

Τμήμα
Μηχανικών
Πληροφορικής τ.ε.

Τεχνολογικό Εκπαιδευτικό Ίδρυμα
Δυτικής Ελλάδας

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

« Εφαρμογές αλγορίθμων μηχανικής μάθησης στη χωρική ανάλυση
της κρουστικής απόκρισης χώρου ακρόασης »

Κριδερά Σταυρούλα - ΑΜ 2468

Πουλή Αλεξάνδρα - ΑΜ 2536

ΕΠΙΒΛΕΠΩΝ: Μιχάλης Παρασκευάς, Επίκουρος Καθηγητής

ΑΝΤΙΠΡΟΪΟ 2019

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Αντίρριο, 2019.

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Μιχάλης Παρασκευάς
2. Ονοματεπώνυμο
3. Ονοματεπώνυμο

Ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε τον Επίκουρο Καθηγητή κ. Μιχαήλ Παρασκευά για την υποστήριξη, την κατανόηση και τη βοήθεια που μας έδωσε έτσι ώστε να πραγματοποιήσουμε την πτυχιακή μας εργασία. Επίσης τις ευχαριστίες μας θα θέλαμε να τις εκφράσουμε προς όλους τους καθηγητές του τμήματος Μηχανικών Πληροφορικής του Τεχνολογικού Εκπαιδευτικού Ιδρύματος Δυτικής Ελλάδας για τις πολύτιμες γνώσεις που μας προσέφεραν όλα αυτά τα χρόνια.

Τέλος, θέλουμε να εκφράσουμε ένα τεράστιο ευχαριστώ στις οικογένειες μας, για την στήριξη και την εμπιστοσύνη που μας έδειξαν όλα αυτά τα χρόνια των σπουδών μας.

Περιεχόμενα

Ευχαριστίες.....	4
Περίληψη.....	9
Abstract.....	10
Κεφάλαιο 1	11
1.1 Εισαγωγή	11
1.2 Ιστορική Αναδρομή	11
1.3 Ήχος.....	12
1.3.1 Διάδοση ήχου σε ελεύθερο πεδίο (free field).....	12
1.3.2 Διάδοση ήχου σε μη ελεύθερα πεδία	12
1.4 Ανάκλαση του ήχου (Reflection).....	15
1.5 Απορρόφηση του ήχου	16
1.6 Διάχυση του ήχου	16
1.7 Διέλευση του ήχου	17
Κεφάλαιο 2	18
2.1 Εισαγωγή	18
2.2 Βασικές Ακουστικές Παράμετροι.....	18
2.2.1 Χρόνος αντήχησης RT60 (Reverberation Time)	18
2.2.2 Ευκρίνεια (Clarity).....	22
2.2.3 Ηχηρότητα (Sound Stength,G10).....	23
2.2.4 Αντιληπτότητα των συλλαβών (Definition).....	23
2.3 Πρώιμος Χρόνος Αντήχησης (Early Delay Time – EDT).....	24
2.4 Άλλες Ακουστικές Παράμετροι.....	24
2.4.1 Ηχώ.....	24
2.4.2 Χρωματισμός (Coloration).....	25
2.4.3 Ζεστασιά (Warmth).....	25
2.4.4 Οικειότητα (Intimacy).....	25

2.5 Μοντέλο των ειδώλων πηγής (Image Source Model)	25
Κεφάλαιο 3	27
3.1 Κρουστική Απόκριση (Impulse Response).....	27
3.1.1 Κρουστική απόκριση σε ηχητικό σύστημα	28
3.2 Μέτρηση Κρουστικής Απόκρισης με τη μέθοδο MLS.....	29
3.3 Μοντέλο All pole Filter	30
Κεφάλαιο 4	32
4.1 Περιγραφή του προβλήματος	32
4.2 Προσομοίωση ακουστικού χώρου και δημιουργία κρουστικής απόκρισης	32
4.2.1 Με τον Room Response	32
4.2.2 Με τον All Pole Filter.....	33
4.2.3 Μέθοδος γραμμικής πρόγνωσης.....	35
Κεφάλαιο 5	37
5.1 Εισαγωγή στην Μηχανική Μάθηση	37
5.2 Εξόρυξη Δεδομένων (Data Mining)	37
5.3 Αλγόριθμοι Εξόρυξης δεδομένων	38
5.3.1 Αλγόριθμος K-Means.....	38
5.3.2 Αλγόριθμος SVM (Support Vector Machine).....	39
5.3.3 Αλγόριθμος APRIORI.....	40
5.3.4 Αλγόριθμος AdaBoost.....	40
5.3.5 Αλγόριθμος NaiveBayes	41
5.3.6 Αλγόριθμος CART	43
5.3.7 Αλγόριθμος EM (Expectation Maximization)	44
Κεφάλαιο 6	47
6.1 Εισαγωγή στα Νευρωνικά Δίκτυα	47
6.1.1 Ανθρώπινο Ακουστικό Σύστημα.....	47
6.1.2 Μηχανικό σύστημα Ακρόασης	48

6.2 Εντοπισμός ήχου με τη Χρήση Νευρωνικών Δικτύων	49
6.3 Σύγκριση Μοντέλων DNN-GMM	51
Βιβλιογραφία	53

Περιεχόμενα Σχημάτων

Σχήμα 1.1 Συνθήκες ακρόασης στη σταδιακή μετάβαση από ανοιχτό σε κλειστό χώρο	13
Σχήμα 1.2 Κατοπτρική Ανάκλαση	15
Σχήμα 1.3 Φαινόμενο Ανάκλασης.....	17
Σχήμα 2.1 Χρόνος Αντήχησης RT60	19
Σχήμα 2.2 Βέλτιστος Χρόνος Αντήχησης RT60	20
Σχήμα 2.3 Απεικόνιση σχέσης χρόνου αντήχησης, όγκου και απορρόφησης	21
Σχήμα 2.4 Διάγραμμα εύρεσης του χρόνου αντήχησης κατά Norris – Eyring	22
Σχήμα 2.5 Απεικόνιση Μοντέλου Image Source	26
Σχήμα 3.1 Κρουστική απόκριση δωματίου	27
Σχήμα 3.2 Απόκριση σε Ανοιχτό Χώρο	28
Σχήμα 3.3 Απόκριση επιφάνειας που ανακλά τον ήχο.....	29
Σχήμα 3.4 Απόκριση κλειστού χώρου.....	29
Σχήμα 4.1 Κρουστική απόκριση δωματίου συναρτήση χρόνου και φάσματος	33
Σχήμα 4.2 Διαγράμματα αποτελεσμάτων του προσομοιωτή All Pole Filter	35
Σχήμα 5.1 Παράδειγμα Ομαδοποίησης του αλγορίθμου K-Means.....	38
Σχήμα 5.2 Παράδειγμα Αλγορίθμου SVM.....	39
Σχήμα 5.3 Παράδειγμα Αλγορίθμου APRIORI	40
Σχήμα 5.4 Παράδειγμα Αλγορίθμου EM	46

Περίληψη

Το θέμα της παρούσας πτυχιακής εργασίας είναι οι Εφαρμογές αλγορίθμων μηχανικής μάθησης στη χωρική ανάλυση της κρουστικής απόκρισης χώρου ακρόασης. Στην εργασία αυτή, αρχικά αναλύθηκε η ακουστική του ήχου και η διάδοση των ηχητικών κυμάτων σε κλειστό χώρο και σε ελεύθερο πεδίο όπως και οι βασικές ακουστικές παράμετροι. Πιο συγκεκριμένα, το πεδίο της ακουστικής χώρου εξετάζει τη διάδοση της ηχητικής ενέργειας και της αλληλεπίδρασης του ήχου με το περιβάλλον στο οποίο διαδίδεται. Επίσης, η ακουστική χώρου μελετάει την διάδοση των ηχητικών κυμάτων, τόσο σε ένα ελεύθερο πεδίο όσο και σε ένα μη ελεύθερο πεδίο, δηλαδή έναν κλειστό χώρο, λαμβάνοντας υπόψιν την ανάκλαση, την απορρόφηση, τη διάχυση και την διέλευση του ήχου στα εμπόδια που υπάρχουν στους χώρους αυτούς. Μέσα από την ακουστική χώρου, προκύπτουν παράμετροι οι οποίες μπορούν να χρησιμοποιηθούν προκειμένου να αξιολογήσουν έναν χώρο ως προς την ακουστική του. Αυτές οι παράμετροι είναι ο χρόνος αντήχησης, η ευκρίνεια, η ηχηρότητα, η αντιληπτότητα των συλλαβών καθώς και πιο ποιοτικές παράμετροι όπως είναι η ηχώ, ο χρωματισμός, η ζεστασιά και η οικειότητα. Παράλληλα με τις ακουστικές παραμέτρους, ένας χώρος μπορεί επίσης να μοντελοποιηθεί και με το μοντέλο ειδώλων πηγής για τη διάκριση των γραμμών διάδοσης του ήχου εντός του χώρου αυτού. Η κρουστική απόκριση ενός δωματίου, δηλαδή η απόκριση του δωματίου σε έναν κρουστικό παλμό, αποτελεί μία πλήρη περιγραφή της ακουστικής του δωματίου και των παραπάνω παραμέτρων. Ωστόσο, πολλές φορές είναι επιθυμητή η προσομοίωση της κρουστικής απόκρισης ενός πεδίου, χωρίς την πραγματική της μέτρηση. Σε τέτοιες περιπτώσεις χρειάζεται να προσομοιωθεί η κρουστική απόκριση ενός χώρου με δεδομένες τις παραπάνω παραμέτρους καθώς και επιπρόσθετα η διάδοση του ήχου στο χώρο αυτό. Αυτή η προσομοίωση περιγράφεται και υπολογίζεται στην συνέχεια της παρούσας εργασίας με τη χρήση δύο προσομοιωτών του Room Impulse Response και του All Pole Filter. Η υλοποίηση αυτή βασίζεται στο λογισμικό MATLAB και τις επιπλέον βιβλιοθήκες που διαθέτει. Το πεδίο της μηχανικής μάθησης μπορεί να προβλέψει τη διάδοση ενός ήχου σε ένα χώρο. Με τη χρήση μεθόδων εξόρυξης δεδομένων και την υλοποίηση νευρωνικών δικτύων είναι δυνατή η προσομοίωση της ακουστικής του χώρου από την πλευρά του ακροατή και συνεπώς η αξιολόγηση του χώρου ως προς την ακουστική του ιδιότητα.

Abstract

In this research we investigate the Applications of machine learning algorithms in the spatial analysis of the impulse response of an auditorium. In this research, firstly were analyzed the acoustics and sound propagation of sound waves indoors and free field such as and basic acoustic parameters. More specifically, the field of acoustics focuses on the transmission of sound energy and the sound reaction with the environment. Also, acoustics study the transmission of sound waves on a free field as much as on an enclosed space, taking into account sound properties such as reflection, absorption, diffusion and transition that occur inside these fields. Through acoustics, several acoustic parameters are distinguished like reverberation time, clarity, sound strength, definition and early delay time (EDT) as well as echo, coloration, warmth and intimacy. Apart from the acoustic parameters, a space can be modeled by the image source model which can identify the lines of sound transmission within that space. Impulse response, the acoustic response of a room on a pulse, fully describes the room acoustic ability and its acoustic parameters. However, many times it's more practical to simulate the impulse response of a field, without measuring the actual impulse response. On such occasions, impulse response and sound transmission within a space, needs to be simulated based on defined acoustic parameters. This simulation described and simulated in the following research using two simulators Room Impulse Response and All Pole Filter. This implementation is based on software MATLAB and the additional libraries which are available. Through the field of machine learning, it is possible to predict the spatial transmission of the sound. With data mining methods and the utilization of neural networks it is possible to simulate the room acoustics from the listener's perspective and thus evaluate the the room acoustic ability.

Κεφάλαιο 1

1.1 Εισαγωγή

Η ακουστική των χώρων είναι ένα θέμα που απασχόλησε τον άνθρωπο από τα αρχαία χρόνια. Η ακουστική αφορά τον κλάδο της φυσικής η οποία μελετά τις ιδιότητες, τη συμπεριφορά του ήχου καθώς και τις εφαρμογές του. Επιπλέον μελετά την παραγωγή, τη μετάδοση, τη λήψη και την αλληλεπίδραση του ήχου με τα διάφορα υλικά μέσα, καθώς επίσης και τις τεχνικές ελέγχου και επεξεργασίας για την εκμετάλλευση του ήχου σε εξειδικευμένες εφαρμογές.

1.2 Ιστορική Αναδρομή

Η ακουστική ξεκίνησε τον 6ο αιώνα π.Χ με τις μελέτες του Πυθαγόρα και των Πυθαγορείων οι οποίοι κατάφεραν να προσδιορίσουν τις σχέσεις που υπάρχουν μεταξύ του μήκους μιας παλλόμενης χορδής και του ύψους του παραγόμενου ήχου. Οι Πυθαγόρειοι διαχώρισαν τις δύο τυπικές όψεις των σπουδών της ακουστικής: τη μουσική όψη και τη φυσικομαθηματική όψη. Μετά την ελληνορωμαϊκή αρχαιότητα ο Leonardo da Vinci (1452-1519) συνειδητοποίησε ότι απαιτείται να υπάρχει αέρας ως μέσο για τη διάδοση του ήχου και ότι ο ήχος διαδίδεται με πεπερασμένη ταχύτητα. Η φυσική μελέτη και ερμηνεία της ακουστικής σημείωσε πρόοδο με τον Γαλιλαίο (1564–1642) και έναν σύγχρονό του, τον Γάλλο Maren Mersen(1588–1648), οι οποίοι προσδιόρισαν πειραματικά τις μαθηματικές σχέσεις μεταξύ συχνότητας, μήκους, τάσης και μάζας μιας παλλόμενης χορδής, συμπληρώνοντας το έργο των Πυθαγορείων πριν από 2000 χρόνια. Ο Joseph Sauveur (1653-1716) εισήγαγε τον όρο "ακουστική" για τη μελέτη του ήχου, το 1701. Η μαθηματική θεωρία διάδοσης του ήχου μπορεί να θεωρηθεί ότι αρχίζει με τον Isaak Newton(1642-1727), του οποίου το έργο "Principia" (1687) περιείχε μια μηχανική ερμηνεία των παλμών ηχητικής πίεσης και της ταχύτητας των κυμάτων κατά τη διάδοσή τους σε στερεά μέσα. Έτσι υπολόγισε πρώτος την ταχύτητα του ήχου λαμβάνοντας υπόψην του κάποιες θεωρητικές εκτιμήσεις. Η ακουστική θεωρία σε σταθερότερη μαθηματική και φυσική βάση διατυπώθηκε από τους Euler (1707-1783), Lagrange (1736-1813) και d'Alembert (1717-1783). Ο Leonhard Euler διατύπωσε την κυματική εξίσωση για τον ήχο, στη μορφή που χρησιμοποιείται σήμερα. Ο Wallace Clement Sabine (1868-1919), θεωρείται ότι ήταν ο πρώτος που συνέβαλε στην ανάπτυξη αυτού που ονομάζεται ακουστική χώρων ή αρχιτεκτονική ακουστική, με στόχο τη βελτίωση της ακουστικής ποιότητας των δωματίων. Η εισαγωγή των ηλεκτρονικών λυχνιών και η εφεύρεση του σωλήνα ηλεκτρονίων (1907) συνέβαλε αποφασιστικά στο πεδίο της ακουστικής, καθώς επέτρεψε την κατασκευή μεγαφώνων, μικροφώνων, ενισχυτών και καταγραφέων για συχνότητες, ακουστικές και μη, ενώ οι ακουστικές μετρήσεις βελτιώθηκαν σε ανάλυση και ακρίβεια. Η σύγχρονη αντίληψη

της ακουστικής χαρακτηρίζεται από την τάση να διαχωρίζονται οι διάφορες επιστημονικές και τεχνικές πλευρές. Έτσι λοιπόν ο ίδιος ο όρος ήχος, που κάποτε οριζόταν αναλυτικά είτε ως φυσική διαταραχή έξω από το αφτί είτε, κατ' αντιδιαστολή, ως φυσιολογικό αίσθημα, θεωρείται σήμερα το σύνολο των φυσικών φαινομένων έξω και μέσα στο αφτί και στον ανθρώπινο εγκέφαλο, που συνδέονται με τη διαταραχή, μέχρι το σημείο όπου η ψυχοακουστική, ανατομική και ηλεκτροφυσιολογική έρευνα είναι ικανή μέχρι σήμερα να διαπιστώσει τη διάδοση του σήματος.

1.3 Ήχος

Ο ήχος είναι η αίσθηση που προκαλείται λόγω της διέγερσης των αισθητηρίων οργάνων της ακοής από μεταβολές πίεσης του ατμοσφαιρικού αέρα. Αυτές οι μεταβολές διαδίδονται με τη μορφή ηχητικών κυμάτων.

1.3.1 Διάδοση ήχου σε ελεύθερο πεδίο (free field)

Ο ήχος διαδίδεται σε γενικές γραμμές ελεύθερα σε έναν χώρο χωρίς εμπόδια, με μοναδική αλλοίωση την ενεργειακή απόσβεση λόγω της τριβής των μορίων του αέρα. Έτσι τα ηχητικά κύματα διαδίδονται σε ευθεία διεύθυνση και η ελεύθερη διάδοση του ήχου συγχέεται με τις συνθήκες που επικρατούν στους ανοιχτούς χώρους. Όταν ο χώρος δεν έχει εμπόδια οι επιφάνειες που αλληλεπιδρούν με το ηχητικό κύμα είναι ελάχιστες και κυριότερη από όλες τις επιφάνειες είναι αυτή του εδάφους. Τα πιο χαρακτηριστικά φαινόμενα κατά τη διάδοση του ήχου σε ανοιχτούς χώρους είναι η καθυστέρηση, η απόσβεση και η ηχώ. Στο ελεύθερο πεδίο ο ρυθμός μείωσης της πίεσης του ήχου είναι σταθερός και ίσος με 6 dB για κάθε διπλασιασμό της απόστασης, στην περίπτωση σφαιρικών ηχητικών κυμάτων.

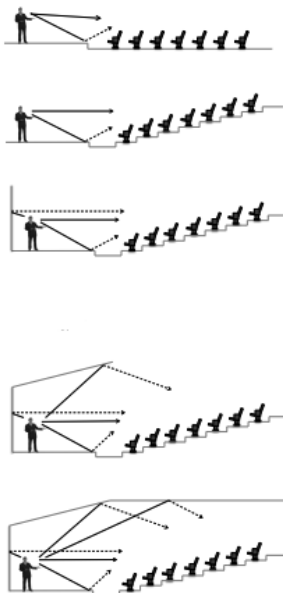
1.3.2 Διάδοση ήχου σε μη ελεύθερα πεδία

Είναι χαρακτηριστική η διαφορά της συμπεριφοράς που έχει ο ήχος σε κλειστούς χώρους σε σχέση με τους ανοιχτούς. Ο ήχος είναι ένα φαινόμενο που δεν υπάρχει στατικά. Για να παραχθεί και να διαδοθεί ο ήχος σε κάποιο μέσο απαιτείται η ύπαρξη ενός ελαστικού μέσου το οποίο θα μπορεί να βρίσκεται είτε σε στερεή, είτε σε υγρή, είτε σε αέρια κατάσταση και να υπακούει στους νόμους διατήρησης της μάζας και της ορμής. Με αυτόν τον τρόπο δεν μπορεί να διαδοθεί κανένας ήχος αφού δεν υπάρχει κανένα ελαστικό μέσο.

Το ηχητικό κύμα όταν ταξιδεύει, καταναλώνει την αρχική ενέργεια που του έχει δοθεί και συναντά πολλών ειδών εμπόδια, τα οποία μεταβάλλουν τα χαρακτηριστικά του. Τα ηχητικά κύματα σπανίως διαδίδονται σε εντελώς ελεύθερα πεδία. Έτσι είναι πολύ πιθανό να

συναντήσουν στην διαδρομή τους πολλών ειδών εμπόδια: τα φυσικά ή τεχνητά, τα ορατά ή αόρατα. Για ένα ηχητικό κύμα, σαν εμπόδιο μπορεί να θεωρηθεί οποιαδήποτε μεταβολή της φύσης του μέσου διάδοσης, βέβαια για την ακουστική κλειστών χώρων μεγαλύτερο ενδιαφέρον παρουσιάζουν τα τεχνητά και ορατά εμπόδια, όπως οι επιφάνειες και τα έπιπλα ενός δωματίου.

Σε έναν κλειστό χώρο το ηχητικό πεδίο «φυλακίζεται» από τις περιβάλλουσες επιφάνειες και η ενέργειά του διατηρείται μεγαλύτερο χρονικό διάστημα εντός του δωματίου, αντίθετα με ό,τι συμβαίνει στο ύπαιθρο. Το φαινόμενο αυτό ονομάζεται αντήχηση. Η αντήχηση εμφανίζει διαφορετικά χαρακτηριστικά ανάλογα με τη μορφή του χώρου στον οποίο βρίσκεστε. Συνήθως σε ένα μικρό χώρο ο ήχος σβήνει πιο γρήγορα από ό,τι σε ένα μεγαλύτερο. Ο χρόνος, ο οποίος χρειάζεται να παρέλθει για να μηδενιστεί η πίεση του ήχου ύστερα από τη διακοπή της πηγής σε ένα δωμάτιο, αποτελεί ένα από τα σημαντικότερα μεγέθη στην ακουστική κλειστών χώρων και ονομάζεται χρόνος αντήχησης (RT, Reverberation Time). Η αίσθηση της αντήχησης οφείλεται στις πολλαπλές ανακλάσεις του ήχου, οι οποίες φθάνουν στο σημείο ακρόασης μέσω της πρόσκρουσης των κυμάτων στους τοίχους, ενώ η χρονική απόσταση που τις χωρίζει είναι πολύ μικρή (μερικών msec).



Σχήμα 1.1 Συνθήκες ακρόασης στη σταδιακή μετάβαση από ανοιχτό σε κλειστό χώρο

Κάποια από τα μεγαλύτερα προβλήματα που αντιμετωπίζει η ακουστική είναι: τα ηχητικά κύματα που είναι αόρατα, η συμπεριφορά τους η οποία είναι εντελώς διαφορετική ανάλογα με τη συχνότητα τους και τα αποτελέσματα της συμβολής τους με τα διάφορα εμπόδια που συναντούν στην πορεία τους. Όταν ο ήχος διαδίδεται σε κλειστό χώρο λαμβάνουν χώρα διάφορα φαινόμενα, κάποια από αυτά είναι η ανάκλαση, η απορρόφηση, η διάχυση, η περίθλαση, η διάδοση μέσω του εμποδίου και η διασπορά.

Κλειστοί χώροι και Συμπεριφορά του ήχου

Ο ήχος έχει διαφορετική συμπεριφορά μέσα σε κλειστούς χώρους σε σχέση με τους ανοιχτούς χώρους. Στους ανοιχτούς χώρους, ο ήχος φθάνει απευθείας από την πηγή στον δέκτη, ενώ σε κλειστούς χώρους, ο ήχος εκτός από την απευθείας διάδοση φθάνει στον δέκτη μετά από διαδοχικές ανακλάσεις στα τοιχώματα του χώρου με αποτέλεσμα να αλλοιώνεται η αρχική κυματομορφή του εκπεμπόμενου ήχου.

Βασικό ρόλο στην ακουστική συμπεριφορά κλειστών χώρων παίζει η απορρόφηση του ήχου και οι διαστάσεις του χώρου. Όταν μελετάμε την ακουστική των κλειστών χώρων θα πρέπει να έχουμε υπόψη μας την κυματική φύση του ήχου. Οι χώροι όσο αφορά τον τρόπο μελέτης τους διακρίνονται σε μικρούς, μεγάλους και πολύ μεγάλους. Ένας χώρος θεωρείται μεγάλος όταν η μικρότερη διάστασή του είναι κατά δύο μήκη κύματος τουλάχιστον μεγαλύτερη από το μήκος κύματος της συχνότητας που μας ενδιαφέρει. Ο ίδιος ο χώρος θεωρείται πολύ μεγάλος όταν οι αποστάσεις των επιφανειών που ορίζουν τον χώρο είναι τόσο μεγάλες ώστε η απορρόφηση του ήχου από τον αέρα να είναι σημαντική.

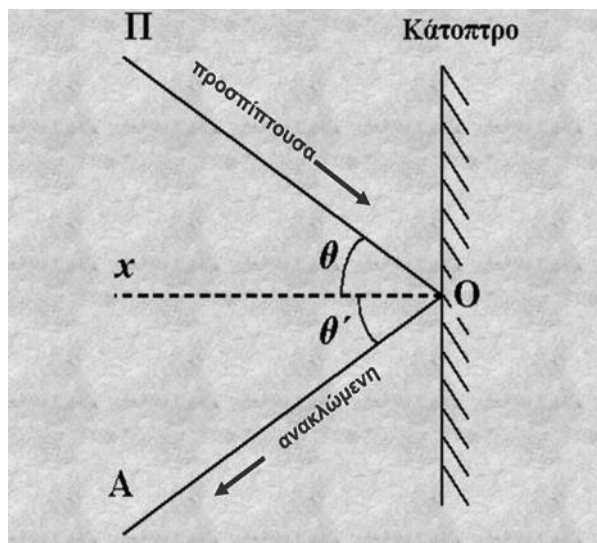
Υπάρχουν τρεις τρόποι μελέτης της ακουστικής συμπεριφοράς των κλειστών χώρων:

- A. Η κυματική ακουστική: με τη μέθοδο αυτή μελετάμε τους μικρούς χώρους όπου τα κυματικά φαινόμενα και κυρίως αυτά της συμβολής είναι πολύ έντονα.
- B. Η στατιστική ακουστική: η μέθοδος αυτή χρησιμοποιείται σε μεγάλους και πολύ μεγάλους χώρους. Η προσέγγιση του προβλήματος γίνεται με τη βοήθεια της στατιστικής ανάλυσης και αυτό γίνεται με τα κυματικά φαινόμενα που είναι πολύ «μπλεγμένα».
- C. Η γεωμετρική ακουστική: Και αυτή η μέθοδος χρησιμοποιείται σε μεγάλους χώρους, οι δε υπολογισμοί γίνονται με την βοήθεια των τύπων της γεωμετρικής οπτικής.

1.4 Ανάκλαση του ήχου (Reflection)

Όπως ανακλάται το φως, έτσι και ο ήχος ανακλάται όταν πέσει πάνω σε μια λεία, αρκετά μεγάλη και σκληρή επιφάνεια, όσο γίνεται χωρίς προεξοχές και κοιλότητες. Ανακλάσεις είναι η αιτία και των στάσιμων κυμάτων αλλά στο παρόν, ως ανακλάσεις θεωρούμε φαινόμενα τα οποία λαμβάνουν χώρα στην περιοχή συχνοτήτων, όπου τα μήκη κύματος (λ) είναι αρκετά μικρά, έτσι ώστε να εφαρμόζουμε τις αρχές της γεωμετρικής ακουστικής. Μια ανάκλαση ηχητικού κύματος δεν είναι διαφορετική από την ανάκλαση οπτικής ακτίνας. Μπορούμε να θεωρήσουμε ότι ο ήχος διαδίδεται ευθύγραμμα και ότι όταν συναντήσει ένα ακλόνητο εμπόδιο υπό μια γωνία, αλλάζει κατεύθυνση, δηλαδή ανακλάται και επιστρέφει στον χώρο.

Η γωνία πρόσπτωσης (θ) και η γωνία ανάκλασης (θ') είναι ίσες. Ως γωνία πρόσπτωσης ορίζουμε την γωνία που σχηματίζει η προσπίπτουσα ακτίνα με την κάθετη στην επιφάνεια (x). Αντίστοιχα ως γωνία ανάκλασης ορίζουμε την γωνία που σχηματίζει η ανακλώμενη ακτίνα με την κάθετη στην επιφάνεια. Αυτό φαίνεται στο Σχήμα 1.2.



Σχήμα 1.2 Κατοπτρική Ανάκλαση

Η ανάκλαση του ήχου είναι η αιτία για την ηχώ και την αντήχηση. Κατά την ηχώ ή «αντίλαλο» ο ακροατής ακούει τον ήχο που εκπέμπει η πηγή ήχου και τον ανακλώμενο ήχο σαν δυο διαφορετικούς ήχους, που ο ένας ακούγεται μετά τον άλλο.

1.5 Απορρόφηση του ήχου

Ο ήχος, όπως και το φως μπορεί να απορροφηθεί, δηλαδή να «εξαφανιστεί». Και συγκεκριμένα όταν ο ήχος συναντήσει στην πορεία του μια μαλακή ή πορώδη ή τραχιά επιφάνεια, τότε ένα πολύ μεγάλο μέρος του απορροφάται από την επιφάνεια λόγω των εσωτερικών τριβών και των συντονισμών, που προκαλούνται μέσα στο υλικό και ένα μικρό μέρος αλλάζει κατεύθυνση και ανακλάται. Το φαινόμενο αυτό ονομάζεται απορρόφηση και έχει μεγαλύτερη ισχύ στις υψηλές συχνότητες οι οποίες απορροφώνται εύκολα λόγω του μικρού μήκους κύματος από πορώδη υλικά. Συνήθως, η ηχητική ενέργεια μετατρέπεται σε θερμική. Η απορρόφηση εξαρτάται εκτός από το υλικό στο οποίο διαδίδεται ο ήχος και από την απόσταση ανάμεσα στην ηχητική πηγή και το δέκτη.

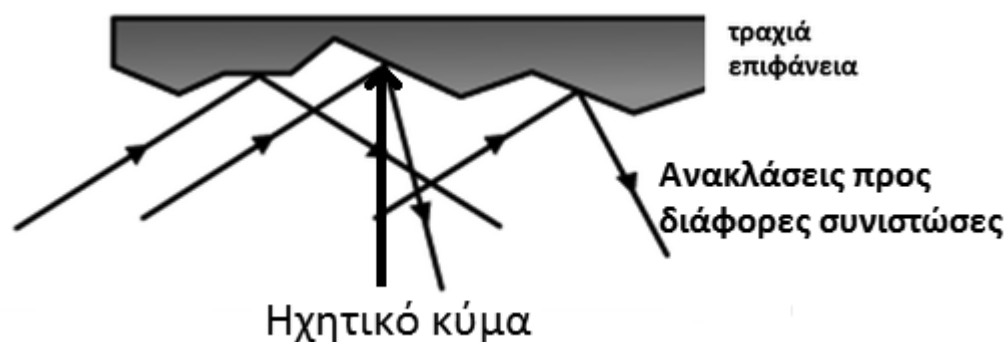
Συντελεστής απορρόφησης

Ο συντελεστής απορρόφησης μιας επιφάνειας ή ενός υλικού, σε ορισμένη συχνότητα και ορισμένες συνθήκες είναι ο λόγος της ηχητικής ισχύος που απορροφάται από την επιφάνεια, προς την ηχητική ισχύ που προσπίπτει στην επιφάνεια. Ο συντελεστής απορρόφησης συμβολίζεται με α , είναι αδιάστατο μέγεθος και έρχεται σε συνάρτηση με την γωνία πρόσπτωσης. Η τιμή του συντελεστή απορρόφησης α κυμαίνεται από το 0 που είναι για πλήρη ανάκλαση έως το 1 που είναι για πλήρη απορρόφηση.

1.6 Διάχυση του ήχου

Η διάχυση είναι το φαινόμενο κατά το οποίο ένα ηχητικό κύμα, αρχικά ομαλό (π.χ. επίπεδο ή σφαιρικό), προσπίπτει σε μια ανώμαλη επιφάνεια. Το αποτέλεσμα είναι ένα σύνολο ανακλάσεων, διαθλάσεων και περιθλάσεων, έτσι ώστε να μην μπορεί να θεωρηθεί πια ως ένα κύμα, αλλά μόνο ως μια πυκνότητα ηχητικής ενέργειας, μεταβλητή από σημείο σε σημείο, η οποία συνιστά και τη χαρακτηριστική παράμετρο του φαινομένου. Ένα τέλει διαχυτικό πεδίο ήχου είναι αυτό που έχει ορισμένες ακουστικές ιδιότητες οι οποίες παραμένουν ίδιες οπουδήποτε μέσα στο χώρο. Ένα μη διαχυτικό ηχητικό πεδίο έχει διαφορετικό χρόνο αντήχησης, αφού ο ακροατής μετακινείται μέσα στο χώρο. Όταν έχουμε ικανοποιητική διάχυση σε μια αίθουσα, οι ακροατές έχουν την αίσθηση ότι ο ήχος έρχεται από όλες τις κατευθύνσεις, με τις ίδιες στάθμες. Για την ακουστική βελτίωση των χώρων ακρόασης και κυρίως για μικρούς χώρους χρησιμοποιούνται οι διαχυτές.

Στην διάχυση, όταν η επιφάνεια δεν είναι λεία τότε το αρχικό κύμα ανακλάται προς διάφορες συνιστώσες, όπως φαίνεται και στο Σχήμα 1.3.



Σχήμα 1.3 Φαινόμενο Ανάκλασης

1.7 Διέλευση του ήχου

Εάν δεν ισχύουν οι ιδανικές συνθήκες τότε ένα ποσοστό της ενέργειας του ηχητικού κύματος εισχωρεί στο εμπόδιο. Όταν αποκτά αυτήν την ενέργεια, το εμπόδιο τίθεται σε ταλάντωση δημιουργώντας ένα νέο ηχητικό κύμα που εκπέμπεται από την άλλη πλευρά. Η αποφυγή της διέλευσης του ήχου από την πίσω πλευρά του εμποδίου είναι το πιο σημαντικό πρόβλημα στηχομόνωση χώρων και επιτυγχάνεται μόνο αν καταφέρουμε να μην δονηθεί το εμπόδιο. Αυτό εξαρτάται από την ελαστικότητα και την πυκνότητά του, οπότε αν αυξήσουμε την μάζα του υλικού σταματάμε τη διέλευση του ήχου.

Κεφάλαιο 2

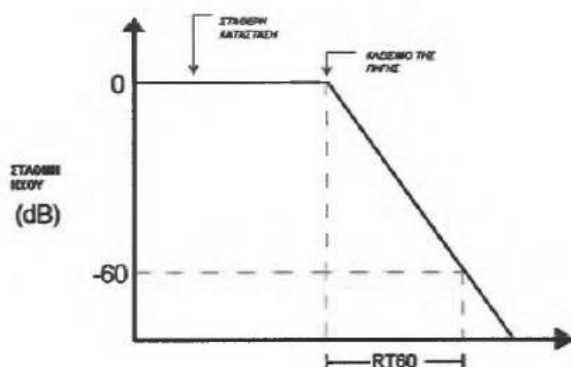
2.1 Εισαγωγή

Από τον 19^ο αιώνα, η ακουστική συμπεριφορά των χώρων δεν λαμβανόταν υπόψη όταν σχεδιάζονταν οι χώροι. Στην εποχή της Αναγέννησης το 1850, ο Joseph Henry για πρώτη φορά βελτίωσε την ακουστική συμπεριφορά των κλειστών χώρων εφόσον είχε γίνει η μετάβαση από τα ανοικτά θέατρα στους κλειστούς χώρους. Εκτός από αυτό, ο Henry παρατήρησε ότι οι κλειστοί χώροι είχαν κακή ακουστική εξαιτίας του φαινομένου της ηχούς και της εστίασης των ήχων σε συγκεκριμένα σημεία των δημοσίων χώρων. Αργότερα ο Wallace Sabine, καθηγητής στο Harvard University, συνειδητοποίησε ότι η αντήχηση ήταν πολύ σημαντική για την ακουστική ποιότητα των χώρων και για την επίδραση του που είχε η απορρόφηση των υλικών στην αντήχηση των χώρων.

2.2 Βασικές Ακουστικές Παράμετροι

2.2.1 Χρόνος αντήχησης RT60 (Reverberation Time)

Ο χρόνος αντήχησης ορίζεται σαν τον χρόνο που ο ήχος εξασθενεί κατά 60 ντεσιμπέλ (dB) σε έναν χώρο δηλαδή όταν έχουμε ένα πολύ δυνατό ήχο, χρειάζεται κάποιος χρόνος έτσι ώστε να εξασθενήσει τόσο πολύ μέχρι να μην ακούγεται. Ο χρόνος αντήχησης μετριέται σε δευτερόλεπτα (msec). Ο χρόνος μπορεί εύκολα να μετρηθεί με ειδικά όργανα μέτρησης, όμως είναι δύσκολο να βρούμε το περιθώριο των 60 dB έτσι ώστε να μετρήσουμε το χρόνο αντήχησης επειδή ο θόρυβος του περιβάλλοντος είναι τέτοιος και για να βρούμε το περιθώριο των 60 dB θα πρέπει η στάθμη της ισορροπίας των χώρων να είναι της τάξης των 120 dB. Σαν στάθμη ισορροπίας ενός χώρου εννοούμε την παραγωγή ενός ήχου που αυξάνεται μέχρι μια συγκεκριμένη στάθμη δηλαδή χάνεται μια ορισμένη ποσότητα ηχητικής ενέργειας εξαιτίας των τριβών με τον αέρα και τις επιφάνειες αφού έχει σταθεροποιηθεί η ενέργεια μέσα στο χώρο.



Σχήμα 2.1 Χρόνος Αντήρησης RT60

Ο βέλτιστος χρόνος αντήρησης αποτελεί ένα πολύ σημαντικό κριτήριο για την καλή ακουστική ενός χώρου. Κάθε χώρος χρησιμοποιείται για διαφορετικούς σκοπούς με αποτέλεσμα να απαιτείται διαφορετικός χρόνος αντήρησης για κάθε από αυτούς, δηλαδή σε έναν χώρο που γίνεται μια ομιλία, ο χρόνος αντήρησης δεν είναι ο ίδιος όπως σε μια συναυλία. Ο βέλτιστος χρόνος αντήρησης μας φανερώνει τις προτεινόμενες τιμές του χρόνου αντήρησης που έχει ένας χώρος ανάλογα με τον όγκο του και τον σκοπό που εξυπηρετεί. Οι διαφορετικές απαιτήσεις για βέλτιστο χρόνο αντήρησης οδήγησαν στην δημιουργία χώρων με μη σταθερή ακουστική που υπάρχουν για πολλούς σκοπούς. Στον παρακάτω Σχήμα 2.2 θα παρατηρήσουμε το εύρος του χρόνου αντήρησης που μπορεί να έχουν κάποιοι χώροι και οι χρήσεις τους.

ΧΩΡΟΣ-ΧΡΗΣΗ	Βέλτιστος Χρόνος Αντήρησης RT60 (sec)
Μικρά στούντιο	0,4-0,6
Αίθουσες σχολείου	0,5-0,8
Διαλέξεις	0,6-1,4
Κινηματογράφοι	0,65-1,3
Ορχήστρες	0,8-1,3
Μικρά θέατρα	1,1-1,5
Ρεσιτάλ Δωματίου	1,2-1,9
Όπερα	1,3-1,9
Αμφιθέατρα	1,3-1,9
Εκκλησίες	1,3-3,4
Χορωδικά- Συμφωνική	1,6-2,2

Σχήμα 2.2 Βέλτιστος Χρόνος Αντήχησης RT60

Θεωρητικός Υπολογισμός του Χρόνου Αντήχησης:

A. Εξίσωση Sabine

Ο χρόνος αντήχησης σχετίζεται με τον όγκο και τη συνολική απορρόφηση των επιφανειών ενός δωματίου. Ο μαθηματικός τύπος υπολογισμού του χρόνου αντήχησης κατά τον Sabine είναι ο εξής:

$$RT60 = 0.161 * V / S * \alpha \quad (2.1)$$

Όπου,

1. RT60: ο χρόνος αντήχησης σε second
2. 0.161: σταθερά
3. V: ο όγκος του δωματίου σε κυβικά μέτρα (m^3)
4. S: η συνολική επιφάνεια του δωματίου σε τετραγωνικά μέτρα (m^2)
5. α : ο μέσος συντελεστής απορρόφησης των επιφανειών του δωματίου
6. $S * \alpha$: η συνολική απορρόφηση του Sabine

(Θέτουμε όπου $S * \alpha = A$)

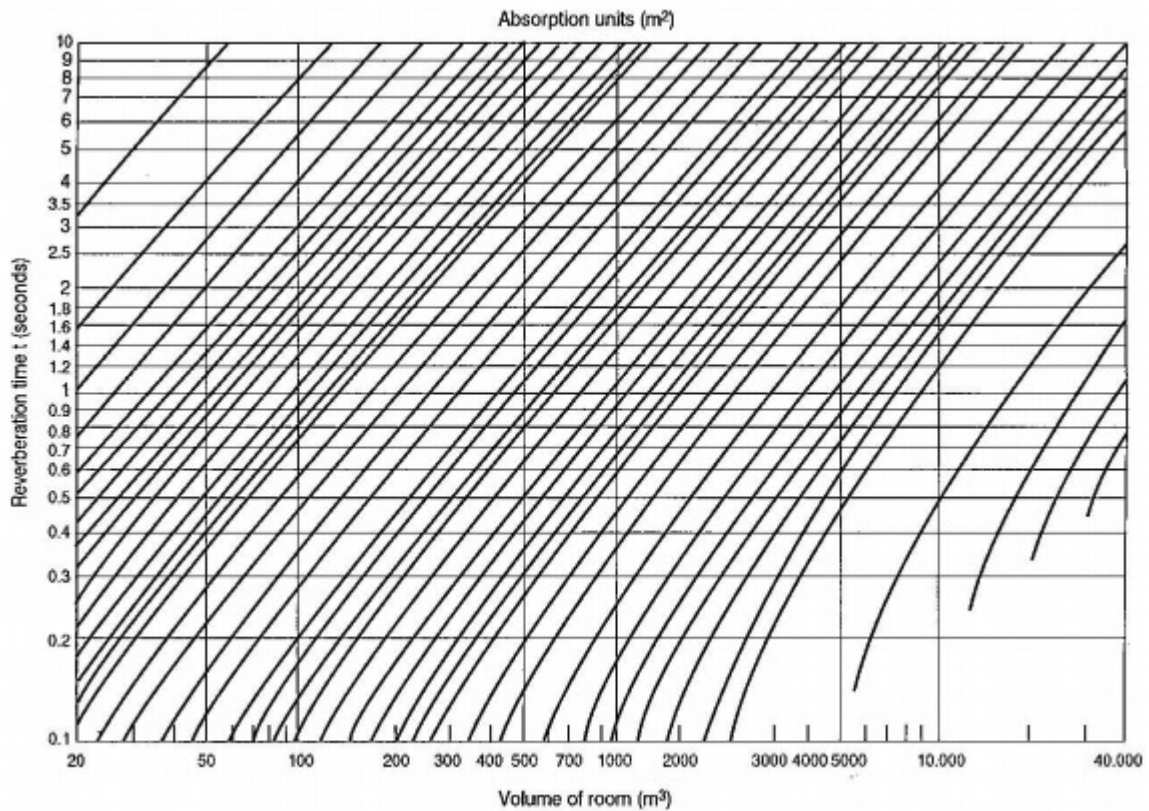
Η συνολική απορρόφηση A δίνεται από τον τύπο:

$$A = \sum S_i * \alpha_i \quad (2.2)$$

Όπου,

1. S: το εμβαδόν κάθε επί μέρους επιφάνειας
2. α : αντίστοιχοι συντελεστές απορρόφησης

Στην παρακάτω Σχήμα 2.3 απεικονίζεται η σχέση μεταξύ του χρόνου αντήχησης, του όγκου και της απορρόφησης του χώρου.



Σχήμα 2.3 Απεικόνιση σχέσης χρόνου αντήχησης, όγκου και απορρόφησης

B. Εξίσωση Norris – Eyring

Η εξίσωση Norris – Eyring δημιουργήθηκε έτσι ώστε να έχουμε μια πιο ακριβής προσέγγιση του χρόνου αντήχησης και αυτή υποθέτει ότι ο ήχος συναντάει όλες τις επιφάνειες με ίση πιθανότητα και σε όλες τις δυνατές γωνίες πρόσπτωσης με μία μέση ελεύθερη διαδρομή V . Η εξίσωση αυτή έχει καλύτερα αποτελέσματα σε μεγάλους χώρους, η οποία είναι η εξής:

$$RT60 = -0.161 * V / S * \ln (1-\alpha_{\mu}) \quad (2.3)$$

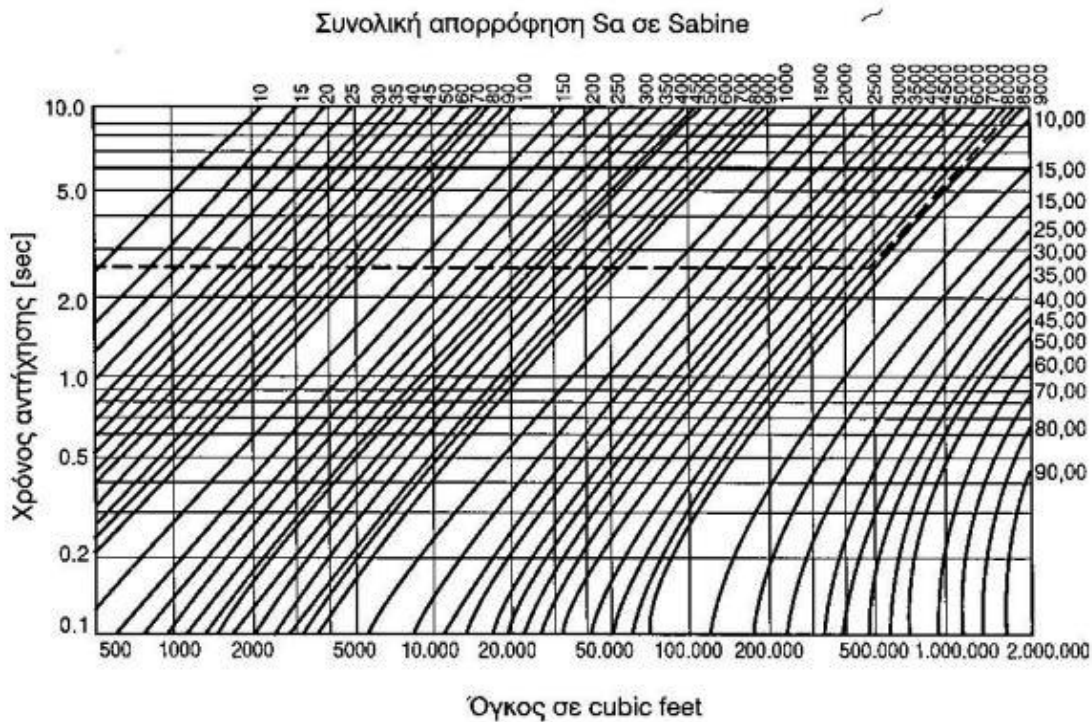
Όπου,

1. RT60: ο χρόνος αντήχησης σε second
2. 0.161: σταθερά
3. V : ο όγκος του δωματίου σε κυβικά μέτρα (m^3)
4. S : η συνολική επιφάνεια του δωματίου σε τετραγωνικά μέτρα (m^2)

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

5. \ln : λογαριθμική συνάρτηση
6. α_m : ο μέσος συντελεστής απορρόφησης των επιφανειών του δωματίου
7. $1-\alpha$: η αναλογία της ενέργειας του κύματος που ανακλάται πίσω στο δωμάτιο.

Στην παρακάτω Σχήμα 2.4 απεικονίζεται το διάγραμμα εύρεσης του χρόνου αντήχησης κατά Norris – Eyring.



Σχήμα 2.4 Διάγραμμα εύρεσης του χρόνου αντήχησης κατά Norris – Eyring

2.2.2 Ευκρίνεια (Clarity)

Η ευκρίνεια είναι σε μεγάλο βαθμό μια ιδότητα συμπληρωματική προς την αντήχηση, επίσης είναι ανάλογη με τον λογάριθμο του λόγου του ήχου που φτάνει τα πρώτα 50 ή 80 msec ως προς αυτόν που φτάνει μετά τα 50 ή 80 msec. Το όριο του ολοκληρώματος είναι ανάλογα 50 ή 80 msec. Η C_{50} χρησιμοποιείται για ομιλία ενώ η C_{80} για μουσική. Η μονάδα μέτρησης της ευκρίνειας είναι τα dB. Όσο πιο μεγάλος είναι ο χρόνος αυτός, τόσο καλύτερη αντιληπτότητα και ευκρίνεια θα υπάρχει σε αντίθεση με την αντήχηση. Στο παρακάτω ολοκλήρωμα παρουσιάζεται η ευκρίνεια (C_{50} ή C_{80}):

$$C_{(50)} = 10 * \log \frac{\int_0^{50} h^2(t) dt}{\int_{50}^{\infty} h^2(t) dt} \quad (2.4)$$

ή

$$C_{(80)} = 10 * \log \frac{\int_0^{80} h^2(t) dt}{\int_{80}^{\infty} h^2(t) dt} \quad (2.5)$$

Με $h(t)$ την κρουστική απόκριση του χώρου.

2.2.3 Ηχηρότητα (Sound Stength, G10)

Είναι το επίπεδο της ηχητικής πίεσης που λειτουργεί στα 10 μέτρα από την πηγή όσο αφορά το ελεύθερο πεδίο της ηχητικής πίεσης. Η ηχηρότητα ορίζεται σαν το δεκαδικό λογάριθμο του λόγου του τετραγώνου της ηχητικής ενέργειας της κρουστικής απόκρισης ως προς το τετράγωνο της ηχητικής ενέργειας της κρουστικής απόκρισης που μετριέται σε ελεύθερο πεδίο και βρίσκεται σε απόσταση 10 m από την πηγή. Η μαθηματική σχέση από την οποία περιγράφεται είναι η εξής:

$$G = 10 * \log \frac{\int_0^{\infty} h^2(t) dt}{\int_0^{\infty} h_{10}^2(t) dt} \quad (2.6)$$

Όπου,

1. $h(t)$: κρουστική απόκριση
2. $h_{10}(t)$: κρουστική απόκριση, σε απόσταση 10 μέτρων πηγή-δέκτη.

Σύμφωνα με τον Barron, σαν ηχηρότητα υπολογίζεται η λογαριθμική εξίσωση της συνολικής ηχητικής στάθμης μείον την ηχητική στάθμη του απευθείας ήχου σε απόσταση 10 m από την πηγή.

$$G = 10 * \log (h_{total}^2) - 10 * \log (h_{10}^2) \quad (2.7)$$

Όπου,

1. $h_{10}(t)$: στιγμιαία ηχητική πίεση της κρουστικής απόκρισης σε απόσταση 10 m από την πηγή που βρίσκεται σε ελεύθερο πεδίο.
2. $h_{total}(t)$: η συνολική κρουστική απόκριση.

Η κρουστική απόκριση μετριέται σε msec.

2.2.4 Αντιληπτότητα των συλλαβών (Definition)

Η αντιληπτότητα είναι μια παρόμοια παράμετρος με την ευκρίνεια. Αυτό που την διαφοροποιείται είναι ότι χρησιμοποιείται για να χαρακτηρίσει σήματα ομιλίας. Ως αντιληπτότητα ορίζεται το ποσοστό του αριθμού των συλλαβών που γίνονται αντιληπτές από

έναν μέσο ακροατή, προς το σύνολο των συλλαβών που εκφωνούνται κατά τη διάρκεια μιας συνήθους ομιλίας. Ο Thiele προσπάθησε για πρώτη φορά να αναπτύξει την έννοια της διακριτότητας η οποία ορίζεται ως την λογαριθμική αναλογία, της ηχητικής ενέργειας της κρουστικής απόκρισης, κατά τη διάρκεια των πρώτων t_2 σε θέση μέτρησης, ως προς την συνολική ηχητική ενέργεια της κρουστικής απόκρισης. Η αντιληπτότητα εξαρτάται από τον λόγο σήματος προς το θόρυβο (SNR) και από τον χρόνο αντήχησης. Η αντιληπτότητα ορίζεται από τον τύπο:

$$D_{50} = 10 * \log \frac{\int_0^{50} h^2(t) dt}{\int_0^{\infty} h^2(t) dt} \% \quad (2.8)$$

Όπου,

1. $h(t)$: κρουστική απόκριση στο σημείο λήψης.

2.3 Πρώιμος Χρόνος Αντήχησης (Early Delay Time – EDT)

Οι πρώτες ανακλάσεις παίζουν καθοριστικό ρόλο στην διαμόρφωση της ακουστικής των χώρων, σε αντίθεση με τις καθυστερημένες ανακλάσεις που δεν μεταφέρουν ηχητική ενέργεια και έχουν μια μικρή επίδραση στην συνολική ακουστική. Όταν ο ρυθμός της πτώσης δεν έχει ομαλή κλίση, τότε χρησιμοποιούμε τον πρώιμο χρόνο αντήχησης. Ο πρώιμος χρόνος αντήχησης είναι ο χρόνος που απαιτείται για την πτώση των πρώτων 10 dB στη στάθμη του ήχου. Ο χρόνος που λαμβάνει υπόψη του τις πρώτες ανακλάσεις, ορίζεται ως έξι φορές η χρονική περίοδος που χρειάζεται ώστε να μειωθεί κατά 10 dB μετά τον μηδενισμό της πηγής. Ο πρώιμος χρόνος αντήχησης χρησιμοποιεί το ίδιο μέτρο και ποσοστό με το χρόνο αντήχησης, η μόνη διαφορά τους είναι το διάστημα το οποίο είναι από το 0 έως -10.

$$EDT = 6 (t_{-10}) \quad (2.9)$$

Ο EDT μετριέται σε sec και είναι μικρότερος από τον RT (χρόνος αντήχησης) και τον RT_{60} .

2.4 Άλλες Ακουστικές Παράμετροι

2.4.1 Ηχώ

Η ηχώ, αλλιώς αντίλαλος, αποτελεί ακουστικό φαινόμενο που οφείλεται στην ανάκλαση των ηχητικών κυμάτων. Όταν μια πηγή εκπέμπει έναν ήχο που βρίσκεται σε κλειστό χώρο, τότε γίνεται αμέσως αντιληπτός από κάποιον που βρίσκεται στον ίδιο χώρο. Αν αμέσως μετά ακουστεί το ηχητικό σήμα που έχει ανακλαστεί και ο ακροατής το αντιληφθεί σαν ξεχωριστό ηχητικό γεγονός, το φαινόμενο αυτό ονομάζεται ηχώ. Η ηχώ και η αντήχηση είναι εντελώς διαφορετικά φαινόμενα, επειδή στην αντήχηση δεν διακρίνεται κάποιο ξεχωριστό ηχητικό

γεγονός. Η ηχώ σε αντίθεση με την αντήχηση δεν είναι επιθυμητή, επειδή καταστρέφει την καλή ακουστική των χώρων αφού δυσκολεύει την επικοινωνία και αλλοιώνει τον ήχο.

Το 1950, ο Haas έδειξε με πειράματα ότι, αν η ανάκλαση καθυστερεί από 5 έως 35 msec, για να είναι διακριτή ως ηχώ, θα πρέπει η στάθμη της να είναι γύρω στα 10 dB μεγαλύτερη από τον ήχο που την δημιουργεί. Έτσι, αν η καθυστέρηση είναι μεγαλύτερη των 100 msec, τότε το φαινόμενο υπάρχει ανεξάρτητα από την στάθμη του ήχου που έχει καθυστερήσει. Αν η στάθμη της καθυστερημένης ηχητικής ενέργειας δεν είναι τόσο μεγάλη, απλώς ενισχύεται ο αρχικός ήχος.

2.4.2 Χρωματισμός (Coloration)

Χρωματισμός ονομάζεται το φαινόμενο κατά το οποίο αλλάζει η χροιά των ήχων όταν συμβάλλουν οι φράσεις και η τυχών καθυστέρηση των ανακλώμενων ακτινών είναι μικρή (δηλαδή της τάξης των 10 msec).

2.4.3 Ζεστασιά (Warmth)

Με τον όρο ηχητική «ζεστασιά» αναφερόμαστε στην τιμή του χρόνου αντήχησης που έχουν οι υψηλές και μεσαίες συχνότητες, δηλαδή όταν σε έναν χώρο υπάρχει μουσική σε χαμηλές συχνότητες, τότε ο χρόνος αντήχησης θα είναι μεγάλος, ενώ στις υψηλές συχνότητες θα είναι μικρός.

2.4.4 Οικειότητα (Intimacy)

Μια σημαντική ακουστική παράμετρος για χώρους είναι η οικειότητα η οποία προορίζεται για ομιλία και μουσική. Σε μια μικρή αίθουσα παίζεται μουσική, η οποία ακούγεται σαν να παίζεται σε μικρή αίθουσα, τότε επιτυγχάνεται η οικειότητα. Αυτό οφείλεται στο ότι οι καθυστερήσεις των ανακλώμενων ηχητικών ακτινών και τον απευθείας εκπεμπόμενων καθορίζουν την τιμή της οικειότητας. Όταν οι ανακλάσεις που φθάνουν στους ακροατές είναι μικρής καθυστέρησης (20 msec), τότε δεν διαφέρουν πολύ από τον απευθείας ήχο και δίνουν την αίσθηση της ακουστικής οικειότητας. Η οικειότητα των χώρων μουσικής πρέπει να είναι μικρότερη από 25 msec.

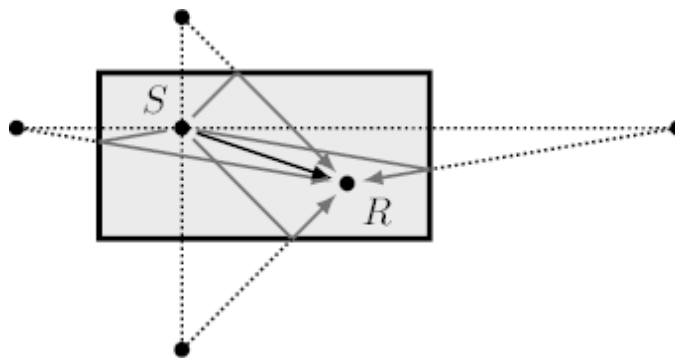
2.5 Μοντέλο των ειδώλων πηγής (Image Source Model)

Το μοντέλο των ειδώλων πηγής είναι μια τεχνική που χρησιμοποιείται για την ακουστική των κλειστών χώρων και εξετάζει την ακουστική χώρων με την μέθοδο της γεωμετρικής ακουστικής. Η μέθοδος αυτή είναι απλή στη χρήση, όμως το μεγαλύτερο της μειονέκτημα είναι η εκθετική αύξηση του υπολογιστικού χρόνου σε αναλογία με την τάξη των ανακλάσεων και

τον αριθμό των επιφανειών μιας αίθουσας. Βέβαια, ο υπολογιστικός χώρος αυξάνεται περισσότερο όταν η ακουστική μοντελοποίηση συμβαίνει σε πολλαπλές πηγές ήχου.

Σε μία αίθουσα έχουμε μια πηγή (S) και ένα δέκτη (R). Στο μοντέλο ειδώλων πηγής, η πηγή ήχου αντανακλάται στις επιφάνειες του δωματίου και έτσι δημιουργούνται οι εικονικές πηγές. Η ευθεία γραμμή από μια εικονική πηγή προς το δέκτη αντιστοιχεί σε μια διαδρομή διάδοσης του ήχου στην αίθουσα η οποία περιλαμβάνει μια ή περισσότερες επιφανειακές αναλύσεις.

Εάν γνωρίζουμε το μήκος της γραμμής και την απορρόφηση των επιφανειών που τέμνει η γραμμή, μπορούμε να υπολογίσουμε τη συμβολή της αντίστοιχης διαδρομής διάδοσης του ήχου στην απόκριση του παλμικού χώρου. Το μοντέλο ειδώλων πηγής εγγυάται ότι θα εντοπίσει όλες τις διαδρομές διάδοσης σε μία αίθουσα, βέβαια αν έχουμε αναφορές υψηλών τάξεων τότε το μοντέλο είναι ανεπαρκής επειδή ο αριθμός των εικονικών πηγών αυξάνεται γρήγορα.



Σχήμα 2.5 Απεικόνιση Μοντέλου Image Source

Κεφάλαιο 3

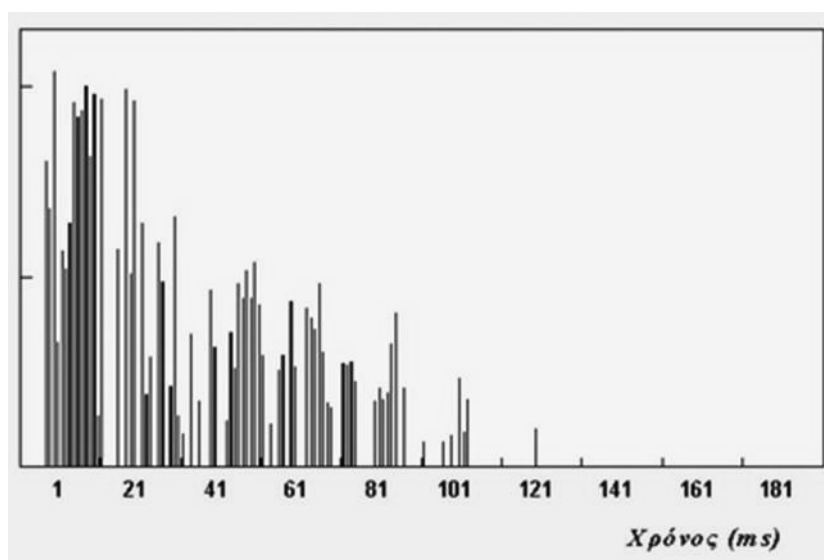
3.1 Κρουστική Απόκριση (Impulse Response)

Η κρουστική απόκριση ενός συστήματος ονομάζεται η απόκριση εξόδου του συστήματος όταν στην είσοδό του εφαρμοστεί ένας παλμός με άπειρη διάρκεια και μοναδιαίο ύψος. Η κρουστική διέγερση μοντελοποιείται θεωρητικά με τη συνάρτηση $\delta(t)$ (Dirac) στο πεδίο του συνεχούς χρόνου. Γνωρίζοντας την κρουστική απόκριση ενός γραμμικού και χρονικά αμετάβλητου (ΓΧΑ) συστήματος μπορούμε να υπολογίσουμε την απόκρισή του σε οποιοδήποτε σήμα εισόδου. Η κρουστική απόκριση υπολογίζεται μέσω της συνέλιξης της με την απόκρισή ενός χώρου:

$$h(t) = g(t) * x(t) \quad (3.1)$$

Όπου:

1. $g(t)$: η κρουστική απόκριση
2. $x(t)$: η διέγερση που εφαρμόζεται
3. * συνέλιξη



Σχήμα 3.1 Κρουστική απόκριση δωματίου

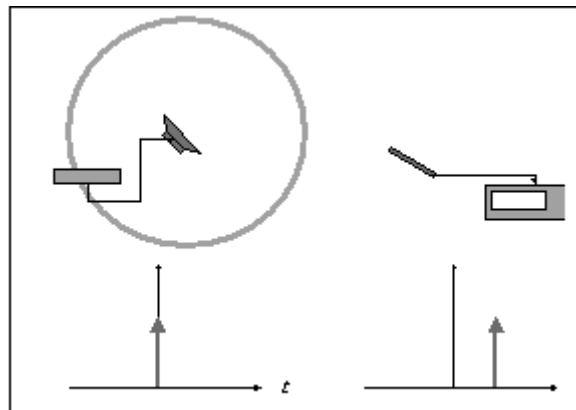
Στην παραπάνω εικόνα φαίνονται οι μετατοπισμένες συναρτήσεις δέλτα (Dirac) που αντιστοιχούν στις ανακλάσεις του ήχου από τα τοιχώματα του χώρου. Η μορφή του διαγράμματος εξαρτάται από τις διαστάσεις, το σχήμα του χώρου, την απορρόφηση του χώρου καθώς και από τη θέση πηγής – δέκτη.

Ανάλογα με την κρουστική απόκριση του χώρου υπολογίζονται όλοι οι παράμετροι με τους οποίους μετράμε την ακουστική ποιότητα ενός χώρου. Έτσι η ακριβής μέτρηση της κρουστικής απόκρισης παίζει σημαντικό ρόλο στον ακριβή υπολογισμό των παραμέτρων που μετρούν την ακουστική ποιότητα του χώρου, δηλαδή για να βελτιωθεί η κρουστική απόκριση χρειάζεται μεγάλος χρόνος σάρωσης, μεγάλος αριθμός δειγμάτων και ειδικής μορφής διεγέρσεις.

3.1.1 Κρουστική απόκριση σε ηχητικό σύστημα

A) Απόκριση σε ανοικτό χώρο

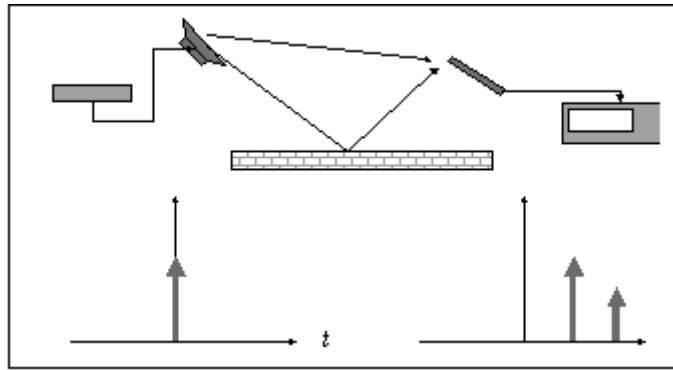
Σε ένα ελεύθερο χώρο υπάρχει μια ηχητική πηγή, για παράδειγμα ένα μεγάφωνο και εκπέμπει μια συνάρτηση δέλτα. Το μικρόφωνο θα καταγράψει επίσης μια συνάρτηση δέλτα μετατοπισμένη όμως ως προς τον χρόνο, βέβαια η μετατόπιση αυτή θα έχει διάρκεια τόση όσο θα κάνει ο ήχος να φθάσει από το μεγάφωνο στο μικρόφωνο.



Σχήμα 3.2 Απόκριση σε Ανοικτό Χώρο

B) Απόκριση επιφάνειας που ανακλά τον ήχο

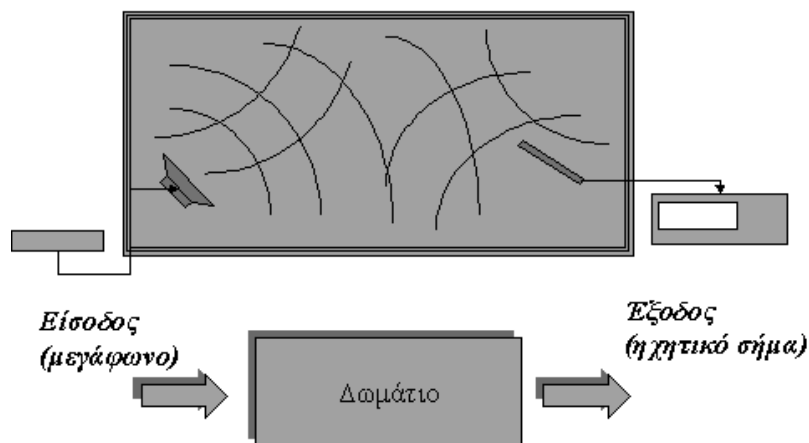
Έχουμε μια πηγή η οποία βρίσκεται πάνω από μια ανακλώμενη επιφάνεια, δηλαδή η απόκριση του μικροφώνου θα είναι τώρα δύο συναρτήσεις δέλτα μία που αντιστοιχεί στον απ' ευθείας ήχο και μία στον ανακλώμενο. Το εύρος της δεύτερης εξαρτάται από τα ακουστικά χαρακτηριστικά της επιφάνειας.



Σχήμα 3.3 Απόκριση επιφάνειας που ανακλά τον ήχο

C) Απόκριση κλειστού χώρου

Το σύστημα βρίσκεται σε κλειστό χώρο και το μεγάφωνο εκπέμπει μια κρουστική συνάρτηση. Το μικρόφωνο εκτός από τον απ' ευθείας διαδιδόμενο ήχο λαμβάνει και αυτόν που προέρχεται από τις διάφορες ανακλάσεις. Έτσι ο ήχος που λαμβάνεται δεν είναι η κρουστική συνάρτηση αλλά μια σειρά μετατοπισμένων και εξασθενημένων συναρτήσεων δέλτα.



Σχήμα 3.4 Απόκριση κλειστού χώρου

3.2 Μέτρηση Κρουστικής Απόκρισης με τη μέθοδο MLS

Για να έχουμε μια ακριβής μέτρηση της κρουστικής απόκρισης θα χρειαστούμε ένα καλώς ορισμένο σήμα διέγερσης. Η ISO (Διεθνής Οργανισμός Τυποποίησης) πρότεινε την χρήση της ψευδοτυχαίας ακολουθίας μεγίστου μήκους (Maximum Length Sequence). Η χρήση της μεθόδου MLS προτάθηκε για πρώτη φορά από τον Schroeder το 1979. Εφαρμόζεται σε κάθε γραμμικό και χρονικά αμετάβλητο σύστημα, ακουστικό, μηχανικό ή ηλεκτρονικό και βασίζεται στην διέγερση του συστήματος με μία εξειδικευμένη κατηγορία σημάτων εισόδου που ονομάζονται ακολουθίες μεγίστου μήκους και αναπτύχθηκαν στα πλαίσια της θεωρίας αριθμών.

Μια ακολουθία μεγίστου μήκους αποτελείται από μία σειρά μοναδιαίων παλμών και παύσεων οι οποίες είναι μηδενικές αφού εναλλάσσονται κατά τέτοιο τρόπο έτσι ώστε σε κάθε χρονική στιγμή, η πιθανότητα να εμφανιστεί ένας θετικός παλμός ή να γίνει παύση να είναι ίση και ανεξάρτητη από το προηγούμενο δείγμα.

Τα βασικά χαρακτηριστικά της ακολουθίας μεγίστου μήκους είναι τα ακόλουθα:

- Παράγεται με αυστηρά καθορισμένο και ντετερμινιστικό τρόπο ,δηλαδή το αποτέλεσμα της ακολουθίας μεγίστου μήκους είναι βέβαιο.
- Ο αριθμός των μονάδων είναι ίσος με τον αριθμό των μηδενικών, βέβαια για μία μονάδα την αρχική δεν ισχύει αυτό.
- Η συνάρτηση της αυτοσυσχέτισης (auto correlation function) προσφέρει τη δυνατότητα δημιουργίας ενός τέλει θεωρητικού παλμού που αναπαριστά ακριβώς τη συνάρτηση δέλτα (Dirac).
- Ο μετασχηματισμός Fourier της ακολουθίας περιέχει σταθερή ενέργεια ανά συχνότητα όπως μια δέλτα συνάρτηση ή ο λευκός θόρυβος.
- Η ακολουθία μεγίστου μήκους είναι ένα περιοδικό σήμα που εξαρτάται από έναν ακέραιο N , ο οποίος ονομάζεται τάξη της ακολουθίας. Πιο συγκεκριμένα πρόκειται για μια δυαδική ακολουθία σταθερού πλάτους με περίοδο $L = 2^N - 1$, ενώ παρουσιάζει επίπεδη συχνοτική απόκριση.

Το βασικό μειονέκτημα της MLS είναι ότι όλες οι συχνότητες μαζί με την αντήχηση που προκαλούν, υπάρχουν και καταγράφονται μέσα στο χώρο την ίδια χρονική στιγμή. Αυτό δημιουργεί προβλήματα αφού τόσο οι αρμονικές παραμορφώσεις όσο και η απόκριση του συστήματος στο ηχογραφημένο σήμα έχουν την μορφή του θορύβου και είναι δύσκολο να διαχωριστούν . Η υπερευαισθησία της μεθόδου στη μη γραμμική συμπεριφορά του ηχητικού εξοπλισμού αλλά και στη χρονική μεταβλητότητα του συστήματος εξαιτίας της επανειλημμένης αναπαραγωγής του σήματος διέγερσης που απαιτείται, έχει ως αποτέλεσμα να ενσωματώνονται ενοχλητικοί θορύβοι κατά μήκος της υπό μέτρησης κρουστικής απόκρισης.

3.3 Μοντέλο All pole Filter

Στην επεξεργασία σήματος, ένα φίλτρο είναι μια συσκευή ή μια διαδικασία που μπορεί να αφαιρεί κάποια ανεπιθύμητα στοιχεία ή χαρακτηριστικά από ένα σήμα. Κάθε ψηφιακό φίλτρο μπορεί να προσδιοριστεί από τους πόλους του και τα μηδενικά του. Οι πόλοι και τα μηδενικά δίνουν χρήσιμες γνώσεις για την απόκριση ενός φίλτρου και μπορούν να χρησιμοποιηθούν ως βάση για το σχεδιασμό ενός ψηφιακού φίλτρου. Όλα τα φίλτρα αποτελούνται από δύο βασικά

στοιχεία: την συνάρτηση μεταφοράς (Transfer Function) και την κρουστική απόκριση (Impulse Response). Αυτά τα δύο στοιχεία συνδέονται μεταξύ τους με τον σχηματισμό Fourier.

Η συνάρτηση μεταφοράς χρησιμοποιείται στην επεξεργασία σήματος για να περιγράψει η συμπεριφορά ενός γραμμικού και αιτιατού συστήματος. Η συνάρτηση μεταφοράς ενός φίλτρου ορίζεται πιο συχνά στην περιοχή των σύνθετων συχνοτήτων. Η λειτουργία μεταφοράς $H(s)$ ενός φίλτρου είναι η αναλογία του σήματος εξόδου $Y(s)$ με εκείνη του σήματος εισόδου $X(s)$ ως συνάρτηση της σύνθεσης συχνότητας s :

$$H(s) = \frac{Y(s)}{X(s)} \quad (3.2)$$

Υπάρχουν δύο τρόποι με τους οποίους υλοποιείται ένα ψηφιακό φίλτρο:

- 1) Με την χρονική καθυστέρηση ενός αντιγράφου τους σήματος εισόδου και τον συνδυασμό του καθυστερημένου σήματος με το καινούργιο σήμα εισόδου. Στην περίπτωση αυτή μιλάμε για φίλτρα FIR (Finite Impulse Response).
- 2) Με τη καθυστέρηση ενός αντιγράφου του σήματος εξόδου και τον συνδυασμό του με το σήμα εισόδου, σε αυτήν την περίπτωση μιλάμε για φίλτρα IIR (Infinite Impulse Response).

Σύγκριση FIR και IIR φίλτρων:

Η εφαρμογή ενός φίλτρου σε ένα σήμα μεταβάλλει πέρα από το πλάτος του και τη φάση του. Η εφαρμογή των FIR φίλτρων είναι γενικά προτιμότερη σε ηχητικά σήματα αφού είναι δυνατό να σχεδιαστούν τέτοια φίλτρα με γραμμική απόκριση φάσης. Μια τέτοια απόκριση προλαμβάνει πιθανές παραμορφώσεις φάσης που εισάγει το φιλτράρισμα ενός ήχου.

Επιπλέον, τα FIR φίλτρα είναι περισσότερο σταθερά από ότι τα IIR μιας και δεν περιέχουν ανάδραση. Από την άλλη πλευρά, η υλοποίηση των FIR φίλτρων απαιτεί περισσότερες αριθμητικές πράξεις και μνήμη από ότι η υλοποίηση IIR φίλτρων με αντίστοιχα χαρακτηριστικά.

Κεφάλαιο 4

4.1 Περιγραφή του προβλήματος

Ένας κλειστός χώρος παρουσιάζεται ως ένα γραμμικό και χρονικά αναλλοίωτο (ΓΧΑ) σύστημα, όπου η κρουστική απόκριση του χώρου περιγράφει τα χαρακτηριστικά της μετάδοσης του απευθείας και του ήχου που έχει ανακλαστεί από μια πηγή σε ένα δέκτη. Η κατευθυντικότητα της πηγής και του δέκτη ή οι μικρές αλλαγές των συνθηκών που έχει το περιβάλλον δεν χαρακτηρίζουν την κρουστική απόκριση που αυτή εξαρτάται μόνο από τις θέσεις της πηγής και του δέκτη. Όταν μια πηγή εκπέμπει ένα παλμό σε ένα κλειστό χώρο τότε η κρουστική απόκριση που θα καταγραφεί σε ένα δέκτη, θα είναι η εξής: καταρχάς θα καταγραφεί παλμός που θα φτάνει στο δέκτη απευθείας από την πηγή, μετά από λίγα δευτερόλεπτα θα φανερωθούν στον δέκτη οι πρώτες διακριτές ανακλάσεις που υπάρχουν στις επιφάνειες του κλειστού χώρου. Αυτές οι ανακλάσεις που έχουν πλάτος και ακριβείς χρόνους αφίξεων, εξαρτώνται από τις θέσεις της πηγής και του δέκτη. Ο ήχος που θα φτάσει στον δέκτη αποτελείται από ανακλάσεις που επικαλύπτονται χρονικά, δηλαδή θα έχουμε το φαινόμενο της αντήχησης.

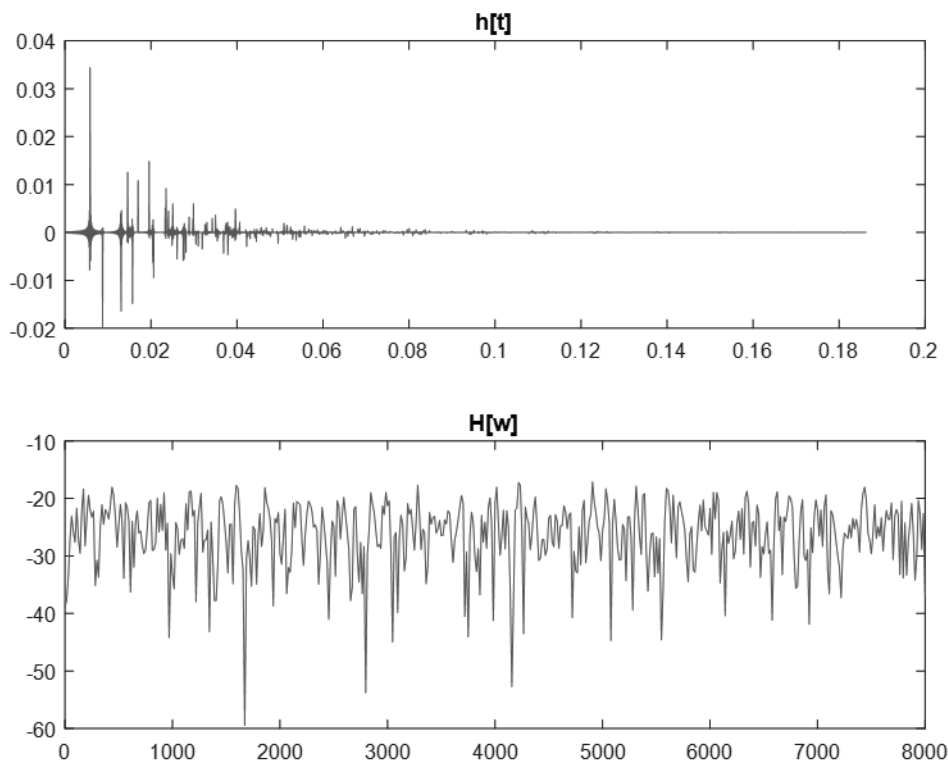
Κρουστική Απόκριση Δωματίου

Σε αυτό το κεφάλαιο θα περιγράψει και θα υπολογιστεί η κρουστική απόκριση ενός κλειστού δωματίου. Αυτό θα γίνει με 2 προσομοιωτές τον Room Impulse Response και τον All Pole Filter. Η υλοποίηση βασίζεται στο λογισμικό MATLAB και τις επιπλέον βιβλιοθήκες που διαθέτει.

4.2 Προσομοίωση ακουστικού χώρου και δημιουργία κρουστικής απόκρισης

4.2.1 Με τον Room Response

Με τον προσομοιωτή Room Response παίρνοντας τις διαστάσεις ενός δωματίου, την τοποθεσία του αισθητήρα και της πηγής, τους συντελεστές απορρόφησης του τοίχου, την συχνότητα δειγματοληψίας και τον χρόνο αντήχησης, μπορούμε να υπολογίσουμε την κρουστική απόκριση του δωματίου. Παρακάτω παραθέτονται 2 γραφήματα τα οποία παρουσιάζουν την κρουστική απόκριση του δωματίου συναρτήση του χρόνου $h(t)$ και το φάσμα της κρουστικής απόκρισης από τον Μετασχηματισμό Fourier $H(w)$ (σε κλίμακα dB).

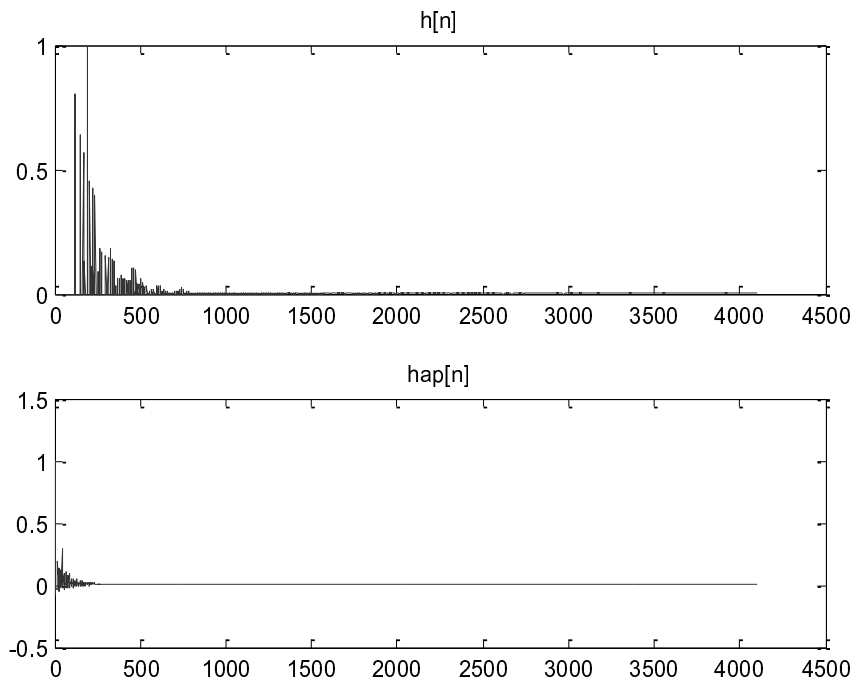


Σχήμα 4.1 Κρουστική απόκριση δωματίου συναρτήση χρόνου και φάσματος

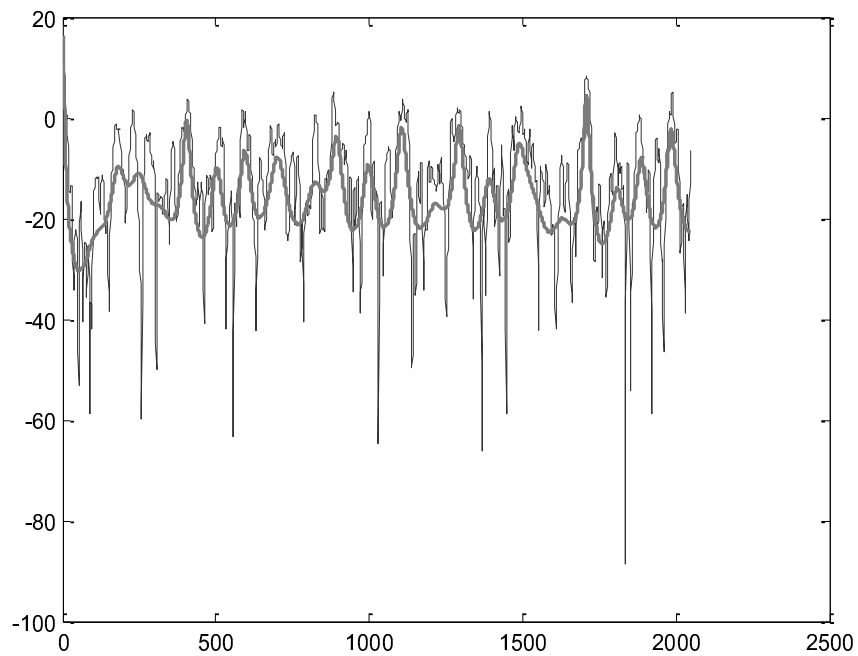
4.2.2 Με τον All Pole Filter

Με τον προσομοιωτή All Pole Filter παίρνοντας τις διαστάσεις ενός δωματίου, την τοποθεσία του αισθητήρα και της πηγής, τους συντελεστές απορρόφησης του τοίχου, την συχνότητα δειγματοληψίας και τον χρόνο αντήχησης, μπορούμε να υπολογίσουμε την κρουστική απόκριση του δωματίου. Στο (1) παρουσιάζεται η κρουστική απόκριση με τη χρήση του Διακριτού Μετασχηματισμού Fourier $h[n]$ και $h_{ap}[n]$ είναι οι αποκρίσεις με τον Αντίστροφο Μετασχηματισμό Fourier. Στο (2) παρατίθενται δύο προσεγγίσεις: ενός πραγματικού φάσματος (μπλέ διάγραμμα) και ενός προσεγγιστικού φάσματος «μόνο πόλων» (μώβ διάγραμμα), που έχει προκύψει με την μέθοδο γραμμικής πρόγνωσης (linear prediction) χρησιμοποιώντας τον αλγόριθμο του Levinson-Durbin. Οι υπολογισμοί αυτοί είναι σε λογαριθμική κλίμακα. Τέλος, στο (3) παρουσιάζονται οι αποδόσεις ανάλογα με την τάξη και την τιμή του σφάλματος (Error) για την πρώτη προσέγγιση και για τα κριτήρια πληροφόρισης (information criterion) για την δεύτερη προσέγγιση. Όσο πιο μικρή είναι η αναλογία αυτή (όσο αφορά το σφάλμα και το κριτήριο πληροφόρισης) τόσο πιο καλή θα είναι και η προσέγγιση του αντίστοιχου μοντέλου.

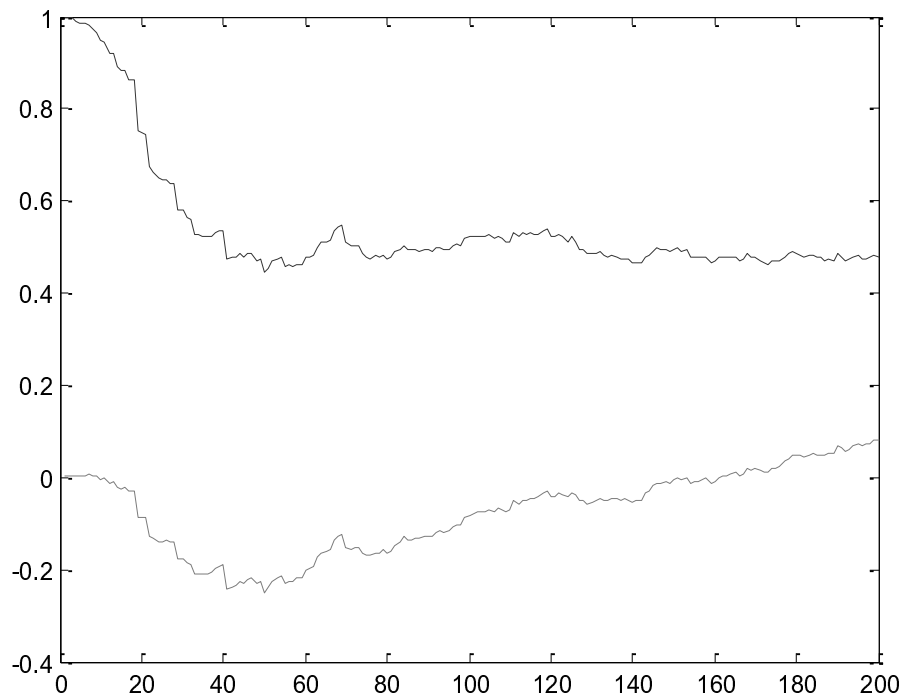
Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης



(1)



(2)



(3)

Σχήμα 4.2 Διαγράμματα αποτελεσμάτων του προσομοιωτή All Pole Filter

4.2.3 Μέθοδος γραμμικής πρόγνωσης

Η Γραμμική Πρόγνωση (linear prediction) είναι μια απλή μέθοδος η οποία προβλέπει τις μελλοντικές τιμές με βάση έναν γραμμικό συνδυασμό των τιμών που υπάρχουν ήδη. Μπορούμε να υπολογίσουμε την γραμμική πρόγνωση με τον παρακάτω τύπο:

$$\tilde{x}(n) = \sum_{k=1}^p a_k * x(n - k) \quad (4.1)$$

Όπου,

$\tilde{x}(n)$: η προβλεπόμενη τιμή σήματος

a_k : ο συντελεστής γραμμικής πρόγνωσης

k : οι προηγούμενες τιμές

Υπολογίζοντας τον συντελεστή γραμμικής πρόγνωσης a_k , μπορώ να ξαναφτάσω στο αρχικό μου σήμα (speech synthesis). Επίσης υπολογίζοντας τον συντελεστή a_k μπορούμε να υπολογίσουμε και το σφάλμα πρόγνωσης. Το σφάλμα πρόγνωσης δίνεται από τον τύπο:

$$e(n) = x(n) - \tilde{x}(n) \quad (4.2)$$

όπου,

$x(n)$: η πραγματική τιμή σήματος

Το σφάλμα πρόγνωσης θα πρέπει να είναι μικρότερο από το αρχικό σήμα. Αυτές τις εξισώσεις μπορούν να χρησιμοποιηθούν όταν έχουμε μονοδιάστατα σήματα. Για πολυδιάστατα σήματα η μέτρηση σφάλματος ορίζεται συχνά ως:

$$e(n) = \| x(n) - \tilde{x}(n) \| \quad (4.3)$$

όπου,

$\| \cdot \|$ ο κατάλληλος κανόνας φορέα που επιλέγεται.

Ο κανόνας φορέα είναι μια συνάρτηση που αποδίδει ένα αυστηρά θετικό μήκος ή μέγεθος σε κάθε διάνυσμα σε ένα διανυσματικό χώρο.

Οι διαφορές των μονοδιάστατων από τα πολυδιάστατα σήματα εντοπίζονται στον τρόπο με τον οποίο επιλέγονται οι συντελεστές πρόβλεψης a_i . Για την βελτιστοποίηση αυτών των παραμέτρων γίνεται χρήση ενός κριτηρίου μέσης τετραγωνικής ρίζας που λέγεται και κριτήριο αυτοσυσχέτισης. Με αυτή τη μέθοδο ελαχιστοποιείτε η αναμενόμενη τιμή του τετραγωνικού σφάλματος $E[e^2(n)]$. Η τιμή αυτή δίνεται από την εξίσωση :

$$\sum_{k=1}^p a_k R(i-k) = R(i) \quad (4.4)$$

όπου ,

όπου R είναι η αυτοσυσχέτιση του σήματος $x(n)$.

Η αυτοσυσχέτιση R ορίζεται ως εξής :

$$R(i) = E\{x(n)x(n-k)\} \quad (4.5)$$

όπου E η αναμενόμενη τιμή.

Οι εξισώσεις αυτές ονομάζονται κανονικές εξισώσεις ή εξισώσεις Yule-Walker.

Όμως αυτό το κριτήριο αυτοσυσχέτισης δεν μπόρεσε να εκτιμήσει γρήγορα τις παραμέτρους της γραμμικής πρόγνωσης. Αργότερα όμως δόθηκε η λύση με τη χρήση ενός ταχύτερου αλγορίθμου του Levinson. Η επανάληψη του Levinson είναι ένας αλγόριθμος για την εξεύρεση ενός all-pole IIR φίλτρου με προκαθορισμένη ντετερμινιστική αλληλουχία αυτοσυσχέτισης. Το φίλτρο που παράγει ο Levinson είναι η ελάχιστη φάση.

Κεφάλαιο 5

5.1 Εισαγωγή στην Μηχανική Μάθηση

Η Μάθηση είναι μία νοητική διεργασία κατά την οποία ένα άτομο αποκτά νέες δεξιότητες και γνώσεις. Οι επιστήμονες του χώρου της τεχνικής νοημοσύνης προσπάθησαν να δημιουργήσουν υπολογιστικά συστήματα που θα μπορούσαν να μάθουν και να βελτιώνουν την απόδοσή τους σε κάποιους τομείς μέσω της αξιοποίησης της προηγούμενης γνώσης και εμπειρίας που είχαν. Αυτή είναι η λεγόμενη Μηχανική Μάθηση.

Οι εργασίες της μηχανικής μάθησης συνήθως ταξινομούνται σε τρεις μεγάλες κατηγορίες που χρησιμοποιούνται ανάλογα με τη φύση του προβλήματος.

- Μάθηση με επίβλεψη (Supervised Learning): Είναι μια διαδικασία κατά την οποία το υπολογιστικό πρόγραμμα παίρνει κάποιες εισόδους και τα επιθυμητά αποτελέσματα από έναν «δάσκαλο» και ο στόχος του είναι να μπορέσει να αντιστοιχίσει αυτές τις εισόδους με τα αποτελέσματα που πήρε. Αυτή η κατηγορία μηχανικής μάθησης χρησιμοποιείται σε προβλήματα: Ταξινόμησης, Πρόγνωσης και Διερμηνείας.
- Μάθηση χωρίς επίβλεψη (UnSupervised Learning): Σε αυτή τη διαδικασία ο αλγόριθμος κατασκευάζει ένα μοντέλο για κάποιο σύνολο εισόδων με μορφή παρατηρήσεων χωρίς να γνωρίζει τις επιθυμητές εξόδους. Αυτή η κατηγορία χρησιμοποιείται σε προβλήματα: Ανάλυσης Συσχετισμών και Ομαδοποίησης.
- Ενισχυτική μάθηση (Reinforcement Learning): Το σύστημα μάθησης προσπαθεί να μάθει μέσω άμεσης αλληλεπίδρασης με το περιβάλλον. Η κατηγορία αυτή χρησιμοποιείται σε προβλήματα: Σχεδιασμού ,για παράδειγμα ο έλεγχος κίνησης των ρομπότ και η βελτιστοποίηση εργασιών σε εργοστασιακούς χώρους.

Για κάθε πρόβλημα προς επίλυση στο χώρο της Μηχανικής Μάθησης υπάρχει ένας κατάλληλος τρόπος μάθησης και για κάθε τρόπο μάθησης υπάρχει τουλάχιστον ένας κατάλληλος αλγόριθμος που μπορεί να χρησιμοποιηθεί.

5.2 Εξόρυξη Δεδομένων (Data Mining)

Η Μηχανική Μάθηση έχει σχέση με την Εξόρυξη Δεδομένων. Η Εξόρυξη των Δεδομένων προέκυψε για την παραγωγή νέας γνώσης η οποία προκύπτει από πληροφορίες, οι

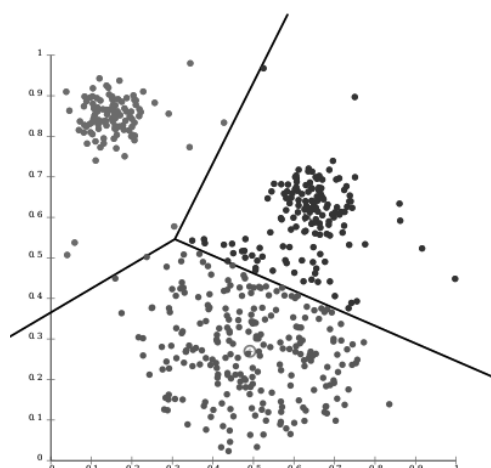
οποίες έχουν συσσωρευτεί κατά το παρελθόν και μπορούν να οδηγήσουν στην βελτίωση της υπάρχουσας κατάστασης. Επίσης, σχετίζεται άμεσα με την απόσπαση των πληροφοριών από μεγάλες βάσεις δεδομένων, οι οποίες συνδυάζονται μέσω της στατιστικής με σκοπό την εξαγωγή ασφαλών συμπερασμάτων.

5.3 Αλγόριθμοι Εξόρυξης δεδομένων

5.3.1 Αλγόριθμος K-Means

Ο αλγόριθμος K-Means είναι ένας αλγόριθμος που επαναλαμβάνεται και ομαδοποιεί αντικείμενα βάσει των χαρακτηριστικών των k-συστάδων. Ο αλγόριθμος ξεκινά καθορίζοντας με τυχαίο τρόπο k-συστάδες, στη συνέχεια προσδιορίζεται η απόσταση κάθε αντικειμένου από το κέντρο κάθε συστάδας και κάθε αντικείμενο τοποθετείται στη συστάδα από την οποία απέχει λιγότερο. Τα καινούργια κέντρα υπολογίζονται σαν ο μέσος όρος των αντικειμένων που ανήκουν μέχρι στιγμής σε κάθε συστάδα. Η διαδικασία επαναλαμβάνεται μέχρις ότου οι συστάδες να σταματήσουν να μεταβάλλονται. Τα μειονεκτήματα του αλγορίθμου K-Means είναι ότι η απόδοση του αλγορίθμου δεν είναι βέλτιστη. Η ποιότητα της τελικής λύσης εξαρτάται πολύ από το αρχικό σύνολο ομάδων και μπορεί να είναι χαμηλότερη από το συνολικό βέλτιστο.

Επίσης ένα άλλο μειονέκτημα του αλγορίθμου είναι ότι ο αριθμός των ομάδων πρέπει να οριστεί εξ αρχής. Στην παρακάτω εικόνα παρουσιάζεται ένα παράδειγμα ομαδοποίησης του αλγορίθμου K-Means.

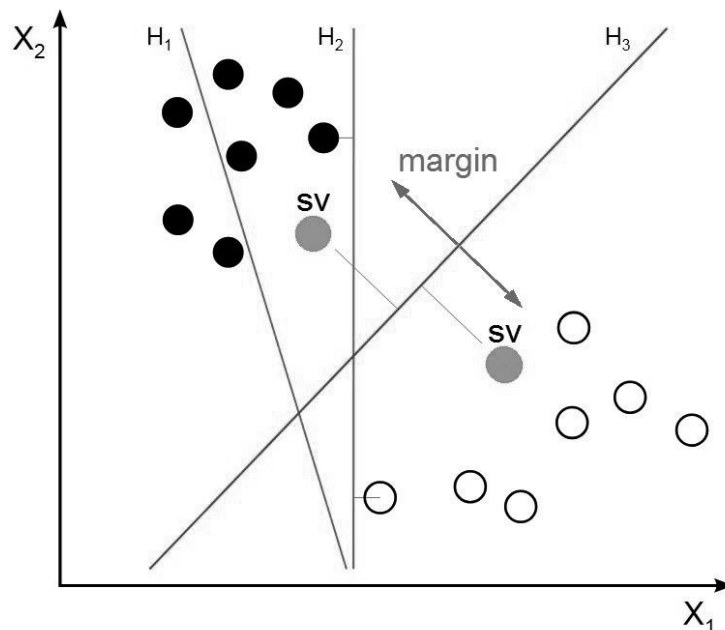


Σχήμα 5.1 Παράδειγμα Ομαδοποίησης του αλγορίθμου K-Means

5.3.2 Αλγόριθμος SVM (Support Vector Machine)

Ο αλγόριθμος Support Vector Machine είναι ένα σύνολο μεθόδων μάθησης με επίβλεψη (supervised learning) και χρησιμοποιείται ευρύτατα σε προβλήματα ταξινόμησης και παλινδρόμησης παρουσιάζοντας υψηλή απόδοση σε χώρους με δεδομένα υψηλής διάστασης. Οι ταξινομητές SVMs μπορεί να χρησιμοποιηθούν για γραμμικά και μη γραμμικά δεδομένα. Στόχος ενός ταξινομητή βασισμένου σε διανύσματα υποστήριξης είναι να «κατασκευάσει» το βέλτιστο διαχωριστικό υπερεπίπεδο σε ένα πολυδιάστατο χώρο χαρακτηριστικών (feature space). Με τον όρο «βέλτιστο» αναφερόμαστε στην περίπτωση που υπάρχουν γραμμικές διαχωρίσιμες κατηγορίες όπου το υπερεπίπεδο που θα κατασκευαστεί να είναι αυτό με το μέγιστο περιθώριο (ή απόσταση) από τα πιο κοντινά παραδείγματα στον πολυδιάστατο χώρο των χαρακτηριστικών. Η μέθοδος SVM βρίσκει αυτό το υπερεπίπεδο χρησιμοποιώντας:

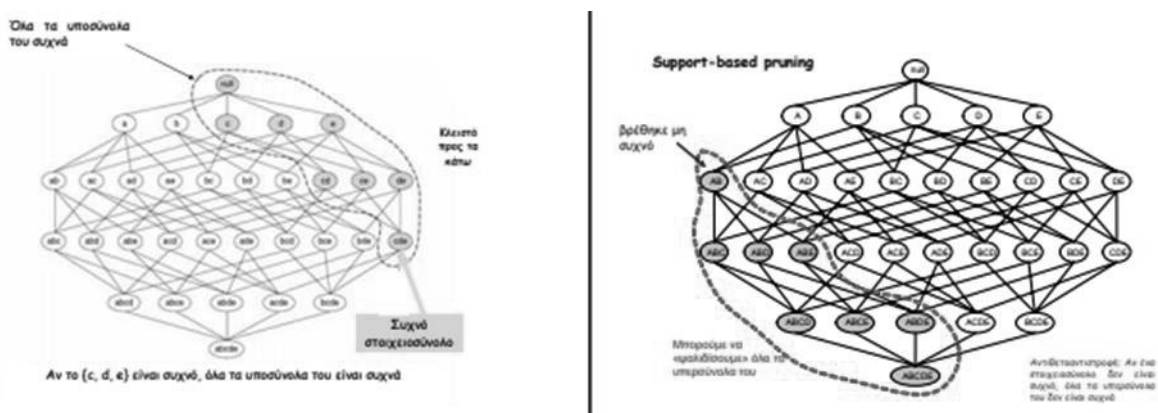
- Διανύσματα υποστήριξης (support vectors), τα οποία αποτελούν τα σημαντικότερα στιγμιότυπα εκπαίδευσης.
- Όρια – margins, που καθορίζονται από τα διανύσματα υποστήριξης.



Σχήμα 5.2 Παράδειγμα Αλγορίθμου SVM

5.3.3 Αλγόριθμος APRIORI

Ο αλγόριθμος APRIORI χρησιμοποιείται για την συχνή στοιχειοδοτημένη εξόρυξη και την συσχέτιση των στοιχείων που βασίζονται στις συναλλαγές των βάσεων δεδομένων. Αρχικά, γίνεται το πέρασμα (διάβασμα) του πίνακα που θέλουμε, στη συνέχεια εφαρμόζεται η ελάχιστη υποστήριξη έτσι ώστε να βρεθούν όλα τα συχνά στοιχεία που υπάρχουν μέσα σε μία βάση δεδομένων. Αυτά τα στοιχεία επιπλέον επεκτείνονται σε μεγαλύτερα σύνολα με την προϋπόθεση όμως ότι εμφανίζονται πολύ συχνά στις βάσεις δεδομένων. Ο APRIORI χρησιμοποιεί μια προσέγγιση “από κάτω προς τα πάνω”, όπου τα συχνά υποσύνολα επεκτείνονται σε ένα στοιχείο κάθε φορά (υποψήφιο στοιχείο) και οι ομάδες των υποψηφίων δοκιμάζονται ενάντια στα δεδομένα. Ο αλγόριθμος τερματίζεται όταν δεν βρεθούν άλλες επιτυχείς επεκτάσεις. Το πλεονέκτημα αυτού του αλγορίθμου είναι ότι σε κάθε φάση γίνεται ένα μόνο πέρασμα από τον αρχικό πίνακα και επιπλέον ο αλγόριθμος αυτός υλοποιείται πολύ εύκολα. Είναι όμως ένας αναποτελεσματικός αλγόριθμος αφού απαιτεί πολλά σαρώματα στην βάση δεδομένων και για το λόγο αυτό έχουν δημιουργηθεί και άλλοι αλγόριθμοι.



Σχήμα 5.3 Παράδειγμα Αλγορίθμου APRIORI

5.3.4 Αλγόριθμος AdaBoost

Ο AdaBoost (ή Adaptive Boosting) είναι ένας αλγόριθμος ενίσχυσης ο οποίος εστιάζει στα προβλήματα ταξινόμησης και στοχεύει στη μετατροπή ενός συνόλου από αδύναμους ταξινομητές σε έναν ισχυρό. Με τον αλγόριθμο αυτό παρατηρείται βελτίωση της απόδοσης ενός ταξινομητή και επιπλέον η ενίσχυση που γίνεται είναι μία από τις πιο σημαντικές τεχνικές μηχανικής μάθησης. Ο Adaboost αναφέρεται σε μια συγκεκριμένη μέθοδο εκπαίδευσης ενός ενισχυμένου ταξινομητή. Κάθε ασθενής ταξινομητής εκπαιδεύεται χρησιμοποιώντας ένα

τυχαίο υποσύνολο του συνόλου εκπαίδευσης. Μετά την εκπαίδευση ενός ταξινομητή σε οποιοδήποτε επίπεδο, ο AdaBoost αποδίδει βάρος σε κάθε εκπαιδευτικό αντικείμενο.

Το λανθασμένο στοιχείο έχει υψηλότερο βάρος, έτσι ώστε να εμφανίζεται στο υποσύνολο εκπαίδευσης του επόμενου ταξινομητή με υψηλότερη πιθανότητα. Αφού εκπαιδευτεί ο κάθε ταξινομητής, το βάρος αποδίδεται στον ταξινομητή και με βάση την ακρίβεια. Για τον ακριβέστερο ταξινομητή αποδίδεται μεγαλύτερο βάρος έτσι ώστε να έχει μεγαλύτερο αντίκτυπο στο τελικό αποτέλεσμα. Σε έναν ταξινομητή το βάρος αντιστοιχίζεται ανάλογα με την ακρίβεια που διαθέτει ο καθένας, δηλαδή όσο πιο ακριβής είναι ο ταξινομητής τόσο μεγαλύτερο είναι το βάρος. Για παράδειγμα σε έναν ταξινομητή με ακρίβεια 50% δίνεται ένα μηδενικό βάρος ενώ σε έναν ταξινομητή με ακρίβεια μικρότερη από 50% δίνεται αρνητικό βάρος. Αυτό σημαίνει ότι συνδυάζουμε την πρόβλεψή του με την ανατροπή του σημείου. Έτσι, ακόμη και αν ο ταξινομητής θέλει να εκτελέσει μια τυχαία μαντεψιά εξακολουθεί να συμβάλλει στην τελική πρόβλεψη.

Ο AdaBoost όπως και ο ταξινομητής Random Forest δίνει πιο ακριβή αποτελέσματα από τους άλλους αλγορίθμους μηχανικής μάθησης αφού εξαρτάται από πολλούς αδύναμους ταξινομητές για να βγει η τελική απόφαση.

Πλεονεκτήματα Adaboost

- Εύκολος, απλός και γρήγορος προγραμματισμός
- Δεν απαιτείται εκ των προτέρων γνώση του αδύναμου ταξινομητή άρα μπορεί να εφαρμοστεί με οποιαδήποτε μέθοδο αδύναμης ταξινόμησης
- Εντοπίζει ακραίες τιμές αφού επικεντρώνεται σε παρατηρήσεις οι οποίες ταξινομούνται δυσκολότερα. Άρα οι παρατηρήσεις με το υψηλότερο βάρος συχνά αποδεικνύεται ότι είναι ακραία σημεία.

Μειονέκτημα AdaBoost

Λόγω του ότι ο AdaBoost δίνει τόση πολλή “προσοχή” στα σημεία που ταξινομούνται λάθος δεν είναι καθόλου ακριβής σε δεδομένα με θόρυβο.

Αυτός ο αλγόριθμος εφαρμόζεται σε συστήματα αναγνώρισης προσώπων.

5.3.5 Αλγόριθμος NaiveBayes

Ο αλγόριθμος NaiveBayes είναι μια τεχνική ταξινόμησης η οποία βασίζεται στο θεώρημα του Bayes και επιπλέον θεωρεί πως υπάρχει ανεξαρτησία μεταξύ των ανεξάρτητων μεταβλητών

που υπάρχουν στον ταξινομητή. Ένα NaiveBayes μοντέλο δημιουργείται εύκολα αφού δεν χρειάζεται περίπλοκη ανάθεση τιμών στις παραμέτρους οι οποίες είναι ιδιαίτερα χρήσιμες όταν έχουμε πολύ μεγάλα σύνολα δεδομένων. Επιπλέον, σε αυτόν τον ταξινομητή όταν υπάρχει ένα συγκεκριμένο χαρακτηριστικό σε μια κλάση σίγουρα δεν προϋποθέτει και την παρουσία ενός άλλου χαρακτηριστικού. Το θεώρημα Bayes έχει έναν τρόπο να υπολογίζει την μεταγενέστερη πιθανότητα $P(c|x)$, από τις πιθανότητες $P(c)$, $P(x)$ και $P(x|c)$:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \quad (5.1)$$

Όπου,

1. $P(c|x)$ είναι η μεταγενέστερη πιθανότητα της τάξης (στόχου) c , όταν υπάρχει το χαρακτηριστικό x .
2. $P(c)$ είναι η προηγούμενη πιθανότητα της κλάσης c .
3. $P(x|c)$ είναι η πιθανότητα που είναι η πιθανότητα να ανήκει το χαρακτηριστικό x στην κλάση c εφόσον υπάρχει η κλάση c .
4. $P(x)$ είναι η προηγούμενη πιθανότητα του χαρακτηριστικού x .

Ο αλγόριθμος αυτός υπολογίζει την μεταγενέστερη πιθανότητα για κάθε κλάση και η κλάση που θα έχει την μεγαλύτερη μεταγενέστερη πιθανότητα θα διεκδικήσει το χαρακτηριστικό.

Πλεονεκτήματα NaiveBayes

Ο ταξινομητής Naive Bayes είναι ένας απλός αλγόριθμος που λειτουργεί εκπληκτικά και χρησιμοποιείται ευρέως επειδή συχνά αποδίδει πολύ καλά τις πιο εξελιγμένες μεθόδους ταξινόμησης. Ως πλεονεκτήματα του μπορεί να θεωρηθούν η εύκολη και γρήγορη πρόβλεψη των κατηγοριών των δεδομένων, οι οποίες πολύ πιθανόν να είναι πολλαπλές. Ο NaiveBayes λειτουργεί καλύτερα από τους άλλους αλγορίθμους όταν υπάρχει ανεξαρτησία μεταξύ των χαρακτηριστικών. Επιπλέον σε σχέση με τους άλλους αλγορίθμους χρειάζεται λιγότερα δεδομένα εκπαίδευσης. Τέλος, η απόδοσή του είναι καλή όταν παρουσιάζονται κατηγορικές μεταβλητές. Όταν υπάρχουν αριθμητικές μεταβλητές γίνεται παραδοχή για ύπαρξη κανονικής κατανομής.

Μειονεκτήματα NaiveBayes

Με τη χρήση του Naive Bayes εμφανίζεται σε ορισμένες περιπτώσεις κατηγορικών μεταβλητών το πρόβλημα της «μηδενικής συχνότητας – zero frequency». Το πρόβλημα αυτό θα εμφανιστεί στην περίπτωση που ένα χαρακτηριστικό δεν προκύπτει για μία τιμή της τάξης

με αποτέλεσμα να μηδενιστεί η πιθανότητα και δεν θα μπορεί να γίνει πρόβλεψη για αυτήν την κατηγορία. Για να λυθεί το πρόβλημα αυτό θα πρέπει να χρησιμοποιηθούν τεχνικές εξομάλυνσης όπως η εκτίμηση Laplace. Ουσιαστικά θα γίνει μία προσθήκη μονάδας στην απαρίθμηση κάθε συνδυασμού μιας τιμής χαρακτηριστικού-τάξης. Άλλο ένα μειονέκτημα αυτού του ταξινομητή είναι ότι η χρήση αυτών των ανεξάρτητων χαρακτηριστικών που θεωρεί ο Naive Bayes δεν συμβαίνει σε πραγματικά προβλήματα και αυτό έχει ως αποτέλεσμα να μην θεωρείται αξιόπιστος εκτιμητής.

Ο ταξινομητής NaiveBayes χρησιμοποιείται για προβλέψεις πραγματικού χρόνου, για πρόβλεψη πολλαπλών κλάσεων, όταν θέλουμε να κατηγοριοποιήσουμε ένα κείμενο, στην ανίχνευση spam ηλεκτρονικού ταχυδρομείου και στην ανάλυση συναισθήματος. Σε συνδυασμό με τεχνικές συνεργατικού φιλτραρίσματος (collaborative filtering) ο ταξινομητής NaiveBayes μπορεί να δημιουργήσει συστήματα συστάσεων (recommendation systems) για την εξόρυξη της πληροφορίας.

5.3.6 Αλγόριθμος CART

Ο αλγόριθμος Cart ή αλγόριθμος δέντρων κατηγοριοποίησης και παλινδρόμησης (classification and regression tree) είναι ένας αλγόριθμος ο οποίος δημιουργεί ένα δυαδικό δέντρο απόφασης από τα δεδομένα και περιγράφεται από δύο συνεχείς και διακριτές λειτουργίες. Όταν έχουμε συνεχή χαρακτηριστικά, όλες τις πιθανές δυαδικές διασπάσεις, τις διαχωρίζουμε σε διαστήματα $(-\infty, a]$ και $(a, +\infty)$, ενώ όταν έχουμε διακριτά χαρακτηριστικά, η ανάλυση αφορά όλες τις πιθανές διασπάσεις του συνόλου των συμβόλων σε δύο ασυνεχή και συμπληρωματικά υποσύνολα. Η μεθοδολογία CART αποτελείται από τρία μέρη: την κατασκευή του μέγιστου δένδρου, την επιλογή του σωστού μεγέθους δένδρου και την ταξινόμηση των δεδομένων με τη χρήση κατασκευασμένων δένδρων. Ένα δέντρο απόφασης χρησιμοποιεί την εντροπία έτσι ώστε να μπορεί να επιλέγει το καλύτερο γνώρισμα διάσπασης και το καλύτερο κριτήριο τερματισμού. Όταν βρεθεί το καλύτερο σημείο διάσπασης τότε γύρω από αυτό θα γίνει η σ και θα δημιουργηθούν δύο απόγονοι. Για κάθε καινούργιο βήμα του αλγορίθμου θα γίνεται μια πλήρης αναζήτηση έτσι ώστε να καθοριστεί η καλύτερη διάσπαση. Η καλύτερη διάσπαση δίνεται από τον τύπο:

$$\Phi (s/t) = 2 P_L P_R \sum_{j=1}^m |P(C_j|t_L) - P(C_j|t_R)| \quad (5.2)$$

Όπου,

1. t : ο τρέχων κόμβος.
2. s : κάθε πιθανό γνώρισμα διάσπασης (κριτήριο).

3. L : το αριστερό υποδένδρο του τρέχοντα κόμβου του δέντρου.
4. R : το δεξιό υποδένδρο του τρέχοντα κόμβου του δέντρου.
5. P_L : είναι η πιθανότητα μια πλειάδα του συνόλου εκπαίδευσης να ανήκει στην αριστερή πλευρά του δένδρου.
6. P_R : είναι η πιθανότητα μια πλειάδα του συνόλου εκπαίδευσης να ανήκει στην δεξιά πλευρά του δένδρου.

Βέβαια σε περίπτωση που έχουμε ισοβαθμία επιλέγεται η δεξιά διακλάδωση.

Όπου,

1. $P(C_j | t_L)$: η πιθανότητα μια πλειάδα να ανήκει στην κατηγορία C_j και στο αριστερό υποδένδρο.
2. $P(C_j | t_R)$: η πιθανότητα μια πλειάδα να ανήκει στην κατηγορία C_j και στο δεξιό υποδένδρο.

Σε κάθε βήμα μόνο ένα κριτήριο επιλέγεται σαν το καλύτερο από όλα τα πιθανά. Όταν δεν έχουμε καμία διάσπαση που να μπορεί να βελτιώσει την απόδοση του αλγορίθμου τότε σταματάει η ανάπτυξη του δέντρου. Όμως, αυτή η προσέγγιση μπορεί να μην είναι η καλύτερη δυνατή για τα δεδομένα που μπορεί να προστεθούν μελλοντικά.

ΠΛΕΟΝΕΚΤΗΜΑΤΑ & ΜΕΙΟΝΕΚΤΗΜΑΤΑ CART

Ο αλγόριθμος Cart είναι κατάλληλος για την πρόβλεψη των νέων περιπτώσεων. Επιπλέον, χαρακτηρίζεται για την ευρωστία του για ακραίες τιμές αφού ο αλγόριθμος διαχωρισμού θα απομονώσει τις ακραίες τιμές σε μεμονωμένο κόμβο ή κόμβους. Τα δέντρα αποφάσεων που χρησιμοποιεί ο αλγόριθμος είναι εύκολα να ερμηνευτούν όμως απαιτούν έναν αριθμό παραμέτρων που είναι δύσκολο να βρεθούν. Ένα από μειονεκτήματα αυτού του αλγορίθμου είναι η υπερβολική προσαρμογή των δεδομένων όμως προσπαθεί να περιορίσει το μέγεθος του προκύπτοντος δένδρου. Τέλος, ο CART διαχειρίζεται ελλιπή δεδομένα αγνοώντας την συγκεκριμένη εγγραφή, στον υπολογισμό για την ορθότητα της διάσπασης ενός γνωρίσματος.

5.3.7 Αλγόριθμος EM (Expectation Maximization)

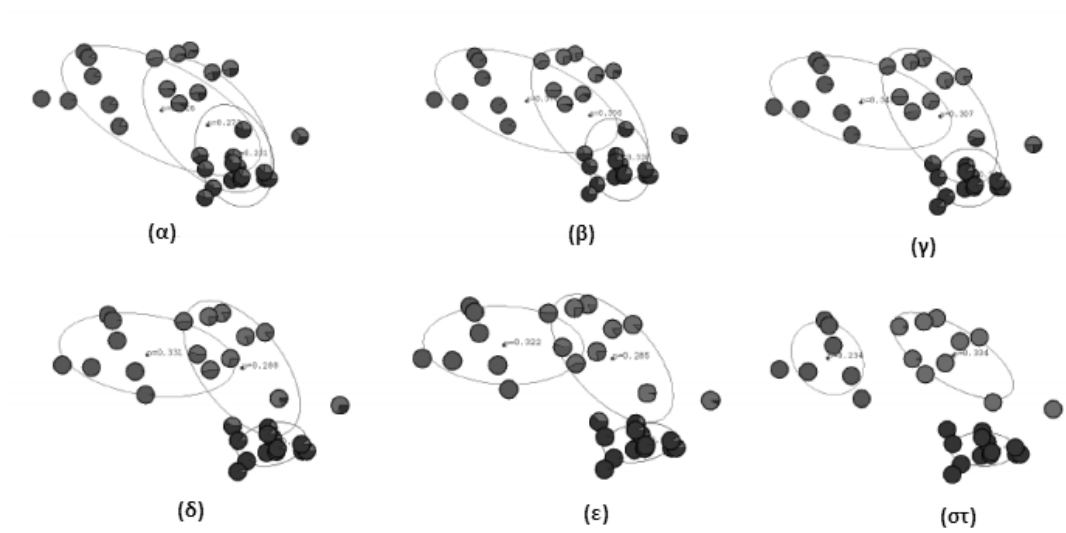
Ο αλγόριθμος Προσδοκίας – Μεγιστοποίησης (Expectation Maximization) είναι μια επαναληπτική μέθοδος για την εύρεση κάποιων τοπικών παραμέτρων μέγιστης πιθανότητας ενός στατιστικού μοντέλου όταν οι εξισώσεις δεν μπορούν να επιλυθούν σε μικρό χρονικό διάστημα. Αυτά τα μοντέλα περιλαμβάνουν εκτός από άγνωστες παραμέτρους και γνωστές

παρατηρήσεις δεδομένων, δηλαδή είτε υπάρχουν ελλειπείς τιμές μεταξύ των δεδομένων, είτε το μοντέλο διατυπώνεται πιο απλά με την χρήση κάποιων μη παρατηρούμενων σημείων δεδομένων. Αυτός ο αλγόριθμος βρίσκει τις παραμέτρους μέγιστης πιθανότητας χρησιμοποιώντας μια διαδικασία δύο βημάτων: την εκτίμηση και την μεγιστοποίηση. Ο αλγόριθμος EM χειρίζεται τα μη παρατηρούμενα σημεία με τον εξής τρόπο. Αρχικά αντικαθιστά τις ελλειπείς τιμές με τις κατ' εκτίμηση τιμές και εκτιμά τις παραμέτρους. Στη συνέχεια έχουμε μια εκ' νέου επανεκτίμηση των ελλειπουσών τιμών υποθέτοντας ότι οι νέες εκτιμήσεις των παραμέτρων είναι σωστές. Τέλος, επαναλαμβάνεται η προαναφερθείσα διαδικασία μέχρι να έχουμε σύγκλιση των διαδοχικών εκτιμήσεων των παραμέτρων.

Κατά το βήμα της εκτίμησης η αναμενόμενη τιμή του λογαρίθμου της μέγιστης πιθανότητας του πλήρους σετ δεδομένων προκύπτει, λαμβάνοντας υπόψη τα παρατηρηθέντα στοιχεία και τις κατ' εκτίμηση παραμέτρους από μία προηγούμενη επανάληψη. Κατά το βήμα της μεγιστοποίησης έχουμε μεγιστοποίηση της δεσμευμένης αναμενόμενης τιμής του λογαρίθμου της μέγιστης πιθανότητας του πλήρους σετ δεδομένων. Η τιμή αυτή μεγαλώνει μέχρι να φτάσουμε σε ένα στάσιμο σημείο.

Στην παρακάτω εικόνα παρατηρούμε τις επαναλήψεις που γίνονται κατά τον αλγόριθμο EM. Στο σχήμα (α) φαίνονται τα αρχικά μοντέλα και σε ποιο μοντέλο ανήκει το κάθε δεδομένο. Από το σχήμα φαίνεται πως έχουμε 3 ομάδες (πράσινη, κόκκινη, μπλέ), το κάθε δεδομένο είναι χρωματισμένο ανάλογα με την πιθανότητα του να ανήκει σε μία από τις 3 ομάδες. Στα επόμενα 5 σχήματα (β, γ, δ, ε, στ) παρουσιάζεται ο επαναυπολογισμός των παραμέτρων των μοντέλων με αποτέλεσμα να αλλάζουν τα ποσοστά των χρωμάτων των δεδομένων, όπως επίσης το μέγεθος του σχήματος της κάθε ομάδας και τα κέντρα κάθε ομάδας. Στο τελευταίο σχήμα (στ) παρατηρούμε πως τα δεδομένα δεν είναι πλέον πολύ σκορπισμένα αλλά έχουν ομαδοποιηθεί και μπορούμε να δούμε ξεκάθαρα τις ομάδες. Για να τερματίσει ο αλγόριθμος θα πρέπει να υπάρχουν ασήμαντες μεταβολές στις τιμές των παραμέτρων των μοντέλων.

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης



Σχήμα 5.4 Παράδειγμα Αλγορίθμου EM

Κεφάλαιο 6

6.1 Εισαγωγή στα Νευρωνικά Δίκτυα

Το νευρωνικό δίκτυο είναι ένα δίκτυο από απλούς υπολογιστικούς κόμβους (νευρώνες, νευρώνια), διασυνδεδεμένους μεταξύ τους. Είναι εμπνευσμένο από το Κεντρικό Νευρικό Σύστημα (ΚΝΣ), το οποίο και προσπαθεί να προσομοιώσει. Οι νευρώνες είναι τα δομικστοιχεία του δικτύου. Κάθε τέτοιος κόμβος δέχεται ένα σύνολο αριθμητικών εισόδων από διαφορετικές πηγές (είτε από άλλους νευρώνες, είτε από το περιβάλλον), επιτελεί έναν υπολογισμό με βάση αυτές τις εισόδους και παράγει μία έξοδο. Αυτή η έξοδος είτε κατευθύνεται στο περιβάλλον, είτε τροφοδοτείται ως είσοδος σε άλλους νευρώνες του δικτύου. Υπάρχουν τρεις τύποι νευρώνων: οι νευρώνες εισόδου, οι νευρώνες εξόδου και οι υπολογιστικοί νευρώνες ή κρυμμένοι νευρώνες. Τα νευρωνικά δίκτυα είναι μοντέλα εποπτευόμενης μηχανικής μάθησης. Για την εκπαίδευση ενός μοντέλου εποπτευόμενης μηχανικής χρειάζονται δεδομένα τα οποία αποτελούνται από τιμές εισόδου, που δίνονται στο μοντέλο, μαζί με τις αντίστοιχες πραγματικές τιμές εξόδου. Τις τιμές εισόδου τις επεξεργάζεται το μοντέλο και παράγει κάποιες υποθετικές τιμές εξόδου.

Αυτές οι υποθετικές τιμές εξόδου συγκρίνονται με τις πραγματικές τιμές εξόδου και αν το σφάλμα είναι μεγάλο τότε το μοντέλο ρυθμίζεται προκειμένου να βελτιστοποιηθεί το σφάλμα στα δεδομένα εκπαίδευσης. Μετα την εκπαίδευση προκύπτει ένα μοντέλο το οποίο είναι προσαρμοσμένο στο πρόβλημα. Συνεπώς τα μοντέλα εποπτευόμενης μάθησης μπορούν να υπολογίσουν τη λύση για ένα δύσκολο πρόβλημα εφόσον υπάρχουν κάποια στιγμιότυπα με τις τιμές εισόδου και τις αντίστοιχες τιμές εξόδου.

6.1.1 Ανθρώπινο Ακουστικό Σύστημα

Το αυτί εσωτερικά, έχει τη δυνατότητα να διασπάσει ένα σύνθετο ηχητικό κύμα στις ηχητικές συχνότητες που το συνθέτουν και έπειτα να δώσει την πληροφορία αυτή στους νευρώνες προκειμένου να προσδιοριστεί η κατεύθυνση της πηγής. Το σήμα που φτάνει στο τύμπανο του αυτιού, μεταφέρεται μέσω παλμών στον κοχλία (cochlea), όπου κατά μήκος του διεγείρονται διαφορετικά ακουστικά κύτταρα, ανάλογα με τις συχνότητες των σημάτων. Οι διεγέρσεις αυτές μεταδίδονται στους νευρώνες του ακουστικού νεύρου και φτάνουν στον εγκέφαλο όπου και συνδυάζονται με το ακουστικό σήμα του άλλου αυτιού, προκειμένου να προσδιοριστούν οι διαφορές άφιξης (Interaural Time Difference - ITD) και οι διαφορές πλάτους των επιμέρους συχνοτήτων (Interaural Level Differences - ILD).

Το ITD είναι η διαφορά χρόνου άφιξης των σημάτων στα δύο αυτιά, η οποία μπορεί να προσδιορίσει την πλευρά στην οποία βρίσκεται η πηγή, καθώς ο ήχος από εκείνη την πλευρά φτάνει προτίτερα στο αντίστοιχο αυτί. Αντίστοιχα το ILD, η διαφορά των επιπέδων των συχνοτήτων φάσματος του σήματος εκμεταλεύεται την λειτουργία του κεφαλιού ως ακουστική σκιά που μεταβάλλει τις συχνότητες των σημάτων που τις διαπερνούν προκειμένου να εξάγει συμπεράσματα για την θέση της πηγής. Με βάση αυτές τις δύο συνιστώσες ο άνθρωπος μπορεί να εντοπίσει την θέση της πηγής με μεγάλη ακρίβεια καθώς και να χειριστεί τις ανακλάσεις.

Η παραπάνω διαδικασία μετατροπής του ηχητικού σήματος σε ακουστικό δείχνει πως μπορεί να υπάρξει μία υπολογιστική προσομοίωση με στόχο τον προσδιορισμό της κατεύθυνσης της πηγής. Δύο δέκτες μπορούν να διεγείρονται από σήματα της πηγής, είτε άμεσα, είτε προερχόμενα από ανακλάσεις, να αναλύουν τα σήματα στις επιμέρους συχνότητες και να τροφοδοτούν τις πληροφορίες αυτές (μαζί πιθανώς και πληροφορίες για το χώρο) σε ένα νευρωνικό δίκτυο το οποίο θα μπορεί να εξάγει ως αποτέλεσμα τη θέση της πηγής στο χώρο. Η κρουστική απόκριση λαμβάνει υπόψη, όχι μόνο τον απευθείας ήχο από την πηγή στον ακροατή, αλλά και όλες τις ανακλάσεις που θα προέρχονταν από την αναπαραγωγή αυτού του ήχου σε ένα πραγματικό δωμάτιο αυτό ονομάζεται Αμφιωτική Κρουστική Απόκριση Δωματίου (Binaural Room Impulse Response - BRIR).

6.1.2 Μηχανικό σύστημα Ακρόασης

Τα νευρωνικά δίκτυα μπορούν να γενικοποιηθούν όσο αφορά τον άμεσο ήχο αλλά είναι λιγότερα ακριβής όσον αφορά την κατεύθυνση άφιξης των ανακλάσεων. Επίσης, τα μηχανικά συστήματα δεν μπορούν να εντοπίσουν με ακρίβεια την θέση της πηγής όταν παρεμβάλλονται πολλές πηγές και όταν βρίσκονται σε περιβάλλοντα με αντήχηση. Αυτό συμβαίνει διότι στα μηχανικά συστήματα η θέση ενός ήχου δεν μπορεί να προσδιοριστεί με μοναδικό τρόπο από την ITD και την ILD αφού εμφανίζεται ομοιότητα των διαυλικών σημάτων (binaural cues) στα μπροστινά και στα πίσω ημιπεδία με αποτέλεσμα να δημιουργούνται "front-back" σφάλματα. Για την επίλυση αυτών των σφαλμάτων των μηχανικών συστημάτων προτίνεται ένα σύστημα εντοπισμού της θέσης της πηγής χρησιμοποιώντας τα βαθιά νευρωνικά δίκτυα DNNs (Deep Neural Networks) και μία στρατηγική κίνησης της κεφαλής. Όμως, ο εντοπισμός αυτός θα πρέπει να γίνεται σε όλο τον κύκλο γύρω από τον δέκτη (360 μοίρες αζιμουθίου) και όχι σε κάποιο συγκεκριμένο ημιπεδίο.

6.2 Εντοπισμός ήχου με τη Χρήση Νευρωνικών Δικτύων

Στη συνέχεια παρουσιάζεται το προτεινόμενο διαυλικό σύστημα εντοπισμού του ήχου σε πλήρες εύρος αζιμούθιου 360°. Κατά την διάρκεια της εκπαίδευσης του συστήματος χωροθετήθηκαν τα καθαρά σήματα ομιλίας χρησιμοποιώντας κρουστικές αποκρίσεις που σχετίζονται με το κεφάλι HRIRs (Head-Related Impulse Responses) και προστέθηκε διάχυτος θόρυβος προτού υποβληθεί σε επεξεργασία από το διαυλικό μοντέλο για την εξαγωγή των χαρακτηριστικών. Η διαδικασία της εκπαίδευσης ξεκινάει αρχικά με την διαυλική εξαγωγή των χαρακτηριστικών δηλαδή τον υπολογισμό της σχετικής θέσης του ακροατή σε σχέση με την κάθε πηγή. Ο ορισμός της σχετικής θέσης γίνεται μέσω της απόστασης και της γωνίας που σχηματίζει η ευθεία ακροατή-πηγής και ο κάθετος άξονας του δωματίου. Η απόσταση πηγής-ακροατή, επηρεάζει το σήμα εξασθενώντας το και καθυστερώντας το. Στη συνέχεια τα σήματα πρέπει να συνελιχθούν με τις HRTF (Head Related Transfer Function) συναρτήσεις.

Τα σήματα που προκύπτουν καταλήγουν στον ακροατή. Στη συνέχεια τα σήματα που έχουν φτάσει σε κάθε αυτί φιλτράρονται από μια Gammatone τράπεζα φίλτρων. Ο αριθμός και το πλάτος των φίλτρων δίνονται με βάση την κλίμακα ERB (equivalent rectangular bandwidth). Για κάθε σήμα που προκύπτει ορίζεται και μία διαφορετική περιοχή συχνοτήτων. Κάθε σήμα περνά στη συνέχεια από το στάδιο της απόκρισης του ακουστικού νεύρου, το οποίο περιλαμβάνει ένα χαμηλοδιαβατό φίλτρο. Όμως στην συγκεκριμένη περίπτωση δεν χρησιμοποιήθηκε το χαμηλοδιαβατό φίλτρο αφού δεν χρειάστηκε να προσομοιωθεί η χαμηλή χρονική απόκριση των τριχοειδών των νευρικών κυττάρων αφού μελέτες έχουν δείξει ότι οι ταξινομητές είναι σε θέση να εκμεταλλευτούν τη δομή υψηλής συχνότητας. Τελικό στάδιο στην παρούσα υλοποίηση είναι φυσικά ο υπολογισμός των αμφιωτικών παραμέτρων ITD και ILD.

Ενας τρόπος εντοπισμού της θέσης της πηγής γίνεται με την χρήση Βαθιών Νευρωνικών Δικτύων DNNs (Deep Neural Networks). Τα θορυβώδη διαυλικά χαρακτηριστικά χρησιμοποιήθηκαν για να εκπαιδευθούν τα DNNs έτσι ώστε να κατανοήσουν τη σχέση μεταξύ των διαυλικών σημάτων και των αζιμούθιων ήχων. Με τη χρήση των DNN γίνεται η χαρτογράφηση όλων των διαστάσεων (34) των διαυλικών χαρακτηριστικών στις αντίστοιχες γωνίες αζιμουθίου. Για κάθε ζώνη συχνοτήτων εκπαιδεύτηκε διαφορετικό DNN. Για τον εντοπισμό των ταυτόχρονων πηγών ήχου είναι σημαντική η χρήση DNN που εξαρτώνται από την συχνότητα. Σύμφωνα με την έννοια του Bregman κάθε ζώνη συχνοτήτων κυριαρχείται από

μια μόνο πηγή. Λόγω της έννοιας αυτής η εκπαίδευση γίνεται με τη χρήση δεδομένων μονής πηγής.

Η δομή ενός DNN είναι η εξής: αποτελείται από ένα στρώμα εισόδου, δύο κρυφά στρώματα και ένα στρώμα εξόδου. Το στρώμα εισόδου περιλαμβάνεται από έναν αριθμό κόμβων και κάθε κόμβος θεωρήθηκε ότι είναι μια τυχαία Γκαουσιανή μεταβλητή. Στις εισόδους όλων των διαυλικών χαρακτηριστικών προστέθηκε λευκός Γκαουσιανός θόρυβος έτσι ώστε να αποφευχθεί η υπερφόρτωση πριν χρησιμοποιηθούν ως είσοδοι στο DNN. Όσο αφορά τα κρυφά στρώματα έχουν λειτουργίες ενεργοποίησης "sigmoid" δηλαδή μία λειτουργία με την οποία λαμβάνετε ένα οποιοδήποτε εύρος πραγματικού αριθμού και επιστρέφεται η τιμή εξόδου και κάθε στρώμα περιέχει έναν αριθμό από συγκεκριμένους κρυφούς κόμβους. Η λειτουργία "Sigmoid" χρησιμοποιείται για τη δυαδική ταξινόμηση. Το στρώμα εξόδου περιέχει αντίστοιχο αριθμό κόμβων και γωνιών αζιμουθίου. Το στρώμα εξόδου έχει μια λειτουργία ενεργοποίησης την "softmax" δηλαδή μια λειτουργία που υπολογίζει την κατανομή των πιθανοτήτων κάθε τάξεις εισόδου σε όλες τις πιθανές τάξεις εξόδου.

Η λειτουργία "softmax" χρησιμοποιείται για την πολλαπλή ταξινόμηση. Για όλες τις ζώνες συχνοτήτων χρησιμοποιήθηκε η ίδια αρχιτεκτονική DNN.

Κατά την διάρκεια της εκπαίδευσης το νευρωνικό δίκτυο αρχικοποιήθηκε με ένα μόνο κρυφό στρώμα, επίσης σε κάθε φάση της εκπαίδευσης χρησιμοποιήθηκαν μικρές ομάδες και κάποιος ρυθμός ορμής. Στο τέλος κάθε φάσης της εκπαίδευσης προστέθηκε ένα επιπλέον κρυφό στρώμα μεταξύ του τελευταίου κρυμμένου στρώματος και του στρώματος εξόδου και η φάση της εκπαίδευσης επαναλήφθηκε μέχρι να επιτευχθεί ο επιθυμητός αριθμός των κρυφών στρωμάτων. Ο εντοπισμός του ήχου διεξήχθη για ένα μπλοκ σήματος που αποτελείται από χρονικά πλαίσια T. Συνεπώς τα πίσω πλαίσια μετρήθηκαν κατά μέσο όρο διαχρονικά για την παραγωγή μίας οπίσθιας κατανομής από δραστηριότητες της ηχητικής πηγής. Ανάλογα με το αζιμούθιο που δίνονταν παρατηρούνταν και η αντίστοιχη τοποθεσία της θέσης της πηγής. Η έξοδος από το DNN συνδυάζεται με μια στρατηγική κίνησης της κεφαλής για τον σταθερό εντοπισμό των πολλαπλών ομιλητών σε περιβάλλοντα αντήχησης. (Αυτή η στρατηγική εξηγείται στην επόμενη παράγραφο.)

Το πρόβλημα όμως των front-back σφαλμάτων εξακολουθεί να υπάρχει, γι' αυτό το λόγο το προτεινόμενο μοντέλο εντοπισμού χρησιμοποιεί ένα στάδιο ανατροφοδότησης που καθοδηγεί τις κινήσεις της κεφαλής εάν η θέση της πηγής δεν μπορεί να εκτιμηθεί. Χρησιμοποιώντας τα ήδη εκπαιδευμένα DNN και ένα μπλοκ από σήματα μπορούμε να υπολογίσουμε τις τοπικές

κορυφές της οπίσθιας κατανομής που αντιστοιχούν στα αζιμούθια των αληθινών πηγών. Όμως λόγω της ομοιότητας των διαυλικών χαρακτηριστικών στο εμπρόσθιο και στο οπίσθιο ημιπεδίο, οι φανταστικές πηγές μπορούν επίσης να γίνουν εμφανείς ως κορυφές στην οπίσθια κατανομή του αζιμούθου. Εάν συμβεί αυτό, τότε ενεργοποιείται μία τυχαία κίνηση της κεφαλής εντός μια συγκεκριμένης περιοχής για την επίλυση του εντοπισμού του σφάλματος. Το προτεινόμενο σύστημα είναι σε θέση πλέον να αντιμετωπίσει τις επικαλυπτόμενες πηγές ήχου υπό συνθήκες αντήχησης.

6.3 Σύγκριση Μοντέλων DNN-GMM

Στη συνέχεια παρουσιάζεται η σύγκριση δύο μοντέλων του προτεινόμενου DNN (Deep Neural Network) και του μοντέλου GMM (Gaussian Mixture Model) όσον αφορά την επίδραση τριών παραμέτρων: της MCT (Multi-Conditional Training), της ILD (Interaural Level Difference) και την χρήση της κίνησης της κεφαλής.

Το μοντέλο DNN επωφελήθηκε από την επίδραση της MCT αλλά η βελτίωση της απόδοσης ήταν μικρή και παρατηρούνταν μόνο σε σενάρια πολλαπλών ομιλητών. Σε αντίθεση, στο μοντέλο GMM η επίδραση της MCT έφερε μεγάλη βελτίωση σε σενάρια πολλαπλών ομιλητών και κατά την παρουσία αντήχησης στο δωμάτιο. Η μικρή βελτίωση στο DNN οφείλεται στο γεγονός ότι στην καθαρή εκπαίδευση η απόδοση του DNN είναι ήδη ισχυρή στις περισσότερες συνθήκες σε σχέση με το μοντέλο GMM. Έτσι το DNN μπορεί να εξαγάγει αποτελεσματικά τα σήματα από τα καθαρά χαρακτηριστικά CCF (Cross-Correlation Function), ILD όταν ο εντοπισμός περιορίζεται στο μπροστινό ημιπεδίο. Στην περίπτωση του πλήρους εντοπισμού (360°) το σύστημα GMM που χρησιμοποιεί την καθαρή εκπαίδευση απέτυχε να εντοπίσει με ακρίβεια τους συνομιλητές όταν υπήρχαν πολλαπλοί ταυτόχρονοι ομιλητές. Αντίθετα, το σύστημα DNN που χρησιμοποιεί και αυτό την καθαρή εκπαίδευση ήταν ισχυρότερο από το GMM αλλά υπήρξε μείωση της απόδοσης κατά την παρουσία περισσότερων ομιλητών.

Χωρίς την επίδραση των ILD σημάτων η απόδοση εντοπισμού του μοντέλου DNN μειώθηκε. Το μοντέλο αυτό χρησιμοποίησε μόνο το σύνολο χαρακτηριστικών CCF. Με αυτόν τον τρόπο επιτυγχάνεται μεγαλύτερη ανθεκτικότητα στις πολλαπλές συνθήκες αντήχησης. Από την άλλη το σύστημα GMM χρησιμοποίησε μόνο τη λειτουργία ITD. Μόνο με αυτή τη λειτουργία η απόδοση ήταν χαμηλή σε αίθουσες με αντήχηση και η ακρίβεια εντοπισμού μειώθηκε όταν εντοπίστηκαν πολλαπλοί επικαλυπτόμενοι ομιλητές. Το GMM δεν μπόρεσε να εξαγάγει τη συστηματική δομή στη CCF λόγω της κατώτερης ικανότητάς του να διαμορφώνει τα συσχετιζόμενα χαρακτηριστικά. Έτσι, η μέση ακρίβεια εντοπισμού ήταν καλύτερη στο

σύστημα DNN, αυτό έχει ως αποτέλεσμα το σύστημα να είναι πιο ανεκτικό σε συνθήκες με αντήχηση.

Με την χρήση της κίνησης της κεφαλής επωφελήθηκαν και τα δύο συστήματα. Βέβαια, το σύστημα DNN υπερβαίνει του συστήματος GMM όταν τα σφάλματα εντοπισμού δεν οφείλονται στα “front-back” σφάλματα όσον αφορά τα πλευρικά αζιμούθια. Από την άλλη στα κεντρικά αζιμούθια τα σφάλματα εντοπισμού οφείλονται στα “front-back” σφάλματα. Αυτό το γεγονός κάνει το GMM να έχει πιο αξιόπιστο εντοπισμό του ήχου στις κεντρικές θέσεις της πηγής από ότι στις πλευρικές θέσεις όπως το DNN. Όμως χωρίς τη χρήση της κίνησης της κεφαλής τα ποσοστά σφάλματος στα κεντρικά αζιμούθια ήταν υψηλά λόγω των “front-back” σφαλμάτων και στα δύο συστήματα. Το σύστημα DNN πέτυχε ακρίβεια στον εντοπισμό σε όλες τις θέσεις της πηγής, σε αντίθεση το GMM είχε μικρή ακρίβεια στον εντοπισμό στο πίσω μέρος της πηγής αλλά σε άλλες θέσεις η απόδοση ήταν καλύτερη. Τέλος, στον εντοπισμό δύο και τριών πηγών και τα δύο συστήματα επωφελήθηκαν από τη χρήση της κίνησης της κεφαλής στις περισσότερες θέσεις αζιμουθίου.

Βιβλιογραφία

[1] Michael Lovedee-Turner and Damian Murphy, “Application of machine learning for the spatial analysis of binaural room impulse responses” , Communication Technologies Research Group, Department of Electronic Engineering, University of York, 30 October 2017. [Online]. Available: <https://www.mdpi.com/2076-3417/8/1/105/pdf>

[2] Juha Backman and Matti Karjalainem, “Modeling of human directional and spatial hearing using neural networks”, Helsinki University of Tegnology, Acoustics Laboratory Otakaari 5A, Finland May 1993. [Online] Available <https://www.researchgate.net/publication/3567442>

[3] Ning Ma, Tobias May, and Guy J. Brown, “Exploiting deep neural networks and head movements for robust binaural localisation of multiple sources in reverberant environments”, IEEE Transactions on Audio, Speech, and Language Processing. ISSN 1558-7916 (12, Dec. 2017). [Online]. Available: <https://ieeexplore.ieee.org/document/8086216>

[4] dr.ir. Emanuel A.P. Habets “Room Impulse Response Generator”, September 20, 2010. [Online]. Available: https://www.researchgate.net/publication/259991276_Room_Impulse_Response_Generator

[5] J. Mourjopoulos and M. A. Paraskevas “Pole and Zero Modeling of Room Transfer Functions” , Wire Communications Laboratory, Electrical Engineering Department, University of Patras, Journal of Sound and Vibration (1991) [Online]. Available: http://www.wcl.ece.upatras.gr/audiogroup/publications/pdfs/pole_zero_md.pdf

[6] Μ. Χαλκίδη, Μ. Βαζιργιάννης, “Εξόρυξη Γνώσης από Βάσεις Δεδομένων και τον Παγκόσμιο Ιστό”, Β' Έκδοση, Εκδόσεις "τυπωθήτω – Γιώργος Δάρδανος", 2005.

[7] Τετερίνα Κατερίνα ,“Ακουστική συμπεριφορά της αίθουσας τελετών του Πανεπιστημίου Μακεδονίας”, Θεσσαλονίκη Φεβρουάριος 2010, Πτυχιακή Εργασία [Online]. Available: https://dspace.lib.uom.gr/bitstream/2159/13785/2/Teterina_Pe2010.pdf

[8] Παναγιώτης Ζαχαράκης ,“Δημιουργία εκπαιδευτικών μαθημάτων μουσικής τεχνολογίας, με τη χρήση νέων τεχνολογιών για μορφές τηλεεκπαίδευσης”, Θεσσαλονίκη 2005, Πτυχιακή Εργασία [Online]. Available: <https://dspace.lib.uom.gr/bitstream/2159/3456/4/ZacharakisPanagiotisPe2005.pdf>

[9] Αναστασίου Ελένη, “Ακουστική διόρθωση αιθουσών διδασκαλίας μουσικών οργάνων”, Θεσσαλονίκη Σεπτέμβριος 2009, Πτυχιακή Εργασία [Online]. Available: <https://dspace.lib.uom.gr/bitstream/2159/13625/2/AnastasiouPE2009.pdf>

[10] Καπετανόγλου Νικόλαος, Παγίδας Ιωάννης “Μελέτη για ηχητική εγκατάσταση προδιαγραφών κινηματογραφικής αίθουσας στο χώρο του αμφιθεάτρου ”, Ρέθυμνο 2007, Πτυχιακή Εργασία [Online]. Available: http://nefeli.lib.teicrete.gr/browse/sefe/mta/2007/KapetanoglouNikolaos,PagidasIoannis/attached-document/Kapetanoglou_Pagidas.pdf

[11] Μπακατσάκης Ιωάννης, “Εφαρμογή Υπολογιστικής Ακουστικής σε Κλειστούς Χώρους”, Ρέθυμνο Μάρτιος 2012, Πτυχιακή Εργασία [Online]. Available: <http://nefeli.lib.teicrete.gr/browse/sefe/mta/2012/MpakatsakisIoannis/attached-document-1355394703-328006-15821/MpakatsakisIoannis2012.pdf>

[12] Claus Lynge Christensen, George Koutsouris, Juan Gil, Jens-Holger Rindel, “Applying_In-situ_Recalibration_For_Sound_Strength_Measurements_in_Auditoria”, Proceedings of the Institute of Acoustics, Odeon A/S, Denmark [Online] Available: https://odeon.dk/pdf/Applying_Insitu_Recalibration_For_Sound_Strength_Measurements_in_Auditoria.pdf

[13] Θωμάς Σ. Ζαρούχας, “Μοντελοποίηση και επεξεργασία ηχητικών δεδομένων για αναπαραγωγή σε χώρους με αντήχηση” Πάτρα Ιούνιος 2010, Διδακτορική Διατριβή [Online]. Available: http://nemertes.lis.upatras.gr/jspui/bitstream/10889/3968/1/ThomasZarouchas_PhD.pdf

[14] Λάμπρος Ν. Ρήγας, “Χρήση τεχνικών εξόρυξης γνώσης σε ιατρικά δεδομένα”, Πάτρα Μάρτιος 2014, Διδακτορική Διατριβή [Online]. Available: <http://nemertes.lis.upatras.gr/jspui/bitstream/10889/8561/1/datamining%20medical%20data.pdf>

[15] Ι. Βλαχάβας, Π. Κεφαλάς, Ν. Βασιλειάδης, Φ. Κόκκορας, Η. Σακελλαρίου, Τεχνητή Νοημοσύνη - Β' Έκδοση [Online]. Available: <http://aibook.csd.auth.gr/include/slides/pdf>

[16] Lisa Tagliaferri, “An Introduction to Machine Learning”, Σεπτέμβριος 28, 2017 [Online]. Available: <https://www.digitalocean.com/community/tutorials/an-introduction-to-machine-learning>

[17] Alex Smola and S.V.N. Vishwanathan, “Introduction to Machine Learning”, Cambridge University Press 2008 [Online]. Available: <http://alex.smola.org/drafts/thebook.pdf>

[18] S. B. Kotsiantis, I. D. Zaharakis, P. E. Pintelas “Machine learning: a review of classification and combining techniques”, Springer Science+Business Media B.V, 10 November 2007 [Online]. Available: http://www.cs.bham.ac.uk/~pxt/IDA/class_rev.pdf

[19] Jason Brownlee, “Naive Bayes for Machine Learning”, April 11, 2016 [Online]. Available: <https://machinelearningmastery.com/naive-bayes-for-machine-learning/>

Κώδικας Υλοποίησης σε Περιβάλλον MATLAB

Παράρτημα Α

Για το αρχείο ROOM Response.m

```
clear all; clc

Room = [5.0 5.0 4.0]; % Room dimensions
Sensor = [2.5 2.5 2.0]; % Sensor location
Source = [4.5 2.5 2.0]; % Source location
AbsCoefs = [0.75 0.75 0.8 0.25 0.3 0.9]; % Wall absorption coefficients

Fs = 16000; % Sampling frequency
RT_Type = 'T60'; % or 'T20'
RT_Value = 0.25; % Reverberation Time (sec)

% Calculate the room impulse response h[n]
h = ISM_RoomResp(Fs, AbsCoefs, RT_Type, RT_Value, Source, Sensor, Room);

% Create time scale
n = 0:1/Fs:(length(h)-1)/Fs;

% Plot impulse response h(t)
figure(1); subplot(211); plot(n, h);

N = 1024; % FFT length
H = fft(h, N); % Calculate FFT

% Create frequency scale
f = 0 : 1*(Fs/N) : (N/2)*(Fs/N);

% Plot magnitude response |H(w)|
subplot(212); plot(f, 20*log10(abs(H(1:N/2+1))))
```

Για το αρχείο All zero filter.m

```
clc; clear all; close all;

% Describe room parameters
fs = 10000; % sampling frequency
VC = 40; % num of virtual sources
r = 0.5; % absorption coeffs
rm = [5.0 5.0 4.0]; % Room dimensions
mic = [0.5 2.5 2.0]; % mic location
src = [4.5 2.5 2.0]; % source location

% Reverberation Time and Critical Distance
[RT60, CR] = RT60(rm, r);

hlen = 0.5*fs; % length of h[n]

n = [0:hlen-1]/fs;
```


Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```

% h = exp(-2*n) .* (sin(5000*pi*n)+2*cos(7000*pi*n));

% Calculate room impulse response h[n]
h = rir(fs, mic, VC, r, rm, src);
h = h';

Ne = 1000; % Active length

M = 4096; % DFT length
H = fft(h, M); % Original RTF
P = abs(H).^2 ./ max(abs(H)); % Normalized power spectrum

i = 0;

for p = 1:1:200;

    i = i+1; order(i) = p;

    R = autocorr(h, p); % Calculate autorrelation of h[n]
    [d, e, k] = levinson(R); % Levison - Durbin algorithm

    D = fft(d, M); % DFT of All Zero
    E = H.*D; % Error spectrum

    Hap = 1./D; % All Pole function
    hap = ifft(Hap, M); % All pole hap[n]

    Pap = abs(Hap).^2 ./ max(abs(Hap)); % Normalized power spectrum

    Vmin(p) = exp(sum(log10(Pap))/M) / R(1); % Error
    I(p) = log10(Vmin(p)) + 2*p/Ne; % Information criterion

end

figure(3);
plot(order, Vmin, 'b'); hold on; %ylim([-10,10]);
plot(order, I, 'm'); hold off; %ylim([-10,10]);

[Vmin,Ipos] = min(I)

order(Ipos);

% Calculate for minimum order

R = autocorr(h, Ipos); % Calculate autorrelation of h[n]
[d, e, k] = levinson(R); % Levison - Durbin algorithm

D = fft(d, M); % DFT of All Zero
E = H.*D; % Error spectrum

Hap = 1./D; % All Pole function
hap = ifft(Hap, M); % All pole hap[n]
Pap = abs(Hap).^2 ./ max(abs(Hap)); % Normalized power spectrum

figure(1);
subplot(211); plot(h(1:M)); title('h[n]') %ylim([-1.2 1.2])
subplot(212); plot(hap(1:M)); title('hap[n]'); %ylim([-1.2 1.2])

```

```
figure(2);  
plot(20*log10(P (1:M/2+1))); hold on; %ylim([-10,10]);  
plot(20*log10(Pap(1:M/2+1)), 'm', 'LineWidth', 2) ; hold off; %ylim([-  
10,10]);
```

Για το αρχείο rir_generator.cpp

/*

Program : Room Impulse Response Generator

Description : Computes the response of an acoustic source to one or more microphones in a reverberant room using the image method [1,2].

[1] J.B. Allen and D.A. Berkley,
Image method for efficiently simulating small-room acoustics,
Journal Acoustic Society of America, 65(4), April 1979, p
943.

[2] P.M. Peterson,
Simulating the response of multiple microphones to a single
acoustic source in a reverberant room, Journal Acoustic
Society of America, 80(5), November 1986.

Author : dr.ir. E.A.P. Habets (ehabets@dereverberation.org)

Version : 2.1.20141124

History : 1.0.20030606 Initial version
1.1.20040803 + Microphone directivity
+ Improved phase accuracy [2]
1.2.20040312 + Reflection order
1.3.20050930 + Reverberation Time
1.4.20051114 + Supports multi-channels
1.5.20051116 + High-pass filter [1]
+ Microphone directivity control
1.6.20060327 + Minor improvements
1.7.20060531 + Minor improvements
1.8.20080713 + Minor improvements
1.9.20090822 + 3D microphone directivity control
2.0.20100920 + Calculation of the source-image position
changed in the code and tutorial.
This ensures a proper response to reflections
in case a directional microphone is used.
2.1.20120318 + Avoid the use of unallocated memory
2.1.20140721 + Fixed computation of alpha
2.1.20141124 + The window and sinc are now both centered
around t=0

Copyright (C) 2003-2014 E.A.P. Habets, The Netherlands.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
*/

```
#define _USE_MATH_DEFINES

#include "matrix.h"
#include "mex.h"
#include "math.h"

#define ROUND(x) ((x)>=0?(long)((x)+0.5):(long)((x)-0.5))

#ifndef M_PI
#define M_PI 3.14159265358979323846
#endif

double sinc(double x)
{
    if (x == 0)
        return(1.);
    else
        return(sin(x)/x);
}

double sim_microphone(double x, double y, double z, double* angle, char
mtype)
{
    if (mtype=='b' || mtype=='c' || mtype=='s' || mtype=='h')
    {
        double gain, vartheta, varphi, rho;

        // Polar Pattern          rho
        // -----
        // Bidirectional           0
        // Hypercardioid           0.25
        // Cardioid                 0.5
        // Subcardioid             0.75
        // Omnidirectional         1

        switch(mtype)
        {
            case 'b':
                rho = 0;
                break;
            case 'h':
                rho = 0.25;
                break;
            case 'c':
                rho = 0.5;
                break;
            case 's':
                rho = 0.75;
                break;
        };
    };
}
```

```
vartheta = acos(z/sqrt(pow(x,2)+pow(y,2)+pow(z,2)));
varphi = atan2(y,x);

gain = sin(M_PI/2-angle[1]) * sin(vartheta) * cos(angle[0]-varphi)
+ cos(M_PI/2-angle[1]) * cos(vartheta);
gain = rho + (1-rho) * gain;

return gain;
}
else
{
return 1;
}
}

void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray
*prhs[])
{
if (nrhs == 0)
{
mexPrintf("-----\n"
-----\n"
"| Room Impulse Response Generator
|\n"
"|
|\n"
"| Computes the response of an acoustic source to one or more
|\n"
"| microphones in a reverberant room using the image method
[1,2]. |\n"
"|
|\n"
"| Author : dr.ir. Emanuel Habets
(ehabets@dereverberation.org) |\n"
"|
|\n"
"| Version : 2.1.20141124
|\n"
"|
|\n"
"| Copyright (C) 2003-2014 E.A.P. Habets, The Netherlands.
|\n"
"|
|\n"
"| [1] J.B. Allen and D.A. Berkley,
|\n"
"| Image method for efficiently simulating small-room
acoustics,|\n"
"| Journal Acoustic Society of America,
|\n"
"| 65(4), April 1979, p 943.
|\n"
"|
|\n"
"| [2] P.M. Peterson,
|\n"
"| Simulating the response of multiple microphones to a
single |\n"
"| acoustic source in a reverberant room, Journal Acoustic
|\n"
```

```

        | Society of America, 80(5), November 1986.
|\n"
-----\n\n"
function [h, beta_hat] = rir_generator(c, fs, r, s, L, beta,
nsample,\n"
" mtype, order, dim, orientation, hp_filter);\n\n"
"Input parameters:\n"
" c          : sound velocity in m/s.\n"
" fs         : sampling frequency in Hz.\n"
" r          : M x 3 array specifying the (x,y,z) coordinates
of the\n"
"            receiver(s) in m.\n"
" s          : 1 x 3 vector specifying the (x,y,z) coordinates
of the\n"
"            source in m.\n"
" L          : 1 x 3 vector specifying the room dimensions
(x,y,z) in m.\n"
" beta       : 1 x 6 vector specifying the reflection
coefficients\n"
"            [beta_x1 beta_x2 beta_y1 beta_y2 beta_z1
beta_z2] or\n"
"            beta = reverberation time (T_60) in seconds.\n"
" nsample    : number of samples to calculate, default is
T_60*fs.\n"
" mtype      : [omnidirectional, subcardioid, cardioid,
hypercardioid,\n"
"            bidirectional], default is omnidirectional.\n"
" order      : reflection order, default is -1, i.e. maximum
order.\n"
" dim        : room dimension (2 or 3), default is 3.\n"
" orientation : direction in which the microphones are pointed,
specified using\n"
"            azimuth and elevation angles (in radians),
default is [0 0].\n"
" hp_filter  : use 'false' to disable high-pass filter, the
high-pass filter\n"
"            is enabled by default.\n\n"
"Output parameters:\n"
" h          : M x nsample matrix containing the calculated
room impulse\n"
"            response(s).\n"
" beta_hat   : In case a reverberation time is specified as an
input parameter\n"
"            the corresponding reflection coefficient is
returned.\n\n");
    return;
}
else
{
    mexPrintf("Room Impulse Response Generator (Version 2.1.20141124)
by Emanuel Habets\n"
"Copyright (C) 2003-2014 E.A.P. Habets, The Netherlands.\n");
}

// Check for proper number of arguments
if (nrhs < 6)
    mexErrMsgTxt("Error: There are at least six input parameters
required.");
if (nrhs > 12)
    mexErrMsgTxt("Error: Too many input arguments.");

```

```
if (nlhs > 2)
    mexErrMsgTxt("Error: Too many output arguments.");

// Check for proper arguments
if (!(mxGetN(prhs[0])==1) || !mxIsDouble(prhs[0]) ||
mxIsComplex(prhs[0]))
    mexErrMsgTxt("Invalid input arguments!");
if (!(mxGetN(prhs[1])==1) || !mxIsDouble(prhs[1]) ||
mxIsComplex(prhs[1]))
    mexErrMsgTxt("Invalid input arguments!");
if (!(mxGetN(prhs[2])==3) || !mxIsDouble(prhs[2]) ||
mxIsComplex(prhs[2]))
    mexErrMsgTxt("Invalid input arguments!");
if (!(mxGetN(prhs[3])==3) || !mxIsDouble(prhs[3]) ||
mxIsComplex(prhs[3]))
    mexErrMsgTxt("Invalid input arguments!");
if (!(mxGetN(prhs[4])==3) || !mxIsDouble(prhs[4]) ||
mxIsComplex(prhs[4]))
    mexErrMsgTxt("Invalid input arguments!");
if (!(mxGetN(prhs[5])==6 || mxGetN(prhs[5])==1) || !mxIsDouble(prhs[5])
|| mxIsComplex(prhs[5]))
    mexErrMsgTxt("Invalid input arguments!");

// Load parameters
double        c = mxGetScalar(prhs[0]);
double        fs = mxGetScalar(prhs[1]);
const double* rr = mxGetPr(prhs[2]);
int           nMicrophones = (int) mxGetM(prhs[2]);
const double* ss = mxGetPr(prhs[3]);
const double* LL = mxGetPr(prhs[4]);
const double* beta_input = mxGetPr(prhs[5]);
double*       beta = new double[6];
int           nSamples;
char*         microphone_type;
int           nOrder;
int           nDimension;
double        angle[2];
int           isHighPassFilter;
double        reverberation_time = 0;

// Reflection coefficients or reverberation time?
if (mxGetN(prhs[5])==1)
{
    double V = LL[0]*LL[1]*LL[2];
    double S = 2*(LL[0]*LL[2]+LL[1]*LL[2]+LL[0]*LL[1]);
    reverberation_time = beta_input[0];
    if (reverberation_time != 0) {
        double alfa = 24*V*log(10.0)/(c*S*reverberation_time);
        if (alfa > 1)
            mexErrMsgTxt("Error: The reflection coefficients cannot be
calculated using the current "
                "room parameters, i.e. room size and
reverberation time.\n        Please "
                "specify the reflection coefficients or change
the room parameters.");
        for (int i=0;i<6;i++)
            beta[i] = sqrt(1-alfa);
    }
    else
    {
        for (int i=0;i<6;i++)
```

```
        beta[i] = 0;
    }
}
else
{
    for (int i=0;i<6;i++)
        beta[i] = beta_input[i];
}

// High-pass filter (optional)
if (nrhs > 11 && mxIsEmpty(prhs[11]) == false)
{
    isHighPassFilter = (int) mxGetScalar(prhs[11]);
}
else
{
    isHighPassFilter = 1;
}

// 3D Microphone orientation (optional)
if (nrhs > 10 && mxIsEmpty(prhs[10]) == false)
{
    const double* orientation = mxGetPr(prhs[10]);
    if (mxGetN(prhs[10]) == 1)
    {
        angle[0] = orientation[0];
        angle[1] = 0;
    }
    else
    {
        angle[0] = orientation[0];
        angle[1] = orientation[1];
    }
}
else
{
    angle[0] = 0;
    angle[1] = 0;
}

// Room Dimension (optional)
if (nrhs > 9 && mxIsEmpty(prhs[9]) == false)
{
    nDimension = (int) mxGetScalar(prhs[9]);
    if (nDimension != 2 && nDimension != 3)
        mexErrMsgTxt("Invalid input arguments!");

    if (nDimension == 2)
    {
        beta[4] = 0;
        beta[5] = 0;
    }
}
else
{
    nDimension = 3;
}

// Reflection order (optional)
if (nrhs > 8 && mxIsEmpty(prhs[8]) == false)
```

```
{
    nOrder = (int) mxGetScalar(prhs[8]);
    if (nOrder < -1)
        mexErrMsgTxt("Invalid input arguments!");
}
else
{
    nOrder = -1;
}

// Type of microphone (optional)
if (nrhs > 7 && mxIsEmpty(prhs[7]) == false)
{
    microphone_type = new char[mxGetN(prhs[7])+1];
    mxGetString(prhs[7], microphone_type, mxGetN(prhs[7])+1);
}
else
{
    microphone_type = new char[1];
    microphone_type[0] = 'o';
}

// Number of samples (optional)
if (nrhs > 6 && mxIsEmpty(prhs[6]) == false)
{
    nSamples = (int) mxGetScalar(prhs[6]);
}
else
{
    if (mxGetN(prhs[5])>1)
    {
        double V = LL[0]*LL[1]*LL[2];
        double alpha = ((1-pow(beta[0],2))+(1-
pow(beta[1],2))*LL[1]*LL[2] +
        ((1-pow(beta[2],2))+(1-pow(beta[3],2)))*LL[0]*LL[2] +
        ((1-pow(beta[4],2))+(1-pow(beta[5],2)))*LL[0]*LL[1];
        reverberation_time = 24*log(10.0)*V/(c*alpha);
        if (reverberation_time < 0.128)
            reverberation_time = 0.128;
    }
    nSamples = (int) (reverberation_time * fs);
}

// Create output vector
plhs[0] = mxCreateDoubleMatrix(nMicrophones, nSamples, mxREAL);
double* imp = mxGetPr(plhs[0]);

// Temporary variables and constants (high-pass filter)
const double W = 2*M_PI*100/fs; // The cut-off frequency equals 100 Hz
const double R1 = exp(-W);
const double B1 = 2*R1*cos(W);
const double B2 = -R1 * R1;
const double A1 = -(1+R1);
double X0;
double* Y = new double[3];

// Temporary variables and constants (image-method)
const double Fc = 1; // The cut-off frequency equals fs/2 - Fc is the
normalized cut-off frequency.
```


Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```

const int      Tw = 2 * ROUND(0.004*fs); // The width of the low-pass FIR
equals 8 ms
const double cTs = c/fs;
double*       LPI = new double[Tw];
double*       r = new double[3];
double*       s = new double[3];
double*       L = new double[3];
double        Rm[3];
double        Rp_plus_Rm[3];
double        refl[3];
double        fdist,dist;
double        gain;
int           startPosition;
int           n1, n2, n3;
int           q, j, k;
int           mx, my, mz;
int           n;

s[0] = ss[0]/cTs; s[1] = ss[1]/cTs; s[2] = ss[2]/cTs;
L[0] = LL[0]/cTs; L[1] = LL[1]/cTs; L[2] = LL[2]/cTs;

for (int idxMicrophone = 0; idxMicrophone < nMicrophones ;
idxMicrophone++)
{
    // [x_1 x_2 ... x_N y_1 y_2 ... y_N z_1 z_2 ... z_N]
    r[0] = rr[idxMicrophone + 0*nMicrophones] / cTs;
    r[1] = rr[idxMicrophone + 1*nMicrophones] / cTs;
    r[2] = rr[idxMicrophone + 2*nMicrophones] / cTs;

    n1 = (int) ceil(nSamples/(2*L[0]));
    n2 = (int) ceil(nSamples/(2*L[1]));
    n3 = (int) ceil(nSamples/(2*L[2]));

    // Generate room impulse response
    for (mx = -n1 ; mx <= n1 ; mx++)
    {
        Rm[0] = 2*mx*L[0];

        for (my = -n2 ; my <= n2 ; my++)
        {
            Rm[1] = 2*my*L[1];

            for (mz = -n3 ; mz <= n3 ; mz++)
            {
                Rm[2] = 2*mz*L[2];

                for (q = 0 ; q <= 1 ; q++)
                {
                    Rp_plus_Rm[0] = (1-2*q)*s[0] - r[0] + Rm[0];
                    refl[0] = pow(beta[0], abs(mx-q)) * pow(beta[1],
abs(mx));

                    for (j = 0 ; j <= 1 ; j++)
                    {
                        Rp_plus_Rm[1] = (1-2*j)*s[1] - r[1] + Rm[1];
                        refl[1] = pow(beta[2], abs(my-j)) *
pow(beta[3], abs(my));

                        for (k = 0 ; k <= 1 ; k++)
                        {

```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```

Rm[2];
pow(beta[5], abs(mz));

Rp_plus_Rm[2] = (1-2*k)*s[2] - r[2] +
refl[2] = pow(beta[4],abs(mz-k)) *

dist = sqrt(pow(Rp_plus_Rm[0], 2) +
pow(Rp_plus_Rm[1], 2) + pow(Rp_plus_Rm[2], 2));

if (abs(2*mx-q)+abs(2*my-j)+abs(2*mz-k) <=
nOrder || nOrder == -1)
{
    fdist = floor(dist);
    if (fdist < nSamples)
    {
        gain =
sim_microphone(Rp_plus_Rm[0], Rp_plus_Rm[1], Rp_plus_Rm[2], angle,
microphone_type[0])
*
refl[0]*refl[1]*refl[2]/(4*M_PI*dist*cTs);

        for (n = 0 ; n < Tw ; n++)
            LPI[n] = 0.5 * (1 -
cos(2*M_PI*((n+1-(dist-fdist))/Tw)) * Fc * sinc(M_PI*Fc*(n+1-(dist-fdist)-
(Tw/2)));

            startPosition = (int) fdist-
(Tw/2)+1;

            for (n = 0 ; n < Tw; n++)
                if (startPosition+n >= 0 &&
startPosition+n < nSamples)
                    imp[idxMicrophone +
nMicrophones*(startPosition+n)] += gain * LPI[n];
            }
        }
    }
}

// 'Original' high-pass filter as proposed by Allen and Berkley.
if (isHighPassFilter == 1)
{
    for (int idx = 0 ; idx < 3 ; idx++) {Y[idx] = 0;}
    for (int idx = 0 ; idx < nSamples ; idx++)
    {
        X0 = imp[idxMicrophone+nMicrophones*idx];
        Y[2] = Y[1];
        Y[1] = Y[0];
        Y[0] = B1*Y[1] + B2*Y[2] + X0;
        imp[idxMicrophone+nMicrophones*idx] = Y[0] + A1*Y[1] +
R1*Y[2];
    }
}

if (nlhs > 1) {
    plhs[1] = mxCreateDoubleMatrix(1, 1, mxREAL);
    double* beta_hat = mxGetPr(plhs[1]);
}

```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```
        if (reverberation_time != 0) {
            beta_hat[0] = beta[0];
        }
        else {
            beta_hat[0] = 0;
        }
    }

    delete[] beta;
    delete[] microphone_type;
    delete[] Y;
    delete[] LPI;
    delete[] r;
    delete[] s;
    delete[] L;
}
```

Για το αρχείο T60.m

```
function [ T60 ] = T60(x,Fs)
x2 = sqrt(x.^2);
L = length(x);
maxA = cumsum(x2);
invert_x2 = x2';
for n = 1:L; n1 = 0;
    if 20*log10(sum(x2(n:L))./maxA) <=-60;
        n1 = n1;
    else
        n1 = n1+1;
    end
end
T60 = n1/Fs;
end
```

Για το αρχείο SetUserVars.m

```
function [EvalStr] = SetUserVars(VarList,varargin)
%SetUserVars Checks and sets user-definable variables
%
% [evalstr] = SetUserVars(VarList,UserVarCell)
%
% This function reads in a series of 'ArgName' -- 'ArgVal' parameters from
% the cell array 'UserVarCell' and checks them with the list of
% user-definable parameters in the cell array 'VarList'. It returns a
% string 'evalstr' that can be evaluated back in the calling function and
% that sets the parameters to either the user-defined values 'ArgVal' or
% the default values in 'VarList'. Use as in example below.
%
% function [] = DoIt(DoIt_in,varargin);
% %DoIt Executes some code
% %[DoIt_out] = DoIt(DoIt_in,'ArgName1',ArgVal1,'ArgName2',ArgVal2,...)
% %This function accepts a series of argument pairs to set user variables
%
```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```
% %User-definable parameters with default values:
% VarList = {'DoItParam1' 5.7; % numerical parameter
% 'DoItParam2' [1 0; 0 1]; % matrix parameter
% 'DoItParam3' 'none'; % string parameter
% eval(SetUserVars(VarList,varargin)); % set the function parameters to
either the
% % default values or user-defined
values
%
% ... your "DoIt" function code using variables DoItParam1, DoItParam2,
etc. ...
%

% Release date: August 2008
% Author: Eric A. Lehmann, Perth, Australia (www.eric-lehmann.com)

if length(varargin)==1,
    varargin = varargin{1}; % if SetUserVars is called within other
function as in the example above
end

if mod(length(varargin),2)~=0,
    error('Number of arguments is odd: arguments must be passed in pairs
('ArgName',ArgVal).');
end

EvalStr = [];
NumSetVars = 0;
for ii=1:size(VarList,1),
    tmp = find(strcmp(varargin,VarList{ii,1})); % Find argument index
in varargin cell vector.
    if length(tmp)==1, % If argument is found and unique, set variable
appropriately.
        if ischar(varargin{tmp+1}),
            EvalStr = [EvalStr VarList{ii,1} '=' varargin{tmp+1} '''];
        else
            EvalStr = [EvalStr VarList{ii,1} '=' mat2str(varargin{tmp+1})
'''];
        end
        NumSetVars = NumSetVars + 1; % Check the number of variables
set from varargin.
    else % Assign default value if parameter not passed as
function argument.
        if ischar(VarList{ii,2}),
            EvalStr = [EvalStr VarList{ii,1} '=' VarList{ii,2} '''];
        else
            EvalStr = [EvalStr VarList{ii,1} '=' mat2str(VarList{ii,2})
'''];
        end
    end
end

if NumSetVars~=length(varargin)/2,
    error('Some of the input arguments could not be set properly (check
syntax)')
end
```

Για το αρχείο RT60.m

```
function [ RT60, CD ] = RT60( rm, r)

V = rm(1) * rm(2) * rm(3);
A = 2 * r * ( rm(1)*rm(2) + rm(1)*rm(3) + rm(2)*rm(3) );

RT60 = 0.161 * V / A;           % Reverberation time
CD = 0.057 * sqrt(V / RT60);   % Critical distance

end
```

Για το αρχείο rir.m

```
function [h] = rir(fs, mic, n, r, rm, src);

%RIR Room Impulse Response.
% [h] = RIR(FS, MIC, N, R, RM, SRC) performs a room impulse
% response calculation by means of the mirror image method.
%
% FS = sample rate.
% MIC = row vector giving the x,y,z coordinates of
% the microphone.
% N = The program will account for (2*N+1)^3 virtual sources
% R = reflection coefficient for the walls, in general -1<R<1.
% RM = row vector giving the dimensions of the room.
% SRC = row vector giving the x,y,z coordinates of
% the sound source.
%
% EXAMPLE:
%
% >>fs=44100;
% >>mic=[19 18 1.6];
% >>n=12;
% >>r=0.3;
% >>rm=[20 19 21];
% >>src=[5 2 1];
% >>h=rir(fs, mic, n, r, rm, src);
%
% NOTES:
%
% 1) All distances are in meters.
% 2) The output is scaled such that the largest value of the
% absolute value of the output vector is equal to one.
% 3) To implement this filter, you will need to do a fast
% convolution. The program FCONV.m will do this. It can be
% found on the Mathworks File Exchange at
% www.mathworks.com/matlabcentral/fileexchange/. It can also
% be found at http://www.sgm-audio.com/research/rir/fconv.m
% 4) A paper has been written on this model. It is available at:
% http://www.sgm-audio.com/research/rir/rir.html
%
% Version 3.4.2
% Copyright © 2003 Stephen G. McGovern
```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

%Some of the following comments are references to equations the my paper.

```
nn = -n:1:n; % Index for the sequence
rms = nn + 0.5 - 0.5*(-1).^nn; % Part of equations 2,3,& 4
srcs = (-1).^nn; % part of equations 2,3,& 4
xi = srcs*src(1) + rms*rm(1) - mic(1); % Equation 2
yj = srcs*src(2) + rms*rm(2) - mic(2); % Equation 3
zk = srcs*src(3) + rms*rm(3) - mic(3); % Equation 4

[i,j,k] = meshgrid(xi,yj,zk); % convert vectors to 3D matrices
d = sqrt(i.^2+j.^2+k.^2); % Equation 5
time = round(fs*d/343) + 1; % Similar to Equation 6

[e,f,g] = meshgrid(nn, nn, nn); % convert vectors to 3D matrices
c = r.^(abs(e)+abs(f)+abs(g)); % Equation 9
e = c./d; % Equivalent to Equation 10

h = full(sparse(time(:),1,e(:))); % Equivalent to equation 11
h = h/max(abs(h)); % Scale output
```

Για το αρχείο reusefig.m

```
function [FigOH,Exists] = reusefig(TagName,varargin)
%reusefig Creates and/or re-uses a figure with specific tag
%
% [FigOH,Exists] =
reusefig(TagName,'PropName1',PropVal1,'PropName2',PropVal2,...)
%
% Looks for the figure with tag 'TagName' and makes it current, or
% creates it if necessary. Returns the handle 'FigOH' of the figure.
%
% This function can be used in order to re-use the same plot windows
% for several calls of the same plotting routine (avoids creating new
% figures at every call or plotting in windows "belonging" to other
% functions).
%
% A list of arguments pairs can be passed on to this function which will
% use them to set the various properties of the figure (see SET).
%
% The function returns Exists=1 if the figure already existed before, 0 if
% it had to create a new figure.

% Release date: August 2008
% Author: Eric A. Lehmann, Perth, Australia (www.eric-lehmann.com)

FigOH = findobj('type','figure','tag',TagName);

if isempty(FigOH),
    % Create a new figure if no existing one found:
    FigOH =
figure('tag',TagName,'IntegerHandle','off','NumberTitle','off','Name',TagNa
me,varargin{:});
    % Uses non-integer handle to reduce the probability of other functions
inadvertently
    % plotting into it (which is why the function reusefig is done for!).
    Exists = 0;
```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```
elseif isequal(size(FigOH), [1 1]),
    figure(FigOH); % If already existing, makes it current and
brings it to screen.
    if ~isempty(varargin),
        set(FigOH, varargin{:}); % Set figure properties to desired user
values.
    end
    Exists = 1;
    %shg;
else
    error('More than one object found with same tag!');
end
```

Για το αρχείο PrintLoopPCw.m

```
function [Percent] = PrintLoopPCw(tt, NumFrames, oldPercent)
%PrintLoopPCw Prints computation percentage on screen
%
% [PC] = PrintLoopPCw(LOOP, NUMLOOPS, oldPC) prints the current percentage of
% computations done in a loop. Use and initialize as in the following
% examples. Also enables the use with WHILE loops.
%
% PrintLoopPCw('\n\n Starting computations. ');
% for LOOP=1:NUMLOOPS,
%     PrintLoopPCw(LOOP, NUMLOOPS);
%
%     % ...
%     % ... your loop code ...
%     % ...
% end
%
% Nested FOR loops example:
%
% PrintLoopPCw('\n\n Starting computations. ');
% for jj=1:NUMJJ,
%     for kk=1:NUMKK,
%         PrintLoopPCw((jj-1)*NUMKK+kk, NUMJJ*NUMKK);
%
%         % ...
%         % ... your loop code ...
%         % ...
%     end
% end
%
% WHILE loop example:
%
% LOOP = 1;
% PC = PrintLoopPCw('\n\n Starting computations. ');
% while LOOP<=NUMLOOPS,
%     PC = PrintLoopPCw(LOOP, NUMLOOPS, PC);
%
%     % ...
%     % ... your loop code including "LOOP = LOOP + 1;" at some stage
(even conditionally) ...
%     % ...
% end
```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```
% Release date: August 2008
% Author: Eric A. Lehmann, Perth, Australia (www.eric-lehmann.com)

Percent = 0;

if nargin==0,          % initialisation
    fprintf('Loop execution: done 0%% ');

elseif nargin==1,     % customised initialisation
    if ~ischar(tt),
        error('Single input parameter must be a string');
    end
    fprintf([tt 'Loop execution: done 0%% ']);

elseif nargin==2,     % use with FOR loop
    Percent = floor(100*tt/NumFrames);
    PercentOld = floor(100*(tt-1)/NumFrames);
    if PercentOld~=Percent,
        fprintf('\b\b\b\b\b%-4s', [num2str(Percent) '%']);
    end
    if tt==NumFrames, % safe to say that this is the last call to
PrintLoopPC...
        fprintf('\n');
    end
end

else                  % use with WHILE loop
    Percent = floor(100*tt/NumFrames);
    if oldPercent~=Percent,
        fprintf('\b\b\b\b\b%-4s', [num2str(Percent) '%']);
        if tt==NumFrames, % safe to say that this is the last call to
PrintLoopPC...
            fprintf('\n');
        end
    end
end
end
```

Για το αρχείο license.txt

Copyright (c) 2009-2012, Eric A. Lehmann
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE

LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Για το αρχείο ISM_setup.m

```
function [SetupStruc] = ISM_setup()
%ISM_setup Environmental parameters for image-source method simulation
%
% [SetupStruc] = ISM_setup()
%
% This function can be used as a template for the definition of the
% different parameters for an image-source method simulation, typically
% providing inputs to the functions 'ISM_RIR_bank.m' as well as
% 'fast_ISM_RIR_bank.m' (Lehmann & Johansson's ISM implementations). This
% function returns the structure 'SetupStruc' with the following fields:
%
%     Fs: sampling frequency in Hz.
%     room: 1-by-3 vector of enclosure dimensions (in m),
%           [x_length y_length z_length].
%     mic_pos: N-by-3 matrix, [x1 y1 z1; x2 y2 z2; ...] positions of N
%               microphones in the environment (in m).
%     src_traj: M-by-3 matrix, [x1 y1 z1; x2 y2 z2; ...] positions of M
%               source trajectory points in the environment (in m).
%     T20 or T60: scalar value (in s), desired reverberation time.
%     c: (optional) sound velocity (in m/s).
%     abs_weights: (optional) 1-by-6 vector of absorption coefficients weights,
%                 [w_x1 w_x2 w_y1 w_y2 w_z1 w_z2].
%
% The structure field 'c' is optional in the sense that the various
% functions developed in relation to Lehmann & Johansson's ISM
% implementation assume a sound velocity of 343 m/s by default. If defined
% in the function below, the field 'SetupStruc.c' will take precedence and
% override the default value with another setting.
%
% The field 'abs_weight' corresponds to the relative weights of each of the
% six absorption coefficients resulting from the desired reverberation time
% T60. For instance, defining 'abs_weights' as [0.8 0.8 1 1 0.6 0.6] will
% result in the absorption coefficients (alpha) for the walls in the
% x-dimension being 20% smaller compared to the y-dimension walls, whereas
% the floor and ceiling will end up with absorption coefficients 40%
% smaller (e.g., to simulate the effects of a concrete floor and ceiling).
% Note that setting some of the 'abs_weight' parameters to 1 does NOT mean
% that the corresponding walls will end up with a total absorption! If the
% field 'abs_weight' is omitted, the various functions developed in
% relation to Lehmann & Johansson's ISM implementation will set the
% 'abs_weight' parameter to [1 1 1 1 1 1], which will lead to uniform
% absorption coefficients for all room boundaries.
%
% The structure 'SetupStruc' may contain one of the two fields 'T60' or
% 'T20'. T60 corresponds to the time required by the impulse response to
% decay by 60dB, whereas T20 is defined as the time required for the
% impulse response to decay from -5 to -25dB. Simply define either one of
% these fields in the file below. Set this value to 0 for anechoic
```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```
% environments (direct path only).

SetupStruc.Fs = 16000; % sampling frequency in Hz

% SetupStruc.c = 343; % (optional) propagation speed of
acoustic waves in m/s

SetupStruc.room = [3 4 2.5]; % room dimensions in m

SetupStruc.mic_pos = [1.6 1 1.3; % [x y z] positions of a microphone
array in m
1.4 1 1.3;
1.2 1 1.3];

SetupStruc.src_traj = [linspace(1,2,101).' ones(101,1)*3
ones(101,1)*1.7]; % [x y z] positions of source trajectory in m.
%% defines a straight line in front of the mic array, with 101 source
points along the trajectory (1cm distance increment).

SetupStruc.T60 = 0.2; % reverberation time T60, or define a
T20 field instead!
% SetupStruc.T20 = 0.15; % reverberation time T20, or define
a T60 field instead!

SetupStruc.abs_weights = [0.6 0.9 0.5 0.6 1.0 0.8]; % (optional)
weights for the resulting alpha coefficients.
%% simulates a carpeted floor, and sound-absorbing material on the ceiling
and the second x-dimension wall.

%% Uncomment the following for a 3D plot of the above setup:
%
plot3(SetupStruc.src_traj(:,1),SetupStruc.src_traj(:,2),SetupStruc.src_traj
(:,3),'ro-','markersize',4); hold on;
%
plot3(SetupStruc.mic_pos(:,1),SetupStruc.mic_pos(:,2),SetupStruc.mic_pos(:,
3),'ko','markerfacecolor',ones(1,3)*.6);
% axis equal; axis([0 SetupStruc.room(1) 0 SetupStruc.room(2) 0
SetupStruc.room(3)]);
% box on; xlabel('x-axis (m)'); ylabel('y-axis (m)'); zlabel('z-axis (m)');
```

Για το αρχείο ISM_RT_check.m

```
function [medRT,RTvec] = ISM_RT_check(setupstruc,varargin)

%ISM_RT_check Reverberation time analysis for image-method setup
%
% [RT_MED,RT_DATA] = ISM_RT_check(SETUP_STRUC)
% [RT_MED,RT_DATA] = ISM_RT_check(SETUP_STRUC,'arg1',val1,'arg2',val2,...)
%
% This function performs an analysis of the reverberation time that would
% result from image-source computations (using Lehmann & Johansson's
% implementation, see "Prediction of energy decay in room impulse responses
% simulated with an image-source model", J. Acoust. Soc. Am., vol. 124(1),
% pp. 269-277, July 2008) with a specific configuration defined by the
% structure SETUP_STRUC. This structure needs to contain the following
```

```
% fields:
%
%     Fs: sampling frequency (in Hz).
%     room: 1-by-3 vector of enclosure dimensions (in m),
%           [x_length y_length z_length].
%     T20 or T60: scalar value (in s), desired reverberation time.
%     c: (optional) sound velocity (in m/s).
% abs_weights: (optional) 1-by-6 vector of absorption coefficients weights,
%             [w_x1 w_x2 w_y1 w_y2 w_z1 w_z2].
%
% If the field SETUP_STRUC.c is undefined, the function assumes a default
% value of sound velocity of 343m/s.
%
% The field 'abs_weights' corresponds to the relative weights of each of
% the six absorption coefficients resulting from the desired reverberation
% time T60. For instance, defining 'abs_weights' as [1 1 0.8 0.8 0.6 0.6]
% will result in the absorption coefficients (alpha) for the walls in the
% y-dimension being 20% smaller compared to the x-dimension walls, whereas
% the floor and ceiling will end up with absorption coefficients 40%
% smaller (e.g., to simulate the effects of a concrete floor and ceiling).
% If this field is omitted, the parameter 'abs_weight' will default to
% [1 1 1 1 1 1], which leads to uniform absorption coefficients for all
% room boundaries.
%
% The structure SETUP_STRUC may contain one of the two fields 'T60' or
% 'T20'. This function will automatically determine which reverberation
% type is used and analyse the reverberation time accordingly. T20 is
% defined as the time required for the impulse response energy to decay
% from -5 to -25dB, whereas T60 corresponds to the time required by the
% impulse response energy to decay by 60dB.
%
% HOWEVER: it must be noted that in case T60 is used, it is usually
% unfeasible to simulate a room impulse response (RIR) down to -60dB in
% practice in order to compute the RIR's exact T60 (due to numerical
% error and roundoff noise). Consequently, the value of T60 measured by
% this function is instead interpolated using the initial slope of the
% RIR's energy decay curve (EDC). Because a RIR's energy decay curve is
% in general not exactly linear (especially for the case where the RIR
% is computed using non-uniform absorption weights), this will typically
% lead to discrepancies between the measured and the true T60 value in
% the RIR, even if the environmental parameters are set to actually
% achieve a correct T60 value. This function is therefore better suited
% for an analysis of the reverberation time T20, which does not suffer
% from the above drawback (EDC only required down to about -25dB).
%
% In addition, a number of other (optional) parameters can be set using a
% series of 'argument'--value pairs. The following parameters (arguments)
% can be used:
%
%     'PlotRes': set to 1 if plots of intermediate results are desired
%               (execution will be paused!). Defaults to 0.
%     'NumConfig': number of source-receiver configurations to average the
%                 results over. Defaults to 50.
%     'SilentFlag': set to 1 to disable on-screen messages during execution.
%                  Defaults to 0.
%
% This function simulates a number of transfer functions (RIRs) using
% Lehmann & Johansson's implementation of the image-source method (see
% above reference), and measures the resulting ("true") reverberation time
% from the energy decay curve, computed using Schroeder's integration
% method. This process is repeated for a series of randomly selected
```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```
% source--receiver configurations to obtain a statistically representative
% set of measurements. The source and receiver positions are chosen to be
% located within the inner 80% of the room volume (away from the walls),
% and further than 0.75m from each other.
%
% This function returns 'RT_MED', the median reverberation time value (T60
% or T20) over the 'NumConfig' measurements, and 'RT_DATA', the vector of
% individual reverberation time measurements (T60 or T20). If 'SilentFlag'
% is set to 0, the function also reports the results of the reverberation
% time analysis on screen in the Matlab command window.

% Release date: November 2009
% Author: Eric A. Lehmann, Perth, Australia (www.eric-lehmann.com)

% User input variables:
VarList = {'PlotRes'          0;          % set to 1 if plots of intermediate
results desired
          'NumConfig'       50;          % total number of src-rcv
configurations considered
          'SilentFlag'     0};          % set to 1 to disable on-screen
messages during execution
eval(SetUserVars(VarList,varargin)); % set user-definable variables

Fs = setupstruc.Fs;
room = setupstruc.room;
if isfield(setupstruc,'abs_weights'),
    weights = setupstruc.abs_weights;
else
    weights = ones(1,6);
end
if isfield(setupstruc,'c'),
    cc = setupstruc.c;
else
    cc = 343;
end
if isfield(setupstruc,'T60'),
    warning(['The function 'ISM_RT_check' cannot practically measure the
true T60 \n' ...
           'value in room impulse responses. Instead, it measures T60 by
interpolating the \n' ...
           'initial slope of the RIR's energy decay, which might not
deliver accurate \n' ...
           'measurements. Please refer to ISM_RT_check's online help for
more information.%s'], '');
    rtstr = 'T60'; rtval = setupstruc.T60;
    T60 = setupstruc.T60;
    alpha = ISM_AbsCoeff('t60',T60,room,weights,'LehmannJohansson','c',cc);
else
    rtstr = 'T20'; rtval = setupstruc.T20;
    T20 = setupstruc.T20;
    T60 = [];
    alpha = ISM_AbsCoeff('t20',T20,room,weights,'LehmannJohansson','c',cc);
end
beta = sqrt(1-alpha);

RTvec = zeros(1,NumConfig);
if ~SilentFlag, PrintLoopPCw(' [ISM_RT_check] Computing sample TFs (this
may take a while!). '); end
for ii=1:NumConfig,
    if ~SilentFlag, PrintLoopPCw(ii,NumConfig); end
```

```

%==:- Select source/receiver points ==:-
X_src = rand(1,3).*room*.8 + .1*room; % avoid positions close to
walls
X_rcv = rand(1,3).*room*.8 + .1*room;
while norm(X_src-X_rcv)<.75, % choose new points if they are too
close to each other
    X_src = rand(1,3).*room*.8 + .1*room;
    X_rcv = rand(1,3).*room*.8 + .1*room;
end

%==:- Compute TF ==:-
TFcoeffs =
ISM_RoomResp(Fs,beta,rtstr,rtval,X_src,X_rcv,room,'c',cc,'SilentFlag',1,'De
lta_dB',50);
TFlen = length(TFcoeffs);

%==:- Compute EDC ==:-
EDCvec = zeros(1,TFlen);
for nn=1:TFlen,
    EDCvec(nn) = sum(TFcoeffs(nn:end).^2); % Energy decay using
Schroeder's integration method
end
EDCvec = EDCvec/EDCvec(1);
EDCvec(EDCvec==0) = eps;
EDCvec = 10*log10(EDCvec); % Decay curve in dB.

%==:- Measure RT ==:-
timevec = [0:TFlen-1]/Fs;
if isempty(T60), % measure T20
    sind = find(EDCvec>=-5,1,'last');
    deltaX = timevec(2)-timevec(1);
    deltaY = EDCvec(sind+1)-EDCvec(sind);
    deltaA = -5 - EDCvec(sind);
    t1 = timevec(sind) + deltaA*deltaX/deltaY;
    sind = find(EDCvec>=-25,1,'last');
    deltaY = EDCvec(sind+1)-EDCvec(sind);
    deltaA = -25 - EDCvec(sind);
    t2 = timevec(sind) + deltaA*deltaX/deltaY;
    RTvec(ii) = t2-t1;
else % measure T60
    dp_del = norm(X_src-X_rcv)/cc; % measure slope after direct
path
    intstart = find(timevec<=dp_del,1,'last');
    intstop = find(EDCvec>=-45,1,'last'); % find best slope
up to -45dB in the EDC
    numpts = intstop-intstart+1;
    if numpts<=1,
        error('Problem encountered while measuring T60.');
```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```

end
using the error median avoids influence of end part of the curve
foo = find(errorvec==nanmin(errorvec),1,'last'); % T60 estimate
as decay line minimising squared error
RTvec(ii) = 60/abs(slopevec(1,foo))/Fs; % T60 estimate from TF
computations
polyp = slopevec(:,foo).';
slopeind = foo;
end

%==-- Plot results if necessary ==--
if PlotRes,
reusefig('ISM Reverberation Time Analysis (ISM_RT_check.m)'); clf;
ylimmin = -60;
foo = TFcoeffs/sum(TFcoeffs.^2);
plot(timevec,10*log10(foo.^2),'color',[.85 .85 1]); hold on;
plot(timevec,EDCvec,'k');
xlabel('time (s)'); ylabel('EDC (dB)');
if isempty(T60), % measure T20
plot([0 t1 t1],[-5 -5 ylimmin],'k:');
plot([0 t2 t2],[-25 -25 ylimmin],'k:');
tstr = ['TF #' num2str(ii) ' of ' num2str(NumConfig) ' :
measured T_{20} = ' num2str(RTvec(ii),'%.3f') 's'];
else % measure T60
plot(timevec([1 end]),polyval(polyp,[1 TFlen]),'r');

plot(timevec(slopeind),polyval(polyp,slopeind),'color','r','marker','o','ma
rkersize',4);

plot(timevec(intstart),polyval(polyp,intstart),'color','r','marker','o','ma
rkersize',4);
tstr = ['TF #' num2str(ii) ' of ' num2str(NumConfig) ' :
measured T_{60} = ' num2str(RTvec(ii),'%.3f') 's'];
end
axis tight; ylim([ylimmin 5]);
if ii<NumConfig,
title([tstr ' (hit any key to continue)']);
pause; title(tstr); shg; pause(.001);
else
title(tstr);
end
end

end

medRT = median(RTvec);
iqrRT = iqr(RTvec);

%==-- Output results on screen ==--
if ~SilentFlag,
fprintf('\n ==-- ==-- Results from reverberation time analysis
('ISM_RT_check.m') ==-- ==--');
fprintf('\n Simulation parameters: Fs = %.5gHz, c = %.5gm/s, room =
[%.5gm %.5gm %.5gm]',Fs,cc,room(1),room(2),room(3));
fprintf('\n
abs_weights = [%.5g %.5g %.5g %.5g
%.5g] ',weights(1),weights(2),weights(3),weights(4),weights(5),weights(6));
if isempty(T60),
fprintf('\n Desired reverb. time: T20 = %.5gs',T20);
else
fprintf('\n Desired reverb. time: T60 = %.5gs',T60);
end
end

```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```
end
fprintf('\n Reflection coeff. for this setup (determined by Lehmann &
Johansson's method):')
fprintf('\n      beta = [%0.5g %0.5g %0.5g %0.5g %0.5g
%0.5g]',beta(1),beta(2),beta(3),beta(4),beta(5),beta(6));
fprintf(['\n\n Resulting ' rtstr ' measured in room (median value over
%d configurations):'],NumConfig);
fprintf(['\n      ' rtstr ' = %0.5f s (inter-quartile range: %0.5g
s)'],medRT,iqrRT);
if ~isempty(T60),
    fprintf(' ~~~ [T60 measurements may be inaccurate!]);
end
fprintf('\n ==:== ==:== ==:== ==:== ==:== ==:== ==:== ==:== ==:==
==:== ==:== ==:== ==:==\n\n');
end
```

Για το αρχείο ISM_RoomResp.m

```
function [RIRvec] =
ISM_RoomResp(Fs,beta,rt_type,rt_val,X_src,X_rcv,room,varargin)
%ISM_RoomResp RIR based on Lehmann & Johansson's image-source method
%
% RIR = ISM_RoomResp(Fs,BETA,RT_TYPE,RT_VAL,SOURCE,SENSOR,ROOM)
% RIR = ISM_RoomResp( ... , 'arg1',val1, 'arg2',val2,...)
%
% This function generates the room impulse response (RIR) between a sound
% source and an acoustic sensor, based on various environmental parameters
% such as source and sensor positions, enclosure's dimensions and
% reflection coefficients, etc., according to Lehmann and Johansson's
% implementation of the image-source method (see below). The input
% parameters are defined as follows:
%
% Fs: scalar, sampling frequency (in Hz). Eg: 8000.
% BETA: 1-by-6 vector, corresponding to each wall's reflection
% coefficient: [x1 x2 y1 y2 z1 z2]. Index 1 indicates wall
closest
to the origin. This function assumes strictly non-negative BETA
coefficients. Set to [0 0 0 0 0 0] to obtain anechoic response
(direct path only), in which case the value of RT_VAL is
discarded. E.g.: [0.75 0.75 0.8 0.25 0.3 0.9].
% RT_TYPE: character string, measure of reverberation time used for the
% definition of the coefficients in BETA. Set to either 'T60' or
% 'T20'.
% RT_VAL: scalar, value of the reverberation time (in seconds) defined by
% RT_TYPE. Set to 0 to obtain anechoic response (same effect as
% setting BETA to [0 0 0 0 0 0]), in which case the BETA
% coefficients are discarded. E.g.: 0.25.
% SOURCE: 1-by-3 vector, indicating the location of the source in space
% (in m): [x y z]. E.g.: [1 1 1.5].
% SENSOR: 1-by-3 vector, indicating the location of the microphone in
% space (in m): [x y z]. E.g.: [2 2 1.5].
% ROOM: 1-by-3 vector, indicating the rectangular room dimensions
% (in m): [x_length y_length z_length]. E.g.: [4 4 3].
%
% In addition, a number of other (optional) parameters can be set using a
% series of 'argument'--value pairs. The following parameters (arguments)
% can be used:
%
% 'c': scalar, speed of acoustic waves (in m/s). Defaults to 343.
% 'Delta_dB': scalar (in dB), parameter determining how much the
resulting
```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```
% impulse response is cropped (i.e. RIR length): the impulse
% response is computed until the time index where its overall
% energy content has decreased by 'Delta_dB' decibels, after
% which the computations stop. Not relevant if
BETA=zeros(1,6).
% Defaults to 50.
% 'MaxDelay': scalar (in seconds), defines the desired length for the
% computed RIR. If defined as non-empty, this parameter
% overrides the setting of 'Delta_dB'. Use 'MaxDelay' if the
% RIR length is known exactly prior to its computation. Not
% relevant if BETA=zeros(1,6). Defaults to [].
% 'SilentFlag': set to 1 to disable all on-screen messages from this
% function. Defaults to 0.
%
% This function returns the time coefficients of the filter (transfer
% function) in the parameter RIR. The filter coefficients are real and
% non-normalised. The first value in the vector RIR, i.e., RIR(1),
% corresponds to time t=0. The number of coefficients returned is variable
% and results from the value of 'Delta_dB' defined by the user: the filter
% length will be as large as necessary to capture all the relevant
% highest-order reflections.
%
% This implementation uses Lehmann and Johansson's variant (see "Prediction
% of energy decay in room impulse responses simulated with an image-source
% model", J. Acoust. Soc. Am., vol. 124(1), pp. 269-277, July 2008) of
% Allen & Berkley's "Image Method for Efficiently Simulating Small-room
% Acoustics" (J. Acoust. Soc. Am., vol. 65(4), April 1979). This function
% implements a phase inversion for each sound reflection off the room's
% boundaries, which leads to more accurate room impulse responses (when
% compared to RIRs recorded in real acoustic environments). Also, the
% computations make use of fractional delay filters, which allow the
% representation of non-integer delays for all acoustic reflections.
%
% Release date: March 2012
% Author: Eric A. Lehmann, Perth, Australia (www.eric-lehmann.com)
%
% Explanations for the following code -----
% This implementation of the image method principle has been specifically
% optimised for execution speed. The following code is based on the
% observation that for a particular dimension, the delays from the image
% sources to the receiver increases monotonically as the absolute value of
% the image index (m, n, or l) increases. Hence, all image sources whose
% indices are above or equal to a specific limit index (for which the
% received delay is above the relevant cut-off value) can be discarded. The
% following code checks, for each dimension, the delay of each received
% path and automatically determines when to stop, thus avoiding unnecessary
% computations (the amount of TF cropped depends on the 'Delta_dB'
% parameter).
% The resulting number of considered image sources hence automatically
% results from environmental factors, such as the room dimensions, the
% source and sensor positions, and the walls' reflection coefficients. As a
% result, the length of the computed transfer function has an optimally
% minimum length (no extra padding with negligibly small values).
%-----
VarList = {'SilentFlag'      0;          % set to 1 to disable on-screen
messages
          'c'                343;        % sound propagation speed
          'Delta_dB'         50;         % attenuation limit
          'MaxDelay'         []];       % predefined RIR length, overrides
value of 'Delta_dB'
```


Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```

eval (SetUserVars (VarList, varargin));

global RIRvec TimePoints           % not too pretty, but this avoids
passing potentially large         % vectors to frequently called
subfunctions...

%--:=- Check user input:
if X_rcv(1)>=room(1) || X_rcv(2)>=room(2) || X_rcv(3)>=room(3) ||
X_rcv(1)<=0 || X_rcv(2)<=0 || X_rcv(3)<=0,
    error('Receiver must be within the room boundaries!');
elseif X_src(1)>=room(1) || X_src(2)>=room(2) || X_src(3)>=room(3) ||
X_src(1)<=0 || X_src(2)<=0 || X_src(3)<=0,
    error('Source must be within the room boundaries!');
elseif ~isempty(find(beta>=1,1)) || ~isempty(find(beta<0,1)),
    error('Parameter ''BETA'' must be in the range [0...1].');
end

beta = -abs(beta);           % implement phase inversion in Lehmann &
Johansson's ISM implementation

X_src = X_src(:);           % Source location
X_rcv = X_rcv(:);           % Receiver location
beta = beta(:);             % Reflection coefficients
Rr = 2*room(:);             % Room dimensions

%--:=- Calculate maximum time lag to consider in RIR --:=-
if ~isequal(beta(:), zeros(6,1)) && rt_val~=0,           % non-anechoic case:
compute RIR's decay time necessary to reach
    if isempty(MaxDelay),           % Delta_dB (using Lehmann & Johansson's
EDC approximation method)
        MaxDelay = ISM_RIR_Decomposition(Delta_dB,rt_type,rt_val,1-
beta.^2,room,X_src,X_rcv,Fs,c);
    end
else                               % Anechoic case: allow for 5 times
direct path in TF
    DPdel = norm(X_rcv - X_src)/c; % direct path delay in [s]
    MaxDelay = 5*DPdel;
    beta = zeros(6,1);           % in case rt_val=0 only
end
TForder = ceil(MaxDelay*Fs);       % total length of RIR [samp] to reach
Delta_dB

TimePoints = ([0:TForder-1]/Fs).';
RIRvec = zeros(TForder,1);

%--:=- Summation over room dimensions:
if ~SilentFlag, fprintf(' [ISM_RoomResp] Computing transfer function ');
end;
for a = 0:1
    for b = 0:1
        for d = 0:1
            if ~SilentFlag, fprintf('.'); end;

            m = 1;                 % Check delay values for m=1 and above
            FoundLValBelowLim =
Check_lDim(a,b,d,m,X_rcv,X_src,Rr,c,MaxDelay,beta,Fs);
            while FoundLValBelowLim==1,
                m = m+1;

```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```

        FoundLValBelowLim =
Check_lDim(a,b,d,m,X_rcv,X_src,Rr,c,MaxDelay,beta,Fs);
        end

        m = 0;                % Check delay values for m=0 and below
        FoundLValBelowLim =
Check_lDim(a,b,d,m,X_rcv,X_src,Rr,c,MaxDelay,beta,Fs);
        while FoundLValBelowLim==1,
            m = m-1;
            FoundLValBelowLim =
Check_lDim(a,b,d,m,X_rcv,X_src,Rr,c,MaxDelay,beta,Fs);
        end

    end
end
if ~SilentFlag, fprintf('\n'); end;

%=====
function [FoundLValBelowLim] =
Check_lDim(a,b,d,m,X_rcv,X_src,Rr,c,MaxDelay,beta,Fs)

FoundLValBelowLim = 0;

l = 1;                % Check delay values for l=1 and above
FoundNValBelowLim =
Check_nDim(a,b,d,l,m,X_rcv,X_src,Rr,c,MaxDelay,beta,Fs);
while FoundNValBelowLim==1,
    l = l+1;
    FoundNValBelowLim =
Check_nDim(a,b,d,l,m,X_rcv,X_src,Rr,c,MaxDelay,beta,Fs);
end
if l~=1, FoundLValBelowLim = 1; end;

l = 0;                % Check delay values for l=0 and below
FoundNValBelowLim =
Check_nDim(a,b,d,l,m,X_rcv,X_src,Rr,c,MaxDelay,beta,Fs);
while FoundNValBelowLim==1,
    l = l-1;
    FoundNValBelowLim =
Check_nDim(a,b,d,l,m,X_rcv,X_src,Rr,c,MaxDelay,beta,Fs);
end
if l~=0, FoundLValBelowLim = 1; end;

%=====
function [FoundNValBelowLim] =
Check_nDim(a,b,d,l,m,X_rcv,X_src,Rr,c,MaxDelay,beta,Fs)

global RIRvec TimePoints

FoundNValBelowLim = 0;

n = 1;                % Check delay values for n=1 and above
dist = norm( [2*a-1; 2*b-1; 2*d-1].*X_src + X_rcv - Rr.*[n;l;m] );
foo_time = dist/c;
while foo_time<=MaxDelay,    % if delay is below TF length limit for n=1,
    check n=2,3,4...

```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```
    foo_amplitude = prod(beta.^abs([n-a; n; l-b; l; m-d; m])) /
(4*pi*dist);
    RIRvec = RIRvec + foo_amplitude * sinc((TimePoints-foo_time)*Fs);
    n = n+1;
    dist = norm( [2*a-1; 2*b-1; 2*d-1].*X_src + X_rcv - Rr.*[n;l;m] );
    foo_time = dist/c;
end
if n~=1, FoundNValBelowLim = 1; end;

n = 0;          % Check delay values for n=0 and below
dist = norm( [2*a-1; 2*b-1; 2*d-1].*X_src + X_rcv - Rr.*[n;l;m] );
foo_time = dist/c;
while foo_time<=MaxDelay, % if delay is below TF length for n=0, check
n=-1,-2,-3...
    foo_amplitude = prod(beta.^abs([n-a; n; l-b; l; m-d; m])) /
(4*pi*dist);
    RIRvec = RIRvec + foo_amplitude * sinc((TimePoints-foo_time)*Fs);
    n = n-1;
    dist = norm( [2*a-1; 2*b-1; 2*d-1].*X_src + X_rcv - Rr.*[n;l;m] );
    foo_time = dist/c;
end
if n~=0, FoundNValBelowLim = 1; end;
```

Για το αρχείο ISM_RIRpow_approx.m

```
function [amppts1,timepts,okflag] =
ISM_RIRpow_approx(aa,room,cc,timepts,rt_type,rt_val)
%ISM_RIRpow_approx Approximation of ISM RIR power (Lehmann & Johansson's
method)
%
% [P_VEC,T_VEC,OK_FLAG] =
ISM_RIRpow_approx(ALPHA,ROOM,C,T_VEC,RT_TYPE,RT_VAL)
%
% This function returns the predicted values of RIR power in P_VEC (as
% would result from ISM simulations) estimated by means of the EDC
% approximation method described in: "Prediction of energy decay in room
% impulse responses simulated with an image-source model", J. Acoust. Soc.
% Am., vol. 124(1), pp. 269-277, July 2008. The values of P_VEC are
% computed for the time points given as input in T_VEC (in sec), which is
% assumed to contain increasing values of time. The vector T_VEC (and
% corresponding vector P_VEC) will be cropped if the numerical computation
% limits are reached for the higher time values in T_VEC (for which NaNs
% are generated in P_VEC), in which case the output parameter OK_FLAG will
% be set to 0 (1 otherwise).
%
% The environmental setting is defined via the following input parameters:
%
% ALPHA: 1-by-6 vector, corresponding to each wall's absorption
% coefficient: [x1 x2 y1 y2 z1 z2]. Index 1 indicates wall
closest
% to the origin. E.g.: [0.5 0.5 0.45 0.87 0.84 0.32].
% RT_TYPE: character string, measure of reverberation time used for the
% definition of the coefficients in ALPHA. Set to either 'T60' or
% 'T20'.
% RT_VAL: scalar, value of the reverberation time (in seconds) defined by
% RT_TYPE. E.g.: 0.25.
% ROOM: 1-by-3 vector, indicating the rectangular room dimensions
```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```
%           (in m): [x_length y_length z_length]. E.g.: [4 4 3].
%           C: scalar (in m/s), propagation speed of sound waves. E.g.: 343.

% Release date: November 2009
% Author: Eric A. Lehmann, Perth, Australia (www.eric-lehmann.com)

numradpts = length(timepts);
radpts = cc * timepts;           % radius values corresponding to time
points

bxx = ( sqrt(1-aa(1))*sqrt(1-aa(2)) )^(1/room(1));
byy = ( sqrt(1-aa(3))*sqrt(1-aa(4)) )^(1/room(2));
bzz = ( sqrt(1-aa(5))*sqrt(1-aa(6)) )^(1/room(3));

if bxx==byy && byy==bzz,
    intcase = 1;
elseif bxx==byy && bxx~=bzz,
    intcase = 2;
elseif byy==bzz && bzz~=bxx,
    if bzz<bxx,           % coordinate swap x<->z
        foo = bxx; bxx = bzz; bzz = foo;
        intcase = 2;
    else
        intcase = 3;
    end
elseif bxx==bzz && bzz~=byy,
    if bzz<byy,           % coordinate swap y<->z
        foo = byy; byy = bzz; bzz = foo;
        intcase = 2;
    else
        intcase = 4;
    end
else
    intcase = 5;
    if bxx>bzz && bxx>byy,           % coordinate swap x<->z
        foo = bxx; bxx = bzz; bzz = foo;
    elseif byy>bzz && byy>bxx,           % coordinate swap y<->z
        foo = byy; byy = bzz; bzz = foo;
    end
end

amppts1 = zeros(1,numradpts);
for ss=1:numradpts,           % compute amplitude/energy estimates
    Bx = bxx^(radpts(ss)); Bx(Bx==0) = eps;
    By = byy^(radpts(ss)); By(By==0) = eps;
    Bz = bzz^(radpts(ss)); Bz(Bz==0) = eps;
    switch intcase
        case 1
            int2 = Bx;
        case 2
            int2 = (Bx-Bz) / log(Bx/Bz);
        case 3
            n1 = log(Bz/Bx);
            int2 = Bz*( expint(n1) + log(n1) + 0.5772156649 ) / n1;
        case 4
            n1 = log(Bz/By);
            int2 = Bz*( expint(n1) + log(n1) + 0.5772156649 ) / n1;
        otherwise
            n1 = log(Bz/By);
            n2 = log(Bz/Bx);
```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```
        int2 = Bz*(log(n1/n2) + expint(n1) - expint(n2)) / log(Bx/By);
    end
    amppts1(ss) = int2/radpts(ss);          % 'propto' really...
end

okflag = 1;
foo = find(isnan(amppts1),1);
if ~isempty(foo),
    amppts1 = amppts1(1:foo-1);
    timepts = timepts(1:foo-1);
    okflag = 0;
end

if nargin==6,
    switch lower(rt_type) % offset correction
        case 't60', sl = exp(3.05*exp(-1.85*rt_val));
        case 't20', sl = exp(3.52*exp(-7.49*rt_val));
    end
    amppts1 = amppts1 ./ exp(sl*(timepts-timepts(1)));
end
```

Για το αρχείο ISM_RIR_DecayTime.m

```
function DTime_vec =
ISM_RIR_DecayTime(delta_dB_vec,rt_type,rt_val,aa,room,X_src,X_rcv,Fs,cc)

%ISM_RIR_DecayTime RIR decay time using Lehmann & Johansson's EDC
approximation method
%
% DT =
ISM_RIR_DecayTime(DELTA_dB,RT_TYPE,RT_VAL,ALPHA,ROOM,SOURCE,SENSOR,Fs)
% DT =
ISM_RIR_DecayTime(DELTA_dB,RT_TYPE,RT_VAL,ALPHA,ROOM,SOURCE,SENSOR,Fs,C)
%
% This function determines the time DT taken by the energy in a RIR to
% decay by DELTA_dB when using Lehmann & Johansson's image-source method
% implementation (see: "Prediction of energy decay in room impulse
% responses simulated with an image-source model", J. Acoust. Soc. Am.,
% vol. 124(1), pp. 269-277, July 2008). The parameter DELTA_dB can be
% defined as a vector of dB values, in which case this function returns DT
% as a vector containing the corresponding decay times. Note that DT does
% not necessarily correspond to the usual definition of quantities such as
% T20, T30, or T60. The resulting DT values are computed according to
% Lehmann and Johansson's EDC (energy decay curve) approximation method
% (see above reference) used in conjunction with a RIR reconstruction
% method based on diffuse reverberation modeling (see "Diffuse
% reverberation model for efficient image-source simulation of room
% impulse responses", IEEE Trans. Audio, Speech, Lang. Process., 2010).
%
% The environmental room setting is given via the following parameters:
%
%     Fs: scalar, sampling frequency (in Hz). Eg: 8000.
%     ALPHA: 1-by-6 vector, corresponding to each wall's absorption
%            coefficient: [x1 x2 y1 y2 z1 z2]. Index 1 indicates wall
%            closest to the origin. E.g.: [0.5 0.5 0.45 0.87 0.84 0.32].
%     RT_TYPE: character string, measure of reverberation time used for the
%            definition of the coefficients in ALPHA. Set to either 'T60' or
```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```

%           'T20'.
%   RT_VAL: scalar, value of the reverberation time (in seconds) defined by
%           RT_TYPE. E.g.: 0.25.
%   SOURCE: 1-by-3 vector, indicating the location of the source in space
%           (in m): [x y z]. E.g.: [1 1 1.5].
%   SENSOR: 1-by-3 vector, indicating the location of the microphone in
%           space (in m): [x y z]. E.g.: [2 2 1.5].
%   ROOM: 1-by-3 vector, indicating the rectangular room dimensions
%          (in m): [x_length y_length z_length]. E.g.: [4 4 3].
%   C: (optional) scalar (in m/s), propagation speed of sound waves.
%       If omitted, C will default to 343m/s.

% Release date: March 2012
% Author: Eric A. Lehmann, Perth, Australia (www.eric-lehmann.com)

if nargin==8,
    cc = 343;
end
delta_dB_vec = abs(delta_dB_vec);

%--:-- Check user input:
if ~isempty(find(aa>1,1)) || ~isempty(find(aa<=0,1)),
    error('Parameter ''ALPHA'' must be in range (0...1).');
elseif ~isempty(find(delta_dB_vec==0,1)),
    error('Parameter ''DELTA_dB'' must contain non-zero scalar values.');
```

```

elseif min(size(delta_dB_vec))~=1,
    error('Parameter ''DELTA_dB'' must be a 1-D vector.');
```

```

end

if isequal(aa(:),ones(6,1)) || rt_val==0,
    error('ISM_RIR_DecayTime cannot be used for anechoic environments.');
```

```

end

switch lower(rt_type)
    case 't60'
        t60_appval = rt_val;
    case 't20'
        t60_appval = rt_val*3; % coarse t60 estimate to determine end
time in EDCapprox computations
    otherwise
        error('Unknown ''RT_TYPE'' argument.');
```

```

end

%--:-- Pre-processing --:--
dp_del = norm(X_src-X_rcv)/cc; % direct path
delta_dB_max = max(delta_dB_vec);
n_ddbv = length(delta_dB_vec);
starttime = max([1.4*mean(room)/cc dp_del]); % start time t0, ensure >=
dp_delay
starttime_sind = floor(starttime*Fs); % sample index
RIR_start_DTime = 2*starttime;

%--:-- select window size --:--
n_win_meas = 6; % approximate nr of (useful)
measurements
TT = (RIR_start_DTime-starttime)/n_win_meas;
w_len = floor(TT*Fs); % current window length (samples)
w_len = w_len + mod(w_len,2) - 1; % make w_len odd
w_len_half = floor(w_len/2);
```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```

%==:- pre-compute start of RIR for lambda correction ==:-
RIR = ISM_RoomResp(Fs,sqrt(1-
aa),rt_type,rt_val,X_src,X_rcv,room,'SilentFlag',1,'MaxDelay',RIR_start_DTi
me,'c',cc);
RIRlen = length(RIR);

%==:- Measure average energy ==:-
fit_time_perc = 0.35;
we_sind_vec = [starttime_sind+w_len_half:w_len:RIRlen]; % window end
indices
wc_sind_vec = we_sind_vec - w_len_half; % window centre
indices
wb_sind_vec = wc_sind_vec - w_len_half; % window beginning
indices
if wb_sind_vec(1)<=0, wb_sind_vec(1) = 1; end % case where t0 is
less than a half window
n_win_meas = length(wc_sind_vec);
en_vec_meas = zeros(1,n_win_meas);
for ww=1:n_win_meas,
    en_vec_meas(ww) = mean( RIR(wb_sind_vec(ww):we_sind_vec(ww)).^2 );
end
t_vec_meas = wc_sind_vec/Fs;
fit_starttime = RIRlen*fit_time_perc/Fs;
fit_start_wind = find(t_vec_meas>=fit_starttime,1); % window index of
start of fit

%==:- Decay time estimate ==:-
DTime_vec = NaN*ones(size(delta_dB_vec));
stind = 3;
while stind>0,
    % compute + lambda-adjust EDC approximation
    stoptime = stind * delta_dB_max/60*t60_appval; % IMapprox computed
up to several times what linear decay predicts
    timepts = [starttime_sind:w_len:stoptime*Fs]/Fs;
    stoptime = timepts(end);
    [amppts1,timepts,okflag] =
ISM_RIRpow_approx(aa,room,cc,timepts,rt_type,rt_val); % compute EDC
approximation
    foo = en_vec_meas(fit_start_wind:end) ./
amppts1(fit_start_wind:n_win_meas); % offset compensation (lambda)
    amppts1 = amppts1 * mean(foo);

    % reconstruct approx. full RIR for proper EDC estimation (logistic
noise approx.)
    amppts1_rec = interp1(timepts,amppts1,[RIRlen+1:stoptime*Fs]/Fs);
    RIR_rec = [RIR.' sqrt(amppts1_rec)];
    RIR_rec_len = length(RIR_rec);

    % approx. full RIR EDC
    edc_rec = zeros(1,RIR_rec_len);
    for nn=1:RIR_rec_len,
        edc_rec(nn) = sum(RIR_rec(nn:end).^2); % Energy decay using
Schroeder's integration method
    end
    edc_rec = 10*log10(edc_rec/edc_rec(1)); % Decay curve in dB.
    tvec_rec = [0:RIR_rec_len-1]/Fs;

    % Determine time of EDC reaching delta_dB decay:
    if edc_rec(end)>-delta_dB_max,
        stind = stind + 1;

```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```
        if okflag==0, error('Problem computing decay time (parameter
'DELTA_dB' may be too large)'); end
        if stind>=25, error('Problem computing decay time (parameter
'DELTA_dB' may be too large)'); end
        continue
    end

    for nn=1:n_ddbv,
        foo = find(edc_rec<=-delta_dB_vec(nn),1);
        DTime_vec(nn) = 1.15*tvec_rec(foo); % statistical offset
correction...
    end

    if max(DTime_vec)>stoptime*2/3, % make sure IM approx was computed
for more than 3/2 the resulting decay time.
        stind = stind + 1; % increase time if necessary
        if okflag==0, error('Problem computing decay time (parameter
'DELTA_dB' may be too large)'); end
        if stind>=25, error('Problem computing decay time (parameter
'DELTA_dB' may be too large)'); end
        continue
    else
        stind = 0; % stop the computations
    end
end
end
```

Για το αρχείο ISM_RIR_bank.m

```
function [RIR_cell] = ISM_RIR_bank(setupstruc,RIRFileName,varargin)
%ISM_RIR_bank Bank of RIRs using Lehmann & Johansson's image-source method
%
% [RIR_CELL] = ISM_RIR_bank(SETUP_STRUC,RIR_FILE_NAME)
% [RIR_CELL] = ISM_RIR_bank( ... , 'arg1', val1, 'arg2', val2, ...)
%
% This function generates a bank of room impulse responses (RIRs) for a
% particular user-defined room setup, using Lehmann and Johansson's
% implementation of the image-source method (see: "Prediction of energy
% decay in room impulse responses simulated with an image-source model", J.
% Acoust. Soc. Am., vol. 124(1), pp. 269-277, July 2008). The input
% SETUP_STRUC is a structure of enviromental parameters containing the
% following fields:
%
%     Fs: sampling frequency (in Hz).
%     room: 1-by-3 vector of enclosure dimensions (in m),
%           [x_length y_length z_length].
%     mic_pos: N-by-3 matrix, [x1 y1 z1; x2 y2 z2; ...] positions of N
%               microphones (in m).
%     src_traj: M-by-3 matrix, [x1 y1 z1; x2 y2 z2; ...] positions of M
%               source trajectory points (in m).
%     T20 or T60: scalar value (in s), desired reverberation time.
%     c: (optional) sound velocity (in m/s).
%     abs_weights: (optional) 1-by-6 vector of absorption coefficients weights,
%                 [w_x1 w_x2 w_y1 w_y2 w_z1 w_z2].
%
% If the field SETUP_STRUC.c is undefined, the function assumes a default
% value of sound velocity of 343 m/s.
%
```


Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```
% The field 'abs_weight' corresponds to the relative weights of each of the
% six absorption coefficients resulting from the desired reverberation
time.
% For instance, defining 'abs_weights' as [1 1 0.8 0.8 0.6 0.6] will result
% in the absorption coefficients (alpha) for the walls in the y-dimension
% being 20% smaller compared to the x-dimension walls, whereas the floor
% and ceiling will end up with absorption coefficients 40% smaller (e.g.,
% to simulate the effects of a concrete floor and ceiling). If this field
% is omitted, the parameter 'abs_weight' will default to [1 1 1 1 1 1],
% which leads to uniform absorption coefficients for all room boundaries.
%
% The structure SETUP_STRUC may contain one of the two fields 'T60' or
% 'T20'. This function will automatically determine which reverberation
% type is used and compute the desired room absorption coefficients
% accordingly. T20 is defined as the time required for the impulse response
% energy to decay from -5 to -25dB, whereas T60 corresponds to the time
% required by the impulse response energy to decay by 60dB. Setting the
% corresponding field value to 0 achieves anechoic impulse responses
% (direct path only).
%
% In addition, a number of other (optional) parameters can be set using a
% series of 'argument'--value pairs. The following parameters (arguments)
% can be used:
%
%   'Delta_dB': scalar (in dB), parameter determining how much the
resulting
%               impulse response is cropped: the impulse response is
%               computed until the time index where its overall energy
%               content has decreased by 'Delta_dB' decibels, after which
%               the computations stop. Not relevant if the reverberation
%               time is set to 0 (anechoic case). Defaults to 50.
%   'SilentFlag': set to 1 to disable this function's on-screen messages.
%               Defaults to 0.
%
% This function returns a 2-dimensional cell array RIR_CELL containing the
% RIRs for each source trajectory point and each microphone, organised as
% follows: RIR_CELL{mic_index, traj_index}. The resulting filter length
% may differ slightly in each computed RIR.
%
% This function also saves the computation results on file. The argument
% RIR_FILE_NAME determines the name of the .mat file where the variable
% RIR_CELL is to be saved. If a file already exists with the same name as
% the input argument, the user will be prompted to determine whether the
% file is to be overwritten or not. The given parameter RIR_FILE_NAME can
% be a full access path to the desired file. If no access path is given,
% the file is saved in the current working directory.

% Release date: November 2009
% Author: Eric A. Lehmann, Perth, Australia (www.eric-lehmann.com)

VarList = {'Delta_dB'      50;           % maximum attenuation in RIR
computation
           'SilentFlag'   0};          % set to 1 to disable on-screen
messages
eval(SetUserVars(VarList, varargin)); % set user-definable variables

if length(RIRFileName) <= 4 || ~strcmpi(RIRFileName(end-3:end), '.mat'),
    RIRFileName = [RIRFileName '.mat'];
end

if exist(RIRFileName, 'file') == 2,
```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```
    foo = input(' [ISM_RIR_bank] The file name passed as input already
exists. Overwrite? [yes/no]: ', 's');
    if ~strcmpi(foo, 'yes');
        fprintf(' [ISM_RIR_bank] Terminating execution now (no data was
saved).\n');
        return
    end
end

Fs = setupstruc.Fs;
room = setupstruc.room;
micpos = setupstruc.mic_pos;
straj = setupstruc.src_traj;

if isfield(setupstruc, 'abs_weights'),
    weights = setupstruc.abs_weights;
else
    weights = ones(1,6);
end
if isfield(setupstruc, 'c'),
    cc = setupstruc.c;
else
    cc = 343;
end
if isfield(setupstruc, 'T60'),
    alpha =
ISM_AbsCoeff('t60', setupstruc.T60, room, weights, 'LehmannJohansson', 'c', cc);
    rtttype = 'T60'; rtval = setupstruc.T60;
elseif isfield(setupstruc, 'T20'),
    alpha =
ISM_AbsCoeff('t20', setupstruc.T20, room, weights, 'LehmannJohansson', 'c', cc);
    rtttype = 'T20'; rtval = setupstruc.T20;
else
    error('Missing T60 or T20 field. ');
end
beta = sqrt(1-alpha);

nMics = size(micpos,1);    % number of microphones
nSPts = size(straj,1);    % number of source trajectory points

%--:=- Compute RIR bank --:=-
RIR_cell = cell(nMics, nSPts); % pre-allocate cell array
if ~SilentFlag, PrintLoopPCw(' [ISM_RIR_bank] Computing room impulse
responses. '); end;
for mm=1:nMics,
    X_rcv = micpos(mm, :);
    for tt=1:nSPts,          % compute ISM room impulse response for each
source-receiver combinations
        if ~SilentFlag, PrintLoopPCw((mm-1)*nSPts+tt, nMics*nSPts); end;
        X_src = straj(tt, :);
        RIR_cell{mm, tt} =
ISM_RoomResp(Fs, beta, rtttype, rtval, X_src, X_rcv, room, 'SilentFlag', 1, 'c', cc, 'D
elta_dB', Delta_dB);
    end
end

%--:=- Save results into .mat file --:=-
save(RIRFileName, 'RIR_cell');
if ~SilentFlag, fprintf(' [ISM_RIR_bank] RIR bank parameter 'RIR_cell'
saved in file '%s'\n', RIRFileName); end;
```

Για το αρχείο ISM_AudioData.m

```
function [AuData] = ISM_AudioData(RIRFileName,SrcSignal,varargin)
%ISM_AudioData  Creates audio samples from a pre-computed RIR bank
%
% [AUDIO_DATA] = ISM_AudioData(RIR_FILE_NAME, SRC_SIGNAL)
% [AUDIO_DATA] = ISM_AudioData( ... , 'arg1', val1, 'arg2', val2, ...)
%
% This function generates samples of audio data based on a bank of
% pre-computed room impulse responses (RIRs). The input variable
% RIR_FILE_NAME determines the name of the .mat file where the bank of RIRs
% is stored (the .mat file must contain a cell array parameter named
% 'RIR_cell'). The parameter RIR_FILE_NAME can be a full access path to the
% desired file. If no access path is given, the function looks for the
% desired file in the current working directory. The second input parameter
% SRC_SIGNAL corresponds to a one-dimensional vector of signal data emitted
% by the acoustic source. The length of the source audio sample, together
% with the source trajectory defined by the RIRs in RIR_FILE_NAME, define
% the velocity of the speaker across the environment (constant velocity
% motion).
%
% This function returns AUDIO_DATA, the matrix of audio data (non-
% normalised) generated at the receivers. This matrix is arranged so that
% each column represents the signal received at the corresponding sensor.
%
% The specific simulation parameters, such as room dimensions, microphone
% positions, number of microphones, source trajectory, sampling frequency,
% etc., are implicitly defined by the set of RIRs contained in
% RIR_FILE_NAME. It is hence up to the user to check the integrity of the
% resulting audio data by ensuring consistency of the considered source
% signal and the environmental setup parameters that were used to compute
% the impulse responses. For instance, the source signal defined in
% SRC_SIGNAL is assumed to have the same sampling frequency as that defined
% for the RIRs.
%
% In addition, a number of other (optional) parameters can be set using a
% series of 'argument'--value pairs. The following parameters (arguments)
% can be used:
%
% 'AudioFileName': character string defining the name of the file used to
% save the resulting audio data. Can be defined as a
% '.mat' or '.wav' string to save the data in the
% corresponding format. The data is saved as a single
% matrix of multi-channel data, with each column
% containing the data generated for the corresponding
% microphone. If saving as a '.wav' file, the audio data
% is first normalised in order to avoid clipping during
% the wavwrite operation. 'AudioFileName' may contain a
% full access path to the desired file; if no access
% path is given, the resulting audio data is saved in
% the current working directory. If the file name
% already exists, the user will be prompted to overwrite
% it. If this parameter is set to the empty matrix [],
% no data will be saved. Defining 'AudioFileName' as a
% '.wav' file also requires the parameter 'Fs' to be
% defined (see below). Defaults to [].
%
% 'Fs': scalar value of sampling frequency (in Hz). Only
```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```
% required if the audio data is saved as a .wav file.
% Defaults to [].
% 'AddNoiseFlag': set this flag to 1 in order to add white Gaussian
% noise to the resulting sensor data. Defaults to 0.
% 'NoiseSNR': scalar value (in dB), desired SNR level of the
additive
% Gaussian noise if 'AddNoiseFlag' is set to 1
(irrelevant
% otherwise). The SNR is computed as a time average
% across all sensors. Defaults to 20.
% 'TrajDir': direction of the source along the trajectory path. Can
% be defined as one of the following strings (value):
% * 'SE': the source follows the path defined by the
% RIRs in RIR_FILE_NAME from Start to End
% * 'ES': the source follows the path backwards, from
% End to Start
% * 'SES': the source follows the path from the Start
% to End point, then returns back to the
% Start position
% * 'ESE': the source follows the path backwards from
% the End to the Start point, then returns
% back to the End position
% * 'SMS': the source follows half the path from the
% Start to Middle point, then returns back to
% the Start position
% * 'EME': the source follows half the path backwards
% from the End to Middle point, then returns
% back to the End position
% Default is 'SE'.
% 'TruncateMicSig': due to the effects of reverberation, the resulting
% sensor signals will typically end up slightly longer
% than the source signal in SRC_SIGNAL. The parameter
% 'TruncateMicSig' can be set to 1 in order to truncate
% the resulting audio data to the same length as the
% input signal, thereby cropping out the reverberation
% effects created by the room on the last few samples
% of input signal. Defaults to 1.
% 'SilentFlag': set to 1 to disable this function's on-screen
messages.
% Defaults to 0.
%
% This function creates the audio data by splitting the source data in
% SRC_SIGNAL into as many (non-overlapping) frames as the number of source
% trajectory points. For a given sensor, each frame of signal is then
% convolved with the impulse response for the corresponding trajectory
% location, and the convolution results are then combined additively to
% generate the microphone signal. This process is repeated for each
% trajectory point and each microphone (overlap-add method). This leads to
% the implicit approximation that the source remains stationary during each
% frame. Also note that some clicking noise might result in the audio data
% if the RIRs differ too much from one trajectory point to the next (to a
% given sensor). It is thus up to the user to ensure that trajectory points
% are close enough to each other so as to minimise the resulting
% discrepancies between consecutive RIRs at each microphone.

% Release date: August 2008
% Author: Eric A. Lehmann, Perth, Australia (www.eric-lehmann.com)

VarList = {'AudioFileName'    [];          % file name to save audio data
           'Fs'                [];         % sampling frequency if data to
be saved as .wav file
```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```

        'AddNoiseFlag'      0;           % Set to 0 if you don't want
additive noise
        'NoiseSNR'         20;          % Desired SNR in the resulting
signals (in dB)
        'TrajDir'          'SE';        % direction of the source along
the trajectory
        'TruncateMicSig'   1;           % truncate mic signals to same
length as input signal
        'SilentFlag'       0};         % set to 1 for silent
behaviour.
eval(SetUserVars(VarList,varargin)); % set user-definable variables

if min(size(SrcSignal))~=1,
    error('Source signal must be one-dimensional (single channel).');
end
SrcSignal = SrcSignal(:); % make sure vector is column vector

if ~isempty(AudioFileName), % check save file name
    if length(AudioFileName)<=4 || (~strcmpi(AudioFileName(end-
3:end),'.mat') && ~strcmpi(AudioFileName(end-3:end),'.wav')),
        AudioFileName = [AudioFileName '.mat']; % if not specified,
save data as .mat file
    end
    if strcmpi(AudioFileName(end-3:end),'.wav') && isempty(Fs),
        error('Sampling frequency ''Fs'' must be defined to be able to save
audio data as .wav file.');
```

```

    end
    if exist(AudioFileName,'file')==2,
        foo = input(' [ISM_AudioData] The audio file name already exists.
Overwrite? [yes/no]: ','s');
        if ~strcmpi(foo,'yes');
            fprintf('                Please call this function again with
a different file name argument.\n');
            return
        end
    end
end

load(RIRFileName); % load RIR bank 'RIR_cell' from file

nSamp = length(SrcSignal); % total number of samples in the audio data
nMics = size(RIR_cell,1); % number of mics
nFrames = size(RIR_cell,2); % number of trajectory points

if strcmpi(TrajDir,'ES'),
    RIR_cell = RIR_cell(:,end:-1:1);
elseif strcmpi(TrajDir,'SMS'),
    RIR_cell = RIR_cell(:, [1:floor(nFrames/2) ceil(nFrames/2):-1:1]);
elseif strcmpi(TrajDir,'EME'),
    RIR_cell = RIR_cell(:, [end:-1:ceil(nFrames/2+1)
floor(nFrames/2+1):end]);
elseif strcmpi(TrajDir,'SES'),
    RIR_cell = RIR_cell(:, [1:end-1 end:-1:1]);
    nFrames = 2*nFrames-1;
elseif strcmpi(TrajDir,'ESE'),
    RIR_cell = RIR_cell(:, [end:-1:1 2:end]);
    nFrames = 2*nFrames-1;
end

```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```

nSampPerFrame = ceil(nSamp/nFrames); % total number of frames in the
source signal, last frame might be partially filled

% pre-determine max number of samples in the resulting mic signals (RIR
lengths are variable!):
maxEndPt = 0;
for tt=1:nFrames,
    if tt==nFrames,
        FrameStopInd = nSamp;
    else
        FrameStopInd = tt*nSampPerFrame;
    end
    for mm=1:nMics,
        RIRlen = length(RIR_cell{mm,tt}); % length of current RIR can
be variable!
        maxEndPt = max(maxEndPt,FrameStopInd+RIRlen-1); % endpoint of
last source audio sample in current frame convolved with RIR
    end
end

%==:- Compute audio data ==:-
AuData = zeros(maxEndPt,nMics);
if ~SilentFlag, PrintLoopPCw(' [ISM_AudioData] Computing audio data. ');
end;
for tt=1:nFrames,
    FrameStartInd = (tt-1)*nSampPerFrame+1; % Start/end indices of the
current frame in the overall audio sample.
    if tt==nFrames,
        FrameStopInd = nSamp;
    else
        FrameStopInd = tt*nSampPerFrame;
    end
    FrameData = SrcSignal(FrameStartInd:FrameStopInd); % get one frame of
data
    for mm=1:nMics, % Compute the received signal by convolving the
source signal with the RIR
        if ~SilentFlag, PrintLoopPCw((tt-1)*nMics+mm,nFrames*nMics); end;
        hh = RIR_cell{mm,tt};
        RIRlen = length(hh); % length of current RIR (can be variable!)
        EndIndex = FrameStopInd+RIRlen-1; % max length for the current
convolution.
        AuData(FrameStartInd:EndIndex,mm) = AuData(FrameStartInd:EndIndex,mm)
+ freq_conv(hh,FrameData);
    end
end

%==:- Truncate signals ==:-
if TruncateMicSig==1,
    AuData = AuData(1:nSamp,:);
end

%==:- Additive random noise ==:-
if AddNoiseFlag==1,
    av_pow = mean( sum(AuData.^2,1)/size(AuData,1) ); % Average mic
power across all received signals.
    sigma_noise = sqrt( av_pow/(10^(NoiseSNR/10)) ); % st. dev. of
white noise component to achieve desired SNR.
    AuData = AuData + sigma_noise*randn(size(AuData)); % Add some
random noise
end

```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```
%-:=- Save results to file -:=-
if ~isempty(AudioFileName),
    if strcmpi(AudioFileName(end-3:end),'.mat')
        save(AudioFileName,'AuData');
        if ~SilentFlag, fprintf(' [ISM_AudioData] Audio data 'AuData'
saved in file '%s'\n',AudioFileName); end;
    else strcmpi(AudioFileName(end-3:end),'.wav')),
        foo = max(max(abs(AuData)));
        AuData = AuData*0.99/foo; % avoid clipping with wavwrite
        wavwrite(AuData,Fs,AudioFileName);
        if ~SilentFlag, fprintf(' [ISM_AudioData] Audio data saved in file
'%s'\n',AudioFileName); end;
    end
end
```

Για το αρχείο ISM_AbsCoeff.m

```
function [out,OKflag] = ISM_AbsCoeff(rttype,rt,room,weight,method,varargin)

%ISM_AbsCoeff Calculates absorption coefficients for a given reverberation
time
%
% [ALPHA,OKFLAG] = ISM_AbsCoeff(RT_TYPE,RT_VAL,ROOM,ABS_WEIGHT,METHOD)
% [ALPHA,OKFLAG] = ISM_AbsCoeff( ... , 'c',SOUND_SPEED_VAL)
%
% Returns the six absorption coefficients in the vector ALPHA for a given
% vector of room dimensions ROOM and a given value RT_VAL of reverberation
% time, with RT_TYPE corresponding to the desired measure of reverberation
% time, i.e., either 'T60' or 'T20'. Calling this function with RT_VAL=0
% simply returns ALPHA=[1 1 1 1 1 1] (anechoic case), regardless of the
% settings of the other input parameters.
%
% The parameter ABS_WEIGHTS is a 6 element vector of absorption
% coefficients weights which adjust the relative amplitude ratios between
% the six absorption coefficients in the resulting ALPHA vector. This
% allows the simulation of materials with different absorption levels on
% the room boundaries. Leave empty or set ABS_WEIGHTS=ones(1,6) to obtain
% uniform absorption coefficients for all room boundaries.
%
% If the desired reverberation time could not be reached with the desired
% environmental setup (i.e., practically impossible reverberation time
% value given ROOM and ABS_WEIGHTS), the function will issue a warning on
% screen accordingly. If the function is used with two output arguments,
% the on-screen warnings are disabled and the function sets the flag OKFLAG
% to 0 instead (OKFLAG is set to 1 if the computations are successful).
%
% The returned coefficients are calculated using one of the following
% methods, defined by the METHOD parameter:
%
% * Lehmann and Johansson (METHOD='LehmannJohansson')
% * Sabine (METHOD='Sabine')
% * Norris and Eyring (METHOD='NorrisEyring')
% * Millington-Sette (METHOD='MillingtonSette')
% * Fitzroy (METHOD='Fitzroy')
% * Arau (METHOD='Arau')
% * Neubauer and Kostek (METHOD='NeubauerKostek')
```

```
%  
% In case the first computation method is selected (i.e., if METHOD is set  
% to 'LehmannJohansson'), this function also accepts an additional  
% (optional) argument 'c', which will set the value of the sound wave  
% propagation speed to SOUND_SPEED_VAL. If omitted, 'c' will default to 343  
% m/s. This parameter has no influence on the other six computation  
% methods.  
%  
% Lehmann & Johansson's method relies on a numerical estimation of the  
% energy decay in the considered environment, which leads to accurate RT  
% prediction results. For more detail, see: "Prediction of energy decay in  
% room impulse responses simulated with an image-source model", J. Acoust.  
% Soc. Am., vol. 124(1), pp. 269-277, July 2008. The definition of T20 used  
% with the 'LehmannJohansson' method corresponds to the time required by  
% the energy--time curve to decay from -5 to -25dB, whereas the definition  
% of T60 corresponds to the time required by the energy--time curve to  
% decay by 60dB from the time lag of the direct path in the transfer  
% function.  
%  
% On the other hand, the last six calculation methods are based on various  
% established equations that attempt to predict the physical reverberation  
% time T60 resulting from given environmental factors. These methods are  
% known to provide relatively inaccurate results. If RT_TYPE='T20', the  
% value of T20 for these methods then simply corresponds to T60/3 (linear  
% energy decay assumption). For more information, see: "Measurement of  
% Absorption Coefficients: Sabine and Random Incidence Absorption  
% Coefficients" in the online room acoustics teaching material "AEOF3/AEOF4  
% Acoustics of Enclosed Spaces" by Y.W. Lam, The University of Salford,  
% 1995, as well as the paper: "Prediction of the Reverberation Time in  
% Rectangular Rooms with Non-Uniformly Distributed Sound Absorption" by R.  
% Neubauer and B. Kostek, Archives of Acoustics, vol. 26(3), pp. 183-202,  
% 2001.  
  
% Release date: November 2009  
% Author: Eric A. Lehmann, Perth, Australia (www.eric-lehmann.com)  
  
VarList = {'c'          343}; % default sound propagation speed  
eval(SetUserVars(VarList,varargin));  
  
if ~strcmpi(rttype,'t60') && ~strcmpi(rttype,'t20'),  
    error('Unrecognised ''RT_TYPE'' parameter (must be either ''T60'' or  
    ''T20'').');  
end  
  
if rt==0,  
    out = ones(size(weight));  
    OKflag = 1;  
    return  
end  
  
if isempty(weight),  
    weight = ones(1,6);  
else  
    weight = weight./max(weight);  
end  
  
if strcmpi(method,'sabine'),  
    if strcmpi(rttype,'t20'), rt = 3*rt; end % linear energy decay  
assumption  
    out = fminbnd(@sabine, 0.0001, 0.9999, [], rt, room, weight);
```



```
elseif strcmpi(method,'norriseyring'),
    if strcmpi(rttype,'t20'), rt = 3*rt; end           % linear energy decay
assumption
    out = fminbnd(@norriseyring, 0.0001, 0.9999, [], rt, room, weight);
elseif strcmpi(method,'millingtonsette'),
    if strcmpi(rttype,'t20'), rt = 3*rt; end           % linear energy decay
assumption
    out = fminbnd(@millington_sette, 0.0001, 0.9999, [], rt, room, weight);
elseif strcmpi(method,'fitzroy'),
    if strcmpi(rttype,'t20'), rt = 3*rt; end           % linear energy decay
assumption
    out = fminbnd(@fitzroy, 0.0001, 0.9999, [], rt, room, weight);
elseif strcmpi(method,'arau'),
    if strcmpi(rttype,'t20'), rt = 3*rt; end           % linear energy decay
assumption
    out = fminbnd(@arau, 0.0001, 0.9999, [], rt, room, weight);
elseif strcmpi(method,'neubauerkostek'),
    if strcmpi(rttype,'t20'), rt = 3*rt; end           % linear energy decay
assumption
    out = fminbnd(@neubauer_kostek, 0.0001, 0.9999, [], rt, room, weight);
elseif strcmpi(method,'lehmannjohansson'),
    if strcmpi(rttype,'t20'),
        out = fminbnd(@lehmann_johansson_20, 0.0001, 0.9999, [], rt, room,
weight, c);
    else
        out = fminbnd(@lehmann_johansson_60, 0.0001, 0.9999, [], rt, room,
weight, c);
    end
else
    error('Unrecognised ''METHOD'' parameter (see help for a list of
accepted methods).');
end

foo = optimset('fminbnd');
tolx = foo.TolX;
if out<.0001+3*tolx,
    if nargout<2,
        warning(['Some absorption coefficients are close to the allowable
limits (alpha->0). The \n' ...
                'resulting reverberation time might end up lower than
desired for the given environmental \n' ...
                'setup. Try to relax some environmental constraints so
that the desired reverberation time \n' ...
                'is physically achievable (e.g., by increasing the room
volume, increasing the maximum gap \n' ...
                'between the absorption weights, or decreasing the desired
RT value).%s'], '');
    end
    OKflag = 0;
elseif out>.9999-3*tolx,
    if nargout<2,
        warning(['Some absorption coefficients are close to the allowable
limits (alpha->1). The \n' ...
                'resulting reverberation time might end up higher than
desired for the given environmental \n' ...
                'setup. Try to relax some environmental constraints so
that the desired reverberation time \n' ...
                'is physically achievable (e.g., by reducing the room
volume, reducing the maximum gap \n' ...
                'between the absorption weights, or increasing the desired
RT value).%s'], '');
    end
end
```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```
end
OKflag = 0;
else
    OKflag = 1;
end
out = weight*out;
```

%===== Sabine's formula

```
function err=sabine(a,rt,room,weight)
```

```
alpha = a*weight;
V = prod(room); % Room volume
Sx = room(2)*room(3); % Wall surface X
Sy = room(1)*room(3); % Wall surface Y
Sz = room(1)*room(2); % Wall surface Z
A = Sx*(alpha(1) + alpha(2)) + Sy*(alpha(3) + alpha(4)) + Sz*(alpha(5) +
alpha(6));
err = abs(rt - 0.161*V/A);
```

%===== Millington-Sette's formula

```
function err=millington_sette(a,rt,room,weight)
```

```
alpha = a*weight;
V = prod(room); % Room volume
Sx = room(2)*room(3); % Wall surface X
Sy = room(1)*room(3); % Wall surface Y
Sz = room(1)*room(2); % Wall surface Z
A = -(Sx*(log(1-alpha(1)) + log(1-alpha(2))) + ...
      Sy*(log(1-alpha(3)) + log(1-alpha(4))) + ...
      Sz*(log(1-alpha(5)) + log(1-alpha(6))));
err = abs(rt - 0.161*V/A);
```

%===== Norris and Eyring's formula

```
function err=norris_eyring(a,rt,room,weight)
```

```
alpha = a*weight;
V = prod(room); % Room volume
Sx = room(2)*room(3); % Wall surface X
Sy = room(1)*room(3); % Wall surface Y
Sz = room(1)*room(2); % Wall surface Z
St = 2*Sx+2*Sy+2*Sz; % Total wall surface
A = Sx*(alpha(1) + alpha(2)) + Sy*(alpha(3) + alpha(4)) + Sz*(alpha(5) +
alpha(6));
am = 1/St*A;
err = abs(rt + 0.161*V/(St*log(1-am)));
```

%===== Fitzroy's approximation

```
function err=fitzroy(a,rt,room,weight)
```

```
alpha = a*weight;
V = prod(room); % Room volume
Sx = room(2)*room(3); % Wall surface X
Sy = room(1)*room(3); % Wall surface Y
Sz = room(1)*room(2); % Wall surface Z
St = 2*Sx+2*Sy+2*Sz; % Total wall surface
```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```
tx = -2*Sx/log(1-mean(alpha(1:2)));
ty = -2*Sy/log(1-mean(alpha(3:4)));
tz = -2*Sz/log(1-mean(alpha(5:6)));
err = abs(rt - 0.161*V/(St^2)*(tx+ty+tz));
```

%===== Arau's formula

```
function err=arau(a,rt,room,weight)
```

```
alpha = a*weight;
V = prod(room); % Room volume
Sx = room(2)*room(3); % Wall surface X
Sy = room(1)*room(3); % Wall surface Y
Sz = room(1)*room(2); % Wall surface Z
St = 2*Sx+2*Sy+2*Sz; % Total wall surface
Tx = (0.161*V/(-St*log(1-mean(alpha(1:2)))))^(2*Sx/St);
Ty = (0.161*V/(-St*log(1-mean(alpha(3:4)))))^(2*Sy/St);
Tz = (0.161*V/(-St*log(1-mean(alpha(5:6)))))^(2*Sz/St);
err = abs(rt - (Tx*Ty*Tz));
```

%===== Neubauer and Kostek's formula

```
function err=neubauer_kostek(a,rt,room,weight)
```

```
V = prod(room); % Room volume
Sx = room(2)*room(3); % Wall surface X
Sy = room(1)*room(3); % Wall surface Y
Sz = room(1)*room(2); % Wall surface Z
St = 2*Sx+2*Sy+2*Sz; % Total wall surface
r = 1 - a*weight;
rww = mean(r(1:4));
rcf = mean(r(5:6));
rb = mean(r);
aww = log(1/rb) + (r(1)*(r(1)-rww)*Sx^2 + r(2)*(r(2)-rww)*Sx^2 + ...
    r(3)*(r(3)-rww)*Sy^2 + r(4)*(r(4)-rww)*Sy^2)/(rww*(2*Sx+2*Sy)^2);
acf = log(1/rb) + (r(5)*(r(5)-rcf)*Sz^2 + r(6)*(r(6)-
    rcf)*Sz^2)/((rcf*2*Sz)^2);
err = abs(rt - 0.32*V/(St^2) * (room(3)*(room(1)+room(2))/aww +
    room(1)*room(2)/acf));
```

%===== Lehmann & Johannsson's EDC approximation method

```
function err=lehmann_johansson_60(a,t60,room,weight,cc)
```

```
starttime = 1.4*mean(room)/cc; % start time t0
DPTtime = mean(room)/cc; % direct path "estimate"
aa = a*weight;

numradpts = 60;
stoptime = 2*t60;
while 1, % loop to determine appropriate stop time
    timepts = linspace(starttime,stoptime,numradpts); % time points where
    to compute data

    [amppts1,timepts,okflag] = ISM_RIRpow_approx(aa,room,cc,timepts);

    for ii=1:length(amppts1),
        amppts1(ii) = sum(amppts1(ii:end));
    end
```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```

amppts1 = 10*log10( amppts1/amppts1(1) );

if amppts1(end)>=-60,
    if okflag==0,
        error('Problem computing EDC approximation!');
    end
    numradpts = numradpts + 30;      % more points are required for
accurate T60 estimate
    stoptime = stoptime + t60;
    continue
end

sind = find(amppts1>=-60,1,'last');
deltaX = timepts(2)-timepts(1);
deltaY = amppts1(sind+1)-amppts1(sind);
deltaA = -60 - amppts1(sind);
t2 = timepts(sind) + deltaA*deltaX/deltaY;

if t2>stoptime*2/3,
    numradpts = numradpts + 30;      % more points are required for
accurate T60 estimate
    stoptime = stoptime + t60;
    if okflag==0,
        %error('Problem computing EDC approximation!');
        break % use current time point if numerical limit is reached
    end
    continue
else
    break
end
end

t60est = t2-DPtime;
err = abs(t60 - t60est);

%===== Lehmann & Johannson's EDC approximation method
function err=lehmann_johansson_20(a,t20,room,weight,cc)

starttime = 1.4*mean(room)/cc;      % start time t0
aa = a*weight;

numradpts = 40;
stoptime = 5*t20;
while 1, % loop to determine appropriate stop time
    timepts = linspace(starttime,stoptime,numradpts); % time points where
to compute data

    [amppts1,timepts,okflag] = ISM_RIRpow_approx(aa,room,cc,timepts);

    for ii=1:length(amppts1),
        amppts1(ii) = sum(amppts1(ii:end));
    end
    amppts1 = 10*log10( amppts1/amppts1(1) );

    if amppts1(end)>=-25,
        if okflag==0,
            error('Problem computing EDC approximation!');
        end
end

```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```
        numradpts = numradpts + 30;          % more points are required for
accurate T20 estimate
        stoptime = stoptime + 3*t20;
        continue
    end

    sind = find(amppts1>=-5,1,'last');
    deltaX = timepts(2)-timepts(1);
    deltaY = amppts1(sind+1)-amppts1(sind);
    deltaA = -5 - amppts1(sind);
    t1 = timepts(sind) + deltaA*deltaX/deltaY;

    sind = find(amppts1>=-25,1,'last');
    deltaY = amppts1(sind+1)-amppts1(sind);
    deltaA = -25 - amppts1(sind);
    t2 = timepts(sind) + deltaA*deltaX/deltaY;

    if t2>stoptime*2/3,
        numradpts = numradpts + 30;          % more points are required for
accurate T20 estimate
        stoptime = stoptime + 3*t20;
        if okflag==0,
            %error('Problem computing EDC approximation!');
            break % use current time point if numerical limit is reached
        end
        continue
    else
        break
    end
end

t20est = t2-t1;
err = abs(t20 - t20est);
```

Για το αρχείο freq_conv.m

```
function rr = freq_conv(xx,yy)
%freq_conv Performs convolution in frequency domain
%
% C = freq_conv(X,Y)
%
% This function computes the convolution of X and Y in the frequency domain
% and returns the result in C. Both X and Y must be one-dimensional (line
% or column) vectors. The format of the output C (i.e., line or column
% vector) matches that of the vector X.
%
% This function allows significant savings in execution time compared to
% the time-domain equivalent, i.e., Matlab's conv function.

% Release date: August 2008
% Author: Eric A. Lehmann, Perth, Australia (www.eric-lehmann.com)

xxs = size(xx); yys = size(yy);
if min(xxs)~=1 || min(yys)~=1,
    error('Both input vectors must be one-dimensional.');
```

```
end
```

Εφαρμογές αλγορίθμων μηχανικής μάθησης (machine learning) στη χωρική ανάλυση (spatial analysis) της κρουστικής απόκρισης (impulse response) χώρου ακρόασης

```
xx = xx(:); yy = yy(:);
rlen = length(xx)+length(yy)-1;
rlen_p2 = 2^nextpow2(rlen);
XX = fft(xx,rlen_p2);
YY = fft(yy,rlen_p2);
rr = ifft(XX.*YY,'symmetric');
rr = rr(1:rlen); %column vector

if xxs(1)==1, % output rr in same format as xx
    rr = rr.';
end
```

Για το αρχείο fconv.m

```
function [y] = fconv(x, h)

%FCONV Fast Convolution
% [y] = FCONV(x, h) convolves x and h, and normalizes the output
% to +-1.
%
% x = input vector
% h = input vector
%
% See also CONV
%
% NOTES:
%
% 1) I have a short article explaining what a convolution is. It
% is available at http://stevem.us/fconv.html.
%
%Version 1.0
%Coded by: Stephen G. McGovern, 2003-2004.

Ly = length(x) + length(h) - 1; %
Ly2 = pow2(nextpow2(Ly)); % Find smallest power of 2 that is > Ly

X = fft(x, Ly2); % Fast Fourier transform
H = fft(h, Ly2); % Fast Fourier transform

Y = X.*H; %

y = real(ifft(Y, Ly2)); % Inverse fast Fourier transform
y = y(1:1:Ly); % Take just the first N elements
y = y / max(abs(y)); % Normalize the output
```
