

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2017

Bc. Karel Čapek



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

VYTVOŘENÍ WEBOVÉ APLIKACE PRO OBJEKTIVNÍ ANALÝZU HYPOKINETICKÉ DYSARTRIE VE FRAMEWORKU DJANGO

DJANGO FRAMEWORK BASED WEB APPLICATION FOR OBJECTIVE ANALYSIS OF HYPOKINETIC
DYSARTHRIA

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Karel Čapek

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Zoltán Galáž

BRNO 2017



Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Karel Čapek

ID: 145980

Ročník: 2

Akademický rok: 2016/17

NÁZEV TÉMATU:

Vytvoření webové aplikace pro objektivní analýzu hypokinérické dysartrie ve frameworku Django

POKyny PRO VYPRACOVÁNÍ:

V rámci diplomové práce budou nastudovány a implementovány metody analýzy řeči v časové a frekvenční oblasti. Dále bude vytvořena webová aplikace ve frameworku Django sloužící pro analýzu dysartrické řeči.

DOPORUČENÁ LITERATURA:

[1] PELED, Abraham. Digital signal processing: theory, design, and implementation. New York: John Wiley & Sons, 1976.

[2] PSUTKA, Josef. Mluvíme s počítačem česky. Praha: Academia, 2006. Česká matice technická, roč. 111, č. spisu 502. ISBN 80-200-1309-1.

Termín zadání: 1.2.2017

Termín odevzdání: 24.5.2017

Vedoucí práce: Ing. Zoltán Galáž

Konzultant:

doc. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato diplomová práce se zabývá výpočtem parametrů, které by byly schopny rozlišit zdravou řeč a řeč postiženou hypokinetickou dysartrií. Byla zde nastudována hypokinetická dysatrie, což je motorická porucha řečového a hlasového ústrojí. Byly nastudovány metody zpracování řečového signálu. Dále byly nastudovány parametry, které by mohly dobře rozlišovat zdravou a nemocnou řeč. Následně byly tyto parametry naprogramovány v programovacím jazyce Python. Dalším krokem bylo vytvoření webové aplikace ve frameworku Django, která slouží pro analýzu dyzartické řeči.

KLÍČOVÁ SLOVA

Hypokinetická dysartrie, Parkinsonova nemoc, segmentace, řečový signál, tvorba řeči, řečové ústrojí, Python, Numpy, Matplotlib, Django

ABSTRACT

This master's thesis deals with the calculation of parameters that would be able to differentiate healthy speech and speech impaired by hypokinetic dysarthria. There was staged hypokinetic dysarthria, which is a motoric disorder of speech and vocal tract. Were studied speech signal processing methods. Further parameters were studied, which could well differentiate healthy and diseased speech. Subsequently, these parameters were programmed in Python programming language. The next step was to create a web application in Django framework, which is used for the analysis of the dyzartic speech.

KEYWORDS

Hypokinetic dysarthria, Parkinson's disease, segmentation, speech signal, creation of speech, speech tract, Python, Numpy, Matplotlib, Django.

ČAPEK, K. *Vytvoření webové aplikace pro objektivní analýzu hypokinetické dysartrie ve frameworku Django*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 51 s. Vedoucí diplomové práce Ing. Zoltán Galáž.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma Vytvoření webové aplikace pro objektivní analýzu hypokinetické dysartrie ve frameworku Django jsem vypracoval samostatně pod vedením vedoucího diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové 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 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.

V Brně dne

.....

(podpis autora)

Výzkum popsáný v této diplomové práci byl realizován v laboratořích podpořených projektem Centrum senzorických, informačních a komunikačních systémů (SIX); registrační číslo CZ.1.05/2.1.00/03.0072, operačního programu Výzkum a vývoj pro inovace.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu mé diplomové práce, panu Ing. Zoltánu Galážovi, za odborné vedení, konzultace, trpělivost a poskytnutí cenných rad a materiálů potřebných pro realizaci této diplomové práce.

V Brně dne

.....

(podpis autora)

OBSAH

Seznam obrázků	11
Úvod	12
1 Hypokinetická dysartrie	14
1.1 Porucha fonace (dysfonie)	14
1.2 Porucha artikulace.....	15
1.3 Porucha prozódie	15
2 Parametrizace řeči	16
2.1 Předzpracování signálu	16
2.2 Parametry založené na časové oblasti.....	18
2.2.1 Jitter	18
2.2.2 Shimmer.....	18
2.2.3 Short-time energy (STE).....	19
2.2.4 Low short-time energy ratio (LSTER).....	19
2.2.5 Zero-crossing rate (ZCR).....	19
2.2.6 Teager-Kaiser energy operator (TKEO).....	20
2.3 Parametry založené na frekvenční oblasti	20
2.3.1 Spectral centroid (SC).....	20
2.3.2 Spectral spread (SS).....	21
2.3.3 Spectral flux (SF).....	21
2.3.4 Spectral roll-off (SR)	21
2.3.5 Spectral flatness (SL).....	22
3 Příklady vypočtených parametrů	23
3.1 Parametry v časové oblasti	25
3.2 Parametry ve frekvenční oblasti	28
4 Praktická část	32
4.1 Django.....	32
4.1.1 Důležité skripty.....	32

4.1.2 MVC	32
4.2 HTML	33
4.3 CSS	34
4.4 Blokové schéma	35
4.5 Popis systému	36
4.6 Aplikace	36
4.7 Spuštění aplikace	38
4.8 Ukázka aplikace	39
5 Závěr	42
Použitá literatura	44
Seznam symbolů, veličin a zkratk	46
Seznam příloh	48
A Obsah přiloženého DVD	49
B Ukázka programu PyCharm	50
C Ukázka výstupního souboru	51

SEZNAM OBRÁZKŮ

Obr. 2.1: Segmentovaný signál a) bez překrytí, b) s překrytím.....	17
Obr. 3.1: Časový průběh zkušební nahrávky: <i>alkoholi8k.wav</i>	23
Obr. 3.2: První segment použité nahrávky.	24
Obr. 3.3: Krátkodobá energie	26
Obr. 3.4: Počet průchodů nulovou úrovní.....	26
Obr. 3.5: Teager-Kaiserův energetický operátor	26
Obr. 3.6: Spektrální těžiště	30
Obr. 3.7: Spektrální rozsah	30
Obr. 3.8: Spektrální tok.....	30
Obr. 3.9: Spektrální roll-off	31
Obr. 3.10: Spektrální plochost	31
Obr. 4.1 Architektura MVC (Model-View-Controller)	33
Obr. 4.2 Hierarchie programu	37
Obr. 4.3 Propojení jednotlivých tlačítek aplikace s funkcemi	37
Obr. 4.4 Úvodní stránka aplikace.	39
Obr. 4.5 Druhá stránka aplikace po stisknutí tlačítka Nahrát.	40
Obr. 4.6 Třetí stránka s po stisknutí tlačítka Vypočítej.	40
Obr. 4.7 Stránka s výsledky.	41
Obr. B.1 Ukázka programovacího prostředí PyCharm.....	50
Obr. C.1 Ukázka výstupního textového souboru	51

ÚVOD

Řeč je jedním ze základních dorozumívacích prostředků člověka. V řeči je člověk schopen nejen vyjádřit svoji myšlenku, ale také třeba momentální emoční stav. Lidé s Parkinsonovou nemocí přesněji s hypokinetickou dysartrií, která se ve velké míře u těchto pacientů vyskytuje, však tyto schopnosti většinou ztrácí. Hypokinetická dysartrie je tedy motorická porucha řečového a hlasového ústrojí, při nichž jsou v různé míře narušeny jednotlivé dílčí složky řeči, kterými jsou například fonace, artikulace či prozódie. Řeč pacientů s touto chorobou se tedy stává méně srozumitelná. Hlas může být nevýrazný, tichý a monotónní, díky čemuž může řeč ztratit veškeré řečníkovi emoce. Dále bývá pozorován drsný, chraptivý hlas a problematické je pro ně i zesílení či zeslabení hlasu. U pacientů se často vyskytuje neefektivní hospodaření s dechem a se stává, že náhle zvyšují a snižují tempo řeči, vytváří velké pauzy a na jeden nádech vysloví jen malý počet slabik.

Velkou nevýhodou určování této choroby je subjektivita vyšetřujících lékařů. Každý lékař může projevy této choroby vnímat s různou odlišností. V dnešní době je proto snahou nalézt neinvazivní metody určování této choroby pomocí analýzy řečového projevu. Pro určení hypokinetické dysartrie je potřeba nalézt parametry, které budou nejlépe oddělovat zdravou řeč a řeč postiženou touto chorobou. Velkou výhodou této metody by mělo být objektivní posouzení řečníka za základě předem naučených výsledků z řad jak pacientů s hypokinetickou dysartrií tak i zdravých řečníků. Pro přesné stanovení výsledků je však zapotřebí velké množství těchto osob.

Jako nejlepší pro stanovení základních parametrů se zdá být vyslovení hlásky „a“. Tato hláska se v řečových cvičeních vyskytuje nejvíce. Jeho výhodou je nezávislost na demografických a lingvistických charakteristikách jednotlivých řečníků. Pro účely této práce je však použita věta „Alkoholik je člověk, který pije víc než jeho lékař“ a to z důvodu lepšího zobrazení výsledků.

V první kapitole je zmíněna samotná Parkinsonova choroba, především pak její působení na hlasové ústrojí v podobě hypokinetické dysartrie. Druhá kapitola pojednává o parametrech, které by dobře mohly rozlišit zdravou a postiženou řeč. Jsou zde

parametry vypočtené jak z časové tak i frekvenční oblasti řečového signálu. Třetí kapitola ukazuje jednotlivé kódy parametrů vytvořené v programovacím jazyce Python, popřípadě i výsledky zobrazené jako grafy průběhů těchto parametrů. Ve čtvrté kapitole jsou popsány programovací jazyky, ve kterých je tato práce realizovaná. Je zde také blokové schéma samotného programu a popis jednotlivých dílčích částí. Na konci je ukázán i reálný výpočet samotných parametrů z ukázkové hlasové nahrávky.

1 HYPOKINETICKÁ DYSARTRIE

Parkinsonova nemoc (dále jen PN) je neurodegenerativní onemocnění, které má charakteristické důsledky na hybnosti zasaženého člověka. Zpravidla se u těchto pacientů (70-80%) vyskytuje i porucha řečového a hlasového ústrojí, která se nazývá hypokinetická dysartrie (dále jen HD). HD však mohou způsobit i úrazy hlavy např. při autonehodě [16].

Shopnost mluvit je jedinečný a komplexní proces, který je rozdělen do několika částí, a to: dýchání, fonace, artikulace a prozodie. HD postihuje všechny tyto části potřebné k tvorbě řeči. Poruchy řeči lze shrnout jako: hasový třes, hypomimie, dysfonie, dysprozodie, nazalita, artikulační problémy, palilalie, bradyfemie a tachyfemie. Pacienti s těmito příznaky mohou být také považováni svým okolím za méně inteligentní či dokonce dementní jedince, což může mít neblahý dopad na jejich každodenní život [6][8].

Právě těmito poruchami hlasového a řečového ústrojí z hlediska zpracování řečových signálů se bude zabírat tato diplomová práce.

1.1 Porucha fonace (dysfonie)

Fonací se rozumí kmitání hlasivek a tím tedy tvorba hlasu jako takového. Je to zpravidla první oblast, ve které se HD projevuje. Svalstvo hlasivek bývá ztuhlé a z toho důvodu nepracují hlasivky správně. Proudění vzduchu z plic také výrazně kolísá, což se ve výsledku projevuje hlasovým třesem pacienta.

U pacientů můžeme pozorovat hypofonii. Hypofonie je snížení intenzity hlasu, hlas je tichý, nevýrazný a monotóní.

Jako další je zde přítomná dysfonie, což je narušení kvality řeči. U pacientů s dysfonií bývá pozorován drsný, chraptivý hlas. Problematické je pro ně také zesílení nebo zeslabení hlasu.

Řečníci také neefektivně hospodaří s dechem. U pacientů se stává, že náhle zvyšují a snižují tempo řeči, vytváří velké pauzy a na jeden nádech vysloví jen malý počet slabik [16].

1.2 Porucha artikulace

Projevuje se nepřesnou artikulací řečníka. Opět je příčinou nesprávná práce svalstva účastnícího se tvorby řeči. Jedná se v tomto případě o aktivní artikulační orgány, kterými jsou zejména jazyk, rty a měkké patro. Řečníci nejsou v tomto případě schopni využít v plném rozsahu své artikulační orgány k dotvoření řečového signálu do podoby srozumitelné posluchači.

Největší problémy mají řečníci při výslovnosti ploziv (p, t, k, b, d, g). K hodnocení kvality a preciznosti artikulace se využívají tzv. Diadocho-kinetické úlohy (DDK). V těchto úlohách řečníci vyslovují opakující se slabiky v kombinaci ploziva a samohlásky např. „pa ta ka pa ta ka ...“. Při vyslovení řady těchto kombinací dochází k prudkým změnám v postavení artikulačních orgánů. Pacienti v některých případech nejsou schopni tyto rychlé změny provádět, v jiných případech jsou schopni kombinace těchto slabik vyslovit avšak na úkor snížené pohyblivosti mluvidel. To má za následek zvýšenou nesrozumitelnost [16].

1.3 Porucha prozodie

Mezi prozodické faktory patří zejména kmitočet základního tónu, intenzita a tempo řeči. Jedná se o tzv. suprasegmentální parametry. Mezi další prozodické jevy patří také např. rytmus, melodie a intonace. U pacientů s PN je většina těchto parametrů narušena. Takto poškozený řečový projev je nazýván jako tzv. dysprozodický.

Prozodické faktory jsou významným nositelem emočního vyjádření mluvčího. Dodávají význam či zdůrazňují obsah vyjádření. Pacienti s PN proto příliš v řeči neumí vyjádřit aktivní emoce jako je např. radost, vztek, údiv, atd [16].

2 PARAMETRIZACE ŘEČI

V této kapitole jsou popsány parametry, které by mohli co nejlépe rozpoznat zdravou řeč od řeči postižené hypokinetickou dysartrií. Analýza řeči bude probíhat v časové a frekvenční oblasti. Názvy jednotlivých parametrů jsou zde uváděny v originálním anglickém jazyce. Tyto názvy jsou v oblasti akustického zpracování dysartrické řeči ustálené a běžně používané. U každého parametru je však uveden i jeho český ekvivalent, avšak překlad do českého jazyka je spíše subjektivní a každý čtenář si jej může vyložit jinak.

2.1 Předzpracování signálu

Na řečový signál je po použítu preemfázové filtrace, která je potřebná ke kompenzaci kmitočtové charakteristiky prostředí. Tato filtrace potlačí vyšší kmitočtové složky spektra řečového signálu. Jde tedy o zvýraznění vyšších frekvenčních složek signálu za účelem vyrovnání kmitočtového spektra řeči. Pro potlačení těchto vlivů se používá filtr typu horní propust (FIR). Přenosová funkce preemfázového filtru má tvar [4]:

$$H(z) = 1 - a_1 z^{-1}, \quad (2.1)$$

kde a_1 je tzv. koeficient filtru a běžně nabývá hodnot 0,9 až 1.

Dále je potřeba z řečového signálu odstranit tzv. stejnosměrnou složku. Tato složka je totiž pro určení parametrů nepodstatná a v některých případech může být dokonce nežádoucí, proto je vhodné tuto složku odstranit. Její odstranění neboli tzv. ustředění se provádí jednoduchým odečtením stejnosměrné složky od signálu. Známe-li však celý vstupní řečový signál předem, což je i případ této diplomové práce pak platí [4]:

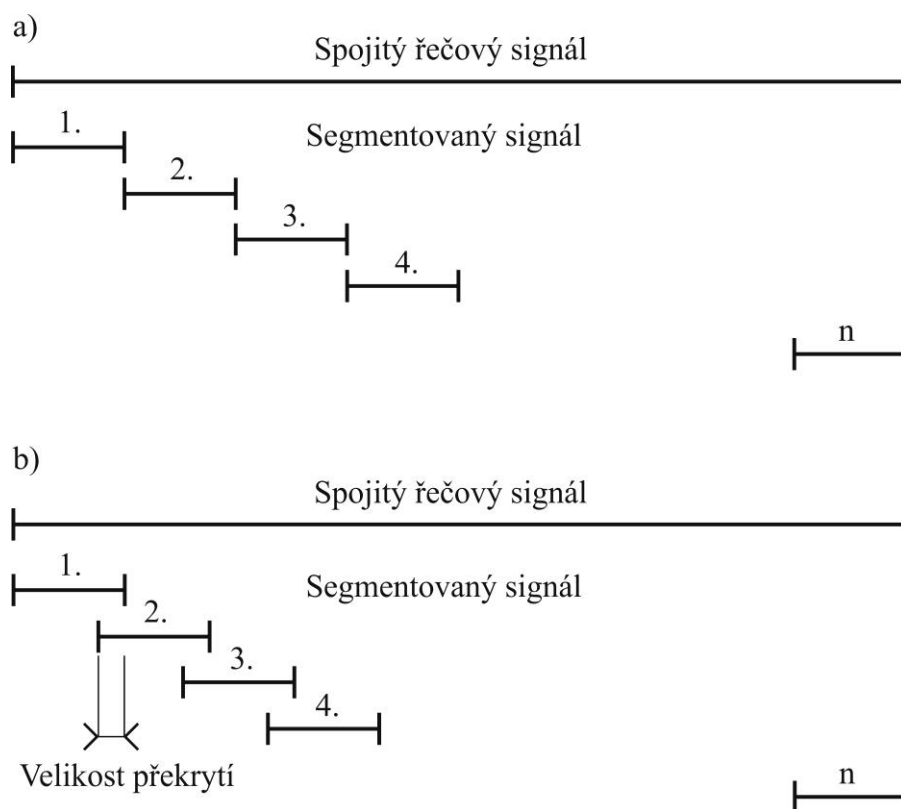
$$s'[n] = \frac{1}{N} \sum_{n=0}^{N-1} s[n], \quad (2.2)$$

kde N je počet vzorků signálu a $s[n]$ jsou jednotlivé vzorky signálu.

Segmentace znamená rozložení řečového signálu před samotným zpracováním na menší úseky tzv. rámce (frame). Jelikož řečový signál není stacionární a dá se považovat

za náhodný, je třeba rozdělit signál na již zmíněné rámce. Na těchto rámcích se už dá signál považovat za kvazy-stacionární a je tedy možné ho dále zpracovávat metodami, které uvažují deterministický průběh vstupního signálu. Délka tohoto rámce se většinou volí mezi 20-25 ms, tak aby byla dostatečně malá a bylo možné považovat průběh jednoho segmentu za stacionární a zároveň dostatečně velká k přesnému určení požadovaného parametru [4].

Většinou se používá tzv. segmentace s překryvem. To znamená, že se sousední rámce z části překrývají. Běžně se volí při segmentaci překrytí 50%. Na segmentovaný signál je použito Hammingovo okno blíže popsané na začátku kapitoly 3.



Obr. 2.1: Segmentovaný signál a) bez překrytí, b) s překrytím

2.2 Parametry založené na časové oblasti

2.2.1 Jitter

Jitter, neboli také frekvenční nestabilita se používá jako parametr pro měření kvality hlasu. Tímto parametrem lze například dobře rozpoznat stres řečníka. Bývá označovat též jako třes nebo chvění hlasivek. Lze jej definovat jako krátkodobou nepravidelnost jednotlivých délek základní periody (převrácená hodnota základní frekvence) řečového signálu. Výpočet tohoto parametru se dá definovat jako rozdíl v délce dvou sousedních period hlasivkových pulzů podělený průměrnou délkou periody. Vztah pro výpočet jitteru [15]:

$$J_g = \frac{\frac{1}{N-1} \sum_{i=1}^{N-1} |T_g[i] - T_g[i-1]|}{\frac{1}{N} \sum_{i=1}^{N-1} T_g[i]}, \quad (2.3)$$

kde N je počet period hlasivkových pulzů a $T_g[i]$ je i -tá hodnota základní periody.

Dalšími variantami jitteru jsou např. absolute (průměrná absolutní odchylka mezi následujícími periodami), RAP (relativní průměr odchylek délek period vyjádřený v procentech), PPQ5 (kvocient odchylek délek pěti sousedních period vyjádřený v procentech).

2.2.2 Shimmer

Amplitudová nestabilita řečového signálu. Stejně jako perioda u hlasového tónu popsaná v kapitole 2.2.1 se může i amplituda lišit v rámci velmi krátkého úseku řečového signálu. Jedná se tedy o kolísání amplitudy hlasivkových impulzů a bývá označováno také jako vibrace nebo kolísání pulzů. Výpočet se dá definovat jako rozdíl velikosti dvou sousedních pulzů dělený jejich průměrnou velikostí. Vztah pro výpočet shimmeru [15]:

$$S_g = \frac{\frac{1}{N-1} \sum_{i=1}^{N-1} |A_g[i] - A_g[i-1]|}{\frac{1}{N} \sum_{i=1}^{N-1} A_g[i]}, \quad (2.4)$$

kde N je počet amplitud hlasivkových pulzů a $A_g[i]$ je i -tá hodnota amplitudy.

Shimmer má stejně jako výše popsaný jitter více variant. Je to shimmer vyjádřený v decibelech, APQ3 (kvocient tří sousedních amplitudových odchylek vyjádřený

v procentech) a APQ5 (kvocient pěti sousedních amplitudových odchylek vyjádřený v procentech).

2.2.3 Short-time energy (STE)

Short-time energy, neboli krátkodobá energie je parametr vyjadřující energii řečového signálu v jednotlivých částech (segmentech) segmentované řeči. Mezi znělými a neznělými hláskami je v tomto parametru velký rozdíl. Zatímco znělé hlásky jsou energeticky vyšší, neznělé hlásky mají šumový charakter a tím pádem obsahují mnohem méně energie. Často jsou nazývány jako šumové hlásky. Toho se dá využít například k rozpoznání přechodu mezi znělým a neznělým úsekem signálu nebo při oddělování segmentů ticha od segmentů řeči např v tzv. detektoru řečové aktivity. Vztah pro výpočet krátkodobé energie jednoho segmentu řečového signálu je definován jako [14]:

$$STE = \sum_{k=0}^{N-1} s[k]^2, \quad (2.5)$$

kde $s[k]$ je vzorek diskrétního řečového signálu a N je počet vzorků segmentu.

2.2.4 Low short-time energy ratio (LSTER)

Je poměr počtu segmentů, ve kterých je krátkodobá energie (STE) nižší jako 0,5 násobek průměrné krátkodobé energie v sekundovém úseku segmentovaného signálu. Tento parametr se používá především pro diferenci mezi řečí a hudebním signálem. V řeči se totiž nachází více segmentů obsahující tiché části než v signálu hudebním. Zde by ale mohl rozlišit zdravou a patologickou řeč, jelikož se předpokládá, že u pacientů s HD bude vykazovat vyšší hodnoty než u řečníků zdravých. Vztah pro výpočet tohoto parametru je [1]:

$$LSTER = \frac{1}{2N} \sum_{n=0}^{N-1} [\text{sgn}(0,5\overline{STE} - STE(n)) + 1], \quad (2.6)$$

kde STE je krátkodobá energie segmentovaného signálu a N je počet segmentů signálu.

2.2.5 Zero-crossing rate (ZCR)

Je to parametr popisující počet průchodu signálu nulovou úrovní. U řečového

signálu nastane průchod nulou, pokud mají dva sousední vzorky různé znaménko. Tato hodnota dává také omezenou informaci o kmitočtu daného signálu. Neznělé hlásky mají vyšší kmitočty, což vede i k vyššímu počtu průchodů nulovou úrovní. Výhodou tohoto parametru je také to, že není závislý na energii signálu. Vztah pro výpočet průchodu nulovou úrovní v jednom segmentu signálu je definován jako [14]:

$$ZCR = \sum_{k=0}^{N-1} |\text{sign}(s[k]) - \text{sign}(s[k-1])| / 2, \quad (2.7)$$

$$\text{sign}(s[k]) = \begin{cases} +1 & \text{když } s[k] > 0 \text{ nebo } s[k] = 0 \wedge s[k-1] > 0, \\ -1 & \text{když } s[k] < 0 \text{ nebo } s[k] = 0 \wedge s[k-1] < 0. \end{cases} \quad (2.8)$$

kde $s[k]$ jsou vzorky segmentovaného signálu, k je pořadí vzorků v segmentu a N je počet vzorků segmentu.

2.2.6 Teager-Kaiser energy operator (TKEO)

Je nelineární operátor, který může poskytnout odhad okamžité frekvence a amplitudy z FM a AM modulace. Rozdílem oproti výpočtu krátkodobé energie (STE) je ten, že Teager-Kaiserův energetický operátor je počítán pro celý řečový signál. TKEO je tedy definován jako [13]:

$$\psi[s(n)] = s(n)^2 - s(n+1)s(n-1), \quad (2.9)$$

kde $s(n)$ je vstupní řečový signál.

2.3 Parametry založené na frekvenční oblasti

2.3.1 Spectral centroid (SC)

Spektrální těžiště neboli také centrum spektra. Udává průměrnou frekvenci celé frekvenční oblasti řečového signálu a naznačuje, kde se nachází největší část spektra. Tento parametr se počítá pomocí Fourierovy transformace. Vztah pro výpočet spektrálního těžiště [1]:

$$SC = \frac{\sum_{k=1}^N k \cdot F[k]}{\sum_{k=1}^N F[k]} \quad (2.10)$$

kde $F[k]$ je hodnota k -té složky spektra, N je celkový počet bodů spektra.

2.3.2 Spectral spread (SS)

Spektrální rozsah. Udává hodnotu o šířce pásma daného spektra okolo jeho střední hodnoty. Vzorec pro výpočet spektrálního rozsahu [11]:

$$SS = \frac{\sum_{k=1}^N (k - SC)^2 \cdot F[k]}{\sum_{k=1}^N F[k]}, \quad (2.11)$$

kde $F[k]$ je hodnota k -té složky spektra, N je celkový počet bodů spektra a SC je hodnota spektrálního těžiště.

2.3.3 Spectral flux (SF)

Spektrální tok. Je to parametr, který vyjadřuje změnu výkonového spektra signálu v čase. Je počítán jako normovaný rozdíl aktuálního a předcházejícího výkonového spektra. Vztah pro výpočet spektrálního toku [10]:

$$SF = \sum_{i=0}^{N-1} \sqrt{(S_k(i) - S_0)^2 (S_{(k-1)}(i) - S_0)^2}, \quad (2.12)$$

kde S_0 je normalizační faktor vypočtený jako průměr střední hodnoty výkonových spekter.

2.3.4 Spectral roll-off (SR)

Spektrální roll-off koeficient. Jedná se o frekvenci, pod kterou je koncentrováno p procent energie daného signálu. Většinou se jedná o 85 nebo 95%. Tento koeficient může tedy udávat mezní frekvenci mezi harmonickým signálem a šumem. Vztah pro výpočet spektrálního roll-off koeficientu [1]:

$$SR = \sum_{k=1}^{fr} F[k] = p \cdot \sum_{k=1}^N F[k], \quad (2.13)$$

kde $F[k]$ je hodnota k -té složky spektra a p je hodnota v procentech.

2.3.5 Spectral flatness (SL)

Spektrální plochost. Plochost spektra je vhodná pro rozlišení znělých a neznělých úseků. Znělé části reprezentují hodnoty blíže k nule a neznělé hodnoty blízké jedné.

Vzorec pro výpočet spektrální plochosti [2]:

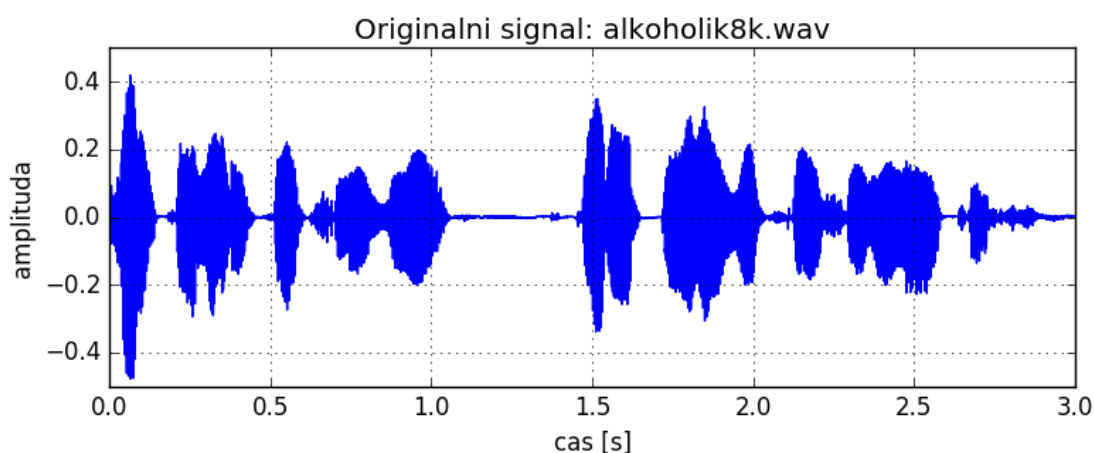
$$SL = \frac{\sqrt[N]{\sum_{k=0}^{N-1} x(k)}}{\frac{1}{N} \sum_{k=0}^{N-1} x(k)}, \quad (2.14)$$

kde $x(k)$ je hodnota k -té složky spektra.

3 PŘÍKLADY VYPOČTENÝCH PARAMETRŮ

Práce je implementována v prostředí Python. Je to interpretovaný a objektově orientovaný programovací jazyk. Dále je zde pracováno s knihovnou NumPy [9]. Tato vědecká knihovna přidává rozšíření o práci s číselnými poli, maticemi a operace s nimi. Je zde také knihovna Matplotlib [7], která přidává rozšíření v podobě vizualizace dat a vytváření grafů, které jsou vhodné pro vizuální publikaci vypočtených parametrů.

Pro výpočet parametrů byla použita zkušební nahrávka *alkoholik8k.wav*. V nahrávce je řečena věta: „Alkoholik je člověk, který pije víc než jeho lékař“. Časový průběh pro tuto nahrávku zobrazuje obrázek níže.



Obr. 3.1: Časový průběh zkušební nahrávky: *alkoholik8k.wav*.

K načtení nahrávky je použit skript *nacti_wav.py*. Tento skript převede nahrávku do číselového tvaru pro další zpracování a zjistí vzorkovací frekvenci F_{vz} . K segmentaci nahrávky je použit skript *segmentace.py*. Tímto skriptem stanovíme šířku okna v našem případě 160 vzorků v jednom segmentu. Dále překrytí segmentů, které je stanoveno na 50%.

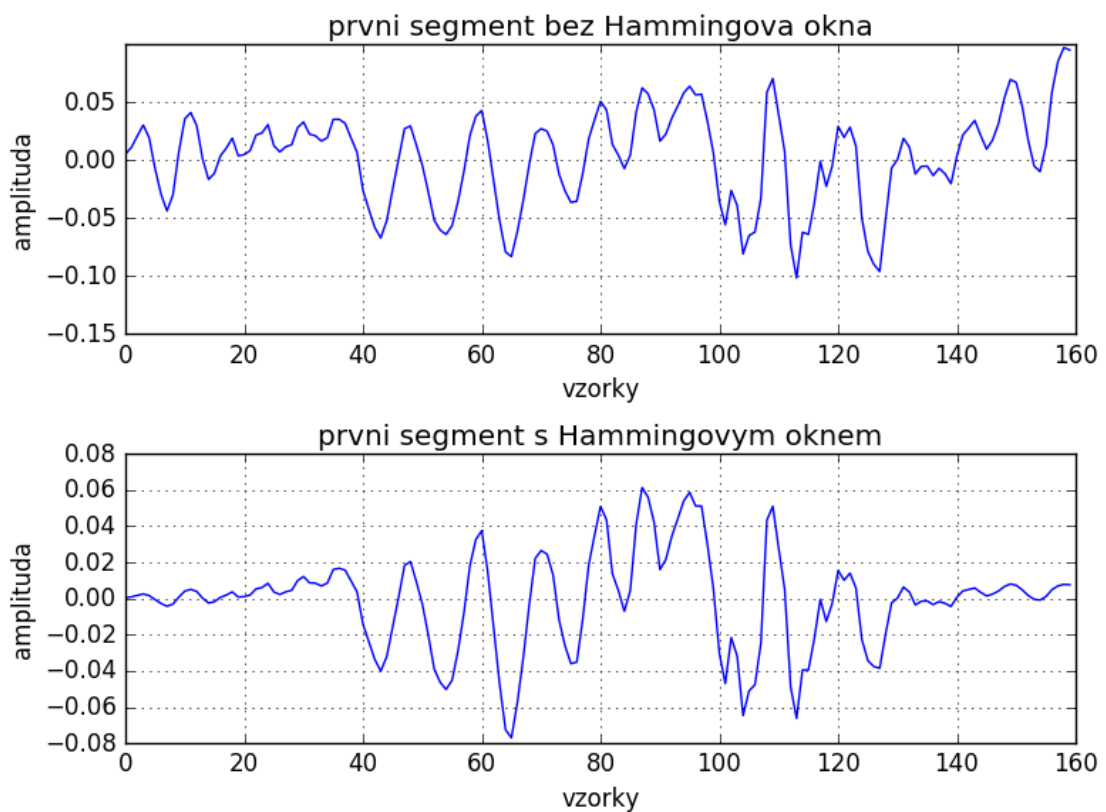
Na segmentovaný signál je dále použito Hammingovo okno, které potlačuje vzorky na okrajích segmentů, čímž zabraňuje vzniku spektrálních nehomogenit a je zvýšená stabilita některých výpočtů.

Vztah pro výpočet Hammingova okna [15]:

$$\begin{aligned} w[n] &= 0,54 - 0,46 \cos \left[n \frac{2\pi}{N} \right], & \text{pro } n = 0, 1, \dots, N - 1 \\ w[n] &= 0, & \text{pro ostatní } n \end{aligned} \quad (3.1)$$

kde N je počet vzorků segmentu.

Na obrázku Obr. 3.2 je vidět rozdíl mezi segmentem s Hammingovým oknem a bez něj. Obrázek nahoře je bez použití Hammingova okna a jsou zde vidět nepotlačené vzorky, které by mohly negativně ovlivňovat výpočty některých parametrů. Na spodním obrázku s použitím Hammingova okna už jsou tyto postraní vzorky potlačeny.



Obr. 3.2: První segment použité nahrávky.

3.1 Parametry v časové oblasti

Funkce pro výpočet Krátkodobé energie (STE):

```
1. def short_time_energy(frames):
2.     # V případě potřeby provede segmentaci
3.     frames = seg.segment(frames)
4.     # Vypocet short time energy (po segmentu)
5.     STE = []
6.     for i in xrange(0, frames.shape[1]):
7.         frame = frames[:, i].copy()
8.         energy = np.sum(frame ** 2)
9.         STE.append(energy)
10.    # Vrati vektor
11.    return np.array(STE)
```

Funkce pro výpočet poměru krátkodobé energie signálu (LSTER):

```
1. def low_short_time_energy_ratio(ramce):
2.     # V případě potřeby provede segmentaci
3.     ramce = seg.segment(ramce)
4.     # Vypocet a segmentace STE
5.     delka = ramce.shape[1]
6.     STE = seg.segment(short_time_energy(ramce))
7.     # Vypocte LSTER
8.     LSTER = np.sum(np.sign(0.5*np.mean(STE,axis=0)-STE)+1)/(2*delka)
9.     # Vrati hodnotu LSTER
10.    return LSTER
```

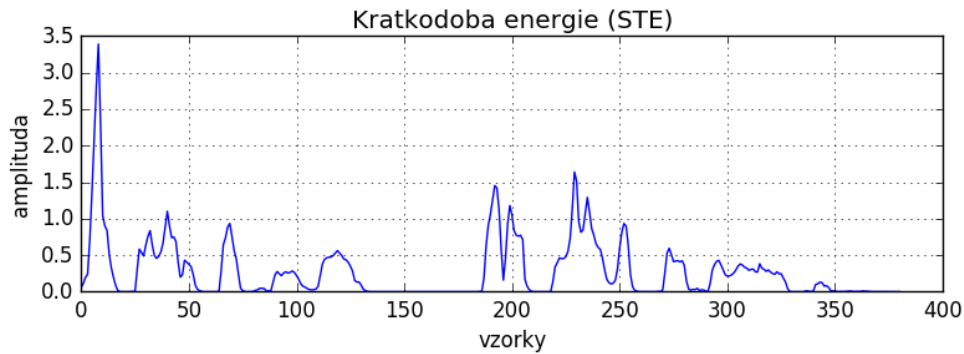
Funkce pro výpočet průchodu nulovou úrovní (ZCR):

```
1. def zero_crossing_rate(ramce):
2.     # V případě potřeby provede segmentaci
3.     ramce = seg.segment(ramce)
4.     # vypocet zero-crossing rate (po segmentu)
5.     ZCR = []
6.     for i in xrange(0, ramce.shape[1]):
7.         ramec = ramce[:, i].copy()
8.         pocet = np.sum(np.abs(np.diff(np.sign(ramec)))) / 2
9.         ZCR.append(np.float64(pocet) / np.float64(len(ramec) - 1.0))
10.    # Vrati zero-crossing rate vector
11.    return np.array(ZCR)
```

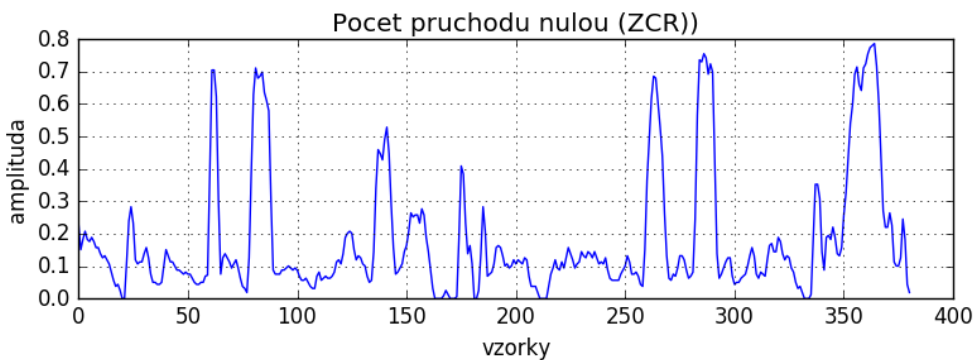
Funkce pro výpočet Teagerova-Kaiserova energetického operátoru (TKEO):

```
1. def teager_kaiser_energy_operator(signal):
2.     # Vypocet
3.     TKEO = np.zeros((len(signal) - 1, 1), dtype=np.float64);
4.     for i in xrange(1, len(signal) - 1):
5.         a = float(signal[i].copy())
6.         b = float(signal[i - 1].copy())
7.         c = float(signal[i + 1].copy())
8.         TKEO[i - 1] = a ** 2 - c * b
9.     # Vrati TKEO vektor
10.    return np.array(TKEO)
```

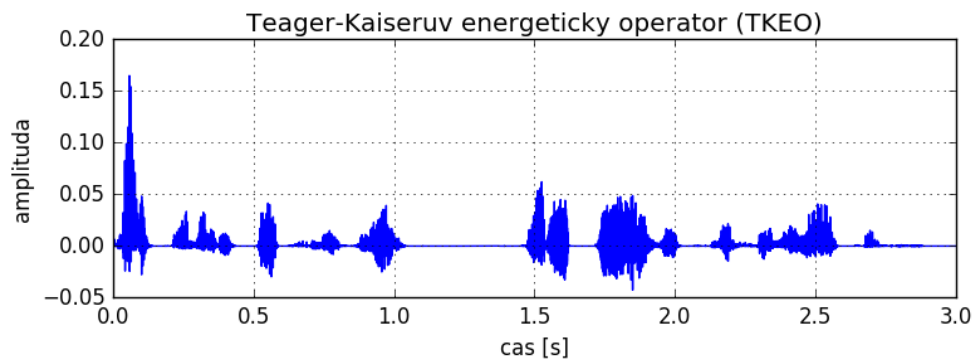
Vstupem těchto funkcí je segmentovaný signál rozdělený na rámce. Výstupy funkcí STE a ZCR jsou grafy viz obrázky Obr. 3.3 a Obr. 3.4. Výstupem funkce LSTER je koeficient udávající poměr průchodu signálu nulovou úrovní. Vstupem funkce pro výpočet TKEO není segmentovaný signál, ale celý nesegmentovaný průběh signálu. Průběh parametru ukazuje Obr. 3.5: Teager-Kaiserův energetický operátor



Obr. 3.3: Krátkodobá energie.



Obr. 3.4: Počet průchodů nulovou úrovní.



Obr. 3.5: Teager-Kaiserův energetický operátor.

Funkce pro výpočet Jitteru:

```
1. def jitter(F0):
2.     # vypocet jitter (absolute)
3.     jitt_abs = np.sum(np.abs(np.diff(F0))) / np.float64(len(F0) - 1.0)
4.     # vypocet jitter (relative)
5.     jitt_rel = jitt_abs / np.mean(F0) * 100
6.     # vypocet jitter (rap)
7.     pom_rap = 0.0
8.     for i in xrange(1, len(F0) - 1):
9.         pom_rap += abs(F0[i] - sum(F0[i - 1:i + 1].copy())) / 3.0
10.    jitt_rap=(np.sum(pom_rap)/np.float64(len(F0)-1.0))/np.mean(F0)*100
11.    # vypocet jitter (ppq5)
12.    pom_ppq = 0.0
13.    for i in xrange(2, len(F0) - 2):
14.        pom_ppq += abs(F0[i] - sum(F0[i - 2:i + 2].copy())) / 5.0
15.    jitt_ppq=(np.sum(pom_ppq)/np.float64(len(F0)-1.0))/np.mean(F0)*100
16.    # Vrati vypoctene hodnoty
17.    return (jitt_abs, jitt_rel, jitt_rap, jitt_ppq)
```

Funkce pro výpočet Shimmeru:

```
1. def shimmer(A0):
2.     # Vypocet shimmer (local)
3.     A0=abs(A0)
4.     shimm_local = np.sum(np.abs(np.diff(A0)))/np.float64(len(A0)- 1.0)
5.     # Vypocet shimmer (local, dB)
6.     pom_db = []
7.     for i in xrange(1, len(A0)):
8.         pom = float(A0[i - 1]) / float(A0[i] + 0.001)
9.         if pom == 0.0 or np.isnan(pom) or np.isinf(pom):
10.            continue
11.        else:
12.            pom_db.append(20 * log10(pom))
13.    shimm_db = np.sum(pom_db) / np.float64(len(A0) - 1.0)
14.    # Vypocet shimmer (apq3)
15.    pom_apq3 = 0.0
16.    for i in xrange(1, len(A0) - 1):
17.        pom_apq3 += abs(A0[i] - sum(A0[i - 1:i + 1].copy())) / 3.0
18.    shimm_apq3 = (np.sum(pom_apq3)\
19.        /np.float64(len(A0) - 1.0))/np.mean(A0) * 100
20.    # Vypocet shimmer (apq5)
21.    pom_apq5 = 0.0
22.    for i in xrange(2, len(A0) - 2):
23.        pom_apq5 += abs(A0[i] - sum(A0[i - 2:i + 2].copy())) / 5.0
24.    shimm_apq5 = (np.sum(pom_apq5) \
25.        /np.float64(len(A0) - 1.0))/np.mean(A0) * 100
26.    # Vrati vypoctene hodnoty
27.    return (shimm_local, shimm_db, shimm_apq3, shimm_apq5)
```

3.2 Parametry ve frekvenční oblasti

Funkce pro výpočet spektrálního těžiště (Spectral centroid):

```
1. def spectral_centroid(ramce, Fs):
2.     # V pripade potreby provede segmentaci
3.     ramce = seg.segment(ramce)
4.     # Vypocet spectral centroid (po segmentu)
5.     SC = []
6.     a = (np.arange(1, ramce.shape[0] + 1)) * (Fs / (2.0 * ramce.shape[0]))
7.     for i in xrange(0, ramce.shape[1]):
8.         ramec = ramce[:, i].copy()
9.         FFT = abs(fft(ramec)[0:ramec.shape[0]])
10.        FFT /= FFT.max()
11.        b = np.sum(a * FFT)
12.        c = np.sum(FFT) + 0.010
13.        SC.append((b / c) / (Fs / 2.0))
14.    # Vrati spectral centroid vektor
15.    return np.array(SC)
```

Funkce pro výpočet spektrálního rozsahu (Spectral spread):

```
1. def spectral_spread(ramce, Fs):
2.     # V pripade potreby provede segmentaci
3.     ramce = seg.segment(ramce)
4.     # Vypocet spectral spread (po segmentu)
5.     SS = []
6.     a = (np.arange(1, ramce.shape[0] + 1)) * (Fs / (2.0 * ramce.shape[0]))
7.     for i in xrange(0, ramce.shape[1]):
8.         ramec = ramce[:, i].copy()
9.         FFT = abs(fft(ramec)[0:ramec.shape[0]])
10.        FFT /= FFT.max()
11.        b = np.sum(a * FFT)
12.        c = np.sum(FFT) + 0.010
13.        SC = ((b / c) / (Fs / 2.0))
14.        SS.append(np.sqrt(np.sum(((a - SC) ** 2) * FFT) / c))
15.    # Vrati spectral spread vektor
16.    return np.array(SS)
```

Funkce pro výpočet spektrálního toku (Spectral flux):

```
1. def spectral_flux(ramce):
2.     # V pripade potreby provede segmentaci
3.     ramce = seg.segment(ramce)
4.     # Vypocet spectral flux (po segmentu)
5.     SF = []
6.     for i in xrange(1, ramce.shape[1]):
7.         a_frame = ramce[:, i - 1].copy()
8.         a_FFT = abs(fft(a_frame)[0:ramce.shape[0]])
9.         a_FFT /= a_FFT.max()
10.        b_frame = ramce[:, i].copy()
11.        b_FFT = abs(fft(b_frame)[0:ramce.shape[0]])
12.        b_FFT /= b_FFT.max()
13.        b_celk = b_FFT / np.sum(b_FFT)
14.        a_celk = a_FFT / np.sum(a_FFT)
15.        SF.append(np.sum((b_celk - a_celk) ** 2))
16.    # Vrati spectral flux vektor
17.    return np.array(SF)
```

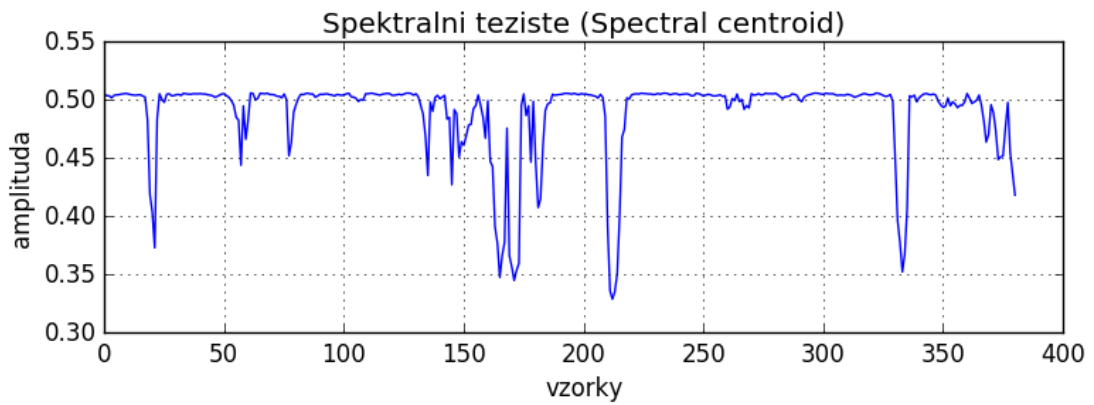
Funkce pro výpočet spektrálního roll-offu (Spectral roll-off):

```
1. def spectral_rolloff(ramce, c=0.8):
2.     # V pripade potreby provede segmentaci
3.     ramce = seg.segment(ramce)
4.     # Vypocet spectral roll-off (po segmentu)
5.     SR = []
6.     for i in xrange(0, ramce.shape[1]):
7.         ramec = ramce[:, i].copy()
8.         FFT = abs(fft(ramec)[0:ramce.shape[0]])
9.         FFT /= FFT.max()
10.        [pos, ] = np.nonzero(np.cumsum(FFT ** 2) > c * np.sum(FFT ** 2))
11.        if len(pos) > 0:
12.            SR.append(np.float64(pos[0]) / (float(len(FFT))))
13.        else:
14.            SR.append(0.0)
15.    # Vraci spectral roll-off vektor
16.    return np.array(SR)
```

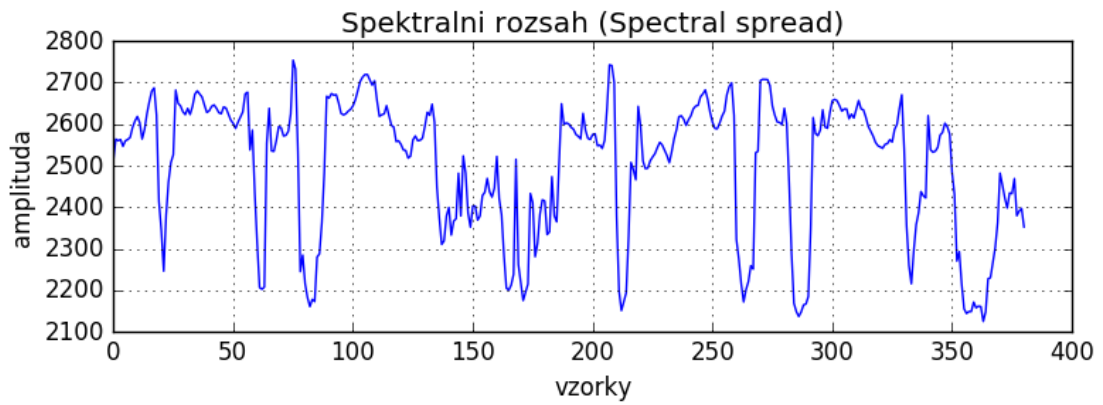
Funkce pro výpočet spektrální plochosti (Spectral flatness):

```
1. def spectral_flatness(ramce):
2.     # V pripade potreby provede segmentaci
3.     ramce = seg.segment(ramce)
4.     # Vypocet spectral flatness (po segmentu)
5.     SF = []
6.     for i in xrange(0, ramce.shape[1]):
7.         ramec = ramce[:, i].copy()
8.         FFT = abs(fft(ramec)[0:ramce.shape[0]]) ** 2
9.         FFT /= FFT.max()
10.        a = np.float64(0)
11.        b = 0
12.        for i in range(len(FFT)):
13.            vzorek = np.float64(FFT[i])
14.            a += np.float64(log(vzorek))
15.            b += vzorek
16.        a /= np.float64(len(FFT))
17.        a = exp(a)
18.        b /= float(len(FFT))
19.        SF.append(a / b)
20.    # Vraci spectral flatness vektor
21.    return np.array(SF)
```

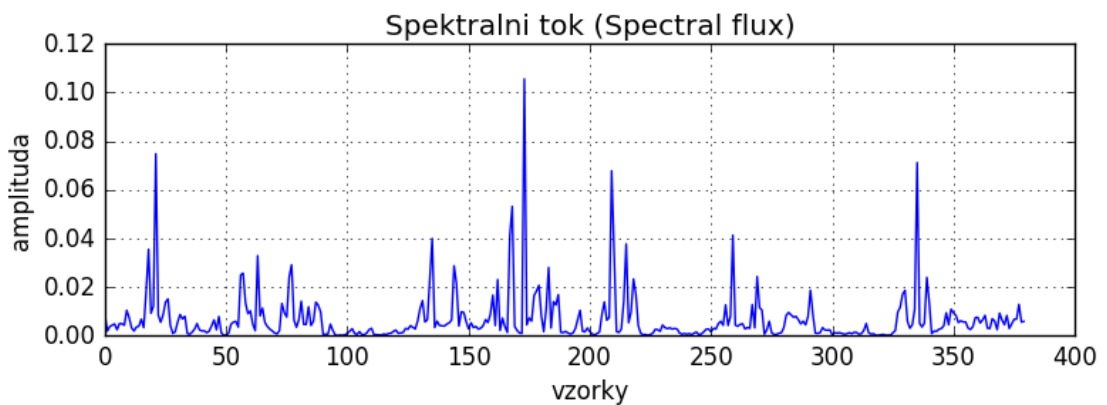
Vstupem pro všechny parametry počítané ve frekvenční oblasti je segmentovaný signál rozdělený na rámce. Výstupy jednotlivých funkcí jsou vidět na obrázcích níže. Jsou to Obr. 3.6 pro spektrální těžiště, Obr. 3.7 pro spektrální rozsah, Obr. 3.8 pro spektrální tok, Obr. 3.9 pro spektrální roll-off a poslední spektrální plochost ukazuje Obr. 3.10.



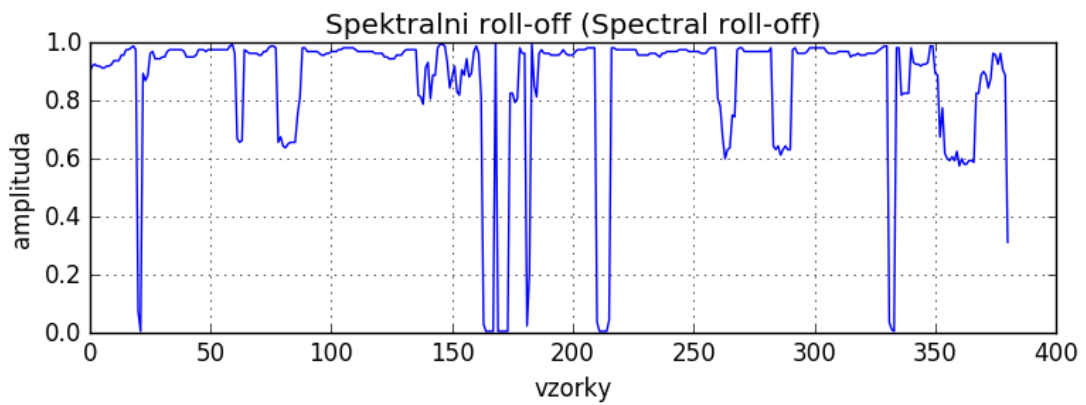
Obr. 3.6: Spektrální těžiště.



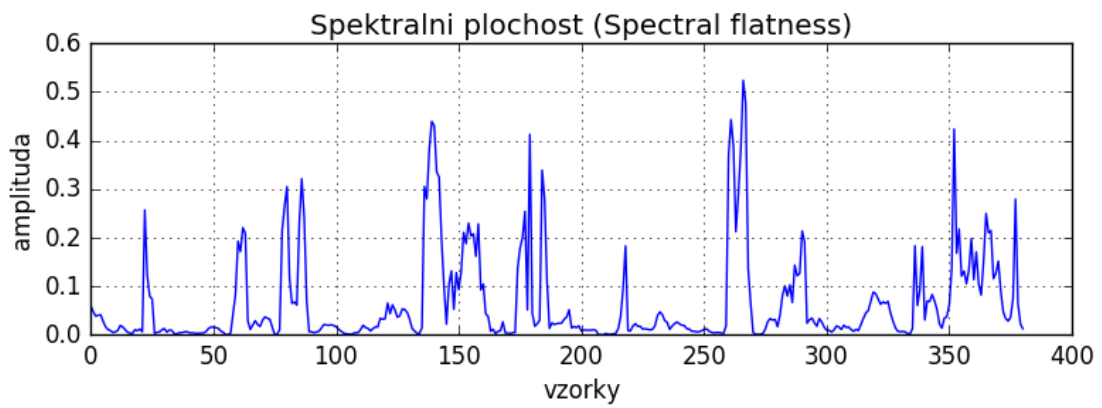
Obr. 3.7: Spektrální rozsah.



Obr. 3.8: Spektrální tok.



Obr. 3.9: Spektrální roll-off.



Obr. 3.10: Spektrální plochost.

4 PRAKTICKÁ ČÁST

V této kapitole bude popsáno samotné vytvoření webové aplikace ve frameworku django.

4.1 Django

Je to webový rámec, neboli framework, který je napsán v jazyce Python. Tento open source webový framework je sada komponent sloužících k vývoji webových stránek. Tento Framework je postaven na MCV (Model-View-Controller) architektuře. Viz kapitola 4.1.2 [5].s

4.1.1 Důležité skripty

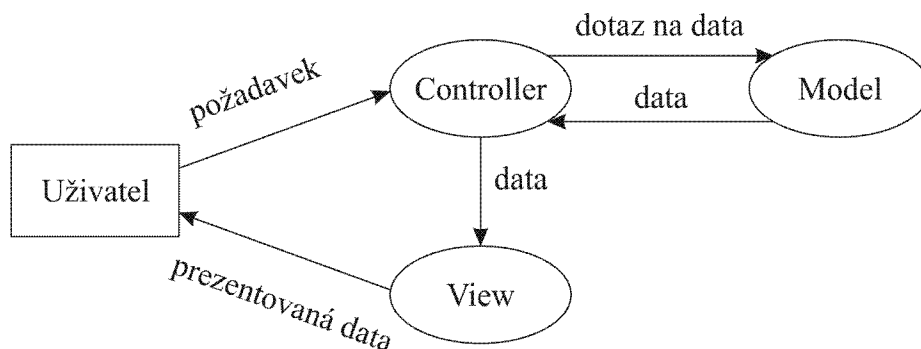
Settings.py – V tomto skriptu se nachází nastavení samotného jádra systému. Nastavuje se zde například název projektu, časové pásmo, seznam použitých aplikací, atd.

Urls.py - Zde se nachází rozcestník webových adres. Jsou zde namapovány url adresy, které odkazují na konkrétní funkce ve skriptu *views.py*.

Views.py – Jsou zde obslužné funkce volané z již zmiňovaného skriptu *urls.py*. Ve funkcích jsou prováděny různé akce. Výsledky těchto funkcí jsou posílány do HTML šablony, která je následně vrácena uživateli jako výstupní HTML stránka.

4.1.2 MVC

Tato architektura se skládá ze tří částí, které mezi sebou mají určité vazby a jsou na sobě nezávislé. První částí je Model. Model reprezentuje informace, s kterými aplikace pracuje. Další částí je View neboli pohled. Tato část je zodpovědná za převod dat reprezentovaná modelem do podoby vhodné pro prezentaci výstupních dat pro uživatele. V této práci je pro tyto účely použita html stránka. Poslední částí je Controller neboli řadič. Ten se stará o propojení mezi prvními dvěma částmi. Jeho hlavní úlohou je přijímat požadavky od uživatele a na základě nich zajišťovat změny v modelu [3]. Architektura MVC viz Obr. 4.1.



Obr. 4.1 Architektura MVC (Model-View-Controller).

4.2 HTML

Je to jazyk pro tvorbu hypertextových dokumentů. Tento jazyk anglicky HyperText Markup Language je jeden ze základních formátů pro tvorbu webových stránek. Zdrojový kód se zapisuje a je ukládán do obyčejného textového souboru. HTML kód je poté interpretován HTML prohlížečem. Tento jazyk je tvořen množinou značek tzv. tagů a jejich vlastností.

Zde je ukázka jednoduchého HTML kódu:

```

1. <html>
2. <head>
3. <title>Titulek stránky</title>
4. </head>
5. <body>
6. <h1>Nadpis stránky</h1>
7. <p>Odtavec</p>
8. </body>
9. </html>
  
```

Skládá se z kořenového elementu (<html> a </html>) tímto je reprezentován celý dokument. Poté je zde (<head> a </head>), což je hlavička dokumentu. Mohou se v ní nacházet metadata vztahující se k celému dokumentu. V tomto případě je zde obsažen titulek stránky (<title> a </title>). Posledním důležitým prvkem je (<body> a </body>). Tento prvek je tělo HTML stránky a obsahuje vlastní obsah dokumentu. Zde se v těle

dokumentu na ukázkou nachází nadpis, který reprezentují tagy (<h1> a </h1>) a (<p> a </p>), což je tag pro odstavec.

4.3 CSS

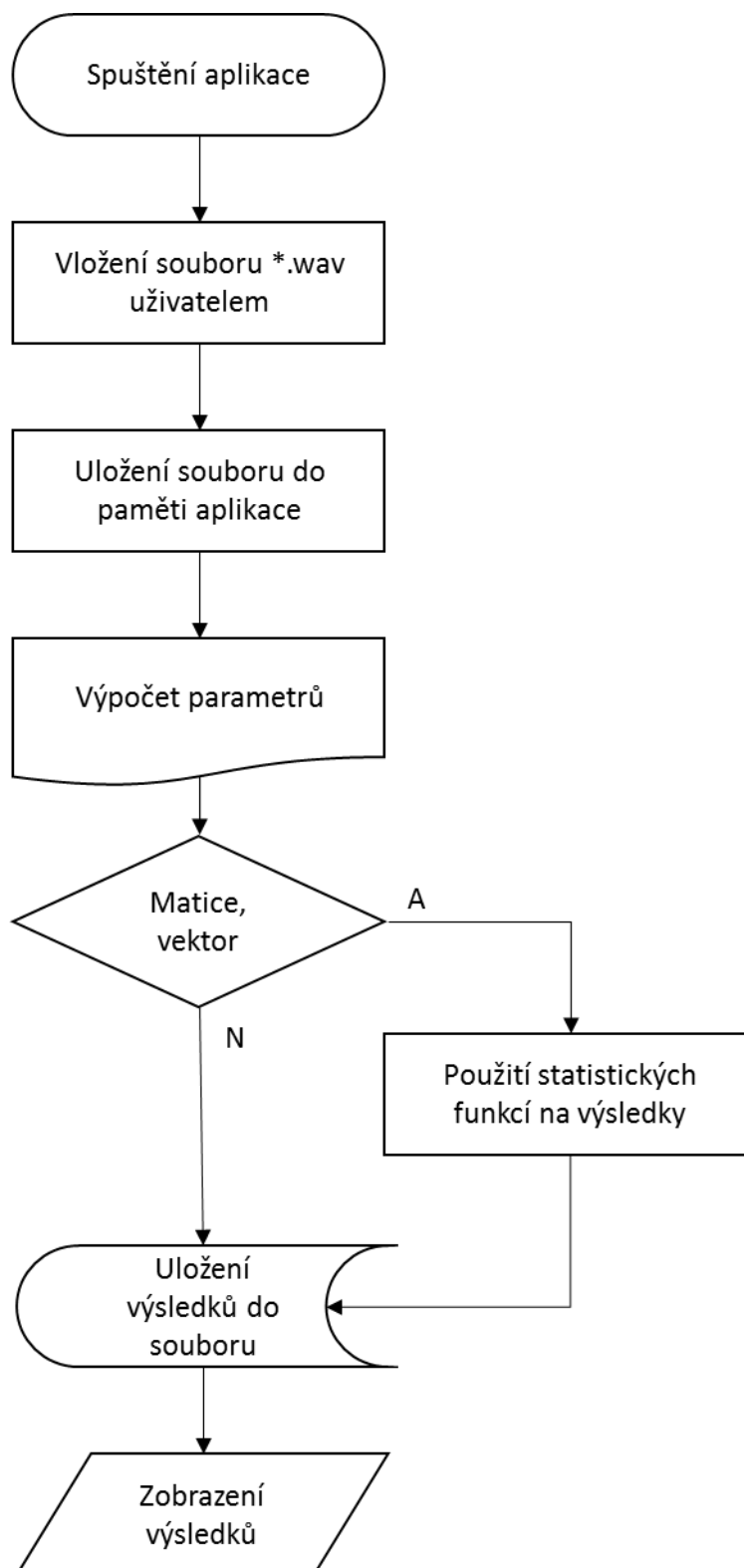
Kaskádové styly, neboli Cascading Style Sheets, jsou jazykem vytvořeným pro popis vzhledu webových stránek. Hlavní výhodou kaskádových stylů je striktní oddělení vzhledu webových stránek od jejich obsahu. Tento styl je většinou samostatný soubor, který se k HTML dokumentu připojuje pomocí tagu zapsaného do hlavičky.

Ukázka CSS stylu:

```
1. h1 { font-size: 20px;  
2.     color: blue;  
3.     text-align: center;  
4. }
```

Všechny nadpisy v dokumentu s tagem (<h1> a </h1>) pak budou mít parametry, které udává tento CSS styl.

4.4 Blokové schéma



4.5 Popis systému

Prvním krokem je spuštění aplikace v jakémkoliv webovém prohlížeči. Uživatel zadá adresu, na které se tato aplikace nachází. Pro účely této práce je používán virtuální server dostupný pouze v konkrétním PC, na kterém je spuštěn. Více o spuštění viz kapitola 4.7.

Dalším krokem je vložení zvukové nahrávky uživatelem. Tato nahrávka musí být vložena ve formátu *.wav. Poté je zvukový soubor uložen do paměti aplikace pro jeho další využití.

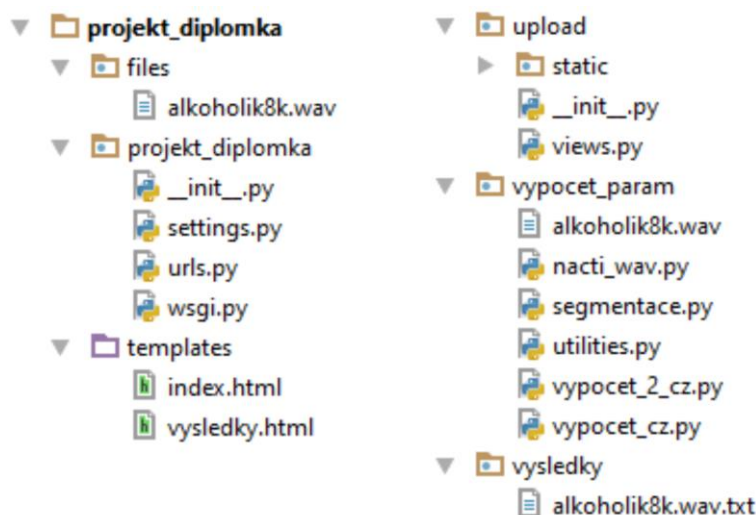
Hlavním krokem aplikace je samotný výpočet parametrů. Zvukový soubor je načten a segmentován na menší úseky. Z tohoto segmentovaného signálu jsou jednotlivé parametry vypočteny.

Na parametry jsou poté, je-li to potřeba, uplatněny statistické funkce. Konkrétně je to minimum, maximum, střední hodnota a odchylka.

Posledním krokem je uložení výsledku do textového souboru, pro případné další využití a zároveň zobrazení výsledků.

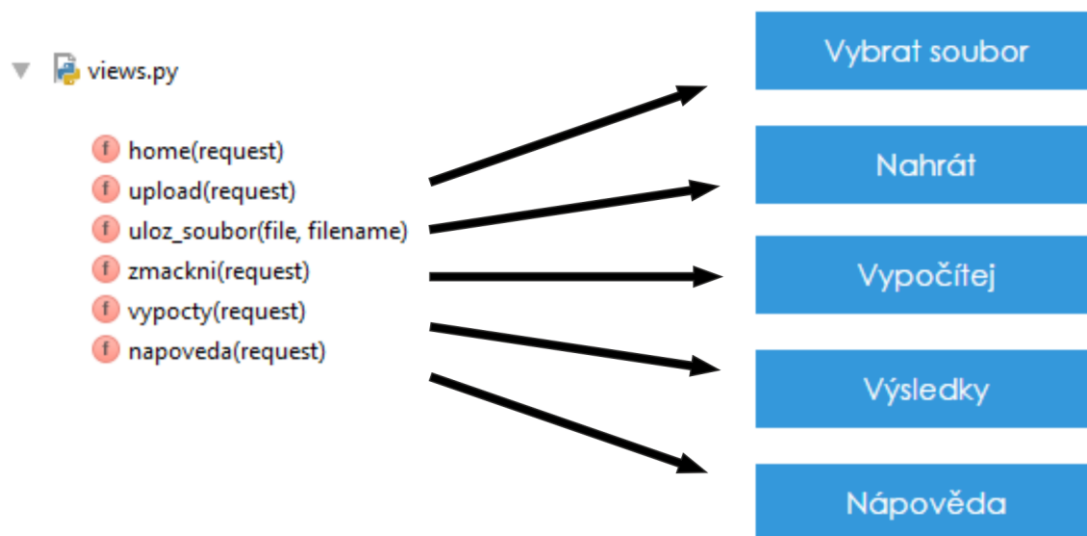
4.6 Aplikace

Aplikace se skládá z několika částí. Celou hierarchii ukazuje Obr. 4.2. Hlavní částí aplikace je soubor *views.py*. Do tohoto souboru se zapisují veškeré funkce potřebné k samotnému chodu a jsou odsud zároveň generovány dynamické odpovědi do html šablon. K tomuto plnění šablon je použit šablonovací jazyk Jinja. Tento jazyk vychází ze syntaxe django. Je velmi oblíbený pro své jednoduché propojení s html kódem.



Obr. 4.2 Hierarchie programu.

V souboru *views.py* jsou naprogramované následující funkce. Pro lepší představu je zde Obr. 4.3 ukazující i jejich propojení s tlačítky v menu uživatelského rozhraní:



Obr. 4.3 Propojení jednotlivých tlačítek aplikace s funkcemi.

home – tato funkce načte html stránku při prvním spuštění webové aplikace uživatelem.

upload – společně s funkcí *uloz_soubor* se stará o nahrání souboru uživatelem a následným uložením do složky *files*. Odtud se ze samotným nahraným souborem ve formátu *.wav pracuje.

zmackni – zde se nachází samotný výpočet parametrů. Funkce načte soubor z již zmiňované složky *files*. Poté ze složky *vypocet_param* ze skriptu *nacti_wav.py* zavolá funkci *cti_wav*, která načte zvukový soubor. Další funkcí je funkce *segment*, která zvuk rozdělí na požadované úseky pro další zpracování. Oba tyto skripty jsou posány v kapitole 2.1. Následně jsou vypočítány všechny parametry již popsané v kapitolách 2.2 a 2.3.

vypocty - touto funkcí se do šablony *vysledky.html* zobrazí vypočtené parametry vybraného souboru.

napoveda – tato funkce pouze zobrazí nápovědu o ovládání webové aplikace.

4.7 Spuštění aplikace

Pro vývoj webové aplikace je použit program PyCharm ve verzi 2016.2.3. Tento program obsahuje kompletní sadu pro vývoj v programovacím jazyce Python. Poskytuje funkce pro profesionální vývoj webových aplikací ve frameworku Django. Pro vývoj těchto webových aplikací se dá použít i jednoduchý příkazový řádek. Tato aplikace však zpříjemňuje vývojové prostředí. Zde jsou popsány příkazy nutné pro založení a spuštění aplikace.

Vytvoření projektu:

```
django-admin.py startproject <nazev projektu>
```

tento příkaz vytvoří základní skripty pro chod aplikace. Jsou to skripty popsané v kapitole 4.1.1.

Start aplikace:

```
python manage.py runserver
```

tímto příkazem se spustí virtuální server na kterém webová aplikace běží. Po tomto spuštění serveru už jen stačí v jakémkoliv webovém prohlížeči zadat adresu:

```
http://127.0.0.1:8000/
```

po zadání této adresy zobrazí vytvořená webová aplikace.

4.8 Ukázka aplikace

V této kapitole bude popsán reálný postup vložení a výpočtu parametrů z ukázkové nahrávky. Ukázkovou nahrávkou je soubor *alkoholik8k.wav*.

Uvodní obrazovku při zadání adresy ve webovém prohlížeči popsané v kapitole 4.4 ukazuje Obr. 4.4. Na této obrazovce uživatel stiskne tlačítko Vybrat soubor a ze svého zařízení vybere soubor, který chce analyzovat. V tomto případě je to soubor *alkoholik.wav*. Dále se pokračuje tlačítkem Nahrát.



Obr. 4.4 Úvodní stránka aplikace.

Po stisknutí tlačítka Nahrát se soubor uloží do složky *files* v paměti aplikace pro následné zpracování. Umístění složky znázorňuje Obr. 4.2. Uživateli se poté zobrazí stránka, na které je popsán další postup. Tuto stránku ukazuje Obr. 4.5.

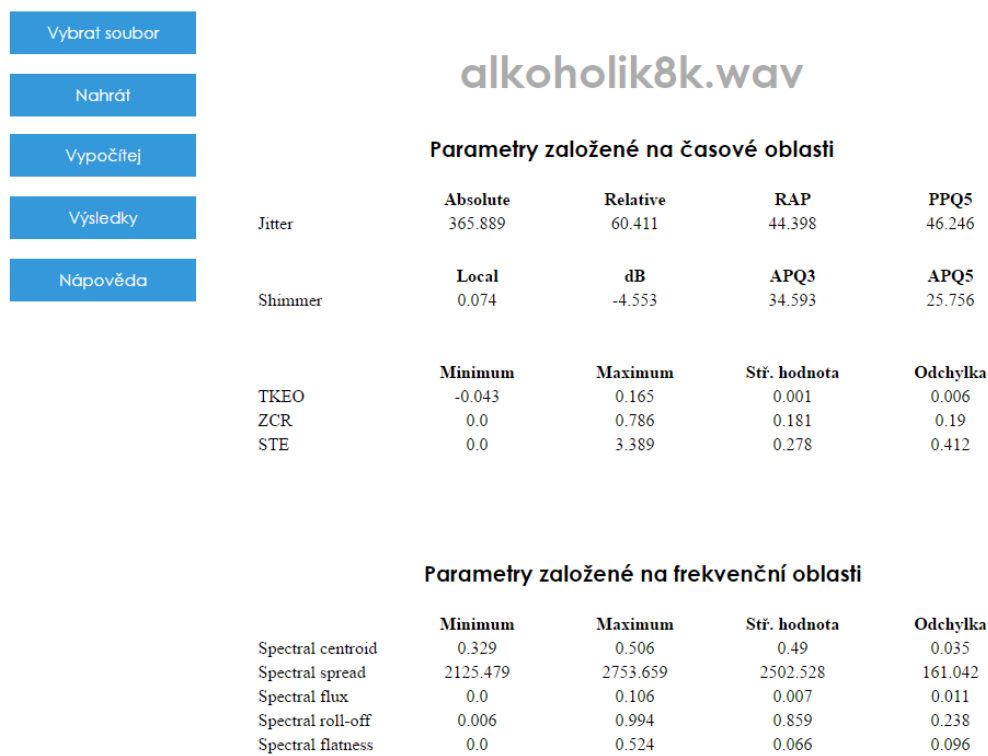


Obr. 4.5 Druhá stránka aplikace po stisknutí tlačítka Nahrát.

Stiskem tlačítka Vypočítej se provedou veškeré potřebné funkce pro samotný výpočet parametrů jak je popsáno v kapitole 4.6. Je-li potřeba tak jsou následně na výsledky uplatněny statistické funkce a to konkrétně: Minimum, Maximum, Střední hodnota a Odchylka. Dalším krokem je stisknutí tlačítka Výsledky.



Obr. 4.6 Třetí stránka s po stisknutí tlačítka Vypočítej.



Obr. 4.7 Stránka s výsledky.

Stránka s výsledky zobrazuje konečné výpočty aplikace. Jsou zde dvě tabulky. První zobrazuje parametry založené na časové frekvenci. Jsou to parametry: jitter, shimmer, TKEO, ZCR a STE. Více viz kapitola 2.2.

Druhá tabulka zobrazuje parametry založené na frekvenční oblasti. Těmito parametry jsou spectral centroid, spectral spread, spectral flux, spectral roll-off a spectral flatness. Popsány v kapitole 2.3.

5 ZÁVĚR

Cílem diplomové práce bylo vytvořit webovou aplikaci pro objektivní analýzu hypokinetické dysartrie. Pro tyto účely byla tato hypokinetická dysartrie v rámci diplomové práce nastudována a popsána. Tato porucha hlasového ústrojí se ve velké míře projevuje u pacientů s Parkinsonovou nemocí. Dále zde byly popsány tři základní složky projevující se u pacientů s hypokinetickou dysartrií, které při nesprávném fungování negativně ovlivňují tvorbu řeči. Těmito složkami jsou oblasti fonace, artikulace a prozodie. Dále zde byla popsáno předzpracování řečového signálu a jeho segmentace pro další použití v této práci. V dalším kroku byly popsány parametry řeči, které by mohly dobře rozlišovat zdravou řeč a řeč napadenou hypokinetickou dysartrií. Jsou zde popsány parametry hodnotící řeč v časové oblasti a parametry hodnotící řeč ve frekvenční oblasti. V další kapitole byly jednotlivé parametry z obou skupin hodnotící řeč zobrazeny v podobě kódů napsaných v moderním a stále se rozšiřujícím programovacím jazyce Python. Pro lepší představivost toho, co je výsledkem těchto kódů, zde byly zobrazeny i průběhy těchto parametrů v grafické podobě.

Další částí práce bylo nastudování a následná realizace webové aplikace na analýzu hlasového signálu. Pro tuto realizaci byl použit open source webový aplikační framework Django napsaný v jazyce Python. Tento jazyk se volně drží architektury MVC (Model-View-Controller), která je blíže popsána v kapitole 4.1.2. Dále zde byl použit moderní šablonovací systém Jinja2. Tento jazyk umožňuje jednoduché, dynamické plnění html šablon. Další nezbytnou částí této práce bylo také nastudování HTML jazyka. Tento značkovací jazyk se používá pro tvorbu webových stránek. V této práci je tímto jazykem napsána html šablona, která je následně pomocí jazyku Jinja2 dynamicky plněna požadovanými datami. Pro úpravu vzhledu webové stránky jsou použity kaskádové sytly, neboli anglicky Cascading Style Sheets (CSS). Hlavní výhodou těchto sytlů je oddělení vzhledu dokumentu od jeho struktury a obsahu. HTML a CSS jsou blíže popsány v kapitole 4.2 a v kapitole 4.3.

Tato práce by mohla být v budoucnu rozšířena o několik zajímavých vylepšení. V první řadě by to mohlo být vykreslení různých grafů k jednotlivým zvukovým nahrávkám. Například grafy formantů., což jsou oblasti s vyšší koncentrací akustické energie vznikající v důsledku průchodu hlasivkového tónu zejména dutinou hrdelní

a ústní a následnou rezonací v těchto dutinách. Dále například vykreslení parametru popisujícího kmitání hlasivek a mnohé další parametry zobrazené grafickou podobou. Dalším zajímavým rozšířením by bylo diagnostikovat případnou patologii řeči na základě porovnání vložené hlasové nahrávky se vzorky zdravými. Toto rozšíření by však požadovalo nahrání velkého množství jak vzorků zdravých řečníků tak i vzorky pacientů napadených hypokinetickou dysartrií, popřípadě i jiných patologií, které je možné nějakým způsobem analyzovat prostřednictvím hlasového signálu.

POUŽITÁ LITERATURA

- [1] ALEXANDRE-CORTIZO, E., M. ROSA-ZURERA a F. LOPEZ-FERRERAS. Application of Fisher Linear Discriminant Analysis to Speech/Music Classification. In: *EUROCON 2005 - The International Conference on "Computer as a Tool"* [online]. IEEE, 2005, s. 1666-1669 [cit. 2016-11-28]. DOI: 10.1109/EURCON.2005.1630291. ISBN 142440049X.
- [2] BECKER, J.M. a CH. ROHLFING. A Segmental Spectral Flatness Measure for Harmonic-Percussive Discrimination. *Institute of Communications Engineering, RWTH Aachen University, 52056 Aachen, Germany*. 2013.
- [3] ČÁPKA, David. *MVC architektura* [online]. [cit. 2017-04-10]. Dostupné z: <https://www.itnetwork.cz/navrhove-vzory/mvc-architektura-navrhovy-vzor/>
- [4] ČERNOCKÝ, J. *Zpracování řečových signálů*. Ústav počítačové grafiky a multimédií, FIT, VUT Brno, 2006.
- [5] *Django Česká republika* [online]. Dostupné z: <http://www.djangoproject.cz/>
- [6] ELIASOVA, I., J. MEKYSKA, M. KOSTALOVA, R. MARECEK, Z. SMEKAL a I. REKTOROVA. Acoustic evaluation of short-term effects of repetitive transcranial magnetic stimulation on motor aspects of speech in Parkinson's disease. *Journal of Neural Transmission* [online]. 2013, **120**(4), 597-605 [cit. 2016-11-21]. DOI: 10.1007/s00702-012-0953-1. ISSN 03009564.
- [7] *Matplotlib* [online]. Dostupné z: <http://matplotlib.org/>
- [8] MEKYSKA, J., I. REKTOROVA a Z. SMEKAL. Selection of optimal parameters for automatic analysis of speech disorders in Parkinson's disease. In: *2011 34th International Conference on Telecommunications and Signal Processing (TSP)* [online]. IEEE, 2011, s. 408-412 [cit. 2016-11-21]. DOI: 10.1109/TSP.2011.6043700. ISBN 978-1-4577-1410-8.
- [9] *NumPy* [online]. Dostupné z: <http://www.numpy.org/>

- [10] Onset detection and spectral flux [online]. Dostupné z URL: <http://www.badlogicgames.com/wordpress/?p=161>
- [11] PEETERS, G. *A Large Set of Audio Features for Sound Description(similarity and classification) in the CUIDADO project* [online]. 2004. Dostupné z: http://recherche.ircam.fr/equipes/analyse-synthese/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf
- [12] PSUTKA, Josef, et al. *Mluvíme s počítačem česky*. Praha: ACADEMIA, 2006. 752 s. ISBN 80-200-1309-1.
- [13] QUATIERI, T. F. *Discrete-time speech signal processing: principles and practice*. Upper Saddle River, NJ: Prentice Hall, c2002. ISBN 013242942X.
- [14] SIGMUND, Milan. *Rozpoznávání řečových signálů: přednášky*. 1. vyd. Brno: VUT FEKT, ústav radioelektroniky, 2007, 122 s. ISBN 978-80-214-3526-1.
- [15] SMÉKAL, Z.: *Číslíkové zpracování řeči (MZPR)*. Elektronická skripta pro magisterská studia, VUT Brno, 2010.
- [16] ZAMIŠKOVÁ, Gabriela, Pavel RESSNER, Jana DLOUHÁ a Dana ŠIGUTOVÁ. Poruchy řeči u Parkinsonovy nemoci: Speech disorders in Parkinson's disease. *Neurologia pre prax*. Bratislava: SOLEN, 2010, **11**(2), 112. ISSN 13359592.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

APQ3	kvocient tří sousedních amplitudových odchylek
APQ5	kvocient pěti sousedních amplitudových odchylek
FIR	filtr typu horní propust
HD	hypokinetická dysartrie
LSTER	poměr nízké krátkodobé energie
PN	Parkinsonova nemoc
PPQ5	kvocient odchylek délek pěti sousedních period
RAP	relativní průměr odchylek délek period
SC	spectral centroid (spektrální těžiště)
SF	spectral flux (spektrální tok)
SL	spectral flatness (spektrální plochost)
SR	spectral roll-off (spektrální roll-off)
SS	spectral spread (spektrální rozsah)
STE	krátkodobá energie
TKEO	Teagero-Kaiserův energetický operátor
ZCR	počet průchodů signálu nulovou úrovní
a_1	koefficient preemfázového filtru
$A_g[i]$	i -tá hodnota amplitudy
$F[k]$	hodnota k -té složky spektra
$H(z)$	přenosová funkce preemfázového filtru
J_g	jitter
S_g	shimmer

$s[n]$	řečový signál
$s'[n]$	řečový signál po ustředění
$s[k]$	vzorek diskrétního řečového signálu
$T_g[i]$	i -tá hodnota základní periody
$w[n]$	váhovací posloupnost

SEZNAM PŘÍLOH

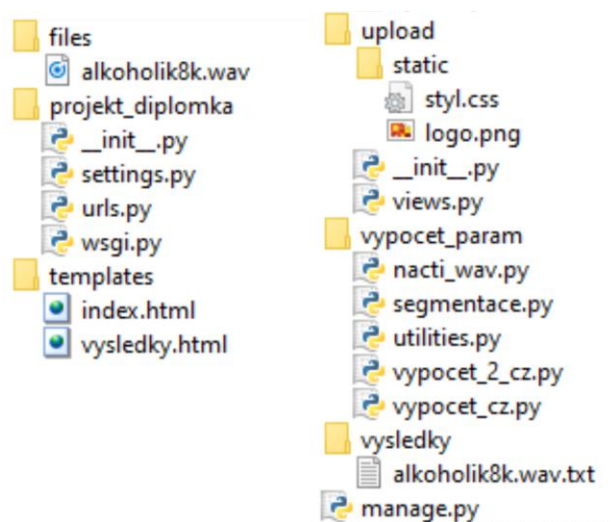
Příloha 1. Obsah přiloženého DVD

Příloha 2. Ukázka programu PyCharm

Příloha 3. Ukázka výstupního souboru

A OBSAH PŘILOŽENÉHO DVD

- projekt_diplomka - tato složka obsahuje kompletní aplikaci



- diplomová práce.pdf

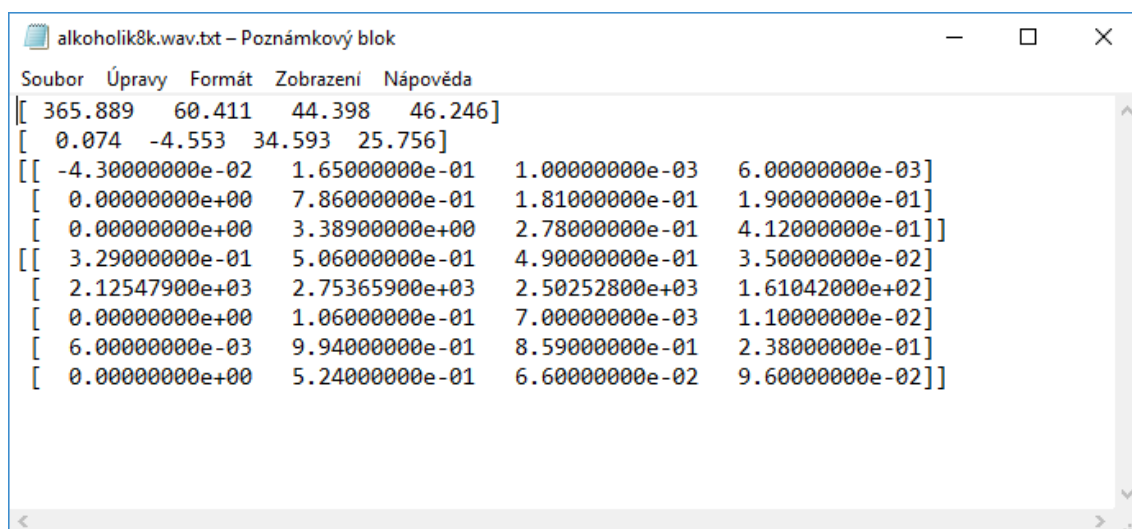
B UKÁZKA PROGRAMU PYCHARM



Obr. B.1 Ukázka programovacího prostředí PyCharm.

C UKÁZKA VÝSTUPNÍHO SOUBORU

Obr. C.1 ukazuje textový soubor, který je vytvářen společně s výstupem do webové aplikace. Jednotlivé řádky představují konkrétní parametry. První řádek je Jitter, dále pak Shimmer, TKEO, ZCR, STE, Spectral centroid, Spectral spread, Spectral flux, Spectral roll-off, a Spectral flatness.



```
alkoholik8k.wav.txt - Poznámkový blok
Soubor Úpravy Formát Zobrazení Nápověda
[[ 365.889 60.411 44.398 46.246]
 [ 0.074 -4.553 34.593 25.756]
 [[ -4.30000000e-02 1.65000000e-01 1.00000000e-03 6.00000000e-03]
 [ 0.00000000e+00 7.86000000e-01 1.81000000e-01 1.90000000e-01]
 [ 0.00000000e+00 3.38900000e+00 2.78000000e-01 4.12000000e-01]]
 [[ 3.29000000e-01 5.06000000e-01 4.90000000e-01 3.50000000e-02]
 [ 2.12547900e+03 2.75365900e+03 2.50252800e+03 1.61042000e+02]
 [ 0.00000000e+00 1.06000000e-01 7.00000000e-03 1.10000000e-02]
 [ 6.00000000e-03 9.94000000e-01 8.59000000e-01 2.38000000e-01]
 [ 0.00000000e+00 5.24000000e-01 6.60000000e-02 9.60000000e-02]]
```

Obr. C.1 Ukázka výstupního textového souboru.