

Glazba i matematika

Habdija, Robert

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:211:005540>

Rights / Prava: [Attribution-ShareAlike 3.0 Unported](#)/[Imenovanje-Dijeli pod istim uvjetima 3.0](#)

Download date / Datum preuzimanja: **2024-07-09**



Repository / Repozitorij:

[Faculty of Organization and Informatics - Digital Repository](#)



**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN**

Robert Habdija

GLAZBA I MATEMATIKA

ZAVRŠNI RAD

Varaždin, 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ź D I N

Robert Habdija

Matični broj: 0016142191

Studij: Informacijski sustavi

GLAZBA I MATEMATIKA

ZAVRŠNI RAD

Mentor :

mag. educ. math et phys. Damjan

Klemenčić

Varaždin, rujan 2023.

Robert Habdija

Izjava o izvornosti

Izjavljujem da je moj ZAVRŠNI RAD izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

U ovom će se radu prikazati povezanost glazbe i matematike koristeći matematičke izraze i izrađeno programsko rješenje. U početnom će dijelu biti prikazan povijesni pregled povezivanja glazbe i matematike te će matematički biti definirani zvuk i ton. Programsko će rješenje primjenjivati matematičke temelje glazbe na dva načina: za reprodukciju tona te za identificiranje temeljne frekvencije dobivenog zvuka. Identificiranje frekvencije se temelji na Fast Fourier Transform algoritmu koji će također biti objašnjen unutar rada. Korišteni algoritmi i programski kod će biti objašnjeni detaljno i na temelju primjera. Programsko je rješenje izrađeno u programskom jeziku C# koristeći .NET Framework verziju 4.7.2 te Windows Forms sučelje.

Ključne riječi: glazba; matematika; zvuk; ton; frekvencija; Fourier; transformacije; algoritam;

Sadržaj

1. Uvod	1
2. Metode i tehnike rada	2
3. Povijesni pregled	3
3.1. Žica i Pitagora	3
3.1.1. Sveti tetraktis	3
3.1.2. Monokord i omjeri	3
3.1.2.1. Fizikalne značajke monokorda	4
3.1.3. Univerzalna proporcija	4
3.1.4. Utemeljenje kvintnog kruga	6
3.1.5. Glazba kugli	7
3.2. Postepena standardizacija	7
3.2.1. Francuski štim	7
3.2.2. Stuttgartski štim i uvođenje standarda	8
4. Matematički opis tona	10
4.1. Stojni val i usporedba Pitagorejskog štimanja sa štimanjem jednakog tempera- menta	10
4.1.1. Pola žice - interval oktave	12
4.1.2. Trećina žice - interval kvinte	13
4.1.3. Četvrtina žice - interval kvarte	14
4.2. Fourierove transformacije	15
4.2.1. Povijest i vrste transformacija	15
4.2.2. <i>Fast Fourier Transform</i> algoritam	16
5. Programska dokumentacija	18
5.1. Korisničko sučelje	18
5.1.1. Reprodukcija tonova i intervala	18
5.1.2. Odjeljak za štimer	19
5.2. Programski kod	20
5.2.1. Klasa <i>Tonovi.cs</i>	20
5.2.1.1. Strukture i atributi	20
5.2.1.2. Funkcije	21
5.2.2. Forma <i>Form1.cs</i>	21
5.2.2.1. Strukture i atributi	21
5.2.2.2. Funkcije	22

6. Zaključak	25
Popis literature	26
Popis slika	27
Popis tablica	28

1. Uvod

Ovaj će rad prikazati povezanost glazbe i matematike. Kroz rad će biti objašnjene veze između glazbe, zvuka, tona i frekvencije na temelju nekoliko primjera te povijesnog pregleda ovih termina. Ova tema ima meni posebno značenje. U svojih deset godina glazbene edukacije, nikad nisam očekivao spojiti ove dvije sfere. Usprkos tome, kroz izradu ovoga rada sam dobio nova razumijevanja koja su me nadasve zadivila. Sve sam svoje glazbeno znanje primijenio u izradi ovog završnog rada i programskog rješenja. Ovaj je rad dodao novi sloj ljepote na moje umjetničko poimanje glazbe te me na zanimljiv način podsjetio zašto volim glazbu.

2. Metode i tehnike rada

Pri razradi teme je korišteno nekoliko izvora s interneta te knjige "Matematika i muzika" autora Zvonimira Šikića i Zorana Šćekića te "Znanstvenikov i inženjerov vodič za digitalno procesiranje signala (eng. *The Scientist and Engineer's Guide to Digital Signal Processing*) autora Stevena W. Smitha. Programsko je rješenje izrađeno u programskom jeziku C# koristeći .NET Framework verziju 4.7.2 te Windows Forms sučelje. Za izračunavanje temeljne frekvencije koristiti će se FFT (eng. *Fast Fourier Transform*) algoritam te će isti biti detaljno objašnjen u dokumentaciji programskog rješenja.

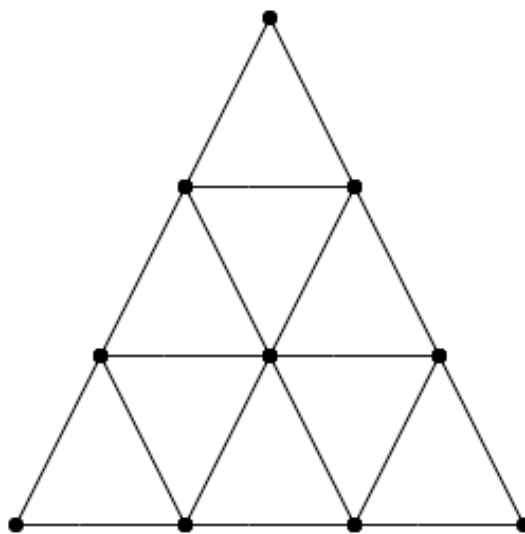
3. Povijesni pregled

U ovom će dijelu rada biti opisana istraživanja povezanosti glazbe i matematike kroz povijest, počevši od Pitagore i njegovih sljedbenika, pa sve do modernog doba s uvođenjem frekvencije 440 Hz kao općeprihvaćenog ISO (eng. *International Organization for Standardization*) standarda.

3.1. Žica i Pitagora

3.1.1. Sveti tetraktis

Prvi ljudi koji su razmišljali o povezanostima glazbe i matematike su bili antički filozofi. Najvažniji među njima su bili Pitagora i njegovi sljedbenici te Platon. Pitagora nije ostavio nikakve rasprave na ovu temu, već su njegovi sljedbenici ostavili zapise njegovih glazbenih djelovanja. Jedna od glavnih forma koju je Pitagora iskoristio za iskazivanje matematičke povezanosti cijelog svemira je bio takozvani tetraktis, odnosno, po njemu, grupa od četiri broja preko koje se sve skladnosti svemira mogu prikazati i povezati. Ta je grupa brojeva 1, 2, 3, 4.

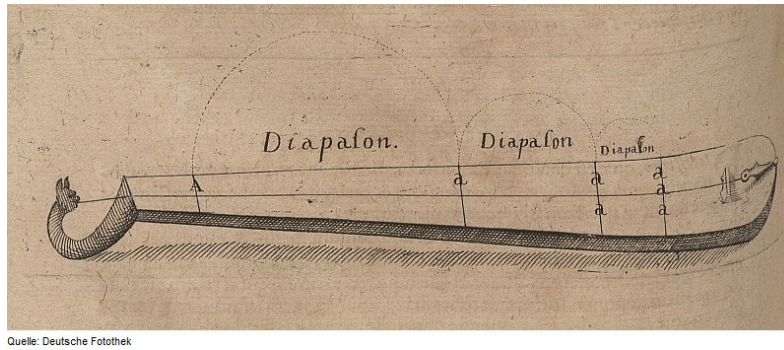


Slika 1: Izgled tetraktisa (autorski rad)

Pitagora nalaže da je tetraktis simbol aritmetičkih, geometrijskih i glazbenih omjera koji čine temelje svemira, te su njegovi sljedbenici smatrali da je tetraktis ilustrirani prikaz strukture cijelog svemira te svega što u njemu postoji. Kroz svoje nauke, počeli su shvaćati tetraktis kao svetinju.

3.1.2. Monokord i omjeri

Pitagorini sljedbenici su koristili vrlo jednostavan instrument kako bi potvrdili navedene omjere i uspostavili temelje glazbene teorije. Taj se instrument zove monokord, te se sastoji od daske, žice i rezonantne kutije.



Quelle: Deutsche Fotothek

Slika 2: Primjer monokorda (Izvor: [1])

Koristeći ovaj instrument, Pitagorini su sljedbenici otkrili da su glazbeni intervali obrnuto proporcionalni duljini žice. Tako su ustanovili da se točno na polovici žice nalazi interval oktave ($\frac{1}{2}$), na trećini žice interval kvinte ($\frac{2}{3}$) te na četvrtini žice interval kvarte ($\frac{3}{4}$). Kroz ovo su saznanje utvrdili vjerodostojnost koncepta tetraktisa kao temelja glazbe.

3.1.2.1. Fizikalne značajke monokorda

Kao što je već navedeno, monokord se sastoji od daske, žice i rezonantne kutije. Rezonantna kutija određuje značajku tona koja neće biti dotaknuta u ovom radu, a to je boja tona. Fokus će biti na žici i daski, pošto one izravno utječu na frekvenciju dobivenog tona.

Za potrebe objašnjenja načina rada monokorda koristiti će se pojmovi: masa po duljini žice $\mu = \frac{\text{masa}}{\text{duljina}} [\frac{kg}{m}]$, sila napetosti žice $F_T = konst.$ te brzina širenja vala na napetoj žici $|v| = \sqrt{\frac{F_T}{\mu}}$. Sila napetosti žice varira od instrumenta do instrumenta zbog različitih veličina instrumenata te različitih debljina žica koje se na njima koriste.

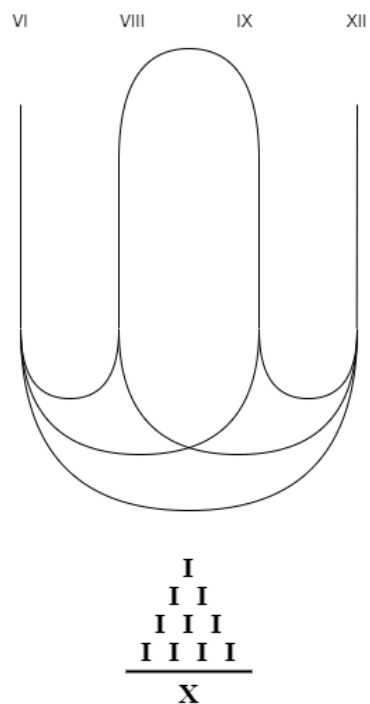
Na zatrzanjoj praznoj žici, valna duljina (λ) je definirana duljinom žice (l). U potpoglavlju 4.1. će biti pobliže objašnjen odnos valne duljine i duljine žice na temelju primjera. Mijenjanjem napetosti žice (F_T) mijenja se brzina zvuka (v). Uzmemo li za primjer monokord koji na odsviranoj praznoj žici producira ton A frekvencije 110 Hz te da je duljina žice koja titra 650 mm, odnosno 0.65 m, dobivamo brzinu vala (v):

$$f = \frac{v}{\lambda} \rightarrow v = f \cdot \lambda = 2 \cdot f \cdot l = 2 \cdot 110Hz \cdot 0.65m = 143 \frac{m}{s}$$

Ukoliko želimo da osnovni ton na praznoj žici bude 220 Hz, odnosno za interval oktave viši, potrebno je napeti žicu 4 puta jače ili 4 puta smanjiti masu žice po jedinici duljine.

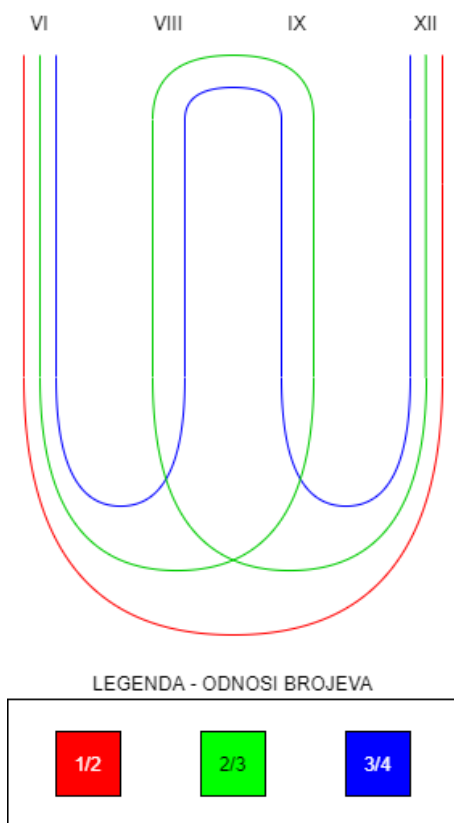
3.1.3. Univerzalna proporcija

Jedna od inačica svetog tetraktisa vidljiva je na Rafaelovoj slici *Atenska škola*[2]. Jedan od učenika koji sjedi kraj Pitagore drži tabelu koja na sebi sadrži rimske brojeve 6, 8, 9 i 12. Ti su brojevi povezani s nekoliko lukova koji čine različite omjere, te je ispod slike nacrtan sveti tetraktis.



Slika 3: Izgled tabele s Rafaelove slike *Atenska škola* (autorski rad)

Niz brojeva 6-8-9-12 sadrži sličnosti sa tetraktisom u iscrtanim omjerima: brojevi 6 i 12 čine omjer $\frac{1}{2}$, 6 i 9 te 8 i 12 čine omjer $\frac{2}{3}$, 6 i 8 te 9 i 12 čine omjer $\frac{3}{4}$.



Slika 4: Vizualno objašnjenje tabele (autorski rad)

3.1.4. Utemeljenje kvintnog kruga

U Pitagorejskom shvaćanju glazbe, tonski se sustav temeljio na nizu intervala čistih kvinta, u kojem je ton C već tada bio svojevrsna sredina niza. Uzlazno taj niz izgleda:

C - G - D - A - E - B - F#,

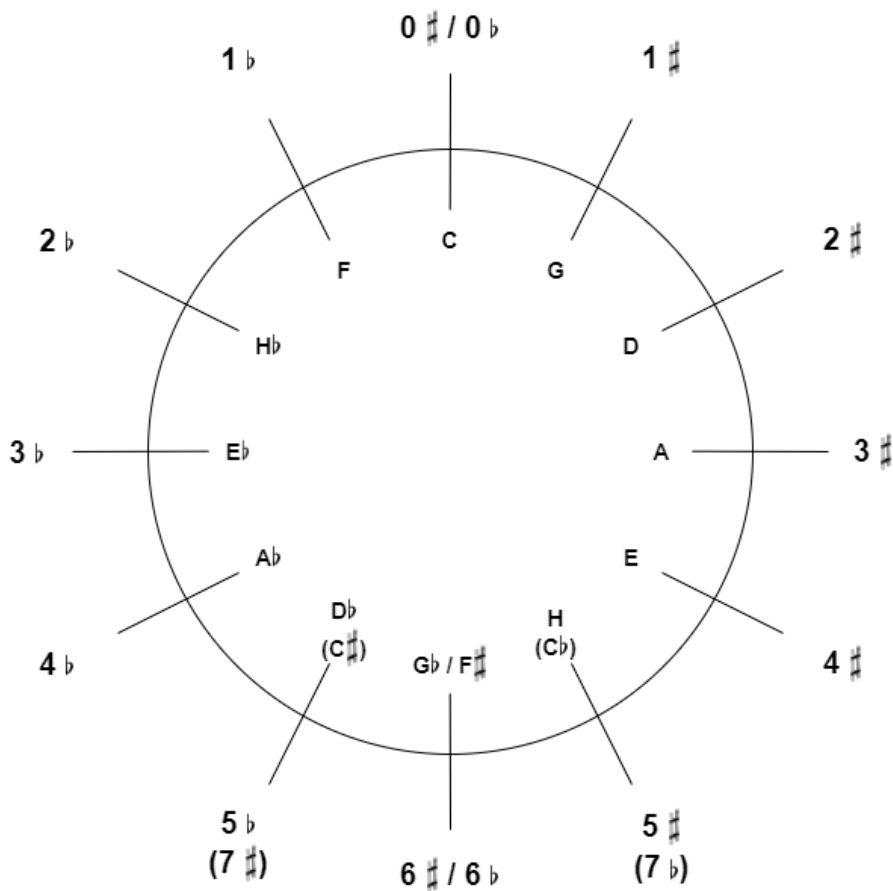
dok silazno izgleda:

C - F - Bb - Eb - Ab - Db - Gb.

Cjelokupni niz se i danas primjenjuje u glazbi kao slijed durskih tonaliteta sa povisilicama odnosno snizilicama, gdje je C-dur tonalitet bez ikakvih povisilica i snizilica:

Gb - Db - Ab - Eb - Bb - F - C - G - D - A - E - B - F#.

U modernoj se glazbenoj teoriji ovakav niz tonova, odnosno tonaliteta, naziva kvintni krug.



Slika 5: Ilustracija kvintnog kruga (autorski rad)

3.1.5. Glazba kugli

Jedno od Pitagorinih učenja je bilo razlikovanje tri tipova glazbe: *musica instrumentalis*, odnosno glazba koju proizvode glazbeni instrumenti; *musica humana*, odnosno nečujna glazba svakog pojedinca koja je skladna ili neskladna ovisno o odnosu duha i tijela; te *musica mundana*, odnosno glazba nebeskih tijela. *Musica mundana* navodi kako svako nebesko tijelo, uključujući Mjesec, Sunce i ostale planete Sunčeva sustava, svojim pokretima proizvodi šum. Smatralo se da svako nebesko tijelo ima jedinstveni šum te, premda su navedeni šumovi nečujni ljudskom uhu, navodi da njihov sklad utječe na kvalitetu života na zemlji. *Musica mundana* je imala utjecaj na nekoliko velikana filozofije i matematike, od kojih je među važnijima Johannes Kepler. Pitagorina su razmišljanja imala značajan utjecaj na Keplera, te o istima diskutira u svom djelu *Harmonices Mundi*. Koristeći svoje znanje o gibanjima planeta, otkrio je da su odnosi između orbitalnih vremena pojedinih planeta približno jednaki odnosima između intervala poput oktave i kvarte. S tim je saznanjima imao velik doprinos razmišljanju da koncept "glazbe sfera" može biti utemeljen u prirodi, ali, usprkos tome, nije vjerovao da je "glazba sfera" stvarna fizička pojava, već je smatrao da je to samo simboličan način za interpretirati harmoniju svemira.

3.2. Postepena standardizacija

Do 19. stoljeća nije bilo koordiniranih pothvata za standardizaciju ijedne frekvencije kao glavne frekvencije za štimanje. U to su se doba po Europi mogle naći mnoge različite frekvencije, ponekad do te razine da su se u istom gradu koristile dvije do tri različite frekvencije za štimanje. Prvi je veći korak prema standardizaciji bio izum zvučne vilice 1711. godine. Premda je zvučna vilica omogućila lakše uštimanje, kroz godine su se pojavljivale brojne varijacije: jedna vilica iz 1740. godine povezana s Händelom je podešena na 422,5 Hz, druga iz 1780. godine je podešena na 409 Hz, dok je vilica koja je pripadala Ludwigu van Beethovenu oko 1800. godine postavljena na 455,4 Hz. Krajem 18. stoljeća generalno je raspon frekvencija za štimanje bio između 400 Hz i 450 Hz, ali je bilo odstupanja, što je vidljivo u slučaju Beethoveneve zvučne vilice.

3.2.1. Francuski štim

Prvi uspješni pokušaj formalne standardizacije frekvencija za štimanje se pojavio u Francuskoj 1859. godine, kada je tadašnja vlada uvela zakon koji je postavio frekvenciju tona A na vrijednost od 435 Hz. Ubrzo je ta frekvencija postala popularna i u okolnim zemljama te je poprimila kolokvijalni naziv "francuski štim". Ovaj je štim čak ušao i u jedan od najvažnijih dokumenata 20. stoljeća: Versajski ugovor, mirovni sporazum kojim je okončan Prvi svjetski rat. U Versajskom se ugovoru nalazi nekoliko odredbi i međunarodnih sporazuma koje je Njemačka morala prihvatiti pri predaji, među kojima je sporazum "glede utvrđivanja koncertnog štima" (eng. *Convention (...) regarding the establishment of a concert pitch..*

- (12) Conventions of September 23, 1910, respecting the unification of certain regulations regarding collisions²⁶ and salvage at sea.²⁷
- (13) Convention of December 21, 1904, regarding the exemption of hospital ships from dues and charges in ports.²⁸
- (14) Convention of February 4, 1898, regarding the tonnage measurement of vessels for inland navigation.
- (15) Convention of September 26, 1906, for the suppression of night-work for women.
- (16) Convention of September 26, 1906, for the suppression of the use of white phosphorus in the manufacture of matches.
- (17) Conventions of May 18, 1904,²⁹ and May 4, 1910, regarding the suppression of the White Slave Traffic.
- (18) Convention of May 4, 1910, regarding the suppression of obscene publications.³⁰
- (19) Sanitary Conventions of January 30, 1892, April 15, 1893, April 3, 1894, March 19, 1897, and December 3, 1903.³¹
- (20) Convention of May 20, 1875, regarding the unification and improvement of the metric system.³²
- (21) Convention of November 29, 1906, regarding the unification of pharmacopoeial formulae for potent drugs.³³
- (22) Convention of November 16 and 19, 1885, regarding the establishment of a concert pitch.
- (23) Convention of June 7, 1905, regarding the creation of an International Agricultural Institute at Rome.³⁴
- (24) Conventions of November 3, 1881, and April 15, 1889, regarding precautionary measures against phylloxera.
- (25) Convention of March 19, 1902, regarding the protection of birds useful to agriculture.
- (26) Convention of June 12, 1902, as to the protection of minors.

Slika 6: Dio Versajskog ugovora (Izvor: *Versajski ugovor*)

3.2.2. Stuttgartski štim i uvođenje standarda

Nekoliko godina prije uspostave francuskog štima, njemački proizvođač svile i samouki glazbenik Johann Heinrich Scheibler stvorio je uređaj koji je nazvao "tonometar". Navedeni je uređaj napravio od 56 zvučnih vilica i jednog metronoma. Na temelju njegova rada i studija, Društvo njemačkih prirodoslovaca i liječnika (njem. *Gesellschaft Deutscher Naturforscher und Ärzte*) preporučilo je frekvenciju od 440 Hz kao standardnu frekvenciju za štimanje na njihovoj konferenciji u Stuttgartu 1834. godine. Frekvencija 440 Hz je i danas poznata pod nazivom Stuttgartski štim, te je više od jednog stoljeća kasnije, 1939. godine, upravo ta frekvencija dobila međunarodnu preporuku te naziv "koncertni štim". 1955. godine ova frekvencija dobiva status tehničkog standarda od strane Međunarodne organizacije za standardizaciju (ISO, eng. *International Organization for Standardization*) te ju ova institucija 1975. godine ponovno potvrđuje kao standard ISO 16.

ISO 16:1975

Acoustics — Standard tuning frequency (Standard musical pitch)

This standard was last reviewed and confirmed in 2022.
Therefore this version remains current.

Abstract

 **Preview**

Specifies the frequency for the note A in the treble stave and shall be 440 Hz.
Tuning and retuning shall be effected by instruments producing it within an
accuracy of 0,5 Hz.

General information

Status : Published

Publication date : 1975-01

Edition : 1

Number of pages : 1

Technical Committee : [ISO/TC 43](#) Acoustics

ICS : [17.140.01](#) Acoustic measurements and noise abatement in general

Slika 7: Opis standarda ISO 16 na službenim stranicama ISO-a. (Izvor: [4])

4. Matematički opis tona

U ovom će dijelu rada biti prikazane fizikalne značajke tona kao stojnog vala, matematički postupak kojim se dolazi do frekvencija određenih tonova na primjeru stojnog vala te princip funkcioniranja Fourierovih transformacija.

4.1. Stojni val i usporedba Pitagorejskog štimanja sa štimanjem jednakog temperameta

Stojni je val vrsta vala kod kojega određene točke stalno miruju. Najčešće su to dvije točke, npr. na žičanim instrumentima poput gitare. Udarcom o žicu gitare nastaju valovi koji se kreću u međusobno suprotnim smjerovima. Ti valovi interferiraju te stvaraju područja minimalnog i maksimalnog gibanja, koji se nazivaju čvorovi odnosno trbusi stojnog vala.

Štimanje se može napraviti na nekoliko načina, od kojih su dva najpoznatija Pitagorejsko štimanje i tzv. štimanje jednakog temperameta (eng. *equal temperament tuning*). Pitagorejsko štimanje uzima u obzir prije navedene Pitagorejske omjere te se temelji na čistim konsonancama, odnosno intervalima kvarte, kvinte i oktave. S druge strane, štimanje jednakim temperamentom se temelji na baznoj frekvenciji i množenjem iste s brojem $2^{\frac{n}{12}}$, odnosno $\sqrt[12]{2^n}$.

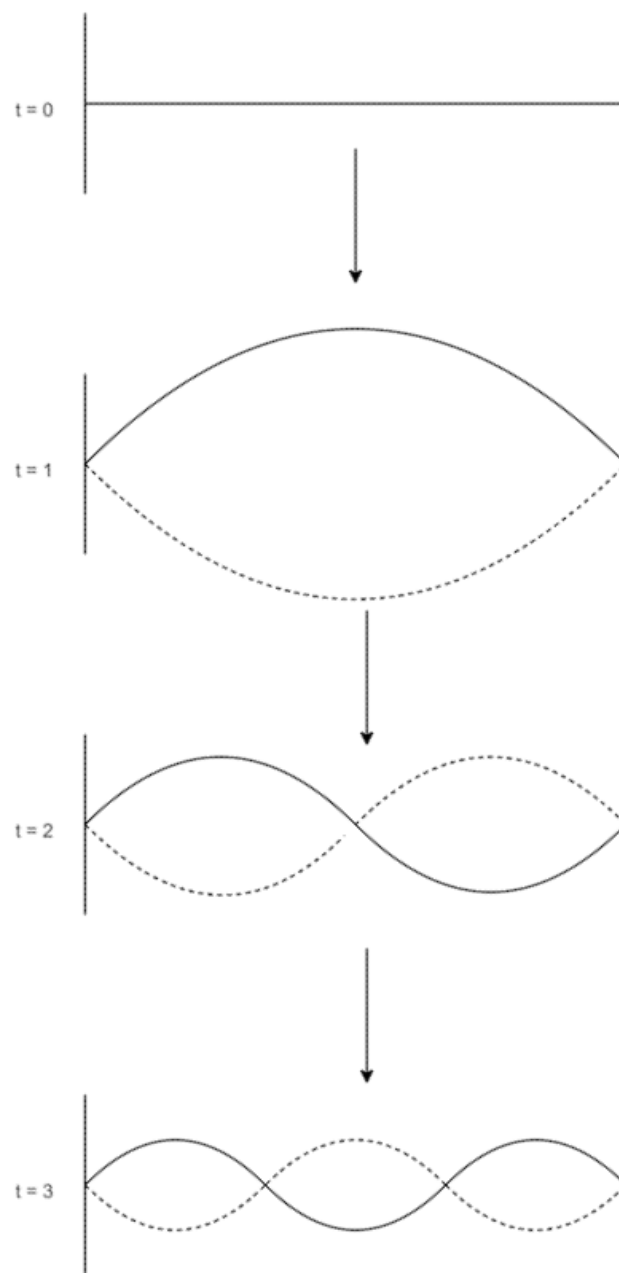
Vrijednosti frekvencija tonova kromatske ljestvice vidljive su u slijedećoj tablici:

Tablica 1: Prikaz vrijednosti frekvencije, korijena i intervala koji ton čini s baznim tonom A

Ton	Frekvencija	Korijen	Interval
A	440.000	$\sqrt[12]{2^0}$	Čista prima (Č1)
Ais / B	466.164	$\sqrt[12]{2^1}$	Mala sekunda (m2)
H	493.883	$\sqrt[12]{2^2}$	Velika sekunda (V2)
C	523.251	$\sqrt[12]{2^3}$	Mala terca (m3)
Cis / Des	554.365	$\sqrt[12]{2^4}$	Velika terca (V3)
D	587.330	$\sqrt[12]{2^5}$	Čista kvarta (Č4)
Dis / Es	622.254	$\sqrt[12]{2^6}$	Tritonus (Pov4/Sm5)
E	659.255	$\sqrt[12]{2^7}$	Čista kvinta (Č5)
F	698.456	$\sqrt[12]{2^8}$	Mala seksta (m6)
Fis / Ges	739.989	$\sqrt[12]{2^9}$	Velika seksta (V6)
G	783.991	$\sqrt[12]{2^{10}}$	Mala septima (m7)
Gis / As	830.609	$\sqrt[12]{2^{11}}$	Velika septima (V7)
A2	880.000	$\sqrt[12]{2^{12}}$	Čista oktava (Č8)

(Autorski rad)

Svjetloplavom su bojom označeni intervali na kojima će se prikazati razlike između Pitagorejskog štimanja i štimanja jednakog temperameta. U svrhu ove usporedbe biti će prikazana jedna žica učvršćena na oba kraja te stojni valovi koji se mogu pojaviti na istoj.



Slika 8: Primjer stojnog vala (autorski rad)

Kroz formule se navedeni stojni val može prikazati na slijedeći način:

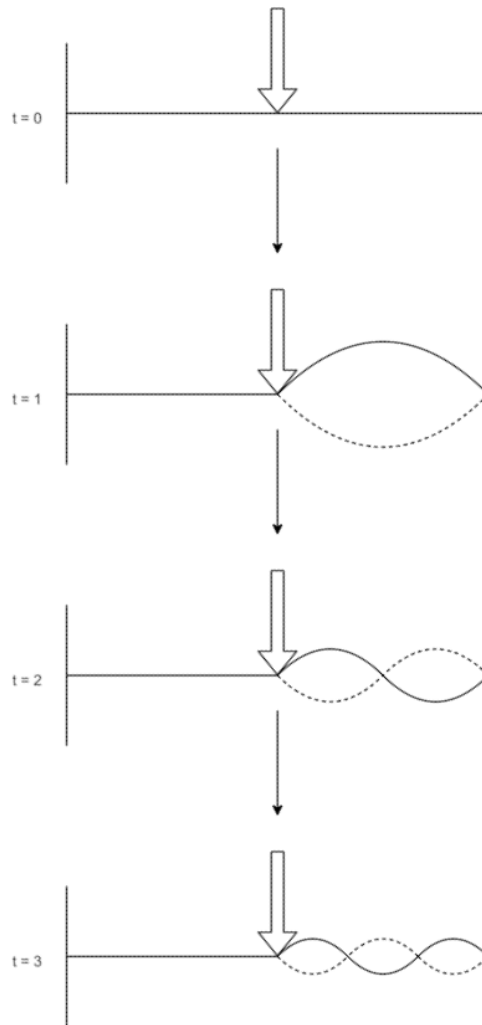
$$l = \frac{\lambda}{2}, \lambda = 2 \cdot l, f = \frac{v}{2l}$$

gdje je l duljina žice, λ valna duljina, f frekvencija te v brzina vala. U početku je $\lambda = 2 \cdot l$ zbog značajke stojnog vala da je ono što titra uvijek jednako pola vrijednosti valne duljine. U

daljnjim će primjerima stojni val sa slike 8 biti korišten kao referenca za izračune te će vrijednost frekvencije ovog vala biti $f_0 = 440\text{Hz}$. Slijedeći će primjeri istražiti što se točno dogodi kada se žica pritisne u svakom od Pitagorejskih omjera.

4.1.1. Pola žice - interval oktave

Pitagorejci su otkrili kako se pritiskom na polovicu žice dobiva interval oktave:



Slika 9: Primjer stojnog vala na žici pritisnutoj na polovici (autorski rad)

Ako je frekvencija prazne odnosno nepritisnute žice 440 Hz, slijedi:

$$f_0 = 440\text{Hz} = \frac{v}{2l} = \frac{v}{2 \cdot \frac{\lambda}{2}} = \frac{v}{\lambda}$$

što vodi do zaključka da je $\frac{v}{\lambda} = 440\text{Hz}$.

Žica pritisnuta na polovici svoje duljine podrazumijeva da je:

$$l_1 = \frac{1}{2}l = \frac{1}{2} \cdot \frac{\lambda}{2} = \frac{\lambda}{4}.$$

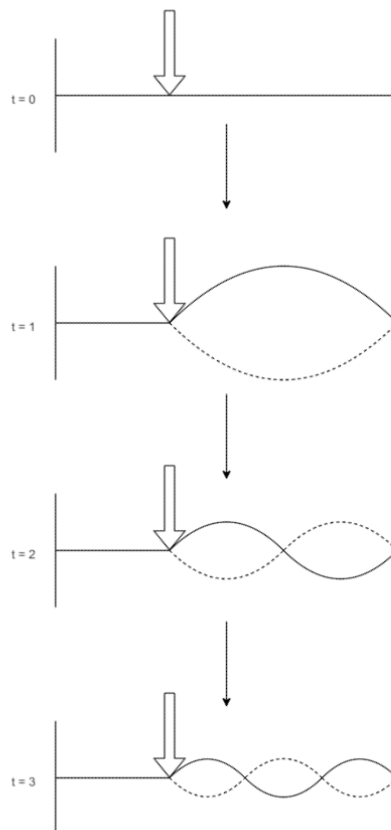
Kod žice pritisnute na polovici duljine, titra samo pola žice, a ne cijela žica. Znajući da je ono što titra pola valne duljine, dolazimo do zaključka da je u ovom slučaju valna duljina jednaka duljini žice. Uzimajući ovo u obzir, vrijedi slijedeće:

$$f_1 = \frac{v}{2l_1} = \frac{v}{2 \cdot \frac{\lambda}{4}} = \frac{v}{\frac{\lambda}{2}} = 2 \cdot \frac{v}{\lambda} = 2 \cdot 440Hz = 880Hz.$$

Ova se vrijednost točno poklapa sa vrijednošću frekvencije za interval oktave.

4.1.2. Trećina žice - interval kvinte

Pitagorejci su također otkrili kako se pritiskom žice na 1/3 njezine duljine dobiva interval kvinte:



Slika 10: Primjer stojnog vala na žici pritisnutoj na trećini (autorski rad)

Žica pritisnuta na trećini svoje duljine podrazumijeva da dvije trećine žice titraju:

$$l_2 = \frac{2}{3}l = \frac{2}{3} \cdot \frac{\lambda}{2} = \frac{\lambda}{3}.$$

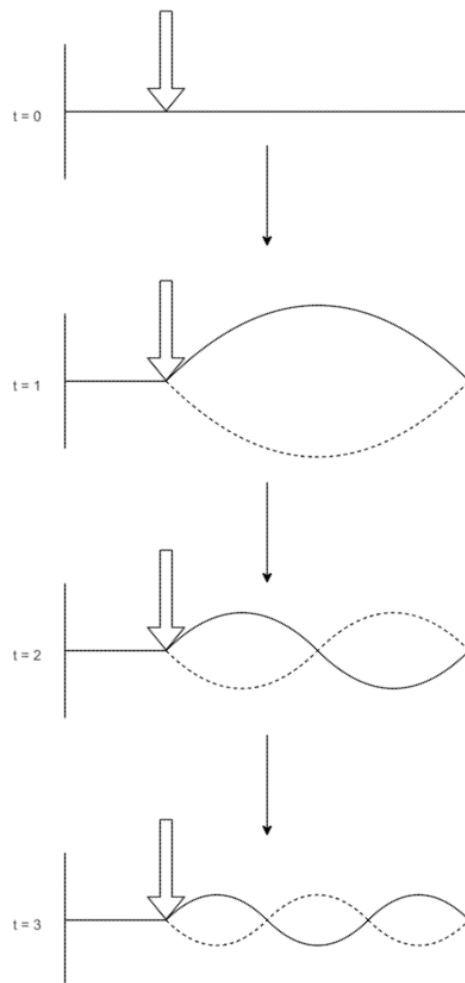
Uzimajući ovo u obzir, vrijedi slijedeće:

$$f_2 = \frac{v}{2l_2} = \frac{v}{2 \cdot \frac{\lambda}{3}} = \frac{v}{\frac{2\lambda}{3}} = \frac{3}{2} \cdot \frac{v}{\lambda} = \frac{3}{2} \cdot 440 \text{ Hz} = 660 \text{ Hz}.$$

Ova je vrijednost za 0.745 Hz viša od vrijednosti intervala kvinte iz tablice 1.

4.1.3. Četvrtina žice - interval kvarte

Pitagorejci su isto tako otkrili kako se pritiskom žice na 1/4 duljine dobiva interval kvarte:



Slika 11: Primjer stojnog vala na žici pritisnutoj na četvrtini (autorski rad)

Žica pritisnuta na četvrtini svoje duljine podrazumijeva da tri četvrtine žice titraju:

$$l_3 = \frac{3}{4}l = \frac{3}{4} \cdot \frac{\lambda}{2} = \frac{3\lambda}{8}.$$

Uzimajući ovo u obzir, vrijedi slijedeće:

$$f_3 = \frac{v}{2l_3} = \frac{v}{2 \cdot \frac{3\lambda}{8}} = \frac{v}{\frac{3\lambda}{4}} = \frac{4}{3} \cdot \frac{v}{\lambda} = \frac{4}{3} \cdot 440\text{Hz} = 586.67\text{Hz}.$$

Ova je vrijednost za 0.66 Hz niža od vrijednosti iz tablice 1.

4.2. Fourierove transformacije

4.2.1. Povijest i vrste transformacija

Fourierova analiza je skup matematičkih metoda i tehnika koje se temelje na rastavljanju signala na sinusoide. Dobila je naziv po francuskom znanstveniku imena Jean Baptiste Joseph Fourier, koji je prvi ukazao na brojne praktične mogućnosti korištenja navedenih tehnika. Premda se njegov prvobitni znanstveni rad bavio propagacijom topline, u njemu je naveo tada-kontroverznu tvrdnju da se svaki neprekidni periodični signal može prikazati kao suma određenih sinusoida. Postoje četiri različite Fourierove transformacije, koje se bave sa svakom od četiri mogućih vrsta signala. Signal može biti:

- Periodičan - Isti se isječak signala ponavlja od $-\infty$ do $+\infty$.
- Aperiodičan - Npr. od $-\infty$ do n je vrijednost signala 0, promijeni se te je nakon promjene vrijednost signala opet 0 do $+\infty$.
- Neprekidan - Signal je prikazan u obliku crte na grafu.
- Diskretan - Signal je prikazan u obliku skupa određenih točaka na grafu.

Koristeći ova četiri atributa, određujemo četiri različite vrste signala te nazive njihovih transformacija:

- Neprekidni aperiodični signal - Fourierova transformacija
- Neprekidni periodični signal - Fourierov red
- Diskretni aperiodični signal - Diskretna vremenska Fourierova transformacija
- Diskretni periodični signal - Diskretna Fourierova transformacija

Za uspješno objašnjenje FFT algoritma, fokus će biti na diskretnoj Fourierovoj transformaciji jer ona čini temelj FFT algoritma. Prije nego se objasni princip rada FFT algoritma, potrebno je definirati neke od značajki DFT-a.

Ulazni signal DFT-a se navodi kao signal u *vremenskoj domeni*. Ovaj se naziv koristi jer se generalno ulazni signal očituje kao kolekcija uzoraka snimljenih u određenim vremenskim intervalima. Izlazni podaci se također nazivaju i signal u *frekvencijskoj domeni*. Ovisno o namjeni, izlazni podaci mogu biti realni brojevi ili kompleksni brojevi. U svrhu objašnjenja FFT

algoritma, koristit će se kompleksni brojevi kao i u popratnom programskom rješenju. Dakle, DFT uzima signal u vremenskoj domeni u obliku skupa realnih brojeva te ih pretvara u signal u frekvencijskoj domeni u obliku skupa *kompleksnih* brojeva. Za izračun DFT-a najčešće se uzimaju duljine signala koje su potencije broja 2. Jedan od razloga takvoj duljini signala je činjenica da su potencije broja 2 "prirodne" računalu pošto sva računala koriste binarno adresiranje podataka.

4.2.2. *Fast Fourier Transform* algoritam

Jedan od problema kod prevođenja Fourierovih transformacija u računalni program je slijedeće: kako računalo, koje ima konačnu duljinu ulaza, može izračunati bilo koji od navedenih signala ako su svi signali nad kojima se može izvršiti Fourierova transformacija beskonačni? Odgovor je da treba promijeniti podatke dobivene ulazom da izgledaju kao da su beskonačni. Računalo može shvatiti samo one signale koji su diskretni i konačni. Ako se podaci promijene tako da za svrhe izračuna izgledaju beskonačno, može se primjeniti diskretna Fourierova transformacija.

Na primjer, uzmemo li ulazni diskretni signal duljine 500 točaka, moramo ga dopuniti s nulama do duljine jedne od potencija broja 2. Najbliža je potencija 512, te se na kraj signala doda 12 nula. Ovaj se signal sada može ponoviti koliko je god puta potrebno da bi računalo moglo izračunati izlazne podatke.

FFT algoritam jedan je od najvažnijih i najkompliciranijih algoritama koji postoje. Otkriće ovog algoritma se pripisuje J.W. Cooleyu i J.W. Tukeyu 1965. godine, premda ga je koristio i njemački matematičar Karl Freidrich Gauss više od jednog stoljeća prije Cooleya i Tukeya. Njegovo se temeljno funkcioniranje može opisati na slijedeći način: signal u vremenskoj domeni od N točaka se raspodijeli na N signala u vremenskoj domeni koji se sastoje od jedne točke. Zatim se za svaki od N signala izračuna frekvencijski spektar te se dobivenih N frekvencijskih spektara sastavlja u jedan frekvencijski spektar. Zanimljiva je činjenica da za računanje frekvencijskog spektra signala koji se sastoji od jedne točke ne treba napraviti ništa: frekvencijski spektar signala s jednom točkom je jednak samom sebi.

Prvi dio, odnosno raspodjela signala sa N točaka na N signala s jednom točkom, postiže se vrlo jednostavnom metodom: invertiranjem bitova (eng. *bit reversal*). Za bilo koji dobiveni signal se naprave slijedeći koraci:

- Kao primjer signala ćemo uzeti brojeve 3, 6, 10 i 15.
- 1. razdioba signala na parne i neparne vrijednosti te nakon toga na zasebne signale

$$[3, 6, 10, 15] \rightarrow [6, 10], [3, 15] \rightarrow [6], [10], [3], [15]$$

- 2. pretvorba vrijednosti signala u binarni oblik

$$[6] = [0110], [10] = [1010], [3] = [0011], [15] = [1111]$$

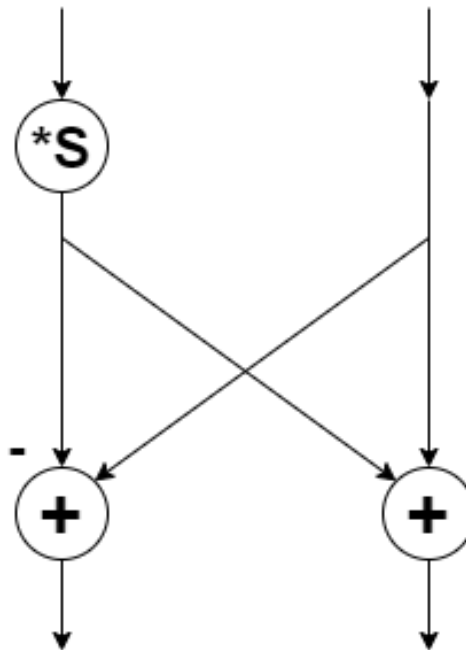
- 3. inverzija bitova

$$[0110] \rightarrow [1001] = [9], [1010] \rightarrow [0101] = [5], [0011] \rightarrow [1100] = [12], [1111] \rightarrow [0000] = [0]$$

- 4. sastavljanje frekvencijskog spektra točno obrnutim redoslijedom iz 1. koraka

$$[9], [5], [12], [0] \rightarrow [12, 0], [9, 5] \rightarrow [12, 9, 5, 0]$$

Ovaj je primjer vrlo pojednostavljen prikaz funkcioniranja FFT algoritma, ali obuhvaća srž onoga što se događa. U većim se signalima postepeno komplicira dio sastavljanja frekvencijskog spektra zbog puno većih duljina signala. Do ovoga problema dolazi zato što se, ovisno o duljini signala, prvi korak mora izvesti na nekoliko razina (u primjeru su to 2. i 3. razdioba u 1. koraku), te sukladno s tim i četvrti korak. Za izračun će trebati razina jednako broju koji je potencija duljine signala. (npr. signal dug 16 točaka zahtjeva 4 razine, signal dug 64 točke zahtjeva 6 razina itd.) U tom se slučaju koristi kalkulacijski element poznat pod nazivom "leptir". Navedeni element uzima dvije ulazne točke te ih pretvara u dvije izlazne točke.



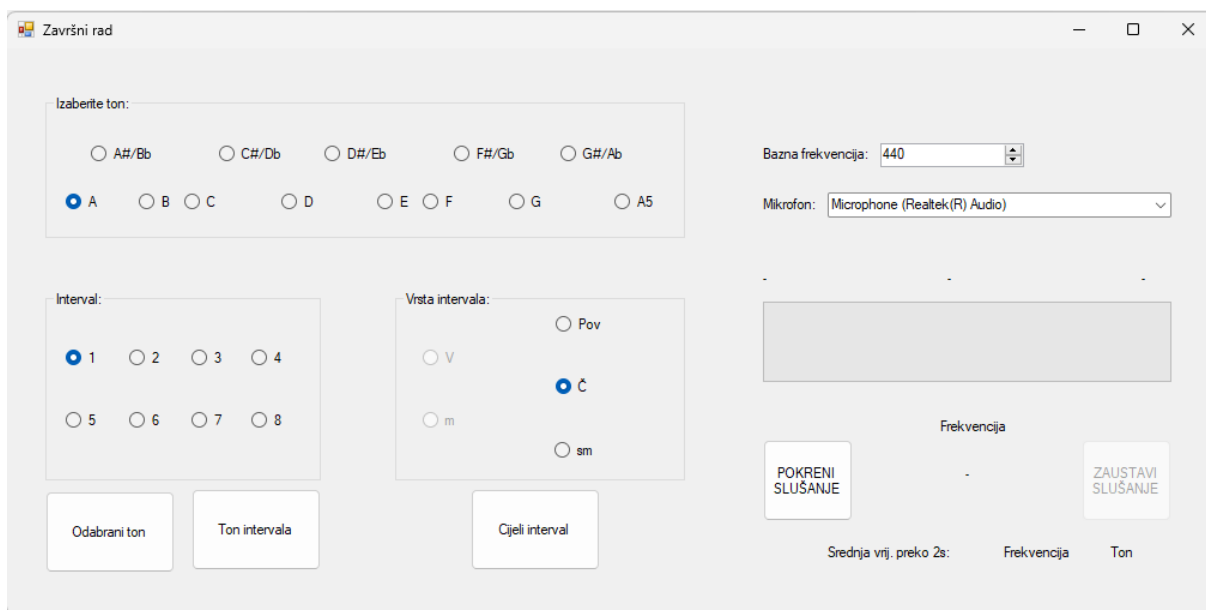
Slika 12: Izgled jednog FFT leptira (Prema: Smith, 1999)

Ovaj se kalkulacijski element koristi na svakoj razini kalkulacije frekvencijskog spektra. Na slici 12 je u slijedu vidljiv jedan krug u kojem je upisano "*S". Prilikom pretvorbe signala iz vremenske u frekvencijsku domenu, vremenska se domena za određene signale popunjava s nulama na parnim mjestima, a za određene na neparnim mjestima. Ovo se na pretvoreni signal preslikava tako da se vrijednosti frekvencijske domene ponove. Ovaj se proces mora napraviti i na signalima s parno popunjenim nulama i na signalima s neparno popunjenim nulama. Za signale s neparno popunjenim nulama, ponavljanje vrijednosti se postiže množenjem sa sinusoidom. Nakon navedenog množenja, dva se elementa mogu sumirati uz pomoć navedenog FFT leptira na svakoj razini.

5. Programska dokumentacija

U ovom će dijelu rada biti prikazan dizajn korisničkog sučelja te će biti objašnjen programski kod koji se koristi unutar samog programa.

5.1. Korisničko sučelje



Slika 13: Korisničko sučelje programskog rješenja (autorski rad)

Korisničko sučelje programa se sastoji od dva dijela:

- Odjeljak za reprodukciju tonova i intervala (lijevo)
- Odjeljak za štimer (desno)

5.1.1. Reprodukcija tonova i intervala

Odjeljak za reprodukciju tonova i intervala sadrži tri odabira i tri gumba:

- Odabir temeljnog tona intervala
- Odabir razmaka odnosno intervala
- Odabir vrste intervala
- Gumb za reprodukciju odabranog temeljnog tona
- Gumb za reprodukciju odabranog intervala na temeljni ton
- Gumb za reprodukciju cijelog intervala, odnosno temeljnog tona i tona intervala

Nakon što se odabere temeljni ton, odabire se interval koji se želi reproducirati. Valja napomenuti da intervali prime (1), kvarte (4), kvinte (5) i oktave (8) mogu biti čisti (Č), povećani odnosno povišeni (Pov) te smanjeni (sm), dok intervali sekunde (2), terce (3), sekste (6) i septime (7) mogu biti samo veliki (V) ili mali (m). Ova je limitacija vidljiva i u programskom kodu te će biti objašnjena logika iste. Odabirom željene kombinacije temeljnog tona, intervala i vrste intervala se može stisnuti jedan od triju gumba: "Odabrani ton", "Ton intervala" i "Cijeli interval". Pritiskom gumba "Odabrani ton" se reproducira odabrani temeljni ton. Gumb koji se nalazi odmah do njega, "Ton intervala", uzima u obzir temeljni ton, odabrani interval te vrstu intervala i reproducira traženi interval. Gumb "Cijeli interval" je kombinacija spomenuta dva gumba te reproducira prvo temeljni ton te nedugo nakon njega reproducira odabrani interval.

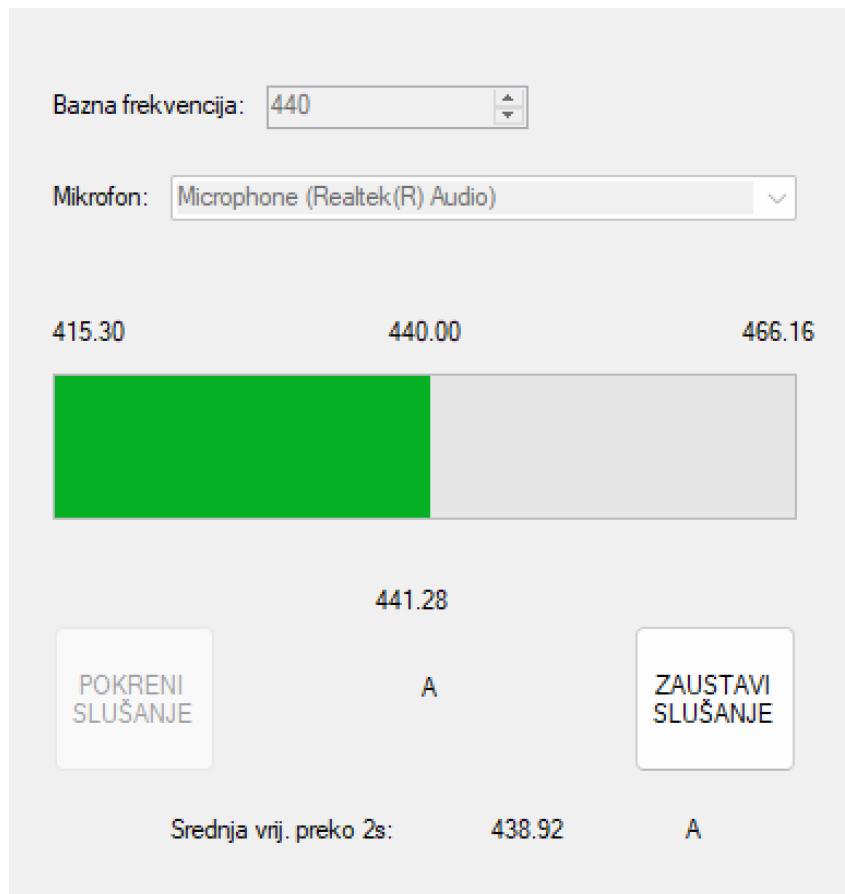
5.1.2. Odjeljak za štimer

Odjeljak za štimer izrađen je od slijedećih komponenti:

- Brojčani odabir bazne frekvencije
- Odabir mikrofona
- Dinamički prikaz frekvencija i točnosti dobivenog signala
- Gumb za pokretanje slušanja
- Gumb za zaustavljanje slušanja
- Prikaz srednje vrijednosti frekvencija i tona preko dvije sekunde snimanja

Prije samog pokretanja snimanja, korisnik mora odabrati koju baznu frekvenciju želi koristiti kao referencu za štimanje. Utjecaj ovog broja će biti prikazan u objašnjenju programskog koda. Pri pokretanju programa, bazna će frekvencija biti postavljena na 440 Hz te se maksimalno može povećati do 444 Hz. Također će pri pokretanju programa biti popunjen odabir mikrofona s trenutnim sistemski zadanim mikrofonom, što je vidljivo na slici 13. Pritiskom na gumb za pokretanje slušanja navedenim će odabirima biti onemogućene promjene te će se upogoniti dinamički prikaz frekvencija.

Dok je slušanje upaljeno, signali dobiveni kroz mikrofona se obrađuju i prikazuju na dinamičkom prikazu frekvencija uz pomoć nekoliko vrijednosti. S gornje strane prikaza nalaze se tri vrijednosti, od kojih je srednja vrijednost frekvencija tona koju se želi dostići. Vrijednost lijevo je frekvencija jednog polutona niže, dok je vrijednost desno frekvencija jednog polutona više. Ispod ovih triju vrijednosti te ispod prikaza nalazi se dominantna frekvencija dobivenog signala te ton koji je najbliži navedenoj frekvenciji. Dodatno je ispod toga postavljena još jedna frekvencija, koja kumulacijom snage frekvencija preko dvije sekunde preciznije prikazuje dominantnu frekvenciju signala. Uz to se, naravno, prikazuje i ton koji pripada danoj frekvenciji.



Slika 14: Upaljeni štimer (autorski rad)

5.2. Programski kod

Programski kod sadrži dva glavna dijela: klasu *Tonovi.cs* te formu *Form1.cs*.

5.2.1. Klasa *Tonovi.cs*

5.2.1.1. Strukture i atributi

Klasa *Tonovi.cs* sadrži slijedeće strukture i atribute:

- `public Dictionary<string, double> ToneBases`

→ Struktura koja sadrži parove ključ-vrijednost u obliku ton-izračunata frekvencija.

- `public double currentBaseFreq`

→ Atribut koji sadrži trenutnu baznu frekvenciju.

Pri pozivu konstruktorske funkcije atribut *currentBaseFreq* dobiva vrijednost. Na temelju ovog atributa se izračunavaju vrijednosti frekvencija svakog tona u strukturi *ToneBases* uz pomoć funkcije *multiplierRoot(n)*.

5.2.1.2. Funkcije

Klasa *Tonovi.cs* sadrži slijedeće funkcije:

- `public double multiplierRoot(double n)`

→ Funkcija koja vraća decimalni broj dobiven formulom $2^{\frac{n}{12}}$, odnosno $\sqrt[12]{2^n}$.

- `public Tonovi(double baseFreq)`

→ Konstruktorska funkcija. Postavlja vrijednost atributa *currentBaseFreq* na temelju numeričkog unosa te izračunava vrijednosti frekvencija svakog tona uz pomoć atributa *currentBaseFreq* i funkcije *multiplierRoot(n)* te ih upisuje u strukturu *ToneBases*.

- `public double normalizeFrequency(double currVal)`

→ Funkcija koja uzima trenutnu frekvenciju i skalira ju na vrijednost od 415 do 870 Hz.

- `public string[] getCurrentToneUDVals(double currVal)`

→ Funkcija koja za ulazni podatak uzima danu frekvenciju, a vraća u jednom polju prepoznati ton, točnu frekvenciju prepoznatog tona, frekvenciju jednog polutona ispod i iznad prepoznatog tona te vrijednost frekvencije nakon izvršavanja funkcije *normalizeFrequency*.

- `public double getOctaveMultiplier(double freq)`

→ Funkcija koja za unesenu frekvenciju vraća broj kojim se frekvencija mora pomnožiti kako bi ona bila u rasponu od 415 do 870 Hz.

5.2.2. Forma *Form1.cs*

Forma *Form1.cs* sadrži funkcije vezane uz omogućavanje i onemogućavanje gumba u sučelju u obliku događaja te atribute i strukture koje se koriste za preračunavanje podataka i njihovo privremeno spremanje.

5.2.2.1. Strukture i atributi

Forma *Form1.cs* sadrži slijedeće atribute:

- `public int baznaFrekvencija`

→ Cjelobrojni atribut u koji će se popunjavati bazna frekvencija.

- `WaveInEvent waveInVar`

→ Struktura koja proizlazi iz biblioteke *NAudio*. Ova je struktura potrebna kako bi programski kod mogao pročitati vrijednosti signala dobivenog mikrofonom.

- `public double[] bufferVals`

→ Polje decimalnih brojeva u koje se upisuje neobrađeni signal.

- `public Tonovi ToneBase`

→ Atribut tipa klase *Tonovi* koji sadrži sve atribute i funkcije navedene za tu klasu.

- `public double[] bigbuffer`

→ Polje decimalnih brojeva u kojem se kumuliraju vrijednosti snaga frekvencija kroz dvije sekunde snimanja.

5.2.2.2. Funkcije

Forma *Form1.cs* sadrži funkcije koje čine srž cijelog programskog rješenja:

- `public Form1()`

→ Konstruktorska funkcija. Unutar nje se pronalaze spojeni mikrofoni i dodaju se u odabir mikrofona te se prvi dostupni postavlja kao odabrani. Također se unutar ove funkcije provjerava vrijednost u numeričkom odabiru bazne frekvencije te se atribut *baznaFrekvencija* postavlja na tu vrijednost. To omogućava inicijaliziranje klase *Tonovi* pod nazivom *ToneBase*.

- `void DisableIneligible()`

→ Funkcija koja, ovisno o odabranom intervalu, omogućuje odnosno onemogućuje određene vrste intervala.

- `RadioButton GetOznaceniRadio(Control container)`

→ Funkcija koja vraća trenutno označeni radio-gumb unutar spremnika *container* sa svim njegovim atributima. Atributi radio-gumba su važni za daljnje računanje intervala koji će biti reproduciran.

- `public int[] izracunajInterval(double baseTone, int interval, string intervalType)`

→ Funkcija koja kao ulazni podatak uzima decimalnu vrijednost frekvencije odabranog baznog tona, cjelobrojni razmak odnosno interval te vrstu intervala. Na temelju ovih podataka se računa ukupni množitelj kojim će se vrijednost frekvencije baznog tona pomnožiti kako bi se dobio točan interval. Ova funkcija vraća polje koje sadrži dva cijela broja: frekvenciju baznog tona te frekvenciju preračunatog intervala. Ove vrijednosti moraju biti cjelobrojne jer funkcija koja reproducira zvuk prihvaća samo cijele brojeve kao parametar.

Slijedeće su funkcije takozvani rukovatelji događaja:

- **private void** WaveIn_DataAvailable(**object** sender, EventArgs e)

→ Funkcija koja će se izvršiti svaki put kada postoje podaci signala. Ova je funkcija temelj snimanja zvučnog signala.

- **private void** timer1_Tick(**object** sender, EventArgs e)

→ Najvažnija funkcija cijelog programa. Izvodi se svakih 0.1 sekundi pošto je vremenski okidač postavljen na 100 milisekundi. Decimalni brojevi koji se nalaze u polju *bufferVals*, odnosno čisti signal, se kopira u novi niz. Taj se niz zatim pomnoži sa dvije "prozorske funkcije" koje smanjuju spektralno curenje te čine signal periodičnim. Pošto FFT algoritam zahtjeva periodični signal koji je duljine neke potencije broja 2, mora se također izvršiti funkcija nul-punjenja (eng. *zero-padding*). Funkcija nul-punjenja osigurava da je signal duljine potencije broja 2. Na takvom se nizu zatim izvršava FFT algoritam koji dobiveni niz decimalnih brojeva pretvara u niz kompleksnih brojeva, odnosno u spektar. Iz dobivenog spektra se mogu dobiti snage određenih frekvencija u obrađenom signalu te se na temelju najjače frekvencije prikazuju i mijenjaju vrijednosti u dinamičkom prikazu frekvencija. Ove se snage također pribrajaju u polje *bigbuffer*, koje će se koristiti u slijedećoj funkciji.

- **private void** timer2_Tick(**object** sender, EventArgs e)

→ Druga najvažnija funkcija u programu. Na početku funkcije je moguće u programskom kodu odrediti varijablu faktor koja će se koristiti u slijedećim izračunima. Na temelju kumuliranih vrijednosti upisanih u polje *bigbuffer* se provodi slijedeći algoritam: Izračunavaju se minimalna i maksimalna vrijednost niza *bigbuffer*. Stvaraju se dva nova privremena niza. U prvi se privremeni niz upisuju normalizirane snage u vrijednosti između 0 i 1 uz pomoć iterativne formule: $X - \min/\max - \min$, gdje je X trenutni broj u nizu. Pomoću srednje vrijednosti prvog privremenog niza se dobiva granica i to preko formule: $G = Avg + (1 - Avg) \cdot F$, gdje je *Avg* srednja vrijednost prvog privremenog niza te *F* zadani faktor koji je početno postavljen na 0.995. Na temelju vrijednosti prvog privremenog niza provjerava se i upisuje slijedeće: je li trenutni broj prvog niza manji od granice? Ako je, u drugi niz se upisuje nula; u suprotnom se piše vrijednost iz prvog niza. Ovim se postupkom dobije frekvencija s najvećom snagom u dvije sekunde snimanja. Na kraju funkcije se vrijednosti u nizu *bigbuffer* inicijaliziraju.

- **private void** cmbMics_SelectedIndexChanged(**object** sender, EventArgs e)

→ Funkcija koja se izvršava kada se promijeni odabir mikrofona. Unutar funkcije se ponovno postavlja atribut *waveInVar*.

- **private void** buttonListen_Click(**object** sender, EventArgs e)

→ Funkcija koja pokreće snimanje zvuka dok se klikne gumb "POKRENI SLUŠANJE". Ova funkcija također pokreće vremenske okidače *timer1* i *timer2* te poziva metodu atributa *waveInVar* koja započinje snimanje.

- `private void buttonStop_Click(object sender, EventArgs e)`

→ Funkcija koja zaustavlja snimanje zvuka, zaustavlja vremenske okidače i inicijalizira dinamički prikaz frekvencija dok se klikne na gumb "ZAUSTAVI SLUŠANJE".

- `private void buttonPlay_Click(object sender, EventArgs e)`

→ Funkcija koja reproducira odabrani bazni ton u odjeljku za reprodukciju tona klikom na gumb "Odabrani ton".

- `private void buttonTonIntervala_Click(object sender, EventArgs e)`

→ Funkcija koja reproducira odabrani interval na bazni ton u odjeljku za reprodukciju tona klikom na gumb "Ton intervala".

- `private void buttonOdvojeno_Click(object sender, EventArgs e)`

→ Funkcija koja reproducira odabrani bazni ton te odabrani interval na bazni ton u odjeljku za reprodukciju tona klikom na gumb "Cijeli interval".

- `private void numUDBaznaFrekv_ValueChanged(object sender, EventArgs e)`

→ Funkcija koja ažurira vrijednosti u strukturi *ToneBase* svaki put kada se promijeni bazna frekvencija u odabiru bazne frekvencije.

Postoji još osam funkcija koje spadaju u rukovatelje događaja, ali sve sadrže istu funkciju unutar sebe. Ta je funkcija *DisableIneligible()*, koja mijenja omogućene vrste intervala ovisno o odabranom intervalu. Za svaki radio-gumb u odjeljku "Interval" (prima, sekunda, terca, kvarta, kvinta, seksta, septima, oktava) postoji ovakva funkcija:

- `private void radioButtonPrima_CheckedChanged(object sender, EventArgs e)`

→ Funkcija koja klikom na radio-gumb za interval prime onemogućava izbor vrste intervala V i m.

6. Zaključak

Predmet ovog završnog rada je FFT algoritam te njegovo korištenje za procesiranje mikrofonskoga signala u frekvencije koje se naposljetku mogu koristiti za štimanje instrumenata. Za izradu programskog rješenja korišten je programski jezik C# i Windows Forms sučelje te Microsoft Visual Studio 2022. Prije izrade ovog završnog rada, imao sam pretpostavku da neću otkriti puno novih stvari. Ta je pretpostavka nestala od prvog dana kako sam počeo izrađivati rad. Kroz povijesni pregled naučio ponešto o matematici i glazbi, ali i neke stvari koje nikad ne bih mogao očekivati. Kroz matematički opis sam ponovio znanje o stojnom valu, potvrdio neke zanimljive činjenice te shvatio srž djelovanja jednog od najvažnijih algoritama računalnog doba.

Popis literature

- [1] K. Rötzel. „Monokord [Slika].” (2009.), adresa: <https://snl.no/monokord>. Pristupano 29.8.2023.
- [2] P. Zenić. „Pitagora i glazba.” (bez dat.), adresa: <http://likovna-kultura.ufzg.unizg.hr/Ucimo-gledati-zine/Broj%201/pitagora%20i%20glazba.htm>. Pristupano 17.7.2023.
- [3] *Versajski ugovor*. 1919.
- [4] I. O. for Standardization. „ISO 16:1975 - Acoustics — Standard tuning frequency (Standard musical pitch).” (siječanj 1975.), adresa: <https://www.iso.org/standard/3601.html>. Pristupano 23.08.2023.
- [5] S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 1999., ISBN: 0-9660176-6-8.

Popis slika

1.	Izgled tetraktisa (autorski rad)	3
2.	Primjer monokorda (Izvor: [1])	4
3.	Izgled tabele s Rafaelove slike <i>Atenska škola</i> (autorski rad)	5
4.	Vizualno objašnjenje tabele (autorski rad)	5
5.	Ilustracija kvintnog kruga (autorski rad)	6
6.	Dio Versajskog ugovora (Izvor: <i>Versajski ugovor</i>)	8
7.	Opis standarda ISO 16 na službenim stranicama ISO-a. (Izvor: [4])	9
8.	Primjer stojnog vala (autorski rad)	11
9.	Primjer stojnog vala na žici pritisnutoj na polovici (autorski rad)	12
10.	Primjer stojnog vala na žici pritisnutoj na trećini (autorski rad)	13
11.	Primjer stojnog vala na žici pritisnutoj na četvrtini (autorski rad)	14
12.	Izgled jednog FFT leptira (Prema: Smith, 1999)	17
13.	Korisničko sučelje programskog rješenja (autorski rad)	18
14.	Upaljeni štimer (autorski rad)	20

Popis tablica

1. Prikaz vrijednosti frekvencije, korijena i intervala koji ton čini s baznim tonom A . 10