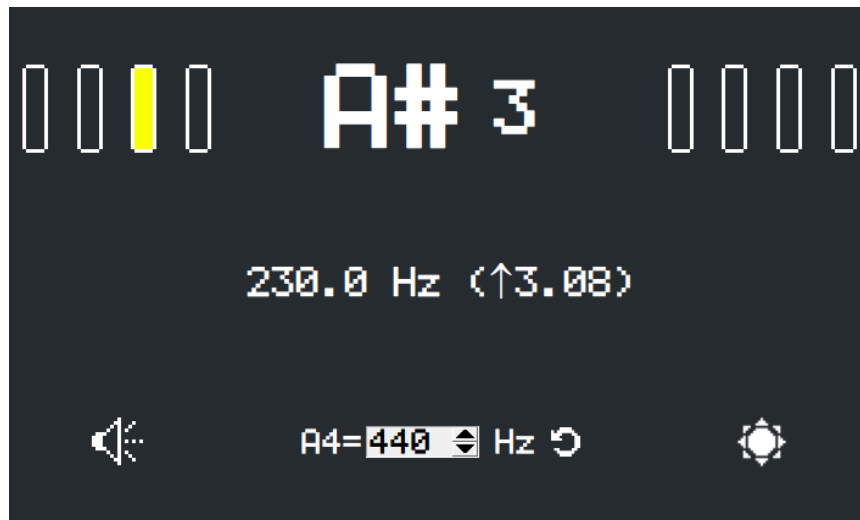


# FineTuned

Ένα Χρωματικό Κουρδιστήρι Υλοποιημένο σε Python



Ευάγγελος Δημητριάδης (1115201700287)

Διονύσιος Μανιατάκος (1115201800104)

Μουσική Πληροφορική

Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

06/2022

## Περίληψη

Σκοπός της παρούσας εργασίας είναι η ανάπτυξη μιας εφαρμογής που μπορεί να χρησιμοποιηθεί ως μουσικό κουρδιστήρι. Εξετάζουμε την είσοδο και ανάγνωση του ήχου, την εξαγωγή του συχνотικού του περιεχομένου, την ανίχνευση της στοιχειώδους συχνότητας και την αντιστοίχιση της στην πιο κοντινή νότα. Γίνεται αναφορά σε ορισμένες προκλήσεις που προκύπτουν κατά την υλοποίηση των συγκεκριμένων διαδικασιών αλλά και σε παραμετροποιήσεις και τεχνικές που μπορούν να συμβάλουν στην αντιμετώπιση τους. Τέλος, παρουσιάζουμε ένα γραφικό περιβάλλον που μπορεί να χρησιμοποιηθεί ως διεπαφή, προσφέροντας πλήθος ενδείξεων και παραμετροποιήσεων, δίνοντας τη δυνατότητα στον χρήστη να χρησιμοποιήσει την εφαρμογή σε ένα πραγματικό σενάριο κουρδίσματος ενός μουσικού οργάνου.

**Πίνακας Περιεχομένων**

<b>Περίληψη</b>	<b>1</b>
<b>Εισαγωγή</b>	<b>3</b>
<b>Είσοδος/Ανάγνωση Ήχου</b>	<b>4</b>
<b>Ανίχνευση Θεμελιώδους Συχνότητας</b>	<b>4</b>
<b>Ανίχνευση Πλησιέστερης Νότας</b>	<b>6</b>
<b>Γραφική Διεπαφή Χρήστη (GUI)</b>	<b>8</b>
<b>Ανάπτυξη GUI</b>	<b>9</b>
<b>Οδηγίες Εγκατάστασης</b>	<b>9</b>
<b>Δοκιμή/Επίδειξη Εφαρμογής</b>	<b>10</b>
<b>Μελλοντικές Προσθήκες/Βελτιώσεις</b>	<b>10</b>
<b>Επίλογος</b>	<b>12</b>
<b>Αναφορές</b>	<b>13</b>

## Εισαγωγή

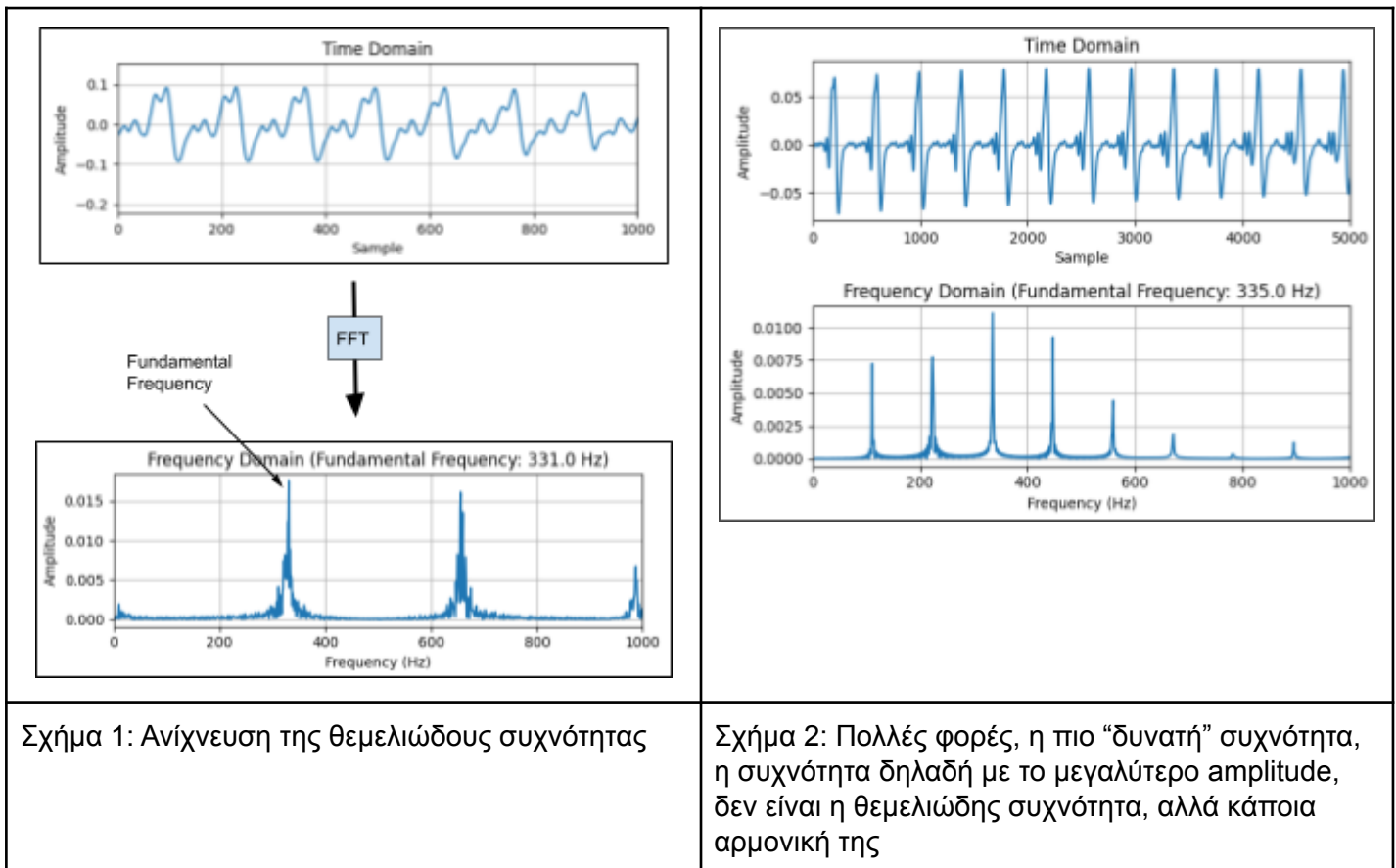
Με τον όρο κούρδισμα αναφερόμαστε στην ρύθμιση του τονικού ύψους που παράγουν τα μέλη ενός μουσικού οργάνου (όπως οι χορδές σε ένα έγχορδο όργανο) ώστε να επιτυγχάνονται τα καθιερωμένα διαστήματα μεταξύ των νοτών και το αποτέλεσμα να ακούγεται εύηχο. Υπάρχουν κάποια ευρέως υιοθετημένα πρότυπα κουρδίσματος τα οποία χρησιμοποιούν οι μουσικοί ώστε τα όργανα μιας ορχήστρας ή ενός μουσικού συνόλου να βρίσκονται σε συμφωνία μεταξύ τους, όσον αφορά τις συχνότητες που παράγουν [1]. Το βασικότερο εργαλείο το οποίο χρησιμοποιείται κατά τη διάρκεια του κουρδίσματος ενός οργάνου, είναι το μουσικό κουρδιστήρι. Ένα μουσικό κουρδιστήρι είναι μία ηλεκτρονική συσκευή η οποία ανιχνεύει και εμφανίζει στον χρήστη το τονικό ύψος του ήχου που παράγεται από ένα μουσικό όργανο, υποδεικνύοντάς του την κατεύθυνση στην οποία πρέπει να κουρδίσει προκειμένου να φέρει το όργανο στο σωστό κούρδισμα. Ο χρήστης, στη συνέχεια, ακολουθεί τις οδηγίες της συσκευής έως ότου πετύχει το σωστό τονικό ύψος, γεγονός που μπορεί να αντιληφθεί από την ανάλογη ένδειξη. Αν και τα μουσικά κουρδιστήρια έχουν, κυρίως, τη μορφή μιας ανεξάρτητης ηλεκτρονικής συσκευής, την τελευταία δεκαετία η ραγδαία αύξηση της χρήσης των φορητών συσκευών (όπως έξυπνα τηλέφωνα και ταμπλέτες) έχουν οδηγήσει σε μεγάλο πλήθος εφαρμογών κινητού που προσομοιώνουν την λειτουργία ενός ηλεκτρονικού μουσικού κουρδιστηριού [2]. Παράλληλα, διάφορες εφαρμογές ήχου, όπως DAWs (Digital Audio Workstations) και Audio Plugins, εντάσσουν λειτουργίες κουρδιστηριών, ώστε να μην χρειάζεται ο χρήστης να καταφύγει στην αγορά ή χρήση κάποιας ανεξάρτητης συσκευής. Προκειμένου να μπορεί να γίνει κατανοητός ο τρόπος που λειτουργεί ένα κουρδιστήρι, είναι σημαντικό να αναφέρουμε την έννοια της θεμελιώδους συχνότητας και των αρμονικών συχνοτήτων. Ο περιοδικός ήχος που παράγεται από ένα όργανο (παραδείγματος χάρη από μία χορδή στην κιθάρα), δεν αποτελείται από μία μόνο συχνότητα, αλλά από μία σύνθεση διαφορετικών συχνοτήτων. Πιο συγκεκριμένα, η χαμηλότερη συχνότητα που παράγεται καλείται θεμελιώδης συχνότητα και οι υπόλοιπες καλούνται αρμονικές συχνότητες. Οι αρμονικές, αυτές, συχνότητες βρίσκονται σε πολλαπλάσια της θεμελιώδους συχνότητας [3]. Παραδείγματος χάρη, αν η θεμελιώδης συχνότητα βρίσκεται στα 440 Hz, η πρώτη αρμονική θα βρίσκεται στα 880 Hz, η δεύτερη αρμονική στα 1320 Hz και ούτω καθεξής. Αν αντιστοιχίσουμε τις συχνότητες αυτές σε νότες, παρατηρούμε ότι η θεμελιώδης συχνότητα είναι η νότα A4, η πρώτη αρμονική είναι η ίδια νότα μία οκτάβα πιο ψηλά (A5) και η δεύτερη αρμονική είναι προσεγγιστικά η νότα (E6). Αν και η μελέτη των αρμονικών συχνοτήτων έχει ιδιαίτερο ενδιαφέρον, ένα συμβατικό μουσικό κουρδιστήρι εστιάζει στην θεμελιώδη συχνότητα, που είναι και η συχνότητα που αντιλαμβάνεται το ανθρώπινο αυτί.

## Είσοδος/Ανάγνωση Ήχου

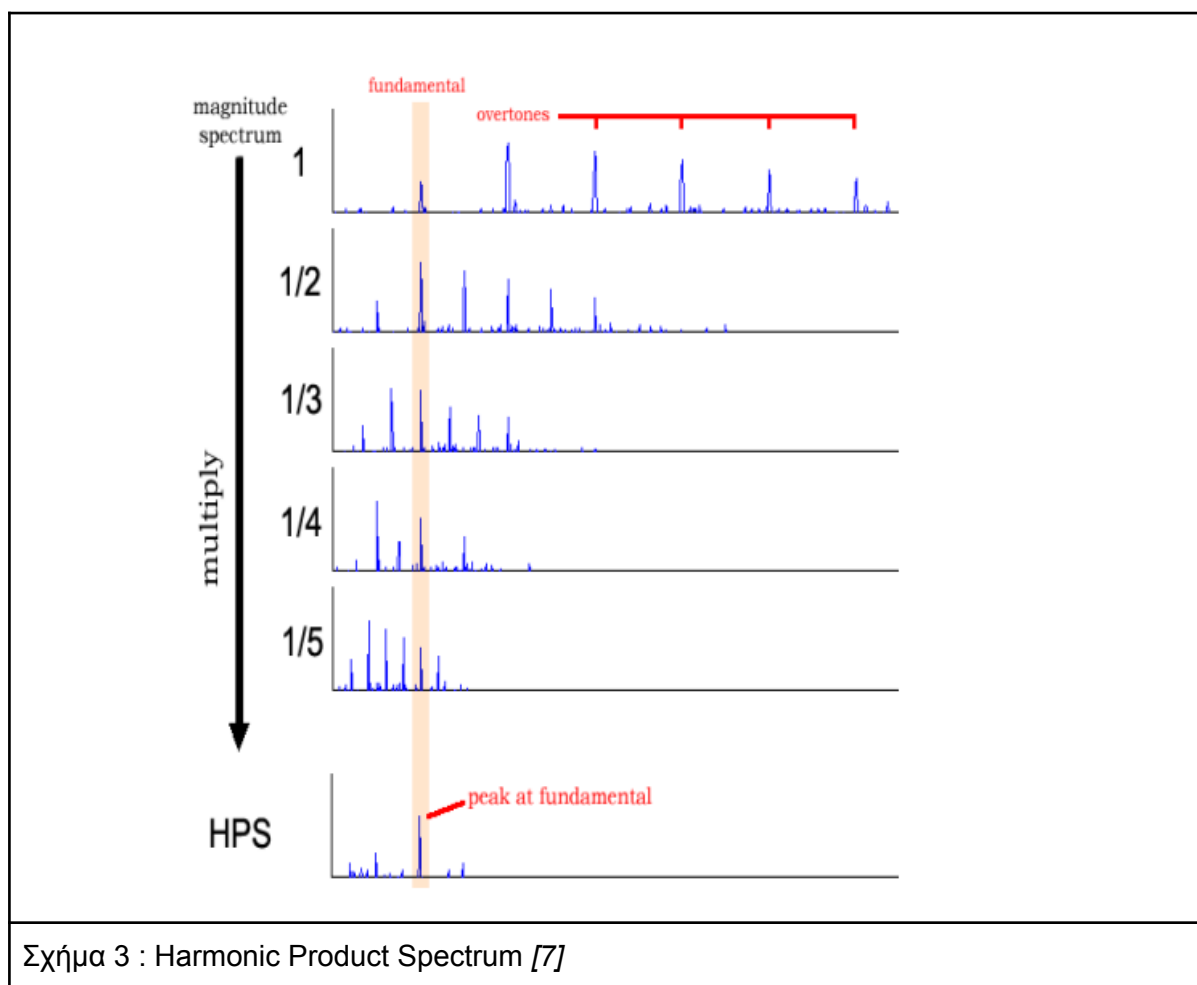
Προκειμένου να μπορούμε να αναλύσουμε τον ήχο που παράγεται από ένα μουσικό όργανο και να εξαγάγουμε τη θεμελιώδη συχνότητα του, χρειάζεται πρώτα να έχουμε το σήμα σε ψηφιακή μορφή. Από τη στιγμή που η εφαρμογή μας είναι μια εφαρμογή πραγματικού χρόνου, αυτή η διαδικασία πρέπει να γίνεται επαναλαμβανόμενα. Πιο συγκεκριμένα, καθώς ο χρήστης προσπαθεί να κουρδίσει ένα όργανο, παράγει συνεχώς ήχους, κάθε ένας από τους οποίους διαφέρει από τον προηγούμενο, όσον αφορά το τονικό ύψος. Η εφαρμογή πρέπει να είναι σε θέση να διαβάζει ένα μικρό “κομμάτι” από την είσοδο (χρησιμοποιώντας μία συσκευή καταγραφής και μία κάρτα ήχου), να το αναλύει, να εμφανίζει τα αποτελέσματα και στη συνέχεια να επαναλαμβάνει την ίδια διαδικασία για το επόμενο κομμάτι. Το πλήθος των δειγμάτων που βρίσκονται σε ένα τέτοιο κομμάτι που διαβάζεται και αναλύεται, καλείται “chunk size”, ορολογία την οποία έχει υιοθετήσει η βιβλιοθήκη που χρησιμοποιούμε για τη συγκεκριμένη διαδικασία (PyAudio [4]). Αν τα “κομμάτια” αυτά είναι πολύ μικρά σε μέγεθος, αν δηλαδή έχουμε μικρό “chunk size”, θα είναι αδύνατο να ανιχνεύσουν κάποιες συχνότητες στα επιμέρους αυτά σήματα, ιδιαίτερα για μεγάλα μήκη κύματος (χαμηλές συχνότητες). Από την άλλη, αν το “chunk size” είναι υπερβολικά μεγάλο, θα υπάρχει καθυστέρηση στην εμφάνιση των αποτελεσμάτων από την εφαρμογή αφού θα πρέπει πρώτα να “ηχογραφείται” ένα μεγαλύτερο “κομμάτι” πριν αυτό αναλυθεί. Ύστερα από πειραματισμούς, καταλήγουμε στο ότι, για τις συνθήκες και τα μέσα με τα οποία δοκιμάστηκε η εφαρμογή, μία καλή επιλογή για την συγκεκριμένη παράμετρο ήταν τα 44100 δείγματα (έχοντας ως συχνότητα δειγματοληψίας τα 44100 Hz). Τα ψηφιακά σήματα αποθηκεύονται χρησιμοποιώντας πίνακες της βιβλιοθήκης NumPy [5].

## Ανίχνευση Θεμελιώδους Συχνότητας

Έχοντας το ψηφιακό σήμα, το επόμενο βήμα είναι ανιχνευθεί η θεμελιώδης συχνότητα. Για να γίνει αυτό, πρέπει το σήμα να μετατραπεί από το πεδίο του χρόνου στο πεδίο της συχνότητας. Σε αυτό χρησιμοποιούμε τον Διακριτό Μετασχηματισμό Fourier μέσω της βιβλιοθήκης SciPy [6]. Στη συνέχεια, οργανώνουμε το amplitude της κάθε συχνότητας σε ένα λεξικό της Python, όπου τα κλειδιά είναι οι συχνότητες και οι τιμές οι τιμές του amplitude. Τέλος, μπορούμε να αναζητήσουμε την συχνότητα με το μεγαλύτερο amplitude και να την θεωρήσουμε ως θεμελιώδη. Παρ' όλα αυτά, πολλές φορές, η πιο “δυνατή” συχνότητα, η συχνότητα δηλαδή με το μεγαλύτερο amplitude, δεν είναι η θεμελιώδης συχνότητα, αλλά κάποια αρμονική της.



Ως εκ τούτου, μπορεί η συχνότητα που ανιχνεύεται να είναι διαφορετική από τη νότα που “παίζει” ο χρήστης. Είναι, λοιπόν, απαραίτητο να βρεθεί ένας πιο αξιόπιστος τρόπος ανίχνευσης της θεμελιώδους συχνότητας. Μία προσέγγιση που μπορούμε να ακολουθήσουμε είναι να χρησιμοποιήσουμε το αποκαλούμενο “Harmonic Product Spectrum” ώστε να απομονώσουμε τη θεμελιώδη συχνότητα. Με την προσέγγιση αυτή, παίρνουμε το διακριτό σήμα στο πεδίο της συχνότητας, όπως έχει προκύψει από τον μετασχηματισμό Fourier και το υποδειγματίζουμε ανά 2, ανά 3, ανά 4 και ούτω καθεξής. Αν υποδειγματίσουμε το σήμα ανά 2 (αν δηλαδή λάβουμε υπόψη κάθε 2 bins στον οριζόντιο άξονα), παρατηρούμε ότι η πρώτη αρμονική “μεταφέρεται” στο ίδιο σημείο που είναι η θεμελιώδης συχνότητα στο αρχικό σήμα (αφού άλλωστε η πρώτη αρμονική είναι το διπλάσιο της θεμελιώδους). Αν τώρα πολλαπλασιάσουμε τα δύο αυτά σήματα (το αρχικό και το υποδειγματισμένο), παρατηρούμε ότι θα ενισχυθεί το amplitude στην περιοχή της θεμελιώδους συχνότητας (αφού εκεί υπάρχει περιεχόμενο και στο αρχικό και στο υποδειγματισμένο σήμα). Συνεχίζουμε την ίδια διαδικασία για τα υπόλοιπα υποδειγματισμένα σήματα (ανά 2, ανά 3 κλπ.) και στο τέλος καταλήγουμε με ένα σήμα στο οποίο η θεμελιώδης συχνότητα είναι εμφανώς ενισχυμένη σε σχέση με τις άλλες.



Σχήμα 3 : Harmonic Product Spectrum [7]

Συνήθως, αρκούν λίγες επαναλήψεις του αλγορίθμου ώστε να επιτευχθεί το επιθυμητό αποτέλεσμα, να ενισχυθεί δηλαδή η περιοχή της θεμελιώδους συχνότητας.

## Ανίχνευση Πλησιέστερης Νότας

Ένα μουσικό κουρδιστήρι εμφανίζει στον χρήστη πόσο “μακριά” βρίσκεται από τη νότα στόχο. Επομένως, αρκεί να γνωρίζουμε τη συχνότητα που προέρχεται από την είσοδο (συχνότητα εισόδου) και τη συχνότητα της νότας που βρίσκεται πλησιέστερα στην συχνότητα εισόδου (τη νότα στόχο δηλαδή). Παραδείγματος χάρη, αν η συχνότητα εισόδου είναι η 439 Hz, είναι πιθανό ο χρήστης να θέλει να “φτάσει” τη νότα A4 που έχει συχνότητα 440 Hz. Μία προσέγγιση που μπορούμε να χρησιμοποιήσουμε για να βρούμε την πλησιέστερη νότα, είναι να φτιάξουμε για κάθε νότα μία hard-coded λίστα από τις συχνότητες που αντιστοιχούν στη νότα αυτή. Στη συνέχεια μπορούμε να διατρέξουμε τις λίστες αυτές και να βρούμε την πλησιέστερη συχνότητα. Βέβαια, σε περίπτωση που ο χρήστης θέλει ένα διαφορετικό πρότυπο κουρδίσματος (Tuning Standard) ο πίνακας αυτός παύει να ισχύει. Παραδείγματος χάρη, αν επιλεγεί το πρότυπο A4=432 Hz, η νότα A4 παύει να βρίσκεται στα 440 Hz, και το ίδιο συμβαίνει και για όλες τις υπόλοιπες νότες. Αν και θα μπορούσαμε να μεταβάλλουμε αναλόγως τον πίνακα σε μια τέτοια περίπτωση, η συγκεκριμένη λύση δεν είναι η πλέον

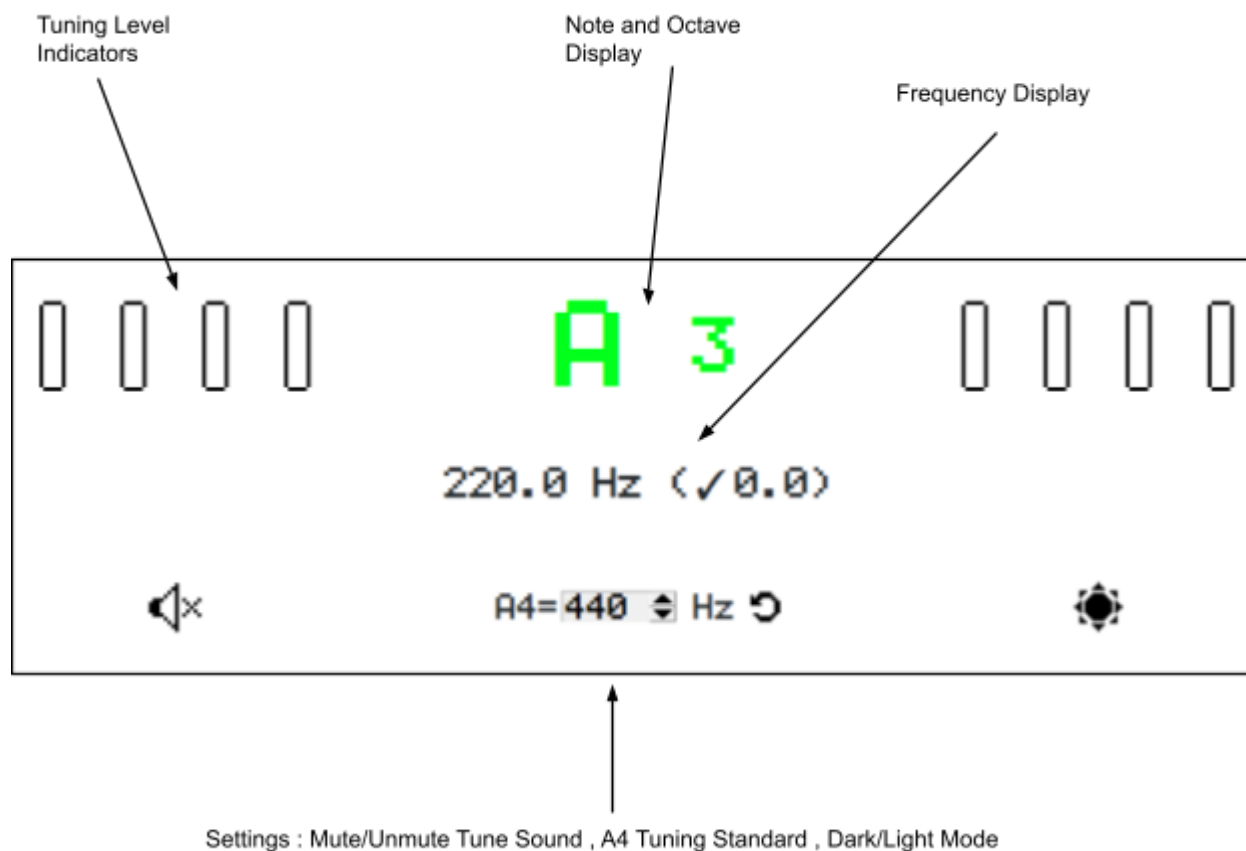
αποδοτική. Η δεύτερη, και περισσότερο κομψή, προσέγγιση είναι να βρίσκουμε την συχνότητα της κοντινότερης νότας χρησιμοποιώντας μία μαθηματική φόρμουλα από την οποία προκύπτουν οι συχνότητες των νοτών συναρτήσει του πρότυπου κουρδίσματος.

$$f_n = f_0 * a^n$$

Όπου  $f_0$  το πρότυπο κουρδίσματος, δηλαδή η συχνότητα της νότας A4 (πχ  $f_0 = 440$ ),  $a = 2^{(1/12)}$  και  $n$  ο δείκτης (index) της νότας (ξεκινώντας από  $n=0$  για τη νότα A4). Έτσι, μπορούμε να υπολογίζουμε το  $f_n$  για διαφορετικά  $n$  (σε ένα λογικό εύρος από -50 έως + 50, καλύπτοντας περίπου 100 νότες, δηλαδή περίπου 8 οκτάβες) και να βρίσκουμε το  $n$  για το οποίο προκύπτει  $f_n$  πλησιέστερο στη συχνότητα εισόδου [8].



## Γραφική Διεπαφή Χρήστη (GUI)



Στόχος ήταν να δημιουργηθεί ένα απλό στην χρήση και διαισθητικό UI. Ο χρήστης απλά ανοίγει την εφαρμογή, κάνει γρήγορη παραμετροποίηση στις ρυθμίσεις, κουρδίζει το όργανο του και κλείνει την εφαρμογή .

Η ένδειξη της νότας/οκτάβας και της συχνότητας έχουν τοποθετηθεί στο κέντρο της οθόνης και έχουν μεγάλο μέγεθος ώστε να είναι ευανάγνωστα .

Στα πλαϊνά της νότας υπάρχουν ενδείξεις που δείχνουν την κατεύθυνση κουρδίσματος (πάνω η κάτω) και το επίπεδο του κουρδίσματος με την απόσταση από την νότα και αντίστοιχο χρωματικό κωδικό.

Οι ρυθμίσεις της εφαρμογής είναι απλές και βρίσκονται στο ίδιο παράθυρο ώστε να μην χρειάζεται ο χρήστης να ανοίγει νέο παράθυρο/σελίδα πράγμα που τον καθυστερεί.

Οι ρυθμίσεις περιλαμβάνουν:

- Σίγαση/ενεργοποίηση ήχου-feedback σωστού κουρδίσματος.
- Αλλαγή του κουρδίσματος της νότας ΛΑ/Α4 στην επιθυμητή συχνότητα και reset του κουρδίσματος στην default τιμή 440 , για ευκολία χρήσης.
- Αλλαγή χρωματικής παλέτας (dark mode/light mode).

Όταν ο χρήστης κουρδίζει σωστά μια νότα δίνεται ακουστικό feedback με ένα σύντομο ήχο και οπτικό feedback με το πρασίνισμα της ένδειξης νότας .

## Ανάπτυξη GUI

Η εφαρμογή έχει αναπτυχθεί με την χρήση της βιβλιοθήκης tkinter [9]. Ο τρόπος που λειτουργεί είναι παρόμοιος με αντίστοιχες βιβλιοθήκες ανάπτυξης εφαρμογών desktop: δημιουργείται ένα πλέγμα (στην συγκεκριμένη εφαρμογή ένα πλέγμα με 3 σειρές και 9 στήλες) και πάνω στο πλέγμα τοποθετούνται διάφορα widgets. Ένα widget είναι ένα αντικείμενο από το οποίο μπορούμε είτε να εξάγουμε πληροφορίες για να τις χρησιμοποιήσουμε στην εκτέλεση του κώδικα είτε να εισάγουμε πληροφορίες για να δείξουμε στον χρήστη τα αποτελέσματα του κώδικα αυτού.

Για παράδειγμα στο παράθυρο η ένδειξη νότας είναι ένα label το οποίο κάνουμε update σε κάθε loop με την νότα που παίζει, το κουμπί είναι ένα button που όταν πατιέται καλείται αυτόματα μια συνάρτηση που κάνει mute τον ήχο στο παράθυρο και η ρύθμιση συχνότητας της ΛΑ είναι ένα spinbox από το οποίο παίρνουμε την τιμή σε κάθε loop ώστε να την χρησιμοποιήσουμε ως παράμετρο  $f_0$  στον τύπο για την εύρεση της πλησιέστερης νότας σε μια συχνότητα.

## Οδηγίες Εγκατάστασης

Σημείωση : Η εφαρμογή έχει αναπτυχθεί και δοκιμαστεί μόνο σε λειτουργικό Microsoft Windows.

### Εγκατάσταση

Download source code or clone from repo with :

```
git clone https://github.com/VangelisDimi/FineTuned
```

Go to source code directory and install dependencies:

```
pip install -r requirements.txt
```

Execute the program:

```
python3 main.py
```

### Για χρήστες Windows

Η βιβλιοθήκη PyAudio που απαιτείται για να τρέξει ο κώδικας έχει κάποια προβλήματα κατά την εγκατάσταση στα Windows. Για να την εγκαταστήσετε μπορείτε να ακολουθήσετε τα παρακάτω βήματα.

In your terminal:

```
pip install pipwin
```

```
pipwin install pyaudio
```

Alternatively:

Visit [https://www.lfd.uci.edu/~gohlke/pythonlibs/#\\_pyaudio](https://www.lfd.uci.edu/~gohlke/pythonlibs/#_pyaudio)

Download PyAudio-....whl (depending on your system)

```
pip install PyAudio-....whl
```

## Δοκιμή/Επίδειξη Εφαρμογής

Η εφαρμογή έχει δοκιμαστεί από εμάς τους ίδιους καθώς και από ένα τρίτο πρόσωπο το οποίο δεν έχει οικειότητα με αυτήν. Είναι εύκολη στην κατανόηση της λειτουργίας της και στο πως παραμετροποιείται και είναι πλήρως λειτουργική ως κουρδιστήρι.

Στο [βίντεο επίδειξης](#) φαίνεται η χρήση της εφαρμογής για το κούρδισμα κιθάρας.

## Μελλοντικές Προσθήκες/Βελτιώσεις

Η εφαρμογή βρίσκεται σε μια λειτουργική κατάσταση . Υπάρχουν όμως αρκετές βελτιώσεις και προσθήκες που μπορούν να γίνουν στο μέλλον ώστε να γίνει πιο ολοκληρωμένη.

### Audio

#### **Βελτίωση ακρίβειας και αξιοπιστίας ανίχνευσης συχνοτήτων**

Μια από τις βελτιώσεις που μπορούν να γίνουν είναι η διόρθωση ενός λάθους στην ανίχνευση συχνότητας, σπάνια η νότα που ανιχνεύεται έχει απόκλιση μιας οκτάβας από την νότα που παίζει ο χρήστης.

#### **Εφαρμογή ζωνοπερατού φίλτρου στο σήμα για μείωση εξωτερικών θορύβων**

Μπορεί να χρησιμοποιηθεί ζωνοπερατό φίλτρο το οποίο θα αφαιρεί τις πολύ χαμηλές και πολύ υψηλές νότες από το σήμα. Αποτέλεσμα της μεθόδου αυτής θα είναι να αφαιρεθούν ήχοι από το σήμα οι οποίοι είναι σε πολύ χαμηλή ή υψηλή συχνότητα για να έχουν προέλθει από το μουσικό όργανο , οπότε είναι εξωτερικοί θόρυβοι.

#### **Μείωση latency**

Optimization για την μείωση του χρόνου που χρειάζεται για την επεξεργασία και ανάλυση του σήματος για να βρεθεί η fundamental συχνότητα. Με την μείωση του χρόνου αυτού η εφαρμογή έχει μικρότερο διάστημα από το παίξιμο της νότας μέχρι την ανίχνευση της και έτσι η εμπειρία χρήσης γίνεται πιο ομαλή.

#### **Δοκιμή και παραμετροποίηση με μεγάλο εύρος μουσικών οργάνων**

### GUI

#### **Ρυθμίσεις μικροφώνου (επιλογή συσκευής μικροφώνου/παραμετροποίηση έντασης μικροφώνου)**

Ρυθμίσεις που επιτρέπουν στον χρήστη να παραμετροποιήσει το μικρόφωνο του για καλύτερα αποτελέσματα ανίχνευσης. Για την ρύθμιση της έντασης θα υπάρχει και αντίστοιχη ένδειξη που θα δείχνει το επίπεδο έντασης του μικροφώνου και αν είναι πολύ χαμηλό ή πολύ δυνατό.

#### **Χρήση βιβλιοθήκης GUI πιο κατάλληλης για δυναμικές εφαρμογές**

Η βιβλιοθήκη tkinter είναι προορισμένη κυρίως για χρήση σε πιο στατικές εφαρμογές. Για δυναμικές εφαρμογές σαν και αυτήν υπάρχουν αρκετοί περιορισμοί. Μια καλή εναλλακτική θα ήταν η βιβλιοθήκη PyGame η οποία υποστηρίζει και τα widgets του tkinter αλλά προσφέρωντας περισσότερη ελευθερία.

#### **Υποστήριξη για screen readers**

Τα αντικείμενα στο παράθυρο μπορούν να γίνουν συμβατά με screen readers ώστε να υπάρχει προσβασιμότητα για τυφλά άτομα ή άτομα με περιορισμένη όραση.

Ο χρήστης θα μπορεί να περιηγηθεί και να αλλάξει τις ρυθμίσεις και μετά να περιηγηθεί στην ένδειξη νότας και συχνότητας ώστε να λαμβάνει ηχητικές πληροφορίες σχετικά με το κούρδισμα.

Η υλοποίηση της λειτουργίας είναι σχετικά απλή ,αυτήν την στιγμή δεν είναι δυνατή λόγω ελλείψεων στην βιβλιοθήκη που χρησιμοποιείται για το GUI.

#### **Υλοποίηση βελώνας κουρδίσματος με ομαλή κίνηση**

Στην τωρίνη έκδοση το επίπεδο και η κατεύθυνση κουρδίσματος επικοινωνούνται στο χρήστη με την χρήση φωτεινών ενδείξεων. Η υλοποίηση αυτή είναι απλή αλλά υστερεί καθώς οι ενδείξεις προσφέρουν περιορισμένη κίνηση που είναι μη ομαλή . Μπορεί να βελτιωθεί με την υλοποίηση μιας “βελώνας” η οποία κινείται συνεχώς ανάλογα το επίπεδο και την κατεύθυνση κουρδίσματος και θα έχει αντίστοιχη χρωματική κωδικοποίηση. Ο χρήστης θα βλέπει άμεσα αλλαγή ανάλογα με το κούρδισμα πράγμα που θα βελτιώσει την ομαλότητα της εμπειρίας χρήσης.

#### **Ανάπτυξη και testing για τα υπόλοιπα λειτουργικά συστήματα (MacOs και Linux)**

Μέχρι στιγμής έχει γίνει ανάπτυξη και testing μόνο για Windows.

#### **Build εκτελέσιμης έκδοσης**

Σκοπός της εφαρμογής είναι να είναι η απλότητα και ταχύτητα χρήσης. Με την δημιουργία εκτελέσιμης έκδοσης θα είναι πολύ εύκολο στον χρήστη το κατέβασμα και η εκτέλεση της εφαρμογής.

## Επίλογος

Στη συγκεκριμένη εργασία εξετάσαμε την ανάπτυξη μιας desktop εφαρμογής που μπορεί να χρησιμοποιηθεί ως μουσικό κουρδιστήρι. Η κεντρική ιδέα πίσω από τη λειτουργία του προγράμματος ήταν η εξαγωγή της θεμελιώδους συχνότητας από τα σήματα εισόδου, διαδικασία για την οποία χρησιμοποιήθηκε ο διακριτός μετασχηματισμός Fourier και στη συνέχεια η τεχνική Harmonic Product Spectrum. Στη συνέχεια, παρουσιάσαμε ένα εύκολο στη χρήση και στην κατανόηση γραφικό περιβάλλον (GUI) που μπορεί να χρησιμοποιηθεί ως διεπαφή στις διάφορες λειτουργίες του προγράμματος. Τέλος, κάναμε αναφορά σε πιθανές βελτιώσεις και προσθήκες που μπορούν να ενταχθούν στην εφαρμογή προκειμένου να κάνουν την λειτουργία της πιο αξιόπιστη, ομαλή και προσβάσιμη.

## Αναφορές

1. “Musical Tuning”, Wikipedia, [https://en.wikipedia.org/wiki/Musical\\_tuning](https://en.wikipedia.org/wiki/Musical_tuning)
2. “Electronic Tuner”, Wikipedia, [https://en.wikipedia.org/wiki/Electronic\\_tuner](https://en.wikipedia.org/wiki/Electronic_tuner)
3. “Fundamental Frequency”, Wikipedia, [https://en.wikipedia.org/wiki/Fundamental\\_frequency](https://en.wikipedia.org/wiki/Fundamental_frequency)
4. PyAudio Documentation, <http://people.csail.mit.edu/hubert/pyaudio/docs/>
5. NumPy Documentation, <https://numpy.org/>
6. Scipy Documentation, <https://docs.scipy.org/doc/scipy/index.html>
7. Mary Lourde R., Anjali Kuppayil Saji (2009). “A Digital Guitar Tuner”,  
[https://www.researchgate.net/publication/45888076\\_A\\_Digital\\_Guitar\\_Tuner](https://www.researchgate.net/publication/45888076_A_Digital_Guitar_Tuner)
8. “Physics of Music-Notes”, <https://pages.mtu.edu/~suits/NoteFreqCalcs.html>
9. tkinter Documentation , <https://docs.python.org/3/library/tk.html>