

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO
FAKULTETA ZA MATEMATIKO IN FIZIKO

Lazar Milinković

**Sintetizator skladb na podlagi
glasbenih značilk**

DIPLOMSKO DELO
UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN MATEMATIKA

MENTOR: doc. dr. Zoran Bosnić

Ljubljana 2012

Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .



Št. naloge: 00018/2012

Datum: 13.04.2012

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko ter Fakulteta za matematiko in fiziko izdaja naslednjo nalogo:

Kandidat: **LAZAR MILINKOVIĆ**

Naslov: **SINTETIZATOR SKLADB NA PODLAGI GLASBENIH ZNAČILK**
SONG SYNTHESIZER BASED ON THE MUSICAL FEATURES

Vrsta naloge: Diplomsko delo univerzitetnega študija prve stopnje

Tematika naloge:

V svetu multimedije obstaja množica sistemov, ki glasbene skladbe opišejo z njihovimi značilkami kot so ritem, harmonije, zvrst, opis vokala, itd. Cilj diplomske naloge je v prvem koraku implementirati analizo nabora skladb, ki bo skladbe opisala z množico formalnih značilk. V drugem koraku naj kandidat izpeljane vrednosti značilk uporabi v obratni smeri: za avtomatsko sintezo skladbe (glasbeni sintetizator), pri kateri naj se poskusi približati lastnostim izvirne skladbe. Pri delu lahko kandidat uporabi obstoječa orodja in knjižnice za procesiranje signalov oz. glasbe, predstavi pa naj tudi sodobne glasbene značilke, ki se uporabljajo v literaturi.

Mentor:

doc. dr. Zoran Bosnić



Dekan Fakultete za računalništvo in informatiko:

prof. dr. Nikolaj Zimic

Dekan Fakultete za matematiko in fiziko:

akad. prof. dr. Franc Forstnerič



IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Lazar Milinković, z vpisno številko **63090150**, sem avtor diplomskega dela z naslovom:

Vzorec diplomskega dela

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Zorana Bosnića,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 11. januarja 2011

Podpis avtorja:

Zahvalil bi se mentorju doc. dr. Zoranu Bosniću za pomoč pri diplomskem delu. Predvsem pa bi se zahvalil družini in svoji dragi Anji za podporo tekom študija.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Hierarhično razvrščanje skladb v skupine na podlagi glasbenih značilk	3
2.1	MIRToolbox in opis značilk	3
2.2	Rezultati razvrščanja in analiza	13
3	Sinteza skladbe na podlagi značilk	17
3.1	Opis problema in cilji	17
3.2	Sinteza inštrumentov	17
3.2.1	Fazni kodirnik glasu in sinteza klavirja	18
3.2.2	Sinteza klarineta	19
3.3	Izbira značilk za sintezo in metode za njihovo uporabo	24
3.3.1	Čas in naklon faze naraščanja amplitude	25
3.3.2	Tempo	25
3.3.3	Povprečna amplituda in dinamika	26
3.3.4	Ključ	28
3.3.5	Svêtlost	29
3.3.6	Disonanca	31
4	Merjenje uspešnosti sinteze	33

KAZALO

5 Sklepne ugotovitve	37
Literatura	38

Povzetek

V diplomskem delu je predstavljena implementacija programa, ki sintetizira solo skladbe na podlagi vnaprej določenih značilk. Program njihove vrednosti izračuna na določeni skladbi, ki jo nato poskuša posnemati s sintezo skladbe z enakimi značilkami. Za računanje vrednosti značilk je uporabljena knjižnica MIRToolbox iz okolja Matlab. Pri sintezi so uporabljeni povprečna amplituda in tempo ter ključ, svetlost in disonanca skladbe, ki določajo množico tonov v skladbi. Implementirana je sinteza klavirja in klarineta, inštrument pa je izbran s pomočjo dveh dodatnih značilk. Na podlagi dveh načinov merjenja je ugotovljena delna uspešnost sinteze. Predstavljenih je nekaj idej za izboljšave v implementaciji, ki bi doprinesle k večji podobnosti med skladbama.

Abstract

The diploma thesis presents the implementation of a program which synthesizes solo compositions on the basis of predefined attributes. The program calculates their values on a particular track, which it later attempts to emulate with the synthesis of a track with the same attributes. Attribute values are calculated using Matlab library MIRToolbox. Average amplitude, tempo, key, brightness and dissonance are used in synthesis, with the last three attributes defining a set of all possible tones in a track. Synthesis of a piano and a clarinet is implemented. Two more attributes are used for an instrument to be chosen automatically. Based on two methods of measuring, a partial success of the synthesis is observed. Presented are some ideas for improvements in the implementation, which would result in a greater similarity between compositions.

Poglavje 1

Uvod

Z razvojem računalništva se je na glasbenem področju začelo mnogo raziskav. Predstavljenih je bilo veliko metod za analizo zvočnih signalov in pridobivanje njihovih značilk. Slednje so postale uporabne predvsem pri prepoznavi govora in razvrščanju glasbenih del. Najbolj znana primera sta radio Pandora in Last.fm, ki uporabniku na podlagi tega, kar posluša, poiščeta skladbe s podobnimi lastnostmi. Pri umetni inteligenci so šli še dlje in razvili algoritme za računalniško komponiranje glasbe. Vprašanje, ki se nam lahko zastavi, je, kako ustvariti skladbo, ki bi posnemala neko drugo skladbo in ji bila čim bolj podobna.

V diplomskem delu smo problem poskušali rešiti z značilkami. Na podlagi množice vrednosti značilk, ki opisujejo dano skladbo, smo želeli sintetizirati skladbo s čim bolj podobnimi vrednostmi. Za pridobivanje značilk bi uporabili že obstoječi programski paket. Za sintezo bi morali najprej raziskati pomen značilk in način, kako programski paket izračuna njihove vrednosti. Njihovo informativnost bi najprej preizkusili pri hierarhičnem razvrščanju skladb v skupine.

Odločili smo se uporabiti knjižnico MIRTtoolbox v okolju Matlab, ki podpira pridobivanje veliko glasbenih in statističnih značilk. Pri razvrščanju smo smiselno izbrali število skupin, vsaki skupini določili razred in z mero Relief ugotovili, da so najbolj informativne značilke tiste, ki določajo tonaliteto

skladbe. Pri sintezi smo se omejili na solo skladbe, natančneje klavirske in s klarinetom. Za zvok klavirja smo uporabili posnetke nekaj tonov, ostale pa smo dobili z uporabo faznega kodirnika glasu. Pri klarinetu smo raziskali njegove spektralne lastnosti in na podlagi tega določili kombinacijo sinusnih funkcij, uporabljenih pri generiranju zvoka. Prav tako smo raziskali, kako se pri igranju posameznega tona spreminja amplituda in s tem določili amplitudno ovojnico. 24 značilk, kjer je vsaka enaka verjetnosti, da je skladba v določenem ključu, smo uporabili tako, da smo izbrali dve značilki z največjo verjetnostjo in pripadajoča ključa uporabili pri sintezi. Glede na ključ smo določili glasbeno lestvico, verjetnosti za pojavitev posameznega tona v skladbi in tri tone, ki se skupaj pojavljajo v akordu. Za dve značilki, čas in naklon naraščanja amplitude, smo ugotovili, da uspešno ločujeta solo skladbe in smo ju zato uporabili pri avtomatičnem odločanju o tem, kateri inštrument bo uporabljen pri sintezi. Svêtlost zvoka smo uporabili pri določanju oktav, iz katerih sintetizator izbira tone, povprečno amplitudo zvoka pri določanju glasnosti tonov, tempo pa pri izračunu dolžine četrтинke v sekundah. Deleže dolžin tonov smo modelirali približno po normalni porazdelitvi s četrтinko kot najpogostejšo dolžino. Senzorično disonanco zvoka smo uporabili le delno in njegovo vrednost podali programu kot parameter.

V poglavju 2 je opisan pomen značilk, ki smo jih uporabili pri hierarhičnem razvrščanju skladb. Podrobno je opisano, na kakšen način MIR-Toolbox izračuna njihove vrednosti. Nato so podani rezultati razvrščanja s komentarjem in informativnost značilk. V poglavju 3 je bolj podrobno razloženo, kako smo nekatere izmed značilk uporabili pri sintezi skladbe. Predstavljena je približna shema programa, ki sintetizira skladbe. Opisane so metode, s katerimi smo sintetizirali zvok klavirja in klarineta. V poglavju 4 sta predstavljena dva načina merjenja uspešnosti sinteze in komentar k rezultatom merjenja.

Poglavje 2

Hierarhično razvrščanje skladb v skupine na podlagi glasbenih značilk

2.1 MIRToolbox in opis značilk

MIRToolbox je knjižnica iz okolja Matlab, namenjena predvsem pridobivanju velikega števila glasbenih značilk iz zvočnih datotek, kot pove tudi kratica imena (MIR = Music information retrieval). Razvita je bila na finski univerzi Jyväskylä, pri našem delu pa smo uporabljali verzijo 1.3.4, ki je izšla 13. decembra, 2011. MIRToolbox omogoča pridobivanje približno 50 glasbenih in statističnih lastnosti glasbe. Glasbene značilke se delijo v pet skupin: ritem, dinamika, barva zvoka, višina najboljših frekvenc in tonaliteta. Za značilke primerov (skladb) pri razvrščanju smo jih izbrali 16. V nadaljevanju je za vsako opisan njen pomen in funkcija, ki izračuna njeno vrednost. Vse funkcije imajo v imenu predpono mir, tako da so funkcije MIRToolboxa jasno ločene od Matlabovih. Bolj podroben opis o knjižnici in njenih funkcijah je na voljo na [1].

mirrms

Funkcija izračuna povprečno amplitudo signala s formulo:

$$s_{rms} = \sqrt{\frac{1}{n} \sum_{i=1}^N s_i^2} \quad (2.1)$$

Funkcija je pravzaprav ekvivalentna ukazu $\text{sqrt}(\text{mean}(s.^2))$ v Matlabu, le da zaradi delitve signala na okvirje (angl. *frames*) deluje nekoliko počasneje. Zato smo v kodi namesto *mirrms* uporabljali zgoraj omenjeni ukaz.

mirlowenergy

Izračuna odstotek okvirjev signala, pri katerih je amplituda vselej pod njenim povprečjem. Signal je predhodno razdeljen na okvirje, ki se polovično prekrivajo in imajo dolžino 50 milisekund.

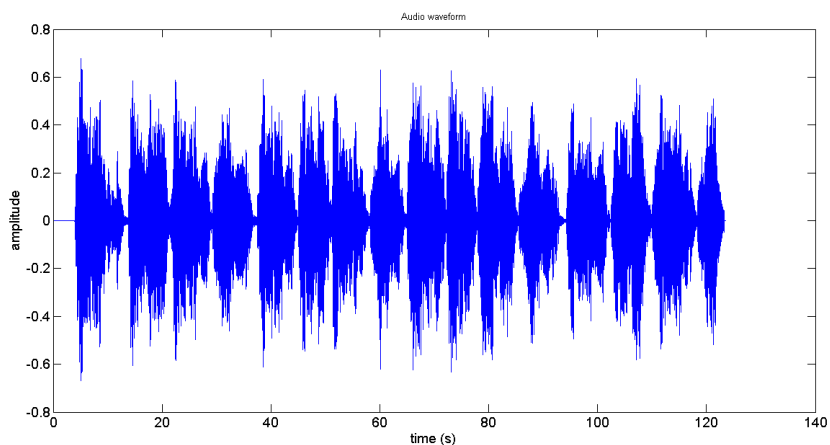
mireventdensity

Izračuna amplitudno ovojnico (angl. *envelope*) signala, označi lokalne maksimume ovojnice in izračuna povprečno število lokalnih maksimumov na sekundo. To je eden od načinov, kako izračunati tempo signala. Čeprav funkcija ne vrne vrednosti v udarcih na minuto (angl. *beats per minute*, to ni toliko pomembno, saj nas ne zanima tempo, temveč samo razvrščanje na podlagi vrednosti, ki jih funkcija vrne.

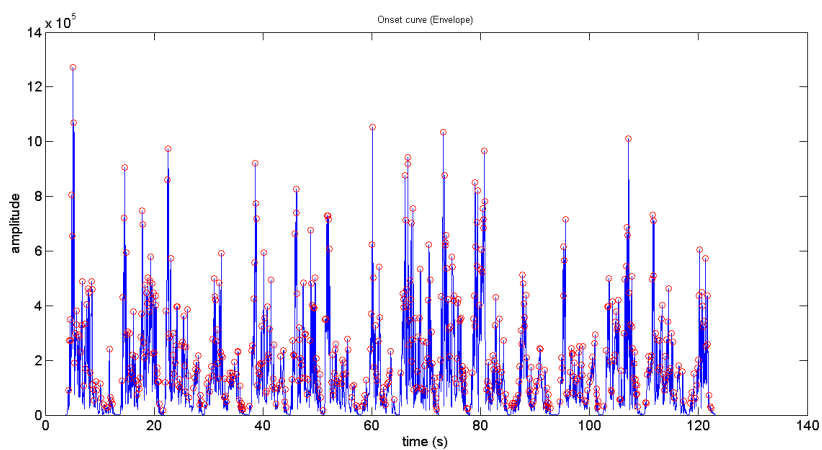
Na sliki 2.2 je primer amplitudne ovojnice signala, prikazanega na sliki 2.1. Označeni so tudi lokalni maksimumi.

mirtempo

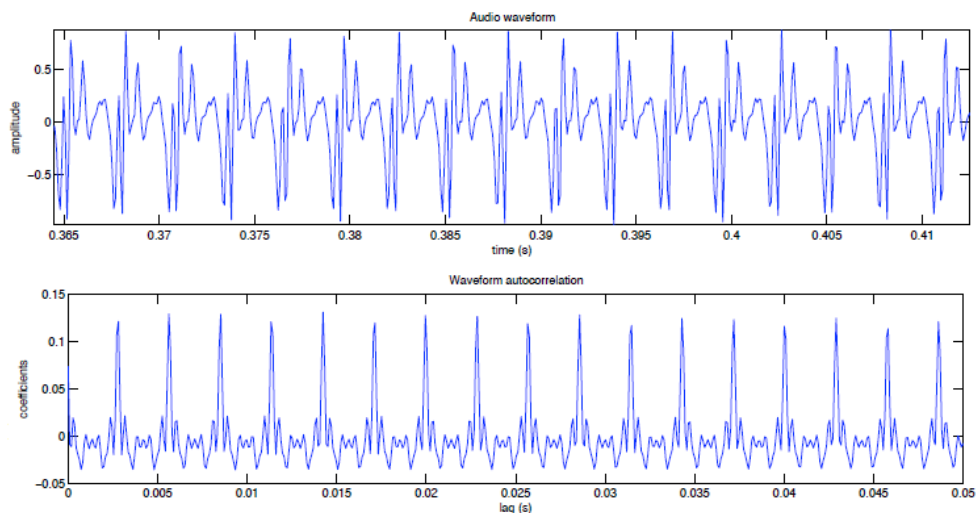
Oceni tempo z zaznavanjem periodičnosti v amplitudni ovojnici signala. S funkcijo *mirautocor* izračuna avtokorelacijsko funkcijo amplitudne ovojnice. Za odmik j je avtokorelacija izračunana kot skalarni produkt signala z istim signalom, zamaknjenim za j vzorcev (angl. *samples*). Avtokorelacija je sorodna konvoluciji, razlika med njima je le predznak odmika signala. Formula



Slika 2.1: Primer valovanja signala.



Slika 2.2: Amplitudna ovojnica signala na sliki 2.1. Lokalni maksimumi so označeni z rdečimi krogi.



Slika 2.3: Valovanje zvoka trobente (zgoraj) in avtokorelacijska funkcija. Funkcija doseže maksimum pri vsakem odmiku, ki je večkratnik periode signala.

konvolucije dveh signalov je:

$$x[n] * h[n] = \sum_{k=0}^{\infty} h[k]x[n - k], \quad (2.2)$$

formula avtokorelacijske funkcije signala pa:

$$corr(x[n], x[n]) = \sum_{k=0}^{\infty} x[k]x[n + k]. \quad (2.3)$$

Ko odmik ustreza periodi signala, sta signala najbolj korelirana in korelacijska vrednost je takrat zelo velika (primer na sliki 2.3). *mirtempo* vzame največjo korelacijsko vrednost in pripadajoči odmik uporabi pri izračunu tempa s formulo $tempo = \frac{60}{odmik}$.

mirpulseclarity

Oceni ritmično jasnost (angl. *pulse clarity*) oziroma moč beatov, ki so ocenjeni s pomočjo funkcije tempo. Za hevristiko vzame največjo korelacijsko vrednost v avtokorelacijski funkciji amplitudne ovojnice.

mirattacktime in mirattackslope

Lokalni maksimumi v amplitudni ovojnici določajo tudi konec faze naraščanja amplitude skladbe. Z izračunom lokalnih minimumov, ki se pojavijo tik pred lokalnimi maksimumi, lahko izračunamo povprečni čas naraščanja in povprečni naklon naraščanja amplitude.

mirbrightness

Funkcija izračuna delež signala s frekvencami, ki so višje od fiksne frekvence, podane kot vhodni parameter (privzeta vrednost je 1500 Hz). Najprej funkcija *mirspectrum* s hitro Fourierjevo transformacijo poda signal v frekvenčni domeni in vrne graf njihovih amplitud (spekter). Světlost (angl. *brightness*) signala s fiksno frekvenco f_k je tako definirana kot:

$$brightness = \frac{\sum_{i=k+1}^N a(f_i)}{\sum_{i=0}^N a(f_i)}, \quad (2.4)$$

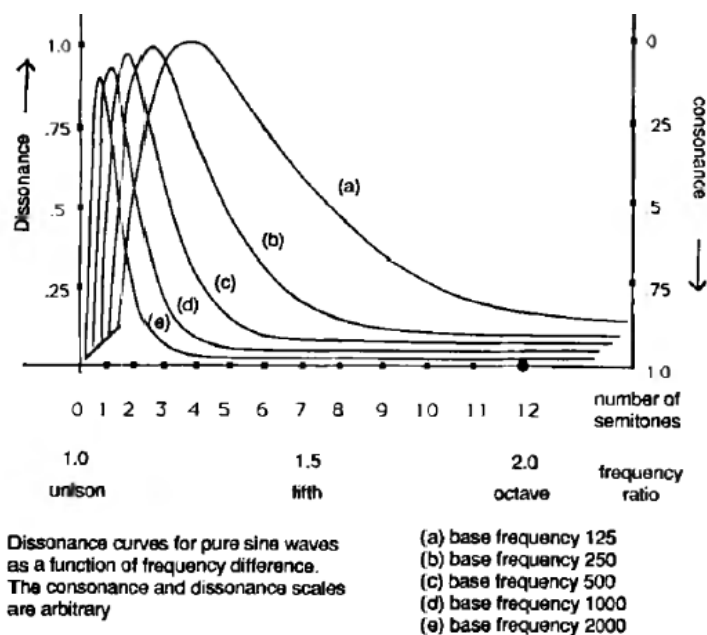
kjer je $a(f_i)$ amplituda frekvence f_i .

mirroughness

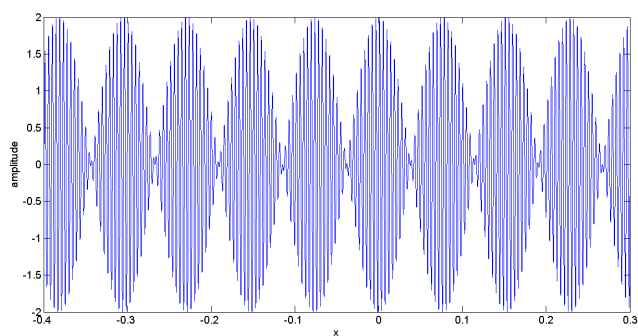
Plomp in Levelt sta leta leta 1965 predstavila graf disonance kot funkcijo razlike med frekvencama dveh sinusoid, pri čemer je prva frekvenca fiksna, druga pa spremenljivka [2]. Krivulja je odvisna tudi od fiksne frekvence. Če je slednja visoka, disonanca hitreje narašča z večanjem razlike frekvenc, a nato tudi hitreje pada (slika 2.4).

Dva pojava, ki ju pri disonanci človeško uho lahko loči, sta bitje (angl. *beating*) in hrapavost (angl. *roughness*). Oba sta tesno povezana z nihanjem amplitude signala. Stopnja nihanja amplitude, ki nastane kot kombinacija dveh sinusoid, je enaka njuni razliki v frekvencah.

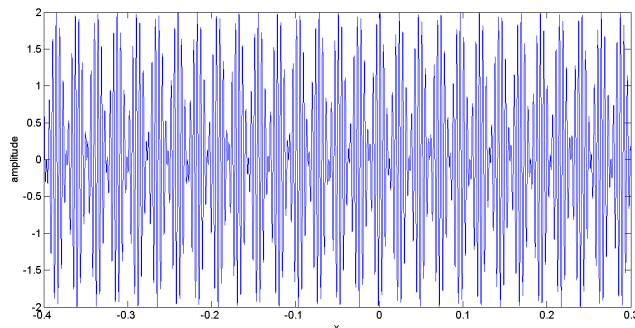
Sinusoidi z zelo majhno razliko, ki pa jo človeško uho še vedno opazi, povzročita počasnejše nihanje amplitude, ki je podobno bitju. Prikazano je na sliki 2.5. Z večanjem razlike frekvenc se večja tudi amplitudno nihanje. Takrat ima signal na videz vedno enako glasnost, nihanje pa povzroči



Slika 2.4: Funkcija disonance kot razlika med frekvencama dveh sinusoid [2]. Graf je pri nižji fiksni frekvenci (125 Hz pri krivulji (a)) bolj položen kot pri višji (1000 Hz pri krivulji (d)).



Slika 2.5: Valovanje vsote sinusoid s frekvencama 220 Hz (fundamentalna harmonija tona A3) in 233.1 Hz (fundamentalna harmonija tona B3b). Amplitudno nihanje je počasnejše, zato človeško uho zazna spremembo glasnosti kot bitje.



Slika 2.6: Valovanje vsote sinusoid s frekvencama 220 Hz (ton A3) in 261.6 Hz (ton C4). Glede na graf na sliki 2.4 je hrapavost zvoka med tema sinusoidama največja, s tem pa je tudi amplitudno nihanje najhitrejše.

hrapavost zvoka [3]. Primer je prikazan na sliki 2.6. Kot prikaže slika 2.4, hrapavost doseže maksimum pri določeni razliki frekvenc, nato pa hitro pade in doseže minimum pri frekvencah z razliko ene oktave. Takrat je ena frekvenca dvakrat večja od druge, njuni sinusoidi pa se boljše ujemata, ker sta njuni vrednosti enaki pri periodi sinusoide z nižjo frekvenco. Kombinacija takih dveh sinusoid je torej konsonantna.

Funkcija *mirroughness* meri senzorično disonanco. Vsak naravni ton je kombinacija več sinusoid, prevladujejo pa sinusoide s fundamentalno frekvenco in njeno harmonično vrsto. Funkcija zato izračuna spekter signala, izbere frekvence z največjimi amplitudami in izračuna disonance med vsemi pari. William Sethares je leta 1998 predstavil metodo za izračun senzorične disonance [4]. Ta je enaka vsoti vseh disonanc, vsaka disonanca pa je izračunana kot produkt amplitud obeh frekvenc in utežena z vrednostjo njune razlike na grafu iz slike 2.4. To metodo smo tudi uporabili pri računanju senzorične disonance skladb, ki smo jih razvrščali. Signal je najprej razdeljen na okvirje s širino 50ms. Za vsak okvir funkcija izračuna Fourierjevo transformacijo in disonanco izmeri med frekvencami z največjo amplitudo. Senzorična disonanca je izmerjena za vsak posamezen okvir, ker se disonanca pojavi samo

med frekvencami, ki se v signalu pojavijo ob istem času. Za vrednost značilke smo uporabili povprečno senzorično disonanco okvirjev.

mirregularity

Nepravilnost (angl. *irregularity*) signala je stopnja različnosti med maksimumi spektra. Funkcija najprej določi lokalne maksimume spektra (amplitude frekvenc) in nepravilnost izračuna kot normalizirano vsoto kvadratnih razlik dveh zaporednih maksimalnih amplitud:

$$irregularity = \left(\sum_{k=1}^N (a_k - a_{k+1})^2 \right) / \sum_{k=1}^N a_k, \quad (2.5)$$

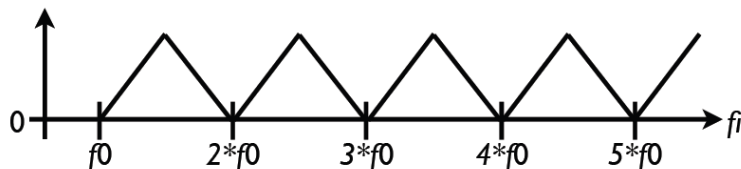
kjer je a_{N+1} enak 0. Formulo je leta 1999 predstavil Kristoffer Jensen v svoji doktorski disertaciji *Timbre Models of Musical Sounds*, ki je na voljo na [5].

mirpitch

Funkcija izračuna spekter signala s hitro Fourierjevo transformacijo in vrne lokalne maksimume spektra, tj. frekvence z največjimi amplitudami. Če bi kot parameter za število lokalnih maksimuumov vzeli 1, bi funkcija vrnila najboljšo višino tona (angl. *pitch*), kar je koristno pri iskanju fundamentalne frekvence (prve frekvence v harmonični seriji) signala z enim tonom (na primer zaigrana nota C4 na klavirju), ker ta določa višino tona. Ker skladbe vsebujejo več tonov, je spekter bolj porazdeljen in je za njegov opis bolje vzeti več višin in povprečiti njihovo frekvenco.

mirinharmonic

S funkcijo *mirpitch* najprej izračuna fundamentalno frekvenco (najboljšo višino), nato pa delež frekvenc, ki niso njeni večkratniki oziroma niso del idealne harmonične vrste. Za to uporabi preprosto žagasto funkcijo, prikazano na sliki 2.7.



Slika 2.7: Funkcija, s katero *mirinharmonicity* izračuna delež neharmoničnih frekvenc.

mirchromagram

Funkcija najprej izračuna spekter signala v logaritemskih enotah, nato pa porazdelitev tega spektra vzdolž dvanajstih razredov višin tonov (na primer frekvence 185Hz, 370Hz, 740Hz spadajo v razred $F\sharp$), imenovanega kromogram. Za attribute smo uporabili deleže vseh razredov tonov.

mirkeystrength

Izračuna moč ključa (ki lahko zavzame vrednost med -1 in 1) za vsakega od 24 kandidatov tako, da izračuna navzkrižno korelacijo med porazdelitvijo višin tonov funkcije *mirchromagram* in porazdelitvijo, ki ustreza porazdelitvi višin tonov signala, ki je v enakem ključu kot ključ, katerega moč merimo. Vrednost 1 torej pomeni korelacijo in s tem največjo verjetnost, da je skladba v danem ključu. Tudi tukaj smo za attribute vzeli moči vseh 24 ključev.

mirmode

Oceni modalnost signala (dur ali mol) kot razliko med največjo močjo durovega ključa in največjo močjo molovega ključa. Modalnost lahko tako zavzame vrednosti med -1 in 1: višja je vrednost, večja je verjetnost, da je modalnost dur in obratno.

mirnovelty

Funkcija najprej izračuna podobnostno matriko, ki prikazuje podobnost med vsemi pari okvirjev signala. Konvolucija glavne diagonale matrike z gaussovimi jedrom vrne krivuljo znatnih teksturnih sprememb vzdolž signala. Za atribut smo uporabili povprečno vrednost krivulje.

Za attribute smo poleg glasbenih uporabili še dve statistični lastnosti: število presečišč signala skozi ničlo in sploščenost signala.

mirzerocross

Funkcija izmeri, kolikokrat signal zamenja predznak. To sicer ni samo statistični podatek, ampak tudi preprost indikator hrupa (angl. *noisiness*).

mirflatness

Statistični podatek, ki pove, koliko je porazdelitev sploščena (v tem primeru gre za sploščenost signala). Vrednost izračuna kot razmerje med geometrijskim in aritmetičnim povprečjem:

$$\frac{\sqrt[N]{\prod_{n=0}^{N-1} x(n)}}{\left(\frac{\sum_{n=0}^{N-1} x(n)}{N}\right)} \quad (2.6)$$

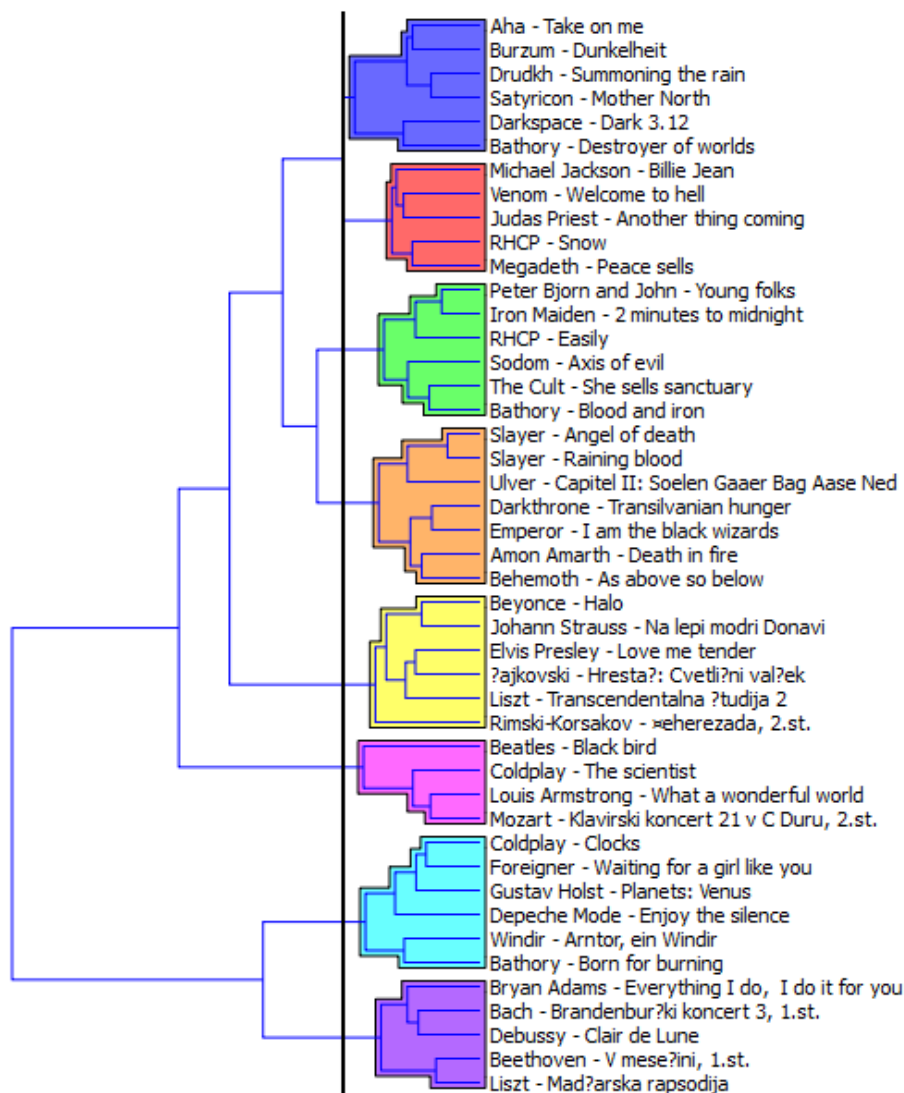
Ko smo izračunali vrednost vseh 54 značilnik iz 45 skladb različnih zvrsti, smo uporabili programski paket Orange za hierarhično razvrščanje z evklidsko razdaljo med primeri in Wardovo razdaljo med skupinami [6]. Da bi lahko ocenili uspešnost razvrščanja v skupine, smo za primere izbrali skladbe znanih zvrsti (v grobem bi jih lahko razdelili na popularno glasbo, klasično glasbo, metal in rock).

2.2 Rezultati razvrščanja in analiza

Na sliki 2.8 je prikazano hierarhično drevo, ki je rezultat razvrščanja skladb v skupine. Skladbam, ki na drevesu pripadajo isti skupini, smo določili isti razred in na podlagi tega ocenili informativnost značilke.

V tabeli 2.1 so podane značilke in njihova informativnost glede na mero ReliefF, ki smo jo izračunali s pomočjo programskega paketa Orange. Podatki so v padajočem vrstnem redu. Izpuščena je večina značilke, dobljenih s funkcijama *mirchromagram* in *mirkeystrength*, ker so si med sabo zelo podobne. Za izpuščena mesta v tabeli je torej privzeto, da so na njih omenjene značilke. Prvih trinajst informativnih značilke pripada ravno omenjenima funkcijama, ki ocenjujeta tonaliteto signala. Na 14. mestu je povprečni čas naraščanja amplitude, ki ocenjuje barvo signala. Pomembnost te značilke je vidna na hierarhičnem drevesu, saj so v zadnji skupini vse skladbe klavirske. Kljub temu, da gre za zelo enostavni značilki, sta bili informativni tudi povprečna amplituda in število presečišč z ničlo. Med funkcijama *mirtempo* in *mireventdensity*, ki ocenjujeta tempo skladbe, se je za bolj informativno izkazala slednja, neharmoničnost in delež majhne amplitude pa sta najbolj neinformativni značilki in nista prispevali k razvrščanju skladb.

Na podlagi žanrov, ki jih poznamo za analizirane skladbe, lahko razvrščanje ocenimo kot uspešno. V prvi skupini so tako večinoma “black metal” skladbe, v drugi in tretji rock skladbe, v četrti “death metal” skladbe, v peti plesne skladbe, v že omenjeni osmi skupini pa so klavirske skladbe. Šesto skupino bi lahko označili kot skupino z mirnimi skladbami, medtem ko med skladbami v sedmi skupini ne moremo najti smiselne povezave. Kljub temu model z izbranimi značilkami ni najboljši pri razvrščanju elektronskih in rap skladb skupaj s skladbami v drevesu, saj ta žanra ne dajeta poudarka na tonaliteti, v katero spada največ značilke v našem modelu. Pri razvrščanju smo v tem primeru zato dobili bolj nesmiselne skupine. Kljub temu lahko elektronske skladbe jasno ločimo od ostalih že na podlagi ritmične jasnosti. Moč ritmičnih udarcev (angl. *beat*) je pri takih skladbah zelo velika, zato ima pri njih ta značilka bistveno višje vrednosti kot pri ostalih.



Slika 2.8: Rezultat hierarhičnega razvrščanja v skupine z evklidsko razdaljo med primeri in Wardovo razdaljo med skupinami.

	Atribut	ReliefF
1	chr A#	0.260
2	chr G#	0.206
3	minor D	0.205
14	attack time	0.135
23	zero-cross	0.103
25	brightness	0.102
27	pitch	0.098
38	irregularity	0.074
40	rms	0.067
41	eventDensity	0.065
42	pulseClarity	0.063
44	flatness	0.040
45	roughness	0.040
46	novelty	0.039
47	mode	0.036
48	attackSlope	0.031
50	tempo	-0.011
51	inharmonic	-0.023
52	lowEnergy	-0.028

Tabela 2.1: Informativnost atributov glede na mero ReliefF. Podatki za verjetnosti tonov v kromogramu in ključev so izpuščeni, razen pri prvih treh mestih.

*POGLAVJE 2. HIERARHIČNO RAZVRŠČANJE SKLADB V SKUPINE
16 NA PODLAGI GLASBENIH ZNAČILK*

Poglavje 3

Sinteza skladbe na podlagi značilk

3.1 Opis problema in cilji

Rdeča nit diplomske naloge je bila uporabiti glasbene značilke za sintezo skladbe. V prejšnjem poglavju smo za vhod podali skladbo in nato izračunali vrednost njenih značilk. Naš program, sintetizator glasbe na podlagi značilk, se ne ustavi tukaj, temveč seznam značilk uporabi za sintezo skladbe, ki ima enake vrednosti značilk kot vhodna skladba. S poslušanjem obeh skladb lahko hitro ocenimo uspešnost sinteze. Zaradi kompleksnosti večinštrumentalnih skladb smo se odločili za posnemanje oziroma sintezo solo skladb, klavirskih in s klarinetom. Naloga je bila zato najprej sintetizirati zvok teh dveh inštrumentov. V nadaljevanju smo nato izbrali seznam značilk, ki jih program uporabi pri sintezi.

3.2 Sinteza inštrumentov

Eden izmed problemov je bil generirati take kombinacije sinusoid, da bi signal zvenel čim bolj podobno določenemu inštrumentu. To se je izkazalo kot precej kompleksen problem predvsem pri klavirju, ki je bil eden od inštrumentov, ki

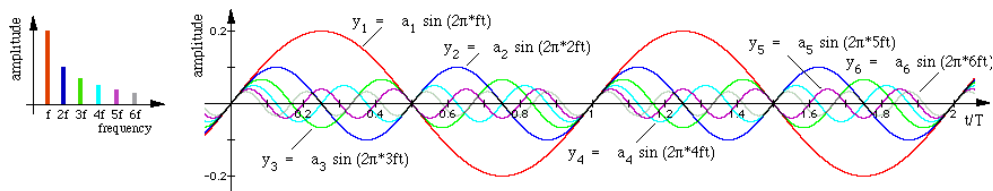
smo jih hoteli uporabiti pri sintezi. Ker sinteza inštrumentov ni bila osrednja naloga diplomske naloge, smo se odločili uporabiti drugačen pristop, opisan v nadaljevanju. Sinteza klarineta pa se je izkazala kot lažja naloga, saj je bilo precej bolj enostavno dobiti sprejemljiv zvok. Kako smo prišli do njega, je prav tako opisano v nadaljevanju.

3.2.1 Fazni kodirnik glasu in sinteza klavirja

Fazni kodirnik glasu (angl. *phase vocoder*, kot kombinacija besed *voice* in *coder*) je eden izmed algoritmov digitalnega procesiranja signalov, ki vhodnemu signalu lahko spreminja dolžino in višino. Osrednji del algoritma je časovno kratka Fourierjeva transformacija, ponavadi implementirana kot hitra Fourierjeva transformacija. Z njo fazni kodirnik transformira signal iz časovne v frekvenčno definicijsko območje (spekter), s čimer je možno spreminjanje amplitud in faz posameznih frekvenčnih komponent zvoka. Z inverzno Fourierjevo transformacijo je na podlagi spektra signal ponovno sintetiziran in pretvorjen nazaj v časovno definicijsko območje. Bolj obširna in teoretična razlaga delovanja faznega kodirnika je na voljo na [7].

Z njegovo uporabo smo se lahko izognili sintezi klavirja. Na [8] so na voljo posnetki vseh tonov, zaigranih s klavirjem. Ker bi bilo prostorsko potratno shraniti vseh 88 posnetkov, smo shranili le tone A iz petih oktav (A2 do A6). Samo en ton se ni izkazal za dobro rešitev, ker fazni kodirnik ni dobro deloval pri spreminjanju višine za več oktav. Za vsak ton, ki smo ga želeli dobiti pri sintezi skladbe, smo uporabili fazni kodirnik, ki za vhod vzame ton A iz iste oktave ter razmerje med njeno frekvenco in frekvenco želenega tona. Pri tem smo uporabili že obstoječo implementacijo faznega kodirnika v Matlabu. Celotna koda in primer uporabe je na voljo na [9].

Prav tako smo imeli v vektorju shranjene samo frekvence tonov A. Program smo implementirali tako, da imamo vse tone oštevilčene (prvi ton je E2), na podlagi tega pa je program za želen ton našel najbližji ton A. Da bi lahko uporabil razmerje med njunima frekvencama v faznem vokoderju,



Slika 3.1: Valovanje strune (rdeča sinusoida) in strun z dvakrat, trikrat, štirikrat, petkrat in šestkrat krajšo dolžino. Amplituda valovanja je odvisna od napetosti, s katero zanihamo struno [10].

program izračuna frekvenco želenega tona s formulo:

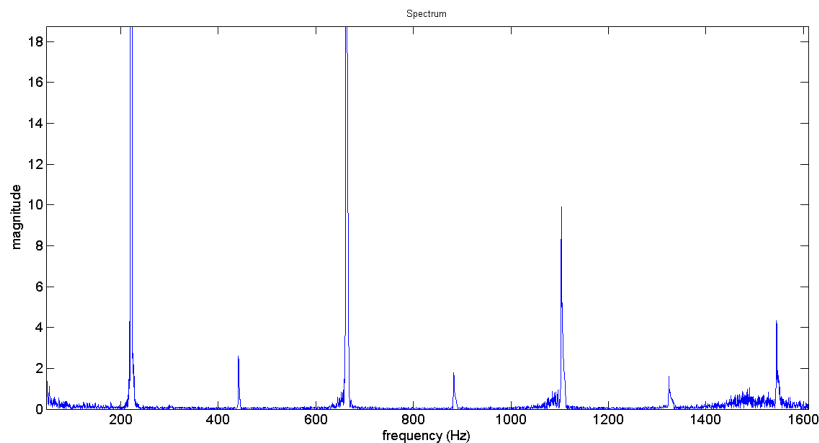
$$f_{ton} = f_A \times 2^{\frac{n}{12}}, \quad (3.1)$$

kjer je n razlika med številko želenega tona in številko tona A . Iz vektorja tona, ki s pomočjo faznega vokoderja dobi svojo višino, program na podlagi predhodno določene dolžine tona vzame le njegovih prvih n elementov.

3.2.2 Sinteza klarineta

Harmonije

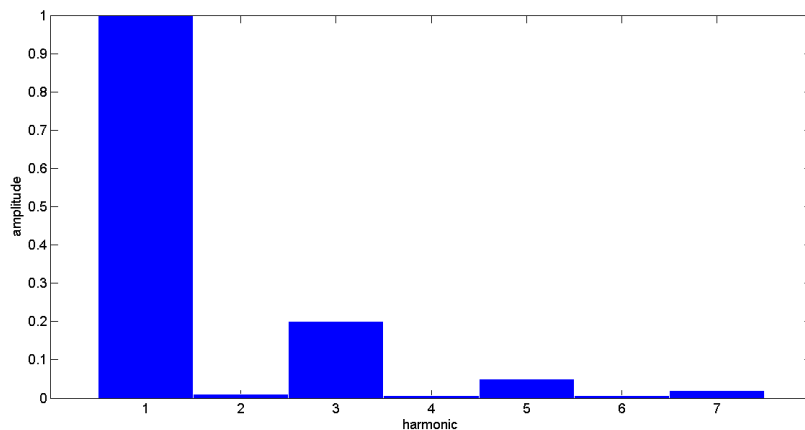
Pitagora je pred 2500 leti odkril, da dobimo prijeten zvok, če zanihamo dve struni s takima dolžinama, da se da njuno razmerje izraziti s celimi števili. Vzemimo za primer razmerje 1:2. Če zanihamo struno na polovici njene dolžine, dobimo določeno valovanje. Njeno dolžino razpolovimo tako, da jo držimo na polovici. Če jo zanihamo na eni od polovic, zaniha z dvakrat večjo frekvenco. Splošneje, če struno razdelimo na n enakih delov, vsak del strune zaniha z n -krat večjo frekvenco (slika 3.1). Geometrijskemu zaporedju takih frekvenc, kjer ima n -ti člen n -krat večjo frekvenco od prve frekvence v zaporedju, pravimo harmonična serija. Členom pravimo harmonije, prvemu členu pa tudi fundamentalna harmonija. V večini primerov je to višina, ki jo prepoznamo ob poslušanju določenega tona. Kljub temu to ni edina harmonija, ki jo slišimo, ko zanihamo struno. Vsak ton je kombinacija več harmonij z



Slika 3.2: Spekter tona A3 (220 Hz), zaigranega s klarinetom, ki ga funkcija *mirspectrum* izračuna s hitro Fourierjevo transformacijo. Iz spektra je razvidno, da prevladujejo lihe harmonije (220Hz, 660Hz, 1100Hz in 1540Hz), medtem ko so sode zadušene. Opomba: slika je povečana in ne prikazuje celotnega spektra zaradi relativno dosti večje magnitude frekvence 220Hz (290) in frekvence 660Hz (50).

določeno amplitudo [11]. S pomočjo hitre Fourierjeve transformacije lahko vsak signal namesto v časovni predstavimo v frekvenčni domeni in tako dobimo spekter frekvenc, ki pokaže marsikatero zanimive lastnosti signala. Če primerjamo spektre tona, zaigranega z različnimi inštrumenti, vidimo, da kljub enaki višini tona spektri niso enaki. Če bi bili, ne bi razločili med zvokom dveh inštrumentov. Lastnosti spektra dajo vsakemu inštrumentu barvo zvoka, zaradi katere ga lahko ločimo od ostalih.

Na sliki 3.2 je prikazan spekter tona A v tretji oktavi, zaigranega s klarinetom. Njegova največja posebnost je, da imajo lihe harmonije bistveno večjo amplitudo in zato večjo vlogo pri barvi tona. To informacijo smo uporabili pri sintezi klarineta oziroma njegovega približka. Spekter se namreč spreminja ne le pri zamenjavi inštrumenta, temveč tudi pri spreminjanju višine tona. Za nižje tone velja, da so bolj kompleksni (so sestavljeni iz več harmoničnih in



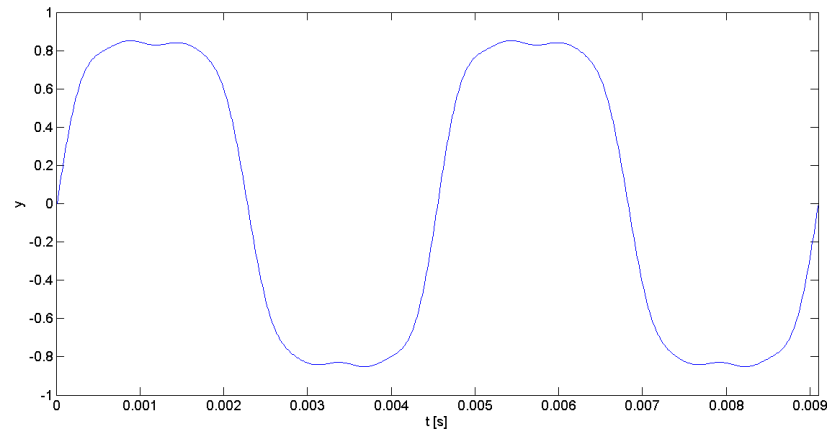
Slika 3.3: Amplitude vseh sedmih harmonij, uporabljenih pri sintezi klarineta.

neharmoničnih frekvenc) kot višji toni. Mi smo pri sintezi za vse tone uporabili enake harmonije in enake amplitude. Vsak ton sintetiziranega klarineta je tako sestavljen iz sedmih sinusoid s frekvencami, ki ustrezajo prvim sedmim členom harmonične vrste in amplitudami, prikazanimi na sliki 3.3. Del dobljenega signala je prikazan na sliki 3.4.

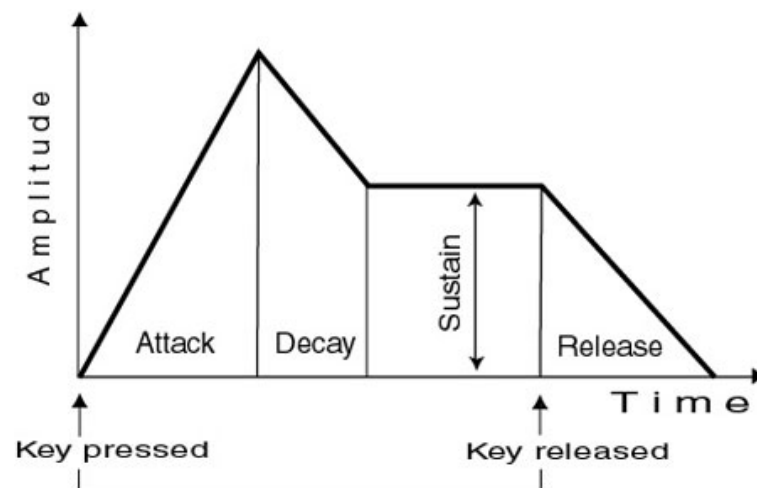
Ovojnica ADSR

Poleg glavnih frekvenc in njihovih amplitud barvo inštrumenta določa tudi krivulja amplitude v odvisnosti od časa, imenovana amplitudna ovojnica. To lahko vidimo, če primerjamo valovanja tona, zaigranega z različnimi inštrumenti. V osnovi ima vsaka amplitudna ovojnica štiri faze, katerih začetnice dajejo ime ovojnici: rast (angl. *attack*), padec (angl. *decay*), ohranjanje (angl. *sustain*) in spust (angl. *release*). Faze so prikazane na sliki 3.5. Klavir ima na primer hitro rast, nekoliko počasnejši padec, nato pa amplituda zelo počasi pada proti nič.

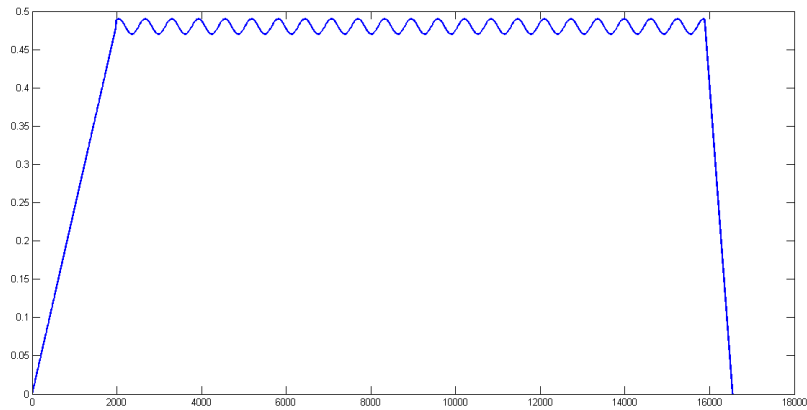
Ovojnica ADSR pihalnih inštrumentov nima faze padca, zato njena oblika spominja na trapez. Ker pri pihalih in trobilih izvajalci pogosto uporabljajo



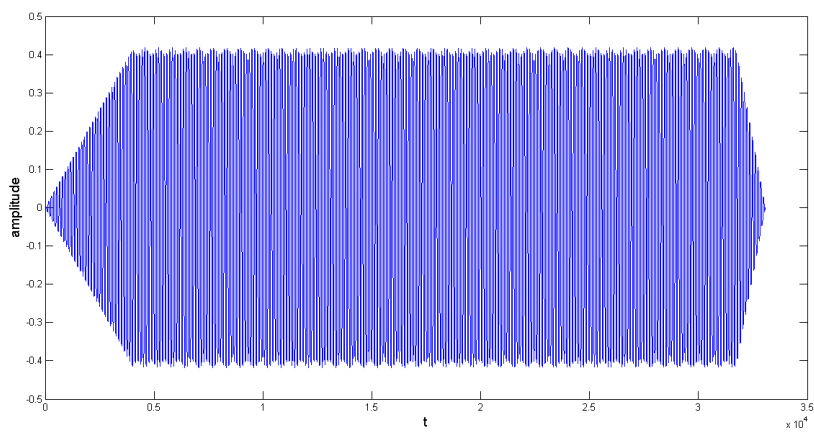
Slika 3.4: Signal tona klarineta, generiran kot vsota sedmih harmonij, pri čemer je fundamentalna harmonija 220Hz (ton A3).



Slika 3.5: Splošni model ovojnice ADSR [12]



Slika 3.6: Ovojnica ADSR klarineta, uporabljena pri sintezi.



Slika 3.7: Valovanje signala s frekvenco 349.2 (ton F4) po skalarnem produktu z ovojnico ADSR.

efekt *tremolo* (rahlo nihanje v amplitudi), smo v fazi ohranjanja namesto premice generirali sinusoido. Ovojnica ADSR, ki smo jo na koncu uporabili pri sintezi klarineta, je prikazana na sliki 3.6, valovanje signala tona po skalarnem produktu z ovojnico ADSR pa na sliki 3.7.

3.3 Izbira značilk za sintezo in metode za njihovo uporabo

Iz opisa načina pridobivanja značilk v poglavju 2 je moč opaziti, da so nekateri med njimi precej kompleksni, njihova uporaba pri sintezi pa bi bila posledično še težja. Prav tako jih veliko računa podobne lastnosti. Zato smo se odločili uporabiti le nekatere značilke. Podrobneje smo raziskali Setharesovo metodo za izračun senzorične disonance [4] in jo poskusili uporabiti pri sintezi. To se je izkazalo za nekoliko težjo nalogo zaradi izbiranja parametrov Setharesove formule, ki ga izvaja sistem MIRToolbox, in na žalost nam je za njegovo obravnavo zmanjkalo časa. Značilke torej nismo uspeli uporabiti pri posnemanju skladb, kljub temu pa smo jo uporabili pri sintezi, kjer smo njeno vrednost v naših enotah podali kot parameter programu. Značilka funkcije *mirregularity* ocenjuje barvo signala. Ker je ta že določena z inštrumentom, uporabljenim pri sintezi, nam značilke ni bilo potrebno uporabiti. Kljub temu nam je dala idejo za še eno značilko, čigar vrednost bi podali kot parameter. Namesto računanja razlik med amplitudami zaporednih frekvenc v spektru ta značilka na podlagi vhodne vrednosti izračuna, kolikšno je lahko odstopanje amplitud not od povprečne amplitude sintetizirane skladbe. Zaradi tega smo značilko poimenovali dinamika zvoka. Značilki čas in naklon faze naraščanja amplitude govorita o fazi rasti v ovojnici ADSR, opisani v poglavju 3.2.2. Kot smo že omenili tam, ima vsak inštrument drugačno ovojnico ADSR, zato smo ti značilki uporabili pri izbiri inštrumenta.

V nadaljevanju so opisane metode, s katerimi smo značilke uporabili pri sintezi. Pri tem njihov vrstni red ustreza tistemu, ki ga uporablja program. Metode so namreč odvisne druga od druge, saj za uporabo določenih značilk

potrebujemo že znane rezultate obravnave nekaterih drugih.

3.3.1 Čas in naklon faze naraščanja amplitude

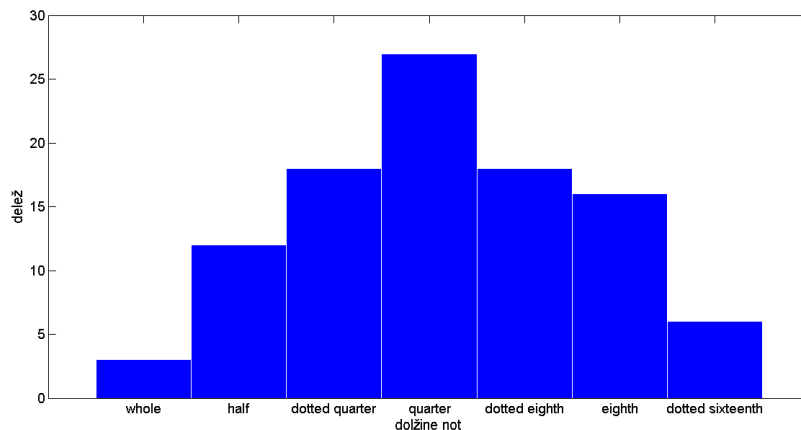
Značilki spadata med tiste, ki merijo barvo skladbe. Ta je odvisna od inštrumentov v skladbi, saj ima vsak inštrument svojo barvo, ki določa njegov prepoznaven zvok. Program vrednost značilk izračuna za nekaj klavirskih skladb in skladb s klarinetom, ki jih imamo shranjene v bazi. Nato izračuna njuni vrednosti za vhodno skladbo in s primerjavo parov značilk ugotovi, kateri inštrument se v njej uporablja. Vsak par obravnava kot vektor in uporabi evklidsko razdaljo, da izračuna, kateri vektor je najbolj podoben vektorju para značilk vhodne skladbe. Na podlagi tega program za sintezo izbere inštrument, ki ga uporablja skladba, najbolj podobna vhodni.

3.3.2 Tempo

Eden od potrebnih parametrov pri sintezi je tudi dolžina skladbe. Ker ta pravzaprav nič ne prispeva k podobnosti med skladbama, smo jo programu podali kot parameter v sekundah. Vsaka sintetizirana skladba ima frekvenco vzorčenja 44100 Hz, torej je njen vektor dimenzij 1×44100 , če traja eno sekundo. Tako na podlagi dolžine signala poznamo dolžino njegovega vektorja. S tem ko vemo kolikšen je tempo skladbe (izračunan z *mirtempo*), poznamo tudi dolžino vektorja note z določeno dolžino. Pri tem smo privzeli, da je mera udarci na minuto, v kateri je podan tempo, ekvivalentna meri četrтинke na minuto (angl. *quarters per minute*). To sicer velja za vse skladbe s četrтinskim taktom, ki se tudi najpogosteje pojavlja [15]. Rezultat izraza $\frac{60}{tempo}$ nam torej vrne dolžino četrтинke v sekundah, s tem pa poznamo dolžino vseh not.

Vprašanje, ki se nam je nato zastavilo, je bilo, katere in koliko not z določeno dolžino uporabiti. Tukaj smo za vsako dolžino vnaprej določili njen delež izmed vseh not v skladbi. Deleži so prikazani na sliki 3.8.

Dolžine imajo tako približno Gaussovo porazdelitev s četrтinko kot pov-

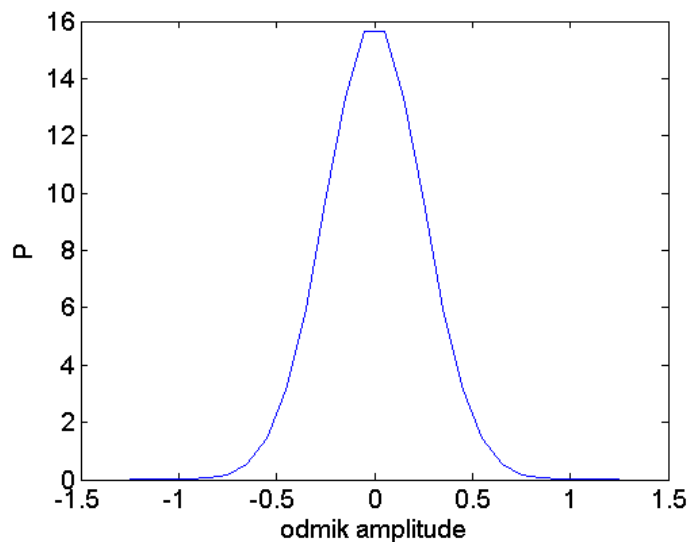


Slika 3.8: Deleži dolžin not v sintetizirani skladbi.

prečno in najbolj pogosto noto. Za četrtniko smo se odločili ravno zaradi predpostavljene ekvivalence obeh mer. Glede na dolžino skladbe in tempo program na podlagi deležev določi, kolikokrat se vsaka dolžina note pojavi. Pri tem vsota njihovih vektorjev lahko nekoliko odstopa od dolžine vektorja skladbe s podano dolžino v sekundah, zato je sintetizirana skladba lahko za kakšno sekundo krajša ali daljša od dolžine, podane programu.

3.3.3 Povprečna amplituda in dinamika

Ko je enkrat znano zaporedje dolžin not, program lahko za vsako od njih izračuna amplitudo, s katero bo zaigran. Če namreč hočemo, da je skladba dinamična, moramo pri tem upoštevati, da imajo take skladbe na nekaterih notah večje in na drugih manjše poudarke. Koliko bo skladba glasna in koliko dinamična, je odvisno od značilne povprečne amplitude in dinamike. Amplitudo note smo modelirali kot slučajno spremenljivko z normalno porazdelitvijo s parametroma $\mu = \text{povprečna amplituda}$ in $\sigma = \frac{1}{\text{dinamika}}$. Večjo vrednost ima torej parameter *dinamika*, manjši je standardni odklon in s tem verjetnost, da bo amplituda drugačna od njenega povprečja. Če je vrednost parametra enaka 0, imajo vse note enako, povprečno amplitudo, kar pomeni,



Slika 3.9: Gaussova porazdelitev za odmik amplitude od njene povprečne vrednosti pri parametru *dinamika* = 4. Odmik je kasneje zmanjšan za faktor 10, tako da se z ignoriranjem verjetnosti, manjših od 0,01, in pri povprečni amplitudi z vrednostjo 0,3 amplituda vedno giblje med 0,24 in 0,36.

da je skladba z vidika glasnosti monotona. Za vsako noto program generira naključno število med 1 in 100 in poišče prvi indeks v vektorju kumulativne normalne porazdelitve, pri katerem je vrednost vektorja večja ali enaka generiranemu številu. Indeks uporabi pri vektorju normalne porazdelitve, da poišče amplitudo note. Na sliki 3.9 je prikazan primer normalne porazdelitve s parametroma *povprečna amplituda* = 0.3 in *dinamika* = 4.

Ko program določi noti višino tona, jo lahko sintetizira s klavirjem oziroma pihalom. Takrat ima nota že neko povprečno amplitudo. Pri klavirju je to povprečna amplituda vektorja, dobljenega s pomočjo faznega kodirnika glasu, pri pihalu pa od amplitudne ovojnice. Zato program najprej izračuna povprečno amplitudo sintetizirane note, in z njeno vrednostjo deli vektor note. Tako ima nota novo povprečno amplitudo, v vrednosti približno 1. Na koncu program pomnoži vektor note s pripadajočo amplitudo.

ton	verjetnost za pojavitev
tonika	0.23
ton lestvice	0.12
ton izven lestvice	0.01

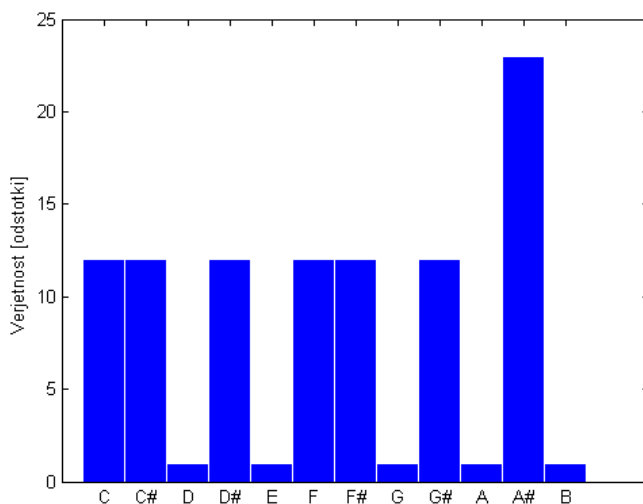
Tabela 3.1: Verjetnosti glede na ključ in pripadajočo lestvico

3.3.4 Ključ

Kot smo ugotovili pri razvrščanju skladb na podlagi njihovih značilk, je ključ najbolj informativna značilka. Iz glasbene teorije vemo, da je vsaka glasbena lestvica odvisna od ključa, ki ga sestavljata tonika in modalnost, na primer C dur. Tonika je prvi in glavni ton v lestvici, nadaljne zaporedje pa je odvisno od modalnosti. Durova lestvica je določena z zaporedjem intervalov W-W-H-W-W-W-H, kjer je W cel ton in H polton, (naravna) molova lestvica pa z zaporedjem intervalov W-H-W-W-H-W-W. Zelo podrobna razlaga tonalne glasbe je na voljo na [13]. Na podlagi ključa program določi verjetnosti za vsak ton, navedene v tabeli 3.1. Na sliki 3.10 je prikazana porazdelitev glede na ključ A mol. Porazdelitve so zelo podobne tistim, ki jih dobimo s funkcijo *mirchromagram*. V večini primerov se skladbe vrtijo okoli tonike in upoštevajo lestvico. Nekoliko bolj atonalne skladbe imajo sicer lahko bolj enakomerno porazdelitev, mi smo se pa kljub temu odločili, da program vedno sintetizira tonalne skladbe.

Program nato izračuna kumulativno porazdelitev, ki jo uporabi pri izbiri tona. Za vsako noto iz seznama, ki smo ga dobili pri določanju dolžin, naključno izbere število v intervalu $[1,100]$, ki ga uporabi tako, da poišče prvi indeks kumulativne porazdelitve (vektor z dvanajstimi elementi, pri čemer prvi element ustreza tonu E, zadnji pa D), ki ima vrednost večjo ali enako naključnemu številu.

Pri daljših skladbah lahko pride do menjave ključa, zato program s funkcijo *mirkeystrength* izračuna njihove verjetnosti in izbere dva najboljša ključa. Skladba je sintetizirana tako, da najprej uporablja najboljši ključ, v drugi

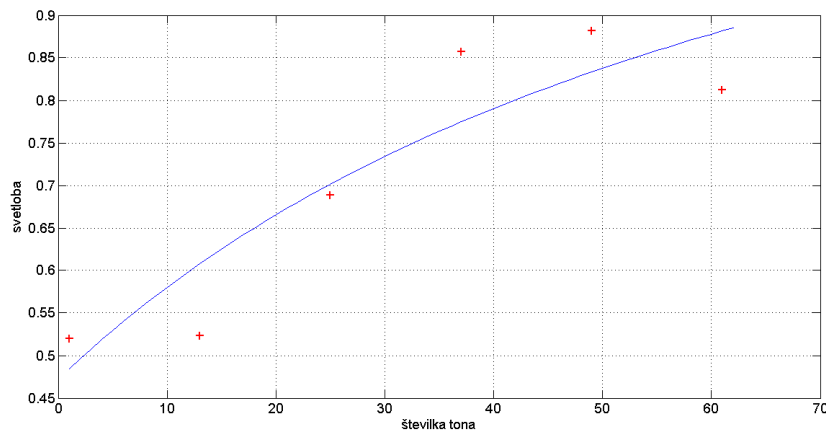


Slika 3.10: Porazdelitev verjetnosti pojavitve tona pri sintezi skladbe v A molu.

polovici pa drugega.

3.3.5 Svêtlost

Ko smo določili verjetnosti za vsak razred tonov v okviru ene oktave, je bil naslednji korak izbrati interval, iz katerega bi program naključno izbral višino tona za noto. Tukaj smo se omejili na interval 25 tonov, torej dve oktavi, čeprav bi lahko sintetizator implementirali tudi za več oktav. Razlog za to je samo večja verjetnost, da bo tako sintetizirana skladba imela več smisla in bila bolj poslušljiva. Interval je bil izbran glede na svêtlost skladbe in inštrument, uporabljen pri sintezi. Za nižje tone namreč velja, da so sestavljeni iz manjšega deleža sinusoid s frekvenco, ki je višja od določene frekvence. Za slednjo smo določili vrednost 370 Hz (ton F4♯), ki smo jo zato tudi predhodno uporabili pri ugotavljanju svêtlosti vhodne skladbe s funkcijo *mirbrightness*. Ker je spekter frekvenc drugačen pri vsakem inštrumentu, je svêtlost skladbe odvisna tudi od uporabljenega inštrumenta. Program za

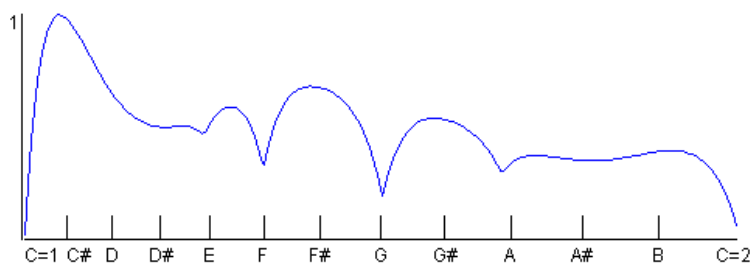


Slika 3.11: Aproximacija svêtlosti tonov z Gauss-Newtonovo metodo. Z rdečimi križci so označene svêtlosti tonov A iz prvih šestih oktav.

vsak interval izračuna svêtlost tako, da svêtlost vsakega tona v intervalu obteži z njegovo verjetnostjo pojavitve. Ko najde interval s svêtlostjo, katere vrednost je najbližje svêtlosti vhodne skladbe, preneha z iskanjem.

Pri klavirju bi morali ugotoviti svêtlost vsakega tona s pomočjo funkcije *mirbrightness*. Ker ta za vhod ne sprejema vektorja, temveč samo glasbene datoteke, bi morali najprej vsakič vektor shraniti v formatu *wav*, kar pa ni prišlo v poštev. Druga možnost bi bila že predhodno izračunati svêtlost za vsak ton posebej in vektor vrednosti uporabiti v programu. Odločili smo se za izračun svêtlosti šestih tonov A, ostale vrednosti pa smo aproksimirali z Gauss-Newtonovo metodo za problem najmanjših kvadratov (slika 3.11).

Ker smo klarinet sintetizirali sami, smo za vsak ton vedeli, kakšen spekter ima, zato smo njegovo svêtlost lahko izračunali brez funkcije *mirbrightness*. Na podlagi višine tona smo namreč vedeli, kakšne so njena fundamentalna frekvenca in pripadajoče harmonije, amplitude frekvenc pa so za vse tone enake.



Slika 3.12: Graf senzorične disonance med baznim tonom A3 (220Hz) in naslednjimi dvanajstimi toni, kot ga prikaže java applet [14]. Amplitude ustrezajo srednje močnem udarcu klavirskih tipk. Kot bazni ton je na abscisni osi vedno prikazan ton C.

3.3.6 Disonanca

Kot smo že omenili, vrednosti disonance nismo pridobili iz vhodne skladbe, temveč smo jo predhodno podali programu kot parameter. Pri tem smo uporabili Setharesovo formulo, ki se glasi:

$$d(f, g, A_f, A_g) = A_f A_g [e^{-0.84Q} - e^{-1.38Q}], \quad (3.2)$$

kjer velja:

$$Q = \frac{g - f}{0.021f + 19}. \quad (3.3)$$

f in g sta dani frekvenci, A_f in A_g pa njuni pripadajoči amplitudi. Razlika eksponentnih funkcij nam da Plompovo in Leveltovo funkcijo, prikazano na sliki 2.4. Tako kot smo že omenili, je torej disonanca dveh sinusoid oziroma čistih tonov enaka produktu njunih amplitud, uteženem s Plompovo in Leveltovo funkcijo. Za harmonične tone izračunamo senzorično disonanco tako, da seštejemo disonance med vsemi pari harmonij: za vsako harmonijo prvega tona izračunamo disonanco med njo in vsako harmonijo drugega tona. Pri tem frekvenca g v enačbi 3.2 vedno ustreza višji od obeh frekvenc. Za oba tona podamo vektor amplitud njunih harmonij. Na [14] se nahaja java applet, ki za bazni ton f prikaže graf senzorične disonance v odvisnosti of g . Primer je prikazan na sliki 3.12.

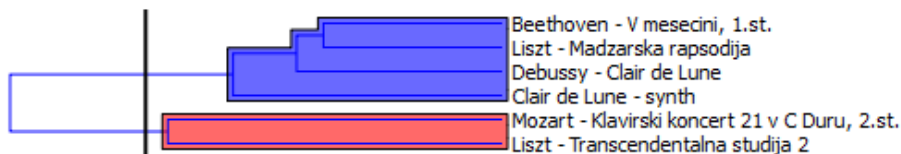
Program smo implementirali tako, da nekaj odstotkom not, ki že imajo določeno dolžino in višino tona, doda še eno oziroma dve noti. V prvem primeru dobimo diado, v drugem triado. Med vsemi toni v triadi je interval vedno enak in je odvisen od ključa. Če je skladba v duru, sestavljajo triado tonika, velika terca in čista kvinta, ki pa niso nujno iz iste oktave. Na primer za ključ v C duru triado sestavljajo toni C, E in G. Pri skladbi v molu je triada sestavljena iz tonike, male terce in čiste kvinte (C, E \flat in G). Skladba je v splošnem sintetizirana tako, da je nekaj več kot polovica akordov diad, preostanek pa triad. Tudi za diade smo uporabljali tak par tonov, da je interval med njima vedno enak. Za določanje intervala smo uporabili Setharesovo funkcijo. Kot parameter uporabnik poda število n med 1 in 12, ki določa stopnjo disonance. Stopnja 1 pomeni najmanjšo vrednost (globalni minimum) Setharesove funkcije. Ta je na sliki 3.12 enaka oktavi, stopnja 12 pa ustreza globalnemu maksimumu in je na sliki 3.12 enaka poltonu. Ker se krivulja spreminja z višanjem frekvence baznega tona, smo za klavir n -ti minimum funkcije izračunali za dvanajst baznih tonov, ki ustrezajo prvim dvanajstim tonom v intervalu tonov, ki ga program uporablja za sintezo. Interval, ki je največkrat določen kot n -ti minimum, je nato uporabljen za diado. Pri klarinetu program izbira tone samo iz ene oktave, zato je funkcija izračunana samo enkrat.

Poglavje 4

Merjenje uspešnosti sinteze

Prvi test, katerega je moral naš program prestati, je primerjava vrednosti značilk, uporabljenih pri sintezi, z vrednostmi, ki jih MIRToolbox izračuna na sintetizirani skladbi. Vrednosti bi si morale biti med sabo čim bolj podobne. Za povprečno amplitudo, katere vrednost je preprosto izračunati, smo pričakovano dobili skoraj iste vrednosti. Manjša odstopanja so možna in odvisna od značilke *dinamika*. Presenetljivo dobre rezultate smo dobili za tempo skladbe. Vrednosti značilke v obeh skladbah sta skoraj enaki, relativna napaka je namreč manjša od 3%. Rezultati so nas pozitivno presenetili zaradi preprostega načina uporabe značilke. Z njeno vrednostjo smo namreč samo določili dolžino četrтинke pri predpostavki, da sta enoti *bpm* in *qpm* ekvivalentni. Očitno je bila odločitev, da bo četrтинka najpogostejša in tudi povprečna dolžina note v skladbi, prava. Ker smo svétlost tonov aproksimirali z Gauss-Newtonovo metodo in višine tonov modelirali kot slučajno spremenljivko, skladba nima natanko enake svétlosti, temveč v večini primerov nekoliko odstopa. V povprečju se je izkazalo, da je relativna napaka največ 10%. Najbolj verjetna ključa, ki ju izračuna MIRToolbox na sintetizirani skladbi, sta vedno tistima, ki sta bila podana za sintezo. Za več kot četrтинo lahko odstopajo verjetnosti nekaterih izmed ostalih ključev, vendar je takih le približno sedmina.

Naslednja stvar, ki smo jo hoteli preveriti, je, kam sintetizirane skladbe



Slika 4.1: Razvrščanje klavirskih skladb v skupine. Razdalja med Debussyjevo Clair de Lune in sintetizirano skladbo je večja od razdalje od Beethovneve in Lisztove skladbe.

skladba	Clair de lune - synth
Debussy - Clair de lune	47 107
Beethoven - V mesečini, 1. stavek	53 637
Beethoven - Patetična, 2. stavek	55 755
Liszt - Transcendentalna študija 2	61 177
Debussy - Reverie	65 032
Liszt - Madžarska rapsodija	71 976

Tabela 4.1: Razlika med vektorjem skladbe Clair de Lune - synth in vektorji ostalih skladb.

uvrsti hierarhično razvrščanje. Ker pri sintezi posnemamo neko skladbo, želimo, da je njuna razdalja med vrednostmi značilnik čim manjša. Če bi bila razdalja najmanjša, bi izpolnili zadan cilj, saj bi to pomenilo, da bi kakšna aplikacija, ki uporablja hierarhično razvrščanje, po poslušanju Debussyjeve Clair de Lune kot prvi predlog podala njen sintetiziran par. Na sliki 4.1 je prikazan primer razvrščanja petih klavirskih skladb, ki smo jih uporabili tudi v poglavju 2 in ene sintetizirane skladbe. Vidimo, da razdalja parov skladb ni najmanjša, zato razvrščanje na podlagi značilnik ni uspešno.

Namesto značilnik lahko za razdaljo uporabimo tudi vektorje signalov. Primerjamo torej njihova valovanja. Ker ima amplituda signala tukaj veliko vlogo, smo vsak vektor skladbe delili z njeno povprečno amplitudo in množili

s povprečno amplitudo sintetizirane skladbe. V tabeli 4.1 je prikazana absolutna razlika med vektorjem sintetizirane skladbe Clair de lune - synth in nekaterimi drugimi skladbami. Vidimo, da imata najmanjšo razdaljo ravno Debussyjeva Clair de Lune in njen sintetiziran par. Primerjava signalov nam da torej bolj vzpodbudne rezultate, vendar bi meritve morali opraviti na dosti večji bazi, da bi se lahko prepričali o uspešnosti sinteze na podlagi takšnega primerjanja.

Poglavje 5

Sklepne ugotovitve

Nepričakovano veliko sintetiziranih skladb, ki jih je program ustvaril, je poslušljivih in smiselnih. To je najbrž posledica zelo majhne svobode, ki smo jo pri sintezi pustili programu. Omejili smo se namreč na tone iz lestvice, triade so vedno sestavljene iz istih treh tonov (ki pa so lahko iz različnih oktav), diade pa vedno iz tonov, oddaljenih za enak interval. Nesmisel skladbe najpogosteje povzroči zaporedje dolžin tonov, ki je lahko včasih takšno, da nikakor ne moremo vsaj približno ujeti ritma skladbe. Dosti skladb je v posameznih delih tako dobro ustvarjenih, da bi lahko skladateljem oziroma na splošno ljudem iz sveta glasbe dali idejo za svojo skladbo, ki bi jo nadalje lahko sami razvijali. Uporabnik bi določil hitrost, ključ, disonanco itd. ter s tem podal smer, v kateri naj program poišče možno idejo. Sam bi ocenil, da je uporaba na tem področju dosti bolj koristna, za bolj uspešno sintezo s posnemanjem skladbe pa bi bilo potrebno nadaljnje raziskovanje v tej smeri.

Kaj bi lahko izboljšali? Značilke, ki smo jih pri sintezi uporabili, so imele vse globalno vrednost. MIRToolbox omogoča razdelitev skladbe na okvirje enakih dolžin (pri čemer lahko dolžino poljubno določimo) in izračun značilke za vsak posamezen okvir. S tem dobimo graf vrednosti značilke skozi čas. Sintezo bi potem lahko naredili na takšen način, da bi sintetizirali vsak okvir posebej in okvirje na koncu povezali skupaj. Tako bi recimo pri posnemanju skladbe s tihim uvodom in zelo glasnim nadaljevanjem zagotovili, da bi

sintetizirana skladba imela tudi take lastnosti. Naša implementacija sicer omogoča lokalno spreminjanje amplitude, ampak ta še vedno variira okoli svojega povprečja.

Spregledali smo dvanajst značilk, dobljenih s funkcijo *mirchromagram*. Spomnimo, da funkcija izračuna porazdelitev signala vzdolž dvanajstih tonov na podlagi magnitud frekvenc. Porazdelitev smo sicer aproksimirali s svojo porazdelitvijo tonov, ki pa ni nujno vedno najboljša. Izračunane vrednosti značilk bi lahko tako uporabili kot verjetnosti za vsak posamezen ton. Pri tem bi morali najprej raziskati, če se verjetnosti resnično ujemaajo z deležem tonov v skladbah. Vsak ton je namreč sestavljen iz več harmonij. Tako ton C3# s frekvenco 138.6 Hz ne poveča verjetnosti samo tonu C#, temveč tudi tonu G#, ker ima ton G4# frekvenco 415 Hz, ta pa je tretja v harmonični vrsti. Problem bi lahko torej nastal pri skladbah z inštrumenti, pri katerih ima namesto fundamentalne frekvence večjo magnitudo kakšna druga frekvenca v harmonični vrsti. MIRTtoolboxovo funkcijo bi lahko uporabili tudi tako, da se ne bi omejili na porazdelitev signala po dvanajstih razredih tonov, temveč po tonih v intervalu več oktav. Tako bi z upoštevanjem svetlosti lahko bolj natančno določili delež oktav v skladbi. Senzorične disonance - kljub manjši informativnosti ene zanimivejših značilk - nam ni uspelo uporabiti pri posne-manju skladb, zato bi lahko še bolj poglobljeno raziskali, kako MIRTtoolbox računa njeno vrednost in kako bi ta podatek uporabili pri sintezi.

Literatura

- [1] MIRtoolbox — Humanistinen tiedekunta. Dostopno na:
<https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox>

- [2] Potential Mathematical Models for the Western Musical Scale. Dostopno na:
<http://www.skytopia.com/music/theory/scale-dissertation.html#Plomp>

- [3] Consonance and dissonance - Wikipedia, the free encyclopedia. Dostopno na:
<http://en.wikipedia.org/wiki/Dissonant#Dissonance>

- [4] W. A. Sethares, *Tuning, Timbre, Spectrum, Scale*, Springer-Verlag, 1998.

- [5] K. Jensen (1999), Timbre Models of Musical Sounds. Dostopno na:
<http://vbn.aau.dk/files/46619185/TMoMS.pdf>

- [6] Orange – Data Mining Fruitful & Fun. Dostopno na:
<http://orange.biolab.si/>

- [7] Jens Johansson ~ The Phase Vocoder: A Tutorial. Dostopno na:
<http://www.panix.com/~jens/pvoc-dolson.par>

- [8] University of Iowa Electronic Music Studios. Dostopno na:
<http://theremin.music.uiowa.edu/MISpiano.html>

- [9] A Phase Vocoder in Matlab. Dostopno na:
<http://labrosa.ee.columbia.edu/matlab/pvoc/>
- [10] What is a Sound Spectrum? Dostopno na:
<http://www.phys.unsw.edu.au/jw/sound.spectrum.html>
- [11] Synth Secrets, Part 1: What's In A Sound? Dostopno na:
<http://www.soundonsound.com/sos/may99/articles/synthsec.htm>
- [12] Synthesizer tutorial || Stage 3. Dostopno na:
<http://www.ico.mmu.ac.uk/09476866/tutorial/stage3.html>
- [13] A. Cadwallar, D. Gagné, *Analysis of Tonal Music: A Schenkerian Approach, Second Edition*, Oxford University Press, 2007.
- [14] Dissonance Curve Applet. Dostopno na:
<http://jjensen.org/DissonanceCurve.html>
- [15] Meter and Rhythm : Musical Tempo : How Music Works. Dostopno na:
<http://www.howmusicworks.org/509/Meter-and-Rhythm/Musical-Tempo>
- [16] Dissonance Theory. Dostopno na:
http://davide.gipibird.net/A_folders/Theory/t_dissonance.html