

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Εξαγωγή και επαναχρησιμοποίηση ρυθμικών και φασματικών χαρακτηριστικών  
από μουσικό περιεχόμενο (ηχογραφήσεις μπουζουκιού).



Του φοιτητή  
Σιούτα Στεφάνου  
Αρ. Μητρώου:587043

Επιβλέπων:  
Κωτσάκης Ρήγας  
Αναπληρωτής Καθηγητής

Ημερομηνία 29-05-2025

Τίτλος Δ.Ε.

Εξαγωγή και επαναχρησιμοποίηση ρυθμικών και φασματικών χαρακτηριστικών από μουσικό περιεχόμενο (ηχογραφήσεις μπουζουκιού)

Κωδικός Δ.Ε.: 24221

Όνοματεπώνυμο φοιτητή: Σιούτας Στέφανος

Όνοματεπώνυμο εισηγητή: Ρήγας Κωτσάκης

Ημερομηνία ανάληψης Δ.Ε.: 24-08-2024

Ημερομηνία περάτωσης Δ.Ε.: 29-05-2025

*Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.*

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Σιούτα Στεφάνου που την εκτόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

*«Στην γυναίκα μου Κορνήλια»*

## Πρόλογος

Βρισκόμαστε στα μέσα της δεκαετίας του 70, εγώ μαθητής της πρώτης δημοτικού, σε ένα μικρό ορεινό χωριό της Καλαμπάκας. Μαζί με το νεόφερτο δίκτυο της τότε ΔΕΗ στο πατρικό μας, ήρθε και ένα κασετόφωνο της TELEFUNKEN, συνοδευόμενο από κάποιες κασέτες με τραγούδια εποχής. Μια από αυτές είχε ορχηστρικά τραγούδια παιγμένα με μπουζούκι από τον αείμνηστο Γιώργο Ζαμπέτα. Αυτό ήταν. Αμέσως κατάλαβα ότι στην ζωή μου θα υπήρχε σίγουρα το μπουζούκι, όπως και έγινε. Αργότερα, προς το τέλος της δεκαετίας του 80, ως φοιτητής πια, έρχομαι σε γνωριμία με τους πρώτους ηλεκτρονικούς υπολογιστές. Γοητεύομαι αφάνταστα, επιλέγω για εργασία την τεχνολογία και έτσι ζω όλη την εξέλιξη τους ως σήμερα. Σήμερα στα μέσα της δεκαετίας του 20 ως τελειόφοιτος του τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΠΙΑΕ, επιλέγω για πτυχιακή εργασία ένα θέμα που συνδυάζει την αγάπη μου για το μπουζούκι και τη μουσική με την πληροφορική και τις γνώσεις μου, με την αμέριστη συμπαράσταση και καθοδήγηση του καθηγητή μου κ Κωτσάκη.

## Περίληψη

Η παρούσα διπλωματική εργασία έχει σαν αντικείμενο μελέτης την επεξεργασία στο πεδίο του χρόνου και της συχνότητας μελωδίας που έχει ηχογραφηθεί από μουζούκι έτσι ώστε να εξαχθεί κάποια μορφή βασικής παρτιτούρας της μελωδίας. Για το σκοπό αυτό συντάχθηκε κώδικας σε γλώσσα Matlab, ο οποίος βασίζεται σε βασικές συναρτήσεις της Matlab και κατάλληλη χρήση της ειδικής βιβλιοθήκης επεξεργασίας ήχου Mirtoolbox.

Η διαδικασία που χρησιμοποιήθηκε περιλαμβάνει αρχικά την εγγραφή μουσικού αποσπάσματος που παίζεται από μουζούκι σε αρχείο wav. Με τη χρήση των συναρτήσεων της προαναφερόμενης βιβλιοθήκης, συναρτήσεων της Matlab και κατάλληλους αλγορίθμους, πραγματοποιείται επεξεργασία αρχικά στο πεδίο του χρόνου και κατόπιν στο πεδίο της συχνότητας. Με πολλές προσπάθειες και εναλλακτικές προσεγγίσεις κατέστη δυνατή η εξαγωγή της κύριας συχνότητας του ήχου του μουζουκιού κατά τη διάρκεια συγκεκριμένης χρονικής περιόδου, η οποία είχε πριν κατάλληλα εκτιμηθεί με βάση τα χτυπήματα της πέννας του μουζουκιού. Με αυτές τις πληροφορίες και την αντιστοίχιση με πίνακα συχνότητας-νότας εξάγονται οι νότες που παίχθηκαν και οι αξίες τους (διάρκεια τετάρτου, ογδού, δέκατου έκτου, κλπ.). Με τη χρήση αυτών των δεδομένων και κατάλληλη παραμετροποίηση ο κώδικας δημιουργεί επιπρόσθετα ένα αρχείο που ακολουθεί το πρότυπο musicxml, το οποίο είναι αναγνώσιμο από τα συνήθη προγράμματα επεξεργασίας παρτιτούρας, όπως το Muse Score.

Η παραπάνω διαδικασία εφαρμόστηκε σε αρκετές περιπτώσεις γνωστών μουσικών κομματιών και διαπιστώθηκε η ιδιαίτερα ικανοποιητική αξιοπιστία της καθώς η πρωτότυπη παρτιτούρα (από την οποία πραγματοποιήθηκε η εκτέλεση του μουζουκιού) και η εξαγόμενη παρτιτούρα, από το πρόγραμμα που δημιουργήθηκε ήταν, σχεδόν πανομοιότυπες, με πολύ μικρές και στις περισσότερες περιπτώσεις αναμενόμενες διαφορές σε συγκεκριμένες μόνο περιπτώσεις. Συμπερασματικά, η μεθοδολογία που ακολουθήθηκε ήταν επιτυχής, όπως αποδείχθηκε πειραματικά, σε πολύ ικανοποιητικό βαθμό.



# **Extraction and reuse of rhythmic and spectral features from musical content (bouzouki recordings)**

Stefanos Sioutas

## **Abstract**

The subject of this thesis is the temporal and spectral processing of a basic melody, recorder by a bouzouki, so that to extract a basic score of the recorded melody. Towards this end, a script was developed in Matlab, using basic Matlab functions and the special sound processing toolbox Mirtoolbox.

The procedure that was used includes recording of the music excerpt played by bouzouki in a wav file. Using the functions of the toolbox above-mentioned and appropriate algorithms that were developed, the wav file is processed first at the time domain and then in the frequency domain. After many different approaches and efforts, we managed to extract the basic frequency tone played by the bouzouki over a specific period which has been estimated based on the hits of the plectrum. With this information and via a table that maps the basic frequencies to bouzouki notes, the melody notes and their values (durations of quarters, eighths, sixteenths, etc.) are extracted. These data are appropriately parametrized within the code to create a musicxml file that is readable by most usual music score processing software, such as Muse Score.

This procedure was applied in several cases of well-known music parts showing particularly successful results, since the original score (from which the bouzouki melody was played) and the exported by the program score were almost identical with minor and in most cases anticipated differences in some cases. In conclusion, the methodology followed was successful, as experimentally proven, to a very satisfactory degree.

## Ευχαριστίες

Τώρα που βρίσκομαι στο τέλος αυτού του ωραίου ταξιδιού, θα ήθελα να ευχαριστήσω όλους όσους με οποιονδήποτε τρόπο με βοήθησαν να τελειώσω την εργασία αυτή.

Πρώτον και καλύτερο τον αναπληρωτή Καθηγητή μου, κ. Ρήγα Κωτσάκη. Με τις γνώσεις του και την καθοδήγηση του με βοήθησε ουσιαστικά στην εκπλήρωση του στόχου μου.

Αμέσως μετά θεωρώ ότι οι συμφοιτητές μου ήταν δίπλα μου οπότε χρειάστηκα βοήθεια για να ξεκολλήσω την σκέψη μου από κάποιο τέλμα.

Εδώ πρέπει να αναφέρω την πολύτιμη βοήθεια που έλαβα από τον Καθηγητή μου και πλέον καλό μου φίλο κ. Ιωσηφίδη Αθανάσιο, με τον οποίο μας συνέδεε η κοινή μας συμμετοχή σε κάποια ορχήστρα του ωδείου Καλαμαριάς. Φίλε μου Θανάση σε ευχαριστώ πολύ

Τέλος, στους φίλους μου και στην οικογένεια μου που μου συμπαραστάθηκαν ηθικά όταν το είχα ανάγκη.

# Περιεχόμενα

Πρόλογος.....	iv
Περίληψη.....	v
Abstract .....	vii
Ευχαριστίες .....	viii
Περιεχόμενα .....	ix
Κατάλογος Εικόνων .....	xiii
Κατάλογος Πινάκων.....	xiv
Συντομογραφίες.....	xv
Κεφάλαιο 1 <sup>ο</sup> : Εισαγωγή .....	1
1.1    Υπόβαθρο .....	1
1.1.1    Muse Score .....	1
1.1.2    Music 21 .....	2
1.1.3    Cecilia.....	3
1.1.4    Ardour .....	3
1.2    Σκοπός της πτυχιακής εργασίας .....	3
1.2.1    Φασματική ανάλυση του ήχου του μπουζουκιού .....	3
1.2.2    Αναγνώριση νότας.....	4
1.2.3    Εύρεση αξίας και διαστημάτων.....	4
1.2.4    Δημιουργία συναρτήσεων για εξαγωγή παρτιτούρας.....	4
1.3    MusicXML .....	4
1.5    Ανάπτυξη της πτυχιακής εργασίας.....	5
1.6    Επίλογος .....	5
Κεφάλαιο 2 <sup>ο</sup> : Το μουσικό όργανο μπουζούκι .....	7
2.1    Εισαγωγή.....	7
2.2    Περιγραφή.....	7
2.3    Ιστορική αναδρομή.....	8
2.4    Κατασκευή και τεχνικά χαρακτηριστικά.....	9
2.5    Μουσικές δυνατότητες του οργάνου .....	11
2.6    Επίλογος .....	13
Κεφάλαιο 3 <sup>ο</sup> : Περί μουσικής θεωρίας.....	15
3.1    Εισαγωγή.....	15

3.2	Τι είναι η μουσική .....	15
3.3	Το πεντάγραμμο .....	15
3.4	Οι νότες ( Φθόγγοι ) .....	16
3.5	Τα κλειδιά.....	16
3.6	Μουσική έκταση – Οκτάβα.....	16
3.7	Μουσικές αξίες.....	17
3.8	Μέτρο .....	19
3.9	Οπλισμός .....	19
3.10	Επίλογος.....	20
Κεφάλαιο 4 <sup>ο</sup> : Ο Ήχος στην Φυσική και στην Πληροφορική .....		21
4.1	Εισαγωγή.....	21
4.2	Τι είναι ήχος .....	21
4.3	Χαρακτηριστικά του ήχου.....	22
4.4	Η αντίληψη του ήχου από τον άνθρωπο.....	24
4.5	Ψηφιακός Ήχος .....	24
4.5.1	Δειγματοληψία .....	25
4.5.2	Κβάντιση .....	25
4.5.3	Μέγεθος αρχείων.....	26
4.5.4	Ρυθμός μετάδοσης.....	26
4.5.5	Συμπύεση .....	26
4.5.6	Αποθήκευση και μεταφορά .....	27
4.5.7	Συμπέρασμα .....	28
4.6	Επίλογος.....	28
Κεφάλαιο 5 <sup>ο</sup> : Η βιβλιοθήκη MIRtoolbox .....		29
5.1	Εισαγωγή.....	29
5.2	Τι είναι το MATLAB .....	29
5.3	MIR toolbox και βασικές συναρτήσεις .....	29
5.3.1	Συνάρτηση miraudio.....	30
5.3.2	Συνάρτηση mirframe.....	30
5.3.3	Συνάρτηση mirfilterbank.....	30
5.3.4	Συνάρτηση mirsegment.....	30
5.3.5	Συνάρτηση mirspectrum .....	30
5.3.6	Συνάρτηση mirenvelope .....	31
5.3.7	Συνάρτηση mirrms .....	31

5.3.8	Συνάρτηση <code>mironsets</code> .....	31
5.3.9	Συνάρτηση <code>mirpitch</code> .....	31
5.3.10	Συνάρτηση <code>get</code> .....	31
5.3.11	Συνάρτηση <code>mirmidi</code> .....	32
5.4	Συναρτήσεις Matlab .....	32
5.4.1	Συνάρτηση <code>audiorecorder</code> .....	32
5.4.2	Συνάρτηση <code>recordblocking</code> .....	32
5.4.3	Συνάρτηση <code>getaudiodata</code> .....	33
5.4.4	Συνάρτηση <code>audiowrite</code> .....	33
5.4.5	Συνάρτηση <code>fft</code> .....	33
5.4.6	Συνάρτηση <code>findpeaks</code> .....	33
5.4.7	Συνάρτηση <code>xcorr</code> .....	34
5.4.8	Συνάρτηση <code>find</code> .....	34
5.5	Συμπέρασμα .....	34
5.6	Επίλογος.....	34
Κεφάλαιο 6 <sup>ο</sup> : Αλγοριθμική επεξεργασία ήχτικου περιεχομένου.....		35
6.1	Εισαγωγή.....	35
6.2	Φασματική ανάλυση στις νότες του μπουζουκιού .....	36
6.3	Αλγόριθμος επεξεργασίας: Παρουσίαση και επεξηγήσεις.....	41
6.3.1	Καθορισμός παραμέτρων και εισαγωγή αρχείου ήχου .....	41
6.3.2	Επεξεργασία αρχείου ήχου στο πεδίο του χρόνου. ....	44
6.3.3	Επεξεργασία αρχείου ήχου στο πεδίο της συχνότητας.....	48
6.3.4	Εμφάνιση αποτελεσμάτων στην οθόνη .....	51
6.3.5	Οργάνωση παραμέτρων με βάση το πρότυπο <code>musicxml</code> .....	52
6.3.6	Υπολογισμός μέτρων παρτιτούρας.....	54
6.3.7	Δημιουργία αρχείου <code>musicxml</code> .....	56
6.4	Επίλογος.....	59
Κεφάλαιο 7 <sup>ο</sup> : Αποτελέσματα.....		61
7.1	Εισαγωγή.....	61
7.2	Αποτελέσματα .....	61
7.2.1	Εγγραφή 1 <sup>η</sup> "Χαράματα η ώρα τρεις" .....	61
7.2.2	Εγγραφή 2 <sup>η</sup> "Απονη ζωή" .....	66
7.2.3	Εγγραφή 3 <sup>η</sup> "Η Ρόζα η Ναζιάρα" .....	71
7.4	Επίλογος.....	76

Κεφάλαιο 8 <sup>ο</sup> : Συμπεράσματα και προτάσεις βελτίωσης .....	77
8.1    Συμπεράσματα.....	77
8.2    Προτάσεις βελτίωσης .....	78
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	79
ΠΑΡΑΡΤΗΜΑ: Αλγόριθμος επεξεργασίας Matlab .....	81

## Κατάλογος Εικόνων

Εικόνα 2.1 Τα μέρη του μπουζουκιού	7
Εικόνα 2.2 Πανδούρα, ένα αρχαίο Ελληνικό όργανο.	8
Εικόνα 2.3 Μπουζούκι οργανοποιείου NEORION.	10
Εικόνα 2.4 Ο Μανώλης Χιώτης με μπουζούκι Ζοζέφ.	11
Εικόνα 2.5 Η γραφή στο πεντάγραμμο της νότας Ρε κάτω.	12
Εικόνα 2.6 Η γραφή στο πεντάγραμμο της νότας Ρε επάνω.	12
Εικόνα 2.7 Η γραφή στο πεντάγραμμο της νότας Λα μεσαία.	13
Εικόνα 3.1 Η δομή του πενταγράμμου.	15
Εικόνα 3.2 Οι νότες πάνω στο πεντάγραμμο.	16
Εικόνα 3.3 Οι νότες στο κλειδί του Σολ πάνω στο πεντάγραμμο.	16
Εικόνα 3.4 Δύο οκτάβες στο κλειδί του Σολ της δεύτερης γραμμής.	17
Εικόνα 3.5 Οι νότες στην ταστιέρα του μπουζουκιού.	17
Εικόνα 3.6 Αναλογία του ολόκληρου με τις υπόλοιπες αξίες.	18
Εικόνα 3.7 Η έννοια του μέτρου.	19
Εικόνα 3.8 Η χρονική διάρκεια του μετρίου.	19
Εικόνα 3.9 Η σειρά του οπλισμού πάνω στο πεντάγραμμο.	19
Εικόνα 4.1 Ήχος.	21
Εικόνα 4.2 Αντικείμενα που παράγουν ηχητικά κύματα.	21
Εικόνα 4.3 Πνευστό όργανο παράγει ήχο.	22
Εικόνα 4.4 Γραφική παράσταση ήχου.	22
Εικόνα 4.5 Συχνότητες που ακούει το ανθρώπινο αυτί.	23
Εικόνα 4.6 Πλάτος κυματομορφής ηχητικού κύματος.	23
Εικόνα 4.7 Διάγραμμα του ανθρώπινου αυτιού.	24
Εικόνα 4.8 Δειγματοληψία.	25
Εικόνα 4.9 Ψηφιοποίηση.	26
Εικόνα 4.10 Συνοψίζοντας την ψηφιοποίηση.	28
Εικόνα 5.1 Κυματομορφή από την εγγραφή μπουζουκιού μετά τη χρήση της συνάρτησης get.	32
Εικόνα 6.1 Διαχωρισμός σε διαστήματα μετά τη χρήση της συνάρτησης mirsegment.	36
Εικόνα 6.2 D4 στη χορδή D με πένα θέση και άρση.	38
Εικόνα 6.3 E4 στη χορδή D με πένα θέση και άρση.	38
Εικόνα 6.4 G4 στη χορδή D με πένα θέση και άρση.	39

Εικόνα 6.5 C5 στη χορδή D με πένα θέση και άρση.	39
Εικόνα 6.6 A3 στη χορδή A με πένα θέση και άρση.	40
Εικόνα 6.7 C4 στη χορδή A με πένα θέση και άρση.	40
Εικόνα 6.8 D4 στη χορδή A με πένα θέση και άρση.	40
Εικόνα 6.9 G4 στη χορδή A με πένα θέση και άρση.	41
Εικόνα 6.10 Γενικό διάγραμμα ροής του κώδικα.	42
Εικόνα 6.11 Εισαγωγή δεδομένων και παραμετροποίηση.	44
Εικόνα 6.12 Επεξεργασία στο πεδίο του χρόνου.	45
Εικόνα 6.13 Αποτελέσματα χρήσης των συναρτήσεων του mirtoolbox.	46
Εικόνα 6.14 Επεξεργασία στο πεδίο της συχνότητας.	47
Εικόνα 6.15 Αυτοσυσχέτιση φάσματος πλάτους δύο περιπτώσεων.	49
Εικόνα 6.16 Εκτύπωση αποτελεσμάτων στην οθόνη.	52
Εικόνα 6.17 Οργάνωση μεταβλητών.	53
Εικόνα 6.18 Υπολογισμός των μέτρων της παρτιτούρας.	55
Εικόνα 6.19 Δημιουργία αρχείου musicxml	57
Εικόνα 7.1 Κυματομορφή από την 1η εγγραφή στο πεδίο του χρόνου.	61
Εικόνα 7.2 Κυματομορφή με τη χρήση της mirenvelope και διαφορισμού.	61
Εικόνα 7.3 Παρτιτούρα όπως προκύπτει με την χρήση του κώδικα.	65
Εικόνα 7.4 Παρτιτούρα που παίχτηκε από τον μουσικό στο μπουζούκι.	65
Εικόνα 7.5 Κυματομορφή από την 2η εγγραφή στο πεδίο του χρόνου .	66
Εικόνα 7.6 Κυματομορφή με τη χρήση της mirenvelope και διαφορισμού	66
Εικόνα 7.7 Παρτιτούρα όπως προκύπτει με την χρήση του κώδικα.	69
Εικόνα 7.8 Παρτιτούρα που παίχτηκε από τον μουσικό στο μπουζούκι .	69
Εικόνα 7.9 Αυτοσυσχέτιση φάσματος πλάτους..	70
Εικόνα 7.10 Κυματομορφή από την 3η εγγραφή στο πεδίο του χρόνου .	71
Εικόνα 7.11 Κυματομορφή με τη χρήση της mirenvelope και διαφορισμού.	71
Εικόνα 7.12 Αυτοσυσχέτιση φάσματος πλάτους..	74
Εικόνα 7.13 Παρτιτούρα όπως προκύπτει με την χρήση του κώδικα.	74
Εικόνα 7.14 Παρτιτούρα που παίχτηκε από τον μουσικό στο μπουζούκι.	75

## Κατάλογος Πινάκων

Πίνακας 3.1 Αριθμητικά δεδομένα	18
---------------------------------	----

## Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
MACOS	Λειτουργικό σύστημα της Apple
MIDI	Musical Instrument Digital Retrieval
MIR	Music Information Retrieval
PDF	Portable Document Format
MP3	MPEG-1 Audio Layer 3
WAV	Waveform Audio File Format
AIFF	Audio Interchange File Format
FFT	Fast Fourier Transform
BPM	Beats per minute
XML	eXtensible Markup Language



## Κεφάλαιο 1<sup>ο</sup>: Εισαγωγή

Η παρούσα πτυχιακή εργασία έχει σαν αντικείμενο μελέτης την επεξεργασία αρχείων τύπου wav, που περιέχουν ηχητικό υλικό από ηχογραφήσεις μπουζουκιού. Η επεξεργασία αυτή σε πρώτη φάση περιλαμβάνει χρήση αλγόριθμων, που θα αναλυθούν παρακάτω, και από την οποία θα προκύψουν φασματικές αναλύσεις κατάλληλες για περαιτέρω μελέτη. Εκεί θα δούμε την αναγνώριση μουσικών νότων, την εύρεση μουσικής αξίας και διαστημάτων, και τη δημιουργία συναρτήσεων με απώτερο σκοπό την εξαγωγή κάποιας βασικής παρτιτούρας.

### 1.1 Υπόβαθρο

Τα εργαλεία που προσφέρει η σύγχρονη μουσική τεχνολογία για παρόμοιες αναλύσεις είναι αρκετά, μεταξύ των οποίων κάποια ανοικτού κώδικα για δωρεάν χρήση, και άλλα με αγορά για πιο επαγγελματική χρήση όμως κανένα εξ αυτών δεν μπορεί ακόμη να προσφέρει την ολοκληρωμένη και αξιόπιστη επεξεργασία ώστε να προκύψει από αρχείο wav, που περιέχει μουσικό κομμάτι, παρτιτούρα μουσικά ολοκληρωμένη, έτοιμη να διαβαστεί και να αναπαραχθεί από μουσικούς. Αυτό φυσικά συμβαίνει λόγω της εξαιρετικά πολύπλοκης και δύσκολης προσέγγισης του θέματος διότι υπάρχουν χιλιάδες μουσικά όργανα και αμέτρητες προσεγγίσεις στο εύρος της ανθρώπινης φωνής που καθιστά αδύνατο τη δημιουργία βάσης δεδομένων για σύγκριση και παραγωγή πλήρης και αξιόπιστης παρτιτούρας. Στην εποχή μας και με τη χρήση της τεχνητής νοημοσύνης τα μοντέλα μηχανικής μάθησης προσεγγίζουν πολύ καλύτερα το θέμα και σύντομα πιστεύω πως θα προσφέρουν πιο ολοκληρωμένες λύσεις. Ωστόσο δεν πιστεύω ότι θα μπορέσουν να αποτελέσουν αξιόπιστη αντικατάσταση του ανθρώπινου παράγοντα στη συγγραφή παρτιτούρας από ένα αρχείο πολύπλοκου μουσικού έργου. Παρακάτω θα σας παρουσιάσω μερικά από τα κυριότερα προγράμματα μουσικής επεξεργασίας που είναι διαθέσιμα στους ενδιαφερομένους

- Muse score
- Music 21
- Cecilia
- Ardour

Τα πρώτα τρία είναι ανοικτού κώδικα και μπορούμε να τα βρούμε στο διαδίκτυο και να τα χρησιμοποιήσουμε ελεύθερα ενώ το τέταρτο εξ αυτών αποτελεί επαγγελματικό εργαλείο με πολλές δυνατότητες που διατίθεται μόνον στους αγοραστές του. Θα ακολουθήσει μια περιορισμένη παρουσίαση για να σας τα κάνω γνωστά.

#### 1.1.1 Muse Score

Το **Muse Score** είναι ένα πρόγραμμα σύνθεσης και επεξεργασίας μουσικού κειμένου, ανοικτού κώδικα και διατίθεται δωρεάν. Θεωρείται ένα από τα καλύτερα προγράμματα του είδους που κυκλοφορούν γιατί λειτουργεί σε περιβάλλον και Windows, και MacOS και Linux. Τα πλεονεκτήματα που προσφέρει στον χρήστη αναφέρονται παρακάτω και δικαιολογούν πλήρως τον τίτλο του καλύτερου δωρεάν προγράμματος στον κόσμο

- Εύκολη εισαγωγή & επεξεργασία: Υποστηρίζει εισαγωγή με ποντίκι, πληκτρολόγιο, Midi keyboard, και σάρωση παρτιτούρων.
- Πλούσια βιβλιοθήκη οργάνων και συμβόλων: Περιλαμβάνει μεγάλη ποικιλία οργάνων και συμβόλων για διάφορα είδη μουσικής.

- Εργαλεία ενορχήστρωσης: Μεταφορά μερών, αλλαγή κλειδιού, και πολλά άλλα εργαλεία για εύκολη ενορχήστρωση.
- Δημιουργία παρτιτούρων για πολλά όργανα: Σύνθεση για ορχήστρες, μπάντες, χορωδίες, και μικρότερα σύνολα.
- Ρεαλιστική αναπαραγωγή: Χρησιμοποιεί υψηλής ποιότητας samples ήχου για ρεαλιστική απεικόνιση της μουσικής.
- Εξαγωγή σε διάφορες μορφές: Αποθήκευση μουσικής σε διάφορες μορφές όπως pdf, Musicxml, Midi, Mp3, Wav, και άλλα.
- Εκτύπωση παρτιτούρων: Εύκολη εκτύπωση παρτιτούρων σε διάφορα μεγέθη και μορφές.
- Σύνδεση με Midi συσκευές: Χρήση Midi πληκτρολογίου ή άλλης συσκευής για εισαγωγή και αναπαραγωγή μουσικής.
- Ενεργή κοινότητα: Μεγάλη διαδικτυακή κοινότητα για βοήθεια, μαθήματα, και ανταλλαγή αρχείων.

Επιπλέον, αξίζει να σημειωθεί ότι είναι πλήρως εξελληνισμένο χάρη στη μετάφραση από το Τμήμα Μουσικών Σπουδών του Ιονίου Πανεπιστημίου που έχει κάνει εξαιρετική δουλειά και θα βοηθήσει στην ακόμη μεγαλύτερη επέκταση του στη χώρα μας. Είναι επίσης ιδανικό και για αρχάριους και για επαγγελματίες μουσικούς, καθώς προσφέρει εύχρηστο περιβάλλον εργασίας και απεριόριστες δυνατότητες.[1]

### 1.1.2 Music 21

Το **Music 21** είναι μια ισχυρή βιβλιοθήκη Python και χρησιμοποιείται για ανάλυση επεξεργασία και δημιουργία μουσικής. Ιδιαίτερα χρήσιμη βιβλιοθήκη για μουσικολόγους, σύνθετες και προγραμματιστές που πρέπει να δουλέψουν μουσικά δεδομένα, Έχει πολλές δυνατότητες όπως

- Ανάλυσης παρτιτούρας
- Σύνθεση και δημιουργία μουσικής τόσο αλγοριθμικά όσο και άμεσα
- Δημιουργία μουσικών παραδειγμάτων στη διδασκαλία βασικών αρχών της μουσικής θεωρίας
- Δημιουργίας μουσικής, μετατροπής μεταξύ διαφορετικών μορφών αρχείων (όπως MusicXML, MIDI)
- Επεξεργασία μουσικής σημειογραφίας, δημιουργία και πολλά ακόμη.

Το **Music 21** υπάρχει από το 2008 και συνεχώς αναπτύσσεται και επεκτείνεται. Η προέλευση του είναι το MIT.όπου όλα τα μαθήματα έχουν αριθμούς και η μουσική, μαζί με κάποια άλλα τμήματα ανθρωπιστικών επιστημών, αριθμούνται 21. Τα μουσικά τμήματα του MIT, μαζί με τα κολέγια Harvard, Smith και Mount Holyoke, βοήθησαν να δημιουργηθεί αυτή η βιβλιοθήκη στην εύκολα κατανοητή γλώσσα προγραμματισμού "Python". Το **Music21** μπορεί να είναι απλό στη χρήση αλλά είναι επίσης εξαιρετικά ισχυρό Για να χρησιμοποιηθεί απαιτείται κάποια εξοικείωση με τη γλώσσα προγραμματισμού Python, η οποία θεωρείται ως μια από τις ευκολότερες γλώσσες για εκμάθηση και συχνά σήμερα διδάσκεται ως πρώτη γλώσσα προγραμματισμού.[2]

### 1.1.3 Cecilia

Το **Cecilia** είναι ένα περιβάλλον επεξεργασίας ηχητικού σήματος Θεωρείται σύγχρονο εργαλείο για επεξεργασία ήχου με τον ευρύτερο ορισμό του. Ήχος ως μουσική, ήχος ως σύνθεση και ήχος ως αξεσουάρ. Είναι ιδανικό για σχεδιαστές ήχου Η Cecilia σας επιτρέπει να δημιουργήσετε το δικό σας γραφικό περιβάλλον χρήστη (γράφημα, ρυθμιστικά, εναλλαγές, αναδύομενα μενού) χρησιμοποιώντας μια απλή σύνταξη. Η **Cecilia** έρχεται με πολλές πρωτότυπες ενσωματωμένες μονάδες για ηχητικά εφέ και σύνθεση Η **Cecilia** εξυπηρετεί τον σύνθετη με μικρή εμπειρία προγραμματισμού και τον προγραμματιστή με μικρή εμπειρία σύνθεσης Δημιουργήθηκε πρώτη φορά το 1995-96 για να αντικαταστήσει τα παλιά αναλογικά/MIDI στούντιο, ως σύστημα σύνθεσης musique concrete Όλη η **Cecilia** είναι γραμμένη σε tcl/tk μια αργή αλλά κατάλληλη γλώσσα σεναρίων μιας και δε χρειάζεται να κάνει μαθηματικούς υπολογισμούς ήχου. Τώρα ξαναγράφεται εξολοκλήρου σε python.wxPython και χρησιμοποιεί τη μηχανή ήχου pyo που δημιουργήθηκε για τη γλώσσα προγραμματισμού Python και υποστηρίζεται από τον Jean Piche.[3]

### 1.1.4 Ardour

Το **Ardour** είναι επαγγελματικό πρόγραμμα ηχογράφησης, επεξεργασίας, και μίξης ήχου Πρόκειται για ένα DAW δηλαδή Digital Audio Workstation και στα ελληνικά μεταφράζεται σαν πρόγραμμα που μπορεί να ηχογραφήσει, να επεξεργαστεί και να μιξάρει αρχεία ήχου Οι δυνατότητες του **Ardour** ξεκινούν από την ηχογράφηση είτε μέσω μικρόφωνου είτε μιας διαφορετικής συσκευής εισόδου ήχου Έπειτα μπορούμε να το επεξεργαστούμε να κόψουμε να συνδυάσουμε ή να διαγράψουμε σημεία που δε μας ενδιαφέρουν και στο στάδιο τη μίξης οι γνώστες θα βρουν όλες τις απαραίτητες λειτουργίες για να φέρουν το κομμάτι στα μετρά τους Παρά πολλά εφέ και ρυθμίσεις είναι διαθέσιμα και οι δυνατότητες είναι πραγματικά άπειρες.[4]

## 1.2 Σκοπός της πτυχιακής εργασίας

Ο σκοπός της παρούσας διπλωματικής εργασίας είναι η δημιουργία κατάλληλων συναρτήσεων που θα επιτρέπουν τη φασματική ανάλυση μιας μελωδίας από ένα αρχείο ήχου τύπου **WAV**, με μονοφωνικές ηχογραφήσεις μπουζουκιού και απώτερο στόχο την εξαγωγή βασικής παρτιτούρας. Αρχικά, στο πρώτο μέρος, στόχος είναι η φασματική ανάλυση της μελωδίας του ήχου του μπουζουκιού που έχει ηχογραφηθεί σε αρχεία ήχου **WAV**. Σε δεύτερο χρόνο η αναγνώριση των νοτών που συνθέτουν την ηχογραφημένη μελωδία και θα προκύπτει μέσω της ανάλυσης του ηχητικού φάσματος, χρησιμοποιώντας αλγόριθμους που περιγράφονται στα επόμενα κεφάλαια. Ωστόσο, η ανάλυση του ήχου δεν είναι μια απλή διαδικασία, διότι οι συχνότητες συχνά αλληλεπιδρούν και περιπλέκονται μεταξύ τους, ειδικά στο μπουζούκι που έχει δίπλες χορδές και παράγει πολλές σειρές αρμονικών σε κάθε νότα. Επιπλέον, ψάχνουμε την εύρεση μουσικών αξιών και διαστημάτων και τέλος δημιουργούμε κώδικα που παρέχει αναγνώριση μελωδίας και εξαγωγή παρτιτούρας. Αυτό το τελευταίο κομμάτι της πτυχιακής εργασίας, δηλαδή η εξαγωγή παρτιτούρας, θα γίνει με τη βοήθεια του πρότυπου **Music XML** στο οποίο θα αναφερθεί με κάποιες λεπτομέρειες παρακάτω.

### 1.2.1 Φασματική ανάλυση του ήχου του μπουζουκιού

Η φασματική ανάλυση ενός ηχητικού αρχείου μπουζουκιού γίνεται για να μας αποκαλύψει τις συχνότητες που συνθέτουν τον ήχο του, επιτρέποντας την κατανόηση της αρμονικής δομής και των χαρακτηριστικών του. Αυτή η διαδικασία περιλαμβάνει τη χρήση λογισμικού ανάλυσης ήχου, που υπάρχει στο **MIR toolbox** (βιβλιοθήκη της **MATLAB**) και την χρήση συναρτήσεων για την εφαρμογή μετασχηματισμού Fourier (FFT) στο ηχητικό αρχείο. Χρησιμοποιώντας αυτή την ανάλυση

ήχου και βλέποντας στο φάσμα συχνοτήτων τις κορυφές, μπορούμε να εντοπίσουμε τις θεμελιώδεις και αρμονικές συχνότητες.

### 1.2.2 Αναγνώριση νότας

Για να βρούμε τις νότες από τις θεμελιώδεις συχνότητες και τις αρμονικές, χρειάζεται να αντιστοιχίσουμε τις συχνότητες στις αντίστοιχες μουσικές νότες. Αυτό γίνεται μέσω της σχέσης των συχνοτήτων με τα μουσικά διαστήματα. Κατά αρχάς θεωρούμε ότι η θεμελιώδης συχνότητα είναι η χαμηλότερη και βασική συχνότητα που εκπέμπει ο ήχος. Κατά δεύτερον και χρησιμοποιώντας πινάκες αντιστοίχισης νοτών με συχνότητες συνδέουμε την θεμελιώδη συχνότητα με κάποια νότα. Υπάρχουν συγκεκριμένα εργαλεία στο **MIR toolbox** που κάνουν αυτή την εργασία αυτόματα. Οι αρμόνικες συχνότητες (πολλαπλάσια της θεμελιώδους) είναι και αυτές με τη σειρά τους σημαντικές ίσως σε κάποια άλλη επεξεργασία γιατί είναι αυτές που ενισχύουν τη χροιά του ήχου.

### 1.2.3 Εύρεση αξίας και διαστημάτων

Για να βρούμε τη μουσική αξία (διάρκεια νότας) δηλαδή αν είναι ολόκληρη νότα, μισή, τέταρτο και ούτω καθεξής, χρησιμοποιούμε πάλι κάποια συγκεκριμένη συνάρτηση που αναγνωρίζει με συγκεκριμένο λογισμικό αυτές τις μουσικές αξίες. Στα αμέσως επόμενα κεφάλαια της πτυχιακής θα αναπτύξουμε και τη μουσική θεωρία αλλά πολύ δε μάλλον τη βιβλιοθήκη και τις συναρτήσεις. Όσον αφορά τα διαστήματα εκφράζουν την απόσταση σε συχνότητες ή μουσικές νότες και λέγονται μελωδικά διαστήματα ή αφορούν νότες που ακούγονται ταυτόχρονα και αναφέρονται σαν αρμονικά διαστήματα (συγχορδίες).

### 1.2.4 Δημιουργία συναρτήσεων για εξαγωγή παρτιτούρας

Για την εξαγωγή παρτιτούρας δεν μπορούμε να χρησιμοποιήσουμε άμεσα τη βιβλιοθήκη του **MIR toolbox**. Με το **MIR toolbox** θα κάνουμε όλες τις αναλύσεις στο ηχητικό αρχείο και αφού το ολοκληρώσουμε και έχουμε νότα, οκτάβα, και χρονική διάρκεια, θα τα εισάγουμε στο πρότυπο **Music XML** και αφού τα επεξεργαστούμε κατάλληλα θα τα εξάγουμε στο **Muse Score**, που δέχεται σαν είσοδο αρχείο **Music XML** για να έχουμε τελικά την παρτιτούρα μας.

## 1.3 MusicXML

Το **MusicXML** είναι ένα πρότυπο αρχείο ανοικτού κώδικα βασισμένο στην **XML** (extensible markup language), δηλαδή μια γλώσσα σήμανσης με κανόνες για την ηλεκτρονική κωδικοποίηση κείμενων). Χρησιμοποιείται για την αναπαράσταση της δυτικής μουσικής σημειογραφίας. Δημιουργήθηκε για να διευκολύνει την ανταλλαγή μουσικών παρτιτούρων μεταξύ διαφορετικών προγραμμάτων σημειογραφίας, όπως το **Finale**, το **Sibelius**, και το **Muse Score**. Στόχος είναι να δημιουργηθεί μια καθολική μορφή για κοινή δυτική μουσική σημειογραφία, παρόμοια με το ρόλο που εξυπηρετεί η μορφή MP3 στην ηχογραφημένη μουσική. Το **Music XML** υποστηρίζει την εισαγωγή και εξαγωγή δεδομένων, επιτρέποντας στους χρήστες να μεταφέρουν μουσικές παρτιτούρες από ένα πρόγραμμα σε άλλο χωρίς να χάνονται πληροφορίες. Όταν ένα αρχείο **Music.XML** ανοίγει σε ένα πρόγραμμα σημειογραφίας, το πρόγραμμα προσπαθεί να αναπαραστήσει τη μουσική όσο το δυνατόν πιο πιστά, μπορεί ωστόσο να χρειαστεί κάποια επεξεργασία για να ταιριάξει η εμφάνιση της παρτιτούρας με την αρχική. Επίσης, το **Music.XML** προσφέρει μια σειρά από δυνατότητες που το καθιστούν ιδανικό για την ανταλλαγή μουσικών παρτιτούρων, όπως:

- Συμβατότητα: Υποστηρίζεται από πολλά προγράμματα σημειογραφίας, όπως το **Finale**, το **Sibelius**, και το **MuseScore**, επιτρέποντας την εύκολη μεταφορά αρχείων μεταξύ τους.

- Πλούσια Σημειογραφία: Μπορεί να αναπαραστήσει σύνθετα στοιχεία σημειογραφίας, όπως αρμονίες, διαγράμματα συγχορδιών, και ειδικές τεχνικές για όργανα.
- Συμπιεσμένα Αρχεία: Υποστηρίζει τη μορφή .mxl, που μειώνει το μέγεθος των αρχείων.
- Ευελιξία: Επιτρέπει την προσαρμογή και καθαρισμό των εισαγόμενων παρτιτούρων για να ταιριάζουν στις ανάγκες του χρήστη [5] [6]

## 1.5 Ανάπτυξη της πτυχιακής εργασίας

Στο πρώτο κεφάλαιο γίνεται εισαγωγή στο θέμα που αποτελεί το αντικείμενο μελέτης της πτυχιακής εργασίας καθώς και στον σκοπό αυτής. Περιγράφεται συνοπτικά η μεθοδολογία και τα εργαλεία που χρησιμοποιούνται παρακάτω, καθώς και τι περιέχουν τα κεφάλαια που την απαρτίζουν

Στο δεύτερο κεφάλαιο παρουσιάζεται το μουσικό όργανο μπουζούκι. Αρχικά, περιγράφεται το όργανο, και μια ιστορική αναδρομή στην προέλευση του. Επίσης παρουσιάζεται η τεχνική κατασκευή του, και τέλος οι μουσικές του δυνατότητες

Στο τρίτο κεφάλαιο γίνεται μια σύντομη αναφορά στη μουσική θεωρία και μια επαφή γνωριμίας με την τέχνη της μουσικής και τη σημειογραφία της. Ευχής έργο στο τέλος του κεφαλαίου να έχουμε βασικές γνώσεις για το γράψιμο και το διάβασμα της μουσικής

Στο τέταρτο κεφάλαιο γίνεται σύντομη αναφορά στον ήχο. Δίνουμε ορισμό και βασικά στοιχεία για τα χαρακτηριστικά του. Μιλούμε για την ακουστική και τον τρόπο της ανθρώπινης ακοής. Επικεντρωνόμαστε στον ψηφιακό ήχο και μελετούμε τη μετατροπή, τη δειγματοληψία, τη κβάντιση, το μέγεθος, το ρυθμό μετάδοσης, τη συμπίεση και τέλος τη μεταφορά την αποθήκευση και αναπαραγωγή των αρχείων του ήχου.

Στο πέμπτο κεφάλαιο ξεκινά η παρουσίαση της βιβλιοθήκης MIR toolbox που χρησιμοποιούμε μαζί με τη MATLAB για την ανάλυση και επεξεργασία ηχητικών αρχείων. Από την τεράστια αυτή βιβλιοθήκη φυσικά επικεντρωνόμαστε στις συναρτήσεις στις όποιες χρησιμοποιήσαμε είτε μας οδήγησαν στον σκοπό μας, είτε όχι.

Το έκτο κεφάλαιο αποτελεί τη βάση της εργασίας, και παρουσιάζεται όλη η αλγοριθμική επεξεργασία που γράφτηκε προκειμένου να καταλήξουμε στο ποθούμενο αποτέλεσμα. Χωρίζεται σε δυο βασικές ενότητες. Η πρώτη έχει τον αλγόριθμο από την αρχή της επεξεργασίας μέχρι τη στιγμή της εξαγωγής νότας, οκτάβας, και χρονικής αξίας. Η δεύτερη ενότητα έχει το κομμάτι του αλγορίθμου από το αρχείο **music.xml** και ως το τέλος, που με τη βοήθεια του **musescore** παίρνουμε την παρτιτούρα.

Στο έβδομο κεφάλαιο, συνοψίζονται τα αποτελέσματα που προέκυψαν κατά την ολοκλήρωση της πτυχιακής εργασίας. Παρουσιάζονται οι παρτιτούρες των ηχογραφήσεων που μελετήθηκαν και αντισυμβάλλονται με αυτές που προέκυψαν από τον κώδικα μετά την επεξεργασία τους. Έτσι με τη σύγκριση αυτή προκύπτουν και τα αποτελέσματα της μεθόδου

Τέλος, στο κεφάλαιο οκτώ παρουσιάζονται τα συμπεράσματα της μεθοδολογίας που αναπτύχθηκε στην παρούσα πτυχιακή εργασία. Επίσης, εξετάζονται οι περιορισμοί της παρούσας μεθοδολογίας, οι προκλήσεις που αντιμετωπίστηκαν κατά την υλοποίηση του προγράμματος και τελικά προτείνονται μέθοδοι για την περεταίρω εξέλιξή της.

## 1.6 Επίλογος

Σε αυτό το κεφάλαιο παρουσιάστηκε το θέμα της πτυχιακής εργασίας. Έγινε μια μικρή αναφορά σε τέσσερα προγράμματα που μπορούν να κάνουν αντίστοιχα πράγματα με τον αλγόριθμο που θα

## Κεφάλαιο 1

αναπτυχτεί παρακάτω. Μετά αναφέρονται τα βήματα που θα ακολουθηθούν κατά την εργασία μας, και παρουσιάζεται και το αρχείο **Music XML**. Τέλος περιγράφεται το περιεχόμενο όλων των κεφαλαίων της πτυχιακή εργασίας.

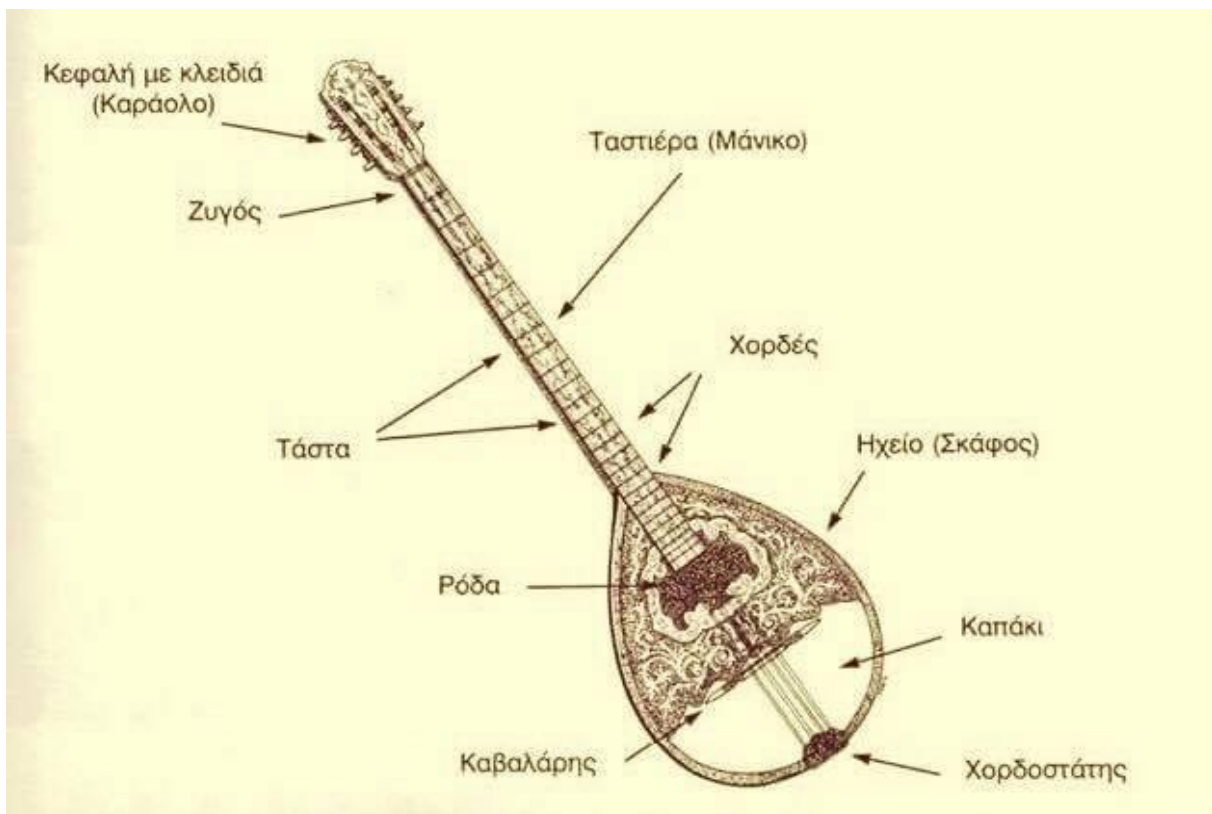
## Κεφάλαιο 2<sup>ο</sup>: Το μουσικό όργανο μπουζούκι

### 2.1 Εισαγωγή

Στο δεύτερο κεφάλαιο θα παρουσιαστεί το μουσικό όργανο μπουζούκι. Αρχικά, θα γίνει μια μικρή περιγραφή του οργάνου, μετά μια ιστορική αναδρομή στην προέλευση του και στην ονομασία του, καθώς και στην εξέλιξη του έως σήμερα. Επίσης θα ειπωθούν λίγα λόγια για την τεχνική του στην κατασκευή, και τέλος θα μιλήσουμε για το μουσικό του εύρος, τις δυνατότητες του και τις κλίμακες που καλύπτει.

### 2.2 Περιγραφή

Το μπουζούκι είναι λαουτοειδές έγχορδο λαϊκό μουσικό όργανο με αχλαδόσχημο αντηχείο (σκάφος) και μακρύ βραχίονα, μπράτσο η μάνικο με κλειδιά στην άκρη για το χόρδισμα (κούρδισμα). Το σύγχρονο μπουζούκι (εικόνα 2.1) διαθέτει τρεις η τέσσερις διπλές χορδές τις οποίες χτυπά ο μουσικός με την πένα. Αρχικά το μπουζούκι είχε τρεις χορδές σε ζεύγη, κουρδισμένες σε τόνους ΡΕ-ΛΑ-ΡΕ, ενώ αργότερα απέκτησε και τέταρτο ζεύγος χορδών και κούρδισμα που αντέγραψε από την κλασσική κιθάρα βαρύτερο κατά ένα τόνο ΡΕ-ΛΑ-ΦΑ-ΝΤΟ.



Εικόνα 2.1: Τα μέρη του μπουζουκιού [7]

Το μπουζούκι χρησιμοποιήθηκε ευρύτατα στο 2<sup>ο</sup> μισό του 20<sup>ου</sup> αιώνα στη λαϊκή μας δισκογραφία (ρεμπέτικη, λαϊκή και έντεχνη) ως πρωταρχικό βασικό μουσικό όργανο παραγκωνίζοντας παραδοσιακά λαϊκά όργανα όπως το κλαρίνο και το βιολί. Από τα μπουζούκια (νυχτερινά μαγαζιά που έχουν πάρει το όνομα τους από το μουσικό όργανο) μέχρι τα ρεμπετάνικα και τις μουσικοταβέρνες, το μπουζούκι είναι το βασικό όργανο, πάνω στο οποίο στηρίζεται όλη η μπάντα.

### 2.3 Ιστορική αναδρομή

Για την καταγωγή του οργάνου και την ονομασία του έχουν γραφτεί παρά πολλά και με εντελώς διαφορετικές προσεγγίσεις. Από όλα αυτά οι δυο επικρατέστερες απόψεις που θα αναφέρω παρακάτω συνδέουν το μπουζούκι με την αρχαιοελληνική Πανδούρα και έχουν ως εξής:

A) Το μπουζούκι ως μουσικό όργανο είναι ένα είδος μετεξέλιξης της αρχαιοελληνικής Πανδούρας. Στην αρχαία Ελλάδα υπήρχε το αντίστοιχο όργανο το οποίο ήταν γνωστό με τις ονομασίες ‘Πανδουρίδιον’ ή αλλιώς και ‘τρίχορδο’ καθώς είχε τρεις χορδές. Το όργανο αυτό πέρασε από τους αρχαίους Έλληνες στους Βυζαντινούς και επέζησε κατά τη διάρκεια της τουρκοκρατίας. Στις αρχές του 20<sup>ου</sup> αιώνα αμφισβητήθηκε και στη συνέχεια γνώρισε την άνθηση και την καταξίωση ως δημοφιλές λαϊκό όργανο. Έντονη απόδειξη της πηγής αυτής είναι τα μάρμαρα της εποχής, με γνωστότερο αυτό της Μαντινείας (4<sup>ος</sup> αιώνας π.χ.), που σήμερα εκτίθεται στο Εθνικό Αρχαιολογικό Μουσείο της Αθήνας (εικόνα 2.2), και απεικονίζει το μυθικό αγώνα, μεταξύ Απόλλωνος και Μαρσύα, όπου η αρχαιοελληνική πανδούρα παίζεται από μια μούσα καθισμένη πάνω σε έναν βράχο.



Εικόνα 2.2: Πανδούρα, ένα αρχαίο Ελληνικό μουσικό όργανο [8]

B) Το μπουζούκι είναι ένα μουσικό έγχορδο όργανο που ανήκει στην κατηγορία των εγχόρδων που γέννησε η αρχαία ελληνική πανδουρίς (> πανδούρα > ταμπούρα > ταμπουράς, κλπ.). Το όνομά του προέρχεται από την απώτατη αρχαία ελληνική λέξη βυζός > βούζος, που σημαίνει μεγάλο εξόγκωμα, εικόνα που μας παραπέμπει τόσο στο σχήμα του, αφού το σκάφος του οργάνου μοιάζει με τέτοιο, όσο και στη θέση που αυτό ακουμπά στο σώμα του παίκτη καθώς παίζετε, που είναι κάτω ακριβώς από το στήθος. Παλαιότερα το όργανο αυτό το έπαιζαν κυρίως γυναίκες ιέρειες, σε μυσταγωγία και καταγωγή, οι οποίες πολλές φορές ήταν ημίγυμνες (από τη μέση και πάνω) και έτσι ταυτιζόταν το όργανο με το στήθος τους. Η ιστορία του οργάνου, πιθανόν, να κρατά ακόμη και από τη μυσταγωγία των ελευσίνιων μυστηρίων. Σε αυτές τις μυστικές τελετές ακουγόταν μια συγκεκριμένη θρησκευτική μουσική μύησης, η οποία παιζόταν από ένα ιερό μουσικό θρησκευτικό όργανο το οποίο αγνοούμε. Δεν είναι απίθανο λοιπόν το μουσικό όργανο που υποβοηθούσε σε όλη αυτή τη μυσταγωγία των ελευσίνιων μυστηρίων, και που παιζόταν από τον αρχιερέα βουζύγη, να ήταν το εμβουζύγιον. Αυτό το αρχαίο όργανο έφτασε ως τις μέρες μας παραφρασμένο ως μπουζούκι. Όργανο, που στην αρχή της ιστορίας του όλοι ξέρουμε ότι παιζόταν κνηγημένο και παρεξηγημένο, σε υπόγεια και καταγωγή,

όπου ετελείτο η σύγχρονη λαϊκή πια μύηση, στη ζωή και στον έρωτα. Η εξέλιξη του συνέχισε να βρίσκει ιδιαίτερη απήχηση σε πόλεις με λιμάνια (Πειραιάς, Βόλος, Θεσσαλονίκη, Σμύρνη, Σύρος, κλπ.), διατρανώνοντας και συνεχίζοντας έτσι τον λαϊκό του χαρακτήρα [9].

## 2.4 Κατασκευή και τεχνικά χαρακτηριστικά

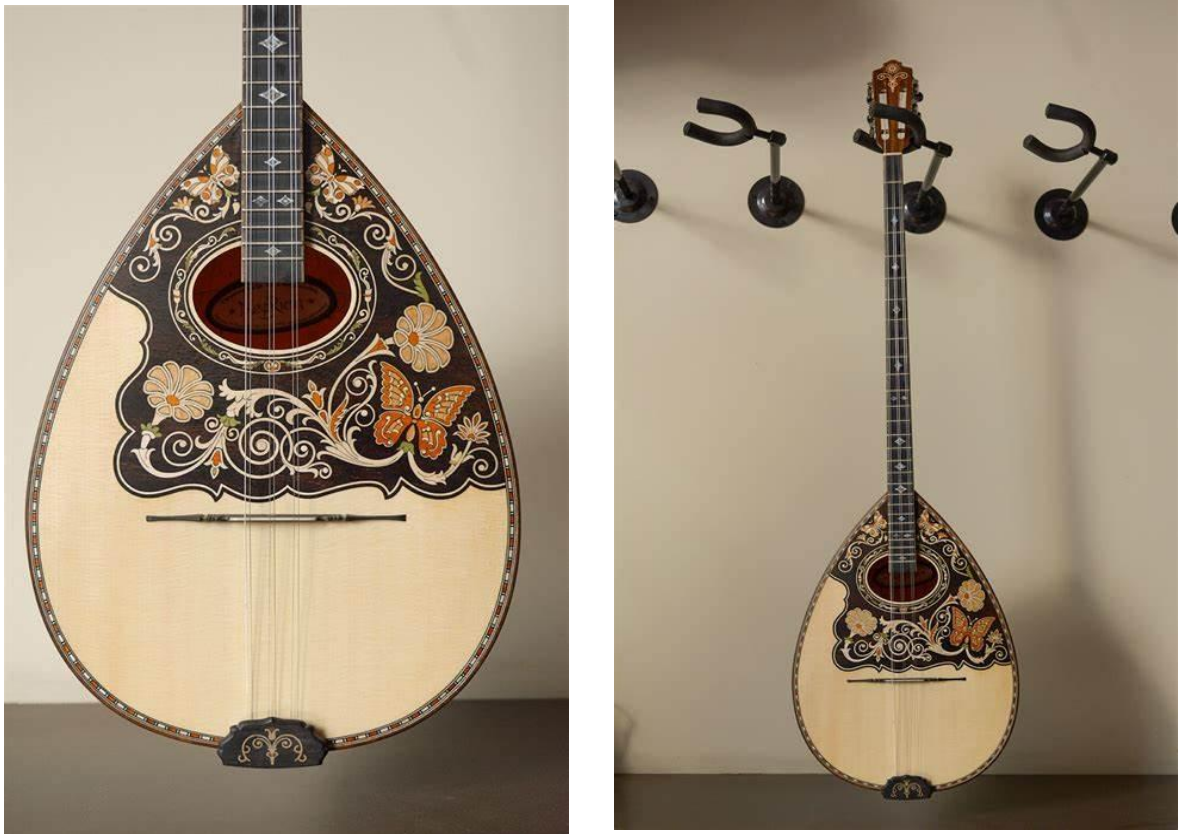
Όπως προαναφέρθηκε παραπάνω, τα σύγχρονα μπουζούκια είναι τρίχορδα ή τετράχορδα και αντιστοίχως ο ήχος τους και το παίξιμο τους ταιριάζουν καλύτερα στη ρεμπέτικη ή στη λαϊκή ορχήστρα. Από κατασκευαστικής άποψης τα μπουζούκια μπορούν να έχουν διαφορές και σε άλλα χαρακτηριστικά τους όπως μήκος μανικού, πλάτος βάθους όσο και ύψος του ηχείου ή σκάφους. Τις διαφορές αυτές τι καθορίζει ο κάθε κατασκευαστής ώστε να πετυχαίνει τον ήχο που προσδοκά (πιο οξύ, πιο βαθύ ή πιο βαρύ ήχο)τροποποιώντας τα λειτουργικά στοιχεία του οργάνου.

Το μέγεθος του σκάφους ενός μπουζουκιού παίζει ρόλο στην τονικότητα του, ενώ από το μήκος του μανικού και κατά επέκταση των χορδών, δίνουν τη διαφορά στην τονικότητα. Εννοείται ότι όλα τα μάνικα έχουν τον ίδιο αριθμό τάστων και αλλάζει μόνο το πλάτος τους σε κάθε όργανο. Πρωταρχικό ρόλο στον ήχο του μπουζουκιού έχει και η ποιότητα των ξύλων από τα οποία κατασκευάστηκε το όργανο.

Για την κατασκευή του σκάφους προτιμούμε τα ξύλα να είναι της μουριάς, της κερασιάς, της ακακίας, της φτελιάς και ακολουθούν της καρδιάς, του πλάτανου, της καστανιάς. Το ξύλο του σκάφους πρέπει να είναι συμπαγές, ιδιότητα που έχουν εκείνα τα ξύλα που προέρχονται από δέντρα βραδείας ανάπτυξης. Το καπάκι του σκάφους πρέπει να είναι από κέδρο ή έλατο (κατά προτίμηση ερυθροελάτη) αν είναι δυνατό, μονοκόμματο. Το καπάκι είναι που παίζει τον κύριο ρόλο στον ήχο γιατί αυτό πάλλεται, ενισχύει και παρατείνει τους παλμούς των χορδών.

Στην ποιότητα του ήχου παίζει ρόλο ο λούστρος και η επεξεργασία του λουστραρίσματος. Καλύτερος είναι ο φυσικός λούστρος από γομμαλάκκα που είναι περασμένος με το χέρι σε πολλά στρώματα, με τον παραδοσιακό τρόπο. Έτσι οι επιφάνειες των ξύλων γίνονται πιο συμπαγείς και πιο ανακλαστικές, πέρα από το καλύτερο αισθητικό αποτέλεσμα. Το μάνικο πρέπει να είναι από πολύ ξερό και σκληρό ξύλο για να μη σκεβρώσει και απομακρύνει τις χορδές από την ταστιέρα, οπότε το όργανο γίνεται φάλτσο και δυσκολόπαιχτο. Για να το πετύχουν αυτό οι οργανοκατασκευαστές χρησιμοποιούν διάφορες τεχνικές κι ο καθένας έχει τα δικά του μυστικά.\

Το καλύτερο ξύλο για μάνικο είναι το σφενδάμι, η φτελιά και η καρδιά. Η πλάκα που κάθετα πάνω στο μάνικο και σ' αυτήν σφηνώνονται τα συρμάτινα τάστα πρέπει να είναι πολύ σκληρό, συμπαγές, ανθεκτικό ξύλο αλλά και όμορφο. Τα καλύτερα για αυτή τη θέση είναι ο έβενος, ο παλίσανδρος, το πυξάρι και η τριανταφυλλιά, ανάλογα με το χρώμα που θέλουμε να έχει η ταστιέρα (ο έβενος μαύρο, ο παλίσανδρος κοκκινιάρικο σκούρο, η τριανταφυλλιά καστανοκόκκινο με όμορφα νερά και το πυξάρι ανοιχτόχρωμο). Τέλος έρχονται τα διακοσμητικά στοιχεία που προσθέτουμε στο σύγχρονο μπουζούκι (εικόνα 2.3). Αυτά όσο πιο πολλά είναι, τόσο ο ήχος γίνεται πιο μουντός.



Εικόνα 2.3 μπουζούκι οργανοποιείου NEORION

Γι' αυτό από τα λιτά μπουζούκια θα ακούσετε πιο καθαρό και πιο "καμπανάτο" ήχο. Οι περισσότεροι κατασκευαστές σήμερα χρησιμοποιούν στα καλά τους όργανα διακοσμητικά από φυσικά υλικά (ξύλο, σεντέφι, ελεφαντόδοντο, ταρταρούγα) και έτσι πετυχαίνουν και "πλουμιστά" και καθαρόηχα μπουζούκια.

Ξακουστός κατασκευαστής ήταν ο περιβόητος Ζοζέφ που έφτιαξε μπουζούκια για τους μεγαλύτερους καλλιτέχνες. Λέγεται ότι το πρώτο τετράχορδο το έφτιαξε αυτός, στις αρχές της δεκαετίας του 1950, κατά παραγγελία και υποδείξεις του Μανώλη Χιώτη (εικόνα 2.4).



Εικόνα 2.4 Ο Μανώλης Χιώτης με μπουζούκι του Ζοζέφ [10]

## 2.5 Μουσικές δυνατότητες του οργάνου

Το μπουζούκι στη νεότερη ιστορία του στις αρχές του 20<sup>ου</sup> αιώνα υπήρξε όργανο της δημοτικής ορχήστρας. Η σύγχρονη ιστορία του ως λαϊκό όργανο αρχίζει από την κλασική περίοδο του ρεμπέτικου, από το 1930 δηλαδή και μετά. Ο Μάρκος Βαμβακάρης ήταν αυτός που γύρω στο 1930 σύστησε την πρώτη λαϊκή ορχήστρα με σολιστικό όργανο το μπουζούκι, το οποίο και ηχογράφησε για πρώτη φορά στην Ελλάδα το 1933. Μεσω αυτών των δισκίων, που κυκλοφόρησαν αμέσως στα αστικά κέντρα, το μπουζούκι άρχισε να γίνεται γνωστό στο ευρύ κοινό. Οι χαρισματικοί δεξιότεχνες καθώς και τα λαϊκά τραγούδια που αναδείκνυαν το μπουζούκι, ήταν αναμφίβολα δυο από του βασικούς παράγοντες που αγαπήθηκε τόσο πολύ και έγινε σύμβολο της ελληνικής λαϊκής μουσικής.

Το μπουζούκι από κατασκευής του σαν έγχορδο όργανο μπορεί να παίζει τρεις ήμισι οκτάβες, έτσι δίνει στον μουσικό τη δυνατότητα που έχουν όλα τα κλασικά έγχορδα για παίξιμο οποιασδήποτε μελωδίας όπως αντίστοιχα το βιολί και το μαντολίνο. Το μόνο που χρειάζεται για να παιχτούν τα κλασικά έργα είναι η προσαρμογή τους στο μπουζούκι.

Ο ποιο ενδεδειγμένος τρόπος για τη γραφή του μπουζουκιού στην παρτιτούρα είναι να γράφεται εκεί που ακούγεται, να γράφεται δηλαδή η χαμηλότερη νότα ρε στη 4<sup>η</sup> βοηθητική γραμμή κάτω από το πεντάγραμμο ή η νότα ντο για το τετράχορδο κάτω από τη 4<sup>η</sup> βοηθητική γραμμή κάτω από το πεντάγραμμο, μέχρι την υψηλότερη νότα φα που γράφεται πάνω από τη 3<sup>η</sup> βοηθητική γραμμή πάνω από το πεντάγραμμο. Καταυτόν τον τρόπο οι βοηθητικές γραμμές κατανέμονται ισόποσα πάνω και κάτω από το πεντάγραμμο με αποτέλεσμα την ευκολότερη ανάγνωση της παρτιτούρας. Οι παρακάτω εικόνες (εικόνα 2.5, 2.6, 2.7) μας δείχνουν ακριβώς αυτό. [11]



Εικόνα 2.5 Η γραφή στο πεντάγραμμο της νότας Ρε κάτω[11]



Εικόνα 2.6 Η γραφή στο πεντάγραμμο της νότας Ρε επάνω[11]





## Κεφάλαιο 3<sup>ο</sup>: Περί μουσικής θεωρίας

### 3.1 Εισαγωγή

Στο τρίτο κεφάλαιο γίνεται μια σύντομη αναφορά στη μουσική θεωρία για να έρθουμε σε επαφή γνωριμίας με την τέχνη της μουσικής και τη σημειογραφία της. Ελπίζω όλων στο τέλος της παρουσίασης να μπορούμε να αντιληφθούμε το σύστημα γραφής της μουσικής (παρτιτούρα), ώστε να γίνει αντιληπτό στη συνέχεια το πειραματικό μέρος της εργασίας και κάποια τεχνικά χαρακτηριστικά αυτής.

### 3.2 Τι είναι η μουσική .

Η μουσική είναι η τέχνη εκείνη η οποία εκφράζεται μέσα από την πλοκή και τον συνδυασμό των ήχων. Για την παραγωγή των ήχων χρησιμοποιούμε την ανθρώπινη φωνή είτε διάφορα μουσικά όργανα. Η μουσική λοιπόν οφείλει την ύπαρξή της στους ήχους. Θεωρία είναι το μάθημα που μας μαθαίνει το γράψιμο και το διάβασμα της μουσικής. Τα χαρακτηριστικά γνωρίσματα των μουσικών ήχων είναι τα παρακάτω.[12]

- Το Ύψος (συχνότητα, ψηλότερος ή χαμηλότερος ήχος)
- Η Διάρκεια (χρονική παράταση των ήχων - άρα των παλμικών δονήσεων)
- Η Δύναμη (δυνατότερος ή σιγανότερος ήχος)
- Η Χροιά (η διαφορετική ονομασία μουσικών οργάνων και φωνών, εξαιτίας του φαινομένου κατά το οποίο ο τόνος ενός οργάνου διαφέρει από τον ίδιο τον τόνο ενός άλλου οργάνου στο ίδιο ύψος)

Τα κυριότερα μουσικά σημεία είναι τα εξής:

- Το πεντάγραμμο
- Τα φθογγόσημα ή νότες
- Τα κλειδιά
- Οι παύσεις
- Οι αλλοιώσεις
- Τα σημεία χρωματισμού

όλα τα παραπάνω γράφονται σε ειδικά φύλλα που ονομάζονται παρτιτούρες.

### 3.3 Το πεντάγραμμο

Πεντάγραμμο ονομάζεται το σχήμα το οποίο αποτελείται από 5 οριζόντιες παράλληλες και ίσες γραμμές ενώ οι αποστάσεις που σχηματίζουν οι 5 γραμμές είναι 4 και ονομάζονται διαστήματα. Τις γραμμές και τα διαστήματα του πενταγράμμου τα μετράμε από κάτω προς τα πάνω. Όλα λοιπόν τα μουσικά σημεία γράφονται ή πάνω στις 5 γραμμές ή μέσα στα 4 διαστήματα. Άρα το πεντάγραμμο είναι το φιλόξενο σπίτι των μουσικών σημείων (εικόνα 3.1).



Εικόνα 3.1 Η δομή του πενταγράμμου[12]

### 3.4 Οι νότες ( Φθόγγοι )

Οι νότες είναι 7 και έχουν τα εξής ονόματα: Ντο, Ρε, Μι, Φα, Σολ, Λα, Σι, γράφονται στις γραμμές και στα διαστήματα, πάνω και κάτω από το πεντάγραμμο. Για να γράψουμε νότες κάτω ή πάνω από το πεντάγραμμο χρησιμοποιούμε τις βοηθητικές γραμμές (εικόνα 3.2).



Εικόνα 3.2 Οι νότες πάνω στο πεντάγραμμο[12]

### 3.5 Τα κλειδιά

Για να μπορέσουν οι νότες να γραφούν σωστά στο πεντάγραμμο πρέπει απαραίτητα στην αρχή κάθε πενταγράμμου να υπάρχει το σημείο που ονομάζεται κλειδί, Το κλειδί του σολ της δεύτερης γραμμής (κλειδί του βιολιού) είναι το κλειδί το οποίο χρησιμοποιείται από τα περισσότερα μουσικά όργανα, Αυτά έχουν σαν χαρακτηριστικό τους γνώρισμα το υψηλό τονικό ύψος (ψηλή φωνή). Γράφεται στη δεύτερη γραμμή του πενταγράμμου ονομάζοντας τη νότα που περνάει πάνω από αυτή τη γραμμή σολ. Παίρνοντας λοιπόν σαν βάση τη νότα Σολ, καθορίζουμε το όνομα και για τις υπόλοιπες νότες έτσι προκύπτουν τα ακόλουθα (εικόνα 3.3).



Εικόνα 3.3 Οι νότες στο κλειδί του Σολ πάνω στο πεντάγραμμο[12]

### 3.6 Μουσική έκταση – Οκτάβα

Μουσική έκταση ονομάζουμε το σύνολο από νότες που παράγει ένα οποιοδήποτε μουσικό όργανο ή η ανθρώπινη φωνή, από τη χαμηλότερη έως την ψηλότερη νότα. Τη μουσική έκταση τη μετράμε σε οκτάβα. Οκτάβα ονομάζουμε το σύνολο οκτώ συνεχόμενων φθόγγων, όπου ο πρώτος και ο όγδοος φθόγγος έχουν το ίδιο όνομα αλλά διαφορετικό ύψος. Ο όγδοος φθόγγος είναι ταυτοχρόνως και

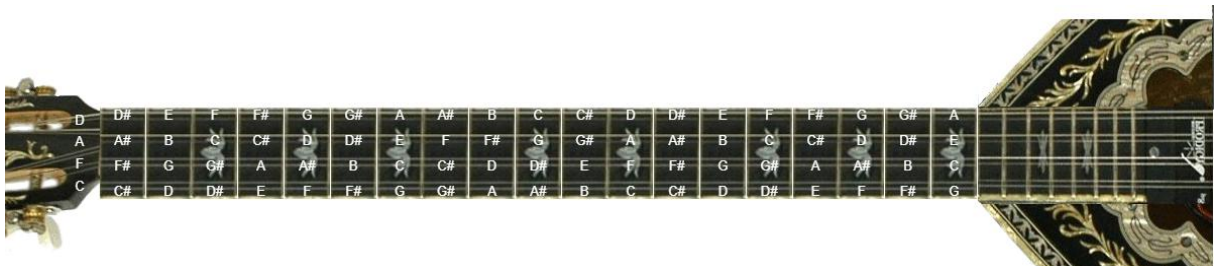
πρώτος της επόμενης οκτάβας (εικόνα 3.4). Όσο περισσότερες οκτάβες έχει ένα μουσικό όργανο τόσο μεγαλύτερη μουσική έκταση έχει.



Εικόνα 3.4 Δυο οκτάβες στο κλειδί του Σολ της δεύτερης γραμμής[12]

Από το παραπάνω παράδειγμα προκύπτει ότι το άθροισμα 2 οκτάβων είναι 15 νότες και όχι 16. Αυτό συμβαίνει επειδή η όγδοη νότα της πρώτης οκτάβας υπολογίζεται μόνο μία φορά. Άρα για να υπολογίσουμε σωστά τον αριθμό των φθόγγων ενός μουσικού οργάνου (την έκταση του), πολλαπλασιάζουμε τον αριθμό των οκτάβων που έχει με 7 και προσθέτουμε το ένα

Στο μπουζούκι λοιπόν έχουμε ξεκινώντας από τη χαμηλότερη ντο, στην πάνω χορδή, και φτάνοντας στην υψηλότερη φα στην κάτω χορδή (καντίνι), ηχούμε σχεδόν τέσσερις οκτάβες όπως και στο βιολί (εικόνα 3.5).



Εικόνα 3.5 Οι νότες στην ταστιέρα του μπουζουκιού[13]

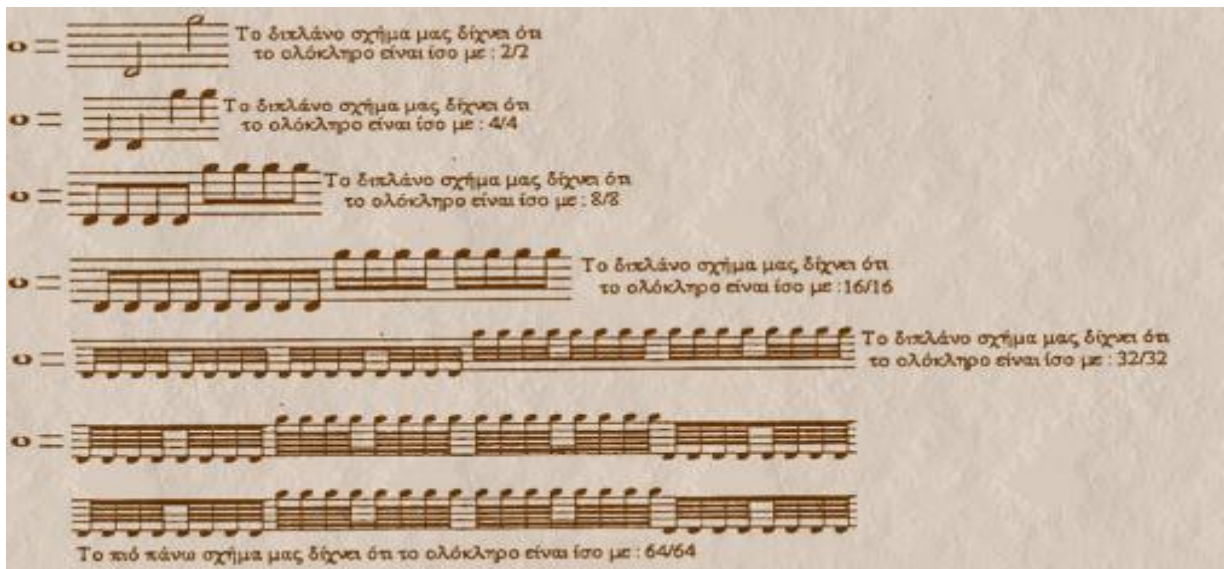
### 3.7 Μουσικές αξίες

Οι μουσικές αξίες προκύπτουν από τα διάφορα σχήματα τα οποία είναι υπεύθυνα για τη χρονική διάρκεια της νότας. Η κάθε μία λοιπόν από τις επτά νότες μπορεί να έχει κάποια από τις αξίες που παρατίθενται στο πίνακα 3.1.

Πίνακας 3.1: Οι μουσικές αξίες και ο συμβολισμός τους

Σύμβολο	Αξία
	ολόκληρου
	μισού
	τετάρτου
	ογδού
	δεκάτου έκτου
	τριακοστού δευτέρου
	εξηκοστού τετάρτου

Το ολόκληρο είναι η μεγαλύτερη χρονική αξία η οποία υπολογίζεται συνήθως μετρώντας 4 χρόνους, δηλαδή χωρίζεται σε 4 ισόποσες χρονικές υποδιαιρέσεις και αντίστοιχα κινούνται όλες οι μικρότερες χρονικές αξίες, όπως φαίνεται και παρακάτω (εικόνα 3.6).



Εικόνα 3.6 Αναλογία του ολόκληρου με τις υπόλοιπες αξίες [12]

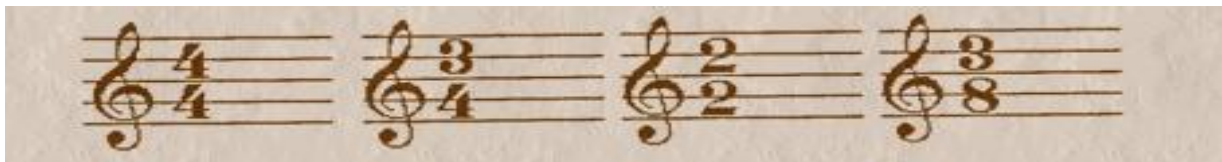
### 3.8 Μέτρο

Μέτρο ονομάζουμε το χώρισμα του πενταγράμμου με κάθετες γραμμές σε μικρότερα μέρη ίσης χρονικής διάρκειας το καθένα. Η κάθε γραμμή του χωρίσματος ονομάζεται διαστολή. Όταν ένα μουσικό κομμάτι τελειώσει τότε χρησιμοποιούμε διπλή διαστολή (εικόνα 3.7).



Εικόνα 3.7 Η έννοια του μέτρου[12]

Ο κλασματικός αριθμός ή το σύμβολο το οποίο μπαίνει αμέσως μετά το κλειδί αποκλειστικά και μόνο στο πρώτο μέρος του κομματιού, ορίζει τη χρονική διάρκεια του πρώτου μέτρου που την παίρνουμε σαν βάση και καθορίζουμε τη χρονική διάρκεια και των υπόλοιπων μέτρων, έως, και το τέλος (εικόνα 3.8).



Εικόνα 3.8 Η χρονική διάρκεια του μετρίου[12]

Έχουμε απλά μέτρα όπως τα 2/4, 3/4, 4/4, έχουμε σύνθετα μέτρα όπως τα 6/4, 6/8, 12/8 και τέλος έχουμε και τα μεικτά μέτρα όπως τα 5/4, 5/8, 7/8 και 9/8.

### 3.9 Οπλισμός

Οπλισμό ονομάζουμε το σύνολο των διέσεων ή των υφέσεων που μπαίνει αμέσως μετά το κλειδί του σολ, και φανερώνει ποιες από τις νότες του κομματιού θα παίζονται με δίεση ή ύφεση χωρίς αυτές οι αλλοιώσεις να αναγράφονται πάνω στις νότες. Η σειρά που τα γράφουμε είναι προκαθορισμένη και είναι η εξής (εικόνα 3.9).



Εικόνα 3.9 Η σειρά του οπλισμού πάνω στο πεντάγραμμο[12]

Εδώ θα κλείσουμε το κεφάλαιο με την μουσική θεωρία Φυσικά και δεν μπορεί να χωρέσει σε ένα κεφάλαιο ο τεράστιος όγκος της μουσικής θεωρίας όμως κάναμε μια μικρή περίληψη των πιο σημαντικών στοιχείων.

### 3.10 Επίλογος

Σε αυτό το κεφάλαιο παρουσιάσαμε την μουσική θεωρία συνοπτικά Αρχίσαμε από το πεντάγραμμο και τους φθόγγους και στην συνέχεια μιλήσαμε για μουσικές αξίες, μέτρα, χρόνους και οπλισμούς Ότι θα χρειαστεί για να κατανοήσουμε καλύτερα την παρτιτούρα.

## Κεφάλαιο 4<sup>ο</sup> : Ο Όχος στην Φυσική και στην Πληροφορική

### 4.1 Εισαγωγή

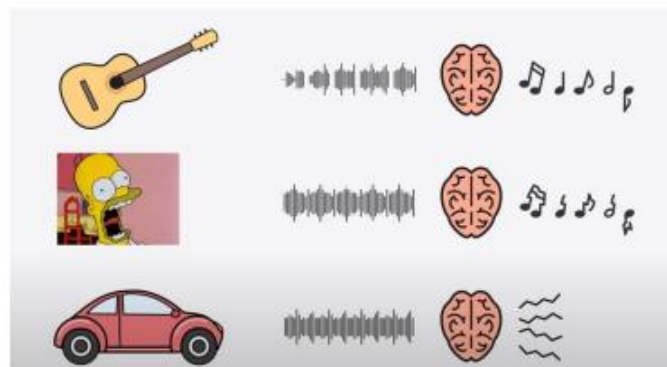
Στο τέταρτο κεφάλαιο θα μιλήσουμε σύντομα για τον ήχο. Θα δώσουμε ορισμό και βασικά στοιχεία για τα χαρακτηριστικά του. Κατόπιν θα αναφερθούμε στην ακουστική και τον τρόπο της ανθρώπινης ακοής. Αμέσως μετά θα επικεντρωθούμε στον ψηφιακό ήχο και θα αναφερθούμε στη μετατροπή, τη δειγματοληψία, την κβάντιση, το μέγεθος, τον ρυθμό μετάδοσης, τη συμπίεση και τέλος τη μεταφορά την αποθήκευση και αναπαραγωγή των αρχείων του ήχου.

### 4.2 Τι είναι ήχος

Όχος είναι ο τρόπος που μεταφράζονται στο μυαλό μας τα ηχητικά κύματα τα οποία προέρχονται από διάφορα αντικείμενα που δονούνται (εικόνα 4.1).

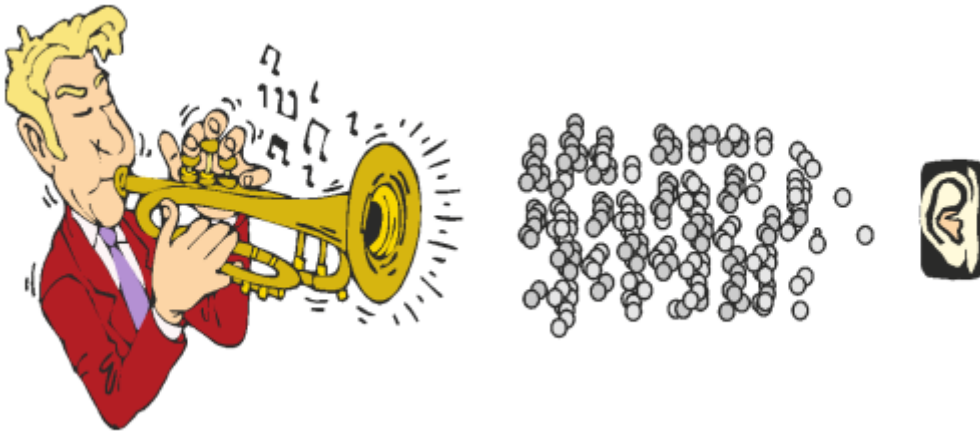


Εικόνα 4.1 Όχος [14]



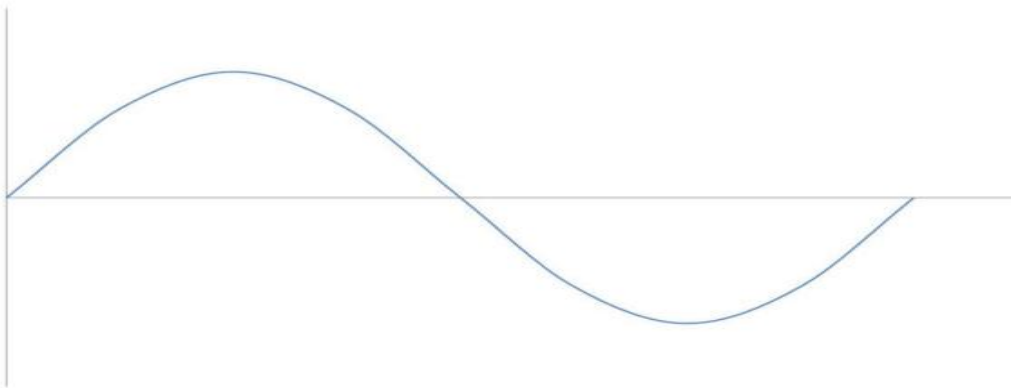
Εικόνα 4.2 Αντικείμενα που παράγουν ηχητικά κύματα [14]

Ο ήχος είναι ένα κύμα που μεταδίδεται σε κάποιο μέσο όπως ο αέρας, το νερό κτλ, και δημιουργείται από δονούμενα αντικείμενα, όπως για παράδειγμα οι φωνητικές χορδές των ανθρώπων ή κάποιο μουσικό όργανο (εικόνα 4.2). Παράγεται από μια ταχεία μεταβολή στη μέση πυκνότητα, ή την πίεση των μορίων του αέρα, πάνω και κάτω από την τρέχουσα ατμοσφαιρική πίεση. Αυτές οι μικρές αλλαγές στην ατμοσφαιρική πίεση (εικόνα 4.3) αναφέρονται ως πίεση ήχου (sound pressure) και οι διακυμάνσεις της πίεσης ως ηχητικά κύματα (sound waves).[14]



Εικόνα 4.3 Πνευστό όργανο παράγει ήχο [14]

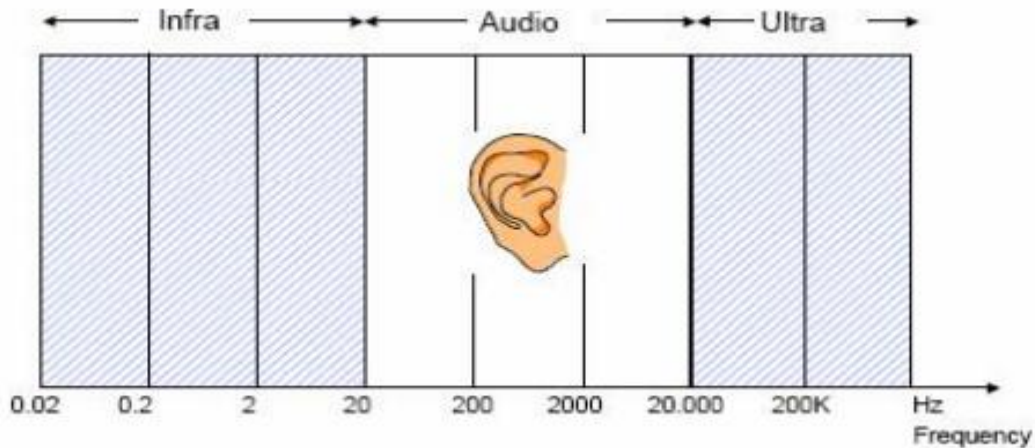
Όταν η πίεση των μορίων του αέρα φτάσει στο τύμπανο του αυτιού ή σε κάποια συσκευή όπως το μικρόφωνο, τότε το τύμπανο πάλλεται μετατρέποντας τη μηχανική ενέργεια της κίνησης σε ηλεκτρική ενέργεια που μεταδίδεται στον εγκέφαλο για περαιτέρω επεξεργασία. Το ηχητικό σήμα αναπαρίσταται γραφικά ως μια αρμονική ημιτονοειδής καμπύλη που ονομάζεται κυματομορφή (wave form) και παριστάνει τη μετατόπιση των μορίων του αέρα, που δημιουργείται λόγω της πίεσης, σε συνάρτηση με το χρόνο (εικόνα 4.4).



Εικόνα 4.4 Γραφική παράσταση ήχου [14]

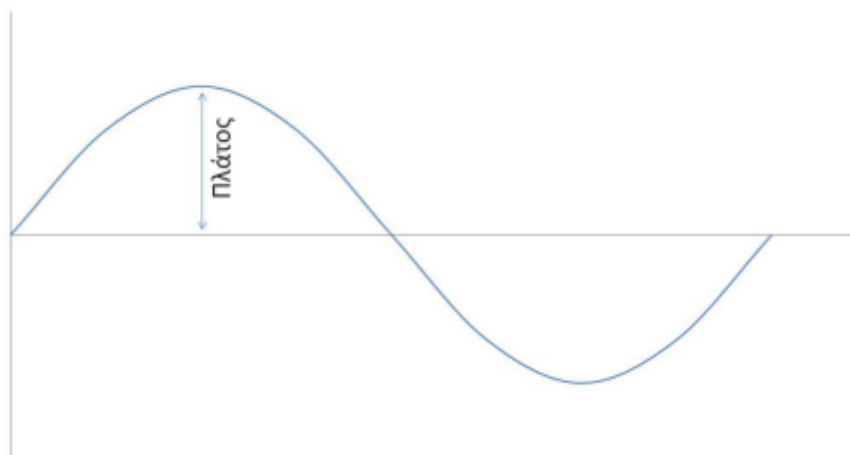
### 4.3 Χαρακτηριστικά του ήχου

Η συχνότητα (frequency), με βάση την επιστήμη της Φυσικής, είναι ο αριθμός των κυμάτων που δημιουργούνται από το σώμα που δονείται ανά δευτερόλεπτο και ορίζεται σε Hertz (Hz), δηλαδή κύκλους σήματος ανά δευτερόλεπτο. Το ανθρώπινο αυτί μπορεί να αντιληφθεί ήχους από 20Hz ως περίπου 20KHz (εικόνα 4.5). {Κωτσάκη βασικά στοιχεία ακουστικής}



Εικόνα 4.5 Συχνότητες που ακούει το ανθρώπινο αυτί [15]

Η ένταση (amplitude) καθορίζει πόσο δυνατά ακούγεται ένας ήχος. Η απόσταση του μέγιστου σημείου μιας κυματομορφής ήχου από τη μέση τιμή του, ονομάζεται πλάτος (εικόνα 4.6). Μεγαλύτερο πλάτος σημαίνει μεγαλύτερη ένταση ήχου. Ως φυσικό μέγεθος, η ένταση αφορά τη δύναμη του ηχητικού κύματος και εκφράζεται σε  $\text{Watt/m}^2$ , αλλά για λόγους απλότητας και συσχέτισης με την ανθρώπινη αντίληψη, χρησιμοποιείται η μονάδα decibel (db).



Εικόνα 4.6 Πλάτος κυματομορφής ηχητικού κύματος [14]

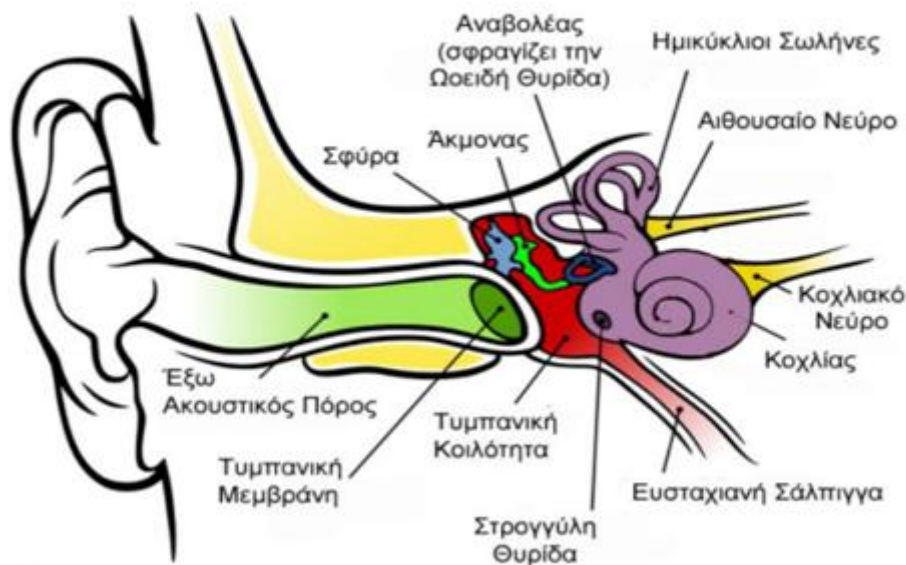
Η ακουσικότητα (loudness) είναι η υποκειμενική ένταση του ήχου, που εξαρτάται από την ακουστική ικανότητα του ακροατή, τη διάρκεια του ήχου και τη συχνότητα. Το ανθρώπινο αυτί παρουσιάζει διαφορετική ευαισθησία στις διάφορες συχνότητες, με μέγιστη ευαισθησία στα 3-4 KHz. Για παράδειγμα, η ελάχιστη ισχύς μια πηγής ήχου που προκαλεί ένα ήχο μόλις αντιληπτό από το ανθρώπινο αυτί στα 30 Hz, είναι κάπου ένα εκατομμύριο φορές μεγαλύτερη, από ότι στα 4 KHz [15]

Το ύψος (pitch) είναι ένα υποκειμενικό χαρακτηριστικό που μας επιτρέπει να διακρίνουμε τους ήχους σε οξείς (μεγάλης συχνότητας), μέσους (μεσαίας συχνότητας) και σε βαρείς (μικρής συχνότητας). Το

ύψος αφορά τον τρόπο με τον οποίο αντιλαμβανόμαστε τη συχνότητα. Οι βαρείς ή βαθείς ήχοι είναι τα μπάσα (bass), και παράγονται από όργανα όπως το τύμπανο, ενώ οι οξείς ήχοι είναι τα πρίμα (treble) και παράγονται από το φλάουτο και το βιολί. Με διαφορετικά λόγια, αν και δύο όργανα ή υλικά μπορούν να παράγουν ήχους παρόμοιας συχνότητας, ακούγονται διαφορετικά, δηλαδή έχουν διαφορετικό ύψος.[14]

#### 4.4 Η αντίληψη του ήχου από τον άνθρωπο

Ο άνθρωπος αντιλαμβάνεται τον ήχο μέσω της ακοής του δηλαδή με μία από τις ανθρώπινες αισθήσεις. Επομένως, επεξεργάζεται τους ήχους που λαμβάνει από το περιβάλλον του λόγω του οργάνου ακοής που είναι το ανθρώπινο αυτί. Το ανθρώπινο αυτί αποτελείται από τρία μέρη: το εξωτερικό αυτί, το μέσο αυτί και το εσωτερικό αυτί. Το εσωτερικό αυτί αποτελείται από δύο μέρη (εικόνα 4.7), το αιθουσαίο νεύρο για την ισορροπία και τον κοχλία για την ακοή. Το εξωτερικό και το μέσο αυτί μεταφέρουν τον ήχο στον κοχλία, ο οποίος επεξεργάζεται τα ηχητικά σήματα και διαχωρίζει τους ήχους ως προς τη συχνότητά τους προτού μετατραπούν σε νευρικό κώδικα στις ίνες του ακουστικού νεύρου[16]. Η ανθρώπινη ακοή κυμαίνεται από τα 20 Hz μέχρι τα 20.000 Hz. Οι ήχοι που γίνονται αντιληπτοί από το ανθρώπινο αυτί ονομάζονται ακουστοί ήχοι. Αντίθετα, οι ήχοι συχνοτήτων κάτω από τα 20 Hz ονομάζονται υπόηχοι και πάνω από τα 20.000 Hz υπέρηχοι. Τέλος, αξίζει να αναφερθεί ότι το κατώφλι πόνου της ανθρώπινης ακοής ξεκινάει από ήχους που ξεπερνούν τα 120 dB.[17]



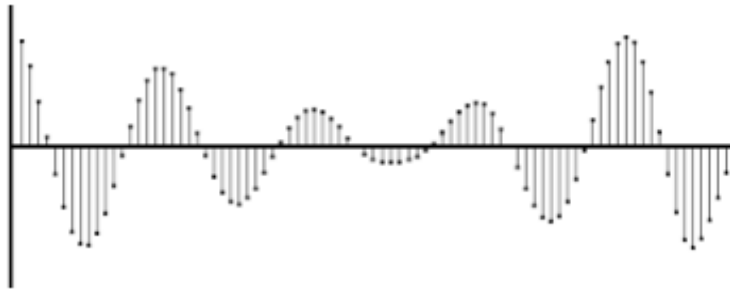
Εικόνα 4.7 Διάγραμμα του ανθρώπινου αυτιού [16]

#### 4.5 Ψηφιακός Ήχος

Το ηχητικό σήμα είναι αναλογικό, δηλαδή σε κάθε χρονική στιγμή υπάρχει διαφορετική ένταση. Για να μπορέσουμε όμως να το επεξεργαστούμε ή να το αναπαράγουμε με τους υπολογιστές, πρέπει να περάσει από τη φάση της ψηφιοποίησης, ώστε το αναλογικό σήμα να μετατραπεί σε ψηφιακό. Η ψηφιοποίηση (digitization) μετατρέπει το αναλογικό σήμα σε ψηφιακή μορφή και περιλαμβάνει τη φάση της δειγματοληψίας (sampling) και τη φάση της κβάντισης (quantization).

### 4.5.1 Δειγματοληψία

Κατά τη δειγματοληψία λαμβάνονται δείγματα (samples) από το αναλογικό σήμα, τα οποία θα αναπαρασταθούν τελικά με ψηφιακές τιμές. Ο αριθμός λήψης δειγμάτων (εικόνα 4.8) ανά δευτερόλεπτο ονομάζεται ρυθμός ή συχνότητα δειγματοληψίας (sampling rate) και μετριέται σε ΗΖ (δείγματα ανά δευτερόλεπτο).

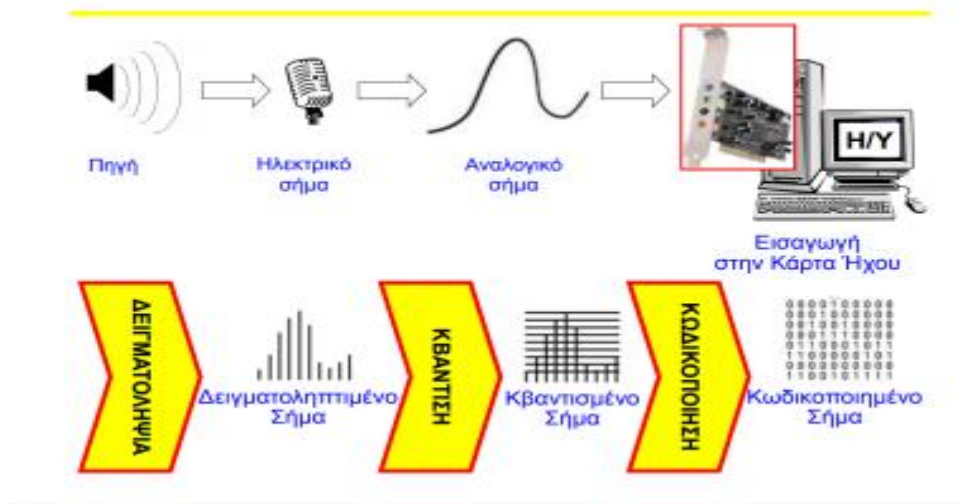


Εικόνα 4.8 Δειγματοληψία [14]

Αυτό το οποίο πρέπει να λάβουμε υπόψη για σωστή δειγματοληψία είναι το Θεώρημα Nyquist που ορίζει, ότι η μέγιστη συχνότητα δειγματοληψίας πρέπει να είναι τουλάχιστον διπλάσια από τη μέγιστη συχνότητα του αναλογικού σήματος στο οποίο γίνεται δειγματοληψία. Για παράδειγμα, αν η μέγιστη συχνότητα του αναλογικού συστήματος είναι 8 KHz, η συχνότητα δειγματοληψίας πρέπει να είναι τουλάχιστον 16 KHz.

### 4.5.2 Κβάντιση

Στη διαδικασία της κβάντισης κάθε δείγμα του ήχου θα αναπαρασταθεί τελικά ως ένας δυαδικός αριθμός. Τα δείγματα μεταβάλλονται ώστε να λαμβάνουν τιμές από ένα προκαθορισμένο σύνολο διακριτών τιμών. Για παράδειγμα, αν επιβάλλεται να χρησιμοποιήσουμε μόνο 4 δυαδικούς αριθμούς, τότε όλα τα δείγματα θα αναπαρασταθούν με χρήση δεκαέξι σταθμών έντασης. Εδώ πρέπει να αναφερθεί η διαφορά μεταξύ της δειγματοληψίας και της κβάντισης. Η δειγματοληψία έχει να κάνει με τη συχνότητα του σήματος (aliasing), ενώ η κβάντιση έχει να κάνει με το πλάτος του σήματος (θόρυβος κβάντισης) (εικόνα 4.9).



Εικόνα 4.9 Ψηφιοποίηση [14]

### 4.5.3 Μέγεθος αρχείων

Το μέγεθος ενός ασυμπίεστου ψηφιακού αρχείου ήχου που προκύπτει μετά την ψηφιοποίηση μιας κυματομορφής εξαρτάται κατά κύριο λόγο από τον ρυθμό δειγματοληψίας και το μέγεθος (ανάλυση) δείγματος. Ο αριθμός καναλιών και η χρονική διάρκεια είναι επίσης, παράγοντες που καθορίζουν το τελικό μέγεθος του ασυμπίεστου ήχου. Για να γίνει πιο σαφές, θα δούμε το παρακάτω παράδειγμα. Έχοντας στερεοφωνικό αρχείο ήχου (2 κανάλια), διάρκειας 1 λεπτού (60 sec), με ρυθμό δειγματοληψίας 22050 Hz και μέγεθος δείγματος 16 bit, ο τύπος μας δίνει το αποτέλεσμα:  $2 \cdot 22050 \cdot 16 \cdot 60 = 42.336.000$  bit μέγεθος αρχείου. Διαιρώντας το αποτέλεσμα διά 8, έχουμε μέγεθος ήχου 5.292.000 bytes. Περαιτέρω διαίρεση δια 1.024, μετατρέπει το μέγεθος σε περίπου 5.168 KB και συνεχίζοντας την διαίρεση του 5.168 με το 1.024, μετατρέπουμε το μέγεθος σε MB και ισούται με περίπου 5 MB.[14]

### 4.5.4 Ρυθμός μετάδοσης

Ο ρυθμός μετάδοσης ή μεταφοράς (bit rate) είναι ο ρυθμός επεξεργασίας ή μεταφοράς πληροφορίας από ένα μέσο, και εκφράζεται σε bits/sec ενώ συντομογραφικά αναφέρεται ως bps. Ο αριθμός αυτός είναι σημαντικός στον ήχο, όπου έχουμε συνεχής και δυναμική ροή πληροφορίας, διότι, αν ο ρυθμός μεταφοράς είναι μικρός, δεν μπορεί να γίνει σωστή αναπαραγωγή της πληροφορίας. Στο παραπάνω παράδειγμα είχαμε ήχο διάρκειας 1 λεπτού και μεγέθους 42.336.000 bit. Για να μπορέσει να αναπαραχθεί σωστά ο ήχος, π.χ. μέσω διαδικτύου, απαιτείται ρυθμός μεταφοράς ίσος με  $42.336.000 \text{ bit} / 60 \text{ sec} = 705.600 \text{ bps} = 705,6 \text{ Kbps} \approx 0,7 \text{ Mbps}$ . [14]

### 4.5.5 Συμπίεση

Η συμπίεση ήχου είναι μια τεχνική μέθοδος ή διαδικασία που χρησιμοποιείται για τη μείωση του μεγέθους των ψηφιακών αρχείων, διατηρώντας παράλληλα μια αποδεκτή ποιότητα για τα αρχεία ήχου. Επιπλέον, η συμπίεση χρησιμοποιεί διάφορους αλγόριθμους για την εξάλειψη περιττών ή λιγότερο σημαντικών δεδομένων από ένα ηχητικό σήμα, μειώνοντας το μέγεθός του. Βασικά, υπάρχουν δύο κύριοι τύποι συμπίεσης ήχου και αυτοί είναι. Χωρίς απώλειες γνωστή σαν μη απωλεστική (lossless), και με απώλειες γνωστή σαν απωλεστική (lossy). Από τη μία πλευρά, η μη

απωλεστική συμπίεση είναι ο τύπος που διατηρεί όλα τα αρχικά δεδομένα, γεγονός που επιτρέπει την τέλεια αποκατάσταση όταν το αρχείο αποσυμπιέζεται, καθιστώντας το σκόπιμο κατάλληλο για αρχειοθέτηση. Από την άλλη πλευρά, η συμπίεση με απώλειες είναι μια διαδικασία που απορρίπτει επιλεκτικά ορισμένες λιγότερο βασικές πληροφορίες ήχου, με αποτέλεσμα μικρότερα μεγέθη αρχείων. Ωστόσο, αυτό μειώνει ελαφρώς την ποιότητα του ήχου, η οποία είναι σε σύγκριση με τη συμπίεση ήχου χωρίς απώλειες. Την παραπάνω τεχνική συμπίεσης την πραγματοποιούν οι κωδικοποιητές, όμως υπάρχουν και άλλοι τρόποι να λάβουμε πιο μικρά αρχεία, όπως με αλλαγή ρυθμού δειγματοληψίας και μεγέθους δείγματος. Εν τω μεταξύ, οι πιο δημοφιλείς κωδικοποιητές, όπως ο MP3 και ο WMA (Windows Media Audio) είναι απωλεστικοί, και έχουν γίνει η τυπική απαίτηση για την αποθήκευση και τη ροή περιεχομένου ήχου, λόγω της δίκαιης ισορροπίας μεταξύ μεγέθους αρχείου και ποιότητας ήχου.[18]

#### 4.5.6 Αποθήκευση και μεταφορά

Η αποθήκευση και η μεταφορά ψηφιακού ήχου είναι ένα ενδιαφέρον και πολυδιάστατο θέμα. Όταν μιλάμε για την μεταφορά του ψηφιακού ήχου συνήθως εννοούμε την μετάδοση του. Η ψηφιακή μετάδοση έχει επικρατήσει ολοκληρωτικά στην μεταφορά του ήχου έναντι της αναλογικής μετάδοσης, για τους παρακάτω λόγους.

- Παράγει λιγότερα σφάλματα σε σχέση με την αναλογική.
- Επειδή τα δεδομένα που μεταδίδονται είναι δυαδικά (0 ή 1) είναι ευκολότερο να εντοπίσουμε και να διορθώσουμε λάθη.
- Επιτρέπει υψηλότερους ρυθμούς μετάδοσης, οι οπτικές ίνες π.χ., είναι σχεδιασμένες για ψηφιακή μετάδοση.
- Είναι πιο αποτελεσματική. Είναι δυνατόν να στείλουμε περισσότερα δεδομένα μέσω ενός κυκλώματος ψηφιακής, παρά αναλογικής μετάδοσης.
- Είναι πιο ασφαλής γιατί τα ψηφιακά δεδομένα κρυπτογραφούνται ευκολότερα.

Η μεταφορά ψηφιακού ήχου περιλαμβάνει τη διαδικασία της αποστολής των ψηφιακών δεδομένων μέσω διαφόρων μέσων και πρωτοκόλλων. Μερικά κοινά μέσα μεταφοράς είναι.

- Δίκτυα Υπολογιστών: Η μετάδοση ήχου μέσω του διαδικτύου (π.χ., Spotify, YouTube) ή μέσω τοπικών δικτύων.
- USB & άλλες συσκευές αποθήκευσης: Αρχεία ήχου μεταφέρονται μέσω USB stick, σκληρών δίσκων ή άλλων συσκευών αποθήκευσης.
- Bluetooth: Για ασύρματη μετάδοση ήχου σε ακουστικά ή ηχεία.

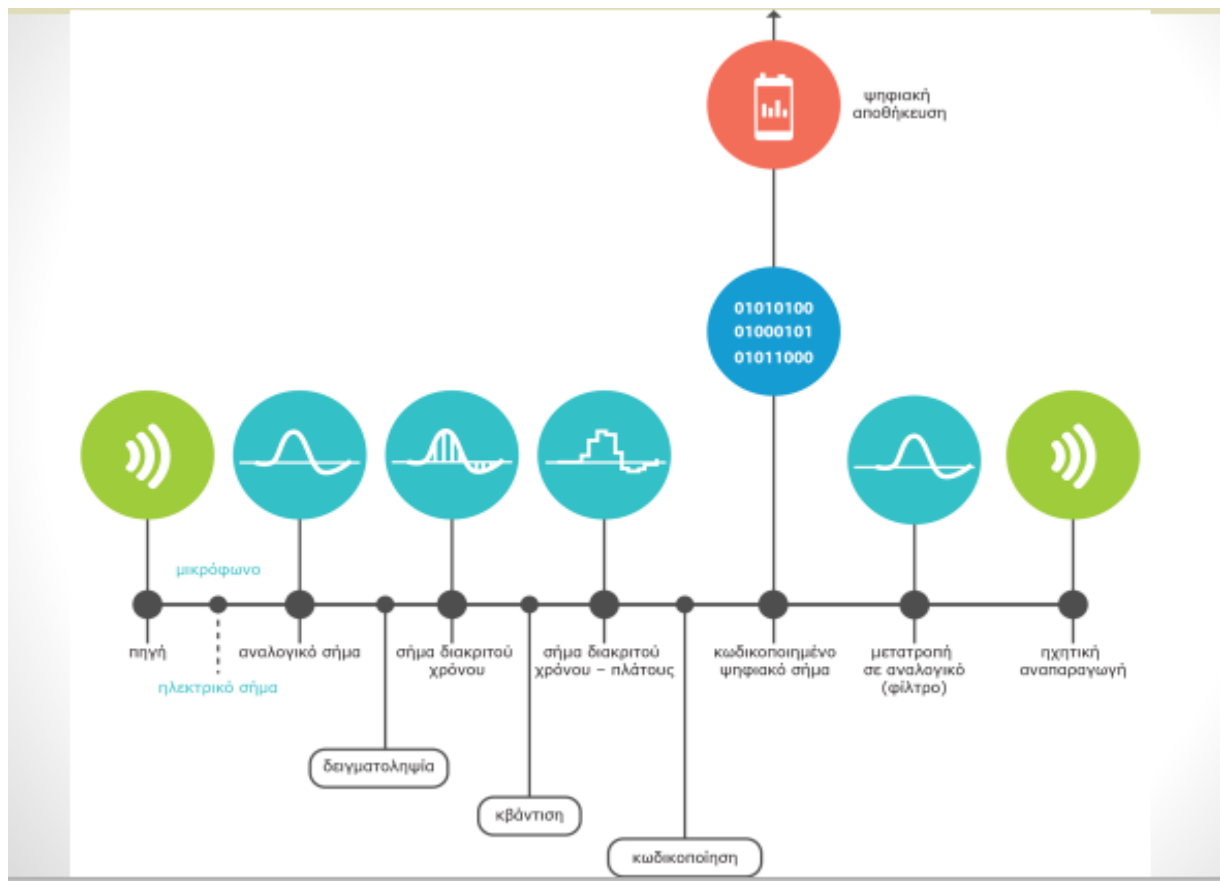
Για την αποθήκευση τώρα και αφού έχει προηγηθεί η δειγματοληψία και ο κβαντισμός χρησιμοποιούμε κάποιες μορφές γνωστών αρχείων οι οποίες διαφέρουν ανάλογα με την ποιότητα και τη συμπίεση του ήχου. Μερικές από τις πιο κοινές είναι.

- WAV (Waveform Audio File): Αναλλοίωτο (uncompressed) αρχείο ήχου, που συνήθως χρησιμοποιείται για υψηλή ποιότητα ήχου, αλλά έχει μεγάλο μέγεθος. Χρησιμοποιείται ευρέως στα συστήματα Windows.
- MP3 (MPEG Audio Layer 3): Συμπιεσμένο αρχείο ήχου, που χρησιμοποιεί αλγόριθμους για να μειώσει το μέγεθος του αρχείου χωρίς να χαθεί η ποιότητα. Χρησιμοποιείται ευρέως για μουσική.
- FLAC (Free Lossless Audio Codec): Συμπιεσμένο αλλά χωρίς απώλειες αρχείο ήχου. Διατηρεί την ποιότητα του αρχικού ήχου αλλά μειώνει το μέγεθος του αρχείου.

- AAC (Advanced Audio Codec): Παρόμοιο με το MP3 αλλά προσφέρει καλύτερη ποιότητα ήχου για το ίδιο μέγεθος αρχείου, χρησιμοποιείται ευρέως για ροή (streaming) ήχου.[13]

#### 4.5.7 Συμπέρασμα

Η αποθήκευση και η μεταφορά ψηφιακού ήχου περιλαμβάνει μια σειρά από τεχνικές, από την ψηφιοποίηση του αναλογικού ήχου μέχρι την επιλογή κατάλληλων μορφών και κωδικοποιητών για τη βέλτιστη μεταφορά και αποθήκευση (εικόνα 4.10). Τα δεδομένα του ήχου, όταν μεταφέρονται και αποθηκεύονται σωστά, μπορούν να αναπαραχθούν με υψηλή ποιότητα σε διάφορες συσκευές.



Εικόνα 4.10 Συνοψίζοντας την ψηφιοποίηση [19]

#### 4.6 Επίλογος

Σε αυτό το κεφάλαιο παρουσιάσαμε τον ήχο και τα χαρακτηριστικά του, καθώς και την αντίληψη του από τον άνθρωπο. Μετά επικεντρωθήκαμε στον ψηφιακό ήχο στον δειγματισμό, στην κβαντίση, στην μεταφορά και την αποθήκευση του, στην σύγχρονη εποχή.

## Κεφάλαιο 5<sup>ο</sup> : Η βιβλιοθήκη MIRtoolbox

### 5.1 Εισαγωγή

Η MIR toolbox είναι μια βιβλιοθήκη (συλλογή εργαλείων) για την επεξεργασία σήματος στη MATLAB που επιτρέπει την έξυπνη διαχείριση ηχητικών αρχείων παρέχοντας εργαλεία στους χρήστες για ανάλυση και επεξεργασία σημάτων με ευκολία, ακρίβεια και ταχύτητα. Η βιβλιοθήκη αυτή είναι ελεύθερο λογισμικό και χρησιμοποιείται ευρέως στην ακαδημαϊκή έρευνα Αρχικά αναπτύχθηκε στο πλαίσιο ενός ευρωπαϊκού έργου "Tuning the brain for Music" και σαν συντονιστή είχε το "Cognitive Brain Research Unit of the Department of Helsinki. Η βιβλιοθήκη υποστηρίζει μουσική ανάλυση, ανάκτηση μουσικών πληροφοριών και εξαγωγή μουσικών χαρακτηριστικών χρησιμοποιώντας τη MATLAB. Στη συνέχεια θα παρουσιαστεί μια μικρή αναφορά στη MATLAB και κατόπιν θα παρουσιαστούν οι βασικές συναρτήσεις της MIR toolbox που θα χρησιμοποιηθούν παρακάτω.

### 5.2 Τι είναι το MATLAB

Το λογισμικό MATLAB, που παίρνει το όνομά του από τις λέξεις MATrix LABoratory, είναι ένα σύγχρονο ολοκληρωμένο μαθηματικό πακέτο που χρησιμοποιείται εκτενώς στα πανεπιστήμια και στη βιομηχανία. Είναι ένα διαδραστικό (interactive) πρόγραμμα για αριθμητικούς υπολογισμούς και για κατασκευή γραφημάτων, αλλά παρέχει επίσης και τη δυνατότητα προγραμματισμού, κάτι που το καθιστά ένα χρησιμότερο εργαλείο για όλους όσους ασχολούνται με τις θετικές επιστήμες (και όχι μόνο). Το MATLAB στις αρχικές του εκδοχές δεν έκανε συμβολικούς υπολογισμούς. Στις νεότερες εκδοχές του, το πακέτο περιλαμβάνει εργαλείοι που επιτρέπουν και συμβολικούς υπολογισμούς. Όπως υποδηλώνεται και από το όνομα του, το MATLAB είναι ειδικά σχεδιασμένο για υπολογισμούς με πίνακες, όπως η επίλυση γραμμικών συστημάτων, η εύρεση ιδιοτιμών και ιδιοδιανυσμάτων, η αντιστροφή τετραγωνικού πίνακα κλπ. Επιπλέον, το πακέτο αυτό είναι εφοδιασμένο με πολλές επιλογές για γραφικά (δηλ. την κατασκευή γραφικών παραστάσεων) και προγράμματα γραμμένα στη δική του γλώσσα προγραμματισμού για την επίλυση άλλων προβλημάτων όπως η εύρεση των ριζών μη γραμμικής εξίσωσης, η επίλυση μη γραμμικών συστημάτων, η επίλυση προβλημάτων αρχικών τιμών με συνήθεις διαφορικές εξισώσεις κ.α. Η γλώσσα προγραμματισμού του MATLAB δίνει την ευχέρεια στον χρήστη να το επεκτείνει με δικά του προγράμματα.

Το MATLAB είναι σχεδιασμένο για την αριθμητική επίλυση προβλημάτων σε αριθμητική πεπερασμένη ακρίβειας (finite-precision arithmetic). Με άλλα λόγια, δεν βρίσκει την ακριβή λύση αλλά μια προσεγγιστική λύση ενός προβλήματος. Αυτή είναι και η βασική του διαφορά από τα συστήματα συμβολικών υπολογισμών όπως το Maple και το Mathematica. Το πακέτο είναι εφοδιασμένο με ένα εκτενές σύστημα βοήθειας, όπου κάθε εντολή επεξηγείται αναλυτικά και με αντιπροσωπευτικά παραδείγματα. Η πιο σημαντική εντολή της MATLAB είναι η help (βοήθεια)! Μια πληθώρα πληροφοριών, τόσο για αρχάριους όσο και προχωρημένους, είναι διαθέσιμη στην επίσημη ιστοσελίδα της MATLAB.[20]

### 5.3 MIR toolbox και βασικές συναρτήσεις

Η βιβλιοθήκη MIR toolbox περιέχει μια σειρά λειτουργιών που εισάγονται με τη χρήση συναρτήσεων και κατάλληλη παραμετροποίηση τους, για να γίνει η εισαγωγή, η επεξεργασία και η εξαγωγή επιθυμητών αποτελεσμάτων. Στην περίπτωση μας έχουμε τις παρακάτω απαιτήσεις.[21]

- Εισαγωγή και ανάγνωση αρχείων τύπου wav με μονοφωνικές ή στερεοφωνικές ηχογραφήσεις μουζουκίου. Η απαίτηση αυτή καλύπτεται από τη συνάρτηση **miraudio**.
- Ανάλυση των ηχητικών σημάτων που περιλαμβάνει χρονική κατάτμηση του αρχείου, φιλτράρισμα, φασματική ανάλυση, εκτίμηση έντασης ήχου, ισχυρών σημείων, κλπ. που πραγματοποιούνται με τις συναρτήσεις **mirframe**, **mirfilterbank**, **mirsegment**, **mirspectrum**, **mirenvelope**, **mirrms**, **mironsets**.
- Ανάλυση μελωδίας, η οποία μπορεί να γίνει με τη συνάρτηση **mirpitch**.
- Εξαγωγή συναρτήσεων στη MATLAB, η οποία μπορεί να γίνει με τη συνάρτηση **get**
- Εξαγωγή σε αρχείο MIDI, η οποία μπορεί να γίνει με τη συνάρτηση **mirmidi**.

Οι συναρτήσεις που χρησιμοποιούμε για την επεξεργασία των σημάτων εγγραφής παρουσιάζονται παρακάτω.

### 5.3.1 Συνάρτηση **miraudio**

Χρησιμοποιείται για να φορτώνει αρχεία ήχου, να εμφανίζει και να εκτελεί λειτουργίες στην κυματομορφή. Επιτρέπει στους χρήστες την διαχείριση διαφορετικών αρχείων όπως WAV, MP3, AIFF και άλλα και παρέχει ποικιλία από συναρτήσεις για την επεξεργασία ήχου όπως φίλτρα εφέ και ανάλυση σήματος. Η συνάρτηση **miraudio** αποτελεί την βασικότερη εντολή της βιβλιοθήκης MIR toolbox, καθώς πριν προχωρήσουμε περαιτέρω σε επεξεργασία, απαιτείται η φόρτωση του αρχείου ήχου. Μέσω της **miraudio**, ο χρήστης μπορεί να εξάγει τα ηχητικά δεδομένα, να αλλάζει την ανάλυση σε δείγματα (sampling rate) και να διαχειρίζεται τα κανάλια του ήχου (mono ή stereo).

### 5.3.2 Συνάρτηση **mirframe**

Χρησιμοποιείται για να χωρίσουμε σε υποπλαίσια το αρχείο με χρονικούς παραμέτρους. Κάνει διαχωρισμό σε διαδοχικά καρέ χρονικής αξίας 0,05sec εκτός και αν ορίσουμε διαφορετικό χρόνο.

### 5.3.3 Συνάρτηση **mirfilterbank**

Είναι συχνά ενδιαφέρον ο διαχωρισμός ενός ηχητικού σήματος σε μια σειρά ηχητικών σημάτων διαφορετικής συχνότητας για να μελετηθεί ξεχωριστά η κάθε μια από αυτές. Εκεί χρησιμοποιείται η παραπάνω συνάρτηση.

### 5.3.4 Συνάρτηση **mirsegment**

Χρησιμοποιείται για να ανιχνεύουμε τις μεταβολές ενός ηχητικού σήματος. Ουσιαστικά διαιρεί την ηχητική ροή σε αυτοτελή τμήματα (segment) βασισμένο σε χαρακτηριστικά όπως η ένταση, το φάσμα ή ακόμα η χροιά του ηχητικού σήματος. Είναι ιδανική για προχωρημένες ηχητικές εφαρμογές όπως η ανάλυση μουσικής δομής.

### 5.3.5 Συνάρτηση **mirspectrum**

Χρησιμοποιείται για να υπολογίσουμε το φάσμα συχνοτήτων ενός ηχητικού σήματος. Ουσιαστικά αναλύει το ηχητικό σήμα σε συνιστώσες συχνότητες παρέχοντας πληροφορία για την ενεργειακή κατανομή στο φάσμα. Αυτό είναι εξαιρετικά χρήσιμο για την ανάλυση και τον εντοπισμό τόνων ή διαστήματων και τον προσδιορισμό των συχνοτήτων που το συνθέτουν. Είναι πολύ χρήσιμη συνάρτηση γιατί αποκαλύπτει ποικιλία δομικών πληροφοριών του ήχου, που δεν είναι πάντα εμφανής στο ακουστικό πεδίο.

### 5.3.6 Συνάρτηση `mirenvelope`

Χρησιμοποιείται για να εξάγει την περιβάλλουσα (συνολικό εξωτερικό σχήμα του ηχητικού σήματος) του πλάτους ως προς το χρόνο, δηλαδή τη γραφική αναπαράσταση της μεταβολής της εντάσεως του ήχου ως προς το χρόνο. Έτσι ανιχνεύουμε τα χρονικά σημεία κορυφών και χαμηλών σε ένα ηχητικό σήμα.

### 5.3.7 Συνάρτηση `mirrms`

Χρησιμοποιείται για να υπολογίσουμε το μέτρο της μέσης ισχύος ή έντασης του ήχου, ενός ηχητικού σήματος (root mean square). Ουσιαστικά κατηγοριοποιεί σε πλαίσια, όπου εμείς ορίζουμε τον χρόνο, και υπολογίζει σε αυτά τη μέση ισχύς. Έτσι προκύπτουν μια σειρά από τιμές (κυματομορφή ισχύος στο χρόνο), και αυτό μας βοηθά να αναγνωρίζουμε τις αλλαγές στην ένταση του σήματος, στην εξέλιξη του χρόνου.

### 5.3.8 Συνάρτηση `mironsets`

Χρησιμοποιείται για την ανίχνευση σημείων ενάρξεως ήχων (onsets) ενός ηχητικού σήματος. Με την παραπάνω συνάρτηση και παίρνοντας τις κορυφές ισχύος στο χρόνο ουσιαστικά αναγνωρίζονται οι αλλαγές που συμβαίνουν σε ένα μουσικό κομμάτι, όπως χτυπήματα τυμπάνων ή αρχές από νότες. Έτσι παραμετροποιώντας με ρυθμίσεις αυτή την ευαισθησία μας δίνεται η δυνατότητα να εξετάσουμε τα onsets για μουσικούς ρυθμούς ή για νότες.

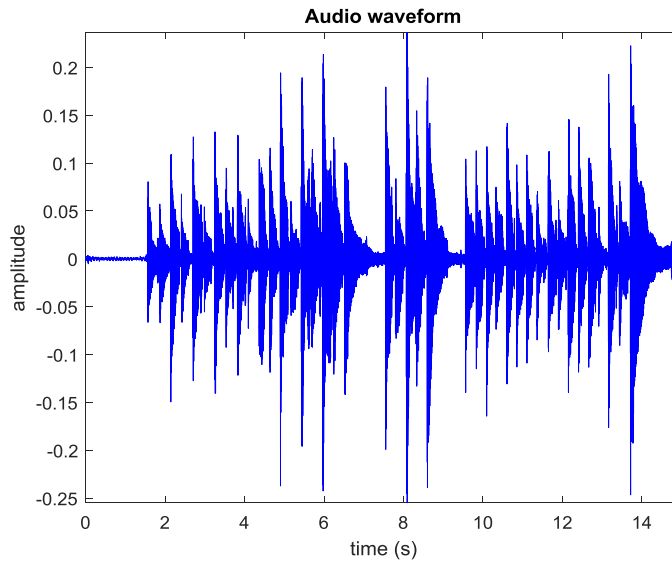
### 5.3.9 Συνάρτηση `mirpitch`

Χρησιμοποιείται για την ανάλυση της μελωδίας των ηχητικών σημάτων και τον προσδιορισμό των μελωδικών χαρακτηριστικών. Παρέχει πληροφορίες για τη βασική συχνότητα και έτσι μπορούμε να προσδιορίσουμε την τονικότητα σε ένα μουσικό αρχείο.

### 5.3.10 Συνάρτηση `get`

Μια από τις πλέον χρήσιμες συναρτήσεις του MIR toolbox είναι η `get`, η οποία μας επιτρέπει να εξάγουμε σε μορφή κατάλληλη για επεξεργασία με τη χρήση απλών συναρτήσεων και πινάκων της Matlab, τις τιμές των ηχητικών σημάτων που δημιουργεί το MIR toolbox στο δικό του format. Για παράδειγμα ο παρακάτω κώδικας μεταφέρει τα περιεχόμενα του αρχείου `testHARAMATA.wav` στη μεταβλητή `InMusic` η οποία έχει τη μορφοποίηση του MIR toolbox. Εν προκειμένω, η γραφική παράσταση του σήματος στο πεδίο του χρόνου φαίνεται στην εικόνα 5.1. Στη συνέχεια, με τη συνάρτηση `get` λαμβάνονται οι χρονικές τιμές και τα αντίστοιχα πλάτη του σήματος σε αυτές και αντιγράφονται στις μεταβλητές-πίνακες `Time` και `Amp` για περαιτέρω επεξεργασία. Σημειώνεται ότι οι διαφορετικές συναρτήσεις του MIR toolbox έχουν διαφορετική δομή ανάλογα με το είδος των τιμών που περιλαμβάνουν.

```
xfile = 'testHARAMATA.wav';
InMusic = miraudio(xfile)
aa = get(InMusic, 'Data');
tt = get(InMusic, 'Pos');
Amp = aa{1,1}{1,1};
Time = tt{1,1}{1,1};
```



Εικόνα 5.1 Κυματομορφή από την εγγραφή μπουζουκιού μετά τη χρήση της συνάρτησης `get`

### 5.3.11 Συνάρτηση `mirmidi`

Το `midi` είναι ακρωνύμιο για το “Musical Instrument Digital Interface” (ψηφιακή διεπαφή μουσικών οργάνων). Αποτελεί ένα πρωτόκολλο επικοινωνίας δεδομένων μεταξύ μουσικών οργάνων, κατασκευαστών μουσικού εξοπλισμού, υπολογιστών και λογισμικού. Σημαντικό να αναφερθεί ότι ένα `midi` αρχείο δεν είναι αρχείο ήχου, άλλα ένα πακέτο πληροφοριών που περιέχει οδηγίες, όπως για παράδειγμα το τέμπο και η τονικότητα του μουσικού αρχείου. Χρησιμοποιώντας λοιπόν την παραπάνω συνάρτηση μετατρέπουμε δεδομένα σχετικά με τα `pitch`, όπως τα πήραμε από την `mirpitch` σε αριθμοσειρές `midi`, κατάλληλες για την εξαγωγή παρτιτούρας όταν μπουν σε κάποιο κατάλληλο πρόγραμμα.

## 5.4 Συναρτήσεις Matlab

Στη μελέτη μας χρησιμοποιήσαμε συστηματικά τις παρακάτω συναρτήσεις της Matlab.

`AudioRecorder`, `recordblocking`, `getaudiodata`, `audiowrite`, `fft`, `findpeaks`, `xcorr` και `find`

### 5.4.1 Συνάρτηση `audiorecorder`

Η συνάρτηση `audiorecorder` χρησιμοποιείται για την εγγραφή ήχου. Με αυτή τη συνάρτηση μπορεί ο χρήστης να δημιουργήσει αντικείμενα εγγραφής ήχου, για την καταγραφή ηχητικών δεδομένων μέσω μικροφώνου ή άλλης πηγής ήχου που είναι συνδεδεμένη στον υπολογιστή. Έτσι μας παρέχει τη δυνατότητα καθορισμού παραμέτρων εγγραφής όπως, συχνότητα δειγματοληψίας, αριθμό καναλιών, έλεγχο στην διαδικασία εγγραφής μέσω πλήκτρων `start`, `stop` και `pause` και πρόσβαση στην εγγραφή για αποθήκευση και περαιτέρω ανάλυση.

### 5.4.2 Συνάρτηση `recordblocking`

Η συνάρτηση `recordblocking` χρησιμοποιείται για την εγγραφή ήχου σε πραγματικό χρόνο. Η ιδιαιτερότητα αυτής της συνάρτησης είναι ότι μπλοκάρει (σταματάει) την εκτέλεση του υπόλοιπου κώδικα μέχρι να ολοκληρωθεί η εγγραφή. Είναι απαραίτητη η χρήση της όταν πρέπει να διασφαλίσουμε ότι η εγγραφή θα ολοκληρωθεί πριν προχωρήσει το πρόγραμμά μας σε επόμενες ενέργειες. Χρησιμοποιείται και συνδυαστικά με την συνάρτηση `audiorecorder` και μπορεί να

δεχθεί και σαν είσοδο την συνάρτηση `audiorecorder`. Η πιο βασική της παράμετρο είναι το `duration`, καθώς μετρά τον χρόνο που σταματά να τρέχει το υπόλοιπο πρόγραμμα.

### 5.4.3 Συνάρτηση `getaudiodata`

Η συνάρτηση `getaudiodata` χρησιμοποιείται για την ανάκτηση δεδομένων ήχου από μια εγγραφή, τύπου `audiorecorder`. Η κυρία χρήση της μας επιτρέπει την εξαγωγή δεδομένων σε μορφή κατάλληλη για μαθηματική επεξεργασία (INT16), όπως φιλτράρισμα, μετασχηματισμός Fourier (FFT), μείωση θορύβου ή ακόμη και γραφήματα της κυματομορφής ήχου προκειμένου να αναπαρασταθεί οπτικά το ηχητικό σήμα.

### 5.4.4 Συνάρτηση `audiowrite`

Η συνάρτηση `audiowrite` χρησιμοποιείται για την εγγραφή (αποθήκευση) δεδομένων ήχου σε αρχεία ήχου με διάφορους υποστηριζόμενους τύπους αρχείων, όπως WAV, FLAC ή MP4. Αυτή η συνάρτηση επιτρέπει την εξαγωγή ηχητικών δεδομένων για περαιτέρω χρήση ή αναπαραγωγή εκτός MATLAB. Η συνάρτηση `audiowrite` της Matlab επιτρέπει την παραμετροποίηση εκτός από τις βασικές απαιτήσεις όπως, όνομα αρχείου, δεδομένα ήχου και συχνότητα δειγματοληψίας, και τον καθορισμό επιπλέον ιδιοτήτων έλεγχου όπως παρακάτω.

- `BitsPerSample` Καθορισμός των αριθμών των bits ανά δείγμα για την ποιότητα του ήχου.
- υποστήριξη διάφορων μορφών αρχείων όπως WAV, FLAC, MP3 και άλλα. Το format καθορίζεται από την επέκταση στο όνομα αρχείου (π.χ. wav, flac, .mp4).
- υποστήριξη διάφορων μορφών δεδομένων όπως μονοφωνικά (mono) ή στερεοφωνικά (stereo).

### 5.4.5 Συνάρτηση `fft`

Η συνάρτηση `fft` χρησιμοποιείται για την εφαρμογή του γρήγορου Μετασχηματισμού Fourier (Fast Fourier Transform). Ο σκοπός της είναι να μετατρέψει ένα σήμα από το πεδίο του χρόνου στο πεδίο της συχνότητας. Αυτός ο μετασχηματισμός είναι εξαιρετικά χρήσιμος για την ανάλυση του φάσματος των σημάτων ιδίως με την χρήση του υπολογιστή, καθώς απλοποιούνται οι μαθηματικές του πράξεις. Μερικές εφαρμογές της συνάρτησης δίνονται παρακάτω.

- Αφαίρεση θορύβου: Μετασχηματίζοντας τον ήχο στο πεδίο της συχνότητας μπορούμε να εντοπίσουμε και να αφαιρέσουμε ανεπιθύμητες συχνότητες που αντιπροσωπεύουν θόρυβο.
- Δημιουργία ηχητικών εφέ: Μπορούμε να ενισχύσουμε ή να τροποποιήσουμε συγκεκριμένες συχνότητες για να δημιουργήσουμε ιδιαίτερα ηχητικά εφέ.
- Ανάλυση φάσματος ήχου: Μπορούμε να αναγνωρίσουμε ποιες συχνότητες είναι πιο έντονες στο σήμα, κάτι που είναι χρήσιμο για την έρευνα ή την μουσική παραγωγή.
- Φιλτράρισμα συχνότητας: Εφαρμογή φίλτρων χαμηλής, υψηλής ή ζωνικής διέλευσης, για προσαρμογή του ήχου.
- Συμπίεση ήχου: Μείωση του μεγέθους των δεδομένων ήχου μέσω ανάλυσης και ανασύνθεσης, με βάση τη φασματική περιεκτικότητα.

### 5.4.6 Συνάρτηση `findpeaks`

Η συνάρτηση `findpeaks` χρησιμοποιείται για την εύρεση τοπικών μεγίστων (κορυφών) σε ένα ηχητικό αρχείο. Μπορεί να μας δίνει τις τιμές των κορυφών και τις αντίστοιχες χρονικές θέσεις τους.

όπως επίσης το πλάτος και την προεξοχή των κορυφών. Είναι ιδιαίτερα χρήσιμη συνάρτηση ιδίως όταν προσαρμόσται με τις βασικές της παραμέτρους που περιλαμβάνει.

- **MinPeakHeight:** Καθορίζει το ελάχιστο ύψος των κορυφών που θα ανιχνευθούν.
- **MinPeakDistance:** Ορίζει την ελάχιστη απόσταση μεταξύ γειτονικών κορυφών.
- **Threshold:** Φιλτράρει κορυφές βάσει του ύψους τους, σε σχέση με τα γειτονικά σημεία.
- **SortStr:** Επιτρέπει την ταξινόμηση των κορυφών, π.χ. από ψηλότερη προς χαμηλότερη.
- **Width:** Υπολογίζει το πλάτος των κορυφών.
- **Prominence:** Υπολογίζει την προεξοχή των κορυφών.

### 5.4.7 Συνάρτηση `xcorr`

Η συνάρτηση `xcorr` χρησιμοποιείται για τον υπολογισμό της συσχέτισης (cross-correlation) ή της αυτοσυσχέτισης (autocorrelation) μεταξύ δύο διακριτών σημάτων. Με την αυτοσυσχέτιση, δυναμώνουμε τις κορυφές ενός σήματος και εξασθενούμε τον θόρυβο. Μερικές βασικές χρήσεις της `xcorr` περιλαμβάνουν.

- Ανάλυση σημάτων για την εύρεση μοτίβων ή καθυστερήσεων.
- Εφαρμογές σε ραντάρ, υπερήχους και επεξεργασία εικόνας.
- Υπολογισμό της ομοιότητας μεταξύ δύο σημάτων σε διαφορετικές χρονικές καθυστερήσεις.

### 5.4.8 Συνάρτηση `find`

Η συνάρτηση `find` χρησιμοποιείται για τον εντοπισμό των θέσεων και των τιμών των μη μηδενικών στοιχείων σε έναν πίνακα.. Επίσης, υπό συγκεκριμένες συνθήκες μπορεί να χρησιμοποιηθεί για να βρει στοιχεία που ικανοποιούν κάποιες παραμέτρους, όπως το να είναι μικρότερα από έναν αριθμό ή να ισούνται με μια συγκεκριμένη τιμή. Επίσης σε περίπτωση που χρησιμοποιήσουμε περισσότερα από ένα κριτήρια, δύναται να κάνουμε επεξεργασία και ανάλυση δεδομένων σε επιστημονικές και μηχανικές εφαρμογές.

## 5.5 Συμπέρασμα

Με την βοήθεια της βιβλιοθήκης `MIR toolbox` όσο και κάποιων συναρτήσεων της `MATLAB` από τις πάμπολλες που υπάρχουν και με καταλλήλους συνδυασμούς αυτών, μελετούμε το θέμα μας. Στο επόμενο κεφάλαιο θα δούμε πως ακριβώς δουλεύουν και τι αποτελέσματα φέρνουν στα υπό επεξεργασία αρχεία ήχου.

## 5.6 Επίλογος

Σε αυτό το κεφάλαιο παρουσιάσαμε τις συναρτήσεις της βιβλιοθήκης `MIR toolbox` καθώς και της `MATLAB` που θα χρησιμοποιηθούν για την μελέτη της πτυχιακής εργασίας.

## Κεφάλαιο 6<sup>ο</sup> : Αλγοριθμική επεξεργασία ήχητικού περιεχομένου.

### 6.1 Εισαγωγή

Στο παρόν κεφάλαιο, παρουσιάζεται και επεξηγείται τμηματικά όλη η αλγοριθμική επεξεργασία με την οποία έγινε η μελέτη της εργασίας μας. Αρχικά, περιγράφονται οι συνδυασμοί των συναρτήσεων που μελετήθηκαν πριν από την επιλογή του τελικού αλγόριθμου, και παρουσιάζονται οι λόγοι για τους οποίους επιλέχθηκε αυτός, έναντι των άλλων. Επειδή παρακάτω θα μιλάμε συνεχώς για συχνότητες που αντιστοιχούν σε συγκεκριμένες νότες, παραθέτονται σε μορφή πίνακα, όλες οι νότες σε όλες τις πιθανές οκτάβες και τις ακριβείς συχνότητες τους, για να αποτελούν σημεία αναφοράς στην μελέτη μας.

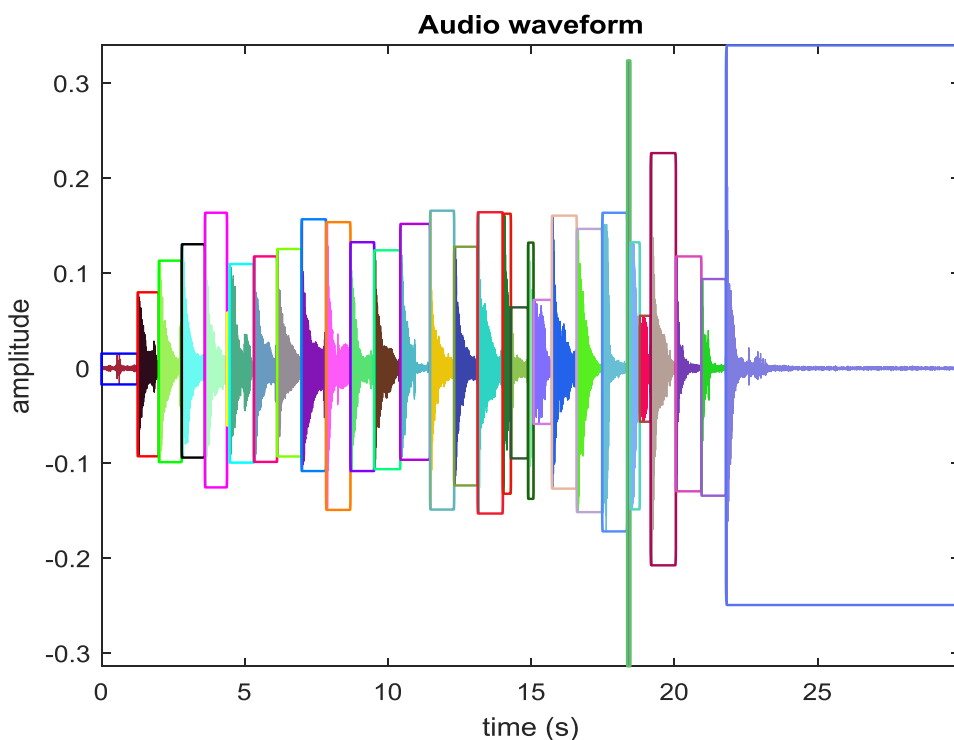
Oct	0	1	2	3	4	5	6	7	8	
[01]	C:	16.35,	32.70,	65.41,	130.81,	261.63,	523.25,	1046.50,	2093.00,	4186.01
[02]	C#:	17.32,	34.65,	69.30,	138.59,	277.18,	554.37,	1108.73,	2217.46,	4434.92
[03]	D:	18.35,	36.71,	73.42,	146.83,	293.66,	587.33,	1174.66,	2349.32,	4698.63
[04]	D#:	19.45,	38.89,	77.78,	155.56,	311.13,	622.25,	1244.51,	2489.02,	4978.03
[05]	E:	20.60,	41.20,	82.41,	164.81,	329.63,	659.25,	1318.51,	2637.02,	5274.04
[06]	F:	21.83,	43.65,	87.31,	174.61,	349.23,	698.46,	1396.91,	2793.83,	5587.65
[07]	F#:	23.12,	46.25,	92.50,	185.00,	369.99,	739.99,	1479.98,	2959.96,	5919.91
[08]	G:	24.50,	49.00,	98.00,	196.00,	392.00,	783.99,	1567.98,	3135.96,	6271.93
[09]	G#:	25.96,	51.91,	103.83,	207.65,	415.30,	830.61,	1661.22,	3322.44,	6644.88
[10]	A:	27.50,	55.00,	110.00,	220.00,	440.00,	880.00,	1760.00,	3520.00,	7040.00
[11]	A#:	29.14,	58.27,	116.54,	233.08,	466.16,	932.33,	1864.66,	3729.31,	7458.62
[12]	B:	30.87,	61.74,	123.47,	246.94,	493.88,	987.77,	1975.53,	3951.07,	7902.13

Το μοναδικό όργανο που έχει όλες τις οκτάβες όπως φαίνονται παραπάνω είναι το πιάνο. Το μπουζούκι εν προκειμένου περιέχει ως χαμηλότερη την νότα C3 και σαν υψηλότερη την νότα F6. Εδώ θα αναφερθεί και κάτι που κάνει το μπουζούκι ιδιαίτερο όργανο. Ο ήχος του παράγεται από την πένα που κτυπάει την κάθε χορδή, διαφοροποιείται αν είναι προς τα κάτω το χτύπημα (θέση) ή προς τα πάνω (άρση). Στην θέση χτυπάμε και τις δυο χορδές ταυτόχρονα, ενώ στην άρση μόνον την μια. Έτσι προκύπτουν εντελώς διαφορετικές σειρές αρμονικών συχνοτήτων παρότι παίζουμε την ίδια νότα Αυτό αποτελεί έναν δισεπίλυτο γρίφο, όπως θα διαπιστωθεί παρακάτω, στην φασματική ανάλυση.

## 6.2 Φασματική ανάλυση στις νότες του μπουζουκιού

Πριν τη διεξοδική φασματική ανάλυση των κυματομορφών ήχου από ηχογραφήσεις συγκεκριμένων μουσικών κομματιών, κρίθηκε απαραίτητο να γίνει μια πρώτη φασματική ανάλυση στις νότες του οργάνου όταν παίζονται ανεξάρτητα και σε ελεύθερους χρόνους.

Ο αλγόριθμος που χρησιμοποιήθηκε για τη φασματική ανάλυση είναι ο παρακάτω. Αρχικά γίνεται ανάγνωση του κατάλληλου αρχείου. Το αρχείο έχει δημιουργηθεί με εγγραφή όπου οι νότες του μπουζουκιού στις χορδές D και A παίζονται αργά διαδοχικά και με τεχνική πέννας θέση (χτύπημα προς τα κάτω) και άρση (χτύπημα προς τα πάνω). Με δεδομένη τη συχνότητα δειγματοληψίας στα 44100 Hz, με την οποία έγινε και η εγγραφή καθορίζεται η ανάλυση (resolution) συχνότητας και κατόπιν οι τιμές των συχνοτήτων για ένα σύνολο 65536 δειγμάτων. Στη συνέχεια γίνεται η ανάγνωση του αρχείου και χρησιμοποιείται η συνάρτηση `mironsets` για να βρεθούν τα σημεία εκείνα στα οποία υπάρχει απότομη αλλαγή πλάτους έτσι ώστε να καθοριστεί η στιγμή χτυπήματος της χορδής. Αφού βρεθούν οι στιγμές αυτές, μετά χρησιμοποιείται η συνάρτηση `mirsegment` για να διαχωριστεί η ροή στα διαστήματα εκείνα που έχει παιχτεί διαφορετική νότα. Ενδεικτικά, τα τμήματα στα οποία γίνεται ο διαχωρισμός φαίνονται στην εικόνα 6.1.



Εικόνα 6.1 Διαχωρισμός σε διαστήματα μετά τη χρήση της συνάρτησης `mirsegment`

Σημειώνεται ότι αυτό δεν είναι πάντα εφικτό λόγω της σύνθετης δομής των αρχείων ήχου, αλλά εν προκειμένω οι συγκεκριμένες στιγμές ήταν εκ των προτέρων προσεγγιστικά γνωστές και ευδιάκριτες δια γυμνού οφθαλμού στο διάγραμμα του ήχου στο πεδίο του χρόνου, οπότε έγιναν οι σχετικές διορθώσεις. Μετά το διαχωρισμό σε τμήματα, χρησιμοποιείται η συνάρτηση `mirspectrum` σε κάθε τμήμα ξεχωριστά για να ληφθούν μέσω της συνάρτησης `get` τα φάσματα πλάτους της κάθε νότας.

Τέλος, τα φάσματα πλάτους απεικονίζονται με τη χρήση τη συνάρτησης plot της Matlab, με περιορισμό μέχρι τα 3 KHz για να είναι ορατές οι βασικές αρμονικές.

```

clear
close all
clc

fs = 44100;

freqValue = [];
freqRes = 22050/65536;
FreqPosInit = [0, (1:65536)*freqRes];

xfile = 'testa23.wav';

InMusic = miraudio(xfile)
aa = get(InMusic, 'Data');
tt = get(InMusic, 'Pos');
Amp = aa{1,1}{1,1};
Time = tt{1,1}{1,1};

Onsets = mironsets(InMusic, 'attacks', 'Detect', 'Peaks')
att = get(Onsets, 'AttackPos');
pt = get(Onsets, 'Pos');
PT = pt{1,1}{1,1};
Onsetindex = att{1,1}{1,1}{1,1};
attacktimes = PT(Onsetindex);
attacktimes = [Time(1);attacktimes];
numofpeaks = length(attacktimes);
for ii = 1:numofpeaks-1
    if attacktimes(ii+1)-attacktimes(ii) < LeastValue
        attacktimes(ii+1)=0;
    end
end
zeroindex = find(attacktimes==0);
attacktimes(zeroindex) = [];

seg = mirsegment(InMusic, attacktimes)

segtimes = [attacktimes;Time(end)];
for ii = 1: length(segtimes)-1
    Seg
miraudio(xfile, 'Extract', segtimes(ii), segtimes(ii+1), 'TrimThreshold', 0.
2);
    SegFreqs = mirspectrum(Seg, 'MinRes', 0.5, 'Normal');
    fv = get(SegFreqs, 'Data');
    fp = get(SegFreqs, 'Pos');
    fvnew = fv{1,1}{1,1};

```

```

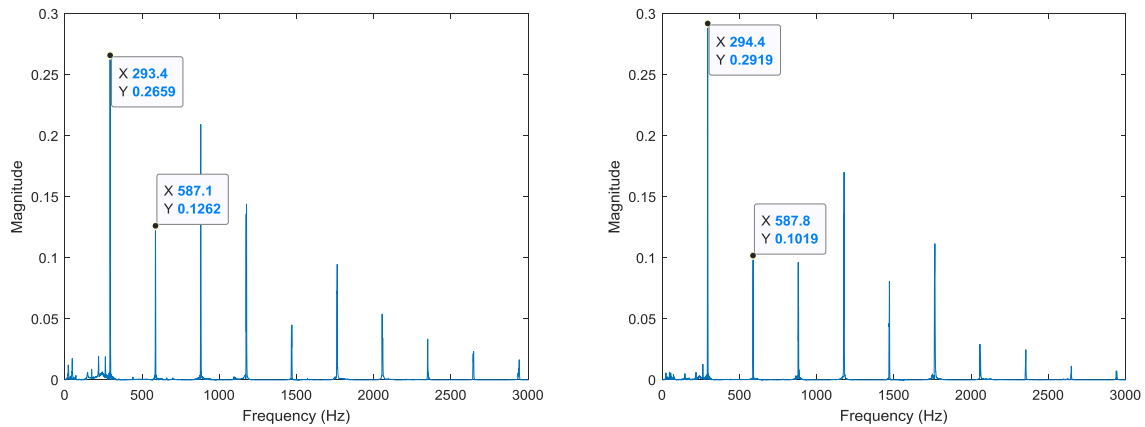
if length(fvnew)>65537
    fvnew(65538:end)=[];
end
freqValue(ii,:) = fvnew.'; % spectrum of note played

end
freqPos = fp{1,1}{1,1}.';

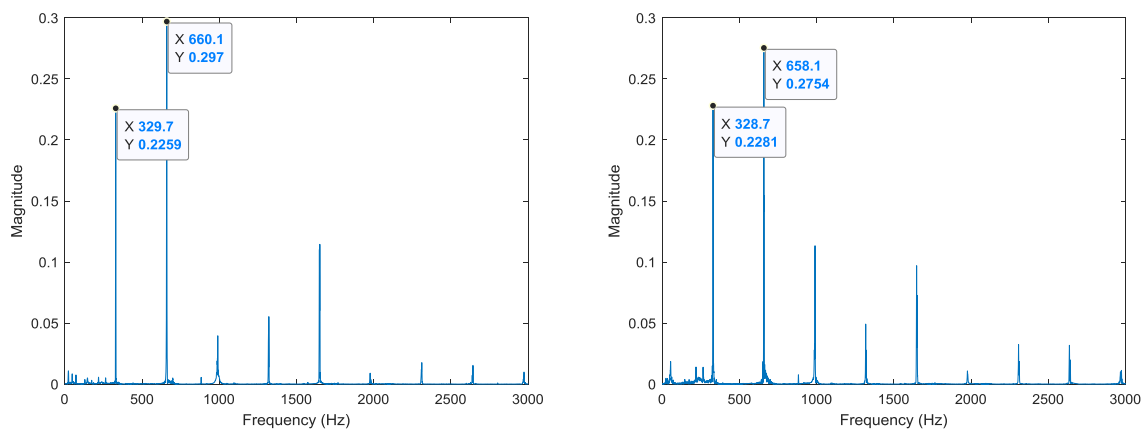
for kk = 1:20
    figure
    plot(freqPos(1:65537),freqValue(kk,:));
    xlabel('Frequency (Hz)');
    ylabel('Magnitude');
    xlim([0 3000]);
End

```

Τα αποτελέσματα παρουσιάζονται στις παρακάτω εικόνες 6.2 έως 6.9 κατά αντιπαράθεση, με πένα σε θέση και σε άρση, για να δούμε καλύτερα αυτό που αναφέρθηκενωρίτερα για την διαφορά στις αρμονικές που προκύπτουν με διαφορετικό τρόπο παιξίματος.



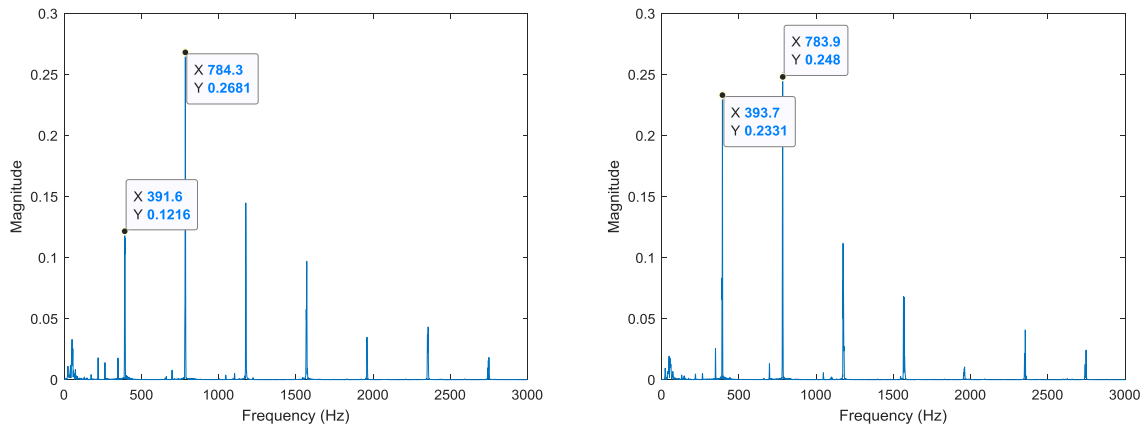
Εικόνα 6.2 D4 στη χορδή D με πένα θέση και άρση



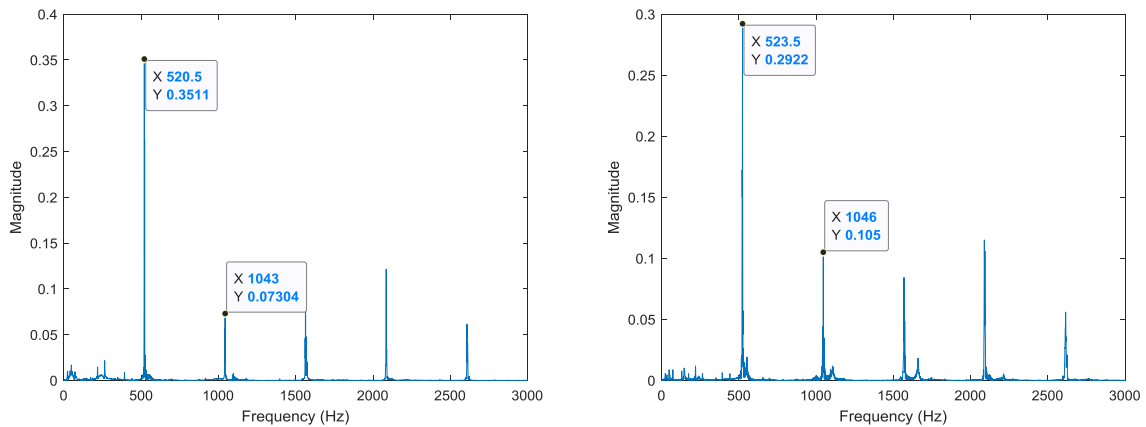
Εικόνα 6.3 E4 στη χορδή D με πένα θέση και άρση

Στην εικόνα 6.2 βλέπουμε στην νότα D4 που είναι άταστη (ανοικτή χορδή) τα έξης: Κατά πρώτον το κούρδισμα επειδή γίνεται με πένα θέση, στο παίξιμο με άρση υπάρχει μια μικρή απόκλιση συχνότητας ως προς τον τόνο της νότας. Κατά δεύτερον η βασική νότα είναι η ισχυρότερη (έχει το μεγαλύτερο πλάτος) και ακολουθούν οι αρμονικές με μικρότερα πλάτη, κάτι που περιμένουμε ως προφανές.

Στην εικόνα 6.3 βλέπουμε στην νότα E4 τα ίδια όπως παραπάνω, όσον αφορά το κούρδισμα δηλαδή υπάρχει και εδώ όπως και σε όλες τις παρακάτω αντιστοιχίσεις, μια μικρή απόκλιση συχνότητας ως προς τον τόνο της νότας. Αυτό όμως που μας εκπλήσσει εδώ είναι ότι, και με τα δυο παιξίματα της πέννας η δεύτερη αρμονική είναι ισχυρότερη από την βασική αρμονική, κάτι που δεν περιμέναμε. Αυτό είναι κάτι που μάλλον χαρακτηρίζει τα μπουζούκια κατασκευαστικά σαν όργανα, και ίσως, είναι και ο κύριος λόγος για το ωραίο άκουσμα που συνοδεύει το παίξιμο τους, και αρέσει τόσο πολύ.



Εικόνα 6.4 G4 στη χορδή D με πένα θέση και άρση

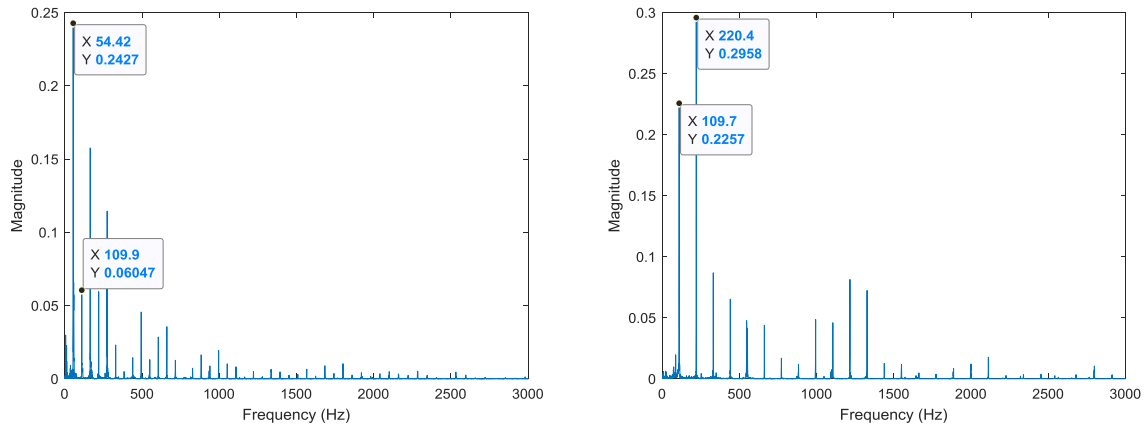


Εικόνα 6.5 C5 στη χορδή D με πένα θέση και άρση

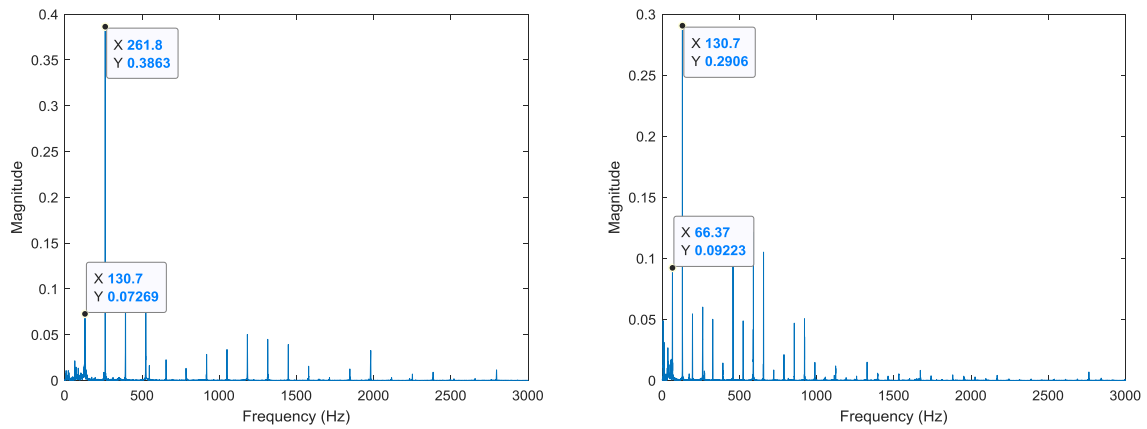
Στην εικόνα 6.4 στην νότα G4 βλέπουμε επίσης όπως και προηγουμένως, τη δεύτερη αρμονική να είναι ισχυρότερη από τη βασική αρμονική, κάτι που πλέον δεν μας εκπλήσσει. Μετά την πρώτη αρμονική βλέπουμε την αλληλουχία των επόμενων αρμονικών να εξελίσσεται κανονικά.

Στην εικόνα 6.5 στην νότα C5 βλέπουμε τη βασική αρμονική όπως και στη νότα D4, ισχυροτερη από τη δεύτερη αρμονική, κάτι που πλέον μας δίνει σοβαρή ένδειξη ότι η πιο δυνατή συχνότητα μπορεί να είναι όχι μόνον η βασική, αλλά και κάποια πρώτη ή ίσως, και δεύτερη αρμονική.

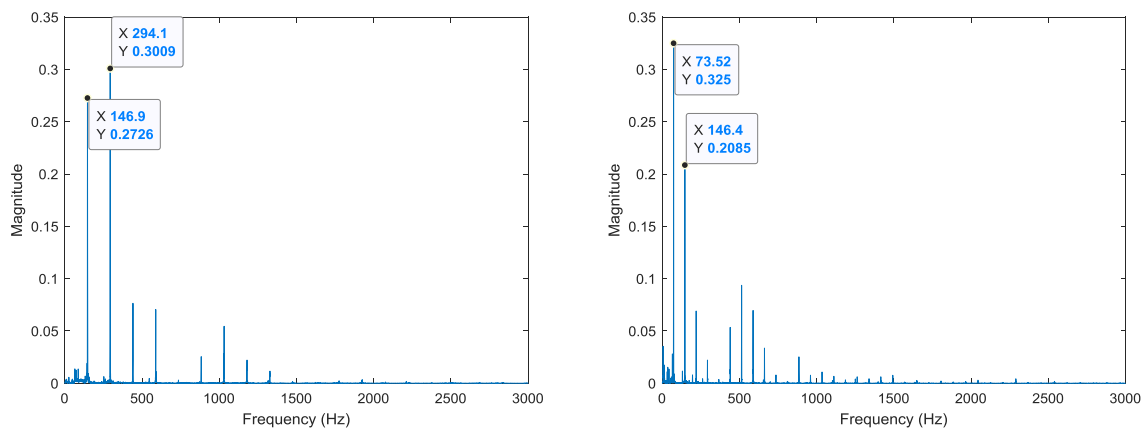
## Κεφάλαιο 6



Εικόνα 6.6 A3 στη χορδή Α με πένα θέση και άρση



Εικόνα 6.7 C4 στη χορδή Α με πένα θέση και άρση

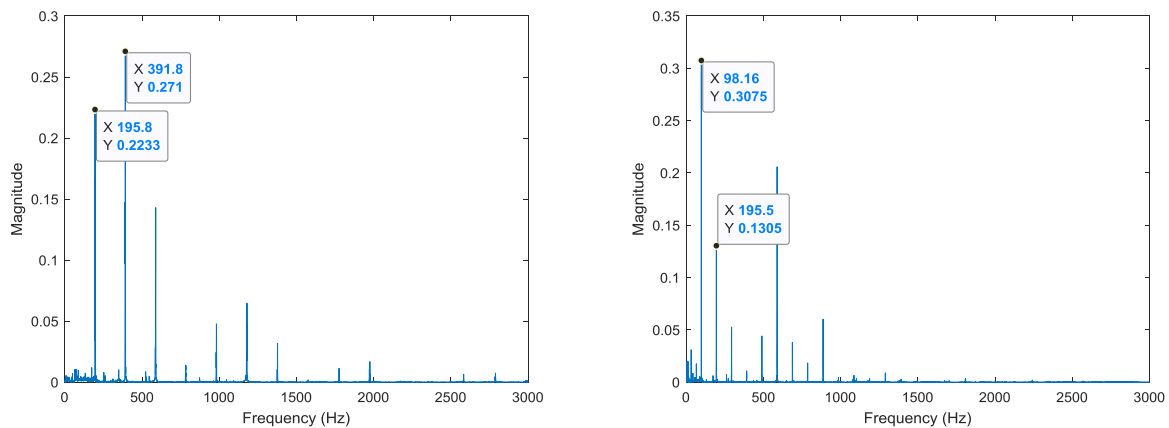


Εικόνα 6.8 D4 στη χορδή Α με πένα θέση και άρση

Στην εικόνα 6.6 στη νότα A3 που είναι και αυτή άταστη (ανοικτή χορδή) βλέπουμε κάτι καινούριο πλέον. Στην άρση η πιο ισχυρή συχνότητα είναι η πρώτη αρμονική, αλλά στην θέση ισχυρότερη είναι η προηγούμενη αρμονική από την βασική συχνότητα, κάτι που περιπλέκει ακόμη περισσότερο τα πράγματα.

Στην εικόνα 6.7 στη νότα C4 που είναι στη χορδή ΛΑ βλέπουμε κάτι καινούριο πάλι Στην άρση η πιο ισχυρή συχνότητα είναι η πρώτη προηγούμενη αρμονική, αλλά στη θέση ισχυρότερη είναι η βασική.

Στην εικόνα 6.8 στη νότα D4 που είναι στη χορδή Α βλέπουμε τα συγκριτικά με την χορδή D. Στη θέση η πιο ισχυρή συχνότητα είναι η βασική αρμονική, όπως και στην εικόνα 6.2, αλλά στην άρση, ισχυρότερη είναι η δεύτερη προηγούμενη συχνότητα σε σχέση με την πρώτη προηγούμενη κάτι που δεν είναι αναμενόμενο.



Εικόνα 6.9 G4 στη χορδή Α με πένα θέση και άρση

Τέλος στην εικόνα 6.9 στη νότα G4 που είναι στη χορδή Α βλέπουμε τα συγκριτικά με την χορδή D Στη θέση η πιο ισχυρή συχνότητα είναι η πρώτη προηγούμενη αρμονική, όπως και στην εικόνα 6.4., αλλά στην άρση ισχυρότερη είναι η δεύτερη προηγούμενη συχνότητα σε σχέση με την πρώτη προηγούμενη, κάτι που το συναντήσαμε και στην εικόνα 6.8

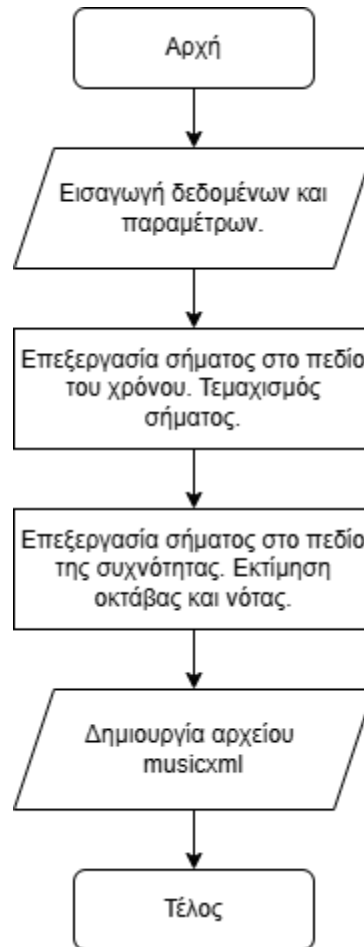
Συνοψίζοντας τα παραπάνω, είμαστε σίγουροι ότι δεν υπάρχει εύκολη οδός καθορισμού της βασικής νότας που έχει παιχτεί, βάση της δυνατώτερης συχνότητας που προκύπτει από το φασματογράφημα. Έτσι είμαστε αναγκασμένοι να κάνουμε συνδυαστικές τεχνικές επεξεργασίας για να προσεγγίσουμε καλύτερα την μελέτη μας

### 6.3 Αλγόριθμος επεξεργασίας: Παρουσίαση και επεξηγήσεις

Σε αυτήν την ενότητα θα παρουσιαστεί ο αλγόριθμος επεξεργασίας με τις απαραίτητες επεξηγήσεις όπου χρειάζεται, για να γίνει αντιληπτή η εργασία σε όλη την έκταση της. Θα χωρίσουμε τον αλγόριθμο σε ενότητες, καθώς κρίνεται αναγκαίο να μιλήσουμε ξεχωριστά για την κάθε ενότητα, δίνοντας μεγαλύτερη έμφαση εκεί που έχει γραφεί αλγόριθμος και όχι στις ενότητες που χρησιμοποιήθηκαν έτοιμες συναρτήσεις από την βιβλιοθήκη. Στην εικόνα 6.10 φαίνεται ένα γενικό διάγραμμα ροής του αλγόριθμου που παρουσιάζει περιγραφικά αυτές τις ενότητες, που θα παρουσιάσουμε λεπτομερώς στην συνέχεια.

#### 6.3.1 Καθορισμός παραμέτρων και εισαγωγή αρχείου ήχου.

Στην εικόνα 6.11 φαίνεται ένα αδρό διάγραμμα ροής του τμήματος του αλγόριθμου που αντιστοιχεί στην παραμετροποίηση και την εισαγωγή του μουσικού αρχείου. Το σχετικό τμήμα φαίνεται αμέσως παρακάτω.



Εικόνα 6.10 Γενικό διάγραμμα ροής του κώδικα.

```

clear
close all
clc

NoteFreqs = [16.35, 32.70, 65.41, 130.81, 261.63, 523.25, 1046.50,
2093.00, 4186.01;
             17.32, 34.65, 69.30, 138.59, 277.18, 554.37, 1108.73,
2217.46, 4434.92;
             18.35, 36.71, 73.42, 146.83, 293.66, 587.33, 1174.66,
2349.32, 4698.63;
             19.45, 38.89, 77.78, 155.56, 311.13, 622.25, 1244.51,
2489.02, 4978.03;
             20.60, 41.20, 82.41, 164.81, 329.63, 659.25, 1318.51,
2637.02, 5274.04;
             21.83, 43.65, 87.31, 174.61, 349.23, 698.46, 1396.91,
2793.83, 5587.65;
             23.12, 46.25, 92.50, 185.00, 369.99, 739.99, 1479.98,
2959.96, 5919.91;
             24.50, 49.00, 98.00, 196.00, 392.00, 783.99, 1567.98,
3135.96, 6271.93;
    
```

```

        25.96, 51.91, 103.83, 207.65, 415.30, 830.61, 1661.22,
3322.44, 6644.88;
        27.50, 55.00, 110.00, 220.00, 440.00, 880.00, 1760.00,
3520.00, 7040.00;
        29.14, 58.27, 116.54, 233.08, 466.16, 932.33, 1864.66,
3729.31, 7458.62;
        30.87, 61.74, 123.47, 246.94, 493.88, 987.77, 1975.53,
3951.07, 7902.13];

Notes = {'C', 'C#', 'D', 'D#', 'E', 'F', 'F#', 'G', 'G#', 'A', 'A#', 'B'};
Octaves = 0:8;
xfile = 'testHARAMATA.wav';
fs = 44100;
bpm = 115;
rythmnom = 4;
rythmdenom = 4;

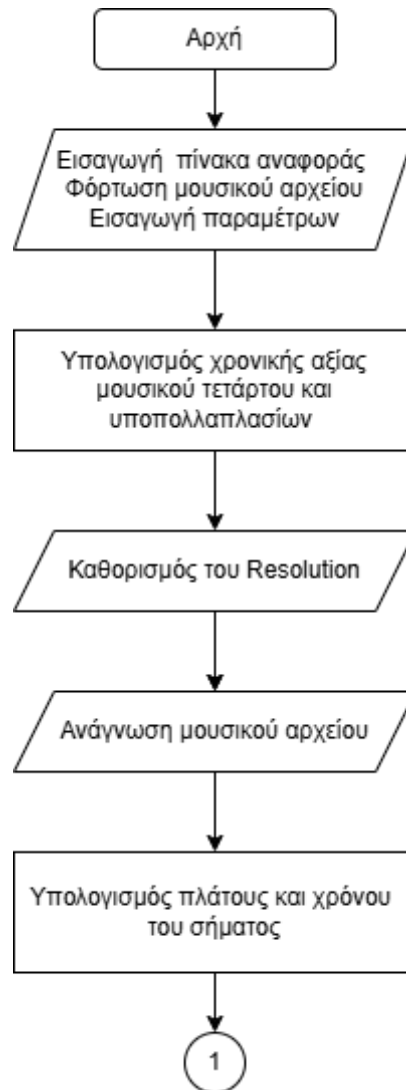
quarter = 60/bpm; % a quarter
eighth = quarter/2;
sixteenth = quarter/4;
thirtysecond = quarter/8;
sixtyfourth = quarter/16;
LeastValue = sixteenth;
divisions = 4; % 4 for sixteenth, 8 for thirtysecond;

freqValue = [];
freqRes = 22050/65536;

InMusic = miraudio(xfile)
aa = get(InMusic, 'Data');
tt = get(InMusic, 'Pos');
Amp = aa{1,1}{1,1};
Time = tt{1,1}{1,1};

```

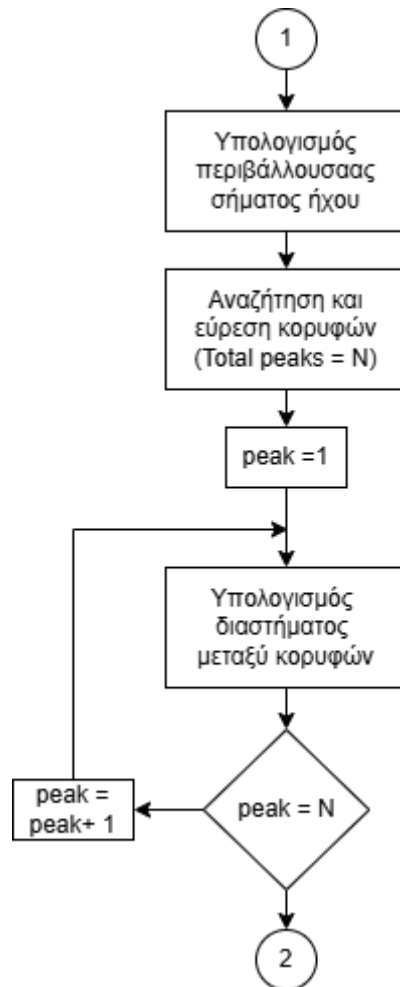
Ξεκινά με τις εντολές καθαρισμού, και αμέσως φορτώνονται σε ένα πίνακα αναφοράς οι συχνότητες και η αντιστοίχιση αυτών, με όλες τις νότες. Φορτώνεται το μουσικό αρχείο που έχουμε ηχογραφήσει και εν προκειμένω έχει τον τίτλο **'testHARAMATA.wav'**. Με δεδομένη τη συχνότητα δειγματοληψίας στα 44100 Hz, και το ρυθμό εγγραφής beats per minute (bpm), π.χ. 115 στην περίπτωση αυτή, με τον οποίο έγινε η εγγραφή, ορίζουμε το χρόνο του μουσικού τετάρτου και όλων των υποπολλαπλασίων αυτού. Κατόπιν καθορίζουμε την ανάλυση (resolution) συχνότητας. Στη συνέχεια γίνεται η ανάγνωση του αρχείου και εξαγωγή των δειγμάτων πλάτους ανά χρονική στιγμή δειγματοληψίας από αυτό, όπως φαίνεται για παράδειγμα στην εικόνα 5.1.



Εικόνα 6.11 Εισαγωγή δεδομένων και παραμετροποίηση.

### 6.3.2 Επεξεργασία αρχείου ήχου στο πεδίο του χρόνου.

Στην εικόνα 6.12 φαίνεται ένα αδρό διάγραμμα ροής του τμήματος του αλγόριθμου που αντιστοιχεί στην επεξεργασία του σήματος στο πεδίο του χρόνου. Αρχικά, χρησιμοποιώντας την συνάρτηση **miEnvelope** βρίσκουμε την περιβάλλουσα του σήματος. Έπειτα με την **mirpeaks** βρίσκουμε τις κορυφές, θεωρώντας ότι εκεί είναι και τα χτυπήματα της πένας του μπουζουκιού. Αφού τις βρούμε κάνουμε κάποιες ρυθμίσεις για να προκύψουν οι χρονικές στιγμές, κανονικοποιούμε ως προς τη συχνότητα δειγματοληψίας, και έτσι βρίσκονται οι πραγματικοί χρόνοι, οι οποίοι μας δίνουν το **segtimes**. Από το **segtimes** θα προκύψουν τα τμήματα που θα χωριστεί η κυματομορφή, ώστε να μπορέσουμε να υπολογίσουμε σε κάθε τμήμα τη νότα. Τέλος σε αυτό το loop βρίσκουμε την αξία κάθε νότας, δηλαδή πόσο χρόνο διαρκεί η νότα. Αυτόν τον χρόνο τον μετράμε ως προς το **LeastValue**, που είναι η ελάχιστη αξία νότας που έχει καθοριστεί από τον χρήστη. Μετά σπάμε σε κομμάτια για να δουλέψουμε φασματικά, το κάθε κομμάτι ξεχωριστά.



Εικόνα 6.12 Επεξεργασία σήματος στο πεδίο του χρόνου

```

testxx = mirenvelope(InMusic,'HalfWaveDiff','Normal','PostDecim',1);

FirstPeaks = mirpeaks(testxx,'Contrast',0.05,'Reso','LeastValue','Only');
PAmp = get(FirstPeaks,'Data');
PPos = get(FirstPeaks,'Pos');
peakamptemp = PAmp{1,1}{1,1};
peakpos = find(peakamptemp);
peakamps = peakamptemp(peakpos);

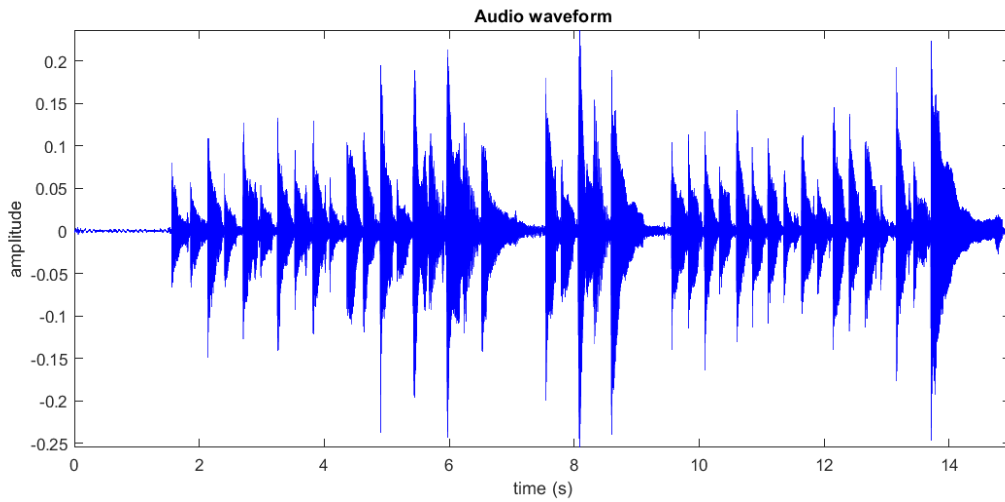
peaktimes = peakpos/fs;

segtimes = [peaktimes;Time(end)]-0.025;

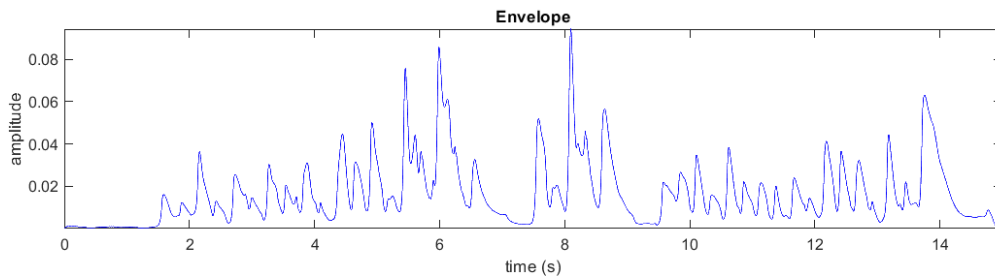
figure
stem(peaktimes,peakamps);
xlim([0 Time(end)]);

for ii = 1:length(segtimes)-1
    segDiffs(ii) = segtimes(ii+1)-segtimes(ii);
end
    
```

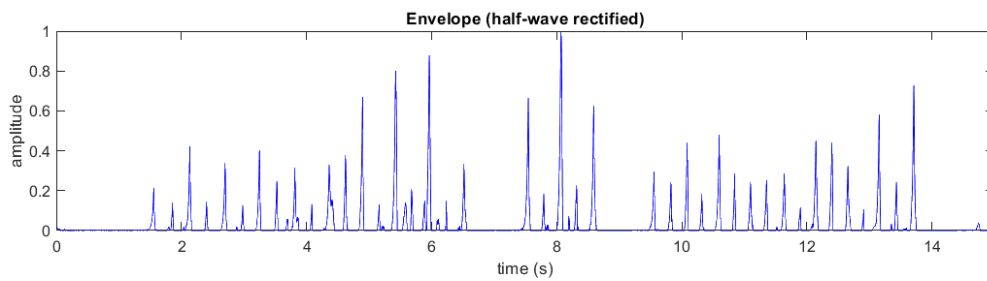
```
for ii = 1:length(segtimes)-1
    durations(ii) = 2*round(segDiffs(ii)/(2*LeastValue));
end
```



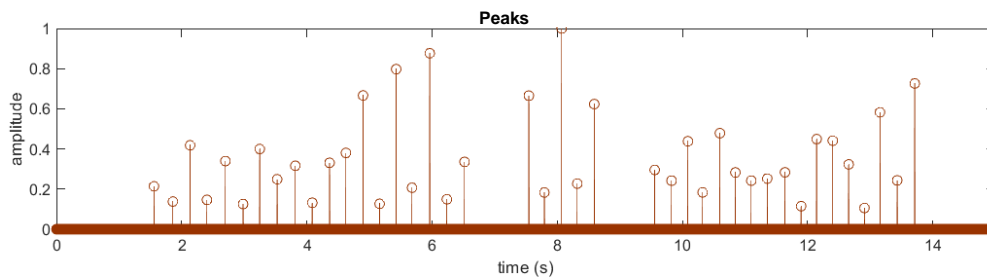
(α)



(β)

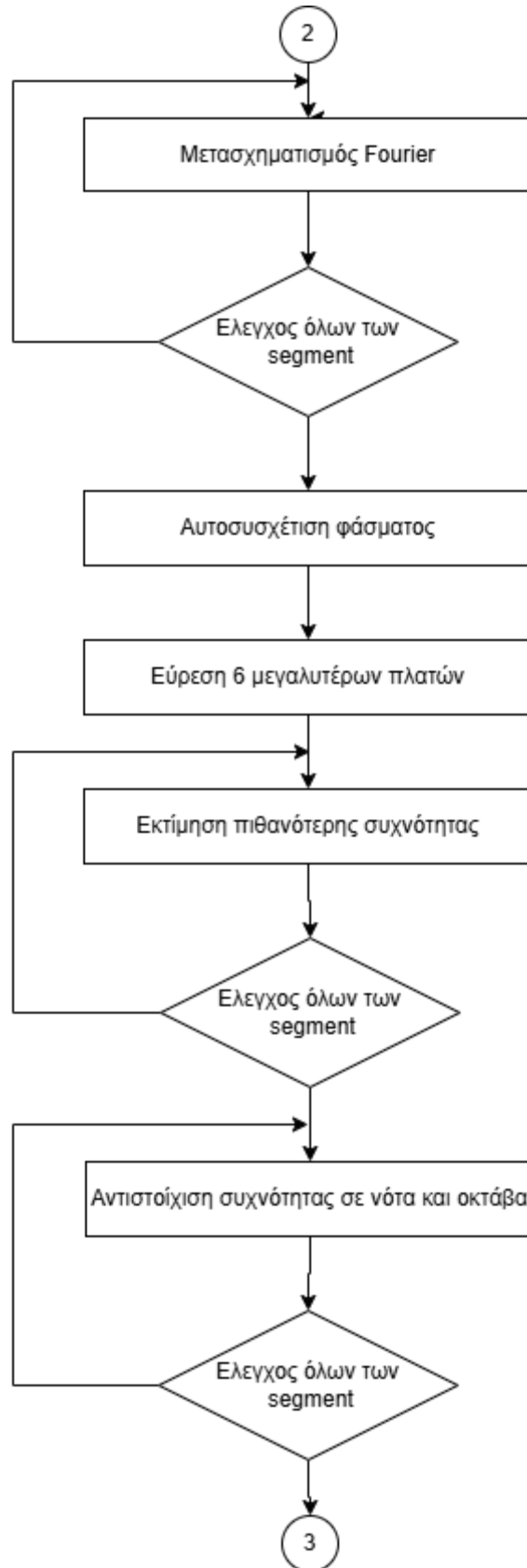


(γ)



(δ)

Εικόνα 6.13 Αποτελέσματα χρήσης των συναρτήσεων του `mirtoolbox`. (α) Αρχική κυματομορφή, (β) απλή χρήση `miirenvelope`, (γ) χρήση `miirenvelope` με διαφορισμό, (δ) χρήση `mirpeaks` για την αναγνώριση των κορυφών της περιβάλλουσας.



Εικόνα 6.14 Επεξεργασία στο πεδίο της συχνότητας

Στην εικόνα 6.13 φαίνεται αναλυτικά η διαδικασία που ακολουθείται για την εύρεση των κορυφών μέσω των οποίων θα γίνει ο διαχωρισμός των τμημάτων που με τη σειρά τους θα καθορίσουν τις αξίες των νοτών που παίζονται. Σημειώνουμε, ότι η διαδικασία αυτή μελετήθηκε διεξοδικά με διάφορες εναλλακτικές μεθόδους και συναρτήσεις του **miraudio**, πριν καταλήξουμε σε αυτή την μεθοδολογία. Στο διάγραμμα 6.13(α) σχεδιάζεται το αρχικό σήμα στο πεδίο του χρόνου. Στο διάγραμμα 6.13(β) φαίνεται το αποτέλεσμα χρήσης της συνάρτησης **mirenvelope** χωρίς κάποια ιδιαίτερη παραμετροποίηση. Διαπιστώθηκε ότι από αυτή την κυματομορφή, η οποία δημιουργείται με τη χρήση ανόρθωσης σήματος και χαμηλοπερατού φίλτρου, εξασθενούν οι ισχυρές τιμές πλάτους που αντιστοιχούν στα χτυπήματα της χορδής. Για το λόγο αυτό η συνάρτηση **mirenvelope** παραμετροποιήθηκε έτσι ώστε να γίνεται διαφορισμός της (αφαίρεση κάθε τιμής πλάτους από την προηγούμενη). Με τον τρόπο αυτό προκύπτει η κυματομορφή του διαγράμματος 6.13(γ), η οποία δεν αντιστοιχεί μεν, σε ακριβή αναπαράσταση της περιβάλλουσας, διατηρεί όμως και τονίζει τις ζητούμενες κορυφές πλάτους. Κατόπιν με τη χρήση της **mirpeaks** διατηρούνται μόνον οι κορυφές που ξεχωρίζουν από την υπόλοιπη κυματομορφή, όπως φαίνεται στο διάγραμμα 6.13(δ). Επιπλέον, σημειώνεται ότι σε κάποιες περιπτώσεις απαιτείται αφαίρεση κάποιων κορυφών (με τη χρήση ενός κατωφλίου γιατί δεν αντιστοιχούν σε χτυπήματα χορδής αλλά σε δευτερεύοντα συντονισμό που οδηγεί σε ενίσχυση του πλάτους στο πεδίο του χρόνου).

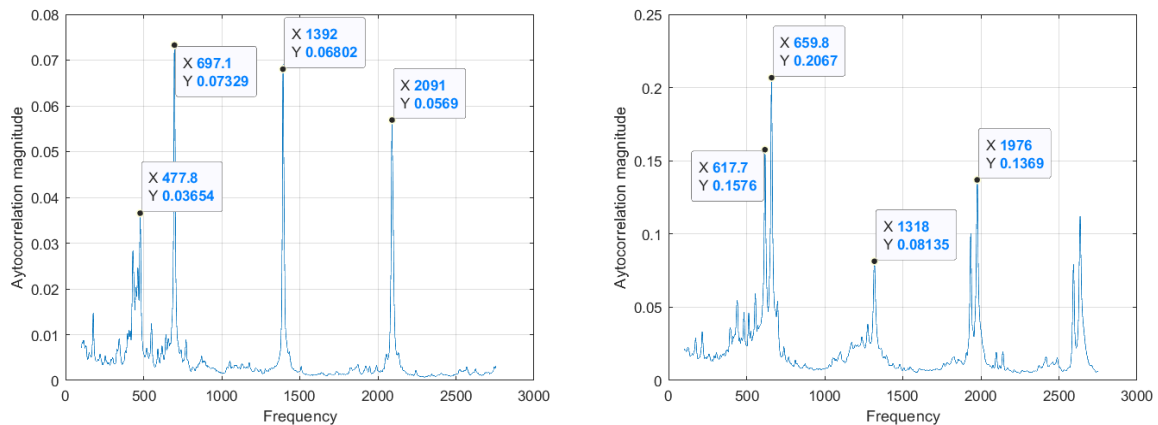
### 6.3.3 Επεξεργασία αρχείου ήχου στο πεδίο της συχνότητας.

Στην εικόνα 6.14 φαίνεται ένα αδρό διάγραμμα ροής του τμήματος του αλγόριθμου που αντιστοιχεί στην επεξεργασία του σήματος στο πεδίο της συχνότητας. Ξεκινάμε την επεξεργασία με έναν βρόχο για κάθε τμήμα (όπως αυτά έχουν καθοριστεί στην επεξεργασία στο πεδίο του χρόνου), και για κάθε τμήμα (**segment**), εφαρμόζουμε μετασχηματισμό Fourier με την συνάρτηση **mirspectrum**. Αφού τον υπολογίσουμε για κάθε τμήμα ξεχωριστά, με την συνάρτηση **get** παίρνουμε τις τιμές συχνότητας και πλάτους ανά συχνότητα, δηλαδή το φάσμα πλάτους, και αυτά τα δεδομένα τα βάζουμε σε έναν πίνακα για περαιτέρω επεξεργασία, όπως φαίνεται στον παρακάτω αλγόριθμο.

```
for ii = 1:length(segtimes)-1
    Seg =
miraudio(xfile, 'Extract', segtimes(ii), segtimes(ii+1), 'TrimThreshold', 0.
2);
    SegFreqs = mirspectrum(Seg, 'MinRes', 0.5, 'Normal');
    fv = get(SegFreqs, 'Data');
    fp = get(SegFreqs, 'Pos');
    fvnew = fv{1,1}{1,1};
    if length(fvnew) > 65537
        fvnew(65538:end) = [];
    end
    freqValue(ii,:) = fvnew.'; % spectrum of note played
end
freqPos = fp{1,1}{1,1}.';
```

Στη συνέχεια, για να γίνει η εκτίμηση της νότας που έχει παιχτεί, υπολογίζουμε την αυτοσυσχέτιση του φάσματος πλάτους με την συνάρτηση **xcorr**, και προσπαθούμε να βρούμε που είναι οι κορυφές δηλαδή σε ποιες συχνότητες είναι οι μεγαλύτερες κορυφές. Αυτό προκύπτει με την χρήση της **findpeaks** και επειδή οι κορυφές είναι περισσότερες από μια, εξετάζουμε ποιες από αυτές έχουν

αρμονικές. Για παράδειγμα στην εικόνα 6.15, φαίνονται δύο περιπτώσεις αυτοσυσχέτισης του φάσματος πλάτους για δύο διαστήματα. Στην πρώτη περίπτωση (αριστερά), η νότα που παίχτηκε είναι η νότα F (ονομαστική συχνότητα 698.66 Hz). Είναι φανερό πως εκτός από τη συχνότητα 697.1 Hz (η οποία είναι επαρκώς κοντά στην ονομαστική) και τις αρμονικές της (1392 και 2091 Hz) υπάρχει και η ισχυρή συχνότητα 477.8 Hz. Αντίστοιχα, στο δεξί σχήμα, εκτός από τη συχνότητα 659.8 Hz (που αντιστοιχεί στη νότα E) και τις αρμονικές υπάρχουν συχνότητες με υψηλή τιμή αυτοσυσχέτισης. Σε αρκετές περιπτώσεις οι «παρασιτικές» συχνότητες μπορεί να είναι ισχυρότερες, από τη συχνότητα της νότας που παίχτηκε.



Εικόνα 6.15 Αυτοσυσχέτιση φάσματος πλάτους δύο περιπτώσεων

Για να αποφύγουμε σφάλματα στην εκτίμηση της νότας, τρέχουμε έναν αλγόριθμο που παίρνει τις έξι συχνότητες μεγαλύτερου πλάτους, και εξετάζουμε κάθε μια αν έχει σχέση αρμονικής με τις υπόλοιπες. Με αυτόν τον τρόπο, βρίσκουμε την συχνότητα που έχει τις περισσότερες αρμονικές (με κάποιο περιθώριο σφάλματος στη συχνότητα, το οποίο τίθεται ως ποσοστό της υπό εξέταση συχνότητας) μέσα σε αυτές τις έξι που επικρατήσαν, και αυτήν ορίζουμε σαν την κύρια συχνότητα που ψάχνουμε. Έτσι έχουμε τώρα τις συχνότητες για κάθε διάστημα.

Στη συνέχεια αφαιρούμε συχνότητες που δεν είναι μέσα στη ζώνη θεμελιωδών συχνοτήτων που παράγει το μπουζούκι (περίπου μεταξύ 130 Hz και 1500 Hz). Σε περίπτωση μη ευρέσεως κάποιας συχνότητας με ισχυρές αρμονικές (δηλαδή μεταξύ των έξι ισχυρότερων συχνοτήτων), επιλέγεται η συχνότητα με τη μεγαλύτερη τιμή αυτοσυσχέτισης. Ο σχετικός αλγόριθμος ακολουθεί.

```

PeakHeight = [];
for ii = 1:length(segtimes)-1

    PeakHeight(ii) = 0.025;
    C(ii,:) = xcorr(freqValue(ii,:));
    C1(ii,:) = C(ii,65537:65537+8192);
    F_low = floor(100/freqRes); % Low frequency in number of samples
    [allpeaks,locs] =
findpeaks(C1(ii,F_low:end),'MinPeakHeight',PeakHeight(ii),'MinPeakDistance',15);
    tempFreqs = (F_low+locs-2)*freqRes;

    while length(locs) > 6
        PeakHeight(ii) = PeakHeight(ii) + 0.01;
    end
end
    
```

```

    [allpeaks,locs] =
findpeaks(C1(ii,F_low:end),'MinPeakHeight',PeakHeight(ii),'MinPeakDista
nce',15);
    end
    tempFreqs = (F_low+locs-2)*freqRes;
    tempAmps = C1(ii,F_low+locs-1);

    estfreqs{ii,:} = tempFreqs;

    CurrentFreqs = estfreqs{ii,:};
    ll = 0;
    pp = 0;
    PossibleFreq = [];
    for jj = 1:length(CurrentFreqs)-1
        testFreq = CurrentFreqs(jj);

        mm = 0;
        testmat = CurrentFreqs;
        testmat(jj) = [];
        for kk = jj+1:length(CurrentFreqs)
            CheckThreshold = 0.03*testFreq;
            testrem = abs(CurrentFreqs(kk) -
round(CurrentFreqs(kk)/testFreq)*testFreq);
            if round(CurrentFreqs(kk)/testFreq) > 1 && testrem <
CheckThreshold
                mm = mm+1;
            end
        end
    end
    if mm > 0
        pp = pp+1;
        PossibleFreq(pp,1) = jj;
        PossibleFreq(pp,2) = testFreq;
        PossibleFreq(pp,3) = tempAmps(jj);
        PossibleFreq(pp,4) = mm;
    end
end
if isempty(PossibleFreq)
    FoundTemp = 0;
    while FoundTemp < 130 || FoundTemp > 1500
        [maxF,locF] = max(allpeaks);
        FoundTemp = tempFreqs(locF);
        allpeaks(locF) = [];
    end
    FoundFreq{ii,1} = FoundTemp;
    if FoundTemp == 0
        [maxF,locF] = max(allpeaks);
        FoundFreq{ii,1} = tempFreqs(locF);
    end
end

```

```

        end
    else
        [maxF,locF] = max(PossibleFreq(:,4));
        FoundFreq{ii,1} = PossibleFreq(locF,2);
    end
end

end
NumberofNotesFound = length(segtimes)-1;

```

Τέλος, κάνουμε την αντιστοίχιση της συχνότητας σε νότα και οκτάβα, υπολογίζοντας την απόσταση της εκτιμηθείσας, με την προηγούμενη διαδικασία, συχνότητας από όλες τις πιθανές συχνότητες που έχουν οριστεί αρχικά στον πίνακα **NoteFreqs**, όπως φαίνεται στον παρακάτω αλγόριθμο.

```

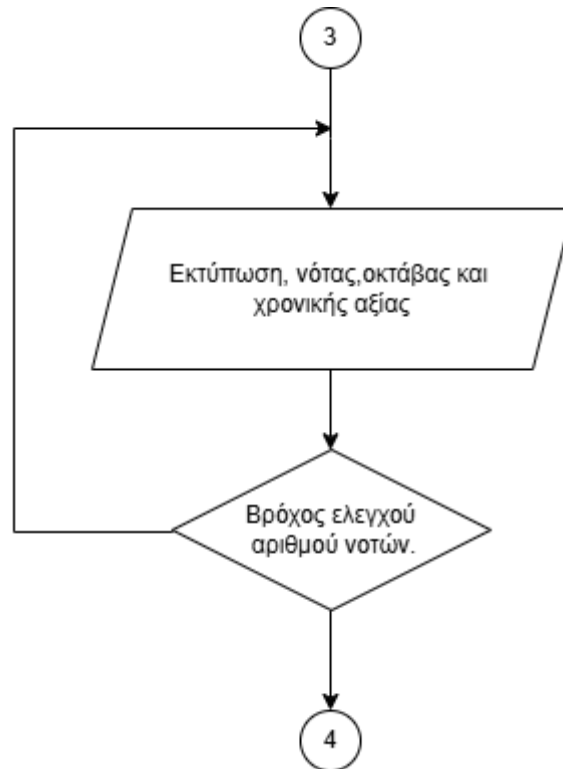
kk = 0;
for ii = 1:length(segtimes)-1
    tempFreqs = FoundFreq{ii};
    if isempty(tempFreqs)
        NumberofNotesFound = NumberofNotesFound-1;
    else
        for jj = 1:length(tempFreqs)
            [xx,yy] = min(abs(tempFreqs(jj)-NoteFreqs));
            [zz,vv] = min(xx);
            Oct{ii,jj} = vv-1;
            ResNote{ii,jj} = yy(vv);
            kk = kk+1;
            InitInstants(kk) = segtimes(ii);
        end
    end
end

end

```

### 6.3.4 Εμφάνιση αποτελεσμάτων στην οθόνη\

Στην εικόνα 6.16 φαίνεται ένα αδρό διάγραμμα ροής του τμήματος του αλγόριθμου που αντιστοιχεί στην εντολή για την εκτύπωση των αποτελεσμάτων στην οθόνη μας. Εκτυπώνουμε λοιπόν στην οθόνη τη νότα, την οκτάβα, και τη χρονική διάρκεια αυτής, όπως προκύπτει από την αντιστοίχιση με τους πίνακες αναφοράς.



Εικόνα 6.16 Εκτύπωση αποτελεσμάτων στην οθόνη.

```

NoteFinal = ResNote{ii,1};
tr{ii,1} = OctFinal;
tr{ for ii = 1:NumberofNotesFound
OctFinal = Oct{ii,1};
ii,2} = Notes{NoteFinal};
tr{ii,3} = durations(ii);
fprintf("%d. Note = %s%d, Time = %5.3f, Duration = %d \n",ii,
Notes{NoteFinal},OctFinal,InitInstants(ii),durations(ii));
end
  
```

### 6.3.5 Οργάνωση παραμέτρων με βάση το πρότυπο musicxml

Σε αυτό το κομμάτι του αλγορίθμου, μορφοποιούμε τις μουσικές πληροφορίες που έχουμε βρει νωρίτερα με βάση το πρότυπο **musicxml**, ώστε να μπορεί μετά να διαβαστεί από κάποιο κλασικό πρόγραμμα, εν προκειμένου το **muse score**, και να εξαχθεί η παρτιτούρα. Ακολουθεί όπως και παραπάνω, η εικόνα 6.17 που απεικονίζει το αδρό διάγραμμα ροής και ο αλγόριθμος με τις απαραίτητες επεξηγήσεις.



Εικόνα 6.17.Οργάνωση μεταβλητών

```

rnom = rythmnom;
rdenom = rythmdenom;
divs = divisions;
transcript = tr;
notestable = ['C', 'D', 'E', 'F', 'G', 'A', 'B'];

measurelength = rnom*divs;
for ii = 1:size(transcript,1)
    Octave(ii) = transcript{ii,1};
    Note{1,ii} = transcript{ii,2};
    dur(ii) = transcript{ii,3};
end
alter = zeros(1,length(Note));

for ii = 1:length(Note)
    st = Note{1,ii};
    fnote(ii) = find(st(1)==notestable);
    if strlength(st) == 2
        if st(2)=="#"
            alter(ii) = 1;
        end
    end
end
    
```

```

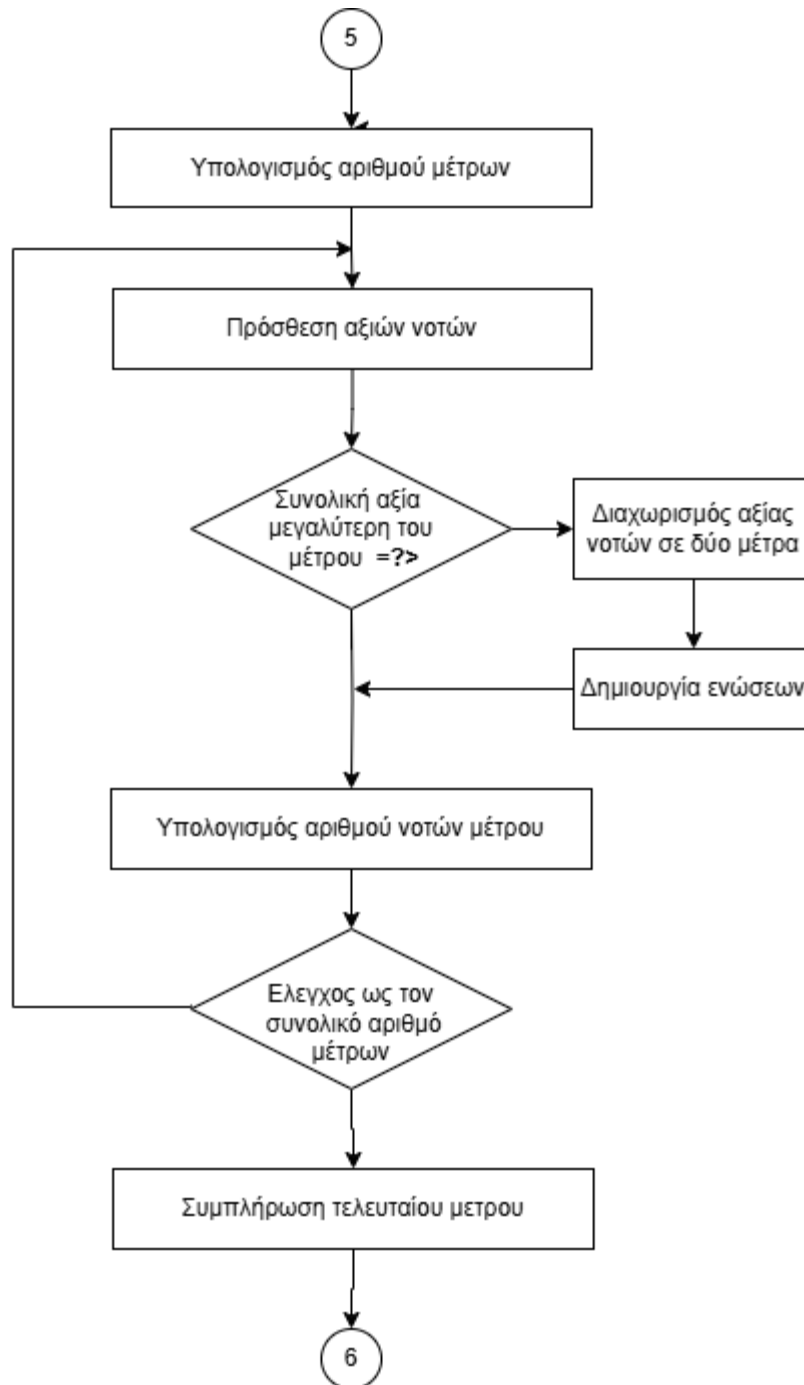
        else
            alter(ii) = -1;
        end
    end
end
end

```

Σε αυτό το τμήμα του κώδικα που μόλις είδαμε, ξεκινάμε ορίζοντας καινούριες μεταβλητές για να μην χρησιμοποιήσουμε τις ίδιες με το πρώτο κομμάτι, και ορίζουμε τον πίνακα με τις επτά βασικές νότες (ντο ρε μι φα σολ λα σι). Έτσι μπορούμε να μετρήσουμε πόσες νότες ανά μετρό, και πόσα μέτρα συνολικά απαρτίζουν την μελωδία που παίζαμε. Εκτός όμως από την μεταβλητή της νότας, της οκτάβας και της διάρκειας αυτής, δημιουργούμε και μια καινούρια μεταβλητή, την **alter**, η οποία αποθηκεύει τον αλλοίωση της νότας, δηλαδή πότε και που έχουμε δίεση ή ύφεση στην νότα. Αυτό γίνεται γιατί έτσι είναι δομημένο το **muse score**, ώστε να μπορεί να γράψει παρτιτούρα. Έτσι έχουμε όλες τις νότες που είχαμε γράψει προηγουμένως με την μορφή χαρακτήρων, όπως C#, F# D#, και σπάμε αυτό το κομμάτι στα δυο. Το μεν πρώτο κομμάτι που είναι η νότα το βάζουμε στον πίνακα **fnote**, το δε δεύτερο κομμάτι το βάζουμε στον πίνακα **alter**. Οι τιμές του πίνακα αυτού είναι +1, -1, και 0, και σημαίνουν αντίστοιχα, δίεση, ύφεση ή νότα καθαρή χωρίς οπλισμό.

### 6.3.6 Υπολογισμός μέτρων παρτιτούρας

Στην εικόνα 6.18 φαίνεται ένα αδρό διάγραμμα ροής του τμήματος του αλγορίθμου που αντιστοιχεί στον υπολογισμό των μέτρων από το πρώτο μέχρι το τελευταίο. Ξεκινάμε να δημιουργούμε το κάθε μέτρο από το πρώτο και μέχρι το τελευταίο, βάζοντας την αξία κάθε νότας όπως την έχουμε από τις προηγούμενες μετρήσεις. Κάθε φορά που ολοκληρώνουμε ένα μέτρο, προχωράμε στο επόμενο, μέχρι και την τελευταία νότα. Επειδή όμως συχνά υπάρχουν νότες που εμπλέκονται στο τέλος του μέτρου και στην αρχή του επόμενου, είναι πολύ σημαντικό να δημιουργήσουμε τις συνδέσεις αυτές και να τις γράψουμε στον πίνακα **ties**. Έτσι στο τέλος αυτής της ενότητας του αλγορίθμου έχουμε όχι μόνον τα μέτρα, αλλά και τις νότες που υπάρχουν σε αυτά, καθώς και τις νότες που συνδέουν συνεχόμενα μέτρα. Εδώ έχουμε τελειώσει με την προεργασία για το **music xml**.



Εικόνα 6.18 Υπολογισμός των μέτρων της παρτιτούρας

```

totalDuration = sum(dur);
numofMeasures = ceil(totalDuration/measurelength);
tot_dur = cumsum(dur);
dur_temp = dur;
l1 = 0;
ties = zeros(1,length(fnote));
for jj = 1:numofMeasures-1
    ds = 0;
    kk = 0;

```

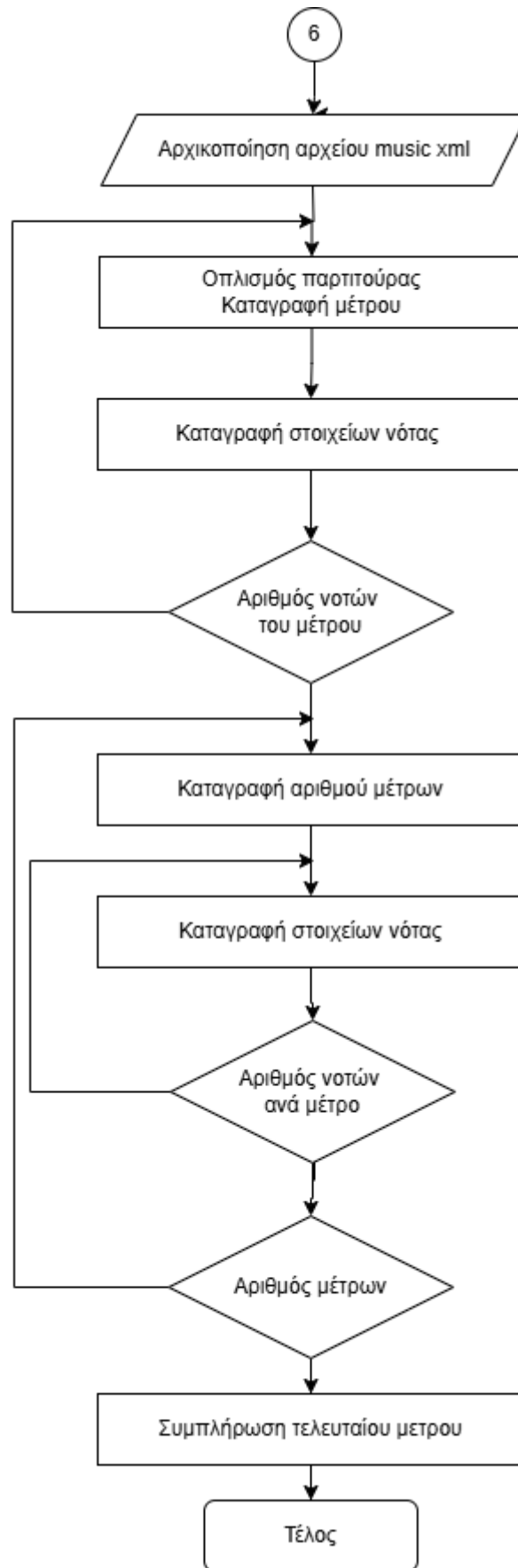
```

while ds < measurelength
    kk = kk+1;
    ll = ll+1;
    if ll <= length(dur_temp)
        ds = ds + dur_temp(ll);
    end
end
if ds > measurelength
    tomove = ds - measurelength;
    tokeep = dur_temp(ll) - tomove;
    dur_temp(ll) = tokeep;
    dur_temp = [dur_temp(1:ll), tomove, dur_temp(ll+1:end)];
    Octave = [Octave(1:ll), Octave(ll), Octave(ll+1:end)];
    fnote = [fnote(1:ll), fnote(ll), fnote(ll+1:end)];
    alter = [alter(1:ll), alter(ll), alter(ll+1:end)];
    if ll == length(dur_temp)-1
        ties = [ties(1:ll-1), 1, 2];
    else
        ties = [ties(1:ll-1), 1, 2, ties(ll+1:end)];
    end
end
notesperMeas(jj) = kk;
end
% last measure
notesperMeas(numofMeasures) = length(dur_temp)-ll;
lastmeasurelength = sum(dur_temp(length(dur_temp)-
notesperMeas(numofMeasures)+1:length(dur_temp)));
if lastmeasurelength < measurelength
    xx = measurelength - lastmeasurelength;
    dur_temp(end) = dur_temp(end) + xx;
end

```

### 6.3.7 Δημιουργία αρχείου musicxml

Στην εικόνα 6.19 φαίνεται ένα αδρό διάγραμμα ροής του τμήματος που περιγραφεί τον αλγόριθμο μέχρι το τέλος Ξεκινάμε και ανοίγουμε ένα αρχείο που το ονομάζουμε '**xx.musicxml**' με write. Το fid είναι ο δείκτης που χρησιμοποιούμε για να κάνουμε print στο αρχείο. Εδώ γράφουμε όλα τα βασικά χαρακτηριστικά που πρέπει να έχει η παρτιτούρα για να μπορεί να διαβαστεί από οποιοδήποτε άλλο αρχείο Συνεχίζοντας, γράφουμε το πρώτο κομμάτι του μέτρου το οποίο έχει κάποια ιδιαίτερα χαρακτηριστικά. καθώς σε αυτό αναγράφεται ο χρόνος, το κλειδί της παρτιτούρας και ο οπλισμός της Μετά έχουμε έναν βρόχο, ο οποίος τρέχει όλες τις νότες του πρώτου μέτρου, και με τα χαρακτηριστικά που έχουμε γράψει παραπάνω, πάει και τα περνά στο αρχείο. Στην συνέχεια ξεκινάμε και γράφουμε, αφού έχουμε τελειώσει με το πρώτο μέτρο, βασιζόμενοι στην ίδια λογική. και όλα τα υπόλοιπα μέτρα, χωρίς φυσικά τα ιδιαίτερα χαρακτηριστικά που υπάρχουν μοναχά στο πρώτο μέτρο. Προσοχή χρειάζεται στο τελευταίο μέτρο, καθώς φροντίζουμε να βγαίνει σωστά η αξία του μέτρου για να ολοκληρώνεται και η παρτιτούρα.



Εικόνα 6.19 Δημιουργία αρχείου music xml

```

fid = fopen('xx.musicxml','w');
text = '<?xml version="1.0" encoding="UTF-8" standalone="no"?>';
fprintf(fid,text); fprintf(fid,"\n");
text = '<!DOCTYPE score-partwise PUBLIC "-//Recordare//DTD MusicXML 4.0
Partwise//EN" "http://www.musicxml.org/dtds/partwise.dtd">';
fprintf(fid,text); fprintf(fid,"\n");
text = '<score-partwise version="4.0">';
fprintf(fid,text); fprintf(fid,"\n");
text = '  <part-list>';
fprintf(fid,text); fprintf(fid,"\n");
text = '    <score-part id="P1">';
fprintf(fid,text); fprintf(fid,"\n");
text = '      <part-name>Music</part-name>';
fprintf(fid,text); fprintf(fid,"\n");
text = '    </score-part>';
fprintf(fid,text); fprintf(fid,"\n");
text = '  </part-list>';
fprintf(fid,text); fprintf(fid,"\n");
text = '  <part id="P1">';
fprintf(fid,text); fprintf(fid,"\n");

fprintf(fid,'    <measure number="%d">',1); fprintf(fid,"\n");
fprintf(fid,'      <attributes>'); fprintf(fid,"\n");
fprintf(fid,'        <divisions>%d</divisions>',divs);
fprintf(fid,"\n");
fprintf(fid,'        <key><fifths>0</fifths></key>');
fprintf(fid,"\n");
fprintf(fid,'        <time><beats>%d</beats><beat-type>%d</beat-
type></time>',rnom,rdenom); fprintf(fid,"\n");
fprintf(fid,'
<clef><sign>G</sign><line>2</line></clef>'); fprintf(fid,"\n");
fprintf(fid,'      </attributes>'); fprintf(fid,"\n");

for jj = 1:notesperMeas(1)
  fprintf(fid,'
<note><pitch><step>%c</step><alter>%d</alter><octave>%d</octave></pitch
>',notestable(fnote(jj)),alter(jj),Octave(jj)); fprintf(fid,"\n");
  fprintf(fid,'      <duration>%d</duration>',dur_temp(jj));
fprintf(fid,"\n");
  if ties(jj)==1
    fprintf(fid,'        <notations><tied
type="start"/></notations>'); fprintf(fid,"\n");
    end
  if ties(jj)==2
    fprintf(fid,'        <notations><tied
type="stop"/></notations>'); fprintf(fid,"\n");

```

```

    end
    fprintf(fid, '    </note>'); fprintf(fid, "\n");
end
fprintf(fid, '    </measure>'); fprintf(fid, "\n");
fnote(1:notesperMeas(1))=[];
alter(1:notesperMeas(1))=[];
Octave(1:notesperMeas(1))=[];
dur_temp(1:notesperMeas(1))=[];
ties(1:1:notesperMeas(1))=[];

for ii = 2:numofMeasures
    fprintf(fid, '    <measure number="%d">', ii); fprintf(fid, "\n");
    for jj = 1:notesperMeas(ii)
        fprintf(fid, '
<note><pitch><step>%s</step><alter>%d</alter><octave>%d</octave></pitch
>', notestable(fnote(jj)), alter(jj), Octave(jj)); fprintf(fid, "\n");
        fprintf(fid, '    <duration>%d</duration>', dur_temp(jj));
fprintf(fid, "\n");
        if ties(jj)==1
            fprintf(fid, '    <notations><tied
type="start"/></notations>'); fprintf(fid, "\n");
            end
            if ties(jj)==2
                fprintf(fid, '    <notations><tied
type="stop"/></notations>'); fprintf(fid, "\n");
            end
            fprintf(fid, '    </note>'); fprintf(fid, "\n");
        end
    end

    fprintf(fid, '    </measure>'); fprintf(fid, "\n");
    fnote(1:notesperMeas(ii))=[];
    alter(1:notesperMeas(ii))=[];
    Octave(1:notesperMeas(ii))=[];
    dur_temp(1:notesperMeas(ii))=[];
    ties(1:1:notesperMeas(ii))=[];
End

fprintf(fid, ' </part>'); fprintf(fid, "\n");
fprintf(fid, '</score-partwise>');
fclose(fid);

```

Εδώ ουσιαστικά έχουμε και το τέλος του αλγορίθμου επεξεργασίας.

## 6.4 Επίλογος

Σε αυτό το κεφάλαιο παρουσιάστηκε ο αλγόριθμος επεξεργασίας στο σύνολο του Ξεκινήσαμε με την φασματική ανάλυση στις νότες του μπουζουκιού σε ασφαλές περιβάλλον, δηλαδή παίζοντας με τον

## Κεφάλαιο 6

ίδιο τρόπο και χρόνο όλες τις νότες δυο χορδών με την σειρά. Έτσι είδαμε πως συμπεριφέρεται ο αλγόριθμος μας και κάναμε τους συνδυασμούς στις συναρτήσεις που επιλέξαμε, στοχεύοντας στην καλύτερη απόκριση του. Κατόπιν πήραμε το ηχογραφημένο αρχείο **testHARAMATA.wav** και τρέξαμε κομμάτι κομμάτι τον αλγόριθμο, χωρισμένο κατά αδρά διαγράμματα ροής, που να αποτελούν αυτοτελή κομμάτια, και στο τέλος φτιάξαμε το αρχείο **music.xml**. Έτσι πρόεκυψε η τελική παρτιτούρα.

## Κεφάλαιο 7<sup>ο</sup> : Αποτελέσματα

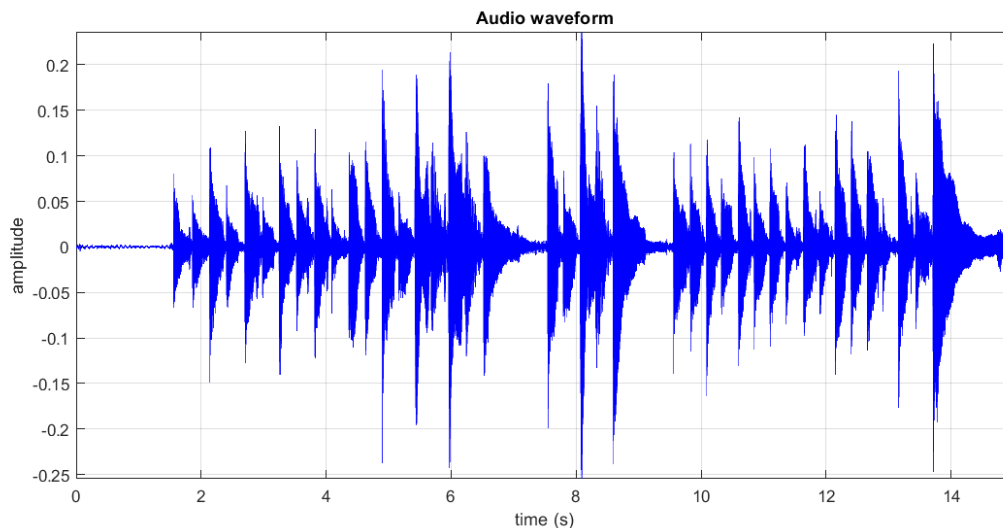
### 7.1 Εισαγωγή

Στο έβδομο κεφάλαιο παρουσιάζονται τα αποτελέσματα που έχουν προκύψει από την μελέτη πάνω στο θέμα της πτυχιακής μας εργασίας. Αρχικά παίρνουμε τρία μουσικά αρχεία wav που έχουμε ηχογραφήσει, και τρέχουμε σε αυτά τον αλγόριθμο μας. Έτσι προκύπτουν τρεις παρτιτούρες, και αφού ολοκληρώσουμε τον αλγόριθμο, τις αντιπαραβάλλουμε με τις παρτιτούρες που χρησιμοποιήσαμε για να γράψουμε τα μουσικά αρχεία. Στη συνέχεια συγκρίνονται ανά δυο οι παρτιτούρες και βλέπουμε την ομοιότητα τους, καταγράφοντας φυσικά τυχόν διαφορές τους. Τέλος υπολογίζουμε την επιτυχία της μεθόδου μας, και σχολιάζουμε πάνω σε αυτό.

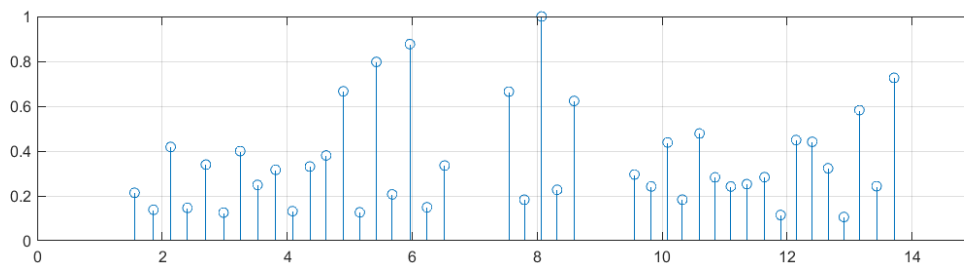
### 7.2 Αποτελέσματα

Παρακάτω θα επακολουθήσουν οι τρεις ηχογραφημένες έγγραφες και η παρουσίαση της παρτιτούρας που προκύπτει για την καθεμιά με την χρήση του αλγορίθμου μας.

#### 7.2.1 Εγγραφή 1<sup>η</sup> "Χαράματα η ώρα τρεις"



Εικόνα 7.1 Κυματομορφή από την 1η εγγραφή στο πεδίο του χρόνου και τη χρήση της συνάρτησης get.



Εικόνα 7.2 Κυματομορφή με τη χρήση της mirenvelope και διαφορισμού.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE score-partwise PUBLIC "-//Recordare//DTD MusicXML 4.0
Partwise//EN" "http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise version="4.0">
```

```

<part-list>
  <score-part id="P1">
    <part-name>Music</part-name>
  </score-part>
</part-list>
<part id="P1">
  <measure number="1">
    <attributes>
      <divisions>4</divisions>
      <key><fifths>0</fifths></key>
      <time><beats>4</beats><beat-type>4</beat-type></time>
      <clef><sign>G</sign><line>2</line></clef>
    </attributes>
    <note><pitch><step>A</step><alter>0</alter><octave>4</octave></pitch>
      <duration>2</duration>
    </note>
    <note><pitch><step>G</step><alter>1</alter><octave>4</octave></pitch>
      <duration>2</duration>
    </note>
    <note><pitch><step>A</step><alter>0</alter><octave>4</octave></pitch>
      <duration>2</duration>
    </note>
    <note><pitch><step>C</step><alter>1</alter><octave>5</octave></pitch>
      <duration>2</duration>
    </note>
    <note><pitch><step>E</step><alter>0</alter><octave>5</octave></pitch>
      <duration>2</duration>
    </note>
    <note><pitch><step>A</step><alter>0</alter><octave>4</octave></pitch>
      <duration>2</duration>
    </note>
    <note><pitch><step>C</step><alter>1</alter><octave>5</octave></pitch>
      <duration>2</duration>
    </note>
    <note><pitch><step>E</step><alter>0</alter><octave>5</octave></pitch>
      <duration>2</duration>
    </note>
  </measure>
  <measure number="2">
    <note><pitch><step>G</step><alter>0</alter><octave>5</octave></pitch>
      <duration>2</duration>
    </note>
    <note><pitch><step>F</step><alter>1</alter><octave>5</octave></pitch>
      <duration>2</duration>
    </note>
    <note><pitch><step>G</step><alter>0</alter><octave>5</octave></pitch>
      <duration>2</duration>
    </note>
    <note><pitch><step>G</step><alter>1</alter><octave>5</octave></pitch>
      <duration>2</duration>
    </note>
  </measure>

```



```

<note><pitch><step>G</step><alter>1</alter><octave>4</octave></pitch>
  <duration>2</duration>
</note>
<note><pitch><step>B</step><alter>0</alter><octave>4</octave></pitch>
  <duration>2</duration>
</note>
<note><pitch><step>E</step><alter>0</alter><octave>4</octave></pitch>
  <duration>2</duration>
</note>
<note><pitch><step>G</step><alter>1</alter><octave>4</octave></pitch>
  <duration>2</duration>
</note>
<note><pitch><step>B</step><alter>0</alter><octave>4</octave></pitch>
  <duration>2</duration>
</note>
</measure>
<measure number="6">
  <note><pitch><step>D</step><alter>0</alter><octave>5</octave></pitch>
    <duration>2</duration>
  </note>
  <note><pitch><step>C</step><alter>1</alter><octave>5</octave></pitch>
    <duration>2</duration>
  </note>
  <note><pitch><step>D</step><alter>0</alter><octave>5</octave></pitch>
    <duration>2</duration>
  </note>
  <note><pitch><step>E</step><alter>0</alter><octave>5</octave></pitch>
    <duration>2</duration>
  </note>
  <note><pitch><step>D</step><alter>0</alter><octave>5</octave></pitch>
    <duration>2</duration>
  </note>
  <note><pitch><step>C</step><alter>0</alter><octave>5</octave></pitch>
    <duration>2</duration>
  </note>
  <note><pitch><step>B</step><alter>0</alter><octave>4</octave></pitch>
    <duration>2</duration>
  </note>
  <note><pitch><step>C</step><alter>0</alter><octave>5</octave></pitch>
    <duration>2</duration>
  </note>
</measure>
<measure number="7">
  <note><pitch><step>A</step><alter>0</alter><octave>4</octave></pitch>
    <duration>16</duration>
  </note>
</measure>
</part>
</score-partwise>

```



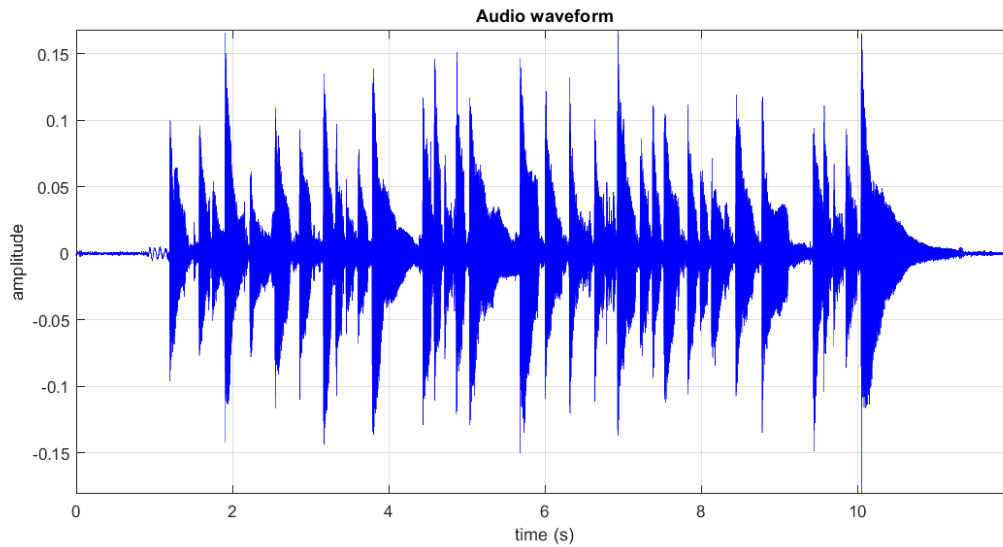
Εικόνα 7.3 Παρτιτούρα όπως προκύπτει με την χρήση του αλγορίθμου.



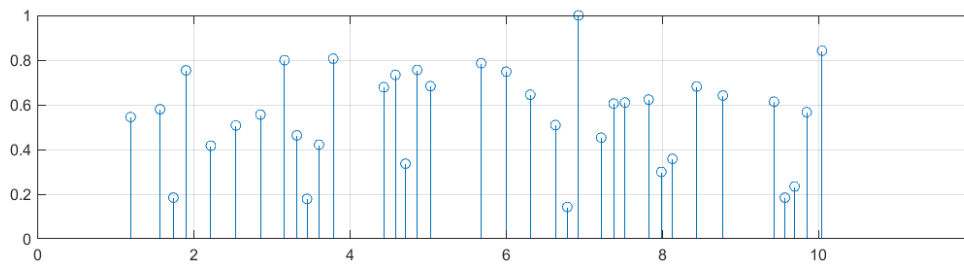
Εικόνα 7.4 Παρτιτούρα που παίχτηκε από τον μουσικό στο μπουζούκι.

Στην πρώτη εγγραφή που τρέξαμε τον αλγόριθμο μας και αποτελεί μικρό απόσπασμα από το τραγούδι με τίτλο "χαράματα η ώρα τρεις" έχουμε τα έξης. Στην εικόνα 7.1 και 7.2 φαίνεται η κυματομορφή της εγγραφής πριν και μετά την χρήση της παραμετροποίησης, με διαφορική της περιβάλλουσας του σήματος. Ακολουθεί ο αλγόριθμος παραγωγής του πρότυπου **musicxml** και στην εικόνα 7.3 έχουμε την παρτιτούρα που προκύπτει. Αν τώρα την συγκρίνουμε με την εικόνα 7.4 που είναι η πρωτότυπη παρτιτούρα, έχουμε απόλυτη ταύτιση εκτός από το 5ο μέτρο στις νότες 6,7,και 8 στις οποίες ο μουσικός έπαιξε την δεύτερη φωνή αντί της πρώτης που είναι γραμμένη, και ο αλγόριθμος μας σωστά βγάζει αυτή την διαφορά.

## 7.2.2 Εγγραφή 2<sup>η</sup> "Απονη ζωή"



Εικόνα 7.5 Κυματομορφή από την 2η εγγραφή στο πεδίο του χρόνου και τη χρήση της συνάρτησης get.



Εικόνα 7.6 Κυματομορφή με τη χρήση της mirenvelope και διαφορισμού

```
\<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE score-partwise PUBLIC "-//Recordare//DTD MusicXML 4.0
Partwise//EN" "http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise version="4.0">
  <part-list>
    <score-part id="P1">
      <part-name>Music</part-name>
    </score-part>
  </part-list>
  <part id="P1">
    <measure number="1">
      <attributes>
        <divisions>8</divisions>
        <key><fifths>0</fifths></key>
        <time><beats>4</beats><beat-type>4</beat-type></time>
        <clef><sign>G</sign><line>2</line></clef>
      </attributes>
      <note><pitch><step>G</step><alter>0</alter><octave>4</octave></pitch>
```





```

    <duration>2</duration>
  </note>
  <note><pitch><step>F</step><alter>1</alter><octave>5</octave></pitch>
    <duration>2</duration>
  </note>
  <note><pitch><step>E</step><alter>0</alter><octave>5</octave></pitch>
    <duration>2</duration>
  </note>
  <note><pitch><step>D</step><alter>0</alter><octave>5</octave></pitch>
    <duration>16</duration>
    <notations><tied type="start"/></notations>
  </note>
</measure>
<measure number="5">
  <note><pitch><step>D</step><alter>0</alter><octave>5</octave></pitch>
    <duration>32</duration>
    <notations><tied type="stop"/></notations>
  </note>
</measure>
</part>
</score-partwise>

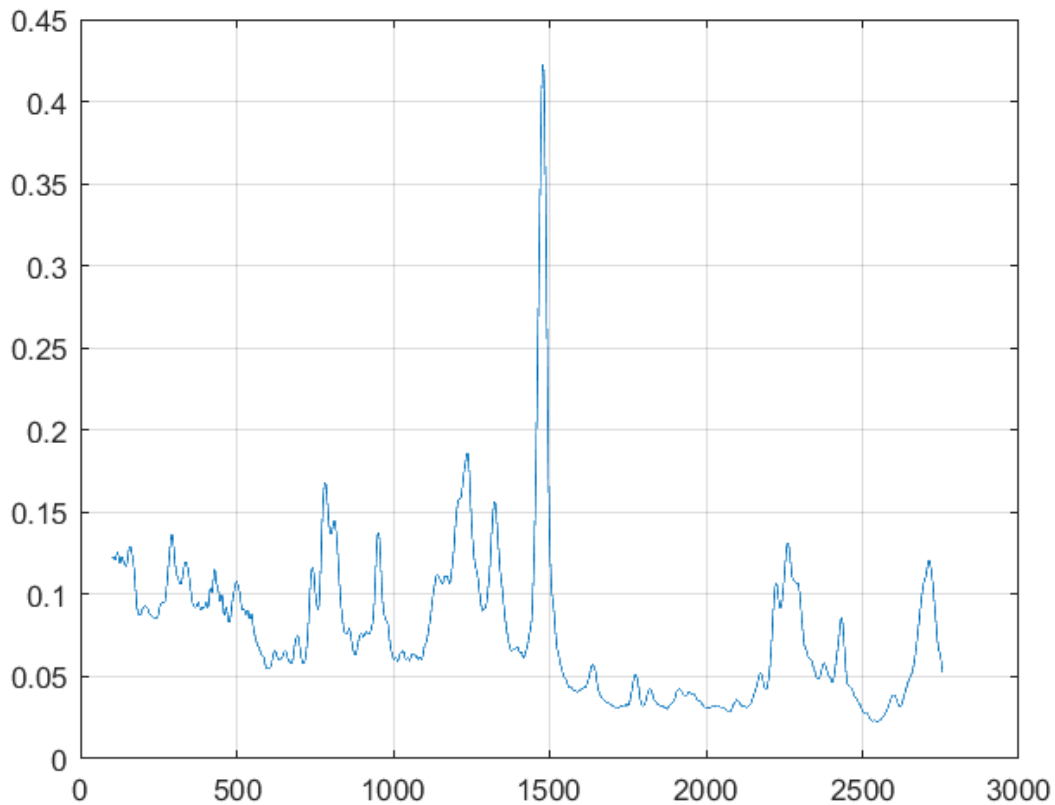
```



Εικόνα 7.7 Παρτιτούρα όπως προκύπτει με την χρήση του αλγορίθμου.



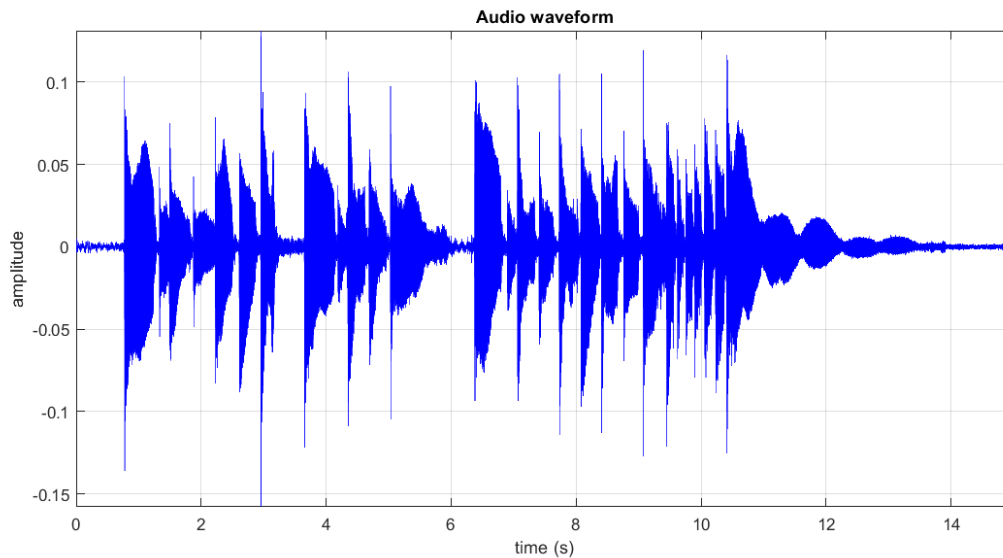
Εικόνα 7.8 Παρτιτούρα που παίχτηκε από τον μουσικό στο μπουζούκι



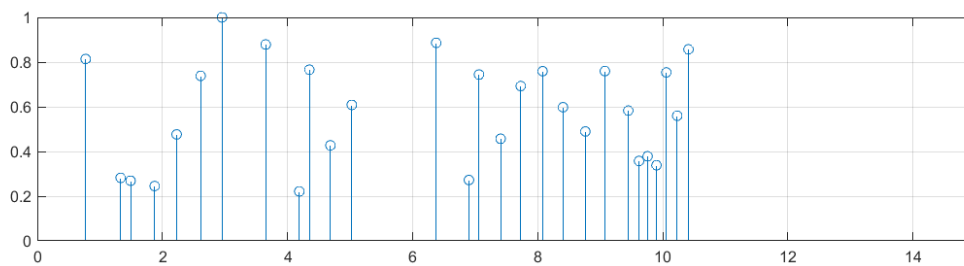
Εικόνα 7.9 Αυτοσυσχέτιση φάσματος πλάτους.

Στη δεύτερη εγγραφή που τρέξαμε στον αλγόριθμο μας και αποτελεί μικρό απόσπασμα από το τραγούδι με τίτλο "άπονη ζωή" έχουμε τα έξις Στην εικόνα 7.5 και 7.6 φαίνεται η κυματομορφή της εγγραφής πριν και μετά τη χρήση της παραμετροποίησης, με διαφοροση της περιβάλλουσας του σήματος, όπως και στο προηγούμενο. Ακολουθεί ο αλγόριθμος παραγωγής του πρότυπου **musicxml** και στην εικόνα 7.7 έχουμε την παρτιτούρα που προκύπτει. Αν τώρα την συγκρίνουμε με την εικόνα 7.8 που είναι η πρωτότυπη παρτιτούρα έχουμε και εδώ απόλυτη ταύτιση εκτός από το 1ο μέτρο στη νότα 10 στην οποία ο αλγόριθμος μας βγάζει τη σωστή νότα κατά μια οκτάβα όμως υψηλότερα. Στην εικόνα 7.9, έχουμε την αυτοσυσχέτιση του φάσματος πλάτους για αυτή την νότα, η οποία είναι η νότα F4 δίοση. (ονομαστική συχνότητα 739.99 Hz). Είναι φανερό πως εκτός από τη συχνότητα 739,99Hz η πρώτη αρμονική της F5 δίοση (1479,98 Hz) υπάρχει και μάλιστα με πολύ υψηλή τιμή αυτοσυσχέτισης. Αυτό εξηγείται με την λογική του οργάνου που παραπάνω είχαμε αναλύσει στο κεφαλαίο 6.2 και είχαμε πει ότι, η ιδιαιτερότητα της κατασκευής και των δίπλων χορδών μας δίνει μερικές φορές υψηλότερη τιμή πλάτους σε κάποια αρμονική, έναντι της βασικής συχνότητας, όπως ακριβώς συμβαίνει στην περίπτωση μας.

### 7.2.3 Εγγραφή 3<sup>η</sup> "Η Ρόζα η Ναζιάρα"



Εικόνα 7.10 Κυματομορφή από την 3η εγγραφή στο πεδίο του χρόνου και τη χρήση της συνάρτησης get.



Εικόνα 7.11 Κυματομορφή με τη χρήση της mirenvelope και διαφορισμού.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE score-partwise PUBLIC "-//Recordare//DTD MusicXML 4.0
Partwise//EN" "http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise version="4.0">
  <part-list>
    <score-part id="P1">
      <part-name>Music</part-name>
    </score-part>
  </part-list>
  <part id="P1">
    <measure number="1">
      <attributes>
        <divisions>8</divisions>
        <key><fifths>0</fifths></key>
        <time><beats>4</beats><beat-type>4</beat-type></time>
        <clef><sign>G</sign><line>2</line></clef>
      </attributes>
      <note><pitch><step>F</step><alter>0</alter><octave>4</octave></pitch>
        <duration>6</duration>
      </note>
    </measure>
  </part>
</score-partwise>
```

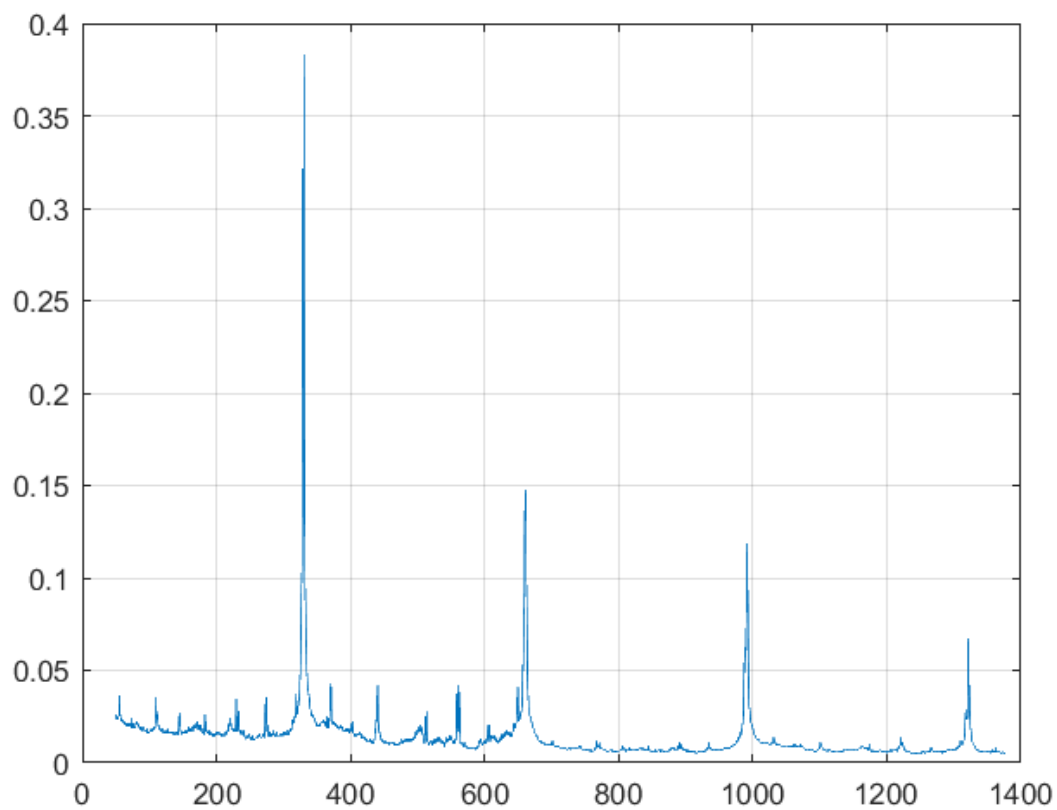


```

</note>
<note><pitch><step>F</step><alter>0</alter><octave>4</octave></pitch>
  <duration>4</duration>
</note>
<note><pitch><step>G</step><alter>0</alter><octave>4</octave></pitch>
  <duration>4</duration>
</note>
<note><pitch><step>G</step><alter>1</alter><octave>4</octave></pitch>
  <duration>4</duration>
</note>
</measure>
<measure number="4">
  <note><pitch><step>G</step><alter>0</alter><octave>4</octave></pitch>
    <duration>4</duration>
  </note>
  <note><pitch><step>F</step><alter>0</alter><octave>4</octave></pitch>
    <duration>2</duration>
  </note>
  <note><pitch><step>G</step><alter>0</alter><octave>4</octave></pitch>
    <duration>2</duration>
  </note>
  <note><pitch><step>F</step><alter>0</alter><octave>4</octave></pitch>
    <duration>2</duration>
  </note>
  <note><pitch><step>E</step><alter>0</alter><octave>4</octave></pitch>
    <duration>2</duration>
  </note>
  <note><pitch><step>C</step><alter>1</alter><octave>4</octave></pitch>
    <duration>2</duration>
  </note>
  <note><pitch><step>F</step><alter>0</alter><octave>4</octave></pitch>
    <duration>2</duration>
  </note>
  <note><pitch><step>E</step><alter>0</alter><octave>5</octave></pitch>
    <duration>16</duration>
    <notations><tied type="start"/></notations>
  </note>
</measure>
<measure number="5">
  <note><pitch><step>E</step><alter>0</alter><octave>5</octave></pitch>
    <duration>32</duration>
    <notations><tied type="start"/></notations>
  </note>
</measure>
<measure number="6">
  <note><pitch><step>E</step><alter>0</alter><octave>5</octave></pitch>
    <duration>32</duration>
    <notations><tied type="stop"/></notations>
  </note>
</measure>
</part>

```

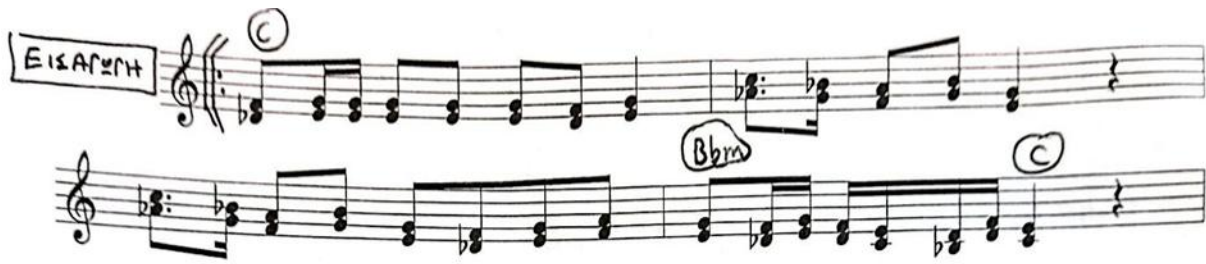
</score-partwise>



Εικόνα 7.12 Αυτοσυσχέτιση φάσματος πλάτους.



Εικόνα 7.13 Παρτιτούρα όπως προκύπτει με την χρήση του αλγορίθμου.



Εικόνα 7.14 Παρτιτούρα που παίχτηκε από τον μουσικό στο μπουζούκι

Στην τρίτη και τελευταία εγγραφή που τρέξαμε στον αλγόριθμο μας και αποτελεί μικρό απόσπασμα από το τραγούδι με τίτλο "η Ρόζα η ναζιάρα" έχουμε τα έξι Στην εικόνα 7.10 και 7.11 φαίνεται η κυματομορφή της εγγραφής πριν και μετά την χρήση της παραμετροποίησης, με διαφόριση της περιβάλλουσες του σήματος, όπως και στο προηγούμενο. Ακολουθεί ο αλγόριθμος παραγωγής του πρότυπου `musicxml` και στην εικόνα 7.13 έχουμε την παρτιτούρα που προκύπτει από τον αλγόριθμο. Αν τώρα την συγκρίνουμε με την εικόνα 7.14 που είναι η πρωτότυπη παρτιτούρα έχουμε και εδώ απόλυτη ταύτιση εκτός από το 1ο μέτρο στη νότα 1, η οποία από λάθος του μουσικού κρατήθηκε 1/16 παραπάνω σε χρόνο, και την οποία ο αλγόριθμος μας βγάζει τη σωστή χρονική διάρκεια. Στην εικόνα 7.12, έχουμε την αυτοσυσχέτιση του φάσματος πλάτους για αυτή τη νότα, η οποία είναι η νότα G4. Επίσης πρέπει να αναφερθεί ότι, στο πρόγραμμα μπορούμε να γράψουμε τις νότες μόνο με δίσηση και όχι με ύφεση, οπότε έχουμε την μετατροπή που βλέπουμε παραπάνω, όπου στην θέση της νότας με ύφεση υπάρχει η ίδια νότα όμως γραμμένη σαν προηγούμενη με δίσηση.

### 7.3 Ποσοστά επιτυχίας και σχολιασμός

Είναι προφανές λοιπόν ότι ο αλγόριθμος επεξεργασίας του ηχητικού μας αρχείου τα πήγε εξαιρετικά στην αναγνώριση και δημιουργία παρτιτούρας σε αυτά τα μικρά πλην δύσκολα μουσικά κομμάτια. Η ταύτιση στις νότες ήταν απόλυτα σωστή και μόνον σε μια περίπτωση είχαμε σωστή νότα σε λανθασμένη οκτάβα. Τα αποτελέσματα του αλγορίθμου μας ικανοποιούν σε μεγάλο βαθμό, γιατί θεωρούμε ότι οι απαιτήσεις μας από την πτυχιακή στο ξεκίνημα της, επιτέθηκαν σε πολύ υψηλό βαθμό. Εδώ πρέπει να αναφερθεί και η διαδικασία κατά την οποία υιοθετήθηκε ο συγκεκριμένος αλγόριθμος που αναλύθηκε παραπάνω. Αρκετές φορές χρειάστηκε να προβούμε σε μικροαλλαγές ή και εντελώς ριζική αλλαγή συναρτήσεων, για να ξεπεραστούν προβλήματα που προέκυπταν στην εξέλιξη. Ξεκινήσαμε με τις εγγραφές των τριών αποσπασμάτων από τα τρία τραγούδια, και στη διάρκεια της πρώτης μας ανάλυσης στο χρόνο, διαπιστώθηκε ότι ο ήχος του μετρονόμου δεν μπορεί να αφαιρεθεί απόλυτα με κάποιο κατάλληλο φίλτρο. Έτσι υπήρχε πρόβλημα στον εντοπισμό των κορυφών. Ξαναγραφήκαν τα κομμάτια με τη χρήση όμως οπτικού μετρονόμου (ρυθμική αλλαγή φωτός στην συχνότητα που θέλαμε) και έτσι ξεπεράστηκε αυτή η κατάσταση. Κατόπιν, στον εντοπισμό των νοτών διαπιστώθηκε ότι προκύπτουν και συχνότητες που δεν είναι χτυπήματα πένας, αλλά θόρυβος που ενισχύεται και μπερδεύοντας τον αλγόριθμο, μεταφράζεται σε νότα που δεν έχει παιχτεί. Αποφασίστηκε λοιπόν να γραφεί μουσικό αρχείο με τις νότες του μπουζουκιού παιγμένες σε χρόνο τετάρτου και μόνο με πένα άρση ή θέση. Έτσι, με εύκολο πλέον τον διαχωρισμό των κορυφών, κάναμε τις κατάλληλες αλλαγές στον αλγόριθμο μας. Παίρνοντας τις έξι μεγαλύτερες συχνότητες μετά την αυτοσυσχέτιση που κάναμε στα κομμάτια που είχε χωριστεί το παραπάνω αρχείο, τρέξαμε ένα κώδικα εντοπισμού σχέσης τονικής και αρμονικών συχνοτήτων. Αυτό μας έδωσε πολύ καλά αποτελέσματα στον εντοπισμό των νοτών και υιοθετήθηκε στον αλγόριθμο μας. Κατόπιν, στη δημιουργία του πρότυπου αρχείου `musicxml` δεν έγινε πλήρη παραμετροποίηση και στην παρτιτούρα

μας δεν υπάρχει στο πρώτο μέτρο οπλισμός. Έτσι στη δημιουργία της παρτιτούρας δεν μπορεί να γραφεί η ύφεση παρά μοναχά η δίεση. Στην προκειμένη περίπτωση αυτό κρίθηκε ασήμαντο γιατί δεν επηρεάζετε σχεδόν καθόλου η ουσία της εργασίας. Επίσης πρέπει να αναφερθεί ότι σκοπίμως απορρίφτηκε εντελώς το τρέμουλο στο παίξιμο των τραγουδιών, ως τεχνική πέννας που υιοθετήθηκε από το μαντολίνο και δεν χαρακτηρίζει το παίξιμο του μπουζουκιού. Τέλος τρόποι παιξίματος που αφορούν διφωνίες με τρίτες και έκτες επίσης αποκλείστηκαν από τα μουσικά μας αρχεία, διότι, είναι πολύ δύσκολη η ανάγνωση κορυφών και η επιλογή των segments, για να γίνει η επεξεργασία στο πεδίο της συχνότητας.

### **7.4 Επίλογος**

Σε αυτό το κεφάλαιο παρουσιάσαμε τα τρία μουσικά αρχεία που ηχογραφήθηκαν στην αρχή και αφού τα τρέξαμε στον αλγόριθμο μας στο τέλος με το πρότυπο Musicxml πήραμε τις παρτιτούρες τις όποιες συγκρίναμε και κατόπιν, σχολιάσαμε τα αποτελέσματα της σύγκρισης, τα οποία μας ικανοποίησαν αρκετά.

## Κεφάλαιο 8<sup>ο</sup> : Συμπεράσματα και προτάσεις βελτίωσης

### 8.1 Συμπεράσματα

Τελειώνοντας την πτυχιακή εργασία θαρρώ ήρθε η ώρα να μοιραστούμε γνώσεις και συμπεράσματα που πρόεκυψαν από την τριβή μας με το θέμα της εξαγωγής παρτιτούρας από ηχητικό περιεχόμενο σε μορφή wav, που ηχογραφήθηκε από μπουζούκι.

Ξεκινήσαμε την εργασία σχεδιάζοντας ένα γενικό διάγραμμα ροής στο οποίο περιγράψαμε προσεγγιστικά τις θεμελιώδεις ενότητες που θα αποτελούσαν τον αλγόριθμο. Γνωρίζαμε ότι το μπουζούκι λόγω της κατασκευής του, και έχοντας διπλές χορδές που παίζονται ταυτόχρονα για να εξαχθεί νότα, θα μας δημιουργούσε κάποιο θέμα στην εύρεση των νοτών λόγω των πολλών αρμονικών και του θορύβου που παράγονται και ενισχύονται υπέρμετρα από την ιδιαιτερότητα της κατασκευής του. Δυστυχώς, όχι απλώς επιβεβαιώθηκαν οι φόβοι μας αυτοί, αλλά επιπλέον, ο τρόπος παίξιματος της πέννας πρόσθεσε ένα ακόμη μεγάλο και δύσκολο πρόβλημα. Η τονική συχνότητα τις περισσότερες φορές δεν ήταν η δυνατότερη σε πλάτος όπως ευλόγως περιμέναμε, αλλά αποτελούσε τη δεύτερη ή ακόμη και την τρίτη κατά σειρά μεγέθους πλάτους συχνότητα εντοπισμού. Το πρώτο πρόβλημα το ξεπεράσαμε κατά την επεξεργασία του αρχείου μας στο πεδίο του χρόνου, με κατάλληλη επεξεργασία της περιβάλλουσας τους σήματος. Αλλάζοντας συνεχώς την παραμετροποίηση καταφέραμε και εξαγάγαμε τις κορυφές που αντιστοιχούσαν σε χτυπήματα πέννας. Κατόπιν τούτου, για την εύρεση της χρονικής αξίας των νοτών δεν αντιμετωπίσαμε κάποιο ιδιαίτερο πρόβλημα. Αυτό έγινε γιατί, κατά πρώτον είχαμε άπλα μέτρα στα μουσικά μας αρχεία, που δεν δυσκόλευαν τις συναρτήσεις μας στην εύρεση των χρονικών αξιών, και κατά δεύτερον στη διάρκεια εγγραφής των αρχείων εφαρμόσαμε χρησιμοποιήσαμε κατευθυντικό μικρόφωνο καλής ποιότητας και μετρονόμο πολύ μικρού θορύβου, που λειτουργούσε και οπτικά.

Το επόμενο θέμα της εύρεσης της κατάλληλης συχνότητας, το αντιμετωπίσαμε στην επεξεργασία του αρχείου μας στο πεδίο της συχνότητας. Εκεί διαπιστώθηκε ότι εκτός της συχνότητας που είχε παιχτεί, υπήρχαν πολλές αρμονικές με πολύ μεγάλο πλάτος και παρασιτικές συχνότητες (πιθανόν θόρυβος) που και αυτές ήταν πολύ μεγάλες σε πλάτος. Για να το ξεπεράσουμε κάναμε καταρχήν τον μετασχηματισμό Fourier σε κάθε τμήμα σήματος που είχαμε υπολογίσει στο πεδίο του χρόνου. Μετά υπολογίσαμε τη συνάρτηση αυτοσυσχέτισης του φάσματος πλάτους και ενισχύσαμε κατά πολύ τις κορυφές όλων των συχνοτήτων που βρίσκαμε. Κατόπιν, γράψαμε τον αλγόριθμο έτσι ώστε να επιλέγονται οι έξι συχνότητες με τα μεγαλύτερα πλάτη ανά τμήμα, και να ελέγχονται μεταξύ τους αν τις συνδέει σχέση αρμονικών συχνοτήτων. Αφού εντοπιζόταν οι αρμονικές, κρατούσαμε αυτήν που εμφανίζει τις περισσότερες αρμονικές σαν την τονική συχνότητα που είχε παιχτεί. Κατά αυτό τον τρόπο καταφέραμε και εντοπίσαμε με μεγάλη επιτυχία τις νότες που είχαν παιχτεί από τον μουσικό.

Έχοντας τελειώσει με την εύρεση των πληροφοριών νότας, οκτάβας και χρονικής αξίας προχωρήσαμε στο επόμενο βήμα, που ήταν η δημιουργία της βασικής παρτιτούρας. Χωρίς δυσκολία υιοθετήθηκε για τη δημιουργία της παρτιτούρας το πρότυπο **MusicXML**. Αυτό έγινε γιατί το συγκεκριμένο πρότυπο είναι ανοικτού κώδικα και υποστηρίζει την αξιόπιστη εισαγωγή και εξαγωγή δεδομένων παρτιτούρας. Επίσης, υποστηρίζεται από όλα τα γνωστά προγράμματα σημειογραφίας της μουσικής. Τέλος με την φόρτωση του εξαγόμενου από το πρόγραμμα αρχείο musicxml στο λογισμικό Musescore, καταφέραμε να τυπώσουμε τις παρτιτούρες των μουσικών μας αρχείων. Ο βαθμός ικανοποίησης μας είναι μεγάλος γιατί, καταφέραμε σαν αποτέλεσμα να έχουμε ταυτόσημες παρτιτούρες με τις αρχικές στις περισσότερες των περιπτώσεων. Με σιγουριά θα λέγαμε πάντως ότι η

επεξεργασία μελωδίας από φυσικά όργανα είναι δύσκολη εργασία, απαιτεί συνδυασμό θεωρητικών και πρακτικών γνώσεων και πολύ όρεξη για δουλειά. Επίσης το αποτέλεσμα ίσως αποτελέσει την ευκαιρία να ασχοληθώ στο μέλλον περισσότερο με το θέμα του εντοπισμού των χρονικών αξιών και την αναγνώριση των νοτών, σε πιο περίπλοκα μουσικά θέματα.

### **8.2 Προτάσεις βελτίωσης**

Συμπερασματικά, η χρήση και ο συνδυασμός των κατάλληλων συναρτήσεων μας επέτρεψαν να επιτύχουμε ικανοποιητικά αποτελέσματα στην εξαγωγή μελωδίας από ηχητικό περιεχόμενο μπουζουκιού, ωστόσο, τα αποτελέσματα αυτά είναι περιορισμένα σε απλά μουσικά κομμάτια που περιέχουν μία κύρια μελωδία. Είναι σαφές ότι αν παιχτεί ολόκληρο το κομμάτι πολυφωνικά από πλήρη ορχήστρα τα πράγματα δυσκολεύουν αφάνταστα. Θα πρέπει, εκτός του εντοπισμού των νοτών, να γίνει και ο εντοπισμός των συγχορδιών και ίσως και της μπασογραμμής. Επιπλέον, θα χρειαστεί περισσότερη επεξεργασία στον εντοπισμό των χρονικών αξιών, ιδίως αν πρόκειται για δεξιοτεχνικά μουσικά κομμάτια, που έχουν πολύ μεγάλες εναλλαγές σε χρόνους ή είναι πολύ γρήγορα παιγμένα. Τέλος θα ήθελα να αναφερθώ και σε τεχνικές παιξίματος που σκόπιμως δεν μελετηθήκαν. Κάποιες από αυτές είναι το τρέμουλο και οι μαντολινάτες διφωνίας με τρίτες και έκτες. Αυτό θα δημιουργούσε μεγάλη δυσκολία στον εντοπισμό της περιβάλλουσας σήματος και κατ' επέκταση στην επιλογή των χρονικών τμημάτων (και της αντίστοιχης αξίας κάθε νότας) που θα πήγαιναν για επεξεργασία στο πεδίο της συχνότητας. Από την εμπειρία που αποκομίσαμε, αυτό το θέμα αποτελεί και την βασικότερη πρόταση για βελτίωση και επέκταση της εργασίας στο μέλλον.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] “Free music composition and notation software | MuseScore,” Musescore.org, 2025. [Online]. Available: <https://musescore.org/el>. Accessed: Dec. 20, 2024.
- [2] “music21 Documentation — music21 Documentation,” Music21.org. [Online]. Available: <https://www.music21.org/music21docs/index.html> Accessed: Dec. 19, 2024.
- [3] “Cecilia - Graphic User Interface to Csound,” Sourceforge.net, 2025. [Online]. Available: <https://cecilia.sourceforge.net/>. Accessed: Dec. 21, 2024.
- [4] “Ardour DAW”. [Online]. Available: <https://ardour.org/>. Accessed: Dec. 21, 2024.
- [5] “MusicXML for Exchanging Digital Sheet Music,” MusicXML. [<https://www.musicxml.com/>]. Accessed: Dec. 21, 2024.
- [6] Wikipedia, “XML”. [Online]. Available: <https://el.wikipedia.org/wiki/XML>. Accessed: Dec. 10, 2024.
- [7] “Μπουζούκι ... το ελληνικό,” Blogspot.com, 2022. [Online]. Available: <https://free-bouzouki-greece.blogspot.com/>. Accessed: Nov. 26, 2025.
- [8] Βικιπαιδεία, “Μπουζούκι”. [Online]. Available: <https://el.wikipedia.org/wiki/Μπουζούκι>. Accessed: Nov. 26, 2025.
- [9] Γ. Λεκάκης, “Η πανδούρα των αρχαίων,” Τα Νέα, 11 Νοεμβρίου, 2011.
- [10] Βικιπαιδεία Μπουζούκι 2011 [Online]. Available <https://el.wikipedia.org/wiki/CE%23>, 2018
- [11] Β. Τρίγκας, *Μέθοδος για τρίχορδο μπουζούκι, τόμος 1*. Ναύπλιο, εκδ. Β. Τρίγκα, 2018.
- [12] Χ. Λέκκας, *Η θεωρία της μουσικής 1*. Ανοιχτή Βιβλιοθήκη. [Online]. Available: <https://www.openbook.gr/i-theoria-tis-mousikis/>. Accessed: Nov. 20, 2024.
- [13] “Πρακτική μέθοδος για μπουζούκι,” Kallitexnorama.gr. [Online]. Available: <https://kallitexnorama.gr/mpouzouki/tetraxordo.htm>. Accessed: Nov. 20, 2024.
- [14] Φ. Λαζαρίνης, *Πολυμέσα. Κάλλιπος – Ανοικτές Ακαδημαϊκές Εκδόσεις*, 2015. [Online]. Available: <https://repository.kallipos.gr/handle/11419/2045>
- [15] Ρ. Κωτσάκης, “Βασικά στοιχεία ακουστικής”, Σημειώσεις μαθήματος *Συστήματα Μέσων Μαζικής Επικοινωνίας*, Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων, ΔΙΠΑΕ, Θεσσαλονίκη, 2023.
- [16] Α. R Moller, *Hearing: Anatomy, Physiology, and Disorders of the Auditory System*. Plural Publishing, 2012.
- [17] T. Dittmar, *Audio Engineering 101: A Beginner’s Guide to Music Production*. New York, Abingdon: Routledge, 2018.
- [18] Blu-ray Master. Συμπύεση ήχου: Πλήρης γνώση και συμβουλής για την συμπύεση του ήχου 2023 <https://www.bluraypys.com/images/article/logo.svg>
- [19] Ρ. Κωτσάκης, “Ψηφιοποίηση-Κωδικοποίηση Ήχου,” Σημειώσεις μαθήματος *Τεχνολογία Ήχου και Εικόνας*, Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων, ΔΙΠΑΕ, Θεσσαλονίκη, 2023.

- [20] X. Ξενοφώντος, *Οδηγός MATLAB για αρχάριους*. Πανεπιστήμιο Κύπρου, Τμήμα Μαθηματικών και Στατιστικής. [Online]. Available: [https://www.mas.ucy.ac.cy/xenophon/pubs/matlab\\_intro.pdf](https://www.mas.ucy.ac.cy/xenophon/pubs/matlab_intro.pdf)
- [21] O. Lartillot, *MIRtoolbox 1.6.3 User's Manual*. Apr. 7th, 2017.

## ΠΑΡΑΡΤΗΜΑ: Αλγόριθμος επεξεργασίας Matlab

### Αλγόριθμος εγγραφής αρχείου ήχου από μπουζούκι

```
clear
close all
clc
fs = 44100; nbits=16;
audio = audiorecorder(fs,nbits,1);
recDuration = 15;
user_input = input('Press a key to play and record');
recordblocking(audio,recDuration);
disp("End of recording.")

bouzouki_data = getaudiodata(audio);
sound(bouzouki_data,fs);
audiowrite('testROZA3.wav',bouzouki_data,fs);
```

### Αλγόριθμος επεξεργασίας αρχείου wav και εξαγωγή παρτιτούρας σε μορφή musicxml

```
clear
close all
clc

NoteFreqs = [16.35, 32.70, 65.41, 130.81, 261.63, 523.25, 1046.50,
2093.00, 4186.01;
             17.32, 34.65, 69.30, 138.59, 277.18, 554.37, 1108.73,
2217.46, 4434.92;
             18.35, 36.71, 73.42, 146.83, 293.66, 587.33, 1174.66,
2349.32, 4698.63;
             19.45, 38.89, 77.78, 155.56, 311.13, 622.25, 1244.51,
2489.02, 4978.03;
             20.60, 41.20, 82.41, 164.81, 329.63, 659.25, 1318.51,
2637.02, 5274.04;
             21.83, 43.65, 87.31, 174.61, 349.23, 698.46, 1396.91,
2793.83, 5587.65;
             23.12, 46.25, 92.50, 185.00, 369.99, 739.99, 1479.98,
2959.96, 5919.91;
             24.50, 49.00, 98.00, 196.00, 392.00, 783.99, 1567.98,
3135.96, 6271.93;
             25.96, 51.91, 103.83, 207.65, 415.30, 830.61, 1661.22,
3322.44, 6644.88;
             27.50, 55.00, 110.00, 220.00, 440.00, 880.00, 1760.00,
3520.00, 7040.00;
             29.14, 58.27, 116.54, 233.08, 466.16, 932.33, 1864.66,
3729.31, 7458.62;
             30.87, 61.74, 123.47, 246.94, 493.88, 987.77, 1975.53,
3951.07, 7902.13];
```

```

Notes = {'C','C#','D','D#','E','F','F#','G','G#','A','A#','B'};
Octaves = 0:8;

% xfile = 'testROZA_BOUZ2.wav';
% xfile = 'testAPONIZOI.wav';
% xfile = 'testHARAMATA.wav';
xfile = 'testZILIARIKA.wav';

fs = 44100;
bpm = 85;
rythmnom = 4;
rythmdenom = 4;

quarter = 60/bpm; % a quarter
eighth = quarter/2;
sixteenth = quarter/4;
thirtysecond = quarter/8;
sixtyfourth = quarter/16;
LeastValue = thirtysecond;
divisions = 8; % 4 for sixteenth, 8 for thirtysecond;

freqValue = [];
freqRes = 22050/65536;
% FreqPosInit = [0,(1:65536)*freqRes];

InMusic = miraudio(xfile)
aa = get(InMusic,'Data');
tt = get(InMusic,'Pos');
Amp = aa{1,1}{1,1};
Time = tt{1,1}{1,1};

testxx = mirenelope(InMusic,'HalfWaveDiff','Normal','PostDecim',1);

FirstPeaks = mirpeaks(testxx,'Contrast',0.05,'Reso',LeastValue,'Only');
PAmp = get(FirstPeaks,'Data');
PPos = get(FirstPeaks,'Pos');
peakamptemp = PAmp{1,1}{1,1};
peakpos = find(peakamptemp);
peakamps = peakamptemp(peakpos);
%%%%%
zz = find(peakamps < 0.1);
peakpos(zz) = [];
peakamps(zz) = [];
%%%%%%%%
peaktimes = peakpos/fs;

segtimes = [peaktimes;Time(end)]-0.025;

```

```

figure
stem(peaktimes,peakamps);
xlim([0 Time(end)]);

for ii = 1:length(segtimes)-1
    segDiffs(ii) = segtimes(ii+1)-segtimes(ii);
end
for ii = 1:length(segtimes)-1
    durations(ii) = 2*round(segDiffs(ii)/(2*LeastValue));
end

for ii = 1:length(segtimes)-1
    Seg
miraudio(xfile,'Extract',segtimes(ii),segtimes(ii+1),'TrimThreshold',0.
2);
    SegFreqs = mirspectrum(Seg,'MinRes',0.5,'Normal');
    fv = get(SegFreqs,'Data');
    fp = get(SegFreqs,'Pos');
    fvnew = fv{1,1}{1,1};
    if length(fvnew)>65537
        fvnew(65538:end)=[];
    end
    freqValue(ii,:) = fvnew.';    % spectrum of note played

end
freqPos = fp{1,1}{1,1}.';

for ii = 1:length(segtimes)-1
    Seg
miraudio(xfile,'Extract',segtimes(ii),segtimes(ii+1),'TrimThreshold',0.
2);
    SegFreqs = mirspectrum(Seg,'MinRes',0.5,'Normal');
    fv = get(SegFreqs,'Data');
    fp = get(SegFreqs,'Pos');
    fvnew = fv{1,1}{1,1};
    if length(fvnew)>65537
        fvnew(65538:end)=[];
    end
    freqValue(ii,:) = fvnew.';    % spectrum of note played

end
freqPos = fp{1,1}{1,1}.';

PeakHeight =[];
for ii = 1:length(segtimes)-1

```

```

PeakHeight(ii) = 0.025;
C(ii,:) = xcorr(freqValue(ii,:));
C1(ii,:) = C(ii,65537:65537+8192);
figure
plot(freqPos(300:8193),C1(ii,300:8193));
grid
F_low = floor(100/freqRes);% Low frequency in number of samples
[allpeaks,locs] =
findpeaks(C1(ii,F_low:end),'MinPeakHeight',PeakHeight(ii),'MinPeakDistance',15);
tempFreqs = (F_low+locs-2)*freqRes;

while length(locs) > 6
    PeakHeight(ii) = PeakHeight(ii) + 0.01;
    [allpeaks,locs] =
findpeaks(C1(ii,F_low:end),'MinPeakHeight',PeakHeight(ii),'MinPeakDistance',15);
end
tempFreqs = (F_low+locs-2)*freqRes;
tempAmps = C1(ii,F_low+locs-1);

estfreqs{ii,:} = tempFreqs;

CurrentFreqs = estfreqs{ii,:};
ll = 0;
pp = 0;
PossibleFreq = [];
for jj = 1:length(CurrentFreqs)-1
    testFreq = CurrentFreqs(jj);

    mm = 0;
    testmat = CurrentFreqs;
    testmat(jj) = [];
    for kk = jj+1:length(CurrentFreqs)
        CheckThreshold = 0.03*testFreq;
        testrem = abs(CurrentFreqs(kk) -
round(CurrentFreqs(kk)/testFreq)*testFreq);
        if round(CurrentFreqs(kk)/testFreq) > 1 && testrem <
CheckThreshold
            mm = mm+1;
        end
    end
    if mm > 0
        pp = pp+1;
        PossibleFreq(pp,1) = jj;
        PossibleFreq(pp,2) = testFreq;
        PossibleFreq(pp,3) = tempAmps(jj);
        PossibleFreq(pp,4) = mm;
    end
end

```

```

        end
    end
    if isempty(PossibleFreq)
        FoundTemp = 0;
        while FoundTemp < 130 || FoundTemp > 1500
            [maxF,locF] = max(allpeaks);
            FoundTemp = tempFreqs(locF);
            allpeaks(locF) = [];
        end
        FoundFreq{ii,1} = FoundTemp;
        if FoundTemp == 0
            [maxF,locF] = max(allpeaks);
            FoundFreq{ii,1} = tempFreqs(locF);
        end
    else
        [maxF,locF] = max(PossibleFreq(:,4));
        FoundFreq{ii,1} = PossibleFreq(locF,2);
    end
end

end

NumberofNotesFound = length(segtimes)-1;
kk = 0;
for ii = 1:length(segtimes)-1
    tempFreqs = FoundFreq{ii};
    if isempty(tempFreqs)
        NumberofNotesFound = NumberofNotesFound-1;
    else
        for jj = 1:length(tempFreqs)
            [xx,yy] = min(abs(tempFreqs(jj)-NoteFreqs));
            [zz,vv] = min(xx);
            Oct{ii,jj} = vv-1;
            ResNote{ii,jj} = yy(vv);
            kk = kk+1;
            InitInstants(kk) = segtimes(ii);
        end
    end
end

end

for ii = 1:NumberofNotesFound
    OctFinal = Oct{ii,1};
    NoteFinal = ResNote{ii,1};
    tr{ii,1} = OctFinal;
    tr{ii,2} = Notes{NoteFinal};
    tr{ii,3} = durations(ii);
end

```

```

    fprintf("%d. Note = %s%d, Time = %5.3f, Duration = %d \n",ii,
Notes{NoteFinal},OctFinal,InitInstants(ii),durations(ii));
end

rnom = rythmnom;
rdenom = rythmdenom;
divs = divisions;
transcript = tr;
notestable = ['C','D','E','F','G','A','B'];

measurelength = rnom*divs;
for ii = 1:size(transcript,1)
    Octave(ii) = transcript{ii,1};
    Note{1,ii} = transcript{ii,2};
    dur(ii) = transcript{ii,3};
end
alter = zeros(1,length(Note));

for ii = 1:length(Note)
    st = Note{1,ii};
    fnote(ii) = find(st(1)==notestable);
    if strlen(st) == 2
        if st(2)=="#"
            alter(ii) = 1;
        else
            alter(ii) = -1;
        end
    end
end
end

totalDuration = sum(dur);
numofMeasures = ceil(totalDuration/measurelength);
tot_dur = cumsum(dur);
dur_temp = dur;
ll = 0;
ties = zeros(1,length(fnote));
for jj = 1:numofMeasures-1
    ds = 0;
    kk = 0;
    while ds < measurelength
        kk = kk+1;
        ll = ll+1;
        if ll <= length(dur_temp)
            ds = ds + dur_temp(ll);
        end
    end
end
if ds > measurelength
    tomove = ds - measurelength;

```

```

    tokeep = dur_temp(ll) - tomove;
    dur_temp(ll) = tokeep;
    dur_temp = [dur_temp(1:ll), tomove, dur_temp(ll+1:end)];
    Octave = [Octave(1:ll), Octave(ll), Octave(ll+1:end)];
    fnote = [fnote(1:ll), fnote(ll), fnote(ll+1:end)];
    alter = [alter(1:ll), alter(ll), alter(ll+1:end)];
    if ll == length(dur_temp)-1
        ties = [ties(1:ll-1), 1, 2];
    else
        ties = [ties(1:ll-1), 1, 2, ties(ll+1:end)];
    end
end
notesperMeas(jj) = kk;
end
% last measure
notesperMeas(numofMeasures) = length(dur_temp)-ll;
lastmeasurelength = sum(dur_temp(length(dur_temp)-
notesperMeas(numofMeasures)+1:length(dur_temp)));
if lastmeasurelength < measurelength
    xx = measurelength - lastmeasurelength;
    dur_temp(end) = dur_temp(end) + xx;
end

fid = fopen('ZILIARIKA.musicxml','w');
text = '<?xml version="1.0" encoding="UTF-8" standalone="no"?>';
fprintf(fid,text); fprintf(fid,"\n");
text = '<!DOCTYPE score-partwise PUBLIC "-//Recordare//DTD MusicXML 4.0
Partwise//EN" "http://www.musicxml.org/dtds/partwise.dtd">';
fprintf(fid,text); fprintf(fid,"\n");
text = '<score-partwise version="4.0">';
fprintf(fid,text); fprintf(fid,"\n");
text = ' <part-list>';
fprintf(fid,text); fprintf(fid,"\n");
text = ' <score-part id="P1">';
fprintf(fid,text); fprintf(fid,"\n");
text = ' <part-name>Music</part-name>';
fprintf(fid,text); fprintf(fid,"\n");
text = ' </score-part>';
fprintf(fid,text); fprintf(fid,"\n");
text = ' </part-list>';
fprintf(fid,text); fprintf(fid,"\n");
text = ' <part id="P1">';
fprintf(fid,text); fprintf(fid,"\n");

fprintf(fid,' <measure number="%d">',1); fprintf(fid,"\n");
fprintf(fid,' <attributes>'); fprintf(fid,"\n");
fprintf(fid,' <divisions>%d</divisions>',divs);
fprintf(fid,"\n");

```

```

fprintf(fid, '                                <key><fifths>0</fifths></key>');
fprintf(fid, "\n");
fprintf(fid, '                                <time><beats>%d</beats><beat-type>%d</beat-
type></time>', rnom, rdenom); fprintf(fid, "\n");
fprintf(fid, '
<clef><sign>G</sign><line>2</line></clef>'); fprintf(fid, "\n");
fprintf(fid, '    </attributes>'); fprintf(fid, "\n");
for jj = 1:notesperMeas(1)
    fprintf(fid, '
<note><pitch><step>%c</step><alter>%d</alter><octave>%d</octave></pitch
>', notestable(fnote(jj)), alter(jj), Octave(jj)); fprintf(fid, "\n");
    fprintf(fid, '                                <duration>%d</duration>', dur_temp(jj));
fprintf(fid, "\n");
    if ties(jj)==1
        fprintf(fid, '                                <notations><tied
type="start"/></notations>'); fprintf(fid, "\n");
    end
    if ties(jj)==2
        fprintf(fid, '                                <notations><tied
type="stop"/></notations>'); fprintf(fid, "\n");
    end
    fprintf(fid, '    </note>'); fprintf(fid, "\n");
end
fprintf(fid, '    </measure>'); fprintf(fid, "\n");
fnote(1:notesperMeas(1))=[];
alter(1:notesperMeas(1))=[];
Octave(1:notesperMeas(1))=[];
dur_temp(1:notesperMeas(1))=[];
ties(1:1:notesperMeas(1))=[];

for ii = 2:numofMeasures
    fprintf(fid, '    <measure number="%d">', ii); fprintf(fid, "\n");
    for jj = 1:notesperMeas(ii)
        fprintf(fid, '
<note><pitch><step>%s</step><alter>%d</alter><octave>%d</octave></pitch
>', notestable(fnote(jj)), alter(jj), Octave(jj)); fprintf(fid, "\n");
        fprintf(fid, '                                <duration>%d</duration>', dur_temp(jj));
fprintf(fid, "\n");
        if ties(jj)==1
            fprintf(fid, '                                <notations><tied
type="start"/></notations>'); fprintf(fid, "\n");
        end
        if ties(jj)==2
            fprintf(fid, '                                <notations><tied
type="stop"/></notations>'); fprintf(fid, "\n");
        end
        fprintf(fid, '    </note>'); fprintf(fid, "\n");
    end
end

```

```
fprintf(fid,' </measure>'); fprintf(fid,"\n");
fnote(1:notesperMeas(ii))=[];
alter(1:notesperMeas(ii))=[];
Octave(1:notesperMeas(ii))=[];
dur_temp(1:notesperMeas(ii))=[];
ties(1:1:notesperMeas(ii))=[];
end

fprintf(fid,' </part>'); fprintf(fid,"\n");
fprintf(fid,'</score-partwise>');
fclose(fid);
```