Česká zemědělská univerzita v Praze

Technická fakulta

Analýza, syntéza a rozpoznání řeči v aplikacích

Diplomová práce

Vedoucí diplomové práce: Ing. Miloslav Linda, Ph.D.

Autor práce: Bc. Jan Holý

PRAHA 2013

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Katedra elektrotechniky a automatizace

Technická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Holý Jan

Informační a řídící technika v agropotravinářském komplexu

Název práce

Analýza, syntéza a rozpoznání řeči v aplikacích.

Anglický název

Speech recognition, analysis and synthesis in application.

Cíle práce

Cílem práce je analýza možností řešení a aplikovatelnosti rozpoznávání řeči. Možnosti řešení spočívají v teoretickém rozboru problematiky rozpoznávání řeči v oblasti analýzy a syntézy signálů, určení lineární predikční systézy řeči. Užití nástrojů sloužících k potlačení různých rušivých vlivů, užití parametrických modelů užívaných při zpracování řeči, užití klasifikátorů slov v reálném čase. Možnosti aplikovatelnosti se zabývají řešením možného užití vybraných metod v průmyslové praxi a při ovládání aplikací řidicích systémů.

Metodika

Provedení teoretického rozboru. Rozbor vybraných metod. Výběr aplikací k řešení problémů rozpoznávání řeči. Možnosti ovládání hlasovým vstupem. Aplikace na vybraném problému řízení.

Osnova práce

1. Úvod.
2. Cíle práce.
3. Zvolené metody zpracování.
4. Přehled o současném stavu problematiky a odborné literatury.
5. Výsledky práce s uvedením nových poznatků.
6. Závěry a doporučení pro využití poznatků v praxi.
7. Seznam odborné literatury.

Rozsah textové části
60 stran, bez příloh

Klíčová slova
Analýza řečí, systéza řeči, rozpoznávání, programování, Fourierova transformace, filtrace, implementace, restaurace signálů.

Doporučené zdroje informací

Kolektiv autorů. Analýza, syntéza a rozpoznávání řeči. 1. vyd. Praha : ČSVTS - FEL - ČVUT, 1985. 156 s.

UHLÍŘ, Jan. SOVKA, Pavel. Číslicové zpracování signálů. Vyd. 2. přeprac. Praha : ČVUT, 2002. 327 s. ISBN 80-01-02613-2.

TUČKOVÁ, Jana. Vybrané aplikace umělých neuronových sítí při zpracování signálů. Praha : ČVUT, 2009. 224 s. ISBN 978-80-01-04229-8.

JAN, Jiří. Číslicová filtrace, analýza a restaurace signálů. 2. upr. a rozš. vyd. Brno: VUTIUM, 2002. 427 s. ISBN 80-214-1558-4.

DAVÍDEK, Vratislav. Implementace algoritmů číslicového zpracování signálů v reálném čase. Vyd. 1. Praha: ČVUT, 2004. 171 s. ISBN 80-01-03114-4.

POULTON, A. S.. Microcomputer Speech Synthesis and Recognition. 1st ed. Wilnslow: Sigma Technical Press, 1983. 194 s.

RABINER, Lawrence, JUANG, Biing-Hwang. Fundamentals of Speech Recognition. 1st ed. Englewood Cliffs: Prentice Hall, 1993. 507 s. ISBN 0-13-015157-2.

LEE, Chin-Hui., SOONG, Frank K, PALIWAL, Kuldip K.. Automatic Speech and Speaker Recognition: advanced topics. 1st ed. Boston: Kluwer, 1996. 517 s. ISBN 0-7923-9706-1.

Vedoucí práce
Linda Miloslav, Ing.

Termín zadání
listopad 2011

Termín odevzdání
duben 2013

prof. Ing. Volf Jaromír, DrSc.
Vedoucí katedry

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE
-3-
TECHNICKÁ FAKULTA

prof. Ing. Vladimír Jurča, CSc.
Děkan fakulty

V Praze dne 6.2.2012

Čestné prohlášení

Prohlašuji, že jsem svoji diplomovou práci na téma „Analýza, syntéza a rozpoznání řeči v aplikacích“ vypracoval samostatně a že jsem uvedl všechny literární zdroje a prameny, ze kterých jsem čerpal.

V Praze dne ……

Bc. Jan Holý

Poděkování

Na tomto místě bych chtěl zejména poděkovat svému vedoucímu diplomové práce Ing. Miloslavu Lindovi, Ph.D. za jeho podporu a mnoho cenných rad při vedení diplomové práce.

Dále bych rád poděkoval svým rodičům a přátelům za podporu během studia.

Abstrakt: Cílem diplomové práce je zhodnocení systémů rozpoznávající řečový signál a vlastní realizace systému reagujícího na hlasové povely. Úvodní část je věnována popisu základních metod analýzy a syntézy řečových signálů. V této části jsou popsány nejdůležitější metody. Problematika rozpoznání řečového signálu je velmi rozsáhlá. Další kapitola se zabývá porovnáním a popisem řečových systémů navzájem. Závěr práce je věnován vlastnímu řešení využití těchto systémů.

Klíčová slova: analýza řeči, syntéza řeči, rozpoznání, programování, Fourierova transformace, filtrace, implementace, restaurace signálů

Speech recognition, analysis and synthesis in application

Summary: The aim of this thesis is the evaluation of systems that recognize speech signal followed by the realization of the system responding to voice commands. The first part is devoted to a description of the basic methods of analysis and synthesis of speech signals. In this section I describe the most important methods. The issue of recognition of the speech signal is very extensive. The next chapter is about the comparison and description of main speech systems. In the conclusion I describe the actual solution and its application of these systems.

Key words: analysis of speech, speech synthesis, recognition, programming, Fourier transformation, filtration, implementation, restoration of signals

Obsah

1. Úvod

Komunikace pomocí řeči je pro člověka nejpřirozenější. Lidé mezi sebou komunikují od pradávna a řeč patří k nejstarším typům komunikace. Je tedy přirozené, že by člověk také rád ovládal přístroje pomocí řeči. Ovládání pomocí řeči je rychlejší a odbourává překážky mezi člověkem a strojem. Stroje či zařízení se tak stále více přibližují člověku.

Nevýhodou těchto systémů je náročnost na výpočetní výkon a velká databáze řečových signálů. Pro úspěšné rozpoznání je potřeba mít velkou databázi řečových signálů, abychom dosahovali co nejlepších výsledků.

1.1 Historie

Historie rozpoznání řečového signálu spadá do 18. století, kdy byly provedeny první pokusy o rozpoznání řeči pomocí mechanického syntetizátoru. Docházelo k velkému zkreslení signálu vlivem různých šumů. Proto byly pokusy v této oblasti spíše ojedinělé.

Zlom v oboru rozpoznávání řeči nastal až s příchodem číslicových obvodů. Číslicové obvody na rozdíl od analogových pracují jen se dvěma stavy. Je tedy lehké odstranit nežádoucí šumový signál. Dochází, ale i k nepřesnému rozpoznání signálu, neboť celé analogové spektrum, musíme vyjádřit pouze pomocí dvou hodnot.

Další velký zlom v rozpoznání řečového signálu přišel s příchodem prvních osobních počítačů. Možnost ovládání počítače pomocí řeči se jevila nejsnadněji. V té době však technologie nebyla na takové úrovni, aby se počítač mohl ovládat pomocí lidské řeči. Tento princip vyžadoval vysoký výpočetní výkon.

Od té doby vznikaly různé systémy rozpoznání řeči, které byly založeny na odlišných principech.

1.2 Současnost

Rozpoznání řeči v dnešní době zasahuje do mnoha oborů. Metody rozpoznání řeči můžeme rozdělit do třech oblastí. Převod textu na řeč, převod řeči na text a převod řeči na řeč.

Převod textu na řeč umožňuje nevidomým překonávat bariéry mezi nimi a dnešními moderními přístroji. Neboť s rozpoznávači řeči se setkáváme v mobilních telefonech, tabletech, počítačích a dalších přístrojích. Nevidomým tato metoda dává možnost zpětné vazby s těmito přístroji a hlavně jim pomáhá v komunikaci s ostatními lidmi a začleněním

se do společnosti. Přístroj jim přečte zprávu z telefonu, tlačítka mají hlasovou odezvu, programy umožňují zpětnou vazbu s uživatelem pomocí řečového signálu.

Druhou oblastí je převod řeči na text. Zde je opět vhodné využití pro handicapované lidi a to z pohledu titulkování textu. Díky tomu se zrakově postižení lidé mohou dívat na jakýkoliv video záznam. Z videa je rozpoznána řeč a ta je převedena do textové podoby a ta je ve formě titulků prezentována na obrazovce. Dále se této metody využívá všude tam, kde je potřeba zaznamenávat řeč. Dochází k nahrazování zapisovatelů, lidé mluví a počítač s příslušným softwarem naslouchá a převádí daná slova do textové podoby. Setkat se s rozpoznávači řeči můžeme zejména u soudu a ve zdravotnictví. Především u soudu jsou tyto programy hojně nasazovány. Daný program však musí obsahovat speciální řečovou databázi. Příkladem programu, který je nasazován u soudu je Newton Dictate. [1] Tento software se rovněž využívá i ve zdravotnictví. Obsahuje jinou databázi hlasových signálů, které jsou uzpůsobeny pro lékařské účely. Lékař může zapisovat zprávu z vyšetření již při samotném vyšetření, může vzdáleně ovládat počítač pomocí hlasových povelů.

Třetím oblastí je převod řeči na řeč. Tato oblast nedosahuje tak velkého rozmachu jako předchozí dva. Zde se naráží na problémy s jednotlivými jazyky. Každý jazyk má svá pravidla a daný slovosled věty. Pokud se odstraní tyto problémy, bylo by vhodné použití těchto systémů k vytvoření plnohodnotného překladače. Nahradila by se tím role překladatelek. Zatím je však použití systémů spíše jen okrajové. Výsledky z těchto systémů nedosahují veliké úspěšnosti. Příkladem takovéto aplikace, která je založena na rozpoznání řeči a následný překlad na cizí řeč, je Trippo VoiceMagic Translator nebo také překladač od Googlu. První jmenovaná aplikace je určena pro mobilní platformy Android a iOS. Překladač od Googlu má tu výhodu, že ho lidé velmi často používají a proto obsahuje velkou databázi řečových signálů. Systémy ovládané řečí slouží rovněž k navigaci ve skladových systémech, kdy obsluha (skladník) provádí vyskladňování a zaskladňování zboží do regálů na základě povelů mobilních terminálů, které navedou obsluhu do přesné buňky ve skladovém systému. Tímto se značně zvýší přesnost a produktivita práce. Tento systém se úspěšně využívá např. ve velkoskladech obchodních řetězců (Globus). [2]

Všechny tyto oblasti mají jedno společné. Lze jimi ovládat systém či zařízení. Zařízení jsou v dnešní době již na takové úrovni, že s námi mohou pomocí hlasového modulu zpětně komunikovat. Jedná se o jakousi umělou inteligenci, avšak její možnosti jsou značně omezené. Většinou jsou tyto systémy prostřednictvím internetu napojeny na

velké řečové databáze. Výhodou je zabudování těchto systémů s hlasovým modulem i do většiny dnešních moderních mobilních telefonů. S rozpoznáním řečového signálu se setkáváme téměř na každém kroku.

Je potřeba zmínit, že občas dochází k problémům s rozpoznáním příkazu. To bývá způsobeno šumovým napětím. Stačí, aby projelo auto nebo se zavřely dveře a již dochází ke zkreslení příslušného řečového povelu. Systémy na to pak reagují různě. Buď je provedena jiná akce, než měla na daný signál nastat nebo systém na daný zvuk nereaguje, neboť jej nemá v databázi. Další nevýhodou je vazba rozpoznávačů na slovníky. Proto není prozatím možné masové nasazení rozpoznávání řeči. Pro daný obor musí existovat obsáhlý řečový slovník.

2. Cíle práce

Cíle této diplomové práce můžeme shrnout do několika bodů:

- zhodnocení využití řeči v aplikacích;
- výhody a nevýhody rozpoznávačů;
- porovnání aplikací využívající rozpoznávače;
- vlastní řešení využívající rozpoznání řeči.

Výstupem této práce je naprogramování dvou jednoduchých příkladů využívající rozpoznání řečového signálu. Každý tento program vychází z jiného principu. První program je aplikace umožňující převod textu do řečové podoby. Druhý program slouží k převodu hlasových povelů do textové podoby. Následně je tento vytvořený program porovnán s obdobnými dostupnými programy, jež se zabývají problematikou převodu řečového signálu.

3. Řeč

Abychom se mohli zabývat zpracováním řeči, je důležité si uvědomit, jak řeč vzniká. Při zpracování řečového signálu se snažíme co nelépe napodobit řečové ústrojí člověka.

Řečové ústrojí můžeme rozdělit na tři části:

- zdroj energie;
- generátor;
- modifikující ústrojí.

Zdrojem energie v řečovém ústrojí člověka jsou plíce. Generátor je zastoupen hrtanem. Hrtan obsahuje hlasivky, které můžeme rozdělit do dvou skupin. [3] První skupinou jsou kmitající hlasivky, které vytvářejí periodický zvuk. Ke kmitání dochází při vyslovování samohlásek. Frekvence, při kterých hlasivky kmitají, nazýváme základní frekvence $f_0\ (Hz)$. V tabulce 1 je zobrazena základní frekvence člověka. [3]

Tabulka 1: Základní frekvence člověka

	$f_0\ (Hz)$
muži	90 – 120
ženy	150 – 300
děti	350 – 400

Zdroj: [3]

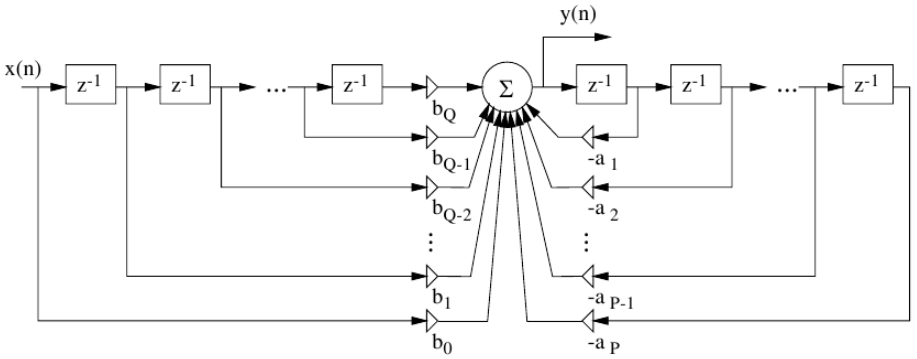
Druhou skupinou jsou hlasivky otevřené, které vytvářejí šum. Dochází k tomu při vyslovování písmen s, f a dalších. Hlasivky v oblasti rozpoznání řeči modelujeme buzením.

Poslední část řečového ústrojí je modifikující ústrojí. Modifikující ústrojí můžeme také nazývat artikulační trakt. Pomocí artikulačního traktu modifikuje signál z hlasivek. Pro účely rozpoznání řeči modelujeme modifikující ústrojí pomocí číslicového filtru. [3]

Tímto byly ve stručnosti popsány základy vzniku řečového signálu u člověka. Dále se tímto popisem nebudu zabývat, neboť to není primární účel této diplomové práce. Bylo však důležité zde uvést tyto základy pro lepší pochopení celé problematiky.

4. Lineární filtrace

Úkolem filtrů je výběr požadovaných složek signálu a potlačení ostatních složek. Proto jsou filtry pro účely rozpoznání řečového signálu velice důležité. Pokud je na vstup číslicového filtru přiveden jednotkový skok, na výstupu filtru se vytvoří impulzní odezva. Impulzní odezva může být konečná (FIR filtr) nebo nekonečná (IIR filtr). Obecně můžeme schéma lineární filtru vyjádřit následujícím obrázkem:



Obrázek 1: Číslicový filtr

Zdroj: [3]

V oblasti řečového signálu používáme nejčastěji filtry k sestavení modelu artikulačního traktu a k potlačení šumu. [3] [4]

4.1 IIR filtr

IIR filtr má nekonečnou impulzní odezvu. Tento filtr je výrazně nižšího řádu, než FIR filtr se stejnými vlastnostmi. Nižší výpočetní náročnost souvisí rovněž s kratším zpožděním mezi vstupem a výstupem. Nevýhodou tohoto filtru je jeho poměrně složité sestavení a citlivost na přesnost výpočtů. Není rovněž zaručena stabilita filtru. Stabilita je závislá na parametrech. Nutná je alespoň jedna zpětná vazba. IIR filtr je zobrazen na obrázku 1. [3]

Diferenční rovnice IIR filtru vypadá následovně:

$$y(n) = \sum_{k=0}^{Q} b_k x(n-k) - \sum_{k=1}^{P} a_k y(n-k) \qquad (1)$$

kde:

$x(n-k)$ *(-)* ..... jsou aktuální a zpožděné vstupy

$y(n-k)$ *(-)* ..... jsou zpožděné výstupy

a_k, b_k *(-)* ..... jsou vstupní a výstupní koeficienty filtru

4.2 FIR filtr

FIR filtr má naopak od předchozího filtru konečnou impulzní odezvu. FIR filtr nemá žádnou zpětnou vazbu a jeho výstup je určen výrazem:

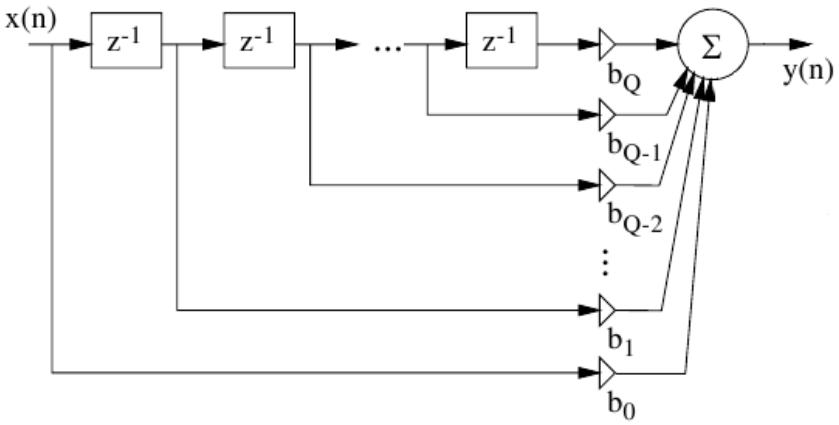
$$y(n) = \sum_{k=0}^{Q} b_k x(n-k) \qquad (2)$$

kde:

$x(n-k)$ *(-)* ..... jsou aktuální a zpožděné vstupy

b_k *(-)* ..... jsou vstupní koeficienty filtru

Tento filtr může mít lineární fázi a nemůže být nestabilní. Neexistuje jeho obdoba v analogové technice. Někdy bývají tyto filtry nazývány konvoluční nebo nerekurzivní. Schéma FIR filtru je zobrazeno na obrázku 2. [5]



Obrázek 2: FIR filtr

Zdroj: [3]

5. Analýza řečového signálu

Analýza slouží jako příprava k hodnocení signálu a identifikaci vlastností. Vstupem je analyzovaný signál. Výstupem analýzy je popis tohoto signálu vhodnými parametry.

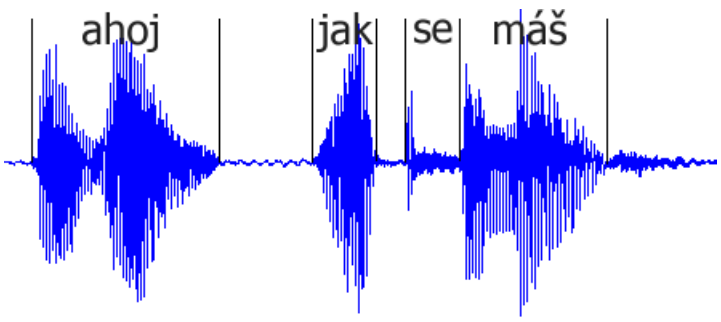
Rozlišujeme dva druhy signálů:

- deterministické;
- náhodné.

Deterministickým signálem je například sinusový nebo zjednodušený obdélníkový průběh. Tyto signály mají tu nevýhodu, že obsahují pouze omezené množství informací. Můžeme u nich určit pouze 2 nebo 3 parametry a lze je jednoduše popsat pomocí matematické rovnice. Řeč patří mezi náhodné signály. Proto rovněž řečový signál nemůžeme nikdy přesně popsat matematickou rovnicí. Dochází k neustálým změnám paramentů. [3]

Máme-li hlasový signál, je nutné ho převést do digitální podoby. Toto zajišťuje pulsní kódová modulace. Před tím, než provedeme pulsní kódovou modulaci je vhodné signál předzpracovat. [3]

Pro analýzu signálu byla použita zkušební větu: „Ahoj, jak se máš“. Tato věta byla nahrána do počítače pomocí programu Audacity (jedná se o volně dostupný software). Data byla uložena ve formátu wav. Vzorkovací frekvence *Fs = 8 kHz*, počet bitů 16 (kvantovací úroveň). Následně bylo provedeno zpracování pomocí modelovacího programu Matlab. Zkušební věta je zobrazena na obrázku 3. Jak je z obrázku patrné, po slově „ahoj“ vznikla pomlka v řeči.



Obrázek 3: vyobrazení zkušební věty v Matlabu

Zdroj: vlastní

5.1 Předzpracování řečového signálu

V častých případech je vhodné řečový signál nejprve předzpracovat. Po předzpracování signálu můžeme signál snadněji analyzovat.

5.1.1 Střední hodnota signálu

Střední hodnota signálu neobsahuje žádnou užitečnou informaci, naopak může být rušivým elementem. Proto by v ideálním případě měla být rovna nule, záleží však na použitém hardwaru. Střední hodnotu signálu můžeme vypočítat podle vzorce:

$$s_s = \frac{1}{N} \cdot \sum_{n=1}^{N} s\,(n) \qquad (3)$$

Kde:

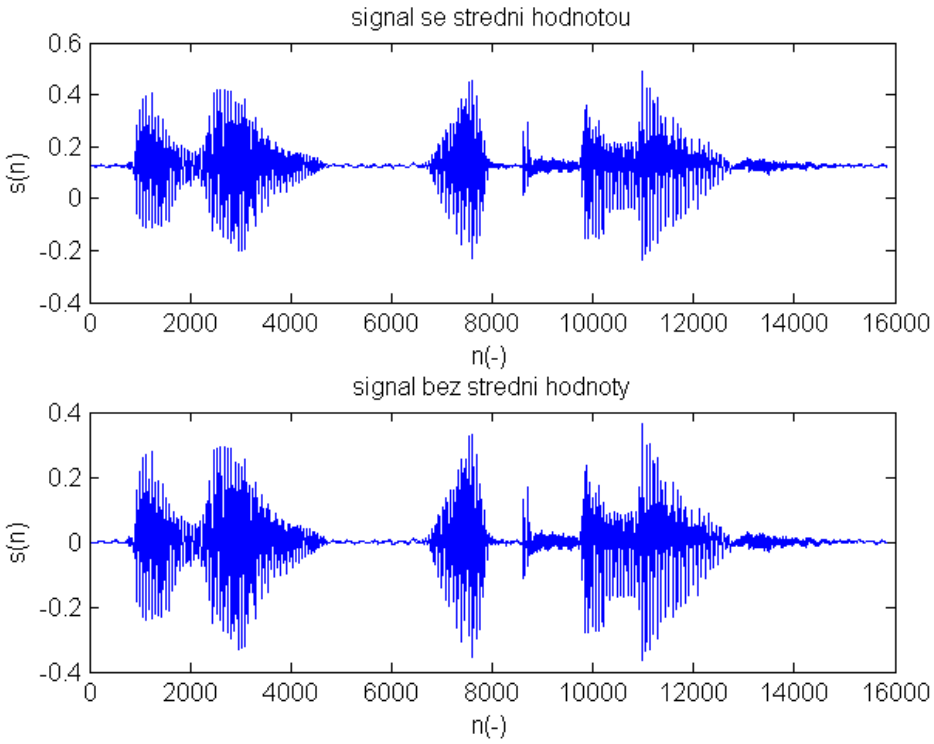
$s\,(n)$ *(-)* ..... je velikost signálu v n-tém vzorku

N (-) ..... je celkový počet vzorků

Ustředněný signál pak vypočteme:

$$s_n(n) = s(n) - s_s \qquad (4)$$

Tím se zbavíme střední hodnoty signálu. Na obrázku 4 je znázorněno odstranění střední hodnoty signálu ze zkušebního souboru.



Obrázek 4: Odstranění střední hodnoty signálu

Zdroj: vlastní

Tento postup je možný pouze tehdy, známe-li konec řečového souboru. Při zpracování řeči v reálném čase však nelze daný postup využít. Proto se pro zpracování řeči v reálném čase signál rozdělí na jednotlivé bloky. Následně se z každého bloku vypočítá střední hodnota a ta se pak odečte od příslušného bloku. [3]

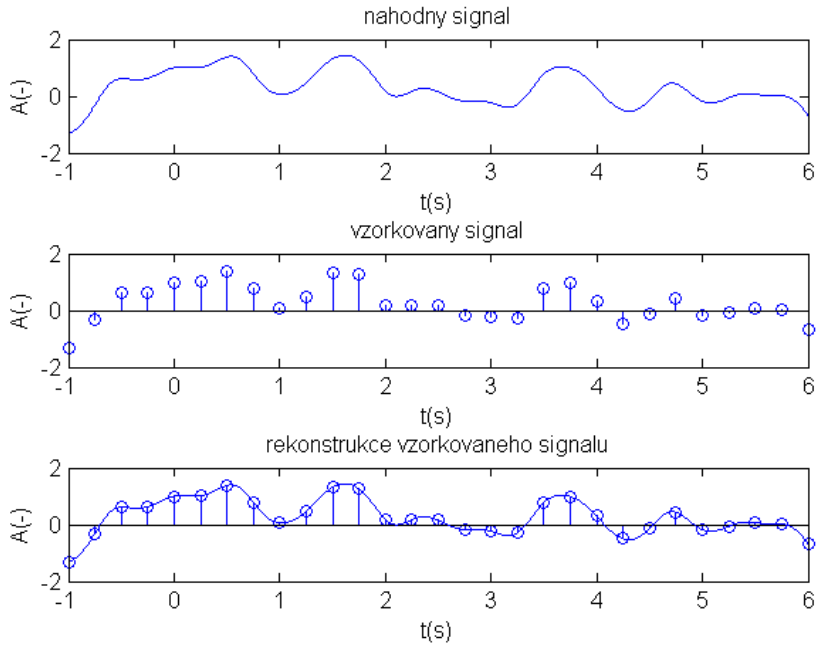
5.2 Pulsní kódová modulace (PCM)

Tento proces můžeme také nazvat digitalizace. Jedná se o převod řeči do posloupnosti čísel, která pokrývají celý spojitý signál. Každé číslo reprezentuje určitý spojitý úsek signálu. Digitalizaci můžeme rozdělit na dvě fáze, a to vzorkování a kvantizaci s kódováním. [6]

5.2.1 Vzorkování

Vzorkování popisuje následující definice: *„Vzorkování je transformace signálu* $s(t)$ *spojitého v čase, na posloupnost vzorků* $s_n = (n \cdot T)$ *diskrétních v čase. Přitom toto vzorkování probíhá v časových okamžicích* $t_n = n \cdot T$*, kde T je perioda vzorkování a n je z množiny přirozených čísel včetně nuly.“* [6]

Vzorkování je možné si představit jako vynásobení analogového signálu Dirakovy impulsy, které mají stejnou periodu. Dirakův impuls je nekonečně malý impuls s nekonečnou amplitudou, jehož plocha je rovna jedné. Ukázka vzorkování signálu je zobrazena na obrázku 5.



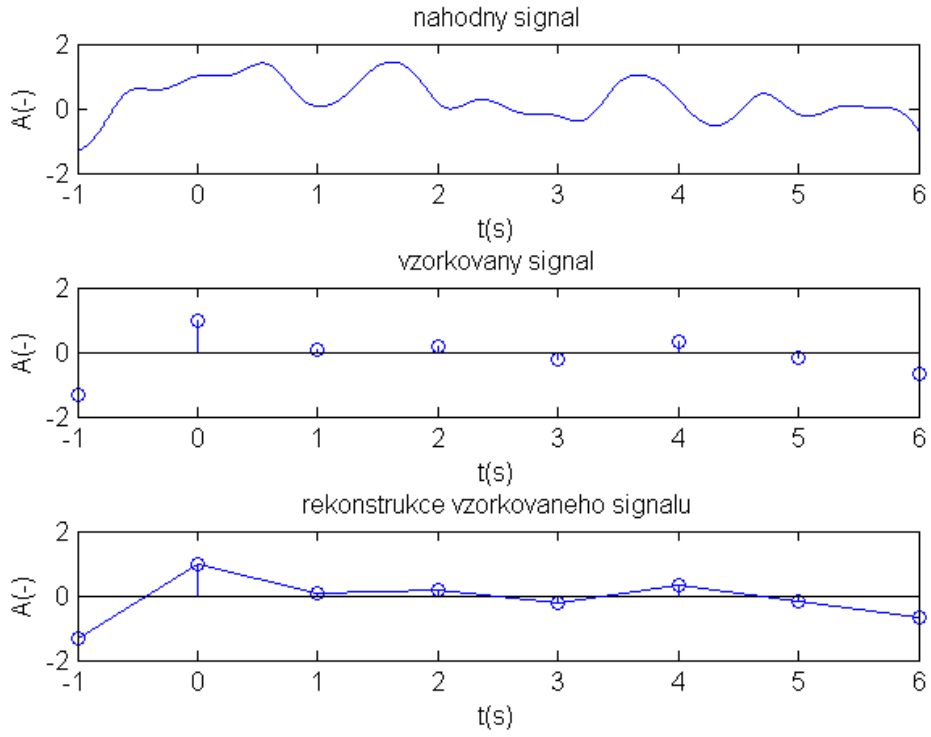
Obrázek 5: Vzorkování signálu

Zdroj: vlastní

Při vzorkování je potřeba brát v úvahu omezení vzorkovací frekvence, která určuje kolikrát za jednu sekundu je provedeno měření. Frekvence vzorkování f_v *(Hz)* by měla být podle Shannon – Kotělníkova vzorkovacího teorému minimálně dvojnásobná oproti maximální frekvenci f_m *(Hz)* obsažené v daném signálu. Musí tedy platit [6]:

$$f_v \geq 2 \cdot f_m \qquad (5)$$

Neplatí-li tento vztah, dochází ke zkreslení, které se nazývá aliasing. Aliasing způsobuje zkreslení vyšších složek frekvence. Pokud není možné zajistit minimálně dvojnásobná vzorkovací frekvenci, je potřeba signál nejprve předzpracovat pomocí antialiasingového filtru. Jedná se o filtr s dolní propustí, který signál ořeže o vysoké frekvence. Důsledek aliasingu je zobrazen na obrázku 6. [6]

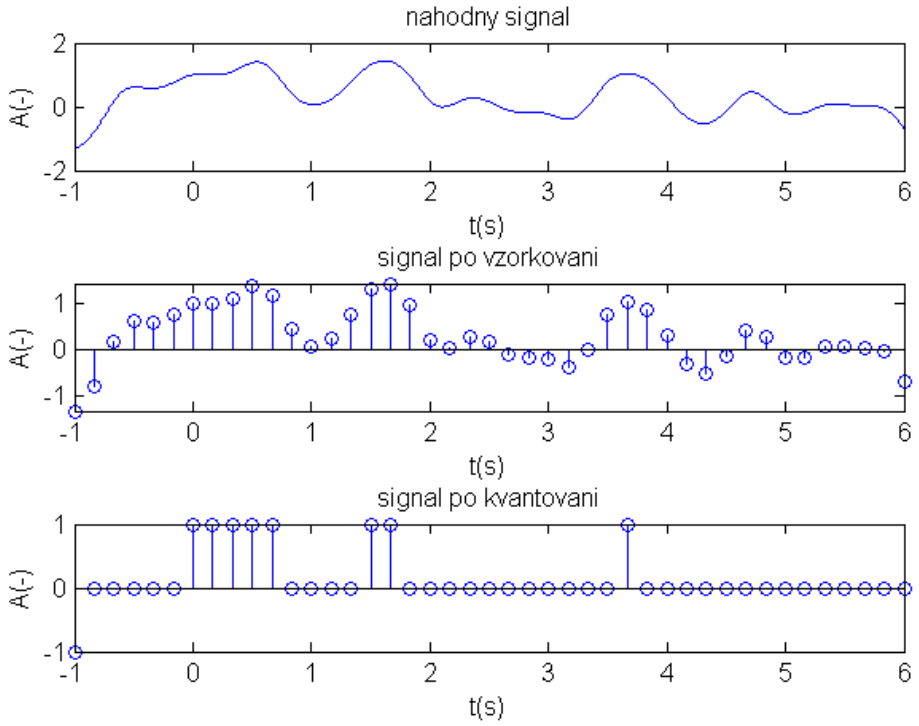


Obrázek 6: Vznik aliasingu

Zdroj: vlastní

5.2.2 Kvantizace a kódování

Kvantování a následné kódování signálu je převod analogové hodnoty po vzorkování do množiny číselných hodnot. Tento proces je prováděn A/D převodníkem. Velikost množiny je určena počtem úrovní kvantování. Úroveň kvantování je ve tvaru 2^B, kde B určuje počet bitů v binárním kódu. Je nutné zvolit hodnotu B takovou, abychom pokryli celý rozsah signálu. Při použití 8 bitového kvantizéru dostáváme 256 hodnot, pro rozlišení. Kvantizací dochází k určité chybě, neboť máme omezenou množinu číselných hodnot. Vzorky se proto zaokrouhlují na nejbližší nižší hodnotu. Tím dochází ke ztrátě informace. Tento proces je nazýván kvantizačním zkreslením nebo také kvantizačním šumem. Snažíme se zajistit, aby byl kvantizační šum co nejmenší. Použijeme-li však velkou úrovně kvantování, dochází ke značným nárokům na paměť. Proto se snažíme volit optimální úroveň kvantování což je s kvantovací úrovní 8 nebo 16 bitů v závislosti na přesnosti určení. Princip kvantování je zobrazen na obrázku 7. [6][7]



Obrázek 7: Kvantování

Zdroj: vlastní

5.3 Zpracování v časové oblasti

Pro zpracování signálu v časové oblasti je nutné rozdělit signál na menší úseky, kterým říkáme rámce nebo také okna. Typická délka rámce je 20 až 25 ms. Při vzorkovací frekvenci $f_v = 8000\,Hz$ je velikost rámce 160 až 200 vzorkům. Jednotlivé rámce se mohou překrývat nebo mohou být bez překrytí. [3]

Rámce bez překrytí nebo s velmi malým překrytím mají tu výhodu, že kladou menší nároky na výpočetní výkon. Hodnoty parametrů mezi jednotlivými rámci se však mohou výrazně lišit. Rámce s velkým překrytím mají velké nároky na výpočetní výkon, hodnoty parametrů mezi sousedními rámci se příliš neliší. Proto je potřebné zvolit optimální překrytí rámců. Nejpoužívanějším překrytím rámců je 10 ms. Počet rámců, můžeme stanovit podle vzorce:

$$N_{ram} = 1 + \left|\frac{N - l_{ram}}{s_{ram}}\right| \qquad (6)$$

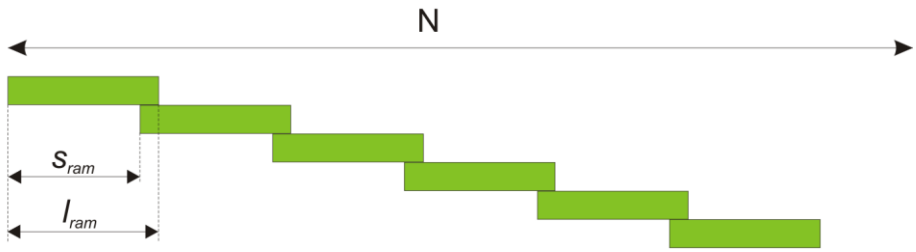
Kde:

N (-)..... je velikost signálu;

l_{ram} *(-)*..... je velikost rámce;

s_{ram} *(-)*..... je velikost rámce bez překrytí.

Názorněji je to ukázáno na obrázku 8.



Obrázek 8: Překrytí rámců

Zdroj: vlastní

Pro výběr rámců se používají okna.

5.3.1 Okna

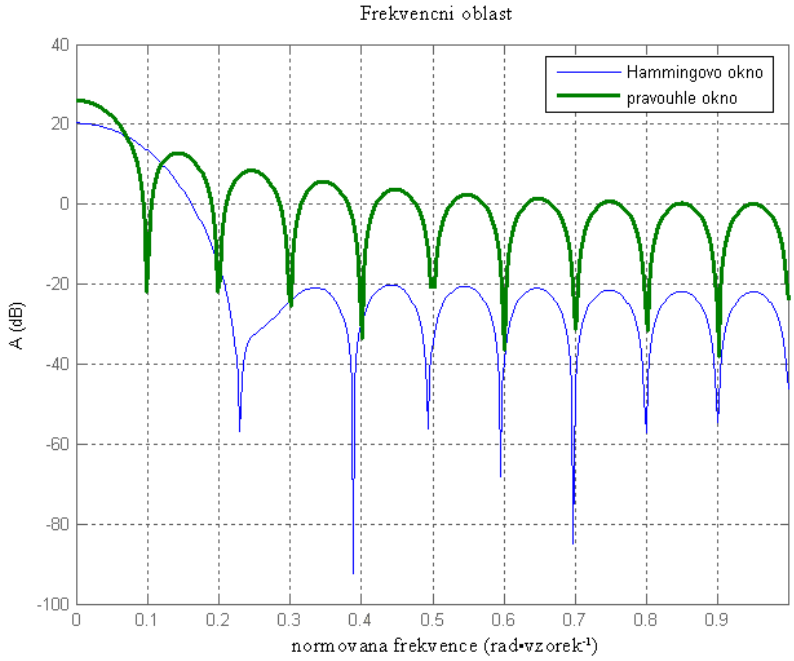
V oblasti rozpoznání řeči používáme dva druhy oken. Tím prvním je pravoúhlé okno, které je definováno vztahem:

$$w(n) = \begin{cases} 1 & pro \quad 0 \leq n \leq l_{ram} - 1 \\ 0 & pro \quad ostatní\ n \end{cases} \qquad (7)$$

Dalším používaným oknem je Hammingovo okno. U tohoto okna dochází k tlumení signálu na okrajích. Hammingovo okno je definováno vztahem [8]:

$$w(n) = \begin{cases} 0.54 - 0.46\, cos\frac{2\pi n}{l_{ram}-1} & pro \quad 0 \leq n \leq l_{ram} - 1 \\ 0 & pro \quad ostatní\ n \end{cases} \qquad (8)$$

V případě pravoúhlého i Hammingova okna se jedná o filtr s dolní propustí. Na obrázku 9 jsou znázorněny spektrální charakteristiky obou oken. Charakteristika pravoúhlého okna je zobrazena zeleně. Je patrné, že šířka pásma propustnosti pravoúhlého okna je dvakrát menší než u Hammingova okna. Z toho vyplývá, že při použití Hammingova okna dochází k vyšším útlumům vyšších frekvencí. [8]



Obrázek 9: Spektrální charakteristiky Hammingova a pravoúhlého okna

Zdroj: vlastní

5.3.2 Střední krátkodobá energie

Střední krátkodobá energie je energie v jednom rámci. Pro výpočet střední krátkodobé energie se používá tento vzorec [3]:

$$E = \frac{1}{l_{ram}} \sum_{n=0}^{l_{ram}-1} x^2\,(n) \qquad (9)$$

kde:

l_{ram} *(-)* ..... je velikost rámce;

$x^2\,(n)$ *(-)* ..... je druhá mocnina hodnoty signálu n-tého vzorku.

Pomocí střední krátkodobé energie můžeme určit přítomnost řeči, to lze ale pouze prostředí, ve kterém se nevyskytuje žádný šum. Dále můžeme pomocí střední energie detekovat hlásky. [9]

5.3.3 Počet průchodů nulou

Počet průchodů nulou se určuje v každém rámci. Je definován vztahem:

$$Z = \frac{1}{2} \sum_{n=1}^{l_{ram}-1} |sign\ x(n) - sign\ x(n-1)| \qquad (10)$$

Kde *sign (x)* je znaménková funkce, která je definována následovně:

$$sign\ x(n) = \begin{cases} +1 & pro \quad x(n) \geq 0 \\ -1 & pro \quad x(n) < 0 \end{cases} \qquad (11)$$

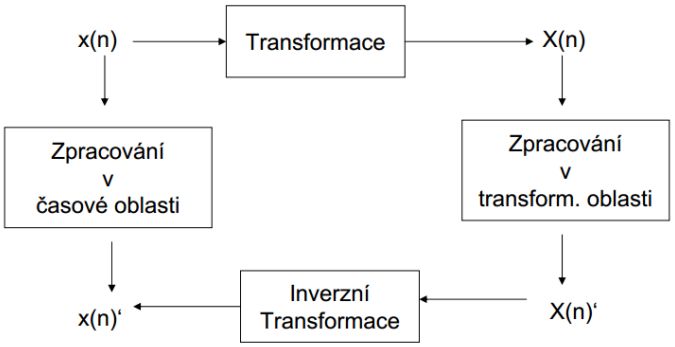
Počet průchodů nulou se používá pro určení začátku a konce mluvy nebo pro určení základního tónu. [9]

5.4 Zpracování ve frekvenční oblasti

Řeč je ve frekvenční oblasti reprezentována velikostí frekvenčních složek. Základ mnoha metod ve frekvenční oblasti je založen na aplikaci Fourierovy transformace. Neboť definiční vzorec Fourierovy transformace je integrál a při číslicovém zpracování dostáváme pouze hodnoty v diskrétních okamžicích, nikoli spojitou funkci. Proto se používá diskrétní Fourierova transformace. [10]

5.4.1 Diskrétní Fourierova transformace

Diskrétní Fourierova transformace je označována DFT. Jejím výsledkem je spektrální funkce, spektrum. DFT se můžeme představit takto:



Obrázek 10: Podstata DFT

Zdroj: [11]

Definice přímé transformace:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j2\pi\frac{nk}{N}} \qquad (12)$$

kde:

$x(n)$ *(-)*..... je signál v n-tém vzorku,

N (-) ..... je počet vzorků.

Inverzní (zpětná) transformace:

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k) \cdot e^{j2\pi\frac{nk}{N}} \qquad (13)$$

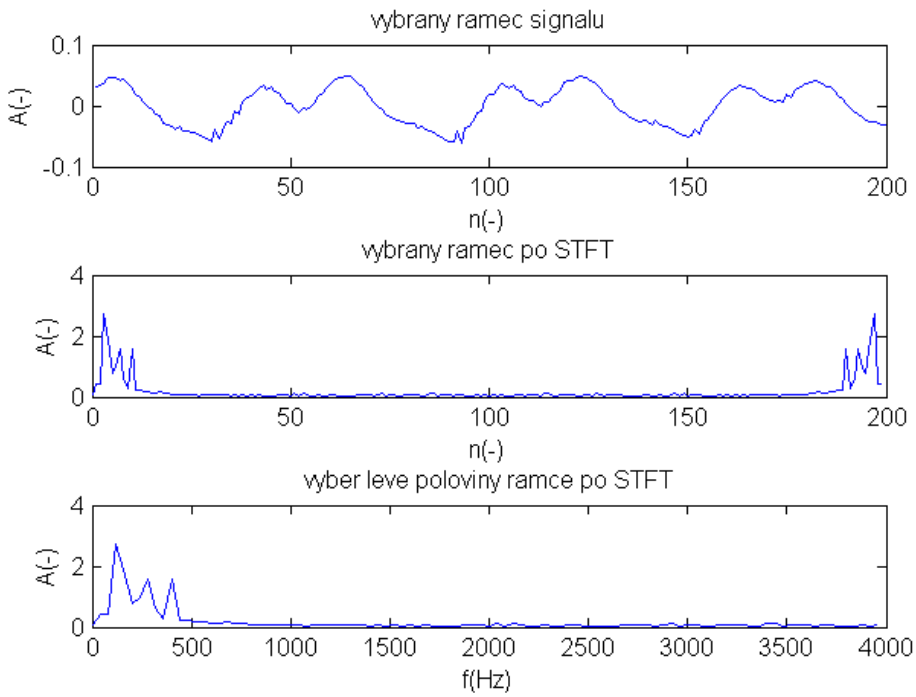
V důsledku diskretizace časové a frekvenční oblasti jsou posloupnosti *x(n)* a *X(k)* periodické. Pro zrychlení výpočtu se používá krátkodobá diskrétní Fourierova transformace. [5]

5.4.2 Krátkodobá diskrétní Fourierova transformace

Pro označení krátkodobé diskrétní Fourierovy transformace se používá značka STFT. Jejím výsledkem je časový průběh krátkodobého spektra. Při použití klasické Fourierovy transformace není možné časově lokalizovat spektra. Při použití signálu vynásobeného oknem, získáme spektrum, jež je funkcí dvou proměnných frekvence a časového posunutí. [12]

Krátkodobá diskrétní Fourierova transformace je zobrazena na obrázku 11. Ze stejné zkušební věty: „Ahoj, jak se máš“, byl vybrán rámec o velikosti 200 vzorků. Následně byla provedena krátkodobá Fourierova transformace. Tato transformace je vykreslena na prostředním obrázku. Jak je z obrázku patrné, jedná se o komplexně sdruženou funkci (neboť levá část je stejná jako pravá). Postačí proto vybrat pouze levou část. To je zobrazeno na posledním obrázku. Pokud bychom tyto hodnoty umocnili na druhou, dostaneme spektrální hustotu výkonu.

Vypočítali jsme spektrum vzorkovaného signálu. Spektrum je periodické s periodou N čísel, čemuž odpovídá vzorkovací frekvence F_s. Protože byl rámec signálu vybrán pomocí okna, tak výsledné spektrum obsahuje vlastnosti tohoto okna.



Obrázek 11: Krátkodobá diskrétní Fourierova transformace

Zdroj: vlastní

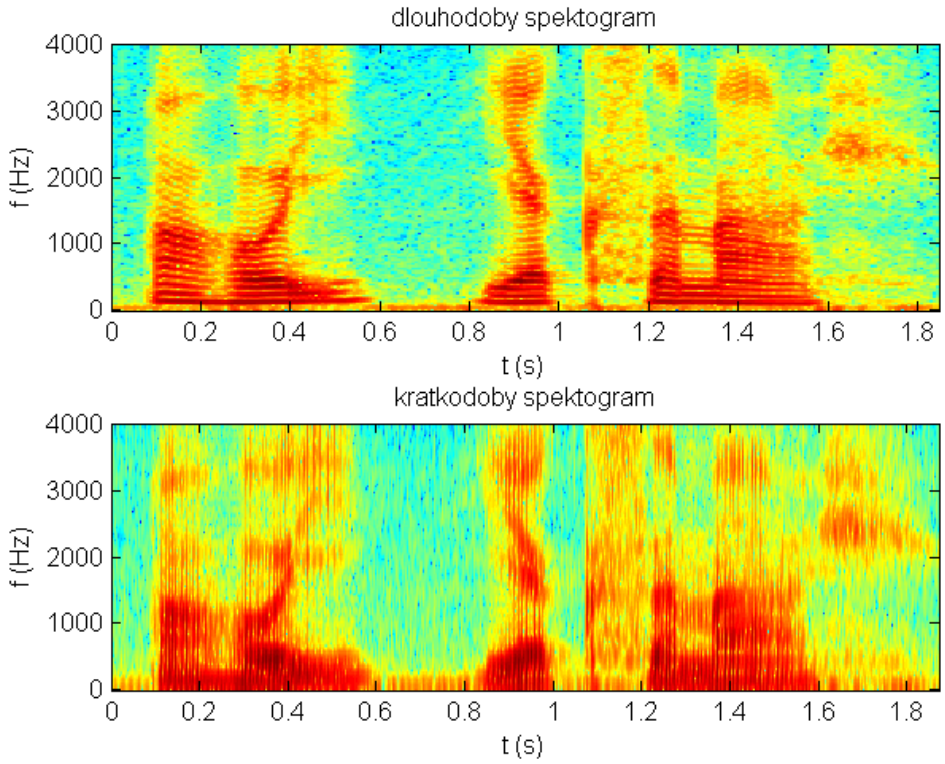
5.4.3 Spektogram

Spektogram zobrazuje spektrální hustotu výkonu daného signálu v čase. Spektrální hustotu výkonu dostaneme použitím krátkodobé Fourierovy transformace. Následně absolutní hodnotu výsledku umocníme na druhou.

Podle délky rámce rozlišujeme dva druhy spektogramu:

- dlouhodobý;
- krátkodobý.

Na obrázku 12 je možné vidět porovnání obou spektogramů. Na horním obrázku je dlouhodobý spektogram, můžeme lépe určit jednotlivá maxima. Na spodním obrázku se nachází krátkodobý spektogram u něhož lze snadno určit časovou základnu. [3]



Obrázek 12: Grafické znázornění dlouhodobého a krátkodobého spektrogramu

Zdroj: vlastní

5.5 Kepstrální analýza

Pojem kepstrum vznikl přesmyčkou slova spektrum. Slouží pro oddělení buzení a modifikace. Oddělením buzení se zbavujeme závislosti na řečníkovi. Názvy veličin v kepstrální analýze jsou stejně jako samotné kepstrum odvozené. Vodorovná osa je nazývána kvefrence a její jednotkou je čas. Dále zde můžeme narazit na pojem liftrování což je přesmyčka od filtrování. Definice kepstra je následující [12]:

$$C(\tau) = F^{-1}\{\log|F\{x(t)\}|^2\} \qquad (14)$$

Jedná se o zpětnou Fourierovu transformaci logaritmu spektrální hustoty výkonu. Signál je dán konvolucí buzení a modifikačního ústrojí. Celou závislost můžeme vyjádřit pomocí následujícího vzorce [3]:

$$s(n) = e(n) \cdot h(n) \qquad (15)$$

kde:

e(n) (-) ..... představuje buzení;

h(n) (-) ..... modifikační ústrojí.

Pro lepší pochopení upravíme definici kepstrální funkce. Využijeme linearity zpětné Fourierovy transformace a dostaneme výraz [3]:

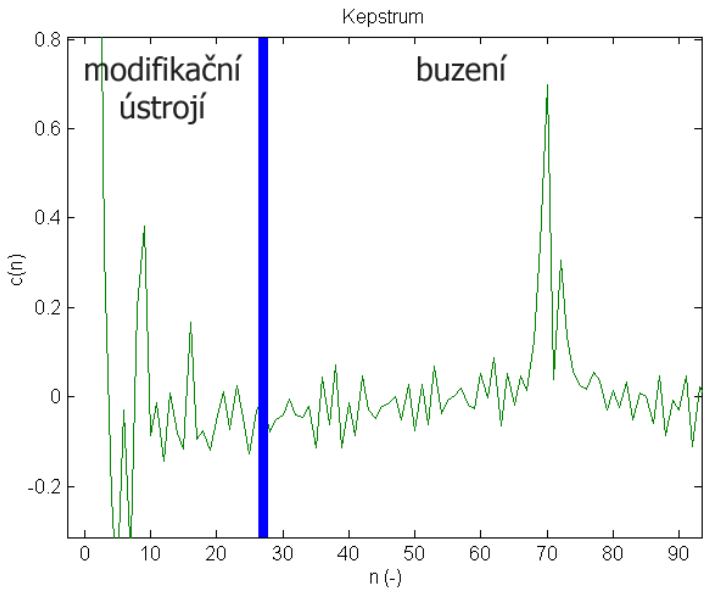
$$c(n) = F^{-1}\{\ln|E(f)|^2\} + F^{-1}\{\ln[H(f)]^2\} = c_e(n) + c_h(n) \qquad (16)$$

kde:

$c_e(n)$ *(-)* ..... je kepstrální koeficient pro buzení;

$c_h(n)$ *(-)* ..... je kepstrální koeficient pro modifikační ústrojí.

První člen výrazu je zpětná Fourierova transformace spektrální hustoty výkonu buzení, druhý člen je zpětná Fourierova transformace spektrální hustoty výkonu modifikačního ústrojí. Aplikace kepstra na vzorovou větu: „Ahoj, jak se máš.“ je vidět na obrázku 13.



Obrázek 13: Kepstrum

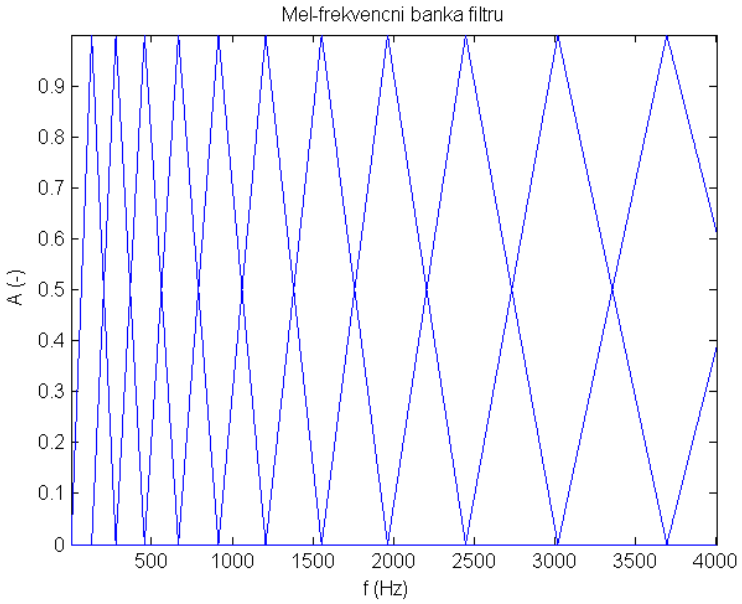
Zdroj: vlastní

5.5.1 Mel-frekvenční kepstrum

Kepstrum vypočítané pomocí diskrétní Fourierovy transformace má stejné frekvenční rozlišení. Lidské ucho však rozezná lépe nižší frekvence než vyšší. Proto se zavádí melovské kepstrální koeficienty, které se snaží kompenzovat nelineární vnímání frekvencí. Využívá se přitom nahrazení DFT za trojúhelníkové pásmové filtry rozložené v melovských frekvencích. Je nutné upravit kmitočtovou osu podle následujícího vzorce:

$$f_{mel} = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \qquad (17)$$

Hodnota f_{mel} je v melech a f je uváděna v hertzích. Na obrázku 14 je vidět Mel-frekvenční banka filtrů. Jak je z obrázku patrné, dochází k rozšiřování filtrů pro vyšší frekvence. [3] [6]

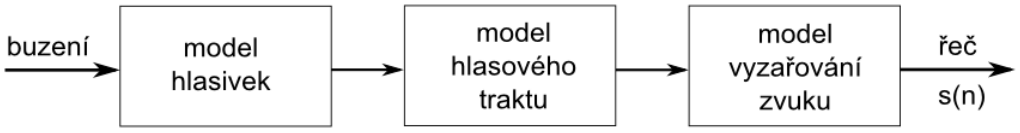


Obrázek 14: Mel-frekvenční banka filtrů

Zdroj: Vlastní

5.6 Lineární prediktivní analýza (LPC)

Jedná se o nejefektivnější metodu analýzy akustického signálu. Metoda se snaží odhadnout parametry modelu hlasového traktu při nízké výpočetní zátěži. Předpoklad LPC metody je, že k-tý vzorek signálu $s(k)$ lze popsat pomocí lineární kombinace předchozích vzorků a buzení u(k). Model artikulačního ústrojí (viz. obrázek 15) se skládá ze tří částí. Nejdůležitějším prvkem je hlasové ústrojí. [6]



Obrázek 15: Model artikulačního ústrojí

Zdroj: [3]

Hlasivky modelujeme jako dolní propust druhého řádu (IIR filtr druhého řádu). Hlasový trakt se modeluje pomocí kaskády dvojpólových rezonátorů, kdy každý rezonátor je IIR filtr druhého řádu. Počet rezonátorů v kaskádě je závislý na vzorkovací frekvenci. Model vyzařování zvuku je zastoupen FIR filtrem. Výsledný model artikulačního ústrojí pak dostaneme právě vynásobením jednotlivých složek. Po úpravě vznikne následující vzorec [3]:

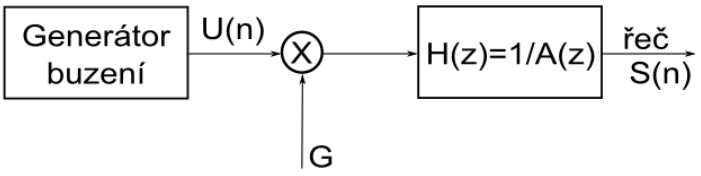
$$H(z) = \frac{1}{A(z)} = \frac{1}{1 + \sum_{i=1}^{P} a_i z^{-i}} \qquad (18)$$

kde:

P (-) ..... je řád systému;

a_i *(-)* ..... koeficienty filtru.

Jedná se o čistý IIR filtr. Pro vzorkovací frekvenci F_s = *8 kHz* se často volí hodnota *P = 10*. Po zapojení tohoto filtru do soustavy dostáváme následující obvod (obrázek 16).



Obrázek 16: Model vytvoření řeči s lineárním číslicovým filtrem

Zdroj: [3]

diferenční rovnice tohoto obvodu je:

$$s(n) = Gu(n) - \sum_{i=1}^{P} a_i s(n-i) \qquad (19)$$

Kde:

G (-) ..... představuje zesílení;

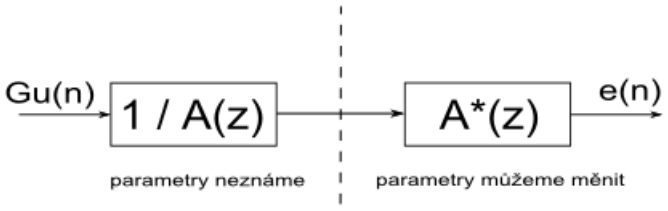
$u(n)$ *(-)* ..... představuje budící signál;

$s(n-i)$ *(-)* ..... jsou zpožděné výstupy;

P (-) ..... je řád systému;

a_i *(-)* ..... koeficienty filtru.

Parametry filtru jsou neznámé a musíme je identifikovat. Identifikace se provádí, připojením inverzního filtru FIR, který je na obrázku 17 znázorněn jako *A*(z)* filtr. Parametry tohoto filtru nastavujeme tak dlouho, až na výstupu dostaneme minimální energii *e(n)*. Po dosáhnutí tohoto stavu jsou koeficienty obou filtrů shodné.
Platí *A(z) = A*(z).* [3] [6]



Obrázek 17: Určování parametrů IIR filtru

Zdroj: vlastní

Princip lineární predikce spočívá v odhadu skutečného vzorku ze vzorků předchozích. Známe předchozí vzorky signálu a na jejich základě odhadujeme současný vzorek. Při odhadování vzniká chyba predikce, která je dána rozdílem skutečné a odhadované hodnoty. Při dobře určených koeficientech IIR filtru je chyba predikce rovna buzení. [6]

6. Metody rozpoznání řeči

Při rozpoznání řečového signálu jsou mnohá úskalí, která brání k vytvoření univerzálního rozpoznávače řeči. Nejčastějšími úskalími jsou:

- proměnlivost prostředí, ve kterém je řeč rozpoznávána;
- variabilita řečníka;
- složitost řešené úlohy.

Hlasy jednotlivých osob se od sebe liší. To je způsobeno odlišnými parametry hlasového ústrojí a odlišným způsobem artikulace. Každý člověk tedy vytváří jedinečný specifický řečový signál. Proto se u rozpoznávání řečových signálů rozdělují úlohy do dvou skupin.

První skupinou jsou systémy závislé na řečníkovi. Tyto systémy jsou naučené a natrénované podle jednoho hlasu a reagují správně pouze na tento hlas. Z toho také vyplývá jejich nevýhoda, neboť správně rozeznávají pouze hlas jedné osoby. Naopak výhodou je malá náročnost na učení v porovnání s druhou skupinou.

Druhou skupinu tvoří systémy, které nejsou závislé na řečníkovi. Jejich velkou výhodou je právě univerzálnost. Je potřeba natrénovat daný systém na stovky i tisíce hlasů.

Překážkou při rozpoznání řeči je i odlišnost hlasu řečníka při různých situacích. Řeč je závislá na našich emocích. Jiný je řečový signál pokud slovo vyslovíme šeptem, v nachlazení, v rozčilení. Je nemožné, aby člověk řekl dané slovo vždy stejně. Pokaždé existují v řečovém signálu nějaké odchylky.

Dále potom rovněž hodně záleží na prostředí, ve kterém je řeč rozpoznávána. Toto nazýváme akustické prostředí. Jedná se zejména o přítomnost okolního šumu. V ideálních podmínkách by bylo vhodné, aby akustika prostředí byla stejná jak při učení systémů, tak při jejich používání. V praxi však není možné zajistit stejné a neměnné laboratorní prostředí bez okolního šumu, proto se v praxi k odstranění šumu používají různé filtry.

Z hlediska metody rozpoznávání dělíme systémy do dvou skupin:

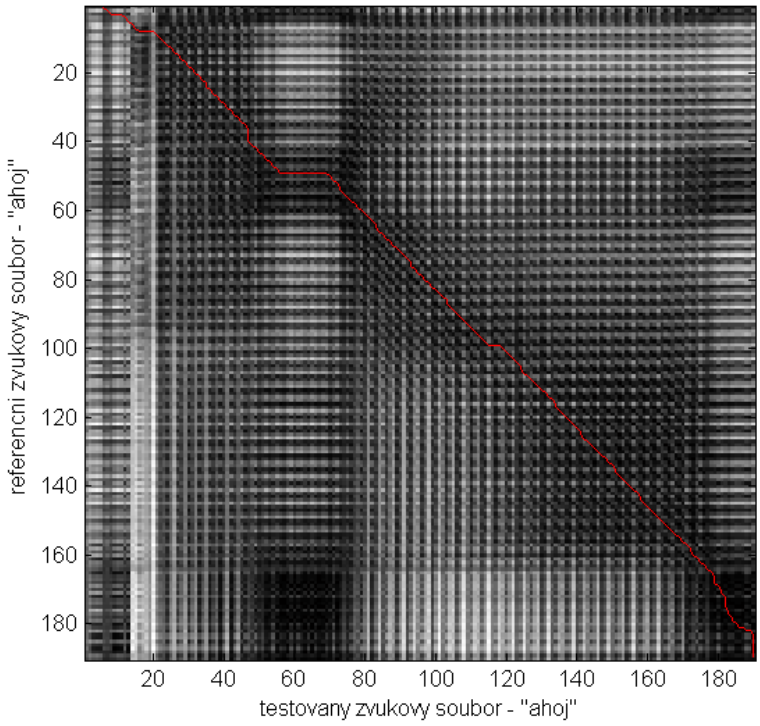
- porovnání se vzory;
- statické metody rozpoznání.

Porovnání se vzory se používalo dříve. Principem této metody bylo přiřadit slovo k danému vzoru. Vypočítávala se vzdálenost slova ke vzoru. Slovo pak bylo přiřazeno ke vzoru, s nímž mělo nejkratší vzdálenost. Rozhodující bylo určení vzdálenosti mezi vzorem a slovem. Vzdálenost se většinou určuje pomocí metody dynamického programování (DTW), které je popsáno níže.

Statická metody rozpoznání řeči jsou založené na principu modelování slova nebo celého řečového signálu pomocí skrytých Markovových modelů. [13]

6.1 Dynamic Time Warping (DTW)

Do češtiny lze tento výraz přeložit jako dynamické deformování času. Účelem této metody je porovnání dvou zvukových vektorů. Jedním z vektorů je referenční zvukový soubor, který se nachází v naučené řečové databázi rozpoznávače. Druhým vektorem je testovaný zvukový soubor. Principem této metody je najít optimální cestu ve spektrální oblasti, která znázorňuje rozdíly mezi referenčním a testovaným souborem. Vzdálenost cesty záleží na podobnosti obou signálů. Čím menší je cesta, tím více jsou dané zvukové soubory stejné. Tato metoda se používá zejména k rozpoznání samostatných slov, kde dosahuje velmi vysoké úspěšnosti. [13]

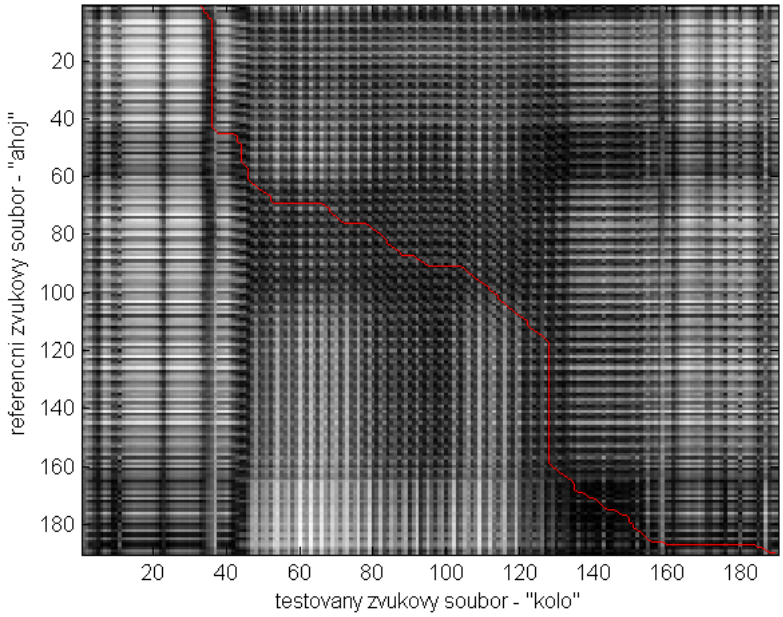


Obrázek 18: Řečový soubor "ahoj" a "ahoj" v DTW

Zdroj: vlastní

Na obrázku 18 je zobrazení DTW v Matlabu pro referenční signál „ahoj“ a testovaný signál „ahoj“. Největší shoda signálu je ve tmavých místech. DTW se snaží nalézt nejkratší optimální cestu. Velikost této cesty je 14,3829 jednotek.

Oproti tomu na obrázku 19 je řečový soubor pro referenční slovo „ahoj“ a testovaným slovem je „kolo“. Na hlavní diagonále ubylo černých míst, značících shodnost signálu. Pro tento obrázek je velikost nejkratší optimální cesty 30,4992 jednotek.

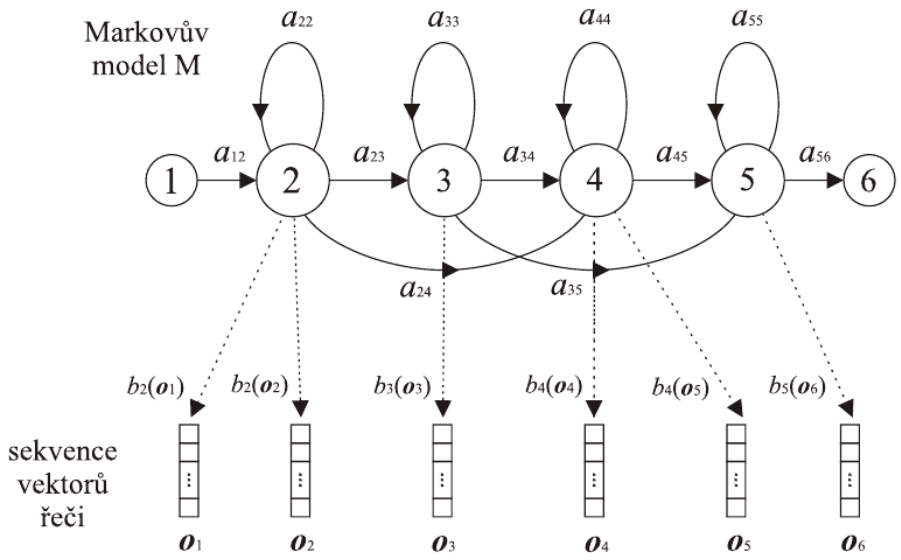


Obrázek 19: Řečový soubor "ahoj" a "kolo" v DTW

Zdroj: vlastní

6.2 Skryté Markovovy modely (HMM)

HMM jsou statickými nástroji pro modelování náhodných signálů. Nejvíce se využívají v oblasti rozpoznání řečového signálu. Fungují na principu pravděpodobnostního konečného automatu. Konečný automat mění v každém časovém okamžiku t svůj stav na stav s_j. Při této změně je generován vektor pozorování o_t , který má výstupní pravděpodobnost $b_j(o_t)$. S rostoucím časem se pak snažíme najít nejpravděpodobnější cestu. Na obrázku 20 je vidět Markovův model, který má šest stavů. [15]



Obrázek 20: Markovův model se šesti stavy

Zdroj: [14]

6.2.1 Rozpoznání izolovaných slov

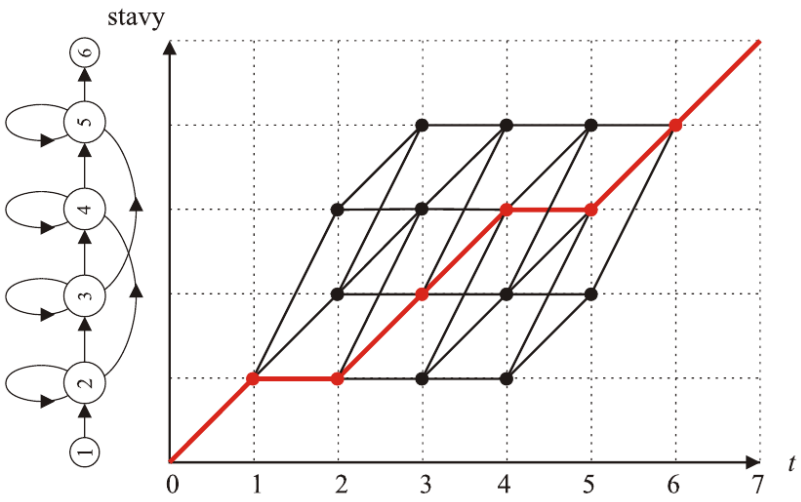
Pro účely rozpoznání samostatných slov je potřeba uvažovat vektory pozorování $o_1, o_2, ..., o_T$ za parametrické vektory, které jsou vypočítány pro jednotlivé rámce řečového signálu. Symbol T představuje počet časových signálů, při nichž dochází ke generování vektorů pozorování. Vektory pozorování pak obsahují typické parametry řečového signálu, jako kepstrální koeficienty, lineární predikční koeficienty apod. Jednotlivá slova jsou popsána posloupností těchto vektorů [16]:

$$O = [o_1, o_2, \dots, o_T] \qquad (20)$$

Pro zjednodušení se můžeme představit, že máme databázi jednotlivých slov. Řečový signál, který chceme určit, si rozdělíme na rámce. V každém rámci pak určíme nějaký parametr. Každý jeden rámec odpovídá jednomu vektoru pozorování. Posloupnost těchto vektorů je popsána rovnicí *(20).* Poté hledáme v databázi slovo, které se nejvíce odpovídá dané posloupnosti. [16]

6.2.2 Rozpoznání plynulé řeči

Při rozpoznání plynulé řeči se používá Viterbiho metoda, která provádí eliminaci nepravděpodobných průchodů. Principem rozpoznání plynulé řeči je rozložení řečového signálu na velmi malé části (slabiky, hlásky,…). Jednotlivé hlásky se pak trénují zvlášť. Výsledná natrénovaná databáze je pak mnohem menší než databáze jednotlivých slov. Při rozpoznávání řeči se hledá nejpravděpodobnější shoda jednotlivých hlásek v databázi. Tento princip rozpoznávání je mnohem rychlejší oproti předchozí metodě rozpoznávání izolovaných slov. Na obrázku 21 je vidět přechod mezi jednotlivými stavy. [14]



Obrázek 21: Stavová posloupnost přechodů mezi jednotlivými stavy

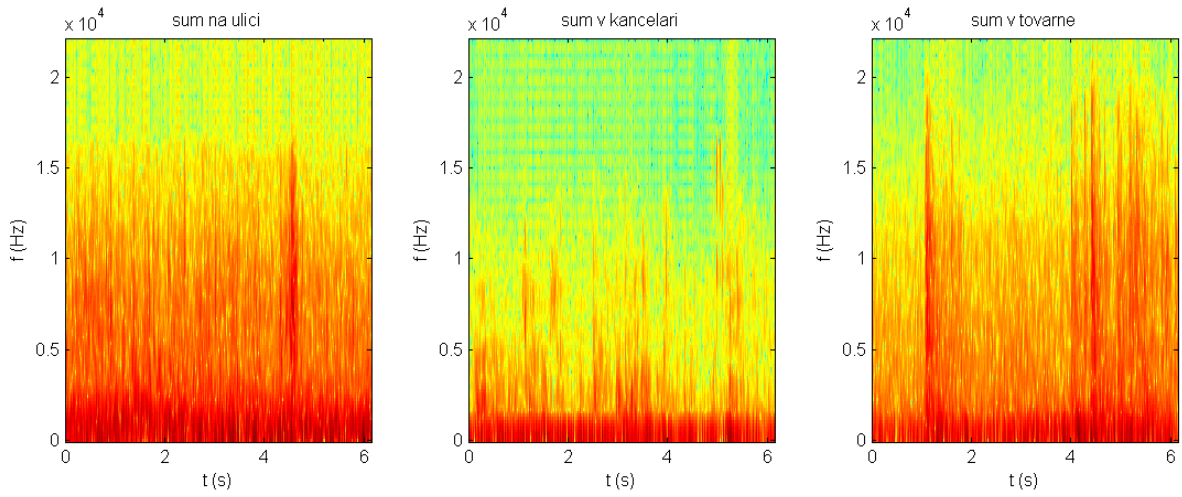
Zdroj: [14]

Problematika rozpoznání řeči pomocí HMM je daleko složitější, než je zde popisováno. Účelem této práce není dopodrobna rozebrat zmiňované metody, ale vytvořit ucelený přehled principů, jež jsou využívány v rozpoznávačích řeči. Další teoretické principy popisující HMM a jejich užití lze nalézt v knize Przemyslawa Dymarského Hidden Markov Model, Theory and Applications.

7. Šumový signál

Je negativní parametr, který omezuje úspěšnost rozpoznávání jednotlivých systémů. Nelze nikdy přesně určit jeho velikost, neboť se v čase mění. Pro různá prostředí jsou odlišné úrovně šumového napětí. Z tohoto důvodu nelze určit jednotnou metodu pro jeho potlačení.

Na obrázku 22 je znázorněna krátkodobá spektrální analýza hluku v různých prostředích. Největší hluk vykazuje prostředí továren. Naopak nejmenší úrovně hluku je dosaženo v kanceláři. Metody pro potlačení hluku v daných prostředích se budou od sebe odlišovat. V prostředí továrního provozu je nutné odfiltrovat vyšší šumové napětí, než v případě kancelářského prostředí. Nejjednodušší odfiltrování bude na ulici, kde úroveň hluku můžeme považovat za téměř konstantní.



Obrázek 22: Šum v různých prostředích

Zdroj: vlastní

Dalším nepříjemný vliv, který vede ke vzniku rušivého napětí pro rozpoznávače, je Lombardův efekt. Je to automatická změna hlasového profilu mluvčího v hlučném prostředí. Pro člověka je přirozené v hlučném prostředí zvýšit hlas. Zvýšením hlasu se změní parametry řečového profilu. Pro účely rozpoznávání může být Lombardův jev mnohem škodlivější než samotný šumový signál. Ian McLoughlin ve své knize uvádí, že pokud dojde ke zvýšení úrovně hluku o 1 dB, zvýší řečník úroveň svého hlasu automaticky o 0,5 dB. [17]

7.1 Způsoby potlačení šumu

Pro potlačení šumu se používají nejčastěji dva způsoby:

- předzpracování řečového signálu;
- přizpůsobení šumu;

Při použití předzpracování řeči, může dojít k odstranění důležité informace ze signálu. Pro předzpracování se nejčastěji používá metoda spektrálního odečítání. Abychom mohli danou metodu použít, musíme znát spektrum šumu. Následně vypočteme spektrum původního signálu jako rozdíl spektra zašuměného signálu a spektra šumu. Výhodou této metody jsou nízké nároky na výpočetní složitost a její jednoduché použití. [18]

Další způsob je přizpůsobit rozpoznávač šumovému napětí. Zde je využíván učící režim rozpoznávače. Učení se provádí v prostředí, ve kterém bude rozpoznávač používán. Tento způsob je omezený jen na konkrétní prostředí. Nemůže se však stát, že by došlo k odstranění důležité informace ze signálu.

8. Rozpoznání řeči v počítačích

V této kapitole se zaměřím na využití rozpoznávačů řeči v počítačích. K těmto účelům je možné využít velké množství komerčních i nekomerčních softwarů pro rozeznání řečového signálu. Proto budou v této kapitole popsány jen některé nejdůležitější programy. Tyto programy můžeme podle typu rozpoznávaného jazyka rozdělit do dvou skupin:

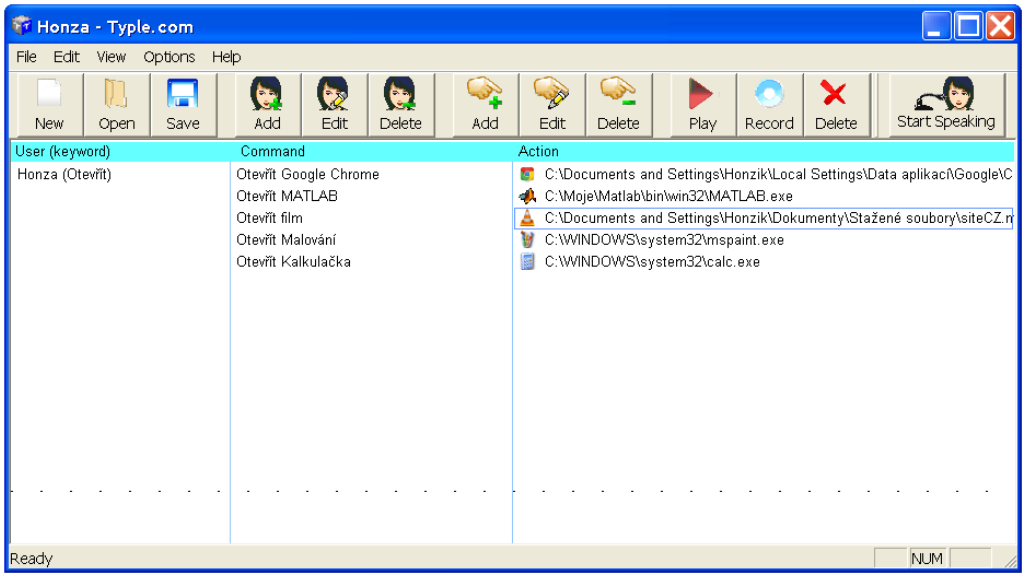
- anglické rozpoznávače;
- české rozpoznávače.

8.1 Aplikace rozpoznávající angličtinu

Neboť většina programů je primárně v angličtině, tak i většina rozpoznávačů řeči reaguje na angličtinu. Proto bude v této části představeno několik programů pro ovládání počítače pomocí hlasových povelů v angličtině.

8.1.1 Typle

Jedná se o program umožňující využít hlasových povelů ke spuštění jednotlivých aplikací. Je možné spouštět aplikace, přehrávat videa nebo písničky pomocí hlasových povelů. Program je zdarma, ale je možné pořídit premium verzi za 13 USD, která obsahuje větší množství příkazů pro ovládání počítače. Na obrázku 23 je vidět uživatelské rozhraní tohoto programu.



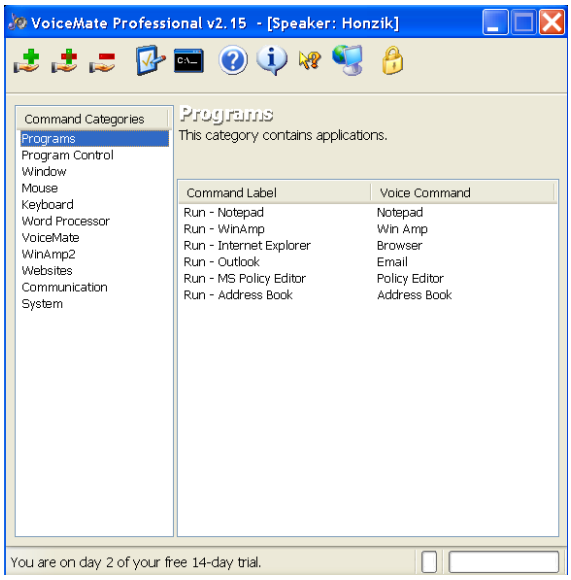
Obrázek 23: Program Typle

Zdroj: vlastní

Program funguje na principu přiřazení příkazu k určitému řečovému signálu. Nejprve se vytvoří uživatele a k němu odpovídající referenční řečový příkaz („otevřít“, „přehrát“, apod.), který je nutné namluvit. Následně se zvolí aplikace, jež chceme pomocí hlasových povelů spouštět. U každé aplikace je potřeba namluvit její název nebo označení, pod nímž budeme danou aplikaci spouštět („film“, „malování“, „kalkulačka“). Pak už stačí jen program spustit. Typle běží na pozadí operačního systému a poslouchá hlasové příkazy, jakmile slyší příkaz odpovídající nějaké operaci, provede tuto operaci. Ukázkou takového příkladu je: „otevři kalkulačku“, „otevři malování“. Pokud při učení příkazů nahrajeme příslušné hlasové povely v češtině, reaguje pak program na povely v češtině.

8.1.2 VoiceMate Professional

Jedná se o komerční software. Pro vyzkoušení si uživatel může stáhnout trial verzi tohoto programu, kterou může používat čtrnáct dnů. Po uplynutí této doby je nutné zakoupit licenci. Program umožňuje spouštět aplikace podle předem nastavených příkazů. Uživatel si může tyto příkazy změnit nebo nadefinovat vlastní nové příkazy. Pokud si software uživatel neupraví podle vlastní potřeby, je nutné se naučit předvolené příkazy. Tyto jsou dosti strojové. Rozpoznávač má rovněž problémy s jejich rozpoznáváním, což je způsobeno univerzálností mluvčího. Grafický vzhled programu je vidět na obrázku 24.



Obrázek 24: Program VoiceMate Professional

Zdroj: vlastní

8.1.3 Dragon Naturally Speaking

Jedná se o nejvíce světově používaný komerční software pro rozpoznání řečového signálu. Program podporuje několik světových jazyků, podpora češtiny chybí. Umožňuje ovládání počítače pomocí příkazů. Další jeho výhoda spočívá v přepisu řeči do textové podoby. Díky této funkci můžeme lépe a rychleji napsat a odeslat email nebo dopis. Program má několik licencí, které se liší cenou v závislosti na množství implementovaných příkazů. Spolupracuje s mnoha různými aplikacemi, takže umožňuje téměř plnohodnotné ovládání počítače s využitím řečových příkazů. Tento software rovněž nabízí časově omezenou verzi pro vyzkoušení. Její nevýhodou je však množství implementovaných příkazů. Program umožňuje pomocí hlasových příkazů pohybovat myší po obrazovce, vyhledávat data, tisknout soubory a další instrukce. Jeho největší přednosti jsou:

- minimální počáteční nastavení;
- univerzálnost mluvčího, systém není fixován na žádný předem naučený hlas;
- velká databáze příkazů;
- dobré rozpoznání řeči při přepisu do textové podoby.

Nevýhoda spočívá v nutnosti používání předvolených příkazů pro ovládání systému. Uživatel si tak musí zvyknout na dané výrazy. Základní příkazy jsou však velice dobře zapamatovatelné a není potřeba se je učit dlouho. Další nevýhoda spočívá v dlouhé pomlce mezi slovy. Pokud mluvčí udělá mezi slovy delší pomlku, než program očekává, rozpoznávač nereaguje přesně na daný povel a je třeba tento povel vyslovit opakovaně.

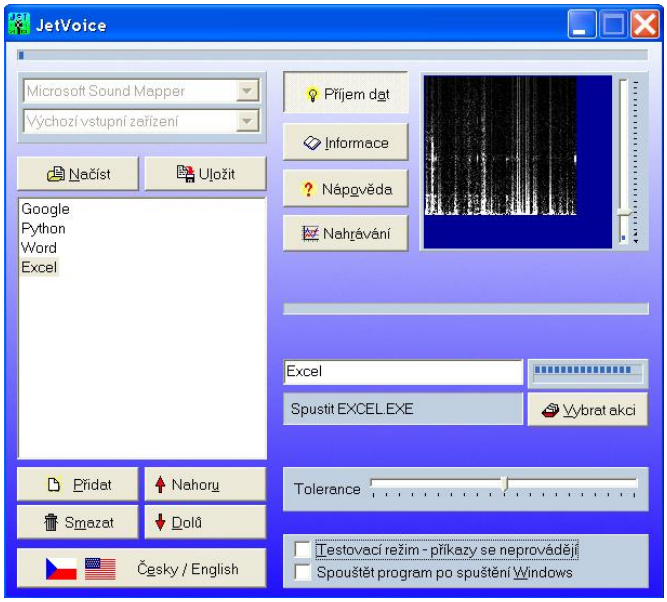
Základní verze programu je prodávána za 99 USD, nejpokročilejší premium verze pak stojí 600 USD. Cena je srovnatelná s cenou českých komerčních programů, které jsou popsány níže.

8.2 Aplikace rozpoznávající češtinu

Do této skupiny patří jen velmi málo programů, neboť čeština nepatří mezi světové jazyky. Jedná se především o komerční programy. Výjimkou je program JetVoice, jež je zdarma. Jeho možnosti jsou však značně omezené. V této kapitole budou popsány nejdůležitější programy využívající rozpoznání češtiny. Bohužel ani jeden z komerčních programů není v žádné omezené verzi volně dostupný, proto tyto programy nemohly být vyzkoušeny.

8.2.1 JetVoice

Jedná se o volně šiřitelný software, který umí spolupracovat s počítačem pomocí hlasových povelů. Po prvním spuštění je nutné program nastavit. Nastavení spočívá ve vyslovení povelu a vybrání akce, která bude po vyslovení tohoto povelu provedena. Záleží tak na uživateli, jestli po vyslovení slova dojde ke spuštění programu či k ovládání myši a podobně. Program je uživatelsky přívětivý a velice dobře reaguje na povely. Výhodou tohoto softwaru je velká úspěšnost rozpoznaných slov, nastavení vlastních příkazů pro ovládání a možnost přepnutí do anglického jazyka. Oproti programu Typle, kde je rovněž možno použít češtinu, má tento software více možností pro nastavení akcí. Naopak jistou nevýhodou je nutnost určitého počátečního nastavení, dále jeho neuniverzálnost při rozpoznání osoby. Program reaguje správně pouze na osobu, která mu nastavila dané řečové pokyny. Na obrázku 25 je vidět uživatelské rozhraní daného programu.



Obrázek 25: Program JetVoice

Zdroj: vlastní

8.2.2 Newton Dictate

Mezi nejpoužívanější software u nás patří Newton Dictate. Používá zejména u soudu a v medicíně. Pro tyto dva obory má velkou řečovou databázi, která umožňuje převod mluveného slova do textové podoby. Je důležité uzpůsobit řeč tomuto programu. Software se může sám zdokonalovat. Přizpůsobuje se mluvčímu a můžeme jej naučit nová slova, která nezná. Software vychází ve třech licencích, přitom cena základní licence basic je 4 900,- Kč (s DPH). Nevýhodou kromě ceny je také vyšší výpočetní náročnost. Program slouží k převodu řeči do textové podoby v textových editorech, nikoliv k ovládání počítače jako takového.[1]

8.2.3 MyVoice

Dalším softwarem, který pracuje s češtinou je MyVoice. Jedná se o software, který umožňuje ovládání počítače pomocí hlasových povelů. Pomocí hlasových povelů můžeme nahradit klávesnici a myš počítače. Dále je možné zadávat nové příkazy pro ovládání programů. Tento software obsahuje také jednoduchý převod řeči do textové podoby, je však omezen velikostí řečové databáze a kvalitou vyslovování jednotlivých slov. Primární použití tohoto softwaru je pro handicapované lidi. Cena licence tohoto software je 6600 Kč s DPH. Program je určen pouze pro ovládání počítače, neumožňuje převod řeči do textové podoby. Je to v podstatě opak výše uváděného software Newton Dictate. [19]

9. Vlastní aplikace využívající rozpoznání řeči

Cílem této diplomové práce bylo vytvořit jednoduché příklady aplikací využívající rozpoznání řečového signálu. Byly vytvořeny dvě jednoduché aplikace, které demonstrují využití rozpoznání řeči v aplikacích. Aplikace byly naprogramovány v programovacím jazyce Python. Výhoda tohoto jazyka je jeho otevřenost a objektový přístup. Použitá verze Pythonu je 2.5.4, neboť má nejlépe propracovaný modul umožňující využívání rozpoznávače řeči. Pro účely rozpoznání řeči byl vybrán rozpoznávač od Microsoftu (Microsoft Speech SDK 5.1).

9.1 Microsoft Speech SDK 5.1

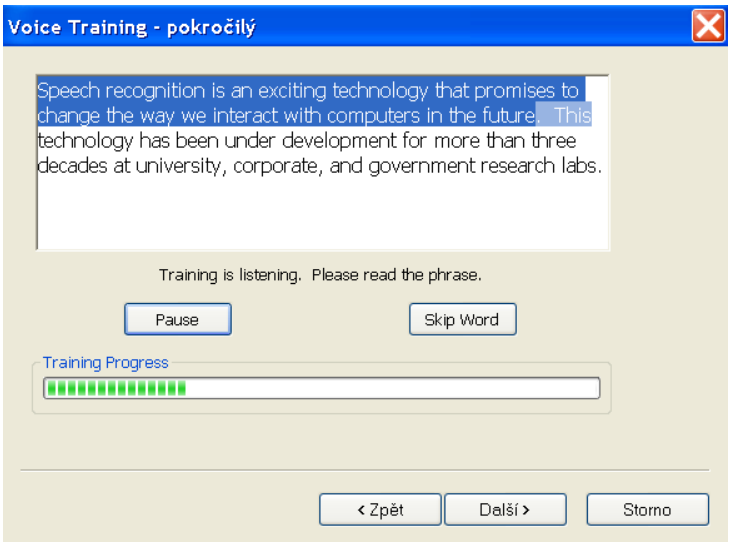
Jedná se o nekomerční software pro majitele operačního systému od firmy Microsoft. Jeho nasazení je vhodné zvláště pro operační systém Windows XP, neboť novější operační systémy od Microsoftu již mají rozpoznávač řeči implementovaný v základu. Po nainstalování tohoto rozpoznávače dostaneme SAPI (Speech Application Programming Interface) řečové rozhraní pro programování aplikací. Jedná se o sbírku procedur, tříd a knihoven, které je možné využít k programování aplikací ovládané hlasem. Tento rozpoznávač umí pracovat s velkým množstvím programovacích jazyků (Python, Visual Basic, C, C++, JavaScript a další).

Rozpoznávač má v sobě integrovány moduly pro rozpoznávání řeči a čtení textu. Moduly pro čtení textu umí číst text pouze v americké angličtině a zjednodušené čínštině. Pro angličtinu jsou k dispozici tři různí mluvčí. Jedná se o řečové profily Sam, Mike a Mary. Z hlediska kvality lze označit jako nejlepší profil Mike. U ostatních profilů je totiž velmi výrazně znát strojové zpracování textu.

Modul pro rozpoznání řeči je o něco bohatší a dokáže ještě navíc rozpoznat řeč v japonštině (což nemohlo být ověřeno z důvodu neznalosti tohoto jazyka). Microsoft Speech SDK pracuje nejčastěji na principu rozpoznání jedné osoby. Pokud bychom chtěli docílit nezávislosti daného rozpoznávače na mluvčím, museli bychom rozpoznávači předložit v učícím režimu různé hlasové profily.

Nejprve se musí rozpoznávač naučit rozeznávat řečové parametry osoby, k tomu slouží režim učení. Učení je velmi důležité, neboť velice záleží na množství naučených slov. Čím více je rozpoznávači během tohoto režimu předloženo slov, tím lépe pak pracuje v aplikacích. Příklad učení rozpoznávače je vidět na obrázku 26. Součástí rozpoznávače jsou také předvolené texty. Čtením těchto textů dochází k lepší analýze hlasového ústrojí

mluvčího. Rozpoznávač pak lépe reaguje na povely či příkazy. Čtení předloženého textu by mělo být monotónní. Nemělo by být ovlivněno emocemi. Záleží také na okolních podmínkách při čtení tohoto textu. Je zapotřebí docílit co nejmenšího šumového napětí. Záleží také na kvalitě použitého mikrofonu. Nejvhodnější je směrový mikrofon. Režim rozpoznání patří k nejdůležitějším režimům, neboť má velký vliv na výslednou kvalitu rozpoznávání řeči. Proto by tento režim neměl být podceňovaný.

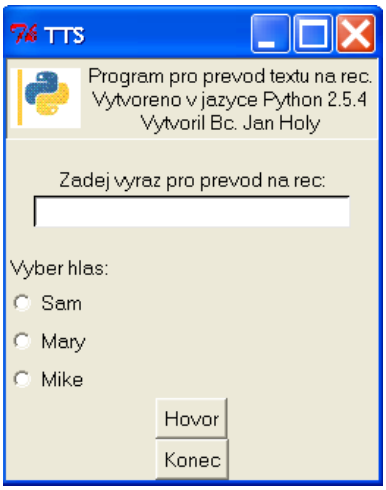


Obrázek 26: Učení rozpoznávače – Microsoft Speech Recognition Training Wizard

Zdroj: vlastní

9.2 Převod textu na řeč (TTS)

Jedná se o aplikaci typu TTS (Text To Speech). Cílem této aplikace je převést text do řečové podoby. Program umí pracovat pouze s angličtinou, proto je potřeba aby dané výrazy byly v tomto jazyce. Bylo vytvořeno jednoduché grafické uživatelské rozhranní, které umožňuje měnit řečový profil mluvčího. Na výběru jsou řečové profily implementované z rozpoznávače. Jedná se o profil Sam, Mike a Mary. Jako implicitní byl nastaven hlasový profil Mike, ale je možné jej uživatelsky změnit. Obrázek 27 znázorňuje grafickou podobu aplikace.



Obrázek 27: Aplikace pro převod textu na řeč

Zdroj: vlastní

Po zadání výrazu a následném spuštění programu přes tlačítko „Hovor“ dochází k vyslovení daného výrazu. Pokud není vybrán žádný hlas mluvčího, bere se implicitní nastavení, což je hlasový profil Mike.

Realizace výběru hlasu zajistí funkce `voice`, která má za úkol vybrat daný hlasový profil podle charakteristiky uživatele. Tlačítka pro výběr hlasu mají přiřazenu číselnou hodnotu, která je předávána jako parametr funkci `voice`. Na jejím základě je pak vybrán hlasový profil, který je návratovou hodnotou dané funkce. Definice funkce `voice` je následující:

```
def voice(rec):
    if (rec == 1):
        hlas = 'MSSam'
    elif (rec == 2):
        hlas = 'MSMary'
    else:
        hlas = 'MSMike'
    return hlas
```

Vlastní program je realizován pomocí funkce `speech`. Tato funkce realizuje samotné vyslovení daného výrazu a nastavení vybraného hlasového profilu. Funkce má dva vstupní parametry. Číselnou hodnotu hlasového profilu a výraz pro vyslovení. Číselná hodnota je předávána funkci `voice`. Výstup z této funkce je ukládán do lokální proměnné `hlas`.

```
hlas = voice(zvuk)
```

Následně je vytvořen nový profil rozpoznávače pomocí tohoto příkazu:

```
tts = pyTTS.Create()
```

Nový profil je pojmenován `tts`. K jeho vytvoření bylo nutné použít modul `pyTTS`, jež není standardní součástí programovacího jazyka Python a je zapotřebí jej importovat. Modul obsahuje funkce sloužící ke komunikaci s rozpoznávačem.

Dále následuje příkaz, který předává rozpoznávači informaci o zvoleném hlasovém profilu včetně jeho nastavení.

```
tts.SetVoiceByName(hlas)
```

Nejdůležitější příkaz celého programu je `Speak`. Tento příkaz realizuje vyslovení požadovaného výrazu. Je definován následovně:

```
tts.Speak(text)
```

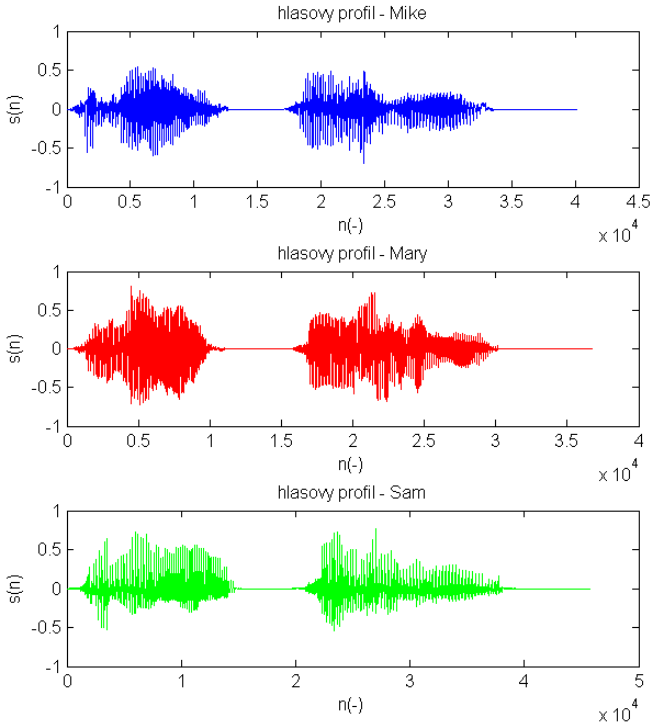
Tímto byly popsány nejdůležitější části zdrojového kódu programu pro realizaci samotné aplikace. Celý zdrojový kód aplikace je umístěn v příloze 1. Program je pro přehlednost rozdělen do dvou souborů. První se stará o grafické uživatelské rozhraní aplikace, ve druhém je komunikace s rozpoznávačem.

9.2.1 Porovnání hlasových profilů mluvčích

Abychom mohli jednotlivé hlasové profily porovnávat mezi sebou je důležité nejprve jejich uložení do souboru. K tomuto účelu využijeme předchozí program pro převod textu na řeč, který doplníme o ukládání výstupu. K tomu slouží příkaz `SpeakToWave`, jež má dva vstupní parametry. Prvním parametrem je jméno výsledného souboru a druhým je ukládaný výraz. Výsledný příkaz vypadá následovně:

```
tts.SpeakToWave('Mike.wav', text)
```

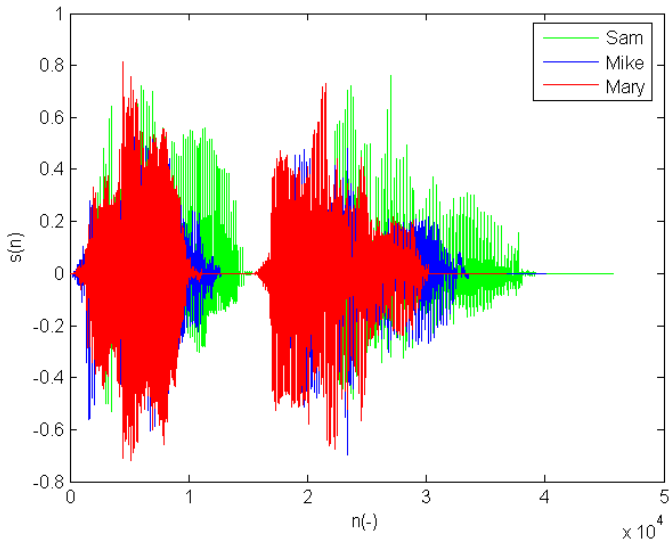
Dále je nutné určit referenční výraz, který necháme vyslovit všemi hlasovými profily. Jelikož rozpoznávač pracuje pouze s angličtinou, byla zvolena následující jednoduchá věta: „Hello, how are you?“. Po vytvoření zvukových souborů byly soubory zobrazeny pomocí programu Matlab. Výsledek je vidět na obrázku 28. Na ose x jsou znázorněny jednotlivé vzorky signálu. Osa y zobrazuje velikost signálu v daném vzorku.



Obrázek 28: Zobrazení hlasových profilů

Zdroj: vlastní

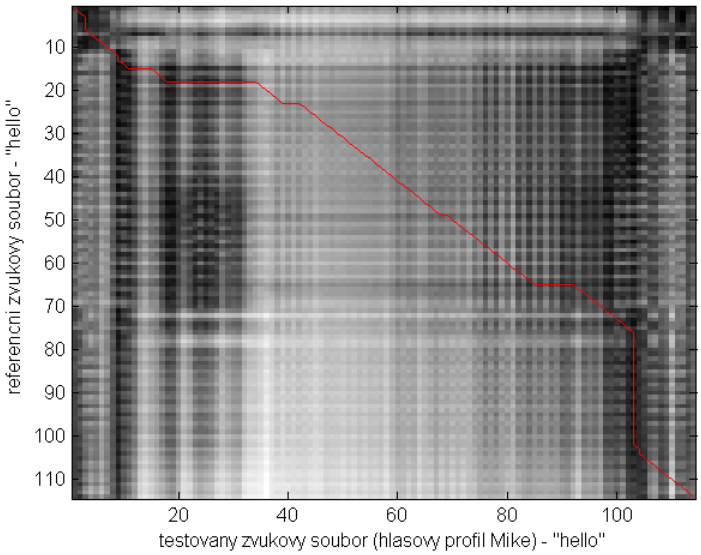
Pro názornost a lepší porovnání výsledků byly výsledné soubory zobrazeny do jednoho grafu. Tím dostávám následující výsledek (obrázek 29). Z grafu je patrné, že nejmenší počet vzorků signálu má hlasový profil Mary. U hlasového profilu Sam, který má naopak největší počet vzorků, dochází k protahování jednotlivých hlásek. Toto můžeme rovněž i slyšet při přehrání zvuku.



Obrázek 29: Porovnání hlasových profilů

Zdroj: vlastní

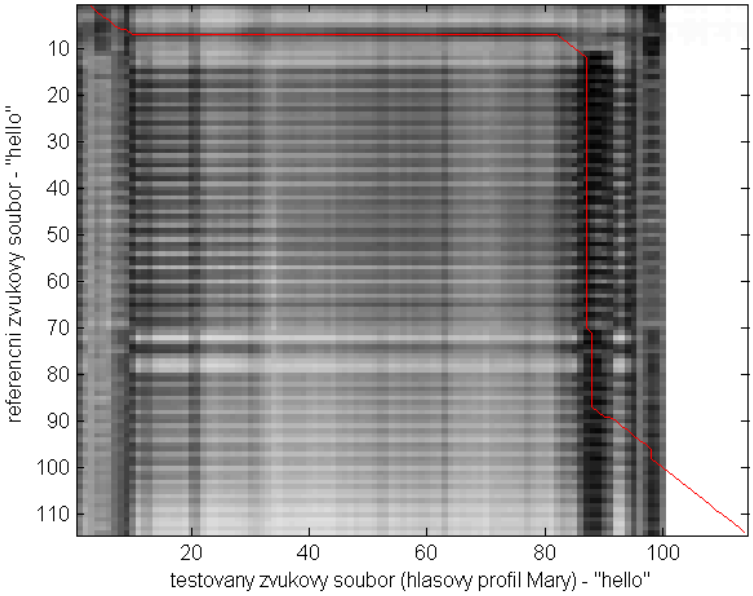
Pro úplnost bylo provedeno srovnávací měření zvukového signálu, který se nejvíce podobá lidskému hlasu. K tomu byla použita metoda DTW, dynamického borcení času. Pomocí vytvořeného programu pro TTS bylo všemi třemi hlasovými profily namluveno slovo „hello“. Následně byla vytvořena referenční nahrávku tohoto slova pomocí lidského hlasu, s využitím programu Audacity. V Matlabu je pak porovnání všech tří hlasových profilů s lidským hlasem. Pro každou možnost je vypočtena optimální cestu ve spektrální oblasti.



Obrázek 30: referenční soubor a hlasový profil Mike

Zdroj: vlastní

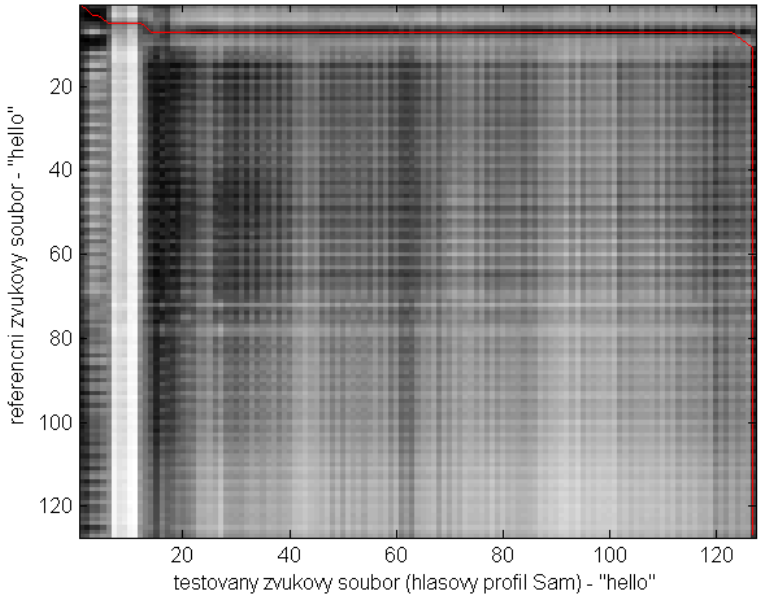
Na obrázku 30 je vidět srovnání DTW pro referenční soubor namluvený lidským hlasem a hlasovým profilem Mike. Velikost jeho optimální cesty je 44,7560 jednotek. Což je podobná hodnota jako pro hlasový profil Sam.



Obrázek 31: referenční soubor a hlasový profil Mary

Zdroj: vlastní

Obrázek 31 znázorňuje metodu měření DTW pro hlasový profil Mary. Zde je velikost optimální cesty 52,2716 jednotek. Jak je vidět, tento hlasový profil má nejdelší optimální cestu. Není proto příliš vhodné tento hlasový profil používat.



Obrázek 32: referenční soubor a hlasový profil Sam

Zdroj: vlastní

Obrázek 32 znázorňuje porovnání referenčního souboru s hlasovým profilem Sam. Velikost optimální cesty je 43,7627 jednotek. Tato hodnota je velmi blízká hodnotě dosažené pro hlasový profil Mike. Z toho vyplývá, že z pohledu DTW vhodné používat hlasový profil Mike nebo Sam.

Pokud však provedeme poslechovou zkoušku daných souborů, jako nejlepší hlasový vychází profil Mike. U profilu Sam je velmi výrazně znát protahování jednotlivých slabik. Zvuk je celkově hodně strojový.

9.3 Převod řeči na text (STT)

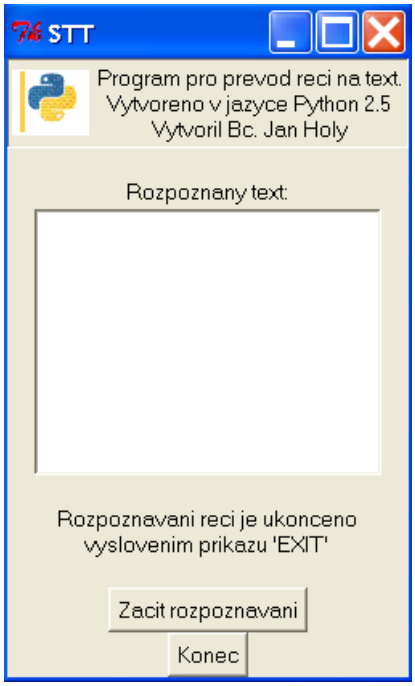
Tato kapitola se zabývá automatickým přepisem řečového výroku do textové podoby. Proces je v literatuře také označován jako ASR (Automatic Speech Recognition). Častěji se pro označení používá pojem STT (Speech To Text). Proto i v této práci bude tento pojem používán pro označení aplikace převodu řečového signálu na text. Důležitým parametrem v tomto oboru je řečový slovník. Nejčastěji se jedná o souvislý řečový signál, kde jednotlivá slova na sebe přirozeně navazují. Pro rozpoznávání je pak důležité učinit pomlky v řečovém signálu. Při nepřetržitém rozpoznávání není možné nalézt hranice mezi jednotlivými řečovými výroky. Podle množství mluvčích, kteří se podílejí na vytvoření řečového slovníku, můžeme systémy rozdělit do dvou skupin:

- závislé na mluvčím;
- nezávislé na mluvčím.

Systém závislý na mluvčím dokáže dobře rozeznat hlasový profil jednoho mluvčího. Je nutné však systém na daný hlasový profil natrénovat v učícím módu. Pokud bychom tomuto systému předložili jiný hlasový profil, než na jaký je tento systém natrénován, došlo by ke zhoršení úspěšnosti rozpoznávání. Naopak systém nezávislý na mluvčím, dokáže dobře rozeznávat jakýkoliv řečový profil. Není nutné již tento systém učit na hlas mluvčího. Nevýhodou je potom však velikost řečového slovníku. Daný slovník musí být vytvořen stovkami či tisícími mluvčími. Takovýto systém si klade velké nároky na výpočetní složitost. [20]

Problematika rozpoznávání řečového signálu je velmi obsáhlá a nelze ji jednoduše popsat. Při rozpoznávání řeči se vždy mohou vyskytnout neočekávané stavy (velký okolní šum, špatná artikulace mluvčího a další). Každý řečový signál můžeme říci pokaždé jinak a s jinou intonací. Záleží pak zejména na nastavení daných rozpoznávačů, jak si s touto problematikou poradí. Největší parazitní složkou v oblasti rozpoznávání je především šum okolí. Není možné vytvořit rozpoznávač, který by dosahoval stejné úspěšnosti rozpoznávání při různé velikosti šumového napětí.

Za účelem zkoumání dané problematiky byla vytvořena jednoduchá aplikaci, která dokáže převést mluvenou řeč do textové podoby. Programovacím jazykem byl opět zvolen jazyk Python ve verzi 2.5.4. Bylo zapotřebí použít jiný modul než v případě aplikace převodu textu na řeč, neboť daný modul uměl pracovat s rozpoznávačem pouze za účelem převodu textu do řečové podoby, nikoli naopak. Proto byl vybrán modul, který má v sobě integrované funkce pro rozeznání slova. Dále bylo vytvořeno jednoduché grafické uživatelské rozhraní pro snadnou orientaci. Grafický vzhled aplikace je zobrazen na obrázku 33.



Obrázek 33: Aplikace pro převod řeči na text

Zdroj: vlastní

Aplikace umí rozeznávat nejen samostatná slova, ale dokáže také rozeznávat celé věty. Nevýhodou aplikace je její jazyková omezenost na angličtinu, neboť bylo nutné použít stejný rozpoznávač jako u předešlé aplikace. Jedná se o Microsoft Speech.

Po stisknutí na tlačítko „Začít rozpoznávání“ dojde k přechodu do režimu naslouchávání. Pokud v tomto okamžiku promluvíme, dojde k rozpoznání slova a k jeho následnému výpisu na obrazovku. Rozpoznávání lze ukončit pomocí hlasového povelu „Exit“ nebo standardním ukončením celé aplikace přes křížek v horní liště. Pokud se rozhodneme o ukončení prostřednictvím příkazu „Exit“, pak po jeho vyslovení budeme upozorněni na ukončení režimu rozpoznávání. Nedochází k výpisu slova „Exit“ na

obrazovku. Dále můžeme pokračovat v rozpoznávání nebo program přes tlačítko konec ukončit. Úkolem programu je zobrazování příslušných hlasových povelů.

Programový cyklus, realizující rozpoznávání řečového signálu je následující:

```
def rec():
    while True:
        rec = speech.input()
        if rec == 'Exit':
            vysledek =
            tkMessageBox.showinfo("Hlaseni","Doslo
            k ukonceni rozpoznavani")
                return
        okno.insert(INSERT, rec)
        okno.insert(INSERT, " ")
        okno.update()
```

Nejprve se načte slovo nebo cyklus slov do proměnné `rec` pomocí příkazu `input()`, který se nachází v modulu `speech`. Následně se provede kontrola, zda vyslovené slovo není „Exit“. Pokud je vysloveným slovem „Exit“ dochází k upozornění na ukončení rozpoznávaní a program se vrací zpět z funkce. Jestliže nedošlo k vyslovení slova „Exit“ je dané slovo vloženo do textového pole. Následně je za dané slovo vložena mezera a textové pole je aktualizovaná. Díky aktualizaci hned vidíme, k jakému slovu rozpoznávač určil daný zvukový soubor. Komunikace s rozpoznávačem je zajištěna pomocí příkazů obsažených v modulu `speech`. Neustálé rozpoznávání zvukových souborů je zajištěno pomocí nekonečné smyčky. Díky této smyčce je neustále monitorován okolní zvuk. Program má jednu nevýhodu. Pokud na vstup přivedeme šumové napětí, rozpoznávač se jej snaží rozeznat a přiřadit šumovému napětí nějaké slovo. To je ovšem nevýhoda většiny programů, které se snaží o převod řeči do textové podoby. Kompletní zdrojový kód programu se nachází v příloze 2.

Tento program je dále rozšiřitelný. Například o příkaz, jež spustí aplikaci na základě hlasového povelu. Příkaz pro spuštění programu vypadá následovně:

```
os.startfile('C:\Windows\system32\mspaint.exe')
```

Je potřeba spustit funkci `startfile`, která se nachází v modulu `os`. Dále už jen stačí zadat cestu, ukazující na umístění daného programu. Pomocí hlasového povelu můžeme spustit libovolný program, který je nainstalovaný na počítači. Celý příkaz včetně referenčního slova pak může vypadat následovně:

```
if rec == 'Word':
    os.startfile('C:\ProgramFiles\MicrosoftOffice\Office12\
    WinWord.exe')
    return
```

Vytvořený program může také sloužit jako jednoduchý překladač. K jednotlivým slovům potřebujeme přiřadit jejich výrazy v cizím jazyce. To nejvhodněji provedeme pomocí datového typu slovník. Ten je definován jako pole skládající se ze dvojic klíč-hodnota. Klíč musí být unikátní a neměnný. Slovník vytvořený v programovacím jazyce Python vypadá následovně:

```
slovnik = {'HELLO':'ahoj', 'LION':'lev', 'APPLE':'jablko'}
```

Slova HELLO, LION a APPLE jsou klíče jednotlivých položek, za nimi pak následuje hodnota těchto položek, což je v našem případě jejich český překlad. Takto lze vytvořit jednoduchý slovník pro překlad slov. Modifikací funkce `rec` z předchozího programu STT dostaneme následující:

```
def rec():
    while True:
        rec = speech.input()
        if rec == 'Exit':
            vysledek = tkMessageBox.showinfo("Hlaseni","Doslo
            k ukonceni rozpoznavani")
            return
        try:
            rec1=slovnik[rec.upper()]
```

```
except KeyError:
      chyba ='!vyraz neni ve slovniku!'
      aktualizace(chyba)
aktualizace(rec1)
rec1=''
```

Zadaná funkce čeká, dokud se na vstupu neobjeví hlasový povel, následně přezkoumá, zda povel nebyl příkaz k ukončení rozpoznávání signálu. Následně je povel vyhledáván ve slovníku mezi klíči. Pokud dojde k nalezení klíče je uložena jeho hodnota do proměnné `rec1`. Následně je proměnná zobrazena na displej a program je připraven k přijmutí dalšího hlasového příkazu.

Metoda `upper()` způsobuje převod malých písmen na velké. Tím je ošetřeno, že nezáleží na tom, v jaké podobě se hlasový povel uloží do proměnné `rec`.

Nevýhodou takto vytvořeného slovníku je jeho omezení, pokud bychom předložili programu slovo, které není klíčem ve slovníku, dojde k chybě. Tato chyba je ošetřena a program na obrazovku vypíše hlášení, že výraz není ve slovníku. Za účelem odstranění duplikace kódu byla vytvořena nová funkce, jež se stará o výpis daného výrazu. Jedná se o následující funkci:

```
def aktualizace(vyraz):
    okno.insert(INSERT, vyraz)
    okno.insert(INSERT, " ")
    okno.update()
    return
```

Nevýhoda tohoto slovníku je v následném poslechu přeloženého výrazu, neboť se využívá rozpoznávač, který umí pracovat pouze s jedním jazykem. Pokud bychom tedy tento rozpoznávač nechali přečíst přeložený výraz, zaslechli bychom změť nesmyslných slov. Pro docílení přečtení i přeloženého slova by bylo nutné použít rozpoznávač, který umí pracovat s více jazyky.

9.3.1 Úspěšnost rozpoznávačů

Je potřeba určit procentuální úspěšnost u rozpoznávačů. Za tímto účelem byly vytvořeny tři hlasové režimy. Jedná se a hlasový režim základní, pokročilý a expert. Režimy se od sebe liší dobou učení. Doba učení jednotlivých režimů je zobrazena v tabulce 2.

Tabulka 2: Doba učení hlasových profilů

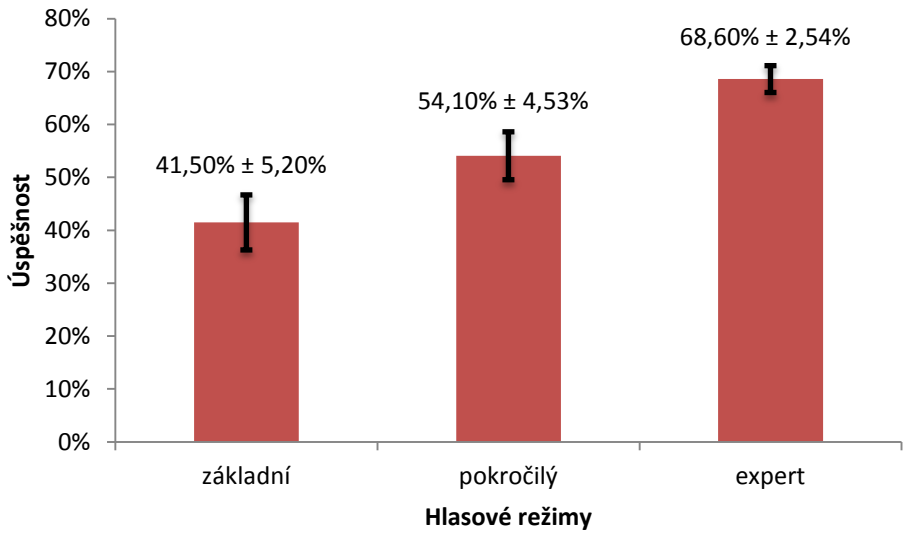
hlasový režim	*doba učení (min)*
Základní	10
Pokročilý	60
Expert	150

Zdroj: vlastní

Základnímu hlasovému režimu byl předložen při učení pouze základní text. Ostatním režimům byl kromě základního textu ještě předložen text rozšiřující. Čtením těchto textů docházelo k učení rozpoznávače. Čtení jednoho textu trvalo přibližně 10 minut. V režimu expert byly rozpoznávači předloženy všechny zkušební texty implementované v rozpoznávači. Měření úspěšnosti probíhalo pomocí aplikace na převod řeči na text napsané v programovacím jazyce Python. Každému režimu byla předkládána stejná slova (příloha 3). Slova byla předkládána prostřednictvím mluvčího, jež dané hlasové profily trénoval na svůj hlas. Celkem bylo toto měření opakováno desetkrát.

Na základě naměřených hodnot byla vypočtena střední hodnota úspěšnosti jednotlivých rozpoznávačů a směrodatná odchylka.

Z níže uvedeného obrázku 34 je patrné, že úspěšnost rozpoznávačů velmi výrazně závisí na adaptaci systému na hlas řečníka (učící režim). Čím více je rozpoznávači během režimu učení předloženo slov, tím lépe pak reaguje na hlasové pokyny mluvčího. Proto jsou systémy založené na rozpoznání řečového signálu tolik náročné na výkon procesoru. Neboť ke správnému určení potřebují velkou databázi řečových signálů. Náročnost lze snížit přesunutím dané databáze do cloudových služeb. Pak je možné k těmto databázím přistupovat i pomocí méně výkonných přístrojů. Vyhledávání odpovídajícího překladu do textové podoby je provedeno na výkonném serveru, k uživateli se dostane pouze nalezený výsledek.



Obrázek 34: Úspěšnost rozpoznání u hlasových režimů

Zdroj: vlastní

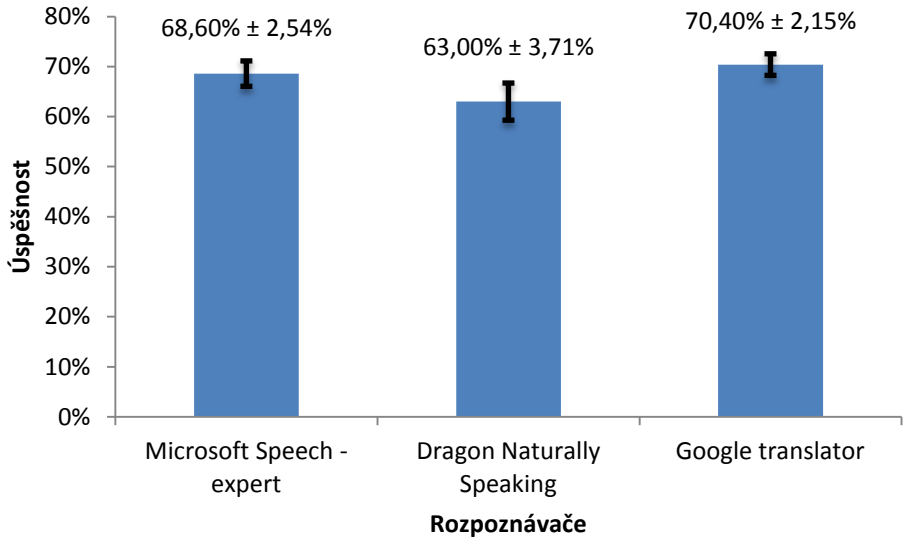
9.3.2 Porovnání rozpoznávačů mezi sebou

Dále byla provedena analýza jednotlivých rozpoznávačů mezi sebou. Účelem této analýzy je prozkoumat jednotlivé rozpoznávače mezi sebou. Byl vybrán rozpoznávač od Microsoftu s hlasovým režimem expert (ten samý jako v předchozí analýze). Dále byl vybrán software Dragon Naturally Speaking, jež má v sobě svůj vlastní rozpoznávač. U tohoto softwaru byl použit mód pro diktování. Rozpoznávač byl vybrán jako zástupce komerčního softwaru. Posledním rozpoznávačem, který se účastnil analýzy, byl online rozpoznávač od firmy Google. Jedná se o Google Translator, jde sice spíše o překladač než o typický rozpoznávač. Google Translator však umí vkládání textu pomocí hlasu. Tato funkce umožňuje využít tuto stránku jako rozpoznávač řečového signálu. Tento druh rozpoznávače je zástupcem cloudové řešení rozpoznávání řeči. Všem překladačům byl předložen stejný zkušební soubor slov jako v předchozí analýze.

Výsledek této analýzy (obrázek 35) je poněkud překvapivý, neboť úspěšnost daných rozpoznávačů se od sebe příliš nelišila. Nejhůře dopadl komerční rozpoznávač Dragon Naturally Speaking, je však nutné podotknout, že se jednalo jen o demoverzi tohoto programu, jenž má značné omezený bázový slovník naučených slov. Lepších parametrů v této analýze dosahoval rozpoznávač od společnosti Google. Naopak nejvíce překvapil rozpoznávač od Microsoftu. Jedná se o nekomerční rozpoznávač, který je dnes

již běžnou součástí nových operačních systému od firmy Microsoft. Pokud danému softwaru předložíme velké množství slov v učícím režimu, můžeme jej kvalitativně srovnat s komerčními programy.

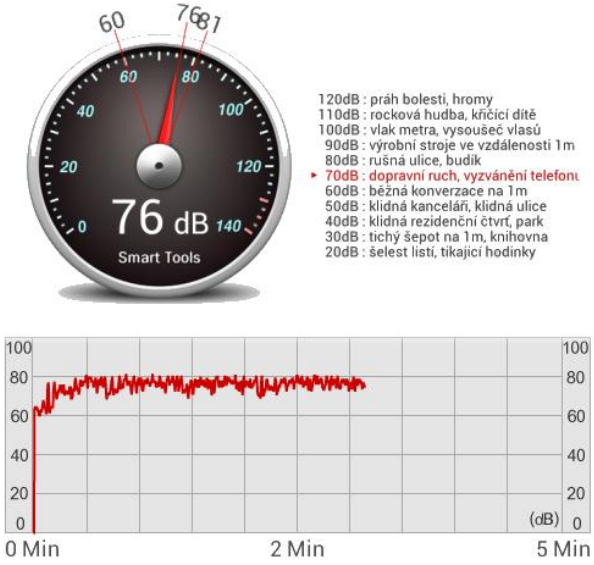
Toto srovnání je však možné pouze pro jednotlivé osoby pracující s daným softwarem. Pokud bychom tuto analýzu provedli s jiným mluvčím, než s jakým byl rozpoznávač naučen, lze předpokládat, že bychom dosáhli mnohem horších výsledků. Výsledky rozpoznávačů od společnosti Google a software Dragon Naturally Speaking by však dosahovaly pořád stejných výsledků. Dále je potřeba zmínit, že Google translator je primárně navržen jako překladač. Tudíž po zapojení slov do příslušných vět, dochází k mnohem lepšímu rozpoznání než v případě samostatných slov. Je také třeba brát v potaz kvalitu použitého mikrofonu. Při všech testech byl použit mikrofon Trust Starzz. Dané porovnání bylo provedeno při minimálním šumovém napětí.



Obrázek 35: Úspěšnost rozpoznávačů při rozpoznávání samostatných slov

Zdroj: vlastní

Další měření je zaměřeno na porovnání rozpoznávačů při nenulové hladině šumu. Jak je patrné z obrázku 36, velikost šumového napětí byla v rozmezí mezi 60 a 80 dB. Okolní hluk byl měřen pomocí mobilní aplikace, umožňující zobrazit graf hluku v čase. Šumový signál byl měřen u mikrofonu. Byla použita stejná testovaná slova jako u předchozího měření. Kromě velikosti šumového napětí se test jinak nijak neměnil od předchozího.



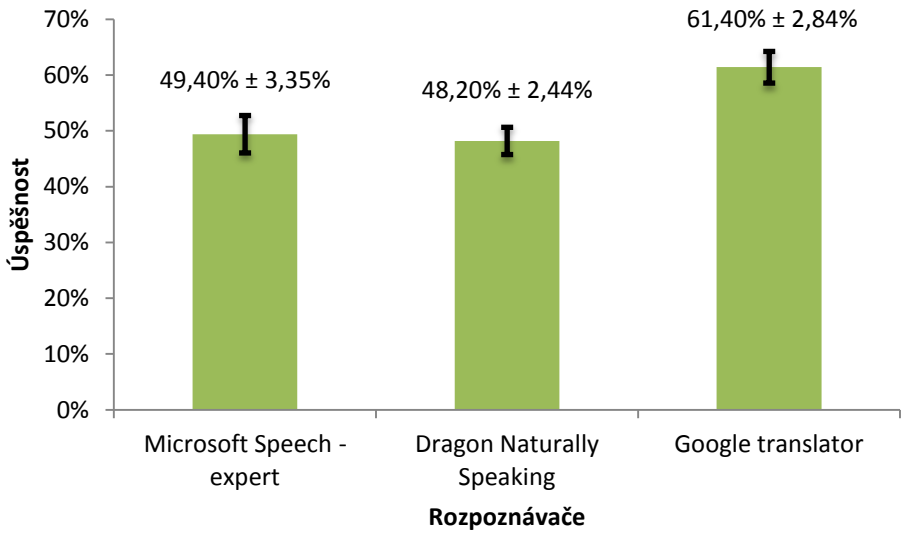
Obrázek 36: úroveň hluku při rozpoznávání

Zdroj: vlastní

Z obrázku 37 je možné vypozorovat, že nejmenší vliv na okolní hluk má rozpoznávač od společnosti Google. Hodnota úspěšnosti rozpoznávání se příliš nelišila od úspěšnosti rozpoznání při nízké hladině hluku. Rozdíl byl pouze o 9 %. To je velmi dobrý výsledek v porovnání s ostatními. Je třeba si uvědomit, že rozpoznávač od Googlu není primárně určen k tomuto procesu. Jeho výsledek byl očekáván, neboť tento rozpoznávač pracuje s velkým slovníkem řečových souborů. Jedná se o cloudové řešení, kde příkaz je vyhodnocován ne v počítači, ale na vzdáleném serveru. Tudíž pro svoji práci nepotřebuje tak veliký výpočetní výkon na daném zařízení. Veškeré rozpoznávání je realizováno na serveru, výsledek je pak vracen na pracovní stanici. Protože tento rozpoznávač využívá mnoho lidí je jeho řečový slovník velmi obsáhlý a zvyklý na různá šumná prostředí. Díky tomu má velmi robustní slovník, který dokáže snáze odfiltrovat šum od daného příkazu. Na druhém místě tedy skončil rozpoznávač Dragon Naturally Speaking, kde byl využit jeho diktovací mód. Tento mód slouží k přepisu hlasových povelů do textové podoby. Přepis byl realizován do textového souboru a následně byla vyhodnocena shodnost s referenčním textem. Nejhorších úspěšnosti dosáhl rozpoznávač od Microsoftu. Pro lepší úspěšnost rozpoznávání by bylo potřeba namluvit učící režim při dané hladině hluku.

Dále je třeba zvážit vliv Lombardova efektu. Jedná se o vliv hlučného prostředí na produkci hlasového signálu. Můžeme si to představit na následujícím příkladu. Pokud je člověk v hlučném prostředí (velké šumové napětí) je pro něj přirozené zvýšit intenzitu

hlasu za účelem vyšší srozumitelnosti. Tento jev je z pohledu rozpoznání hlasového signálu velmi škodlivý, neboť dochází k změně základního řečového kmitočtu f_0. Lombardův jev je závislý na mluvčím a na úrovni hluku. Může se tak stát, že Lombardův efekt má mnohem větší rušivý potenciál než okolní šum. Lombardův efekt je pro člověka přirozený, automaticky jej používáme, aniž bychom si to uvědomovali. Proto mohlo dojít k mírnému zkreslení naměřených výsledků vlivem toto jevu.

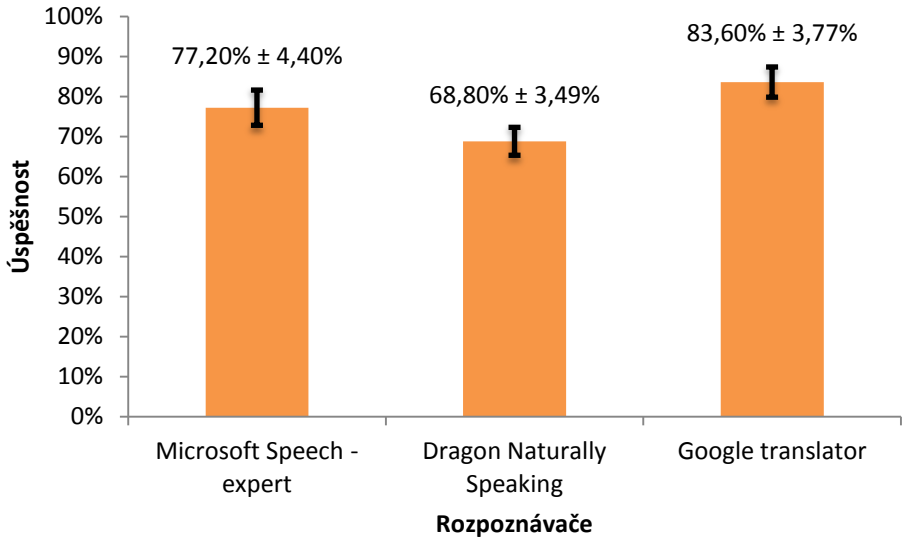


Obrázek 37: Úspěšnost rozpoznávačů při rozpoznávání samostatných slov

(hladina šumového napětí 60 - 80 dB)

Zdroj: vlastní

Následuje měření za účelem zjištění úspěšnosti rozpoznávačů pro věty nebo slovní spojení. Vlivem tohoto měření bychom měli zjistit závislost jednotlivých systémů na přísunu většího množství hlasových povelů na vstup. Doposud byla testována pouze samostatná slova. Nyní jsou rozpoznávačům předkládány věty, či slovní spojení. U některých rozpoznávačů, lze předkládat zlepšení úspěšnosti rozpoznávání. Proto byl vytvořen testovací soubor slovních spojení a vět, které byly postupně předkládány jednotlivým rozpoznávačům. Soubor obsahoval 25 slovních spojení, či vět, které byly každému systému dvakrát předloženy (příloha 4). Na obrázku 38 jsou vidět naměřené výsledky. Nejlepší úspěšnosti dosáhl rozpoznávač od firmy Google, jeho přednosti již byly probrány výše. Jako další skončil rozpoznávač od Microsoftu v režimu expert. Na třetím místě je rozpoznávače Dragon Naturally Speaking, který měl úspěšnost pouze 68,80 %.

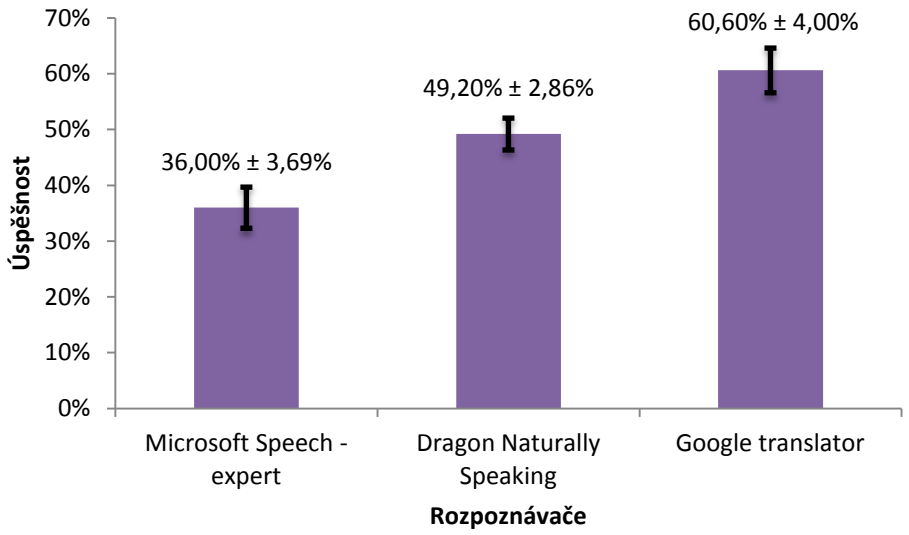


Obrázek 38: Úspěšnost rozpoznávačů při rozpoznávání slovních spojení a vět

Zdroj: vlastní

V tomto testu byly patrné jednotlivé principy fungování rozpoznávačů. Zatímco rozpoznávač od firmy Google se snažil rozpoznat celou větu, Dragon Naturally Speaking rozeznával jednotlivá slova. V důsledku toho, pak docházelo k záměně jednotlivých slov. Záměna mohla být způsobena špatnou artikulací mluvčího, nebo malou robustností řečové slovníku. Nejčastější příčina u Dragon Naturally Speaking byla právě robustnost řečového slovníku. Pokud jsme daný výraz opakovali několikrát, docházelo ke stejnému překladu do textové podoby. Lze předpokládat, že daný systém toto slovo neznal. Na vině také mohou být jazykové znalosti překladatele.

Jazykové schopnosti překladatele a univerzálnost jednotlivých systémů. Jsou předmětem posledního testu. Tento test je zaměřen na schopnosti rozpoznávačů správně určit hlasové povely od jiného mluvčího, než který namluvil učící režim. Lze předpokládat, že nejlepších výsledků dosáhne rozpoznávač od Google, neboť má velmi obsáhlý a robustní řečový slovník. Měření je provedeno pří nízké hladině šumu (do 20 dB). Testovaným souborem jsou opět samostatná slova (příloha 3). Výsledek je zobrazen na obrázku 39.



Obrázek 39: Úspěšnost rozpoznávačů pro jiné hlasové profily

Zdroj: vlastní

Nejhorší úspěšnosti dosáhl rozpoznávač od Microsoftu, což je způsobeno jeho malou robustností vůči jiným hlasovým profilům. Pokud bychom chtěli zvýšit robustnost tohoto rozpoznávače, bylo by nutné mu provést učící režim pro několik různých hlasových profilů. Čím více by mu bylo předloženo hlasových profilů, tím by byla větší jeho robustnost. Dále se potvrdil předpoklad, že nejlepší úspěšnosti dosáhne rozpoznávač od Googlu, jež obsahuje robustní řečový slovník.

V příloze 5 jsou uvedeny tabulky znázorňující úspěšnost pro jednotlivá měření.

10. Závěr

Tato diplomová práce se zabývá zpracováním řečového signálu. Existuje velké množství rozpoznávačů, které pracují nejčastěji na principu skrytých Markovových modelů.

Rozpoznávače rozdělujeme podle použití na univerzální a jednoúčelové. Univerzální rozpoznávače obsahují množství hlasových profilů. K tomu je potřeba velká a obsáhlá řečová databáze, což je jejich nevýhoda. Takto velká řečová databáze potřebuje rovněž vysoký výpočetní výkon, který umožňuje její zpracování. Z tohoto důvodu se tyto rozpoznávače přemisťují na výkonné servery. Server umožňuje dosáhnout většího výpočetního výkonu než běžné stolní počítače či jiná zařízení, která uživatelé používají. Příkladem takového rozpoznávače je Google Translator. Ten je schopen rozpoznat univerzální řečový signál. Rozpoznávač od Googlu je umístěn na serveru, jež je připojen k internetu. Takovýto server musí mít velký výpočetní výkon. Ke zkvalitňování samotného rozpoznávání přispívají samotní uživatelé tím, že jej používají. Při používání tohoto rozpoznávače dochází k nárůstu řečové databáze a tím se zlepšuje úspěšnost rozpoznávání. Pokud se zaměříme na problém rozpoznávání z pohledu výrobce takového softwaru, je nesmírně důležité mít kvalitní a robustní databázi a velký výpočetní výkon hardwaru. Z pohledu uživatele nejsou tyto systému závislé na jejich použitém hardwaru. Uživatel řeší zejména pouze kvalitu a rychlost připojení k internetu.

Druhou skupinou jsou jednoúčelové rozpoznávače. Tyto rozpoznávače mohou také obsahovat velké množství hlasových profilů, jejich řečový slovník je však omezen pouze na určitý obor. Zástupcem tohoto software je Newton Dictate. Výrobce implementuje daný slovník přímo do samotného rozpoznávače. Z pohledu výrobce není nutné provozovat server, který by musel mít vysoký výpočetní výkon. Požadavky na vyšší výpočetní výkon se přesouvají na stranu uživatele. Uživatel k provozu systému potřebuje odpovídající hardware, což představuje určitou nevýhodu oproti serverovému řešení s tenkým klientem. Je potřeba, aby uživatel přizpůsobit styl diktování danému softwaru. Potom dochází ke zvýšení efektivnosti práce uživatele, zejména pokud se jedná o dlouhé texty. V praxi se uvádí úspěšnost přepisovatelky v rozmezí 85 – 90 %. Software Newton Dictate, při správné artikulaci mluvčího, dosahuje úspěšnosti až 90 – 95 %. [21] Tyto systémy jsou používány zejména v soudnictví a zdravotnictví. Jejich výhoda spočívá ve vysoké úspěšnosti rozpoznání pro určité obory, které mají obsáhlou řečovou databázi. Pokud bychom se snažili dosáhnout stejných parametrů u univerzálních rozpoznávačů,

bylo by toto řešení velmi obtížné. Z pohledu uživatele by byl nutný vysoký výpočetní výkon. Výrobce by pak musel vyrobit velmi obsáhlou a robustní řečovou databázi, což by bylo z časového hlediska velice náročné. Toto je i při dnešních hardwarových možnostech osobních počítačů stále nereálné. Všechny univerzální rozpoznávače řeči pracují s omezenou řečovou databází, aby ji bylo možné používat na většině standardních osobních počítačů.

Dále je potřeba rovněž zmínit náročnost při rozpoznávání jednotlivých jazyků. Metody se různě mění vzhledem k použitému jazyku. Nejjednodušší je angličtina, neboť má pevnou stavbu věty. Slovní druhy v oznamovací větě mají přesně danou pozici. Tím angličtina vyniká oproti ostatním jazykům, zejména od češtiny, kde je možné větu říci několika způsoby. Stavba věty není pevně daná.

Posledním neméně důležitým ukazatelem je finanční hledisko. Cena komerčních programů je v dnešní době poměrně vysoká, pohybuje se v řádu desítek tisíc korun. Většina těchto programů je orientována pouze na určité obory, z důvodu dosažení očekávané úspěšnosti. Příkladem je opět software Newton Dictate. Úspěšnost tohoto programu dokazuje, že po roce testování českými soudy, bylo zakoupeno 1600 licencí, zejména pro soudnictví a státní správu. [21] Oproti tomu nekomerční programy mají omezené vlastnosti a je přizpůsobit pro daného uživatele. Nedisponují robustním řečovým slovníkem, který by nebyl závislý na mluvčím. Pokud uživatel používá operační systém Windows, může zdarma využít i jeho integrovaný modul Microsoft Speech, který umožňuje rozpoznání řeči. Tento modul dosahuje poměrně dobrých parametrů, proto byl vybrán k vytvoření aplikací pro rozpoznání řeči.

Cílem této práce bylo vytvoření jednoduchých programů, které se budou zabývat danou problematikou. První program slouží k převodu textu na řeč. V programu je možné si vybrat ze třech různých mluvčích, kteří daný text „přečtou“. Vytvořená aplikace úspěšně reagovala na zadané textové povely. Nevýhody této aplikace jsou:

- nutnost zadávání textových povelů v angličtině;
- „strojový“ výstupní hlas.

Obě výše zmíněné nevýhody souvisí se zvoleným rozpoznávačem. Následně byl proveden test, který měl za úkol vyhodnotit nejpřirozenější hlasový profil. Nejlepších hlasových parametrů dosáhl profil Mike. Význam TTS aplikací je především u zrakově postižených lidí. Elektronická zařízení již umožňují přečíst zprávu či email.

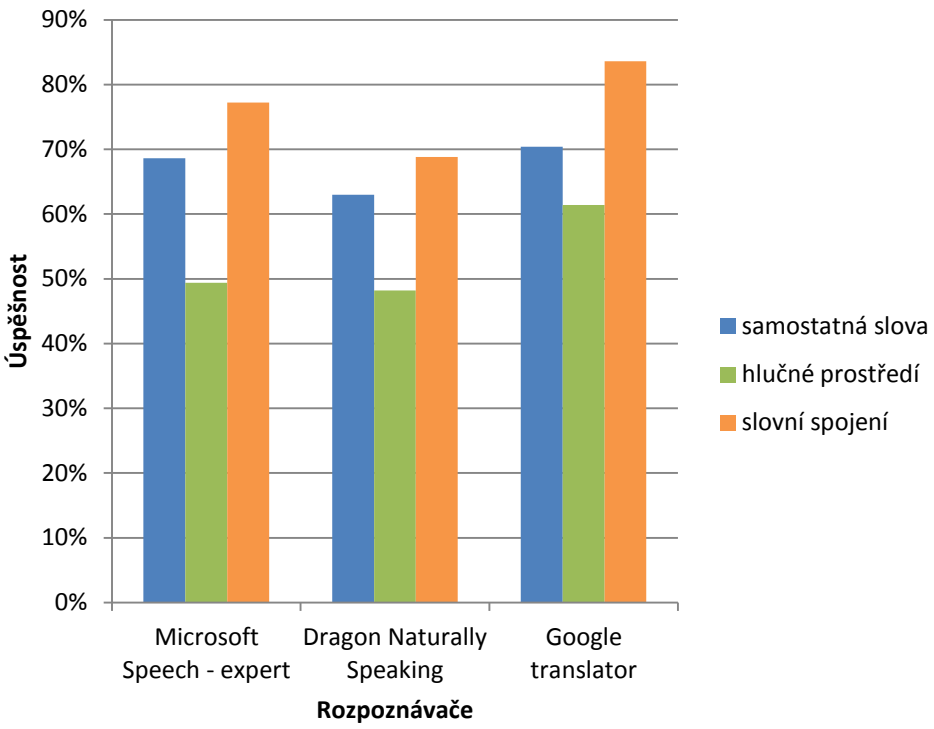
Druhý program, který vznikl na základě této práce, je aplikace pro převod řeči do textové podoby. Jedná se o opačné řešení, kdy je aplikaci předložen řečový signál a ona jej přepíše do textové podoby. Rozpoznávání je mnohem složitější. Vytvořená aplikace převádí hlasové povely do textové podoby. Je možno úpravy této aplikace tak, aby ovládala počítač nebo fungovala jako překladový slovník.

Nevýhodou tohoto softwaru jsou následující:

- závislost na mluvčím;
- nutnost věnovat se učícímu režimu;
- přizpůsobit mluvu softwaru;
- malá robustnost vůči okolnímu šumu.

Dále byly provedeny testy porovnávající úspěšnost několika rozpoznávačů za různých podmínek. Do testu byly zařazeny rozpoznávače, jež reprezentují různé skupiny. Rozpoznávač od Googlu disponuje velkým univerzálním slovníkem, není vázán na výpočetní výkon uživatele, neboť tento jej používá v zejména cloudu. Další rozpoznávač Dragon Naturally Speaking patří ve světě k nejpoužívanějším programům pro rozpoznání řeči. Disponuje omezeným univerzálním slovníkem a je nutný vysoký výpočetní výkon na straně uživatele. Poslední je mnou vyrobená aplikace, která je postavena na rozpoznávači do společnosti Microsoft. Výsledky jednotlivých měření jsou znázorněny na obrázku 40.

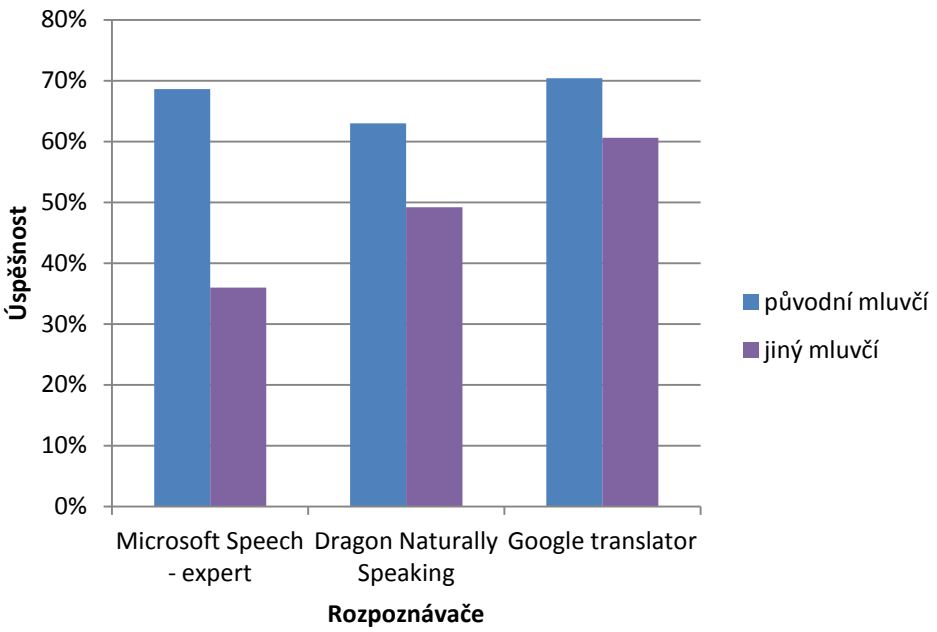
Zde se projevuje obsáhlost řečového slovníku. Největší řečový slovník má rozpoznávač od společnosti Google. Neboť jak bylo výše zmíněno, na rozšiřování slovníku se podílejí samotní uživatelé. Dále je třeba zmínit, že software Dragon Naturally Speaking, má jen omezený slovník. Proto je toto porovnání poněkud zavádějící. Pokud bychom měli u každého rozpoznávače stejně velký slovník, záleželo by pouze na filtrování šumového napětí jednotlivých rozpoznávačů. Z hlediska úspěšnosti je vhodné předkládat rozpoznávačům slovní spojení či celé věty.



Obrázek 40: Celková úspěšnost rozpoznávačů

Zdroj: vlastní

Na obrázku 41 je zvlášť zobrazena závislost na robustnosti rozpoznávače. Zde docházelo k rozpoznávání jednotlivých slov jiným mluvčím. Následně byla úspěšnost porovnána s úspěšnosti mluvčího, jež namluvil učící režim rozpoznávače. Nejmenší robustnosti dosáhla mnou vytvořená aplikace. Je patrné, že velice závisí na velikosti a robustnosti řečového slovníku. U ostatních rozpoznávačů můžeme rozdíly v úspěšnosti zanedbat, neboť se hodnoty příliš neliší. Odchylka je z velké části způsobena špatnou artikulací mluvčího.



Obrázek 41: Robustnost rozpoznávače vůči jinému mluvčímu

Zdroj: vlastní

Pokud tedy shrneme poznatky této práce do několika bodů, pak úspěšnost správného převodu záleží na následujících faktorech:

- velikosti a robustnosti řečového slovníku;
- výpočetním výkonu;
- správné artikulaci mluvčího;
- velikosti okolního šumu a jeho filtraci;
- jazykových schopnostech přizpůsobení požadavku systému.

Pokud tedy nepřijde jiná technologie, která umožní ještě snadnější ovládání (ovládání zařízení pomocí čipu v mozku). Lze předpokládat růst významu této technologie.

Seznam odborné literatury

[1] NEWTON TECHNOLOGIES A.S. Diktovani.cz [online]. 2003 [cit. 2013-04-03]. Dostupné z: www.diktovani.cz

[2] Hlasem řízené vychystávání ve skladech. Logistika.ihned.cz [online]. 2005 [cit. 2012-12-11]. Dostupné z: http://logistika.ihned.cz/c1-15789340

[3] ZRE: Zpracování řečových signálů. [online]. [cit. 2012-12-11]. Dostupné z: http://www.fit.vutbr.cz/study/courses/ZRE/public/

[4] DIGITAL FILTERS. VASEGHI, Saeed. Communication & Multimedia Signal Processing Group: Brunel University London [online]. [cit. 2013-02-14]. Dostupné z: http://dea.brunel.ac.uk/cmsp/Home_Saeed_Vaseghi/Chapter05-DigitalFilters.pdf

[5] DAVÍDEK, Vratislav - SOVKA, Pavel. Číslicové zpracování signálů. Vyd. 1. Praha: ČVUT, 1996. 235 s. ISBN 8001015300.

[6] PSUTKA, Josef - MÜLLER, Luděk - MATOUŠEK, Jindřich - RADOVÁ, Vlasta. Mluvíme s počítačem česky. Praha: Academia, 2006. 752 s. ISBN 80-200-1309-1.

[7] UHLÍŘ, Jan - SOVKA, Pavel - ČMEJLA, Roman. Úvod do číslicového zpracování signálů. Praha: Vydavatelství ČVUT, 2003. 151 s. ISBN 80-01-02799-6.

[8] Spectral audio signal processing. SMITH, Julius O. Center for Computer Research in Music and Acoustics [online]. [cit. 2012-12-11]. Dostupné z: https://ccrma.stanford.edu/~jos/sasp/

[9] RABINER, Lawrence. Digital Speech Processing Course. Eletrical and computer engineering: University of California Santa Barbara[online]. [cit. 2012-12-15]. Dostupné z: http://www.ece.ucsb.edu/Faculty/Rabiner/ece259/

[10] Segmentace řečového signálu. Brno, 2008. Dostupné z: http://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=8160. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Ing. Petr Sysel.

[11] MAUTNER, Pavel. Diskrétní Fourierova transformace. 23 s. Dostupné z: http://www.kiv.zcu.cz/~mautner/Azs/Azs4.pdf

[12] KREIDL, Marcel - ŠMÍD, Radislav. Technická diagnostika: Senzory - Metody - Analýza signálu. Vyd. 1. Praha: BEN, 2006. 408 s. ISBN 80-7300-158-6.

[13] MÜLLER, Meinard. Dynamic Time Warping. [online]. s. 17 [cit. 2013-02-18]. Dostupné z: http://www.springer.com/cda/content/document/cda_downloaddocument/9783540740476 - c1.pdf?SGWID=0-0-45-452103-p173751818

[14] SMÉKAL, Zdeněk - ATASSI, Hicham - STEJSKAL, Vojtěch - MEKYSKA, Jiří. Soubor programů pro práci se skrytými Markovovými modely (HTK).Elektrorevue. 5. 3. 2009, č. 11, s. 42. ISSN 1213-1539. Dostupné z: http://elektrorevue.cz/file.php?id=200000288-a19dba297a

[15] WILPON, Jay G - RABINER, Lawrence - LEE, Chin-Hui - GOLDMAN. Automatic Recognition of Keywords in Unconstrained Speech Using Hidden Markov Models. 1990. 9 s. Dostupné z: http://www.ece.ucsb.edu/faculty/Rabiner/ece259/Reprints/automatic%20recognition%20of%20keywords.pdf

[16] DYMARSKI, Przemyslaw. Hidden Markov Model: Theory and Applications. Croatia: InTech, 2011. 314 s. ISBN 978-953-307-208-1. Dostupné z: http://www.intechopen.com/books/hidden-markov-models-theory-and-applications

[17] MCLOUGHLIN, Ian. Applied speech and audio processing: with Matlab examples. 1st pub. Cambridge: Cambridge University Press, 2009. 206 s. ISBN 978-0-521-51954-0.

[18] MIHELIČ, France - ŽIBERT, Janez. Speech recognition: technologies and applications. Vienna: I-Tech Education and Publishing, 2008. 550 s. ISBN 978-953-7619-299. Dostupné z: http://www.intechopen.com/books/speech_recognition

[19] Fugasoft: MyVoice. [online]. [cit. 2013-02-24]. Dostupné z: http://www.fugasoft.cz/index.php?cont=myvoice

[20] RAMAKRISHNAN S. Modern Speech Recognition: Approaches with Case Studies. Croatia: InTech, 2012. 326 s. ISBN 978-953-51-0831-3. Dostupné z: http://www.intechopen.com/books/modern-speech-recognition-approaches-with-case-studies

[21] MAZAL, Jan. Počítačový převod mluvené češtiny do textové podoby v praxi. [online]. 2010 [cit. 2013-03-17]. Dostupné z: http://pcworld.cz/novinky/doba-kdy-bude-automaticky-prepisovana-lidska-rec-se-diky-vykonnejsim-procesorum-priblizila-9015

Seznam zkratek a symbolů

ASR.....Automatic Speech Recognition

DFT .....Discrete Fourier Transformation

DTW.....Dynamic Time Warping

FIR.....Finite Impulse Response

HMM.....Hidden Markov Model

IIR .....Infinite Impulse Response

LPC .....Linear Predictive Coding

PCM .....Pulse - Code Modulation

SAPI .....Speech Application Programming Interface

STFT .....Short - Time Fourier Transform

STT.....Speech To Text

TTS.....Text To Speech

---

f_0 *(-)* .....základní frekvence

x (-).....vstupy

y (-).....výstupy

a_k, b_k *(-)*.....konstanty

N (-) .....celkový počet vzorků

s(n) (-).....velikost signálu n-tého vzorku

s_s *(-)* .....ustředněný signál

T (s) .....perioda vzorkování

f_v *(Hz)* .....frekvence vzorkování

f_m *(Hz)*.....maximální frekvence

N_{ram} *(-)*.....počet rámců

l_{ram} *(-)*.....velikost rámce

s_{ram} *(-)* .....velikost rámce bez překrytí

E (-) .....střední krátkodobá energie

e(n) (-) .....buzení

h(n) (-) .....modifikační ústrojí

$c_e(n)$ *(-)* .....kepstrální koeficient pro buzení

$c_h(n)$ *(-)* .....kepstrální koeficient pro modifikační ústrojí

f_{mel} *(mel)*.....melovská frekvence

P (-) .....řád systému

o (-).....vektory pozorování

Seznam obrázků

Seznam tabulek

Seznam příloh

Příloha 1: Zdrojový kód aplikace překladu textu na řeč (TTS).

Soubor **start.py**

```
# Aplikace pro prevod textu na rec
# vytvoril: Bc. Jan Holy
from Tkinter import *
import sys
import speech
def sel():
        '''
        funkce, ktera predava vstupni data funkci speech
        (funkce speech se nachazi v souboru speech.py)
        '''
        hodnota = int(var.get())
        vyraz = str(t.get(1.0,END))
        speech.speech(hodnota,vyraz)


hlavniOkno = Tk()
hlavniOkno.title("TTS")
ramec = Frame(hlavniOkno, relief=SUNKEN, border=1)
image = PhotoImage(file='python.gif')
label4 = Label(ramec)
label3 = Label(ramec, text = 'Program pro prevod textu na rec.\nVytvoreno v jazyce
Python 2.5.4\nVytvoril Bc. Jan Holy')
label4['image'] = image
label4.pack(side=LEFT)
label3.pack(side=LEFT)
ramec.pack(side=TOP, expand=True)
popisek = Label(hlavniOkno, text='\nZadej vyraz pro prevod na rec:')
popisek.pack()
t = Text(hlavniOkno, width=30, height=1, bg='white')
t.pack()
label = Label(hlavniOkno, text = '\nVyber hlas:')
label.pack(anchor = W)
```

```
var = IntVar()
R1 = Radiobutton(hlavniOkno, text="Sam", variable=var, value=1)
R1.pack(anchor = W)
R2 = Radiobutton(hlavniOkno, text="Mary", variable=var, value=2)
R2.pack(anchor = W)
R3 = Radiobutton(hlavniOkno, text="Mike", variable=var, value=3)
R3.pack(anchor = W)
tlacitko1 = Button(hlavniOkno, text = 'Hovor', command = sel)
tlacitko1.pack()
tlacitko = Button(hlavniOkno, text = 'Konec', command = hlavniOkno.destroy)
tlacitko.pack()
hlavniOkno.mainloop()
```

Soubor **speech.py**

```
import pyTTS
def voice(rec):
        '''
        funkce pro vyber hlasoveho profilu mluvciho
        '''
        if (rec == 1):
                        hlas = 'MSSam'
         elif (rec == 2):
                hlas = 'MSMary'
         else:
                hlas = 'MSMike'
         return hlas

def speech(zvuk, text):
        '''
        funkce zajistujici vysloveni vyrazu
        '''
        hlas = voice(zvuk)
        tts = pyTTS.Create()
        tts.SetVoiceByName(hlas)
        tts.Speak(text)
        return
```

Příloha 2: Zdrojový kód aplikace překladu řečového signálu do textové podoby (STT).

Soubor **stt.py**

```
from Tkinter import *
import tkMessageBox
import speech
def rec():
        '''
        funkce zajistujici prevod reci do textove podoby,
        je-li vyraz = 'Exit' pak dochazi k ukonceni prevodu
        '''
        while True:
                rec = speech.input()
                        if rec == 'Exit':
                                vysledek = tkMessageBox.showinfo( "Hlaseni","Doslo k
                                ukonceni rozpoznavani")
                        return
                okno.insert(INSERT, rec)
                okno.insert(INSERT, " ")
                okno.update()


def konec(event):
        '''
        funkce ukoncujici program
        '''
        hlavniOkno.destroy()


hlavniOkno = Tk()
hlavniOkno.title("STT")
ramec = Frame(hlavniOkno, relief=SUNKEN, border=1)
image = PhotoImage(file='python.gif')
label4 = Label(ramec)
```

```
label3 = Label(ramec, text = 'Program pro prevod reci na text.\nVytvoreno v jazyce Python 2.5\nVytvoril Bc. Jan Holy')

label4['image'] = image

label4.pack(side=LEFT)

label3.pack(side=LEFT)

ramec.pack(side=TOP, expand=True)

popisek = Label(hlavniOkno, text='\nRozpoznany text:')

popisek.pack()

okno = Text(hlavniOkno, width=30, height=10)

okno.pack()

popisek1 = Label(hlavniOkno, text='\nRozpoznavani reci je ukonceno\nvyslovenim prikazu \'EXIT\'\n')

popisek1.pack()

tlacitko1 = Button(hlavniOkno, text = 'Zacit rozpoznavani', command = rec)

tlacitko1.bind("<Escape>",konec)

tlacitko1.pack()

tlacitko = Button(hlavniOkno, text = 'Konec', command = hlavniOkno.destroy)

tlacitko.pack()

hlavniOkno.mainloop()
```

Příloha 3: Seznam samostatných slov, jež byl předkládán rozpoznávačům při měření úspěšnosti.

car	jobs
hello	milk
bus	gold
john	glass
but	wood
apple	Google
dad	silver
phone	ice
blue	iron
lion	carrot
red	man
dog	hat
cat	tie
fish	bag
fly	face
food	arm
dinner	bone
ten	horse
shop	table
school	chair
pet	pink
one	brown
bank	bed
bridge	pc
factory	wallet

Příloha 4: Seznam slovních spojení a vět, na nichž se testovala úspěšnost rozpoznávačů.

My name is John.

How are you?

I have a blue car.

The sun is yellow.

Where are you from?

There is a three.

Dog and cat;

Red apple;

How old are you?

He has got a lion.

Nice to meet you.

Is he married?

Green grass;

Beautiful day;

Good morning;

Glass of milk;

Gold paper;

Brown horse;

It is lovely.

She is ill.

Need a job;

He is crazy.

I can swim.

He loves her.

We live in London.

Příloha 5: Tabulky znázorňující úspěšnosti pro jednotlivá měření.

Tabulka úspěšnosti rozpoznání pro jednotlivé hlasové režimy:

č. měření		1	2	3	4	5	6	7	8	9	10	průměr	σ(Xi)
hlasové režimy	základní	49%	48%	32%	36%	42%	44%	38%	46%	38%	42%	**41,50%**	**5,20%**
	pokročilý	57%	62%	52%	48%	56%	46%	52%	58%	54%	56%	**54,10%**	**4,53%**
	expert	70%	66%	64%	68%	68%	70%	72%	66%	70%	72%	**68,60%**	**2,54%**

Tabulka úspěšnosti rozpoznání jednotlivých slov:

č. měření	1	2	3	4	5	6	7	8	9	10	průměr	σ(Xi)
Microsoft Speech - expert	70%	66%	64%	68%	68%	70%	72%	66%	70%	72%	**68,60%**	**2,54%**
Dragon Naturally Speaking	68%	58%	62%	60%	58%	66%	64%	66%	60%	68%	**63,00%**	**3,71%**
Google translator	72%	66%	70%	70%	72%	68%	70%	72%	74%	70%	**70,40%**	**2,15%**

Tabulka úspěšnosti rozpoznání jednotlivých slov při šumu 60-80 dB:

č. měření	1	2	3	4	5	6	7	8	9	10	průměr	σ(Xi)
Microsoft Speech - expert	54%	46%	44%	48%	48%	46%	52%	50%	54%	52%	**49,40%**	**3,35%**
Dragon Naturally Speaking	50%	46%	50%	52%	46%	48%	50%	46%	44%	50%	**48,20%**	**2,44%**
Google translator	64%	62%	58%	60%	62%	66%	56%	60%	62%	64%	**61,40%**	**2,84%**

Tabulka úspěšnosti rozpoznání pro slovní spojení nebo věty:

č. měření	1	2	3	4	5	6	7	8	9	10	průměr	σ(Xi)
Microsoft Speech - expert	80%	84%	76%	76%	68%	80%	72%	76%	80%	80%	**77,20%**	**4,40%**
Dragon Naturally Speaking	68%	68%	64%	68%	72%	68%	64%	76%	72%	68%	**68,80%**	**3,49%**
Google translator	88%	80%	84%	88%	84%	80%	76%	84%	88%	84%	**83,60%**	**3,77%**

Tabulka úspěšnosti rozpoznání pro jednotlivá slova namluvená jiným mluvčím:

č. měření	1	2	3	4	5	6	7	8	9	10	průměr	σ(Xi)
Microsoft Speech - expert	42%	30%	32%	40%	34%	38%	34%	40%	36%	34%	**36,00%**	**3,69%**
Dragon Naturally Speaking	48%	44%	52%	50%	48%	54%	46%	50%	52%	48%	**49,20%**	**2,86%**
Google translator	60%	66%	52%	64%	62%	58%	64%	62%	62%	56%	**60,60%**	**4,00%**

σ(Xi) představuje směrodatnou odchylku, která se vypočte podle následujícího vzorce: $\sigma(Xi) = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} (Xi - \bar{X})^2}$,

kde: Xi je hodnota i-tého vzorku měření,

$\bar{X}$ je aritmetický průměr ze základního souboru.