Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Κατάτμηση Σημάτων Φωνής και Εξαγωγή Θεμελιωδών Συχνοτήτων σε Ενσωματωμένη Πλατφόρμα**
Ονοματεπώνυμο Φοιτητή	**Κωστελίδης Βασίλειος**
Πατρώνυμο	**Ηλίας**
Αριθμός Μητρώου	**ΜΠΠΛ/07005**
Επιβλέπων	**Μιχάλης Ψαράκης**

Ημερομηνία Παράδοσης **Απρίλιος 2011**

Τριμελής Εξεταστική Επιτροπή

Μιχάλης Ψαράκης	**Άγγελος Πικράκης**	**Δημήτριος Γκιζόπουλος**
Επίκουρος Καθηγητής	**Λέκτορας**	**Αναπληρωτής Καθηγητής**

Περίληψη

Δυο σημαντικά θέματα της ψηφιακής επεξεργασίας ανθρώπινης φωνής είναι η κατάτμηση σημάτων φωνής ώστε να ανιχνευθεί ομιλία και η εξαγωγή των θεμελιωδών συχνοτήτων της φωνής ενός ομιλητή.

Ένα σύστημα κατάτμησης φωνής με σκοπό ανίχνευση ομιλίας (Voice Activity Detector – VAD) μπορεί να χρησιμοποιηθεί σε τηλεφωνικά κέντρα, σε συστήματα ασφαλείας, σε επαγγελματικά συστήματα τραγουδιστών, σε μεγάλο αριθμό ηλεκτρονικών παιχνιδιών και σε διάφορα άλλα προγράμματα.

Η θεμελιώδης συχνότητα της ανθρώπινης φωνής (pitch) είναι από τα πιο σημαντικά χαρακτηριστικά της. Είναι ο ρυθμός με τον οποίο δονούνται οι φωνητικές χορδές κατά την ανθρώπινη ομιλία. Η εξαγωγή της θεμελιώδους συχνότητας (pitch extraction) μπορεί να χρησιμοποιηθεί για να αναγνωρίζονται ομιλητές σε συστήματα ασφαλείας, για την ανίχνευση της συναισθηματικής τους κατάστασης (emotion detection) [1], για τη διόρθωση της ίδιας της συχνότητας (pitch correction) σε περιπτώσεις τραγουδιστών, για την προπόνηση επαγγελματιών τραγουδιστών, για ηλεκτρονικά παιχνίδια, για σύνθεση ήχου και για αρκετές άλλες εφαρμογές. Οι αλγόριθμοι εξαγωγής pitch είναι πολλοί. Για διαφορετικές εφαρμογές υπάρχουν αλγόριθμοι με πλεονεκτήματα και μειονεκτήματα πάνω στην ακρίβεια και τον χρόνο εκτέλεσης.

Και τα δυο αυτά προβλήματα, η ακριβής κατάτμηση των σημάτων φωνής, σε ομιλία και θόρυβο και η εξαγωγή του pitch συμβάλλουν στην αναγνώριση λέξεων και στην αναγνώριση φωνής του ομιλητή. Για την κατάτμηση τμημάτων φωνής, η επιλογή ενός αλγορίθμου είναι εύκολη, είναι ο γνωστός αλγόριθμος του Rabiner [2][3]. Για τη ανίχνευση pitch μελετήθηκαν μερικοί αλγόριθμοι μέσα από βιβλιογραφία και συγκρίθηκαν τα βασικά τους χαρακτηριστικά. Δυο από αυτούς επιλέχθηκαν για υλοποίηση.

Για τους σκοπούς της πτυχιακής αυτής, ένα ενσωματωμένο σύστημα με έναν 8 – bit μικροελεγκτή προγραμματίστηκε με τους προαναφερθέντες αλγόριθμους για την ανίχνευση ομιλίας και εξαγωγή pitch.

Abstract

Two important issues regarding human speech processing is the detection of the presence of a spoken word and the pitch extraction of a speaker's voice.

A system that detects if a word has been spoken (Voice Activity Detector – VAD) can be used in telephone centers, security systems, professional singing systems, in a large number of computer games and in many other applications.

Pitch is one of the most distinguished characteristics of human voice. It is the rhythm by witch the vocal chords are vibrating during speech. A pitch extractor can be used for voice recognition in security systems, for the emotion detection [1] of a given speaker, the correction of the pitch in singers, training of singers, video games, speech synthesis etc. The existing pitch extraction algorithms are many. For different applications there are algorithms with advantages and disadvantages regarding the accuracy and the execution time.

Both, an accurate voice activity detection and pitch detection in a sound system, are decisive elements for word recognition and speaker recognition. For the voice activity ase, the choice of the algorithm is easy, it is the well known algorithm from Rabiner [2][3]. For the pitch extraction case, some algorithms were studied through bibliography and their basic characteristics were compared. Two of them were selected in order to be implemented.

For the purposes of this thesis, an embedded system with an 8 – bit microcontroller was loaded with the fore mentioned algorithms that perform VAD and pitch detection.

Πίνακας περιεχομένων

Λίστα εικόνων

1 Εισαγωγή

1.1 Ο ρόλος της τεχνολογίας

Η πρόοδος της τεχνολογίας και του αυτοματισμού έχει στόχο να ανεβάσει τα επίπεδα διαβίωσης του ανθρώπου. Αναπτύσσοντας συστήματα που εξαρτώνται όλο και λιγότερο από τον άνθρωπο για την πραγματοποίηση σημαντικών εργασιών, ελαχιστοποιείται το ρίσκο που εισάγεται από πιθανούς κινδύνους. Από υλοποιήσεις στην απλή καθημερινή ζωή, όπως ένα αυτόματο σύστημα που πουλάει αναψυκτικά και την αυτοματοποίηση των τηλεφωνικών κέντρων, ως την λειτουργία πολύπλοκων ιατρικών οργάνων με συστήματα εγχείρησης με laser που πλέον δε βασίζονται στο χειρουργικό νυστέρι, όλα έχουν έναν σκοπό, μεγαλύτερη ασφάλεια από λάθη στον ελάχιστο χρόνο.

1.2 Αλληλεπίδραση ανθρώπου – μηχανής

Η επεξεργασία ήχου, εργάζεται για βελτιστοποίηση πολλών τομέων όπως συστήματα ασφαλείας ή αμυντικά συστήματα. Ένας από αυτούς τους τομείς είναι και η αλληλεπίδρασης ανθρώπου μηχανής μέσα από την εξαγωγή χαρακτηριστικών από τον προφορικό λόγο. Είτε πρόκειται για επικοινωνία ανθρώπου με υπολογιστή είτε με κάποιο μεγαλύτερο υπολογιστικό σύστημα, σε βιομηχανίες και απλές συσκευές όπως κινητά τηλέφωνα.

Χαρακτηριστικό είναι το παράδειγμα ενός Ολλανδικού πρότυπου συστήματος, το οποίο μπορεί να εφαρμοστεί σε τηλεφωνικά κέντρα και αξιολογεί την συναισθηματική ένταση και το στρες της φωνής του ανθρώπου που τηλεφωνεί μέσα από την ένταση της φωνής, την ταχύτητα ομιλίας, τις πολλές αυξομειώσεις του τόνου και τα κενά ανάμεσα στις λέξεις. Το σύστημα τότε μπορεί να δώσει προτεραιότητα σε αυτή την κλήση. Αν πρόκειται για κλήση σε αστυνομία ή για κλήση σε νοσοκομείο, τα αποτελέσματα θα είναι πάντα προς το καλύτερο. Για την υλοποίηση ενός τέτοιου συστήματος, δυο από τα αξιοποιούμενα χαρακτηριστικά της ομιλίας είναι η ανίχνευση των λέξεων και ο τόνος (pitch) της φωνής.

Ο σκοπός αυτής της εργασίας είναι σχετικός με τα δυο αυτά ζητούμενα. Η ανάπτυξη ενός ενσωματωμένου συστήματος που να μπορεί να εξάγει και τα δυο αυτά χαρακτηριστικά από την ομιλία. Να βρίσκει δηλαδή τα χρονικά όρια φωνής μέσα σε ήχο και να ανιχνεύει το pitch της. Στην εργασία αυτή αναλύονται τα χαρακτηριστικά που μελετήθηκαν για την υλοποίηση αυτού του συστήματος.

1.3 Περιγραφή κεφαλαίων

Η πτυχιακή αυτή χωρίζεται στα παρακάτω κεφάλαια:

Στο δεύτερο κεφάλαιο παρουσιάζονται αναφορικά κάποια στοιχεία για τον ήχο. Δίνονται ορισμένα εργαλεία που χρησιμοποιούνται για να γίνει εφικτή η επεξεργασία του ήχου καθώς και κάποιες βασικές μαθηματικές αρχές για την ψηφιακή του επεξεργασία. Τέλος δίνεται μια βασική περίληψη της παραγωγής της ανθρώπινης ομιλίας. Γενικά το κεφάλαιο αυτό έχει σα σκοπό να δώσει τις βασικές αρχές έτσι ώστε ο αναγνώστης να γνωρίζει τι να ψάξει, αν αποφασίσει να ασχοληθεί περισσότερο με το συγκεκριμένο αντικείμενο.

Στο τρίτο κεφάλαιο δίνονται οι ορισμοί της θεμελιώδους συχνότητας και δίνεται έμφαση στη διαφορά μεταξύ της ανθρώπινης αντίληψης της και της μέτρησής της. Παρουσιάζονται κάποια ηχητικά παράδοξα σχετικά με αυτή. Έπειτα, αναφέρονται οι διαφορές μεταξύ εξαγωγής θεμελιώδης συχνότητας από ένα μουσικό κομμάτι και από ανθρώπινη φωνή καθώς και οι δυσκολίες που αντιμετωπίζει η κάθε προσέγγιση

Τα κεφάλαια τέσσερα και πέντε αναλύουν τους αλγορίθμους που χρησιμοποιούνται. Στο τέταρτο κεφάλαιο χωρίζονται σε κατηγορίες οι αλγόριθμοι εξαγωγής θεμελιώδης συχνότητας και παρουσιάζονται ορισμένοι από αυτούς. Ειδικά ένας από αυτούς, ο αλγόριθμος της αυτοσυσχέτισης αναλύεται σε αρκετά μεγάλο βαθμό λόγω της αξιοπιστίας που του προσάπτεται από τις διάφορες υλοποιήσεις του καθώς και το ενδιαφέρον που έχει σαν αλγόριθμος. Σύμφωνα με τις επιδόσεις τους στην υπάρχουσα βιβλιογραφία, επιλέγονται ορισμένοι από αυτούς για να υλοποιηθούν. Στο πέμπτο κεφάλαιο αναλύεται ο αλγόριθμος κατάτμησης φωνής και η εύρεση των ορίων των λέξεων μέσα σε μια ηχογράφηση ήχου. Στην ουσία είναι μια ανάλυση δυο δημοσιεύσεων (papers) [2][3] τα οποία πλέον είναι ευρέως χρησιμοποιούμενα.

Τα κεφάλαια έξι, εφτά και οχτώ είναι κεφάλαια που έχουν να κάνουν με την υλοποίηση του ενσωματωμένου συστήματος. Στο κεφάλαιο έξι γίνεται μια παρουσίαση του λογισμικού που αναπτύχθηκε σε έναν προσωπικό υπολογιστή και το μπλοκ διάγραμμα του προγράμματος μαζί με επεξηγήσεις. Στο κεφάλαιο εφτά επιλέγεται ο μικροεπεξεργαστής και αναλύονται οι δυνατότητές του. Επίσης επιλέγεται η αναπτυξιακή πλατφόρμα πάνω στην οποία θα εκτελεστεί το λογισμικό και γίνεται μια παρουσίαση των περιφερειακών μονάδων της. Το κεφάλαιο οχτώ δίνει το μπλοκ διάγραμμα για την υλοποίηση του λογισμικού στην ενσωματωμένη πλατφόρμα τονίζοντας τις διαφορές που έχει με την υλοποίηση του προγράμματος στον προσωπικό υπολογιστή.

Τέλος το κεφάλαιο εννέα δίνει τις αποδόσεις του ενσωματωμένου συστήματος και τους περιορισμούς που έχει και γίνεται μια περιγραφή σχετικά με τις μελλοντικές αναπτύξεις που θα υλοποιηθούν στη συγκεκριμένη πλατφόρμα και τη μορφή που θα πάρει το τελικό προϊόν.

2 Ήχος και φωνή – θεωρητικό υπόβαθρο

Σε αυτό το κεφάλαιο δίνονται κάποιες βασικές αρχές και ορισμοί περί ήχου και μια σύντομη μαθηματική εισαγωγή στη θεωρία της ψηφιακής του ανάλυσης για να μπορούν να γίνουν πιο εύκολα κατανοητά τα επόμενα κεφάλαια.

2.1 Πληροφορίες σχετικά με τον ήχο

Ήχος είναι ένα μηχανικό κύμα το οποίο είναι μια ταλάντωση πίεσης που μεταδίδεται μέσα από στερεά, υγρά ή αέρια, και παράγεται από συχνότητες μέσα στο φάσμα της ακοής και ένταση ικανή για να ακουστεί [4].

Ταλάντωση πίεσης είναι η τοπική απόκλιση πίεσης από τον μέσο όρο της ατμοσφαιρικής πίεσης. Συχνότητα είναι ο αριθμός των κυμάτων ανά δευτερόλεπτο. Ένταση είναι η ενέργεια του ήχου ανά μονάδα χρόνου μέσα σε μια μονάδα εμβαδού. Στο S.I. η μονάδα είναι το 1 Pa, το οποίο ορίζεται ως εξής: 1 Pa = $kg/(m*s^2)$.

Το φάσμα της ακοής στο οποίο αναφέρεται ο ορισμός είναι κατά μέσο όρο από 20 Hz ως 20 kHz αν και για συγκεκριμένους ανθρώπους μπορεί να ανεβαίνει πολύ το πάνω όριο. Επίσης, καθώς ο άνθρωπος γερνάει, το πάνω όριο μειώνεται αρκετά.

Το απόλυτο κατώφλι ακοής (Absolute Threshold of Hearing – ATH), είναι η κατώτερη ένταση ενός τόνου που το μέσο αυτί μπορεί να ακούσει χωρίς να είναι παρών άλλος ήχος [5]. Ο μέσος όρος είναι 20 μPa (micropascals) στη συχνότητα του 1 kHz. [6].

Σύμφωνα με τον ορισμό, πολλές είναι οι πηγές που μπορεί να παράγουν ήχο, αρκεί να δημιουργηθεί μια ταλάντωση που να πληρεί τις κατάλληλες προϋποθέσεις, όπως εξηγήθηκαν παραπάνω, όταν φτάσει στα ανθρώπινα αυτιά. Τα μουσικά όργανα και η ανθρώπινη φωνή είναι από τα σημαντικότερα.

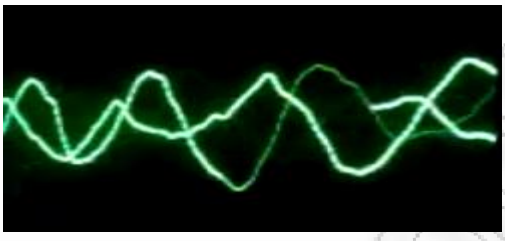
2.2 Βασική θεωρία

Ο ήχος, σαν ακουστικό σήμα, είναι ιδιότητα του αναλογικού κόσμου. Για να γίνει εφικτή η ψηφιακή επεξεργασία του, πρέπει πρώτα να μετατραπεί από ακουστικό σήμα σε ηλεκτρικό αναλογικό σήμα και έπειτα να μετατραπεί σε ψηφιακό σήμα. Έπειτα, μετά το τέλος της επεξεργασίας, πρέπει να ξαναγίνει μετατροπή από ψηφιακό σε αναλογικό σήμα και πάλι σε ακουστικό σήμα.

2.2.1 Αισθητήρας μετατροπής από ακουστικό σε ηλεκτρικό σήμα

Για να μετατραπεί το ακουστικό σήμα σε ηλεκτρικό σήμα πρέπει να περάσει από κάποιον αισθητήρα. Τέτοιοι αισθητήρες ονομάζονται αισθητήρες ήχου (sound sensors) και διαφέρουν μεταξύ τους στον τρόπο λειτουργίας τους. Οι αισθητήρες ενσωματώνονται σε μικρόφωνα, τα οποία, ανάλογα με το είδος του αισθητήρα, χωρίζονται σε πολλές κατηγορίες. Ορισμένοι τύποι μικροφώνων είναι το χωρητικό (condenser), το ηλεκτροστατικό-μαγνητικό χωρητικό (electret condenser), το δυναμικό, το μικρόφωνο άνθρακα, το πιεζοηλεκτρικό κλπ.

Όταν τελικά το ηχητικό σήμα μετατραπεί μέσω του αισθητήρα σε ηλεκτρικό σήμα, μπορεί, να μετρηθεί και να μελετηθεί όπως οποιοδήποτε άλλο ηλεκτρικό σήμα. Το σχήμα 2-1 είναι μια φωτογραφία της οθόνης ενός αναλογικού παλμογράφου όταν διαβάζει ένα τέτοιο ηλεκτρικό σήμα.



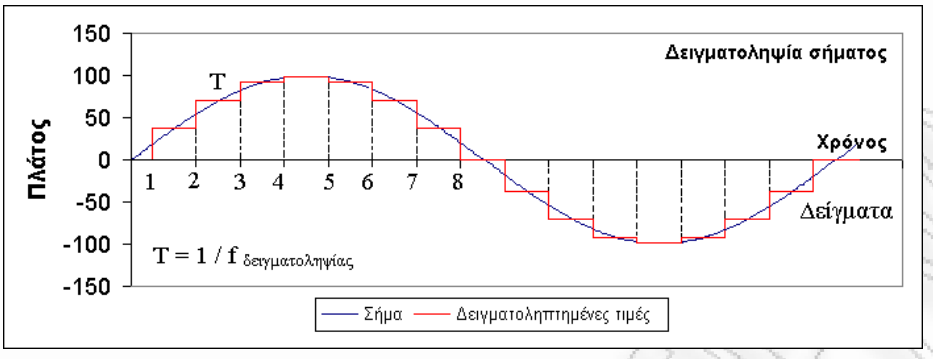
Εικόνα 2-1: Στιγμιότυπο ήχου όπως φαίνεται σε αναλογικό παλμογράφο

2.2.2 Μετατροπέας αναλογικού σήματος σε ψηφιακό (ADC)

Το σήμα πλέον μπορεί να περάσει από έναν μετατροπέα, που θα μετατρέψει το αναλογικό ηλεκτρικό σήμα σε ψηφιακό (Analog to Digital Converters – ADC). Στα ηλεκτρονικά, ο μετατροπέας είναι συνήθως ένα ολοκληρωμένο κύκλωμα που παίρνει σαν είσοδο μια αναλογική τάση ή ρεύμα και σαν έξοδο δίνει την αριθμητική ισοδυναμία της εισόδου.

Υπάρχουν αρκετοί τρόποι μετατροπής και οι αρχές λειτουργίας τους διαφέρουν. Μερικοί από αυτούς είναι οι: άμεσοι μετατροπείς (direct conversion ADC), οι μετατροπείς διαδοχικής προσέγγισης (successive approximation ADC), σίγμα – δέλτα (sigma – delta ADC) κλπ. Η μορφή της εξόδου είναι διαφορετική ανάλογα με το ολοκληρωμένο που χρησιμοποιείται. Συνήθως είναι μια δυαδική αναπαράσταση ως προς δυο (two's complement).

Εκτός από την αρχή λειτουργίας τους, διακρίνονται δυο πολύ σημαντικά χαρακτηριστικά στον τρόπο λειτουργίας τους. Ο ρυθμός δειγματοληψίας (sampling rate), που είναι η χρονική απόσταση μεταξύ δυο διαδοχικών μετατροπών. Το σήμα δε μπορεί να μετρηθεί σε κάθε χρονική στιγμή (αφού ο χρόνος είναι αναλογικό μέγεθος). Είναι προφανές λοιπόν πως χάνεται κάποια ακρίβεια στο πεδίο του χρόνου, αφού δεν είναι δυνατό να προσδιοριστεί το πλάτος ενός σήματος που δειγματοληπτείται, σε στιγμή που δε γίνεται δειγματοληψία. Αν όμως η δειγματοληψία γίνεται σχετικά γρήγορα συγκριτικά με τον ρυθμό αλλαγής του σήματος, τότε μπορούν να γίνουν εικασίες για τη συμπεριφορά του. Σε περιοδικά σήματα η συχνότητα δειγματοληψίας πρέπει να είναι μεγαλύτερη από τη διπλάσια συχνότητα του σήματος (θεώρημα δειγματοληψίας Nyquist), αλλιώς παρατηρείτε το φαινόμενο των ψευδοσυχνοτήτων. Στο σχήμα 2-2 φαίνεται ένα δειγματοληπτημένο σήμα.

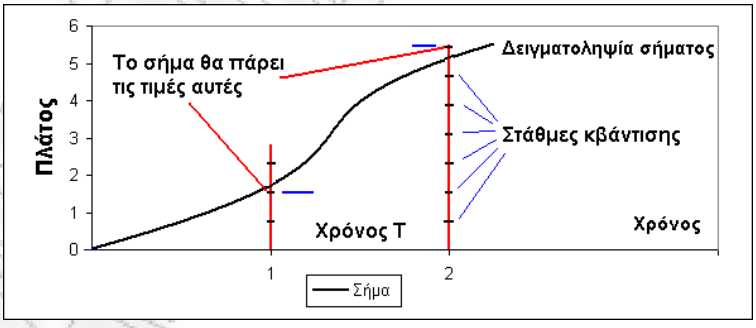


Εικόνα 2-2: Δειγματοληπτημένο ημιτονικό σήμα

Το άλλο χαρακτηριστικό των μετατροπέων είναι η ακρίβεια που δίνουν στο πλάτος του μετρούμενου σήματος. Η ακρίβεια αυτή εξαρτάται από την ανάλυση (resolution) του μετατροπέα. Ανάλυση είναι ο αριθμός των διακριτών σταθμών (τιμών) που μπορεί να πάρει το μετρούμενο σήμα κατά τη διάρκεια της μέτρησης. Οι στάθμες μπορεί να είναι σε γραμμική κλίματα ή σε άλλη κλίμακα, για παράδειγμα λογαριθμική.

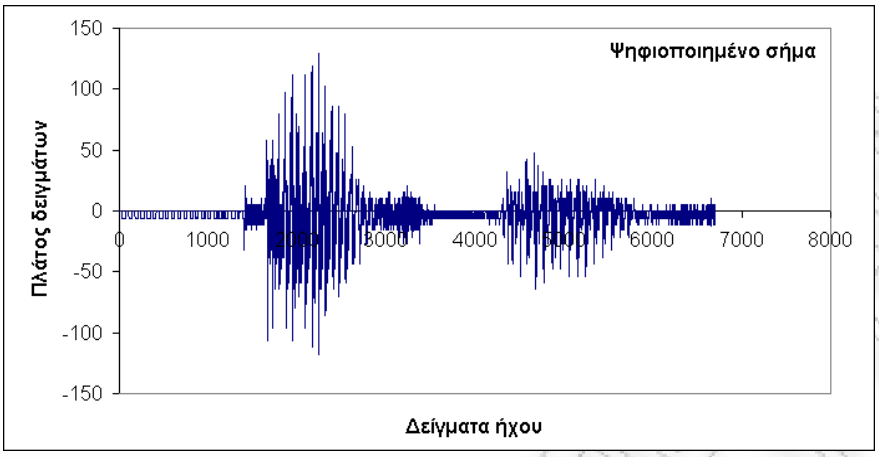
Στην περίπτωση για παράδειγμα ενός γραμμικού μετατροπέα με 256 στάθμες, με μικρότερη τιμή τα 0 volt και μεγαλύτερη τιμή τα 5 volt, η κάθε στάθμη σημαίνει (5 – 0)/256 volt σήματος. Αν λοιπόν ένα μετρούμενο σήμα τη στιγμή της δειγματοληψίας είναι 3.45 volt, θα δώσει 177 στάθμες (3.45 * 256 / 5 = 176.64 με στρογγυλοποίηση στον πλησιέστερο ακέραιο). Η απόκλιση της τιμής μέτρησης σε σχέση με την πραγματική τιμή ονομάζεται λάθος κβάντισης (quantization error).

Επειδή επιπλέον είναι ηλεκτρονικές συσκευές, οι τιμές αυτές αποθηκεύονται στο δυαδικό σύστημα, δηλαδή μια μετρούμενη τιμή θα αναπαρασταθεί σε δυαδική μορφή. Αυτό σημαίνει πως ο αριθμός κβάντισης είναι πάντα δύναμη του δυο. Αν για παράδειγμα ένας μετατροπέας έχει 256 διακριτές στάθμες, τότε η ανάλυσή του είναι 8 bit (2^8 = 256). Ένα γραφικό παράδειγμα δίνεται στο σχήμα 2-3.



Εικόνα 2-3: Επιλογή στάθμης σε κβάντιση σήματος

Επιπλέον, στο σχήμα 2-4 δίνεται η κυματομορφή ήχου που έχει ηχογραφηθεί σε έναν Η/Υ. Αυτό είναι μια πλήρης κυματομορφή.

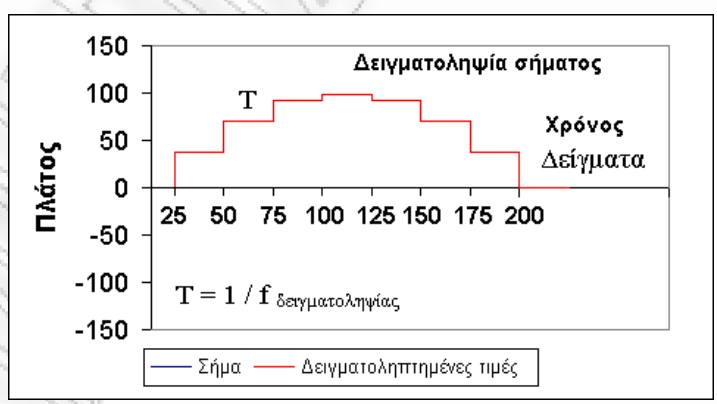


Εικόνα 2-4: Ψηφιακό σήμα, αφού πέρασε από ηχογράφηση στον Η/Υ

2.2.3 Μετατροπέας ψηφιακού σήματος σε αναλογικό (DAC)

Την αντίθετη εργασία, δηλαδή τη μετατροπή από ψηφιακό σήμα σε αναλογικό, κάνει ένας μετατροπέας από ψηφιακό σε αναλογικό σήμα (DAC - Digital to Analog Converter). Οι διαφορετικές αρχές λειτουργίας τους είναι διαμόρφωση πλάτους παλμού (Pulse Width Modulation – PWM), δέλτα σίγμα (delta – sigma DAC) και μερικές άλλες. Ένας DAC παίρνει το σήμα σε ψηφιακή μορφή, πιθανώς σε συμπλήρωμα ως προς δυο και το μετατρέπει σε τάση.

Η συχνότητα δειγματοληψίας διέπεται από την ίδια αρχή όπως και ο ADC, δηλαδή είναι ο χρόνος μεταξύ δυο διαδοχικών μετατροπών. Επίσης, στην περίπτωση αυτή δε γίνεται κβάντιση, οπότε δε χάνεται ακρίβεια, απλά οι αριθμοί που δέχεται στην είσοδο του ο ADC, είναι φιξαρισμένοι στις στάθμες κβάντισης. Αυτό σημαίνει πως η ακρίβεια έχει χαθεί από πριν την μετατροπή. Η έξοδος ενός DAC δίνεται στο σχήμα 2-5.



Εικόνα 2-5: Έξοδος μετατροπέα από αναλογικό σε ψηφιακό (DAC)

2.2.4 Παραθυρική συνάρτηση (window ή weighting function)

Μια παραθυρική συνάρτηση (window function ή weighting function) είναι μια συνάρτηση, η οποία όταν πολλαπλασιάζετε με ένα σήμα, του δίνει ορισμένα χαρακτηριστικά, δίνει «βάρος» (weighting function) σε αυτά. Τα χαρακτηριστικά αυτά μπορεί να είναι η αύξηση του πλάτους των ενδιάμεσων τιμών του σήματος (Hamming window) ή απλά ο μηδενισμός του έξω από κάποια όρια του πεδίου ορισμού (rectangular window function). Η τετραγωνική συνάρτηση είναι η εξής:

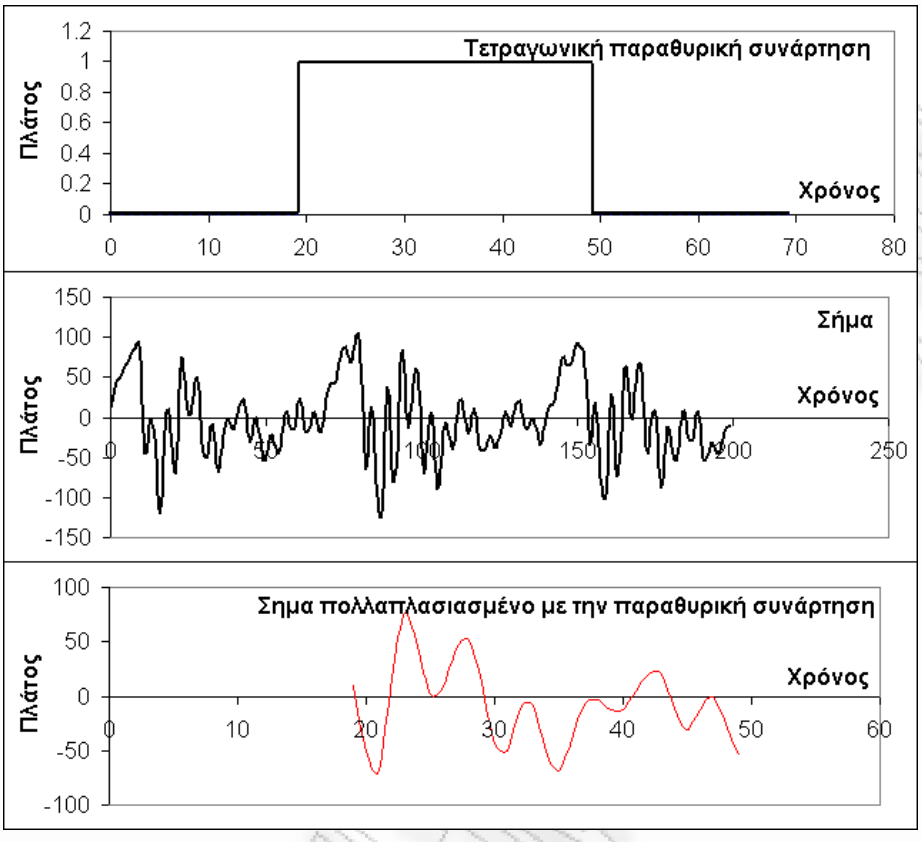
$$W_r(k) = 1,\ K1 < k < K2$$

$$W_r(k) = 0,\ k < K1, k > K2$$

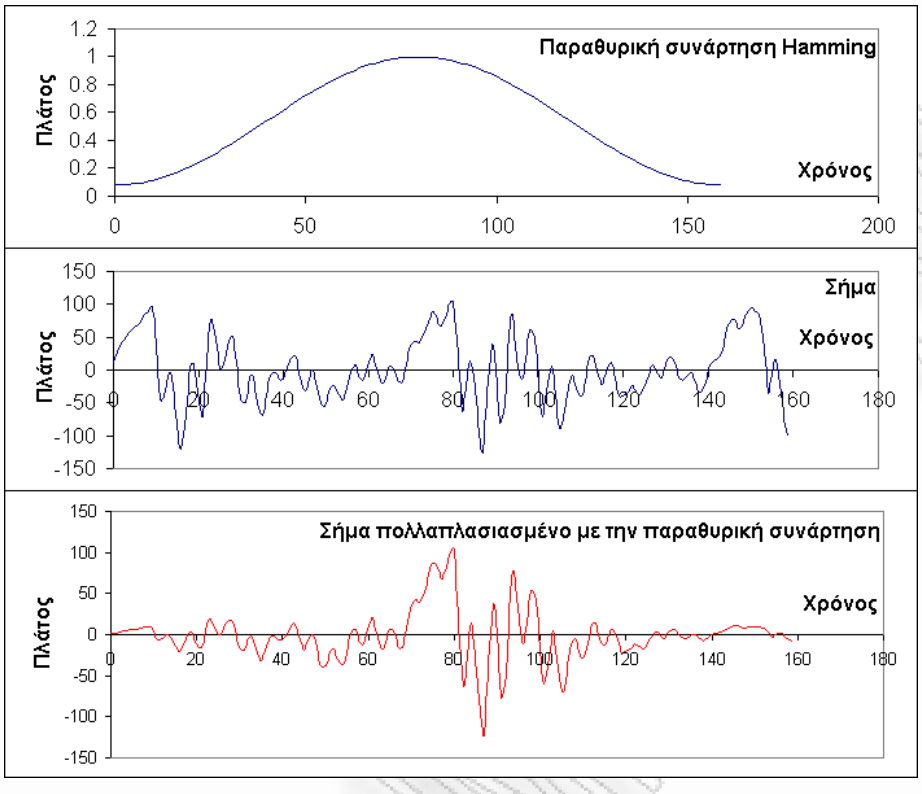
Ενώ η συνάρτηση Hamming είναι η εξής

$$W_r(k) = 0.54 + 0.46 * \cos(2\pi k / K)$$

Στο σχήμα 2-6 δίνεται το αποτέλεσμα του πολλαπλασιασμού ενός σήματος με τετραγωνική παραθυρική συνάρτηση και στο σχήμα 2-7 το ίδιο σήμα πολλαπλασιασμένο με μια hamming. Στην περίπτωση της τετραγωνικής το πλάτος του σήματος, στο συγκεκριμένο χρονικό πλαίσιο μένει ως έχει, ενώ μηδενίζεται έξω από αυτό το περιθώριο. Στη περίπτωση της hamming, τονίζεται το σήμα στο μέσο του χρονικού πλαισίου, συμπιέζοντας το σήμα στα άκρα, δίνοντας λιγότερη ισχύ στα δεξιά και αριστερά όρια.



Εικόνα 2-6: Πολλαπλασιασμός σήματος με τετραγωνική παραθυρική συνάρτηση



Εικόνα 2-7: Πολλαπλασιασμός σήματος με παραθυρική συνάρτηση Hamming

2.2.5 Μετατροπή σήματος στο πεδίο των συχνοτήτων και αρμονικές

Ο διακριτός μετασχηματισμός Fourier (Discrete Fourier Transform – DFT), ορίζεται παρακάτω:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-\frac{2\pi i}{N} kn} \ , \ k = 0,...N-1$$

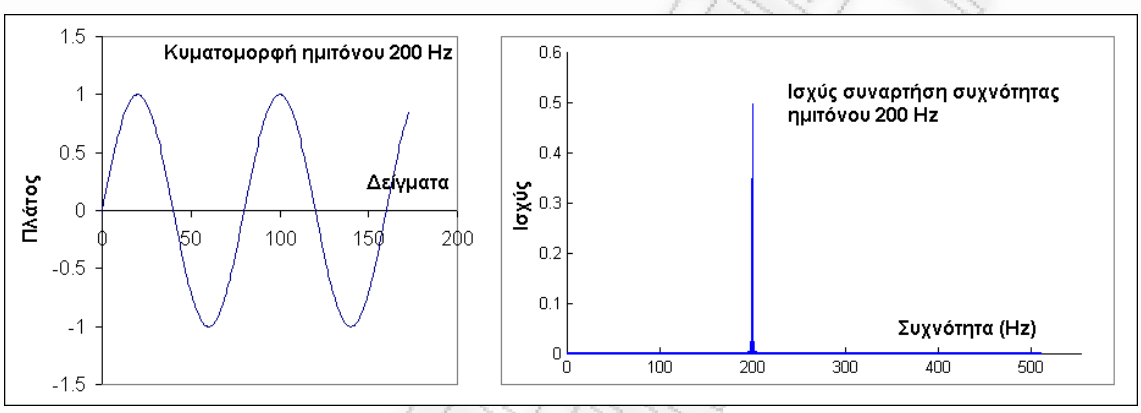
Και μετασχηματίζει μια συνάρτηση, από το πεδίο του χρόνου στο πεδίο των συχνοτήτων. Η διαδικασία αυτή μπορεί να γίνει μόνο με συναρτήσεις διακριτού χρόνου και με πεπερασμένο μήκος. Πλέον με την εισαγωγή των γρήγορων μετασχηματισμών Fourier (Fast Fourier Transform – FFT) όπως radix – 2 και radix – 4 ο μετασχηματισμός γίνεται σχετικά γρήγορα σε ένα μικροϋπολογιστικό σύστημα οπλισμένο με αριθμητική μονάδες κινητής υποδιαστολής. Οι αλγόριθμοι radix – 2 και radix – 4 απαιτούν το μήκος του σήματος να είναι δύναμη του 2 ή του 4 αντίστοιχα.

Αντίστοιχα, η μετατροπή από το πεδίο των συχνοτήτων στο πεδίο του χρόνου γίνεται με τον αντίστροφο μετασχηματισμό Fourier (Inverse Discrete Fourier Transform – IDFT), όπως ορίζεται παρακάτω:

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k) e^{\frac{2\pi i}{N} kn} \ , \ n = 0,...N-1$$

Οι συναρτήσεις αυτές δίνονται για λόγους πληρότητας επειδή σε ορισμένα κεφάλαια θα γίνει λόγος για συχνότητες στο φάσμα και αρμονικές συχνότητες. Αυτά για να βρεθούν, χρησιμοποιούν τους μετασχηματισμούς DFT και IDFT.

Στο σχήμα 2-8 δίνεται ένα ημίτονο και το φάσμα του. Η γραμμή αυτή τοποθετείτε πάνω στον άξονα των συχνοτήτων και αντιπροσωπεύει τη συχνότητα του ημιτόνου.

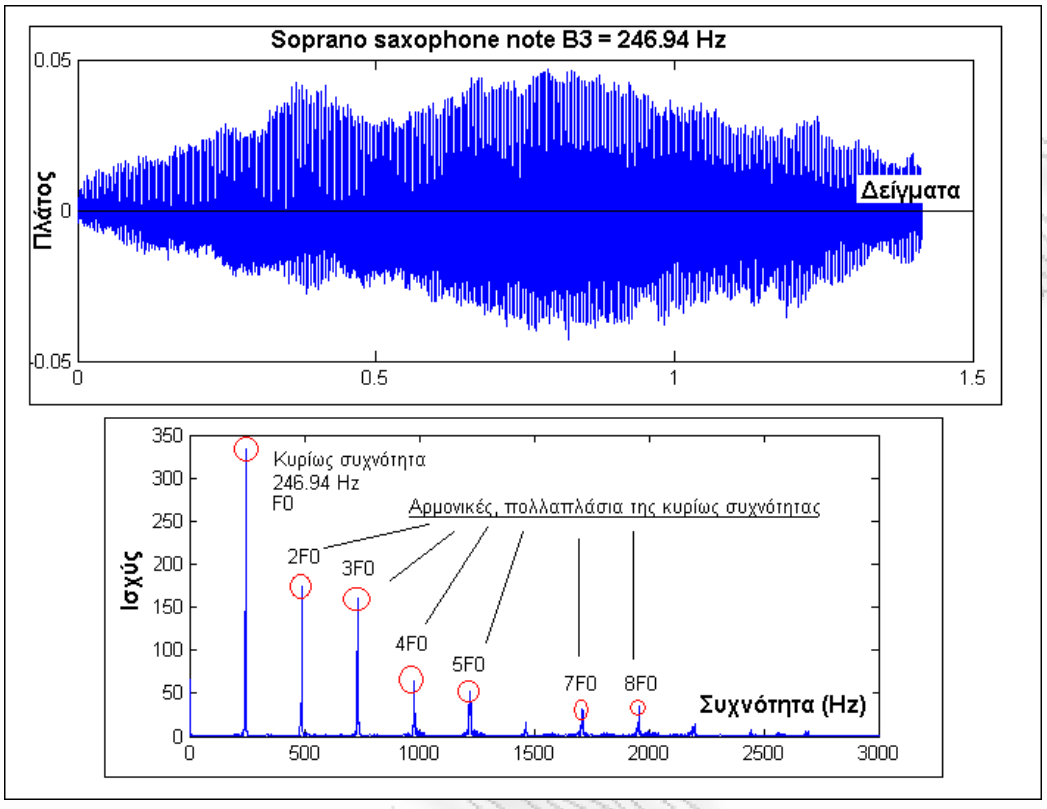


Εικόνα 2-8: Φάσμα ημιτόνου στα 200 Hz

2.2.6 Αρμονικές

Οι αρμονικές συχνότητες ενός σήματος, είναι ακέραια πολλαπλάσια της θεμελιώδης συχνότητας ενός σήματος. Αν η θεμελιώδης συχνότητα είναι f_0, τότε οι αρμονικές είναι 2 f_0, 3 f_0, …. Οι αρμονικές, στο πεδίο των συχνοτήτων, απέχουν μεταξύ τους απόσταση f_0. Τονίζεται πως τα μουσικά όργανα, όταν παίζουν, παράγουν πολλές αρμονικές και το διάγραμμα του πεδίου των συχνοτήτων δε θα είναι μόνο μια γραμμή, αυτή της βασικής συχνότητας της νότας, όπως δόθηκε στο προηγούμενο σχήμα.

Στο σχήμα 2-9 δίνεται κυματομορφή ενός σαξοφώνου που παίζει την νότα B3, που έχει συχνότητα 246.94 Hz και το φάσμα της κυματομορφής αυτής. Ξεχωρίζουν αρκετές αρμονικές, οι οποίες σημειώνονται.



Εικόνα 2-9: Φάσμα σαξοφώνου νότα B3

2.3 Ομιλία

Η ανθρώπινη ομιλία, είναι ήχοι που συνήθως αποτελούνται από σύμφωνα και φωνήεντα, όπως υπάρχουν και στο αλφάβητο, τα οποία συντάσσονται με χρονική σειρά για να σχηματίσουν λέξεις. Τα σύμφωνα, χωρίζονται σε δυο μεγάλες κατηγορίες, τα φωνητικά (voiced) σύμφωνα και τα μη φωνητικά (unvoiced) σύμφωνα.

2.3.1 Φωνητικές χορδές

Οι φωνητικές χορδές, αποτελούνται από δυο δίδυμες μεμβράνες τεντωμένες οριζόντια μέσα στον λάρυγγα. Δονούνται διαμορφώνοντας τη ροή του αέρα που εξέρχεται από τους πνεύμονες παράγοντας φωνή [7]. Μεταξύ των ήχων που παράγονται από τις φωνητικές χορδές είναι τα φωνήεντα και τα φωνητικά σύμφωνα. Αυτά τα δυο, είναι περιοδικοί ήχοι, όπως τα ‘α’, ‘ε’, ‘β’, ‘γ’ κλπ. Σύμφωνα όπως ‘λ’ και ‘κ’ κλπ, περνάνε στιγμιαία από τις φωνητικές χορδές προκαλώντας την ταλάντωσή τους, άρα είναι φωνητικά σύμφωνα.

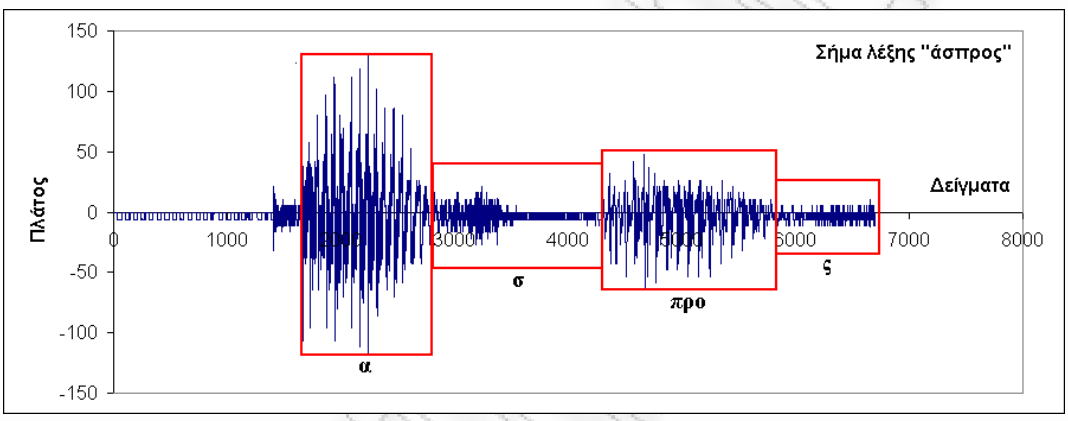
2.3.2 Στοματική κοιλότητα

Σύμφωνα όπως το ‘φ’, ‘σ’, ‘θ’, είναι τα μη φωνητικά σύμφωνα και παράγονται από τη στοματική κοιλότητα, με έξοδο του αέρα από τους πνεύμονες. Ο ήχος τους είναι αποτέλεσμα της τυχαίας κίνησης

των μορίων του αέρα και το αποτέλεσμα είναι μη περιοδικό (για να καταλάβει κάποιος πια σύμφωνα είναι μη-φωνητικά και πια είναι φωνητικά είναι απλό, μπορεί να τοποθετήσει ένα δάχτυλο στο δέρμα έξω από τον λάρυγγα και να προφέρει τα σύμφωνα. Εκεί που θα αισθανθεί μια δόνηση, αυτά είναι τα φωνητικά σύμφωνα. Όταν δεν αισθανθεί δόνηση, το σύμφωνο είναι μη φωνητικό).

2.3.3 Σύνθεση λέξεων

Στο σχήμα 2-10 δίνεται η κυματομορφή της λέξης «άσπρος», με το ενδιαφέρον κόλλημα τριών σύμφωνων μαζί, ένα μη φωνητικό και δυο φωνητικά, για να τονιστεί η διαφορά ανάμεσα στα φωνήεντα και τα σύμφωνα από τα οποία αποτελείται.



Εικόνα 2-10: Κυματομορφή της λέξης "άσπρος"

Φαίνεται πως κατά τη διάρκεια του ‘α’ και του «προ», το σήμα είναι αρκετά τονισμένο, αν και το ‘α’ είναι περισσότερο τονισμένο επειδή κυριολεκτικά τονίζεται η λέξη στο ‘α’, ενώ το ‘σ’ είναι πολύ υποβιβασμένο. Αυτό συμβαίνει στις περισσότερες περιπτώσεις, τα φωνήεντα και τα φωνητικά σύμφωνα έχουν μεγαλύτερο πλάτος από τα μη φωνητικά σύμφωνα.

Τα φωνητικά σύμφωνα μαζί με τα φωνήεντα, εκτός από μεγαλύτερο πλάτος, έχουν και άλλα συγκεκριμένα χαρακτηριστικά, σε αντίθεση με τα μη φωνητικά σύμφωνα που τα χαρακτηρίζει η τυχαιότητα. Ένα από τα κύρια χαρακτηριστικά τους είναι η θεμελιώδης συχνότητα.

2.3.4 Θεμελιώδης συχνότητα

Θεμελιώδης συχνότητα της φωνής είναι το φαινόμενο που επαναλαμβάνεται ανά χρόνο T_0, όπου T_0 ορίζεται ως ο χρόνος μεταξύ δυο διαδοχικών παλμών από τον λάρυγγα [8].

Η θεμελιώδης συχνότητα των φωνητικών χορδών ενός ανθρώπου δεν έχει μεγάλες διακυμάνσεις. Τα όρια της συχνότητας στους άντρες είναι από 85 ως 180 Hz και στις γυναίκες από 165 ως 265 Hz [9].

Όσον αφορά το τραγούδι, οι επαγγελματίες τραγουδιστές μπορούν να τραγουδήσουν σε ένα μεγάλο φάσμα, στη διάρκεια ενός τραγουδιού. Η μέτρηση της θεμελιώδης συχνότητας κατά τη διάρκεια του τραγουδιού δεν είναι αντικείμενο αυτής της εργασίας.

3 Θεμελιώδης συχνότητα (pitch)

Η θεμελιώδης συχνότητα, που για τη συνέχεια αυτής της εργασίας ονομάζεται pitch, είναι μια από τις βασικές ακουστικές ιδιότητες του ήχου, μαζί με τη διάρκεια, την ένταση, την ανίχνευση της πηγής και το τέμπο. Αυτό που αντιλαμβάνονται οι άνθρωποι σαν pitch, είναι πως είναι η συχνότητα ενός ήχου, για αυτό το λόγο, αναφορά σε υψηλό pitch, σημαίνει υψηλές συχνότητες και αναφορά σε χαμηλό pitch, σημαίνει χαμηλές συχνότητες. Όμως η πραγματικότητα είναι λίγο διαφορετική. Παρακάτω δίνονται τρεις από τους ορισμούς του pitch:

- Αντιπροσωπεύει την αντίληψή της θεμελιώδης συχνότητας μιας νότας [10].
- Είναι η υποκειμενική αίσθηση, με την οποία ο ακροατής μπορεί να αναθέσει νότες, σε σχετικές θέσεις πάνω σε μια μουσική κλίματα, βασισμένος κυρίως στην συχνότητα της δόνησης [11] (της πηγής του ήχου).
- Ο ψυχοακουστικός ορισμός ANSI του pitch είναι ο εξής: «pitch είναι αυτή η ιδιότητα της ακοής, βάση της οποίας μπορεί να ταξινομηθεί ένας ήχος σε μια κλίμακα από χαμηλά προς ψηλά».

Ενώ η επιστημονική προσέγγιση του pitch, είναι η μέτρηση της συχνότητας, δηλαδή η μέτρηση της ταλάντωσης των ηχητικών κυμάτων, που αντιπροσωπεύει την δόνηση της πηγής του ήχου, είτε σε απλούς είτε σε πολύπλοκους ήχους, ο ανθρώπινος εγκέφαλος αντιλαμβάνεται τα πράγματα διαφορετικά. Όπως φαίνεται και από τους τρεις ορισμούς, το pitch είναι υποκειμενικό φαινόμενο (αντίληψη – υποκειμενική αίσθηση – ιδιότητα της ακοής) και καθορίζεται σε μεγάλο βαθμό από διάφορους άλλους παράγοντες εκτός της συχνότητας.

3.1 Ανθρώπινη αντίληψη και pitch

Μια επιστήμη που ασχολείται εκτεταμένα με το pitch σαν αντιληπτική ικανότητα, είναι η ψυχοακουστική (psychoacoustics). Η ψυχοακουστική είναι ένας κλάδος της ψυχολογίας, που μελετάει τις ψυχολογικές και φυσικές αντιδράσεις του ανθρώπου, οι οποίες σχετίζονται με τον ήχο. Σαν κλάδος, έχει δώσει αρκετά πρακτικά αποτελέσματα σε διάφορους τεχνολογικούς τομείς όπως:

- Στην ανάπτυξη λογισμικού πάνω σε επεξεργασία ήχου. Για παράδειγμα, έχει δώσει αρκετές εφαρμοσμένες τεχνικές στην κατασκευή του codec mp3 για να αυξηθεί η συμπίεση του [12].
- Στη σχεδίαση συστημάτων ήχου υψηλής ποιότητας (high end) για κινηματογράφους, συναυλιακούς χώρους και για το σπίτι (home theater).
- Αμυντικά συστήματα και συστήματα τήρησης της τάξης.

Και σε διάφορους άλλους τομείς.

Μέσω της ψυχοακουστικής, δίνονται αρκετά ενδιαφέρον συμπεράσματα. Το ανθρώπινο αυτί και ο εγκέφαλος, επεξεργάζονται σαν παραμέτρους πέρα από τη συχνότητα και άλλες μεταβλητές όπως ένταση, διάρκεια, ανίχνευση της πηγής του ήχου και τέμπο. Ακόμα και η όραση του ακροατή τη συγκεκριμένη στιγμή παίζει ρόλο [13][14]. Το αποτέλεσμα είναι πως το pitch είναι διαφορετικό από τη συχνότητα των ήχων που ακούμε. Σαν παραδείγματα δίνονται τα εξής:

- Το ύψος της ηχητικής πηγής δίνει την εντύπωση πως είναι ανάλογο με το ύψος του pitch, ακόμα και αν η ηχητική πηγή παράγει τον ίδιο ήχο [15]. Δηλαδή όσο πιο ψηλά τοποθετείται η πηγή, τόσο ψηλότερο φαίνεται πως είναι το pitch.
- Η πίεση του ήχου (όγκος και ένταση), ειδικά σε συχνότητες κάτω από τα 1000 Hz και πάνω από 2000 Hz, στη πρώτη περίπτωση δίνει την αίσθηση πως το pitch είναι πιο χαμηλό από όσο είναι και στη δεύτερη ότι είναι πιο ψηλό από όσο είναι [16].
- Σε υψηλές συχνότητες, η αύξηση της έντασης, δίνει την εντύπωση πως αυξάνεται και το pitch [16].

Με βάση αυτές και άλλες παρατηρήσεις, έχουν δημιουργηθεί πειράματα τα οποία επιβεβαιώνουν άμεσα τις παραμέτρους αυτές, με τις οποίες ο ανθρώπινος εγκέφαλος συνδέει το pitch. Μερικά από αυτά τα πειράματα, οι «ακουστικές παραισθήσεις» (auditory illusions) όπως ονομάζονται, περιλαμβάνουν

- Binaural beats [17]. Αν στο ένα αυτί εφαρμοστεί ήχος συχνότητας μικρότερης από 1 kHz και στο άλλο αυτί ήχος συχνότητας, πάλι μικρότερης από 1 kHz αλλά με διαφορά από τον πρώτο μικρότερη από 30 Hz, για παράδειγμα 400 και 410 Hz, τότε ο εγκέφαλος θα δημιουργήσει την εντύπωση πως ακούει έναν τόνο 10 Hz. Ο άνθρωπος δε μπορεί να «ακούσει» τόνους τόσο χαμηλής συχνότητας και αυτή η αίσθηση είναι σαν να ακούγεται ένας ρυθμός (beat). Με διαφορετικές διαφορές συχνότητας, παράγονται και διαφορετικές παραισθήσεις. Τα binaural beats μελετούνται από την νευρολογία.

- Παραίσθηση κλίμακας Deutsch [18], με διαφορετικούς ήχους σε κάθε αυτί, όπου συχνά, οι δεξιόχειρες αντιλαμβάνονται πως οι υψηλές νότες έρχονται από το δεξί αυτί και οι χαμηλές από το αριστερό, ενώ το αντίθετο ισχύει για τους αριστερόχειρες.

- Παραίσθηση Glissando [19][20], με παρόμοια εφφέ για αριστερόχειρες και δεξιόχειρες.

- Παραίσθηση συνέχειας των συχνοτήτων[21], όταν ένας τόνος διακόπτεται για χρόνο μικρότερο από περίπου 50 mc, ο άνθρωπος δεν αντιλαμβάνεται τη διακοπή.

- Εφφέ McGurk [13][14], η όραση επηρεάζει την ακοή. Παρατηρώντας ανθρώπους, που φαινομενικά κάνουν έναν ήχο με το στόμα τους, ενώ στην πραγματικότητα ακούγεται ένας άλλος ήχος, ο εγκέφαλος νομίζει πως ειπώθηκε ένας τρίτος, ενδιάμεσος ήχος. Το καταπληκτικό είναι πως ακόμα και όταν ο άνθρωπος γνωρίζει αυτό το εφφέ, πάλι δε μπορεί να κάνει τον

διαχωρισμό μεταξύ των ήχων, σε αντίθεση με τις οπτικές παραισθήσεις, που μόλις αντιληφθεί την παραίσθηση, δε γίνεται ξανά εμφανής.

- Τόνος του Shepard [22]. Είναι ένας ήχος που αποτελείτε από το άθροισμα 12 τόνων που διαφέρουν ανά οκτάβες. Κατά τη διάρκεια της αναπαραγωγής μια περιόδου, το pitch του ήχου ανεβαίνει. Όταν αυτός ο ήχος αναπαράγεται συνέχεια, δημιουργεί την παραίσθηση πως ο τόνος ανεβαίνει ή κατεβαίνει μόνιμα. Τέτοια παραδείγματα μπορούν να βρεθούν σε ένα έργο του J. S. Bach (Jesu, Joy of Man's Desiring) και του Tchaikovsky (Pathetique).

- Άκουσμα μιας συχνότητας που δεν υπάρχει, αν δοθούν άλλα μέρη της αρμονικής σειράς [23][24].

Και διάφορα άλλα πειράματα.

Στη συνέχεια αυτής της εργασίας, το pitch, δε θα έχει να κάνει με την ανθρώπινη αντίληψη. Η ανίχνευσή του ανάγεται στα πλαίσια της ψηφιακής επεξεργασίας σήματος και είναι συνάρτηση της κυματομορφής του ήχου σε ψηφιοποιημένη μορφή.

3.2 Διαφορές εξαγωγής pitch ανθρώπινης φωνής και μουσικής

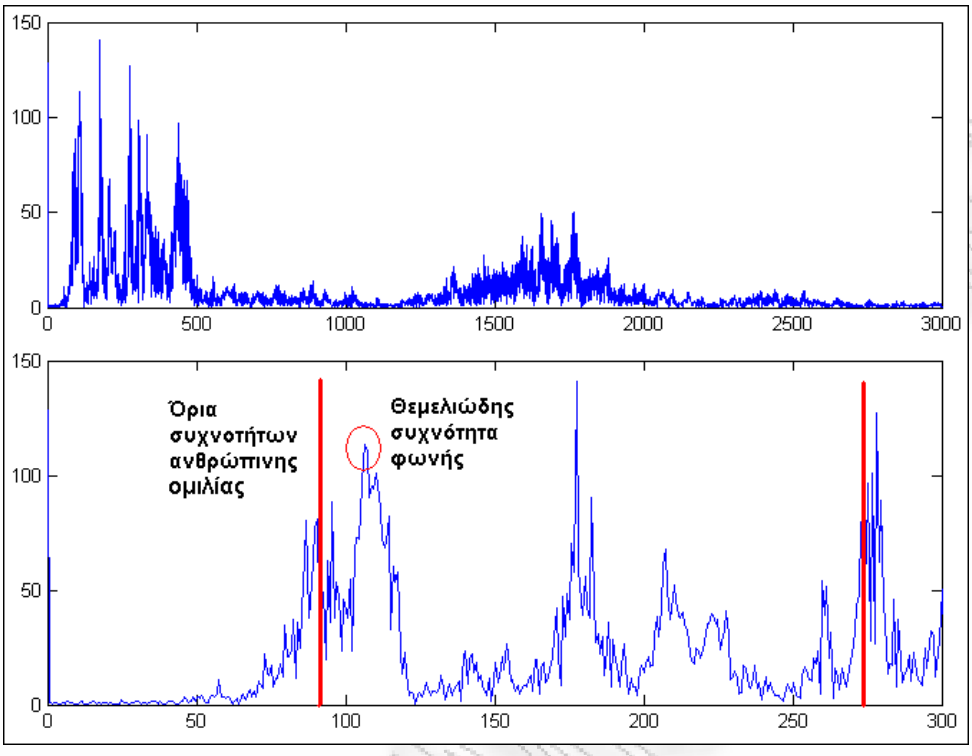
Τα δυο αυτά είδη αλγορίθμων διαφέρουν μεταξύ τους σε κυρίως 3 σημεία:

- Αρμονικές
- Φάσμα συχνοτήτων
- Χρόνος εκτέλεσης

Παρακάτω αναλύονται αυτές οι διαφορές.

3.2.1 Αρμονικές

Το σχήμα 3-1 δείχνει το φάσμα του ήχου που παράγει μια ηχογράφηση ανθρώπινης ομιλίας. Τονίζεται πως το συγκεκριμένο φάσμα δίνει μια συχνότητα μεγαλύτερη από αυτή που είναι σημειωμένη σαν θεμελιώδη συχνότητα φωνής. Αυτό όμως δεν επηρεάζει ορισμένους αλγόριθμους να λειτουργήσουν σωστά και να εξάγουν το σωστό pitch.



Εικόνα 3-1: Φάσμα και αρμονικές σε ανθρώπινη ομιλία

Συγκριτικά με το φάσμα μιας νότας, όπως στο σχήμα 2-9, το ανθρώπινο φάσμα είναι πιο τυχαίο και δεν έχει τόσο αυστηρή αρμονική δομή, στη φωνή, οι διάφορες συχνότητες μπορεί να φτάσουν και στο ίδιο ύψος με τη θεμελιώδη συχνότητα.

Ο αλγόριθμος που βρίσκει το pitch μουσικής πρέπει να εστιάσει στο γεγονός της πολύπλοκης δομής των αρμονικών συχνοτήτων. Η έλλειψη αυτών των ισχυρών αρμονικών στη φωνή δίνει τη δυνατότητα χρησιμοποίησης ενός πιο απλού αλγορίθμου, που μπορεί ακόμα και να αγνοήσει τις αρμονικές, αφού ψάχνει μόνο μια συγκεκριμένη συχνότητα.

3.2.2 Στενό φάσμα συχνοτήτων

Το pitch που έχει η ανθρώπινη φωνή, όπως δόθηκε παραπάνω έχει συγκεκριμένα και πολύ μικρά όρια. Οπότε ο αλγόριθμος μπορεί να σαρώνει τη φωνή ψάχνοντας για πολύ συγκεκριμένο φάσμα. Στη μουσική τα πράγματα είναι διαφορετικά καθώς το pitch που πρέπει να βρεθεί καταλαμβάνει τεράστιο φάσμα.

Ένα πιάνο μόνο, καλύπτει νότες από 16 Hz ως 20 kHz. Αυτό είναι τεράστιο φάσμα. Βέβαια, αν μειωθεί το φάσμα μέτρησης, αυξάνεται η ακρίβεια του αλγορίθμου. Για παράδειγμα μπορούν να γίνουν οι εξής εικασίες και να υπάρχουν αξιόλογα αποτελέσματα [25]

- Ανθρώπινη φωνή : 80 – 1000 Hz
- Πιάνο : 30 – 5000 Hz
- Σαξόφωνο : 50 – 1500 Hz
- Τρομπέτα : 150 – 1000 Hz

- Φλάουτο : 300 – 3000 Hz
- Ακουστική κιθάρα : 70 – 700 Hz

Φυσικά αυτές οι προσεγγίσεις δεν είναι πανάκεια, για παράδειγμα, υπάρχουν όπερες του Mozart, όπου μια σοπράνο coloratura μπορεί να φτάσει τα 1800 Hz (όπως στη «βασίλισσα της νύχτας» από το «μαγικός αυλός»).

3.2.3 Χρόνος εκτέλεσης

Η εξαγωγή pitch για μουσική πρέπει να είναι γρήγορη επειδή για κάθε χρονική στιγμή πρέπει να δείχνει τις αλλαγές στις αρμονικές δομές, οι οποίες είναι μεγάλες, λόγω της χρήσης των μουσικών οργάνων.

Στη περίπτωση ομιλίας ενός ανθρώπου, δεν είναι σκοπός να κάνουμε το πιο ακριβές σκιαγράφημα των συχνοτήτων εκείνη τη στιγμή αλλά να αναγνωριστεί ένας μέσος όρος της θεμελιώδης συχνότητας της φωνής που είναι πολύ πιο απλό.

Υπάρχουν και οι περιπτώσεις που το pitch πρέπει να βρεθεί για εξάσκηση τραγουδιστών. Στη συγκεκριμένη περίπτωση δεν θα γίνει αναγωγή στο πρόβλημα εξαγωγής από ομιλία, καθώς η τραγουδιστή φωνή μπορεί να πιάσει, αρκετά μεγάλο φάσμα.

3.3 Συμπέρασμα

Από τα προηγούμενα, συμπεραίνεται πως οι αλγόριθμοι εξαγωγής pitch φωνής είναι πιο απλή διαδικασία από την εξαγωγή pitch μουσικής ή τραγουδιστής φωνής. Αυτό επιτρέπει την επιλογή φτηνότερου συστήματος που δε χρειάζεται να σηκώσει βαριά επεξεργασία της πληροφορίας.

4 Αλγόριθμοι εξαγωγής pitch

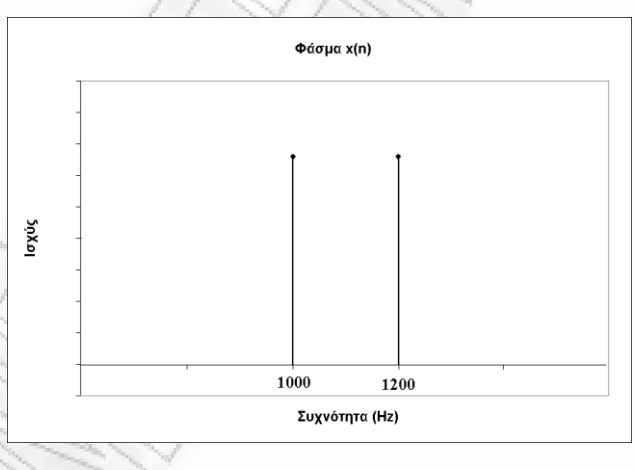
Στο κεφάλαιο αυτό αναλύονται ορισμένοι αλγόριθμοι εξαγωγής pitch και επιλέγονται οι κατάλληλοι για το σύστημα που θα υλοποιηθεί.

4.1 Εισαγωγή

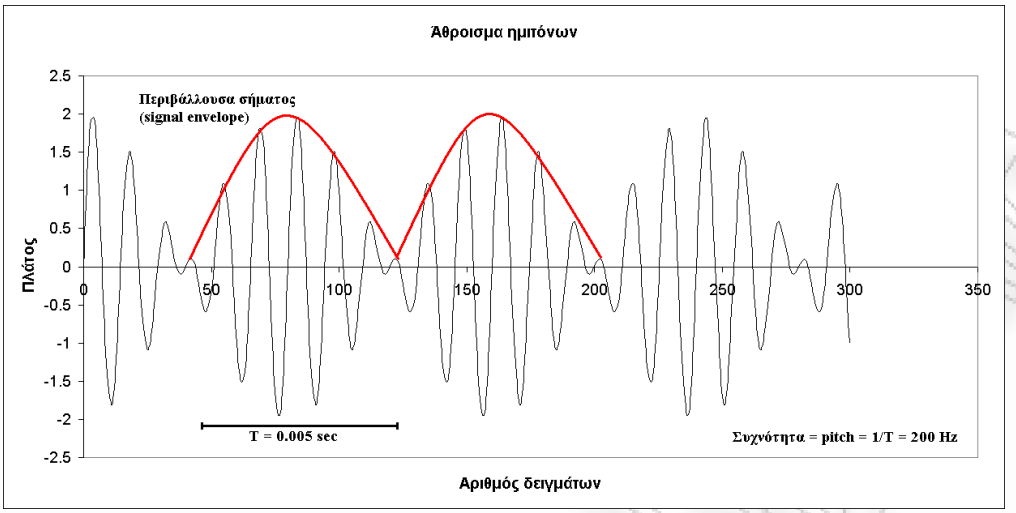
Όπως δόθηκε στο προηγούμενο κεφάλαιο, το pitch σχετίζεται άμεσα με την περίοδο ενός σήματος. Για παράδειγμα, μια κυματομορφή που αποτελείται από το άθροισμα δυο ημιτόνων, 1 kHz και 1.2 kHz:

$$x(t) = \sin(1000 \cdot 2\pi \cdot t) + \sin(1200 \cdot 2\pi \cdot t)$$

θα δώσει τρία pitch, ένα στα 1 kHz, ένα στα 1,2 kHz, που είναι οι συχνότητες που φαίνονται στο φάσμα αυτής της συνάρτησης, όπως φαίνεται στο σχήμα 4-1, και ένα στα 200 Hz, το οποίο παράγεται από την περιβάλλουσα του σήματος (signal envelope) [26]. Στο σχήμα 4-2 φαίνεται η γραφική παράσταση του σήματος και της περιβάλλουσας, που ο άνθρωπος λαμβάνει σαν pitch.

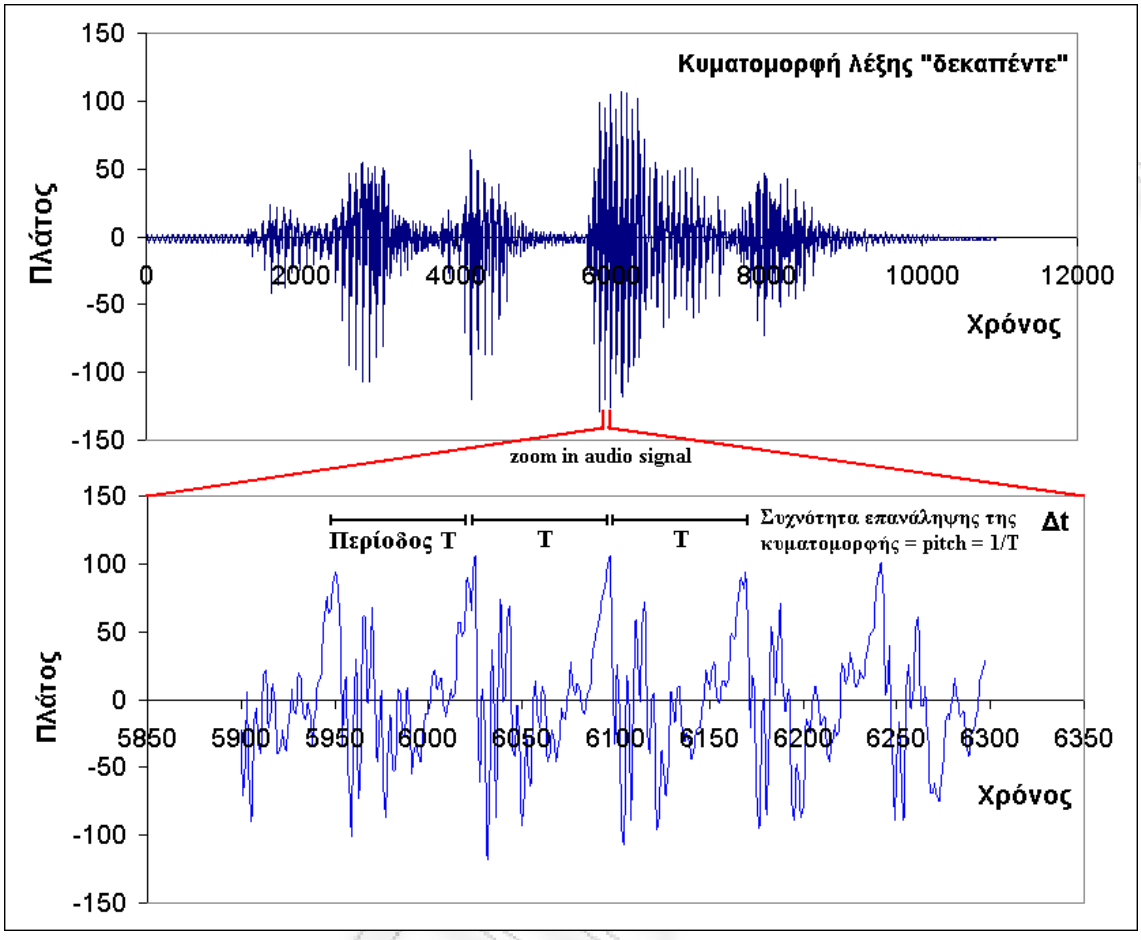


Εικόνα 4-1: Φάσμα σήματος αθροίσματος ημιτόνων 1 και 1.2 kHz



Εικόνα 4-2: Περιβάλλουσα (signal envelope) του προηγούμενου σήματος

Όμως, και στα μη-ημιτονικά σήματα, έτσι και για την περίπτωση της κυματομορφής της ανθρώπινης ομιλίας, διακρίνονται κάποια επαναλαμβανόμενα μοτίβα, όπως φαίνεται από το σχήμα 4-3. Αυτές οι επαναλαμβανόμενες διακυμάνσεις του ήχου, δίνουν την περίοδο, άρα και τη θεμελιώδη συχνότητα, που είναι το pitch. Προφανώς, όσο πιο πολύ μικραίνει ο χρόνος της περιόδου, το pitch μεγαλώνει. Αντίστοιχα, μεγαλύτερος χρόνος περιόδου σημαίνει μικρότερο pitch.



Εικόνα 4-3: Επαναλαμβανόμενα μοτίβα στην κυματομορφή της φωνής

Οι αλγόριθμοι εξαγωγής pitch, όταν χρησιμοποιούνται για ανθρώπινη φωνή, δέχονται σαν δεδομένο πως αυτά τα επαναληπτικά μοτίβα υπάρχουν και βρίσκουν την περίοδο επανάληψής τους. Για την υλοποίηση αυτής της εργασίας, έπρεπε να επιλεχθούν κάποιοι αλγόριθμοι από την πληθώρα που υπάρχουν. Υπάρχει εξαιρετικά μεγάλο πλήθος τέτοιων αλγορίθμων και επειδή δεν έχει βρεθεί η τέλεια μέθοδος, συνεχώς εφευρίσκονται και άλλοι.

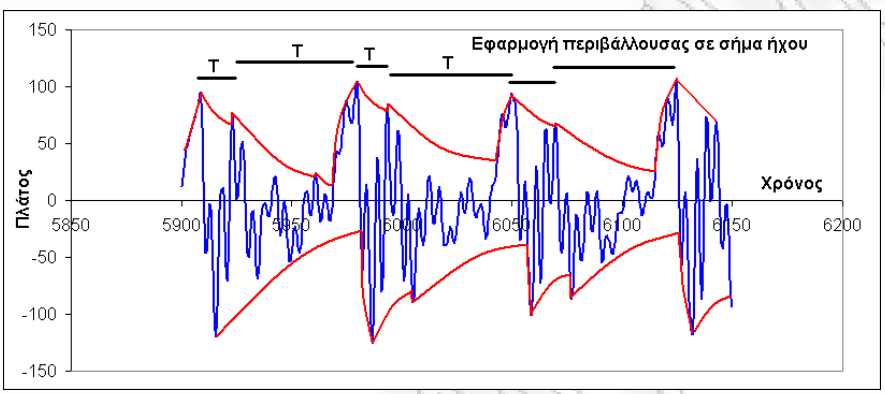
Οι κατηγορίες που αυτοί χωρίζονται είναι κυρίως σε αλγόριθμους που εργάζονται πάνω στο πεδίο του χρόνου, αλγόριθμους που εργάζονται πάνω στο πεδίο των συχνοτήτων, αλγόριθμους που δουλεύουν και στα δυο πεδία και άλλους αλγόριθμους.

Σε αυτή την εργασία, επειδή το κόστος πρέπει να κρατηθεί χαμηλά, η επεξεργασία που θα μπορεί να κάνει ο μικροελεγκτής δε θα μπορεί να σηκώσει έναν FFT. Για αυτό το λόγο παρουσιάζονται μόνο αλγόριθμοι που εργάζονται στο πεδίο του χρόνου.

Παρακάτω αναλύονται οι πιο γνωστοί από αυτούς. Είναι σημαντικό να τονιστεί, πως οι αλγόριθμοι αυτοί προσπαθούν να δουλέψουν με τον ίδιο τρόπο που δουλεύει το ανθρώπινο μάτι, με τη σύγκριση ομοιότητας των κυματομορφών μεταξύ τους.

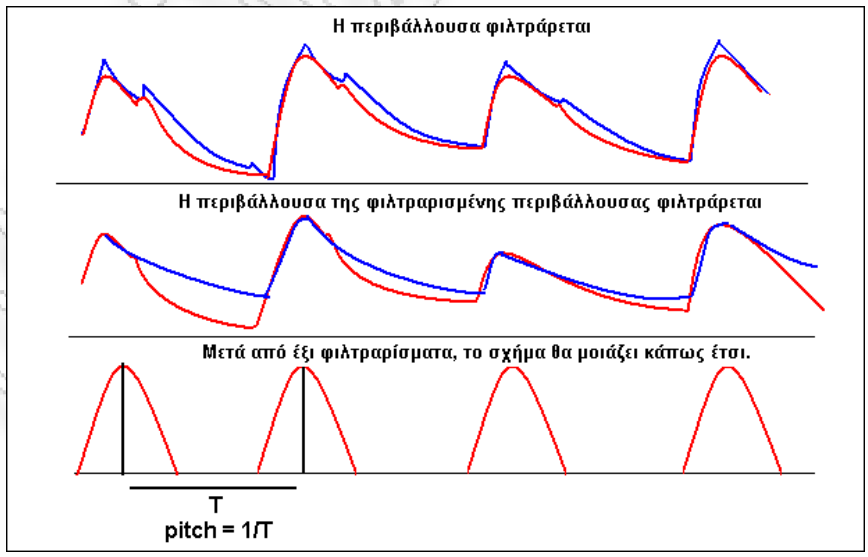
4.2 Απλός ακόλουθος περιβάλλουσας

Αυτός ο αλγόριθμος εμφανίστηκε πρώτη φορά υλοποιημένος με αναλογικά ηλεκτρονικά [26]. Στην πραγματικότητα, αυτή η περιβάλλουσα είναι μια εκθετική μείωση, σαν εκφόρτιση ενός πυκνωτή. Η περίοδος του pitch δίνεται από την διαφορά του χρόνου ανάμεσα σε διαδοχικές κορυφές που βρίσκονται από την περιβάλλουσα του σήματος. Στο σχήμα 4-4 δίνεται η γραφική παράσταση.



Εικόνα 4-4: Απλός ακόλουθος περιβάλλουσας

Στο paper είχε παρθεί μόνο η θετική περιβάλλουσα, όμως μπορεί να αξιοποιηθεί και η αρνητική, όταν όμως δίνουν διαφορετικούς χρόνους πρέπει να επιλεχθεί ο κατάλληλος. Δεν υπάρχει κάποια λύση για αυτό. Όπως φαίνεται αυτή η μέθοδος μπορεί να δίνει και λάθη τα οποία μπορούν να εξαλειφθούν περνώντας την περιβάλλουσα από ένα φίλτρο υψηλών συχνοτήτων ώσπου στο τέλος θα μετριέται μόνο η απόσταση μεταξύ δυο κορυφών (peak) για να δώσει την περίοδο. Το paper που το παρουσίασε πέρασε την περιβάλλουσα από έξι φίλτρα υψηλών, όπως φαίνεται και στο σχήμα 4-5.

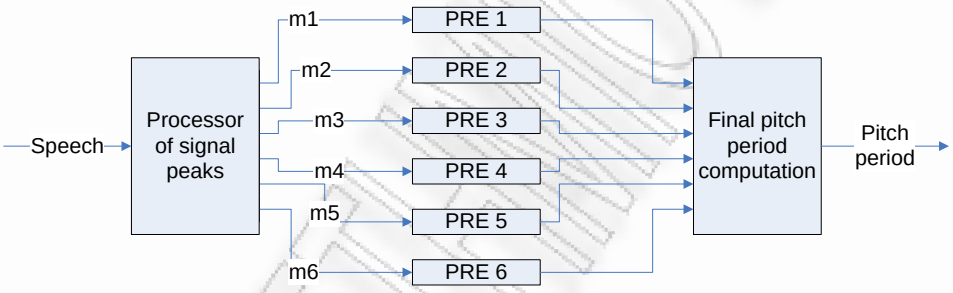


Εικόνα 4-5: Φιλτραρισμένες περιβάλλουσες

Τα αποτελέσματα είναι ικανοποιητικά και ο αλγόριθμος είναι εξαιρετικά γρήγορος. Βέβαια χρειάζεται και εξωτερικό κύκλωμα για να δουλέψει επειδή η διαδικασία ανίχνευσης της γραμμής της περιβάλλουσας κάθε φορά και των σημείων τομής της με το σήμα, αποτελεί μια χρονοβόρα για τον μικροελεγκτή διαδικασία. Ο αλγόριθμος αυτός δεν υλοποιήθηκε επειδή είναι επεξεργαστικά επίπονος χωρίς την προσθήκη εξωτερικού κυκλώματος.

4.3 Εξαγωγή pitch με παράλληλη επεξεργασία από Rabiner και Gold

Αυτός ο αλγόριθμος είναι αρκετά πολύπλοκος. Ονομάζεται αλγόριθμος παράλληλης επεξεργασίας επειδή η κάθε τελική επιλογή τιμής pitch, έχει επιλεχθεί από έξι υποψήφιες τιμές. Επίσης ο Rabiner υποστηρίζει πως οι παράλληλες μέθοδοι επεξεργασίας πολλών υποψηφίων pitch μοιάζουν πιο πολύ με τον τρόπο που το μάτι βρίσκει το pitch Το μπλοκ διάγραμμα, το οποίο ακολουθεί ο αλγόριθμος δίνεται στο σχήμα 4-6. Πριν ο ήχος φτάσει στο πρώτο μπλοκ, έχει ήδη περάσει από ένα φίλτρο χαμηλών συχνοτήτων.



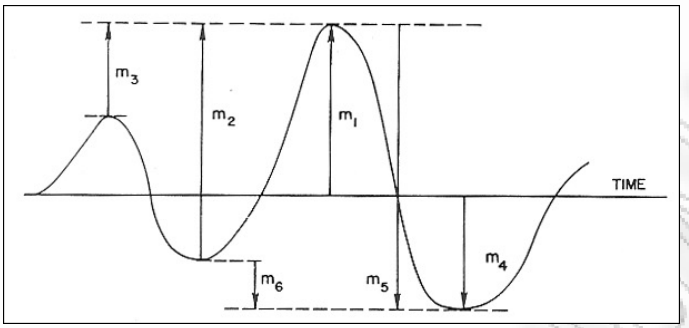
Εικόνα 4-6: Μπλοκ διάγραμμα του αλγορίθμου εξαγωγής pitch από Rabiner και Gold

Το σήμα, μετά από το φίλτρο χαμηλών συχνοτήτων οδηγείται στον αλγόριθμο. Η διαδικασία που ακολουθείται είναι η εξής:

- Εξαγωγή παλμών από τα χαρακτηριστικά του σήματος.
- Εξαγωγή υποψήφιων pitch.
- Επιλογέας pitch μέσα από «συμπτώσεις» (co incidents όπως τα αναφέρει ο Rabiner).

4.3.1 Εξαγωγή παλμών από τα χαρακτηριστικά του σήματος

Για την ανίχνευση των έξι τιμών: Βρίσκονται 2 τοπικά μέγιστα και δυο τοπικά ελάχιστα στη σειρά. Όπως στο σχήμα 4-7.



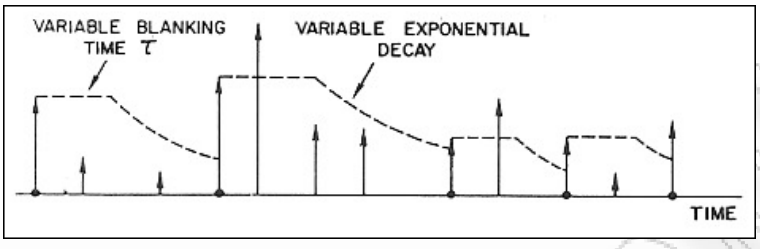
Εικόνα 4-7: Ανίχνευση τοπικών ελάχιστων και μέγιστων στη σειρά

Σύμφωνα με την παραπάνω κυματομορφή, η επιλογή γίνεται ως εξής:

- Οι τιμές m_1, m_2 και m_3 καθορίζονται σε κάθε τοπικό μέγιστο του φιλτραρισμένου σήματος.
- Οι τιμές m_4, m_5 και m_6 καθορίζονται σε κάθε τοπικό ελάχιστο του φιλτραρισμένου σήματος.
- Οι τιμές m_1 και m_4 είναι απλές μετρήσεις των τοπικών μέγιστων και ελάχιστων.
- Οι τιμές m_2, m_3, m_5 και m_6 εξαρτώνται από τα προηγούμενα τοπικά μέγιστα και ελάχιστα, πιο συγκεκριμένα:
    - Οι τιμές m_2 και m_5 είναι μετρήσεις τοπικό μέγιστο – τοπικό ελάχιστο και τοπικό ελάχιστο – τοπικό μέγιστο αντίστοιχα.
    - Οι τιμές m_3 και m_6 είναι μετρήσεις τοπικό μέγιστο με προηγούμενο τοπικό μέγιστο και τοπικό ελάχιστο με προηγούμενο τοπικό ελάχιστο.
    - Οι τιμές m_3 και m_5 δεν επιτρέπονται να γίνουν αρνητικές, αυτό σημαίνει πως αν κάποιο τοπικό μέγιστο ή ελάχιστο δεν είναι μεγαλύτερο από το προηγούμενο τοπικό μέγιστο ή ελάχιστο αντίστοιχα, οι τιμές αυτές μηδενίζονται.

4.3.2 Εξαγωγή υποψήφιων pitch

Αυτές οι τιμές τοποθετούνται σαν πλάτη πάνω στο πεδίο του χρόνου και με αποστάσεις ίδιες με αυτές που έχουν στο σήμα από το οποίο πάρθηκαν. Για μια χρονική περίοδο ‘τ’, η τιμή στον άξονα του πλάτους παραμένει σταθερή και ίση με m_n, έπειτα αρχίζει να μειώνεται εκθετικά, με εκθετική παράμετρο ‘β’. Πιθανή περίοδος είναι το χρονικό διάστημα που ξεκινάει από τη σταθερή τάση ώσπου αυτή να τέμνει κάποιο από τα επόμενα πλάτη m_{n+k}. Έπειτα ξαναρχίζει η διαδικασία. Τα πλάτη των τιμών m_k που βρίσκονται πριν το τέλος της διάρκειας ‘τ’, δεν ανιχνεύονται, για αυτό και το ‘τ’ ονομάζεται κενή περίοδος (blanking period). Το σχήμα 4-8 δίνει την γραφική περιγραφή.



Εικόνα 4-8: "Κενός" χρόνος και εκθετική μείωση

Οι τιμές των 'τ' και 'β' είναι συναρτήσεις της περιόδου. Η περίοδος βρίσκεται με τον τύπο:

$$P_{av}(n) = [P_{av}(n-1) + P_{new}]/2$$

Όπου P_{new} είναι η τελευταία εύρεση της περιόδου, $P_{av}(n)$ είναι η ομολοποιημένη τιμή της περιόδου και $P_{av}(n-1)$ είναι η προηγούμενη ομαλοποιημένη τιμή της περιόδου. Κάθε φορά που ανιχνεύεται ένα καινούριο τοπικό μέγιστο ή ελάχιστο, το P_{av} ανανεώνεται σύμφωνα με τον προηγούμενο τύπο. Το P_{av} πρέπει να είναι μεταξύ 4 και 10 msec. Οι τιμές 'τ' και 'β' είναι

$$\tau = 0.4P_{av}$$

$$\beta = P_{av}/0.695$$

Με αυτό τον τρόπο προκύπτουν έξι υποψήφιες τιμές pitch.

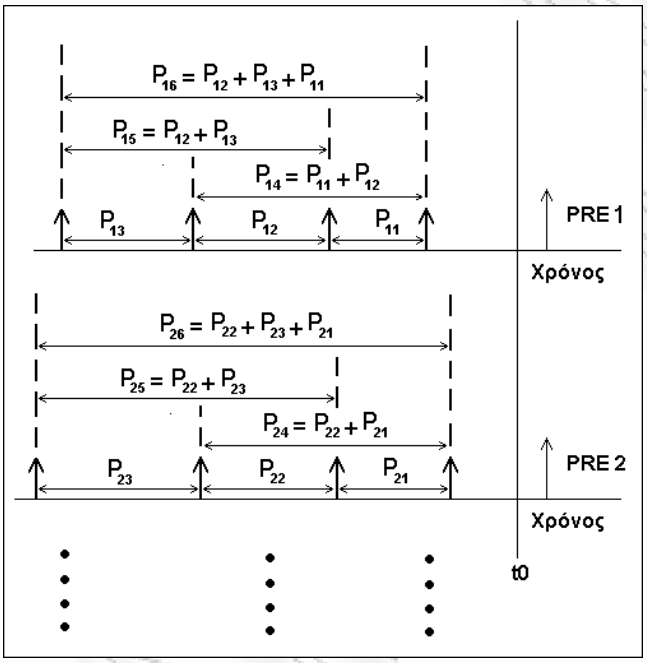
4.3.3 Επιλογέας pitch μέσα από συμπτώσεις

Οι έξι υποψήφιες περίοδοι που προκύπτουν, δίνουν έξι διαφορετικά pitch. Αυτά τοποθετούνται σαν έξι τιμές στη πρώτη γραμμή ενός πίνακα 6x6. Η δεύτερη γραμμή είναι οι 6 προηγούμενες εκτιμήσεις του pitch και η τρίτη γραμμή είναι οι 6 προηγούμενες από τις προηγούμενες εκτιμήσεις. Το κάθε στοιχείο της τέταρτης γραμμής του πίνακα είναι το άθροισμα των αντίστοιχων στοιχείων της πρώτης και δεύτερης γραμμής, αντίστοιχα η πέμπτη γραμμή δημιουργείται από το άθροισμα της δεύτερης και τρίτης γραμμής και η έκτη γραμμή από το άθροισμα της πρώτης, δεύτερης και τρίτης γραμμής. Το σχήμα 4-9 δίνει τη γραφική εξήγηση και η παράταξη των τιμών φαίνεται στον πίνακα 1.

P_{11}	P_{21}	P_{31}	P_{41}	P_{51}	P_{61}
P_{12}	P_{22}	P_{32}	P_{42}	P_{52}	P_{62}

P_{13}	P_{23}	P_{33}	P_{43}	P_{53}	P_{63}
P_{14}	P_{24}	P_{34}	P_{44}	P_{54}	P_{64}
P_{15}	P_{25}	P_{35}	P_{45}	P_{55}	P_{65}
P_{16}	P_{26}	P_{36}	P_{46}	P_{56}	P_{66}

Πίνακας 1: Κατάταξη των διαφόρων τιμών περιόδου



Εικόνα 4-9: Επιλογή περιόδων του αλγορίθμου Rabiner -Gold

Έπειτα κάθε υποψήφια περίοδος συγκρίνεται με όλες τις άλλες 35 τιμές του πίνακα 1 για 4 φορές και μετριέται ο αριθμός των συμπτώσεων που έχει. Σύμπτωση υπάρχει όταν η διαφορά μεταξύ της υποψήφιας περιόδου με τη συγκρινόμενη τιμή είναι μικρότερη από ένα συγκεκριμένο κατώφλι. Το κατώφλι αυτό είναι συνάρτηση του αριθμού επανάληψης της σύγκρισης της υποψήφιας συχνότητας με τον πίνακα 1 καθώς και της χρονικής διάρκειάς της. Ο πίνακας 2 επεξηγεί.

Bias		**1**	**2**	**5**	**7**
Αριθμός επανάληψης		**1**	**2**	**3**	**4**
		Κατώφλι συμπτώσεων (μsec)			
Pitch period range	**1.6 – 3.1**	**100**	**200**	**300**	**400**
	3.1 – 6.3	**200**	**400**	**600**	**800**
	6.3 – 12.7	**400**	**800**	**1200**	**1600**

	12.7 – 25.5	**800**	**1600**	**2400**	**3200**

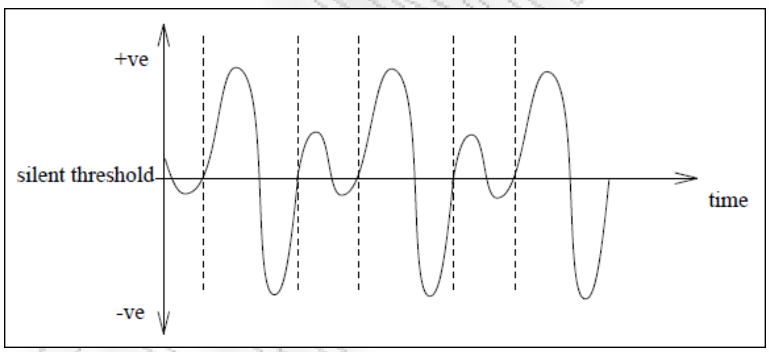
Πίνακας 2: Κατάταξη των διαφόρων τιμών περιόδου

Για παράδειγμα όταν η υποψήφια περίοδος βρίσκεται μεταξύ 3.1 και 6.3 δευτερολέπτων και ο αλγόριθμος βρίσκεται στη πρώτη σάρωση του πίνακα 1, η τιμή κατωφλίου είναι 200 μsec. Όταν όλες οι τιμές συγκριθούν και βρεθεί ο αριθμός των συμπτώσεων για τη πρώτη επανάληψη, αφαιρείται η μονάδα (το Bias, η πρώτη γραμμή του πίνακα 2). Στη δεύτερη επανάληψη, η τιμή κατωφλίου αυξάνεται, δημιουργώντας έτσι πιο πολλές συμπτώσεις, όμως για αντιστάθμιση το bias γίνεται μεγαλύτερο. Στο τέλος όλων των επαναλήψεων όλων των υποψήφιων περιόδων, επιλέγεται για περίοδος η τιμή που έχει τις πιο πολλές συμπτώσεις.

Ο αλγόριθμος αυτός δεν υλοποιήθηκε επειδή η επεξεργασία του, όπως αναφέρουν και οι συγγραφείς, ξεπερνάει κατά πολύ τα πλαίσια του πραγματικού χρόνου.

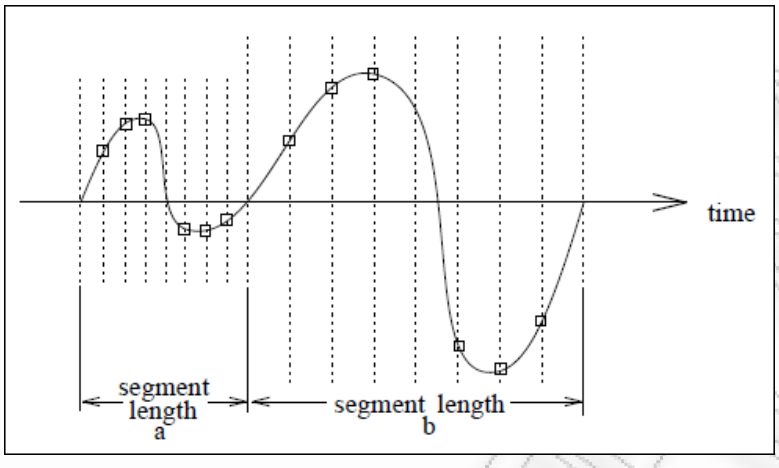
4.4 Αλγόριθμος από Cooper και Ng

Ο αλγόριθμος από τους Cooper και Ng παίρνει το σήμα και το αναλύει σε τμήματα. Το κάθε τμήμα βρίσκεται ανάμεσα σε δυο διαδοχικές ελεύσεις από το μηδέν, από τα αρνητικά προς τα θετικά, όπως στο σχήμα 4-9.



Εικόνα 4-10: Διαίρεση σε τμήματα για τον αλγόριθμο εξαγωγής pitch από Cooper και Ng

Κάθε τμήμα διαιρείτε σε 8 ίσα υπό-τμήματα. Για κάθε τμήμα παίρνονται τα πρώτα 3 και τα τελευταία 3 υπό-τμήματα. Το σχήμα 4-10 δείχνει την υπό-τμηματοποίηση δυο γειτονικών τμημάτων.



Εικόνα 4-11: Υπο τμηματοποίηση του τμήματος

Για να μετρηθεί η ομοιότητα μεταξύ των τμημάτων α και β, χρησιμοποιείτε η αναλογία ομοιότητας τους, η οποία δίνεται από τον τύπο

$$similiarity\_ratio = \frac{a.b}{(a.a)+(b.b)-(a.b)}$$ όπου

$$x.y = \sum_{i=1}^{6} x_i y_i$$

Όταν το η αναλογία ομοιότητας (similarity_ratio) ισούται με 1, τότε τα δυο τμήματα ταιριάζουν τέλεια (perfect match) ενώ η τιμή 0, δείχνει πως τα δυο τμήματα είναι τελείως διαφορετικά. Το τμήμα με το μεγαλύτερο μήκος, συγκρίνεται με όλα τα άλλα τμήματα και μετρούνται οι αποστάσεις και οι αναλογία ομοιότητας με αυτό. Για να βρεθεί η περίοδος ακολουθούνται τα εξής βήματα:

1. Η διαφορά του μήκος μεταξύ των δυο τμημάτων (ένα από αυτά είναι το μεγαλύτερο, όπως αναφέρθηκε) πρέπει να είναι μικρότερη από ένα κατώφλι.
2. Το similarity_ratio μεταξύ δυο τμημάτων πρέπει να είναι ψηλό, στο παράδειγμα το θέτουν.
3. Το όμοιο τμήμα πρέπει να είναι όσο το δυνατό πιο κοντά στο μεγαλύτερο τμήμα.

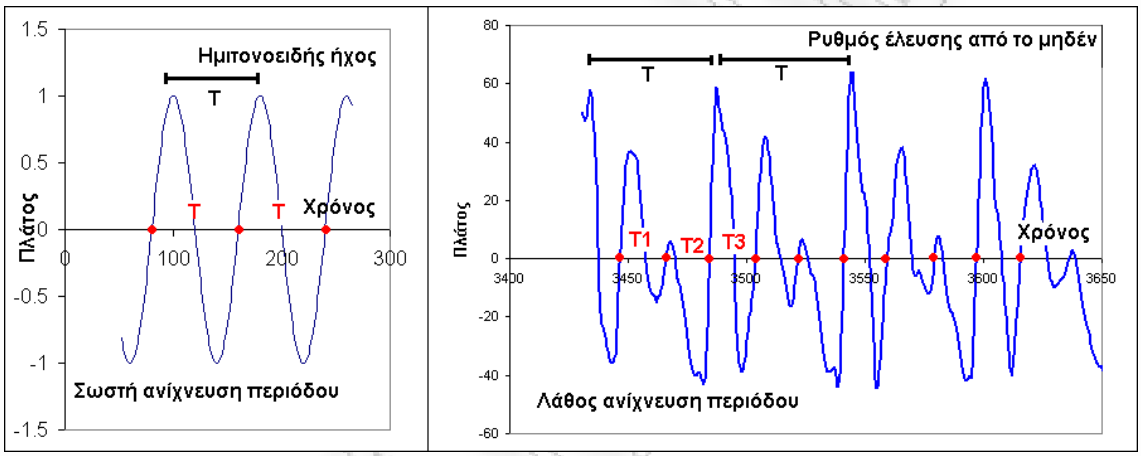
Το pitch υπολογίζεται, ως η απόσταση μεταξύ δυο όμοιων τμημάτων και βρίσκεται αριθμητικά ως εξής

$$pitch = \frac{συχνότητα\_δειγματοληψίας}{απόσταση\_μεταξύ\_δυο\_όμοιων\_τμημάτων}$$

Παρόλο που δεν απαιτεί ισχυρή επεξεργαστική δύναμη, στο τέλος δεν επιλέχθηκε επειδή υπερίσχυσαν άλλες επιλογές, όπως θα δοθεί παρακάτω.

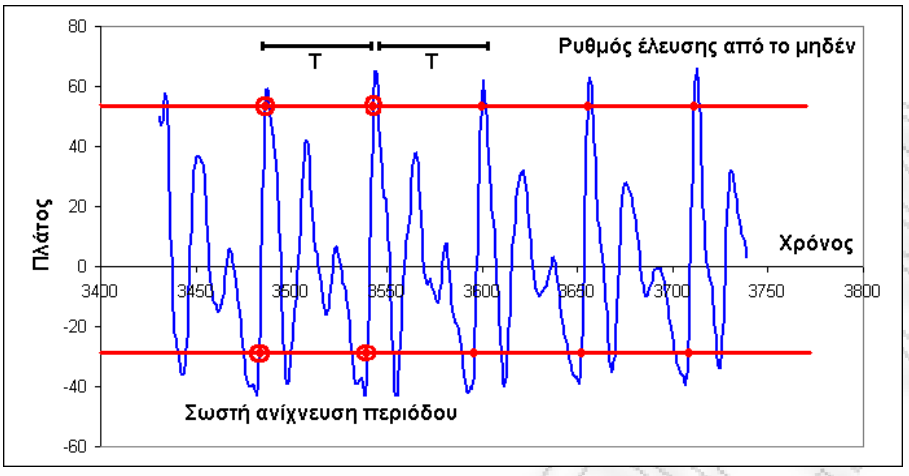
4.5 Αλγόριθμος ανίχνευσης ρυθμού έλευσης από το μηδέν

Ο Αλγόριθμος ανίχνευσης ρυθμού ελεύσεως από το μηδέν (Zero Cross Detection – ZCD) είναι από τους πιο παλιούς που χρησιμοποιήθηκαν. Μετράει το χρόνο που περνάει ανάμεσα σε δυο διαδοχικές αλλαγές του πρόσημου του σήματος, από τα αρνητικά προς τα θετικά. Παράδειγμα δίνεται στο σχήμα 4-11 με ένα απλό ημιτονικό σήμα και μια κυματομορφή ανθρώπινης φωνής.



Εικόνα 4-12: Έλευση από το μηδέν σε ημίτονο και φωνή – προβληματική ανίχνευση

Όπως φαίνεται, αν η κυματομορφή είναι μια απλή ημιτονοειδής καμπύλη, η μέτρηση είναι πολύ χρήσιμη ενώ στην άλλη περίπτωση υπάρχει μεγάλο περιθώριο σφάλματος. Μια ενδιαφέρον παραλλαγή αυτού του αλγορίθμου είναι η αλλαγή κατωφλίου από μηδέν σε μια άλλη τιμή, θετική η αρνητική και η προσθήκη δεύτερου κατωφλίου, με διαφορετικό πρόσημο από το άλλο κατώφλι, όπως φαίνεται στο σχήμα 4-12. Αυτό πρέπει να ανιχνεύσει δυο διαδοχικά περάσματα, από το αρνητικό προς το θετικό κατώφλι για να γίνει ανίχνευση περιόδου, δηλαδή εξαλείφει αρκετές λαθεμένες περιπτώσεις. Βελτιώνει κάπως τα πράγματα. Το πρόβλημα είναι η σωστή τοποθέτηση των κατωφλίων και αυτό είναι δύσκολο καθώς θέλει συνέχεια αλλαγή.



Εικόνα 4-13: Χρήση δυο κατωφλίων για τον αλγόριθμο ανίχνευσης έλευσης από το μηδέν

Τα αποτελέσματα αυτού του αλγορίθμου χειροτερεύουν ακόμα περισσότερο όταν προστεθεί θόρυβος στην πηγή ήχου. Τότε ο αλγόριθμος γίνεται τελείως αναξιόπιστος. Αυτός ο αλγόριθμος έχει το μεγάλο πλεονέκτημα πως είναι ο πιο γρήγορος από όλους τους άλλους. Ο αλγόριθμος αυτός υλοποιήθηκε αλλά λόγω πολλών λάθος αποτελεσμάτων δεν αναφέρετε παραπάνω σε αυτή την εργασία.

4.6 Αλγόριθμος αυτοσυσχέτισης (Autocorrelation function – ACF)

Στο κεφάλαιο αυτό δίνεται σε λεπτομέρεια η εξαγωγή pitch με τον κλασικό αλγόριθμο αυτοσυσχέτισης (Autocorrelation Function – ACF).

Δεδομένου ενός διακριτού σήματος x(n), ορισμένο σε όλα τα n, ο αλγόριθμος της συσχέτισης (cross-correlation) ορίζεται ως εξής:

$$\Phi_{xy}(m) = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x(n) \cdot y(n+m)$$

Ο αλγόριθμος αυτός, είναι ένας τρόπος για να μετρηθεί η ομοιότητα μεταξύ δυο σημάτων. Μια προσεκτική ματιά στη συνάρτηση αυτή, φανερώνει πως το ένα σήμα «ολισθαίνει» μέσα στο άλλο, πάνω στη μεταβλητή m (αριθμός δειγμάτων). Όταν το $\Phi_{xy}(m)$ πάρει τη μέγιστη τιμή του, σημαίνει πως οι δυο κυματομορφές «μοιάζουν» πολύ μεταξύ τους. Η τιμή m λοιπόν, για την οποία η συνάρτηση παίρνει μέγιστη τιμή, δίνει τη χρονική απόσταση σε δείγματα ήχου που πρέπει να ολισθήσει το ένα σήμα μέσα στο άλλο μέχρι οι κυματομορφές τους, να ταιριάζουν όσο το δυνατό γίνεται πιο πολύ μεταξύ τους.

Η αυτοσυσχέτιση (autocorrelation) είναι μια ειδική περίπτωση της συσχέτισης, οπού τα δυο συσχετιζόμενα σήματα είναι τα ίδια. Ο τύπος γίνεται:

$$\Phi_{xx}(m) = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x(n) \cdot x(n+m)$$ [27]

Η αυτοσυσχέτιση, αντίστοιχα με την συσχέτιση, δείχνει την ομοιότητα μεταξύ τμημάτων ενός σήματος. Όταν εφαρμοστεί η αυτοσυσχέτιση σε ένα περιοδικό σήμα, θα δώσει μέγιστη τιμή εκεί που το σήμα ταιριάζει πιο πολύ με τον εαυτό του, δηλαδή σε μια από τις επαναλήψεις του.

Στην περίπτωση της ψηφιακής επεξεργασίας του ήχου, όπως γίνεται σε αυτή την εργασία, δεν είναι πρακτικός ο παραπάνω τύπος. Αυτό που γίνεται είναι το σπάσιμο του προς επεξεργασία σήματος σε τμήματα και η εφαρμογή του αλγορίθμου αυτοσυσχέτισης σε αυτά τα τμήματα. Αυτό σημαίνει πως το όριο θα φύγει από την εξίσωση. Ο τύπος γίνεται ως εξής [27]:

$$\Phi_l(m)\frac{1}{N} = \sum_{n=0}^{N'-1}[x(n+l)w(n)][x(n+l+m)w(n)], 0 \le m \le M_0 - 1$$
$$w(n) = 1, 0 \le n \le N-1$$
$$w(n) = 0, n \le 0, n > N-1$$

Όπου w(n) είναι η απαραίτητη τετραγωνική παραθυρική συνάρτηση που πολλαπλασιάζεται με το σήμα προς ανάλυση (παρακάτω, στο σχήμα 4-16, αναλύεται ο λόγος για τον οποίο είναι μόνο τετραγωνική), N είναι το μήκος του σήματος της επεξεργασίας, N' είναι ο αριθμός των δειγμάτων του σήματος στο παράθυρο, M_0 είναι ο αριθμός των σημείων αυτοσυσχέτισης που θα υπολογισθούν και l είναι το πρώτο χρονικό σημείο του σήματος προς επεξεργασία. Συνήθως για εξαγωγή pitch, N'=N-m. [27].

Το N', δηλαδή το πεδίο ορισμού της παραθυρικής συνάρτησης, ονομάζεται και καθυστέρηση (lag). Η καθυστέρηση είναι το μήκος του κάθε τμήματος στα οποία κομματιάζεται το αρχικό σήμα. Αυτή η τιμή δεν επιλέγεται τυχαία και η διαδικασία επιλογής του θα αναλυθεί στο κεφάλαιο 4.7.1. Όταν το pitch κάθε τμήματος μήκους N' προσδιοριστεί, τότε εξάγεται η γραφική παράσταση του pitch (pitch contour).

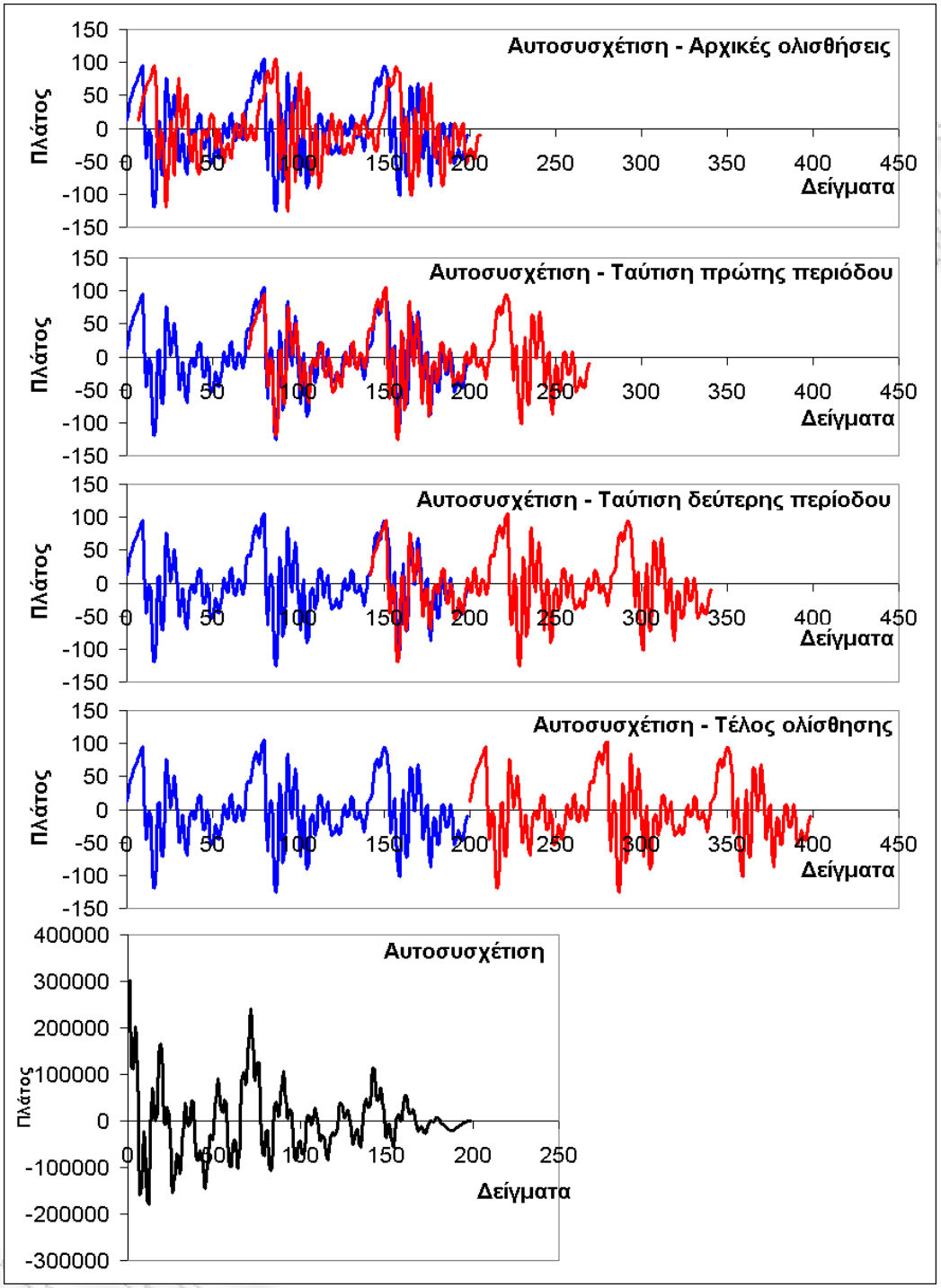
Για απλοποίηση παρατίθεται η μορφή του αλγορίθμου και σε αυτή τη μορφή, η οποία είναι και αυτή που εκτελεί το πρόγραμμα. Το x(n) ορίζεται μόνο από 0 ως N-1.

$$\Phi_{xx}(m) = \sum_{n=0}^{N-1} x(n)x(n+m)$$

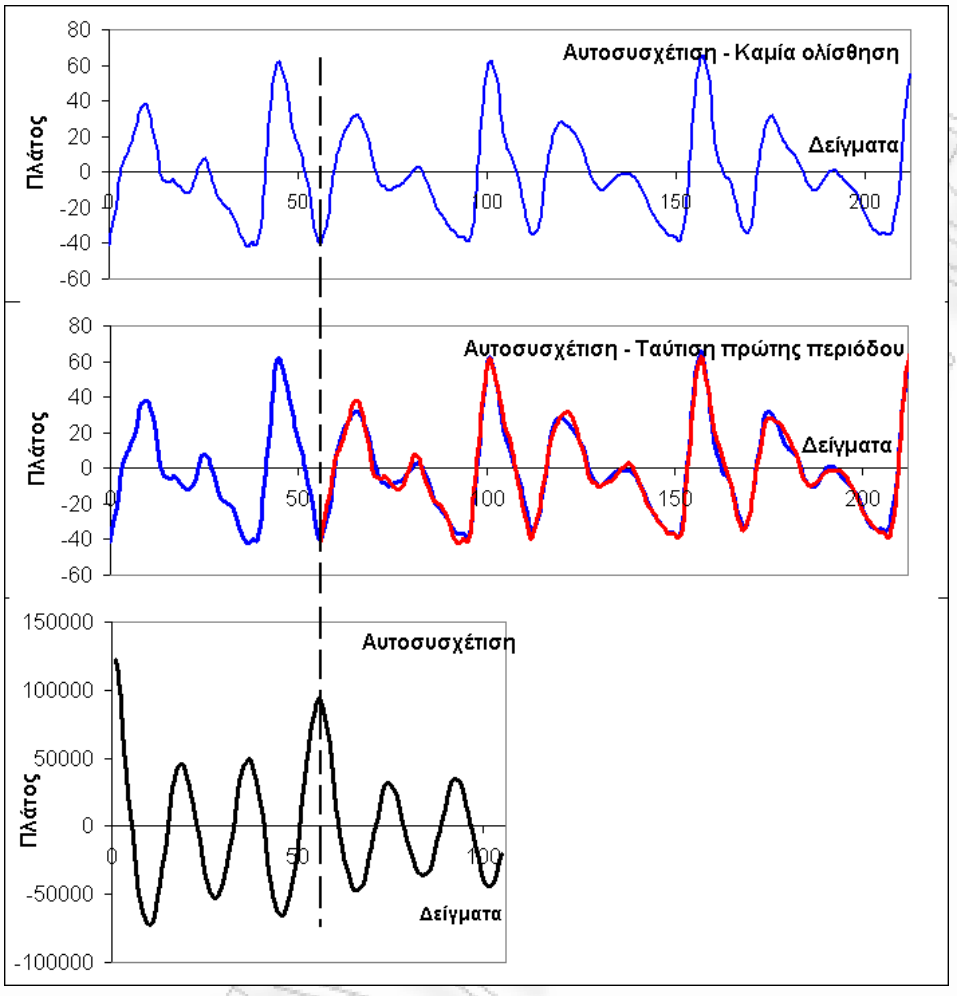
Μια γραφική απεικόνιση της αυτοσυσχέτισης δυο τμημάτων ενός σήματος φωνής δίνεται στο σχήμα 4-13. Μια ακόμα αυτοσυσχέτιση με μια πιο «ομαλή» φωνή δίνεται στο σχήμα 4-14. Τονίζεται η διαφορά των αποτελεσμάτων.

Στα σχήματα αυτά, στην πρώτη εικόνα, οι κυματομορφές σχεδόν ταυτίζονται και οι υπόλοιπες εικόνες δείχνουν μια σταδιακή ολίσθηση προς τα δεξιά. Η τελευταία εικόνα κάθε σχήματος είναι η συνάρτηση της αυτοσυσχέτισης. Στο σημείο που αυτή μεγιστοποιείται, υπάρχει η μέγιστη ταύτιση των κυματομορφών, εκεί που ξεκινάει η κατακόρυφη γραμμή.

Δίνεται έμφαση στη διαφορά τάξεων μεγέθους των τιμών του πλάτους των κυματομορφών και της αυτοσυσχέτισης.

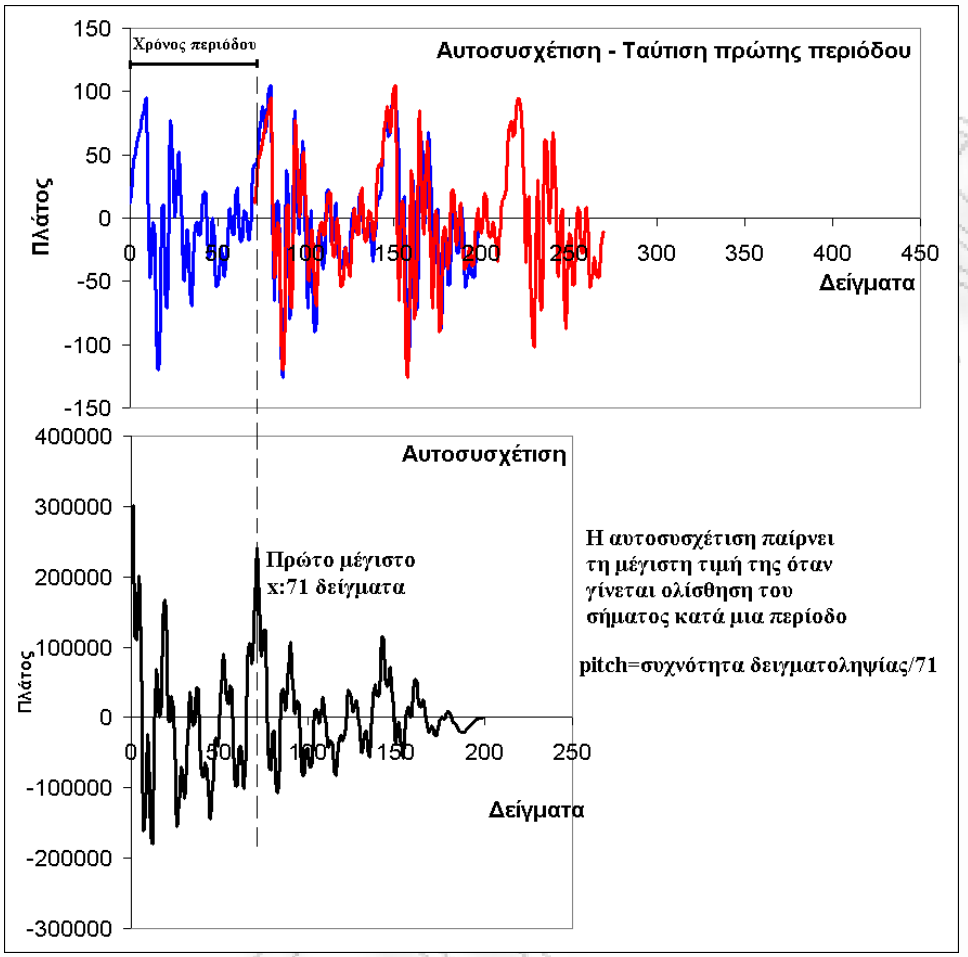


Εικόνα 4-14: Αυτοσυσχέτιση τμήματος ήχου



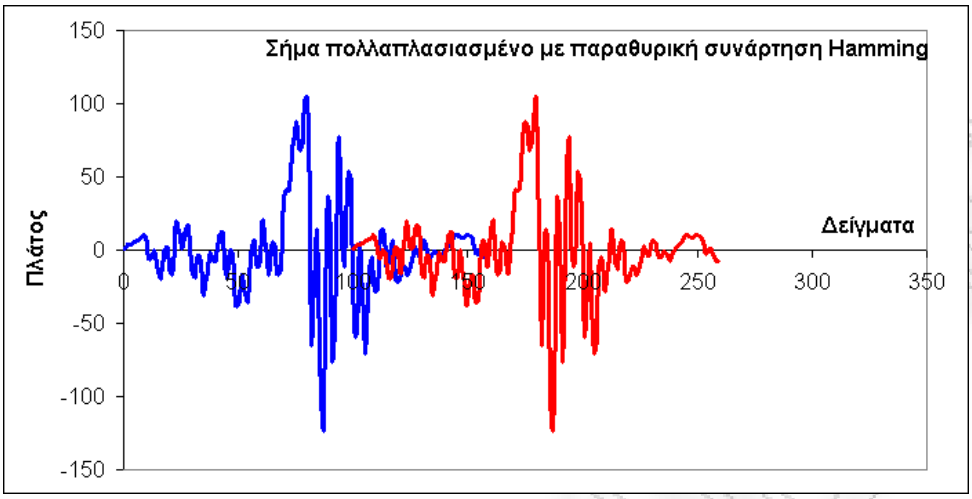
Εικόνα 4-15: Αυτοσυσχέτιση πιο "ομαλού" ήχου

Η γραφική παράσταση της αυτοσυσχέτισης μαζί με την κυματομορφή που δίνει τη μέγιστη τιμή της, δίνονται στο σχήμα 4-15



Εικόνα 4-16: Μέγιστο αυτοσυσχέτισης στην ταύτιση της πρώτης περιόδου

Από τα σχήματα πλέον είναι φανερός ο λόγος που η παραθυρική συνάρτηση επιλέχθηκε να είναι τετραγωνική. Αν είναι οποιαδήποτε άλλη, το σήμα θα χάνει το πλάτος σύγκρισής του καθώς ολισθαίνει, οι κυματομορφές δε θα ταυτίζονται σε κάθε περίοδο και η αυτοσυσχέτιση θα δίνει αποτελέσματα που δε μπορούν να μεταφραστούν με ακρίβεια. Η μη ταύτιση του σήματος δίνεται γραφικά στο σχήμα 4-16.



Εικόνα 4-17: Αυτοσυσχέτιση με σήμα πολλαπλασιασμένο με παραθυρική συνάρτηση Hamming

Όπως αναφέρθηκε και προηγουμένως, ο χρόνος μετριέται με τον αριθμό των δειγμάτων και τον ρυθμό δειγματοληψίας του σήματος. Για οποιαδήποτε διαφορά μεταξύ δειγμάτων του σήματος, με την απλή μέθοδο των τριών, ισχύει πως

Δχρόνου = αριθμός δειγμάτων / συχνότητα δειγματοληψίας ⇔

συχνότητα = συχνότητα δειγματοληψίας / αριθμός δειγμάτων

Η αυτοσυσχέτιση παίρνει τη μέγιστη τιμή της όταν γίνεται ολίσθηση του σήματος κατά αριθμό δειγμάτων m. Αυτό σημαίνει, όπως εξηγήθηκε προηγουμένως, πως η κυματομορφή επαναλαμβάνεται ανά m δείγματα, άρα είναι περιοδική με περίοδο ίση με χρόνο που αντιστοιχεί σε m δείγματα. Οπότε η συχνότητα ισούται με:

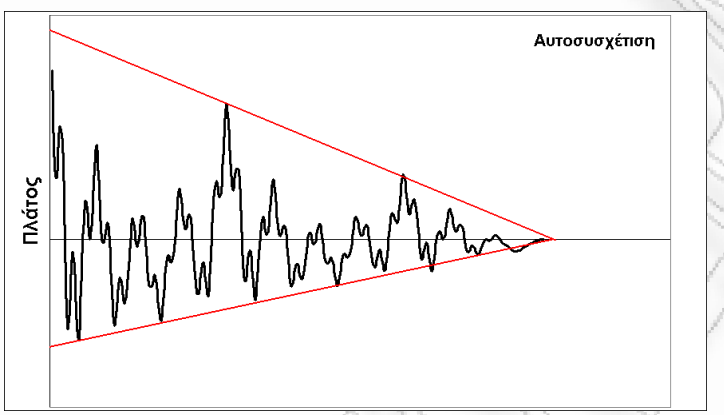
$$f_0 = pitch = f_s / m$$

Και στα δυο σχήματα η συχνότητα δειγματοληψίας είναι 8 kHz. Στο πρώτο σχήμα το pitch βγαίνει 8000/71 = 112.7 Hz και στο δεύτερο σχήμα βγαίνει 8000/56 = 142.9 Hz.

Μια ακόμα παρατήρηση είναι πως στο δεύτερο μέγιστο της γραφικής παράστασης, αντιστοιχεί χρόνος ολίσθησης που είναι ίσος με 2m δείγματα ήχου, δηλαδή ακόμα μια περίοδος, δηλαδή η μισή συχνότητα. Αν ο αλγόριθμος δεχτεί αυτή τη τιμή αντί για την πρώτη, θα δώσει αποτελέσματα αρμονικής συχνότητας, που είναι λάθος.

Το ότι στη μισή συχνότητα, αντιστοιχεί τοπικό μέγιστο που είναι αριθμητικά μικρότερο από το πρώτο μέγιστο, οφείλεται στο ότι ο αλγόριθμος είναι ένα άθροισμα που αποτελείται από όλο και λιγότερα μέρη όσο περνάει ο χρόνος, καθώς, ότι ξεπερνάει τα όρια των επικαλυπτόμενων

κυματομορφών, μηδενίζεται από τη παραθυρική συνάρτηση w(n). Για αυτό το λόγο φαίνεται υπάρχει και η ενδιαφέρον ιδιότητα της συγκεκριμένης υλοποίησης της αυτοσυσχέτισης, ότι, καθώς ο χρόνος περνάει, μοιάζει να μειώνεται το πλάτος της σταθερά, μέχρι να μηδενίσει, όπως φαίνεται και στο σχήμα 4-17.



Εικόνα 4-18: Σταδιακή μείωση πλάτους της συνάρτησης αυτοσυσχέτισης

4.6.1 Επιλογή τιμής καθυστέρησης (lag)

Όπως αναφέρθηκε σε προηγούμενο κεφάλαιο (2.3.4) όρια της συχνότητας της ανθρώπινης φωνής είναι για τους άντρες από 85 Hz ως 180 Hz και για τις γυναίκες από 165 Hz ως 265 Hz περίπου, φυσικά μπορεί να υπάρχουν εξαιρέσεις αλλά δε λαμβάνονται υπ' όψη σε αυτή την εργασία. Αυτό σημαίνει πως οι τυπικές επαναλήψεις της κυματομορφής θα κινούνται σε αυτά τα όρια, δηλαδή ισχύει ο πίνακας 3

	Άντρες		Γυναίκες	
	Συχνότητα (Hz)	Περίοδος (seconds)	Συχνότητα (Hz)	Περίοδος (seconds)
Ελάχιστο	85	0.012	165	0.006
Μέγιστο	180	0.0056	265	0.0038

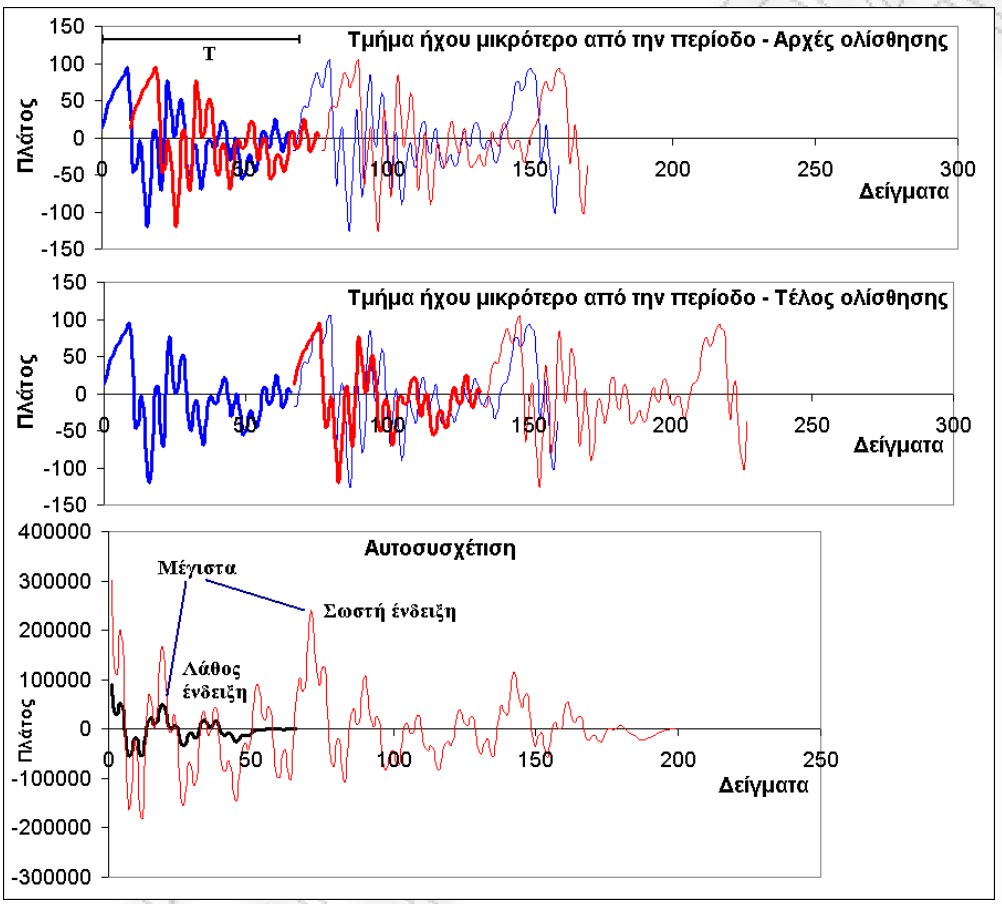
Πίνακας 2 όρια ανθρώπινης φωνής

Για να είναι σίγουρο πως ο αλγόριθμος θα μπορέσει να λειτουργήσει σωστά, δηλαδή να ανιχνεύσει οποιαδήποτε από τις συχνότητες που αναφέρθηκαν παραπάνω, πρέπει να επεξεργαστεί τις κυματομορφές της φωνής, στον μεγαλύτερο από αυτούς τους χρόνους, δηλαδή την ελάχιστη αντρική συχνότητα.

Επίσης, αφού ο αλγόριθμος ψάχνει για επαναλήψεις της ίδιας κυματομορφής μέσα στο χρονικό παράθυρο, πρέπει το παράθυρο αυτό να είναι αρκετά μεγάλο ώστε να περιλαμβάνει τουλάχιστο δυο περιόδους του σήματος, έτσι ώστε να συμπέσουν τα όμοια τμήματα και να πάρει η αυτοσυσυσχέτιση τη μέγιστη τιμή. Οπότε το χρονικό παράθυρο θα είναι T1 = 1/85 * 2 = 0.024 second = 24 msec. Σε

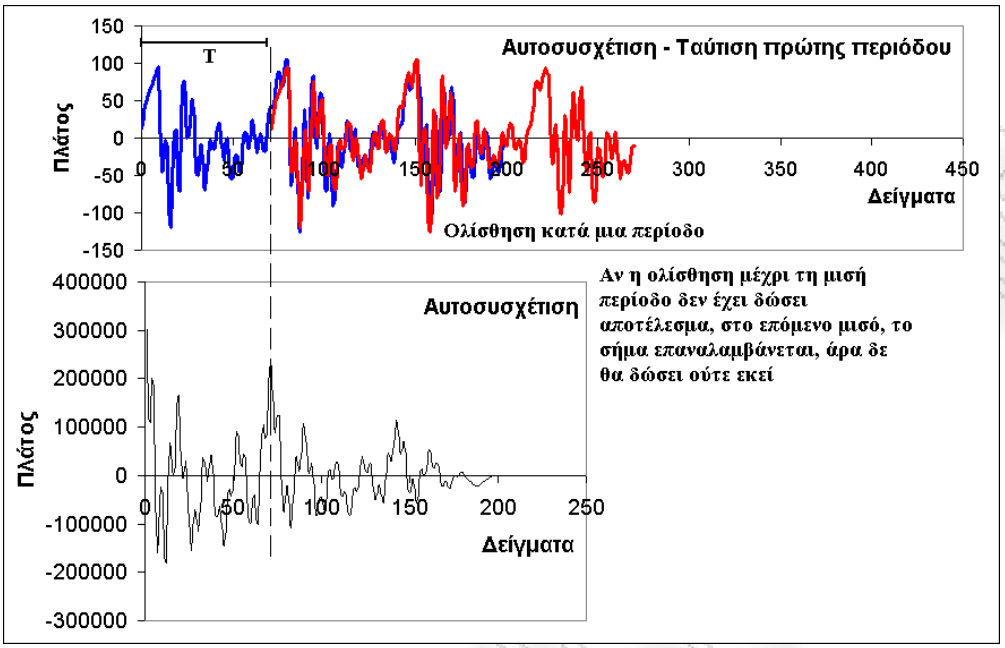
περίπτωση επιλογής μικρότερου χρόνου, T2, έστω T2 < T1, χάνονται συχνότητες. Αυτό φαίνεται και γραφικά στο σχήμα 4-18. Οι γραμμές που δεν είναι πολύ έντονες αντιπροσωπεύουν το πως θα συνεχίζονταν οι κυματομορφές και δε λαμβάνουν τόπο στους υπολογισμούς της αυτοσυσχέτισης.

Όταν τελικά όλο το σήμα αναλυθεί σε τμήματα των 24 msec και για κάθε ένα από αυτά τα τμήματα βρεθεί το pitch, θα γίνει η γραφική παράσταση του pitch για όλη τη διάρκεια της ομιλίας.



Εικόνα 4-19: Λάθος αποτελέσματα στην περίπτωση επιλογής καθυστέρησης μικρότερης από την περίοδο

Μια ακόμα σημαντική παρατήρηση είναι πως ενώ το χρονικό παράθυρο του σήματος πρέπει να είναι το διπλάσιο από τη μικρότερη περίοδο, η ολίσθηση που χρειάζεται να κάνει το σήμα, είναι ο μισός από αυτό τον χρόνο, δηλαδή το σήμα χρειάζεται να ολισθήσει τόσο χρόνο όσο είναι η περίοδός του. Αυτό επειδή, μέσα στον χρόνο της μικρότερης περιόδου, θα υπάρχει σίγουρα το επαναλαμβανόμενο μοτίβο μια φορά (μια φορά το λιγότερο, στην περίπτωση που πρόκειται για τη μικρότερη αντρική συχνότητα), οπότε η ολίσθηση του σήματος θα προλάβει να δώσει μέγιστο στην ταύτιση των κυματομορφών. Ένα γραφικό παράδειγμα φαίνεται και στο σχήμα 4-19.



Εικόνα 4-20: Μετά την ταύτιση της πρώτης περιόδου δε χρειάζεται παραπάνω ολίσθηση

Ένα σημείο που τονίζεται είναι πως η επιλογή του πρώτου τοπικού μέγιστου, μειώνει κατά πολύ το χρόνο εκτέλεσης του αλγορίθμου αυτοσυσχέτισης, αυτό επειδή μόλις βρει το τοπικό μέγιστο σταματάει και επιστρέφει την τιμή του pitch στο συγκεκριμένο τμήμα της κυματομορφής. Αυτός είναι και ο λόγος που στις κυματομορφές των παραδειγμάτων, οι κυματομορφές αρχικά ταυτίζονται και έπειτα ολισθαίνουν, απομακρυνόμενες από την ταύτιση. Αν οι κυματομορφές ήταν στην αρχή απομακρυσμένες και ολίσθαιναν προς την ταύτισή τους, θα έπρεπε να εκτελεστεί ο αλγόριθμος σε όλο το χρονικό διάστημα των δυο περιόδων και μετά να μετρηθεί το τελευταίο τοπικό μέγιστο. Είναι θέμα βελτιστοποίησης του αλγορίθμου. Πράγματι, συμφέρει να μην υπολογιστεί όλη η αυτοσυσχέτιση επειδή είναι χρονοβόρα διαδικασία.

4.6.2 Ψευδοσυχνότητες

Όπως αναφέρθηκε προηγουμένως, ο ελάχιστος χρόνος που επιλέγεται για εξαγωγή pitch, είναι 24 msec. Μια κυματομορφή ήχου από άνθρωπο με υψηλό pitch (π.χ. μια γυναίκα) θα συναντήσει πολλές περιόδους μέσα στο διάστημα των 24 msec, οπότε και θα δίνει πολλές φορές τοπικό μέγιστο σε αυτό το χρονικό διάστημα.. Στο σχήμα 4-20, δίνονται μια κυματομορφή υψηλού pitch και μια χαμηλού pitch και στο σχήμα 4-21 δίνεται η σύγκριση των δυο γραφικών παραστάσεων της αυτοσυσχέτισης τους

Εικόνα 4-21: Φωνή υψηλού pitch και χαμηλού pitch

Εικόνα 4-22: Αυτοσυσχέτιση των δυο φωνών

Φαίνεται πως, ενώ στη συνάρτηση της αυτοσυσχέτισης της φωνής υψηλής συχνότητας το πρώτο μέγιστο είναι και το σωστό (η κυματομορφή ήταν πολύ ομαλή για φωνή οπότε προέκυψαν καλά αποτελέσματα), στην άλλη κυματομορφή, υπάρχουν πολλά τοπικά μέγιστα πριν το σωστό. Αυτό συμβαίνει επειδή κατά τη διάρκεια της ολίσθησης, υπάρχουν τυχαία μοτίβα στη φωνή που ταυτίζονται για λίγο.

Για να εξαλειφτεί αυτό το φαινόμενο, εφαρμόζεται η προσθήκη ενός κατωφλίου. Για να θεωρηθεί ένα τοπικό μέγιστο ικανό να δίνει ταύτιση, πρέπει να ξεπερνάει αυτό το κατώφλι. Στο σχήμα 4-22 δίνεται έμφαση στα διάφορα τοπικά μέγιστα και στις τιμές κατωφλίου. Πρέπει να τονιστεί επίσης, πως οι τιμές που δίνουν συχνότητες που πέφτουν έξω από τα όρια την ανθρώπινης φωνής απορρίπτονται. Ειδικά στο σχήμα αυτό, η συχνότητα της γυναικείας φωνής φτάνει και τα 420 Hz, το οποίο σημαίνει πως θα απορρίπτονταν χωρίς τις κατάλληλες παρεμβάσεις στο λογισμικό (αυτό υπάρχει για να δείξει πως υπάρχουν ακραίες περιπτώσεις στην εξαγωγή pitch).

Εικόνα 4-23: Επιλογή κατωφλίου

Δυστυχώς το κατώφλι δεν είναι τόσο αξιόπιστη λύση, αφού συνήθως βρίσκεται εμπειρικά. Αυτό είναι μεγάλο μειονέκτημα αυτού του αλγορίθμου όταν εφαρμόζεται κανονικά σε μια πηγή ήχου.

4.6.3 Ακρίβεια αλγορίθμου

Ο τρόπος που μετριέται ο χρόνος στον αλγόριθμο, είναι με αριθμό δειγμάτων. Όπως αναφέρθηκε προηγουμένως, ισχύει το εξής:

$$f_0 = pitch = f_s / m$$

Όπου m είναι ο αριθμός δειγμάτων στον οποίο προέκυψε το pitch. Με αυτόν τον τύπο, μπορεί να βρεθεί για κάθε ρυθμό δειγματοληψίας το πιθανό λάθος του αλγορίθμου. Για παράδειγμα, όταν σε ήχο με ρυθμό δειγματοληψίας 8 kHz βρεθεί pitch στο 35^o δείγμα, σημαίνει πως το pitch είναι

$$pitch_{35} = \frac{8000}{35} = 228.5Hz$$

Ενώ αν βρεθεί στο 36^o δείγμα,

$$pitch_{36} = \frac{8000}{36} = 222.2Hz$$

Παρατηρείται πως υπάρχει μια απόσταση 6.5 Hz από το ένα δείγμα ως το άλλο. Δηλαδή το pitch στο 36 δείγμα μπορεί να σημαίνει πως το pitch είναι από 219 ως 225.5 Hz. Αντίστοιχα, για ανίχνευση μέγιστου στο 50^{o} δείγμα και στο 51^{o} δείγμα, ισχύει πως:

$$pitch_{50} = \frac{8000}{50} = 160, pitch_{51} = \frac{8000}{51} = 156.8$$

Και η διαφορά είναι 3.2 Hz. Όσο πιο χαμηλές συχνότητες, τόσο μεγαλύτερη είναι η ακρίβεια. Παρακάτω δίνονται 4 πίνακες, ο κάθε ένας για διαφορετική συχνότητα δειγματοληψίας, 8, 16, 32 και 44,1 kHz. Δίνονται δείγματα, που σε κάθε συχνότητα αντιστοιχούν στις συχνότητες 100, 120, 140, 160 Hz, οι οποίες αντιστοιχούν σε αντρική φωνή και συχνότητες 180, 200, 225 και 250 Hz που αντιστοιχούν σε γυναικεία φωνή. Για κάθε δείγμα δίνεται το pitch, η μικρότερη πιθανή τιμή και η μεγαλύτερη πιθανή τιμή για αυτό το pitch.

Αριθμός δείγματος pitch		**Συχνότητα δειγματοληψίας 8 kHz**		
		Χαμηλότερη τιμή	**Μετρούμενη τιμή**	**Μεγαλύτερη τιμή**
Αντρική φωνή	80	100.629	100	99.3789
	66	122.137	121.212	120.301
	57	141.593	140.351	139.13
	50	161.616	160	158.416
Γυναικεία φωνή	44	183.908	181.818	179.775
	40	202.532	200	197.531
	35	231.884	228.571	225.352
	32	253.968	250	246.154

Πίνακας 3: Ακρίβεια Αυτοσυσχέτισης σε σχέση με τη συχνότητα δειγματοληψίας – 8 kHz

Αριθμός δείγματος pitch	**Συχνότητα δειγματοληψίας 16 kHz**

		Χαμηλότερη τιμή	Μετρούμενη τιμή	Μεγαλύτερη τιμή
Αντρική φωνή	160	100.313	100	99.6885
	133	120.755	120.301	119.85
	114	140.969	140.351	139.738
	100	160.804	160	159.204
Γυναικεία φωνή	88	182.857	181.818	180.791
	80	201.258	200	198.758
	71	226.95	225.352	223.776
	64	251.969	250	248.062

Πίνακας 4: Ακρίβεια Αυτοσυσχέτισης σε σχέση με τη συχνότητα δειγματοληψίας – 16 kHz

Αριθμός δείγματος pitch		**Συχνότητα δειγματοληψίας 32 kHz**		
		Χαμηλότερη τιμή	**Μετρούμενη τιμή**	**Μεγαλύτερη τιμή**
Αντρική φωνή	320	100.156	100	99.844
	266	120.527	120.301	120.075
	228	140.659	140.351	140.044
	200	160.401	160	159.601
Γυναικεία φωνή	177	181.303	180.791	180.282
	160	200.627	200	199.377
	142	226.148	225.352	224.561
	128	250.98	250	249.027

Πίνακας 5: Ακρίβεια Αυτοσυσχέτισης σε σχέση με τη συχνότητα δειγματοληψίας – 32 kHz

Αριθμός δείγματος pitch		**Συχνότητα δειγματοληψίας 44,1 kHz**		
		Χαμηλότερη τιμή	**Μετρούμενη τιμή**	**Μεγαλύτερη τιμή**
Αντρική φωνή	441	100.114	100	99.8867
	367	120.327	120.163	120
	315	140.223	140	139.778
	275	160.656	160.364	160.073

Γυναικεία φωνή	245	180.368	180	179.633
	220	200.911	200.455	200
	196	225.575	225	224.427
	176	251.282	250.568	249.858

Πίνακας 6: Ακρίβεια Αυτοσυσχέτισης σε σχέση με τη συχνότητα δειγματοληψίας – 44.1 kHz

Μια σημαντική παρατήρηση στον αλγόριθμο αυτόν είναι πως δεν επηρεάζεται η απόδοσή του από τον θόρυβο ή από την κβάντιση του σήματος. Δουλεύει εξίσου καλά στα 8, 10, 12 bit ανάλυσης.

4.6.4 Αξιοπιστία αλγορίθμου

Το βασικότερο μειονέκτημα που αντιμετωπίζει ο αλγόριθμος αυτός είναι οι αρμονικές συχνότητες. Η ύπαρξη αρμονικών με μεγάλο πλάτος καθώς και η ύπαρξη υποβιβασμένης θεμελιώδης συχνότητας συχνά μπορεί να μπερδέψει τον αλγόριθμο.

Υπάρχουν αρκετοί τρόποι για παράκαμψη αυτού του φαινόμενου όπως η κατάλληλη επιλογή κατωφλίου η οποία εξηγήθηκε παραπάνω, ή η προσωρινή κράτηση στη μνήμη προηγούμενων τιμών που βρέθηκαν και σύγκριση με την τελευταία τιμή.

Εδώ πρέπει να τονιστεί, πως αν ο ήχος προς επεξεργασία έχει περάσει προηγούμενος από κανονικοποίηση, το κατώφλι μπορεί να παραμείνει σταθερό για διαφορετικές λέξεις. Αυτό επειδή οι τιμές του πλάτους της κυματομορφής παραμένουν στα ίδια επίπεδα για τις διάρκειες των ηχογραφήσεων μικρών λέξεων, έτσι τα αθροίσματα και οι πολλαπλασιασμοί κυμαίνονται περίπου στις ίδιες τιμές. Στην εργασία αυτή, το κατώφλι ρυθμίστηκε μια φορά στην αρχή, για μια ηχογράφηση που πέρασε από κανονικοποίηση. Έπειτα, έγινε κανονικοποίηση της κάθε ηχογράφησης χωρίς την αλλαγή κατωφλίου και η επεξεργασία έδινε αξιόπιστα αποτελέσματα. Τα πράγματα γίνονται πιο δύσκολα στη περίπτωση επεξεργασίας σε πραγματικό χρόνο, αφού το σήμα δεν κανονικοποιείτε (ή είναι χρονοβόρο να κανονικοποιηθεί), οπότε το κατώφλι πρέπει να αλλάζει συνέχεια. Τονίζεται πως παρ' όλο που το κατώφλι δούλεψε καλά, το πρόγραμμα που υλοποιήθηκε χρησιμοποιεί την εύρεση της μέγιστης τιμής για μεγαλύτερη ασφάλεια.

Ένα ακόμα θέμα είναι η ύπαρξη μιας σταθερής τάσης (DC) στο σήμα. Αυτό όμως επιλύεται εύκολα με την προσθήκη ενός φίλτρου υψηλών συχνοτήτων.

4.7 Αλγόριθμος μέσου όρου διαφοράς τάξεως (Average Magnitude Difference Function – AMDF)

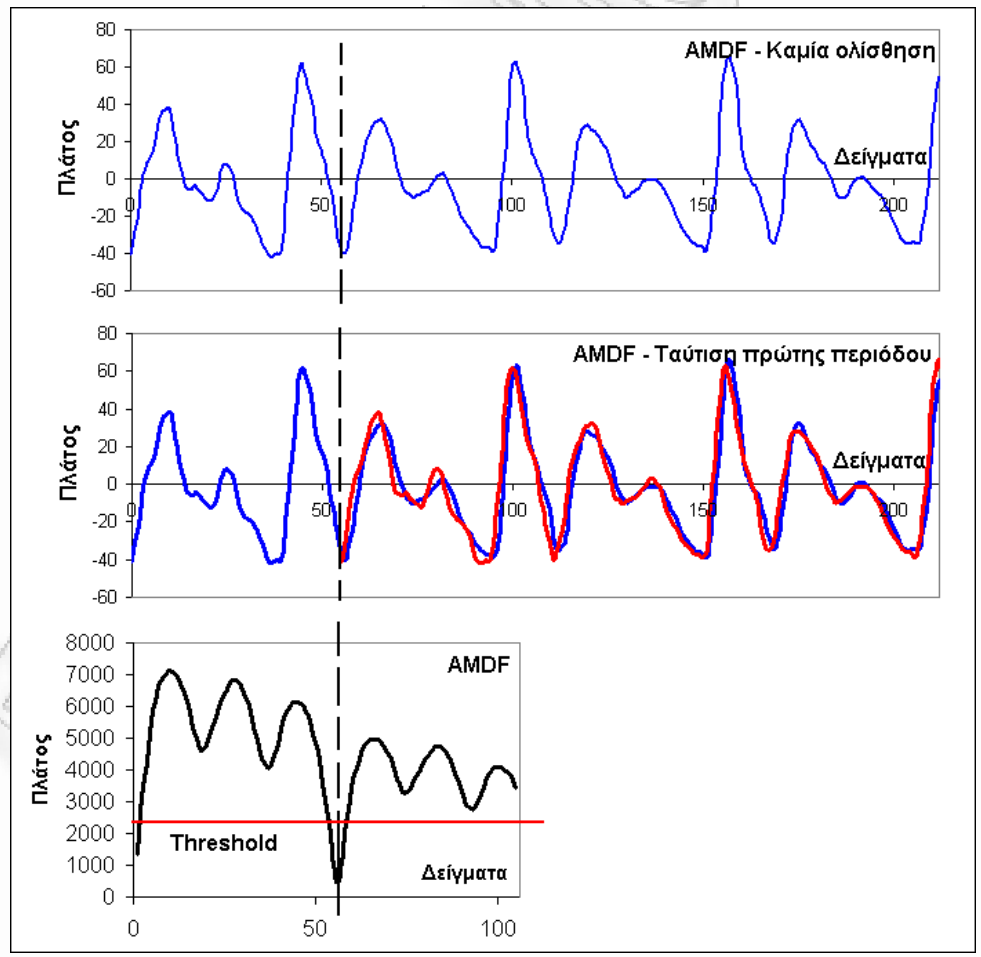
Ο αλγόριθμος AMDF θεωρείται αλγόριθμος αυτοσυσχέτισης. Τα αποτελέσματα που παράγει εξηγούνται όπως και στον ACF και ο μαθηματικός του τύπος εξηγείτε με την ίδια λογική της ολίσθησης. Ο τύπος είναι

$$\Phi_{xx}(m) = \sum_{n=0}^{N-1} x(n) - x(n+m) \text{ [28]}$$

Φαίνεται πως αντί για πολλαπλασιασμό υπάρχει αφαίρεση. Αυτό σημαίνει δυο πράγματα. Στον χρόνο της μεγαλύτερης ταύτισης, αντί για τοπικό μέγιστο θα υπάρχει τοπικό ελάχιστο. Το ζητούμενο τοπικό ελάχιστο θα είναι πιο «κοντά» στις τιμές του από τα άλλα τοπικά ελάχιστα, αυτό επειδή η αφαίρεση δίνει τάξεις μεγέθους μικρότερα αριθμητικά αποτελέσματα από τον πολλαπλασιασμό. Σαν αποτέλεσμα αυτού, είναι δυσκολότερο να επιλεγεί τιμή κατωφλίου και το ίδιο πρόβλημα που υπάρχει στον ACF υπάρχει και εδώ.

Πέρα από αυτά τα προβλήματα, είναι ένας εύκολος και ελαφρύς επεξεργαστικά αλγόριθμος, αφού ο πολλαπλασιασμός και οι αφαιρέσεις είναι πράξεις με περίπου ίδιο χρόνο εκτέλεσης σε ακέραιους αριθμούς, χωρίς μεγάλες αποκλίσεις και χρησιμοποιείτε ευρέως

Μια γραφική απεικόνιση της AMDF του τμήματος ενός σήματος φωνής δίνεται στο σχήμα 4-23 Είναι φανερή η μείωση σε τάξη μεγέθους των τιμών της συνάρτησης AMDF, συγκριτικά με την ACF, Το ACF στα προηγούμενα σχήματα δίνει μέγιστο στα 250000, ενώ το AMDF φτάνει το μέγιστο στα 7000. Στην εικόνα, το pitch βρίσκεται με τον ίδιο τρόπο όπως και στο ACF, με τη διαφορά πως, όπως αναφέρθηκε, αντί για μέγιστο υπάρχει τοπικό ελάχιστο, και μάλιστα πολύ ισχυρό, οπότε εδώ, με συχνότητα δειγματοληψίας 8 kHz, ισχύει πως pitch = 8000/56 = 142.9 Hz.



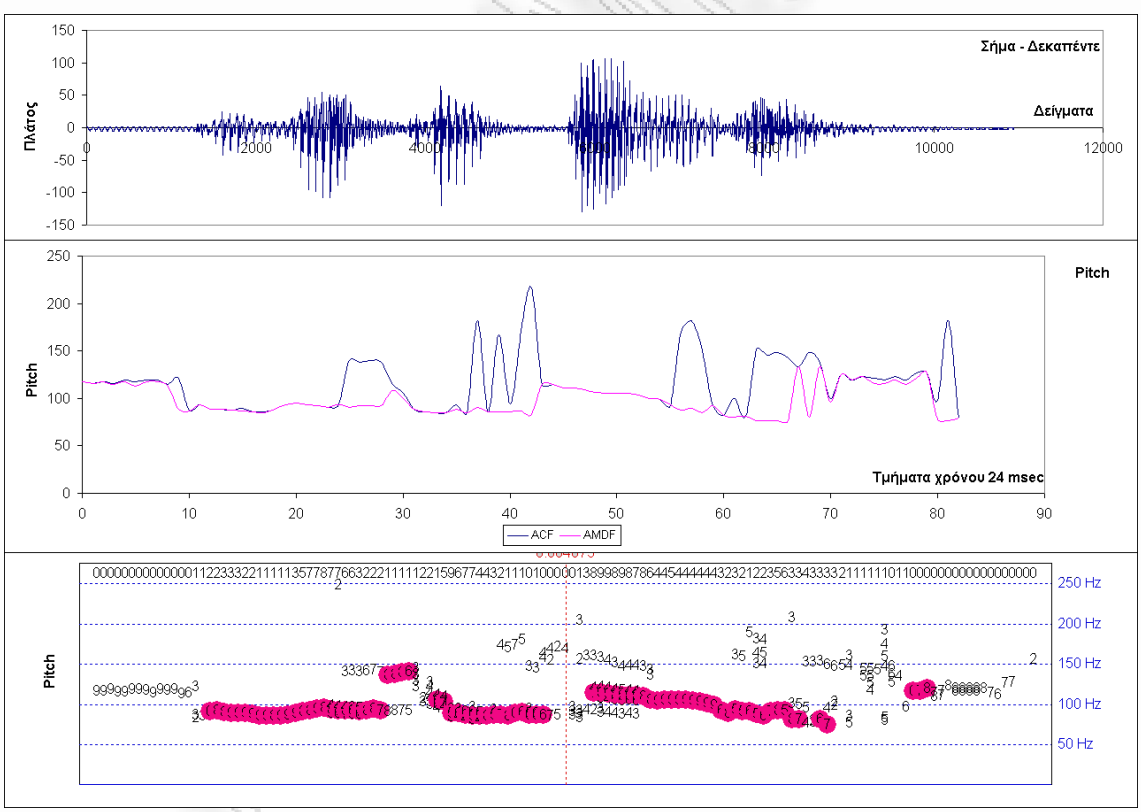
Εικόνα 4-24: Συνάρτηση AMDF

4.8 Επιλογή αλγορίθμων

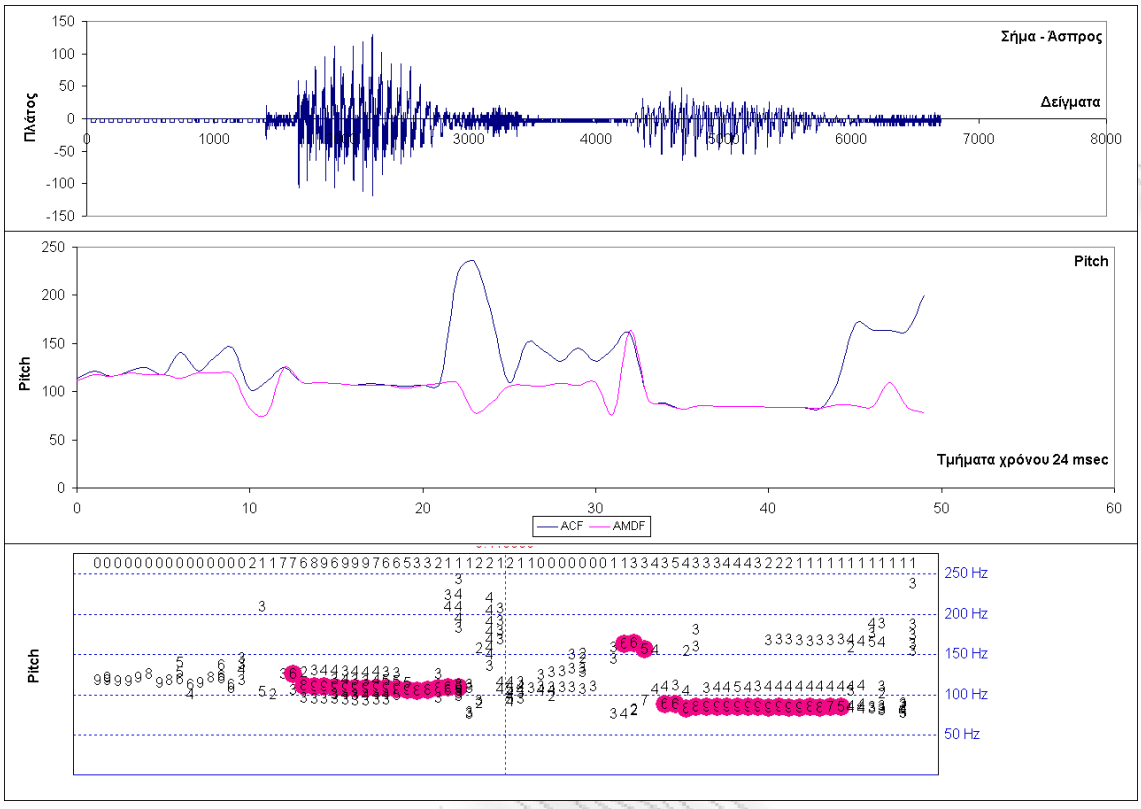
Για την εργασία αυτή επιλέχθηκαν να υλοποιηθούν δυο από τους αναφερθέντες αλγόριθμους. Αυτοί είναι ο κλασικός αλγόριθμος της αυτοσυσχέτισης (ACF), κυρίως επειδή είναι ο πιο διάσημος με τις πιο πολλές αναφορές για την αξιοπιστία του και ο AMDF, επειδή είναι γειτονικός με τον ACF, πιο γρήγορος και με επίσης καλά αποτελέσματα. Οι άλλοι αλγόριθμοι, απορριφθήκαν λόγω της μεγάλης υπολογιστικής ισχύος που απαιτείται για την εκτέλεση τους ή λόγω της πολυπλοκότητας τους. Είναι σημαντικό πως και οι δυο επιλεγόμενοι αλγόριθμοι δουλεύουν πολύ κατανοητά για τον άνθρωπο επειδή το ανθρώπινο μάτι συγκρίνει με τον ίδιο τροπο. Η σύγκριση των δυο αλγορίθμων δεν έδειξε μεγάλες αποκλίσεις στα αποτελέσματα. Και οι δυο παρουσιάζουν σχετικά καλή αξιοπιστία.

4.8.1 Γραφική παράσταση pitch (Pitch Contour)

Η τιμή της αυτοσυσχέτισης για κάθε τμήμα ήχου που αποτελείτε από 24 msec, σχηματίζει τη τελική γραφική παράσταση του pitch (pitch contour). Τα σχήματα 4-24 και 4-25 δίνουν το pitch contour δυο ηχογραφήσεων όπως υπολογίζονται από τους αλγορίθμους ACF και AMDF. Επίσης συγκρίνονται με τα αποτελέσματα του προγράμματος Praat, το οποίο χρησιμοποιεί FFT για την εξαγωγή του pitch. Ένα εμφανές πρόβλημα είναι πως στο pitch contour υπάρχουν τμήματα που το pitch φαίνεται τυχαίο (ενώ στο Praat αυτές οι περίοδοι δε παρουσιάζουν pitch). Αυτό συζητιέται αμέσως μετά.



Εικόνα 4-25: Pitch contour της λέξης "δεκαπέντε", αλγόριθμοι ACF/AMDF και σύγκριση με το πρόγραμμα Praat



Εικόνα 4-26: Pitch contour της λέξης "άσπρος", αλγόριθμοι ACF/AMDF και σύγκριση με το πρόγραμμα Praat

4.9 Προσθήκη τεχνικής κεντρικής ψαλιδοποίησης (center clipping) για όλους τους προηγούμενους αλγόριθμους

Όπως φαίνεται στα σχήματα 4-24 και 4-25, στα σημεία που δεν παρατηρείτε φωνή, δηλαδή στα σημεία που ο ήχος έχει χαμηλό πλάτος, το pitch είναι λάθος, τυχαίο. Αυτό οφείλεται ακριβώς στο γεγονός της έλλειψης φωνητικών συμφώνων και φωνηέντων, οι φωνητικές χορδές παύουν να πάλλονται, τα μόρια του αέρα ταλαντώνονται με τυχαίες κινήσεις και η συχνότητα δεν είναι σταθερή.

Για να αποφευχθεί το φαινόμενο αυτό, υπάρχει η τεχνική κεντρικής ψαλιδοποίησης (center clipping), η οποία βρίσκει αν το πλάτος ενός δείγματος είναι μεγαλύτερο από μια τιμή κατωφλίου (clipping threshold). Αν είναι μικρότερο από αυτή την τιμή, το δείγμα μηδενίζεται. Υπάρχουν τρεις μορφές του αλγορίθμου. Η πρώτη μορφή είναι η συμπιεσμένη (compressed center clipping)

$$
\begin{aligned}
y(n) = clc[x(n)] &= (x(n) - C_L), && x(n) \geq C_L \\
&= 0, && |x(n)| < C_L \\
&= (x(n) + C_L), && x(n) \leq -C_L
\end{aligned}
$$

Όπου C_L είναι το κατώφλι ψαλιδοποίησης και clc σημαίνει clip and compress Η δεύτερη μορφή είναι η απλή (simple center clipping)

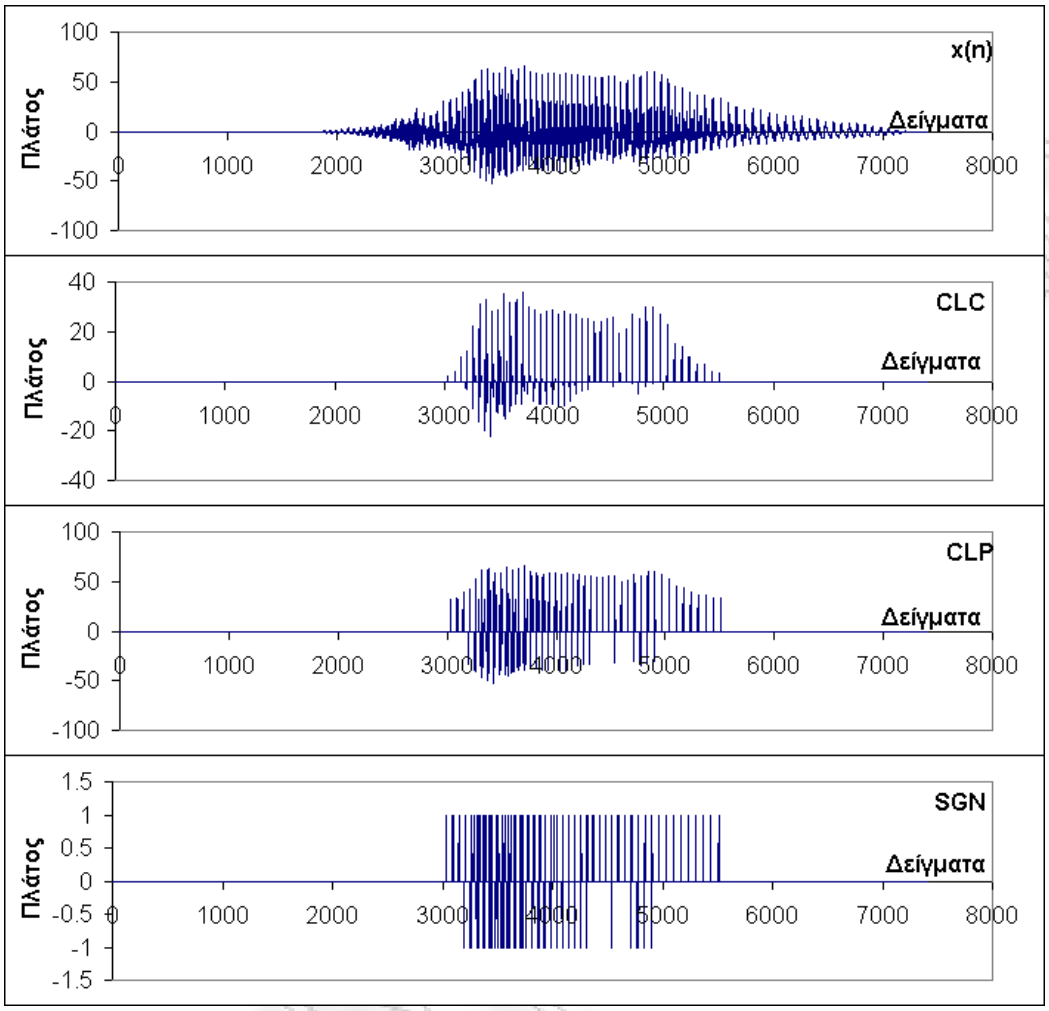
$$\begin{aligned} y(n) = clp[x(n)] &= [x(n)], && x(n) \geq C_L \\ &= 0, && |x(n)| < C_L \\ &= -x(n), && x(n) \leq -C_L \end{aligned}$$

Όπου clp είναι clip Και η τρίτη μορφή είναι ένας συνδυασμός center και compressed (peak) clipping (sgn σημαίνει sign of x)

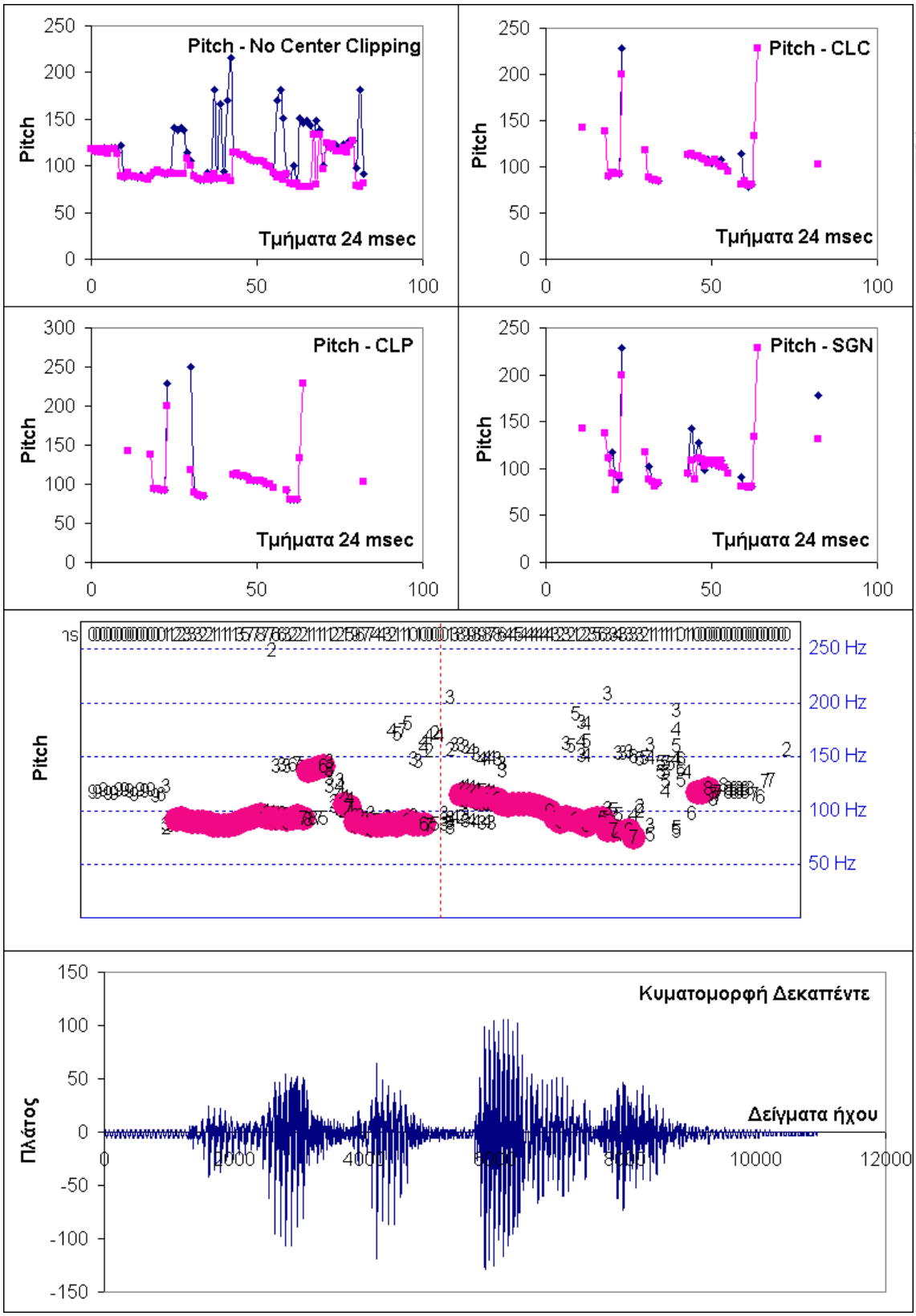
$$\begin{aligned} y(n) = \text{sgn}[x(n)] &= 1, && x(n) \geq C_L \\ &= 0, && |x(n)| < C_L \\ &= -1, && x(n) \leq -C_L \end{aligned}$$

([27] όλα τα παραπάνω)

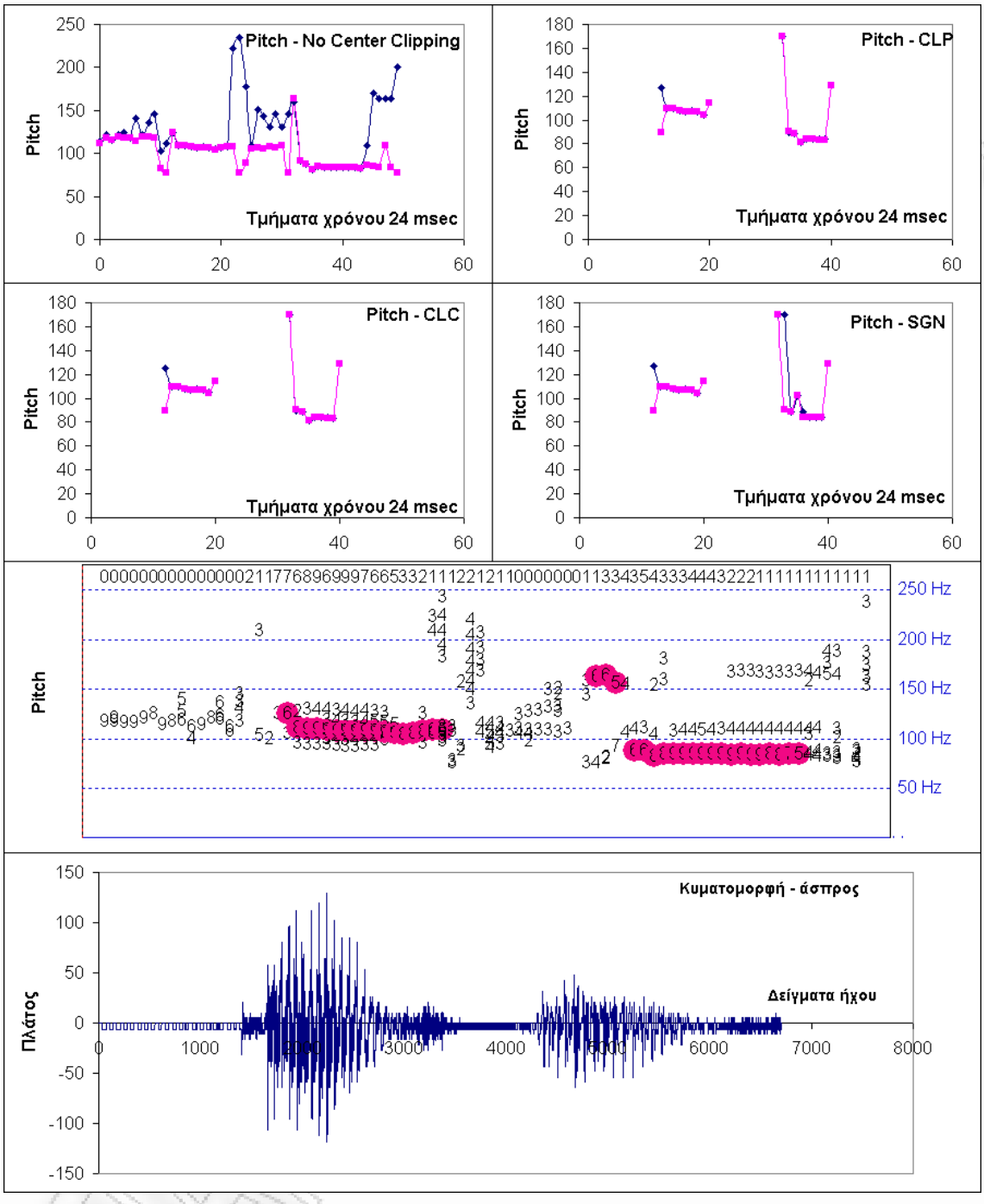
Η κάθε περίπτωση δίνει και από μια διαφορετική κυματομορφή. Το σχήμα 4-26 δίνει ένα σήμα x(n) και την εφαρμογή των διαφορετικών τεχνικών κεντρικών ψαλιδοποιήσεων πάνω στο σήμα αυτό. Η τεχνική αυτή είναι εφαρμόσιμη σε όλους τους προαναφερθέντες αλγόριθμους και στα σχήματα 4-27και 4-28, εφαρμόζονται αυτοί οι αλγόριθμοι με τιμή κατωφλίου 40 στάθμες. Με ανάλυση 8 bit, στα όρια -2.5 ως 2.5 Volt, αυτό μεταφράζεται σε τιμή κατωφλίου 0.78125 Volt. Όσα λοιπόν δείγματα του ήχου δε ξεπερνάνε την τάση των 0.78125 Volt ή -0.78125 Volt, μηδενίζονται.



Εικόνα 4-27: Αποτέλεσμα σήματος αφού πέρασε από διαφορετικές επεξεργασίες κεντρική ψαλιδοποίησης



Εικόνα 4-28: Αποτελέσματα εξαγωγής pitch contour λέξης «δεκαπέντε» αφού πέρασε από τεχνικές κεντρικής ψαλιδοποίησης



Εικόνα 4-29: Αποτελέσματα εξαγωγής pitch contour λέξης «άσπρος» αφού πέρασε από τεχνικές κεντρικής ψαλιδοποίησης

Η τεχνική ψαλιδισμού δίνει καλά αποτελέσματα, όμως ο υπολογισμός των τιμών ψαλιδισμού είναι επίπονος επεξεργαστικά. Είναι προτιμότερο να γνωρίζει ο αλγόριθμος από πριν πια τμήματα του ήχου περιέχουν αξιοποιήσιμη πληροφορία. Η ανίχνευση των τμημάτων ήχου που περιέχουν αξιοποιήσιμη πληροφορίας αναλύεται στο επόμενο κεφάλαιο.

4.9.1 Ταχύτητα αλγορίθμων αυτοσυσχέτισης και AMDF

Η ταχύτητα των δυο υλοποιημένων αλγορίθμων συζητιέται στο κεφάλαιο «Απόδοση συστήματος – συμπεράσματα».

5 Αλγόριθμος κατάτμησης σημάτων φωνής σε ομιλία και θόρυβο (endpoint detection)

Η προ επεξεργασία του ηχητικού σήματος με την κατάτμησή του με σκοπό να βρεθεί που αρχίζει και που τελειώνει η ομιλία μέσα στον ήχο, είναι πολύ σημαντική όταν οι σιωπηλές περίοδοι και ο θόρυβος είναι ανεπιθύμητα. Εφαρμογές όπως αναγνώριση ομιλίας και φωνής χρειάζονται τεχνικές με καλά αποτελέσματα. Με αυτό τον τρόπο μειώνεται ο χρόνος επεξεργασίας πάνω σε τμήματα ήχου που δε περιλαμβάνουν χρήσιμη πληροφορία και κυρίως δεν εισάγεται θόρυβος στην επεξεργασία, που συμβάλει στην αλλοίωση χρήσιμων αποτελεσμάτων.

Στο προηγούμενο κεφάλαιο, η εξαγωγή του pitch για τα τμήματα που δεν έχουν ήχο είναι επίπονη διαδικασία, αντί να υπολογίζει ο αλγόριθμος τις τιμές ψαλιδισμού, μπορεί να χρησιμοποιήσει έναν αλγόριθμο κατάτμησης σήματος φωνής ώστε να γνωρίζει από πριν πια τμήματα σήματος περιέχουν αξιοποιήσιμη πληροφορία, Για τη πτυχιακή αυτή, ο αλγόριθμος που χρησιμοποιήθηκε είναι ο γνωστός endpoint detection αλγόριθμος του Rabiner [2][3].

5.1 Εισαγωγή

Οι αλγόριθμοι της κατάτμησης δουλεύουν όταν οι λέξεις βρίσκονται απομονωμένες μέσα στην ηχογράφηση, δηλαδή πρέπει η λέξη να ακολουθείται και να προηγείται από σιωπηλή περίοδο. Ο συγκεκριμένος αλγόριθμος του Rabiner, χρειάζεται αρχική σιωπηλή περίοδο διάρκειας τουλάχιστο 100 msec για να γίνουν υπολογισμοί για τον θόρυβο.

Έτσι, το πρόγραμμα μπορεί να υποθέσει πως οι λέξεις αυτές χωρίζονται μεταξύ τους. Η διαδικασία απομόνωσης λέξεων από το ηχητικό υπόβαθρο, το οποίο περιέχει λέξεις και σιωπηλή περίοδο (δηλαδή θόρυβο), ονομάζεται «endopoint detection».

Σύμφωνα με τον Rabiner, οι στόχοι της ανάπτυξης αυτού του αλγορίθμου είναι:

- Απλότητα
- Αξιόπιστη ανίχνευση σημαντικών ακουστικών γεγονότων
- Δυνατότητα να εφαρμοστεί σε διαφορετικά περιβάλλοντα θορύβου

Ο αλγόριθμος χωρίζεται σε τρία τμήματα, τα τμήματα αυτά είναι τα εξής, σύμφωνα με τη χρονική σειρά που εκτελούνται:

- Υπολογισμός ενέργειας του σήματος
- Προσαρμόσιμη κανονικοποιημένη στάθμη (adaptive level equalizer)
- Ανίχνευση παλμών ενέργειας

Το κάθε τμήμα εξετάζεται χωριστά στη συνέχεια

5.2 Υπολογισμός ενέργειας του σήματος

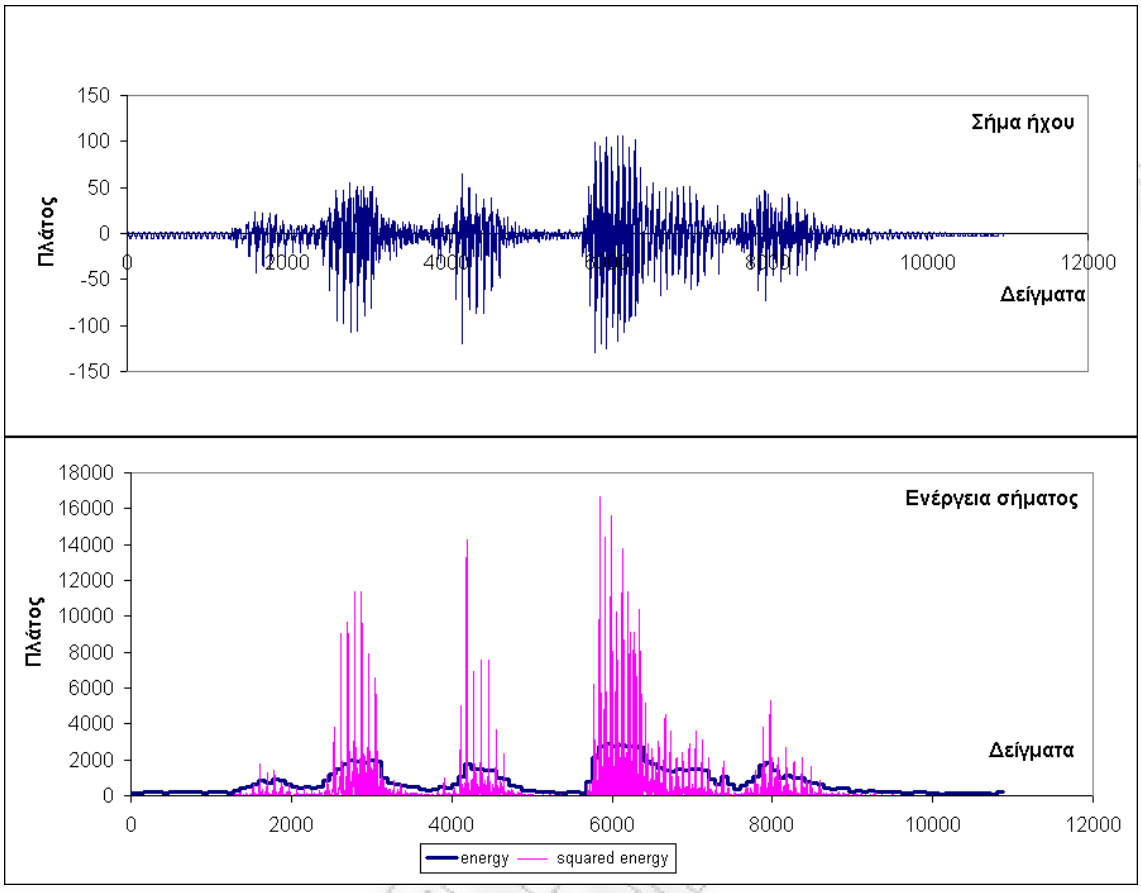
Ο αλγόριθμος αρχίζει χωρίζοντας το σήμα σε χρονικά τμήματα των 10 msec και υπολογίζει την ενέργεια για κάθε ένα από αυτά τα τμήματα. Η ενέργεια, βρίσκεται από το άθροισμα της απόλυτης τιμής κάθε τιμής του σήματος μέσα στη χρονική αυτή διάρκεια. Οπότε, όταν έχουμε ένα σήμα με συχνότητα δειγματοληψίας π.χ. 8000 Hz, το κάθε τμήμα ενέργειας, θα υπολογιστεί αθροίζοντας 80 δείγματα ήχου [2]. Ο τύπος είναι ο εξής:

$$E(n) = \sum_{i=0}^{N} |s(n+i)|$$

Ο τύπος που ο Rabiner ορίζει για την ενέργεια δεν είναι ο γνωστός τύπος των τετραγώνων. Ο λόγος είναι επειδή ο τύπος που χρησιμοποιείται, δεν τονίζει απότομες εξάρσεις πλάτους του ήχου, παράγοντας έτσι μια πιο ομαλή συνάρτηση της ενέργειας. Στον συγκεκριμένο αλγόριθμο, αυτό είναι επιθυμητό επειδή μπορεί να αφαιρεθεί ο μέσος όρος του θορύβου από το σήμα. Παράδειγμα ενός σήματος, της ενέργειας υπολογισμένη με τον τύπο

$$E(n) = \sum_{n=0}^{N} s(n+i)^2$$

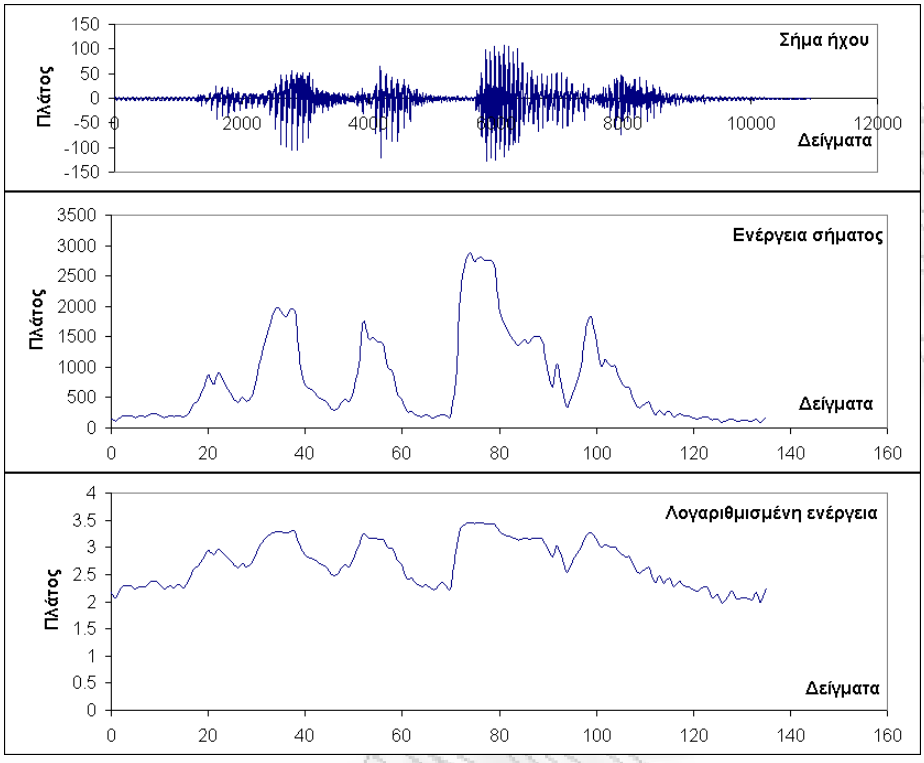
Και με τον τύπο του Rabiner, δίνεται για σύγκριση στο σχήμα 5-1.



Εικόνα 5-1: Σύγκριση ενέργειας Rabiner με ενέργεια τετραγωνοποίησης του σήματος

5.3 Προσαρμόσιμη κανονικοποιημένη στάθμη (adaptive level equalizer)

Στο στάδιο αυτό, λογαριθμείται η ενέργεια του σήματος. Αυτό γίνεται για να καταπιεστεί η ενέργεια στα ψηλά της σημεία και να ενισχυθεί ο θόρυβος, που συνήθως λαμβάνει πιο μικρές τιμές, έτσι, όταν αφαιρεθεί αργότερα η μέση τιμή του θορύβου, θα ομαλοποιηθεί καλύτερα το σήμα της λογαριθμισμένης ενέργειας. Το σχήμα 5-2 δείχνει γραφικά την επίδραση που έχει η λογαρίθμιση της ενέργειας ενός σήματος.

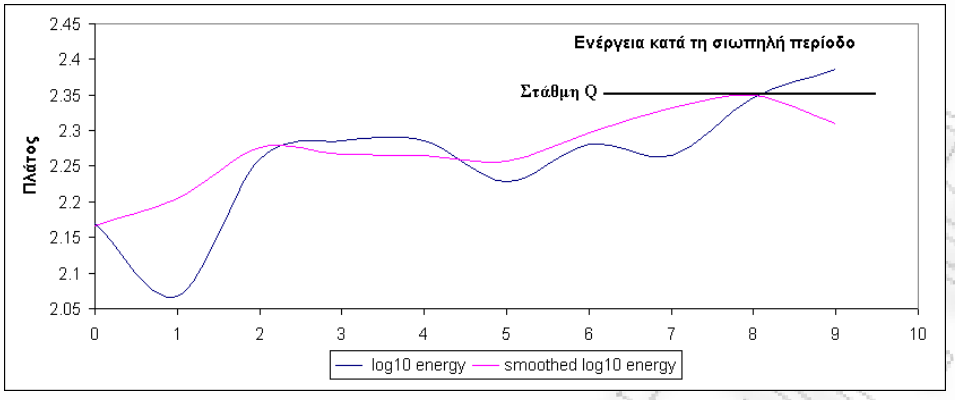


Εικόνα 5-2: Ενέργεια σήματος και λογαριθμοποίηση του

Έπειτα, βρίσκεται το κανονικοποιημένο σήμα ενέργειας, το οποίο από εδώ και πέρα θα ονομάζεται S, και το οποίο υπολογίζεται ως εξής:

$$S = \log(x(n) - Q$$

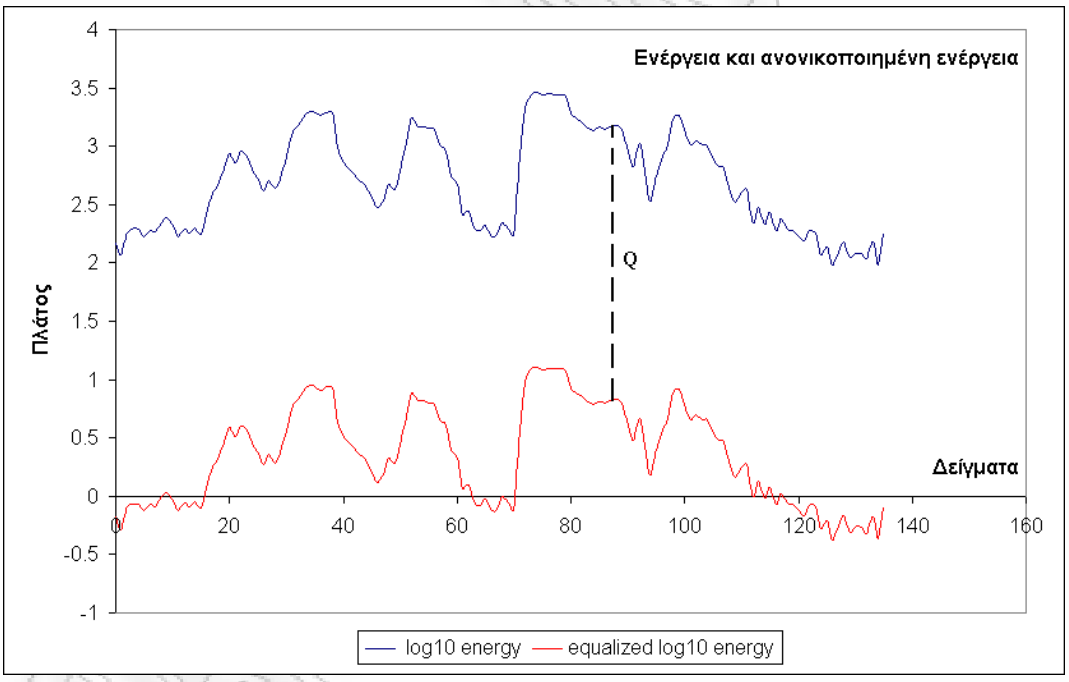
Όπου το Q είναι μια σταθερά, η οποία ορίζεται ως η μέγιστη τιμή της ομαλοποιημένης συνάρτησης μέσου όρου τριών στοιχείων της λογαριθμισμένης ενέργειας του σήματος, για τη σιωπηλή διάρκεια των 100 msec. Αυτό γίνεται για να γίνει ακόμα πιο ομαλή η συνάρτηση του θορύβου κατά τη σιωπηλή διάρκεια, εξαλείφοντας τις απότομες αλλαγές πλάτους, όπως φαίνεται στο σχήμα5-3. Έπειτα, αυτή η στάθμη αφαιρείται από όλο το σήμα,



Εικόνα 5-3: Επιλογή της στάθμης Q αφού ομαλοποιηθούν τα πρώτα 100 msec του ήχου

Το S, έχει την ιδιότητα πως, κατά την περίοδο που δεν υπάρχει φωνή, παίρνει τιμές κοντά στο μηδέν, ενώ κατά τη διάρκεια της φωνής, παίρνει σημαντικά μεγαλύτερες τιμές.

Στο σχήμα 5-4 φαίνεται ο λογάριθμος της ενέργειας, το Q και το S, δηλαδή ο λογάριθμος της ενέργειας αφού αφαιρέθηκε το Q. Είναι μια απλή αφαίρεση.



Εικόνα 5-4: Αφαίρεση της στάθμης Q από την ενέργεια

Αφού βρέθηκε το S, ο αλγόριθμος συνεχίζει για να ανιχνεύσει τους παλμούς της ενέργειας. Δηλαδή τις λέξεις.

5.4 Ανίχνευση παλμών ενέργειας

Από το κανονικοποιημένο σήμα ενέργειας, ορίζονται τέσσερις ενεργειακές στάθμες, K1, K2, K3 και K4. Σύμφωνα με τον αλγόριθμο που εξηγείται παρακάτω, μερικά σημεία τομής αυτών των σταθμών ενέργειας, με το S, θα καθορίσουν τα όρια, A1, A2, A3, A4, ανάμεσα στα οποία θα περικλείεται κάθε παλμός ενέργειας. Οι παλμοί αυτοί, θα θεωρηθούν σαν φωνή μέσα στην ηχογράφηση. Οι στάθμες K1, K2, K3 και K4 εξάγονται εμπειρικά και οι τιμές τους είναι εξαρτώμενες από την μέγιστη τιμή του S, Οι τιμές φαίνονται παρακάτω

K1 = 0

K2 = 0.17 * μέγιστη_λογαριθμημένη_ομαλοποιημένη_ενέργεια

K3 = 0.12733 * μέγιστη_λογαριθμημένη_ομαλοποιημένη_ενέργεια

K4 = μέγιστη_λογαριθμημένη_ομαλοποιημένη_ενέργεια * $\log_{10}(3)$

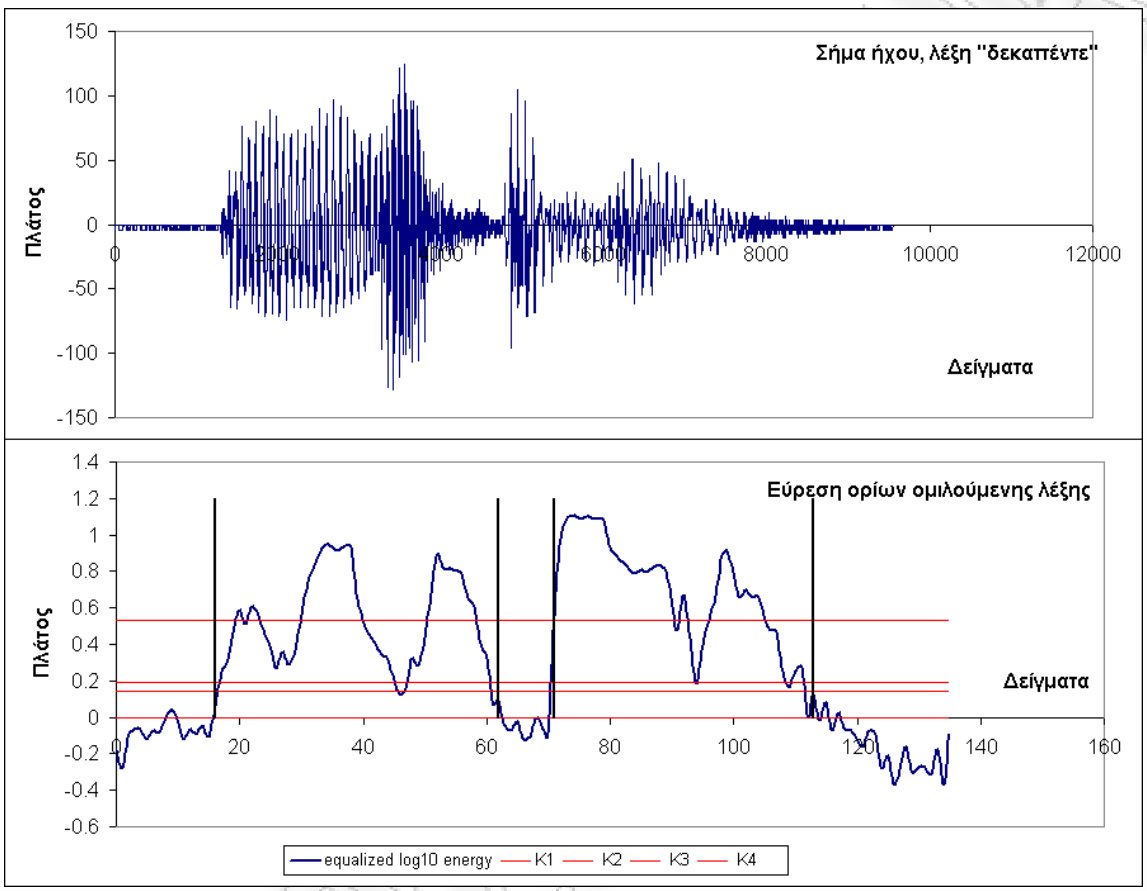
Ο αλγόριθμος σύμφωνα με τον οποίο ανιχνεύονται οι παλμοί ενέργειας εξηγείται εδώ. Οι τιμές του S, ελέγχονται διαδοχικά, από την παλαιότερη, ως την νεότερη χρονικά. Ξεκινώντας την ανίχνευση,

- Όταν μια τιμή του S, ξεπεράσει την πρώτη στάθμη K1, αυτή η τιμή καταγράφεται σαν η τιμή A1.
- Όταν μια μετέπειτα τιμή του S, ξεπεράσει την στάθμη K2, τότε η τιμή A2 καταγράφεται.
- Σε περίπτωση που το S πέσει κάτω από τη στάθμη K1 πριν ξεπεράσει τη στάθμη K2, τότε το A1 απορρίπτεται και η ανίχνευση συνεχίζει κανονικά ψάχνοντας για το A1.
- Η αρχή του παλμού ενέργειας ορίζεται σαν το A1, εκτός αν ο χρόνος ανόδου, δηλαδή η χρονική απόσταση μεταξύ A1 και A2, είναι αρκετά μεγάλη (εμπειρικά οριζόμενο), οπότε σαν αρχή της λέξης, ορίζεται το A2.
- Το τέλος του παλμού βρίσκεται με παρόμοιο τρόπο, η διαφορά είναι πως η στάθμη K3 είναι λίγο μικρότερη σε τιμή από τη στάθμη K2, αυτό γίνεται για να επιτρέψουμε πιο μεγάλους χρόνους καθόδου, το οποίο είναι παρατηρήσιμο φαινόμενο στο τέλος μιας λέξης.

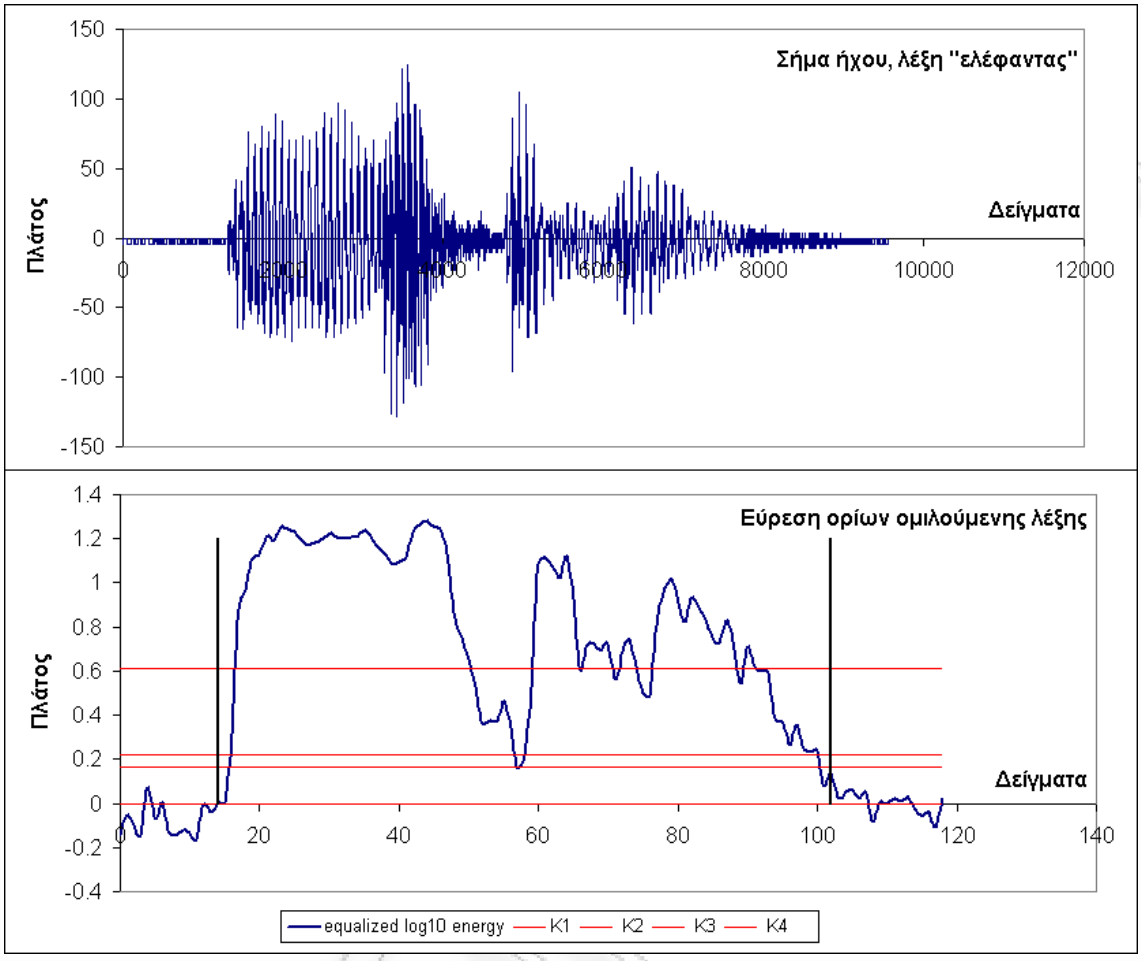
Αφού καθοριστούν τα όρια A1, A2, A3 και A4 για κάθε παλμό ενέργειας που ανιχνεύεται,

- Αν ένας παλμός ενέργειας, δε περιλαμβάνει μέσα του καμία τιμή που να ξεπερνάει τη στάθμη K4, τότε αυτός ο παλμός απορρίπτεται από μέρος της λέξης.
- Αν ο συνολικός χρόνος ενός παλμού ενέργειας είναι μικρότερος από 75 msec, τότε αυτός ο παλμός ενέργειας απορρίπτεται επίσης.

Ένα γραφικό παράδειγμα για δυο διαφορετικά σήματα, που δείχνει και την ακρίβεια των υλοποιημένων αλγόριθμων, δίνεται στα σχήματα 5-5 και 5-6.



Εικόνα 5-5: ανίχνευση ορίων λέξης "δεκαπέντε"



Εικόνα 5-6: Ανίχνευση ορίων λέξης "άσπρος"

Ο αλγόριθμος του Rabiner, στη συνέχεια βρίσκει τις λέξεις που ειπώθηκαν συγκρίνοντας της με αποθηκεμένες λέξεις. Ο σκοπός αυτής της εργασίας όμως δεν είναι η αναγνώριση λέξεων αλλά η μεγαλύτερη ακρίβεια στην εξαγωγή του pitch όπως ειπώθηκε στην αρχή του κεφαλαίου. Οπότε η αναγνώριση λέξεων δεν υλοποιήθηκε.

6 Λογισμικό εξομοίωσης

Το λογισμικό γράφτηκε πρώτα σε προσωπικό υπολογιστή και αφού τελείωσε η ανάπτυξη, έγιναν οι απαραίτητες αλλαγές για να εισαχθεί στο ενσωματωμένο σύστημα. Αυτό έγινε επειδή στον Η/Υ είναι πιο εύκολη η διαδικασία της αποσφαλμάτωσης (debugging) καθώς και πιο γρήγορη η επαλήθευση των αποτελεσμάτων με προγράμματα όπως το excel. Το πρόγραμμα αναπτύχθηκε στη γλώσσα προγραμματισμού C, στον C compiler Microsoft Visual C++ 6.0.

6.1 Μπλοκ διάγραμμα λογισμικού

Το σχήμα 6-1 δίνει το μπλοκ διάγραμμα της ροής του λογισμικού. Το λογισμικό αποτελείτε από 5 μεγάλα τμήματα.

1. Αρχικοποίηση των μεταβλητών, που στο διάγραμμα ονομάζεται «System Init»
2. Εισαγωγή, από τον χρήστη, του ηχογραφημένου αρχείου, το οποίο για την ώρα επιτρέπεται να είναι μόνο μορφής wav, μονοφωνικό αρχείο, 8 ή 16 kHz, 8 bit ανάλυση και στο διάγραμμα ονομάζεται «File Input»
3. Αλγόριθμος κατάτμησης σήματος φωνής, που στο διάγραμμα ονομάζεται Endpoint Detection (το μπλοκ διάγραμμα του αλγορίθμου ανίχνευσης παλμών δε δόθηκε επειδή δεν υπάρχει. Ο αλγόριθμος υλοποιήθηκε σύμφωνα με τις οδηγίες του κεφαλαίου 5.4 και για λόγους πληρότητας δίνεται ο κώδικας στο παράρτημα ΙΙ).
4. Εξαγωγή pitch, που στο διάγραμμα ονομάζεται Pitch Detection
5. Αποθήκευση αποτελεσμάτων σε αρχεία, που στο διάγραμμα ονομάζεται Save to files

6.1.1 Αρχικοποίηση των μεταβλητών

Όπως σε κάθε C πρόγραμμα, εδώ λαμβάνει χώρο η δήλωση των μεταβλητών, τα αρχεία από τα οποία εξαρτώνται άλλα αρχεία (file dependencies), ορίζονται οι πίνακες και οι συναρτήσεις.

6.1.2 Εισαγωγή ηχογραφημένου αρχείου

Ο χρήστης εισάγει ένα ηχογραφημένο αρχείο, το αρχείο πρέπει να έχει ηχογραφηθεί προηγουμένως και η ηχογράφηση πρέπει να έχει συχνότητα δειγματοληψίας 8 ή 16 kHz και ανάλυση 8 bit. Υπάρχει η δυνατότητα να σωθεί σε μορφή .wav και .raw όπως με κάποιο πρόγραμμα σαν το wavelab. Αν το πρόγραμμα δε βρει στο αρχείο τις προαναφερθέντες παραμέτρους τότε θα ξαναζητήσει από τον χρήστη αρχείο.

6.1.3 Αλγόριθμος κατάτμησης σήματος φωνής

Ο αλγόριθμος που αναλύθηκε στο κεφάλαιο «αλγόριθμος κατάτμησης σημάτων φωνής σε ομιλία και θόρυβο», του Rabiner. Τα αποτελέσματα και όλες οι παράμετροι που μπορεί να χρειαστούν σώζονται στο τέλος σε ένα αρχείο.

6.1.4 Εξαγωγή pitch

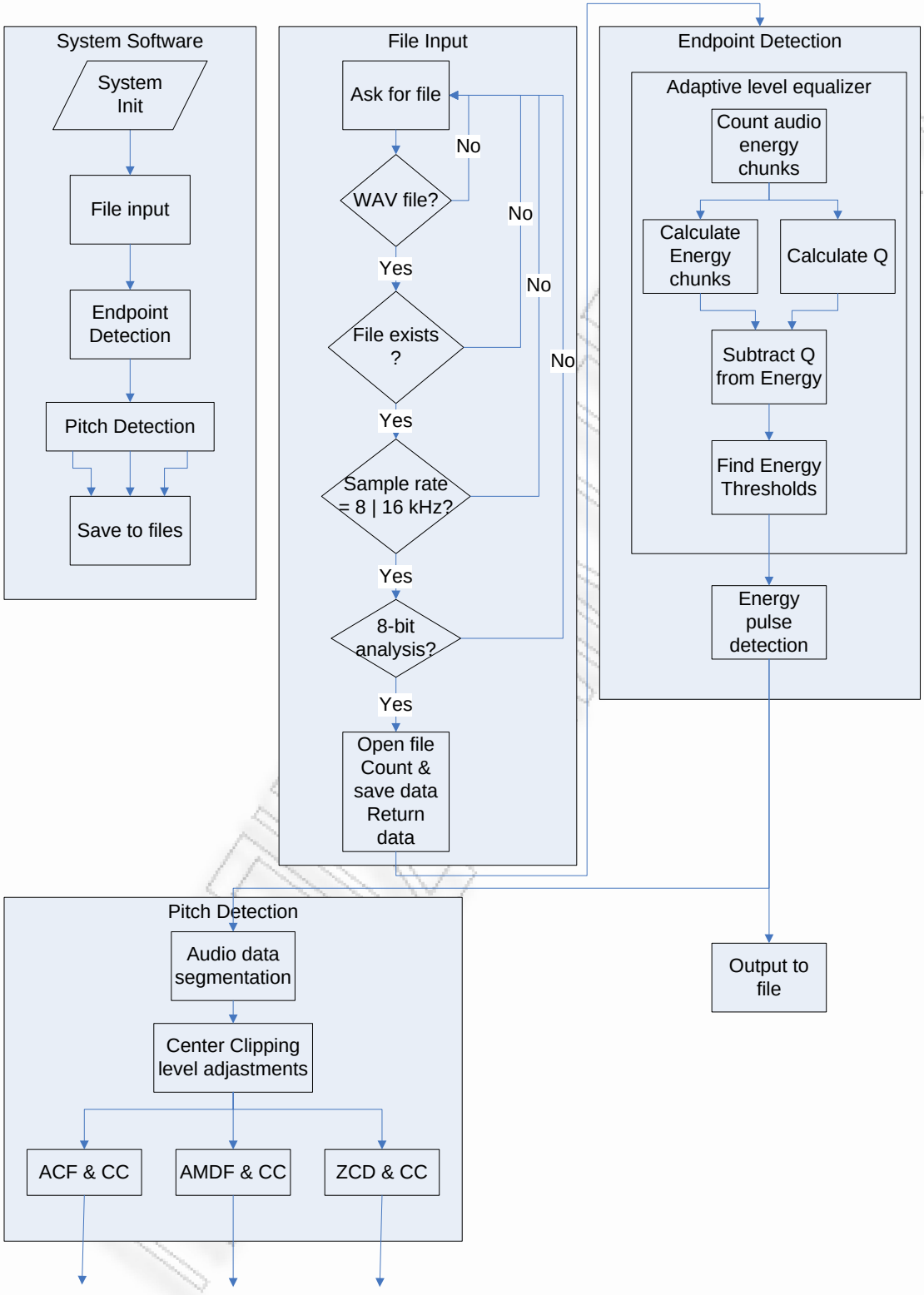
Το πρόγραμμα αφαιρεί από το αρχείο ήχου τις χρονικές διάρκειες που δεν υπάρχει ομιλία και το δίνει στις συναρτήσεις εξαγωγής pitch, που αναλύθηκαν στο κεφάλαιο «αλγόριθμοι εξαγωγής pitch».

6.1.5 Αποθήκευση αποτελεσμάτων σε αρχεία

Όλα τα αποτελέσματα με τις σχετικές παραμέτρους σώζονται σε αρχεία.

6.2 Συμπεράσματα-Αποτελέσματα

Τα αποτελέσματα που πάρθηκαν από τις εκτελέσεις του προγράμματος συγκρίθηκαν με τα αποτελέσματα που έδωσε το πρόγραμμα Praat πάνω στα ίδια αρχεία. Οι διάφορες αποκλίσεις βρέθηκαν ικανοποιητικές. Τα σχήματα που δόθηκαν στο κεφάλαιο ανάλυσης των αλγορίθμων ήταν από το πρόγραμμα αυτό.



Εικόνα 6-1: Μπλοκ διάγραμμα λογισμικού Η/Υ

7 Υλικό (hardware)

Στο κεφάλαιο αυτό εξετάζεται το υλικό που επιλέχθηκε για την υλοποίηση των αλγορίθμων.

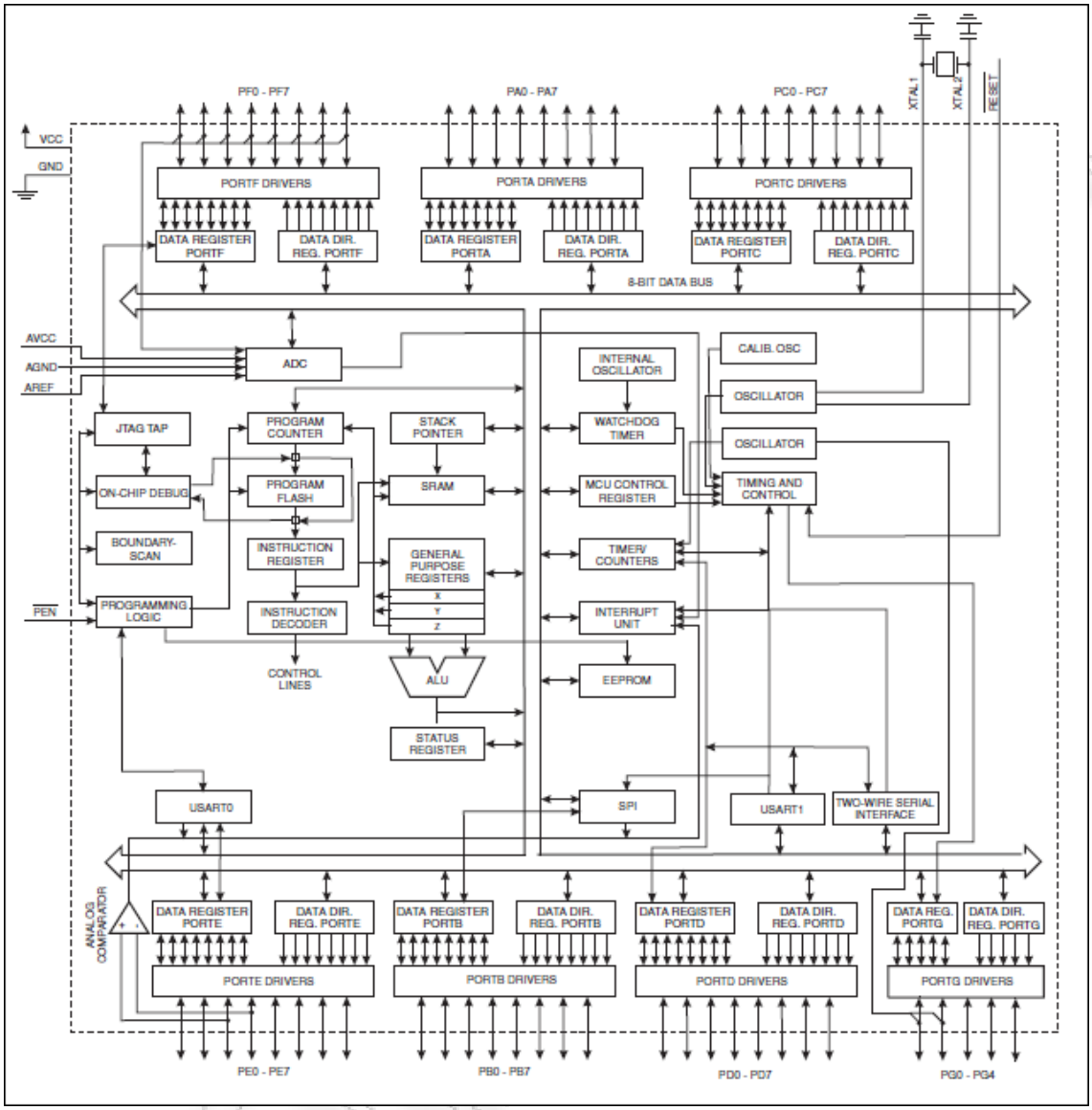
7.1 Παράμετροι για ενσωματωμένο σύστημα

Οι βασικότερες παράμετροι για την επιλογή του συστήματος ήταν ως συνήθως το κόστος και η δύναμη επεξεργασίας, με ιδιαίτερη έμφαση στο χαμηλό κόστος. Οι αλγόριθμοι που επιλέχθηκαν να υλοποιηθούν είναι οι λιγότερο χρονοβόροι υπολογιστικά (εκτός από τον ZCD που δεν υλοποιήθηκε λόγω άσχημων αποτελεσμάτων) και δεν απαιτούν μεγάλες ποσότητες μνήμης RAM ή μνήμης κώδικα. Με βάση αυτά, η αγορά ενός ενσωματωμένου συστήματος με έναν 8 – bit μικροελεγκτή φάνηκε η σωστή επιλογή. Επειδή οι αλγόριθμοι ACF και AMDF είναι της τάξης $O(N^2)$, ο μικροελεγκτής δε θα καταφέρει να λειτουργήσει σε πραγματικό χρόνο, αλλά θα δώσει αρκετά ικανοποιητικά αποτελέσματα για υπάρχουσες εφαρμογές.

Το ενσωματωμένο σύστημα βρίσκεται πάνω σε μια αναπτυξιακή πλατφόρμα με έναν 8 – bit μικροελεγκτή AVR, της εταιρείας Atmel. Στο κεφάλαιο αυτό αναλύονται τα βασικά στοιχεία του μικροελεγκτή καθώς και το αναπτυξιακό σύστημα.

7.2 8 – bit Μικροελεγκτής AVR

Οι μικροελεγκτές AVR της εταιρείας Atmel είναι 8 – bit μικροελεγκτές με χρονισμούς έως και 20 MHz και είναι αρχιτεκτονικής Υπολογιστικής μηχανής μειωμένου συνόλου εντολών (RISC – Reduced Instruction Set Computer). Σε αντίθεση με τις υπολογιστικές μηχανές πολύπλοκου συνόλου εντολών (CISC - Complex Instruction Set Computer), έχουν λιγότερες εντολές αλλά μπορούν να τις εκτελέσουν γρηγορότερα. Η εκτέλεση πιο πολύπλοκων εντολών γίνεται με το λογισμικό συνδυάζοντας εντολές στη σειρά. Όταν το λογισμικό έχει να κάνει με πολύπλοκες μαθηματικές πράξεις, ένας CISC είναι ίσως η καλύτερη επιλογή, σε άλλες περιπτώσεις, όπως αυτή, ένας RISC κρίθηκε καταλληλότερος. Το μπλοκ διάγραμμα ενός AVR δίνεται στο σχήμα 7.1.



Εικόνα 7-1: Μπλοκ διάγραμμα μικροελεγκτή AVR

7.2.1 Εξωτερικός κρύσταλλος χρονισμού

Οι AVR έχουν δυνατότητα χρονισμού και από εξωτερικό κρύσταλλο χρονισμού αλλά και από εσωτερικό ρολόι.

7.2.2 Διακοπές (Interrupts)

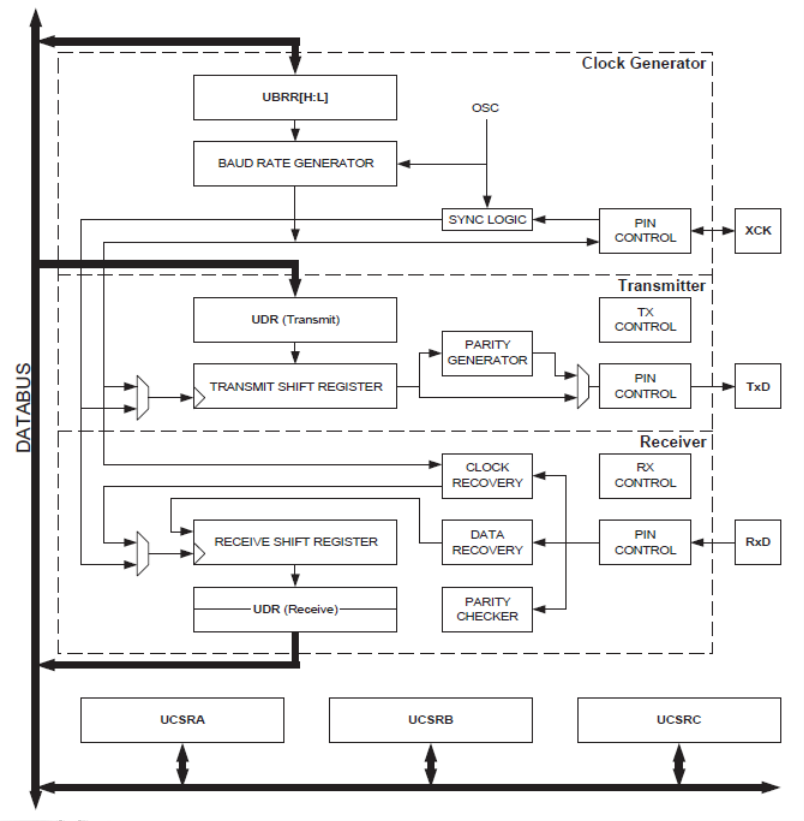
Ο AVR έχει 35 πηγές interrupt, σε αυτές συμπεριλαμβάνονται οι εξωτερικές διακοπές και οι διακοπές της σειριακής θύρας, που αναλύεται παρακάτω.

7.2.3 Σειριακό πρωτόκολλο UART

Η σειρά των AVR έχει κάποιους μικροελεγκτές που έχουν περιφερειακά UART. Ένα σετ καταχωρητών καθώς και 3 διακοπές υπάρχουν για να εξυπηρετούν τη σειριακή θύρα. Τα χαρακτηρηστικά της UART ενός AVR είναι τα εξής:

- Αμφίδρομη λειτουργία (Ανεξάρτητοι καταχωρητές μετάδοσης και λήψης χαρακτήρων)
- Σύγχρονη ή ασύγχρονη λειτουργία
- Γεννήτρια ρυθμού σειριακής θύρας (Baud Rate) υψηλής ανάλυσης
- Υποστηρίζει 5, 6, 7, 8 ή 9 bits δεδομένων, 1 ή 2 stop bits.
- Περιττή ή άρτια ισοτιμία
- Ανίχνευση λάθους από δεδομένα ή από πακέτο
- 3 διαφορετικές διακοπές

Το βασικό μπλοκ διάγραμμα του περιφερειακού της σειριακής θύρας δίνεται στο σχήμα 7-1



Εικόνα 7-2: Σειριακή θύρα AVR

Οι διακοπές της σειριακής θύρας είναι πολύ χρήσιμες, αφού η UART είναι αρκετά αργό πρωτόκολλο. Σε μια επικοινωνία 115200 bits/second, με 8 bit δεδομένων, 1 stop bit και 1 start bit, δηλαδή σύνολο 10

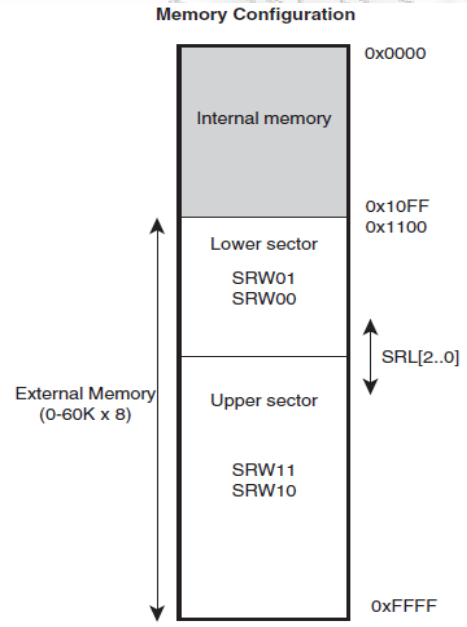
bit μετάδοσης ανά byte, και συνεχή ροή δεδομένων, το κάθε byte έχει λήψη κάθε 86 μs περίπου (1/11520), το οποίο αφήνει σημαντικά μεγάλο περιθώριο για επεξεργασία μέσα στον επεξεργαστή.

7.2.4 Διασύνδεση με εξωτερική μνήμη (SRAM)

Μια από τις δυνατότητες που παρέχουν οι AVR, είναι η διασύνδεση με μια εξωτερική μνήμη SRAM. Τα κύρια χαρακτηρηστικά αυτής της λειτουργίας είναι:

- 4 διαφορετικά στάδια αναμονής (wait states)
- Δυνατότητα να χρησιμοποιηθούν διαφορετικά στάδια αναμονής για διαφορετικές σελίδες εξωτερικής μνήμης
- Δυνατότητα επιλογής του αριθμού των γραμμών διευθυνσιοδότησης (address bus) για μικρότερες μνήμες

Το μοντέλο μνήμης, φαίνεται στο σχήμα 7-2



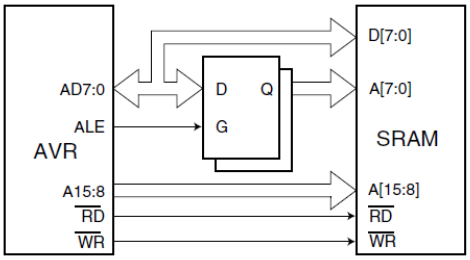
Εικόνα 7-3: Μοντέλο μνήμης AVR

Για να υπάρχει πρόσβαση σε αυτή την εξωτερική μνήμη, χρησιμοποιούνται κάποιες ακίδες (pins) του μικροελεγκτή. Αυτές οι ακίδες είναι οι εξής:

- AD7:0, πολυπλεγμένος δίαυλος δεδομένων και οι λιγότερο 8 σημαντικές γραμμές του διαύλου διευθύνσεων

- A15:8, οι 8 περισσότερο σημαντικές γραμμές του διαύλου διευθύνσεων. Ο αριθμός των γραμμών είναι μεταβαλλόμενος σε περίπτωση που θέλουμε να χρησιμοποιήσουμε μικρότερη μνήμη από 64 kBytes και θέλουμε να χρησιμοποιήσουμε κάποιες από τις ακίδες για άλλες λειτουργίες
- ALE, μανδαλωτής για τον δίαυλο διευθύνσεων
- /RD, ακίδα λειτουργίας διαβάσματος από τη μνήμη
- /WR, ακίδα λειτουργίας εγγραφής στη μνήμη

Ο τρόπος συνδεσμολογίας της εξωτερικής μνήμης περιλαμβάνει και ένα εξωτερικό μανδαλωτή (latch), όπως φαίνεται και στο σχήμα 7-3



Εικόνα 7-4: Σύνδεση AVR με εξωτερική μνήμη

Για να επικοινωνήσει ο μικροελεγκτής με την εξωτερική SRAM, πρώτα δίνονται όλες οι γραμμές διευθυνσιοδότησης. Έπειτα, ο μανδαλωτής μανταλώνει τις 8 λιγότερο σημαντικές γραμμές διευθυνσιοδότησης και τις δίνει στις εξόδους του. Έπειτα, ο δίαυλος δεδομένων δίνεται από τις ίδιες ακίδες που δόθηκαν οι 8 λιγότερο σημαντικές γραμμές, όμως επειδή ο μανδαλωτής έχει κρατήσει τις τιμές των γραμμών διευθυνσιοδότησης, πλέον η μνήμη έχει ότι χρειάζεται για να δώσει δεδομένα, οπότε χρησιμοποιώντας το /RD ή το /WR, γίνεται η επιθυμητή λειτουργία.

Όπως φαίνεται από το προηγούμενο σχήμα 7.2, ο μικροελεγκτής, αντιμετωπίζει την εξωτερική μνήμη SRAM, σαν συνέχεια της εσωτερικής του μνήμης RAM. Αυτό όμως σημαίνει πως τα πρώτα 0x1100 bytes της εξωτερικής μνήμης, θα είναι αχρησιμοποίητα, δε θα διαβαστούν ποτέ.

Φαίνεται λοιπόν πως από μια εξωτερική μνήμη SRAM των 64 kBytes, θα πάνε χαμένα 4 kBytes και μερικά ακόμα bytes. Αν χρησιμοποιηθεί μικρότερη εξωτερική μνήμη SRAM, για παράδειγμα, 32 kBytes, και συνδεσμολογηθεί όπως τη μνήμη των 64 kBytes, χωρίς τη γραμμή διευθυνσιοδότησης A15, τότε θα υπάρχει πάλι το ίδιο πρόβλημα, χαμένα 4 kBytes και από αυτή τη μνήμη.

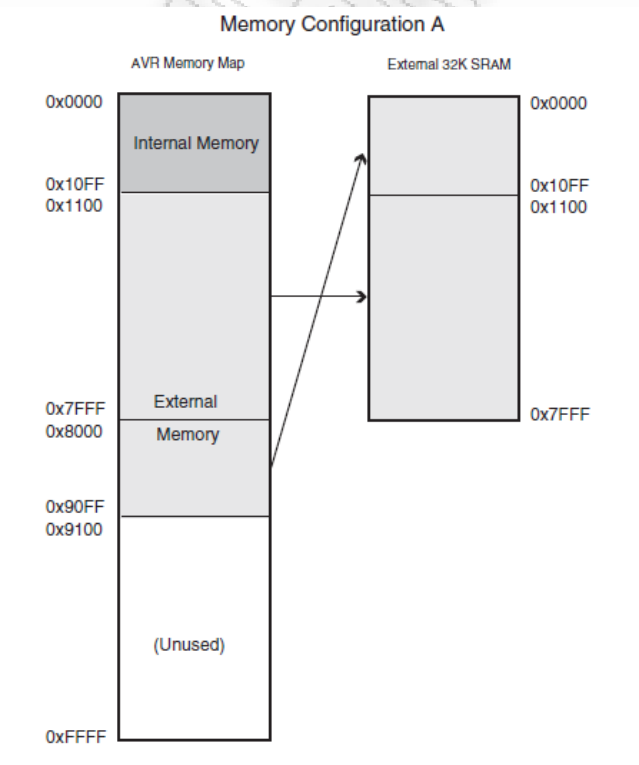
Παρ' όλα αυτά, υπάρχουν λύσεις και για τις δυο περιπτώσεις, δηλαδή να μη χαθούν κομμάτια μνήμης και από μια μνήμη των 64 kBytes αλλά και από μικρότερη εξωτερική μνήμη SRAM.

Στη συγκεκριμένη πτυχιακή, αγοράστηκε ένα έτοιμο αναπτυξιακό εργαλείο, το οποίο χρησιμοποιεί μια εξωτερική μνήμη των 32 kBytes, συνδεσμολογημένη όπως αναφέρθηκε, δηλαδή με τον ίδιο τρόπο όπως μια εξωτερική μνήμη των 64 kBytes, χωρίς τη γραμμή διευθυνσιοδότησης A15 (πιο συγκεκριμένα, στην αναπτυξιακή πλατφόρμα, αυτή η γραμμή πηγαίνει στην ακίδα /CE της εξωτερικής μνήμης, αλλά αυτό θα εξηγηθεί πιο αναλυτικά στο κεφάλαιο που αναλύεται η εξωτερική μνήμη της αναπτυξιακής πλατφόρμας στο κεφάλαιο 7.3.2).

Η λύση λοιπόν που χρησιμοποιείται όταν χρειάζεται να υπάρχει πρόσβαση στα πρώτα 4 kBytes της εξωτερικής μνήμης SRAM, είναι η εξής:

- θα πρέπει αρχικά, ο μικροελεγκτής να είναι προγραμματισμένος για πρόσβαση σε εξωτερική μνήμη SRAM των 64 kBytes, έτσι ώστε να επιτρέπεται από τον assembler και από τον compiler, να δοθεί πρόσβαση σε εξωτερική μνήμη παραπάνω από τα 32 kBytes, αυτό πρακτικά σημαίνει πως ο μικροελεγκτής θα νομίσει πως πρέπει να χρησιμοποιήσει και την γραμμής διευθυνσιοδότησης A15.
- Διευθυνσιοδοτούνται διευθύνσεις μεγαλύτερες από την φυσική χωρητικότητα της εξωτερικής μνήμης SRAM. Οι φυσικές διευθύνσεις είναι από 0x0000 ως 0x8000, και θα χρησιμοποιηθούν οι επιπλέον διευθύνσεις από 0x8000 ως 0x90FF. Με αυτό τον τρόπο, ο μικροελεγκτής διευθυνσιοδοτεί την μνήμη με τις πρώτες 14 γραμμές διευθυνσιοδότησης συν την γραμμή A15. Ενώ η γραμμή A15 δε κάνει τίποτα, οι πρώτες 14 γραμμές διευθυνσιοδοτούν τα πρώτα 4 kBytes της εξωτερικής μνήμης SRAM, έτσι, ο μικροελεγκτής νομίζει πως μιλάει με μια εξωτερική μνήμη μεγαλύτερη των 32 kBytes ενώ στην πραγματικότητα μιλάει με τα πρώτα «αόρατα» 4 kBytes.

Το σχήμα 7-4 δείχνει αυτό τον τρόπο διευθυνσιοδότησης.



Εικόνα 7-5: Μοντέλο μνήμης με διασύνδεση εξωτερικής μνήμης

Παρακάτω δίνεται ένα τμήμα κώδικα στη γλώσσα προγραμματισμού C, που έχει πρόσβαση σε όλη την εξωτερική μνήμη SRAM. Πρώτα γίνεται η προσπέλαση διευθύνσεων από τη διεύθυνση 0x1100 ως το τέλος της φυσικής εξωτερικής μνήμης SRAM, 0x8000, και οι υπόλοιπες προσπελάσεις, από τη διεύθυνση 0x8000 ως τη διεύθυνση 0x90FF, γίνονται στα πρώτα 4352 «χαμένα» bytes.

```
//pointer that points at the first external SRAM address.
unsigned char *external_sram_ptr = (unsigned char *)0x1100;

//When external_sram_ptr reaches address 0x7FFF,
//the next address, 0x8000 will point to the first
//byte of the external SRAM. The next 0x10FE bytes
//after  that, will cover the whole first 4 kBytes
//(plus some other bytes) of the lost memory.
for(i = 0; i < 0x8000; i ++)
{
        *external_sram_ptr = i;
        external_sram_ptr ++;
}
```

Πρέπει να σημειωθεί πως για να λειτουργήσει αυτός ο τρόπος πρόσβασης στην εξωτερική RAM σε πιο πολύπλοκους επεξεργαστές, πρέπει να υπάρχει δήλωση χρησιμοποίησης αυτής της μνήμης και των ορίων της στον linker. Σε αντίθετη περίπτωση θα υπάρξει λάθος από τον χρόνο της μεταγλώτισης. Στον μικροελεγκτή που χρησιμοποιήθηκε σε αυτή την εργασία, όπως θα φανεί και αργότερα, αυτή η μνήμη δε δηλώθηκε, επίσης δεν υπήρχε έλεγχος.

7.2.5 Προγραμματισμός εντός συστήματος (In System Programming και JTAG)

Μερικοί AVR έχουν την δυνατότητα να προγραμματιστούν μέσα από JTAG. Επίσης, όλοι οι AVR, έχουν τη δυνατότητα να προγραμματιστούν πάνω στην πλατφόρμα που θα λειτουργήσουν, αυτή η δυνατότητα λέγεται προγραμματισμός εντός του συστήματος (In System Programming – ISP). Σε αυτή την πτυχιακή έγινε χρήση του προγραμματισμού εντός του συστήματος.

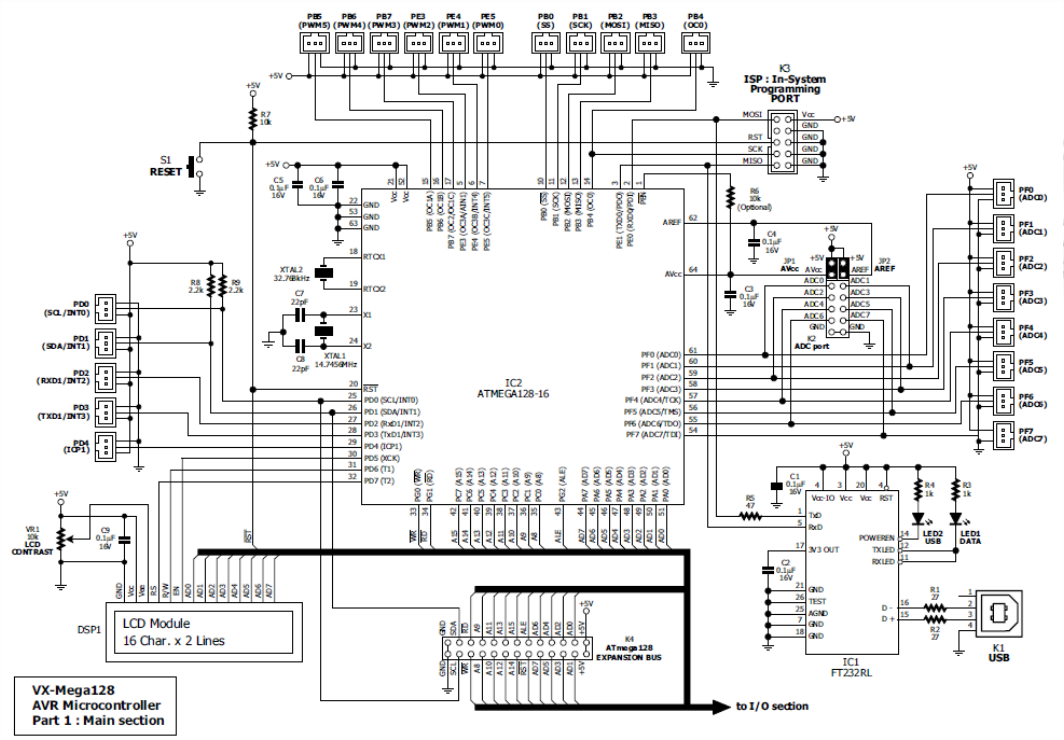
Με αυτή τη λειτουργία, η εσωτερική μνήμη προγράμματος flash του AVR, προγραμματίζεται με το πρωτόκολλο επικοινωνίας SPI. Τα εργαλεία που παρέχει η Atmel για τον προγραμματισμό των AVR, παρέχουν δυνατότητες επαλήθευσης της σωστής εγγραφής του κώδικα (verification).

7.3 Αναπτυξιακή πλατφόρμα (development board)

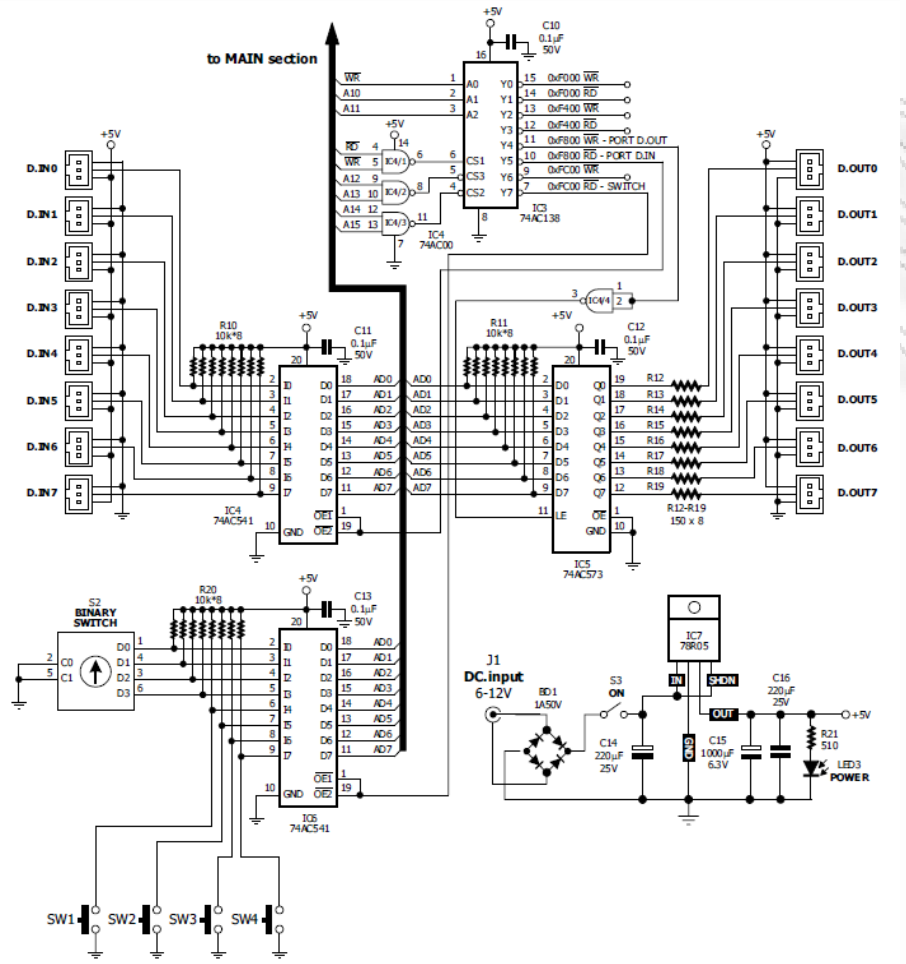
Σε αυτό το κεφάλαιο αναλύονται τα χαρακτηριστικά της αναπτυξιακής πλατφόρμας που χρησιμοποιήθηκαν σε αυτή την εργασία. Συνοπτικά, αυτή η αναπτυξιακή πλατφόρμα περιλαμβάνει

- ATmega128, με μνήμη προγράμματος flash 128 kbytes και εσωτερική μνήμη RAM 4 kBytes
- Εξωτερικός κρύσταλλος στα 14745600 Hz.
- 34 είσοδοι/έξοδοι
- Οθόνη υγρών κρυστάλλων 16 στήλες επί 2 γραμμές
- 4 κουμπιά
- Έναν περιστρεφόμενο διακόπτη
- USB θύρα με το ολοκληρωμένο FT232RL της FTDI
- 32 kBytes εξωτερική μνήμη SRAM, σε ξεχωριστή μικρή πλακέτα
- Υποστηρίζει προγραμματισμό πάνω στην πλατφόρμα (ISP - In System Programming)
- Τάση λειτουργίας από 6 ως 12 Volt, με γέφυρα και ρυθμιστή τάσης (voltage regulator) 500 mA

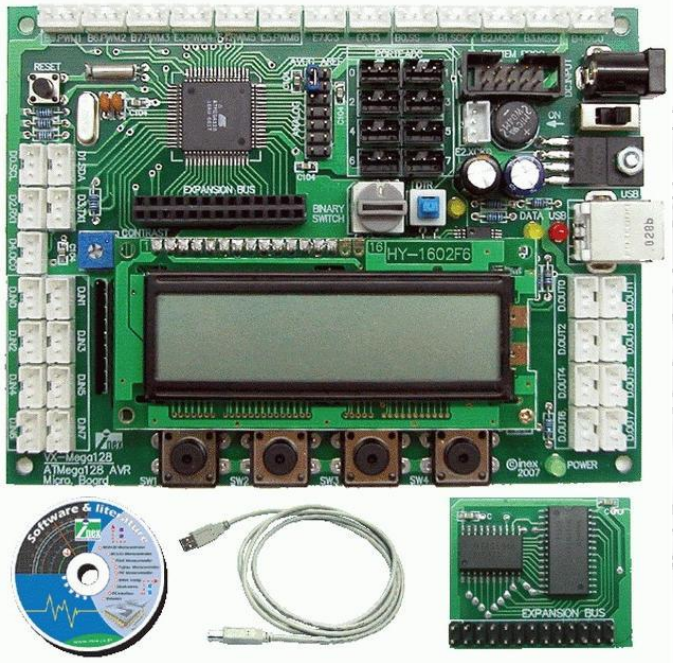
Στα σχήματα 7-5 και 7-6 φαίνονται τα σχηματικά και στο σχήμα 7-8 μια φωτογραφία του αναπτυξιακού.



Εικόνα 7-6: Σχηματικό της αναπτυξιακής πλατφόρμας - μικροελεγκτής



Εικόνα 7-7: Σχηματικό της αναπτυξιακής πλατφόρμας - Latches

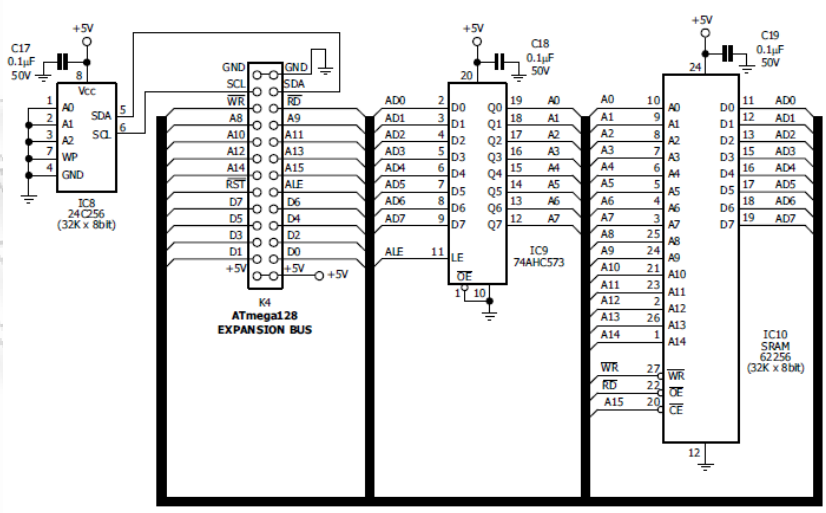


Εικόνα 7-8: Το αναπτυξιακό σύστημα

7.3.1 Εξωτερική μνήμη 32 kBytes

Όπως εξηγήθηκε στο προηγούμενο κεφάλαιο, οι μικροελεγκτές AVR έχουν τη δυνατότητα να υποστηρίξουν εξωτερική μνήμη SRAM με μια συγκεκριμένη συνδεσμολογία (σχήμα 7-8). Αυτή η αναπτυξιακή πλατφόρμα κάνει χρήση αυτής της δυνατότητας και προσφέρει μια εξωτερική πλακέτα με μια SRAM των 32 kBytes και τον απαραίτητο μανδαλωτή (latch).

Επίσης, πάνω σε αυτή την πλακέτα με την μνήμη SRAM και τον μανδαλωτή, βρίσκεται και μια εξωτερική EEPROM που επικοινωνεί με τον μικροελεγκτή μέσω του πρωτοκόλλου I^2C. Αυτή η μνήμη είναι αδιάφορη για αυτή την εργασία, και όπως θα φανεί αργότερα, απενεργοποιήθηκε.



Εικόνα 7-9: Σχηματικό της αναπτυξιακής πλατφόρμας - εξωτερικές μνήμες

7.3.2 Τροποποιήσεις στην αναπτυξιακή πλατφόρμα

Όπως εξηγήθηκε προηγουμένως, χρησιμοποιούνται και τα 32 kBytes μιας εξωτερικής μνήμης SRAM. Σε αυτή την αναπτυξιακή πλακέτα όμως, η γραμμή διευθύνσεων A15 συνδέεται με την ακίδα /OE της εξωτερικής μνήμης SRAM. Η γραμμή διευθυνσιοδότησης A15 έχει αναλάβει τον ρόλο του σήματος Enable της εξωτερικής μνήμης SRAM.

Αυτό σημαίνει πως σε περίπτωση προσπάθειας προσπέλασης της εξωτερικής μνήμης SRAM των πρώτων 4 kBytes, η γραμμή διευθυνσιοδότησης θα γίνει 1, οπότε η εξωτερική μνήμη SRAM θα απενεργοποιηθεί

Για αυτό τον λόγο, κρίθηκε απαραίτητη μια τροποποίηση πάνω στην αναπτυξιακή πλακέτα. Η ακίδα της γραμμής διευθυνσιοδότησης A15, αφαιρέθηκε από την πλακέτα της μνήμης, και σαν enable ακίδα, χρησιμοποιήθηκε η ακίδα επικοινωνίας του μικροελεγκτή με την εξωτερική μνήμη EEPROM που βρίσκεται πάνω στην ίδια πλακέτα με την SRAM. Με αυτό τον τρόπο, χρησιμοποιείτε όλη η εξωτερική μνήμη SRAM και η εξωτερική μνήμη EEPROM είναι απενεργοποιημένη. Το μειονέκτημα είναι πως κάθε φορά που πρέπει να προσπελασθεί η εξωτερική μνήμη SRAM, πρέπει η ακίδα enable, να γίνεται 0.

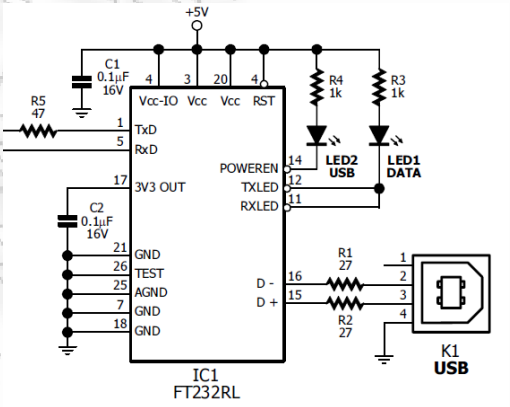
7.3.3 Θύρα USB με το ολοκληρωμένο κύκλωμα FT232RL

Η επικοινωνία του προσωπικού υπολογιστή με την αναπτυξιακή πλατφόρμα χωρίζεται σε δυο μέρη. Στο πρωτόκολλο USB και στο πρωτόκολλο UART.

Επάνω στην αναπτυξιακή πλακέτα βρίσκεται το ολοκληρωμένο κύκλωμα FT232RL της εταιρείας FTDI, το οποίο είναι γέφυρα μεταξύ των πρωτοκόλλων USB και UART.

Η όλη μετατροπή των πρωτοκόλλων γίνεται τελείως διάφανα για τον μικροελεγκτή, οπότε δε χρειάζεται κανένας επιπλέον οδηγός (drivers). Ο μικροελεγκτής επικοινωνεί με τη σειριακή του θύρα με το FT232RL και το FT232RL επικοινωνεί με USB με τον υπολογιστή.

Από την πλευρά του υπολογιστή χρειάζεται η εγκατάσταση ενός προγράμματος που να δημιουργεί μια εικονική (virtual) σειριακή θύρα. Με την εγκατάσταση αυτού του προγράμματος, πλέον ο χρήστης θα δουλεύει με ένα σειριακό τερματικό που θα συνδέεται στην δημιουργημένη virtual σειριακή θύρα. Το σχηματικό κύκλωμα για τη συνδεσμολογία του FT232RL φαίνεται στο σχήμα 7-9.



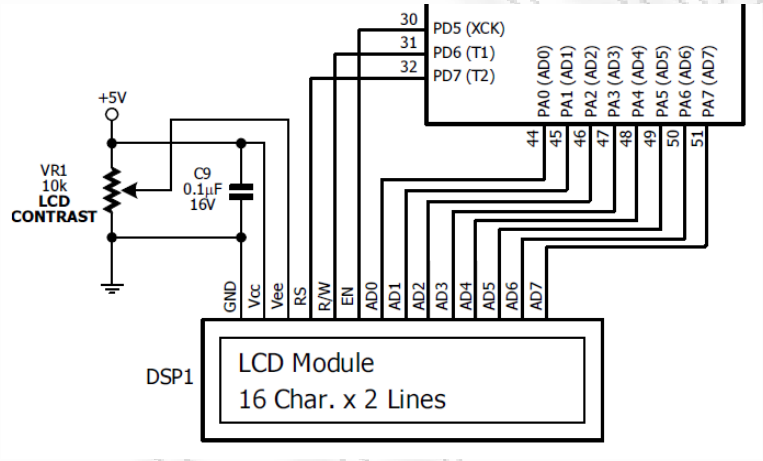
Εικόνα 7-10: Ολοκληρωμένο κύκλωμα IC FTDI και σχηματικό

7.3.4 Εξωτερικός κρύσταλλος 14.745.600 Hz

Ο εξωτερικός κρύσταλλος που χρονίζει τον μικροελεγκτή είναι 14.745.600 Hz για να δίνει 0% λάθος στην σειριακή επικοινωνία της UART με 115200 bits/second.

7.3.5 Οθόνη υγρών κρυστάλλων (LCD)

Η οθόνη υγρών κρυστάλλων που βρίσκεται πάνω στην αναπτυξιακή πλατφόρμα είναι 16 στήλες και 2 γραμμές. Ο ελεγκτής που χρησιμοποιεί είναι ο γνωστός HD44780. Η σύνδεση έχει γίνει με τέτοιο τρόπο ώστε να μπορεί να γίνει πρόσβαση στο LCD σαν να είναι μνήμη (memory mapped). Το σχηματικό διασύνδεσης με τον μικροελεγκτή φαίνεται στο σχήμα 7-10.



Εικόνα 7-11: Σχηματικό κύκλωμα διασύνδεσης με LCD

8 Ενσωματωμένη υλοποίηση λογισμικού

Στην περίπτωση μεταφοράς λογισμικού από μια πλατφόρμα σε μια άλλη, λαμβάνονται υπ' όψη οι διαφορές του υλικό και τα εργαλεία λογισμικού στα οποία έγινε η ανάπτυξη.

Για την ανάπτυξη του λογισμικού για τον μικροελεγτκή, χρησιμοποιήθηκε το AVR studio με χρήση του plug in C compiler AVR GCC. Το AVR studio παρέχει στον προγραμματιστή πολύ καλές δυνατότητες προσομοίωσης (simulation) του κώδικα.

Η επιλογή της γλώσσας C δεν είναι τυχαία. Τα πλεονεκτήματα μιας υψηλής γλώσσας προγραμματισμού (high level programming language), είναι πως οι αλλαγές που λαμβάνουν χώρα, δε θα έχουν να κάνουν με τη σύνταξη της γλώσσας και τη δομή του προγράμματος. Ο προγραμματιστής έχει ένα πρόβλημα λιγότερο να τον απασχολήσει. Καίτοι τα δυο βασικά στοιχεία, οι υλοποιήσεις των αλγορίθμων ACF και AMDF υλοποιήθηκαν σε assembly για καλύτερη ταχύτητα.

Το πρόγραμμα που υλοποιήθηκε δε τρέχει σε πραγματικό χρόνο, αλλά πρώτα περιμένει να δεχτεί τα δεδομένα ήχου από τη σειριακή θύρα. Όπως θα εξηγηθεί στο τελευταίο κεφάλαιο, μπορεί να γίνει μια αντιστοίχηση της σειριακής θύρας με έναν μετατροπέα από αναλογικό σε ψηφιακό σήμα στην ίδια περίπου ταχύτητα.

8.1 Μπλοκ διάγραμμα λογισμικού

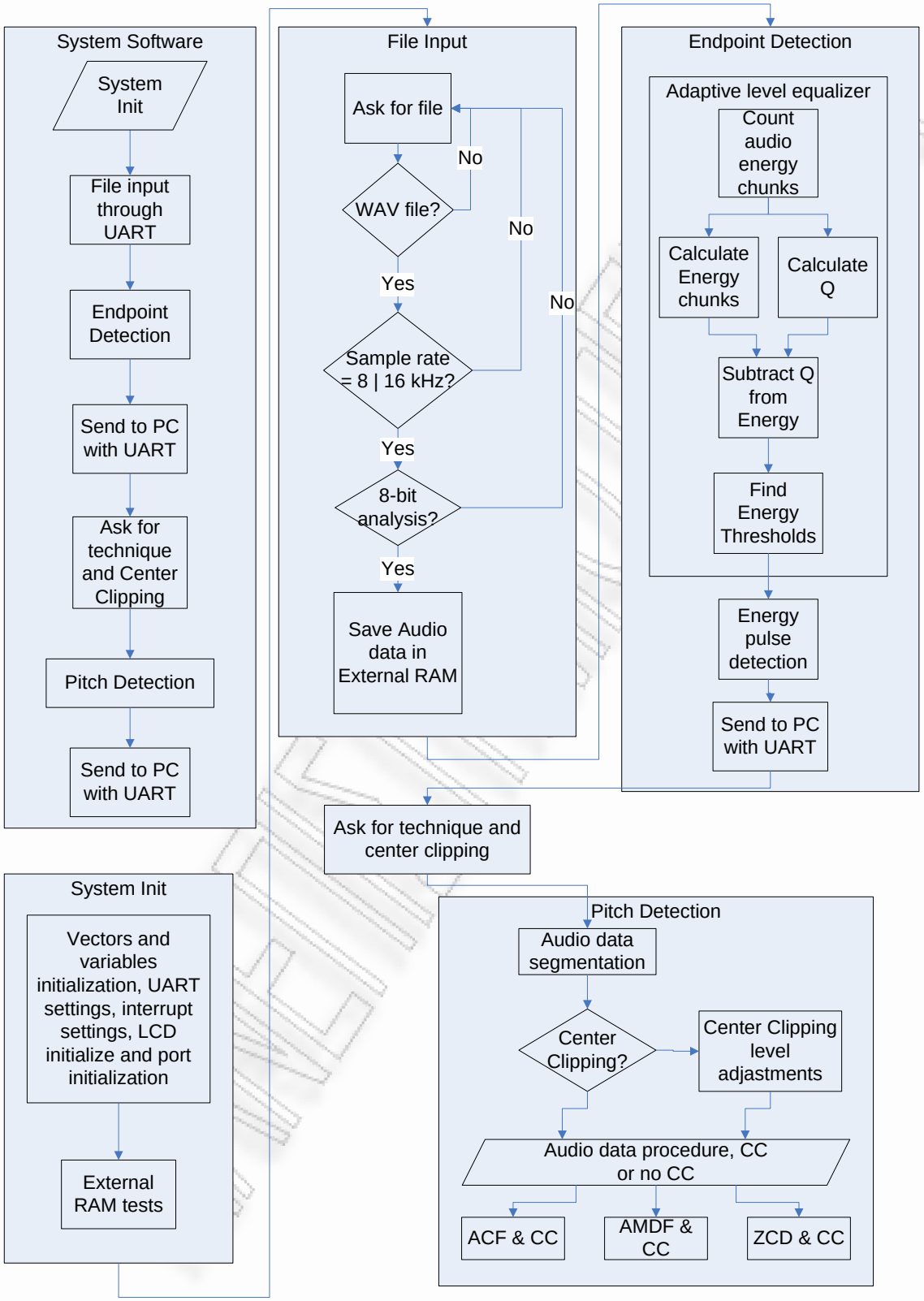
Στο σχήμα 8-1 δίνεται το μπλοκ διάγραμμα του λογισμικού του ενσωματωμένου συστήματος του και μετά δίνεται μια εξήγηση και τονίζονται οι διαφορές του με το λογισμικό της εξομοίωσης στον Η/Υ. Όπως και στην εξομοίωση, το πρόγραμμα χωρίζεται σε τμήματα. Τα τμήματα αυτά είναι

1. Αρχικοποίηση των μεταβλητών, «System Init»
2. Αποστολή αρχείου ήχου μέσω της σειριακής θύρας, «File Input». Μόνο wav αρχείο.
3. Αλγόριθμος κατάτμησης σήματος φωνής, «Endpoint Detection», ο αλγόριθμος ανίχνευσης παλμών είναι ο ίδιος με το μπλοκ διάγραμμα του σχήματος 6.1 που υπάρχει και στο παράρτημα II
4. Αποστολή αποτελεσμάτων της εκτέλεσης του προηγούμενου αλγορίθμου, «Send to PC with UART»
5. Ερώτηση προς τον χρήστη για τις παραμέτρους του Pitch Detection, «Ask for technique and Center Clipping»
6. Εξαγωγή pitch, «Pitch Detection»
7. Αποστολή αποτελεσμάτων της εκτέλεσης του προηγούμενου αλγορίθμου «Send to PC with UART»

8.1.1 Αρχικοποίηση των μεταβλητών

Εδώ γίνονται οι αρχικοποιήσεις των πινάκων, των μεταβλητών, η αρχικοποίηση της σειριακή θύρας στα «115200, N, 8, 1», η αρχικοποίηση των διακοπών, η αρχικοποίηση του LCD και η αρχικοποίηση των εξωτερικών πορτών του μικροελεγκτή. Επίσης, γίνεται το τεστ ελέγχου της εξωτερικής μνήμης RAM.

Τα προγράμματα που ελέγχουν την ορθή λειτουργία μιας μνήμης, συνήθως εκτελούν πολλές λειτουργίες, ανιχνεύοντας λάθη στο δίαυλο δεδομένων, στο δίαυλο διευθύνσεων, στην εσωτερική διευθυνσιοδότηση της μνήμης και στην εσωτερική οργάνωση των δεδομένων της (bitwise tests). Για τη συγκεκριμένη πτυχιακή χρησιμοποιήθηκαν μόνο τα απλά τεστ που βρίσκουν αν υπάρχει πρόβλημα, χωρίς διευκρίνιση για το είδος του προβλήματος. Πιο συγκεκριμένα, ξεκινώντας από την αρχική διεύθυνση μνήμης, γράφεται μια τιμή και σε κάθε αυξανόμενη διεύθυνση γράφεται μια αύξηση της προηγούμενης τιμής κατά 1 (συνήθως ξεκινάει από το μηδέν και αυξάνει). Μόλις γραφτεί όλη η μνήμη, το πρόγραμμα ξεκινάει να διαβάζει από την αρχή και αν βρει ανακρίβεια σταματάει και ενημερώνει τον χρήστη.



Εικόνα 8-1: Μπλοκ διάγραμμα λογισμικού ενσωματωμένου συστήματος

8.1.2 Αποστολή αρχείου ήχου μέσω της σειριακής θύρας

Στη συνέχεια το σύστημα ζητάει μέσω της σειριακής θύρας από τον χρήστη να στείλει το αρχείο, πάλι μέσω της σειριακής θύρας. Εδώ τονίζεται, πως, όπως αναφέρθηκε και στο κεφάλαιο «υλικό», ο Η/Υ επικοινωνεί με το σύστημα μέσω της θύρας USB, όμως και τα δυο νομίζουν πως μιλάνε με σειριακή θύρα, αυτό επειδή το λειτουργικό σύστημα έχει «γεφυρώσει» από τη μεριά του Η/Υ το τερματικό πρόγραμμα της σειριακής θύρας με τη USB θύρα και το FTDI IC έχει κάνει το ίδιο επάνω στο ενσωματωμένο σύστημα. Παραπάνω πληροφορίες για τις ενέργειες του χρήστη, δίνονται στο παράρτημα Ι αυτής της εργασίας.

Ο χρήστης επιλέγει το αρχείο, που πρέπει να πληρεί τα ίδια χαρακτηριστικά όπως και για το πρόγραμμα εξομοίωσης και το στέλνει χωρίς επιπλέον κανόνες επικοινωνίας όπως για παράδειγμα 1 K Xmodem, Kermit, Xmodem, Ymodem κλπ.

Το σύστημα δέχεται το αρχείο και με το που καταφτάνουν τα πρώτα δεδομένα από το header file, εξετάζει τις παραμέτρους (αρχείο wav, αριθμός bit ανάλυσης και ρυθμός δειγματοληψίας). Αν είναι, τότε αρχίζει και τοποθετεί τα δεδομένα στην εξωτερική RAM.

8.1.3 Αλγόριθμος κατάτμησης σήματος φωνής

Ο αλγόριθμος εκτελείτε κανονικά, όπως και στην εξομοίωση.

8.1.4 Αποστολή αποτελεσμάτων

Με τον ίδιο τρόπο αποστέλλονται τα αποτελέσματα από το ενσωματωμένο σύστημα προς τον Η/Υ, πάλι με τη UART.

8.1.5 Ερώτηση για Center Clipping

Για να μην επιβαρυνθεί το σύστημα από όλες τις επεξεργασίες, ρωτάει τον χρήστη ποιον αλγόριθμο να εκτελέσει και αν θέλει να εισαχθεί η τεχνική center clipping. Η ανταλλαγή της πληροφορίας γίνετια μέσω της UART.

8.1.6 Εξαγωγή pitch

Ο αλγόριθμος εκτελείτε κανονικά, όπως και στην εξομοίωση.

8.1.7 Αποστολή αποτελεσμάτων

Όμοια, τα αποτελέσματα αποστέλλονται τα αρχεία από το ενσωματωμένο σύστημα προς τον Η/Υ.

8.2 Διαφορές

Κατά τη μεταφορά του λογισμικού από τον Η/Υ στο ενσωματωμένο σύστημα (migration), μεγάλο μέρος της δομής του προγράμματος έμεινε ίδιο, ακριβώς όπως αναφέρθηκε στο κεφάλαιο «Λογισμικό». Οι

υλοποιήσεις των αλγορίθμων, επίσης δεν άλλαξαν. Υπάρχουν τμήματα κώδικα κρατήθηκαν ακριβώς τα ίδια. Οι αλλαγές που έγιναν, χωρίζονται σε τρεις κατηγορίες

- Βελτιστοποίηση χρήσης χώρου μνήμης κώδικα.
- Βελτιστοποίηση ταχύτητας
- Βελτιστοποίηση αξιοποίησης RAM
- Διαφορές λόγω υλικού – Προσθήκη οδηγών (drivers)

Παρακάτω αναλύονται αυτές οι διαφορές.

8.2.1 Βελτιστοποίηση χρήσης χώρου μνήμης κώδικα

Ο AVR που χρησιμοποιήθηκε έχει 128 kBytes μνήμης Flash για αποθήκευση κώδικα. Ο χώρος που χρησιμοποιήθηκε από αυτό είναι το 3%. Αυτό σημαίνει πως για τη συγκεκριμένη ανάπτυξη, ο χώρος μνήμης κώδικα δεν ήταν λόγος ανησυχίας.

8.2.2 Βελτιστοποίηση ταχύτητας

Η σημαντικότερη διαφορά του AVR GCC με το Microsoft Visual Studio C++ 6.0 προς τον χρήστη είναι η διαχείριση της μνήμης. Λόγω μικρής εσωτερικής μνήμης αλλά και μειωμένης συγκριτικά ταχύτητας, ο AVR GCC δε διαχειρίζεται τη δυναμική δέσμευση μνήμης αποτελεσματικά, με αποτέλεσμα να πρέπει να τροποποιηθούν ή να αλλάξουν μερικά τμήματα κώδικα που αναφέρονται στη μνήμη. Η σημαντικότερη αλλαγή είναι η χρήση της συνάρτησης malloc. Παρακάτω αναλύεται η λειτουργία της malloc στον AVR GCC και εξηγείται για πιο λόγο δε χρησιμοποιήθηκε. Η malloc ορίζεται ως εξής:

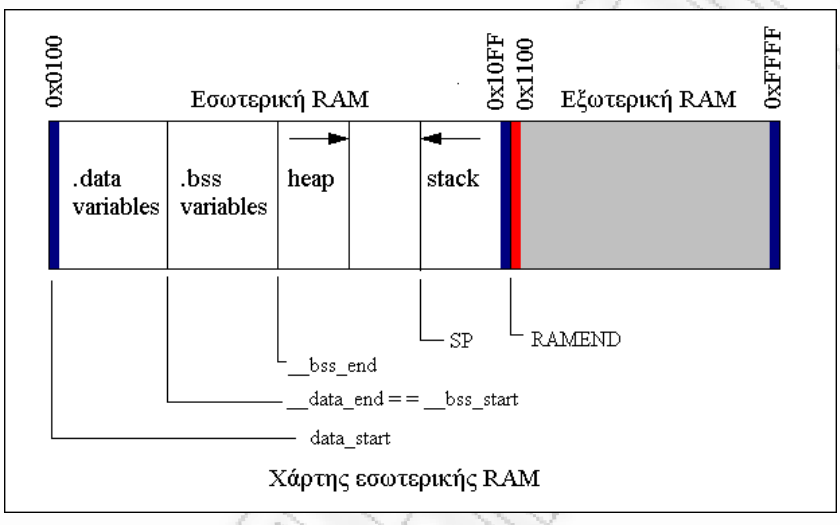
void *malloc(size_t size)

Παίρνει σαν όρισμα έναν αριθμό, δεσμεύει στη μνήμη χώρο ίσο με τον αριθμό του ορίσματος σε bytes και επιστρέφει έναν δείκτη σε void της πρώτης θέσης του χώρου που δέσμευσε. Αν επιστρέψει δείκτη στο μηδέν (0), τότε η δέσμευση χώρου δεν έγινε.

Η μνήμη RAM χωρίζεται στις αρχικοποιημένες και μη αρχικοποιημένες μεταβλητές, στον χώρο που θα αποκτηθεί από τη δυναμική δέσμευση μνήμης (heap) και από στη στοίβα (stack). Οι αρχικοποιημένες μεταβλητές τοποθετούνται σε ένα τμήμα (section) που ονομάζεται «**.**data» και οι μη αρχικοποιημένες σε ένα τμήμα που ονομάζεται «**.**bss». Στους AVR, αντίθετα με μεγαλύτερες αρχιτεκτονικές, δεν υπάρχει ενσωματωμένη διαχείριση μνήμης μέσα από υλικό, η οποία θα βοηθούσε στο να μην υπάρχει επικάλυψη μιας περιοχής της μνήμης με μια άλλη.

Ο τρόπος που διαμορφώνεται η μνήμη RAMστον συγκεκριμένο compiler και linker είναι ο εξής: το τμήμα **.**data τοποθετείτε στη αρχή της εσωτερικής RAM, ξεκινώντας από τις λιγότερο σημαντικές

διευθύνσεις και δεσμεύοντας χώρο προς τις περισσότερο σημαντικές διευθύνσεις. Ακολουθεί το **.**bss με την ίδια λογική. Η στοίβα ξεκινάει από το τέλος της εσωτερικής RAM, δηλαδή από τις περισσότερο σημαντικές διευθύνσεις και αυξάνεται προς τις λιγότερο σημαντικές διευθύνσεις. Η heap, καταλαμβάνει τον χώρο ανάμεσα στο .bss και στη stack, ξεκινώντας από το .bss και αυξάνοντας προς τις περισσότερο σημαντικές διευθύνσεις. Με αυτό τον τρόπο δεν υπάρχει κίνδυνος για υπερκάλυψη των μεταβλητών με τον χώρο της δυναμικής δέσμευσης μνήμης. Υπάρχει όμως κίνδυνος να υπάρξει υπερκάλυψη του με τη στοίβα. Στο σχήμα 8-2 δίνεται η εσωτερική μνήμη ενός AVR όπως τη διαχειρίζεται ο linker



Εικόνα 8-2: Διαμόρφωση μνήμης από τον linker

Σε μια απλή συσκευή όπως ένας AVR, είναι δύσκολο να γραφτεί μια υλοποίηση της malloc, που να διαχειρίζεται αποδοτικά τη δυναμική δέσμευση χώρου.

Όπως αναφέρθηκε στο κεφάλαιο «υλικό», στο ενσωματωμένο σύστημα υπάρχει εξωτερική μνήμη των 32 kBytes. Στην περίπτωση αυτή, τα προγράμματα απλοποιούνται πάρα πολύ με τη μεταφορά της heap στην εξωτερική μνήμη, παρ' όλ' αυτά, η χρησιμοποίηση της malloc είναι ακόμα επιβάρυνση αφού απαιτεί αρκετούς κύκλους ρολογιού για να εκτελεστεί.

Με αυτή τη λογική, αποφεύχθηκε η χρήση της malloc στον μικροελεγκτή. Οι πίνακες που χρησιμοποιήθηκαν έχουν συγκεκριμένο μήκος και ανταποκρίνονται στην συνολική χωρητικότητα της εξωτερικής μνήμης.

Η εξωτερική μνήμη δεν έχει δηλωθεί στο σύστημα και για πρόσβαση σε αυτή χρησιμοποιείτε δείκτης (pointer), όπως αναφέρθηκε στο κεφάλαιο «υλικό». Με αυτό τον τρόπο δε δεσμεύεται καθόλου μνήμη από την εξωτερική RAM και μπορεί να χρησιμοποιηθεί όλη για την αποθήκευση του αρχείου ήχου.

Επίσης, βελτιστοποίηση ταχύτητας προκύπτει κάνοντας χρήση και άλλων τεχνικών, όπως, τοποθέτηση πολλών βρόγχων επανάληψης, που ο καθένας πραγματοποιεί μόνο έναν έλεγχο, αντί για έναν μεγάλο βρόγχο που πραγματοποιεί πολλούς ελέγχους, loop unrolling ή διάφορες άλλες τεχνικές.

8.2.3 Βελτιστοποίηση αξιοποίησης RAM

Επειδή η μνήμη RAM είναι περιορισμένη, χρειάζεται ένας καλός τρόπος αξιοποίησής της. Πράγματι, στις πρώτες εκδόσεις της υλοποίησης του λογισμικού αυτού, υπήρχε πρόβλημα, οι μεταβλητές δε χωρούσαν στη RAM. Για αυτό έγιναν αρκετές αλλαγές, οι πιο πολλοί έχουν να κάνουν με πίνακες.

Πίνακες που ήταν ορισμένοι σαν «int», στο λογισμικό του Η.Υ, δε χρησιμοποιήθηκαν, η πληροφορία που ήταν γραμμένη εκεί, αποθηκεύτηκε πάνω σε άλλους πίνακες με στοιχεία κινητής υποδιαστολής (float) χρησιμοποιώντας δείκτες με χρήση casting (διαφορετική πληροφορία τύπου δεδομένων στην ίδια διεύθυνση για το συγκεκριμένο θέμα), σβήνοντας την επεξεργασμένη και άχρηστη πλέον πληροφορία τους. Με αυτό τον τρόπο, σώθηκε αρκετή μνήμη.

Οι πιο πολλοί πίνακες που δημιουργήθηκαν στην εξομοίωση στον Η/Υ δημιουργήθηκαν με δυναμική δέσμευση χώρου, όπως εξηγήθηκε παραπάνω, αυτό δεν εφαρμόστηκε σε αυτή την περίπτωση. Οι πίνακες εδώ υπολογίστηκαν από την αρχή έτσι ώστε να πιάσουν τον μέγιστο δυνατό χώρο που μπορούν σύμφωνα με την αποθηκευτική μνήμη για τα δεδομένα του ήχου, αυτό σημαίνει πως στην περίπτωση μικρότερης διάρκειας ήχου, θα μείνουν κάποιες θέσεις ανεκμετάλλευτες. Με κάποιες τεχνικές λογισμικού, πολύς χώρος από αυτόν τον ανεκμετάλλευτο χώρο μπορεί να χρησιμοποιηθεί, αυτό θέλει πολύ προσοχή και είναι η πρώτη αιτία που ο προγραμματιστής θα υποψιαστεί για σχετικά σφάλματα (bugs) του κώδικα.

Στο πρόγραμμα διακρίνονται δυο περιπτώσεις δειγματοληψίας, η μία 8 kHz και η άλλη 16 kHz. Στη περίπτωση των 8 kBytes, η εξωτερική RAM θα φτάσει για 4 δευτερόλεπτα ήχου και στη περίπτωση των 16 kBytes για 2. Οι αντίστοιχοι πίνακες που βρίσκονται στην εσωτερική μνήμη πρέπει να κρατάνε πληροφορίες που διαχειρίζονται είτε 4 είτε για 2 δευτερόλεπτα ήχου.

Ο αλγόριθμος κατάτμησης σήματος φωνής προϋποθέτει τμήματα ενέργειας τα οποία αποτελούνται από συγκεκριμένη χρονική διάρκεια, ενώ δηλαδή για ένα τμήμα ενέργειας διάρκειας 10 msec, στα 8 kHz, χρειάζεται να προστεθούν 80 δειγμάτων ήχου, στα 16 kHz χρειάζονται 160 δείγματα ήχου. Αυτό δεν αυξάνει τον απαιτούμενο χώρο. Αντίθετα, οι αλγόριθμοι αυτοσυσχέτισης και AMDF θέλουν τον διπλάσιο χώρο για να λειτουργήσουν, όμως χρειάζονται δείγματα μόνο για 24 msec που δεν είναι σημαντική απαίτηση χώρου, οπότε εκεί δεν υπάρχει πρόβλημα.

8.2.4 Διαφορές λόγω υλικού – Προσθήκη οδηγών (drivers)

Οι διαφορές της υπολογιστικής πλατφόρμας με τον Η/Υ, σε σχέση με το λογισμικό που εκτελείτε συνοψίζεται στον πίνακα 8.

	Η/Υ	**Ενσωματωμένο σύστημα**	**Αλλαγή οδηγών**
Φυσικό μέσο εισαγωγής/εξαγωγής	Σκληρός δίσκος	Σειριακή θύρα	NAI

δεδομένων ήχου			
Μέθοδος εισαγωγής/εξαγωγής ήχου	Windows API/Συναρτήσεις C	Διακοπές (interrupts) σειριακής θύρας	NAI
Είσοδος επικοινωνίας από τον χρήστη	Πληκτρολόγιο Η/Υ	Κουμπιά στην πλακέτα (buttons)	ΟΧΙ
Έξοδος επικοινωνίας προς τον χρήστη	Οθόνη Η/Υ	LCD	NAI

Πίνακας 7: Προσθήκες οδηγών στο ενσωματωμένο σύστημα

Στην περίπτωση των κουμπιών που βρίσκονται πάνω στο ενσωματωμένο σύστημα, δε κρίθηκε απαραίτητη η προσθήκη καινούριων οδηγών αφού η επικοινωνία μπορεί να γίνει επιτυχώς μέσα από τη σειριακή θύρα. Όμως όταν το σύστημα γίνει αυτόνομο, θα πρέπει να γραφτούν.

Όπως φαίνεται, υλοποιήθηκαν οδηγοί για τη σειριακή θύρα, τις διακοπές και το LCD. Από αυτά, αξίζει να αναφερθεί μόνο πως η σειριακή θύρα έχει ρυθμιστεί να δουλεύει στα 115200, 8, N, 1, που σημαίνει 10 bits ανά byte, δηλαδή 11520 bytes ανά δευτερόλεπτο. Αν η υλοποίηση της σειριακής θύρας απαιτούσε και επιπλέον τρόπο συγχρονισμού, όπως για παράδειγμα κάποιο preamble για τη μεταφορά ενός byte δεδομένων, το οποίο χρειάζεται για μεγάλες μεταφορές δεδομένων σε σειριακά πρωτόκολλα, τότε η ταχύτητα θα έπεφτε κατά πολύ.

9 Απόδοση συστήματος – συμπεράσματα

9.1 Εξομοίωση πραγματικού χρόνου

Το σύστημα, δε λειτουργεί σε πραγματικό χρόνο. Παρακάτω δίνονται οι λόγοι.

9.1.1 Απόδοση αλγορίθμου κατάτμησης σήματος φωνής σε πραγματικό χρόνο

Ο αλγόριθμος κατάτμησης σήματος της φωνής μέσα στην ηχογράφηση έχει σταθερές ενεργειακές στάθμες (K1, K2, K3 και K4), οι οποίες απαιτούν να είναι γνωστή η μέγιστη τιμή της ενέργειας (κεφάλαιο 5.4). Αυτό σημαίνει πως μέχρι να πάρει το σύστημα και το τελευταίο δείγμα από τον ήχο, δε μπορεί να αρχίσει την επεξεργασία. Για αυτό το λόγο δίνεται και ηχογραφημένο αρχείο, με σταθερό μήκος, για να γνωρίζει το σύστημα πότε πρέπει να ξεκινήσει ο αλγόριθμος κατάτμησης. Η υλοποίηση αυτού του αλγορίθμου σε πραγματικό χρόνο, αναλύεται σε επόμενο υποκεφάλαιο.

9.1.2 Ταχύτητα σειριακής θύρας

Επίσης, η ταχύτητα της σειριακής θύρας περιορίζει το σύστημα από το να χαρακτηριστεί πραγματικού χρόνου. Στην περίπτωση χρησιμοποίησης ενός μετατροπέα αναλογικού σε ψηφιακό, η είσοδος των δεδομένων θα γίνονταν με τον ρυθμό δειγματοληψίας του μετατροπέα. Η καθυστέρηση που εισάγει αυτό είναι της τάξης των μsec, δηλαδή αμελητέος χρόνος. Στο συγκεκριμένο όμως σύστημα χρησιμοποιείται σειριακή θύρα για τη λήψη των δεδομένων. Οι ρυθμίσεις της σειριακή θύρας είναι 115200,N,8,1, δηλαδή στα 10 bit μεταδίδεται 1 byte, οπότε η ταχύτητα μετάδοσης είναι 11520 bytes/sec. Αυτή η ταχύτητα είναι λίγο μεγαλύτερη από τη συχνότητα δειγματοληψίας στα 8 kHz, οπότε τα δεδομένα έρχονται γρηγορότερα από ότι θα έρχονταν με τον μετατροπέα. Στην περίπτωση όμως επιλογής μετατροπέα συχνότητας δειγματοληψίας στα 16 kHz, υπάρχει καθυστέρηση των δεδομένων. Στην περίπτωση αυτή, αν διπλασιαστεί το bit rate τότε πάλι οι ταχύτητες είναι κοντινές μεταξύ τους.

9.1.3 Απόδοση αλγορίθμων εξαγωγής pitch

Οι αλγόριθμοι υπολογισμού pitch, είναι γραμμένοι είτε για εκτέλεση σε πραγματικό χρόνο είτε όχι, αφού τα δεδομένα εισόδου για επεξεργασία τοποθετούνται σε προσωρινή μνήμη (buffers), η οποία μπορεί να γεμίζει σε πραγματικό χρόνο. Για να κριθούν αυτοί οι αλγόριθμοι ικανοί για εκτέλεση σε πραγματικό χρόνο, πρέπει να εκτελούν τις συναρτήσεις ACF και AMDF σε λιγότερο χρονικό διάστημα από όσο χρειάζεται για να γεμίσει ο επόμενος buffer προς επεξεργασία. Δηλαδή η επεξεργασία ενός buffer πρέπει να γίνεται σε χρόνο μικρότερο από 26.6 msec. Οι μετρήσεις έδειξαν πως αυτό δεν συμβαίνει στα 8 kHz, άρα ούτε και στα 16 kHz. Ο πίνακας 9 δείχνει τα αποτελέσματα αυτά εμφανίζοντας τους χρόνους εκτέλεσης των αλγορίθμων για την επεξεργασία εξαγωγής μιας τιμής συχνότητας.

Απόκριση στα 8 kHz	Απόκριση στα 16 kHz

(msec) μέσος όρος		(msec) μέσος όρος	
ACF	AMDF	ACF	AMDF
0.39	0.51	9.21	10.44

Πίνακας 8: Χρονική απόδοση εκτέλεσης αλγορίθμων pitch

Είναι φανερό από τον πίνακα 9, πως οι αλγόριθμοι εκτελούνται πολύ αργά για να χαρακτηριστεί το σύστημα, σύστημα πραγματικού χρόνου. Όμως, στην περίπτωση των 8 kHz, ο χρόνος που χρειάζεται το σύστημα για να βρει το pitch σε μικρές λέξεις είναι αρκετός για τηλεφωνικές εφαρμογές, όπως αναλύεται και παρακάτω.

Στο σημείο αυτό, αναφέρεται πως σε ανάλυση παραπάνω από 8 bit, ο AVR δε θα μπορέσει να ανταπεξέλθει χρονικά, ένας πολλαπλασιασμός δυο αριθμών 16 bit, σε ένα μηχάνημα 8 bit, θέλει αρκετούς κύκλους ρολογιού για να απαντήσει.

9.2 Ακρίβεια

Η ακρίβεια φαίνεται στις γραφικές παραστάσεις που δόθηκαν στο κεφάλαιο «αλγόριθμοι εξαγωγής pitch» και υπόκεινται στους πίνακες 5.2 και 5.3 του κεφαλαίου 4.7.3. Όλες αυτές οι γραφικές πάρθηκαν από το σύστημα, μέσα από τη σειριακή θύρα, για βεβαιότητα, συγκρίθηκαν με τα αντίστοιχα αποτελέσματα του προγράμματος εξομοίωσης και τα αποτελέσματα συγκρίθηκαν με το πρόγραμμα Praat. Η ακρίβεια κρίνεται ικανοποιητική καθώς δεν υπάρχουν μεγάλες αποκλίσεις από το Praat, το οποίο είναι το πρόγραμμα αναφοράς.

9.3 Ενσωματωμένη εφαρμογή: Αυτόματο τηλεφωνικό σύστημα

Αυτή η εργασία είναι η αρχή για την πραγματοποίηση ενός μεγαλύτερου έργου, το οποίο αναλύεται παρακάτω. Ο τελικός σκοπός του συστήματος, είναι ένα μικρό αυτόματο τηλεφωνικό σύστημα που θα επικοινωνεί με τον χρήστη και θα καταλαβαίνει τι ζητάει αυτός, με σκοπό να τον εξυπηρετήσει αυτόματα.

Η επικοινωνία από την πλευρά του μικροελεγκτή θα γίνεται με έτοιμα ηχογραφημένα μηνύματα που θα καθοδηγούν τον χρήστη να πατήσει κουμπιά ή να μιλήσει ορισμένες μικρές λέξεις.

Η επικοινωνία, από τη μεριά του χρήστη θα γίνεται με τόνους που παράγονται από το πάτημα των πλήκτρων του τηλεφώνου καθώς και με ορισμένες λέξεις. Αυτό θα γίνεται με την προτροπή του τηλεφώνου. Οι λέξεις αυτές θα περιλαμβάνουν τα «ναι», «όχι» και τους δέκα πρώτους αριθμούς του δεκαδικού συστήματος για να μπορεί να τους εισάγει ο χρήστης μιλώντας.

9.3.1 Επεξεργασία

Πρώτα πρέπει να φτιαχτεί το σύστημα έτσι ώστε να ανταποκρίνεται στις απαιτήσεις ενός τέτοιου συστήματος, δηλαδή όταν ο χρήστης λέει μια λέξη, να έχει γίνει εξαγωγή του pitch πριν πει την επόμενη λέξη. Όπως αναφέρθηκε, δε θα υπάρχει επεξεργασία πραγματικού χρόνου αλλά το σύστημα θα μπορεί να κάνει τη δουλειά του. Πρέπει λοιπόν να γίνουν οι εξής εργασίες:

- Να ρυθμιστεί ο αλγόριθμος κατάτμησης σήματος φωνής έτσι ώστε να μπορεί να λειτουργήσει σε πραγματικό χρόνο. Αυτό προϋποθέτει αυτόματη ρύθμιση των ενεργειακών σταθμών K1, K2, K3 και K4.
- Για την εισαγωγή της τεχνικής center clipping, θα χρειαστεί κάποια παρόμοια ρύθμιση, αφού για την ανίχνευση της τιμής κατωφλίου, απαιτείται η γνώση της μέγιστης τιμής του σήματος.
- Για το θέμα κατωφλίου στις συναρτήσεις ACF και AMDF θα υπάρχει πάλι κάποια σχετική ρύθμιση. Σε αυτή τη περίπτωση ίσως να χρειαστεί να μπει κάποιο σύστημα με αυτόματη ρύθμιση κέρδους (Automatic Gain Control) για να παραμένει το κατώφλι στα ίδια επίπεδα.

9.3.2 Περιβάλλον αυτόματου τηλεφωνικού συστήματος

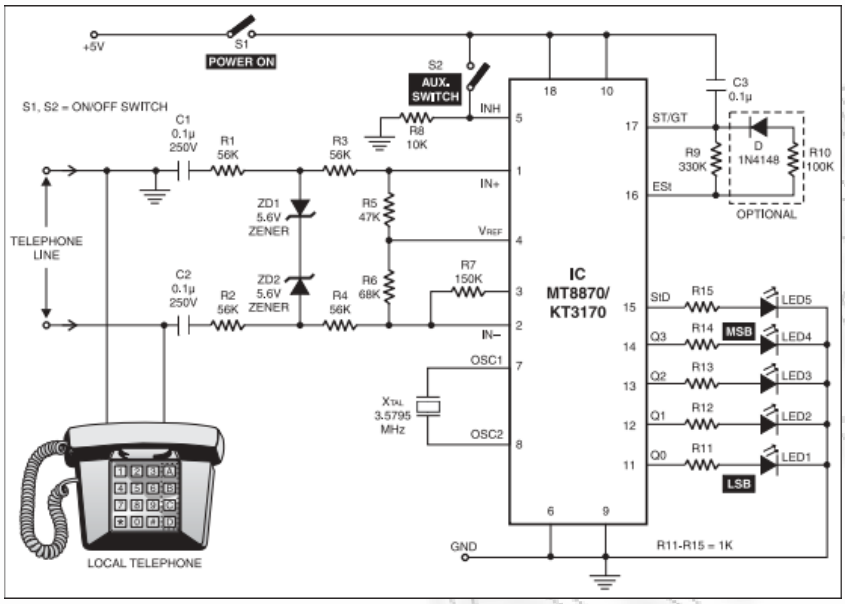
Πρέπει να ενσωματωθεί ένα μικρόφωνο και ένας μετατροπέας από αναλογικό σε ψηφιακό σήμα με την ικανότητα να δειγματοληπτεί στα 8 και 16 kHz και με ανάλυση 8 bit. Οι buffers θα πρέπει να διαβάζονται, να επεξεργάζονται και να σβήνονται γρήγορα για να μην υπάρξει πρόβλημα χώρου.

Θα πρέπει να φτιαχτεί το κατάλληλο κύκλωμα και οι οδηγοί για ένα τηλεφωνικό ολοκληρωμένο κύκλωμα, όπως το πολύ γνωστό ολοκληρωμένο κύκλωμα πληκτρολόγησης τόνων πολλαπλών συχνοτήτων (DTMF IC), MT8870. Το σχηματικό κύκλωμα δίνεται στο σχήμα 10-1.

9.3.3 Ολοκλήρωση λογισμικού

Στο τέλος πρέπει να γραφτεί το λογισμικό που θα οδηγεί τον χρήστη κάθε φορά σε ένα παρακάτω επίπεδο αναζήτησης με τις απαντήσεις του για να του δώσει την πληροφορία που χρειάζεται.

Ένα πολύ γνωστό τέτοιο σύστημα, είναι το σύστημα που έχει η UPS στα τηλεφωνικά της κέντρα, όταν κάποιος παραλήπτης ή αποστολέας ενός πακέτου έχει τον αριθμό εύρεσης (tracking number) ενός πακέτου, μπορεί, παίρνοντας τηλέφωνο τη UPS να πει τον αριθμό αυτό και να γίνει αυτόματη ενημέρωση.



Εικόνα 9-1: Σχηματικό κύκλωμα τηλεφωνικού IC MT8870

Αναφορές (References)

1. Analysis of Emotionally Salient Aspects of Fundamental Frequency for Emotion Detection Carlos Busso, Member, IEEE, Sungbok Lee, Member, IEEE, and Shrikanth Narayanan, Fellow, IEEE. IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. 17, NO. 4, MAY 2009
2. An algorithm for determining the endpoints of isolated utterances. L.R. Rabiner and M.R. Sambur. THE BELL SYSTEM TECHNICAL JOURNAL, FEBROUARY 1975
3. An improved endpoint detector for isolated word recognition. Lori F. Lamel, IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, VOL. ASSP-29, NO. 4, AUGUST 1981
4. The American Heritage Dictionary of the English Language, Fourth Edition, Houghton Mifflin Company, 2000, archived from the original on June 25, 2008, retrieved May 20, 2010
5. Durrant J D., Lovrinic J H. 1984. Bases of Hearing Sciences. Second Edition. United States of America: Williams & Wilkins
6. Gelfand, S A., 1990. Hearing: An introduction to psychological and physiological acoustics. 2nd edition. New York and Basel: Marcel Dekker, Inc.
7. Titze, I. R. (2008). The human instrument. Sci.Am. 298 (1):94-101. PM 18225701
8. Wolfgang Hess “Pitch Determination of Speech Signals”,1983 ISBN 3-540-11933-7 ISBN 0-387-11933-7
9. Titze, I.R. (1994). Principles of Voice Production, Prentice Hall (currently published by NCVS.org), ISBN 978-0137178933.
10. Machlis, Joseph; Forney, Kristine (2007). Payne, Maribeth. ed. The Enjoyment of Music (10 ed.). W. W. Norton. p. 8. ISBN 0393928888.
11. Plack, Christopher J.; Andrew J. Oxenham, Richard R. Fay, eds. (2005). Pitch: Neural Coding and Perception. Springer. ISBN 0387234721.
12. Lars Ahlzen, Clarence Song (2003). The Sound Blaster Live! Book. No Starch Press. ISBN 1886411735.
13. McGurk, H & MacDonald, J (1976); "Hearing lips and seeing voices," Nature, Vol 264(5588).
14. Wright, Daniel and Wareham, Gary (2005); "Mixing sound and vision: The interaction of auditory and visual information for earwitnesses of a crime scene," Legal and Criminological Psychology, Vol 10(1).
15. http://www.aruffo.com/eartraining/research/articles/pratt30.htm Carroll C. Pratt, Journal of Experimental Psychology, 13, 278-85, 1930
16. Olson, Harry F. (1967). Music, Physics and Engineering. Dover Publications. pp. 171, 248–251. ISBN 0486217698.

17. ster G (1973). "Auditory beats in the brain". Sci. Am. 229 (4): 94 102.doi: 10.1038/ scientificamerican1073-94. PMID 4727697.

18. Deutsch, D. (1974). "An illusion with musical scales". Journal of the Acoustical Society of America 56: s25. doi:10.1121/1.1914084 (exei abstract!!)

19. Deutsch, D. (1995). Musical Illusions and Paradoxes. Philomel Records. OCLC 36640949. ASIN B00000228A

20. Deutsch, D., Hamaoui, K., and Henthorn, T. (2007). "The Glissando Illusion and Handedness.". Neuropsychologia 45 (13): 2981–2988. doi:10.1016/j.neuropsychologia.2007.05.015. PMID 17624379 (exei weblink!!)

21. Warren RM, Wrightson JM, Puretz J (1988). "Illusory continuity of tonal and infratonal periodic sounds". Journal of the Acoustical Society of America 84 (4): 1338–42.doi:10.1121/1.396632. PMID 3198869

22. Roger N. Shepard (December 1964). "Circularity in Judgements of Relative Pitch". Journal of the Acoustical Society of America 36 (12): 2346–53. doi:10.1121/1.1919362.

23. John Clark, Colin Yallop and Janet Fletcher (2007). An Introduction to Phonetics and Phonology. Blackwell Publishing. ISBN 1405130830.

24. Christopher J. Plack (2005). Pitch: Neural Coding and Perception. Springer. ISBN 9780387234724.

25. Harry.F.Olson “Musical Engineering” © 1952 s.203

26. Ladislav O. Dolansky “An Instantaneous Pitch-Period Indicator” The Journal of the Acoustical Society ofAmerica, volume 27, nr 1, Jan 1955

27. Lawrence R Rabiner, “On the use of autocorrelation analysis for pitch detection”, IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, VOL.ASSP-25, NO. 1, FEBRUARY 1977.

28. Patricio de la Cuadra, Aaron Master, Craig Sapp. Efficient Pitch Detection Techniques for Interactive Music. [Center for Computer Research in Music and Acoustics, Stanford University].

Παράρτημα Ι

Πως να χρησιμοποιηθεί το ενσωματωμένο σύστημα

Η διαδικασία που ακολουθεί ο χρήστης είναι:

- Ηχογράφηση φωνής.
- Επεξεργασία αρχείου φωνής στον υπολογιστή (προαιρετικό).
- Αποστολή αρχείου φωνής στο μικρουπολογιστικό σύστημα μέσω σειριακής θύρας.

Ηχογράφηση φωνής

Χρησιμοποιώντας οποιοδήποτε λογισμικό ηχογράφησης φωνής για έναν προσωπικό υπολογιστή, ηχογραφούνται μεμονομένες λέξεις. Οι περιορισμοί του συστήματος είναι

- Αποθήκευση του αρχείου σε μορφή *.wav
- Ανάλυση 8 – bit
- Μονοφωνικό αρχείο
- Συχνότητα δειγματοληψίας
    - 8000 Hz (μέχρι τέσσερα δευτερόλεπτα ηχογράφησης)
    - 16000 Hz (μέχρι δυο δευτερόλεπτα ηχογράφησης)

Οι περιορισμοί αυτοί οφείλονται στην λιγοστή μνήμη SRAM που διαθέτει το μικρουπολογιστικό σύστημα.

Το πρότεινόμενο πρόγραμμα είναι το WaveLab. Με αυτό έγιναν οι ηχογραφήσεις των λέξεων που χρησιμοποιήθηκαν για τον έλεγχο της ορθής λειτουργίας του μικρουπολογιστικού συστήματος.

Επεξεργασία αρχείου φωνής στον υπολογιστή (προαιρετικό)

Το Wavelab παρέχει τον χρήστη την δυνατότητα να αφαιρέσει από το ηχογραφημένο αρχείο χρονικές περιόδους που δεν υπάρχει φωνή, αυτό φάνηκε χρήσιμο για την ηχογράφηση μεγάλων λέξεων, όταν η ηχογραφημένη λέξη ξεπερνούσε σε διάρκεια τα τέσσερα δευτερόλεπτα για συχνότητα δειγματοληψίας 8 kHz και τα δυο δευτερόλεπτα για συχνότητα δειγματοληψίας 16 kHz.

Επίσης, αν ο χρήστης επιθυμεί, μπορεί να γίνει κανονικοποίηση του σήματος. Με αυτή τη δυνατότητα, μια ηχογράφηση αποθηκεύονταν σε ένα αρχείο χωρίς να εφαρμόσουμε κανονικοποίηση, επίσης πραγματοποιούνταν μια ακόμα αποθήκευση με κανονικοποίηση και τα αποτελέσματα της εξόδου του μικρουπολογιστικού συστήματος, για κάθε αποθηκεμένο αρχείο, συγκρίνονταν μεταξύ τους. Η εκτέλεση της κανονικοποίησης δίνει κατώφλι που μπορεί να παραμείνει σταθερό όταν χρησιμοποιείτε ο ACF και αντίστοιχα όταν χρησιμοποιείτε ο AMDF.

Αποστολή αρχείου φωνής στο μικροϋπολογιστικό σύστημα μέσω θύρας USB.

Ο προσωπικός υπολογιστής που έχει το αποθηκευμένο αρχείο ήχου, πρέπει να περιέχει μια θύρα USB. Το αρχείο στέλνεται μέσω αυτής της θύρας στο μικροϋπολογιστικό σύστημα. Ο χρήστης ανοίγει το τερματικό σειριακής θύρας και χειρίζεται το αρχείο σαν να ήταν αποστολή μέσω σειριακής.

Παράρτημα ΙΙ

Παρακάτω δίνεται ο κώδικας για τον αλγόριθμο ανίχνευσης των παλμών ενέργειας που είναι μέρος του αλγορίθμου εξαγωγής σήματος φωνής.

```
K1 = 0;
K2 = 0.17 * max_smoothed_log10_energy;
K3 = 0.12733 * max_smoothed_log10_energy;
K4 = max_smoothed_log10_energy * log10(3);

word = 0;
thrK1 = 0;
thrK2 = 0;
StartWord = 0;
EndWord = 0;
number_of_words = 0;

for(i = 0; i < number_of_audio_chunks; i ++)
{
        if((equalized_energy_array[i] > K1) &&
        if((A[word][1] - A[word][0]) > TOO_LONG_RISING)
        {
        A[word][0] = A[word][1];
        }
        else{}
        if((A[word][3] - A[word][2]) < TOO_LONG_FALLING)
        {
        A[word][1] = A[word][3];
        }
        else
        {
        A[word][1] = A[word][2];
        }
        A[word][2] = A[word][3] = 0;        //Now, A[word][1] holds the
        //beginning of the energy pulse and A[word][1] the end.
```

```
(thrK1 == 0) && (StartWord == 0))
{
    thrK1 = 1;
    A[word][0] = i;
    EndWord = 0;
    continue;
}

if((equalized_energy_array[i] < K1) &&
(thrK1 == 1) && (StartWord == 0))
{
    thrK1 = 0;
    A[word][0] = 0;
    continue;
}

if((equalized_energy_array[i] > K2) &&
(StartWord == 0))
{
    A[word][1] = i;
    StartWord = 1;
    continue;
}

if((StartWord) &&
(equalized_energy_array[i] < K2) &&
(thrK2 == 0))
{
    thrK2 = 1;
    A[word][2] = i;
    continue;

if((A[word][1] - A[word][0]) > 5)
{
word ++;
}
else
{
A[word][0] = A[word][1] = 0;
}

continue;
}
}
//Final check, see if A[i][0] is not zero and A[i][1] is zero, then,
//zero A[i][0];
for(i = 0; i < number_of_words; i ++)
{
if((A[i][0] != 0) && (A[i][1] == 0))
{
A[i][0] = 0;
}
}
//Also, in case that the last starting point was detected without the
//ending.
for(;i < MAXIMUM_WORDS; i ++)
{
A[i][0] = A[i][1] = 0;
}
float peak_energy[MAXIMUM_WORDS];          //Problem if more
//than 10 energy pulses. But, I hope it won't happen.
for(i = 0; i < number_of_words; i ++)
{
peak_energy[i] = equalized_energy_array[A[i][0]];
```

```
}

if((StartWord) &&
(equalized_energy_array[i] > K2) &&
(thrK2 == 1))
{
    thrK2 = 0;
    A[word][2] = 0;
    EndWord = 0;
    continue;
}

if((StartWord) &&
(equalized_energy_array[i] < K3) &&
(EndWord == 0))
{
    A[word][3] = i;
    thrK1 = 0;
    thrK2 = 0;
    StartWord = 0;
    EndWord = 1;
    number_of_words ++;

for(j = A[i][0]; j < A[i][1]; j ++)
{
if(equalized_energy_array[j] > peak_energy[i])
{
peak_energy[i] = equalized_energy_array[j];
}
}
}
for(i = 0; i < number_of_words; i ++)
{
if(peak_energy[i] < K4)
{
A[i][0] = A[i][1] = 0;
}
}
printf("\n\nbegining and ending points\n\n");
for(i = 0; i < number_of_words; i ++)
{
if(A[i][0] != 0)
{
printf("%d\t%d\n", A[i][0], A[i][1]);}}
```