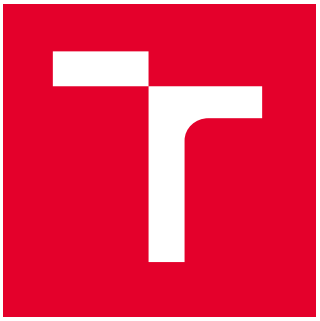
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2023 Dominik Poloček



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

ANALYZÁTOR AKORDŮ KLAVÍRU

PIANO CHORD ANALYZER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE — **Dominik Poloček**

AUTHOR

VEDOUCÍ PRÁCE — **Ing. Matěj Ištvánek**

SUPERVISOR

BRNO 2023

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Audio inženýrství**
specializace Zvuková produkce a nahrávání
Ústav telekomunikací

Student: Dominik Poloček ***ID:*** 230306
Ročník: 3 ***Akademický rok:*** 2022/23

NÁZEV TÉMATU:

Analyzátor akordů klavíru

POKYNY PRO VYPRACOVÁNÍ:

Prozkoumejte problematiku určení výšky jednoho i vícera simultánních tónů u nahrávek klavíru. Implementujte systém v jazyce Python, který určí výšku zahraných tónů podle rovnoměrně temperovaného ladění a následně určí, o jaký akord se jedná. Systém bude umožňovat také přepínání mezi různými metodami a rozpoznávání akordů v reálném čase při použití mikrofonu. Cílem semestrální práce je popis metod určení výšky jednoho tónu a tónů v polyfonické struktuře se zaměřením na klavírní nahrávky. Dalším cílem je vytvoření skriptů pro určení výšky tónů a základní implementace alespoň jedné metody pro určení typu zahraného akordu. V navazující bakalářské práci bude systém rozšířen o další metody, možnost použití v reálném čase a vyhodnocení na vlastních klavírních nahrávkách.

DOPORUČENÁ LITERATURA:

[1] MÜLLER, Meinard. Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications. Cham: Springer International Publishing, 2015. ISBN 978-3-319-21945-5.
[2] JIANG, Nanzhu, GROSHE, Peter, KONZ, Verena a MÜLLER, Meinard. Analyzing Chroma Feature Types for Automated Chord Recognition. In: Proceedings of the AES Conference on Semantic Audio, Ilmenau, Německo, 2011.

Termín zadání: 6.2.2023 ***Termín odevzdání:*** 26.5.2023

Vedoucí práce: Ing. Matěj Ištvánek

doc. Ing. Jiří Schimmel, Ph.D.
předseda rady studijního programu

ABSTRAKT

Předložená práce se věnuje analýze akordů pomocí určování kmitočtů jejich komponentů. Cílem práce je nastínit metody pro určování základních kmitočtů jednoho a vícera tónů a implementovat systém, který dokáže s jejich použitím akordy určovat. Metoda implementovaná v jazyce Python (metoda spektrálních špiček) využívá rychlou Fourierovu transformaci pro zobrazení signálu v kmitočtové rovině a poté hledá spektrální maxima, která po patřičné kontrole vyhodnocuje jako základní kmitočty. Metoda spektrálních špiček byla srovnána s metodou sčítání modulů harmonických složek a se *state-of-the-art* systémem pro přepis nahrávky do MIDI (PianoTranscription) pomocí testů na datasetu vytvořeném pro tuto práci (530 nahrávek akordů a tónů). Nejlepší výsledky prezentuje PianoTranscription ($Accuracy = 0,74, E_{\text{tot}} = 0,23$), druhou nejúspěšnější metodou je metoda spektrálních špiček se známým počtem tónů ($Accuracy = 0,55, E_{\text{tot}} = 0,29$), poté tatáž metoda s neznámým počtem tónů ($Accuracy = 0,52, E_{\text{tot}} = 0,38$) a na konec metoda sčítání modulů harmonických složek ($Accuracy = 0,26, E_{\text{tot}} = 0,81$). Limitací implementovaného systému je neschopnost určit počet tónů (musí být zadán uživatelem) a frekvenční minimum (138,59 Hz), pod kterým jsou odhady chybné, a které je pravděpodobně způsobeno konstrukcí klavíru a opředením některých strun.

KLÍČOVÁ SLOVA

Analyzátor akordů, klavírní nahrávky, Music Information Retrieval, spektrum akordu, určení výšky tónů.

ABSTRACT

The presented thesis deals with the analysis of chords by determining the frequencies of their components. The aim of thesis is to outline methods for determining the fundamental frequencies of single and multiple notes and to implement a system that can determine chords using these methods. The method, implemented in Python (spectral peak method), uses a fast Fourier transform to represent the signal in the frequency domain and then searches for spectral maxima, which it evaluates as fundamental frequencies after proper checking. The spectral peaks method was compared with the harmonic component modulus summation method and with the *state-of-the-art* system for transcribing recordings to MIDI (PianoTransctiprion) by running tests on the dataset created for this thesis (530 chord and note recordings). The best results are presented by PianoTranscription ($Accuracy = 0.74, E_{\text{tot}} = 0.23$), the second best performing method is the spectral peaks method with a known number of tones ($Accuracy = 0.55, E_{\text{tot}} = 0.29$), followed by the same method with unknown number of tones ($Accuracy = 0.52, E_{\text{tot}} = 0.38$) and finally the harmonic component modulus summation method ($Accuracy = 0.26, E_{\text{tot}} = 0.81$). The limitations of the implemented system are the inability to determine the number of tones (must be specified by the user) and the frequency minimum (138.59 Hz), below which the estimates are erroneous, which is probably due to the design of the piano and the braiding of strings.

KEYWORDS

Chord analyzer, chord spectrum, multipitch estimation, Music Information Retrieval, piano recordings.

Vysázeno pomocí balíčku `thesis` verze 4.07; `http://latex.feec.vutbr.cz`

POLOČEK, Dominik. *Analyzátor akordů klavíru*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2023, 65 s. Bakalářská práce. Vedoucí práce: Ing. Matěj Ištvánek

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Dominik Poloček
VUT ID autora:	230306
Typ práce:	Bakalářská práce
Akademický rok:	2022/23
Téma závěrečné práce:	Analyzátor akordů klavíru

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu semestrální práce panu Ing. Matějovi Ištvánkovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Rád bych poděkoval Mgr. Tomaszowi Michnikovi za zprostředkování nahrávací techniky.

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Tato bakalářská práce spadá do oboru „MIR“ (*Music Information Retrieval*), jehož cílem je extrakce informací různého typu z nahrávek (tempo, tónina, sledování melodie, rytmus, metrum, hledání duplikátů atd.). Informace z nahrávek jsou získávány pomocí analýzy různých reprezentací signálu, strojového učení aj. Tato práce se zabývá problémem určování komponentů akordu na základě digitální nahrávky pomocí analýzy signálu v časové, kmitočtové a kmitočtově-časové oblsati. Neuronová síť je zde použita mj. pro srovnávací účely. Práce prezentuje analýzu akordů pomocí určení základních kmitočtů tónů, které znějí v nahrávce a následného vyhodnocení obratu a umístění akordu na klávesnici klavíru. Proces určování výšek tónů v polyfonické struktuře se v MIR nazývá *Multiple f0 estimation*. Výstupem tohoto procesu je výška každého tónu a doba, po kterou tón zní. V této práci pracujeme s akordy, kde se kmitočty jednotlivých tónů v čase nemění (stacionární signál), proto zde zaniká požadavek časového údaje.

Určování akordů se také věnuje aktivita *Chord recognition* – rozpoznávání akordů. Na akord se v tomto případě nahlíží jako na chroma reprezentaci, která nezobrazuje kmitočty, pouze názvy tónů. Cílem této úlohy je pouze určit, jaký akord zní v daném čase nahrávky. Na obraty ani výšku jednotlivých tónů se zde nebere ohled. V této práci akord rozumíme jako tři/čtyti současně znějící tóny, jejichž kmitočty chceme určit.

Odhad základní frekvence jednoho nebo vícera tónů (*single f0 estimation, multipitch tracking*) nachází uplatnění v transkripčních systémech (automatický přepis nahrávky do MIDI), synchronizačních systémech (srovnání různých interpretací stejného díla, sledování notového zápisu v průběhu hry na nástroj), řečových analyzátorech a v neposlední řadě v digitálních ladičkách.

Cílem této práce je popis problematiky nalezení jednoho nebo více základních kmitočtů tónů v klavírní nahrávce se zaměřením na akordy. Cílem teoretické části je vedle popisu metod také objasnění signálových reprezentací. Cílem praktické části je implementace systému pro přesné určování akordů pomocí analýzy kmitočtů jednotlivých tónů. Mimo implementace je také cílem test a srovnání dvou metod pro určení výšky tónů v polyfonické struktuře a srovnání těchto metod s jedním ze *state-of-the-art* systémů, který pro určení výšek tónů používá neuronovou síť. Dalším cílem praktické části je tvorba datasetu.

Přehled náplně kapitol v textu: 1. kapitola popisuje klavír, akordy, notaci, ladění a dataset. Ve 2. kapitole jsou naznačeny použité reprezentace signálu. 3. kapitola uvádí některé metody pro určení výšky jednoho nebo vícera tónů. Ve 4. kapitole jsou popsány použité knihovny a 5. kapitola nabízí zmíněné srovnání metod. V závěru (kap. 6) najdeme shrnutí výsledků a návrhy pro pokračování práce.

1 Klavír

Klavír patří mezi nejpopulárnější hudební nástroje. Má velmi široké zastoupení ve většině hudebních stylů, a to jak v hudbě vážné, tak v hudbě populární. Počátky klavírní hudby můžeme hledat již v období baroka, a i dnes je klavír, nebo spíše jeho elektronický ekvivalent – klávesy (*stage piano*), nedílnou součástí nejedné popové formace. Pokud nebude uvedeno jinak, v tomto textu hovoříme o klasickém dřevěném klavíru s kovovými strunami.

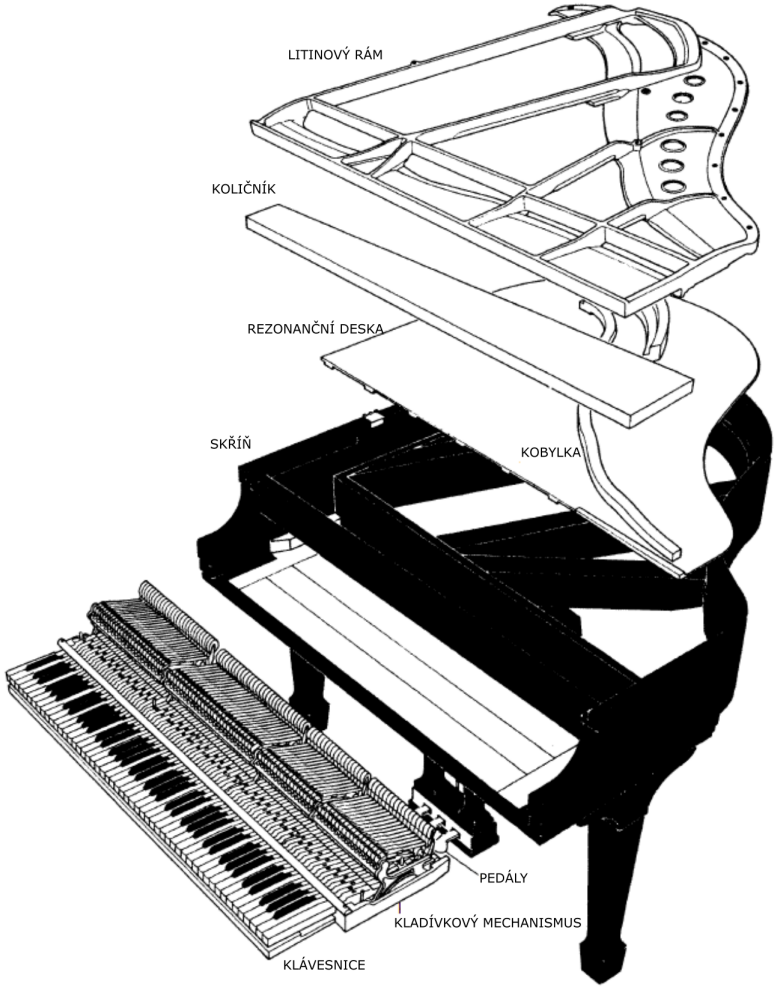
Klavír patří mezi strunné nástroje – *chordofony* úderné. Tón je tvořen úhozem kladívka do struny. Tónový rozsah standardního klavíru je subkontra A–C5. České názvosloví respektuje také zápis subkontra A–pětičárkované C. V této práci budeme používat obecnou MIDI notaci: identifikace not pomocí hodnot p (0–127, rozsah klavíru odpovídá hodnotám 21–108) a klavírní MIDI notaci: identifikace not pomocí čísla a písmena A0 – C8. Význam notace je popsán v kapitole 1.4. Standardní klavír disponuje 88 klapkami (52 bílými a 36 černými). Obvykle jsou na klavíru dva nebo tři pedály, které ovlivňují ADSR (zkratka od slov *attack* – nástup tónu, *decay* – útlum, *sustain* – podržení, *release* – uvolnění) obálku hraných tónů.
V dnešní době se běžně můžeme setkat s klavírem – křídlem či pianinem. Pianino je výrazně menší, má struny nataženy vertikálně a má 4 nohy. Klavír – koncertní křídlo má nohy pouze 3, struny jsou nataženy horizontálně a tvarem připomíná ptačí křídlo. Rozdíl je samozřejmě i ve spektru, o kterém hovoří kapitola 1.2.3.

1.1 Historie

Následující podkapitola čerpá informace primárně z knihy Pavla Kurfürsta: *Hudební nástroje* [1]. Motivací k vytvoření klávesových nástrojů byla snaha o zjednodušení hry na vícestrunné nástroje. Prvním předchůdcem klavíru je *klavichord*. Klavichord měl několik mosazných strun stejně dlouhých a stejně laděných. Struny se rozeznívaly kovovými jazýčky, jež po stisku klávesy udeřily do struny zespodu. Jazýček neměl jen rozeznívající funkci, ale také strunu zkracoval na patřičnou délku, čímž byla dosažena daná výška tónu. Pod strunami byla rezonanční deska z jedlového dřeva.

V druhé polovině 14. století vzniklo *cembalo (klavicembalo)*. Cembalo, na rozdíl od klavichordu, využívá strun různé délky, různého ladění a nerozeznívá je úderem, ale trsnutím zprvu havraního brku, později kouskem tuhé bůvolí kůže nebo kovu. Právě ono trsnutí způsobuje charakteristický cinkavý zvuk cembala. Délce strun je přizpůsoben tvar rezonanční desky, tj. tvar pravoúhlého trojúhelníka, kde upevnění strun kopíruje přeponu. Tak jako u klavichordu jsou na cembalu barvy klapek (černá/bílá) opačně než na klavíru. Později se na cembalu začalo struny zdvojovat nebo dokonce ztrojovat, přidávat další klávesnice a rejstříky.

V první třetině 18. století se výrobci za účelem redukce počtu klávesnic a zrušení rejstříků bez ztráty dynamiky vrátili k excitaci struny úderem. Teprve v 19. století nabyl klavír podoby, kterou známe dnes: skříň, litinový rám, ozvučná deska, struny natažené mezi količníkem a kobylkou, klávesová a pedálová mechanika. Tuto konstrukci klavíru znázorňuje obr. 1.1.



Obr. 1.1: Konstrukce klavíru, převzato z [2]

1.2 Tvorba tónu

Tón klavíru vzniká na struně. Excitátorem je v našem případě celý řetězec prst – klapka – kladívko. Ten rozezní strunu, která svou délkou, silou napnutí a objemovou hustotou definuje výšku tónu – oscilátor. Struna je upevněna ladícími kolíky na količníku, kterými se upravuje mechanické napětí struny, a závěsnými kolíky v závěsné podložce na litinovém rámu. Navíc, vlně na struně v cestě stojí kobylka,

jenž je připevněna k rezonanční desce a definuje délku aktivně kmitající struny. Primární funkcí kobylky je podle [3] podpora účinnějšího kmitání struny a interference. Za druhé, kobylka přenáší vibraci struny na rezonanční desku – rezonátor a zároveň radiátor. Má tedy identickou funkci jako kobylka na smyčcových nástrojích. O ozvučné desce tentýž zdroj tvrdí, že zpracovává kmity strun, reprodukuje, zesiluje a pro výslednou barvu tónu je klíčová.

1.2.1 Kladívkový mechanismus

Klapka funguje jako páka. Ta při zmáčknutí aktivuje mechanismus, který vymrští kladívko, to uhodí do struny, hned se vrací zpět a čeká na uvolnění klávesy. Současně s vymrštěním kladívka se zvedá dusítko, které předtím neaktivní strunu tlumilo. Uvolněním klávesy na strunu dusítko zpět doléhá a celý proces se může opakovat. Nejkratší struny dusítka nemají, protože prakticky nepřeznívají i když klávesa je stlačena. Trvání tohoto procesu je v řádech milisekund. Kladívkový mechanismus je detailně popsaný v [2].

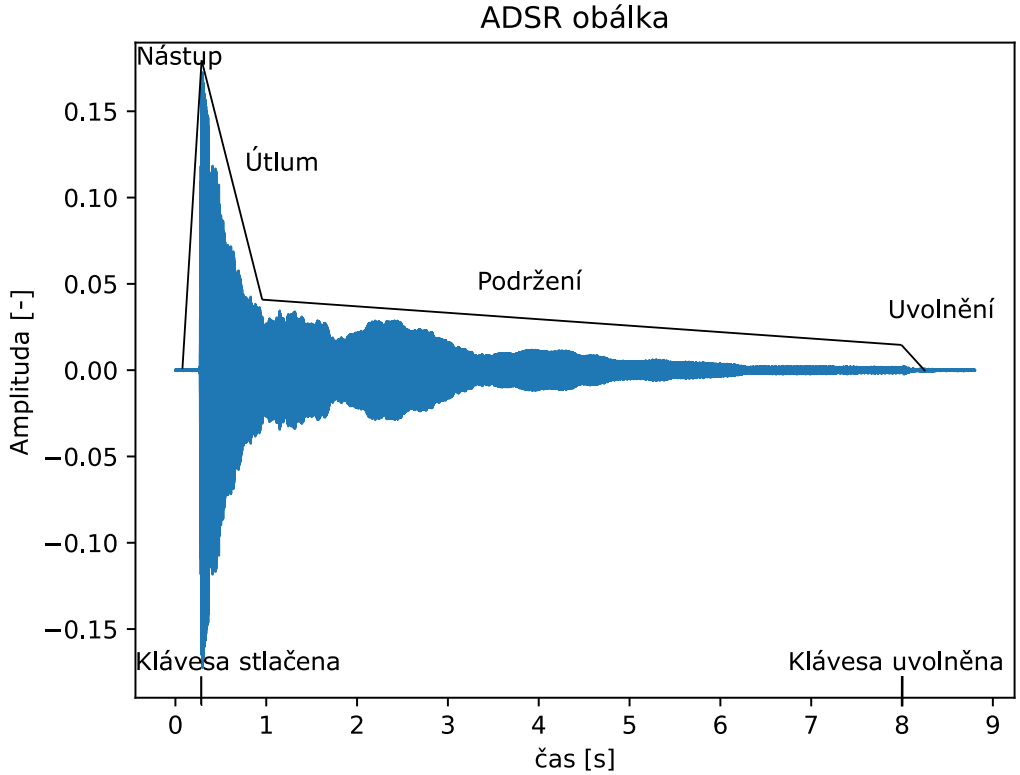
1.2.2 ADSR obálka

Každý tón lze rozdělit do 4 fází: *Attack, decay, sustain, release* (přeloženo výše). Hodnoty nástupu, útlumu a uvolnění se udávají v milisekundách a mají význam doby jednotlivých fází. Hodnota podržení se udává např. v decibelech nebo voltech a má význam hlasitosti tónu nebo amplitudy signálu (v některých aplikacích se nastavuje i čas). Každý tón má svoji ADSR obálku, a proto zvuky dvou různých nástrojů od sebe rozeznáme nejen pomocí rozdílné barvy, ale i podle ADSR obálky. Každý tón začíná fází nástupu. Mohou se zde vyskytovat slyšitelné transienty a inharmonické spektrální složky, které vnáší do zvuku zvonivost a perkusivitu. Ty se do dalších fází nepřenáší, protože jsou krátkodobé. Nástup může být krátký např. u bubnu, nebo dlouhý (u houslí se struna rozeznívá postupně podle doby tažení smyčce). Druhou fází, která udává čas od konce nástupu do začátku podržení je útlum. Útlum určuje dobu potřebnou pro ustálení tónu a odeznění transientů. Podržení je fáze, kde vnímáme výšku tónu, kdy je možné s tónem pracovat, harmonizovat jej atd. Fáze uvolnění udává čas od konce excitace nástroje do úplného zániku zvuku.

ADSR obálka u klavíru

Na ADSR obálku u klavíru mají největší vliv zejména kladívka a pedály, pokud jsou používány. Nástup je dosti krátký, jak vidíme na obrázku 1.2. Jsou zde perkusivní složky, které vznikají v důsledku úderu kladívka do struny. Právě kladívko

vnáší do spektra ve fázi nástupu inharmonicity. Kladívko je nejčastěji potaženo hustými a tvrdými vlněnými vlákny. Stáří vláken a jejich tvrdost perkusivitu ovlivňuje. Tím je hlasitost, přesněji vrchol nástupu vyšší než u jiných nástrojů. Na obrázku 1.2 je začátek fáze nástupu zakreslen v čase dříve, než začíná tón. Je to pouze z důvodu viditelnosti, fáze nástupu samozřejmě začíná se začátkem excitace struny. Doba podržení závisí pouze na hráči, protože končí spadnutím dusítka na strunu, a to se děje uvolněním klávesy. Občas se stává, že dopad dusítka na strunu je slyšitelný. To se může projevit ve spektru a způsobit chybu v identifikaci tónu. V případě klavíru excitace struny trvá jen zlomek vteřiny. Zde se pro fázi podržení rozumí čas, kdy je klávesa stlačena a dusítko zdviženo. Uvolnění může být u klavírů různě dlouhé podle velikosti ozvučné skříně, stáří strun, použití pedálů atd. Doba uvolnění je tak dlouhá, jak dlouho po uvolnění klávesy vibruje rezonanční deska a jak dlouho přeznívají jiné struny zpětně deskou rozvibrované. Přeznívající a zpětně rozvibrované struny mohou také vést k nepřesným výsledkům odhadu základní frekvence tónu.



Obr. 1.2: ADSR obálka tónu C4, bez použití pedálů

ADSR obálku klavíru a také jeho spektrum výrazně ovlivňují pedály. Obvykle jsou dva se striktně určenou funkcí. Pravý pedál, označovaný jako *sustain* zvedá všechna dusítka ze strun, takže po uvolnění klávesy struny nejsou zatlumeny. Fáze podržení je tedy výrazně prodloužena a do spektra přibývají výše zmíněné harmonické složky zpětně rozezněných strun rezonanční deskou. Pokud je sustain pedál aktivní, přechod z fáze podržení do uvolnění je plynulý, ne skokový.
Levý pedál ztišuje celou hru buď posunem kláves a kladívkového mechanismu doprava, nebo přibližuje kladívka ke strunám. U zdvojených strun kladívko uhodí pouze do jedné z nich a u ztrojených do dvou. Celková hlasitost se snižuje. Pokud má levý pedál (*una corda, sostenuto*) druhou funkci z uvedených, kladívka se přiblíží strunám a mají kratší čas na nabrání rychlosti, čímž udeří slaběji.
Třetí pedál, pokud je vestavěný, je umístěn mezi dvěma uvedenými a není určeno jakou má mít funkci. V některých případech aktivuje plsťovou lištu, která přiléhá na struny. Zvuk je pak měkčí, fáze nástupu slabší a kratší, perkusivních složek je pak méně. U jiných klavírů podle [1] střední pedál funguje jako sustain pedál ale pouze pro struny, které byly aktivovány v okamžiku jeho sešlápnutí.
Na téma středního pedálu se poeticky vyjadřuje Arthur A. Reblitz v Piano Servicing, Tunig and Rebuilding [2]: *„V jistém smyslu je sostenuto téměř jako třetí ruka pokročilého klavíristy...“*

1.2.3 Barva zvuku

Barva je jednou z vlastností zvuku, díky které dokážeme jednotlivé zvuky přiřazovat k jejich zdrojům. Barva každého nástroje je jiná. To nám dovoluje nástroje od sebe rozeznat. Barva zvuku např. okaríny je ve srovnání s barvou houslí či varhan dosti chabá, varhany znějí subjektivně o moc komplexněji. Podle Ohmova základního psychoakustického zákona člověk vnímá kmitání sinusového průběhu jako prostý tón. Jiná periodická kmitání, složitější než sinus nebo kosinus jsou ve sluchovém aparátu rozkládány do řady harmonických průběhů. Frekvence a amplituda těchto dílčích průběhů je vnímána jako celkový vjem barvy [4].

Podle [5] lze každou periodickou funkci času (pro nás signál) zkonstruovat pomocí určitého počtu sinů a cosinů s určitou fází a amplitudou. Jinak řečeno, každou periodickou funkci (např. tvar kmitání struny) lze rozložit na dílčí harmonické složky. Poměr amplitud těchto složek a jejich počet definují barvu zvuku nástroje. Složky jsou harmonické, tzn., že jsou vyjádřeny funkcí sinus nebo kosinus a platí, že kmitočet těchto složek je celočíselným násobkem kmitočtu první harmonické složky – fundamentu. Právě s touto skutečností se musí náš analyzátor akordů klavíru poprat, protože sluchově určujeme výšku tónu hlavně podle první harmonické složky. Ne vždy však slyšíme stejnou frekvenci jako tu, kterou má vlna vysílaná zdrojem.

Jednotkou pro subjektivní vjem výšky tónu je *mel*. Závislost vjemu výšky tónu na frekvenci je zhruba logaritmická (při zdvojnásobování frekvence slyšíme konstantní přírustky ve výšce tónu). Subjektivní výška tónu závisí také na hlasitosti [6]. Akord nám definují první harmonické složky min. tří tónů. Proto bude potřeba fundamenty ve spektru vyhledat a oddělit je od zbytku složek, abychom mohli určit kmitočty a tím i výšky tónů, které přispívají k akordu.

Barva zvuku klavíru

V sekci 1.2 bylo uvedeno, že výšku tónu, přesněji kmitočet vibrace struny definuje její délka, síla napnutí a objemová hustota. Je zde nutné dodat, že ze vztahu:

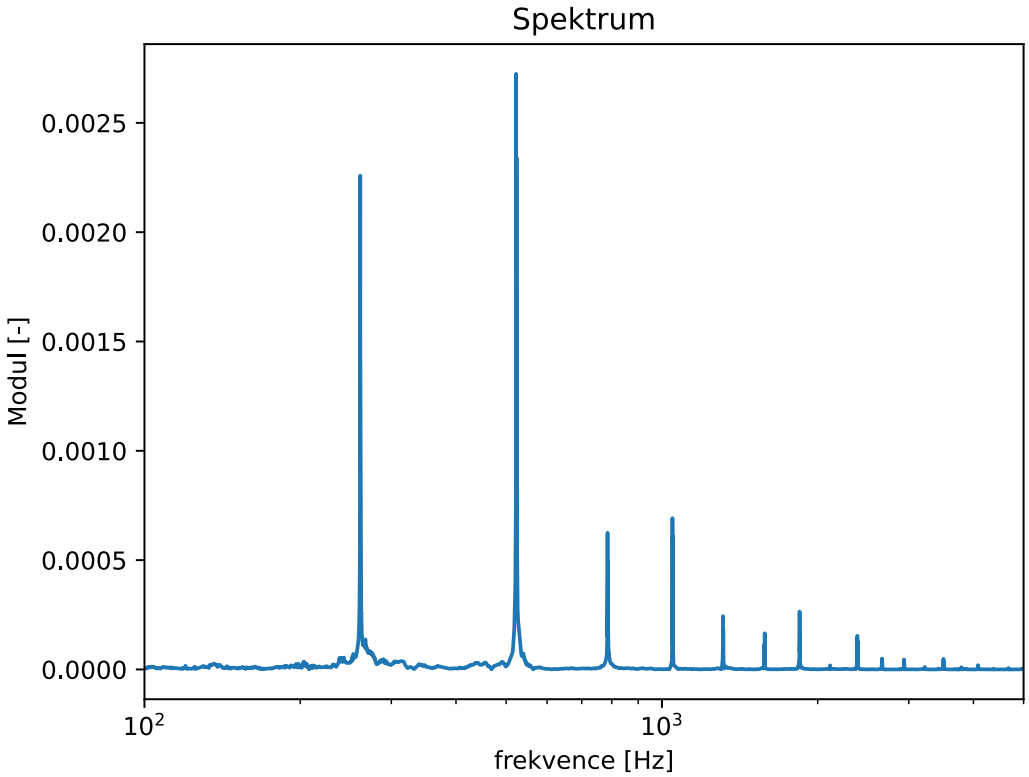
$$f = \frac{1}{Ld}\sqrt{\frac{F}{\pi\rho}}, \tag{1.1}$$

kde podle [3] L je délka struny v metrech, d je průměr struny v metrech, F je tažná síla v Newtonech, ρ je objemová hustota v $\mathrm{kg} \cdot \mathrm{m}^{-3}$, získáme pouze kmitočet první harmonické složky – fundamentu.

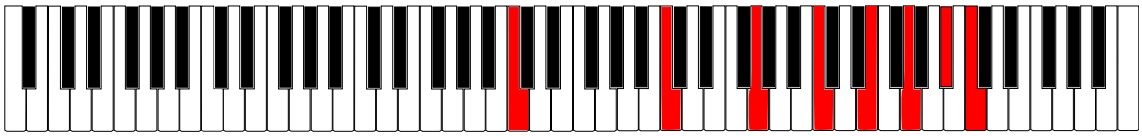
Následující odstavec čerpá informace z [4]. Když kladívko udeří do struny, začne se z místa úderu šířit k jejím oběma upevněným koncům příčné vlnění (výchylka kolmá ke směru šíření). Jelikož je struna na obou koncích upevněna, vlna při odrazu mění svou fázi. Odraz se děje na obou koncích struny, odražené vlny postupují proti sobě, vzájemně interferují a vzniká stojaté vlnění. Tvar kmitání struny je ovlivněn tvarem kladívka a dobou jeho dotyku se strunou. Čím vyšší je základní kmitočet buzené struny, tím více se doba kontaktu kladívka se strunou blíží k periodě příčného vlnění, což způsobuje zmenšení počtu vyšších harmonických složek.

Matematickým popisem barvy tónu je spektrum, které je úzce spjato s tvarem zvukové vlny, kterou nástroj produkuje. Modulové spektrum kmitání struny tónu C4, získané provedením diskrétní Fourierovy transformace je zobrazeno na obrázku 1.3. Je důležité si uvědomit, že spektrum tónu vnímáme najednou. Nevnímáme jej jako jednotlivé tóny, ale jako zabarvení, specifiku fundamentu: obr. 1.4. Hlasitost harmonických složek s jejich rostoucím pořadovým číslem často klesá, ale není tak tomu vždy (viz právě obr. 1.3). Toto je krajní případ, kdy některé metody pro identifikaci výšky tónu můžou havarovat. Z obr 1.3 je patrné, že první harmonická složka nemá nejvyšší modul, ale i přes to její perioda bude periodou postupné vlny na struně. Právě na periodu prvních harmonických složek tónů se budeme zaměřovat.

Struny ovlivňují barvu tónu dvěma způsoby: svým počtem a typem. Spodní struny od A0 do A1 jsou jednotlivé – jednochór. Od A♯1 do D♯3 jsou struny zdvojené, tzn. kladívko, při stisku klávesy bije do dvou strun (ideálně s identickými kmitočty) – dvojchór a od E3 do C8 jsou struny tři – trojchór. Aby struny s nízkými



Obr. 1.3: Modulové spektrum tónu C4, bez použití pedálů



Obr. 1.4: Harmonická řada tónu C4 jako akord (prvních 8 složek)

frekvencemi nemusely být příliš dlouhé, dosahuje se jejich základního kmitočtu zvýšením objemové hustoty. Holá struna je opředená měděným drátem, což zvětší její hmotnost a zpomalí kmitání. Opředení má také veliký vliv na barvu, potažmo na spektrum.

Některými z faktorů, které se podílí na celkové barvě tónu klavíru jsou podle [3]: délka ozvučné skříně (140–290 cm), ozvučná deska, její materiál, způsob konstrukce, žebrování, struny (počet, délka, síla natažení, typ, materiál), mechanické vlastnosti kladívek, dusítka aj. Z pohledu technické analýzy nahrávek také spektrum závisí na typu použitého mikrofonu, jeho poloze, kvalitě a jiných technických specifikacích nahrávacího řetězce.

1.3 Ladění

Celá tato práce předpokládá, že klavír je naladěn v rovnoměrně temperovaném ladění, takže rozdíl mezi kmitočty strun odpovídajících dvou sousedním klapkám je $\sqrt[12]{2}$. Kmitočet jiné struny vzdálené o n klapek od klapky referenční je dán vztahem:

$$f_2 = f_1 \cdot (\sqrt[12]{2})^n. \tag{1.2}$$

Hodnota $\sqrt[12]{2}$ odpovídá v rovnoměrně temperovaném ladění zdvihu o půltón. Pokud bychom potřebovali kroky mezi kmitočty menší než půltón, je zde ještě jednotka *cent*. Cent dělí oktávu na 1200 stupňů, tedy zdvih o půltón odpovídá zdvihu o 100 centů.

Rovnoměrně temperované ladění bylo zavedeno v baroku. Vyznačuje se tím, že dělí oktávu (přesněji kmitočtové pásmo vyznačené určitou frekvencí a jejím dvojnásobkem) na 12 stupňů – půltónů. Mimo rovnoměrně temperované ladění existuje řada jiných: pythagorejské, přirozené, středotónové aj. [4]. Specifikace těchto systémů ladění přesahuje obsah této práce.

1.4 Komunikační rozhraní MIDI

MIDI (*Musical Instruments Digital Interface*) si můžeme představit jako jakýsi jazyk, kterým společně komunikují hudební nástroje a zvuková zařízení. Přesněji se jedná o komunikační protokol mezi zvukovými zařízeními. Rozhraní MIDI bylo prvním komunikačním rozhraním s ucelenou a pevně určenou strukturou a formátem. Podle [7] bylo zavedeno za účelem připojení nástroje (většinou klávesového) k počítači, práce s hudebními daty mimo reálný čas, automatizované hry a hlavně získání nových zvukových barev ze současně znějících nástrojů. Základem je paralelní propojení zařízení. Je důležité uvědomit si, že data přenášená přes MIDI nejsou analogovým signálem ani digitálními vzorky zvuku, ale daty řídícími.

Protokol MIDI používá pro identifikaci not a tónů hodnoty 0–127. Hodnotu 0 má podle oficiální MIDI specifikace [8] sub-subkontra C, f_c = 8,18 Hz a hodnotu 127 má G šestičárkované, f_c = 12543,88 Hz. Většinu tónů v hudbě lze identifikovat právě podle MIDI hodnot. Rozsah klavíru v těchto hodnotách je 21–108. Symbol pro tyto hodnoty je p jako *pitch* a jsou definovány podle vztahu:

$$p(f_c) = 69 + 12\log_2(\frac{f_c}{440}), \tag{1.3}$$

kde f_c je střední frekvence noty, pro kterou hodnotu p zjišťujeme. Střední kmitočet tónu z hodnoty p lze získat podle:

$$f_c(p) = 2^{\frac{(p-69)}{12}} \cdot 440. \tag{1.4}$$

V rovnicích se objevuje hodnota 69, což odpovídá hodnotě p pro komorní A. Ve vztazích uvažujeme střední kmitočet, protože v sekci 2.3.1 se bude hovořit o kmitočtových pásmech popsaných hodnotou p.
Pro klavír má MIDI speciální notaci skládající se z písmena (případně křížku nebo béčka) a čísla. Písmeno označuje notu, číslo označuje číslo oktávy na klavíru, ve které se nota nachází. Jelikož ze subkontra oktávy jsou na klavíru pouze 3 tóny, identifikuje se tato oktáva jako nultá. První klapka zleva má kód A0 a poslední klapka zprava C8.

1.5 Teorie akordů

Celá podkapitola je postavena na [9]. Pro potřeby práce definujeme akord jako souzvuk nejvýše 4 tónů, přičemž nejnižší a nejvyšší tón jsou od sebe vzdáleny nejvýše o oktávu (kmitočet nejvyššího tónu je maximálně dvojnásobkem kmitočtu tónu nejnižšího). Analýza souzvuků složených z 5 a více tónů, nebo akordů, jejichž největší interval mezi tóny je větší než oktáva, přesahuje rámec této práce. Abychom pochopili koncept názvosloví a složení akordů, musíme nejprve objasnit stavební kameny jejich popisu.

1.5.1 Stupnice

Stupnice je řada tónů – tóny seřazené vzestupně nebo sestupně s určitým krokem v pásmu jedné oktávy. Obsahuje jeden hlavní tón – tóniku, od kterého se řada staví a podle něj se taky jmenuje. Stupnici lze poskládat od libovolného tónu. Kmitočet tóniky stanoví jednu hranici stupnice a jeho dvojnásobek druhou. Pásmo vyznačené těmito hranicemi se nazývá oktáva. Existuje velké množství stupnic: durová, mollová, dórská, mixolydická, lydická aj. Tato práce bude brát v úvahu pouze stupnice sedmistupňové: durovou a mollovou aiolskou.
V této práci budou použity americké názvy not. Budeme používat označení B pro tón H. Snížené H budeme reprezentovat označením B♭ nebo A♯, protože použité knihovny a balíčky v praktické části pracují právě s americkou notací. Přípony, které označují typ akordu (dur, moll, maj) a akordové značky zůstávají v evropské notaci.

Durová stupnice sestává ze sedmi nebo chceme-li osmi tónů, přičemž osmý tón je o oktávu vyšší než první. Mezi stupni durové stupnice jsou celotónové a půltónové kroky, které lze shrnout ve vektoru kroků $\mathbf{K} = [1, 1, 1/2, 1, 1, 1, 1/2]$ (první hodnota vektoru odpovídá kroku mezi 1. a 2. tónem stupnice, druhá kroku mezi 2. a 3. krokem atd.). Pro shrnutí, vektor kroků durové stupnice má 3. a 7. místě hodnotu 1/2, na jiných místech jsou hodnoty 1. Nejjednodušší durovou stupnicí je Cdur, můžeme

Tab. 1.1: Výčet základních intervalů

Název	Počet půltónů	Příklad	Specifikace
Prima	0	C1–C1	Čistá
Sekunda	2	C1–D1	Velká
Tercie	4	C1–E1	Velká
Kvarta	5	C1–F1	Čistá
Kvinta	7	C1–G1	Čistá
Sexta	9	C1–A1	Velká
Septima	11	C1–B1	Velká
Oktáva	12	C1–C2	Čistá

ji zahrát postupným mačkáním 8 bílých klapek počínaje např. od C4. Pokud budeme stavět např. stupnici Ddur, začneme od tónu D a následujeme výše definovaný vektor kroků: D+1 tón=E, E+1 tón= F♯ (museli jsme o půl tónu zvýšit F, abychom zachovali celotónový krok mezi druhým a třetím stupněm stupnice), F♯+1/2 tónu = G, dále A, B, B+1 tón = C♯, C♯+1/2 tónu = D. Do stupnice Ddur jsme oproti Cdur přidali dva křížky – dvě zvýšení. Jeden křížek přidáváme do Gdur, dva do Ddur, tři do Adur a řada pokračuje Edur, Hdur, F♯dur, C♯ dur G♯dur... Obecněji, křížky přidáváme do stupnic stavěných od pátého stupně stupnice předchozí. Názvy durových stupnic píšeme velkými písmeny.

Mollová stupnice aiolská má také 7 stupňů, ale vektor kroků je v aiolské stupnici jiný: $\boldsymbol{K}$ = [1, 1/2, 1, 1, 1/2, 1, 1]. Nejprostší aiolskou stupnicí je amoll – všechny bílé klapky v rozmezí oktávy od tónu A. Existují ještě stupnice mollová harmonická: $\boldsymbol{K}$ = [1, 1/2, 1, 1, 1/2, 3/2, 1/2] a melodická: $\boldsymbol{K}$ = [1, 1/2, 1, 1, 1, 1, 1/2] aj. Názvy mollových stupnic píšeme malými písmeny.

1.5.2 Intervaly

Interval je v hudbě vzdálenost mezi dvěma tóny. Rozlišujeme osm základních intervalů + jejich modifikace (zvětšení, zmenšení). Dělíme je na čisté a velké. Souhrn intervalů je v tabulce 1.1: Všechny intervaly lze zvětšovat, nebo zmenšovat. Pokud zmenšíme velký interval, tj. vzdálenost mezi tóny zkrátíme o jeden půltón oproti tabulce, říkáme že je malý. Pokud jej zvětšíme, přidáme půltón a říkáme, že je zvětšený. Např. malá tercie od C4 je Es4 = E♭4 a zvětšená tercie od C4 bude E♯4 = F4. U čistých intervalů říkáme pouze zmenšený/zvětšený, případně dvojzmenšený/dvojzvětšený. Pro zvětšení používáme křížek „♯“, pro snížení béčko „♭“. Příklady najdeme v tabulce 1.2.

Tab. 1.2: Příklady intervalů

Název	Počet půltónů	Příklad
Velká tercie	4	D3–F♯3
Malá sexta	8	F2–D♭3
Zvětšená kvinta	8	F2–C♯3
Zmenšená kvinta	6	F2–C♭3 = B2
Malá sekunda	1	G1–A♭1
Zvětšená kvarta	6	C1–F♯1

1.5.3 Akordy

Akordem se rozumí každé tři a více tónů znějících současně. V této práci se bude brát v úvahu pouze souzvuky složené z nejvýše 4 tónu, přičemž interval mezi nejnižším a nejvyšším tónem akordu nebude větší než oktáva.

Trojzvuky

V této práci budeme brát v úvahu pouze trojzvuky durové a mollové, tj. základní akordy postavené v základních stupnicích. Trojzvuk, přesněji kvintakord sestává z tercie a kvinty od základního tónu. Pokud je tercie velká, jedná se o kvintakord durový, pokud je malá, jde o mollový.
Pokud základní tón posuneme o oktávu výše (C1–E1–G1 → E1–G1–C2), jedná se o sextakord, mezi prvním a třetím tónem je sexta. Pokud posun nejnižšího tónu akordu provedeme znovu, obdržíme kvartsextakord (E1–G1–C2 → G1–C2–E2). Jestli stejnou operaci zopakujeme potřetí, obdržíme znovu originální kvintakord, o oktávu výše posunutý. Pro plný popis trojzvuku udáváme základní tón kvintakordu, typ tercie kvintakordu (dur/moll) a obrat (kvintakord, sextakord, kvartsextakord). V kapitole 5 najdeme mimo jiné i výsledky testů systémů na dvou datasetech. Pro úplnost dodáváme, že jeden z nich obsahuje mimo durové a mollové také zvětšené a zmenšené trojzvuky. Obraty u zvětšených/zmenšených trojzvuků fungují stejně jako u základních. Zvětšený trojzvuk má velkou tercii a zvětšenou kvintu, zmenšený má malou tercii a zmenšenou kvintu. Příklady najdeme v tabulce 1.3.

Čtyřzvuky

Nejprostšími čtyřzvuky jsou akordy odvozeny od akordů popsaných výše. Jedná se o trojzvuk s přidanou oktávou od nejnižšího tónu. Názvy obratů i názvy akordů jsou stejné jako u trojzvuků. Příklady: Ddur kvintakord – [D2, F♯2, A2, D3], bmoll kvartsextakord - [F♯1, B1, D2, F♯2]. Přidání oktávy nejnižšího tónu vnáší problémy do

Tab. 1.3: Příklady akordů

Tóny	Název	Obrat
C2–E2–G2	Cdur	kvintakord
C2–E♭2–G2	cmoll	kvintakord
G♯2–B2–E3	Edur	sextakord
B2–E3–G2	emoll	kvartsextakord
B3–D♯4–F♯4	Bdur	kvintakord
B♯2(= C3)–D♯3–G♯3	G♯dur	sextakord
B♭2–E♭3–G♭3	e♭moll	kvartsextakord
D3–F3–B♭3	B♭dur	sextakord
F4–A♭4–C♭5(=B4)	f zmenšený	kvitakord
A♯5–D6–F♯6	D zvětšený	kvartsextakord

analýzy neboť interval druhé harmonické složky je oktáva od fundamentu. Nejvyšší tón zmíněného typu akordů je zároveň svým fundamentem i druhou harmonickou nejnižšího tónu. V této práci se čtyřzvukem rozumí trojzvuk s přidanou oktávou nejnižšího tónu, ne septakord.

Septakordy

Septakord je čtyřzvuk, který má oproti trojzvukovému akordu přidanou septimu. Septakord tedy sestává z primy, tercie, kvinty a septimy od základního tónu. Podle typu tercie (velká/malá), kvinty (čistá/zmenšená/zvětšená) a septimy (velká/malá/zmenšená) je septakord určen. Tak jako u akordů popsaných výše se septakord jmenuje podle svého základního tónu. U septakordů se naopak nepoužívá označení dur/moll. Název septakordu sestává z základního tónu, typu tercie a typu septimy. Pro akordy s s čistou kvintou a velkou tercií platí označení „tvrdě“, s čistou kvintou a malou tercií „měkce“. Pokud kvinta není čistá, ale zvětšená nebo zmenšená, v názvu je určení právě kvinty, ne tercie. Je nutné dbát na to, že i septakord je složen ze tří tercií (velkých/malých), proto nemůže nastat případ výskytu malé tercie a zvětšené kvinty ani velké tercie a zmenšené kvinty. Proto se u akordů s jiným typem kvinty nezdůrazňuje typ tercie, vyniká totiž z typu kvinty. Poslední slovo v názvu septakordu patří septimě, která může být velká/malá/zmenšená. Při zachování podmínky, že každý tón akordu je o tercii vyšší, než jeho nižší soused existuje sedm septakordů: tvrdě malý, tvrdě velký, měkce malý, měkce velký, zvětšeně velký, zmenšeně malý, zmenšeně zmenšený.

Obraty fungují stejně jako u trojzvuků tj. posunutím nejnižšího tónu o oktávu výše získáme další obrat: septakord (nejnižší tón je základní), kvintsextakord, terckvar-

Tab. 1.4: Příklady septakordů

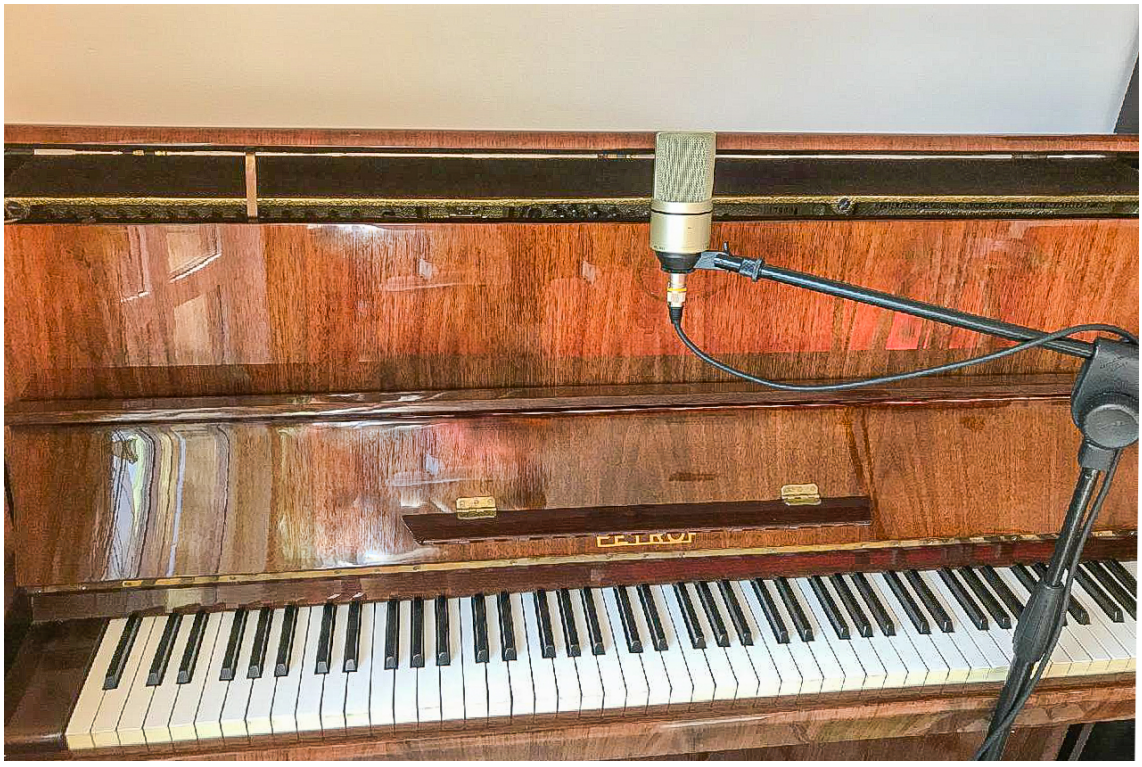
Tóny	Název	Obrat	Značka
C2–E2–G2–B2	C tvrdě velký	septakord	Cmaj
F2–A2–D3–C♯3	D měkce velký	kvintsextakord	dmaj/Dmmaj
B3–E4–D♯4–G4	E tvrdě malý	terckvartakord	E7
E♭4–F4–A♭4–C5	F měkce malý	sekundakord	f7/Fm7
G3–B3–D♯4–F♯4	G zvětšeně velký	septakord	Gmaj/5+
C3–E♭3–G3–A3	A zmenšeně malý	kvintsextakord	a7/5-
B3–D4–F4–A♭4	B zmenšeně zmenšený	septakord	Bdim

takord, sekundakord. Přesněji se v názvu nachází interval, který svírá nejnižší tón se septimou a základním tónem. Příklady najdeme v tabulce 1.4. Velmi specifickým případem je zmenšeně zmenšený septakord, který sestává z tří malých tercií. Existují jen tři a to stavěné od tónů C, C♯, D. Zmenšeně zmenšený septakord od tónu D♯ je zároveň obratem – kvintsextakordem Cdim. Podobně septakord Gdim [G, B♭, D♭, F] je terckvartakordem od D♭dim. Stejně se chovají zmenšeně zmenšené septakordy od zbylých výše naznačených tónů.

1.6 Testovací dataset

Testovací dataset obsahuje 514 3–4sekundových nahrávek klavírních akordů (251 durových, 261 mollových a 2 zmenšeně zmenšené) a 16 nahrávek jednotlivých tónů. Nahrávky pochází z domácího pianina *Petrof* a koncertního křídla *Petrof* ze ZUŠ v Českém Těšíně. V rámci akordů je v datasetu 485 trojzvuků (durových a mollových kvintakordů, jejich obratů ve všech tóninách a ve všech oktávách klavíru), 10 čtyřzvuků a 19 septakordů. 22 akordů má nejnižší tón v 0. oktávě klavíru, 80 v první, 80 v druhé, v třetí a čtvrté po 78, 81 v páté, 77 v šesté a 34 v sedmé. Celkem se v datasetu nachází 1587 fundamentů. Dataset byl nahráván kondenzátorovým mikrofonem MXL 990 umístěným před otevřeným víkem pianina/klavíru, viz obr. 1.5. Mikrofon byl připojen na vstup zvukové karty Steinberg UR22C a její výstup do programu Cubase AI 11. Nahrávání probíhalo s vzorkovací frekvencí 48 kHz a bitovou hloubkou 16 bitů. Během zpracování došlo k převzorkování na 44,1 kHz. Převzorkování našemu systému nevadí, vzorkovací frekvence je získávána pomocí funkce `librosa.get_samplerate()`. Tato práce předpokládá, že kmitočty jednotlivých strun (až na výjimky v 7. oktávě) odpovídají MIDI kmitočtům. Ucho posluchače nezaznamenává slyšitelné rozladění, klavír se běžně používá.

Jednotlivé nahrávky – zvukové soubory mají specifické názvy, které je nejlépe vysvětlit na příkladu: `001_s_G+_2_5`. První tři místa jsou vyznačena pro pořadové číslo nahrávky, podtržítko odděluje informace. Na druhém významném místě je písmeno „h“ nebo „s“, což označuje klavír, na kterém byl akord nahrán. „h“ pro domácí pianino, „s“ pro školní klavír. Třetí místo značí tóninu. Pokud je písmeno velké, jedná se o dur, pokud je malé, jde o moll. Za písmenem může stát znak „+“, což označuje zvýšení tónu vyznačeného písmenem. Čtvrté místo je rezervováno pro obrat: 0 pro kvintakord, 1 pro sextakord, 2 pro kvartsextakord. Poslední místo vyznačuje oktávu, ve které se podle MIDI notace nachází nejnižší nota akordu. Náš ukázkový název znamená: První nahrávka, školní klavír, G♯dur, kvartsextakord s první notou v páté oktávě ($p = [75, 80, 84]$). Pokud se jedná o čtyřzvuk, nebo septakord, název je o jednou číslici delší a ta odpovídá typu septakordu/čtyřzvuku: 10 pro čtyřzvuk, 3 pro tvrdě malý septakord, 4 pro měkce malý, 5 tvrdě velký, 6 měkce velký, 7 zvětšeně velký, 8 zmenšeně malý, 9 zmenšeně zmenšený. Předposlední číslice nadále značí obrat (u čtyřzvuků totožně s trojzvuky) – 0 pro septakord, 1 kvintsextakord, 2 terckvartakord, 3 sekundakord. Poslední číslice zůstává oktávě nejnižšího tónu akordu.



Obr. 1.5: Nahrávání datasetu

2 Analýza hudebního obsahu

V této kapitole budou ukázány signálové reprezentace, které pomohou najít základní frekvenci tónu mezi jeho harmonickými složkami. Základním dělením, použitým v této kapitole, je dělení na časovou a spektrální oblast. Některé metody pracují v obou oblastech (např. spektrogram) a budou komentovány v podkapitole k tomu vyznačené.

2.1 Časová rovina

V časové oblasti sledujeme signál měnící se v čase. Čas nebo vzorky odebírané v ekvidistantních časových okamžicích jsou nezávislou proměnnou, kterou značíme na vodorovné ose. Závislou proměnnou je okamžitá hodnota napětí signálu, nebo jiná veličina, kterou sledujeme. Nevýhodou analýzy v časové oblasti je to, že nemáme informace o vyšších harmonických složkách signálu, nevíme, z čeho je signál složen. Podle [10] je signál s tónovými složkami (bez ruchů a šumů) definován:

$$x(t) = \sum_{k=1}^{n} A_k \cos(2\pi f_k t + \phi_k). \tag{2.1}$$

Signál je složen z n harmonických složek, každá z nich má svou frekvenci $f_k = kf_1$ viz 1.2.3, počáteční fázi ϕ_k a amplitudu A_k. Pro analýzu výšky tónu je naším cílem mezi těmito složkami najít první a pevně určit její kmitočet. Signál, reprezentující trojzvuk (akord) lze pak definovat jako:

$$s(t) = x_1(t) + x_2(t) + x_3(t) = \sum_{i=1}^{3} \sum_{k=1}^{n} A_{ik} \cos(2\pi f_{ik} t + \phi_{ik}). \tag{2.2}$$

Pro potřeby práce ještě definujeme podmínku periodicity signálu:

$$x(t) - x(t+T) = 0, \forall t. \tag{2.3}$$

Signál je periodický, pokud jeho hodnota v čase t je stejná jako hodnota v čase $t+T$, kde T je perioda signálu. $t, T \in \mathbb{R}$. Podmínka platí pro všechna t, tedy pro nekonečný počet časových hodnot. Signál pocházející z klavíru není striktně periodický

2.1.1 A/D převod

Je nutné si uvědomit, že v počítači pracujeme s digitálním signálem, zatímco na membránu mikrofonu dopadá vzduchové vlnění, transformuje se na napětí, jenž je v čase spojité a je analogové (rov. 2.1). Ve zvukové kartě dochází k převodu signálu z analogového na digitální (A/D převod). A/D převod sestává podle [5] z tří kroků:

vzorkování, kvantování a kódování. V průběhu vzorkování jsou z analogového signálu pobírány vzorky v okamžicích, které se periodicky opakují s vzorkovací periodou T_{vz}. Vzorkovací frekvence musí splňovat tzv. vzorkovací poučku: $f_{\text{vz}} \geq 2f_{\text{max}}$ tzn., že kmitočet vzorkování musí být minimálně dvojnásobkem nejvyšší spektrální složky signálu, aby nedošlo k aliasingu (překrytí spekter v důsledku jejich periodizace A/D převodem). Pro nás je důležité slyšitelné pásmo 20 Hz – 20 kHz, takže stačí, když $f_{\text{vz}} \geq 40$ kHz. V praxi se vzorkovací kmitočet nastavuje vyšší: 44,1 kHz nebo 48 kHz. Jedním z důvodů je použití antialiasingového filtru typu dolní propust, který spektrum signálu omezí. Jelikož přechod kmitočtové charakteristiky z propustného do nepropustného pásma není skokový (je zde přechodová oblast, kde charakteristika spojitě klesá, viz 2.2.4), mohlo by se stát, že by některé složky (např. 20,5 kHz pro $f_{\text{vz}} = 40$ kHz) byly utlumeny jen částečně, pronikly by do signálu a způsobily by aliasing. Proto se f_{vz} volí větší, aby se zachovala kmitočtová rezerva pro složky, které kmitočtem spadnou do přechodného pásma, kde útlum ještě není výrazný. Druhým důvodem kmitočtové rezervy je kompatibilita s různými digitálními zvukovými formáty. V důsledku vzorkování získáváme signál s diskrétním časem, zápis $s(t)$ zde přechází v $s[n]$. Cílem kvantování je přiřadit každému vzorku hodnotu na svislé ose. V ideálním případě bude hodnota vzorku rovna hodnotě analogového signálu v okamžiku vzorkování, což ale nelze zaručit, protože disponujeme pouze určitým počtem hodnot (analog nekonečným). V důsledku omezeného počtu hodnot vzniká kvantovací šum. Získáváme diskrétní signál. Kódování přiřazuje hodnotám vzorků binární čísla. Získáváme digitální signál, interpretovatelný počítačem. Jinými slovy, vzorkování dělí časovou osu na body, kvantování dělá totéž s osou okamžitých hodnot signálu a kódování přiřazuje binární hodnoty. V digitální podobě má signál svou binární hodnotu ne v daném čase, ale v daném vzorku.

Jedním z prostředků pro určení základní frekvence tónu, který pracuje v časové rovině je autokorelační funkce. Autokorelační funkce je považována za metodu určování základního tónu, proto je popsaná v kapitole 3.1.

2.2 Kmitočtová rovina

V kmitočtové oblasti můžeme pozorovat složení signálu, jeho harmonickou analýzu. Na vodorovné ose většinou zobrazujeme nezávislou proměnnou – frekvenci, nejlépe v logaritmickém měřítku, na svislé ose zůstává amplituda nebo modul (polovina amplitudy, pokud DFT normujeme počtem vzorků signálu). Nevýhodou je ztráta informací o časových změnách v signálu. Signál je v kmitočtové rovině vyjádřen komplexním číslem.

2.2.1 Fourierova řada

Je známo, že spektrum je technickým popisem barvy a témbru zvuku (sekce 1.2.3). Spektrum periodického signálu získáme jeho rozkladem do Fourierovy řady. Výsledkem rozkladu, tedy obrazem signálu v kmitočtové oblasti je spektrum modulové a fázové. Pokud sečteme hodnoty modulů sobě odpovídajících složek v záporné a kladné části spektra ($|c_{-1}| + |c_1|$, $|c_{-2}| + |c_2|$) a zobrazíme je v kladné části frekvenční osy, obdržíme spektrum amplitudové. Komplexní tvar Fourierovy řady je podle [5] dán vztahy:

$$s(t) = \sum_{k=-\infty}^{\infty} c_k \mathrm{e}^{\mathrm{j}k\omega_1 t},\ c_k \in \mathbb{C},\ \omega_1,\ t,\ s(t) \in \mathbb{R},\ k\ \in \mathbb{Z}, \tag{2.4}$$

$$c_k = \frac{1}{T_1} \int_{-\frac{T_1}{2}}^{\frac{T_1}{2}} C_k \mathrm{e}^{-\mathrm{j}k\omega_1 t} \mathrm{d}t,\ k = 0, \pm 1, \pm 2, \pm 3..., \tag{2.5}$$

$$c_k = |c_k| \mathrm{e}^{\mathrm{j}\varphi_k} = \frac{C_k}{2} \mathrm{e}^{\mathrm{j}\varphi_k}, k \neq 0, \tag{2.6}$$

kde $s(t)$ je periodický signál, c_k je k-tým koeficientem Fourierovy řady, ω_1 a T_1 jsou úhlový kmitočet a perioda signálu. Bohužel tyto vztahy nelze použít pro reálné průběhy z různých důvodů: prvním je skutečnost, že Fourierova řada pracuje se spojitými signály. V počítači pracujeme pouze se signály digitálními, viz kapitola 2.1.1. Zde by se zdála použitelná diskrétní Fourierova řada, ale druhým důvodem, proč FŘ/DFŘ nepoužíváme, je to, že z jejich definicí vychází použitelnost pouze na periodické signály (rov. 2.3). S periodickými signály se u akustických hudebních nástrojů nesetkáváme. Třetím problémem je to, že digitální signál je časově omezený svou délkou a kmitočtově Nyquistovým kmitočtem ($f_{\mathrm{vz}}/2$).

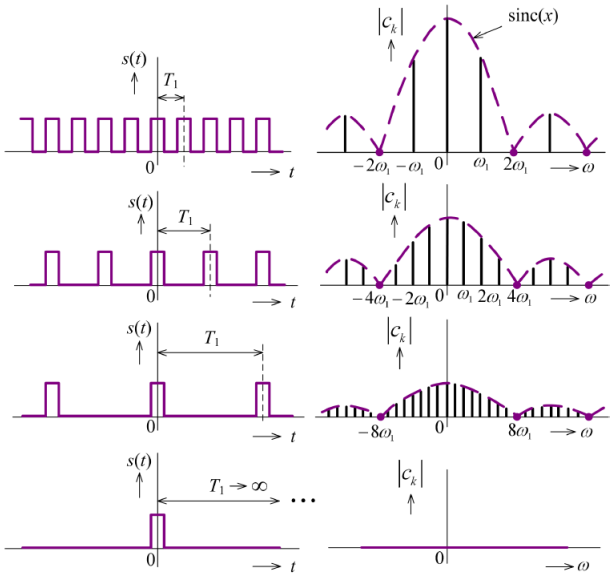
2.2.2 Fourierova transformace

V podkapitole výše jsme se setkali s Fourierovou řadou a jejími koeficienty, které určují kmitočty a jejich hodnoty moduly harmonických složek periodického signálu. Zobecněním Fourierovy řady i pro neperiodické signály je podle [5] Fourierova transformace, definována:

$$X(\omega) = \int_{-\infty}^{\infty} x(t) \mathrm{e}^{-\mathrm{j}\omega t}\, \mathrm{d}t. \tag{2.7}$$

Výsledkem bude znovu reálná a imaginární složka spektrální funkce, tedy modulové a fázové spektrum. Fourierova transformace považuje neperiodické signály za signály s nekonečně dlouhou periodou. Zatímco výsledkem Fourierovy řady bylo spektrum čárové, Fourierova transformace vede ke spektru spojitému. Při prodlužující se periodě se spektrální čáry (na kmitočtech jednotlivých harmonických) zahušťují a jejich velikost klesá. Při nekonečné periodě bude jejich modul nulový a spektrum bude

spojité, viz obr. 2.1. Nevýhodou je to, že spektrální funkce získána FT je průměrována přes celou dobu trvání signálu. S tím se vypořádá krátkodobá Fourierova transformace 2.3.1.



Obr. 2.1: Fourierova transformace pravoúhlého impulzu s prodlužující se periodou, převzato z [5]

Uvnitř integrálu ve vztahu 2.7 se násobí. Integrál vyjadřuje plochu pod grafem vzniklým součinem signálu a harmonické funkce vyjádřené exponenciálně. Pokud hodnota obou funkcí má po většinu času stejné znaménko, jejich součin bude kladný a hodnota integrálu vysoká. V případě, kdy si znaménka hodnot funkcí většinu času neodpovídají, hodnota integrálu bude nižší. Jedná se vlastně o typ míry podobnosti signálu a harmonické funkce [11].

2.2.3 Diskrétní Fourierova transformace

Pro diskrétní signály se používá DFT (*discrete Fourier transform*). DFT, tak jako Fourierova transformace, násobí signál s harmonickou funkcí (měří plochu pod grafem vzniklým jejich součinem). V digitálním prostředí stačí uvážit harmonickou funkci navzorkovanou. Integrál v diskrétní podobě odpovídá sumaci. Získáváme vztah:

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x[n] \mathrm{e}^{-2\pi \mathrm{j} kn/N}. \tag{2.8}$$

Aby $X(k)$ nabývalo konečného počtu hodnot, je třeba signál časově omezit, tedy pevně definovat počet vzorků N a pro výpočet považovat $x[n] = 0$ mimo uvažovaný interval.

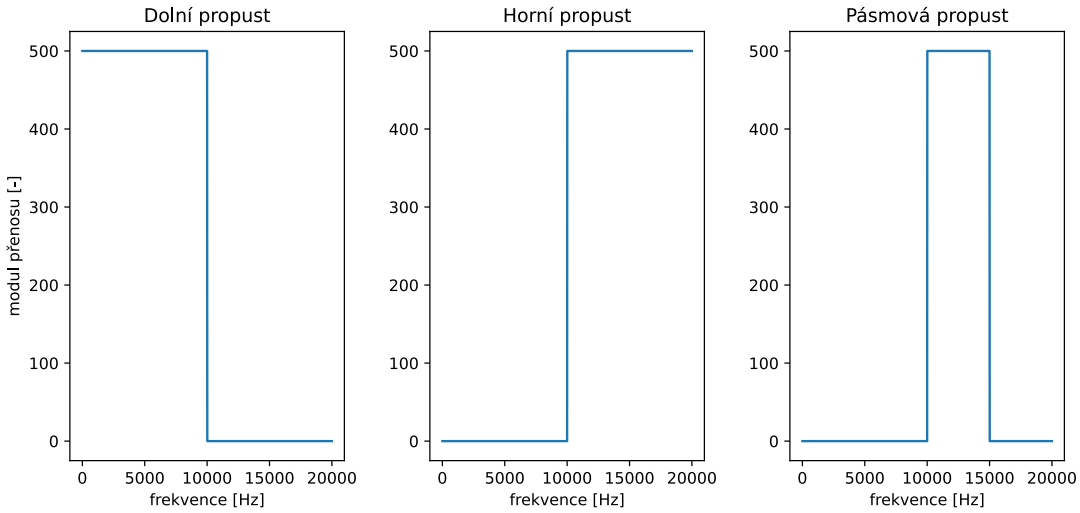
To samé je třeba udělat s kmitočtovou osou, protože nás zajímají harmonické složky do kmitočtu 20 kHz. Jak nastavit frekvenční rozlišení kmitočtové osy je popsáno v [11]. Při hledání základního kmitočtu signálu je pro nás důležitá hodnota k ze vztahu 2.8, která má po úpravě:

$$f_{\mathrm{coef}}(k) = \frac{k \cdot f_{\mathrm{vz}}}{N} \tag{2.9}$$

význam frekvence v Herzích. DFT má kmitočtovou i časovou osu diskrétní. Jednou z hlavních výhod DFT je to, že pokud se počet vzorků N rovná celočíselné mocnině čísla 2, lze ji efektivně spočítat pomocí FFT (*fast Fourier transform*) a snížit tím výpočetní náročnost. DFT je nejběžněji používaný prostředek pro zjištění spektrálního složení signálu.

2.2.4 Filtrace

Jedním z přístupů k identifikaci základního kmitočtu jednoho nebo vícera tónů je odfiltrování nepotřebné části spektra. Pokud analyzujeme základní kmitočty tónů vytvářených klavírem, vystačíme si s kmitočtovým pásmem 27,5–4186 Hz rov. 1.4). Pro odfiltrování části spektra se používají kmitočtové filtry. Kmitočtový filtr se charakterizuje např. pomocí šířky propustného pásma B [Hz], zesílení v propustném pásmu A [dB] (často 0 dB), strmostí kmitočtové charakteristiky v nepropustném pásmu k [dB/okt, dB/dek] a mezního kmitočtu f_{m}. Filtry typu pásmová propust a pásmová zádrž se charakterizuje např. pomocí středního kmitočtu a šířky pásma nebo jakosti Q. Dále lze filtry popisovat pomocí jejich přenosové funkce, impulsní odezvy, aproximace kmitočtové funkce. Popisy najdeme v [12] a [13].
Rozlišujeme různé typy kmitočtových filtrů: horní propust, dolní propust, pásmová propust, pásmová zádrž, hřebenový filtr aj. Kmitočtové charakteristiky některých z nich jsou znázorněny na obr. 2.2.



Obr. 2.2: Ideální modulové kmitočtové charakteristiky kmitočtových filtrů

V praxi takto kmitočtové charakteristiky nevypadají, přechod do nepropustného pásma je postupný, ne skokový. Pro nás jsou důležité filtry typu HP, které pomohou odfiltrovat matoucí harmonické složky a filtry typu PP. Pásmové propusti se používají zejména při odhadu vícera základních frekvencí. Signál se směřuje do banky filtrů typu PP, tím se oddělí fundamenty od sebe a na výstupech filtrů se pak používají metody pro odhad jedné základní frekvence [10].

2.3 Kmitočtově-časová oblast

2.3.1 Krátkodobá Fourierova transformace

Tato kapitola čerpá informace z [11]. Krátkodobá Fourierova transformace (*Short-time Fourier transform* – STFT) dokáže ve spektrální oblasti zachytit informace o časových změnách signálu, zatímco klasická Fourierova transformace ukazuje spektrum průměrované přes celou délku signálu. Hlavní myšlenkou je uvážení pouze malé části signálu, což zajistíme vynásobením funkcí okna, viz sekce 3.1.1. Princip je prostý: Okénkovou funkci se posouvá podél signálu v čase. Po každém posunu se signál s okénkovou funkcí pronásobí (mimo okno jsou všude nulové hodnoty) a spočítá se DFT jejich součinu. Krátkodobá Fourierova transformace signálu se spočítá podle vztahu:

$$\mathcal{X}(m,k) = \sum_{n=0}^{N-1} x(n+mH)W(n)\mathrm{e}^{-2\pi\mathrm{j}kn/N}, \tag{2.10}$$

kde N je délka okénkové funkce ve vzorcích, $W(n)$ je okénková funkce a H (*hop size*) je hodnota posunu okna ve vzorcích. Hodnotu H se často volí jako $H = N/2$.

Výsledek rovnice 2.10 se interpretuje jako hodnota modulu k-tého Fourierova koeficientu v m-tém časovém rámci. Abychom mohli informace získané STFT vyjadřovat v Herzích a sekundách, je třeba frekvenční i časovou osu upravit, čímž získáme časové a frekvenční rámce:

$$f_{\text{coef}}(k) = \frac{k \cdot f_{\text{vz}}}{N} \tag{2.11}$$

$$T_{\text{coef}}(m) = \frac{m \cdot H}{f_{\text{vz}}}. \tag{2.12}$$

Spektrogram

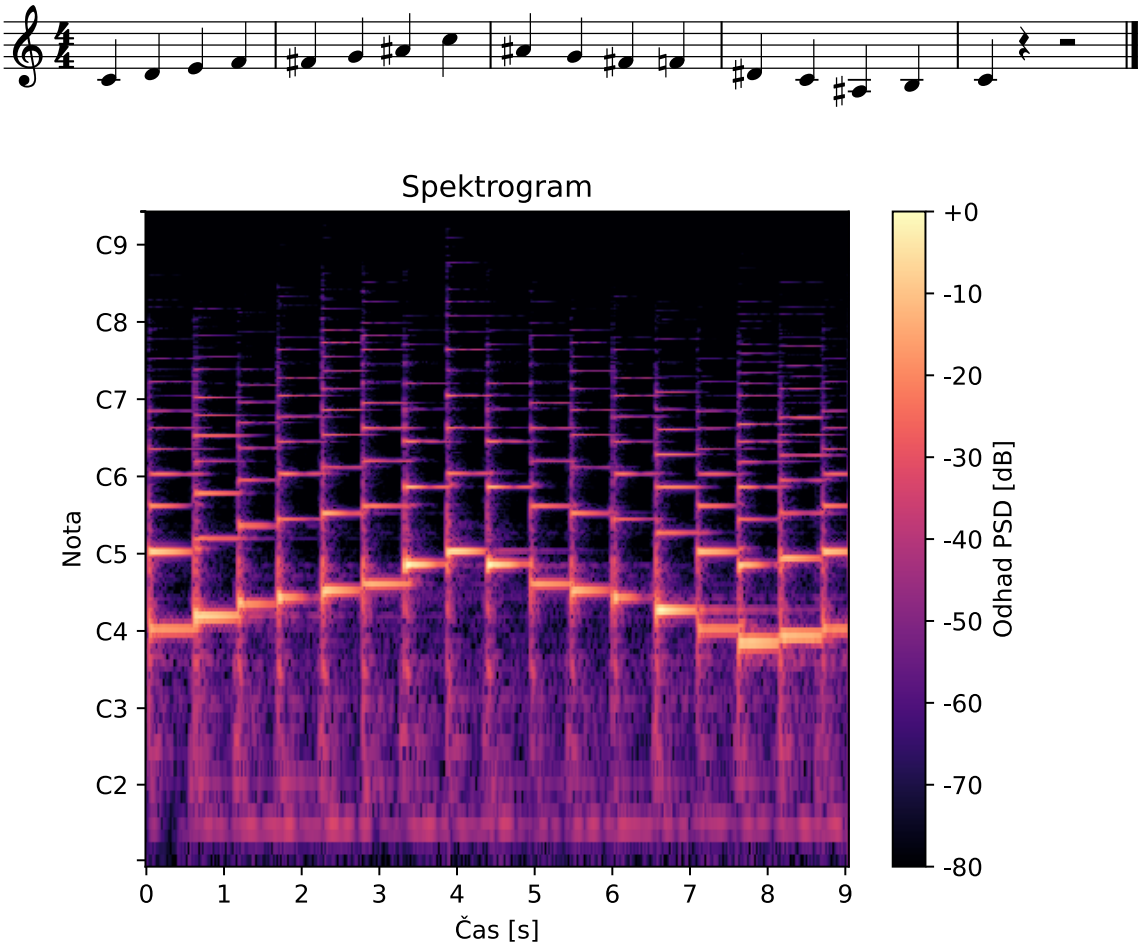
Názornou reprezentací STFT je spektrogram. Jde o graf závislosti druhé mocniny modulu krátkodobé Fourierovy transformace na frekvenci a čase. Jinými slovy, je to grafické znázornění frekvenčního složení signálu, které sleduje časové změny. Jelikož se jedná o graf, ve kterém je třeba vykreslit tři veličiny (časové rámce, kmitočtová pásma odhad výkonové spektrální hustoty v kmitočtových pásmech), je v dvojrozměrném provedení hodnota modulu $\mathcal{X}$ znázorněna různými barvami nebo odstíny barvy jedné. Spektrogram získáme pomocí:

$$\mathcal{Y}(m,k) = |\mathcal{X}(m,k)|^2. \tag{2.13}$$

Rozlišení spektrogramu záleží na délce časového okna W, použitého v rov. 2.10. Použití dlouhého okna přináší zlepšení rozlišení na frekvenční ose za cenu zhoršení rozlišení na ose časové. Krátké okno zlepšuje informaci o čase, ale rozmazává údaje o kmitočtu [14]. Zde je důležité najít kompromis mezi dlouhým a krátkým oknem podle toho, které informace jsou pro nás důležitější (časové nebo frekvenční). Spektrogram, získaný pomocí funkcí `stft` a `specshow` z knihovny `librosa` je zobrazen na obr 2.3. Zkratka PSD v popisu barevné osy značí *power spectral density* – výkonová spektrální hustota, přesněji její odhad. Jelikož spektrogram zobrazuje PSD, získáváme informaci o tom, jaká část výkonu signálu je kumulována v daném kmitočtovém pásmu. Tvar okna má také vliv na tvar spektra. Např. nejprostší okno, které signál jednoduše „ořízne“ a pronásobí jedničkou – obdélníkové, vnáší do spektra zvlnění, které nepochází ze zkoumaného signálu. Je to způsobeno právě ostrými hranami okna. Matematické zdůvodnění najdeme v [5], popis okénkových funkcí v [11].

Spektrogram s logaritmickou frekvenční osou

Na obr. 2.3 je pro názornější zobrazení svislá osa v logaritmickém měřítku. Navíc kmitočty jsou zde převedeny do názvů not. Pokud bychom chtěli kmitočty vyjadřovat v MIDI p hodnotách, které stoupají lineárně, bylo by třeba pro každou hodnotu p definovat kmitočtové pásmo. Jak je popsáno v kapitole 1.4, střední kmitočet f_{c}



Obr. 2.3: Notový zápis a spektrogram jednoduché sekvence tónů

pásma pro danou p hodnotu získáme pomocí vztahu 1.4. Jinými slovy, pro každé p se definuje množina kmitočtů, které bude pokrývat (s využitím vztahů 2.11 a 1.4):

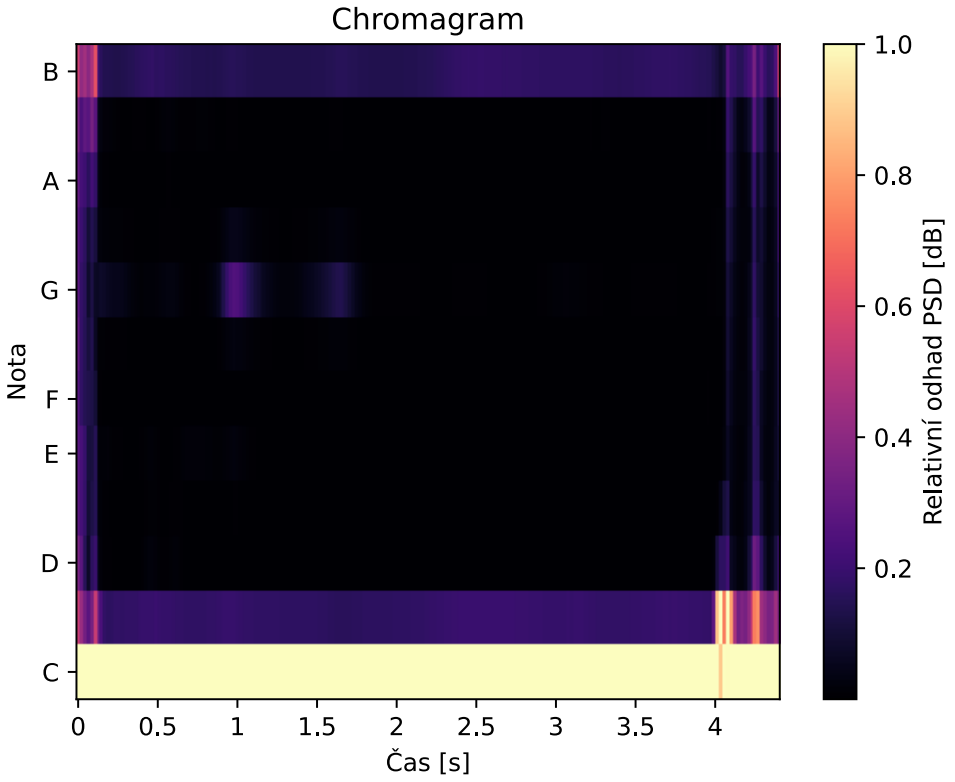
$$P(p) = \{f_{\mathrm{c}}(p - 0,5) \leq f_{\mathrm{coef}}(k) < f_{\mathrm{c}}(p + 0,5)\}. \tag{2.14}$$

Např. hodnota $P = 50$ bude odpovídat pásmu 142,65–150,13 Hz. Spektrogram s logaritmickou svislou osou popsanou MIDI hodnotami získáme z:

$$\mathcal{Y}_{\mathrm{LF}}(m, p) = \sum_{k \in P(p)} |\mathcal{X}(m, k)|^2. \tag{2.15}$$

Chromagram

Chromagram je kmitočtově-časová reprezentace audio signálu, která zobrazuje tónové složení signálu. Na rozdíl od spektrogramu, ve kterém jsou viditelné i jednotlivé harmonické složky signálu, chromagram kmitočtové složení spektra zastírá. Na svislé ose chromagramu je 12 pásem, které odpovídají tónům oktávy v rovnoměrně temperovaném ladění. Ztrácíme zde informaci o spektru, protože všechny harmonické



Obr. 2.4: Chromagram tónu C4, svislá osa vpravo normována k nejvyšší hodnotě PSD

složky budou zaznamenány v jedné oktávě. Příklad chromagramu je na obr. 2.4. Největší hodnotu odhadu PSD vidíme v pásmu C. Je to z toho důvodu, že se zde potkává první, druhá, čtvrtá a osmá harmonická složka (všechny jsou tónem C, ale nacházejí se v různých oktávách). Dále vidíme zastoupení v pásmu G. Tónu G patří třetí, šesté a dvanácté harmonické. V rámci cca prvních 0,25 s nalézáme zastoupení všech půltónů oktávy. To je způsobeno slyšitelným úderem kladívka do strun, což vede ke krátkodobému ruchu. Barevná osa vysvětluje barvy na chromagramu, je normována k jeho nejvyšší hodnotě. Matematická formulace chromagramu je podle [11]:

$$\mathcal{C}(m,c) = \sum_{(p\in[0:127]:p \mod 12=c)} \mathcal{Y}_{\mathrm{LF}}(m,p). \tag{2.16}$$

Například, chceme-li zjistit hodnotu chromagramu v časovém rámci m a tónu E, za c dosadíme 4 (E je 4. půltón v oktávě), hledáme všechny p, které jsou řešením rovnice $p \mathrm{mod} 12 = 4$, tedy $\{28, 40, 52, 64, 76, 88, 100\}$ a sumace sčítá vybrané hodnoty spektrogramu s logaritmickou frekvenční osou. Takuya Fujishima jako první použil chromagram v analýze akordů [15], kde chroma vektory (stavební kameny chromagramu) jednotlivých akordů porovnával s binárními akordovými předlohami.

2.3.2 Konstantní Q transformace – CQT

Jiným zobrazením signálu v kmitočtově-časové rovině je konstantní Q transformace (CQT – *constant Q transform*). CQT, podle [16], lépe zobrazuje frekvenční složení hudebního signálu než DFT, protože používá proměnnou délku okna. Konstantní Q transformace má na rozdíl od DFT proměnné frekvenční rozlišení (špičky spektra získaného DFT jsou na nízkých frekvencích široké, a může být těžké definovat, jakému kmitočtu patří). CQT si lze představit jako banku pásmových propustí s konstantní šířkou pásma a nastavitelným počtem na oktávu. Pokud chceme udržovat konstantní šířku frekvenčních pásem, musíme zmenšovat okno, kterým signál násobíme. Okno se zkracuje nepřímo úměrně s rostoucí frekvencí. Intuitivně bychom počet filtrů na oktávu volili roven 12, což odpovídá počtu půltónů v rovnoměrně temperovaném ladění. Často se tato hodnota volí rovna 24, protože na vyšších kmitočtech se harmonické složky od sebe liší o méně než půltón.
Hlavní předností konstantní Q transformace je konstantnost poměru středního kmitočtu a šířky pásma filtrů (rozlišení), v celém uvažovaném rozsahu.

$$Q = \frac{f_k}{\delta f_k} = \frac{f_k}{f_{k+1} - f_k} = \frac{f_k}{f_k(2^{\frac{1}{b}} - 1)}, \tag{2.17}$$

kde f_k je střední kmitočet k-tého filtru a b je počet filtrů na oktávu. CQT je výpočetně náročnější než DFT. Chromagram jde získat také pomocí CQT. Konstantní Q transformace je podrobněji popsána v [16] a [17].

3 Metody určování výšky tónů

V této kapitole prezentujeme jeden z přístupů k identifikaci zahraných akordů. Pomocí metod na určení výšky jednoho, nebo vícera tónů se snažíme určit kmitočty zahraných tónů. Získaný vektor kmitočtů převedeme do MIDI p hodnot, které přesně definují tóny klavíru v rovnoměrně temperovaném ladění. Pomocí rozdílů těchto hodnot určíme typ akordu, tóninu, obrat a umístění na klávesnici. Některé vzorce jsou v práci uvedeny ve formě pro spojité signály, přestože se v praxi používají v diskrétních podobách. Je to kvůli zachování stejného zápisu vůči použitým zdrojům.

3.1 Metody určování výšky jednoho tónu – *single f0 estimation*

3.1.1 Metody pracující v časové oblasti

Metody, pracující v časové oblasti hledají základní periodu signálu, ne frekvenci. Nezávislou proměnnou jsou zde indexy vzorků běžící v čase. Časový index n-tého vzorku v sekundách určíme prostým násobením $t_n = nT_{\text{vz}}$. Nezávislou proměnnou může být jiná veličina, např. zpoždění, udávané ve vzorcích. Většina metod pracujících v časové oblasti má společné to, že se jedná o operace (násobení, odčítání) signálu se svou zpožděnou verzí. Výhodou těchto metod je to, že nemusí řešit případy, kdy modul druhé nebo vyšší harmonické složky je větší než fundament, protože nahlíží na signál jako na časový průběh.

Průchody nulou

Nejjednodušším způsobem hledání periody signálu je hledat opakující se vzorce v časovém průběhu. U obdélníkového signálu najdeme opakující se vzorec mezi dvěma sousedními nástupnými hranami. U signálů, které nemají hrany stačí najít dvě sousední paralelní průchody nulou. Tyto průběhy nulou musí mít stejný směr (stejné znaménko derivace). Tato metoda má v hudbě jen malé využití, protože často signál prochází nulou vícekrát v rámci jedné periody.

Autokorelační funkce

Autokorelační funkce při hledání základní periody signálu respektuje i průběhy, které projdou nulou vícekrát v rámci jedné periody. Podle [18] jde o hledání podobnosti ke svému obrazu posouvanému v čase. Autokorelační funkce je definována jako:

$$r[\tau] = \sum_{n=0}^{N} x[n]x[n+\tau], \tag{3.1}$$

kde $r[\tau]$ je hodnota ACF (*autocorrelation function*) v hodnotě posunu τ a N je počet uvažovaných vzorků signálu. Maximální hodnotu nabývá ACF v bodě $\tau = 0$, jedná se o míru podobnosti s totožným signálem. Další maximum najdeme v $\tau = T$ tj. v posunu rovném periodě signálu. V [18] se pracuje s modifikací ACF, AMDF (*Average magnitude difference function*), která místo násobení používá odečítání a pomocí lokalizace minim vzniklé funkce odvozuje základní periodu signálu.

Metoda YIN

Knihovna `librosa` nabízí implementaci metody YIN [19] pro odhad základní frekvence tónu či tónové sekvence. Metoda byla vyvinuta v roce 2001. Jejím základem je ACF (rov. 3.1), váhovaná oknem o délce W, aby byla schopna zpracovat signál s proměnnou základní frekvencí v čase. Váhováním oknem se rozumí násobení funkcí, která má nenulové hodnoty pouze v oblasti okna, které se podle potřeby přesouvá. Existují různé typy oken: obdélníkové, trojúhelníkové, Hannovo, Blackmanovo aj. Použití samotné ACF má vysokou chybovost. Článek [19] uvádí, že odchylka 10 % odhadů přesahovala 20% hranici, proto prezentuje kroky, které metodu vylepší a chybovost sníží. Na základě rov. 3.1 je proveden druhý krok: je zavedena rozdílová funkce (SDF – *Squared difference function*) a pomocí její druhé mocniny je hledána perioda signálu. SDF je definovaná jako:

$$d_t[\tau] = \sum_{j=1}^{W}(x[j] - x[j+\tau])^2 = r_t[0] + r_{t+\tau}[0] - 2r_t[\tau], \tag{3.2}$$

kde W je délka okna a t je časový index vzorku, ve kterém okno začíná. Periodu signálu a její násobky najdeme v ideálním případě v nulových bodech této funkce. V praxi se hledají minima. Maxima rovnice 3.1 a minima rovnice 3.2 by se měla shodovat. Protože se člen $r_{t+\tau}[0]$ mění s τ, mohou se maxima ACF a minima SDF lišit. Kdyby první dva členy rovnice 3.2 byly konstantní, maxima ACF a minima SDF by si odpovídala [19]. Chybovost zde klesla z 10 % na 1,95 %.

První nulový bod funkce 3.2 je vždy v $d_t[0]$. Pokud se nestanoví spodní hranice τ (minimální očekávaná perioda), pod níž se základní perioda nebude hledat, bude $d_t[0]$ vyhodnoceno jako perioda, což je samozřejmě chybně. Také silná rezonance první formantové oblasti může do SDF vnášet lokální minima, která mohou být menší než hledané minimum. To se může stát i přes stanovení spodní hranice τ, protože rozsah hledaných frekvencí se může z části pokrývat s formantovou oblastí. Jako řešení těchto problémů je zavedeno normování SDF (třetí krok):

$$d'_t[\tau] = \begin{cases} 1, & \text{pro } \tau = 0, \\ d_t[\tau]/(\frac{1}{\tau}\sum_{j=1}^{\tau} d_t[j]), & \text{jinak.} \end{cases} \tag{3.3}$$

Tato funkce začíná v hodnotě 1, čímž se eliminuje vyhodnocení $d_t[0]$ jako periodu a je menší než 1 jen tehdy, když $d_t[t]$ klesne pod průměrnou hodnotu (reprezentovanou jmenovatelem případu „jinak“ v rovnici výše). Chybovost klesá z 1,95 % na 1,69 %.

Čtvrtým krokem je stanovení prahové hodnoty (threshold – *THR*). Jako odhad základní periody T se pak bere nejmenší hodnota τ, pro kterou platí: $d'_t[\tau] < THR$. Chybovost klesá z 1,69 % na 0,78 %.

Pátým krokem je kvadratická interpolace lokálních minim $d'_t[\tau]$ a jejich okolí. Interpoluje se z toho důvodu, že kroky uvedené výše jsou účelné, jen pokud je hledaná perioda násobkem periody vzorkovací. Pokud není, může být výsledek vzdálen od pravdy až o $T_{\text{vz}}/2$. Interpolace je přesná, pokud signál neobsahuje vysoké harmonické složky (vyšší než $f_{\text{vz}}/4$). Po interpolaci chyba klesá jen o 0,01 %.

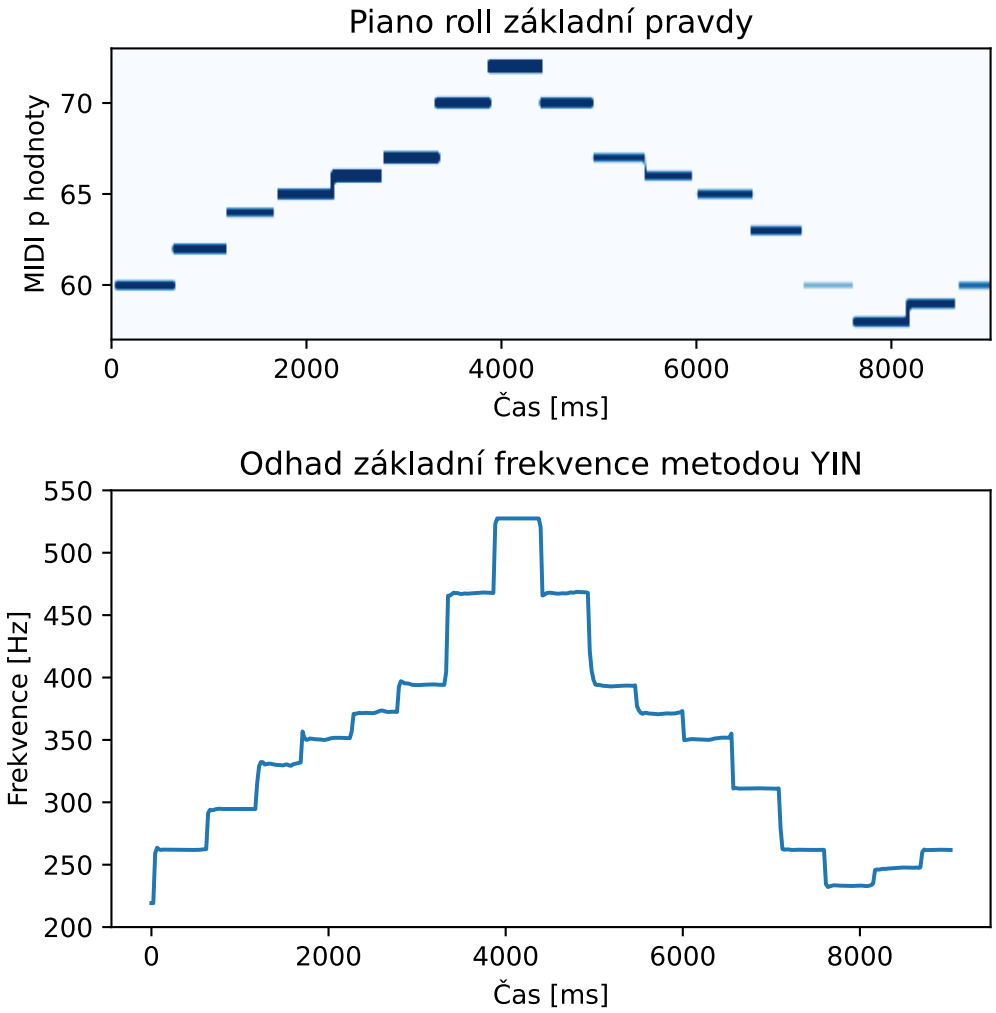
Posledním krokem je hledání nejlepšího lokálního odhadu periody T. Celý algoritmus popsaný výše se provede poprvé: Pro každý vzorek signálu s indexem t z určitého pásma se najde první odhad, pak se pásmo možných period zúží a algoritmus proběhne znovu. Podruhé algoritmus vyhledá konečný odhad základní periody s chybovostí jen 0,5 %.

Údaje o chybovosti pochází z testování metody na vybraných nahrávkách ze čtyř databází nahrávek řeči. Délka okna $W = 25$ ms, posun okna roven jednomu vzorku, rozsah kmitočtů od 40–8000 Hz a prahová hodnota $THR = 0{,}1$. Metoda YIN byla také testována na nahrávce sekvence, jejíž notový zápis je na obr. 2.3. Parametry předány do funkce `yin`: $f_{\min} = 220$ Hz, $f_{\max} = 523{,}25$ Hz, $f_{\text{vz}} = 48$ kHz, délka rámce = 4096 vzorků, délka okna $W = 2048$ vz, posun okna $H = 1024$ vz. Výsledek najdeme na obr. 3.1. Základní pravdou je MIDI soubor vytvořený systémem PianoTranscription.

Metoda pYIN

Modifikací, či rozvinutím metody YIN je pravděpodobnostní YIN. Z odstavce výše víme, že výsledek (odhad frekvence) metodou YIN závisí pouze na jednom časovém rámci signálu (např. 1024 vzorků = 23,2ms okno při $f_{\text{vz}} = 44{,}1$ kHz). Jinými slovy, YIN vrací jeden odhad frekvence, pro každý rámec, v našem případě jeden odhad frekvence každých 23,2 ms. Před vrácením odhadu periody signálu je stanovena prahová hodnota, minima SDF jsou interpolována (u pYIN taktéž) a hledá se nejmenší minimum pod prahovou hodnotou (viz výše). Podle [20] je tento postup nevýhodný, protože prahová hodnota se neopírá o jiné odhady periody (byť chybné) v důsledku analýzy po rámcích. Navíc, pokud SDF nemá minimum menší než prahová hodnota, jako odhad frekvence se bere absolutní minimum, což může vést k chybám.

Metoda pYIN sestává z dvou kroků. První krok vrací odhady periody a hodnoty jejich pravděpodobností, které vynikají právě z hodnoty prahové. Hodnoty pravdě-



Obr. 3.1: Základní pravda a odhad základní frekvence sekvence z obr. 2.3. Odstín modré barvy v horním grafu reprezentuje *velocity* – dynamiku (tmavá pro vysokou)

podobností se ve druhém kroku interpretují jako pozorovací hodnoty (*observation values*) pro skryté Markovovy modely (HMM – *hidden Markov models*, komentováno níže). Algoritmus metody pYIN začíná stejně jako YIN: počítá se autokorelační funkce a pomocí ní se z rovnice 3.2 získá SDF, následuje normalizace. Místo prahové hodnoty THR (funkce `librosa.yin()` nastavuje $THR = 0{,}1$ jako výchozí hodnotu) se použije její rozdělení pravděpodobnosti S_{THR}. V praxi se používá rozdělení pravděpodobnosti beta. Pravděpodobnost, že minimum SDF je v čase τ_0 je:

$$P(\tau = \tau_0 | S_{\text{THR}}, x_t) = \sum_{i=1}^{N} a(s_i, \tau) P(s_i) [Y(x_t, s_i) = \tau], \; P(s_i) \in S_{THR}, \tag{3.4}$$

$$a(s_i, \tau) = \begin{cases} 1, & \text{pro } d'(\tau) < s_i, \\ p_a, & \text{jinak}, \end{cases} \tag{3.5}$$

kde $S_{THR} = \{P(s_1), P(s_2), ..., P(s_N)\}$ je rozdělení pravděpodobnosti prahové hodnoty, p_a je pravděpodobnost, že minimum SDF je větší než prahová hodnota (strategie YIN) a hodnota hranaté závorky je buď 1 nebo 0, podle toho, zda výraz (odhad periody metodou YIN v čase x_t s prahovou hodnotou s_i) uvnitř je pravdivý nebo ne. Dvojice: odhad základní periody τ + hodnota pravděpodobnosti $P(\tau)$ jsou předávány do druhého kroku – HMM. Skryté Markovovy modely jsou pravděpodobnostními modely, podle kterých dokážeme předpovědět stav náhodné veličiny na základě stavu předešlého. Potřebujeme k tomu znát množinu stavů náhodné veličiny (pro nás množina možných základních kmitočtů tónů) a pozorovací hodnoty (získané v prvním kroku), podle kterých se snažíme určit stav skryté náhodné veličiny. Tyto hodnoty, hodnoty pravděpodobností přechodů stavů a počáteční stav se určí z pravděpodobností získaných v předchozím kroku pomocí rovnic a postupů popsaných v [20]. Skryté Markovovy modely se také hojně využívá v systémech pro rozpoznávání akordů (*Chord Recognition*). Popis HMM přesahuje obsah této práce. Metody YIN i pYIN se používají v analýze řeči a jiných aplikacích.

Kepstrum

Mnoho metod pro určování základní periody signálu pracuje s reprezentací zvanou „Kepstrum“. To je podle [21] definováno jako čtverec Fourierovy transformace amplitudového spektra (sekce 2.2.1) signálu, které bylo logaritmováno. Tato práce se kepstrální analýze nevěnuje. Informace o kepstru a jeho použití najdeme v [21] a [22].

3.1.2 Metody pracující v kmitočtové oblasti

V kmitočtové oblasti hledáme základní frekvenci signálu. Jako první řešení se nabízí provést STFT nahrávky a najít kmitočet odpovídající nejnižšímu maximu ve spektru. Tento přístup může být nepoužitelný, pokud nahrávka bude např. silně zašuměná nebo pokud zvukový vzorek bude obsahovat i subharmonické složky.

HPS – *Harmonic product spectrum*

Tato metoda se používá pro určování základního kmitočtu mj. v řečových signálech. Jedná se o podvzorkování spektra získaného pomocí DFT. Pokud počet vzorků snížíme na polovinu, druhá harmonická složka se ocitne na kmitočtu originálního fundamentu. Pokud podvzorkujeme na třetinu vzorků, třetí harmonická bude mít kmitočet fundamentu. Pro získání HPS se podvzorkováná spektra pronásobí [23]. HPS je definováno jako:

$$\begin{aligned} HPS(k) &= \prod_{m=1}^{M} |X(mk)| \\ k_0 &= \text{argmax}\{HPS(k)\}, \end{aligned} \tag{3.6}$$

kde $X(k)$ je spektrum signálu získané DFT, M je počet uvažovaných harmonických, počet podvzorkování, k je frekvenční pásmo DFT, ve kterém se nachází fundament. k_0 je třeba přepočítat na kmitočet pomocí vztahu 2.9. Vstupní signál se zde může před vypočtením spektra vynásobit okénkovou funkcí, aby HPS pak sledovalo časové změny. Jedná se vlastně o použití STFT místo DFT. Podle [23] tato metoda funguje i pro určení výšek tónů v polyfonické struktuře.

3.1.3 Metody pracující v kmitočtově-časové oblasti

Principem metod, které pracují v této oblasti je použití časových i frekvenčních prostředků: např. kmitočtová filtrace + časová metoda (ACF, SDF – kapitola 3.1). Tento přístup prezentuje metoda, která se snaží simulovat proces určení výšky tónu, který probíhá ve sluchovém ústrojí člověka (hlemýžďová filtrace + autokorelace) [24]. Hlavní myšlenkou je rozdělení signálu na kmitočtová pásma pomocí banky filtrů a na výstupu každého z nich se počítá funkce průměrného rozdílu velikosti (AMDF – *Average magnitude difference function*) Jedná se vlastně o ACF s operací odečítání místo násobení.

$$AMDF(\tau) = \int_{t}^{t+W} |S(t) - S(t+\tau)| \tag{3.7}$$

S ADMF výstupních signálů jednotlivých filtrů se pracuje různými způsoby: Jedním z nich je sečtení všech AMDF a hledání minima součtu funkcí. Druhým je normalizace hodnot jednotlivých AMDF (pomocí vydělení okamžité hodnoty průměrem hodnot ve vycentrovaném okně) a až poté sečtení a hledání minima. Dalším je sčítání AMDF z jednocestně usměrněných a vyhlazených (*smoothed*) signálů z filtrů.

3.2 Metody určování výšky vícera tónů – *Multi-pitch estimation*

3.2.1 Metody pracující v časové oblasti

Cílem těchto metod je nalezení dvou, nebo více základních period signálu. Používají k tomu funkce, které nabývají charakteristických hodnot v časech odpovídajícím periodám.

Metoda MMM

Metoda „MMM“ [25] byla vyvinuta v roce 2003. Jedná se vlastně o rozšíření či zobecnění metody YIN popsané výše (3.1.1) pro dva hlasy. Zdroj [25] uvádí, že metoda bude použitelná i na N hlasů, pokud všechny harmonické složky nebudou násobky

$N-1$ (nebo méně) frekvencí v rozsahu kmitočtů, který prohledáváme. Zkoumaný signál $z[t]$ o dvou neznámých periodách U a V se dá složit z dvou periodických signálů $x[t]$ a $y[t]$. Definice jeho periodicity je:

$$z[t] - z[t-V] - z[t-U] + z[t-V-U] = 0. \tag{3.8}$$

Rozšířením SDF pro dvojhlas (rov. 3.2) získáme funkci, která v ideálním případě bude mít hodnotu 0 v τ a ν rovným periodám a jejich násobkům:

$$d_t[\tau,\nu] = \sum_{j=t+1}^{t+W} (z[t] - z[t-V] - z[t-U] + z[t-V-U])^2 = 0. \tag{3.9}$$

Jako odhady period se označuje nejmenší τ a ν, pro které platí uvedená rovnice. Zdroj [25] uvádí, že pokud $x[t]$ a $y[t]$ jsou periodické, funkce garantuje nalezení jejich period až na krajní případ. Krajní případ nastává, když všechny kmitočty, které hledáme jsou násobky jednoho z nich. Pro snížení výpočetní náročnosti se rovnice výše rozšiřuje obdobně jako rov. 3.2, tedy se sumace vyjádří pomocí hodnot autokorelační funkce. Stejně jako u metody YIN se minima rozšířené SDF interpolují parabolou. Podmínku $d_t[\tau,\nu] = 0$ splňují hodnoty podél obou os, pokud τ nebo ν $= 0$. Proto se funkci 3.9 normuje:

$$d'_t[\tau,\nu] = \begin{cases} 1, & \text{pro } \tau = 0 \text{ nebo } \nu = 0, \\ d_t[\tau,\nu] / (\frac{1}{\tau} \sum_{j=1}^{\tau} d_t[j,\nu]), & \text{jinak,} \end{cases} \tag{3.10}$$

$$d''_t[\tau,\nu] = \begin{cases} 1, & \text{pro } \tau = 0 \text{ nebo } \nu = 0, \\ d'_t[\tau,\nu] / (\frac{1}{\nu} \sum_{j=1}^{\tau} d'_t[\tau,j]), & \text{jinak.} \end{cases} \tag{3.11}$$

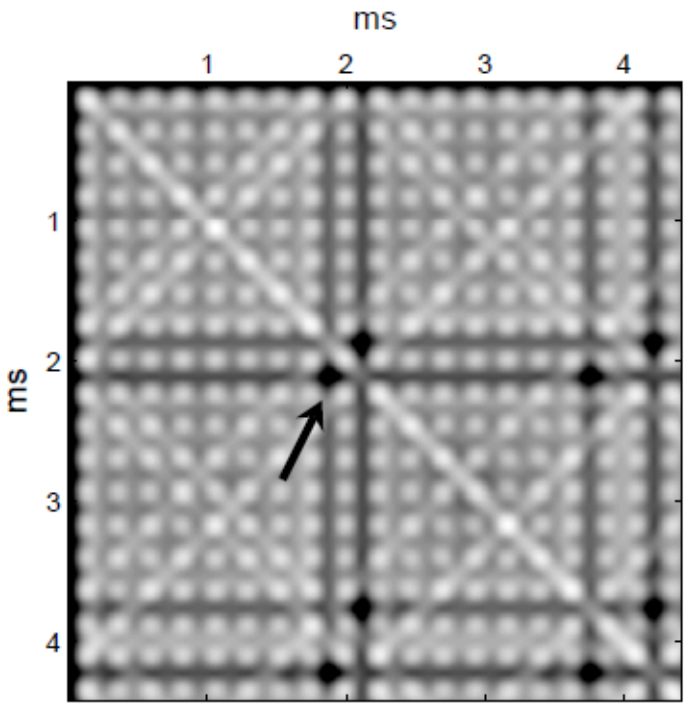
Tímto se zbavíme minim, které mohou být chybně vyhodnoceny jako periody složeného signálu (obdobně jako u YIN rov. 3.3). Výsledná d''_t má minimum v periodách $[U,\ V]$ a jejich násobcích $[kU,\ jV]$. Pro eliminaci násobků period s stanoví prahová hodnota (*THR)*, známá z YIN. Jako periody signálu se pak vyhodnotí minima s nejmenšími souřadnicemi, pro které $d''_t < THR$. Grafické znázornění metody MMM, bez normování, interpolace a *THR* je na obr. 3.2.

3.2.2 Metody pracující v kmitočtové oblasti

Metoda sčítání modulů harmonických složek

Hlavní myšlenkou je výpočet *salience* (významnost, síla) pro daný kmitočet. Významnost je sumací váhovaných součtů amplitud harmonických složek daného kmitočtu (diskrétní verze):

$$s(\tau) = \sum_{m=1}^{M} g(\tau,m) \max_{k \in \kappa_{\tau,m}} |Y(k)|, \tag{3.12}$$



Obr. 3.2: Výstup metody MMM, převzato z [10]. Obě osy znázorňují čas

kde $g(\tau, m)$ je váhovací funkce, $Y(k)$ je vyhlazené spektrum získané DFT a operátor max vybere maximální modul z rozsahu $\kappa_{\tau,m}$. Vstupní signál je pomocí FFT převeden do kmitočtové roviny a spektrum se vyhladí. Z vyhlazeného spektra se získá první odhad f0 a spočítá se zmíněná významnost. Spektrum $X(k)$ se vyhlazuje po pásmech pomocí filtrace bankou pásmových propustí se šířkami pásem odpovídajícími pásmům kritickým. Hodnota kompresního koeficientu γ je získána ze směrodatných odchylek a pásmových kompresních koeficientů jednotlivých filtrů (vztahy najdeme v [26]). Vliv filtru na vstupní signál v kmitočtové oblasti získáme násobením: $Y(k) = \gamma(k)X(k)$. Váhová funkce $g(\tau, m)$ byla získána s použitím trénovacího datasetu ve formě 1000 nahrávek různých počtů nástrojů (1, 2, 3, 4, 6) s určenými referenčními základními kmitočty. Proběhly odhady základních frekvencí pomocí významnosti. Pro nahrávky jednoho tónu se jako f0 považovala maximální hodnota významnosti, pro n tónů se bralo v úvahu n největších hodnot. Odhady byly srovnány s referenčními hodnotami a váhová funkce se postupně upravovala, až se snížila chybovost na nejmenší hodnotu. Jeden z prezentovaných způsobů získávání funkce:

$$g(\tau, m) = g_1(\tau)g_2(m), \tag{3.13}$$

kde g_1 je lineární interpolace 10 rovnoměrně rozložených fundamentů z rozsahu 30–2500 Hz a g_2 je lineární interpolace podobně rozložených harmonických složek. Hodnoty g_1 a g_2 byly před úpravami nastaveny na 1. Zdroj [26] naznačuje i jiné způsoby výpočtu a zacházení s váhovacími funkcemi v této metodě. Metoda sčítání amplitud má mimo své přímé verze popsané výše také verzi, kde se odhad provádí opakovaně a odstraňuje se jej pro další iteraci. Je to toho důvodu, že dvě maxima ve významnostní funkci mohou patřit jednomu fundamentu. Odhad frekvence a její smazání je hlavním principem systému implementovaného pro tuto práci.

Metoda spektrálních špiček – Analyzátor akordů klavíru

Způsob, jak zjistit složení akordu, který se zdá nejintuitivnějším, je pohlédnutí na signál v kmitočtové rovině a zaznamenání kmitočtů tří nebo čtyř nejnižších harmonických složek. Tohle se nedá udělat, pokud analyzovaný zdroj signálu produkuje i subharmonické složky, což se ale u klavíru neděje. Tento postup se hojně využívá při sledování melodie skladby v čase a funguje i pro analýzu akordů: Provádí se krátkodobá Fourierova transformace (sekce 2.3.1) a označí se nejnižší harmonickou. Jelikož vstupem do analyzátoru je nahrávka jediného akordu, který se v čase nemění, zaniká potřeba použití STFT a spektrum je získáno pomocí FFT s délkou rovnou délce signálu. Hlavní funkci algoritmu naznačuje výpis 3.1.

```
x = load(audiofile)
Spektrum = fft(x)
Spektrum2 = Spektrum.copy()
f = []
while i <= pocet_tonu:
    f0_candidate = argmax(Spektrum)
    if Spektrum2[f0_candidate/2] > Spektrum2[f0_candidate/0.89]:
        Spektrum[f0_candidate] = 0
        if f0_candidate in f:
            continue
        else:
            f0_candidate /= 2
    f.append(f0_candidate)
    Spektrum[f0_candidate] = 0
    i += 1
return f
```

Výpis 3.1: Zjednodušený náčrt hlavní funkce algoritmu

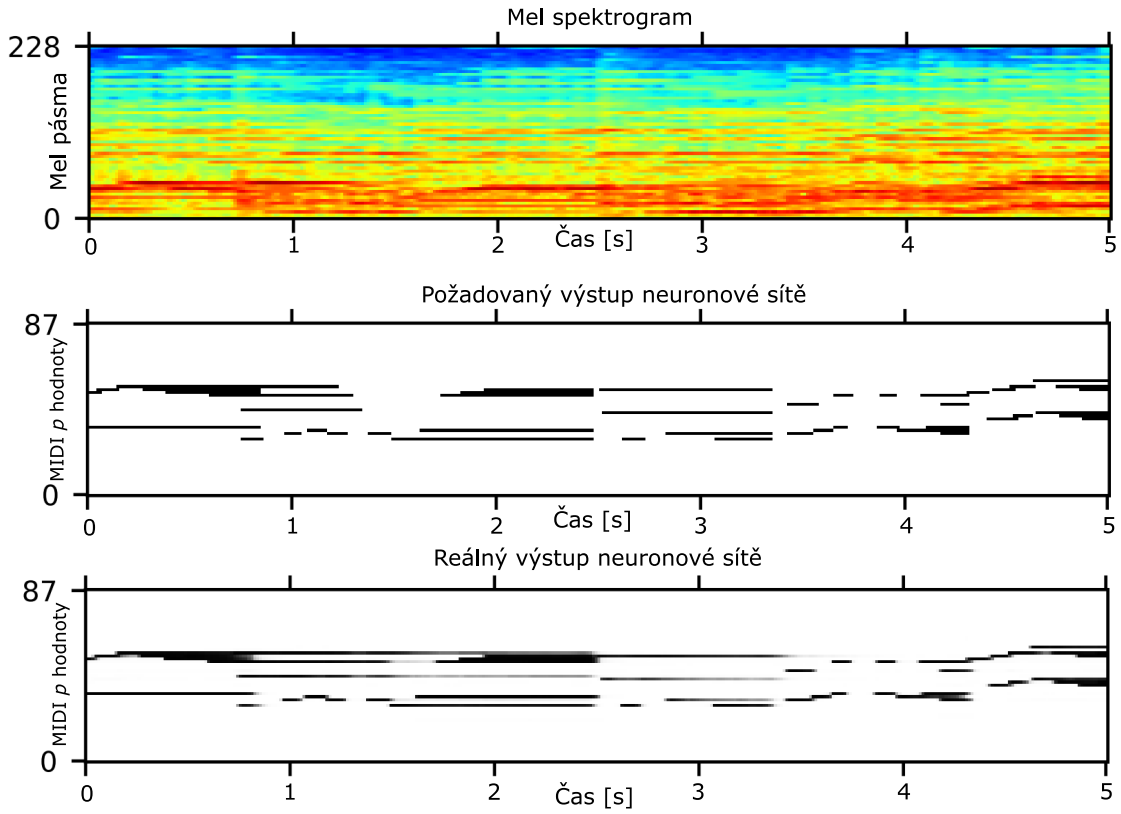
Nejprve je vzorek načten a je spočtena jeho DFT. Pokud to uživatel nastaví, je možné zobrazit si modulové spektrum nahrávky. Spektrum je kmitočtově omezené na pásmo 20 Hz – 20 kHz, protože část spektra mimo zmíněné pásmo pro lidský sluch

nemá význam. Poté se vytvoří kopie spektra, `Spektrum2`, které slouží pouze pro kontrolu. V cyklu, který se provede tolikrát, kolik je tónů v akordu, se najde maximum ve spektru. Jelikož není pravidlem, že fundament tónu má vždy největší modul, je třeba se ujistit, že nezaznamenáváme druhou harmonickou (třetí harmonické už svým modulem většinou nepřesahují první a druhou). Toto ujištění se provede tak, že se srovná modul na kmitočtu rovném polovině kandidáta na f0 (kmitočet spektrálního maxima) s modulem na kmitočtu v neharmonickém poměru ke kmitočtu kandidáta na f0. Zde byla použita hodnota 1/0,89. Pokud systém zaznamenal, že na polovičním kmitočtu kandidáta je důležitá složka, ujistí se ještě, že tuto složku již nezachytil a pokud ne, vloží do výstupní matice `f` polovinu kandidáta na f0. Poté se zaznamenaný fundament odfiltruje spolu s případnými chybnými odhady základního kmitočtu, které byly odchyceny a cyklus se může opakovat. Výstupní matice je přepočítána pomocí funkcí z knihovny `librosa` na MIDI p hodnoty a na řetězce, reprezentující jednotlivé tóny. Tyto matice jsou poté předávány do funkcí, které z rozdílu jednotlivých MIDI p hodnot určí typ a obrat akordu. Tónina je určena ze zmíněných tónových řetězců. Pro čtyřzvuky je připravena funkce `typ_obrat_matcher`, která obsahuje slovník, jehož klíče jsou trojice čísel pro daný obrat daného typu septakordu/čtyřzvuku a hodnotami jsou názvy příslušných akordů a kódy týkající se typu a obratu: `{(4, 3, 4) : ["tvrdě velký septakord", 5, 0, "maj"]}`. Tato funkce využívá toho, že vzdálenosti jednotlivých tónů od sebe jsou stejné, nezávisle na tónině a umístění akordu na klavíru. Pro trojzvuky je rezervována funkce pracující na velmi podobném principu. Umístění akordu na klávesnici se určí pomocí oktávy nejnižšího tónu, která je ukryta na prvním místě tónového řetězce. Pokyny pro výstupy systémů hledajících základní frekvence v polyfonické struktuře najdeme např. na webu [27]. Podle tohoto doporučení má systém vracet časovou značku + odhady daného počtu frekvencí, které zazněly v tomto čase. Náš systém nevrací časové značky, protože se kmitočty jednotlivých harmonických složek v čase nemění.

Hlavním rozdílem mezi naším Analyzátorem akordů klavíru a jinými systémy pro určování výšek tónů je schopnost přesného určování obratů, umístění na klávesnici a použitelnost pouze na stacionární signály. Systém nabízí také variantu pro určování akordů v reálném čase při použití mikrofonu. Analyzátor se spustí, s malou prodlevou je spuštěno i nahrávání, které probíhá po rámcích o délce 1024 vzorků (23 ms pro f_{vz} = 44,1 kHz) a v terminálu se každou sekundu vypisují kytarové značky akordů, jejich obraty a oktávy. Limitací této metody je neschopnost určení počtu zahraných tónů. Počet musí určit uživatel. Pokud se tak nestane, je pro určení počtu tónů použit v úvodu zmíněný algoritmus PianoTranscription (viz níže), který slouží pro přepis klavírních nahrávek do MIDI souborů. Stává se, že systém vyhodnotí počet tónů špatně protože není určen pro krátké nahrávky, ve kterých se prakticky nic nemění. Pokud systém určí počet tónů špatně, přenese se chyba do určování kmitočtů.

3.2.3 Systém PianoTranscription – *state-of-the-art* řešení

V práci bude dále pro systém PianoTranscritpion [28] používána zkratka „PTR“. Systém vznikl v roce 2021. Už tehdy byly nejúspěšnější v hledání výšky tónů neuronové sítě. Neuronová síť je struktura navzájem propojených buněk – neuronů, která dokáže na základě vstupu vydat určitý typ výstupu (rozpoznání objektu, predikce trendu, výstupu systému, rozpoznání kmitočtového složení signálu aj). Inspirací k její vytvoření byla funkce lidského mozku. Neuronová síť se skládá z vrstev neuronů – vstupní, výstupní a skryté. Na vstupní vrstvu se vkládá vstupní data, pro nás např. jeden časový rámec spektrogramu. Na výstupu očekáváme zvolený typ výstupu podle toho, k čemu je síť určena. V našem případě očekáváme vektor hodnot fundamentů akordu. Před tím, než je síť použitelná, musí být natrénovaná. Trénování probíhá tak, že se parametry sítě (váhy neuronů, biasy, prahové hodnoty) nejprve zvolí náhodně, na vstup jsou vložena data a výstup sítě se porovná se správnou odpovědí (pro nás tři/čtyři hodnoty kmitočtů dle základní pravdy). Spočítá se chybovost a pak se mění parametry jednotlivých neuronů tak, aby se chybovost snížila, a to pro různá vstupní data sítě. Trénování probíhá tak dlouho, dokud se chyba nepřestane snižovat. Pokud na vstup vložíme zmíněný spektrogram s logaritmickou frekvenční osou, tedy matici s 88 řádky (každý pro jednu klávesu) a jedním sloupcem, každý řádek matice bude mít hodnotu úměrnou hodnotě spektrogramu v odpovídajícím kmitočtovém pásmu. Výstup sítě se porovná se správnou odpovědí (pro Cdur kvintakord ve 4. oktávě bude mít matice základní pravdy hodnoty 1 na řádcích č. 60, 64, 67, jinde 0). Po vypočtení chybovosti se váhy neuronů přehodnotí (např. neuron pro C8 ve vstupní vrstvě bude mít po natrénování nižší váhu než neuron pro C4, protože C4 se v akordech vyskytuje mnohem častěji). Po natrénování neuronová síť na výstupu vrátí matici ideálně s hodnotami 0 pro klávesy, které nejsou stlačeny a 1 pro aktivní klapky (viz obr. 3.3). Existují různé typy neuronových sítí např. konvoluční, zpětnovazební aj. podle jejich vnitřní struktury. Hlubší popis neuronových sítí přesahuje rámec této práce. Neuronová síť systému PTR byla trénovaná na rozpoznávání různých parametrů not na klavírním datasetu MAESTRO [29]. Dataset MAESTRO obsahuje více než 168 hodin klavírních nahrávek, přičemž každá z nich disponuje svým MIDI zápisem. Byl nahráván na klavírech Yamada Disclavier, což jsou akustické klavíry se zabudovanými senzory, které mapují pohyb kladívek, pedálů aj. Výstup z senzorů se průběžně zapisuje do MIDI souboru. Yamaha Disklavier dokáže také MIDI soubor zpětně reprodukovat, má totiž i hrací mechanismus.



Obr. 3.3: Příklad použití neuronové sítě, převzato z [28]. Barva v prvním grafu má význam PSD [dB] (nejsilnější červená, nejslabší modrá)

Systém PTR vkládá na vstup své neuronové sítě spektrogram s logaritmickou frekvenční osou popsanou v jednotkách mel. Neuronová síť, vrátí výšky aktivních tónů. Určování výšek tónů není jediným procesem, kde systém používá neuronovou síť. Používá se ji také pro určení *velocity* (MIDI reprezentace dynamiky), časů nástupů a zániků tónu aj. Přesnost použitého spektrogramu je limitována hodnotou posunu okna. Kompromis mezi dlouhým a krátkým oknem zmíněný v kap. 2.3.1 je řešen inovativním přístupem k určování časů nástupů not, proto je použito okno o délce 2048 vzorků.

4 Použité knihovny a balíčky

V této krátké kapitole budou popsány programovací moduly a knihovny, použité v implementací systému.

`Librosa`[1] [30] – balíček pro analýzu audio souborů v programovacím jazyce Python. Obsahuje funkce pro tvorbu systému pro MIR. Nabízí funkce pro tvorbu spektrogramu, chromagramu, funkce převodu veličin (kmitočet na MIDI p hodnoty, MIDI p na názvy not v rovnoměrně temperovaném ladění, názvy not na jejich kmitočet), funkce pro načítání audio souborů, funkci pro výpočet autokorelační funkce aj. Disponuje také přímými implementacemi některých metod, např. YIN, pYIN.

`PyAudio`[2] [31] – knihovna, která umožňuje signálové propojení skriptu a zvukového rozhraní počítače. Nabízí funkce pro nahrávání signálu z mikrofonu, přehrávání, filtraci v reálném čase (před záznamem), generování zvukového signálu, vykreslování průběhů aj. V Analyzátoru akordů klavíru je použita stejnojmenná třída této knihovny, `pyaudio.PyAudio` pro čtení výstupu z mikrofonu a analýzu akordů v reálném čase.

`Numpy`[3] [32] – knihovna pro různé matematické operace. Nabízí funkce pro výpočty prostých úloh, jako např. vrácení absolutní hodnoty čísla či funkce pro složité matematické transformace (FFT). V práci použita na vyhledávání argumentu největší hodnoty funkce, převod do absolutních hodnot a formátování polí.

`SciPy`[4] [33] – knihovna pro technické a vědecké výpočty. Pomocí `SciPy` se dá vizualizovat data a manipulovat s nimi. Knihovna nabízí balíčky mj. také pro číslicové zpracování signálu `scipy.signal` nebo pro transformace z časové oblasti do frekvenční a zpět – `scipy.fftpack`. V práci je použitá pro výpočet Fourierovy transformace signálu pomocí FFT.

`Matplotlib`[5] [34] – knihovna pro vizualizaci dat a vykreslování grafů. Umožňuje jejich přiblížení, nastavování velikostí, nastavování měřítek os aj. Nabízí vykreslovaní 2D i 3D grafů. V této práci jsou trojrozměrné grafy vyobrazovány pomocí barvy. Knihovna byla použita pro vizualizaci metody YIN, krátkodobé Fourierovy transformace, chromagramu i volitelnou vizualizaci spektra akordu v skriptu `trojzvuky.py`.

`Tkinter`[6] [35] – knihovna pro tvorbu grafického uživatelského rozhraní. Nabízí funkce pro přípravu oken, tlačítek, polí pro vkládání textu, výpisových, varovných oken aj. V této práci je balíček použit pro demonstrační verzi Analyzátrou akordů klavíru.

[1] `https://librosa.org/doc/latest/index.html` [cit. 17-04-2023]
[2] `https://people.csail.mit.edu/hubert/pyaudio/docs/`[cit. 17-04-2023]
[3] `https://numpy.org/doc/`[cit. 17-04-2023]
[4] `https://docs.scipy.org/doc/scipy/`[cit. 17-04-2023]
[5] `https://matplotlib.org/stable/index.html`[cit. 17-04-2023]
[6] `https://docs.python.org/3/library/tk.html`[cit. 17-04-2023]

`mir_eval`[7] [36] – knihovna, ve které najdeme funkce pro většinu nejčastěji používaných metrik v MIR. Nabízí kvalitativní i kvantitativní metriky pro většinu MIR procesů: určování tempa, akordů, tóniny, znělosti, dob, transkripce. V práci je použita funkce `mir_eval.multipitch.metrics()`.
`PianoTranscription`[8] [28] – implementace transkripčního systému pro převod klavírní nahrávky do MIDI zápisu. V práci použito pro určování počtu tónů akordu a pro srovnání s implementovaným systémem.

[7] `https://craffel.github.io/mir_eval/#`[cit. 17-04-2023]
[8] `https://github.com/bytedance/piano_transcription`[cit. 17-04-2023]

5 Výsledky

V této kapitole budou srovnány metoda sčítání modulů harmonických složek, metoda spektrálních špiček (kap. 3.2.2) a *state-of-the-art* systém PTR (kap. 3.2.3). Metody byly testovány na položkách testovacího datasetu, vytvořeného pro potřeby této práce (kap. 1.6).

Základní pravda u všech testů byla odvozena od názvů položek datasetu funkcí `anotator()`. Funkce vrací pole s kmitočtem, nebo kmitočty stanovenými MIDI doporučením [8]. Odhad frekvence se počítá za úspěšný, pokud se hodnota neliší od referenční hodnoty o více než 3 %, což odpovídá cca čtvrttónu. Je nutné zmínit, že jeden z klavírů, z nichž pochází vzorky datasetu je ve druhé polovině své nejvyšší oktávy mírně rozladěn (o cca půltón). To může lehce zkreslovat výsledky testů. Funkce `anotator()` vrátí přesné kmitočty kláves, které byly stlačeny, zvukově však kmitočty těmto klávesám neodpovídají. Takovýchto případů je v datasetu 15.

5.1 Použité metriky

Metriky, kterými hodnotíme testované metody pochází z „MIR xChange – MIREX“ z let 2017 a 2021 [27]. MIREX je soutěž pořádaná komunitou MIR. Pro všechny aktivity jsou stanoveny přesné metriky a údaje, podle kterých se hodnotí celková úspěšnost a chybovost metod. Metriky zde budou vysvětleny stručně, detailní komentář najdeme v [37] a [38]. Těmito zdroji se také řídí námi použitá funkce `mir_eval.multipitch.metrics()`. Než metriky popíšeme, nastíníme evaluační případy, které mohou nastat. Dodáme, že cílem systému je vrácení matice kmitočtů, jejichž hodnoty se liší nejvýše o 3 % od příslušných hodnot základní pravdy a jejíž rozměr je stejný jako rozměr matice základní pravdy. Pro ilustrační účely vektor kmitočtů základní pravdy jako $\mathbf{P} = [440, 554, 659]$, a výstupní matici systému $\mathbf{V}$.

- *True positive* (TP) – Odhad kmitočtu se shoduje s referenčním kmitočtem. Splňuje 3% pravidlo a byl vrácen ve výstupní matici systému, $\mathbf{V} = [440, 554, 659]$.
- *False positive* (FP) – Falešně pozitivní odhad. Jeho hodnota nesplňuje 3% pravidlo, do výstupní matice nepatří, ale je v ní vrácen, $\mathbf{V} = [440, 554, 659, 1318]$. Tento případ se někdy nazývá *false alarm* – „planý poplach“.
- *False negative* (FN) – Falešně negativní odhad kmitočtu. Odhad splňuje 3% pravidlo ale do výstupní matice nebyl vložen (např. kvůli chybnému určení počtu tónů), $\mathbf{V} = [440, 554]$.
- *Substitution error* – Chybný odhad. Jedná se o spojení FN a FP. Správný odhad ve výstupní matici chybí a je vložen odhad chybný. Počet hodnot výstupní matice se shoduje s počtem hodnot matice základní pravdy, jedna nebo více hodnot nesplňuje 3% pravidlo, $\mathbf{V} = [440, 554, 800]$.

Metriky, kterými se přesnost a použitelnost hodnotí:
Accuracy – přesnost, na rozdíl od dvou dalších metrik dává údaj o celkové úspěšnosti systému. Dvě další musí být uváděny v páru, samy dávají jen částečnou informaci.
Precission – preciznost, udává poměr počtu úspěšných odhadů frekvence k počtu všech vrácených.
Recall – výtěžnost, dává údaj o tom, jaká část správných odhadů byla systémem vrácena.
Hodnoty všech tří výše zmíněných se pohybují v rozmezí 0–1 a lze je vyjadřovat v procentech.

$$Accuracy = \frac{\sum_{n=1}^{N} TP(n)}{\sum_{n=1}^{N} TP(n) + FP(n) + FN(n)}, \tag{5.1}$$

$$Precission = \frac{\sum_{n=1}^{N} TP(n)}{\sum_{n=1}^{N} TP(n) + FP(n)}, \tag{5.2}$$

$$Recall = \frac{\sum_{n=1}^{N} TP(n)}{\sum_{n=1}^{N} TP(n) + FN(n)}, \tag{5.3}$$

kde N je celkový počet nahrávek, na kterých byl systém testován. Je důležité zmínit, že v případě běžných systémů, které se nezaměřují pouze na akordy, ale sledují základní kmitočty měnící se v čase, je místo n časový údaj t (udávaný např. v milisekundách).
Metriky značící chybovost se vztahují k celkovému souboru hodnot vráceného během testu na datasetu (ne pro 1 akord, jak je tomu u TP, FP a FN):

- E_{sub} – Substituční chyba. Určuje, u jaké části odhadů nastal chybný odhad (viz výše).
- E_{miss} – Chyba vynechání. Určuje, jaká část fundamentů nebyla systémem vrácena, v jaké části odhadů nastal případ FN.
- E_{fa} – Chyba přidání. Určuje, v jaké části odhadů nastal případ FP.
- E_{tot} – Součet tří zmíněných

$$E_{\text{sub}} = \frac{\sum_{n=1}^{N} \min(K_{ref}(n), K(n)) - K_{TP}(n)}{\sum_{n=1}^{N} K_{\text{ref}}(n)}, \tag{5.4}$$

$$E_{\text{miss}} = \frac{\sum_{n=1}^{N} \max(0, K_{ref}(n) - K(n)}{\sum_{n=1}^{N} K_{ref}(n)}, \tag{5.5}$$

$$E_{\text{fa}} = \frac{\sum_{n=1}^{N} \max(0, K(n) - K_{ref}(n)}{\sum_{n=1}^{N} K_{ref}(n)}, \tag{5.6}$$

kde N je počet akordů testování, $K(n)$ je počet odhadů základních kmitočtů pro akord n, $K_{ref}(n)$ je počet kmitočtů základní pravdy pro akord n, $K_{TP}(n)$ je počet správných odhadů kmitočtu pro akord n. Z důvodu mezinárodního značení, anglické názvy metrik nebudou v práci dále překládány.

5.2 Evaluace testů metod

Na 514 nahrávkách akordů a 16 nahrávkách jednotlivých tónů proběhly testy metod pro určení výšek tónů. Systémy byly implementovány v jazyce Python. V této sekci srovnáme implementaci metody sčítání modulů harmonických složek[1] Gregoryho Burleta, implementaci metody spektrálních špiček, vytvořenou pro potřeby této práce a profesionální systém PTR. Zmíněná implementace metody sčítání modulů harmonických složek vrací matici odhadu frekvencí v určených časových intervalech. Jako výstupní množina frekvencí pro test byla uvažována matice, která se mezi odhady vyskytovala nejčastěji. Frekvenční rozsah ve zdrojovém kódu byl upraven pro klavír.

Zde je nutné připomenout, že limitací Analyzátoru akordů klavíru (kap. 3.2.2) je určování počtu tónů, proto byl test rozdělen na 2 situace. V první z nich je počet zahraných tónů odvozen z názvu právě testovaného audio souboru, což reprezentuje případ, kdy počet tónu zadá uživatel. V druhém případě je počet tónu určen systémem PTR. Limitace PTR jsou tedy i limitacemi Analyzátoru akordů klavíru. Výsledky testů najdeme v tab. 5.1.

Tab. 5.1: Srovnání metod na celém datasetu

Metoda	*Accuracy*	*Precission*	*Recall*	E_{sub}	E_{miss}	E_{fa}	E_{tot}
Spektrální špičky Počet tónů znám	0,55	0,71	0,71	0,29	0,0	0,0	0,29
Spektrální špičky Počet tónu neznám	0,52	0,69	0,69	0,24	0,07	0,07	0,38
PTR	0,74	0,83	0,88	0,08	0,05	0,11	0,23
Moduly harmonických	0,26	0,37	0,47	0,52	0,01	0,28	0,81

Je zde vidět, že mezi testovanými metodami jednoznačně vítězí PTR. Je to proto, že používá natrénovanou neuronovou síť, což je dnes *state-of-the-art* řešení ve většině procesů MIR. Když srovnáme *accuracy* v prvních dvou řádcích tab. 5.1, zjistíme, že celková přesnost systému spektrálních špiček vzrostla jen o 0,02 při známém počtu tónů oproti stavu s neznámým počtem. Rozdíl 0,03 (3 %) odpovídá při 530položkovém datasetu (1587 fundamentů) 16 vzorkům (zaokrouhleno nahoru). Chybovost zde klesá o 9 %. Druhý a třetí řádek tabulky výše nabízí srovnání mechanismu pro určování fundamentů v PTR (používá také mechanismy pro určování počtu not, jejich délek, aktivity pedálů aj.) s mechanismem, který používá Analyzátor akordů klavíru. Test, jehož výstupem jsou druhý a třetí řádek, probíhal tak, že PTR nejprve určil počet tónů, ten se předal do Analyzátoru akordů klavíru a pak proběhly

[1] `https://github.com/gburlet/multi-f0-estimation`[cit. 01-05-2023]

Tab. 5.2: Srovnání metod na části datasetu

Metoda	*Accuracy*	*Precission*	*Recall*	E_{sub}	E_{miss}	E_{fa}	E_{tot}
Spektrální špičky Počet tónů znám	0,90	0,95	0,95	0,05	0,0	0,0	0,05
Spektrální špičky Počet tónů neznám	0,87	0,95	0,91	0,03	0,06	0,02	0,11
PTR	0,87	0,92	0,94	0,03	0,04	0,06	0,12
Moduly harmonických	0,43	0,55	0,67	0,31	0,02	0,25	0,57

dva odhady fundamentů, tzn., že oba mechanismy pro určování fundamentů měly identické vstupy (nahrávku a počet tónů). Poslední řádek obsahuje relativně malé hodnoty přesnosti a velkou chybovost. Je tomu tak proto, že váhová funkce používaná ve výpočtu významnosti (viz kap. 3.2.2) byla získaná z kmitočtově omezeného datasetu (30–2500 Hz). Povšimněme si ještě hodnoty E_{miss} v posledním řádku. Je nejnižší ze všech (mimo prvního řádku, kde ale k vynechání či přidání odhadu nemohlo dojít) a informuje nás o tom, že metoda sčítání modulů harmonických složek dobře detekuje počet zahraných tónů. Na druhou stranu hodnota E_{fa} je zde největší. Tato hodnota v tomto případě reprezentuje situaci, kdy metoda vrátí ve své výstupní matici vícekrát jeden kmitočet.

Bylo zjištěno, že další limitací metody spektrálních špiček je frekvenční minimum, pod kterým jsou odhady fundamentů chybné. Experimentálně bylo odhaleno, že k chybným odhadům fundamentů dochází u akordů, které obsahují tóny z množiny tónů {A0, C♯3}, jejichž struny jsou uchyceny v pravém segmentu litinového rámu (obr. 5.1). Abychom zjistili, jak poroste úspěšnost, pokud budeme testovat pouze akordy, které neobsahují tóny z této množiny (335 vzorků), byl proveden druhý test. Z tabulky 5.2 lze odvodit, že pokud netestujeme akordy, které obsahují struny uchycené v pravé části litinového rámu, je celková úspěšnost mechanismu hledání fundamentů v Analyzátoru akordů klavírů stejná, jako úspěšnost mechanismu v PTR. Celková chybovost našeho analyzátoru je dokonce o 1 % nižší. Zde je nutno podotknout, že PTR má jiné výhody a je určen pro přepis audio nahrávky do MIDI v čase. Důvodem tohoto 1% rozdílu je to, že PTR někdy najde ve zvuku úderu kladívka do struny tónovou složku (byť velmi krátkou) a její kmitočet považuje za fundament. S tím Analyzátor akordů klavíru počítá (viz rozdíl hodnot E_{fa} ve druhém a třetím řádku tab. 5.2) a jako počet tónů v akordu vezme pouze počet not vrácených PTR, delších než 0,5 s.

Struny jsou na spodní části rámu uchyceny ve dvou sekcích, viz obr. 5.1. V červeně označené části jsou uchyceny všechny jednochórové struny a všechny dvojchó-

rové bez dvou dvojic. Zvuk pocházející z těchto strun má velmi silnou ruchovou složku. To může být způsobeno silným opředením těchto strun. Jelikož se jedná o nejdelší struny na klavíru, mají mnoho harmonických, které jsou navíc velmi silné a doba kontaktu kladívka se strunou je mnohem menší než perioda jejich kmitů (sekce 1.2.3). Jejich moduly jsou mnohdy silnější než moduly fundamentů. Funkce `f0_finder()` (viz výpis 3.1) testuje, zda náhodou nezachytil druhou harmonickou složku zahraného tónu jako spektrální maximum srovnáním jejího modulu s modulem na polovičním kmitočtu. U některých vzorků je třetí harmonická složka výraznější než první a druhá, tudíž ji algoritmus zachytí. Srovná její modul s modulem na polovičním kmitočtu, což není kmitočet fundamentu, ale jeho 1,5násobek (neharmonický poměr). Kdyby funkce testovala i na třetí harmonickou, musela by modul vybrané složky srovnávat s modulem na třetinovém kmitočtu. Tento proces v našem Analyzátoru akordů klavíru neprobíhá, protože těchto krajních případů není mnoho.

Obr. 5.1: Rozdělení strun na rámu. Akordy pocházející ze strun uchycených v červeně označené části rámu byly chybně určeny

Jelikož Analyzátor akordů klavíru byl vyvíjen s použitím jednotlivých vzorků testovacího datasetu (kap. 1.6), proběhl další test na datasetu, který naše zkoumané systémy nikdy neanalyzovaly – *Piano Triad Waveset* (dataset klavírních trojzvuků)[2] Davida Robertse. Tento dataset obsahuje 360 trojzvuků nahraných na digitálním klavíru (klasických, zvětšených a zmenšených) s nejnižšími tóny v 1.–6. oktávě klavíru. Základní pravda byla získána z přiloženého `.csv` souboru. Výsledky najdeme v tab. 5.3. Analýza celého datasetu metodou spektrálních špiček se známým počtem tónů trvala 45 sekund, metodou spektrálních špiček s neznámým počtem tónů a PTR cca tři a čtvrt hodiny a metodu sčítání modulů harmonických složek více než 14 hodin. Pro vysokou časovou, výpočetní náročnost a nízkou úspěšnost byla z testů na druhém datasetu vynechána poslední metoda ze zmíněných. U obou systémů vidíme, že přesnosti jsou dosti nízké a chybovost vysoká. Díky poznatkům získaným při testu

Tab. 5.3: Srovnání metod na druhém datasetu

Metoda	*Accuracy*	*Precission*	*Recall*	E_{sub}	E_{miss}	E_{fa}	E_{tot}
Spektrální špičky Počet tónů znám	0,64	0,78	0,78	0,22	0,0	0,0	0,22
Spektrální špičky Počet tónů neznám	0,24	0,79	0,26	0,06	0,68	0,0	0,75
PTR	0,31	0,94	0,32	0,02	0,67	0,0	0,69

na celém datasetu vytvořeném pro tuto práci byl proveden také druhý test na druhém datasetu s frekvenčním omezením (vynechány akordy, které obsahují nižší tóny než C♯3 – 120 položek). Pohledem druhý a třetí řádek tab. 5.4 zjistíme, že úspěšnost systému PTR a Analyzátoru akordů klavíru je o dost menší než při testech na prvním datasetu i přes vynechání „problematických“ – nízkých kmitočtů.

Tab. 5.4: Srovnání metod na části druhého datasetu

Metoda	*Accuracy*	*Precission*	*Recall*	E_{sub}	E_{miss}	E_{fa}	E_{tot}
Spektrální špičky Počet tónů znám	0,82	0,90	0,90	0,10	0,0	0,0	0,10
Spektrální špičky Počet tónů neznám	0,29	0,95	0,3	0,01	0,69	0,01	0,71
PTR	0,31	0,96	0,31	0,01	0,68	0,01	0,69

[2] `https://www.kaggle.com/datasets/davidbroberts/piano-triads-wavset`

Relativně vysoké hodnoty *precission* nás informují o tom, že většina z provedených odhadů je správně, ale nízký *recall* naznačuje to, že velká část fundamentů nebyla zachycena. Systém spektrálních špiček má v situaci známého počtu tónů výsledky srovnatelné s prvním datasetem. Jeho implementace hledá tolik základních kmitočtů, kolik je tonů v akordu (přičemž minimálně jeden odhad proběhne pokaždé) a tento údaj v druhé situaci získává pomocí PTR. Z toho zde plyne, že PTR cca 70 % fundamentů z druhého datasetu nezachycuje. Experimentálně bylo zjištěno, že se jedná o akordy s tóny ze čtvrté oktávy klavíru a nižšími. Je to pravděpodobně způsobeno tím, že PTR nebyl trénován na nahrávkách digitálního klavíru, ale na nahrávkách klavíru Yamaha Disklavier.

6 Závěr

Tato bakalářská se věnovala určování akordů se zaměřením na klavírní nahrávky pomocí odhadu jejich jednotlivých základních kmitočtů. V teoretické části byl popsán klavír, bylo objasněno názvosloví akordů doplněné příklady a byl vytvořen 530položkový dataset, na kterém probíhaly testy. Dále byly specifikovány použité signálové reprezentace. S jejich využitím byly v praktické části nastíněny některé metody pro určení výšky jednoho tónu: průchody nulou, ACF, YIN, pYIN, kepstrum, HPS a AMDF. Dále byly popsány některé metody pro určování výšek tónů v polyfonické struktuře: MMM, metoda sčítání modulů harmonických složek, navržená a implementovaná v jazyce Python metoda spektrálních špiček a *state-of-the-art* řešení – neuronová síť použitá v systému PTR. Všechny zmíněné metody pro určování výšky vícera tónů mimo MMM byly testovány na datasetu. Pro test metody sčítání modulů harmonických složek byla vybrána implementace Gregoryho Burleta z roku 2012. Limitací Analyzátoru akordů klavíru je určování počtu tónů, proto byl jeho test rozdělen na dva případy: počet tónů zadán uživatelem a počet tónů určen systémem PTR. Nejvyšší celkovou přesnost a nejmenší chybovost na vytvořeném datasetu vykázal systém PTR ($Accuracy = 0{,}74$, $E_{\text{tot}} = 0{,}23$), druhým nejpřesnějším se ukázal systém spektrálních špiček se známým počtem tónů ($Accuracy = 0{,}54$, $E_{\text{tot}} = 0{,}3$), s neznámým počtem tónů ($Accuracy = 0{,}52$ $E_{\text{tot}} = 0{,}38$) a metoda sčítání modulů harmonických složek ($Accuracy = 0{,}26$ $E_{\text{tot}} = 0{,}81$), která je ze zmíněných časově jednoznačně nejnáročnější. Také se ukázalo, že tato metoda není vhodná pro nahrávky klavíru, protože je už ve stádiu svého vývoje kmitočtově omezená na pásmo menší, než je rozsah klavíru. Během testu byla zjištěna druhá, frekvenční limitace metody spektrálních špiček, způsobená pravděpodobně mechanickou konstrukcí klavíru (jeho litinového rámu). Akordy, jejichž tóny pochází ze strun uchycených v pravé části rámu (viz obr. 5.1) byly určovány zcela chybně. Tyto struny totiž vykazují silnou ruchovou složku při úderu kladívka do struny a také jejich vyšší harmonické složky (zpravidla třetí, ty Analyzátor akordů klavíru nekontroluje) stanoví spektrální špičku. Proto byl spuštěn druhý test, který probíhal pouze na akordech složených z tónů vyšších než C♯3 (138,59 Hz). Tímto celkové úspěšnosti vzrostly a nejpřesnější se ukázala metoda spektrálních špiček se známým počtem tónů ($Accuracy = 0{,}9$), stejná metoda s neznámým počtem tónů a PTR mají stejnou přesnost 0,87. Metody byly srovnány i na jiném datasetu trojzvuků. Při testu na druhém datasetu bylo nalezeno další frekvenční minimum, které se týká obou srovnávaných systémů a pohybuje se kolem kmitočtu 525 Hz (C5). Je to pravděpodobně způsobeno použitím digitálního klavíru, ze kterého pochází vzorky druhého datasetu. Systém PTR byl natrénován na nahrávkách z akustického klavíru.

V rámci dalšího vývoje analyzátoru akordů klavíru by bylo dobré zlepšit kontrolu proti zachycení vyšších harmonických složek a najít způsob určování počtu tónů. Řešením obou úloh by mohla být neuronová síť natrénovaná na různých nahrávkách akordů (vysokých i nízkých, s různým počtem tónů). Vstupem sítě by byl spektrogramy s různými délkami oken, aby síť byla schopna zachytit jak nízké tak vysoké kmitočty. Jiným vstupem by mohl být také spektrogram získaný z CQT, která používá proměnlivou délku okna podle analyzovaných kmitočtů. Dále by se dalo systém rozšířit i na analýzu nonových akordů.

Literatura

[1] KURFÜRST, Pavel. *Hudební nástroje*. Praha: Togga, 2002. ISBN 80-902-9121-X.

[2] REBLITZ, Arthur A. *Piano servicing, tuning, and rebuilding: For the professional, the student, and the hobbyist.* 2nd edition. Lanham: VESTAL PRESS, 1993. ISBN 1-879511-03-7.

[3] JOSEF, Prach. *STAVBA KLAVÍRŮ A PIANIN a její problematika*. 1. Praha: Státní pedagogické nakladatelství, 1987. ISBN Není.

[4] SYROVÝ, Václav. *Hudební akustika.* 1. Praha: Akademie múzických umění v Praze, 2013. ISBN 978-80-7331-297-8.

[5] SMÉKAL, Zdeněk. *Analýza signálů a soustav - BASS* [online]. Brno: Vysoké učení techické v Brně, 2012 [cit. 2022-10-20]. ISBN 978-80-214-4453-9. Dostupné z: databáze skript VUT v Brně

[6] SCHIMMEL, Jiří. *Elektroakustika 1* [online]. 1. Brno: VUT v Brně, 2013 [cit. 2022-11-30]. ISBN 978-80-214-4716-5. Dostupné z: `https://www.vut.cz/vav/vysledky/detail?vav_id=99790#vysledek-99790`

[7] SCHIMMEL, Jiří. *Studiová a hudební elektronika* [online]. Druhé. Brno: FEKT, 2015 [cit. 2022-10-16]. ISBN 978-80-214-4452-2. Dostupné z: databáze skript VUT v Brně

[8] *The Complete MIDI 1.0 Detailed Specification: Incorporating all Recommended Practices*. 3rd ed. Los Angeles: The MIDI Manufacturers Association, 2014. ISBN 9780972883108.

[9] ZENKL, Luděk. *ABC Hudební nauky*. 8. Praha: Editio Bärenreiter Praha, 2003. ISBN 80-86385-21-3.

[10] WANG, DeLiang a Guy J. BROWN. *Multiple F0 Estimation. Computational Auditory Scene Analysis*. IEEE, 2006, 2011, 45-79. ISBN 9780470043387. Dostupné z: doi:10.1109/9780470043387.ch2

[11] MÜLLER, Meinard. *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. Cham: Springer, 2015. ISBN 978-3-319-21945-5..

[12] VRBA, Kamil a Jiří MIŠUREC. *Technika analogových obvodů*. Brno: VUTIUM, 2020 [cit. 2022-10-31]. ISBN 978-80-214-5901-4. Dostupné z: `https://dspace.vutbr.cz/bitstream/handle/11012/195807/ANAebook.pdf?\sequence=1&isAllowed=y`

[13] MIŠUREC, Jiří a Zdeněk SMÉKAL. Číslicové zpracování signálů [online]. 1. Brno, 2011 [cit. 2022-10-31]. Dostupné z: https://moodle.vut.cz/pluginfile.php/386820/mod_resource/content/1/DATA/\Skriptum_BCZS_2011.pdf

[14] OPPENHEIM, Alan V. *Speech spectrograms using the fast Fourier transform*. IEEE Spectrum. 1970, 7(8), 57-62. ISSN 0018-9235. Dostupné z: doi:10.1109/MSPEC.1970.5213512

[15] FUJISHIMA, Takuya. *Realtime Chord Recognition of Musical Sound: a System Using Common Lisp Music.* ICMC Proceedings [online]. Stanford, 2000, 1999, 1999, 464-467 [cit. 2022-11-26]. Dostupné z: https://quod.lib.umich.edu/i/icmc/bbp2372.1999.446/--realtime-chord-recognition-of-musical-sound-a-system-using?view=image

[16] BROWN, Judith C. *Calculation of a constant Q spectral transform*. The Journal of the Acoustical Society of America. 1991, 89(1), 425-434. ISSN 0001-4966. Dostupné z: doi:10.1121/1.400476

[17] BROWN, Judith C. a Miller S. PUCKETTE. *An efficient algorithm for the calculation of a constant Q transform*. The Journal of the Acoustical Society of America [online]. 1992, 92(5), 2698-2701 [cit. 2022-11-30]. ISSN 0001-4966. Dostupné z: doi:10.1121/1.404385

[18] MATUŠTÍK, Daniel. *Určování základního hlasového tónu*. Brno, 2013. Diplomová práce. VUT v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky. Vedoucí práce Milan Sigmund.

[19] DE CHEVEIGNÉ, Alain a Hideki KAWAHARA. *YIN, a fundamental frequency estimator for speech and music*. The Journal of the Acoustical Society of America [online]. 2002, 111(4), 1917-1930 [cit. 2022-11-20]. ISSN 0001-4966. Dostupné z: doi:10.1121/1.1458024

[20] MAUCH, Matthias a Simon DIXON. *PYIN: A fundamental frequency estimator using probabilistic threshold distributions*. 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) [online]. IEEE, 2014, 4.5.2014, 659-663 [cit. 2023-04-15]. ISBN 978-1-4799-2893-4. Dostupné z: doi:10.1109/ICASSP.2014.6853678

[21] NOLL, A. Michael. *Cepstrum Pitch Determination*. The Journal of the Acoustical Society of America. 1967, 41(2), 293-309. ISSN 0001-4966. Dostupné z: doi:10.1121/1.1910339

[22] NOLL, A. M. a M. R. SCHROEDER. *Short-Time "Cepstrum" Pitch Detection.* The Journal of the Acoustical Society of America. 1964, 36(5), 1030-1030. ISSN 0001-4966. Dostupné z: doi:10.1121/1.2143271

[23] LEE, Kyogu. *Automatic Chord Recognition from Audio Using Enhanced Pitch Class Profile.* 1. Stanford: ICMC Proceedings, 2006.

[24] DE CHEVEIGNÉ, Alain. *Speech f0 extraction based on Licklider's pitch perception model.* ICPhS. 4. Paříž, 1993, 1993(4), 218-221.

[25] DE CHEVEIGNÉ, Alain a Alexis BASKIND. *F0 estimation of one or several voices.* 8th European Conference on Speech Communication and Technology (Eurospeech 2003). ISCA: ISCA, 2003, 2003-9-1, 833-836. Dostupné z: doi:10.21437/Eurospeech.2003-187

[26] KLAPURI, Anssi. *Multiple Fundamental Frequency Estimation by Summing Harmonic Amplitudes.* Proceedings of the 7th International Conference on Music Information Retrieval, ISMIR [online]. 2006, 8.10.2006, 7(7), 216-221 [cit. 2023-04-16]. Dostupné z: doi:10.5281/zenodo.1416740 [cit. 2023-04-16].

[27] *Mirex Wiki 2021: Multiple Fundamental Frequency Estimation & Tracking* [online]. 2021 [cit. 2022-12-03]. Dostupné z: `https://www.music-ir.org/mirex/wiki/2021:Multiple_Fundamental_Frequency_Estimation_%26_Tracking`

[28] Qiuqiang Kong, Bochen Li, Xuchen Song, Yuan Wan, and Yuxuan Wang. *High-resolution Piano Transcription with Pedals by Regressing Onsets and Offsets Times.* 2020, arXiv preprint arXiv:2010.01815. Dostupné z:`https://arxiv.org/pdf/2010.01815.pdf`

[29] HAWTHRONE, Curtis, Andriy STASYUK, Adam ROBERTS, et al. *Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset.* International Conference on Learning Representations (ICLR). New Orleans, 2018. Dostupné z: doi:10.5281/zenodo.4660569

[30] McFee, B., Metsai, A., McVicar, M., Balke, S., Thomé, C., Raffel, C., Zalkow, F., Malek, A., Dana, Lee, K., Nieto, O., Ellis, D., Mason, J., Battenberg, E., Seyfarth, S., Yamamoto, R., Viktorandreevichmorozov, Choi, K., Moore, J., Bittner, R., Hidaka, S., Wei, Z., Nullmightybofo, Weiss, A., Hereñú, D., Stöter, F., Nickel, L., Friesch, P., Vollrath, M. a Kim, T. *librosa/librosa: 0.9.2.* (Zenodo, 2022, 6). Dostupné z: `https://doi.org/10.5281/zenodo.6759664` [cit. 2022-12-01]

[31] GIANNAKOPOULOS, Theodoros a Gianni PAVAN. *PyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis*. PLOS ONE [online]. 2015, 10(12), 1 [cit. 2023-04-17]. ISSN 1932-6203. Dostupné z: doi:10.1371/journal.pone.0144610

[32] HARRIS, Charles R., K. Jarrod MILLMAN, Stéfan J. VAN DER WALT, et al. *Array programming with NumPy*. Nature. 2020, 585(7825), 357-362. ISSN 0028-0836. Dostupné z: doi:10.1038/s41586-020-2649-2 [cit. 2022-12-01]

[33] VIRTANEN, Pauli, Ralf GOMMERS, Travis E. OLIPHANT, et al. *SciPy 1.0: fundamental algorithms for scientific computing in Python*. Nature Methods. 2020, 17(3), 261-272. ISSN 1548-7091. Dostupné z: doi:10.1038/s41592-019-0686-2 [cit. 2022-12-01]

[34] HUNTER, John D. *Matplotlib: A 2D Graphics Environment*. Computing in Science & Engineering. 2007, 9(3), 90-95. ISSN 1521-9615. Dostupné z: doi:10.1109/MCSE.2007.55 [cit. 2022-12-01]

[35] SHIPMAN, John W. T*kinter 8.4 reference: a GUI for Python*[online]. 2. Mexiko: New Mexico Tech, 2010 [cit. 2023-04-17]. Dostupné z: `https://www.academia.edu/4903094/Tkinter_8_4_reference_a_GUI_for_Python`

[36] RAFFAEL, Collin, Brian MCFEE, Eric J. HUMPHREY, Justin SALAMON, Oriol NIETO, Dawen LIANG a Daniel P. W. ELLIS. *MIR_EVAL: A Transparent Implementation of Common MIR Metrics*. Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR [online]. Taipei, Taiwan, 2014, 27.10.2014, 1-6 [cit. 2023-04-17]. Dostupné z: `doi:10.5281/zenodo.1416528`

[37] POLINER, Graham E. a Daniel P. W. ELLIS. A *Discriminative Model for Polyphonic Piano Transcription*. EURASIP Journal on Advances in Signal Processing [online]. 2006, 2007(1) [cit. 2023-04-20]. ISSN 1687-6180. Dostupné z: doi:10.1155/2007/48317

[38] BAY, Mert, Andreas F. EHMANN a J. Stephen DOWNIE. *Evaluation of multiple f0-estimation and tracking systems*. 10th International Society for Music Information Retrieval Conference (ISMIR 2009) [online]. Japonsko, 2009, 2009, (1), 315-320 [cit. 2023-04-20]. Dostupné z: doi:10.5281/zenodo.1418241

Seznam symbolů a zkratek

f0/F0 Základní kmitočet/fundament

FT *Fourier transform* – Fourierova transformace

DFT *Discrete Fourier transform* – Diskrétní Fourierova transformace

FŘ Fourierova řada

DFŘ Diskrétní Fourierova řada

DTFT *Discrete-time Fourier transform* – Fourierova transformace s diskrétním časem

FFT *Fast Fourier transform* – rychlá Fourierova transformace

STFT *Short-Term Fourier transform* – krátkodobá Fourierova transformace

ACF *Autocorrelation function* – Autokorelační funkce

SDF *Squared difference function* – funkce kvadratického rozdíu

PSD *Power spectral density* – výkonová spektrální hustota

HMM *Hidden Marcov models* – skryté Markovovy modely

AMDF *Average magnitude difference function* – funkce rozdílu průměrné velikosti

HPS *Harmonic product spectrum* – Pronásobené harmonické spektrum

ADSR *Attack, deckay, sustain, release* – Nástup, útlum, podržení, uvolnění

CQT *Constant Q-transform* – Konstantní Q transformace

PTR *PianoTranscription* – Název systému pro přepis klavírní nahrávky do MIDI

A Obsah elektronické přílohy

- `dataset` – složka s datasetem
- `multi-f0-estimation-master` – složka se zdrojovým kódem `bc_metoda2.py` a soubory potřebnými pro spuštění analýzy akordů pomocí metody sčítání modulů harmonických složek (kap. 3.2.2).
- `piano_transcription-master` – složka se zdrojovým kódem a soubory potřebnými pro používání systému `PianoTranscription`. Ve složce je také kód s funkcí pro určení základní pravdy `anotator.py`
- `analyzator_akordu_klaviru_demo.py` – Soubor pro demonstrační verzi Analyzátoru akordů klavíru pro testy na položkách datasetu nebo analýzu v reálném čase
- `bc_metoda1_ptr.py` – Soubor s kódem pro spuštění všech testů metody spektrálních špiček a PTR
- `gui.py` – zdrojový kód funkcí pro grafické rozhraní
- `mikrofon.py` – zdrojový kód s funkcemi pro analýzu výstupu z mikrofonu v reálném čase
- `trojzvuky.py` – zdrojový kód se všemi potřebnými funkcemi, třídami atd.
- `requirements.txt` – seznam využívaných knihoven
- `README.txt` – návod na spuštění a používání kódů