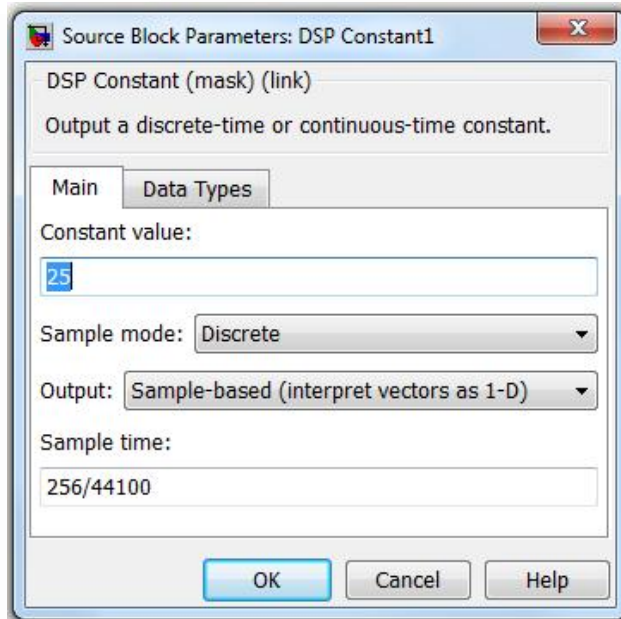
Computation method: Trigonometric fcn

Sample time: 1/44100

Samples per frame: 256

Resetting states when re-enabled: Restart at time zero

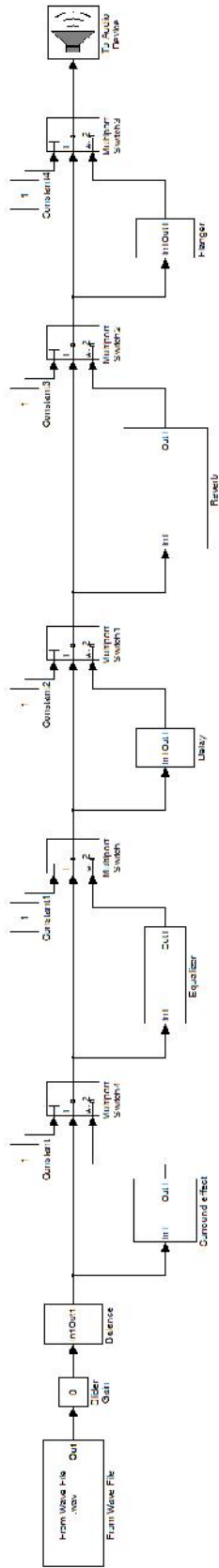
OK Cancel Help



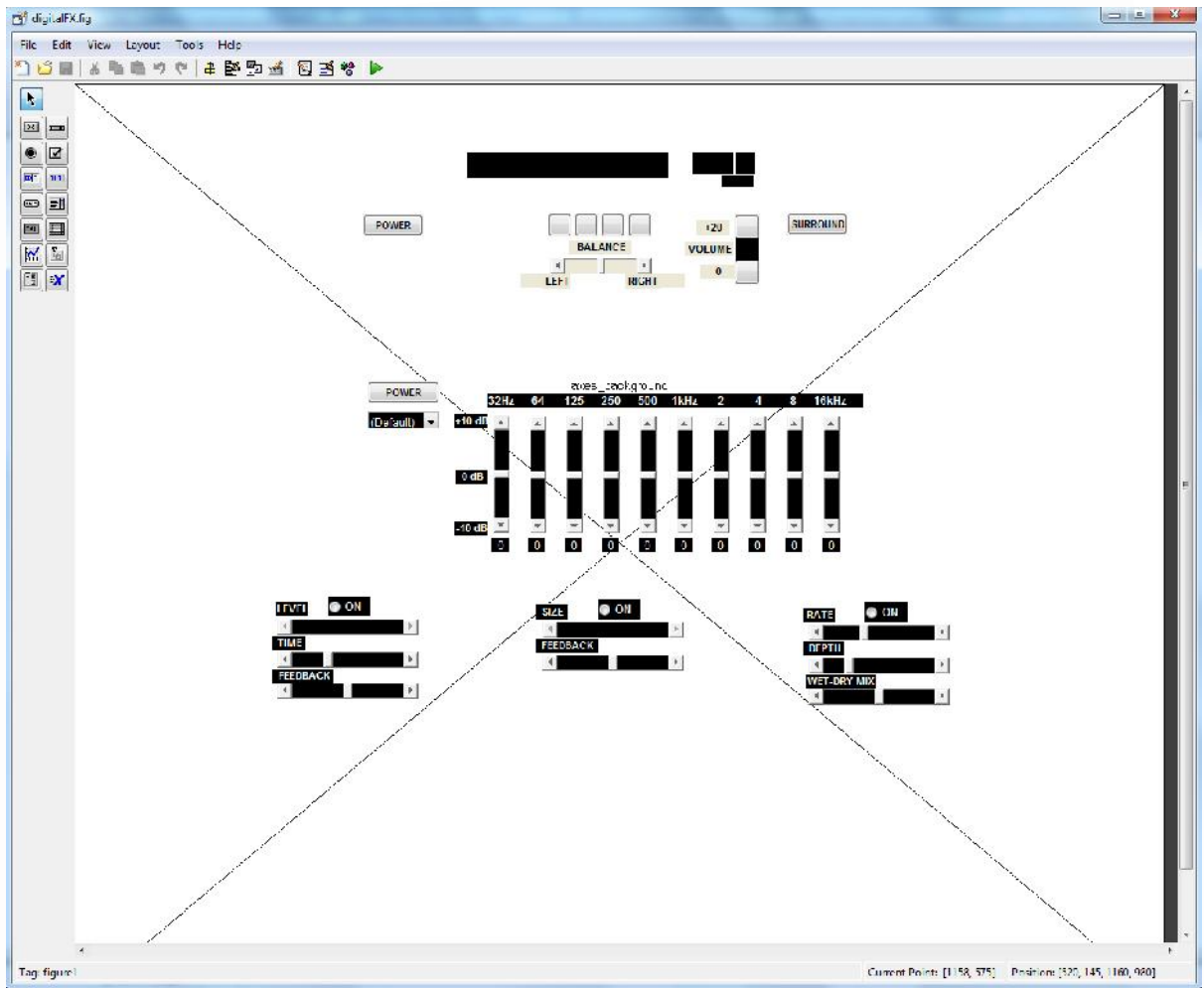
5.7 Συνδυασμός των εφέ

Το πρόγραμμα που υλοποιήσαμε παρουσιάζεται ως μια μοναδική ενότητα που σχηματίζεται από τα εφέ που εξηγήσαμε παραπάνω. Η επεξεργασία που υπόκειται το σήμα αποτελείται από πέντε εφέ, διαιρούμενα σε πέντε χωριστά μπλοκ: τον ενισχυτή σήματος, τον ισοσταθμιστή, το delay, το reverb και το flanger. Αφού τα παραπάνω εφέ λειτουργούν ατομικά το κάθε ένα από αυτά, ο συνδυασμός όλων αυτών είναι απαραίτητος για την υλοποίηση του προγράμματος. Για το σκοπό αυτό, ένα σύστημα παράκαμψης (on/off) για κάθε εφέ θεωρείται απαραίτητο, έτσι ώστε ο χρήστης να έχει τη δυνατότητα να επιλέξει πότε θα χρησιμοποιήσει κάποιο από τα εφέ σε κάθε περίπτωση.

Τα εφέ συνδέονται σειριακά μεταξύ τους σε μια ενιαία γραμμή. Αυτό σημαίνει ότι το αποτέλεσμα της επεξεργασίας ενός εφέ αποτελεί την είσοδο του επόμενου εφέ. Σαν αποτέλεσμα, ο χρήστης μπορεί να επιλέξει πότε και ποιά εφέ ή συνδυασμό εφέ θα χρησιμοποιήσει.



UNIVERSITY OF PERPA



PAWEL



PAWEL

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

Συμπεράσματα

Ο στόχος της παρούσας μεταπτυχιακής διατριβής είναι η ανάπτυξη μιας εφαρμογής λογισμικού για την προσομοίωση ενός ψηφιακού συστήματος ήχου το οποίο υλοποιείται στο λογισμικό πακέτο Matlab, και αποτελείται από έναν ενισχυτή, ένα γραφικό ισοσταθμιστή 10 περιοχών και τρία ψηφιακά εφέ ήχου, το delay, το reverb και το flanger. Ο στόχος αυτός έχει επιτευχθεί. Το πρόγραμμα που υλοποιήσαμε όχι μόνο προσομοιώνει τα πιο δημοφιλή εφέ ήχου, αλλά αναπαριστά μια παρόμοια διάταξη με τα επαγγελματικά ψηφιακά συστήματα ήχου.

Κατά το πρώτο μέρος της ανάπτυξης του προγράμματος, η προτεραιότητα ήταν να συλλέξουμε όλα τα εφέ ατομικά, ώστε να λειτουργούν σωστά. Έγινε εκτενής έρευνα σχετικά με την λειτουργία και την εφαρμογή των διαφόρων εφέ που επιλέχθηκαν, οι πιθανές λύσεις συζητήθηκαν και οι βέλτιστες σχετικά με την χρήση επιλέχθηκαν για κάθε περίπτωση. Έτσι, εφέ όπως η ισοστάθμιση, το delay, το reverb και το flanger υλοποιήθηκαν.

Μόλις ο στόχος αυτός επετεύχθη, το δεύτερο μέρος της ανάπτυξης του προγράμματος ήταν επικεντρωμένο στη δημιουργία του ψηφιακού συστήματος ήχου, βασισμένο στο συνδυασμό των επιμέρους εφέ. Το τελικό αποτέλεσμα παρέχει ευελιξία καθώς και ένα φιλικό προς το χρήστη περιβάλλον διεπαφής.

Είναι σημαντικό να επισημάνουμε την απαραίτητη παρουσία των ψηφιακών φίλτρων κατά τη δημιουργία της διατριβής. Αποδείχθηκε ότι τα δύο φίλτρα κτένας FIR και IIR χρησιμοποιήθηκαν στα περισσότερα από τα εφέ, δείχνοντας έτσι την μεγάλη τους σημασία στην ακουστική επεξεργασία σήματος, καθώς και τη χρησιμότητά τους για πολλούς και διάφορους σκοπούς, όπως τη δημιουργία μιας χρονικής καθυστέρησης.

Μιας και το Matlab έχει σχεδιαστεί για τους υπολογισμούς διανυσμάτων και μητρών, αποδείχθηκε ότι έχει σχετικά αργή ταχύτητα για τους σκοπούς της διατριβής αυτής. Για τον λόγο αυτό, δεν ενδείκνυται για την επεξεργασία σήματος σε πραγματικό χρόνο. Παρ' όλα αυτά, το Matlab μπορεί να θεωρηθεί ως κατάλληλο εργαλείο για τη διαδικασία της προσομοίωσης, όπου οι ιδιότητες των φίλτρων και των εφέ που έχουν σχεδιαστεί μπορούν να ελεγχθούν και οι υπολογισμοί μπορούν να γίνουν. Μπορεί να θεωρηθεί ως ένα βήμα προς την υλοποίηση σε πραγματικό χρόνο.

Λαμβάνοντας υπόψη αυτά τα αποτελέσματα, η εφαρμογή του προγράμματος σε πραγματικό χρόνο έχει επιτευχθεί εν μέρει, καθώς και η βιωσιμότητά του έχει αποδειχθεί από την εφαρμογή των εφέ. Τα θέματα αυτά αντιμετωπίστηκαν στο σχεδιασμό του προγράμματος, χτίζοντας μια γραφική διεπαφή χρήστη και παρέχοντας τον ακριβή έλεγχο των παραμέτρων του συστήματος.

Τέλος, μία περαιτέρω βελτίωση του προγράμματος θα μπορούσε να είναι η εφαρμογή και άλλων εφέ όπως το wah-wah, η παραμόρφωση (distortion) και το tremolo, καθώς και η μεγαλύτερη και πιο λεπτομερής παραμετροποίηση των ήδη εφαρμοσμένων εφέ.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΔΑΛΙΑΣ

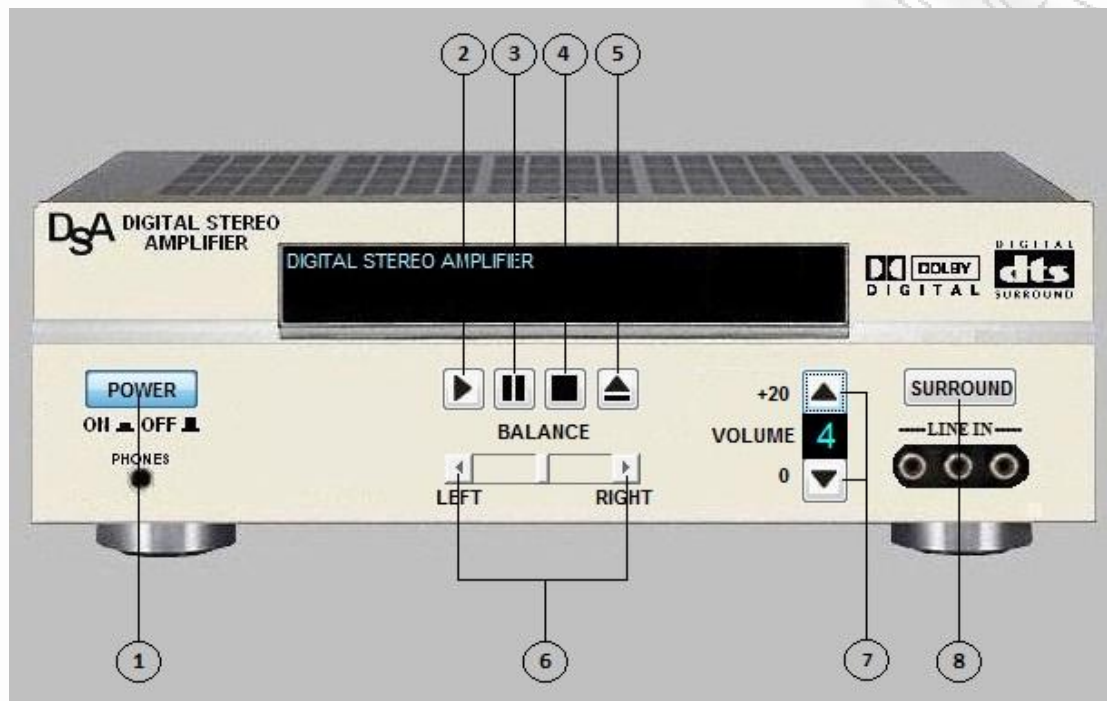
Βιβλιογραφία

- [1] Charles Dodge & Thomas A. Jersey, 1985, *Computer Music. Synthesis, Composition, and Performance*, Schirmer Books
- [2] Curtis Roads, 1999, *The Computer Music Tutorial*, MIT Press
- [3] Eduardo Reck Miranda, 2002, *Computer Sound Design: Synthesis Techniques and Programming*, Focal Press
- [4] F. Alton Everest, 1998, *Εγχειρίδιο ακουστικής*, Εκδόσεις Τζιόλα
- [5] F. Richard Moore, 1990, *Elements of Computer Music*, Prentice Hall
- [6] Glen M. Ballou, 2001, *Handbook for Sound Engineers*, Focal Press
- [7] Meinard Müller, 2007, *Information Retrieval for Music and Motion*, Springer
- [8] Michael Ross, 1998, *Getting Great Guitar Sounds*, Hal Leonard
- [9] Proakis, John G., Manolakis & Dimitris G., 2010, *Ψηφιακή ανάλυση σήματος: Αρχές, αλγόριθμοι, εφαρμογές*, Εκδόσεις Ίων
- [10] Udo Zölzer & Xavier Amatriain, 2002, *DAFX: Digital Audio Effects*, John Wiley and Sons
- [11] Διαμαντόπουλος Ταξιάρχης, 2004, *Προγραμματισμός και Σύθεση Ήχου*, Εκδόσεις Έλλην
- [12] Δώδης Δημήτρης, 2001, *Ηχοληψία: Η δημιουργία με τη σύγχρονη τεχνολογία*, Εκδόσεις Ίων

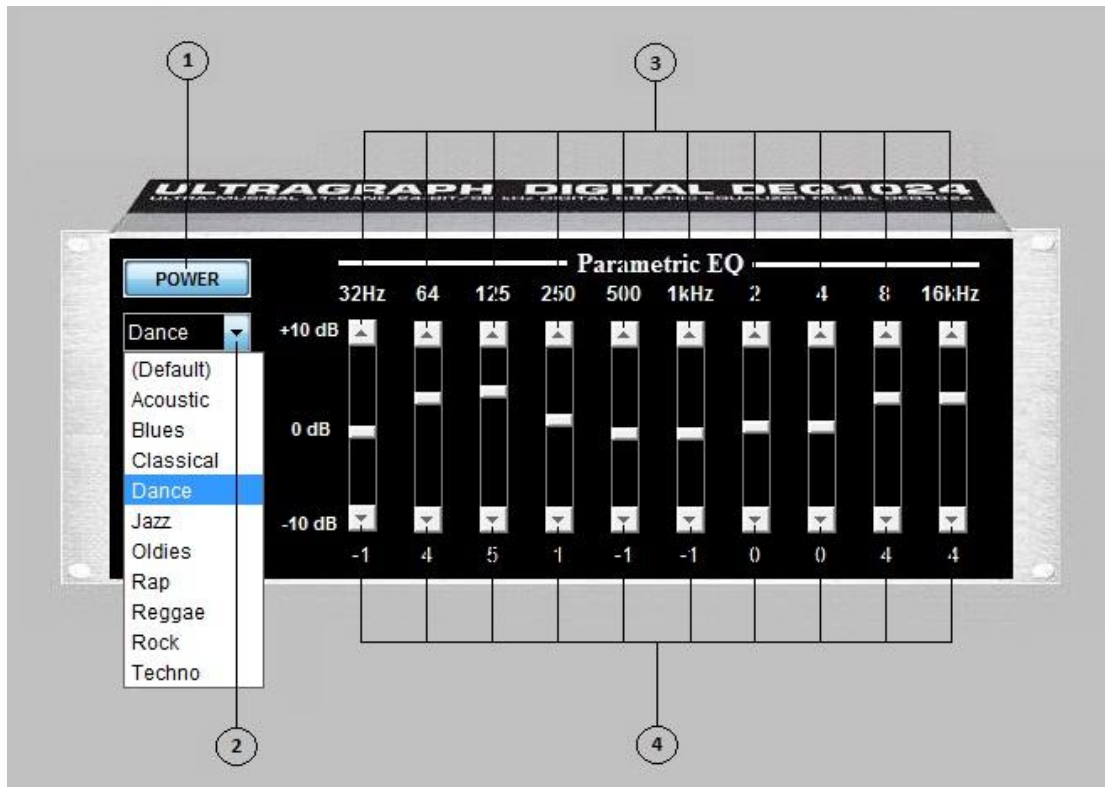
Δικτυακές αναφορές

- <http://www.mathworks.com/>
- <http://www.mathworks.com/matlabcentral/fileexchange/>
- <http://en.wikipedia.org>
- <http://blinkdagger.com/matlab/>
- <http://www.duke.edu/~hpgavin/matlab.html>
- <http://www.geofex.com/>
- <http://www.harmonycentral.com>
- <http://science.howstuffworks.com/environmental/life/human-biology/hearing1.htm>
- <http://cnx.org/content/m22940/latest/>
- <http://www.bores.com/courses/intro/iir/index.htm>

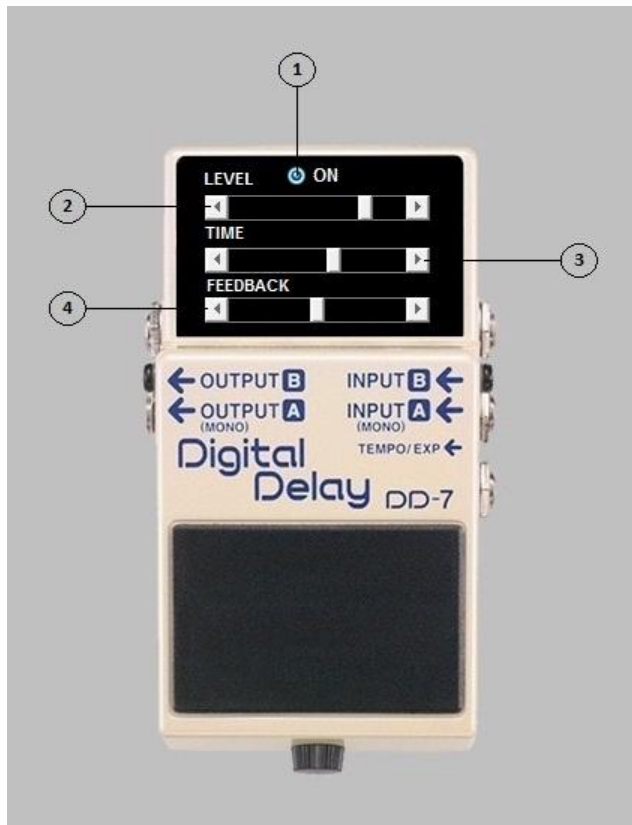
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ



1. POWER. Διακόπτης ενεργοποίησης / απενεργοποίησης (ON/OFF). Πιέζοντας μια φορά το διακόπτη θέτετε τον ενισχυτή σε κατάσταση λειτουργίας, πιέζοντας άλλη μια φορά το κουμπί σβήνετε τον ενισχυτή.
2. PLAY button. Πιέζοντας το κουμπί Play θέτετε σε αναπαραγωγή το αρχείο ήχου.
3. PAUSE button. Πιέζοντας μια φορά το κουμπί θέτετε προσωρινά σε παύση την αναπαραγωγή του αρχείου ήχου, πιέζοντας άλλη μια φορά το κουμπί θέτετε ξανά σε αναπαραγωγή το αρχείο ήχου από το χρονικό σημείο της παύσης.
4. STOP button. Πιέζοντας το κουμπί Stop διακόπτετε την αναπαραγωγή του αρχείου ήχου.
5. OPEN button. Πιέζοντας το κουμπί Open μπορείτε να εκτελέσετε αναζήτηση και να επιλέξετε το αρχείο ήχου που θέλετε να αναπαραγάγετε.
6. BALANCE left / right button. Πιέζοντας το αριστερό / δεξιό κουμπί ρυθμίζετε την ισορροπία μεταξύ αριστερών και δεξιών ηχείων η οποία μετακινείται προς τα αριστερά / δεξιά. Η ένδειξη Left 10 έως Right 10 εμφανίζεται καθώς η ισορροπία μεταξύ αριστερών / δεξιών ηχείων μετακινείται από αριστερά προς τα δεξιά.
7. VOLUME + / - button. Πιέζοντας το κουμπί + / - αυξάνετε / μειώνετε την ένταση του ήχου που αναπαραγάγει αντίστοιχα. Η ένδειξη 0 έως 20 εμφανίζεται καθώς αυξομειώνετε την ηχητική ένταση.
8. SURROUND button. Πιέζοντας το κουμπί Surround δημιουργείται στον ακροατή η αίσθηση της θέσης των ηχητικών πηγών στον χώρο ακρόασης.



1. POWER. Διακόπτης ενεργοποίησης / απενεργοποίησης (ON/OFF). Πιέζοντας μια φορά το διακόπτη θέτετε τον ισοσταθμιστή σε κατάσταση λειτουργίας, πιέζοντας άλλη μια φορά το κουμπί σβήνετε τον ισοσταθμιστή.
 2. EQ PRESETS. Πιέζοντας το κουμπί EQ Presets εμφανίζεται μια λίστα όπου μπορείτε να επιλέξετε ανάμεσα σε δέκα (10) αποθηκευμένες καμπύλες ισοστάθμισης, ανακαλώντας τες εύκολα οποιαδήποτε στιγμή. Οι καμπύλες αυτές είναι: Acoustic, Blues, Classical, Dance, Jazz, Oldies, Rap, Reggae, Rock, Techno.
 3. EQ + button. Πιέζοντας το κουμπί + η στάθμη της περιοχής συχνότητων ισοσταθμιστή αυξάνεται σε dB. Η ένδειξη -10 έως +10 εμφανίζεται καθώς αυξάνεται η στάθμη. (*)
 4. EQ - button. Πιέζοντας το κουμπί - η στάθμη της περιοχής συχνότητων ισοσταθμιστή μειώνεται σε dB. Η ένδειξη -10 έως +10 εμφανίζεται καθώς μειώνεται η στάθμη. (*)
- * Οι επιθυμητές περιοχές συχνότητων ισοσταθμιστή για ρύθμιση είναι οι εξής: 32Hz, 64, 125, 250, 500, 1KHz, 2, 4, 8, 16KHz



1. On. Διακόπτης ενεργοποίησης / απενεργοποίησης (ON/OFF). Πιέζοντας μια φορά το κουμπί επιλογής θέτετε το πετάλι Delay σε κατάσταση λειτουργία, πιέζοντας άλλη μια φορά το κουμπί σβήνετε το πετάλι.
2. Level. Καθορίζει την ένταση του ήχου με Delay. Ρυθμίζοντας το slider προς τα δεξιά αυξάνουμε την ένταση του ήχου με Delay. Στην τέρμα δεξιά θέση, η ένταση του ήχου με Delay ισούται με αυτήν του αρχικού σήματος. Στην τέρμα αριστερά θέση, η ηχητική ένταση του αρχικού μη επεξεργασμένου σήματος είναι μηδενική.
3. Delay Time. Καθορίζει τον αριθμό των περιόδων του δείγματος στην καθυστέρηση του αρχικού σήματος.
4. Feedback. Καθορίζει τον αριθμό επαναλήψεων του ήχου με Delay. Ρυθμίζοντας το slider προς τα δεξιά αυξάνουμε τον αριθμό επαναλήψεων του ήχου με Delay,

Λειτουργίες κουμπιών και ρυθμιστικών του πεταλιού Reverb



1. On. Διακόπτης ενεργοποίησης / απενεργοποίησης (ON/OFF). Πιέζοντας μια φορά το κουμπί επιλογής θέτετε το πετάλι Reverb σε κατάσταση λειτουργία, πιέζοντας άλλη μια φορά το κουμπί σβήνετε το πετάλι.
2. Size. Καθορίζει το μήκος χρόνου της αντήχησης του ήχου, ανάλογα με το μέγεθος του δωματίου. Όσο μεγαλύτερο είναι το μέγεθος, τόσο μεγαλύτερο θα είναι και το μήκος χρόνου της αντήχησης του ήχου.
3. Feedback. Καθορίζει τον αριθμό επαναλήψεων του ήχου με Reverb. Ρυθμίζοντας το slider προς τα δεξιά αυξάνουμε τον αριθμό επαναλήψεων του ήχου με Reverb, ενώ ρυθμίζοντάς το προς τα αριστερά, τον μειώνουμε.

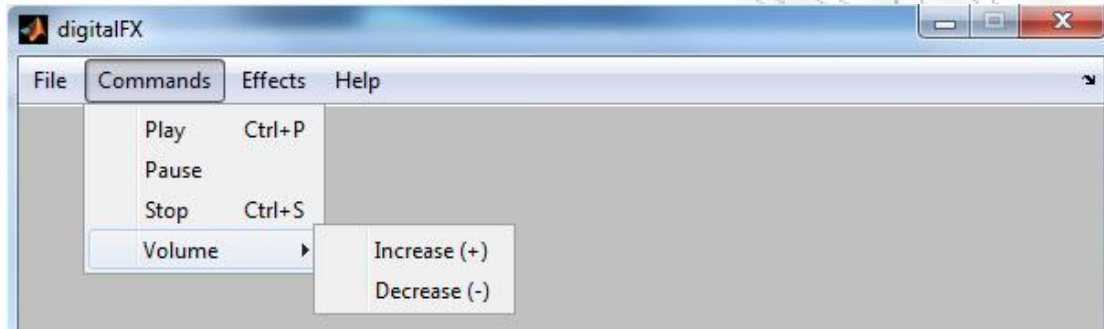
Τα κουμπιά ρυθμίσεως του πεταλιού Reverb παραμένουν ανενεργά όσο το πετάλι είναι απενεργοποιημένο, ενώ τίθενται σε λειτουργία όταν το ενεργοποιήσουμε.



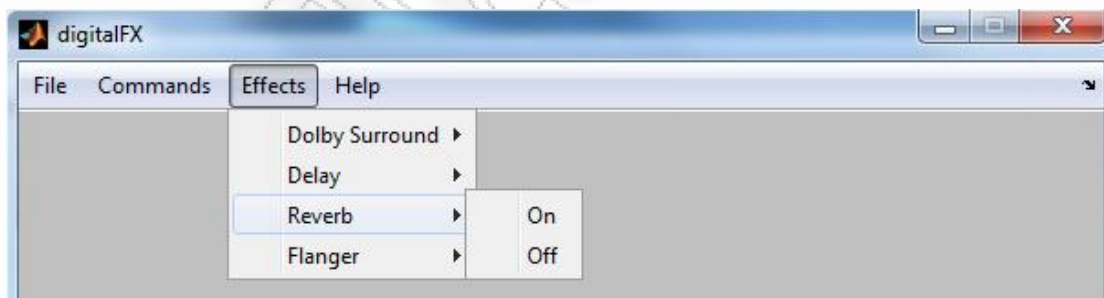
1. On. Διακόπτης ενεργοποίησης / απενεργοποίησης (ON/OFF). Πιέζοντας μια φορά το κουμπί επιλογής θέτετε το πετάλι Flanger σε κατάσταση λειτουργία, πιέζοντας άλλη μια φορά το κουμπί σβήνετε το πετάλι.
2. Rate. Καθορίζει την ταχύτητα, το ρυθμό με τον οποίο θα ολοκληρωθεί ο κύκλος της αλλοίωσης της συχνότητας (του ξεκουρδίσματος δηλ.) και της επαναφοράς στη σωστή συχνότητα. Ρυθμίζοντας το slider προς τα δεξιά, αυξάνουμε την ταχύτητα του flange εφέ.
3. Depth. Καθορίζει το μέγεθος της αλλοίωσης της συχνότητας, δηλαδή πόσο έντονη ή διακριτική θα είναι η αλλοίωση του τόνου. Ρυθμίζοντας το slider προς τα δεξιά, προσδίδουμε περισσότερο βάθος στο flange εφέ.
4. Wet-Dry Mix. Μας επιτρέπει να κάνουμε μια μίξη μεταξύ του πρωτογενούς σήματος (του καθαρού δηλ. όπως βγαίνει από



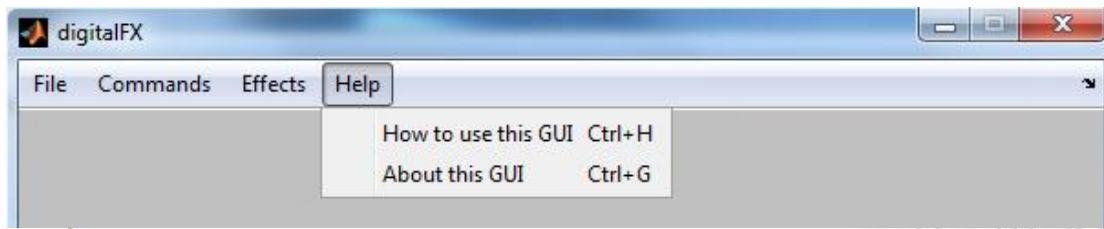
- File ➤ Open: Πιέζοντας το κουμπί Open μπορείτε να εκτελέσετε αναζήτηση και να επιλέξετε το αρχείο ήχου που θέλετε να αναπαράγετε.
- File ➤ Exit: Πιέζοντας το κουμπί Exit τερματίζετε την εφαρμογή.



- Commands ➤ Play: Πιέζοντας το κουμπί Play θέτετε σε αναπαραγωγή το αρχείο ήχου.
- Commands ➤ Pause:
- Commands ➤ Stop: Πιέζοντας το κουμπί Stop διακόπτετε την αναπαραγωγή του αρχείου ήχου.
- Commands ➤ Volume ➤ Increase (+): Πιέζοντας το κουμπί (+) αυξάνετε την ένταση του ήχου που αναπαράγεται.
- Commands ➤ Volume ➤ Decrease (-): Πιέζοντας το κουμπί (-) μειώνετε την ένταση του ήχου που αναπαράγεται.



- Effects ➤ Dolby Surround ➤ On: Ενεργοποίηση του εφέ Dolby Surround.
- Effects ➤ Dolby Surround ➤ Off: Απενεργοποίηση του εφέ Dolby Surround.
- Effects ➤ Delay ➤ On: Ενεργοποίηση του εφέ Delay.
- Effects ➤ Delay ➤ Off: Απενεργοποίηση του εφέ Delay.
- Effects ➤ Reverb ➤ On: Ενεργοποίηση του εφέ Reverb.
- Effects ➤ Reverb ➤ Off: Απενεργοποίηση του εφέ Reverb.
- Effects ➤ Flanger ➤ On: Ενεργοποίηση του εφέ Flanger.
- Effects ➤ Flanger ➤ Off: Απενεργοποίηση του εφέ Flanger.



Help ➤ How to use this GUI: Οδηγίες χρήσης της εφαρμογής.
Help ➤ About this GUI: Πληροφορίες της εφαρμογής.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΠΑ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΔΑΛΙΑΣ

Παράρτημα Β: Κώδικας του προγράμματος στο Matlab

```

function varargout = digitalFX(varargin)
% DIGITALFX M-file for digitalFX.fig
%     DIGITALFX, by itself, creates a new DIGITALFX or raises the
existing
%     singleton*.
%
%     H = DIGITALFX returns the handle to a new DIGITALFX or the
handle to
%     the existing singleton*.
%
%     DIGITALFX('CALLBACK',hObject,eventData,handles,...) calls the
local
%     function named CALLBACK in DIGITALFX.M with the given input
arguments.
%
%     DIGITALFX('Property','Value',...) creates a new DIGITALFX or
raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before digitalFX_OpeningFcn gets called.
An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to digitalFX_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help digitalFX

% Last Modified by GUIDE v2.5 21-Oct-2011 12:47:06

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @digitalFX_OpeningFcn, ...
                  'gui_OutputFcn',  @digitalFX_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

```

```

% --- Executes just before digitalFX is made visible.
function digitalFX_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to digitalFX (see VARARGIN)

% Choose default command line output for digitalFX
handles.output = hObject;

find_system('Name','dfx');open_system('dfx');
%import the background image into matlab
%if image is not in the same directory as the GUI files, you must use
the
%full path name of the image file
backgroundImage = importdata('background.jpg');
%select the axes
axes(handles.axes_background);
%place image onto the axes
image(backgroundImage);
%remove the axis tick marks
axis off

%import the play image into matlab
%if image is not in the same directory as the GUI files, you must use
the
%full path name of the image file
play = importdata('play.jpg');
%set the play image as the button background
set(handles.pushbutton_play, 'CDATA',play);

%import the pause image into matlab
%if image is not in the same directory as the GUI files, you must use
the
%full path name of the image file
pause = importdata('pause.jpg');
%set the pause image as the button background
set(handles.togglebutton_pause, 'CDATA',pause);

%import the stop image into matlab
%if image is not in the same directory as the GUI files, you must use
the
%full path name of the image file
stop = importdata('stop.jpg');
%set the stop image as the button background
set(handles.pushbutton_stop, 'CDATA',stop);

%import the open image into matlab
%if image is not in the same directory as the GUI files, you must use
the
%full path name of the image file
open = importdata('open.jpg');
%set the open image as the button background
set(handles.pushbutton_open, 'CDATA',open);

%import the volume_up image into matlab
%if image is not in the same directory as the GUI files, you must use
the

```



```

%full path name of the image file
open = importdata('volume_up.jpg');
%set the volume_up image as the button background
set(handles.volume_up, 'CDATA', open);

%import the volume_down image into matlab
%if image is not in the same directory as the GUI files, you must use
the
%full path name of the image file
open = importdata('volume_down.jpg');
%set the volume_down image as the button background
set(handles.volume_down, 'CDATA', open);

set_param('dfx/Slider Gain', 'Gain', num2str(0));
set_param('dfx/Balance/Constant_right', 'Value', num2str(1));
set_param('dfx/Balance/Constant_left', 'Value', num2str(1));
set_param('dfx/Equalizer/dB Gain1', 'dB', num2str(0));
set_param('dfx/Equalizer/dB Gain2', 'dB', num2str(0));
set_param('dfx/Equalizer/dB Gain3', 'dB', num2str(0));
set_param('dfx/Equalizer/dB Gain4', 'dB', num2str(0));
set_param('dfx/Equalizer/dB Gain5', 'dB', num2str(0));
set_param('dfx/Equalizer/dB Gain6', 'dB', num2str(0));
set_param('dfx/Equalizer/dB Gain7', 'dB', num2str(0));
set_param('dfx/Equalizer/dB Gain8', 'dB', num2str(0));
set_param('dfx/Equalizer/dB Gain9', 'dB', num2str(0));
set_param('dfx/Equalizer/dB Gain10', 'dB', num2str(0));
set_param('dfx/Constant', 'Value', num2str(1));
set_param('dfx/Constant1', 'Value', num2str(1));
set_param('dfx/Constant2', 'Value', num2str(1));
set_param('dfx/Constant3', 'Value', num2str(1));
set_param('dfx/Flanger/Gain1', 'Gain', num2str(1));
set_param('dfx/Flanger/Gain2', 'Gain', num2str(1));
set_param('dfx/Flanger/Low Frequency
Oscillator', 'Amplitude', num2str(20));
set_param('dfx/Flanger/Low Frequency
Oscillator', 'Frequency', num2str(0.7));
set_param('dfx/Reverb/Comb1/Integer Delay', 'Delay', num2str(816));
set_param('dfx/Reverb/Comb2/Integer Delay', 'Delay', num2str(816*2));
set_param('dfx/Reverb/Comb3/Integer Delay', 'Delay', num2str(816*3));
set_param('dfx/Reverb/Comb4/Integer Delay', 'Delay', num2str(816*4));
set_param('dfx/Reverb/Allpass1/Integer Delay', 'Delay', num2str(441));
set_param('dfx/Reverb/Allpass2/Integer Delay', 'Delay', num2str(713));
set_param('dfx/Reverb/Comb1/Feedback', 'Gain', num2str(0.5));
set_param('dfx/Reverb/Comb2/Feedback', 'Gain', num2str(0.5));
set_param('dfx/Reverb/Comb3/Feedback', 'Gain', num2str(0.5));
set_param('dfx/Reverb/Comb4/Feedback', 'Gain', num2str(0.5));
set_param('dfx/Reverb/Allpass1/Feedback', 'Gain', num2str(0.5));
set_param('dfx/Reverb/Allpass2/Feedback', 'Gain', num2str(0.5));
set_param('dfx/Delay/Gain', 'Gain', num2str(0.8));
set_param('dfx/Delay/Integer Delay', 'Delay', num2str(15000));
set_param('dfx/Delay/Feedback', 'Gain', num2str(0.5));
set(handles.balance, 'Enable', 'Inactive');
set(handles.freq1_var, 'Enable', 'Inactive');
set(handles.freq2_var, 'Enable', 'Inactive');
set(handles.freq3_var, 'Enable', 'Inactive');
set(handles.freq4_var, 'Enable', 'Inactive');
set(handles.freq5_var, 'Enable', 'Inactive');
set(handles.freq6_var, 'Enable', 'Inactive');
set(handles.freq7_var, 'Enable', 'Inactive');
set(handles.freq8_var, 'Enable', 'Inactive');
set(handles.freq9_var, 'Enable', 'Inactive');

```

```

set(handles.freq10_var, 'Enable', 'Inactive');
set(handles.flanger_rate, 'Enable', 'Inactive');
set(handles.flanger_depth, 'Enable', 'Inactive');
set(handles.flanger_mix, 'Enable', 'Inactive');
set(handles.reverb_time, 'Enable', 'Off');
set(handles.reverb_feedback, 'Enable', 'Inactive');
set(handles.delay_level, 'Enable', 'Inactive');
set(handles.delay_time, 'Enable', 'Inactive');
set(handles.delay_feedback, 'Enable', 'Inactive');
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes digitalFX wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = digitalFX_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in amp_power.
function amp_power_Callback(hObject, eventdata, handles)
% hObject handle to amp_power (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of amp_power
button_state = get(hObject, 'Value');
if button_state == get(hObject, 'Max')
    % toggle button is pressed
    set(handles.FilePath, 'Enable', 'On');
    set(handles.FilePath, 'String', 'DIGITAL STEREO AMPLIFIER');
    set(handles.text_Hertz, 'String', '');
    set(handles.text_Hertz, 'Enable', 'On');
    set(handles.sampling_rate, 'String', '');
    set(handles.sampling_rate, 'Enable', 'On');
    set(handles.sample_bits, 'String', '');
    set(handles.sample_bits, 'Enable', 'On');
    set(handles.bits, 'String', '');
    set(handles.bits, 'Enable', 'On');
    set(handles.channels, 'String', '');
    set(handles.channels, 'Enable', 'On');
    set(handles.pushbutton_play, 'Enable', 'On');
    set(handles.togglebutton_pause, 'Enable', 'On');
    set(handles.pushbutton_stop, 'Enable', 'On');
    set(handles.pushbutton_open, 'Enable', 'On');
    set(handles.volume_up, 'Enable', 'On');
    set(handles.volume_down, 'Enable', 'On');
    set(handles.volume, 'Enable', 'On');
    set(handles.volume, 'String', '0');
    set_param('dfx/Slider Gain', 'Gain', num2str(0));
    set(handles.balance, 'Enable', 'On');
    set(handles.balance, 'Value', 0);

```



```

set_param('dfx/Balance/Constant_right','Value',num2str(1));
set_param('dfx/Balance/Constant_left','Value',num2str(1));
set(handles.togglebutton_surround,'Enable','On');
set(handles.menu_open,'Enable','On');
set(handles.menu_commands,'Enable','On');
elseif button_state == get(hObject,'Min')
    % toggle button is not pressed
set(handles.FilePath,'Enable','Off');
set(handles.text_Hertz,'Enable','Off');
set(handles.sampling_rate,'Enable','Off');
set(handles.sample_bits,'Enable','Off');
set(handles.bits,'Enable','Off');
set(handles.channels,'Enable','Off');
set(handles.pushbutton_play,'Enable','Inactive');
set(handles.togglebutton_pause,'Enable','Inactive');
set(handles.pushbutton_stop,'Enable','Inactive');
set(handles.pushbutton_open,'Enable','Inactive');
set(handles.volume_up,'Enable','Inactive');
set(handles.volume_down,'Enable','Inactive');
set(handles.volume,'Enable','Off');
set(handles.balance,'Enable','Inactive');
set(handles.text_left,'String','');
set(handles.text_right,'String','');
set(handles.togglebutton_surround,'Enable','Inactive');
set(handles.menu_open,'Enable','Off');
set(handles.menu_commands,'Enable','Off');
end

% --- Executes on button press in pushbutton_play.
function pushbutton_play_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_play (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global simTime;
sim('dfx',[0 simTime]);
%set_param(gcs,'SimulationCommand','Start')

% --- Executes on button press in togglebutton_pause.
function togglebutton_pause_Callback(hObject, eventdata, handles)
% hObject    handle to togglebutton_pause (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of
togglebutton_pause
button_state = get(hObject,'Value');
if button_state == get(hObject,'Max')
    % toggle button is pressed
    set_param(gcs,'SimulationCommand','Pause')
else
    set_param(gcs,'SimulationCommand','Continue')
end

% --- Executes on button press in pushbutton_stop.
function pushbutton_stop_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_stop (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

set_param(gcs, 'SimulationCommand', 'Stop')

% --- Executes on button press in pushbutton_open.
function pushbutton_open_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton_open (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

[FileName,PathName] = uigetfile('*.wav','Select an audio file');
path = [PathName,FileName];
set_param('dfx/From Wave File','FileName',FileName);
set(handles.FilePath,'String',path);
[x,Fs,bits] = wavread (FileName);
%info = mmfileinfo(FileName);
%simTime = info.Duration;
%sim('dfx',[0 simTime]);
channel = size (x,2);
if channel == 1;
    channels = ('Mono');
else
    channels = ('Stereo');
end
%set(handles.freq1_val,'string',info.Duration);
set(handles.channels,'String',channels);
set(handles.sampling_rate,'String',Fs);
set(handles.sample_bits,'String',bits);
set(handles.text_Hertz,'String','Hz');
set(handles.bits,'String','bits');
global simTime;
simTime = (length (x)/Fs) + 2;

% --- Executes on slider movement.
function freq1_var_Callback(hObject, eventdata, handles)
% hObject handle to freq1_var (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
% get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
var=get(hObject,'value');
set(handles.freq1_val,'string',num2str(var));
set_param('dfx/Equalizer/dB Gain1','dB',num2str(var));

% --- Executes during object creation, after setting all properties.
function freq1_var_CreateFcn(hObject, eventdata, handles)
% hObject handle to freq1_var (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: slider controls usually have a light gray background.

```

```

if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function freq2_var_Callback(hObject, eventdata, handles)
% hObject    handle to freq2_var (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
var=get(hObject,'value');
set(handles.freq2_val,'string',num2str(var));
set_param('dfx/Equalizer/dB Gain2','dB',num2str(var));

% --- Executes during object creation, after setting all properties.
function freq2_var_CreateFcn(hObject, eventdata, handles)
% hObject    handle to freq2_var (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function freq3_var_Callback(hObject, eventdata, handles)
% hObject    handle to freq3_var (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
var=get(hObject,'value');
set(handles.freq3_val,'string',num2str(var));
set_param('dfx/Equalizer/dB Gain3','dB',num2str(var));

% --- Executes during object creation, after setting all properties.
function freq3_var_CreateFcn(hObject, eventdata, handles)
% hObject    handle to freq3_var (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

```

```

end

% --- Executes on slider movement.
function freq4_var_Callback(hObject, eventdata, handles)
% hObject    handle to freq4_var (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
var=get(hObject,'value');
set(handles.freq4_val,'string',num2str(var));
set_param('dfx/Equalizer/dB Gain4','dB',num2str(var));

% --- Executes during object creation, after setting all properties.
function freq4_var_CreateFcn(hObject, eventdata, handles)
% hObject    handle to freq4_var (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function freq5_var_Callback(hObject, eventdata, handles)
% hObject    handle to freq5_var (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
var=get(hObject,'value');
set(handles.freq5_val,'string',num2str(var));
set_param('dfx/Equalizer/dB Gain5','dB',num2str(var));

% --- Executes during object creation, after setting all properties.
function freq5_var_CreateFcn(hObject, eventdata, handles)
% hObject    handle to freq5_var (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function freq6_var_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to freq6_var (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
var=get(hObject,'value');
set(handles.freq6_val,'string',num2str(var));
set_param('dfx/Equalizer/dB Gain6','dB',num2str(var));

% --- Executes during object creation, after setting all properties.
function freq6_var_CreateFcn(hObject, eventdata, handles)
% hObject    handle to freq6_var (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function freq7_var_Callback(hObject, eventdata, handles)
% hObject    handle to freq7_var (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
var=get(hObject,'value');
set(handles.freq7_val,'string',num2str(var));
set_param('dfx/Equalizer/dB Gain7','dB',num2str(var));

% --- Executes during object creation, after setting all properties.
function freq7_var_CreateFcn(hObject, eventdata, handles)
% hObject    handle to freq7_var (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function freq8_var_Callback(hObject, eventdata, handles)
% hObject    handle to freq8_var (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider

```



```

%         get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
var=get(hObject,'value');
set(handles.freq8_val,'string',num2str(var));
set_param('dfx/Equalizer/dB Gain8','dB',num2str(var));

% --- Executes during object creation, after setting all properties.
function freq8_var_CreateFcn(hObject, eventdata, handles)
% hObject    handle to freq8_var (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function freq9_var_Callback(hObject, eventdata, handles)
% hObject    handle to freq9_var (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
var=get(hObject,'value');
set(handles.freq9_val,'string',num2str(var));
set_param('dfx/Equalizer/dB Gain9','dB',num2str(var));

% --- Executes during object creation, after setting all properties.
function freq9_var_CreateFcn(hObject, eventdata, handles)
% hObject    handle to freq9_var (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function freq10_var_Callback(hObject, eventdata, handles)
% hObject    handle to freq10_var (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
var=get(hObject,'value');
set(handles.freq10_val,'string',num2str(var));
set_param('dfx/Equalizer/dB Gain10','dB',num2str(var));

```

```

% --- Executes during object creation, after setting all properties.
function freq10_var_CreateFcn(hObject, eventdata, handles)
% hObject    handle to freq10_var (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on selection change in presets_menu.
function presets_menu_Callback(hObject, eventdata, handles)
% hObject    handle to presets_menu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
presets_menu contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
presets_menu
switch get(handles.presets_menu,'Value')
    case 1
        set(handles.freq1_var,'Value',0);
        set(handles.freq1_val,'string',0);
        set_param('dfx/Equalizer/dB Gain1','dB',num2str(0));
        set(handles.freq2_var,'Value',0);
        set(handles.freq2_val,'string',0);
        set_param('dfx/Equalizer/dB Gain2','dB',num2str(0));
        set(handles.freq3_var,'Value',0);
        set(handles.freq3_val,'string',0);
        set_param('dfx/Equalizer/dB Gain3','dB',num2str(0));
        set(handles.freq4_var,'Value',0);
        set(handles.freq4_val,'string',0);
        set_param('dfx/Equalizer/dB Gain4','dB',num2str(0));
        set(handles.freq5_var,'Value',0);
        set(handles.freq5_val,'string',0);
        set_param('dfx/Equalizer/dB Gain5','dB',num2str(0));
        set(handles.freq6_var,'Value',0);
        set(handles.freq6_val,'string',0);
        set_param('dfx/Equalizer/dB Gain6','dB',num2str(0));
        set(handles.freq7_var,'Value',0);
        set(handles.freq7_val,'string',0);
        set_param('dfx/Equalizer/dB Gain7','dB',num2str(0));
        set(handles.freq8_var,'Value',0);
        set(handles.freq8_val,'string',0);
        set_param('dfx/Equalizer/dB Gain8','dB',num2str(0));
        set(handles.freq9_var,'Value',0);
        set(handles.freq9_val,'string',0);
        set_param('dfx/Equalizer/dB Gain9','dB',num2str(0));
        set(handles.freq10_var,'Value',0);
        set(handles.freq10_val,'string',0);
        set_param('dfx/Equalizer/dB Gain10','dB',num2str(0));
    case 2
        set(handles.freq1_var,'Value',0);
        set(handles.freq1_val,'string',0);

```

```

set_param('dfx/Equalizer/dB Gain1', 'dB', num2str(0));
set(handles.freq2_var, 'Value', 1);
set(handles.freq2_val, 'string', 1);
set_param('dfx/Equalizer/dB Gain2', 'dB', num2str(1));
set(handles.freq3_var, 'Value', 2);
set(handles.freq3_val, 'string', 2);
set_param('dfx/Equalizer/dB Gain3', 'dB', num2str(2));
set(handles.freq4_var, 'Value', 0);
set(handles.freq4_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain4', 'dB', num2str(0));
set(handles.freq5_var, 'Value', 0);
set(handles.freq5_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain5', 'dB', num2str(0));
set(handles.freq6_var, 'Value', 0);
set(handles.freq6_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain6', 'dB', num2str(0));
set(handles.freq7_var, 'Value', 0);
set(handles.freq7_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain7', 'dB', num2str(0));
set(handles.freq8_var, 'Value', 0);
set(handles.freq8_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain8', 'dB', num2str(0));
set(handles.freq9_var, 'Value', 2);
set(handles.freq9_val, 'string', 2);
set_param('dfx/Equalizer/dB Gain9', 'dB', num2str(2));
set(handles.freq10_var, 'Value', 2);
set(handles.freq10_val, 'string', 2);
set_param('dfx/Equalizer/dB Gain10', 'dB', num2str(2));
case 3
set(handles.freq1_var, 'Value', -1);
set(handles.freq1_val, 'string', -1);
set_param('dfx/Equalizer/dB Gain1', 'dB', num2str(-1));
set(handles.freq2_var, 'Value', 0);
set(handles.freq2_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain2', 'dB', num2str(0));
set(handles.freq3_var, 'Value', 2);
set(handles.freq3_val, 'string', 2);
set_param('dfx/Equalizer/dB Gain3', 'dB', num2str(2));
set(handles.freq4_var, 'Value', 1);
set(handles.freq4_val, 'string', 1);
set_param('dfx/Equalizer/dB Gain4', 'dB', num2str(1));
set(handles.freq5_var, 'Value', 0);
set(handles.freq5_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain5', 'dB', num2str(0));
set(handles.freq6_var, 'Value', 0);
set(handles.freq6_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain6', 'dB', num2str(0));
set(handles.freq7_var, 'Value', 0);
set(handles.freq7_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain7', 'dB', num2str(0));
set(handles.freq8_var, 'Value', 0);
set(handles.freq8_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain8', 'dB', num2str(0));
set(handles.freq9_var, 'Value', -1);
set(handles.freq9_val, 'string', -1);
set_param('dfx/Equalizer/dB Gain9', 'dB', num2str(-1));
set(handles.freq10_var, 'Value', -3);
set(handles.freq10_val, 'string', -3);
set_param('dfx/Equalizer/dB Gain10', 'dB', num2str(-3));
case 4
set(handles.freq1_var, 'Value', 0);

```



```

set(handles.freq1_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain1', 'dB', num2str(0));
set(handles.freq2_var, 'Value', 6);
set(handles.freq2_val, 'string', 6);
set_param('dfx/Equalizer/dB Gain2', 'dB', num2str(6));
set(handles.freq3_var, 'Value', 6);
set(handles.freq3_val, 'string', 6);
set_param('dfx/Equalizer/dB Gain3', 'dB', num2str(6));
set(handles.freq4_var, 'Value', 3);
set(handles.freq4_val, 'string', 3);
set_param('dfx/Equalizer/dB Gain4', 'dB', num2str(3));
set(handles.freq5_var, 'Value', 0);
set(handles.freq5_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain5', 'dB', num2str(0));
set(handles.freq6_var, 'Value', 0);
set(handles.freq6_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain6', 'dB', num2str(0));
set(handles.freq7_var, 'Value', 0);
set(handles.freq7_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain7', 'dB', num2str(0));
set(handles.freq8_var, 'Value', 0);
set(handles.freq8_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain8', 'dB', num2str(0));
set(handles.freq9_var, 'Value', 2);
set(handles.freq9_val, 'string', 2);
set_param('dfx/Equalizer/dB Gain9', 'dB', num2str(2));
set(handles.freq10_var, 'Value', 2);
set(handles.freq10_val, 'string', 2);
set_param('dfx/Equalizer/dB Gain10', 'dB', num2str(2));
case 5
set(handles.freq1_var, 'Value', -1);
set(handles.freq1_val, 'string', -1);
set_param('dfx/Equalizer/dB Gain1', 'dB', num2str(-1));
set(handles.freq2_var, 'Value', 4);
set(handles.freq2_val, 'string', 4);
set_param('dfx/Equalizer/dB Gain2', 'dB', num2str(4));
set(handles.freq3_var, 'Value', 5);
set(handles.freq3_val, 'string', 5);
set_param('dfx/Equalizer/dB Gain3', 'dB', num2str(5));
set(handles.freq4_var, 'Value', 1);
set(handles.freq4_val, 'string', 1);
set_param('dfx/Equalizer/dB Gain4', 'dB', num2str(1));
set(handles.freq5_var, 'Value', -1);
set(handles.freq5_val, 'string', -1);
set_param('dfx/Equalizer/dB Gain5', 'dB', num2str(-1));
set(handles.freq6_var, 'Value', -1);
set(handles.freq6_val, 'string', -1);
set_param('dfx/Equalizer/dB Gain6', 'dB', num2str(-1));
set(handles.freq7_var, 'Value', 0);
set(handles.freq7_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain7', 'dB', num2str(0));
set(handles.freq8_var, 'Value', 0);
set(handles.freq8_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain8', 'dB', num2str(0));
set(handles.freq9_var, 'Value', 4);
set(handles.freq9_val, 'string', 4);
set_param('dfx/Equalizer/dB Gain9', 'dB', num2str(4));
set(handles.freq10_var, 'Value', 4);
set(handles.freq10_val, 'string', 4);
set_param('dfx/Equalizer/dB Gain10', 'dB', num2str(4));
case 6

```

```

set(handles.freq1_var, 'Value', 0);
set(handles.freq1_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain1', 'dB', num2str(0));
set(handles.freq2_var, 'Value', 0);
set(handles.freq2_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain2', 'dB', num2str(0));
set(handles.freq3_var, 'Value', 0);
set(handles.freq3_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain3', 'dB', num2str(0));
set(handles.freq4_var, 'Value', 3);
set(handles.freq4_val, 'string', 3);
set_param('dfx/Equalizer/dB Gain4', 'dB', num2str(3));
set(handles.freq5_var, 'Value', 3);
set(handles.freq5_val, 'string', 3);
set_param('dfx/Equalizer/dB Gain5', 'dB', num2str(3));
set(handles.freq6_var, 'Value', 3);
set(handles.freq6_val, 'string', 3);
set_param('dfx/Equalizer/dB Gain6', 'dB', num2str(3));
set(handles.freq7_var, 'Value', 0);
set(handles.freq7_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain7', 'dB', num2str(0));
set(handles.freq8_var, 'Value', 2);
set(handles.freq8_val, 'string', 2);
set_param('dfx/Equalizer/dB Gain8', 'dB', num2str(2));
set(handles.freq9_var, 'Value', 4);
set(handles.freq9_val, 'string', 4);
set_param('dfx/Equalizer/dB Gain9', 'dB', num2str(4));
set(handles.freq10_var, 'Value', 4);
set(handles.freq10_val, 'string', 4);
set_param('dfx/Equalizer/dB Gain10', 'dB', num2str(4));
case 7
set(handles.freq1_var, 'Value', -2);
set(handles.freq1_val, 'string', -2);
set_param('dfx/Equalizer/dB Gain1', 'dB', num2str(-2));
set(handles.freq2_var, 'Value', 0);
set(handles.freq2_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain2', 'dB', num2str(0));
set(handles.freq3_var, 'Value', 2);
set(handles.freq3_val, 'string', 2);
set_param('dfx/Equalizer/dB Gain3', 'dB', num2str(2));
set(handles.freq4_var, 'Value', 1);
set(handles.freq4_val, 'string', 1);
set_param('dfx/Equalizer/dB Gain4', 'dB', num2str(1));
set(handles.freq5_var, 'Value', 0);
set(handles.freq5_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain5', 'dB', num2str(0));
set(handles.freq6_var, 'Value', 0);
set(handles.freq6_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain6', 'dB', num2str(0));
set(handles.freq7_var, 'Value', 0);
set(handles.freq7_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain7', 'dB', num2str(0));
set(handles.freq8_var, 'Value', 0);
set(handles.freq8_val, 'string', 0);
set_param('dfx/Equalizer/dB Gain8', 'dB', num2str(0));
set(handles.freq9_var, 'Value', -2);
set(handles.freq9_val, 'string', -2);
set_param('dfx/Equalizer/dB Gain9', 'dB', num2str(-2));
set(handles.freq10_var, 'Value', -5);
set(handles.freq10_val, 'string', -5);
set_param('dfx/Equalizer/dB Gain10', 'dB', num2str(-5));

```

```

case 8
    set(handles.freq1_var, 'Value', -1);
    set(handles.freq1_val, 'string', -1);
    set_param('dfx/Equalizer/dB Gain1', 'dB', num2str(-1));
    set(handles.freq2_var, 'Value', 0);
    set(handles.freq2_val, 'string', 0);
    set_param('dfx/Equalizer/dB Gain2', 'dB', num2str(0));
    set(handles.freq3_var, 'Value', 2);
    set(handles.freq3_val, 'string', 2);
    set_param('dfx/Equalizer/dB Gain3', 'dB', num2str(2));
    set(handles.freq4_var, 'Value', 2);
    set(handles.freq4_val, 'string', 2);
    set_param('dfx/Equalizer/dB Gain4', 'dB', num2str(2));
    set(handles.freq5_var, 'Value', -1);
    set(handles.freq5_val, 'string', -1);
    set_param('dfx/Equalizer/dB Gain5', 'dB', num2str(-1));
    set(handles.freq6_var, 'Value', -1);
    set(handles.freq6_val, 'string', -1);
    set_param('dfx/Equalizer/dB Gain6', 'dB', num2str(-1));
    set(handles.freq7_var, 'Value', 0);
    set(handles.freq7_val, 'string', 0);
    set_param('dfx/Equalizer/dB Gain7', 'dB', num2str(0));
    set(handles.freq8_var, 'Value', 0);
    set(handles.freq8_val, 'string', 0);
    set_param('dfx/Equalizer/dB Gain8', 'dB', num2str(0));
    set(handles.freq9_var, 'Value', 4);
    set(handles.freq9_val, 'string', 4);
    set_param('dfx/Equalizer/dB Gain9', 'dB', num2str(4));
    set(handles.freq10_var, 'Value', 6);
    set(handles.freq10_val, 'string', 6);
    set_param('dfx/Equalizer/dB Gain10', 'dB', num2str(6));
case 9
    set(handles.freq1_var, 'Value', -1);
    set(handles.freq1_val, 'string', -1);
    set_param('dfx/Equalizer/dB Gain1', 'dB', num2str(-1));
    set(handles.freq2_var, 'Value', 0);
    set(handles.freq2_val, 'string', 0);
    set_param('dfx/Equalizer/dB Gain2', 'dB', num2str(0));
    set(handles.freq3_var, 'Value', 0);
    set(handles.freq3_val, 'string', 0);
    set_param('dfx/Equalizer/dB Gain3', 'dB', num2str(0));
    set(handles.freq4_var, 'Value', -3);
    set(handles.freq4_val, 'string', -3);
    set_param('dfx/Equalizer/dB Gain4', 'dB', num2str(-3));
    set(handles.freq5_var, 'Value', 0);
    set(handles.freq5_val, 'string', 0);
    set_param('dfx/Equalizer/dB Gain5', 'dB', num2str(0));
    set(handles.freq6_var, 'Value', 3);
    set(handles.freq6_val, 'string', 3);
    set_param('dfx/Equalizer/dB Gain6', 'dB', num2str(3));
    set(handles.freq7_var, 'Value', 4);
    set(handles.freq7_val, 'string', 4);
    set_param('dfx/Equalizer/dB Gain7', 'dB', num2str(4));
    set(handles.freq8_var, 'Value', 0);
    set(handles.freq8_val, 'string', 0);
    set_param('dfx/Equalizer/dB Gain8', 'dB', num2str(0));
    set(handles.freq9_var, 'Value', 3);
    set(handles.freq9_val, 'string', 3);
    set_param('dfx/Equalizer/dB Gain9', 'dB', num2str(3));
    set(handles.freq10_var, 'Value', 4);
    set(handles.freq10_val, 'string', 4);

```

```

set_param('dfx/Equalizer/dB Gain10','dB',num2str(4));
case 10
set(handles.freq1_var,'Value',-1);
set(handles.freq1_val,'string',-1);
set_param('dfx/Equalizer/dB Gain1','dB',num2str(-1));
set(handles.freq2_var,'Value',1);
set(handles.freq2_val,'string',1);
set_param('dfx/Equalizer/dB Gain2','dB',num2str(1));
set(handles.freq3_var,'Value',2);
set(handles.freq3_val,'string',2);
set_param('dfx/Equalizer/dB Gain3','dB',num2str(2));
set(handles.freq4_var,'Value',3);
set(handles.freq4_val,'string',3);
set_param('dfx/Equalizer/dB Gain4','dB',num2str(3));
set(handles.freq5_var,'Value',-1);
set(handles.freq5_val,'string',-1);
set_param('dfx/Equalizer/dB Gain5','dB',num2str(-1));
set(handles.freq6_var,'Value',-1);
set(handles.freq6_val,'string',-1);
set_param('dfx/Equalizer/dB Gain6','dB',num2str(-1));
set(handles.freq7_var,'Value',0);
set(handles.freq7_val,'string',0);
set_param('dfx/Equalizer/dB Gain7','dB',num2str(0));
set(handles.freq8_var,'Value',0);
set(handles.freq8_val,'string',0);
set_param('dfx/Equalizer/dB Gain8','dB',num2str(0));
set(handles.freq9_var,'Value',4);
set(handles.freq9_val,'string',4);
set_param('dfx/Equalizer/dB Gain9','dB',num2str(4));
set(handles.freq10_var,'Value',4);
set(handles.freq10_val,'string',4);
set_param('dfx/Equalizer/dB Gain10','dB',num2str(4));
case 11
set(handles.freq1_var,'Value',-6);
set(handles.freq1_val,'string',-6);
set_param('dfx/Equalizer/dB Gain1','dB',num2str(-6));
set(handles.freq2_var,'Value',1);
set(handles.freq2_val,'string',1);
set_param('dfx/Equalizer/dB Gain2','dB',num2str(1));
set(handles.freq3_var,'Value',4);
set(handles.freq3_val,'string',4);
set_param('dfx/Equalizer/dB Gain3','dB',num2str(4));
set(handles.freq4_var,'Value',-1);
set(handles.freq4_val,'string',-1);
set_param('dfx/Equalizer/dB Gain4','dB',num2str(-1));
set(handles.freq5_var,'Value',-1);
set(handles.freq5_val,'string',-1);
set_param('dfx/Equalizer/dB Gain5','dB',num2str(-1));
set(handles.freq6_var,'Value',-2);
set(handles.freq6_val,'string',-2);
set_param('dfx/Equalizer/dB Gain6','dB',num2str(-2));
set(handles.freq7_var,'Value',0);
set(handles.freq7_val,'string',0);
set_param('dfx/Equalizer/dB Gain7','dB',num2str(0));
set(handles.freq8_var,'Value',0);
set(handles.freq8_val,'string',0);
set_param('dfx/Equalizer/dB Gain8','dB',num2str(0));
set(handles.freq9_var,'Value',5);
set(handles.freq9_val,'string',5);
set_param('dfx/Equalizer/dB Gain9','dB',num2str(5));
set(handles.freq10_var,'Value',5);

```



```

        set(handles.freq10_val,'string',5);
        set_param('dfx/Equalizer/dB Gain10','dB',num2str(5));
    otherwise
end

% --- Executes during object creation, after setting all properties.
function presets_menu_CreateFcn(hObject, eventdata, handles)
% hObject    handle to presets_menu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in togglebutton_surround.
function togglebutton_surround_Callback(hObject, eventdata, handles)
% hObject    handle to togglebutton_surround (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of
togglebutton_surround
button_state = get(hObject,'Value');
if button_state == get(hObject,'Max')
    % toggle button is pressed, surround effect is enabled
    set_param('dfx/Constant','Value',num2str(2));
elseif button_state == get(hObject,'Min')
    % toggle button is not pressed, surround effect is disabled
    set_param('dfx/Constant','Value',num2str(1));
end

% --- Executes on slider movement.
function delay_level_Callback(hObject, eventdata, handles)
% hObject    handle to delay_level (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
level=get(hObject,'value');
set_param('dfx/Delay/Gain','Gain',num2str(level));

% --- Executes during object creation, after setting all properties.
function delay_level_CreateFcn(hObject, eventdata, handles)
% hObject    handle to delay_level (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

```

```

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on button press in delay_on.
function delay_on_Callback(hObject, eventdata, handles)
% hObject    handle to delay_on (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of delay_on
status = get(handles.delay_on,'Value');
%status = 0, if the box is unchecked,
%status = 1, if the box is checked
if(status)
    %if box is checked, delay pedal turns on
    set_param('dfx/Constant2','Value',num2str(2));
    set(handles.delay_level,'Enable','On');
    set(handles.delay_time,'Enable','On');
    set(handles.delay_feedback,'Enable','On');
else
    %if box is unchecked, delay pedal turns off
    set_param('dfx/Constant2','Value',num2str(1));
    set(handles.delay_level,'Enable','Inactive');
    set(handles.delay_time,'Enable','Inactive');
    set(handles.delay_feedback,'Enable','Inactive');
end

% --- Executes on button press in reverb_on.
function reverb_on_Callback(hObject, eventdata, handles)
% hObject    handle to reverb_on (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of reverb_on
status = get(handles.reverb_on,'Value');
%status = 0, if the box is unchecked,
%status = 1, if the box is checked
if(status)
    %if box is checked, reverb pedal turns on
    set_param('dfx/Constant3','Value',num2str(2));
    set(handles.reverb_time,'Enable','On');
    set(handles.reverb_feedback,'Enable','On');
else
    %if box is unchecked, reverb pedal turns off
    set_param('dfx/Constant3','Value',num2str(1));
    set(handles.reverb_time,'Enable','Inactive');
    set(handles.reverb_feedback,'Enable','Inactive');
end

% --- Executes on button press in flanger_on.
function flanger_on_Callback(hObject, eventdata, handles)
% hObject    handle to flanger_on (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hint: get(hObject,'Value') returns toggle state of flanger_on
status = get(handles.flanger_on,'Value');
%status = 0, if the box is unchecked,
%status = 1, if the box is checked
if(status)
    %if box is checked, flanger pedal turns on
    set_param('dfx/Constant4','Value',num2str(2));
    set(handles.flanger_rate,'Enable','On');
    set(handles.flanger_depth,'Enable','On');
    set(handles.flanger_mix,'Enable','On');
else
    %if box is unchecked, flanger pedal turns off
    set_param('dfx/Constant4','Value',num2str(1));
    set(handles.flanger_rate,'Enable','Inactive');
    set(handles.flanger_depth,'Enable','Inactive');
    set(handles.flanger_mix,'Enable','Inactive');
end

% --- Executes on button press in equalizer_power.
function equalizer_power_Callback(hObject, eventdata, handles)
% hObject    handle to equalizer_power (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of equalizer_power
button_state = get(hObject,'Value');
if button_state == get(hObject,'Max')
    % toggle button is pressed
    set_param('dfx/Constant1','Value',num2str(2));
    set(handles.presets_menu,'Enable','On');
    set(handles.freq1_var,'Enable','On');
    set(handles.freq2_var,'Enable','On');
    set(handles.freq3_var,'Enable','On');
    set(handles.freq4_var,'Enable','On');
    set(handles.freq5_var,'Enable','On');
    set(handles.freq6_var,'Enable','On');
    set(handles.freq7_var,'Enable','On');
    set(handles.freq8_var,'Enable','On');
    set(handles.freq9_var,'Enable','On');
    set(handles.freq10_var,'Enable','On');
elseif button_state == get(hObject,'Min')
    % toggle button is not pressed
    set_param('dfx/Constant1','Value',num2str(1));
    set(handles.presets_menu,'Enable','Inactive');
    set(handles.freq1_var,'Enable','Inactive');
    set(handles.freq2_var,'Enable','Inactive');
    set(handles.freq3_var,'Enable','Inactive');
    set(handles.freq4_var,'Enable','Inactive');
    set(handles.freq5_var,'Enable','Inactive');
    set(handles.freq6_var,'Enable','Inactive');
    set(handles.freq7_var,'Enable','Inactive');
    set(handles.freq8_var,'Enable','Inactive');
    set(handles.freq9_var,'Enable','Inactive');
    set(handles.freq10_var,'Enable','Inactive');
end

% --- Executes on button press in volume_up.
function volume_up_Callback(hObject, eventdata, handles)
% hObject    handle to volume_up (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

tmp = get(handles.volume, 'String');
if (str2num(tmp) < 20)
vol = str2num(tmp) + 1;
set(handles.volume, 'String', num2str(vol));
set_param('dfx/Slider Gain', 'Gain', num2str(vol));
end

% --- Executes on button press in volume_down.
function volume_down_Callback(hObject, eventdata, handles)
% hObject    handle to volume_down (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
tmp = get(handles.volume, 'String');
if (str2num(tmp) > 0)
vol = str2num(tmp) - 1;
set(handles.volume, 'String', num2str(vol));
set_param('dfx/Slider Gain', 'Gain', num2str(vol));
end

% --- Executes on slider movement.
function delay_time_Callback(hObject, eventdata, handles)
% hObject    handle to delay_time (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'Value') returns position of slider
%        get(hObject, 'Min') and get(hObject, 'Max') to determine range
of slider
time=get(hObject, 'value');
set_param('dfx/Delay/Integer Delay', 'Delay', num2str(round(time)));

% --- Executes during object creation, after setting all properties.
function delay_time_CreateFcn(hObject, eventdata, handles)
% hObject    handle to delay_time (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', [.9 .9 .9]);
end

% --- Executes on slider movement.
function reverb_time_Callback(hObject, eventdata, handles)
% hObject    handle to reverb_time (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'Value') returns position of slider
%        get(hObject, 'Min') and get(hObject, 'Max') to determine range
of slider
time=get(hObject, 'value');
set_param('dfx/Reverb/Comb1/Integer
Delay', 'Delay', num2str(round(time)));

```



```

set_param('dfx/Reverb/Comb2/Integer
Delay','Delay',num2str(round(2*time)));
set_param('dfx/Reverb/Comb3/Integer
Delay','Delay',num2str(round(3*time)));
set_param('dfx/Reverb/Comb4/Integer
Delay','Delay',num2str(round(4*time)));

% --- Executes during object creation, after setting all properties.
function reverb_time_CreateFcn(hObject, eventdata, handles)
% hObject    handle to reverb_time (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function reverb_feedback_Callback(hObject, eventdata, handles)
% hObject    handle to reverb_feedback (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
feedback=get(hObject,'value');
set_param('dfx/Reverb/Comb1/Feedback','Gain',num2str(feedback));
set_param('dfx/Reverb/Comb2/Feedback','Gain',num2str(feedback));
set_param('dfx/Reverb/Comb3/Feedback','Gain',num2str(feedback));
set_param('dfx/Reverb/Comb4/Feedback','Gain',num2str(feedback));

% --- Executes during object creation, after setting all properties.
function reverb_feedback_CreateFcn(hObject, eventdata, handles)
% hObject    handle to reverb_feedback (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function flanger_rate_Callback(hObject, eventdata, handles)
% hObject    handle to flanger_rate (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider

```

```

%         get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
rate=get(hObject,'value');
set_param('dfx/Flanger/Low Frequency
Oscillator','Frequency',num2str(rate));

% --- Executes during object creation, after setting all properties.
function flanger_rate_CreateFcn(hObject, eventdata, handles)
% hObject    handle to flanger_rate (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function flanger_depth_Callback(hObject, eventdata, handles)
% hObject    handle to flanger_depth (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
depth=get(hObject,'value');
set_param('dfx/Flanger/Low Frequency
Oscillator','Amplitude',num2str(depth));

% --- Executes during object creation, after setting all properties.
function flanger_depth_CreateFcn(hObject, eventdata, handles)
% hObject    handle to flanger_depth (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function balance_Callback(hObject, eventdata, handles)
% hObject    handle to balance (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
balance=get(hObject,'value');
if (balance<0)
set_param('dfx/Balance/Constant_right','Value',num2str(1+balance));

```

```

set_param('dfx/Balance/Constant_left','Value',num2str(1));
set(handles.text_left,'String',num2str(round(-10*balance)));
set(handles.text_right,'String','');
elseif (balance>0)
set_param('dfx/Balance/Constant_left','Value',num2str(1-balance));
set_param('dfx/Balance/Constant_right','Value',num2str(1));
set(handles.text_right,'String',num2str(round(10*balance)));
set(handles.text_left,'String','');
elseif (balance==0)
set_param('dfx/Balance/Constant_right','Value',num2str(1));
set_param('dfx/Balance/Constant_left','Value',num2str(1));
set(handles.text_left,'String','');
set(handles.text_right,'String','');
end

% --- Executes during object creation, after setting all properties.
function balance_CreateFcn(hObject, eventdata, handles)
% hObject    handle to balance (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% -----
-
function menu_file_Callback(hObject, eventdata, handles)
% hObject    handle to menu_file (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
-
function menu_open_Callback(hObject, eventdata, handles)
% hObject    handle to menu_open (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[FileName,PathName] = uigetfile('*.wav','Select an audio file');
path = [PathName,FileName];
set_param('dfx/From Wave File','FileName',FileName);
set(handles.FilePath,'String',path);
[x,Fs,bits] = wavread (FileName);
%info = mmfileinfo(FileName);
%simTime = info.Duration;
%sim('dfx',[0 simTime]);
channel = size (x,2);
if channel == 1;
    channels = ('Mono');
else
    channels = ('Stereo');
end
%set(handles.freq1_val,'string',info.Duration);
set(handles.channels,'String',channels);

```

```

set(handles.sampling_rate,'String',Fs);
set(handles.sample_bits,'String',bits);
set(handles.text_Hertz,'String','Hz');
set(handles.bits,'String','bits');
global simTime;
simTime = (length (x)/Fs) + 2;

% -----
-
function menu_exit_Callback(hObject, eventdata, handles)
% hObject    handle to menu_exit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% msgboxText{1} = 'You have tried to open a file while the amplifier
is off power.';
% msgboxText{2} = 'Turn on the amplifier first.';
% msgbox(msgboxText,'Out of power', 'help');
exit

% -----
-
function menu_commands_Callback(hObject, eventdata, handles)
% hObject    handle to menu_commands (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
-
function menu_play_Callback(hObject, eventdata, handles)
% hObject    handle to menu_play (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global simTime;
sim('dfx',[0 simTime]);
%set_param(gcs,'SimulationCommand','Start')

% -----
-
function menu_pause_Callback(hObject, eventdata, handles)
% hObject    handle to menu_pause (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set_param(gcs,'SimulationCommand','Pause')

% -----
-
function menu_stop_Callback(hObject, eventdata, handles)
% hObject    handle to menu_stop (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set_param(gcs,'SimulationCommand','Stop')

% -----
-

```

```

function menu_effects_Callback(hObject, eventdata, handles)
% hObject    handle to menu_effects (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
-
function menu_dbs_Callback(hObject, eventdata, handles)
% hObject    handle to menu_dbs (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
-
function menu_delay_Callback(hObject, eventdata, handles)
% hObject    handle to menu_delay (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
-
function menu_reverb_Callback(hObject, eventdata, handles)
% hObject    handle to menu_reverb (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
-
function menu_flanger_Callback(hObject, eventdata, handles)
% hObject    handle to menu_flanger (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on slider movement.
function flanger_mix_Callback(hObject, eventdata, handles)
% hObject    handle to flanger_mix (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
mix=get(hObject,'value');
if (mix<0)
set_param('dfx/Flanger/Gain1','Gain',num2str(1+mix));
set_param('dfx/Flanger/Gain2','Gain',num2str(1));
elseif (mix>0)
set_param('dfx/Flanger/Gain2','Gain',num2str(1-mix));
set_param('dfx/Flanger/Gain1','Gain',num2str(1));
elseif (mix==0)
set_param('dfx/Flanger/Gain1','Gain',num2str(1));
set_param('dfx/Flanger/Gain2','Gain',num2str(1));
end

```



```

% --- Executes during object creation, after setting all properties.
function flanger_mix_CreateFcn(hObject, eventdata, handles)
% hObject    handle to flanger_mix (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% -----
-
function Untitled_14_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
-
function Untitled_15_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on slider movement.
function delay_feedback_Callback(hObject, eventdata, handles)
% hObject    handle to delay_feedback (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
feedback=get(hObject,'value');
set_param('dfx/Delay/Feedback','Gain',num2str(feedback));

% --- Executes during object creation, after setting all properties.
function delay_feedback_CreateFcn(hObject, eventdata, handles)
% hObject    handle to delay_feedback (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% -----
-

```

```

function menu_volume_Callback(hObject, eventdata, handles)
% hObject    handle to menu_volume (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
%
function menu_volincrease_Callback(hObject, eventdata, handles)
% hObject    handle to menu_volincrease (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
tmp = get(handles.volume, 'String');
if (str2num(tmp) < 20)
vol = str2num(tmp) + 1;
set(handles.volume, 'String', num2str(vol));
set_param('dfx/Slider Gain', 'Gain', num2str(vol));
end

% -----
%
function menu_voldecrease_Callback(hObject, eventdata, handles)
% hObject    handle to menu_voldecrease (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
tmp = get(handles.volume, 'String');
if (str2num(tmp) > 0)
vol = str2num(tmp) - 1;
set(handles.volume, 'String', num2str(vol));
set_param('dfx/Slider Gain', 'Gain', num2str(vol));
end

% -----
%
function menu_flangeron_Callback(hObject, eventdata, handles)
% hObject    handle to menu_flangeron (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set_param('dfx/Constant4', 'Value', num2str(2));
set(handles.flanger_rate, 'Enable', 'On');
set(handles.flanger_depth, 'Enable', 'On');
set(handles.flanger_mix, 'Enable', 'On');

% -----
%
function menu_flangeroff_Callback(hObject, eventdata, handles)
% hObject    handle to menu_flangeroff (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set_param('dfx/Constant4', 'Value', num2str(1));
set(handles.flanger_rate, 'Enable', 'Inactive');
set(handles.flanger_depth, 'Enable', 'Inactive');
set(handles.flanger_mix, 'Enable', 'Inactive');

```

```

% -----
-
function menu_reverbon_Callback(hObject, eventdata, handles)
% hObject    handle to menu_reverbon (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set_param('dfx/Constant3','Value',num2str(2));
set(handles.reverb_time,'Enable','On');
set(handles.reverb_feedback,'Enable','On');

% -----
-
function menu_reverboff_Callback(hObject, eventdata, handles)
% hObject    handle to menu_reverboff (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set_param('dfx/Constant3','Value',num2str(1));
set(handles.reverb_time,'Enable','Inactive');
set(handles.reverb_feedback,'Enable','Inactive');

% -----
-
function menu_delayon_Callback(hObject, eventdata, handles)
% hObject    handle to menu_delayon (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set_param('dfx/Constant2','Value',num2str(2));
set(handles.delay_level,'Enable','On');
set(handles.delay_time,'Enable','On');
set(handles.delay_feedback,'Enable','On');

% -----
-
function menu_delayoff_Callback(hObject, eventdata, handles)
% hObject    handle to menu_delayoff (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set_param('dfx/Constant2','Value',num2str(1));
set(handles.delay_level,'Enable','Inactive');
set(handles.delay_time,'Enable','Inactive');
set(handles.delay_feedback,'Enable','Inactive');

% -----
-
function menu_dbson_Callback(hObject, eventdata, handles)
% hObject    handle to menu_dbson (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set_param('dfx/Constant','Value',num2str(2));

% -----
-
function menu_dbsoff_Callback(hObject, eventdata, handles)
% hObject    handle to menu_dbsoff (see GCBO)

```



```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
set_param('dfx/Constant', 'Value', num2str(1));

% -----
-
function menu_help_Callback(hObject, eventdata, handles)
% hObject handle to menu_help (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% -----
-
function menu_howto_Callback(hObject, eventdata, handles)
% hObject handle to menu_howto (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
open help.pdf

% -----
-
function menu_about_Callback(hObject, eventdata, handles)
% hObject handle to menu_about (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
%this is the first line of the msgbox
msgboxText{1} = 'Digital Audio System v 1.0.1';
%this is the second line
msgboxText{2} = 'Created by mIndC!RcUs';
%this command creates the actual message box
msgbox(msgboxText, 'About', 'help');

```

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ