



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

ELEKTRONIKOS FAKULTETAS

KOMPIUTERIŲ INŽINERIJOS KATEDRA

Arūnas Survila

**GARSO ATPAŽINIMO SISTEMOS TYRIMAS**

**INVESTIGATION OF SOUND RECOGNITION SYSTEM**

Baigiamasis magistro darbas

Kompiuterių inžinerijos studijų programa, valstybinis kodas 621H69001

Kompiuterių technologijų specializacija

Elektronikos ir elektros inžinerijos studijų kryptis

Vilnius, 2012

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS  
ELEKTRONIKOS FAKULTETAS  
KOMPIUTERIŲ INŽINERIJOS KATEDRA

TVIRTINU

**Katedros vedėjas**

\_\_\_\_\_  
(Parašas)  
prof. dr. Algirdas Baškys  
(Vardas, pavardė)

\_\_\_\_\_  
(Data)

Arūnas Survila

**GARSO ATPAŽINIMO SISTEMOS TYRIMAS**  
**INVESTIGATION OF SOUND RECOGNITION SYSTEM**

Baigiamasis magistro darbas

Kompiuterių inžinerijos studijų programa, valstybinis kodas 621H69001

Kompiuterių technologijų specializacija

Elektronikos ir elektros inžinerijos studijų kryptis

**Vadovas**

prof. dr. Dalius Navakauskas \_\_\_\_\_  
(Moksl. laipsnis/pedag. vardas, vardas, pavardė) (Parašas) (Data)

**Lietuvių kalbos konsultantas**

asist. Daiva Macko \_\_\_\_\_  
(Moksl. laipsnis/pedag. vardas, vardas, pavardė) (Parašas) (Data)

Vilnius, 2012

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS  
ELEKTRONIKOS FAKULTETAS  
KOMPIUTERIŲ INŽINERIJOS KATEDRA

MAGISTRANTŪROS STUDIJŲ BAIGIAMOJO DARBO UŽDUOTIS

*Elektronikos fakultetas*  
*Kompiuterių inžinerijos katedra*  
*Technikos mokslų sritis*  
*Elektros ir elektronikos inžinerijos mokslo kryptis*  
*Elektronikos inžinerijos studijų kryptis*  
*Kompiuterių inžinerijos studijų programa*  
*Kompiuterių technologijų specializacija*

TVIRTINU  
Kompiuterių inžinerijos katedros vedėjas  
prof. Algirdas Baškys \_\_\_\_\_  
2011 m. \_\_\_\_\_ mėn. \_\_\_\_\_ d.

Studentui (ei) \_\_\_\_\_ **Arūnui Survilai** \_\_\_\_\_  
(vardas, pavardė)

Baigiamojo darbo tema: \_\_\_\_\_ **Garsų atpažinimo sistemos tyrimas** \_\_\_\_\_

(lietuvių kalba)

Investigation of Sound Recognition System

(anglų kalba)

Patvirtinta 2011 m. lapkričio mėn. 11 d. dekanų įsakymu Nr. 200el

Baigiamojo darbo užbaigimo terminas 2012 m. birželio mėn. 01 d.

Darbo tikslas: Ištirti gitaros garsų atpažinimo realiuoju laiku galimybę, sukuriant sistemą, atpažįstančią gitaros garsus.

BAIGIAMOJO DARBO UŽDUOTIS:

1. Atlikti garsų atpažinimo metodų analitinę apžvalgą.
2. Atrinkti ir vienoje sistemoje įgyvendinti bent vieną gitaros garsų atpažinimo algoritmą.
3. Sudaryti originalų gitaros garsyną.
4. Atlikti sukurtos gitaros garsų atpažinimo sistemos tyrimus.

Baigiamojo darbo rengimo konsultantai: \_\_\_\_\_ - \_\_\_\_\_

(vardas, pavardė, mokslinis laipsnis ir vardas)

Vadovas: \_\_\_\_\_  
(parašas)

prof. dr. Dalius Navakauskas  
(vardas, pavardė, mokslinis laipsnis ir vardas)

Užduotį gavau: \_\_\_\_\_  
(studento parašas)

\_\_\_\_\_  
(data)

Vilniaus Gedimino technikos universitetas

**Elektronikos** fakultetas

**Kompiuterių inžinerijos** katedra

ISBN ISSN

Egz. sk. ....

Data ....-....-....

**Kompiuterių technologijų** studijų programos baigiamasis magistro darbas

Pavadinimas **Garso atpažinimo sistemos tyrimas**

Autorius **Arūnas Survila**

Vadovas prof. dr. **Dalius Navakauskas**

Kalba

lietuvių

užsienio

### **Anotacija**

Baigiamajame magistro darbe nagrinėjamos gitaros garsų (natų) atpažinimo problemos ir būdai, taikomi joms spręsti. Išnagrinėti pagrindiniai metodai, kuriais remiantis galima išskirti dominuojančius dažnius spektre ir taip atpažinti natas. Atliekant baigiamąjį darbą sudarytas originalus gitaros garsynas, susidedantis iš 450 įrašų, atliktas skirtingų metodų taikymo modeliavimas kompiuterine programa *MATLAB*. Įvertinus gautus modeliavimo rezultatus, pasiūlyta gitaros natų atpažinimo metodika pasitelkiant greitąją Furjė transformaciją. Parašyta programa, leidžianti ją įgyvendinti mikroprocesorinėje sistemoje. Sudarytos struktūrinė bei principinė elektrinė schemas, taip pat atliktas šios sistemos kompiuterinis modeliavimas bei tyrimas. Galiausiai surinktas ir eksperimentiškai ištirtas gitaros garsų atpažinimo sistemos bandomasis maketas. Išnagrinėjus gautus realaus maketo tyrimo rezultatus, galima daryti išvadą, kad jis gitaros natas geba tinkamai atpažinti 90,49 % atvejų.

Darbą sudaro: įvadas, 5 skyriai, išvados, literatūros sąrašas.

Darbo apimtis – 56 p. teksto be priedų, 43 iliustr., 7 lent., 38 bibliografiniai šaltiniai.

Atskirai pridedami darbo priedai.

**Prasminiai žodžiai:** Atmega644, autokoreliacija, Furjė transformacija, garsų atpažinimas, gitaros natos, Hanningo langas, HSS algoritmas, požymių išskyrimas, skaitmeninis filtras.

Vilnius Gediminas Technical University  
Faculty of **Electronics**  
Department of **Computer Engineering**

ISBN ISSN  
Copies No. ....  
Date ....-....-....

Master Degree Studies **Computer Engineering** study programme Final Work

Title **Investigation of Sound Recognition**

Author **Arūnas Survila**

Academic supervisor prof. dr. **Dalius Navakauskas**

Thesis language

Lithuanian

Foreign (English)

### **Annotation**

In this thesis guitar sound (notes) recognition problems and solutions are overviewed. Common methods, which facilitate frequency separation and therefore note recognition have been analysed. An original guitar sample bank, which lists 450 tracks, has been compiled, an application modelling of different methods has been performed using MATLAB. Supported by modelling results, a fast Fourier transform-based guitar note recognition technique is suggested. Program code is implemented in a way that makes it possible to perform the latter using a microprocessor system. Block and schematic electrical circuit diagrams have been produced as well as a computer-aided modelling has been carried out. Finally, a prototype of guitar sound recognition system has been assembled and benchmarked. Obtained performance test results facilitate the conclusion that the device has successfully recognised notes in 90,49 % of cases.

Thesis consists of introduction, 5 chapters, conclusions and literature list.

Thesis lists 56 pages of written text, 43 figures, 7 tables and 38 bibliographic sources.

Appendices supplied separately.

**Keywords:** ATmega644, autocorrelation, digital filter, feature extraction, Fourier transform, Hanning window, HPS algorithm, guitar notes, sound recognition.

# TURINYS

Žymenys ir santrumpos .....	8
1. ĮVADAS .....	9
1.1. Darbo aktualumas ir tikslas .....	9
1.2. Darbo uždaviniai.....	9
1.3. Naudoti tyrimo ir analizės metodai .....	9
1.4. Darbo naujumas ir praktinė nauda.....	10
1.5. Darbo struktūra .....	10
2. GITAROS GARSŲ ATPAŽINIMO METODŲ ANALITINĖ APŽVALGA .....	11
2.1. Garso signalo požymių išskyrimo algoritmai.....	11
2.1.1. Greitoji Furjė transformacija .....	12
2.1.2. Autokoreliacija .....	12
2.1.3. Trumpalaikė Furjė transformacija .....	13
2.1.4. Vlnelių transformacija.....	14
2.1.5. Harmonikų spektrų sandaugos algoritmas.....	15
2.1.6. IIR ir FIR filtrų taikymas.....	16
2.2. Garsų klasifikavimas .....	17
2.3. Įtaisai, naudojantys garsų atpažinimo algoritmus .....	18
2.3.1. Muzikos instrumentų derinimo priemonės.....	18
2.3.2. Paukščių skleidžiamų garsų atpažinimo įtaisai .....	20
2.4. Skyriaus išvados .....	21
3. GITAROS GARSŲ SUDARYMAS .....	22
3.1. Gitaros garsų įrašymo sąlygos.....	22
3.2. Įrašytų gitaros garsų redagavimas bei katalogavimas .....	23
3.3. Skyriaus išvados .....	24
4. GITAROS GARSŲ ATPAŽINIMO ALGORITMŲ TYRIMAS <i>MATLAB</i> PROGRAMA.....	25
4.1. Pirminis garso signalų apdorojimas.....	25
4.2. Tyrimų rezultatai naudojant skaitmeninius filtrus.....	29
4.3. Tyrimų rezultatai naudojant HSS algoritmą.....	31
4.4. Tyrimų rezultatai naudojant autokoreliacinį algoritmą .....	35
4.5. Skyriaus išvados .....	37
5. GITAROS GARSŲ ATPAŽINIMO SISTEMOS PROJEKTAVIMAS .....	38
5.1. Gitaros garsų atpažinimo sistemos struktūrinės schemos sudarymas .....	38
5.2. Įtaiso principinės elektrinės schemos sudarymas .....	41
5.2.1. Mikrofono stiprintuvo schemos sudarymas.....	41
5.2.2. ŽD filtrų parinkimas .....	42
5.2.3. Mikrovaldiklio parinkimas .....	43
5.2.4. Skystųjų kristalų indikatoriaus sąsajos sudarymas.....	43
5.2.5. Maitinimo grandinės parinkimas .....	44
5.3. Mikrovaldiklio programos sudarymas.....	45
5.4. Suprojektuotos sistemos modeliavimas programa Proteus .....	48
5.5. Spausdintinės plokštės projektavimas .....	51
5.6. Skyriaus išvados .....	52
6. GITAROS GARSŲ ATPAŽINIMO SISTEMOS PATIKRA.....	53
6.1. Eksperimentų planas.....	53
6.2. Eksperimentų rezultatai .....	54
6.2.1. Pagrindinių sistemos maketo mazgų patikra .....	54
6.2.2. Tiriamojo maketo įeinančių gitaros garsų dažnių intervalų nustatymas .....	55
6.2.3. Gitaros natų atpažinimui reikalingo laiko sąnaudų skaičiavimas.....	56

6.2.4. Gitaros natų atpažinimo sistemos statistinis tikslumo įvertinimas.....	57
6.3. Skyriaus išvados .....	59
IŠVADOS .....	60
LITERATŪROS SĄRAŠAS .....	61
PRIEDAI .....	64
A priedas. Dalyvaujant XV -toje Lietuvos jaunųjų mokslininkų konferencijoje gautos pažymos kopija.....	64
B priedas. Suprojektuotos gitaros garsų atpažinimo sistemos principinė elektrinė schema.....	66
C priedas. Gitaros garsų atpažinimui tirti programa MATLAB naudotų algoritmų išeities tekstai.....	66
D priedas. Parašytos programos mikrovaldikliui ATmega644 išeities tekstai.....	69

## Žymenys ir santrumpos

$f$  virpesių dažnis, Hz  
 $k$  imties numeris  
 $N$  atskaitų skaičius  
 $x[k]$  diskretusis signalas

ASK	analogas-skaičius keitiklis
DACH	dažninė amplitudės charakteristika
DFT	diskrečioji Furjė transformacija
DLSK	dinaminis laiko skalės kraipymas
DNT	dirbtinių neuronų tinklai
DNT	dirbtiniai neuronų tinklai
FIR (angl. <i>Finite Impulse Response</i> )	ribotos impulsinės reakcijos
GFT (angl. <i>Fast Fourier transform</i> )	greitoji Furjė transformacija
GMM	Gauso mišinių modeliai
HSS (angl. <i>Harmonic product spectrum</i> )	harmonikų spektrų sandauga
IIR (angl. <i>Infinite Impulse Response</i> )	neribotos impulsinės reakcijos filtras
IS	ilgalaikė statistika
LED (angl. <i>Light-emitting diode</i> )	puslaidininkinis šviesos diodas
LPC (angl. <i>Linear predictive coding</i> )	tiesinės prognozės modelis
MFCC (angl. <i>Mel-frequency cepstral coefficients</i> )	Melų skalės kepstro koeficientai
MV	mikrovaldiklis
PCA (angl. <i>Principal Component Analysis</i> )	pagrindinių komponenčių analizė
PMM	paslėptieji Markovo modeliai
SK (angl. <i>Liquid crystal display</i> )	skystųjų kristalų indikatorius
SMD (angl. <i>Surface Mount Devices</i> )	paviršinio montažo elementai
STFT (angl. <i>Short-time Fourier transform</i> )	trumpalaikė Furjė transformacija
SVM (angl. <i>Support vector machines</i> )	atraminių vektorių mašinos
UART (angl. <i>Universal Asynchronous Receiver–Transmitter</i> )	universalus asinchroninis imtuvas–siųstuvas
USB (angl. <i>Universal Serial Bus</i> )	universalioji nuosekloji jungtis
VK	vektorių kvantavimas
ŽDF	žemų dažnių filtras



# 1. ĮVADAS

Vis didėjant žmonių gyvenimo tempui, tampa aktualios priemonės, padedančios taupyti laiką bei žmogiškus resursus, rankų darbą. Turbūt daugelis mūsų nebeįsivaizduojame savo gyvenimo be automatinės skalbimo mašinos, mašinų surinkimo linijos neišsiverčia be robotų pagalbos. Ne išimtis būtų ir kompozitorių bei muzikantų darbo palengvinimas, jei jie turėtų įtaisą, galintį natas užrašyti pagal muzikinio instrumento skleidžiamą garsą atliekant vieną ar kitą kūrinį.

## 1.1. Darbo aktualumas ir tikslas

Sparčiai besivystant skaitmeninėms garso bei vaizdo apdorojimo technologijoms, aktualus tampa ir automatinis muzikos garsų atpažinimas. Pastaruoju metu muzikos instrumentų (šiuo atveju – gitaros) garsų transkribavimas yra daug kur naudojamas. Įsivaizduokime situaciją: išgirstame kokį nors kūrinį, kurį norėtume išmokti groti patys, tačiau niekur nerandame natų. Tuo tarpu automatinė garsų atpažinimo sistema sutaupytų daug muzikanto laiko bei resursų. Be to, tokia sistema būtų labai aktuali besimokantiems groti, nes galima būtų sekti, kurioje vietoje padaryta klaida ir panašiai.

Šio darbo tikslas – ištirti gitaros garsų atpažinimo realiuoju laiku galimybę, sukuriant sistemą, atpažįstančią gitaros garsus.

## 1.2. Darbo uždaviniai

Darbo tikslui pasiekti keliami uždaviniai:

1. Atlikti garsų atpažinimo metodų analitinę apžvalgą.
2. Atrinkti ir vienoje sistemoje įgyvendinti bent vieną gitaros garsų atpažinimo algoritmą.
3. Sudaryti originalų gitaros garsyną.
4. Atlikti sukurtos gitaros garsų atpažinimo sistemos tyrimus.

## 1.3. Naudoti tyrimo ir analizės metodai

Darbe panaudoti kompiuterinio modeliavimo principai, sukauptos žinios dirbant su *MATLAB* programa. Pritaikyta skaitmeninių signalų apdorojimo (SSA) teorija. Gitaros garsų atpažinimo sistemoje įgyvendinti garsų požymių išskyrimo metodai – greitoji Furjė transformacija ir autokoreliacija. Kuriant gitaros garsų atpažinimo sistemos maketą panaudotos programavimo C kalba, elektroninių įtaisų projektavimo žinios. Tiriant pagamintą maketą naudotas laboratorinis maketavimas, taip pat empirinių tyrimų metodas – eksperimentas. Gautiems rezultatams analizuoti pritaikyti statistinių tyrimų metodai.

## 1.4. Darbo naujumas ir praktinė nauda

Pastaruoju metu yra sukurta nemažai programų (pastaruoju metu ypač daug pritaikytų išmaniesiems telefonams), kurių pagalba galima suderinti gitarą. Tačiau tokios programos turi vieną trūkumą – jomis galima atpažinti tik vienos stygos virpėjimo dažnį. Taip pat yra profesionalių muzikantams skirtų programų, kurios geba atpažinti visą akordą (polifoninį kelių stygų garsą). Tačiau tokios programos yra mokamos ir jų algoritmai kruopščiai slepiami.

Šiame darbe projektuojama sistema leis atpažinti kelias natas tuo pačiu metu. Pradedantys mokytis groti gitarą dažnai nemoka ar nesugeba tinkamai sugroti kai kurių akordų, todėl šios sistemos pagalba galima bus sekti, ar tinkamai sugrotas akordas, ir jei ne – kur konkrečiai padaryta klaida. Toks prietaisas galėtų būti „elektroninis mokytojas“ ir padėti greičiau išmokyti groti gitarą.

Kitas tokios sistemos panaudojimas galėtų būti skirtas automatiniam gitaros garsų transkribavimui. Tai palengvintų darbą kompozitoriams, nes jie galėtų tiesiog improvizuoti, o jų kūryba iš karto būtų užrašoma natų pavidalu, pavyzdžiui, MIDI formato byloje. Toks natų užrašymo būdas naudojant muzikinius sintezatorius įgalintų lengvai vieno instrumento garsus paversti kito instrumento garsais, pvz. iš gitaros į fleitos ar dūdmaišio garsus.

## 1.5. Darbo struktūra

Magistro darbą sudaro įvadas, šeši skyriai, darbo rezultatų apibendrinimas, literatūros sąrašas ir priedai.

Pirmąjį skyrių sudaro įvadas, kuriame aptariamas darbo aktualumas, tikslai ir uždaviniai.

Antrajame skyriuje atlikta garsų atpažinimo sistemų analizė. Aptarti algoritmai, paprastai naudojami gitaros ir panašių garsų atpažinimui.

Trečiajame skyriuje atliekamas originalaus gitaros garsyno sudarymas. Šis garsynas bus reikalingas tolimesniems tyrimams atlikti.

Ketvirtajame skyriuje atliekamas kompiuterinis algoritmų, skirtų gitaros garsams atpažinti, tyrimas programa *MATLAB*. Tyrimams naudoti skaitmeninių FIR filtrų, harmonikų spektro sandaugos ir autokoreliacijos algoritmai. Tiriamas jų tinkamumas gitaros garsams atpažinti.

Penktajame skyriuje aptariamas gitaros garsų atpažinimo sistemos projektavimas. Detaliau aprašomi svarbiausių sistemos mazgų pasirinkimo kriterijai, sudaromos struktūrinė ir principinė elektrinė schemas, taip pat programa mikrovaldikliui.

Šeštajame skyriuje atliekama gitaros garsų atpažinimo sistemos maketo patikra laboratorinėmis sąlygomis. Tiriamas gitaros įrašų atpažinimas.

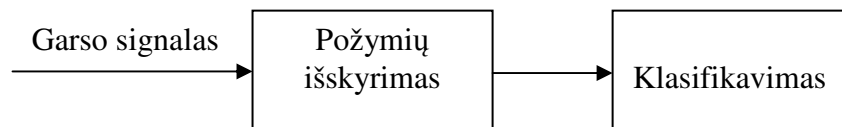
Prieduose pateikiamos suprojektuotos sistemos struktūrinė bei principinė schemas, *MATLAB* programos bei mikrovaldiklio išėties kodai.

## 2. GITAROS GARSŲ ATPAŽINIMO METODŲ ANALITINĖ APŽVALGA

Šiame skyriuje nagrinėjami muzikos instrumentų ir jų skleidžiamų garsų atpažinimo įtaisai, ir įvairūs kiti įtaisai, atpažįstantys garsus (pvz., paukščių identifikavimo pagal giesmę), ir juose taikomi metodai. Aptariami garsų požymių išskyrimo bei klasifikavimo algoritmai.

Garso atpažinimas (ar tai būtų kalbos, ar aplinkos, ar muzikos garsai) paprastai yra atliekamas dviem etapais: pirmasis – požymių išskyrimas, antrasis – klasifikacija (2.1 pav.). Požymių išskyrimas reikalingas norint gauti charakteristikų rinkinį, būdingą tam garsui. Pavyzdžiui, garsą galėtų sudaryti aukšto arba žemo tono garsas (kitai tariant – reikia gauti signalo spektrą).

Antrasis etapas – klasifikacija – reikalingas atpažinti garsams kataloguojant jų požymius tam tikru būdu (sistemos mokymo) ir tada lyginant tiriamą garsą su požymių duomenų baze.



2.1 pav. Tipinė garso atpažinimo proceso diagrama

Požymių išskyrimas gali būti suskirstytas į du tipus: stacionarus (dažnių bazė) požymių išskyrimas ir nestacionarus (laiko – dažnių bazė) požymių išskyrimas. Stacionarus signalo požymių išskyrimas formuoja bendrą rezultatą išsamiai atvaizduojant dažnius, esančius visame signale, t. y. dažnių spektrą. Šiuo atveju nėra jokio skirtumo, kurioje vietoje šie dažniai aptikti signale. Priešingai, nestacionariais signalo požymių išskyrimo atvejais signalas skyla į diskretaus laiko vienetus. Tai leidžia identifikuoti dažnį tam tikroje vietoje signale.

### 2.1. Garso signalo požymių išskyrimo algoritmai

Signalų požymių išskyrimui – kalbos, muzikos instrumentų atpažinimui ir t. t., yra keletas skirtingų algoritmų (kiekvienas iš kelių skirtingų variantų). Pradžioje verta paminėti pačius populiariausius ir dažniausiai garsams atpažinti taikomus algoritmus:

- greitoji Furjė transformacija (GFT);
- Melų skalės keistro koeficientai (angl. *Mel-frequency cepstral coefficients*);
- tiesinės prognozės modelis (angl. *linear predictive coding*);
- nulių kirtimų dažnis (angl. *zero crossing rate*);
- autokoreliacija [1, 2, 3, 5, 7].

Pagrindiniai nestacionaraus požymių išskyrimo algoritmai, kurie yra dažniausiai minimi literatūroje [1, 4], yra šie:

- trumpalaikė Furjė transformacija (angl. *short-time Fourier transform*);
- vilnelių transformacija (angl. *wavelet transform*).

Metodai, pagrįsti tiesinės prognozės modelio koeficientais, yra grindžiami vokoderio [6] idėja, kuri yra žmogaus balso trakto imitacija. Kadangi žmogaus balso trakto neatkartoja aplinkos garsai, šie metodai nėra tinkami atpažinti ne kalbos garsus.

Melų skalės kepstro koeficientai taip pat dažniausiai naudojami kalbos atpažinimo ar žmogaus identifikavimo pagal balsą uždaviniuose, įvairių garsų palyginimo ir kategorizavimo sistemose. Todėl muzikinių garsų tonui nustatyti pastarasis algoritmas nelabai tinkamas [2, 5].

Nulių kirtimų dažnis parodo, kiek kartų per sekundę signalas kerta laiko ašį. Šis algoritmas nenaudotinas, kai tuo pačiu laiko momentu tenka apdoroti kelis signalus (pvz., užgautos kelios gitaros stygos vienu metu), nes tuomet rezultatai bus iškreipti.

Toliau išsamiau aptarsime požymių išskyrimo algoritmus, kurie būtų tinkami įgyvendinant gitaros ar kitokių trumpalaikių periodinių garsų tono atpažinimo uždavinius.

### 2.1.1. Greitoji Furjė transformacija

Praktikoje signalų požymių išskyrimui dažniausiai yra naudojama greitoji Furjė transformacija (angl. *Fast Fourier transform*), pasiūlyta septintame dešimtmetyje J. W. Cooley ir J. W. Tukey [1, 8]. Vienas iš pagrindinių reikalavimų yra tas, kad duomenų ilgis turi būti  $2^M$ . Jeigu signalo ilgis atskaitomis yra kitoks, jis papildomas nuliais iki artimiausio  $2^M$  ilgio ir tuomet vykdomas GFT algoritmas. Nulius galima pridėti tiek nuo signalo pradžios, tiek nuo galo, ir tai nekeičia GFT rezultato. Pavyzdžiui, jeigu signalo ilgis atskaitomis yra 220, jis papildomas nuliais iki 256 ( $2^8$ ) atskaitų ir tuomet vykdoma GFT, kuri išreiškiama formule:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot \omega_N^{jk}, \quad \text{kai } k = 0, \dots, N-1, \quad (2.1)$$

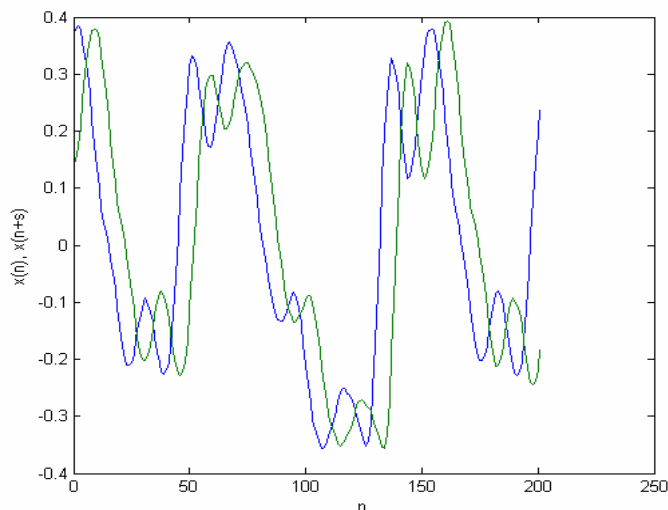
čia:  $\omega_N = e^{-j2\pi/N}$  ir yra dažnis, kurį norime patikrinti;  $n$  – imtis signale;  $N$  – testuojamo signalo ilgis.

Jei diskrečiosios Furjė transformacijos (DFT) sudėtingumas yra  $O(N^2)$ , tai GFT atveju sudėtingumas tampa  $O(N \log_2 N)$ . Tad turint mažai atskaitų, tikslinga naudoti DFT, nes jos skaičiavimai yra šiek tiek paprastesni, bet turint daugiau atskaitų (64, 128, 256 ar daugiau) jau tikslinga naudoti greitosios Furjė transformacijos algoritmus.

### 2.1.2. Autokoreliacija

Tai taip pat labai dažnai signalų apdorojime naudojamas metodas, skirtas nustatyti nagrinėjamo garso tonui, pagrįstas didžiausios autokoreliacijos funkcijos vertės radimu [1, 6, 7, 16]. Koreliacija į kompoziciją labai panaši operacija. Tai dviejų reiškinių (dviejų požymių)

tarpusavio sąryšis. Ji reikalinga išskirti norimą signalą ar jo periodines dedamąsias iš daug signalų ar didelių triukšmų. Iš esmės algoritmas paremtas tuo, kad periodinis signalas, net jei jis nėra gryna sinusoidė, vienu laiko momentu bus panašus į tą patį signalą kitu laiko momentu (2.2 pav.). Tai galioja net jeigu signalo amplitudė kinta laike, jei tie pokyčiai nebūna per greiti.



2.2 pav. Originalus ir paslinktas per  $k$  laike signalas [7]

Jei turime dvi baigtinės energijos sekas, jų tarpusavio koreliacija išreiškiama:

$$r_{xy}(k) = \sum_{n=-\infty}^{\infty} x(n)y(n-k) = \sum_{n=-\infty}^{\infty} x(n+k)y(n), \quad \text{kai } k = 0, \pm 1, \dots, \pm \infty. \quad (2.2)$$

Šioje formulėje  $k$  yra vėlinimo parametras, o indeksai  $x, y$  eilės tvarka rodo kryptį, kuria viena seka yra pastumta kitos atžvilgiu. Tuo atveju, jeigu  $y(n) = x(n)$ , turime sekos  $x(n)$  autokoreliaciją, t. y. signalo palyginimą su pačiu savimi [6]:

$$r_{xx}(k) = \sum_{n=-\infty}^{\infty} x(n)x(n-k) = \sum_{n=-\infty}^{\infty} x(n+k)x(n), \quad \text{kai } k = 0, \pm 1, \dots, \pm \infty. \quad (2.3)$$

Jei  $x(n)$  yra priežastinė  $N$  ilgio seka, tai:

$$r_{xx}(k) = \sum_{n=0}^{N-|k|-1} x(n)x(n-k). \quad (2.4)$$

Kaip matyti iš (2.4), vėlinimo parametras  $k$  kinta nuo 1 iki  $N-1$  ir dažnai vadinamas autokoreliacinės funkcijos eile.  $k$ -tosios eilės autokoreliacija parodo pradinio signalo panašumą su per  $k$  vienetų perstumtu signalu. Paprastai autokoreliacinės funkcijos eilė apytiksliai parenkama  $k = F_d + 4$ , kur  $F_d$  yra diskretizacijos dažnis, išreikštas kHz.

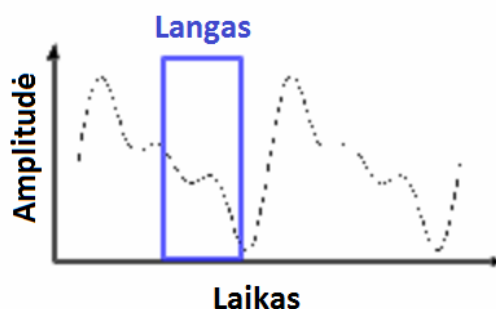
### 2.1.3. Trumpalaikė Furjė transformacija

Kartais DFT reikia skaičiuoti vis iš naujo atitinkamai paslinkus analizės langą per vieną analizuojamos sekos  $x(n)$  elementą. Tada skaičiavimo rezultatas yra ne tik dažnio, bet ir sekos elemento  $n$  funkcija. Tokiam skaičiavimui reikės didelių skaičiavimo resursų, nes paslinkus

langą per vieną sekos elementą, spektrą teks perskaičiuoti iš naujo. Todėl aktualus yra rekursyvusis DFT skaičiavimo algoritmas [1, 5]. Ribotos trukmės sekos (apribotos fiksuotos padėties stačiakampiu langu (žr. 2.3 pav.)) Furjė transformacija dar vadinama trumpalaikė Furjė transformacija (angl. *short-time Fourier transform*). Jei langas slenkamas, tai visas duomenų masyvas skaldomas į iš dalies persidengiančius segmentus. Kiekvienas jų atskirai transformuojamas ir gauti rezultatai surašomi į matricą. Jos įrašai rodo signalo amplitudę ir fazę kiekvienu laiko momentu kiekvienoje dažnių atkarpoje. Šį procesą galima būtų aprašyti taip:

$$X(m, \omega) = \sum_{n=-\infty}^{\infty} x[n] \cdot w[n - m] \cdot e^{-j\omega n}, \quad (2.5)$$

čia:  $x[n]$  – signalas;  $w[n]$  – „langas“ (periodas, kuriam taikoma transformacija).



2.3 pav. Slenkantis langas, atliekant STFT

Galima įžvelgti keletą išraiškos (2.5) privalumų lyginant su GFT algoritmu:

- $N$  gali būti bet koks. Nebūtina, kad jis būtų dviejų kartotinis, kaip GFT atveju;
- dažniai  $\omega$  irgi gali būti bet kokie. Spektrą galima skaičiuoti visoje dažnių srityje arba tik jos dalyje;
- norint gauti  $N$  spektro imčių reikės tik  $2N$  daugybos operacijų.

Algoritmo trūkumas – dėl rekursyviojo skaičiavimo kaupiasi paklaidos.

#### 2.1.4. Vilnelių transformacija

Šiuo metu įprasta pagrindine signalų analizės priemone laikyti Furjė transformaciją. Tai iš esmės yra signalo atvaizdavimas harmoninių virpesių suma. Toks atvaizdavimas atskleidžia visą signalo dažnių turinį, t. y. Furjė spektras parodo, kokio dažnio komponentės sudaro tiriamą signalą, tačiau šios harmoninės funkcijos nėra lokalizuotos laiko srityje, todėl negalima sustatyti tikslių signalo trūkio vietų ir panašiai [3, 4].

Vilnelių transformacija taip pat yra atvaizduojama tam tikrų funkcijų suma, tik, priešingai nei Furjė transformacijoje, vietoje harmonikų  $\{e^{j\omega n}\}$  yra naudojama funkcijų sistema  $\Psi_{a,b}(t) = 1/\sqrt{a} \cdot \Psi((b-t)/a)$ . Tolydinė vilnelių transformacija užrašoma tokiu pavidalu:

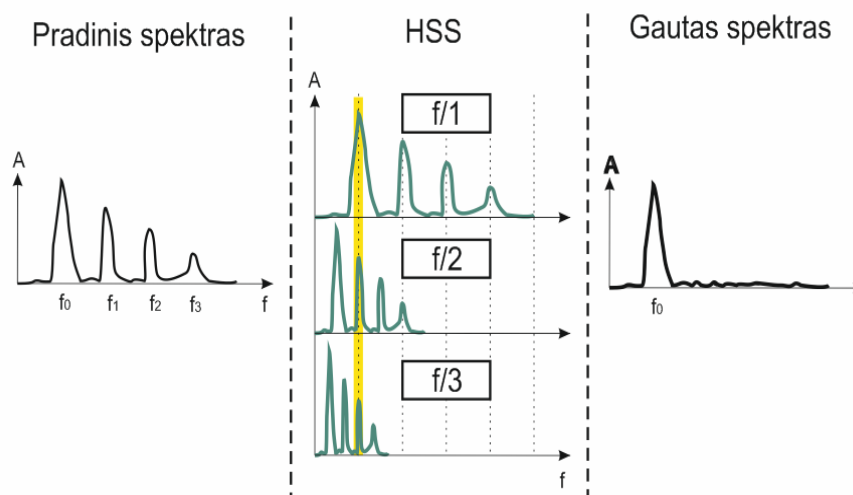
$$W_f(b, a) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \Psi\left(\frac{b-t}{a}\right) dt, \quad (2.6)$$

čia:  $a$  – mastelio parametras;  $b$  – postūmio parametras.

Naudojant skirtingas vilneles, gaunamas skirtingas vilnelių spektras. Jų pasirinkimą lemia konkreti užduotis. Dažniausiai naudojamos Morleto (*Morlet*), Daubechies, meksikietiškos kepurės ir kitos vilnelės [4].

### 2.1.5. Harmonikų spektrų sandaugos algoritmas

Harmonikų spektrų sandaugos algoritmas taikomas signalo spektrui [7]. Jei įėjimo signalas yra muzikinė nata, tada jos spektras turėtų susidėti iš pikinių reikšmių serijos, atitinkančios pagrindinį dažnį ir jam kartotinių harmoninių komponentių. Jeigu „suspaustume“ spektro komponentes keletą kartų ir vis palygintume jas su pradiniu spektru, pastebėtume, kad labiausiai išreikšta harmonika sutampa su kartotinėmis „suspausto“ spektro harmonikomis (2.4 pav.). Taigi sudauginus spektrų dedamąsias gautume pagrindinį dažnį.



2.4 pav. Pagrindinio dažnio dedamosios  $f_0$  radimas taikant harmonikų spektro sumos metodą

Taikant šį algoritmą, pirmą kartą reikia originalų spektrą „suspausti“ du kartus, antrą kartą – tris ir t. t. Šias operacijas išreiškia formulė:

$$HSS(k) = \prod_{r=1}^R X(k \cdot r), \quad \text{kai } 0 \leq k < N, \quad (2.7)$$

čia:  $X(k)$  yra  $N$  atskaitų GFT rezultatas;  $k$  – atskaitos numeris;  $R$  – pasirinktas spektro sutankinimo skaičius. Paprastai užtenka šią operaciją pakartoti keletą kartų, kad būtų nustatytas pagrindinis dažnis (-iai), kuris (-ie) atitinka didžiausią amplitudę.

## 2.1.6. IIR ir FIR filtrų taikymas

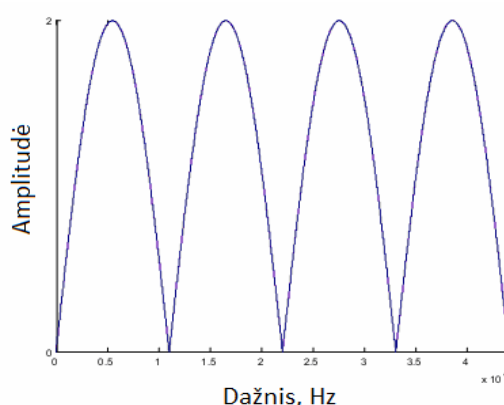
Monografijoje „Multi Pitch Estimation by using IIR Comb Filters“ nagrinėjamas keleto tonų išskyrimas iš garso naudojant modifikuotus neribotos (IIR) ir ribotos (FIR) impulsinės reakcijos filtrus [8]. Šie filtrai dažnai sutinkami automatinėse muzikos transkribavimo sistemose, kalbančiojo atpažinimo įtaisuose ir kituose su signalų apdorojimu susijusiose sistemose.

FIR filtro perdavimo funkcija aprašoma taip:

$$H(z) = 1 + g \cdot z^{-D}, \quad (2.8)$$

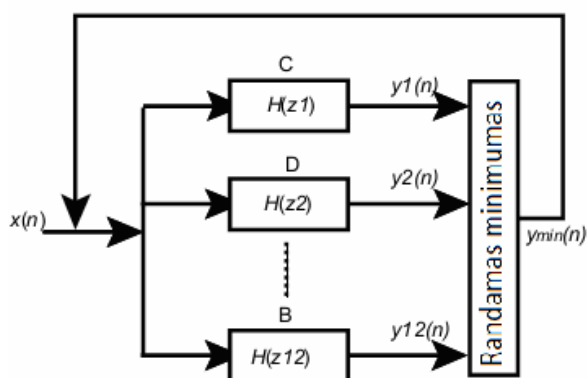
čia  $g$  yra faktorius, kuris dalina filtro slopinimą tarp  $1 + g$  ir  $1 - g$ ,  $D$  – užlaikymas atskaitomis.

Kai  $g = -1$ , diskretizavimo dažnis 44 100 Hz ir  $D = 4$ , FIR filtro DACH pateikta 2.5 pav.



2.5 pav. FIR filtro DACH charakteristika

Autorius naudojo 12 FIR filtrų sujungtų lygiagrečiai (2.6 pav.) kiekvienai oktavos naitai atpažinti. Į visus filtrus buvo siunčiamas garso signalas, o pagal tai, kurio filtro išėjime buvo mažiausia amplitudė, galima spręsti, koks ieškamos natos dažnis. Šis procesas kartojamas, kol gaunamos visos ieškamos natos (2.6 pav.).



2.6 pav. Lygiagrečiai sujungtų FIR filtrų sistema

Naudojant šiuos skaitmeninius filtrus, pavienė nata buvo teisingai atpažįstama visais atvejais, dvi natos – 90,9 %, trys natos – 83,3 %, o keturios natos – 75 % atvejų.



Apibendrinant gautus rezultatus galima teigti, kad tai labai paprastas sprendimas, kurį galima įgyvendinti gitaros natų atpažinimo sistemoje ir jo rezultatai gaunami pakankamai tikslūs. Vienintelis trūkumas, kad šie filtrai yra sąlyginai lėti. Tačiau apdorojant nedidelį kiekį duomenų, juos galima taikyti ir realiu laiku veikiančiose sistemose.

## 2.2. Garsų klasifikavimas

Atlikus garsų požymių išskyrimo operaciją, sekantis žingsnis būtų taikyti klasifikacijos metodus, kurie naudojami kataloguoti signalų požymius ir surašyti juos į duomenų bazę. Tuomet kiekvieno tiriamo garso požymiai gali būti sulyginami su šioje duomenų bazėje esančių „etaloninių“ garsų požymiais.

Sekantys klasifikavimo metodai paprastai yra naudojami kalbančiojo atpažinimui pagal jo balsą:

- dinaminis laiko skalės kraipymas (DLSK);
- paslėptieji Markovo modeliai (PMM);
- vektorių kvantavimas (VK);
- dirbtinių neuronų tinklai (DNT);
- ilgalaikė statistika (IS) [1, 10].

Be šių, taip pat plačiai aprašyti metodai, dažnai naudojami muzikos instrumentų identifikavimui (ne tik pavieniams tonams):

- maksimalaus tikėtimumo vertinimas (angl. *maximum likelihood estimation*);
- Gauso mišinių modeliai (GMM);
- atraminių vektorių mašinos (angl. *support vector machines*).

Buvo sudarytos aptartų metodų palyginimo lentelės [10]. Šios lentelės parodė, kad kai kurie iš šių metodų pagal savo pobūdį negali būti naudojami ne kalbos garsų atpažinimui. Taip yra todėl, kad aplinkos garsai stokoja fonetinės struktūros, ko gausu kalboje. Nėra nustatytas „alfabetas“, pagal kurį tam tikri ne kalbos garsų segmentai galėtų būti atitinkamai padalinti, dėl to paslėptųjų Markovo modelių (PMM) taikymas aplinkos ar muzikos garsams negali būti naudotinas.

Ilgalaikės statistikos (IS) klasifikavimo metodas negali būti taikomas kartu su nestacionariais signalo požymių išskyrimo algoritmais, todėl pastarasis metodas taip pat yra atmestinas.

Apžvelgus literatūrą [1, 9, 10], kurioje aptariamas muzikos, trumpų neperiodinių ir periodinių garsų, bei paukščių skleidžiamų garsų atpažinimas, buvo sudaryta lyginamoji klasifikavimo metodų sąveikos su skirtingais požymių išskyrimo metodais lentelė (žr. 2.1 lentelę).

**2.1 lentelė.** Požymių išskyrimo ir klasifikavimo metodų tarpusavio sąveikos įtaka atpažinimo tikslumui

Požymių išskyrimo metodai \ Klasifikavimo metodai	GFT	STFT	MFCC	SVT
<b>DLSK</b>	iki 60 %	iki 58 %	iki 70 %	iki 70 %
<b>GMM</b>	iki 22 %	iki 45 %	iki 45 %	iki 21 %
<b>VK</b>	iki 52 %	iki 2 %	iki 48 %	iki 55 %
<b>DNT</b>	iki 3 %	iki 3 %	iki 7 %	iki 41 %

Kaip matyti iš 2.1 lentelės, didžiausias ne kalbos signalų atpažinimo tikslumas gaunamas naudojant dinaminį laiko skalės kraipymo metodą (DLSK), net nesvarbu su kuriuo požymių išskyrimo metodu jis taikomas. Didžiausias šio metodo trūkumas – sąlyginai ilgas skaičiavimo procesas, todėl garsams atpažinti realiame laike, o juo labiau naudojant ne specializuotą mikrovaldiklį, jo pritaikyti nepavyks.

Kita vertus, gitaros garsai yra sąlyginai trumpi ir periodiniai. Dėl šių priežasčių nėra būtinybės taikyti garsų klasifikavimo metodų, nes jie tik smarkiai pailgins gitaros garsų atpažinimo procesą. Be to, tada nereikės sistemos „mokyti“ atpažinti natas. Tad įgyvendinant gitaros garsų atpažinimo sistemą, bus apsiribota tik požymių išskyrimo metodais, kurie užtikrina dažnio ir amplitudės radimą iš turimo garso signalo.

## 2.3. Įtaisai, naudojantys garsų atpažinimo algoritmus

Šiame poskyryje pateiksime kelis dažnai naudojamų įtaisų bei kompiuterinių programų pavyzdžius, kurie naudoja vienokius ar kitokius garsų atpažinimo algoritmus. Tai būtų muzikos instrumentų derinimo priemonės, paukščių atpažinimui skirti įtaisai ir programiniai paketai.

### 2.3.1. Muzikos instrumentų derinimo priemonės

Turbūt kiekvienas turintis gitarą ir mokantis ja groti, naudoja vienokias ar kitokias priemones jai suderinti. Šiuo metu galima nusipirkti pačių įvairiausių derinimo prietaisų. Vienas jų – *Musedo Metro-tuner MT-30* (žr. 2.7 pav.) [11]. Tai yra trys prietaisai viename: gitaros derintuvas, liaudiškai vadinamas „tuneriu“, metronomas ir tonų generatorius.



**2.7 pav.** Gitaros derintuvas *Metro-Tuner MT-30*

Šio prietaisėlio derinimo ribos yra nuo A0 (27,5Hz) natos iki A6 (1 760Hz) natos. Jis turi integruotą mikrofoną, o taip pat signalo linijinio jėjimo jungtį, kuri leidžia pasiekti didesnį tikslumą. Duomenys apie parinktą režimą, natos dažnį ir stygos derinimo pobūdį (labiau įtemti ar atleisti) yra rodomi skystųjų kristalų ekrane. Prietaiso pagrindiniai privalumai: yra integruotas metronomas ir tonų generatorius, kurie labai praverčia muzikantams, taip pat mažas dydis, ekonomiškumas (dviejų paprastų AAA dydžio 1,5 volto baterijų užtenka iki pusės metų). Tačiau šio prietaiso negalima būtų vadinti natų atpažinimo įtaisu, nes geriausiu atveju jis tesugeba parodyti pavienės užgautos stygos dažnį ir jį atitinkantį natos žymenį. Tuo tarpu grojant net ir paprasčiausius kūrinius dažnai tenka užgauti kelias, ar net visas gitaros stygas vienu metu.

Taip pat nemažai analogiškai veikiančių derinimo programų galima sutikti kompiuterinėje terpėje. Vienos dažniausiai naudojamų – *PitchPerfect*, *AP Tuner*, *Note detector* ir panašios programos [12]. Jos taip pat leidžia matyti analizuojamo garso dažnį, natos žymenį ir nuokrypį nuo etaloninio dažnio. Derinimui gali būti naudojamas vidinis (nešiojamuose kompiuteriuose) arba išorinis mikrofonas.

Verta paminėti, kad programa *Note detector v.1.0.2.8*. išsiskiria iš kitų, nes yra adaptuota daugeliui muzikos instrumentų. Natos simbolis joje vaizduojamas ir penklinėje. Tai kur kas universalesnis vaizdavimo būdas nei, pavyzdžiui, tik raidinis natų žymėjimas (2.8 pav.).



2.8 pav. *Note detector v.1.0.2.8* programos langas

Šios programos veikimas paremtas greitosios Furjė transformacijos (GFT) algoritmu. Deja, tačiau ir ši programa rodo tik paskutinę atpažintą natą, todėl negalima atpažinti kelių natų vienu laiko momentu.

Pastaraisiais metais labai padidėjęs kompiuterių spartai, analogiškoms programoms galima naudoti net ir sudėtingiausius algoritmus, šie su signalų apdorojimu susitvarkys labai greitai. Pagrindinis trūkumas, kad reikia su savimi turėti kompiuterį, o jis dar nėra toks mažas, pigus ir

lengvas, kad kiekvienas patogiai galėtų nešiotis ir prireikus turėtų jį šalia. Kita vertus, šiuolaikiniai išmanieji telefonai jau beveik prilygsta kompiuteriams.

### 2.3.2. Paukščių skleidžiamų garsų atpažinimo įtaisai

Paukščių atpažinimo pagal giesmę sistemas aptarti taip pat naudinga, nes jose gali būti naudojami panašūs ar net tokie pat metodai, kaip ir muzikos instrumentu natų atpažinime. Todėl šias sistemas, o taip pat jose naudojamus algoritmus trumpai ir aptarsime.

Patyrę ornitologai paukščius lengvai atpažįsta iš klausos. Tačiau yra daugybė gamtos entuziastų, kuriems nėra taip lengva pasakyti, kokio paukščio giesmę jie girdi. Tam yra sukurti įvairūs prietaisai, padedantys identifikuoti paukščius, pagal jų skleidžiamą garsą. Vienas tokių yra *IdentiFlyer Ultra* [13] (2.9 pav.). Jis lengvai telpa į delną, gali atpažinti iki 120 skirtingų paukščių ir 10 varlių rūšių. Papildomai yra įmontuotas skystųjų kristalų (toliau – SK) ekranėlis, kuriame rodomas atpažintas objektas. Kai kuriose įtaiso versijose galima įsigyti papildomas atminties korteles su vandens ar naktinių paukščių bibliotekomis. Tokiuose įtaisuose atpažinimui dažniausiai yra naudojami greitosios Furjė transformacijos, koreliacijos ar Melų skalės kepstro koeficientų algoritmai [14].



2.9 pav. *IdentiFlyer Ultra* paukščių atpažinimo pagal jų giesmę įtaisas [13]

Žinoma, tai nėra prietaisas, skirtas profesionalams. Tačiau jis pakankamai funkcionalus ir tikslus mėgėjams, kurie domisi ornitologija ir paukščiais. Taip pat didelis privalumas, kad jis yra mažas, todėl lengva nešiotis ir visada turėti po ranka. Trūkumas – nepakankamai tikslus, palyginti maža duomenų bazė. Taip pat paukščių giesmė turi būti pakankamai garsi, priešingu atveju prietaisas jos paprasčiausiai neužfiksuos.

Taip pat šiuo metu sparčiai daugėja įvairių programų išmaniesiems telefonams, tad nenuostabu, kad jau yra sukurta nemažai aplikacijų, skirtų būtent paukščių identifikavimui pagal jų giesmę. Viena naujesnių panašaus pobūdžio programų yra *WeBIRD*, kurios nemokama versija pasirodė tik šių, t. y. 2012 m. pavasarį [15]. Ji veikia lygiai taip pat kaip ir ankščiau aptartas prietaisėlis, tačiau pagrindinis privalumas yra tai, kad telefoną dažniausiai turime su savimi, tad

prireikus bet kada galima pasinaudoti šia programa. Pagrindinis trūkumas – nepakankamas mikrofono jautrumas. Dėl šios priežasties nebus galima identifikuoti garsų, sklindančių iš pakankamai toli.

## 2.4. Skyriaus išvados

Šiame skyriuje buvo aptarti literatūroje sutinkami metodai, naudotini garsų atpažinimo sistemose. Aptarti keturi dažniausiai naudojami algoritmai garsų požymiams išskirti, t. y. pagrindiniam dažniui signale rasti: greitoji ir trumpalaikė Furjė transformacijos, autokoreliacija bei vilnelių transformacija. Visi šie algoritmai gali veikti realiuoju laiku.

Taip pat aptartas harmonikų spektrų sandaugos algoritmas, kuris veikia dažnių erdvėje ir leidžia išskirti pagrindinių dažnių dedamąsias iš signalo. Jis dažnai naudojamas muzikinių natų atpažinimui, polifoninių garsų tonams identifikuoti ir panašioms uždaviniais spręsti.

Gitaros natų atpažinimui taip pat gali būti taikomi ir skaitmeniniai filtrai [8]. Jie gali veikti tiek laiko, tiek ir dažnių erdvėje. Jų pagrindinis privalumas – paprasti skaičiavimai, o jei filtro eilė nedidelė, jie atliekami labai greitai, praktiškai realiuoju laiku.

Garsų atpažinimo procese atsisakyta garsų klasifikacijos algoritmų. Visi jie reikalauja iš pradžių „apmokyti“ sistemą, kad ši sugebėtų tinkamai klasifikuoti garsus pagal būdingus jiems požymius. Taip pat reikia atlikti sudėtingus skaičiavimus, tad įgyvendinus juos sistemoje su mikrovaldikliu, labai pailgėtų natų atpažinimo laikas. Be to, gitaros skleidžiami garsai yra trumpi ir periodiniai, o jų dažniai suderinti pagal griežtus reikalavimus šiam instrumentui. Klasifikacijos algoritmų tikslumas tokiems garsams literatūros duomenimis [9, 10] yra palyginti mažas, todėl natoms atpažinti pilnai pakanka jau aptartų požymių išskyrimo algoritmų.

Kadangi atliekant darbą reikės sukurti gitaros garsų atpažinimo sistemą, todėl buvo aptartos ir analogiškos sistemos, skirtos muzikinių bei paukščių garsų atpažinimui. Aptarti jose dažniausiai naudojami algoritmai.

Taigi, apžvelgus literatūrą ir analogiškas garsų atpažinimo sistemas, tyrimams atlikti pasirinkti šie požymių išskyrimo algoritmai: greitoji Furjė transformacija (GFT) ir autokoreliacija. Pasirinkimą lėmė jų paprastumas ir greitaveika. Turint signalo spektrą, galima taikyti harmonikų spektrų sandaugos (HSS) algoritmą pagrindiniams dažniams išskirti. Tad atliekant tolimesnius tyrimus *MATLAB* programa, bus naudojami visi šie algoritmai, o taip pat ir skaitmeniniai filtrai.

### 3. GITAROS GARSYNO SUDARYMAS

Atliekant įvairius gitaros garsų atpažinimui naudojamų algoritmų tyrimus, pravartu turėti sudarytą gitaros garsyną, t. y. kiekvienos natos garsą, įrašytą skaitmeniniu formatu. To reikia, kad paskui galima būtų lengvai pritaikyti anksčiau aptartus skaitmeninius metodus, atpažinti gitaros garsams. Tai labai paspartins tyrimų eigą, sukonstruoto maketo bandymus, nes nereikės naudoti gitaros. Tačiau galutiniai bandymai vis tikrai bus atlikti naudojant ne įrašus, o tikrą gitarą.

#### 3.1. Gitaros garsų įrašymo sąlygos

Garso įrašų kokybė labai priklauso nuo išorinės aplinkos bei naudojamų įrenginių, todėl būtina aptarti, kokioms sąlygoms esant gauti šie įrašai. Taigi, įrašams atlikti buvo naudojama vokiečių firmos *HOHNER* gaminamų muzikinių instrumentų gamos elektroakustinė 6 stygų gitara *DMC-725S* [16]. Elektroakustinė gitara nuo akustinės skiriasi tik tuo, kad turi muzikantų tarpe taip vadinamą „nuėmėją“ (prietaisą, paverčiantį stygų vibracijas elektriniais impulsais), leidžiantį prie jos prijungti gitarinį kubą ar, kaip šiuo atveju, diktofoną. Dažniausiai naudojami magnetiniai arba pjezoelektriniai stygų virpesių jutikliai. Aptartoje gitaroje jis yra pjezoelektrinis. Naudojant tokį jutiklį yra gerai atkuriami aukštesni dažniai, jis nejautrus 50 Hz elektros tinklo kintamosios srovės fonui (magnetiniams davikliams tai rimta problema). Kadangi šis jutiklis integruotas į gitaros vidų, tai leidžia prijungti diktofoną linijiniu laidu, todėl pavyksta išvengti aplinkos triukšmų ir gauti kokybiškesnius įrašus. Nepaisant to, gitaros garsai buvo įrašinėjami uždaroje patalpoje, kurioje buvo laikomasi tylos ir taip išvengiant galimų trukdžių bei pašalinių garsų įtakos.

Gitara pirmiausiai buvo suderinta klausos pagalba, o po to dar patikrinta jau aptarta programa *AP Tuner*. Perderinti nieko nereikėjo, tai liudija fenomenalią gitaristo (ačiū Liudui Kripaičiui už pagalbą įrašinėjant gitaros garsus) klausą.

Skaitmeninių įrašų kokybė didžiaja dalimi priklauso nuo naudojamo diktofono. Įrašams atlikti buvo naudotas skaitmeninis diktofonas *Fostex FR-2LE* [17]. Pastarasis buvo pasiskolintas iš VGTU universiteto. Jis leidžia įrašyti garsus net 24 bitų skyra ir 96 kHz diskretizavimo dažniu. Tai daugiau nei pakanka geros kokybės skaitmeniniams garso įrašams. Tad atliekant įrašus buvo nustatyta 16 bitų kvantavimo skyra ir 44,1 kHz diskretizavimo dažnis. Diktofonas turi galimybę išsaugoti įrašus dviem formatais – MP3 bei WAV (angl. *Waveform Audio File*). Kadangi MP3 yra suglaudintas formatas, todėl įrašai užima mažai vietos, tačiau glaudinimo metu dalis žmogaus ausiai negirdimos informacijos neišvengiamai pradingsta. Kita vertus, garso bylos su *.wav* plėtiniu savo „kepurėje“ saugo informaciją apie diskretizavimo dažnį, kvantavimo

skyrą ir kitus parametrus. Be to, tai vienas paprasčiausių audio formatų, kuriam apdoroti *MATLAB* programa turi integruotas funkcijas. Dėl išvardintų priežasčių, įrašytos bylos išsaugomos būtent šiuo formatu.

Nagrinėta elektroakustine gitara galima išgauti 45 skirtingo dažnio garsus, t. y. pustonius. Pustonis – mažiausias dviejų garsų aukščio santykis [18]. Du pustoniai atitinką toną. Pustonių galime įsivaizduoti kaip juodą klavišą, esantį tarp dviejų baltų pianino klavišų. Beje, nederėtų maišyti tono ir tembro apibrėžimų. Tono fizikinė prasmė – garso virpesių dažnis. Tuo tarpu garso tembras – garso atspalvis, priklausomas nuo obertonų (harmonikų). Šiame darbe yra svarbu atpažinti gitaros natas, t. y. identifikuoti jų dažnį. Garso tembro analizavimas padėtų išskirti gitarą iš kitų instrumentų, tačiau užduotyje jau paminėta, kad tai bus būtent gitara. Kad būtų mažiau painiavos, du gretimų dažnių garsus laikysime tonais, pagal jo fizikinę prasmę.

Garsą kuria tik žemiau metalinės juostelės (ant gitaros grifo) esanti stygos dalis. Skirtingos stygos gali būti prispaustos skirtingose vietose, todėl tą pačią natą galima sugroti užgavus skirtingas stygas. Natos gali būti atvaizduotos keliais būdais. Kad būtų paprasčiau, panaudosime įprastą raidinį gitaros natų žymėjimą: E, F, F#, G, G#, A, A#, B, C# (čia grotelės reiškia diežą, t. y. natos pokytį pustoniu). Oktavos skaičius rašomas po raide, tad šiuo atveju pirmoji gitaros nata bus antros oktavos E2 (82,407 Hz), o paskutinioji – šeštosios oktavos C6 (1 046,5 Hz).

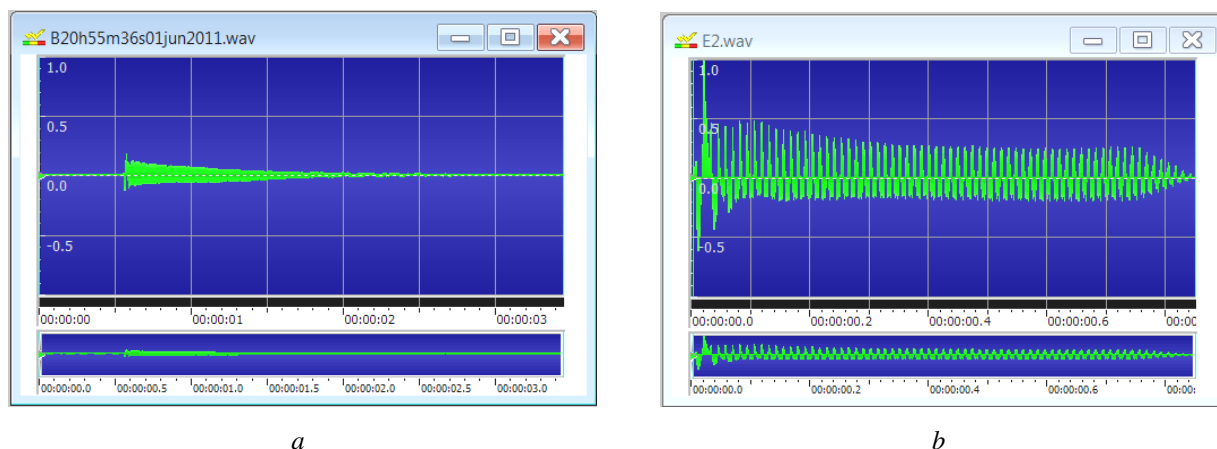
Visi galimi 45 skirtingų tonų gitaros garsai buvo įrašyti pavieniui, kurių kiekvienas atitinkamai po 10 kartų. Tokiu būdu iš viso gauta 450 įrašų. Norint apsunkinti jų atpažinimą, taip pat buvo padaryti kelių gretimų natų įrašai, kurių spektras vėliau buvo palygintas su sintetiniu (programos *MATLAB* pagalba sujungtų pavienių garsų) tų pačių natų garso spektru. Gauti rezultatai buvo beveik identiški. Taigi, garsai bus sintetinami, t. y. iš kelių įrašytu skirtingo tono garsų gaunamas vienas kelių tonų garsas juos susumavus. Taip nuspręsta daryti, nes įrašinėti visus galimus gitaros akordus (natų kombinacijas) po kelis kartus užimtų labai daug laiko.

### **3.2. Įrašytų gitaros garsų redagavimas bei katalogavimas**

Tam, kad galima būtų panaudoti gautus įrašus tyrimams, reikia juos redaguoti, pašalinti tylos intervalus ir panašiai. Taigi, įrašai buvo apdoroti laisvai prieinama programos *GoldWave* v5.58 bandomąja versija [19]. Kadangi jie buvo periodiškai kartojami kas 2–4 sekundes, tai įrašuose atsirasdavo tylos intervalų. Tokie segmentai buvo ištrinami ir paliekami 0,8 s ilgio trukmės garsai. Būtent toks laiko tarpas pasirinktas atsižvelgiant į aukštesniųjų natų amplitudės gesimo laiko intervalą.

Buvo pastebėta, kad įrašinėjant garsus į diktofoną, reikia sumažinti jo linijinio įėjimo jautrumą, nes įsijungdavo perkrovos indikatorius lemputė ir gaudavosi labai nekokybiški įrašai.

Tačiau sumažinus jautrumą, garsų amplitudė taip pat sumažėjo (3.1 pav. *a*). Todėl visų įrašų amplitudės buvo normalizuotos (suvienodinti garsumo lygiai) (3.1 pav. *b*).



**3.1 pav.** E2 natos garso įrašas prieš apdorojimą (*a*) ir po apdorojimo (*b*) programa *GoldWave*

Kad perėjimas tarp dviejų įrašytų garsų būtų švelnus ir neiššauktu trikdžių, pradedant ties 0,7 s laiko žyma, jų amplitudė tiesiškai sumažinama iki 0 dB (3.1 pav. *b*). Kaip jau buvo minėta, garso byla išsaugoma tokiu pat pavadinimu kaip ir nata (šiuo atveju E2).

Sekantis etapas yra įrašytų gitaros garsų katalogavimas. Kadangi visi 45 gitaros natų įrašai buvo kartojami po 10 kartų, tai pirmojo etapo 45 skirtingų natų įrašai yra patalpinti į direktoriją pavadinimu „1“, antrojo – „2“ ir t. t. Tokiu būdu yra gauta visa originali gitaros garsyno duomenų bazė, kuri užima 60,5 MB. Ši bazė patalpinta į laisvai prieinamą serverį [20], tad kiekvienas norintis paeksperimentuoti ar tiesiog atsisiųsti gitaros garsų, gali tai nesunkiai padaryti.

### 3.3. Skyriaus išvados

Šiame skyriuje buvo aptartas originalaus gitaros garsyno sudarymas. Jis yra būtinas, norint greičiau bei patikimiau atlikti gitaros garsų atpažinimo sistemos tyrimus, ypač naudojant kompiuterines modeliavimo ir tyrimo priemones. Kadangi šią sistemą reikia iširti visapusiškai, t. y. išbandyti visas galimas akordų (kelių natų) kombinacijas, tai reikalingų įrašų skaičius būtų labai didelis. Todėl buvo įrašyti monotoniniai garsai, o iš jų gauti polifoniniai pastaruosius sintetinant, t. y. susumuojant norimus garsus. Iš viso buvo įrašyta 450 įrašų (po 10 kiekvienos natos). Vėliau jie apdoroti programa *GoldWave v5.58* (pašalinti tylos intervalai, normalizuota amplitudė, suvienodinta įrašų trukmė). Sudarytas garsynas patalpintas viešai prieinamame serveryje [20].



## 4. GITAROS GARSŲ ATPAŽINIMO ALGORITMŲ TYRIMAS *MATLAB* PROGRAMA

Šiame skyriuje apžvelgsime įrašytų gitaros garsų atpažinimo algoritmų atliktus tyrimus ir jų rezultatus, gautus naudojant programinį paketą *MATLAB 7.10* [21]. Iš pirmo žvilgsnio viskas atrodo labai paprasta – tereikia garsinį signalą iš laiko erdvės pervesti į dažnių erdvę, t. y. gauti signalo spektrą, o po to rasti didžiausios amplitudės dedamąją. Tuo tikslu buvo atlikta greitoji Furjė transformacija visiems 45 įrašytiems gitaros garsams. Atliekant gitaros garsų atpažinimo algoritmų tyrimus *MATLAB* programa, daug naudingos informacijos buvo rasta D. Navakausko bei A. Serackio vadovėlyje „Skaitmeninis signalų apdorojimas taikant *MATLAB*“ [22].

### 4.1. Pirminis garso signalų apdorojimas

Gitaros garsams atpažinti taikysime požymių išskyrimo algoritmus. Vienas populiariausių yra greitoji Furjė transformacija [1, 5, 8]. GFT algoritmų atmainų yra labai daug. Vienos jų optimizuotos greičiui, kitos analizuojamo spektro tikslumui ir panašiai. Gitaros garsinių signalų požymių išskyrimui naudosime išretinto laiko GFT transformacijos algoritmą. *MATLAB* turi jau aprašytą funkciją jam įgyvendinti – *fft()*. Pagal Naikvisto kriterijų, diskretizavimo dažnis turi būti bent dvigubai didesnis nei maksimalus tiriamo garso dažnis. Šiuo atveju aukščiausia gitaros nata yra C6 (1 046,5 Hz). Todėl diskretizuoti signalą pakaktų 2 094 Hz dažniu. Tiriama įrašai diskretizuoti 44,1 kHz dažniu, tačiau, kadangi siekiama gauti maksimalų tikslumą, čia nieko nekeisime. Taikant GFT realiame įtase, šį dažnį tikslinga mažinti, norint gauti optimalų atskaitų skaičių ir tokiu būdu sutrumpinti skaičiavimus.

Prasmę turi tik teigiami dažniai ir spektras iki  $F_d/2$ . Taip yra todėl, kad signalo spektras simetriškai atsikartoja. GFT rezultatas – kompleksinis spektras. Amplitudinis spektras gaunamas:

$$a_k = \sqrt{(\text{Re}(f_k))^2 + (\text{Im}(f_k))^2}, \quad (4.1)$$

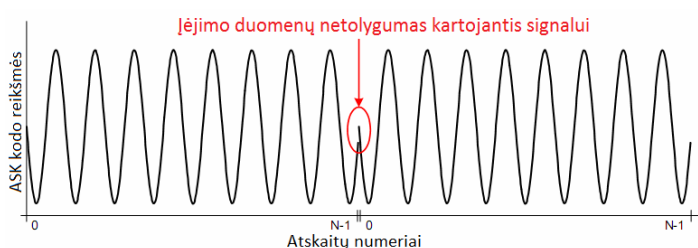
čia:  $a_k$  – amplitudinė atskaitos  $k$  reikšmė;  $Re$  – realioji dalis;  $Im$  – spektro menamoji dalis. *MATLAB* programoje tam galima panaudoti funkciją *abs()*. Po greitosios Furjė transformacijos kiekvienas vektoriaus elementas, kurio indeksas  $k$  atitiks signalo dažnį:

$$f_x = \frac{F_d}{N} \cdot k, \quad \text{kai } k = 0, 1, \dots, N - 1, \quad (4.2)$$

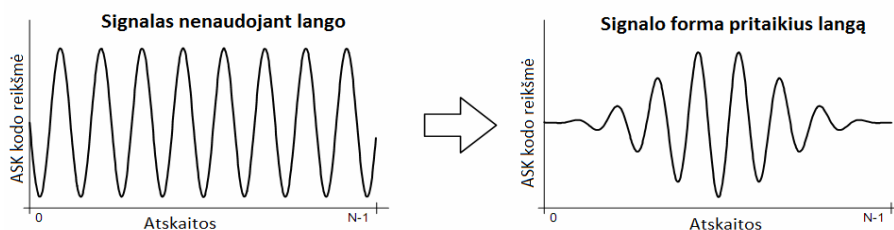
čia:  $f_x$  – ieškomas dažnis;  $F_d$  – diskretizacijos dažnis;  $N$  – atskaitų skaičius;  $k$  – atskaitos numeris.

Pagal (4.1) ir (4.2) formules, po GFT transformacijos iš gauto spektro bus galima rasti ieškomus dažnius ir amplitudes. Paprastai prieš atliekant GFT, įėjimo signalui yra taikoma lango funkcija [23].

Kaip žinia, sudėtingos formos kintantį laike signalą galime išskaidyti į sinusų ir kosinusų sumą (Furjė transformacijos atveju). Būtent šių formų signalai sklisdami aplinka nekeičia savo formos ir dažnio, o kinta tik fazė ir amplitudė. Nagrinėjamo garso signalo kitimas yra tolydus periodinis dydis. Tam tikrais vienodais laiko intervalais matuojant  $N$  atskaitų reikšmes, gauname diskretų signalą. Norint išskaidyti apdorojant garso signalus gautą jų kitimo kreivę į sinuso ir kosinuso funkcijas, iškyla problema: šios funkcijos yra begalinės, o analizuojamas signalas – baigtinės trukmės. Tačiau galime daryti prielaidą, kad analizuojamo diskretaus signalo reikšmės yra periodinės ir kartojasi iki begalybės. Tokiu būdu paskutinė  $N - 1$  atskaitos reikšmė kartojantis signalui tampa gretima pirmajai sekančio periodo atskaitos reikšmei. Tai gali privesti prie signalo trūkių (4.1 pav.). Siekiant išvengti tokio signalo netolygumo, prieš vykdant GFT algoritmą yra naudojami langai. Langų tipų būna įvairių, tačiau dažniausiai naudojami šie: trikampis langas, Haningo, Hamingo ir Blakmano (angl. *Blackman*) langai [23].



4.1 pav. Įėjimo signalo trūkio taškas, vedantis prie signalo spektro „nutekėjimo“



4.2 pav. Įėjimo signalas nenaudojant lango funkcijos (kairėje) ir su lango funkcija (dešinėje)

Skirtingos langų funkcijos turi ir skirtingas joms būdingas charakteristikas. 4.1 lentelėje pateikti šių langų palyginimai.

4.1 lentelė. Langų tipų palyginimas [23]

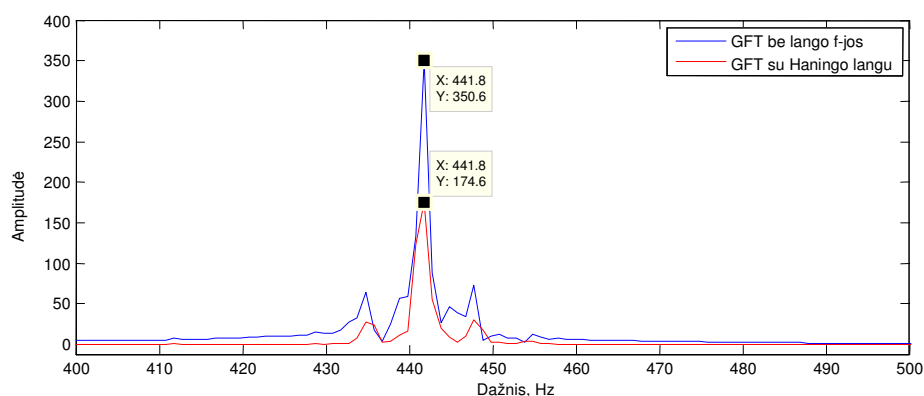
Lango tipas	Pagrindinių spektro dedamųjų plotis	Stiprinimas, dB	Šoninių spektro dedamųjų slopinimas, dB
Be lango	1 dedamoji	1,00	13 dB
Trikampis	3 dedamosios	0,50	27 dB
Haningo	3 dedamosios	0,50	32 dB
Hamingo	3 dedamosios	0,54	43 dB
Blackmano	5 dedamosios	0,42	58 dB

Remiantis 4.1 lentele, buvo pasirinktas Haningo langas. Šis langas skaičiuojamas pagal formulę:

$$W(k) = 0,5 - 0,5 \cos(2\pi k / N), \quad \text{kai } 0 \leq k < N, \quad (4.3)$$

čia:  $W(k)$  – lango koeficientas;  $k$  – atskaitos numeris;  $N$  – atskaitų skaičius.

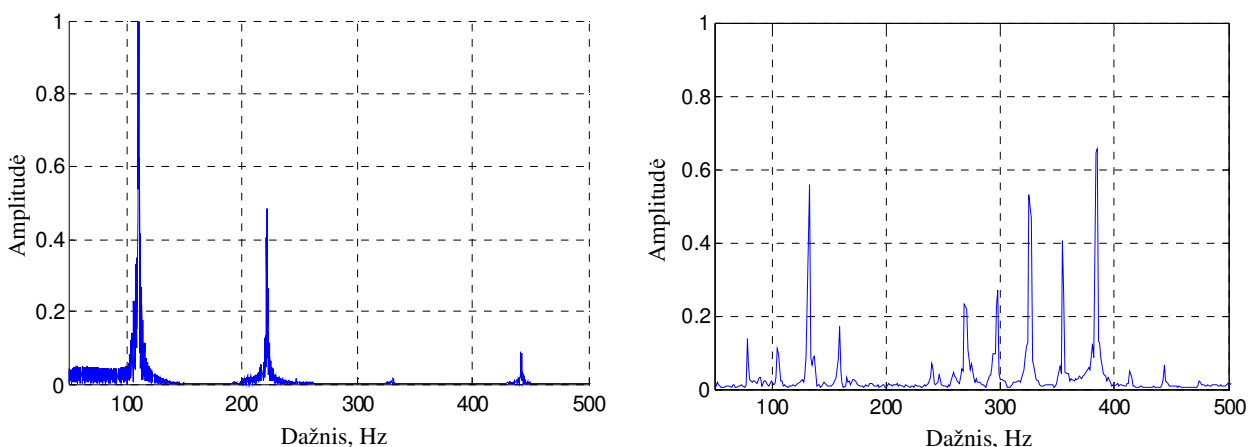
Kaip jau buvo minėta, prieš atliekant GFT tiriamiems signalams reikia pritaikyti lango funkciją. "Langavimas" sumažina signalo energiją, todėl siekiant atstatyti prarastą signalo amplitudės dalį, periodiniams signalams spektrą galima padauginti iš lango svorio koeficiento. *MATLAB* programoje šis koeficientas randamas taip:  $W = 1 / \text{mean}(W)$ .



4.3 pav. A4 natos signalo spektras be lango funkcijos ir pritaikius langą

Iš 4.3 pav. matyti, kad signalo spektro dedamųjų amplitudė pritaikius Haningo lango funkciją sumažėjo dvigubai. Tad šiuo atveju lango svorio koeficientas bus lygus 2. Tokią reikšmę apskaičiavo ir anksčiau minėta *MATLAB* programos funkcija.

Kaip žinia, diskretizuojant signalą atsiranda pagrindiniam dažniui kartotinės harmonikos. Būtent dėl šios priežasties ir iškyla būtinybė taikyti papildomus algoritmus, leidžiančius iš spektro pašalinti nereikalingas harmonikas ir taip rasti centrinį dažnį, kuris atitinka grojamą natą. Siekiant vaizdžiau iliustruoti šią problemą, pateiksime gitaros ir pianino tos pačios natos spektrą (4.4 pav.).



4.4 pav. Gitaros natos A2 (110 Hz) spektras (kairėje) ir tos pačios pianino natos spektras atlikus GFT (dešinėje)

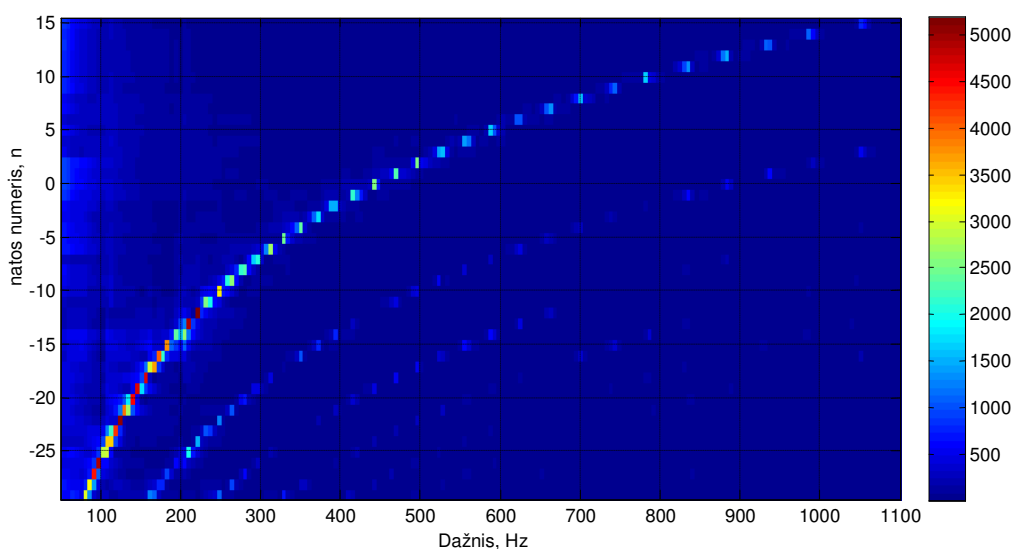
Kaip matyti 4.4 pav., gitaros A2 natos spektras susideda iš pagrindinio dažnio  $f_0$  (110 Hz) ir jam kartotinių harmonikų:  $2f_0$ ,  $3f_0$  ir t. t. Tačiau iš šio spektro galima iš karto pasakyti, kokia yra natos pagrindinio dažnio dedamoji. Tuo tarpu pianino A2 natos spektras yra kur kas sudėtingesnis, net būtų sunku (praktiškai neįmanoma) nustatyti pagrindinį dažnį iš anksto jo nežinant.

Ištyrinėjus visų įrašytų skirtingų 45 gitaros natų spektrus, juose paprastai matoma iki 6 kartotinių harmonikų. Aukštesnių eilių harmonikų amplitudė nykstamai mažėja, lyginant su pagrindinio dažnio dedamąja  $f_0$ . Dar verta pastebėti, kad nė vienos harmonikos amplitudė neviršijo pagrindinio dažnio dedamosios amplitudės. Tai galėtų palengvinti gitaros natų atpažinimą paprasčiausiai randant maksimalią amplitudės reikšmę. Tačiau šis būdas tikėtų nebent tuo atveju, jeigu būtų „užgaunama“ tik viena gitaros styga tam tikru laiko momentu. Jei yra „užgaunamos“ kelios stygos tuo pat metu, jų pagrindinės dažnių dedamosios spektre gali būti skirtingų amplitudžių. Vadinasi, eliminavus visas harmonikas (virpesio dedamąsias, kurių dažniai yra pagr. dažnių kartotiniai), natas galima būtų atpažinti paprasčiausiai radus visus maksimumus spektre.

Tarp visų natų dažnių ir pagrindinės derinimo natos A4 (440 Hz) yra tikslus matematinis ryšys, kurį galima išreikšti formule:

$$f = 440 \cdot 2^{\frac{n}{12}}, \quad (4.4)$$

čia:  $f$  – ieškomos natos dažnis;  $n$  – atstumas pustoniais tarp A4 ir ieškomosios natos. Taigi, natai A4  $n = 0$ , A4# natai  $n = 1$  ir t. t. Taigi, mūsų jau aptartoje gitaroje  $n$  kinta nuo  $-29$  iki  $15$  (E2–C6 natos). Šių natų dažninis išsidėstymas pateiktas spalviniame žemėlapyje (4.5 pav.).



**4.5 pav.** Natų spalvinis žemėlapis, gautas paeiliui nuskaičius visas natas

Kaip matyti iš 4.5 pav., gauti natų dažniai atitinka taikant (4.4) rastus dažnius (jie spalviniame žemėlapyje atitinka šviesesnes spalvas). Tačiau pateiktame paveiksle matyti ne viena natos numerio priklausomybė nuo dažnio, o kelios, t. y. tą patį natos numerį atitinka ne vienas dažnis, o keli. Taip yra dėl jau anksčiau aptartų priežasčių: signalo spektras sudarytas ne tik iš pagrindinės harmonikos, bet ir keleto kartotinių harmonikų. Taip pat galime pastebėti, kad žemesniųjų natų spektro intensyvumas yra didesnis.

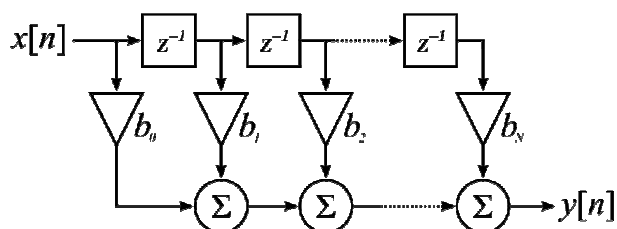
Yra žinoma nemažai algoritmų, kurių pagalba galima rasti pagrindinės dedamosios dažnį. Vieni jų taikomi spektrui, pavyzdžiui, harmonikų spektrų sandaugos (angl. *Harmonic product spectrum*) algoritmas. Kiti (autokoreliacija) gali būti naudojami signalui apdoroti laiko sritys erdvėje. Tuo tarpu įvairūs filtrai gali būti taikomi tiek dažnių tiek laiko sritys. Tolimesniems tyrimams buvo pasirinkti ribotos impulsinės reakcijos FIR (angl. *Finite Impulse Response*) filtrai, harmonikų spektro sandaugos ir autokoreliacijos algoritmai.

## 4.2. Tyrimų rezultatai naudojant skaitmeninius filtrus

Skaitmeniniai filtrai – tai viena iš svarbiausių signalų apdorojimo komponentų. Skaitmeniniai filtrai ypač paplitę tokiose srityse kaip garso signalų apdorojimas. Filtrų taikymas gitaros natų dažniui rasti pasirodė ganėtinai priimtinas dėl jų paprastumo bei greitaveikos (jei žinoma nenaudojamas labai aukštos eilės filtras). Buvo pasirinktas ribotos impulsinės reakcijos FIR filtras [24]. Jis turi eilę privalumų, lyginant su begalinės impulsinės reakcijos IIR (angl. *Infinite Impulse Response*) filtru:

- šie filtrai visada stabilūs;
- filtro eilei nėra jokių apribojimų;
- statesnė filtro charakteristika.

Ribotos impulsinės reakcijos filtro veikimo schema pavaizduota 4.6 pav.



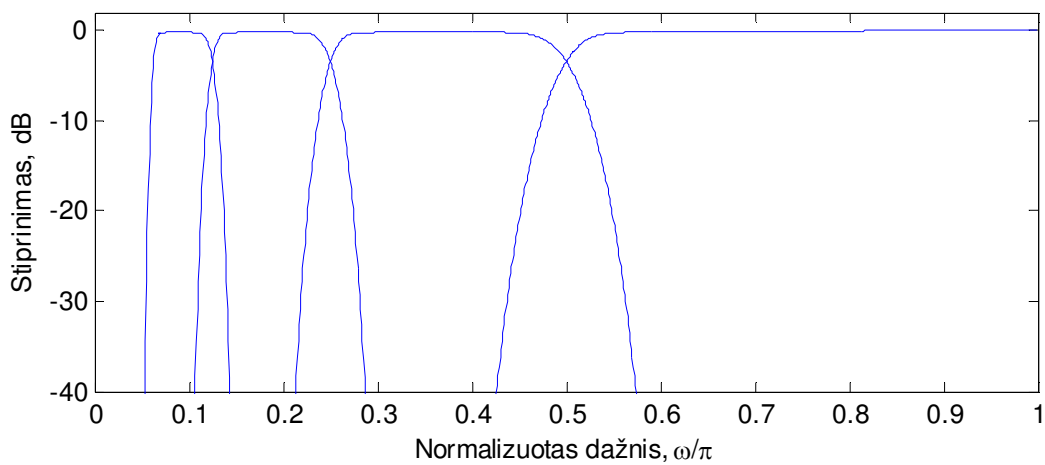
4.6 pav. Diskretaus laiko FIR filtras [24]

Tokio filtro funkciją galime užrašyti formule:

$$y[n] = \sum_{k=0}^N b_k x[n-k],$$

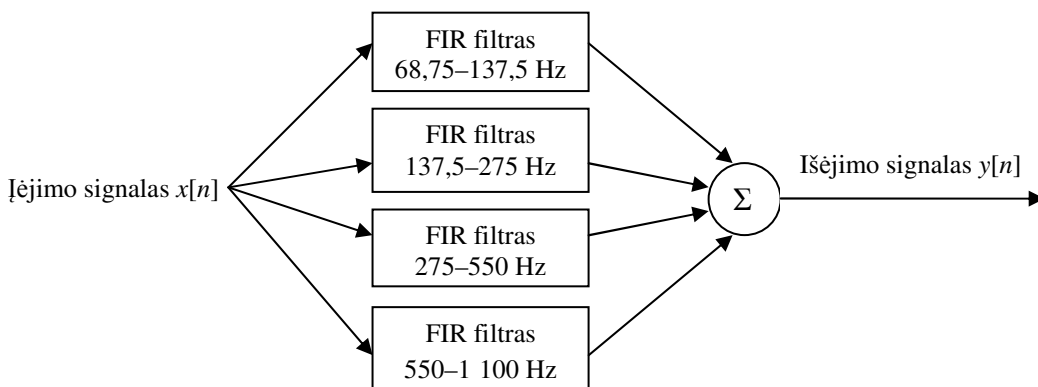
čia:  $N$  – filtro eilė;  $x[n]$  – įėjimo signalas;  $y[n]$  – išėjimo signalas;  $b_k$  – filtro koeficientai.

Kadangi antroji harmonika nutolusi nuo pirmosios (centrinio dažnio) lygiai du kartus, antroji – tris ir t. t., todėl galima būtų įgyvendinti daugiajuostį filtrą, kurio pralaidos juostos būtų dvigubai didesnės viena už kitą (kiekviena oktava būtų filtruojama atskirai). Pavyzdžiui, aukščiausios natos C6 dažnis yra 1 046,5 Hz, o per oktavą žemiau esančios C5 natos dažnis dvigubai mažesnis, C4 natos dar dvigubai mažesnis. Vadinasi, pasirinkus Naikvisto kriterijų tenkinantį diskretizavimo dažnį 2,2 kHz, galėtume gauti juostinių filtrų rinkinį tokiems dažniams: 68,75–137,5 Hz; 137,5–275 Hz; 275–550 Hz; 550–1 100 Hz. *MATLAB* programoje išskvietus *fdatool* (angl. *Filter Design and Analysis Tool*) įrankį ir įvedus norimus parametrus (filto eilę, dažnių diapazonus ir pan.), galime gauti apskaičiuotus filtro koeficientus, taip pat filtro atsako į signalą charakteristiką (4.7 pav.).



4.7 pav. 4 juostų FIR filtro DACH charakteristika, kai pasirinkta filtro eilė  $N = 64$

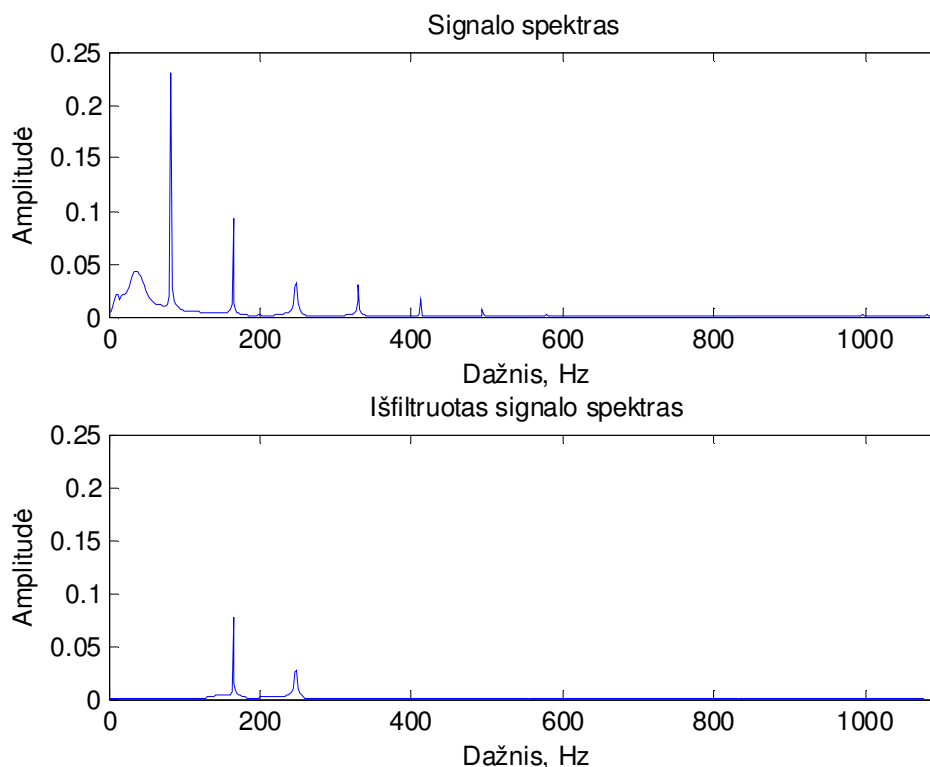
Šitaip išfiltravus signalą lygiagrečiai naudojant 4 filtrus, ir po to vėl susumavus atskiras jo dalis, gautume vientisą signalo spektrą (4.8 pav.).



4.8 pav. 4 juostų FIR filtrų taikymo schema gitaros natų dažniams filtruoti

Turint juostinį 68,75–137,5 Hz filtrą ir filtruojant, pavyzdžiui, E2 (82,407 Hz) natą, jos harmonikos neturėtų pereiti į kito filtro dažnių juostą. Be to, jei pirma harmonika atsiduria už filtro ribų, tai į tą juostą neturėtų patekti ir kartotinės jos harmonikos. Kitaip tariant, jei

panaikiname kartotinių harmonikų priežastį (pirmąją harmoniką), tai turėtų pranykti ir pastarosios. 4.9 pav. pateiktas E2 natos spektras prieš ir po juostinio 137,5–275 Hz FIR filtro.



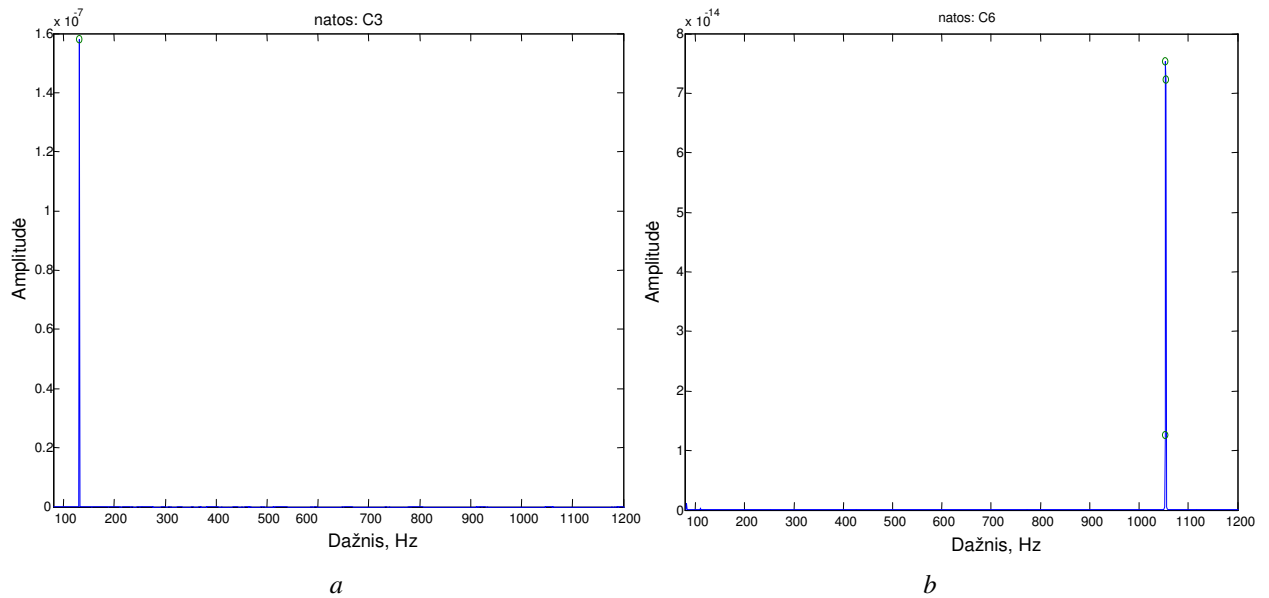
**4.9 pav.** Nefiltruotas natos E2 spektras (viršuje) ir spektras pritaikius skaitmeninį juostinį FIR 137,5–275 Hz filtrą, kurio eilė  $N = 64$  (apačioje)

Kaip matyti iš 4.9 pav., net ir nufiltravus pirmąją harmoniką, kartotinės harmonikos lieka, todėl toks filtrų panaudojimo būdas netenka prasmės, nes galiausiai susumavus visų filtrų išėjimo amplitudes bus gaunamas pradinis signalas. Tai gana netikėta išvada, nes buvo filtruojamas ne gautas spektras, bet pradinis įėjimo signalas laiko srityje, tačiau, pasirodo gaunami rezultatai yra tokie pat. Todėl tolimesniuose tyrimuose išbandysime harmonikų spektrų sandaugos bei autokoreliacinį algoritmus.

### **4.3. Tyrimų rezultatai naudojant HSS algoritmą**

Vienas plačiau taikomų algoritmų kartotinėms harmonikoms panaikinti dažnių srityje yra harmonikų spektrų sandaugos (HSS) algoritmas. Atliekant natų spektrų tyrimus, buvo pastebėta, kad pritaikius HSS algoritmą, spektro dedamųjų amplitudės gerokai sumažėjo, lyginant su pradinėmis. Tačiau šiame tyrimo etape tai neturi lemiamos įtakos, nes jas galima padauginti iš norimo proporcingumo koeficiento. Atliekant tyrimus, spektro sutankinimo ir daugybos operacija buvo pakartota tris kartus, atsižvelgiant į gautus geriausius rezultatus. Beje, net ir pritaikius šį algoritmą, kartais lieka nepageidaujamos kartotinės pagr. dažniui harmonikos. Peržiūrėjus visų natų spektro dedamąsias, buvo nustatyta, kad jos neviršija 1/10 pagrindinės

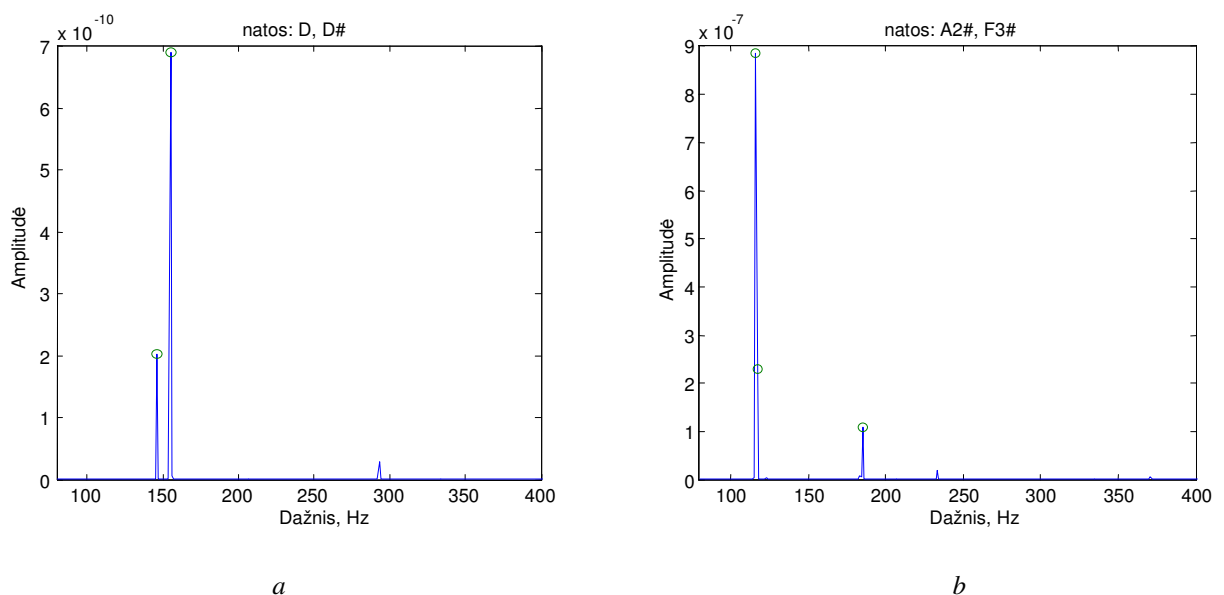
harmonikos aukščio. Atsižvelgiant į tai, buvo įvesta papildoma apsauga nuo neteisingo natos dažnio interpretavimo: radus maksimalią amplitudę, jos vertė padalinama iš 10, ir visos spektro dedamosios, kurios yra žemiau šios vertės, paprasčiausiai ignoruojamos. Tačiau tuomet iškyla kita problema: aukštesniųjų oktavų natoms būdinga mažesnė amplitudė, kaip pavaizduota 4.10 pav. Beje, sėkmingai atpažintų natų dažnių pikinės reikšmės spektre pažymėtos apskritimais.



4.10 pav. Natos C3 (a) ir trimis oktavomis aukščiau esančios natos C6 (b) spektras

Kaip matyti iš 4.10 pav. *a* ir *b*, aukštesnės natos C6 amplitudė yra mažesnė už C3 amplitudę net  $2 \cdot 10^6$  karto. Kitais atvejais tokio drastiško skirtumo nėra. Tai gali privesti prie to, kad, kuomet reikės atpažinti ne atskirą natą, o kelias vienu metu, dėl tokio didelio amplitudžių skirtumo aukštesnioji nata tiesiog bus ignoruojama dėl anksčiau aptartų priežasčių.

Toliau buvo atlikti tyrimai, bandant atpažinti iš karto dvi natas (4.11 pav.).

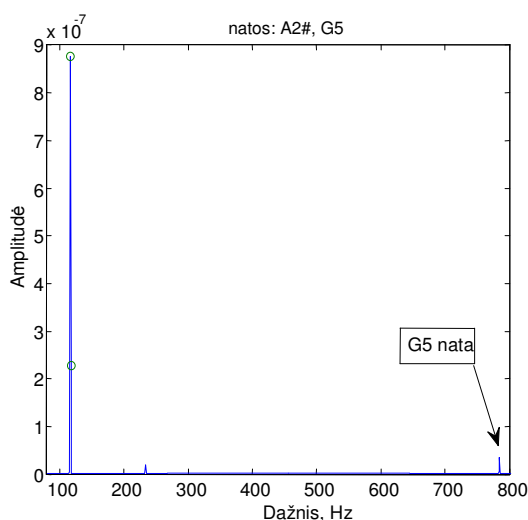


4.11 pav. Dvi teisingai atpažintos gretimos natos D3 ir D3# (a) ir atsitiktinai parinktos A2# bei F3# (b)



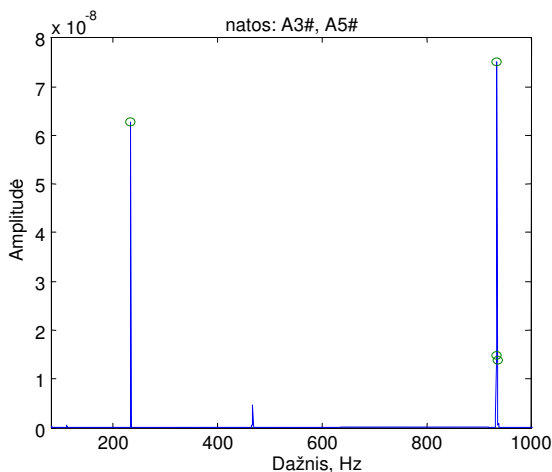
Taigi, pirmiausia buvo bandoma atpažinti dvi šalia viena kitos esančias natas, nes teoriškai tai padaryti turėtų būti sunkiausia dėl mažiausio galimo dažnių skirtumo. Atsitiktine tvarka buvo atrinkta 20 gretimų natų ir atlikus harmonikų spektrų sandaugos algoritmą visos jos buvo atpažintos teisingai. 4.11 pav. *a* pateiktas dviejų gretimų natų D3 ir D3# spektras, kuriame matyti, kad abi natos atpažintos teisingai.

Atsitiktinai pasirinktus ne gretimas natas ir naudojant HSS algoritmą, kai kurias iš jų taip pat pavyko atpažinti be klaidų (4.11 pav. *b*). Iš viso buvo atlikti 70 bandymų su atsitiktinai parinktomis ne gretimomis natų poromis. Deja, tačiau tik 42 natų poros buvo atpažintos teisingai. Tuo tarpu kitos, kurių dažniai skyrėsi keliomis oktavomis, dėl pernelyg didelių amplitudžių skirtumų buvo ignoruojamos (4.12 pav.).



**4.12 pav.** Dviejų vienu metu užgautų natų A2# ir G5 signalo spektro dedamosios. G5 nata liko neatpažinta

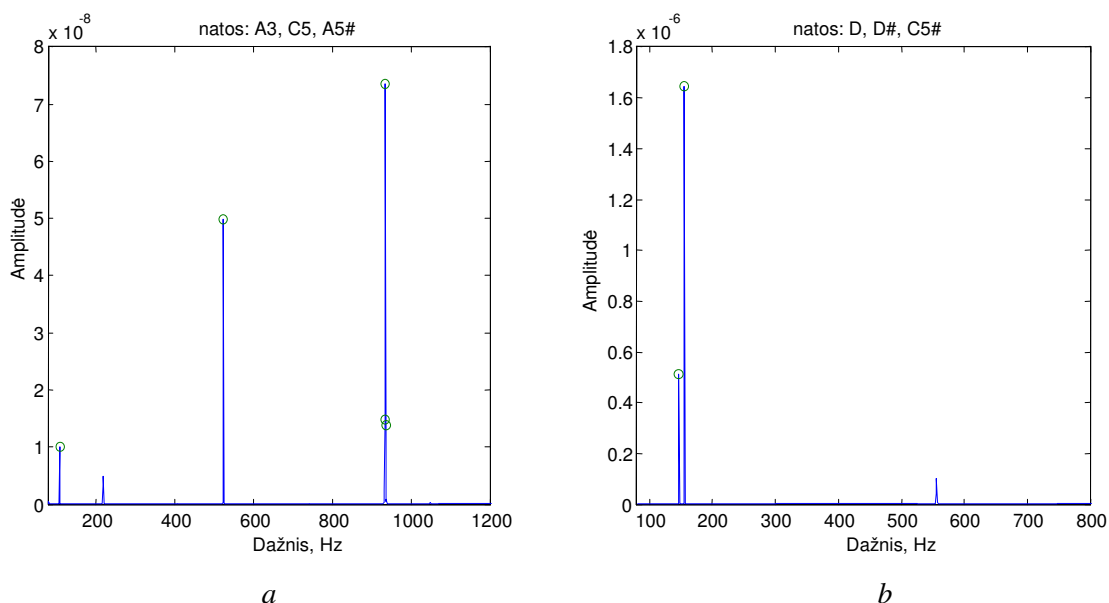
Kitas įdomus atvejis – natos, kurios pasikartoja kas oktavą (žemesniosios oktavos natos harmonikos sutampa su aukštesniosios natos harmonikomis). Tokiais atvejais (4.13 pav.) HSS algoritmas taip pat pasiteisino.



**4.13 pav.** Dvi sėkmingai atpažintos natos A3# ir A5#

Pagal 4.12 ir 4.13 pav. galime spręsti, kad gitaros natų atpažinimo rezultatai naudojant HSS algoritmą yra sunkiai prognozuojami ir priklauso tik nuo tam tikrų natų amplitudžių dydžio ir nuo to, kaip gerai sutampa suglaudinto spektro dedamosios. Kuo daugiau kartų atliekama harmonikų sutankinimo ir sandaugos operacija, tuo geriau nufiltruojamos kartotinės harmonikos, tačiau ir neatpažintų natų skaičius proporcingai didėja.

Gautų rezultatų su įrašytų trijų vienu metu užgautų stygų garsų spektru pateikti 4.14 pav.



**4.14 pav.** Trijų stygų skambesio spektro dedamosios, kuomet pavyko atpažinti visas natas (*a*) ir kuomet ne (*b*)

Iš 4.14 pav. *a* matome, kad natas A3, C5, A5#, net ir išsidėsčiusias palyginti plačiame dažnių diapazone, vis tik pavyko atpažinti teisingai. Kitu atveju (4.14 pav. *b*) dvi natas (D3 ir D3#) identifikuoti pavyko, tačiau trečioji – C5# – liko neatpažinta.

Gauti rezultatai pritaikius harmonikų spektrų sandaugos algoritmą vienai, dviems ir trimis vienu metu užgautų stygų natoms atpažinti pateikti 4.2 lentelėje.

**4.2 lentelė.** Tyrimų rezultatai naudojant HSS algoritmą

<b>Vienu metu užgautų gitaros stygų skaičius</b>	<b>Atliktų bandymų skaičius</b>	<b>Sėkmingų atpažinimų skaičius</b>
1	90	90
2	90	62
3	90	35

Taigi, kaip matyti iš 4.2 lentelės, pritaikius harmonikų spektrų sandaugos algoritmą pagrindinėms spektro dedamosioms rasti, pavienes natas pavyko atpažinti visada (per visus 90 bandymų buvo gautas teisingas centrinis dažnis, atitinkamai teisingai atpažinta ir nata). Bandant atpažinti dviejų užgautų stygų natas vienu metu, gauti šiek tiek kuklesni rezultatai. Iš

viso buvo atlikti 90 skirtingų bandymų, iš kurių abi natas teisingai atpažino 62 kartus. Tuo tarpu užgavus tris stygas vienu metu, teisingai natas pavyko atpažinti tik 35 atveju iš 90 skirtingų natų kombinacijų. Ypač daug klaidų pasitaikydavo bandant atpažinti aukštesnio dažnio natas, nes jų amplitudė mažesnė ir virpesiai greičiau gęsta. Taip gali būti dėl to, kad šios stygos yra gerokai labiau įtemptos ir mažiau vibruoja, o kartotinės harmonikos beveik išnyksta.

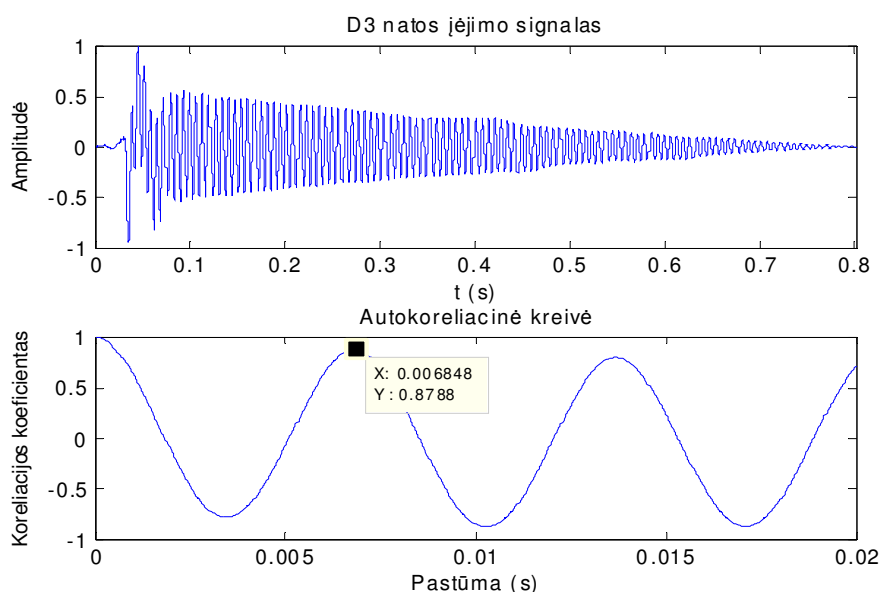
Daugiau bandymų atlikti nėra prasmės, nes aiškiai matyti, kad šio algoritmo tinkamumas kelioms vienu metu grojamoms gitaros natoms yra pakankamai mažas (kuo daugiau natų tenka atpažinti vienu metu, tuo algoritmo tikslumas tendencingai mažėja).

#### 4.4. Tyrimų rezultatai naudojant autokoreliacinį algoritmą

Šiame skyrelyje trumpai aptarsime autokoreliacinio algoritmo bandymų rezultatus. Šis algoritmas, parašytas programa *MATLAB*, buvo rastas internete [25]. Nežymiai jį pakoregavus (pakeitus vėlinimo parametą, t. y. pastūmą), buvo atlikti tokie pat bandymai kaip ir su HSS algoritmu, t. y. nuskaityti visi gitaros įrašai ir peržvelgti gauti rezultatai. Beje, koreliacijos koeficientas parenkamas pagal tiriamų garsų minimalaus tono dažnį. Pavyzdžiui, žmogaus kalbos minimalus tonas gali siekti 50 Hz. Tuomet signalo pastūma apskaičiuojama iš šio dažnio periodo ir būtų lygi 0,02 s.

Tiriamų gitaros garsų žemiausios natos teorinis dažnis yra 82,407 Hz. Tad pastūmos periodą skaičiuosime su atsarga, laikydami, kad mažiausias dažnis gali siekti 80 Hz. Tuomet atliekant autokoreliaciją, signalą savo paties atžvilgiu reikės pastumti per 0,0125 s.

Taikant autokoreliacijos algoritmą, tyrimai buvo atlikti su visais 45 skirtingais gitaros garsų tonais. Žemiau pateiktas D3 natos įrašo (teoriškai jos pagrindinis dažnis yra 146,83 Hz) laikinis signalas ir gauta autokoreliacijos kreivė (4.15 pav.).

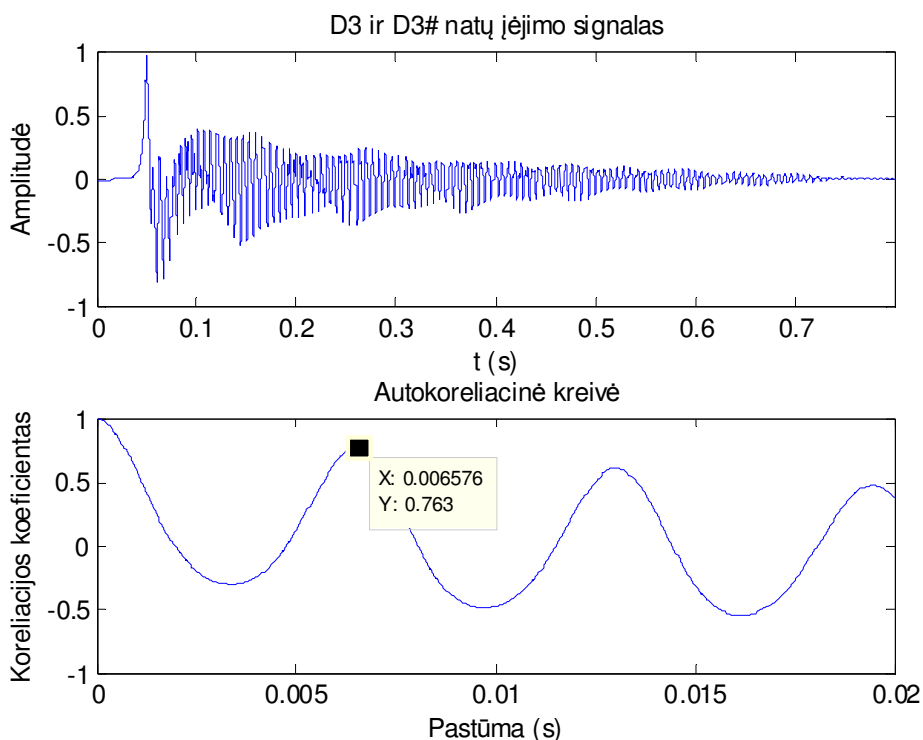


4.15 pav. D3 natos įėjimo signalas (viršuje) ir gauta autokoreliacijos kreivė (apačioje)

Pagal gautą autokoreliacijos kreivę (4.15 pav. apačioje) buvo apskaičiuotas centrinis natos dažnis. Norint jį apskaičiuoti, reikia rasti pirmojo autokoreliacinės kreivės maksimumo reikšmę abscisių ašyje. Ši reikšmė nurodo ieškomo dažnio periodą, kuris duotu atveju lygus 146,03 Hz. Lyginant su teoriniu D3 natos dažniu (146,83 Hz), rezultatai skyrėsi 0,54 %. Atlikus analogiškus bandymus su likusiais devyniais tos pačios natos įrašais, gauti rezultatai labai panašūs – gautas nuokrypis neviršija  $\pm 0,6$  % nuo centrinio dažnio. Šis nuokrypis gali atsirasti dėl to, kad nepakankamai tiksliai suderinta gitara, taip pat ir dėl skaičiavimo paklaidų. Tačiau jis su kaupu tenkina reikalavimus, nes tam, kad nata būtų atpažinta teisingai, nuo centrinio dažnio maksimaliai (paliekant kelių hercų tarpą tarp gretimų natų) galima būtų nukrypti per 3 Hz (gretimų natų centriniai dažniai yra ties 138,59 Hz bei 155,56 Hz), o tai sudarytu  $\pm 2$  % paklaidą.

Su kitomis 45 natomis taip pat gauti labai panašūs rezultatai, o didžiausia apskaičiuoto dažnio paklaida nuo teorinės vertės siekia iki  $\pm 0,74$  %. Taigi, naudojant autokoreliacinį algoritmą, pavienės natos buvo tinkamai atpažintos visais atvejais.

Tokia pat bandymai buvo atlikti ir su polifoniniais gitaros garsais. Deja, tačiau šio algoritmo negalima taikyti kelioms natoms vienu metu atpažinti, nes jis, kaip paaiškėjo bandymų eigoje, randa tik labiausiai dominuojantį dažnį garse. Pavyzdžiui, bandant atpažinti atitinkamai D3 (146,83 Hz) ir D3# (155,56 Hz) natas vienu metu, buvo gautas rezultatas, esantis tarp šių dažnių – 152,07 Hz (4.16 pav.).



**4.16 pav.** Dviejų D3 ir D3# natų įėjimo signalas (viršuje) ir gauta autokoreliacijos kreivė (apačioje)

Beje, buvo pastebėta, kad gautas dviejų natų pagrindinio dažnio rezultatas gali pasislinkti tai į vieną, tai į kitą gretimų natų dažnių pusę, priklausomai kurios iš jų amplitudė yra didesnė. Tas pats galioja, jei bandoma atpažinti tris ir daugiau natų vienu metu.

#### **4.5. Skyriaus išvados**

Šiame skyriuje buvo atliktas kelių dažniausiai naudojamų algoritmų dominuojantiems signalė dažniams išskirti tyrimas *MATLAB* programa. Gauti rezultatai nuviliantys – nei vienas iš tirtų metodų nedavė pakankamai gerų rezultatų. Aptartas skaitmeninių FIR filtrų taikymas atmestinas dėl to, kad nepavyksta pašalinti kartotinių harmonikų (galbūt naudojant kitokio pobūdžio filtrus bei kitokį jų taikymo algoritmą pavyktu gauti geresnių rezultatų). Autokoreliacinis algoritmas taip pat nėra tinkamas, nes negalima atpažinti akordų, o tik pavienes natas. Pačius geriausius rezultatus pavyko gauti naudojant harmonikų spektro sandaugos (HSS) algoritmą. Tačiau šio algoritmo trūkumas – nepavyksta išsaugoti signalo amplitudžių, todėl krenta atpažinimo tikslumas (3 natas vienu laiko momentu tinkamai pavyko atpažinti tik 35 atvejais iš 90 bandymų).

Atlikus visų 45 natų spektro tyrimus buvo pastebėta, kad:

- pirmosios harmonikos amplitudė visuomet didžiausia, o kartotinių harmonikų periodiškai mažėja beveik dvigubai;
- kartotinės harmonikos atsiranda tik dažnio didėjimo kryptimi;
- antroji harmonika atsiranda jau kitos oktavos diapazone.

Atsižvelgus į tai galima daryti išvadą, kad mažiau nei per oktavą nutolusias natas galime atpažinti nenaudojant papildomų algoritmų, pakanka atlikti greitąją Furjė transformaciją. Tuo tarpu natoms, nutolusioms per oktavą viena nuo kitos ir daugiau, galima būtų pritaikyti papildomas sąlygas:

- natos turi būti atpažįstamos dažnio didėjimo kryptimi;
- jei atpažintos natos dažnis yra kartotinis dydis prieš tai buvusiam, tada tikrinama jos amplitudė (jei ji didesnė nei 55 % pirmosios natos amplitudės, tuomet tai nauja nata, o jei nesiekia nustatyto dydžio – kartotinė harmonika).

Ši pasiūlyta metodika natoms atpažinti atliekant tik greitąją Furjė transformaciją ir taikant papildomas sąlygas garso spektro dedamosioms bus įgyvendinta realioje gitaros garsų atpažinimo sistemoje.

## 5. GITAROS GARSŲ ATPAŽINIMO SISTEMOS PROJEKTAVIMAS

Atlikus parinktų algoritmų gitaros garsams atpažinti tyrimus kompiuterine programa *MATLAB*, galime pereiti prie realios sistemos projektavimo ir maketo gamybos. Tad šiame skyriuje aptarsime projektuojamos sistemos gitaros garsams atpažinti struktūrinės ir principinės elektrinės schemų sudarymo principus. Trumpai aprašysime jų veikimą, parametrus, pagrindinių mazgų pasirinkimo kriterijus.

### 5.1. Gitaros garsų atpažinimo sistemos struktūrinės schemas sudarymas

Norint sudaryti struktūrinę gitaros garsų atpažinimo sistemos schemą, pirmiausia reikia aptarti įrenginio funkcinius elementus, jų veikimą, ateinančius ir išeinančius signalus ir panašiai. Atsižvelgdami į užduotį galime teigti, kad bus reikalingas blokas gitaros garsiniams signalams nuskaityti, žemų dažnių filtras (tipinė grandinė prieš atliekant garso signalų diskretizavimą), analogas – skaičius (ASK) keitiklis, mikrovaldiklis, ir pasirinktos indikacijos priemonės. Taip pat planuojama atpažintas natas įrašyti į SD ar MMC tipo atminties kortelę MIDI ar kitokiu formatu, kad vėliau galima būtų gautus rezultatus palyginti, pavyzdžiui, su tam tikro kūrinio natomis. Taip pat projektuojamos sistemos testavimo tikslais labai pravartu būtų gautus duomenis (spektrą) siųsti į kompiuterį per nuosekliają USB (ang. *Universal Serial Bus*) sąsają. Tai leistų stebėti spektro pokyčius esant vienam ar kitam garsui ir būtų galima paderinti programinę bei aparatūrinę dalį, jei kažkas veiktų ne taip. Tad aptarsime kiekvieną iš šių funkcinių dalių išsamiau.

Pirmiausiai, norint atlikti skaitmeninį signalų apdorojimą, reikia turėti ką apdoroti, t. y. reikalingas garso signalas. Jį iš aplinkos galime gauti panaudoję plačiajuostį mikrofoną. Mikrofonai dažniausiai būna elektrodinaminiai arba kondensatoriniai (elektretiniai). Jie taip pat turi ir skirtingas kryptingumo diagramas. Nuo jų pasirinkimo priklausys naudojama schema. Vienu ar kitu atveju, vis tiek bus reikalingas mikrofono signalo stiprintuvas.

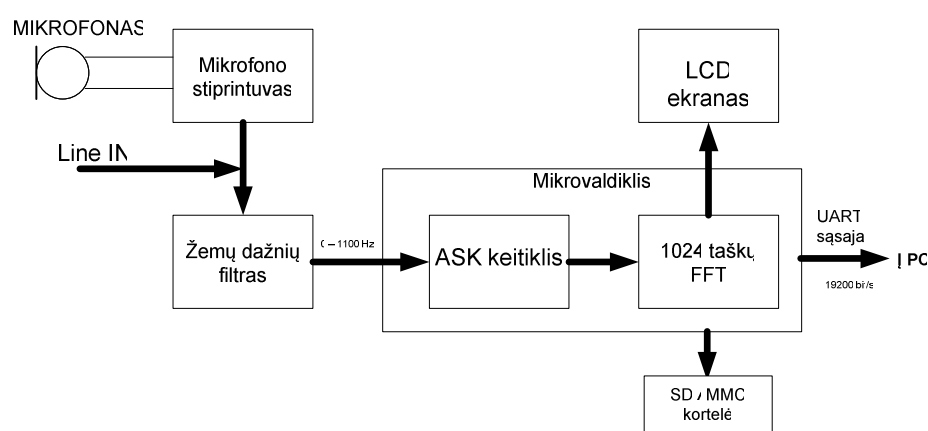
Kadangi gitara su kuria buvo daromi įrašai yra elektroakustinė ir turi linijinį išėjimą, todėl ir projektuojamas įtaisas taip pat turi turėti linijinį įėjimą. Tai leis universaliau panaudoti pastarąjį įtaisą.

Visi mus supantys aplinkos signalai yra analoginiai. Ne išimtis ir garsas. Norint, kad juos suprastų mikrovaldiklis, reikia panaudoti analogas–skaičius keitiklį. Tokiu būdu yra gaunamas skaitmeninis kodas, kuris toliau analizuojamas, skaičiuojama GFT. Dauguma šiuolaikinių mikrovaldiklių jau turi integruotus ASK keitiklius, todėl nereikės naudoti atskiros mikroschemos. Tereiks pasirinkti MV su greičio ir tikslumo poreikius tenkinančiu ASK.

GFT įgyvendinimas reikalauja, kad visi aukštesnio dažnio signalai už maksimalų naudotiną (šiuo atveju 1 046,5 Hz, bet atsargai paliksime 1 100 Hz pralaidos juostą) būtų nufiltruoti žemų dažnių filtru (toliau ŽDF). Tai reiškia, kad prieš skaičius – analogas keitiklį turės būti įgyvendintas ŽDF.

Žmogus savo ruožtu taip pat negali suprasti skaitmeninių signalų, todėl juos reikia pateikti analoginiu pavidalu. Vaizdžiam gauto dažnio atlikus GFT transformaciją pateikimui gali būti naudojamas skystųjų kristalų ekranas (toliau SK), taip pat LED indikacinės lemputės, rodančios kuri styga yra užgauta ir panašiai.

Aptarus minėtus principus, jau galima sudaryti preliminarią projektuojamo įtaiso struktūrinę schemą (5.1 pav.).



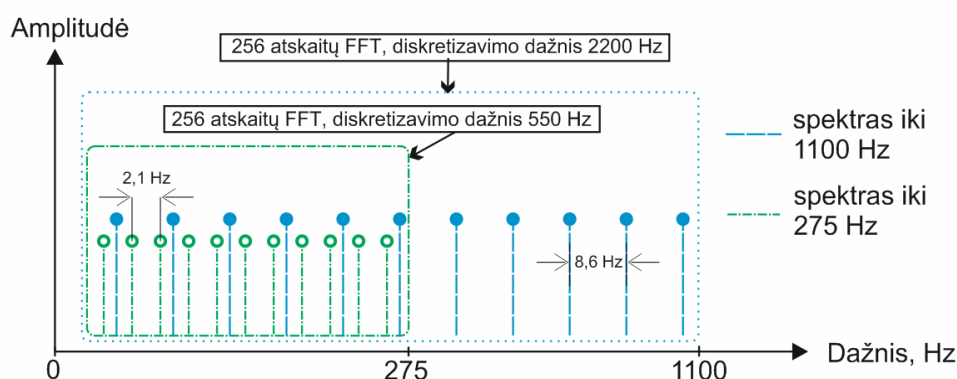
5.1 pav. Preliminari gitaros garsų atpažinimo įtaiso struktūrinė schema

Deja, ši struktūrinė įtaiso schema nėra galutinė. Buvo neatkreiptas dėmesys į atrodytų visai nesvarbų dalyką – mikrovaldiklio duomenų atminties kiekį. Norint gauti tikslesnį spektrą, reikia taikyti bent jau 1 024 taškų GFT transformaciją. Diskretizuojant signalą 2 200 Hz dažniu tai leistų pasiekti 2,15 Hz skyrą tarp gretimų spektro dedamųjų. Tuo tarpu reikalingas duomenų atminties kiekis šiai transformacijai atlikti būtų  $2 \cdot (N + 2 \cdot N + N/2)$ . Jei  $N = 1024$ , tai reikėtų 7 168 bitų RAM atminties. Tokia statinės atminties talpa gali pasigirti toli gražu ne kiekvienas mikrovaldiklis. Be to, optimizuotas GFT algoritmas buvo parsisiųstas iš interneto, jam įvykdyti reikia 73,7 ms, naudojant 16 MHz kvarcinį rezonatorių. Todėl buvo nueita kitu keliu: naudotos dvi 256 taškų GFT transformacijos (atitinkamai du ASK keitiklius ir žemų dažnių filtrus), kurių viena skaičiuotų spektrą iki 275 Hz, o kita iki 1 100 Hz, bet didesniais intervalais. Pagrindiniai tokio sprendimo privalumai:

- dvi 256 taškų GFT transformacijos naudojant tą patį kvarcinį rezonatorių yra įvykdomos greičiau – per 31,8 ms;
- natų dažniai išsidėstę logaritmiu dėsnio, todėl žemesniuose dažniuose reikia didesnės skyros, o aukštesniuose – mažesnės;

- žemesniuose dažniuose gaunama skyra yra tokia pat, kaip ir naudojant 1 024 taškų GFT;
- 256 taškų GFT transformacijai reikia tik 1 792 bitų RAM atminties;
- galiausiai šių dviejų spektrų šalutinės harmonikos nepersidengia, todėl paprasčiau taikyti pasirinktus algoritmus rasti centriniams natų dažniams.

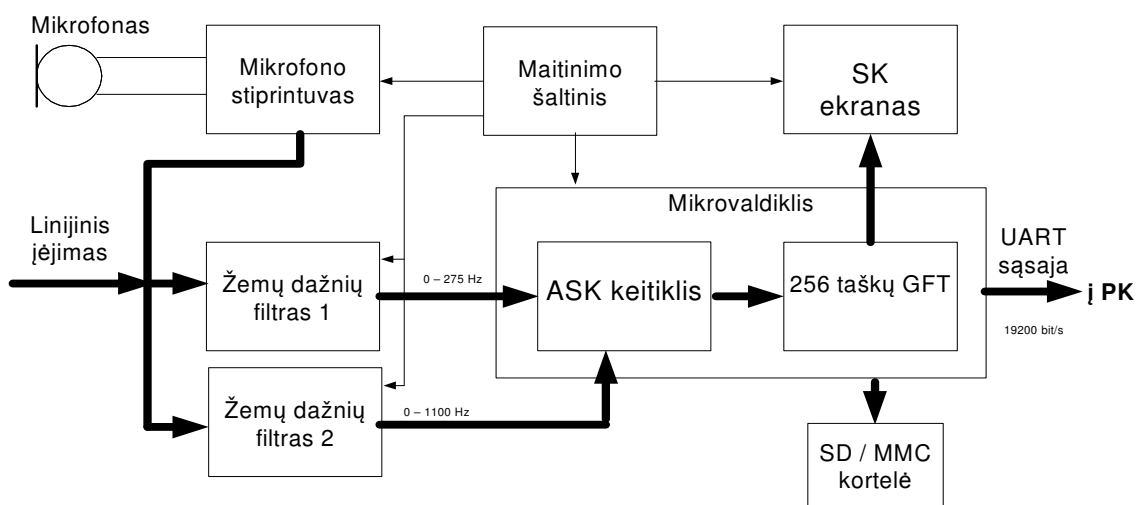
Vienintelis trūkumas – reikia dviejų ŽD filtrų ir ASK keitiklių. Šiuolaikiniai MV dažnai turi 8 ar daugiau integruotų analogas–skaičius keitiklių. Spektra sudėties, gautos atlikus tokį dvigubą GFT skaičiavimą, pavyzdys pateiktas 5.2 pav.



5.2 pav. Analizuojamo signalo spektro persidengimas naudojant dvi GFT transformacijas

Kaip matyti iš 5.2 pav., dažnių spektrui iki 275 Hz atvaizduoti yra naudojamos tankiau išsidėsčiusios atskaitos (su tuščiais apskritimais), o virš šio dažnio – rečiau išsidėsčiusios (su pilnais apskritimais). Jei pastarosios atskaitos spektre yra žemiau 275 Hz, jos paprasčiausiai ignoruojamos.

Pakoreguota projektuojamos sistemos struktūrinė schema su dviem ŽD filtrais pateikta 5.3 pav.



5.3 pav. Galutinė gitaros garsų atpažinimo įtaiso struktūrinė schema



Šioje struktūrinėje schemoje (5.3 pav.) nėra pavaizduoto keitiklio iš UART į USB sąsają bloko, nes jis įtaise nebus integruotas. Tam bus naudojamas paprasčiausias mobilaus telefono duomenų kabelis, nes, kaip jau buvo minėta, šią sąsają planuojama naudoti tik testavimo tikslais.

## 5.2. Įtaiso principinės elektrinės schemos sudarymas

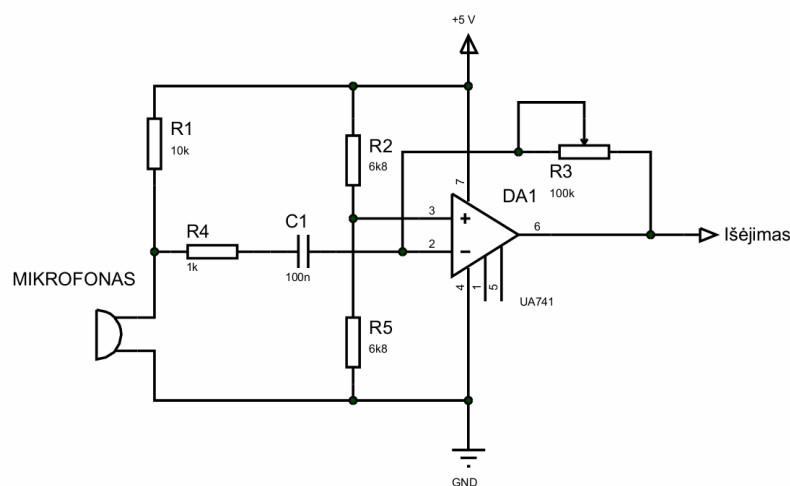
Šiame poskyryje išnagrinėsime atskirų schemos mazgų elementų veikimą, jų parinkimo kriterijus. Didžiausias dėmesys buvo kreipiamas į mikrovaldiklio parinkimą, nes jų yra daug ir įvairių, tačiau skiriasi jų greičiai, integruoti aparatiniai blokai, mašininio ciklo vykdymo laikas ir kiti parametrai.

### 5.2.1. Mikrofono stiprintuvo schemos sudarymas

Tai bene jautriausia trukdžiams schemos dalis. Nuo norimų parametrų garso jautriui ir kokybei, reikia pasirinkti atitinkamą mikrofoną, nuo kurio automatiškai priklausys ir schema. Kadangi gitaros stygų virpesiai periodiniai, o maksimalus dažnis tesiekia 1,1 kHz, tai ypatingos kokybės nereikia. Didelis jautrumas taip pat nepageidautinas, nes bus fiksuojami ne tik naudingi gitaros garsai, bet ir aplinkos triukšmas. Tai gali sąlygoti sistemos nestabilų darbą. Dėl šių priežasčių buvo pasirinktas elektretinis mikrofonas, kurio parametrai tokie:

- dažnių juosta: 50–12 000 Hz;
- maitinimo įtampa: 1,5–9 V;
- stiprinimas 72 dB.

Taip pat buvo paimta standartinė elektretinio mikrofono stiprintuvo schema naudojant mikroschemą UA741 (LM741) [26], kuri pateikta 5.4 pav.



5.4 pav. Standartinė mikrofono stiprintuvo schema su LM741

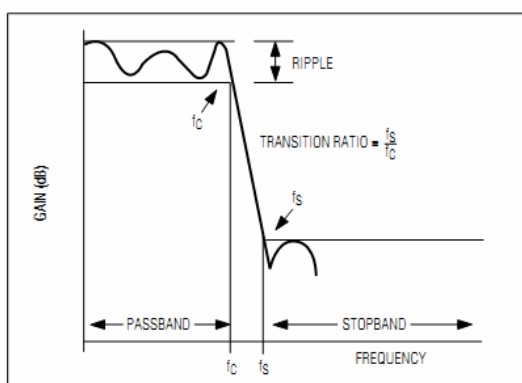
Šioje schemoje rezistoriai R2 ir R5 reikalingi nustatyti nuolatinę įėjimo įtampai, kuri lygi maitinimo įtampai padalintai pusiau. Paderinamas rezistorius R3 reikalingas nustatyti

grįžtamajam ryšiui, juo galima reguliuoti mikroschemos DA1 stiprinimo koeficientą. Paderinamą rezistorių galima pakeisti paprastu, nes jis gerokai mažesnis, be to, nenumatoma kaskart naudojant įtaisą nustatinėti jo reikšmės.

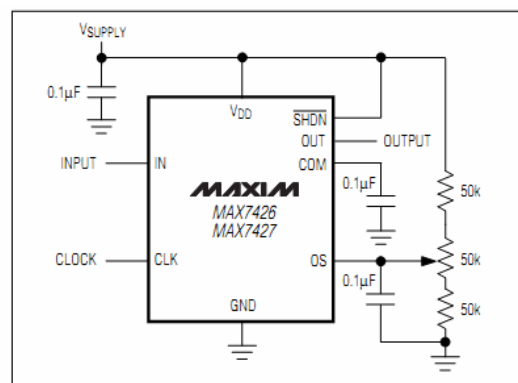
### 5.2.2. ŽD filtrų parinkimas

Prieš apdorojant garso signalą ASK keitikliais, reikia atlikti pirminį jo apdorojimą – nufiltruoti visus aukštesnius už 1 100 Hz dažnius. Šių filtrų konfigūracijų yra visokių. Didžiausias dėmesys buvo kreipiamas į integralumą (kad reikėtų kuo mažiau išorinių komponentų), tai pat į filtravimo laipsnį. Kaip žinia, kuo didesnė filtro eilė, tuo didesnis slopinimas gaunamas už filtruojamo kritinio dažnio. Idėja daryti ŽDF iš operacinių stiprintuvų buvo iš karto atmesta, nes tai užimtų santykinai daug vietos plokštėje, be to, reikėtų daug pasyvių komponentų (kondensatorių, rezistorių ir t. t.).

Po ilgų ieškojimų buvo pasirinktas firmos *MAXIM* produktas – mikroschema MAX7426 [27]. Tai 5-tos eilės žemų dažnių filtras, kurio norimą filtravimo dažnį galima pasirinkti generatoriumi ribose nuo 1 Hz iki 12 kHz. Be to, reikalingas tik minimalus išorinių elementų kiekis. Dar vienas šios mikroschemos privalumas – galima reguliuoti nuolatinės įtampos išėjimo slenkstį (angl. *offset*). Maitinimo įtampa – 4,5–5,5 V. Filtro atsakas į signalą (DACH charakteristika) pateiktas 5.5 pav., o tipinė MAX7426 jungimo schema – 5.6 pav.



5.5 pav. Filtro DACH [27]



5.6 pav. Tipinė MAX7426 jungimo schema [27]

Verta paminėti, kad gamintojas šią mikroschemą ir rekomenduoja aukštos skyros signalo iškraipymams sumažinti atvaizduojant jį mažesne skyra (angl. *Anti-aliasing*) bei garso atpažinimo aplikacijoms. Tiesa, dar reikia atkreipti dėmesį į tai, kad sąryšis tarp praleidžiamų dažnių juostos ir taktinių impulsų šaltinio yra 1:100, t. y. jei norime nufiltruoti signalą ties 1 kHz, į mikroschemos *CLOCK* įvadą reikia prijungti 100 kHz dažnio impulsus.

### 5.2.3. Mikrovaldiklio parinkimas

Gitaros garsų atpažinimo prietaiso širdis yra mikrovaldiklis. Šiam projektui nauduosime gerai žinomą *Atmel AVR* mikrovaldiklių šeimą. Pastaroji pasirinkta dėl to, kad sukaupia nemaža darbo su šiais MV patirtis, turimi visi programavimui reikalingi prietaisai, be to, juos lengva gauti kiekvienoje elektronikos prekių parduotuvėje.

Šioje sistemoje kritinis parametras yra greitis. Todėl pasirinksiame tokį MV, kurio maksimalus kvarcinio generatoriaus dažnis būtų kuo didesnis. Taip pat GFT skaičiavimams reikia labai daug duomenų atminties. Vieną MV teko pakeisti kitu, nes šis turėjo per mažai duomenų atminties, kad leistų atlikti norimą 256 taškų GFT. Nesukompiliavus kodo neįmanoma tiksliai nustatyti, kiek reikės vietos programos ir duomenų atmintyse. Tai padarius, paaiškėjo, kad reikalingas MV turintis bent jau 8 KB programų atmintį ir 4 KB duomenų atmintį. Taip pat būtų gerai, kad jis turėtų vidinį bent 8 bitų skyros ASK keitiklį. Šiuos reikalavimus tenkina ATmega644 mikrovaldiklis [28]. Naudosime DIP40 mikrovaldiklio korpusą, kad prireikus galima būtų nieko nelituojuant perdėti į kitą lizdą, nes surinkus galutinę schemą ir atlikus bandymus, gali tekti ją šiek tiek pakeisti (gali tekti perdaryti ir pačią spausdintinę plokštę).

Aktualiausi MV parametrai yra šie:

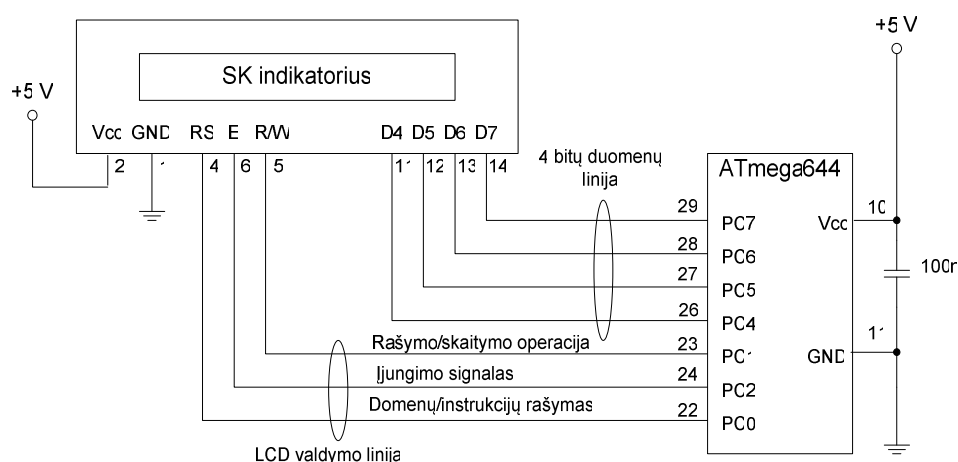
- 64 KB programų *Flash* atmintis;
- 4 KB duomenų *SRAM* atmintis;
- iki 20 MIPS naudojant 20 MHz kvarcinį rezonatorių;
- 8 kanalų, 10 bitų ASK keitiklis;
- kiekvieną išvadą galima panaudoti išorinėms pertraukims (angl. *Pin change interrupt*);
- du 8 bitų ir vienas 16 bitų laikmačiai–skaitikliai;
- maitinimo įtampa 1,8–5,5V.

Taigi, pasirinktas mikrovaldiklis su kaupu tenkina visus jam keliamus reikalavimus. Tačiau rezervą bet kuriuo atveju turėti reikia, nes gali prireikti daugiau išvadų kokiai nors periferijai prijungti ar papildomiems indikacijos elementams. Taip pat gali tekti įgyvendinti kelis ar daugiau algoritmų natų dažniams apskaičiuoti, todėl didesnis atminties kiekis taip pat pageidautinas.

### 5.2.4. Skystųjų kristalų indikatoriaus sąsajos sudarymas

Informacijai apie grojamas natas išvesti panaudosime SK indikatorių. Naudosime 4 bitų duomenų perdavimo režimą. Nors jis perduoda duomenis dvigubai ilgiau (po 4 bitus du kartus), tačiau tam tereikia vieno mikrovaldiklio išvesties prievado. Pasirinktas mikrovaldiklis turi 4 bendros paskirties prievadų grupes (PORTA–PORTD), kurių kiekvienas turi po 8 išvadas.

Pasirinksime PORTC prievadą, nes projektuojant spausdintinio montažo plokštę bus patogiau išvedžioti takelius, be to, šis prievadas turi mažiau specializuotų išvadų, kurių reikės kitiems schemos mazgams. Žinoma, galima būtų duomenis perduoti iš skirtingų prievadų, tačiau tai nėra optimalu programiniu atžvilgiu, nes daug laiko užimtų atskirų prievadų išvadų inicializavimas. Be to, sudarant MV darbinę programą buvo rasta optimizuota SK indikatoriaus valdymo biblioteka, kuri 4 žemiausias prievado skiltis naudoja duomenims perduoti, o keturias aukščiausias skiltis – valdymo signalams. SK indikatoriaus prievadas RS nustato ar bus priimamos instrukcijos ar duomenys. Prievadas R/W nustato ar į jį bus rašomi, ar skaitomi duomenys iš SK indikatoriaus. Signalas E įjungia indikatoriaus valdiklį. Sudaryta skystųjų kristalų indikatoriaus sąsajos schema pateikta 5.7 pav.



5.7 pav. Parinkto SK indikatoriaus sąsajos su mikrovaldikliu schema

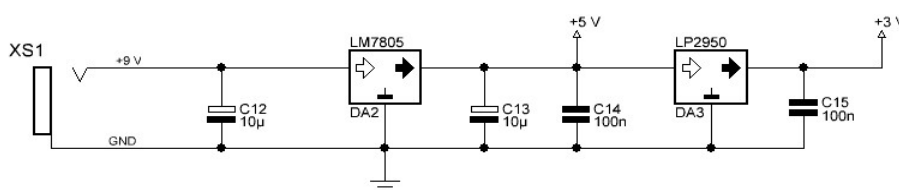
Buvo pasirinktas 16 stulpelių ir 2 eilučių indikatorius su standartiniu kontrolieriu HD44780, galintis atvaizduoti visus ASCII (angl. *American Standard Code for Information Interchange*) lentelės simbolius. Jis neapkrauna mikrovaldiklio bereikalingu darbu, nes turi integruotą atmintį, todėl nereikia pastoviai siųsti informacijos į SK, pakanka ją atnaujinti, kai pasikeičia duomenys. Daugelis standartinių SK indikatorių turi pašvietimą (angl. *Backlight*). Tačiau jo buvo atsisakyta, nes jis naudoja 15–30 mA srovę. Dėl to sutrumpėtų prietaiso darbo laikas.

### 5.2.5. Maitinimo grandinės parinkimas

Maitinimo šaltinio parinkimas – labai svarbus etapas gaminant produktus masiškai. Nuo jo priklauso prietaiso veikimo laikas, ekonomiškumas, dydis ir eilė kitų parametru. Galima rinktis įvairius maitinimo elementus, akumuliatorius ir t. t. Schemoje naudojami elementai vartoja labai nedaug srovės, suminė jų srovė neviršija 100 mA. Paprasčiausias būdas, nereikalaujantis ypatingų schemotechninių sprendimų – naudoti didesnės įtampos maitinimo šaltinį nei reikia komponentams, ir ją sumažinti bei stabilizuoti. To pasakoje įtaisą galima jungti ne tik prie specialaus parinkto šaltinio, bet ir prie plačios gamos akumuliatorių, adapterių ir panašiai.

Nuolatinę įtampą galima labai paprastai sumažinti, pavyzdžiui, naudojant LM78XX serijos analoginį integrinį grandyną. Toks įtampos žeminimo būdas turi nemažą trūkumą – palyginti žemą naudingumo koeficientą. Šiam trūkumui pašalinti naudojami impulsiniai stabilizatoriai, bet jų schemos sudėtingesnės.

Šioje schemoje dauguma mazgų naudoja 5 V įtampą. Todėl parinksime LM7805 teigiamos įtampos stabilizatorių, kurio išėjimo srovė siekia 1 A. Įtaise taip pat planuojama naudoti SD tipo kortelę, o jai maitinti turi būti naudojama 2,7–3,3 voltų įtampa. Tad pasirinksiame 3 voltų stabilizatorių LP2950. Tipinė tokios maitinimo grandinės schema pateikta 5.8 pav.



5.8 pav. Maitinimo grandinės schema

Numatoma, kad gaminamas prietaisas bus maitinamas iš 9 V elemento („kronos“). Šio elemento energijos tikrai pakaks kelioms, o gal ir keliolikai valandų darbo. Tai nėra pats optimaliausias maitinimo šaltinis, tad vėliau tobulinant įtaisą galima bus parinkti kitokį variantą. Ploškštėje numatoma padaryti tik jungtį, prie kurios galima bus prijungti minėtą elementą ar tinklo adapterį.

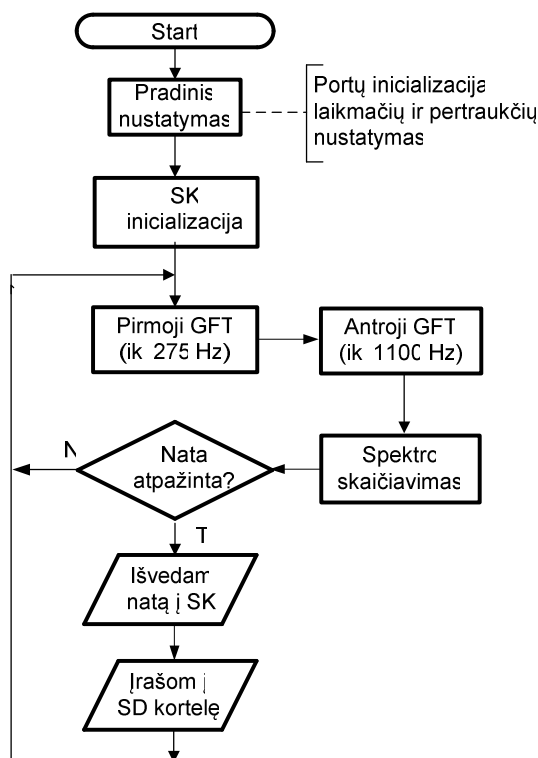
Galutinė projektuojamos gitaros garsų atpažinimo sistemos principinė elektrinė schema sudaryta programa *sPLAN 7.0* ir pateikta prieduose (žr. B priedą).

### 5.3. Mikrovaldiklio programos sudarymas

Programuojant projektuojamą sistemą, buvo pasirinkta C programavimo kalba. Naudoti įrankiai buvo AVR GCC kompiliatorius *WinAVR* bei *AVR Studio 5* programinis paketas [29, 30]. Atlikus paieškas internete, buvo rastas GFT transformacijos optimizuotas kodas, parašytas assemblerio kalba ir pritaikytas būtent 8 bitų AVR šeimos mikrovaldikliams [31]. Šis kodas ir buvo panaudotas atlikti GFT skaičiavimus.

Pirmiausia programuojant buvo apsarašyti kintamieji, nustatytos registrų reikšmės, apskaičiuoti laikmačio–skaitiklio vėlinimai. Garsiniams signalams diskretizuoti buvo pasirinktas *TIMER1* 16 bitų laikmatis. Jis turi dvi 16 bitų skiltis, kuriomis galima valdyti dvi vidines pertrauktis. Viena jų buvo panaudota žemesnio dažnio signalams diskretizuoti, kita – aukštesnio. Vėliau inicializuojamas SK indikatorius, UART duomenų perdavimo sąsaja bei SD kortelės sąsaja. Kuomet visa tai padaryta, įgalinamos laikmačio–skaitiklio pertrauktys ir pradeda skaičiuoti GFT. Šios transformacijos skaičiuojamos nuolat. Tuomet atliekami gauto spektro

skaičiavimai ir tikrinama, ar nata buvo atpažinta. Jei taip, jos duomenys išvedami į skystųjų kristalų indikatorių bei SD atminties kortelę. Vėliau ciklas kartojamas. Supaprastintas mikrovaldiklio programos algoritmas pateiktas 5.9 pav.



5.9 pav. Mikrovaldiklio programos veikimo algoritmas

Kai bus įsitikinta, kad programa veikia neprikaištingai, tuomet galima bus papildomai įvertinti vidutinę signalo energiją, t. y. kai sistemos įėjime garso amplitudė neviršys nustatytos triukšmų normos, skaičiavimai nebus vykdomi. Taip bus taupoma maitinimo šaltinio energija, be to, bus lengviau identifikuoti pauzių tarp natų laiką.

Programos fragmentas, kaip atliekamas diskretizavimas iššaukiant pertrauktis ir jų metu nuskaitant ASK keitiklio duomenis, pateiktas 5.1 pavyzdyje.

### 5.1 programos kodas signalo diskretizavimui

```

#define Timer1_h F_CPU/2200 // nustatom vieną timer1 registrą 2200 Hz dažniui
#define Timer1_l F_CPU/550 // nustatom kitą timer1 registrą 550 Hz dažniui
TCCR1A = 0b00000000; //atliekama timer1 inicializacija
TCCR1B = 0b00011001; // nustatom daliklį (clk/1)
TIMSK1 |= _BV(TOIE1); // įgalinam pertrauktį
OCR1A = Timer1_l;
OCR1B = Timer1_h;
ISR(TIMER1_COMPB_vect) { //iškviečiam pertrauktį
    TCNT1=0;
    TIMSK1 |= (1 << OCIE1B); //įgalinam compare1B
    static uint16_t offset=0;
    uint8_t adc_val;
    ADMUX |= (1 << MUX0); //nuskaitinėjam ASK duomenis
    ADCSRA |= (1 << ADSC);
    while (ADCSRA & (1 << ADSC)) {;}; // laukiam, kol rezultatai bus prieinami
    adc_val = ADCH;
    capture[offset] = (int16_t)(adc_val)-127; // ofsetas 2,5 V
  }
  
```

```

offset++;

if (offset == NUM_SAMPLES) // buferis pilnas
{
    TIMSK1 &= ~_BV(OCIE1B); //išjungiam capture1B pertrauktį
    GFT_input(capture, bfly_buff); //skaičiuojam GFT
    GFT_execute(bfly_buff);
    GFT_output(bfly_buff, spektrum);
    print_GFT(spektrum, NUM_SAMPLES/2);
    offset = 0;
}
}

```

5.1 programos fragmente pateiktas pavyzdys, kaip atliekama 2,2 kHz signalo diskretizavimas (pagal Naikovskio kriterijų  $F_d$  turi būti bent dvigubai didesnis už tiriamą dažnį, kuris yra 1,1 kHz). Analogiškai atliekamas ir 550 Hz diskretizavimo procesas, tik tuomet išjungiamos pertrauktys, skaičiuojančios iki OCR1B laikmačio registro reikšmės (kuomet iškviečiamos 2,2 kHz pertrauktys) ir įjungiamas skaičiavimas iki OCR1A registre nurodytos reikšmės.

Kitas svarbus programos fragmentas, be kurio sistema negalėtų dirbti, tai atitinkamų dažnių filtro mikroschemoms generavimas. Šie dažniai yra generuojami TIMER0 ir TIMER2 skaitiklių pagalba. Programos fragmentas pateiktas 5.2.

### 5.2 programos fragmentas, nustatantis laikmačius TIMER0 ir TIMER2

```

// timer2 nustatymas
TCCR2A = 0b01000010; //pakeičiam OC2A išvado loginį lygį
TCCR2B = 0b00000010; // įjungiamas daliklis iš 8
OCR2A = 45; //skaičiuojame nuo 0 iki 45 (~54 kHz)
// timer0 nustatymas
TCCR0A = 0b01000010; //pakeičiam OC0A išvado loginį lygį
TCCR0B = 0b00000010; //sisteminiis dažnis dalinamas iš 8
OCR0A = 10; //skaičiuojam nuo 0 iki 10 (~227 kHz)
//*****

```

Atlikus skaičiavimus, gauname, kad TIMER0 iškvies pertrauktį tokiu dažniu:

$$T_0 = 20 \cdot 10^6 / 11 / 8 = 227,26 \text{ kHz.}$$

Nepamirškime, kad pertrauktis tik pakeičia loginį lygį MV prievade, tad vienam taktui gauti reikia dažni dar dalinti pusiau. Vadinasi gauname, kad OC0A (PB3 išvadas) generuoja dažnį, lygų 113,63 kHz. Filtrų generatoriaus ir ribinio dažnio koeficientas 1:100, todėl gauname, kad filtras praleis dažnius iki 1,136 kHz. Tai pilnai tenkina reikalavimus.

Analogiškai paskaičiavus TIMER2 pertraukties dažnį, jis yra gaunamas 27,152 kHz. Tai reiškia, kad kitas filtras praleis dažnius iki 271,5 Hz.

Kita svarbi programos dalis – duomenų išvedimas į SK indikatorių. Tam tikslui buvo panaudotos bibliotekos *lcd\_lib.c* ir *lcd\_lib.h*, rastos internete [32]. Norint išvesti simbolį ar skaičių, pakanka pasinaudoti keliomis funkcijomis, kurios pateiktos toliau:

*LCDinit()* – inicializuoti LCD ekraną;

*LCDclr()* – išvalo ekraną;

*CopyStringtoLCD(kintamasis, 3, 0)* – išveda duomenis nurodytoje vietoje;

*LCDGotoXY(0, 1)* – pereiti į nulinį stulpelį ir pirmą eilutę.

Daugiau funkcijų programoje nebuvo naudota, tik šios pagrindinės, nors bibliotekos leidžia progresinio duomenų atvaizdavimo ir kitokias funkcijas.

Kitas svarbus programos fragmentas – duomenų įrašymas į SD tipo kortelę. Kad tai taptų įmanoma, teko įgyvendinti FAT32 failų sistemos palaikymą mikrovaldiklyje. Šios sistemos programinis kodas buvo rastas internete [33]. Taip pat buvo pasinaudota analogišku projektu (MP3 bylų grotuvas iš SD kortelės), kurio kodas taip pat laisvai prieinamas internete [34].

Atpažintos natos į SD kortelę įrašomos populiariu MIDI formatu [35, 36]. Ši rinkmena susideda iš MIDI žinučių. Kai kurios žinutės yra vieno baido ilgio, kai kurios ilgesnės. Paprastai visose žinutėse pirmasis baitas nurodo būseną ir šešiolyktainėje skaičiavimo sistemoje gali būti 0x80–0xFF ribose. Būsenos baido 4 reikšmingiausios skiltys nurodo „įjungti“ ar „išjungti“ natą. Likusios 4 skiltys – kuris MIDI kanalas bus naudojamas. Likę žinutės baitai (t. y. duomenų baitai) bus diapazone nuo 0x00 iki 0x7F. Pirmasis duomenų baitas nurodo natos numerį nuo 0 iki 127. Antrasis duomenų baitas tokiame pačiame diapazone nurodo, koku stiprumu yra grojama nata (garso amplitudė). Pavyzdžiui, jei turėtume duomenų seką 0x90 0x3C 0x40, tai pirmasis baitas rodo, kad nata įjungta, antrasis, kad grojama C4 nata, trečiasis – natos garso amplitudė. Jei ši nata nebegrojama, tuomet reikėtų nusiųsti MIDI žinutę su tokia seka: 0x80 0x3C 0x40 arba 0x90 0x3C 0x00.

MIDI formato natų numeriai apskaičiuojami pagal sekančią formulę [36]:

$$p = 69 + 12 \cdot \log_2 \left( \frac{f}{440} \right),$$

čia:  $p$  – natos numeris MIDI formatu (nuo 0 iki 127);  $f$  – natos dažnis.

Kad viskas veiktų be klaidų, MIDI formatu užrašytos žinutės SD kortelėje yra išsaugomos tekstinėje byloje. Vėliau, įsikėlus bylą į kompiuterį, galima pakeisti jos plėtinį į *.mid* ir naudoti garso sintezatoriuje ar panašiais tikslais.

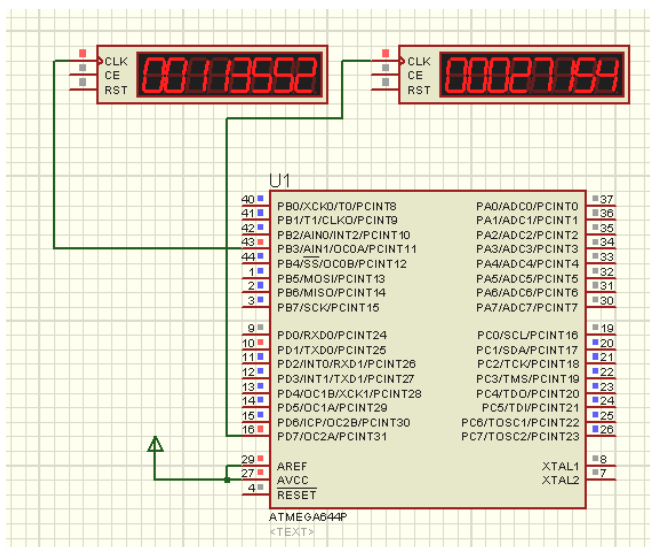
#### **5.4. Suprojektuotos sistemos modeliavimas programa *Proteus***

Prieš gaminant sistemos maketą, labai pravartu iš pradžių atlikti suprojektuotos schemas modeliavimą ir įsitikinti, kad ji veikia gerai. Tam galima panaudoti elektroninių schemų modeliavimui skirtą programą „Proteus ISIS v7.6“. Ja ypač patogu tikrinti MV programos veikseną. Taigi, iš pradžių parašę programą mikrovaldikliui įsitikinsime, kad ji veikia taip, kaip ir turėtų. Kaip parodė praktika, šis programinis paketas leidžia naudoti tik vieną instrumentą,



skirtą realiu laiku apdoroti ir atvaizduoti duomenis, pavyzdžiui: jei naudojamas virtualus oscilografas, tuomet jau nebegalima naudotis virtualiu terminalu ar dažnimačiu. Kartais pavyksta prijungti ir kelis instrumentus, bet tai priklauso nuo situacijos, be to, ir duomenys gali būti ne visai tikslūs. Todėl schemos veikimą tikrinsime dalimis: pirmiausia impulsų generatorių, tuomet SK indikatoriaus sąsają ir GFT veikimą.

Tikrinant impulsų generatorius buvo panaudotas *counter timer* (virtualus *Proteus ISIS*) instrumentas, galintis matuoti dažnį, skaičiuoti impulsus ir t. t. Kaip jau buvo aptarta anksčiau, impulsų generatoriai įgyvendinti panaudojant OC0A ir OC2A MV prievadus. Simuliuojama schema ir gauti rezultatai pateikti 5.10 pav.

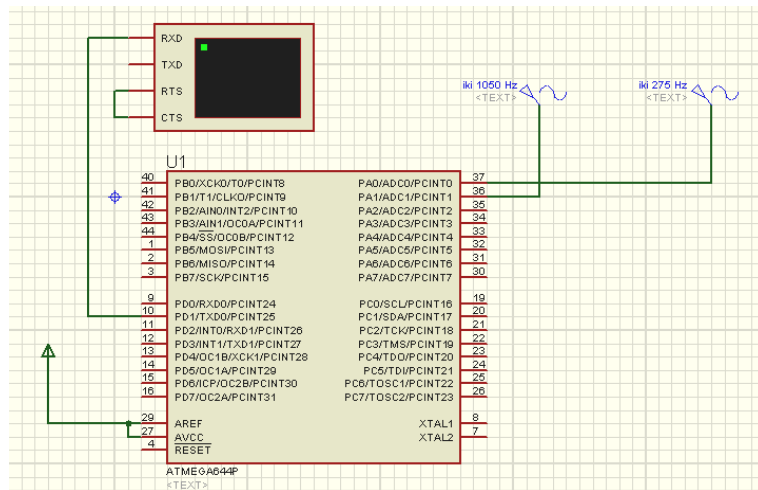


5.10 pav. Simuliuojamos schemos jungimas tikrinant impulsų generavimą

Matome, kad gauti dažniai labai mažai (praktiškai tik keliais hercais) skiriasi nuo apskaičiuotų. Tad galime daryti prielaidą, kad impulsų generatoriai filtrams veikia taip, kaip ir turėtų.

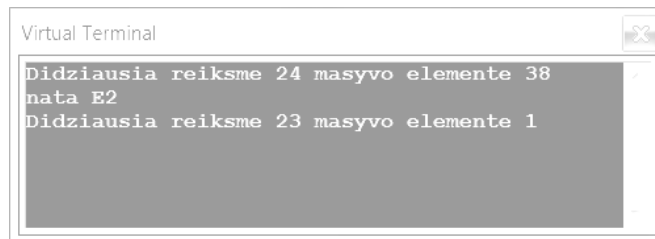
Iškilo problemų norint modeliuoti atpažintos natos išvedimą į SK ekraną. Po ilgų bandymų paaiškėjo, kad to padaryti negalima kai programos modeliuojamame mikrovaldiklyje yra nustatytas išorinis kvarcinis generatorius. Parinkus vidinį mikrovaldiklio 8 MHz taktinių impulsų šaltinį, duomenys į SK buvo išvedami be problemų. Pakeitus į tokį pat 8 MHz išorinį kvarcinį generatorių, buvo matomas tik tuščias ekranas. Taigi, galime ignoruoti šią programos klaidą ir laikyti, kad mikrovaldiklio sąsaja su SK indikatoriumi veikia gerai.

Kadangi programoje *Proteus ISIS* nepavyko duomenų išvesti į skystųjų kristalų indikatorių, tuomet testavimo tikslais buvo užprogramuotas mikrovaldiklio UART prievadas ir duomenys siunčiami per jį. Modeliuojamos schemos fragmentas prijungus virtualaus terminalo įrankį pateiktas 5.11 pav.



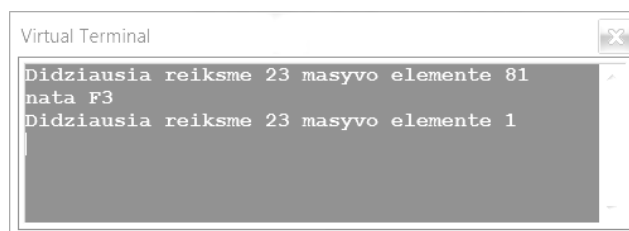
5.11 pav. Simuliuojama schema tikrinant natų atpažinimą

Iš 5.11 pav. matyti, kad abiejų ŽD filtrų signalai prijungti prie PA0 ir PA1 prievadų. Į pirmąjį ateina iki 275 Hz nufiltruotas signalas, į antrąjį iki 1 100 Hz signalas. Įsitikinti, kad sistema veikia gerai, pabandydysime atpažinti kelias pasirinktas natas. Tuo tikslu *Proteus* programoje pasirenkame sinusinio signalo generatorių ir nustatome reikiamus dažnius. Kai tai atlikome, galime spausti *Run* mygtuką, ir stebėti, kas pasirodys *virtual terminal* lange. UART sąsajos greitis buvo parinktas 19 200 bodų.

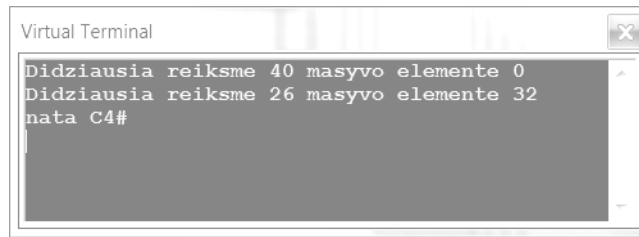


5.12 pav. *Virtual terminal* langas su gautais rezultatais

Iš 5.12 pav. matome, kad yra atvaizduojamos 3 eilutės. Viršutinioji nurodo po GFT transformacijos (žemesnio dažnio) masyve esančių reikšmių didžiausią amplitudę (šiuo atveju 24) ir masyvo vietą, kuriame yra reikšmė. Ši informacija aktuali tik testavimo tikslais. Sekanti eilutė nurodo natą. Jei nata neatpažinta, palieką ją tuščią. Taigi, E2 (82 Hz) nata buvo atpažinta tinkamai. Trečioji eilutė nurodo po GFT transformacijos (aukštesnio dažnio) masyve esančių reikšmių didžiausią amplitudę ir masyvo indeksą. Toliau pateikiami analogiškai gauti bandymų rezultatai su kitomis natomis.



5.13 pav. Atpažinta F3 nata, kurios dažnis 175 Hz

A screenshot of a 'Virtual Terminal' window. The window title is 'Virtual Terminal'. The text inside the terminal is:

```
Didziausia reiksme 40 masyvo elemente 0
Didziausia reiksme 26 masyvo elemente 32
nata C4#
```

5.14 pav. Atpažinta C4# nata, kurios dažnis 277 Hz

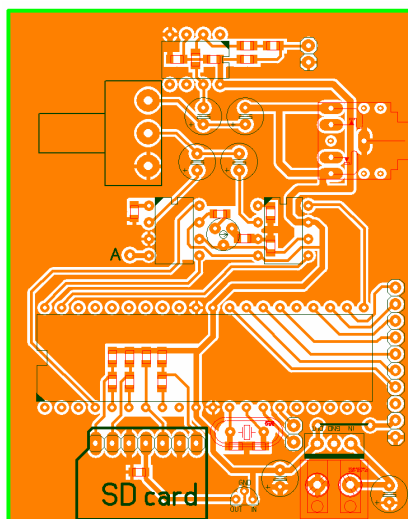
Iš gautų rezultatų (5.13–5.14 pav.) matome, kad sistema veikia ir atpažįsta natas tinkamai. Buvo ištestuoti visi 45 dažniai, kuriems yra priskirtos natos ir net įvedant juos su kelių procentų paklaida gauti tenkinantys rezultatai – tinkamai atpažintos visos natos. Tolimesni sistemos tyrimai bus atliekami su realiu maketu siunčiant duomenis per mikrovaldiklio UART sąsają į kompiuterį ir stebint juos *Tera Term* programos pagalba.

## 5.5. Spausdintinės plokštės projektavimas

Tik surinkus suprojektuotą schemą ant spausdintinio montažo plokštės bus galima atlikti galutinį jos testavimą, nes kartais yra neįvertinami parazitiniai reiškiniai. Kaip žinia, jei ant tos pačios plokštės yra analoginė ir skaitmeninė dalys, tiesiog būtina kuo geriau jas atskirti, kad nebūtų įnešami trukdžiai į analoginį signalą. Tad projektuojant plokštę pagrindinis dėmesys ir buvo teikiamas analoginės dalies (jėjimo signalo traktas) atskyrimui nuo skaitmeninės dalies. Buvo stengiamasi kuo labiau ekranuoti visus garso signalo laidininkus. Nors šiuo atveju net ir trukdžiai (žinoma, nedideli) nėra labai svarbūs, nes garso signalas niekur neišeina (kaip žinia, žmogaus ausis yra labai jautri net ir menkiems trukdžiams), o yra paverčiamas kita informacija, t. y. naudojamas GFT skaičiavimams.

Visi plokštėje esantys takeliai skirti valdantiems ir garso signalams gali būti labai ploni, nes įtaise naudojamų komponentų galios yra labai mažos. Kitų svarbių reikalavimų tenkinimas, pavyzdžiui, komponentų išdėstymas, nėra kritinis, dėl to tiesiog buvo stengiamasi padaryti spausdintinio montažo plokštę kuo kompaktiškesnę. Trasavimui buvo pasirinkta *Sprint Layout 5* programa, nes ji yra labai intuityvi. Programų su automatinio trasavimo galimybe buvo atsisakyta, nes paprastai jis tik sukelia daugiau problemų, ypač kai komponentų kiekis nėra didelis.

Iš viso buvo sutrasuotos dvi spausdintinio montažo plokščių versijos: viena – ATmega644 mikrovaldiklio paviršinio montažo SMD korpusui, kita DIP korpusui. Žinoma, pirmuoju atveju plokštė gavosi kompaktiškesnė, tačiau jei prireiktų ką nors padaryti iškiltų nemažai keblumų. Todėl bandomoji spausdintinio montažo plokštė bus gaminama DIP mikrovaldiklio korpuso variantui (5.15 pav.). Jos matmenys yra 61 × 78 mm.



**5.15 pav.** *Sprint Layout 5* programa suprojektuota montažinė plokštė mikrovaldikliui su DIP korpusu

Visi pasyvūs elementai buvo parinkti kuo mažesni ir kur buvo įmanoma – paviršinio montažo. Skystųjų kristalų indikatorių prie maketinės plokštės prijungti numatoma lanksčia jungtimi. Tokiu būdu nėra prisirišama prie konkretaus skystųjų kristalų indikatoriaus ir galima juos lengvai pakeisti.

## 5.6. Skyriaus išvados

Šiame skyriuje išsamiai aptarėme gitaros garsų atpažinimo sistemos projektavimo eigą. Pirmiausiai buvo sudaryta struktūrinė sistemos schema. Vėliau, parenkant konkrečius elementus, buvo sudaryta ir principinė elektrinė schema. Jos širdis – mikrovaldiklis, kuris parinktas atsižvelgiant į jam numatomas vykdyti užduotis.

Atlikus principinės elektrinės schemos projektavimą, buvo parašyta programa mikrovaldikliui C kalboje. Programuojant apskaičiuoti laikmačių–skaitliukų intervalai, nustatyti išdavai su periferiniais įrenginiais, įgyvendinta FAT32 failų sistema bendravimui su SD tipo atminties kortele. Aptartas MIDI žinučių formatas ir įgyvendintas jų siuntimas į šią kortelę. Taip pat įgyvendinti greitosios Furjė transformacijos algoritmas ir pasiūlyta natų skaičiavimo metodika.

Suprojektuota schema buvo modeliujama programos *Proteus ISIS* pagalba. Patikrintas atskirų schemos mazgų funkcionavimas. Nustatyta, kad mikrovaldiklio laikmačio–skaitliuko generuojami dažniai tinkami ŽDF pralaidos juostai nustatyti. Taip pat buvo atliktas garso signalų atpažinimo modeliavimas virtualiu generatoriumi parenkant dažnius, atitinkančius teorinius natų dažnius. Gauti rezultatai buvo išvedami į *Proteus ISIS* programos virtualų terminalą. Visi jie buvo teigiami, iš ko galima spręsti, kad GFT algoritmas veikia gerai.

Galiausiai buvo suprojektuota viensusė 61 × 78 mm dydžio spausdintinio montažo plokštė.

## 6. GITAROS GARSŲ ATPAŽINIMO SISTEMOS PATIKRA

Atlikus gitaros garsų atpažinimo sistemos projektavimo darbus bei pagaminus bandomąjį maketą, reikia įsitikinti, kad jis veikia būtent taip, kaip ir turėtų. Šiam tikslui pasiekti sudarysime eksperimentų su bandomuoju maketu planą. Trumpai aptarsime su juo atliktų eksperimentų eigą, pobūdį bei gautus rezultatus.

### 6.1. Eksperimentų planas

Tam, kad eksperimentinis maketo tyrimas vyktų nuosekliai ir kryptingai, pirmiausia reikia susidaryti eksperimentų planą, bei išaiškinti kas, kaip ir koku tikslu bus tiriama. Taigi, planuojamų eksperimentų ir tyrimų eiga:

- išbandyti bei išsiaiškinti, ar maketas apskritai veikia, patikrinti pagrindinių jo mazgų funkcionavimą, t. y. ar tinkamai atpažįsta gitaros garsus (natas), atvaizduoja jas skystųjų kristalų ekrane, bendrauja su kompiuteriu per nuosekliają UART sąsają, ar įrašo duomenis į SD (angl. *Secure Digital*) atminties kortelę.
- Naudojant dažnių generatorių patikrinti, kokiame dažnių diapazone atpažįstamos natos, kitaip tariant, išsiaiškinti galimas nuokrypas nuo centrinių natų dažnių.
- Eksperimentiškai ištirti, koks laiko intervalas reikalingas vienai ar kitai naitai atpažinti, įvertinti, kaip greitai galima keisti natas. Įvertinti ar sistema gali dirbti realiuoju laiku.
- Atlikti maketo tyrimus ir įvertinti, ar galimas tinkamas dviejų, trijų ir t. t. šalia viena kitos esančių natų atpažinimas.
- Atlikti maketo tyrimus ir įvertinti, ar galimas tinkamas kelių per oktavą viena nuo kitos nutolusių natų atpažinimas.
- Įvertinti suprojektuotos gitaros garsų atpažinimo sistemos jautrumą („taiklumą“) atpažįstant natas.

Naudojant garso įrašus, bus tirtas sistemos gebėjimas atpažinti kelias natas vienu metu. Jautrumas ir tikslumas [38] yra statistiniai rodikliai, naudojami eksperimentų rezultatams įvertinti. Jautrumas (angl. *sensitivity*) nurodo faktinių teigiamų įvykių, kurie yra identifikuoti eksperimentų metu procentinę išraišką. Tikslumas (angl. *specificity*) nurodo faktinių neigiamų įvykių procentinę išraišką. Šiuo atveju mus domina teigiamų įvykių procentas (jautrumas). Šios dvi sąvokos įvairiuose literatūros šaltiniuose apibrėžiamos šiek tiek skirtingai ir gali kelti nemažai painiavos. Tad jautrumą galima apibūdinti kaip sistemos „taiklumą“ atpažįstant natas.

Atliekamų eksperimentų metu rezultatai gali būti teigiami (natos atpažintos teisingai) bei neigiami (natos atpažintos klaidingai). Tokiu atveju visi galimi eksperimento rezultatai yra:

teisingai teigiami (angl. *true positive*) – nata atpažinta teisingai; klaidingai teigiamas (angl. *false positive*) – nata atpažinta klaidingai (pvz., kartotinė harmonika atpažinta kaip nata); teisingai neigiamas (angl. *true negative*) – atpažinta nata buvo teisingai atmesta (pvz., ji buvo kartotinė harmonika) ir klaidingai neigiamas (angl. *false negative*) – nata atmesta klaidingai (pvz., dėl per mažos amplitudės). Statistiniam gitaros garsų atpažinimo sistemos įvertinimui bus naudojamos šios formulės:

$$J = \frac{T_T}{T_T + K_N},$$

čia:  $J$  – sistemos jautrumas arba „taiklumas“;  $T_T$  – teisingai teigiamų rezultatų skaičius,  $K_N$  – klaidingai neigiamų rezultatų skaičius.

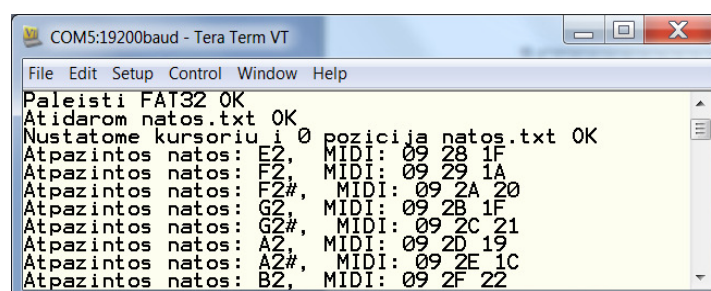
Prognozuojama teigiama vertė (PTV) atspindi teigiamų eksperimento rezultatų tikimybę, kad jie yra teisingi ir išreiškiami formule:  $PTV = T_T / T_T + K_T$ , kur  $K_T$  yra klaidingai teigiamų įvykių skaičius.

## 6.2. Eksperimentų rezultatai

Atlikus aibę įvairių eksperimentų laboratorinėmis bei realiomis sąlygomis su gitaros garsų atpažinimo sistemos bandomuoju maketu, pateiksime jo charakteristikas ir gautus tyrimo rezultatus.

### 6.2.1. Pagrindinių sistemos maketo mazgų patikra

Pirmiausia buvo patikrinta, ar maketas atvaizduoja duomenis skystųjų kristalų ekrane ir bendrauja su kompiuteriu per nuosekliąją UART sąsają. Tam prie jo linijinio įėjimo buvo prijungtas kompiuterio garso plokštės išėjimas. Sąsajai su kompiuteriu sudaryti buvo panaudotas paprastas DCA-540 telefono duomenų kabelis, kuris atlieka UART–USB signalų vertiklio funkciją. Gautiems duomenims atvaizduoti kompiuteryje pasirinkta *Tera Term* programa. Tuomet nustatius maksimalų kompiuterio garso lygį, paeiliui buvo ištestuoti visi 45 skirtingų natų įrašai. Buvo įsitikinta, kad sistema veikia: pavienes natas atpažįsta teisingai bei atvaizduoja jas SK ekrane ir tinkamai išveda duomenis į UART sąsają. *Tera term* programos langas su gautais bandymų rezultatais pateiktas 6.1 pav.



```

COM5:19200baud - Tera Term VT
File Edit Setup Control Window Help
Paleisti FAT32 OK
Atidarom natos.txt OK
Nustatome kursoriu i 0 pozicija natos.txt OK
Atpazintos natos: E2, MIDI: 09 28 1F
Atpazintos natos: F2, MIDI: 09 29 1A
Atpazintos natos: F2#, MIDI: 09 2A 20
Atpazintos natos: G2, MIDI: 09 2B 1F
Atpazintos natos: G2#, MIDI: 09 2C 21
Atpazintos natos: A2, MIDI: 09 2D 19
Atpazintos natos: A2#, MIDI: 09 2E 1C
Atpazintos natos: B2, MIDI: 09 2F 22
  
```

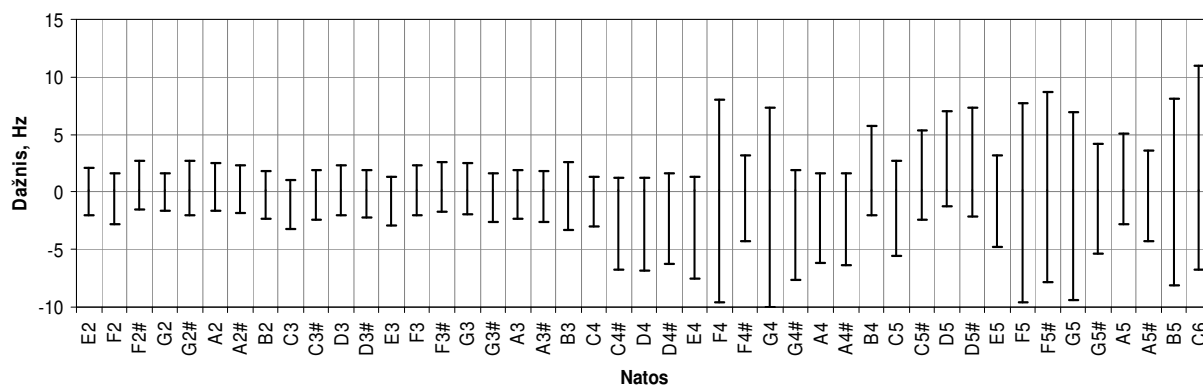
6.1 pav. Programos *Tera Term* langas su maketo bandymo rezultatais

Kaip matyti iš 6.1 pav., pirmiausia yra tikrinama ar įdėta SD atminties kortelė. Jei taip, tuomet aktyvuojama FAT32 failų sistema, kortelėje sukuriama nauja byla *natos.txt* ir joje kursorius nustatomas į pradinę padėtį. Duomenys į SD atminties kortelę taip pat yra korektiškai įrašomi tekstinėje byloje pagal MIDI specifikacijoje pateiktus reikalavimus. Verta paminėti, kad MIDI žinutės turėtų būti siunčiamos pastoviai nustatytu greičiu. Kol kas į SD atminties kortelę yra įrašomos atpažintos natos atsiradimo, jos numerio ir garso intensyvumo žinutės. Ateityje numatoma papildomai įrašyti informaciją ir apie natos trukmę, todėl bus galima tiksliau atkurti muzikinį kūrinį.

Vėliau buvo atliekama mikrofono schemos patikra. Šią schemą reikėtų koreguoti, nes nepakankama garsinio signalo amplitudė GFT transformacijai atlikti. Schemoje buvo numatytas potenciometras (žr. 5.4 pav.), kurio pagalba galima keisti mikrofoninio stiprintuvo stiprinimo koeficientą. Tačiau net ir nustačius didžiausią stiprinimą, signalo lygis yra per mažas, nes maketą reikia laikyti labai arti garso šaltinio (gitaros ar garsiakalbio), kad sistema reaguotų į šiuos garsus. Mikrofono signalo taip pat negalima stiprinti be galo daug kartų, nes prasideda stiprintuvo susižadinimas. Visa tai galima būtų ištaisyti šiek tiek pakoregavus schemą – pridėjus dar vieną papildomą operacinį stiprintuvą, tačiau tam reikėtų perdaryti montažinę plokštę. Tad šią schemos dalį teks tobulinti ateityje, o maketo tyrimams naudosime lininį įėjimą, nes jis veikia puikiai tiek prijungtas prie kompiuterio garso plokštės, tiek ir prie elektroakustinės gitaros.

### 6.2.2. Tiriamojo maketo įeinančių gitaros garsų dažnių intervalų nustatymas

Sekantis tyrimo etapas – nustatyti galimas įeinančių gitaros garsų (natų) dažnių nuokrypas nuo etaloninių dažnių, kuomet jos dar yra tinkamai atpažįstamos. Tam buvo panaudotas funkcinis signalų generatorius *Grundig FG100* [37]. Generatorius buvo prijungtas prie maketo linijinio įėjimo, parinkta 1 V sinusinės amplitudės vertė, dažnis buvo keičiamas 0,1 Hz žingsniu stebint, kokiame dažnių intervale natos yra atpažįstamos. Gauti rezultatai užfiksuoti ir pateikti 6.2 pav.



6.2 pav. Matavimų metu gautos galimos nuokrypos nuo natų centrinių dažnių, kuomet nata dar atpažįstama teisingai

Pateiktame 6.2 pav. nulis ordinačių ašyje atitinka teoriškai apskaičiuotus natų dažnius, o išvesti statmenys („ūsai“) atitinka nuokrypas nuo centrinių dažnių. Kai kurių natų nuokrypiai didesni. Taip yra dėl to, kad centriniai natų dažniai atsidūrė tarp dviejų GFT transformacijos spektro dedamųjų. Kad išvengti klaidingo natų interpretavimo, teko jas priskirti abejoms dedamosioms.

### **6.2.3. Gitaros natų atpažinimui reikalingo laiko sąnaudų skaičiavimas**

Kai kurie muzikiniai kūriniai yra lėti, kiti greitesni. Tai priklauso ir nuo to, kokio stiliaus kūrinys atliekamas. Dėl šios priežasties tampa aktualu gitaros natas atpažinti kaip galima greičiau, kad vieni garsai nespėtų perdengti kitų ir taip būtų išvengiama klaidingo natų interpretavimo. Tad šiame skyrelyje aptarsime bandymų rezultatus naudojant skirtingo ilgio gitaros garsus.

Norint iširti koks laiko intervalas reikalingas natoms atpažinti, pirmiausia reikia įvertinti visų suprojektuotos sistemos naudojamų operacijų (signalų diskretizavimas, GFT skaičiavimas, rezultatų išvedimas į periferinius įrenginius ir pan.) laiką. Galima nesunkiai paskaičiuoti, jog esant diskretizavimo dažniams 2,2 kHz ir 550 Hz, vien užpildyti du duomenų masyvus iki 256 atskaitų reikia atitinkamai 116 ms ir 465 ms. GFT skaičiavimas užtrunka dar papildomas 25,44 ms (prie 20 MHz taktinio dažnio). Tad bendras laikas, reikalingas natoms atpažinti, yra apie 606 ms. Praktiškai gaunami šiek tiek kitokie rezultatai. Tarkime, kad garsinis signalas prasideda tuo pat metu, kai pradedamas pildyti pirmasis GFT duomenų masyvas, ir nutrūksta nebaigus jo užpildyti. Tuomet masyvas iki galo užpildomas nuliais (angl. *Zero padding*). Tai nekeičia GFT rezultato. Tokiu būdu nata gali būti sėkmingai atpažinta iš trumpo įrašo (buvo bandyti 100 ms trukmės įrašai, kurių trukmė buvo apribota *MATLAB* programos pagalba). Tačiau net ir naudojant tą patį įrašą, ne visada pavyksta identifikuoti natą. Tai labai paprastai paaiškinama: jei konkrečiu laiko momentu vykdoma garso signalų diskretizacija žemesniems dažniams (iki 275 Hz), o šis signalas yra aukštesnio dažnio, tuomet nata bus neatpažinta. Polifoniniams garsams, ypač jei jie yra iš skirtingų dažnių ruožo, pavyzdžiui, vienos natos tono aukštis mažesnis, o kito didesnis nei 275 Hz, reikalingas ilgesnis garso intervalas, norint sėkmingai atpažinti gitaros natas.

Jei trumpi (100–300 ms) gitaros garsų įrašai leidžiami iš karto vienas po kito, tai sistema interpretuoja kaip vieną akordą (kelias užgautas stygas vienu metu). Iš esmės tai nėra labai blogai, nes natos atpažįstamos teisingai, tačiau iš tikro jos grojamos skirtingais laiko momentais. Vadinasi užrašius tai MIDI žinučių pavidalu, gausime nekorektiškus rezultatus. Tyrimo tikslais buvo grojami visi 45 skirtingų natų ir skirtingos trukmės įrašai vienas po kito. Gauti rezultatai pateikti 6.1 lentelėje.



**6.1 lentelė.** Natų sėkmingo atpažinimo priklausomybė nuo įrašo trukmės

Vienos natos įrašo trukmė, ms	Natų skaičius, vnt	Atpažintos monotoninės natos, vnt
100	45	0
200	45	2
300	45	6
400	45	15
500	45	31
600	45	45

6.1 lentelėje stulpelis „Atpažintos monotoninės natos“ reiškia, kad grojant tam tikros trukmės gitaros garsų (natų) įrašus vieną po kito, šie visais atvejais yra atpažįstami kaip atskira nata, o ne kaip kelių samplaika.

Apibendrinant skyrelio rezultatus galime teigti, kad tinkamai natos yra atpažįstamos, kai jų įrašo trukmė yra ne mažesnė nei 600 ms. Tai šiek tiek per ilgą laiko tarpą, norint natos atpažinti realiuoju laiku, nebent grojamas kūrinys būtų lėtas. Todėl reikėtų imtis priemonių mažinti signalo diskretizavimo trukmei, nes būtent ši operacija užima daugiausiai laiko ir jos metu mikroprocesorius praktiškai nėra apkrautas įvairiais skaičiavimais.

#### **6.2.4. Gitaros natų atpažinimo sistemos statistinis tikslumo įvertinimas**

Tyrimui buvo panaudotas sudarytas gitaros garsynas. Tyrimo metu garso signalai iš kompiuterio buvo siunčiami į maketą linijiniu laidu įvairiomis sekomis. Pirmiausia buvo bandoma atpažinti dvi gretimas natos, nes jų dažniai skiriasi mažiausiai, todėl tai padaryti turėtų būti sunkiausia. Monotonines natos, kaip jau buvo aptarta anksčiau, sistema atpažino visais atvejais. Pradėta buvo nuo žemiausiųjų natų, t. y. E2 ir F2, F2 ir F2#, F2# ir G2, ir t. t. Vėliau toks pat eksperimentas pakartotas su trimis, keturiomis, penkiomis ir šešiomis natomis vienu metu. Jei visos natos atpažintos teisingai, tuomet įvykis teisingai teigiamas ( $T_T$ ), jei bent viena nata atpažinta neteisingai (ne to dažnio ar pan.) – klaidingai teigiamas ( $K_T$ ), o jei iš viso neatpažinta – klaidingai neigiamas ( $K_N$ ). Gauti rezultatai pateikti 6.2 lentelėje.

**6.2 lentelė.** 1-6 vienu metu grojamų natų tyrimo rezultatai

Grojamų natų skaičius	Skirtingų bandymų skaičius	Teisingai teigiamų ( $T_T$ ) rezultatų skaičius	Klaidingai teigiamų ( $K_T$ ) rezultatų skaičius	Klaidingai neigiamų ( $K_N$ ) rezultatų skaičius
1	45	45	0	0

**6.2 lentelė. Tęsinys**

2	44	41	0	3
3	43	38	1	4
4	42	35	2	5
5	41	35	1	5
6	40	31	3	6

Kitas svarbus tyrimo etapas – patikrinti kaip yra atpažįstamos gitaros natos, kurios nutolusios viena nuo kitos per oktavą, pavyzdžiui, E2 ir E3, A3 ir A4 ir t. t. Taip galima patikrinti, ar tinkamai panaikinamos harmonikos ir ar jos klaidingai neatpažįstamos kaip natos. Tokių eksperimentų rezultatai pateikti 6.3 lentelėje.

**6.3 lentelė. Dviejų per oktavą nutolusių natų tyrimo rezultatai**

Grojamų natų skaičius	Skirtingų bandymų skaičius	Teisingai teigiamų ( $T_T$ ) rezultatų skaičius	Klaidingai teigiamų ( $K_T$ ) rezultatų skaičius	Klaidingai neigiamų ( $K_N$ ) rezultatų skaičius
2	36	32	0	4

Siekiant nustatyti įgyvendintos gitaros garsų atpažinimo sistemos veikimo „taiklumą“, panaudotos ankstesniame poskyryje aprašytos jautrumo įvertinimo formulės. Iš viso buvo atlikta 291 skirtingų bandymų. Juos išanalizavus gauti rezultatai pateikti 6.4 lentelėje.

**6.4 lentelė. Sistemos jautrumo įvertinimo tyrimo rezultatai**

Rodiklis	Rezultatas, vnt.
Teisingai teigiamas ( $T_T$ )	257
Klaidingai teigiamas ( $K_T$ )	7
Klaidingai neigiamas ( $K_N$ )	27

Pagal surinktus statistinius duomenis apskaičiuojame suprojektuotos sistemos jautrumą („taiklumą“) bei prognozuojamą teigiamą vertę ( $PTV$ ) atpažįstant gitaros garsus:

$$J = \frac{T_T}{T_T + K_N} = \frac{257}{257 + 27} \cdot 100 \% = 90,49 \%;$$

$$PTV = \frac{T_T}{T_T + K_T} = \frac{257}{257 + 7} \cdot 100 \% = 97,35 \%.$$

Gauti rezultatai rodo, kad suprojektuota gitaros garsų atpažinimo sistema geba atpažinti 90,49 % natų su 97,35 % tikimybe, jog šios natos yra atpažintos teisingai.

### **6.3. Skyriaus išvados**

Šiame skyriuje buvo atartas maketo eksperimentinis tyrimas. Kad jis būtų atliekamas nuosekliai, tam buvo sudarytas eksperimentų planas. Pateikta trumpa suprojektuotos gitaros garsų atpažinimo sistemos statistinio vertinimo metodologija.

Atlikus pirminę maketo patikrą, buvo įsitikinta, kad jis veikia taip, kaip ir turėtų. Duomenys yra tinkamai išvedami į periferinius įrenginius, taip pat įrašomi SD tipo atminties kortelėje.

Nustatyti dažnių intervalai, kurių ribose natos yra atpažįstamos teisingai. Kitaip tariant, įvertintos galimos nuokrypos nuo centrinių natų dažnių.

Nustatytas minimalus laiko tarpas, reikalingas visiems skaičiavimams atlikti ir natoms atpažinti. Visos natos buvo atpažintos be klaidų, kai įrašų trukmė ne mažesnė nei 600 ms. To pakanka gitara grojant lėtesnius kūrinčius, bet greitesniu tempu besikeičiančias natas sistema gali atpažinti netinkamai arba net visai praleisti.

Taip pat buvo atliktas gitaros garsų atpažinimo sistemos statistinis vertinimas. Atlikus eksperimentus su 291 skirtingais gitaros garsais nustatyta, kad sistema geba tinkamai atpažinti 90,49 % natų su 97,35 % tikimybe, jog šios natos yra atpažintos teisingai.

## IŠVADOS

Ištirta gitaros garsų atpažinimo realiuoju laiku galimybė sukuriant sistemą, atpažįstančią šiuos garsus.

Sudarytas originalaus gitaros garsynas. Jis susideda iš 450 unikalių įrašų (po 10 kiekvienos natos). Garsynas patalpintas viešai prieinamame serveryje [20].

Atliktas parinktų algoritmų gitaros garsams atpažinti tyrimas programa *MATLAB*. Atsižvelgiant į jų rezultatus, pasiūlyta gitaros natų atpažinimo metodika taikant greitąją Furjė transformaciją bei papildomas sąlygas signalo spektro dedamosioms.

Gitaros garsų atpažinimo sistema įgyvendinta naudojant 8 bitų AVR šeimos mikrovaldiklį ATmega644. Ją projektuojant buvo:

1. Sudarytos projektuojamos sistemos struktūrinė ir principinė schemas;
2. Parašyta programa mikrovaldikliui C programavimo kalba;
3. Atliktas virtualus sistemos modeliavimas programa *Proteus ISIS*;
4. Suprojektuota spausdintinio montažo plokštė.

Atlikus eksperimentinius pagaminto gitaros garsų atpažinimo sistemos maketo tyrimus nustatyta:

1. Pagrindiniai suprojektuoto maketo mazgai veikia tinkamai atlikdami visas jiems numatytas funkcijas;
2. Natoms tinkamai atpažinti reikalingas ne mažesnis nei 600 ms trukmės garso signalas;
3. Sistema geba tinkamai atpažinti 90,49 % natų su 97,35 % tikimybe, jog šios natos yra atpažintos teisingai.

## LITERATŪROS SĄRAŠAS

1. KAMARAUSKAS, J. 2009. *Asmens atpažinimas pagal balsą: daktaro disertacija* [interaktyvus]. Vilnius: Technika [žiūrėta 2012 m. gegužės 15 d.]. Prieiga per internetą: <[www.mii.lt/files/mii\\_dis\\_09\\_kamarauskas.pdf](http://www.mii.lt/files/mii_dis_09_kamarauskas.pdf)>
2. ALM, J. F.; WALKER, J. S. 2002. Time-Frequency Analysis of Musical Instruments, *Society for Industrial and Applied Mathematics* [interaktyvus]. 44(3): 457–476. [žiūrėta 2012 m. gegužės 15 d.]. Prieiga per internetą: <[www.uwec.edu/walkerjs/media/38228\[1\].pdf](http://www.uwec.edu/walkerjs/media/38228[1].pdf)>
3. FITCH, J.; SHABANA, W. 1999. *A wavelet-based pitch detector for musical signals* [interaktyvus]. Department of Mathematical Sciences, University of Bath. [žiūrėta 2012 m. gegužės 15 d.]. Prieiga per internetą: <<http://opus.bath.ac.uk/16465/>>
4. YEN, J. 2009. *Wavelet for Acoustics* [interaktyvus]. [žiūrėta 2012 m. gegužės 15 d.]. Prieiga per internetą: <[djj.ee.ntu.edu.tw/Wavelet\\_Acoustics.pdf](http://djj.ee.ntu.edu.tw/Wavelet_Acoustics.pdf)>
5. MCLEOD, P. G. 2008. *Fast, Accurate Pitch Detection Tools for Music Analysis* [interaktyvus]. University of Otago, Dunedin, New Zealand [žiūrėta 2012 m. gegužės 15 d.]. Prieiga per internetą: <[http://miracle.otago.ac.nz/tartini/papers/Philip\\_McLeod\\_PhD.pdf](http://miracle.otago.ac.nz/tartini/papers/Philip_McLeod_PhD.pdf)>
6. KNESEBECK, A.; ZÖLZER, U. 2010. *Comaprision ogf pitch trackers for real-time guitar effects* [interaktyvus]. Dept. of Signal Processing and Communications, Helmut Schmidt University, Hamburg, Germany. [žiūrėta 2012 m. gegužės 15 d.]. Prieiga per internetą: <[http://opus.unibw-hamburg.de/frontdoor.php?source\\_opus=2795](http://opus.unibw-hamburg.de/frontdoor.php?source_opus=2795)>
7. MIDDLETON, G. 2003. *Pitch Detection Algorithms* [interaktyvus]. Connexions [žiūrėta 2012 m. gegužės 15 d.]. Prieiga per internetą: <<http://cnx.org/content/m11714/1.2/>>
8. GAINZA M.; LAWLOR, R.; COYLE, E. 2005. *Multi Pitch Estimation by using IIR Comb Filters* [interaktyvus]. International Symposium focused on Multimedia Systems and Applications (ELMAR), Zadar. [žiūrėta 2012 m. gegužės 15 d.]. Prieiga per internetą: <<http://arrow.dit.ie/argcon/19/>>
9. CAMACHO, A. 2007. *A sawtooth waveform inspired pitch estimator for speech and music: doctor's dissertation* [interaktyvus]. University of Florida. [žiūrėta 2012 m. gegužės 15 d.]. Prieiga per internetą: <<http://www.scribd.com/doc/62950196/24/Harmonic-Product-Spectrum-HPS>>
10. COWLING, M.; SITTE, R. 2002. Analysis of Speech Recognition Techniques for use in a Non-Speech Sound Recognition System [interaktyvus]. *The Kluwer International Series in Engineering and Computer Science* 703: 31–46 [žiūrėta 2012 m. gegužės 21 d.]. Prieiga per internetą: <<http://www.springerlink.com/content/g742814w731866v7/>>
11. *MT-30 Chromatic Metro-Tuner* [interaktyvus]. Musedo Technology Inc. [žiūrėta 2012 m. gegužės 21 d.]. Prieiga per internetą: <<http://store.musicbasics.com/t-mt-30.html>>

12. *Note Detector 1.0.2.9* [interaktyvus]. 2012. Programos namų puslapis [žiūrėta 2012 m. gegužės 15 d.]. Prieiga per internetą: <<http://store.musicbasics.com/t-mt-30.html>>
13. *Identiflyer Ultra* [interaktyvus]. 2012. Nature-Watch [žiūrėta 2012 m. gegužės 15 d.]. Prieiga per internetą: <<http://www.nature-watch.com/identiflyer-ultra-p-1067.html>>
14. GELLING, D. 2010. *Bird Song Recognition using GMMs and HMMs* [interaktyvus]. Department of Computer Science, University of Sheffield. [žiūrėta 2012 m. gegužės 21 d.]. Prieiga per internetą: <[http://www.dcs.shef.ac.uk/intranet/teaching/projects/archive/msc0910/pdf/DGelling\\_dissertation.pdf](http://www.dcs.shef.ac.uk/intranet/teaching/projects/archive/msc0910/pdf/DGelling_dissertation.pdf)>
15. *Digital Bird Song Identification a Reality?* [interaktyvus]. 2011. American Birding Association [žiūrėta 2012 m. gegužės 21 d.]. Prieiga per internetą: <<http://blog.aba.org/2011/10/digital-bird-song-identification-a-reality.html>>
16. *DMC725S - Dreadnaught Acoustic-Electric Cutaway* [interaktyvus]. 2012. Hohner Inc. [žiūrėta 2012 m. gegužės 21 d.]. Prieiga per internetą: <<http://www.hohnerusa.com/index.php?295>>
17. *FR-2LE: CF Field Recorder* [interaktyvus]. 2012. [Foster Electric Co., Ltd](http://www.fostexinternational.com/) [žiūrėta 2012 m. gegužės 21 d.]. Prieiga per internetą: <<http://www.fostexinternational.com/docs/products/FR-2LE.shtml>>
18. VAITKIENĖ, D. 2002. *Muzikos teorijos pagrindai: konspektas*. Alytus : AB "Alytaus spaustuvė". 45 p.
19. Programos *Gold Wave* namų puslapis [interaktyvus]. 2012. GoldWave® Inc [žiūrėta 2012 m. gegužės 21 d.]. Prieiga per internetą: <<http://www.goldwave.com/>>
20. Gitaros natų įrašai [interaktyvus]. 2012. Arūnas Survila [žiūrėta 2012 m. gegužės 31 d.]. Prieiga per internetą: <[http://www.failai.lt/6hva9522v4km/Gitaros\\_natos.rar.htm](http://www.failai.lt/6hva9522v4km/Gitaros_natos.rar.htm)>
21. Programos *MATLAB* namų puslapis [interaktyvus]. 2012. The MathWorks, Inc. [žiūrėta 2012 m. gegužės 23 d.]. Prieiga per internetą: <<http://www.mathworks.se/products/matlab/>>
22. NAVAKAUSKAS, D; SERACKIS, A. 2008. *Skaitmeninis signalų apdorojimas taikant MATLAB* : vadovėlis aukštųjų mokyklų studentams. Vilnius : Technika. 381 p.
23. *Characteristics of Different Smoothing Windows* [interaktyvus]. 2008. National Instruments Corporation [žiūrėta 2012 m. gegužės 23 d.]. Prieiga per internetą: <[http://zone.ni.com/reference/en-XX/help/371361E-01/lvanlsconcepts/char\\_smoothing\\_windows/](http://zone.ni.com/reference/en-XX/help/371361E-01/lvanlsconcepts/char_smoothing_windows/)>
24. *Finite impulse response* [interaktyvus]. 2012. Wikipedia [žiūrėta 2012 m. gegužės 23 d.]. Prieiga per internetą: <[http://en.wikipedia.org/wiki/Finite\\_impulse\\_response](http://en.wikipedia.org/wiki/Finite_impulse_response)>
25. *Project: Pitch Detection* [interaktyvus]. 2008. Naotoshi Seo [žiūrėta 2012 m. gegužės 23 d.]. Prieiga per internetą: <<http://note.sonots.com/SciSoftware/Pitch.html>>
26. *741 Electret Pre-Amp* [interaktyvus]. 2012. All about circuits [žiūrėta 2012 m. gegužės 23 d.]. Prieiga per internetą: <<http://forum.allaboutcircuits.com/showthread.php?t=18395>>

27. *5th-Order, Lowpass, Elliptic, Switched-Capacitor Filters* [interaktyvus]. 2008. Maxim Integrated Products [žiūrėta 2012 m. gegužės 23 d.]. Prieiga per internetą: <<http://www.maxim-integrated.com/datasheet/index.mvp/id/2237>>
28. *Atmega644 mikrovaldiklio aprašymas* [interaktyvus]. 2012. Atmel Corporation [žiūrėta 2012 m. gegužės 23 d.]. Prieiga per internetą: <[http://www.atmel.com/dyn/products/product\\_card.asp?part\\_id=3694](http://www.atmel.com/dyn/products/product_card.asp?part_id=3694)>
29. *AVR Studio 5* [interaktyvus]. 2012. Atmel Corporation [žiūrėta 2012 m. gegužės 23 d.]. Prieiga per internetą: <[http://www.atmel.com/dyn/products/product\\_card.asp?part\\_id=3694](http://www.atmel.com/dyn/products/product_card.asp?part_id=3694)>
30. *WinAVR* [interaktyvus]. WebRing Inc. [žiūrėta 2012 m. gegužės 23 d.]. Prieiga per internetą: <<http://winavr.sourceforge.net/index.html>>
31. *Audio Spectrum Monitor* [interaktyvus]. 2005. Electronic Lives Manufacturing – presented by ChaN [žiūrėta 2012 m. gegužės 23 d.]. Prieiga per internetą: <[http://elm-chan.org/works/akilcd/report\\_e.html](http://elm-chan.org/works/akilcd/report_e.html)>
32. *AVR-GCC Source Examples* [interaktyvus]. 2006. Peter Fleury [žiūrėta 2012 m. gegužės 15 d.]. Prieiga per internetą: <<http://homepage.hispeed.ch/peterfleury/avr-software.html#libs>>
33. *Petit FAT File System Module* [interaktyvus]. 2010. Electronic Lives Manufacturing – presented by ChaN [žiūrėta 2012 m. gegužės 23 d.]. Prieiga per internetą: <[http://elm-chan.org/fsw/ff/00index\\_p.html](http://elm-chan.org/fsw/ff/00index_p.html)>
34. *YAMPP. Yet another MP3 player* [interaktyvus]. 2004. Jesper Hansen [žiūrėta 2012 m. gegužės 23 d.]. Prieiga per internetą: <<http://www.myplace.nu/mp3/>>
35. *The MIDI Specification* [interaktyvus]. Steve Richardson [žiūrėta 2012 m. gegužės 23 d.]. Prieiga per internetą: <<http://www.gweep.net/~prefect/eng/reference/protocol/midispec.html>>
36. *MIDI* [interaktyvus]. 2012. Wikipedia [žiūrėta 2012 m. gegužės 23 d.]. Prieiga per internetą: <<http://en.wikipedia.org/wiki/MIDI>>
37. *Function generator FG100. Technical datasheet* [interaktyvus]. Grundig [žiūrėta 2012 m. gegužės 23 d.]. Prieiga per internetą: <<http://www.farnell.com/datasheets/36083.pdf>>
38. *Type I and type II errors* [interaktyvus]. 2012. Wikipedia [žiūrėta 2012 m. gegužės 26 d.]. Prieiga per internetą: <[http://en.wikipedia.org/wiki/True\\_positive](http://en.wikipedia.org/wiki/True_positive)>

## **PRIEDAI**

**A priedas. Dalyvaujant XV -toje Lietuvos jaunųjų mokslininkų konferencijoje gautos pažymos kopija**

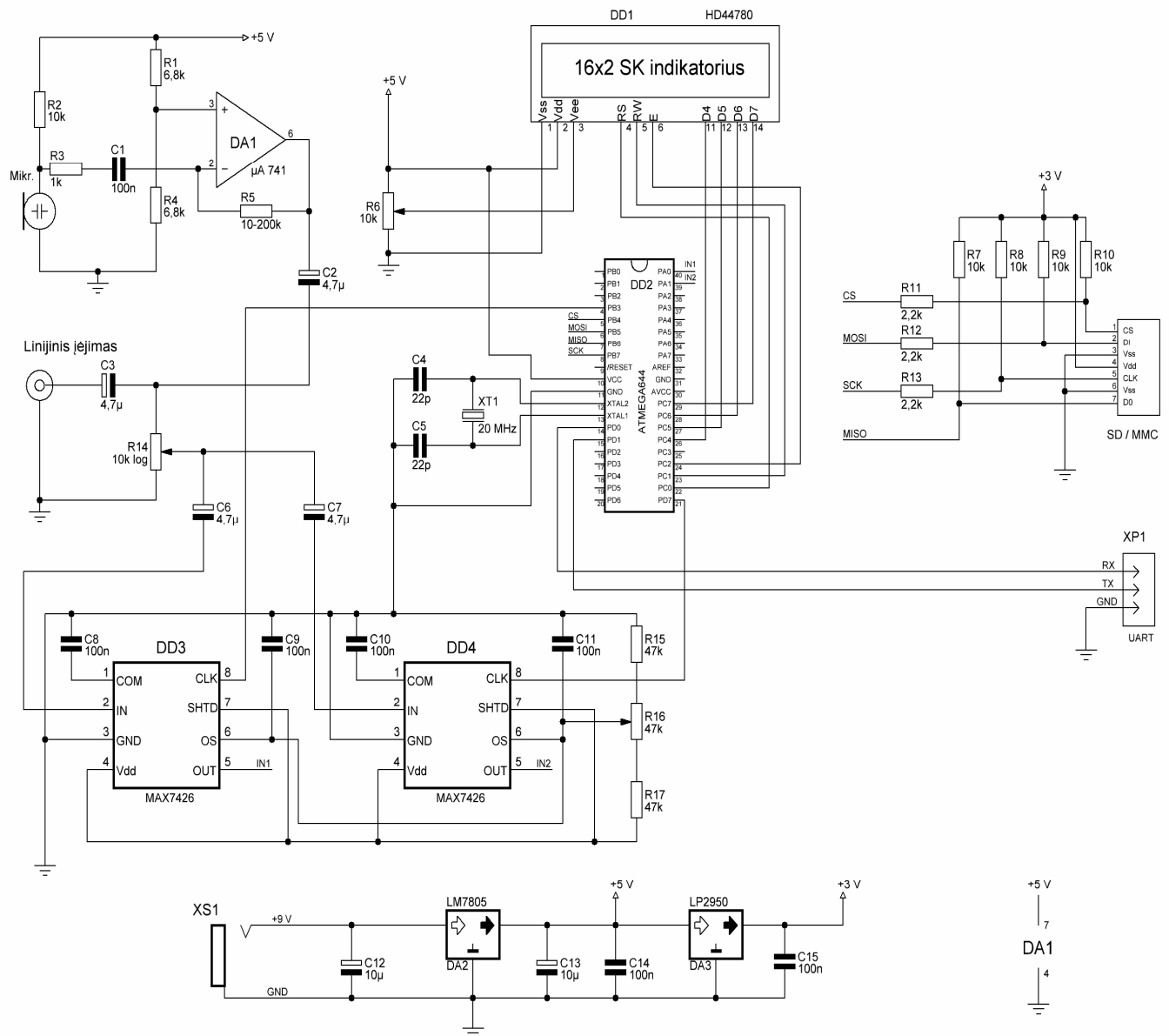
**B priedas. Suprojektuotos gitaros garsų atpažinimo sistemos principinė elektrinė schema**

**C priedas. Gitaros garsų atpažinimui tirti programa MATLAB naudotų algoritmų išeities tekstai**

**D priedas. Parašytos programos mikrovaldikliui ATmega644 išeities tekstai**



**A priedas. Dalyvaujant XV -toje Lietuvos jaunųjų mokslininkų konferencijoje gautos pažymos kopija**



					<b>GGAS-01</b>			
<i>Pak.</i>	<i>Lapas</i>	<i>Dokum. Nr.</i>	<i>Parašas</i>	<i>Data</i>	<b>Gitaros garsų atpažinimo sistema</b>	<i>Litera</i>	<i>Masė</i>	<i>Mastelis</i>
<i>Sudarė</i>		<i>A. Survila</i>		05.30				
<i>Tikrino</i>		<i>D. Navakauskas</i>		05.30				
						<i>Lapa</i> 1	<i>Lapų</i> 1	
					<i>Principinė elektrinė įrenginio schema</i>	<b>KTfm-10</b>		
								66

## C priedas. Gitaros garsų atpažinimui tirti programa *MATLAB* naudotų algoritmų išėities tekstai

### *MATLAB* programa HSS algoritmui realizuoti

```
clear all;
% signalu formavimas
[Fn, Pn]=uigetfile('*.wav', 'Pasikraukim'); %nusiskaitom faila
[S,Fs] = wavread([Pn Fn]);
Fs; %diskretizavimo daznis
T=1:length(S); T=T-1; T=T/Fs; % laiko vektorius atvaizduoti signalui laike
%figure(1);
S = S(:,1); % sudedam duomenis i viena stulpeli
N=length(S);
LNG=hann(N); % Hanningo langas
WH=1/mean((LNG)); % lango svorio koeficientas
S=S.*LNG;
FA=abs(fft(S,N)); %Furje transformacija
FA=2*FA/N*WH; % rezultata isvedame amplitudes reiksmemis
len=length(FA);
fx1=FA(1:2:len); %harmoniku suspaudimas
fx2=FA(1:3:len);
fx3=FA(1:4:len);
%fx4=FA(1:5:len);
len4=length(fx3);
Px=100*(FA(1:len4).*fx1(1:len4).*fx2(1:len4).*fx3(1:len4));
%Px=100*(FA(1:len4).*fx1(1:len4));
%Px1=Px.*(FA(1:len4));
Px([1:80]) = 0; %uzpildome spektro dedamasias 0 iki ~80 Hz
F=0:Fs/(len):Fs;
f=Fs/N.*(0:len4-1); %formuojamas daznio vektorius
[m,I]=max(Px(1:len4));
pitch=F(I);
figure(1);
ind = Px > m/10; % randamos visos pikines reiksmes
peak=(find(ind)-1);
peak1=peak*Fs/N; %randame pikines reiksmes dazni
plot(f, Px, peak1, Px(ind), 'o')
[nat]=natos(peak) %kvieciam funkcija natom atvaizduoti
ht=title(['natos: ',nat]);
xlim([80 1200])

function [nat] = natos(x)
%***** funkcija natom atvaizduoti *****
nn=round(12*log2(x/440)+29); %randamas natos numeris
nn=unique(nn); %istrina pasikartojancius elementus
on=round(nn/16); % oktavos numeris
nn=nn-on*12; %jei pereina i kita oktava, skaiciuoja raides is pradziu
if nn<0
    nn=nn+12;
end;
on=on+2;
on=unique(on);
on=num2str(on); %konvertuojam i simboli
switch nn
    case 0
        nat=['E' on];
    case 1
        nat=['F' on];
    case 2
        nat=['F' on '#'];
    case 3
```

```

        nat=['G' on];
    case 4
        nat=['G' on '#'];
    case 5
        nat=['A' on];
    case 6
        nat=['A' on '#'];
    case 7
        nat=['B' on];
    case 8
        nat=['C' on];
    case 9
        nat=['C' on '#'];
    case 10
        nat=['D' on];
    case 11
        nat=['D' on '#'];
end

```

## Autokoreliacijos programa

```

[Fn, Pn]=uigetfile('*.wav', 'Pasikraukim'); %nusiskaitom failą
[x,fs] = wavread([Pn Fn]); x = x(1:44000);
r = spCorr(x, fs, [], 'plot');
f0 = spPitchCorr(r, fs)
function [r] = spCorrelum(x, fs, maxlag, show)
    %% Initialization
    if ~exist('maxlag', 'var') || isempty(maxlag)
        maxlag = fs/50 % F0 is greater than 50Hz => 20ms maxlag
    end
    if ~exist('show', 'var') || isempty(show)
        show = 0;
    end
    r = xcorr(x, maxlag, 'coeff')
    if show
        %% atvaizduojam signalo grafika
        t=(0:length(x)-1)/fs; % padalinam i laiko periodus
        subplot(2,1,1);
        plot(t,x);
        legend('Waveform');
        xlabel('Time (s)');
        ylabel('Amplitude');
        xlim([t(1) t(end)]);
        %% atvaizduojam autokoreliacijos grafika
        d=(-maxlag:maxlag)/fs;
        subplot(2,1,2);
        plot(d,r);
        legend('Auto-correlation');
        xlabel('Lag (s)');
        ylabel('Correlation coef');
    end
end

function [f0] = spPitchCorr(r, fs)
    % ieškom maksimumų tarp 2ms (=500Hz) ir 20ms (=50Hz)
    ms2=floor(fs/500); % 2ms
    ms20=floor(fs/50); % 20ms
    % viena pusė tik veidrodinis signalo atvaizdas, todėl paliekam tik pusę jo
    r = r(floor(length(r)/2):end);
    [maxi,idx]=max(r(ms2:ms20))
    f0 = fs/(ms2+idx-1)
end

```

## **D priedas. Parašytos programos mikrovaldikliui ATmega644 išeities tekstai**

Parašytos programos mikrovaldikliui išeities tekstai, visos naudotos bibliotekos ir sukompiliuotas mašininis HEX kodas pateikiami pridėtame kompaktiniame diske.