UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Rozpoznání slov diskrétního diktátu

Bc. Miloslav Kočí

Diplomová práce
2010

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2009/2010**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Miloslav KOČÍ**
Osobní číslo:	**I08338**
Studijní program:	**N2612 Elektrotechnika a informatika**
Studijní obor:	**Komunikační a řídicí technologie**
Název tématu:	**Rozpoznání slov diskrétního diktátu**
Zadávající katedra:	**Katedra elektrotechniky**

Zásady pro vypracování:

V teoretické části popište proces vytváření řeči a možnosti zpracování řečového signálu určeného k rozpoznání izolovaných slov pomocí Mel-frekvenčních kepstrálních koeficientů.
V praktické části se zaměřte na rozpoznání slov z diskrétního diktátu, kde budou analyzována podobně znějící slova mužského a ženského hlasu. Pro tyto účely vytvořte softwarovou aplikaci v prostředí Matlab pro záznam hlasu, zpracování řečového signálu a rozpoznávání izolovaných slov. Výsledky přehledně zpracujte pomocí tabulek a vyhodnoťte úspěšnost rozpoznání vybraných slov různých řečníků.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

PSUTKA, Josef. 1995. Komunikace s počítačem mluvenou řečí. Praha : Academia Praha, 1995. ISBN 80-200-0203-0.

PSUTKA, Josef, a další. 2006. Mluvíme s počítačem česky. Praha : Academia Praha, 2006. ISBN-80-200-1309-1.

Vedoucí diplomové práce: **Ing. Zdeněk Němec, Ph.D.**
Katedra elektrotechniky

Datum zadání diplomové práce: **15. ledna 2010**

Termín odevzdání diplomové práce: **21. května 2010**

L.S.

prof. Ing. Simeon Karamazov, Dr.
děkan

Ing. Zdeněk Němec, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2010

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 28. 6. 2010 *Bc. Miloslav Kočí*

Poděkování

V první řadě bych rád poděkoval svému vedoucímu práce Ing. Zdeňku Němcovi, Ph.D. za všechen čas, který mi věnoval a za nepřeberné množství poznámek, které práci obohatily. Poděkovaní patří také panu doc. Dr. Ing. Janu Černockému za materiály, které mi poskytnul a v neposlední řadě panu Ing. Vojtěchu Stejskalovi, Ph.D. za konzultaci, na kterou si dokázal najít čas. Dále bych chtěl poděkovat členům kolektivu Fakulty elektrotechniky a informatiky za jejich vstřícnost a ochotu při vytváření databáze záznamů řeči. Na závěr chci poděkovat mé přítelkyni a kamarádům, kteří mne podporovali v průběhu vytváření práce.

Bc. Miloslav Kočí

Anotace

Diplomová práce se zabývá problematikou zpracování řečového signálu a rozpoznávání slov v diskrétním diktátu. Popsána je posloupnost procesů, které budou řečový signál zpracovávat a jednotlivá slova diskrétního diktátu porovnávat s předem namluvenými referenčními slovy. Podle této posloupnosti je vytvořena softwarová aplikace v prostředí Matlab, která rozpoznává slova v promluvě muže i ženy. Aplikace je testována na řečovém signálu s akusticky si podobnými i zcela odlišnými slovy.

Klíčová slova

Lidská řeč, Mel-frekvenční kepstrální koeficienty, banka filtrů, dynamické borcení času, rozpoznávání slov.

Title

Word Recognition of Discrete Dictate

Annotation

The thesis deals with issues of the processing of speech signal and the differentiation of words in a discrete dictation. It describes the sequence of the processes that will analyze the speech signal and will compare individual words of the discrete dictation with pre-recorded reference words. This sequence is the base of a software application in the Matlab environment that differentiates words in the speech of a man and a woman. The application is tested on a speech signal with similar as well as entirely different words from the point of view of acoustics.

Keywords

Human speech, Mel-frequency cepstral coefficients, Mel filter bank, dynamic time warping, word recognition.

Obsah

Seznam zkratek

DFT	Discrete Fourier Transform
MFCC	Mel-Frequency Cepstral Coefficient
DCT	Discrete Cosine Transform
DTW	Dynamic Time Warping
USB	Universal Seriál Bus
CD	Compact Disk
IR	Infrared Radiation

Seznam obrázků

Seznam tabulek

1 Úvodní informace

V dnešní době plné různých komunikačních aplikací, sociálních sítí i jiných technických vymožeností stále zůstává základním a nejpřirozenějším prostředkem pro předávání informací mezi lidmi mluvená řeč. Pokud chceme, aby člověk řečí předával informace i počítačovému systému, je nutné vyřešit řadu nelehkých úkolů, které se týkají zejména zpracování řečového signálu a rozpoznávání řeči. V případě, že chceme, aby počítačový systém byl schopen člověku odpovědět, je potřeba vyřešit i počítačovou syntézu. Vyřešením těchto úloh se vědci zabývají již několik staletí. Už ve druhé polovině 18. století byly popsány první testy mechanického syntezátoru lidské řeči rakouským vynálezcem von Kempelenem. Od té doby bylo vymyšleno a experimentálně ověřeno několik metod pro zpracování, rozpoznávání a syntézu lidské řeči. Nejvýznamnější milník v této oblasti nastal s příchodem číslicových počítačů. Přes značný pokrok v této problematice, kdy jsou systémy schopny vyřešit řadu úloh pro zjednodušení nebo zefektivnění mnoha činností, je nutné konstatovat, že přirozená plynulá komunikace mezi počítačem a člověkem zůstává pouze příslibem budoucnosti. Je to hlavně díky zatím nepřekonatelným problémům s rozpoznáním smyslu promluvy řečníka. Celý proces od zpracování signálu sluchovým ústrojím až po rozpoznání informace v mozku není dostatečně známý, proto nemáme dostatek informací pro konstrukci obdobného systému.

I přes mnoho zatím nedořešených problémů, s nimiž se systémy hlasové komunikace člověka s počítačem potýkají, je jejich využití v průmyslových nebo společenských systémech stále častější. U většiny případů jde o komunikaci s omezeným počtem slov a v prostředí s předem známým rušivým pozadí. Všeobecně využívány jsou systémy, které ovládají různé stroje a zařízení pomocí hlasových povelů. U takových systémů je pro lepší rozpoznávání doporučováno, aby referenční slova uložená v paměti systému byla namluvená stejným řečníkem, který pak systému příkazy zadává. Velmi známá je také aplikace hlasového vytáčení u telefonů, kde systém vytočí číslo, ke kterému je přiděleno vyslovené jméno. Vysoké přirozenosti v dnešní době dosáhly aplikace s převodem psaného textu na mluvené slovo, což mohou využívat zrakově indisponovaní lidé při čtení e-mailů, nebo SMS zpráv. Sluchově indisponovaní lidé určitě v budoucnu ocení on-line titulkování televizních přenosů, kde není předem známý dialog (např. sportovní přenosy, politické diskuse atd.). V poslední době se intenzivně pracuje na aplikacích překladu z řeči do řeči, které uvažují vstup v jednom jazyce, rozpoznání promluvy, automatického překladu do jiného jazyka a následnou syntézou přeložené věty. V budoucnu by měla být řeč jedním z hlavních způsobů komunikace s počítačem a následně pak i s jakýmkoliv jiným strojem. Stroj by měl nejen dobře rozumět, ale měl by být i schopen identifikovat mluvčího pro případ neoprávněného rozkazu. Zajímavé by bylo, kdyby byl počítač schopen v promluvě rozpoznat třeba vtip. To se ale bude týkat ještě vzdálenější budoucnosti.

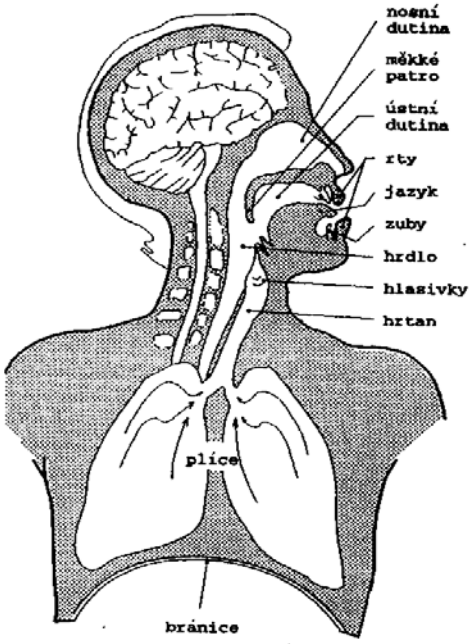
2 Lidská řeč

Mluvená řeč se přenáší komunikačním kanálem v podobě akustického signálu. Podstatou akustického (řečového) signálu je vlnění elastického prostředí v množině slyšitelných frekvencí. Pod pojmem komunikační kanál si představme prostředí, kterým se šíří akustický signál od zvukového ústrojí mluvčího ke sluchovému ústrojí posluchače. V akustickém signálu promluvy jsou zakódovány různé druhy informací. Kromě akustické složky (amplitudově-frekvenční časové spektrum) řečový signál obsahuje lingvistickou složku (fonetická, morfologická, syntaktická, sémantická či pragmatická struktura) vyjadřující význam promluvy. Další složka akustického signálu nese specifické informace o mluvčím závislé na hlasovém ústrojí a způsobu artikulace (intonace, rytmus řeči, barva hlasu atd.) včetně možných anomálií jako jsou třeba vady řeči. Tato složka obsahuje i informaci o emocionálním stavu řečníka (stres, rozčílení, smutek, radost atd.)

2.1 Proces vytváření řeči člověkem

V lidském těle je několik orgánů, které se zabývají vytvářením řeči, souhrnně tyto orgány nazýváme řečové (artikulační) orgány, nebo také jednoduše mluvidla (artikulátory). Ovšem vytváření řeči nebývá primárním úkolem těchto orgánů. Jejich základní funkce jsou v lidském těle různé a většinou spolu ani nijak nesouvisí. Společnou mají až účast na vytváření řeči. Seskupení těchto orgánů v těle tvoří hlasový trakt. Hlasový trakt lze rozdělit na tři základní části (PSUTKA, a další, 2006):

- dechové ústrojí,
- hlasové ústrojí,
- a artikulační ústrojí.



Obrázek 1 - Hlasový trakt člověka (PSUTKA, 1995).

2.1.1 Dechové ústrojí

Dechové ústrojí představuje zdroj energie pro řeč. Je umístěno v hrudním koši a tvořeno přívodní dýchací cestou, plícemi a s nimi funkčně spjatými dýchacími svaly (bránicí). Při nádechu dochází k pohybu vzduchu, který tak poskytuje zdroj energie pro řeč. Při výdechu potom v plicích vzniká výdechový proud vzduchu, který je v zásadě základním materiálem pro tvorbu řeči. Výdechový proud je z plic odváděn průdušnicí, a pak prochází hrtanem a nadhrtanovými dutinami, kde se modifikuje a jako řečový signál je vyzařován rty do okolního prostoru. Síla výdechového proudu ovlivňuje způsob fungování hlasového ústrojí, a tím má vliv na sílu hlasu a částečně i na jeho výšku.

2.1.2 Hlasové ústrojí

Pojmem hlasové ústrojí se často označuje celý systém pro vytváření řeči. Zde budeme pod tímto pojmem rozumět pouze tu část, kde bude docházet k samotnému vzniku hlasu. Hlasové ústrojí je uloženo v hrtanu, který je s plícemi spojen průdušnicí. Z hlediska tvorby řeči nejdůležitější část hlasového ústrojí tvoří hlasivky. Jsou to dvě ostré slizniční řasy, které vedou napříč hrtanem v místě jeho nejužšího průchodu. Při vytváření hlasu (fonaci) se hlasivky nacházejí v tzv. hlasovém (fonačním) postavení. Výdechový proud vzduchu postupuje bez odporu z plic průdušnicí až k hrtanu. Zde se mu do cesty postaví překážka vytvořená hmotou hlasivek, které cestu vzduchu úplně uzavřou. Stažené hlasivky se pod tlakem vzduchu stávají pružnými a začínají kmitat. V důsledku kmitání hlasivek vzniká základní (hlasivkový) tón, který představuje nosný zvuk řeči. Frekvence kmitání hlasivek se označuje F_0 a nazývá se frekvence základního hlasivkového tónu. Tato frekvence nabývá hodnot asi od $60 - 400$ Hz. U mužů se F_0 pohybuje asi mezi $80 - 160$ Hz, u žen je to $150 - 300$ a u dětí asi $200 - 400$ Hz. Fonační postavení hlasivek má za následek vznik hlasivkového tónu a používá se proto při vytváření znělých zvuků řeči (samohlásky a znělé souhlásky). Neznělé zvuky jsou naopak tvořeny při klidovém postavení hlasivek, neobsahují tedy základní hlasivkový tón a vznikají tedy až modifikací výdechového proudu vzduchu v nadhrtanových dutinách.

2.1.3 Artikulační ústrojí

Artikulační ústrojí je posledním ústrojím, které se podílí na tvorbě řeči. Jeho význam spočívá v tom, že umožňuje vytvářet velké množství různých zvuků, které charakterizují mluvený jazyk. Skládá se jednak z nadhrtanových dutin a jednak z artikulačních orgánů, které jsou v těchto dutinách uloženy nebo je obklopují. Mezi nadhrtanové dutiny řadíme dutinu hrdelní, ústní a nosní. Hranici mezi těmito dutinami tvoří čípek, špička měkkého patra, které zamezuje nebo umožňuje přístup vzduchu z dutiny hrdelní do dutiny nosní.

Zatímco se nadhrtanové dutiny účastní procesu tvorby řeči pasivně (nepohybují se), artikulační orgány (artikulátory) se účastní tvorby řeči většinou aktivně – tvoří pohyblivé součásti artikulačního ústrojí a svým pohybem mění velikost nadhrtanových dutin. Z hlediska vytváření řeči mezi nejvýznamnější artikulátory patří jazyk, rty a měkké patro, neboť se podílejí na vytváření největšího počtu různých zvuků. Dalšími artikulátory potom

jsou zuby, tvrdé patro nebo čelisti. Artikulátorem je také hrtan, který se může pohybovat a tím měnit délku celého hlasového traktu.

2.2 Informační obsah řeči

2.2.1 Fonetická forma řeči

Za nejmenší jednotku řeči, která může rozlišovat jednotlivá slova, lze považovat foném. Fonémy lze od sebe rozlišit například podle místa tvoření, podle artikulujícího orgánu, nebo podle sluchového dojmu.

Počet fonémů v existujících světových jazycích se pohybuje od 12 do 60. V českém jazyce je jich 36, v anglickém 42, v ruském 40 apod. Fonémy se spojují do posloupnosti spojených celků, v nichž lze nalézt další stavební jednotku – slabiku (tu lze již přesně srovnávat s psanou formou). Libovolné promluvy jsou vlastně pravidelným opakováním různých posloupností slabik. Slovo je určitou kombinací slabik, přičemž jejich počet tvoří vždy celé číslo. Slovanské jazyky používají kolem $2500 - 3500$ slabik a $45\,000 - 50\,000$ slov.

Při běžném rozhovoru vysloví člověk asi $80 - 130$ slov za minutu, což představuje frekvenci výskytu asi 10 fonémů za sekundu. Jestliže uvážíme průměrné množství informace na jeden foném $H = 3 - 4$ bit, dostaneme pro mluvenou řeč rychlost přenosu informace asi $30 - 40$ bit/s. Tento výsledek tedy charakterizuje informační obsah řeči objevující se v její fonetické struktuře. Z psychoakustických testů bylo zjištěno, že člověk je schopen zpracovat informaci o rychlosti maximálně 50 bit/s (PSUTKA, 1995).

2.2.2 Akustická forma řeči

Poněkud jiné výsledky získáme, zkoumáme-li mluvenou řeč ne z hlediska informačního obsahu její fonetické struktury, nýbrž z hlediska informačního obsahu skutečného průběhu akustického signálu. Akustický signál je charakterizován jednak průběhem amplitudy (energie) v čase a jednak průběhem změn frekvence v čase. Pro kvantitativní vyjádření informačního obsahu zde využijeme Shannonovy věty o výběru. Z této věty vyplývá, že spojitou funkci času, jejíž kmitočtové spektrum je omezené tak, že neobsahuje vyšší kmitočty než F_m, je možno nahradit řadou jejích hodnot (vzorků), jestliže frekvence vzorkování $F_s \geq 2F_m$.

Jestliže budeme uvažovat $F_m = 12 - 16$ kHz a dynamiku řeči 50 dB při minimálním šumu, byla by pro dobrý záznam takového signálu bitová rychlost větší než $C = 200000$ [bit/s].

Z důvodů obrovské informační nadbytečnosti využívá člověk takové interní mechanismy, které mu umožní potlačit v řečovém signálu v tu chvíli nepotřebné údaje (barvu hlasu, intonaci apod.) a zdůraznit pouze několik hlavních opěrných zvukových příznaků. Tyto opěrné příznaky obsahují základní informaci, která je zakódována v řečovém signálu a je shodná pro všechna stejná slova. Samozřejmě, že pokud výsledkem vnímání řeči má být např. identifikace řečníka, bude posluchač využívat jiných

charakteristik promluvy ap. Problémem je, že tyto mechanismy vnímání řeči člověkem, které zahrnují proces od zpracování signálu sluchovým orgánem až po porozumění dané zprávě při zpracování v mozku, nejsou v současné době dostatečně známy, a tak nelze využít ani jejich analogie při návrhu technických a programových prostředků, které by dosahovaly obdobných cílů – rozpoznávání řeči, identifikaci řečníka, porozumění řeči apod. Nicméně je alespoň vhodné využít dostupných znalostí o procesu tvoření řeči a snažit se odhadnout, které charakteristiky akustického signálu nesou podstatnou informaci nutnou pro vyřešení konkrétní úlohy. Intuitivně je například zřejmé, že technické a programové prostředky při rozpoznávání omezené množiny slov mohou být jednodušší než při rozpoznávání plynulé řeči, anebo v případě identifikace řečníka bude třeba zkoumat jiné příznaky akustického signálu než při rozpoznávání slov (PSUTKA, 1995).

3 Záznam řeči a její digitalizace

Signál je fyzikální veličina nesoucí informaci. Lidská řeč také přenáší informaci, můžeme jí proto interpretovat také jako signál. Ve volném prostředí se tento řečový signál šíří jako mechanické kmity vzduchu. Tato forma však není příliš vhodná pro jeho následné zpracování, a proto se pomocí mikrofonu převádí na signál elektrický. Z hlediska spojitosti v čase a úrovni rozlišujeme následující tři typy signálů (LAVICKÝ, 2003):

- signály analogové (spojité v čase i amplitudě),
- signály diskrétní (spojité v amplitudě, diskrétní v čase),
- signály číslicové (diskrétní v čase i v amplitudě).

3.1 Pulzní kódová modulace

Řečový signál zaznamenaný mikrofonem je analogový a pro další zpracování je zapotřebí tyto analogové kmity zpravidla převést do číslicového tvaru tak, aby spojitý signál byl reprezentován posloupností číselných údajů. Tento proces, který se nazývá pulzní kódová modulace nebo též digitalizace, zahrnuje provedení dvou kroků, a to vzorkování a kvantizaci s kódováním (PSUTKA, 1995).

3.1.1 Vzorkování

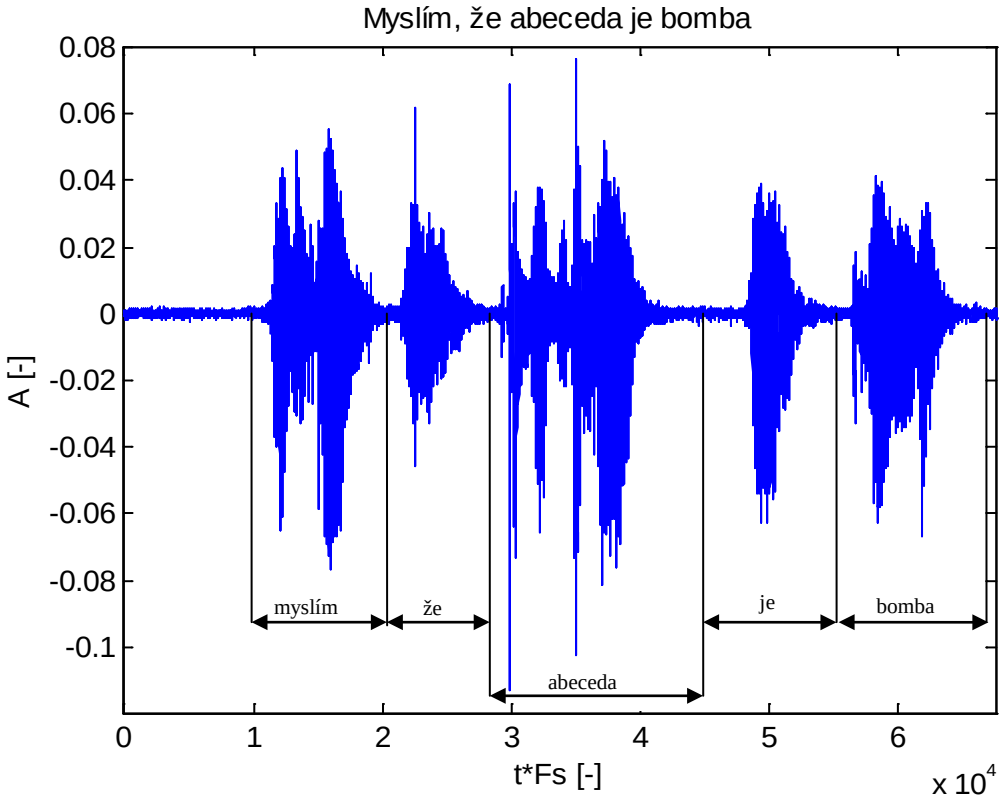
Vzorkování je proces, při němž je analogový signál převeden na diskrétní, tzn. je nahrazován jeho částmi – vzorky. Vzorky jsou od sebe zpravidla rovnoměrně vzdáleny (rovnoměrné vzorkování) o vzorkovací periodu T_S. Rovnoměrné vzorkování lze chápat jako násobení signálu souvislého času periodickým vzorkovacím signálem. Pro zjednodušení výpočtů zavádíme ideální vzorkování, při němž je vzorkovacím signálem posloupnost Diracových impulsů. Důležitou vlastností vzorkovaného signálu je periodicita jeho spektra. Díky této periodicitě, vzniká při vzorkování jisté nebezpečí nevratné ztráty informace v důsledku překrytí spekter dvou sousedních period. Aby k tomuto překrytí nemohlo dojít, je třeba, aby byl splněn Shannonův vzorkovací teorém pospaný v odstavci 2.2.2(LAVICKÝ, 2003).

3.1.2 Kvantizace s kódováním

Kvantizace s následným kódováním je přiřazení analogové hodnoty vzorku signálu jedné hodnotě z konečného počtu číselných hodnot. Každý vzorek je tedy vyjádřen B-bitovým binárním kódem a počet možných úrovní je obvykle 2^B. Nejmenší možný rozdíl mezi dvěma hodnotami kvantovaného signálu se nazývá kvantovací krok. Při kvantizačním procesu dochází k určité ztrátě informace „zaokrouhlováním“ měřených okamžitých velikostí signálu. Tato ztráta se nazývá kvantizační šum (PSUTKA, 1995).

4 Předzpracování a analýza akustického signálu

Před tím, než budou jednotlivé části řečového signálu připraveny k rozpoznávání, musí se signál upravit. V této kapitole se budeme kromě předzpracování zabývat také analýzou řečového signálu. Názorné ukázky použitých aplikací budou demonstrovány na větě „Myslím, že abeceda je bomba“.



Obrázek 2 - Časový průběh věty „Myslím, že abeceda je bomba“.

4.1 Ustředění

Na úvod zpracování odečteme od signálu jeho střední hodnotu, stejnosměrná složka totiž nenese žádnou potřebnou informaci, naopak může být při některých výpočtech na škodu (výpočet krátkodobé energie). Výpočet střední hodnoty je komplikovanější, pokud neznáme průběh celého signálu, viz (ČERNOCKÝ, 2006). V našem případě ovšem celý signál známe, proto jednoduše:

$$\bar{s} = \frac{1}{N}\sum_{n=1}^{N} s[n], \tag{4.1}$$

kde N je celkový počet vzorků signálu,

$s[n]$ jsou jednotlivé vzorky řečového signálu.

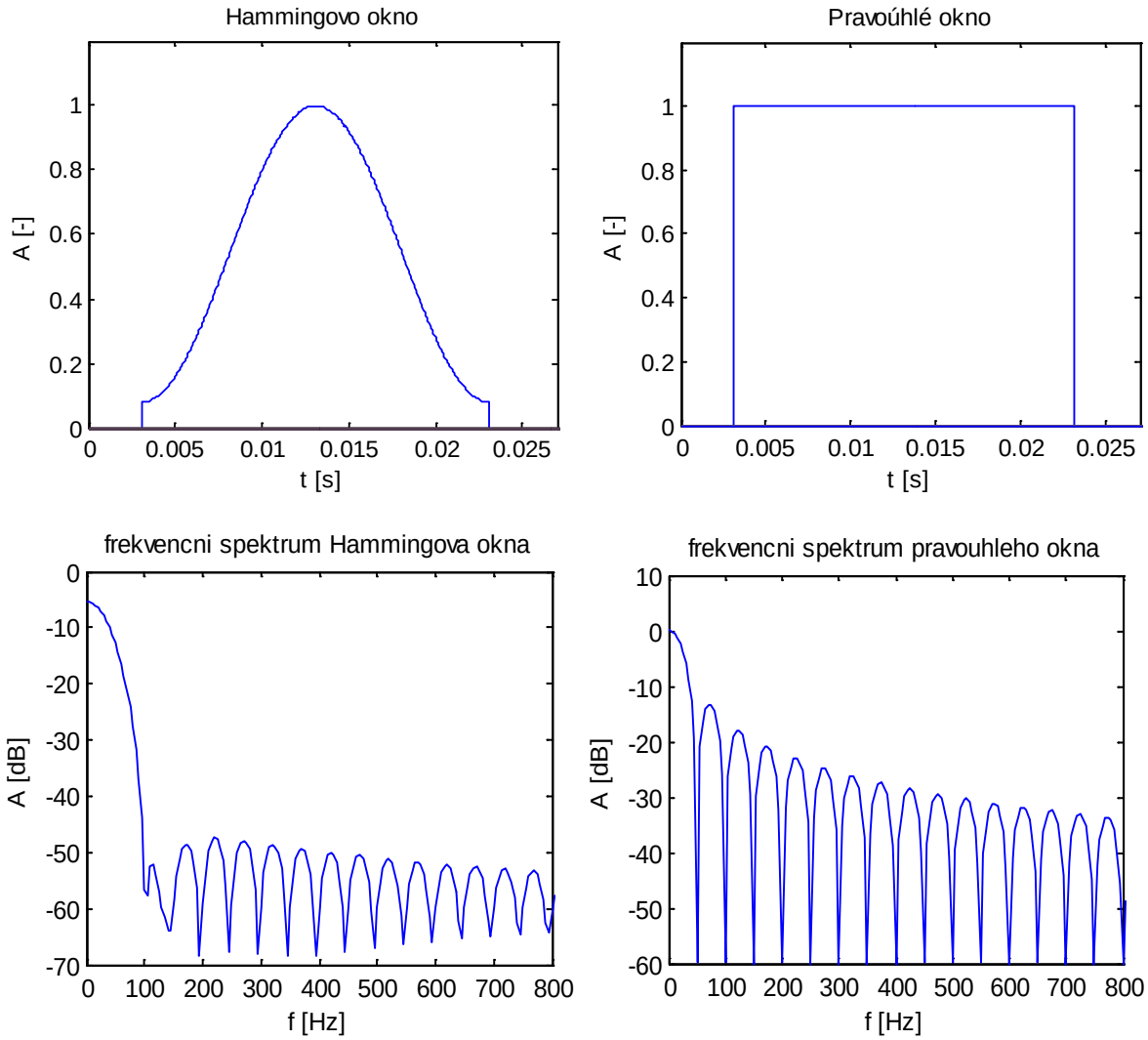
4.2 Preemfáze

Preemfáze je určena k vyrovnání kmitočtové charakteristiky řeči (energie řeči klesá směrem k vyšším frekvencím). Jedná se spíše o historickou operaci, pokud si uvědomíme, co se děje při výpočtu Mel-frekvenčních kepstrálních koeficientů (viz dále v odstavci 5.2) nemá preemfáze žádný vliv na další kroky předzpracování signálu. Více informací o preemfázi je uvedeno v (ČERNOCKÝ, 2006).

4.3 Rámcování signálu

Řečový signál je nutné před dalším zpracováním rozdělit na rámce. Důvod je prostý. Metody pro odhad parametrů dobře pracují se stacionárním signálem, což celý signál určitě není. Pokud chceme zpracovávat stacionární úsek, musíme uvažovat setrvačnost hlasového ústrojí. Signál tedy většinou dělíme na $20 - 25$ ms dlouhé úseky. Délka rámce ve vzorcích při $Fs = 16$ kHz je $L_{ram} = 320 - 400$ vzorků.

Pro lepší zachování kontextu je vhodné, aby se jeden úsek řeči stal součástí několika rámců, tedy aby se sousední rámce z určité části překrývaly. Samozřejmě s rostoucí hodnotou překrytí se zvyšují i nároky na paměť a také si budou sousední rámce více podobné. Nicméně s malým překrytím se hodnoty parametrů mezi sousedními rámci můžou hodně měnit. Proto je vhodné udělat kompromis, kterým je z pravidla překrytí: $P_{ram} = 10\text{-}15$ ms. Samotné rámcování signálu se provádí pomocí vykrojení tzv. okna. Tvar okna může být různý, nejčastěji se ovšem využívá pravoúhlé nebo Hammingovo okno. Pravoúhlé okno má sice ve frekvenčním spektru užší hlavní svazek, ale výrazné postranní laloky. Oproti tomu Hammingovo má širší hlavní svazek, ale postranní laloky jsou značně tlumeny. Ukázka časové i frekvenční charakteristiky pravoúhlého a Hammingova okna ukazuje obrázek 3. Vzorky signálu v rámcích budeme pro další zpracování označovat $x[n]$.



Obrázek 3 - Srovnání frekvenční a časové charakteristiky pravoúhlého a Hammingova okna.

4.4 Počet průchodů nulou

Pokud máme signál rozdělený na rámce, můžeme si pro každý z nich určit počet průchodů nulou Z. Tento parametr se využívá k určení znělých a neznělých hlásek. Při vytváření neznělých hlásek se nepoužívají hlasivky (PSUTKA, 1995), proto se více podobají šumu a mají tedy vyšší Z. Počet průchodů nulou je ale značně náchylný na šum, je proto někdy problém rozlišit neznělou hlásku od šumu. Pro získání počtu průchodů nulou se v Matlabu využívá funkce sign.

$$Z = \frac{1}{2}\sum_{n=1}^{L_{ram}} |sign\ x[n] - sign\ x[n-1]|, \tag{4.2}$$

$$\text{kde} \begin{cases} sign\ x[n] = 1 & pro\ x[n] > 0 \\ sign\ x[n] = -1 & pro\ x[n] < 0 \\ sign\ x[n] = 0 & pro\ x[n] = 0 \end{cases}$$

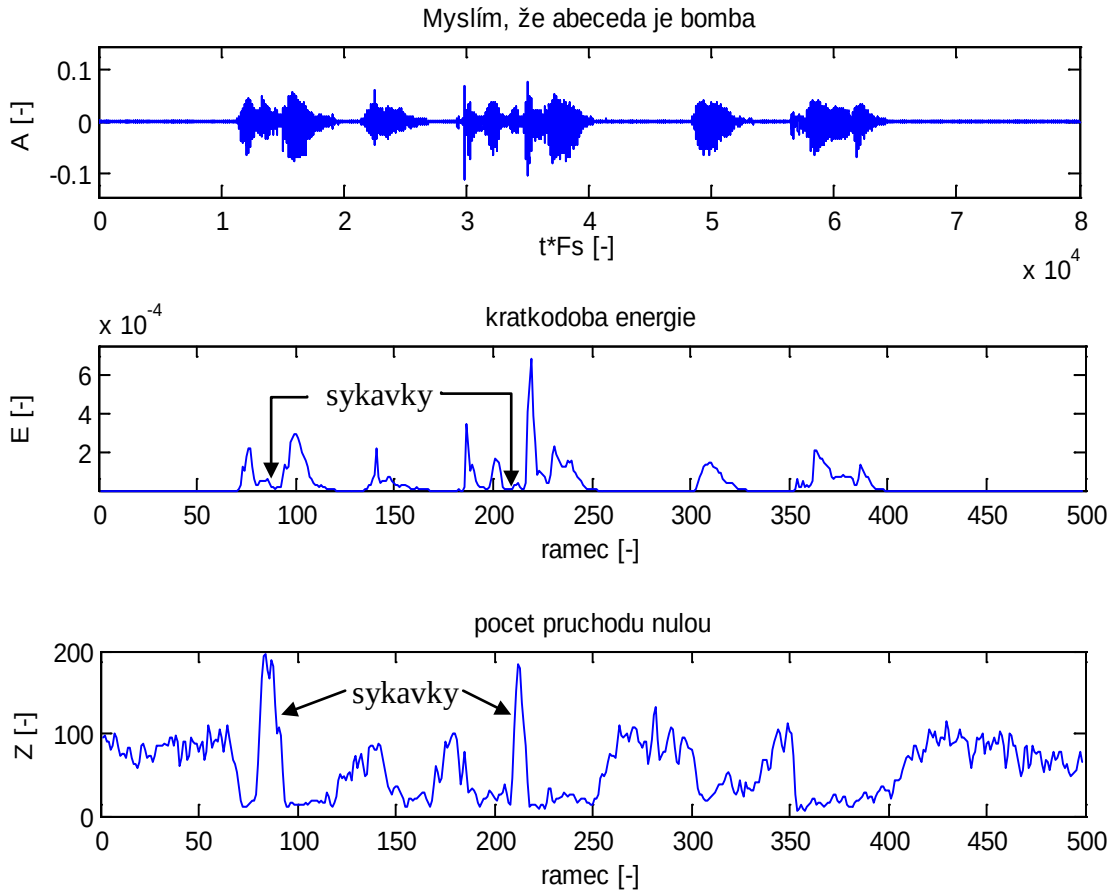
Při průchodu signálu nulou je hodnota v sumě rovna dvěma, proto musíme ještě výsledek o polovinu zmenšit. Problém nastane, pokud signál začíná v nule. V tom případě hodnota v sumě je rovna 1 a ve výsledku se to projeví hodnotou o 0,5 větší.

4.5 Střední krátkodobá energie a intenzita

Podobně jako u počtu průchodů nulou můžeme střední krátkodobou energii využít na rozlišení znělých (vyšší energie) a neznělých (nižší energie) hlásek. Tato energie se také využívá na detekci řečové aktivity, je to vlastně nejjednodušší způsob, jak řeč detekovat. Tato metoda je ovšem stejně jako předchozí hodně náchylná na šum, hlavně u neznělých hlásek metoda často selhává. Krátkodobá energie E se lehce vypočítá jako:

$$E = \frac{1}{L_{ram}} \sum_{n=0}^{L_{ram}-1} x^2[n]. \tag{4.3}$$

Obrázek 4 znázorňuje počet průchodů nulou a krátkodobou energie pro jednotlivé rámce naší věty, ve které řečový signál není příliš ovlivněn šumem, a proto jsme schopni řečovou aktivitu dobře rozpoznat. Všimněme si také, kolik průchodů nulou a jak velkou krátkodobou energii mají tzv. sykavky (s, c, z), které mají silně šumovou strukturu.

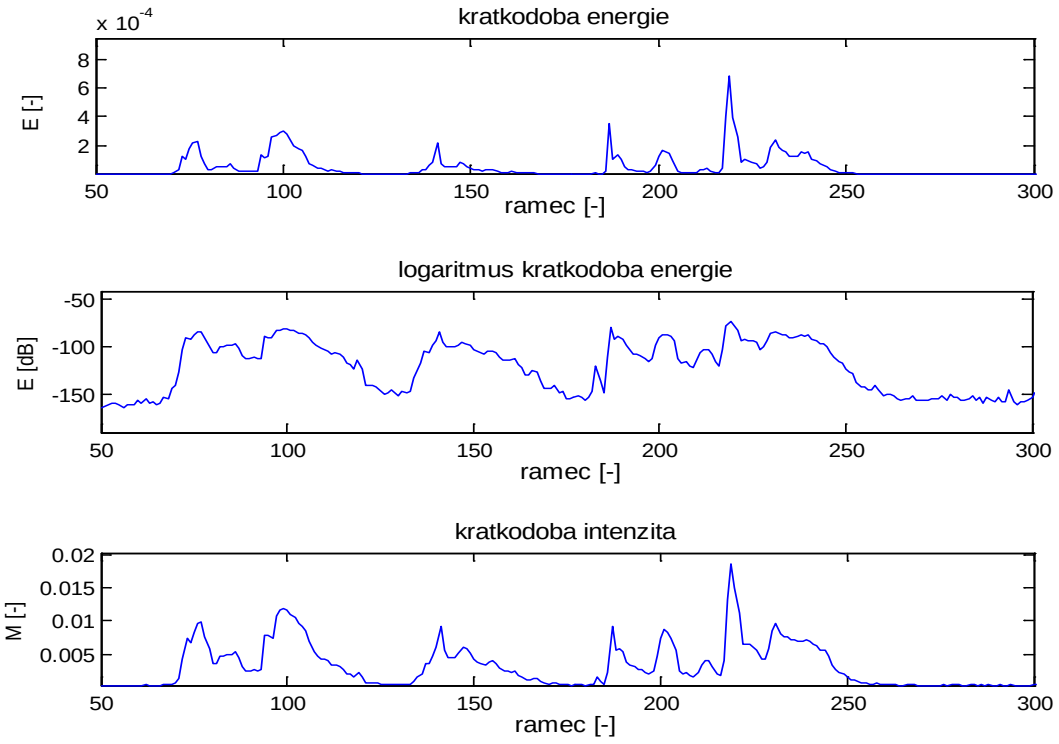


Obrázek 4 - Ukázka počtu průchodů nulou a krátkodobé energie u příkladové věty.

Kvůli zmenšení dynamického rozsahu se často energie uvádí v decibelech, nebo se pracuje s krátkodobou intenzitou M:

$$M = \frac{1}{L_{ram}} \sum_{n=0}^{L_{ram}-1} |x[n]|. \tag{4.4}$$

Srovnání ukazuje obrázek 5, kde je pro lepší porovnání zobrazena první polovina věty a charakteristika přiblížena.



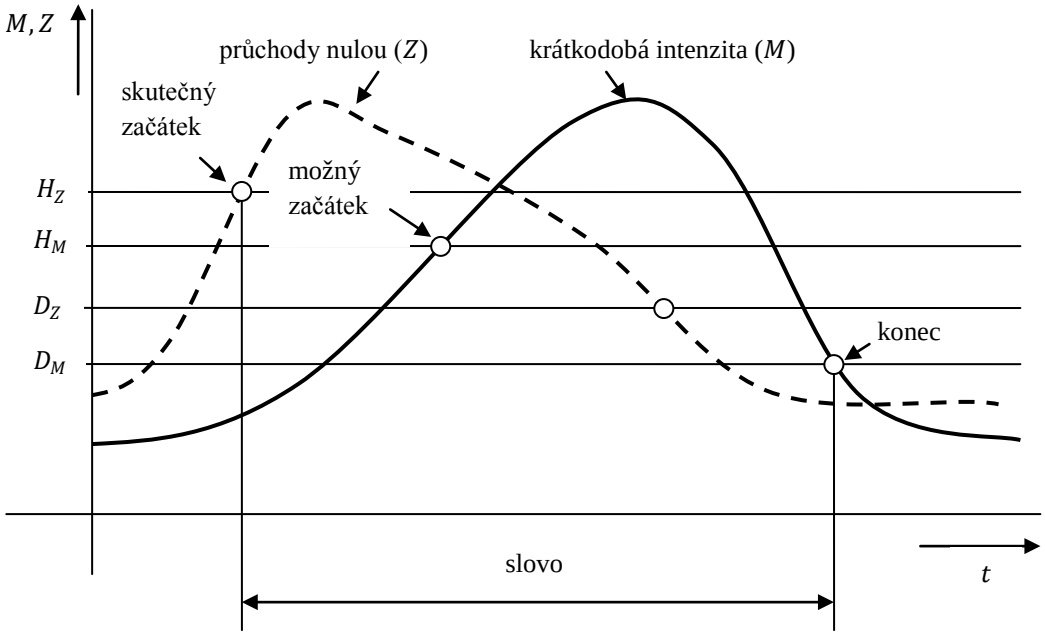
Obrázek 5 - Porovnání krátkodobé energie s jejím logaritmem a krátkodobou intenzitou.

4.6 Určování hraničních bodů promluvy

Závažným problémem při zpracování řečového signálu pro účely rozpoznávání je určení hraničních bodů, tj. začátku a konce, promluvy. Analyzátor většinou nepřetržitě zpracovává veškerý vstupní signál snímaný z mikrofonu, telefonu apod., a jeho příznakovou reprezentaci předává klasifikátoru. Jestliže je zabezpečen vysoký odstup signálu od šumu (velmi tichá místnost, kvalitní mikrofon apod.), pak lze problém nalezení hraničních bodů promluvy vyřešit poměrně snadno na základě zvýšené úrovně intenzity signálu řeči, neboť ta i pro nejslabší zvuky, jako jsou neznělé sykavky, převyšuje intenzitu okolního šumu. Tyto případy však nejsou běžné, naopak v reálných aplikacích pracují zejména klasifikátory izolovaných slov v prostředích s podstatným rušivým pozadím.

Průběh intenzity je pro určení hraničních bodů promluv rozhodující, v některých případech pro přesné stanovení ovšem nestačí. Proto je pro korektní vymezení hraničních bodů vhodné využít i spektrálních charakteristik řečového signálu. Jedna z takových metod

využívá například průběhy krátkodobé intenzity M a krátkodobé funkce středního počtu průchodů signálu nulou Z. V (PSUTKA, 1995) je ukázaná metoda, kde jsou pro určení začátků promluvy nejprve stanoveny horní mez intenzity H_M a horní mez počtu průchodů signálu nulou H_Z. Překročí-li intenzita signálu mez H_M v q mikrosegmentech (rámců) za sebou (vhodné se ukazuje $q = 3$ až), lze usuzovat na výskyt začátku slova. Je-li začátek slova tvořen hláskou s vysokou frekvencí, ale malou intenzitou (např. hlásky f, s apod.), projeví se to zvýšenou hodnotou středního počtu průchodů signálu nulou, takže počáteční hraniční bod bude dodatečně odvozen z překročení H_Z. Podobným způsobem je určován i konec slova. Jsou stanoveny dolní meze pro intenzitu D_M a pro střední počet průchodů signálu nulou D_Z. Konec slova je určen, jestliže obě odpovídající charakteristiky poklesnou pod D_M, popř. D_Z.



Obrázek 6 - Ilustrace určování hraničních bodů slova na základě průběhu funkce krátkodobé intenzity a středního počtu průchodů nulou (PSUTKA, 1995).

Hodnoty H_M, popř. D_M, je možné stanovit na základě experimentálního ověření takto:

$$H_M = M_S + 4\sqrt{M_D}, \tag{4.5}$$

$$D_M = M_S + 2\sqrt{M_D}, \tag{4.6}$$

kde M_S je střední hodnota intenzity šumu,

M_D je disperze od střední hodnoty intenzity šumu.

Střední hodnota intenzity šumu i její disperze jsou určeny přes vhodný počet mikrosegmentů v pauze mezi promluvami. Analogický je i výpočet H_Z a D_Z.

4.7 Frekvenční analýza

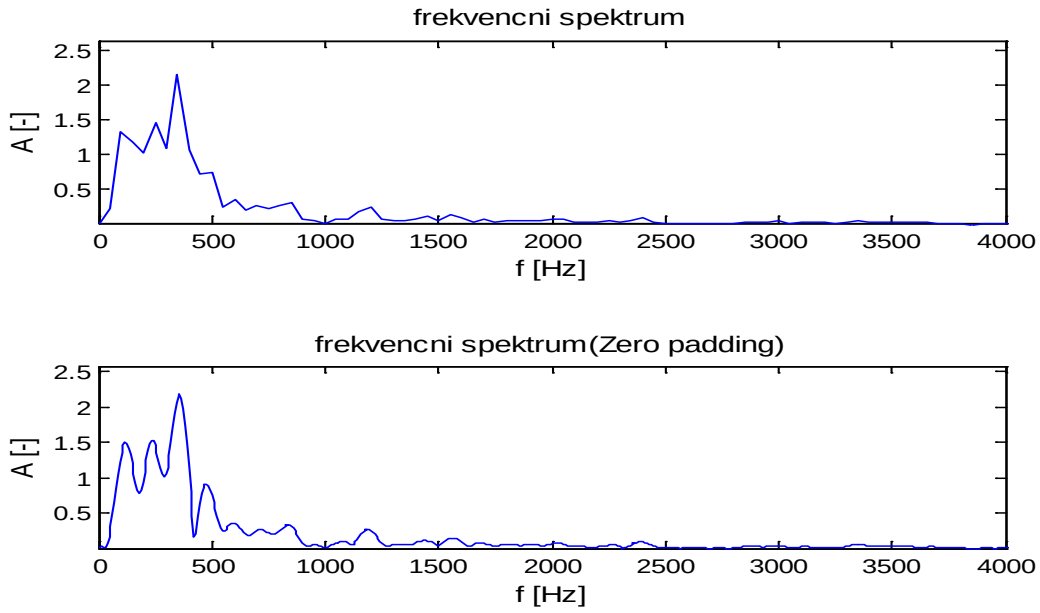
Operacemi v časové oblasti nedokážeme od signálu oddělit nepotřebné informace o řečníkovy. Pro oddělní nepotřebných informací, musíme signál převést do frekvenční oblasti. Využívá se Diskrétní Fourierovy transformace (DFT), která je dána vztahem:

$$X(k) = \sum_{n=0}^{N-1} x[n] e^{-j\,2\pi \frac{nk}{N}}, \qquad pro\ k\ \epsilon\ \langle 0, N-1 \rangle. \tag{4.7}$$

Spočítali jsme spektrum vzorkovaného signálu, které bude periodické a to s periodou rovnou vzorkovací frekvenci. Pokud budeme DFT aplikovat na jednotlivé rámce, projeví se nám ve spektru i to, jakým oknem jsme ze signálu rámec„vyřízli“, viz. (PSUTKA, 1995). Spočítané spektrum je diskrétní s hodnotami kmitočtů vzdálených o velikosti $f_v = F_S/N$. Jelikož je výsledné spektrum symetrické, stačí vykreslit jenom jeho první polovina.

4.7.1 Zero padding

Z předchozího odstavce víme, jak jsou od sebe vzdáleny jednotlivé vzorky spektra. Někdy ale potřebujeme, aby charakteristika byla přesnější, tzn., aby vzorků ve spektru bylo více. Toho se dá docílit použitím metody doplňování nul, tzv. „Zero padding“. Doplněním nul zvýšíme N, ale žádnou informaci do výpočtu spektra nepřidáme.



Obrázek 7 - Porovnání frekvenční charakteristiky znělého rámce bez a s využitím „Zero padding“.

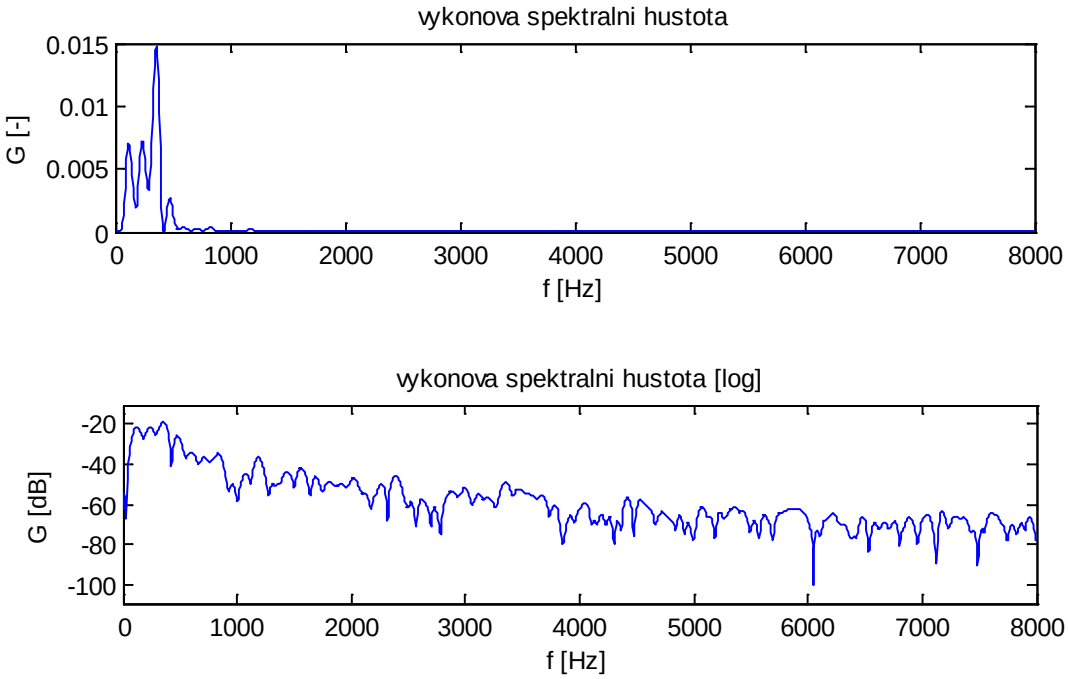
4.8 Výkonová spektrální hustota

Využívá se k analýze náhodného signálu a udává rozdělení výkonu ve frekvenční oblasti. Jeden z odhadů výkonové spektrální hustoty G využívá DFT:

$$G_{DFT}(k\Delta f) = \frac{1}{N}|X[k]|^2. \tag{4.8}$$

Na vzniklé charakteristice ovšem nelze z důvodu vysoké dynamiky pozorovat méně výrazné části, proto se výsledek logaritmuje a zobrazí v decibelech:

$$G_{DFT}(k\Delta f) = 10log_{10}\left(\frac{1}{N}|X[k]|^2\right). \tag{4.9}$$



Obrázek 8 - Výkonová spektrální hustota a její logaritmus.

5 Kepstrální koeficienty

Pro přesnější porovnávání slov je potřeba vzorky v rámci převést na kepstrální koeficienty. Metod pro vytvoření těchto koeficientů je více, zevrubně jsou popsány v (PSUTKA, a další, 2006). V této kapitole se budeme zabývat vytvořením Mel-frekvenčních kepstrálních koeficientů. Tato metoda je obecně velmi rozšířená, protože i přes svoji jednoduchost vede k výborným výsledkům.

5.1 Kepstrum

Řečový signál se dá rozdělit na dvě základní složky - buzení a modifikaci. Buzení je dáno základním tónem (frekvencí) řečníka, oproti tomu modifikaci ovlivňuje artikulační trakt. Pro rozpoznání řeči bez vlivu na typu řečníka buzení nepotřebujeme, a snažíme se proto buzení odstranit. To však není jednoduché, protože buzení je v podobě vyšších harmonických obsaženo v celém spektru. Otázka tedy zní, jak se zbavit buzení v řeči? Řečový signál je v časové oblasti dán konvolucí buzení $q(t)$ a modifikace $h(t)$:

$$s(t) = q(t) * h(t) = \int_{-\infty}^{\infty} g(\tau) \cdot h(t-\tau) d\tau. \tag{5.1}$$

Přeneseme-li signál pomocí DFT do kmitočtové oblasti, z konvoluce se stane součin:

$$S(f) = Q(f) \cdot H(f). \tag{5.2}$$

Bohužel ani v jedné oblasti od sebe nelze dobře složky oddělit. Problém vyřešíme, když uděláme spektrum spektra, tzv. kepstrum (ČERNOCKÝ, 2006). V kepstru se nyní z konvoluce stal součet. Na ose kepstrálních koeficientů se buzení vyskytuje na nižších hodnotách, kdežto modifikace na vyšších. Buzení lze tedy odstranit jednoduchým hornopropustným filtrem. Tento postup však nebere na vědomí fakt, že lidské ucho má na nižších frekvencích lepší rozlišení, než na frekvencích vyšších. Snažíme se tedy o podrobnější analýzu spektra na nižším kmitočtu.

5.2 Mel-frekvenční kepstrální koeficienty (MFCC)

Jak jsme si řekli, naše ucho rozpoznává spíše nižší frekvence, na což je potřeba při výpočtu koeficientů myslet. Metod, kde se klade větší důraz na nižší kmitočty je více. Velmi často se používá metoda s nelineárně rozloženou bankou filtrů ve spektru (MFCC). Tato metoda i přes svoji jednoduchost dosahuje výborných výsledků.

Při výpočtu signál nejdříve převedeme pomocí DFT do spektra, které umocníme a poté vynásobíme trojúhelníkovým filtrem a výsledné hodnoty sečteme. Takto získané energie musíme zlogaritmovat a diskrétní kosinovou transformací (DCT) z nich vypočítat MFCC koeficienty (obrázek 11). DCT je v tomto případě vhodnější náhrada inverzní Fourierovy transformace (využíváme symetrie spektra a toho že výsledek musí vyjít reálný (ČERNOCKÝ, 2006)). Největším problémem je tedy správné navržení filtrů. Z výše

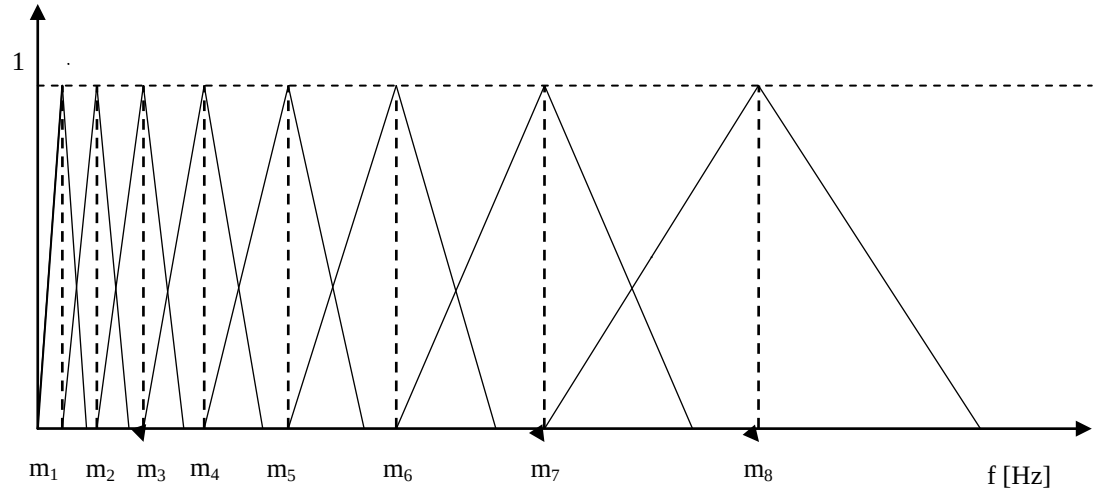
uvedených důvodů se snažíme, aby byly hustěji rozloženy v nižších frekvencích. Jejich počet je závislý na vzorkovací frekvenci, tabulka této závislosti je uvedena v (PSUTKA, a další, 2006). Při utváření banky filtrů můžeme nejdříve upravit frekvenční osu přepočtem Hertzů na Mely, a poté je rozmístit lineárně. Mely se z Hertzů vypočítají podle:

$$F_{Mel} = 2959log_{10}\left(1 + \frac{F_{Hz}}{700}\right). \tag{5.3}$$

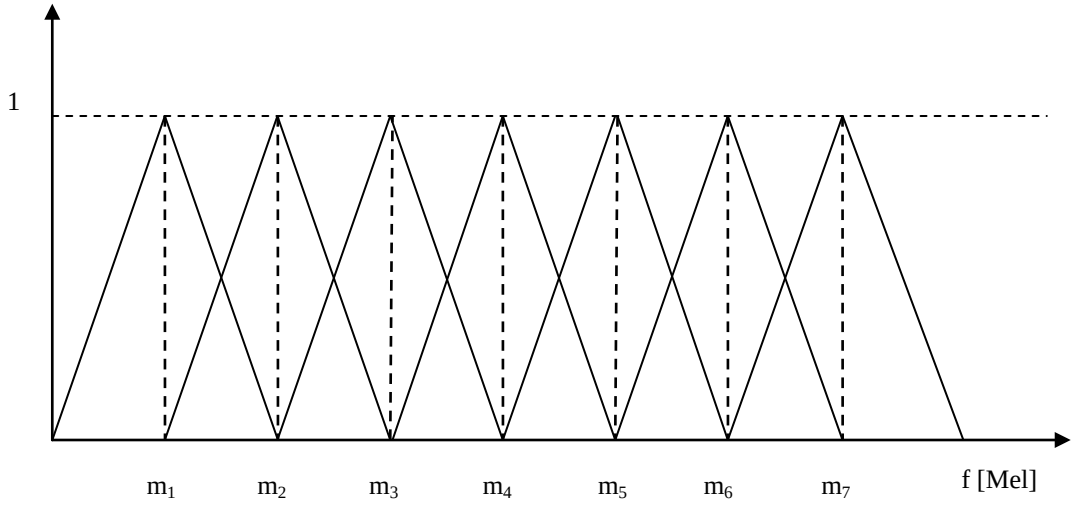
kde F_{Mel} je frekvence v Melech,

F_{Hz} je frekvence v Hertzích.

Všimneme si, že přepočet je vlastně zlogaritmování frekvenční osy. Rozmístění filtrů v závislosti na Hertzích a Melech zobrazuje obrázek 9 a obrázek 10.



Obrázek 9 - Rozmístění filtrů v závislosti na Hertzích.



Obrázek 10 - Rozmístění filtrů v závislosti na Melech.

Filtry jsou vlastně trojúhelníková okna s maximem v hodnotě jedna, kterými násobíme hodnoty spektra. Na okrajích jsou hodnoty spektra značně potlačeny, zatímco ve středu zůstávají takřka nezměněny. V intervalu každého okna hodnoty sečteme a umocníme na druhou a obdržíme energie e_k. Získané energie zlogaritmujeme a diskrétní kosinovou transformací získáme koeficienty MFCC, které jsou vhodné pro reprezentování zvuků.

$$c_m(n) = \sum_{k=1}^{K} log\, e_k cos\left[n(k-0.5)\frac{\pi}{K}\right], \qquad pro\ n \in \langle 0, M\rangle \tag{5.4}$$

kde K značí počet filtrů,

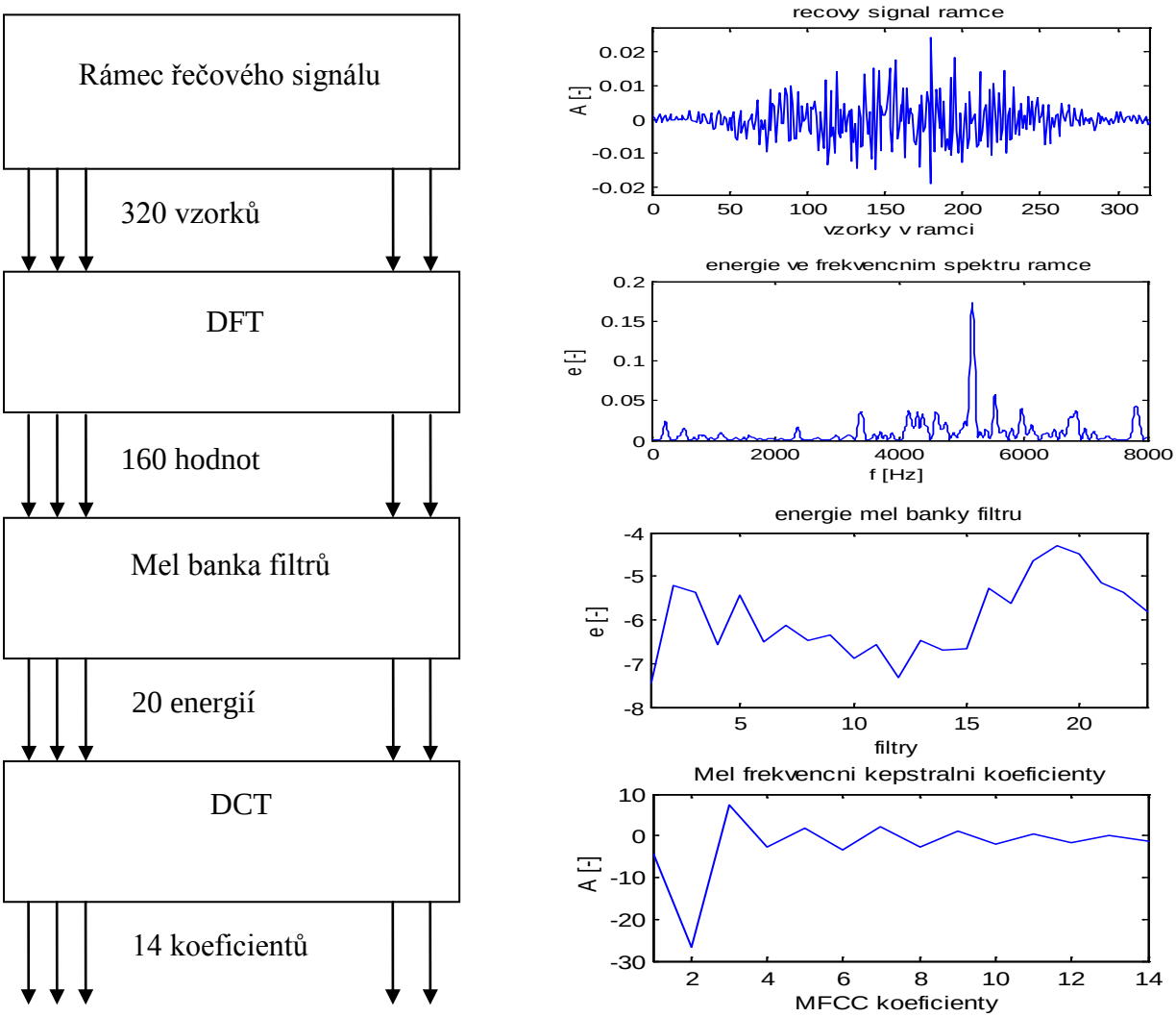
e_k je energie na intervalu daného filtru,

M je počet melovských kepstrálních koeficientů.

Počet koeficientů může být maximálně roven počtu filtrů, ale kdybychom využili všechny k rozpoznání řeči, výpočetní nároky by se značně zvedly. Jelikož je hlavní informace o zvuku obsažena v prvních několika koeficientech, tak bychom nedosáhli ani lepších výsledků. Běžně se používá 10 až 13 koeficientů. Velmi často se přidává na začátek ještě jeden koeficient, který je roven logaritmu krátkodobé energie přímo z řečového signálu rámce (PSUTKA, a další, 2006):

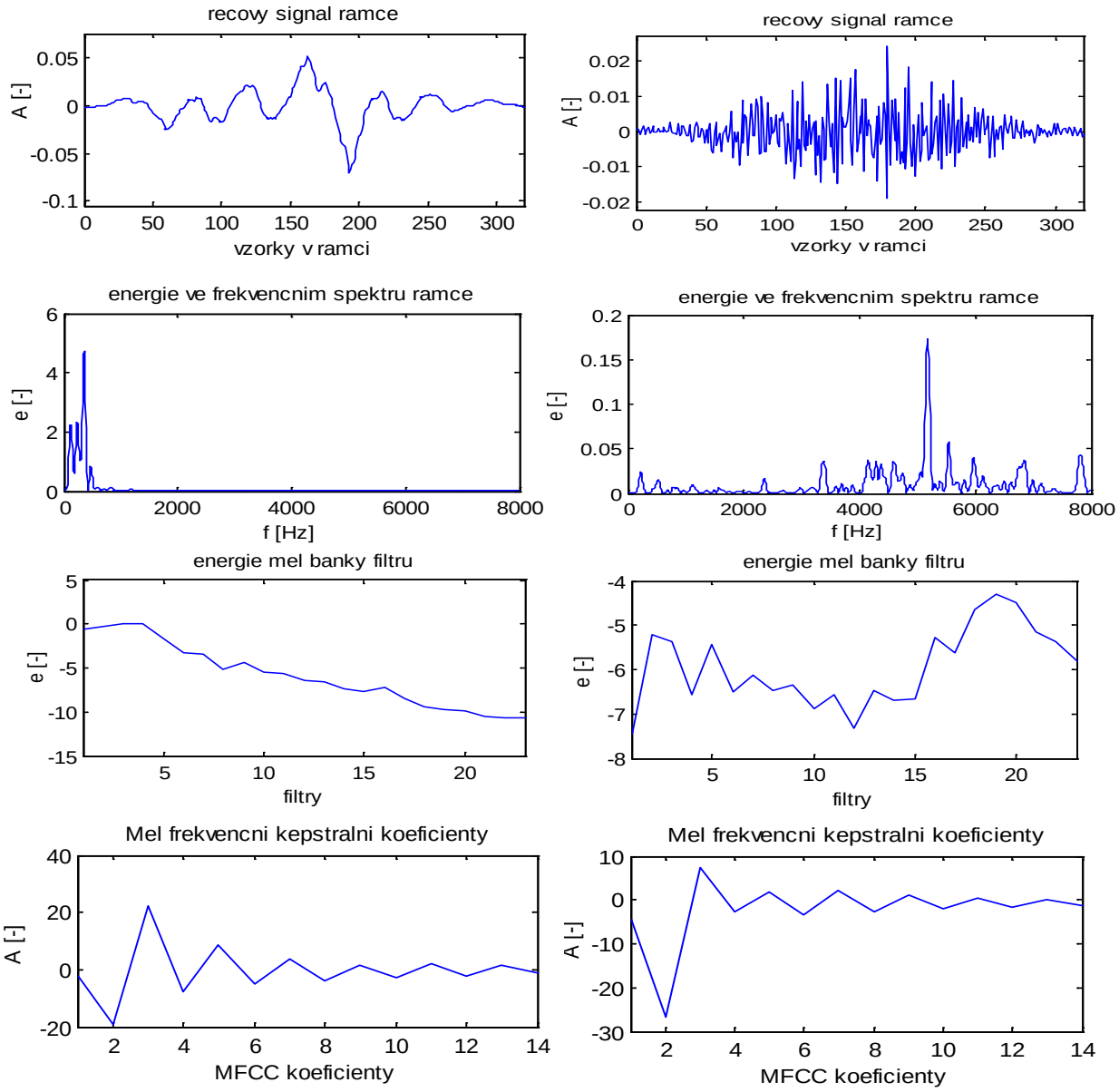
$$c_m(0) = log \sum_{n=0}^{L_{ram}-1} x^2\,[n], \tag{5.5}$$

Princip této metody je vždy stejný, ale tvarů filtru může být více. V diplomové práci je uvažován pouze základní trojúhelníkový filtr.



Obrázek 11 - Blokový postup výpočtu MFCC.

Vzniklé grafy reprezentují rámec obsahující sykavku „c“, což je typický představitel neznělé hlásky. Vidíme, že signál má větší energie na vyšších frekvencích. Za ukázku stojí i porovnání znělého a neznělého rámce, což ukazuje obrázek 12, kde v pravém sloupci je postup výpočtu MFCC pro neznělý rámec a v levém sloupci pro rámec znělý. Postup výpočtu koeficientů aplikujeme na každý rámec slova. Výsledkem bude jeho obraz, který bude vhodněji slovo reprezentovat při rozpoznávání.



Obrázek 12 - Porovnání postupu MFCC u znělého (vlevo) a neznělého (vpravo) rámce.

Jako znělý je vybrán rámec číslo 100, kde je obsažena hláska „í“ ze slova myslím. Neznělou hlásku reprezentuje rámec číslo 212, kde je obsažena hláska „c“ ze slova abeceda. Jenom při pohledu na rámce je vidět, že frekvence v neznělém rámci je mnohem vyšší. To je názorně dokázáno ve frekvenčním spektru rámce. Zajímavý je také pohled na výsledné kepstrální koeficienty, kde rozdíl mezi hodnotami prvních koeficientů je větší a naopak poslední koeficienty jsou si už velmi podobné. To je názorná ukázka, že více než 14 koeficientů pro reprezentaci rámce není potřeba.

6 Rozpoznávání řeči

O počítačové rozpoznání řeči se odborníci zajímají již 50 let. Přesto se rozpoznat libovolné slovo z mluvy neznámého řečníka stále dokonale nedaří. Důvody nezdarů se skrývají v obrovské variabilitě mluvčího, v prostředí, kde se záznam provádí, ale také v obtížnosti řešené úlohy. Každý člověk má originální hlasové ústrojí a odlišný způsob artikulace, to se projevuje rozdílnou barvou hlasu, přízvukem, rychlostí řeči atd. I hlas jednoho řečníka je variabilní a závislý na mnoha aspektech (otázka, příkaz, nálada, nemoc atd.). To se projevuje v délce jednotlivých úseků řeči i v intenzitě řečového signálu. Ve skutečnosti je vlastně nemožné, aby bylo slovo řečeno dvakrát naprosto stejně. Rozpoznávače řeči můžeme podle složitosti rozdělit do tří skupin (PSUTKA, a další, 2006):

- **Rozpoznávání izolovaných slov** (malý slovník, např. číslovky, povely).
- **Rozpoznávání diskrétního diktátu** (rozsáhlejší slovník, slova jsou vyslovována izolovaně s krátkou mezislovní pauzou).
- **Rozpoznávání souvislé řeči** (slovník na desítky tisíc slov).

Metod pro rozpoznávání je také více. Pracuje se s využitím statistických metod, kde jsou slova i celé promluvy modelovány pomocí tzv. skrytých Markovových modelů. Tato metoda je nejrozsáhlejší a je vhodná pro rozpoznávače s velkým slovníkem. Protože v mé práci se zabývám rozpoznáváním diskrétního diktátu s malým slovníkem, nebudu se o této metodě dále rozepisovat. Pro zájemce je metoda na bázi skrytých Markovových modelů dobře popsaná v (ČERNOCKÝ, 2006), (PSUTKA, a další, 2006).

6.1 Dynamické programování

Při vstupu signálu do této metody jsou slova reprezentována sekvencí vektorů, které budeme nazývat obrazem slova. Délka sekvence je dána délkou slova a velikost vektorů počtem kepstrálních koeficientů. Metoda dynamického programování je založena na porovnávání neznámého testovaného a referenčního obrazu:

$$\boldsymbol{O} = [\boldsymbol{o}(1), \dots, \boldsymbol{o}(i), \dots, \boldsymbol{o}(T)], \tag{6.1}$$

kde $\boldsymbol{o}(i)$ jsou vektory kepstrálních koeficientů testovaného obrazu.

$$\boldsymbol{R} = [\boldsymbol{r}(1), \dots, \boldsymbol{r}(j), \dots, \boldsymbol{r}(R)]. \tag{6.2}$$

kde $\boldsymbol{r}(j)$ jsou vektory kepstrálních koeficientů referenčního obrazu.

Referenční obrazy jsou rozdělené do tříd a neznámé obrazy se přidělí do té třídy, kde se nachází obraz, od kterého má nejmenší vzdálenost (součet odchylek jednotlivých vektorů slova). Základní odlišnosti mezi slovy ovšem nejsou ve spektrální části, ale v délce slov či vnitřních částech slova (hlásek). Problém se dá řešit lineární časovou normalizací obrazu, ta ale neovlivní rozdílné délky hlásek. Řešení přináší metoda

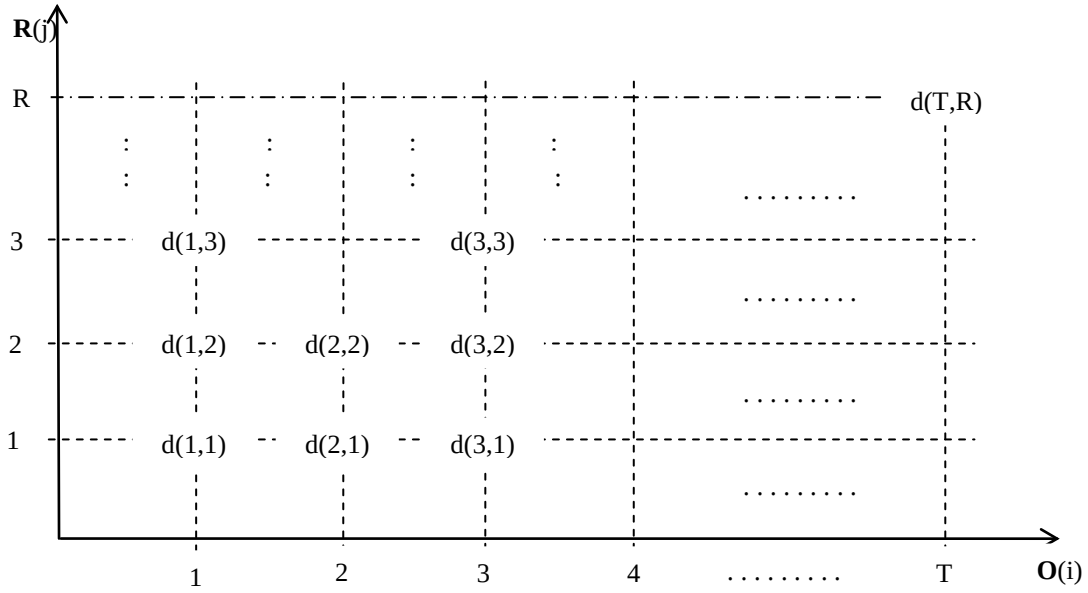
DTW (dynamic time warping), česky dynamické borcení času. Tato funkce minimalizuje rozdíly mezi dvěma obrazy borcením časové osy jednoho z nich.

6.2 DTW

Transformační funkce, která optimálně přizpůsobuje referenční slovo s testovaným, je dána pomocí lokálních vzdáleností mezi body obrazů v rovině (T, R). Při výpočtu funkce DTW většinou uvažujeme testované příznaky podél horizontální osy a referenční podél osy vertikální. Lokální vzdálenosti se můžou jednoduše určit:

$$d(\boldsymbol{o}, \boldsymbol{r}) = d\big(\boldsymbol{o}(i)\boldsymbol{r}(j)\big) = \sqrt{\sum_{n=1}^{M} |o_n(i) - r_n(j)|^2}, \tag{6.3}$$

Z těchto vzdáleností můžeme udělat matici o rozměrech $T \times R$.



Obrázek 13 - Matice vzdáleností DTW.

Nyní spočítáme cesty uprostřed matice, které vedou z počátku $d(1,1)$ do konce $d(T, R)$. Pro další výklad je vhodné zavést časovou proměnou $k = 1,2,3,\ldots,K$, kde K ukazuje délku porovnávací cesty a na ní závislé transformační funkce:

- referenční sekvenci cesty $r(k)$,
- a testovanou sekvenci cesty $t(k)$.

Pro referenční i testovaný obraz musí být délka cesty stejná, i když jsou oba obrazy rozdílné velikosti. Příklad cesty funkce DTW a aplikace proměnné k, ukazuje obrázek 15. Hledaná cesta je ta s minimální celkovou vzdáleností. Pro nalezení minima je nutno

prozkoumat všechny možné cesty. Některé cesty jsou z důvodu nereálných zkreslení zkoumaných slov předem vyloučeny.

6.2.1 Lokální omezení

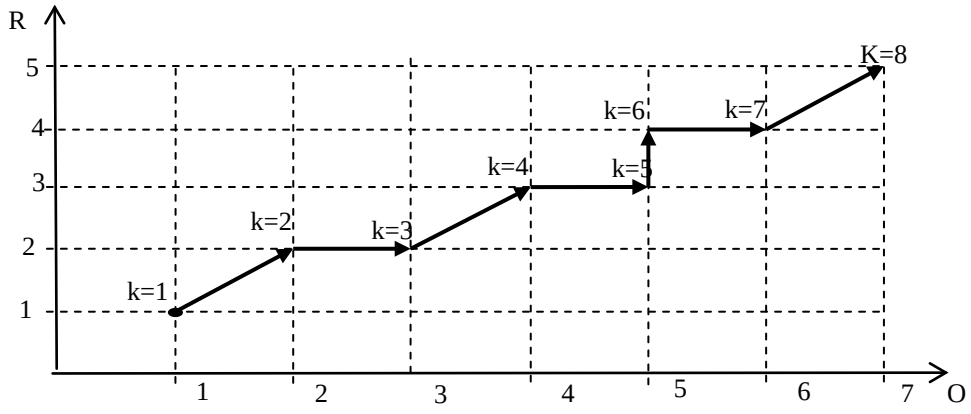
Abychom se vyhnuli nadměrné kompresi nebo expanzi časového měřítka u porovnávacích obrazů, musíme na funkci DTW aplikovat **omezení monotónnosti a souvislosti**.

$$0 \leq r(k) - r(k-1) \leq R^x, \tag{6.4}$$

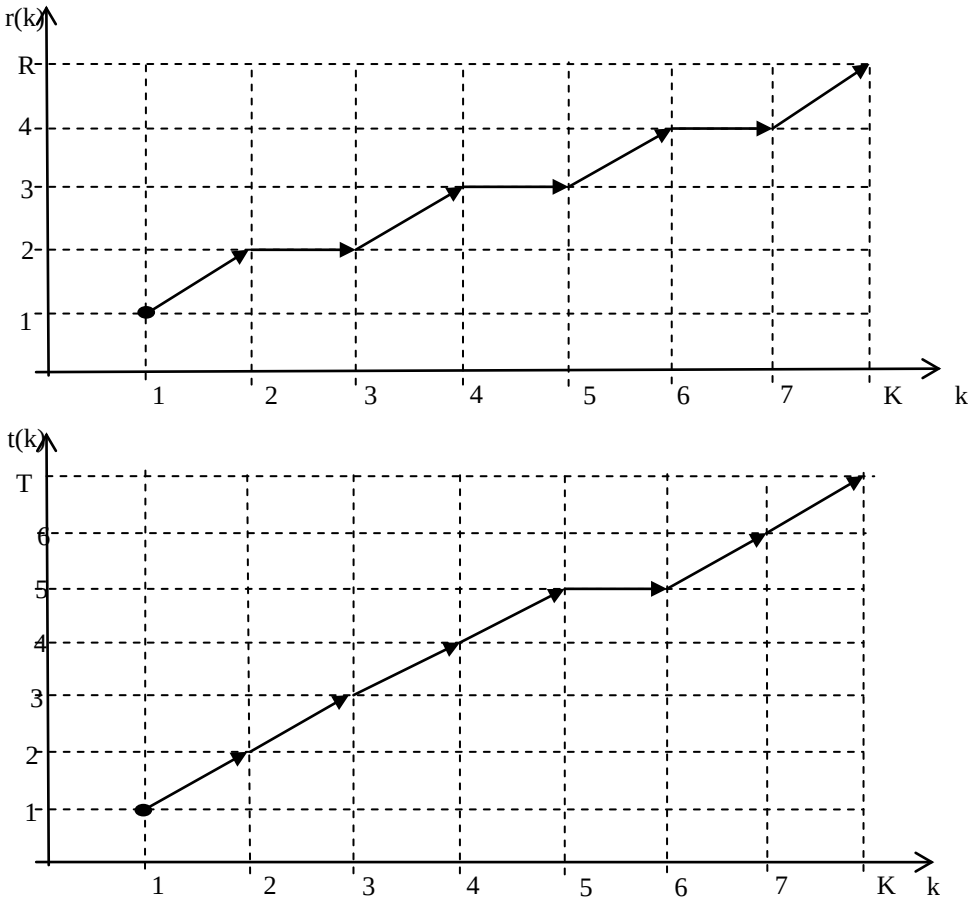
$$0 \leq t(k) - t(k-1) \leq T^x, \tag{6.5}$$

kde R^x, T^x jsou kladná celá čísla.

Jde vlastně o omezení délky kroku funkce DTW. Bude-li jedna z hodnot R^x nebo T^x rovna jedné, funkce nesmí v dané ose žádné segmenty vynechat. V praxi většinou necháváme funkci vynechávat za sebou maximálně dva segmenty. Problém může nastat i v případě, že bude funkce příliš strmá ve směru jedné z os. Proto se zavádí omezení strmosti. Pokud funkce bude postupovat ve směru pouze jedné osy n krát za sebou, není jí dovoleno ve směru pokračovat, dokud nepostoupí m krát ve směru jiném (PSUTKA, 1995). Tabulka 1 obsahuje typy lokálních omezení použitých v této práci. Tabulka nejčastějších lokálních omezení funkce DTW je uvedena v příloze A.



Obrázek 14 - Schematické znázornění funkce DTW.



Obrázek 15 - Ukázka aplikace proměnné *k* na funkci DTW.

6.2.2 Globální vymezení cesty

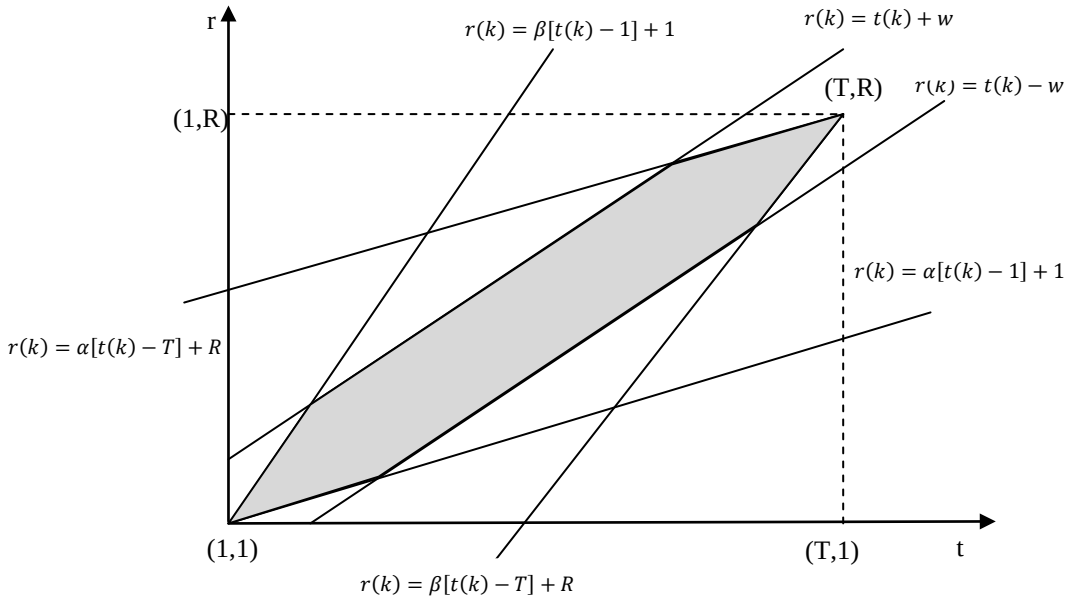
Aplikujeme-li lokální podmínky na celou rovinu $(T \times R)$, lze vymezit oblast přípustnou pro průchod cesty. Globální vymezení můžeme matematicky zapsat:

$$1 + \alpha[t(k) - 1] \leq r(k) \leq 1 + \beta[t(k) - 1], \tag{6.6}$$

$$R + \beta[t(k) - T] \leq r(k) \leq R + \alpha[t(k) - T], \tag{6.7}$$

kde α, β jsou směrnice přímky vymezující přípustnou oblast cesty.

Další omezení vyplývá z předpokladu, že při porovnávání dvou obrazů, které reprezentují stejné slovo, nemůže dojít k zásadním časovým rozdílům mezi příslušnými úseky obou obrazců. Proto můžeme definovat w jako vhodné celé číslo, které vytyčí maximální vzdálenost jednotlivých úseků porovnávaných obrazců (PSUTKA, 1995). Omezení globální cesty znázorňuje obrázek 16.



Obrázek 16 - Globální vymezení pohybu funkce DTW (ČERNOCKÝ, 2006).

6.2.3 Výpočet vzdálenosti

Každá cesta C je jednoznačně dána svou délkou K_c, a průběhem funkcí $t_c(k)$ a $r_c(k)$. Pro tuto cestu se vzdálenost mezi obrazy $\boldsymbol{O}$ a $\boldsymbol{R}$ vypočítá jako:

$$D_c(\boldsymbol{O}, \boldsymbol{R}) = \frac{1}{N_c(\widehat{W})} \sum_{k=1}^{K_c} d\left[\boldsymbol{o}\left(t_c(k)\right), \boldsymbol{r}\left(r_c(k)\right)\right] \widehat{W}_c(k), \tag{6.8}$$

kde $d[\boldsymbol{o}(\dots), \boldsymbol{r}(\dots)]$ je vzdálenost dvou vektorů,

$\widehat{W}_c(k)$ je váha odpovídající k-tému kroku cesty,

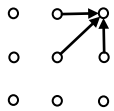
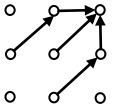
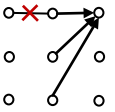
$N_c(\widehat{W})$ je normalizační faktor závislý na vahách.

Vzdálenost obrazů $\boldsymbol{O}$ a $\boldsymbol{R}$ je dána jako minimální vzdálenost ze všech možných cest:

$$D(\boldsymbol{O}, \boldsymbol{R}) = \min_{\{c\}} D_c(\boldsymbol{O}, \boldsymbol{R}). \tag{6.9}$$

Podle typu lokálního omezení, využíváme čtyři druhy váhových funkci $\widehat{W}(k)$ a normalizačních faktorů $N(\widehat{W})$.

Tabulka 1 - Lokálních omezení použité v práci (PSUTKA, 1995).

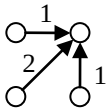
Typ funkce DTW		α	β	Typ W(k)	$g(n,m)$
I		0	∞	a	$min\begin{Bmatrix} g(n,m-1) + d(n,m) \\ g(n-1,m-1) + 2d(n,m) \\ g(n-1,m) + d(n,m) \end{Bmatrix}$
				d	$min\begin{Bmatrix} g(n,m-1) + d(n,m) \\ g(n-1,m-1) + d(n,m) \\ g(n-1,m) + d(n,m) \end{Bmatrix}$
II		1/2	2	a	$min\begin{Bmatrix} g(n-1,m-2) + 2d(n,m-1) + d(n,m) \\ g(n-1,m-1) + 2d(n,m) \\ g(n-2,m-1) + 2d(n-1,m) + d(n,m) \end{Bmatrix}$
III		1/2	2	b1	$min\begin{Bmatrix} g(n-1,m) + \kappa d(n,m) \\ g(n-1,m-1) + d(n,m) \\ g(n-1,m-2) + d(n,m) \end{Bmatrix}$ $\kappa = 1$ pro $j(k-1) \neq j(k-2)$ $\kappa = \infty$ pro $j(k-1) = j(k-2)$

6.2.4 Váhová funkce

Typy funkcí:

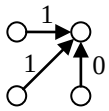
a) symetrická váhová funkce

$$\widehat{W}(k) = [t(k) - t(k-1)] + [r(k) - r(k-1)], \tag{6.10}$$

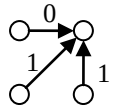


b) asymetrická váhová funkce

b1) $\widehat{W}(k) = t(k) - t(k-1)$, (6.11)



b2) $\widehat{W}(k) = r(k) - r(k-1)$, (6.12)



c) váhová funkce minimální vzdálenosti kroku

$$\widehat{W}(k) = min\{t(k) - t(k-1), r(k) - r(k-1)\}, \tag{6.13}$$

d) váhová funkce maximální vzdálenosti kroku

$$\widehat{W}(k) = max\{t(k) - t(k-1), r(k) - r(k-1)\}. \tag{6.14}$$

6.2.5 Normalizační faktor

Tento faktor je zaveden, aby kompenzoval délku cesty funkce DTW. Je závislý na váhové funkci a můžeme ho spočítat jako (ČERNOCKÝ, 2006):

$$N(\widehat{W}) = \sum_{k=1}^{K} \widehat{W}(k). \tag{6.15}$$

Pro jednotlivé váhovací funkce přiřazujeme tyto faktory:

a) $N(\widehat{W}_a) = \sum_{k=1}^{K}[t(k) - t(k-1) + r(k) - r(k-1)] = T + R$ (6.16)

b)

b1) $N(\widehat{W}_{b1}) = \sum_{k=1}^{K}[t(k) - t(k-1)] = t(K) - t(0) = T,$ (6.17)

b2) $N(\widehat{W}_{b2}) = \sum_{k=1}^{K}[r(k) - r(k-1)] = r(K) - r(0) = R,$ (6.18)

Pro váhové funkce typu c) a d) se osvědčila volba faktoru nezávisle na průběhu funkce DTW:

$$N(\widehat{W}_c) = N(\widehat{W}_d) = T. \tag{6.19}$$

6.3 Postup zjištění optimální cesty funkce DTW

Rozhodnutí o průběhu optimální cesty se nemusí dělat až na konci, ale můžeme postupovat minimální cestou již od začátku. Víme, z jakých předchozích bodů se do toho současného můžeme dostat (lokální omezení), proto můžeme od začátku vybírat jenom ty nejlepší varianty pro jednotlivé body. Na konci obou obrazců v bodě (T, R), je pak k dispozici velikost cesty s nejmenší vzdáleností. Postup této metody je následující (ČERNOCKÝ, 2006):

1. Vytvoříme mřížku $\boldsymbol{d}$ o rozměrech $T \times R$, do které zapíšeme vzdálenosti jednotlivých testovacích a referenčních vektorů (Obrázek 13).
2. Vytvoříme další mřížku $\boldsymbol{g}$ částečných kumulovaných vzdáleností, která má k mřížce $\boldsymbol{d}$ navíc nultý řádek a nultý sloupec, které inicializujeme na:

$$g(0,0) = 0,$$

$$g(0, m \neq 0) = g(n \neq 0, 0) = \infty.$$

3. Částečnou kumulovanou vzdálenost vypočítáme pro každý bod takto:

$$g(m,n) = \min_{\forall \text{předchůdci}} \left[g(\text{předchůdce}) + d(m,n)\widehat{W}(k)\right], \tag{6.20}$$

- možní předchůdci jsou dáni pomocí tabulky lokálních omezení cesty.

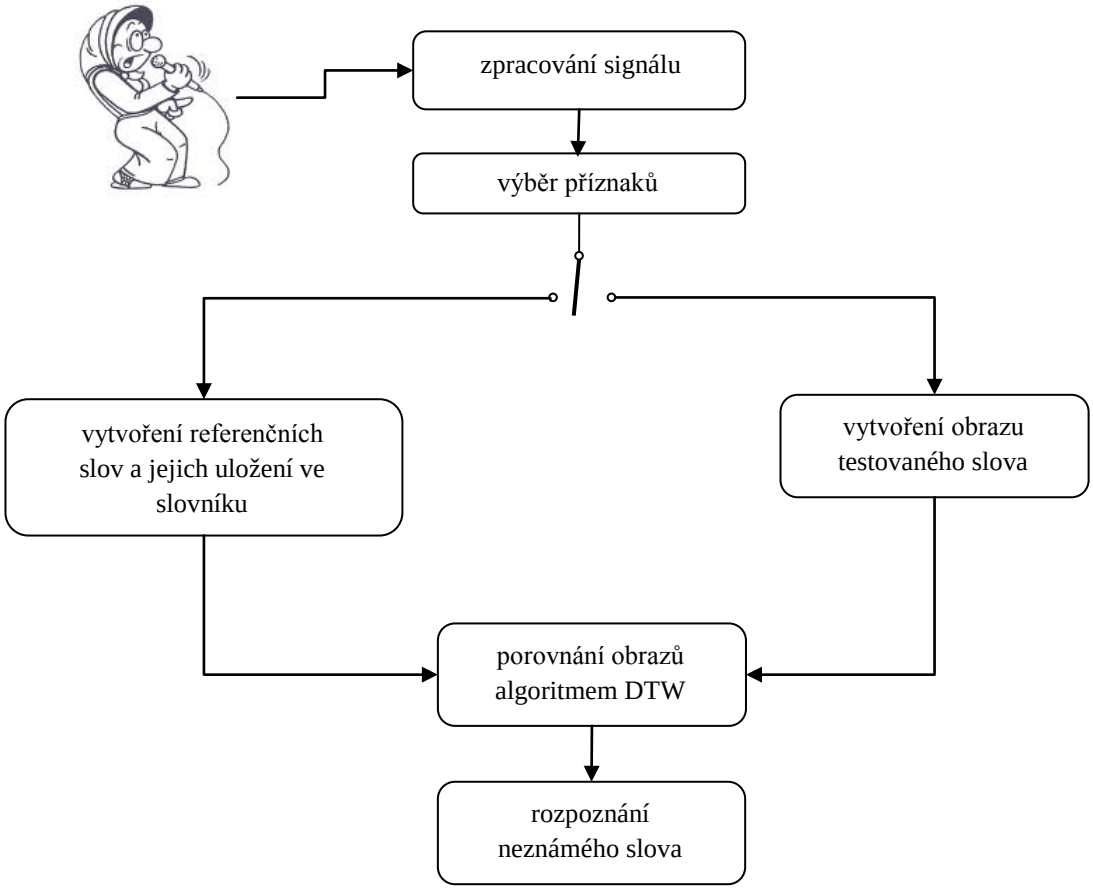
- Váha $\widehat{W}(k)$ odpovídá pohybu z předchůdce do bodu $[m, n]$.
- Vztahy pro výpočet částečné kumulované vzdálenosti pro lokální omezení užitých v práci jsou uvedeny v Tabulce 1.

4. Konečná minimální vzdálenost mezi dvěma obrazy je pak dána:

$$D(\boldsymbol{O}, \boldsymbol{R}) = \frac{1}{N(\widehat{W})} g(T, R). \tag{6.21}$$

6.4 Praktická realizace klasifikátoru slov

Blokové schéma typického klasifikátoru izolovaných slov ukazuje obrázek 17. Z obrázku je patrné, že klasifikátor pracuje ve dvou režimech činnosti. V režimu trénování jsou na základě trénovací množiny vytvářeny a uchovávány referenční obrazy slov slovníku, kdežto v režimu klasifikace dochází aplikováním algoritmu DTW k porovnání testovaného obrazu s referenčními obrazy uloženými ve slovníku s následnou klasifikací obrazu testovaného slova. V dalších odstavcích si vysvětlíme podrobněji náplň těchto dvou režimů.



Obrázek 17 - Blokové schéma typického klasifikátoru slov.

6.4.1 Trénování

V tomto režimu činnosti každý řečník (pro systém, který je trénován na hlas řečníka) nebo skupina řečníků (pro systém nezávislý na hlasu řečníka) vysloví postupně každé slovo požadovaného slovníku (jednou nebo vícekrát). Po vyslovení jednotlivého slova dojde ke zpracování vstupního akustického signálu snímaného mikrofonem, a to nejčastěji využitím pulsní kódové modulace. Aplikací vybrané metody krátkodobé analýzy pak dochází k popisu každého mikrosegmentu snímaného signálu příznaky, kterými mohou být např. hodnoty výstupů pásmových filtrů, kepstrální koeficienty, apod. Důležitým krokem dalšího zpracování je detekce hraničních bodů slova, to znamená nalezení co možná nejpřesnějšího okamžiku začátku a konce slova. Tento krok je náročný na provedení, a to zejména při rušivém pozadí (šum, hluk, vzdálená řeč apod.). Následující krok v režimu trénování je spojen s vytvářením referenčních obrazů. Existuje v zásadě několik metod konstrukce referenčních obrazů, tedy obrazů, které budou reprezentovat jednotlivé třídy v procesu klasifikace (PSUTKA, 1995).

a) Přímé využití obrazů trénovací množiny jako referenčních obrazů.

Řečník nebo skupina řečníků namluví slova ze slovníku jednou nebo i vícekrát a všechny takto vzniklé akustické obrazy jsou využity jako referenční vzory. Metoda dynamického programování nevyžaduje, aby obraz každého slova, který se bude porovnávat algoritmem DTW, byl stejně dlouhý, tj. byl tvořen stejným počtem příznaků.

b) Vytváření průměrného vzorového obrazu pro každou třídu ω_r.

Při vytváření průměrného vzorového obrazu se využívá nejčastěji metoda lineárního průměrování nebo metoda dynamického průměrování. Při lineárním průměrování se předpokládá, že provedeme lineární časovou normalizaci všech akustických obrazů trénovací množiny. U dynamického průměrování se vytváří vzorový obraz využitím algoritmu DTW.

c) Vytváření vzorových obrazů shlukováním

Předpokládejme, že pro třídu ω_r existuje celkem S_r trenovacích obrazů. Rozdělíme těchto S_r obrazů do P_r shluků tak, že obrazy uvnitř jednoho shluku jsou si „podobné“ (mají malou vzájemnou vzdálenost), kdežto obrazy mezi shluky si jsou „nepodobné“. Takový shlukovací proces se dá realizovat různými postupy, které mohou být buď interaktivní nebo automatické, které jsou většinou odvozeny z MacQueenova[1] shlukovacího algoritmu.

6.4.2 Klasifikace

V režimu klasifikace probíhá proces zpracování řečového signálu, výběr příznaků i vytvoření obrazu testovaného slova stejným způsobem jako při zpracování množiny slov v režimu trénování. Pokud jsou referenční obrazy ve slovníku uloženy s normalizovanou délkou, musíme i testovaný obraz normalizovat.

[1] MacQueenův algoritmus je také v anglické literatuře označován jako k-means algorithm.

Při klasifikaci slov metodou dynamického programování se prioritně využívá především pravidlo minimální vzdálenosti a z něj odvozená dvě rozhodující pravidla – pravidlo nejbližšího souseda (Nearest Neighbour rule, zkr. NN rule) a pravidlo S-nejbližšího souseda (S-NN rule). Pokud je každé slovo reprezentováno jediným vzorovým obrazem$\boldsymbol{B_r}$ (vzorový obraz získán například průměrováním obrazů trénovací množiny třídy ω_r anebo namluvením pouze jediného vzorového obrazu), zařadí klasifikátor obraz $\boldsymbol{O}$ neznámého slova do té třídy ω_{r^*}, pro kterou platí:

$$\omega_{r^*} = arg \min_r [D(\boldsymbol{O}, \boldsymbol{B_r})], \tag{6.22}$$

kde $r = 1, \dots, R$ a R je počet tříd.

V případě, že každé slovo je reprezentováno více referenčními obrazy B_{rs} (vzorové obrazy byly získány buď prostým namluvením anebo shlukovacím procesem), pak klasifikaci podle pravidla NN lze uskutečnit pomocí vztahu:

$$\omega_{r^*} = arg \min_{r,s} [D(\boldsymbol{O}, \boldsymbol{B_{rs}})], \tag{6.23}$$

kde $s = 1, \dots, P_r$ a P_r je počet vzorových obrazů ve třídě ω_r.

Lepší výsledky klasifikace než při aplikaci pravidla nejbližšího souseda dává většinou v obdobných případech rozhodovací pravidlo S-NN. Při aplikaci pravidla S-NN vyčíslíme nejprve všechny vzdálenosti $D(\boldsymbol{O}, \boldsymbol{B_{rs}})$ pro $r = 1, \dots, R$ a $s = 1, \dots, P_r$ a pak je pro každé r uspořádáme podle velikosti od nejlepší po nejhorší:

$$D\left[\boldsymbol{O}, \boldsymbol{B_{r[1]}}\right] \leq D\left[\boldsymbol{O}, \boldsymbol{B_{r[2]}}\right] \leq \cdots \leq D\left[\boldsymbol{O}, \boldsymbol{B_{r[P_r]}}\right].$$

O klasifikaci obrazu $\boldsymbol{O}$ neznámého slova do třídy ω_{r^*} pak rozhodneme podle minima průměrné vzdálenosti k-nejbližších sousedů

$$\omega_{r^*} = arg_r \min \frac{1}{S} \sum_{s=1}^{S} D\left[\boldsymbol{O}, \boldsymbol{B_{r[s]}}\right]. \tag{6.24}$$

V praktických aplikacích, kdy počet vzorových obrazů $P_r \approx 6 - 12$, je pro pravidlo S-NN postačující volba $S = 2 - 4$. Vyšší hodnoty s již ke zlepšení klasifikace výrazně nepřispívají, naopak klasifikační proces zpomalují (PSUTKA, 1995).

7 Využití systémů v praxi

Systémy, které využívají rozpoznávání slov, jsou na trhu velmi rozšířeny. Mezi nejznámější určitě patří hlasové vytáčení v mobilním telefonu. Tyto systémy se využívají k rozpoznání omezeného slovníku slov, které musíme systém před použitím naučit. Je pro něj také problém rozpoznat povel v hlučném prostředí, nebo správně rozlišit více podobných příkazů. Obecně můžeme říci, že jednoduché systémy pro rozpoznávání slov fungují tím lépe, čím rozdílnější si navzájem budou referenční slova a čím tišší bude okolní prostředí. V následujících dvou podkapitolách budou popsány dva komerčně dostupné systémy pro rozpoznání řeči, uvedené popisy systémů vycházejí z informací distributora.

7.1 Voice Me

Voice Me je zařízení, které umožňuje uživatelům vydávat hlasové příkazy pro většinu domácích přístrojů fungujících na dálkové ovládání. Tento výrobek od společnosti Hotech poslouchá naše vyřčené rozkazy a přání, mění je na infračervené paprsky s příslušnými kódy, jimiž pak ovládá ovládané komponenty. Výuka Voice Me je jednoduchá, integrovaný ženský hlas napovídá v každém kroku učení. Nejdříve zvolíme oslovení (např. „Poslouchej mě“), kterým budeme ovladač uvádět do stavu bdělosti. Dokud zařízení takto neoslovíme, bude ignorovat všechny zvuky v místnosti a nedojde tak k nechtěným operacím. Pak začneme programovat až třicet jednotlivých pokynů. Nejdříve vyslovíme příkaz (např. „Přehraj CD“) a po jeho kontrolním zopakování namíříme ovladač od CD přehrávače na vysílací část Voice Me (viz obrázek 18) a zmáčkneme tlačítko Play. Stejně postupujeme i u ostatních funkcí. Zvolený příkaz by měl mít několik slabik, je pak lépe rozpoznatelný. Jednotlivé příkazy rozeznává Voice Me spolehlivě, občas je však nutné některé zopakovat. Může dojít i ke špatnému rozpoznání příkazu. Opravdové problémy nastanou v případě, kdy je v místnosti hlučněji. Na oválné krabičce jsou čtyři poosvětlená tlačítka. Jedno je určeno pro učení, druhé slouží pro okamžité ruční ztlumení zvuku (až tří přístrojů), třetím listujeme jednotlivými příkazy a posledním je můžeme mazat. Voice Me má mít jenom jednoho tzv. velitele (NÝVLT, 2003).

Obrázek 18 - Ukázka použití zařízení Voice Me (TopReklama).

Funkce:

- Hlasová aktivace.
- 30 hlasových příkazů.
- Jedním příkazem je možno zapnout 3 různé přístroje.
- Všesměrové emitování infračerveného signálu pro snadné dosažení přístrojů.
- Hlasový návod při nahrávání příkazů.
- Rozpoznání slovního řetězce přikazujícího spouštění a hlasových příkazů.

Specifikace:

- Vzdálenost pro příjem infračerveného signálu: 1 m.
- Mezní vzdálenost pro vstup řeči: 5 m.
- Vymezení úhlu pro příjem IR signálu: $\pm 45°$.

7.2 Hlasem řízený sklad K.voice

Systém K.voice umožňuje rychlejší, spolehlivější, produktivnější a bezpečnější řízení skladu. Způsob komunikace systému s uživatelem je určen tzv. šablonou dialogu, kterou lze připravit na míru podle typu hlasem řízené operace. Šablona určuje, jaké informace budou uživateli pomocí hlasu interpretovány a naopak jaké odpovědi uživatele jsou očekávány (tedy kterým slovům systém „rozumí“).

Obrázek 19 - Systém K.voice pro skladové využití (KODYS).

Páteří systému K.voice je serverová aplikace a speciální hlasové terminály, které tato aplikace prostřednictvím bezdrátové sítě řídí. Vzdálená správa hlasových terminálů je zajištěna internetovou aplikací, která umožňuje sledovat v reálném čase stav každého terminálu a umožňuje i změnu konfigurace jednotlivých terminálů. Nejčastěji jsou využívány hlasové terminály Talkman firmy Vocollect, které jsou vybaveny náhlavní sadou (sluchátka a mikrofon), pomocí které uživatel s terminálem komunikuje. Terminály

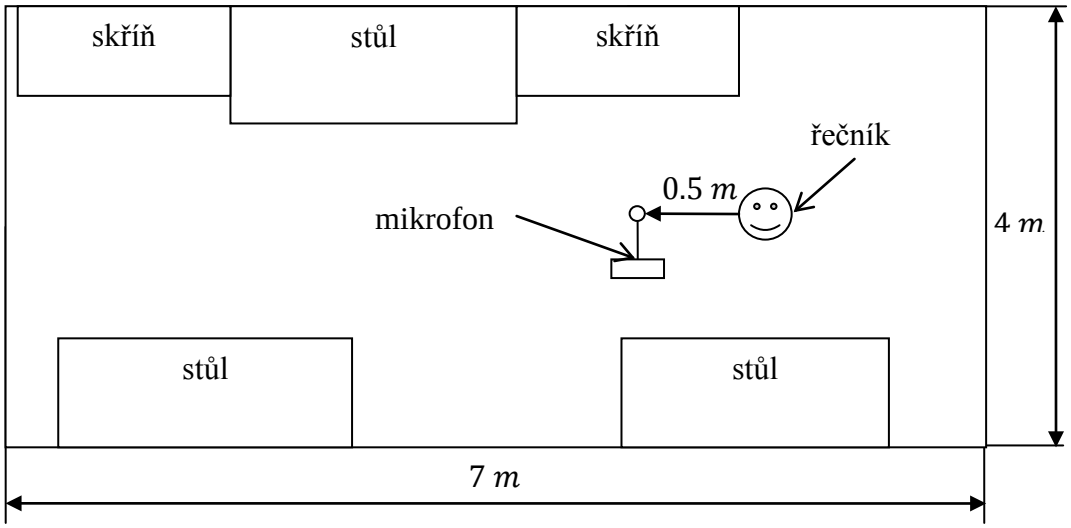
Talkman jsou vyvinuty speciálně pro hlasové aplikace a jsou k tomu tedy velmi dobře uzpůsobeny. K jejich výhodám patří speciální mikrofon, který dokáže filtrovat zvuky na pozadí, robustní a zároveň ergonomické provedení a v neposlední řadě kapacita akumulátoru, který dokáže zajistit chod terminálu celou pracovní směnu. Systém K.voice je primárně určen k řízení skladových operací, zejména vychystávání, zaskladňování a inventury. Jeho použití má značný přínos zejména ve skladech, kde se vychystává zboží ručně (bez využití manipulační techniky), nebo tam, kde by kvůli prostředí byla manipulace s „klasickým“ mobilním terminálem ztížena či znemožněna (sklady chladíren, mrazíren apod.). K.voice lze však s výhodou použít i v jiných oblastech než je sklad. Typickým příkladem může být použití při kontrole a řízení výroby, kdy uživatel hlasem potvrzuje provedení sekvence určitých činností (KODYS).

8 Zaznamenání řeči

V této kapitole je popsáno prostředí, kde se nahrávání všech referenčních i testovaných promluv odehrávalo. Jsou zde popsány přístroje a mikrofon, které byly pro záznam řeči využity.

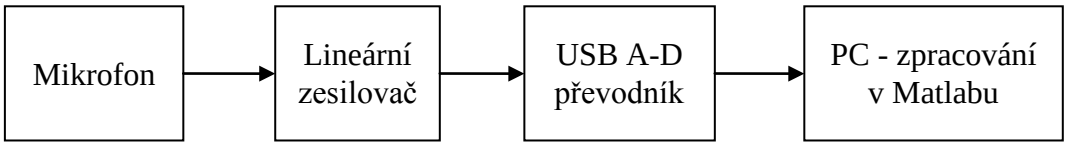
8.1 Pracoviště pro vytváření záznamů

Všechny zvukové záznamy, jež jsou v práci použity, byly získány v laboratoři o rozměrech 4 x 7 metrů. Prostředí bylo záměrně zvoleno tak, aby docházelo alespoň k částečnému rušení odrazy a provozem v okolí laboratoře. Řečník byl od mikrofonu vzdálen přibližně 0,5 metru a mluvil přímo k mikrofonu.



Obrázek 20 - Schéma laboratoře pro nahrávání řeči.

Blokové schéma záznamového řetězce ukazuje, jaké komponenty byly při záznamu řeči využity. Jednotlivé části řetězce jsou popsány níže.



Obrázek 21 - Blokové schéma záznamového řetězce.

8.1.1 Mikrofon

Pro záznam řeči byl použit běžný kapacitní mikrofon s malou impedancí MCD100 zobrazen na obrázku 22. Tento mikrofon má hyperkardiodní směrovou charakteristiku. Další parametry jsou:

- napěťový rozsah je 0 – 10V,
- Standardní napětí je 1,5V,

- Odstup signál/šum je více než 60dB.

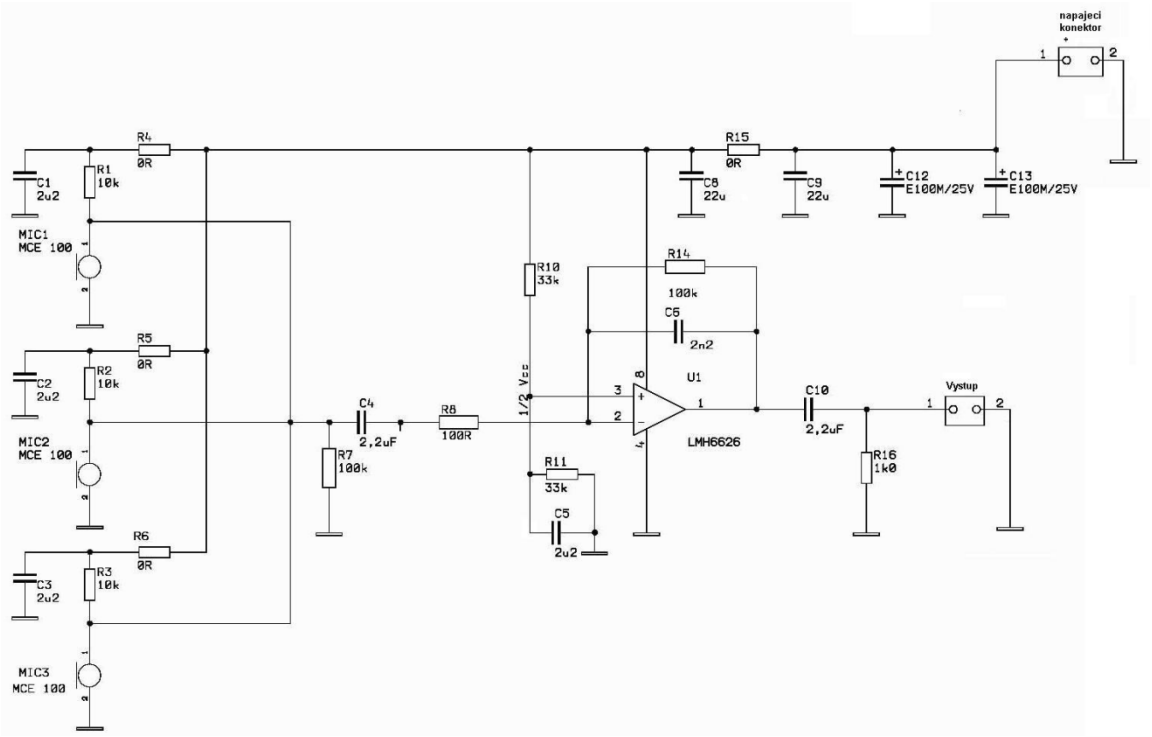
Obrázek 22 - Obrázek mikrofonu.

Citlivost tohoto mikrofonu je znázorněna graficky na obrázku 23. Běžně se pro rozpoznávání řeči akustický signál vzorkuje frekvencí 8 nebo 16 kHz. V této práci je zaznamenaná promluva vzorkována frekvencí 16 kHz a z toho můžeme odvodit, že důležité prvky řeči pro rozpoznání mají frekvenci do 8 kHz. Pro záznam řeči je tento mikrofon dostačující.

Obrázek 23 - Frekvenční odezva mikrofonu.

8.1.2 Lineární zesilovač

Hlavní částí lineárního zesilovače je obvod LMH6626, což je nízko šumový širokopásmový zesilovač, který je napájen napětím $\pm 3{,}3V$. Obvod se vyznačuje malým proudovým a napěťovým offsetem. Na výstupu lineárního zesilovače je umístěna horní propust s mezním kmitočtem 70 Hz. Schéma zapojení lineárního zesilovače je uvedeno na obrázku 24.



Obrázek 24 - Schéma lineárního zesilovače.

8.1.3 A-D převodník

Další částí záznamového řetězce je analogově digitální USB převodník NI USB – 6210 od firmy National Instruments. Tento 16-ti bitový převodník pracuje s maximální vzorkovací rychlostí 250kS/s na jeden kanál. Převodník NI USB – 6210 má:

- 16 analogových vstupů,
- 8 diferenciálních vstupů,
- 4 digitální vstupy,
- 4 digitální výstupy,
- 2 čítače s 32 bitovým rozlišením.

Vstupní rozsah převodníku je ± 10V, vstupní odpor při zapnutém zařízení je větší než 10GΩ s paralelním kondenzátorem o hodnotě 100pF.

Data z analogově digitálního převodníku jsou zpracovávána v prostředí Matlab.

9 Praktické řešení v Matlabu

Jedním z cílů diplomové práce je vytvoření programu v prostředí Matlab Program má být schopen rozpoznávat referenční slova v diskrétním diktátu a byl vypracován pro konkrétní zadání, nelze ho tedy bez úprav využít na jiné úkoly. Celý program můžeme rozdělit do tří skriptů. Jeden z nich je hlavní, probíhá v něm celý proces předzpracování i rozpoznávání slov. Ostatní se týkají úpravy signálu a referenčních slov před vložením do hlavního skriptu. Hlavní skript se skládá hlavně z vlastních funkcí, těch je dohromady vytvořeno osm.

Některé parametry u procesu předzpracování signálu nebo rozpoznávání slov musí být určeny. Tabulka 2 příklady těchto parametrů znázorňuje i s hodnotami, které byly v práci zvoleny. Hodnoty parametrů byly vybrány většinou na základě doporučení ve studované literatuře nebo z empirických zkušeností.

Tabulka 2 - Hodnoty parametrů stanovených v práci.

Vzorkovací frekvence	$F_S = 16\ kHz$
Délka rámce	$L_{ram} = 20\ ms$
Délka překrytí rámce	$P_{ram} = 10\ ms$
Počet Mel filtrů	20
Počet MFCC	14
Počet nejbližších sousedů	$S = 5$
Min. velikost slova	37 $rámc$ů
Max. velikost slova	150 $rámc$ů
Počet rámců, pro určení intenzity šumu	30

9.1 Funkce v matlabu

V této podkapitole budou popsány funkce naprogramované pro účely zpracování a rozpoznání slov v diskrétním diktátu.

9.1.1 Funkce frame

Tato funkce má za úkol rozdělit signál na rámce. Jako vstupní parametry musí být kromě samotného signálu definovány také délka rámce L_{ram}, překryv rámce P_{ram} a vzorkovací frekvence. Délka a překryv jsou zadány v sekundách, funkce si jejich velikosti přepočítá na počet vzorků. Pomocí funkce *frame* vzniká z vektoru vstupního signálu matice, kde sloupce značí jednotlivé rámce a řádky počet vzorků v rámci. Jak je uvedeno v kapitole 4.3, rámcování se nejčastěji dělá za pomocí Hammingova okna. Ve funkci to

znamená, že vzorky v rámci vynásobíme Hammingovým oknem, pro které je v Matlabu předdefinovaná funkce. Výstupními parametry funkce, krom již zmíněné matice vzorků, je délka rámce ve vzorcích, která je důležitá pro další výpočetní procesy.

```
function [mat_s,Lram]=frame(sig,usek,prekryv,Fs)

delka=length(sig);      % délka rámcovaného signálu
Lram = usek*Fs;         % počet vzorků v rámci
prekryv = prekryv*Fs;   % velikost překryvu ve vzorcích
posuv=Lram-prekryv;     % velikost posuvu následujícího rámce ve vzorcích
sloupec = floor(1+((delka-Lram)/posuv)); % Protože z vektoru bude vznikat
matice, musíme určit na kolik sloupců (rámců) signál rozdělíme.
mat_s = zeros(Lram,sloupec); % inicializace matice vzorků
%% cyklus, kde do každého sloupce uložím jeden rámec
for k=1:1:sloupec % pro každý sloupec
   for l=1:1:Lram % pro každý řádek
       if ((k-1)*posuv+l)<=delka % dokud je z čeho načítat
           mat_s(l,k) = sig((k-1)*posuv+l); % plníme matici vzorky ze
signálu
       else
           mat_s(l,k)=0; % Zbytek do konce sloupce doplníme nulami
       end;
   end;
end;
for k=1:1:sloupec % pro všechny rámce
    mat_s(1:Lram,k)=mat_s(1:Lram,k).*(hamming(Lram)); % rámce vynásobíme
Hammingovým oknem
end
```

9.1.2 Funkce hlaska

Pokud chceme zjistit, zda je konkrétní rámec znělý nebo neznělý, respektive jestli je součástí znělé či neznělé hlásky, potřebujeme znát počet průchodů nulou Z, krátkodobou energii E, nebo krátkodobou intenzitu M. Všechny tyto charakteristiky počítá funkce *hlaska*. Vstupním parametrem je matice vzorků vytvořená ze signálu díky funkci *frame*. Výše zmíněné parametry jsou počítány pro každý rámec signálu a seřazeny do řádků výstupní matice, kde v prvním řádku jsou hodnoty Z, v druhém E, a třetí řádek obsahuje hodnoty M. Uvnitř funkce si nastavíme, jestli chceme na druhém řádku energii, nebo její logaritmus. Počet průchodů nulou je počítán pomocí funkce sign, která je v Matlabu definována. Využití funkce Sign je podrobněji popsáno v kapitole 4.4. Výpočet krátkodobé energie i intenzity je uveden v kapitole 4.5.

```
function [ZEM]=hlaska(mat_s)

 Poc_ramcu= length (mat_s(1,:)); % počet rámců signálu
 Lram = length (mat_s(:,1)); % počet vzorků v rámci
 ZEM = zeros(3,Poc_ramcu); % inicializace výstupní matice
 %% Výpočet počtu průchodů nulou
 for l=1:1:Poc_ramcu % pro každý rámec
     Z = 0;  % pro každý rámec začínáme počítat průchody od nuly
     x = mat_s(:,l); % vybereme rámec
     for k=2:1:Lram  % pro všechny vzorky v rámci
         Z = Z+ (abs(sign (x(k)) - sign (x(k-1))))/2; %hledáme průchody
nulou
     end
     ZEM(1,l) = Z; % uložení počtu průchodů nulou pro daný rámec
```

```
 end
%% Výpočet energie a intenzity
for l=1:1:Poc_ramcu % pro každý rámec
     ZEM(2,l)= 10*log(sum((mat_s(:,l)).^2)/Lram); % výpočet logaritmu E
%    ZEM(2,l)= sum((mat_s(:,l)).^2)/Lram;         % výpočet E
     ZEM(3,l)= sum(abs(mat_s(:,l)))/Lram;         % výpočet M
end
```

9.1.3 Funkce mfcc

Jak název napovídá, tato funkce bude z časových vzorků každého rámce signálu vytvářet stanovený počet Mel-frekvenčních kepstrálních koeficientů, které budou lépe reprezentovat řečový signál pro rozpoznávání. Mezi vstupní parametry patří již tradičně matice vzorků signálu, vzorkovací frekvence a počet trojúhelníkových filtrů, které budou na vzorky v každém rámci aplikovány. Celý proces vytváření koeficientů se dá rozdělit do tří fází. V první fázi provedeme diskrétní Fourierovu transformaci každého rámce a pro z ní vypočteme výkonové spektrum. Druhá fáze se týká aplikace Mel-banky filtrů. Protože vytvoření banky filtrů ve frekvenční oblasti je nelineární, je jednodušší provést transformaci frekvenční osy z Hertzů na Mely (vzorec 5.3), kde bude rozmístění filtrů lineární. Na pozměněnou frekvenční osu rovnoměrně aplikujeme zvolený počet trojúhelníkových filtrů[2]. Vzorky obsažené v každém filtru sečteme a normujeme jejich počtem. Každý rámec tedy bude reprezentován vektorem koeficientů o velikosti počtu zvolených filtrů. Před začátkem třetí fáze se získané koeficienty logaritmují. Pro získání melovských kepstrálních koeficientů již stačí provést zpětnou kosinovou transformaci. Z výsledných koeficientů budeme potřebovat jen prvních 13. Čtrnáctý koeficient, který vložíme na první místo je logaritmus krátkodobé energie vypočítaný přímo z časových vzorků rámce. U výstupní matice tedy sloupce značí jednotlivé rámce a řádky jejich kepstrální koeficienty.

```
function [Koef] = mfcc(mat_s,Fs,n)

Lram = length (mat_s(:,1)); % délka rámce
f = (0:1023) / 2048 * Fs; % zero padding
%% DFT
for k=1:1:length(mat_s(1,:)) % pro každý rámec
    ramec = mat_s(:,k); % vyjmu rámec z matice
    X = fft([ramec' zeros(1,2048-Lram)]); % zero padding musím zbytek do
množství vzorků doplnit nulami
    DFT(:,k) = X(1:1024); % výsledný signál je zrcadlový, stačí pracovat
s polovinou
end
DFT = abs(DFT.^2); % výpočet výkonového spektra
f_mel = 2959*log10(1+(f/700)); % přepočet frekvenční osy z Hertzů na Mely
max_fmel = max(f_mel); % zjistíme maximální hodnotu
sirka_f = max_fmel/(n/2); % zjistíme mezní hodnoty lineárně poskládaných
filtrů
%% určení počtu vzorků v každém filtru
for l = 1:1:n % pro každý filtr spočítáme počet vzorků
    v=1; % před každým intervalem vždy začínáme se vzorky od jedničky
```

[2] Aplikací filtru rozumíme vynásobení každého vzorku ve frekvenční oblasti filtru příslušným ziskem trojúhelníkového filtru.

```
    while f_mel(v) <floor((l)*(sirka_f/2)) % mezní hodnotu potřebujeme
zaokrouhlit dolů. Je to kvůli poslednímu vzorku, na ostatní výpočty to
vliv nemá.
        v = v+1; % přičteme další vzorek
    end
    K(l)=v; % uložíme vzdálenost ve vzorcích pro jednotlivé filtry
end

K = [1 K]; % před vektor spočítaných intervalů musíme vložit 1 je to
kvůli matlabu, nebere nuly.
%% Výpočet koeficientů pro každý filtr
for k = 1:1:length(DFT(1,:)), % pro všechny rámce
    for l=2:1:n+1, % pro všechny filtry
        if l <= n % podmínka kvůli poslednímu filtru
           v = K(l+1)-K(l-1); % vypočítám počet vzorků ve filtru
           filtr = triang(v+1); % určeni trojúhelníkového filtru
           a = K(l-1); % počáteční mez každého filtru
           Koef(l-1,k)=sum(DFT(a:a+v,k).*filtr)/v; % vynásobíme filtr s
hodnotami spektra a normujeme počtem vzorků
        else % interval posledního filtru, se počítá od předposlední
hodnoty vektoru intervalů až do konce.
            a = K(l-1); % počáteční mez posledního filtru
            v= length(DFT(a:end,k)); % počet vzorků posledního filtru
            filtr = triang(v); %určeni posledního trojúhelníkového filtru
            Koef(l-1,k)=sum(DFT(a:end,k).*filtr)/(1024-a); % vynásobíme
poslední filtr s hodnotami spektra a normujeme počtem vzorků
        end
    end
end
Koef = log(Koef); % zlogaritmování koeficientů
Koef = idct(Koef); % zpětná Kosinova transformace
%% Doplnění prvního koeficientu
for k =1:1:length(Koef(1,:)) % pro všechny rámce
    e = log (sum(mat_s(:,k).^2)); % vypočet logaritmu energie daného
řečového signálu
    Koeff(:,k)=[e Koef(:,k)']; % dosazení výpočtu jako první koeficient.
end
Koef = Koeff(1:14,:); % chceme jenom prvních 14 koeficientů
```

9.1.4 Funkce slova

Tato funkce má za úkol detekovat jednotlivá slova v promluvě. Detekce je prováděna jenom pomocí krátkodobé intenzity. V odstavci 4.6 se pro detekci počítá i s počtem průchodů nulou, ale protože zkoumaná slova neobsahují na začátku ani na konci žádné sykavky, pracuje se ve funkci jenom s intenzitou. Ta je spolu s maticí vzorků dané věty vstupním parametrem funkce. Nejdříve si vypočítáme střední hodnotu a odchylku krátkodobé intenzity šumu, díky kterým vypočítáme detekční úroveň řeči. Šum je brán z prvních nebo posledních třiceti rámců promluvy. Tato funkce počítá i s krátkým poklesem intenzity pod detekční hranici. Pokud detekujeme řeč do 4 rámců od jejího poklesu pod detekční hranici, načítání slova se nezastaví. Pokud se ale na konci zaznamenané promluvy vyskytuje nedokončené slovo, je nutné uvést podmínku, která slovo ukončí. Podmínkou je myšleno automatické přidání nízkých hodnot intenzity. Počet takto přidaných hodnot musí být roven velikosti uvažovaného poklesu. Slovo je detekováno od chvíle, kdy intenzita rámce stoupne nad počáteční úroveň detekce až do chvíle, kdy intenzita bude pod koncovou úrovní detekce a to minimálně po 4 rámce za

sebou. Z empirických zkušeností víme, že slova, s nimiž je v diplomové práci pracováno, nemůžou být menší než 37 rámců a větší než 150 rámců. Pokud jsou slova kratší, nebo delší jedná se o slova nedokončená, nebo naopak o seskupení několika slov, proto jsou takové úseky promluvy na závěr funkce eliminovány. Výstupními parametry jsou třírozměrná matice jednotlivých detekovaných slov a vektor velikostí pro každé detekované slovo.

```
function [Vyh_Slova,Vyh_vel_slov]= slova(Detekce,mat_s)

Lram = length(mat_s(:,1)); % velikost rámce ve vzorcích
poc_ramcu = 200; % stanovená maximální velikost slov
Slova = zeros(Lram,poc_ramcu,1); % Předdefinování výstupní matice
k=30; % počet rámců, ze kterých se bere vzorek šumu, pro nastavení
hladiny detekce
energie = [Detekce -200 -200 -200 -200]; % přidání hodnot intenzity pro
případ nedomluveného slova na konci
%% pro vzorky šumu ze začátku odkomentuj
% Ms = (sum(Detekce(1:k)))/k; % střední hodnota intenzity šumu
% Md = sum((Detekce(1:k)-Ms).^2)/k; % disperze od střední hodnoty
intenzity šumu
%% pro vzorky šumu z konce odkomentuj
Ms = (sum(Detekce(end-k:end)))/k;% střední hodnota intenzity šumu
Md = sum((Detekce(end-k:end)-Ms).^2)/k;% disperze od střední hodnoty
intenzity šumu
Hm = Ms+4*sqrt(Md); % počáteční hladina detekce
Dm = Ms+2*sqrt(Md); % konečná hladina detekce

ramec=1; % aktuální rámec
s=1;    % určuje počet slov ve větě
while ramec<= (length(energie)), % pro rámce + úpadek
     if energie(ramec)>=Hm,% zjistíme začátek slova
         vel_s = 1; % určení velikosti každého slova

         while energie(ramec)>=Dm || energie(ramec+1)>=Dm ||
energie(ramec+2)>=Dm || energie(ramec+3)>=Dm, % načítáme slovo, dokud je
podmínka splněna
            Slova(:,vel_s,s)= mat_s(:,ramec); % plníme matici hodnotami
detekovaného slova
            vel_s = vel_s+1; % velikost slova s každou iterací narůstá
            ramec = ramec+1; % další rámec
         end
         vel_slov(s)=vel_s-1; % uložíme velikost daného slova
         ramec=ramec-1; % hodnotu rámce musíme vrátit o jedno menší,
protože se na konci cyklu zase o jedno zvětší
         s=s+1; % další slovo
     end
    ramec=ramec+1; % další rámec
end
S = 1; % počet vyhovujících slov
for k = 1:1:s-1, % pro všechny slova
    if vel_slov(k)>36 && vel_slov(k)< 150, % podmínka pro vyhovující
slovo
       Vyh_vel_slov(S)= vel_slov(k); % vyhovující velikost slova uložíme
       Vyh_Slova(:,:,S)=Slova(:,:,k); % vyhovující slovo uložíme
       S=S+1; % počet vyhovujících slov zvýšíme
    end
end
```

9.1.5 Funkce dtw

Funkce *dtw* porovnává referenční slovo se slovem testovaným pomocí metody dynamického borcení času. Tato metoda je podrobněji popsána v odstavci 6.2. Vstupními parametry funkce jsou obrazy testovaného a referenčního slova, které obdržíme z funkce *mfcc*. Jednotlivé rámce obrazů jsou tedy reprezentovány kepstrálními koeficienty. Nejdříve si vytvoříme matici d, kde na horizontální ose jsou uvedeny vektory rámců testovaného slova a na vertikální ose vektory rámců slova referenčního. Matici naplníme hodnotami rozdílu mezi patřičnými vektory obrazů. V další fázi definujeme matici g, která bude mít oproti d o jeden sloupec a jeden řádek více. Hodnoty v prvním sloupci a v prvním řádku definujeme jako nereálně velké, pouze hodnota na prvním řádku a prvním sloupci je rovna nule. Docílíme tak vytvoření jediného bodu, ze kterého můžou všechny cesty vycházet. Takovou matici g vytvoříme pro každou použitou metodu lokálního omezení a budeme jí plnit nejmenšími hodnotami z možných cest omezení tak, že sečteme hodnotu v bodě, ze kterého vycházíme, s hodnotou v bodě, do kterého směřujeme. Velký pozor si ale musíme dát na fakt, že hodnoty matice g jsou vždy posunuty o jeden sloupec a jeden řádek výše, než jim příslušné hodnoty matice d. Počítáme-li tedy hodnotu v bodě $g[3,2]$, musíme sečíst hodnotu z pozice, ze které cesta do tohoto bodu vychází s hodnotou uloženou v matici d na pozici $d[2,1]$. Minimální vzdálenost mezi obrazy referenčního a testovaného slova pak udává hodnota v posledním řádku a posledním sloupci matice g.

Protože druhá a třetí metoda lokálního omezení se pohybuje o dvě pozice v každém směru, je potřeba definovat i počáteční podmínky pro druhý sloupec i řádek. U třetí metody je navíc vytvořena podmínka, kdy cesta nemůže jít dvakrát za sebou jenom v horizontálním směru. Na závěr celkovou vzdálenost pro každou metodu vydělíme příslušnou normalizační hodnotou. Použité metody lokálního omezení a k nim příslušné hodnoty jsou uvedeny v Tabulce 1. Jako výstupní parametry jsou uvedeny nejen hodnoty vzdáleností ale i výsledné matice g pro každé omezení.

```
function[D1,D2,D3,g1,g2,g3]=dtw(ref,test),

[koef_test,ramce_test] = size(test); % Zjištění rozměrů testované matice
[koef_ref,ramce_ref] = size(ref); % Zjištění rozměrů referenční matice
d = zeros(ramce_ref,ramce_test); % inicializace matice vzdáleností
%% naplnění matice d
for n=1:1:ramce_ref, % pro každý referenční rámec
    for m=1:1:ramce_test, %pro každý testovaný rámec
        d(n,m) = sum(abs(ref(:,n)-test(:,m))); %rozdíl mezi referenčním a
testovaným rámcem
    end
end
g1 = zeros(ramce_ref+1,ramce_test+1); % předdefinování matice postupného
součtu vzdáleností
g1(1,:)=1e4; % definujeme omezení cesty
g1(:,1)=1e4; % definujeme omezení cesty
g1(1,1)=0; % jediný výchozí bod
g2 = g1; % matice vzdáleností pro druhou metodu lokálního omezení
g3 = g1; % matice vzdálenostÍ pro třetí metodu lokálního omezení
er = zeros(ramce_ref+1,ramce_test+1); % podmínka aby nedošlo k
vícenásobnému opakováni segmentu
%% naplnění matic g
```

```
for m=2:1:ramce_test+1, % pro všechny testované rámce
     for n=2:1:ramce_ref+1 % pro všechny referenční rámce
%% první metoda lokálního omezení
         c0 = g1(n,m-1) + d(n-1,m-1);     % první možnost směru cesty
         c1 = g1(n-1,m-1)+2*d(n-1,m-1);   % druhá možnost směru cesty
         c2 = g1(n-1,m) +d(n-1,m-1);      % třetí možnost směru cesty
         g1(n,m)= min([c0,c1,c2]); % další bod je roven nejmenší hodnotě
možné cesty
%% Druhá metoda lokálního omezení
         if n == 2 && m == 2, % Podmínka pro bod [2,2]
            c0 = 1e4; % tato cesta je z lok. omezení zakázaná
            c1 = g2(n-1,m-1) + 2*d(n-1,m-1); % jediná možná cesta do
tohoto bodu
            c2 = 1e4;  % tato cesta je z lok. omezení zakázaná
         else if n == 2, % Podmínka pro řádky
                 c0 = 1e4; % tato cesta je z lok. omezení zakázaná
                 c1 = g2(n-1,m-1) + 2*d(n-1,m-1); % první možnost cesty
                 c2 = g2(n-1,m-2) + 2*d(n-1,m-2)+ d(n-1,m-1); % druhá
možnost cesty
              else if m == 2, % Podmínka pro sloupce
                      c0 = g2(n-2,m-1) + 2*d(n-2,m-1)+ d(n-1,m-1); % první
možnost cesty
                      c1 = g2(n-1,m-1) + 2*d(n-1,m-1); % druhá možnost
cesty
                      c2 = 1e4; % tato cesta je z lok. omezení zakázaná
                   else
                      c0 = g2(n-2,m-1) + 2*d(n-2,m-1)+ d(n-1,m-1); % první
možnost cesty
                      c1 = g2(n-1,m-1) + 2*d(n-1,m-1); % druhá možnost
cesty
                      c2 = g2(n-1,m-2) + 2*d(n-1,m-2)+ d(n-1,m-1); % třetí
možnost cesty
                   end
               end
         end
         g2(n,m)= min([c0,c1,c2]); % další bod u druhé metody je roven
nejmenší hodnotě možné cesty
%% třetí metoda lokálního omezení
         if n == 2 % úvodní podmínka pro druhy řádek
            c0 = g3(n,m-1)+ er(n,m-1); % první možnost cesty
            c1 = g3(n-1,m-1); % druhá možnost cesty
            c2 = 1e4; %tato cesta je z lok. omezení zakázaná
         else
            c0 = g3(n,m-1)+ er(n,m-1); % první možnost cesty
            c1 = g3(n-1,m-1); % druhá možnost cesty
            c2 = g3(n-2,m-1); % třetí možnost cesty
         end
         g3(n,m)=d(n-1,m-1)+ min([c0,c1,c2]); % další bod je roven
nejmenší hodnotě možné cesty

         if min([c0,c1,c2])==c0 % když jednou vyjdeme z tohoto bodu
            er(n,m) = 1e4; % nastavím podmínku, abych nemohl jít touto
cestou dvakrát za sebou
         else
         er(n,m) = 0; % když nepůjdu touto cestou, podmínka se neuplatní
         end
     end
 end
D1 = g1(end,end)/(ramce_test+ramce_ref); % normalizovaná vzdálenost pro
1. metodu
```

```
D2 = g2(end,end)/(ramce_test+ramce_ref); % normalizovaná vzdálenost pro
2. metodu
D3 = g3(end,end)/(ramce_test); % normalizovaná vzdálenost pro 3. metodu
```

9.1.6 Funkce obsahuje_XX

Funkce *obsahuje_XX* porovnává slova z testované věty s vybranými referenčními slovy. Protože výpočetní náročnost s každým referenčním slovem stoupá a v práci není potřeba hledat v jedné větě všechna referenční slova, byla tato funkce vytvořena pro různé reference zvlášť. Samotné tělo funkce se nemění, rozdíl je pouze v názvech a načtených referencích uvnitř funkce. Vstupními parametry jsou matice slov testované věty, vektor velikostí testovaných slov a vzorkovací frekvence. Nejdříve si funkce načte referenční slova a jejich velikosti ze souboru vytvořeného skriptem *nacteni_ref_XX*. Každé testované slovo pak funkcí *mfcc* převedeme na matici kepstrálních koeficientů a porovnáme ho se všemi referenčními slovy, na které funkci *mfcc* aplikujeme také. Vzdálenosti od všech referenčních slov se ukládají do vektoru pro každou metodu lokálního omezení zvlášť. Máme tedy tři vektory vzdáleností všech referenčních slov s jedním testovaným. Funkcí *seradit* hodnoty ve vektorech seřadíme podle velikosti od nejmenšího k největšímu. Ze seřazených hodnot vybereme $S = 5$ nejmenších a uděláme jejich průměr. Ten bude reprezentovat vzdálenost každého testovaného slova se slovem referenčním. Výstupním parametrem je pak matice reprezentativních vzdáleností, kde sloupce značí jednotlivá testovaná slova a řádky metodu lokálního omezení. Jako zdrojový kód bude znázorněn postup pro jedno referenční slovo.

```
function [sl_brezen]=obsahuje_brezen(veta,vel_slov,Fs),

load('ref_brezen'); % načtení referenčních hodnot
NS = 5; % počet nejbližších sousedu z kterých počítáme průměr

for k=1:1:length(vel_slov), % pro všechna slova věty
    test = veta(:,1:vel_slov(k),k); % vyber testovaného slova
    test = mfcc(test,Fs,20); % převedeme na MFCC koeficienty

    for l=1:1:length(vel_brezen) % pro všechny referenční slova
        ref_A = brezen(:,1:vel_brezen(l),l); % vybereme jedno referenční
slovo
        ref_A = mfcc(ref_A,Fs,20); % převedeme na MFCC koeficienty
        [D1_A(l),D2_A(l),D3_A(l)] = dtw(ref_A,test); % Vypočítáme
vzdálenost referenčního slova s testovaným a hodnoty uložím
    end
%% seřadím hodnoty pro každou metodu omezení
    D1_A = seradit(D1_A);
    D2_A = seradit(D2_A);
    D3_A = seradit(D3_A);
%% Výsledná hodnota je průměr z nejmenších NS hodnot
    sl_brezen(1,k) = (sum(D1_A(1:NS)))/NS;
    sl_brezen(2,k) = (sum(D2_A(1:NS)))/NS;
    sl_brezen(3,k) = (sum(D3_A(1:NS)))/NS;
 end
```

9.1.7 Funkce seradit

Tato funkce je doplňková a jejím úkolem je seřadit hodnoty ve vektoru podle velikosti od nejmenší k největší. Vstupní parametr je vektor, na který chceme funkci aplikovat. Výstupem funkce je vektor se seřazenými hodnotami. Metoda je jednoduchá, nejdříve vybereme nejmenší hodnotu a její pozici ve vstupním vektoru. Hodnotu uložíme na začátek výstupního vektoru. Ve vstupním vektoru pak pozici, kde se nejmenší hodnota nacházela, přeskočíme a opět hledáme minimální hodnotu.

```
function[b]= seradit(a)

delka=length(a);
for k = 1 : delka
    [Hmin,Poz]=min(a); % najdu minimum a pozici minima
    b(k)=Hmin; % uložím hodnotu minima do "b"
    a=[a(1:Poz-1) a(Poz+1:end)]; % vektor "a" definuji pro další hledání
bez minima
end;
end;
```

9.1.8 Funkce rekni

V průběhu vytváření programu byla potřeba si některé slovo z věty poslechnout. Protože slova z vět vybíráme až po procesu rámcování, jsou samotná slova vždy reprezentována vzorky v rámcích. Hlavním úkolem před poslechem je tedy seřadit vzorky zpět do vektoru. Bohužel při rámcování Hammingovým oknem nevratně ztrácíme informaci z původního signálu. Ale i bez ztracené informace lze touto funkcí řečené slovo dobře poznat. Vstupním parametrem je matice vzorků chtěné promluvy a vzorkovací frekvence. Výstupním parametrem je vektor vstupní promluvy. Funkce uvažuje konstantní překryv rámců roven polovině délky rámce.

```
function [signal]=rekni(mat_s,Fs),

signal = []; % vektor do kterého zrekonstruujeme rámcovaný signál
Lram = length(mat_s(:,1)); % délka rámce
for i =1:1:length(mat_s(1,:)), % pro všechny rámce
    cast_ramce = mat_s(1:(Lram/2),i); % vybereme, vzorky které se
nepřekrývají se sousedními rámci
    signal = [signal cast_ramce']; % vybrané vzorky ukládáme za sebou do
vektoru
end
sound (signal,Fs) % poslechneme vstupní promluvu
```

9.2 Skripty v matlabu

9.2.1 Skript nacteni_ref_XX

Všechny referenční slova byla zaznamenána tak, že do časového úseku bylo jedno slovo vyřčeno několikrát za sebou. Tento skript byl vypracován pro každou referenční množinu zvlášť. Po nahrání všech časových úseků s jedním referenčním slovem se každý úsek narámcoval funkcí *frame*. Na takto upravený signál je aplikovaná funkce *hlaska*, která mimo jiné vypočítá krátkodobou intenzitu, což je jedna ze vstupních proměnných další použité funkce *slova*. Výsledkem pro každý úsek je matice jednotlivých nalezených slov,

spolu s vektorem jejich velikostí. Posledním krokem v tomto souboru je sjednocení všech identifikovaných referenčních slov i jejich vzdáleností do jediné matice (vzniká tak třírozměrná matice), resp. vektoru. V práci je stanoven stejný počet vzorů pro všechna referenční slova. Při načtení, ale můžou být některé referenční matice početnější než ostatní. Je tedy snaha je všechny omezit na stejný počet, což znamená na počet nejméně početného z nich. Vymazaná vzorová slova nejsou ničím výrazná a jejich délka je podobná těm, která v referenční množině zůstávají. Ukázka zdrojového kódu znázorňuje načtení a uložení jen pro dvě věty referenčních slov.

```
veta1=load('brezen'); % načtení věty
veta2=load('brezen1'); % načtení věty

[fveta1,Lram] = frame(veta1.data1,0.02,0.01,Fs); % větu narámcujeme
det = hlaska(fveta1); % vypočteme Z,E,M
[brezen1,vel_brezen1]=slova(det(3,:),fveta1); % zjistíme jednotlivá slova
ve větě
[fveta1,Lram] = frame(veta2.data1,0.02,0.01,Fs); % větu narámcujeme
det = hlaska(fveta1); % vypočteme Z,E,M
[brezen2,vel_brezen2]=slova(det(3,:),fveta1); % zjistíme jednotlivá slova
ve větě

brezen=zeros(320,130,1); % předdefinování matice, kam budeme ukládat
referenční slova
brezen = brezen1; % naplnění matice
brezen(:,:,length(brezen(1,1,:))+1:length(brezen(1,1,:))+
length(brezen2(1,1,:)))=brezen2; % naplnění matice

vel_brezen = vel_brezen1; % musíme zjistit i velikosti jednotlivých slov
a ty ukládáme do vektoru
vel_brezen = [vel_brezen vel_brezen2]; % naplnění vektoru velikostmi

save ('ref_brezen','brezen','vel_brezen'); % uložení matice slov a
vektoru vzdáleností
```

9.2.2 Skript omezeni_vety

Víme, že detekční hranice řeči se určuje podle střední hodnoty a odchylky šumu. Tento šum je odebírán z konců nebo začátků promluvy. Ovšem pokud se v těchto místech nachází šum s vyšší intenzitou (vlivem krátkodobého zvýšení hluku) než mezi slovy, může to detekci řeči značně zkomplikovat. Při vykreslení věty zjistíme, kde a jak bychom měli větu oříznout, aby hodnoty pro detekci byly optimální. V tomto skriptu stačí jen načíst konkrétní větu, určit její hranice a zpět uložit. Tento skript pracuje pouze se soubory typu mat.

```
load('veta'); % nahraná věta
data1 = data1(1:5.5e4); % omezení
save('veta','data1'); % uložení
```

9.2.3 Skript rozpoznáni

Tento soubor je hlavní a je v něm obsažena jak část předzpracování, tak i samotného rozpoznávání. Před spuštěním tohoto souboru je ovšem nutné mít připravená referenční slova v maticích.

Nejdříve si načteme větu, ve které budeme referenční slova hledat. V souboru jsou připraveny načítací funkce pro soubory ve třech různých formátech:

- wav,
- txt,
- mat.

Pokud je načtený signál nahráván jako stereo, je prezentován dvěma vektory. Nám stačí využívat jenom jeden, proto nejdříve nahranou větu omezíme na jeden vektor. Dále od nahraného signálu odečteme jeho střední hodnotu. Takto upravený signál je pomocí funkce *frame* „narámcován“. V skriptu můžeme pro jednotlivé rámce získat frekvenční charakteristiku, či spektrální hustotu a samozřejmě si je vykreslit. Funkce *hlaska* vypočítá pro všechny rámce věty počet průchodů nulou, krátkodobou energii a krátkodobou intenzitu. Pro další zpracování je ale důležitá hodnota krátkodobé intenzity, podle které je ve funkci *slova* určena hladina pro detekci řeči. Výstupem funkce *slova* je třírozměrná matice jednotlivých slov a vektor obsahující délku jednotlivých slov. V této chvíli je každé slovo reprezentováno určitým počtem rámců a v každém rámci stejným počtem vzorků. Stačí už jen porovnat slova věty s referenčními slovy prostřednictvím funkce *obsahuje_XX*. Pokud si budeme chtít nějaké slovo nebo celou promluvu poslechnout, můžeme využít funkci *rekni*.

```
%% Načteni souboru wav
[veta,Fs,Nbits]=wavread('Kdo jsi.wav'); % načteni signálu
veta = veta'; % transformace hodnot ze sloupce do řádku
veta = veta(1,:); % pokud je signál stereo
%% Načteni souboru txt
div = fopen('veta42.txt','r'); % otevření složky se signálem
veta = fscanf(div,'%lg'); % načtení signálu
fclose(div); % zavření složky se signálem
veta = veta(1,:); % pokud je signál stereo
%% Načteni souboru mat
veta = load('dubbud61'); % načteme soubor
veta = veta.data1; % ze souboru načteme signál
veta = veta(:,1); % pokud stereo
Fs = 16e3; % vzorkovací frekvence
%% počáteční výpočty a ustředěni
veta = veta-mean(veta); % ustředěni signálu
delka = length(veta); % načteme si počet vzorků v promluvě
%% rámcování
[Fveta,Lram]=frame(veta,0.02,0.01,Fs); % funkce frame
%% frekvenční analýza pro konkrétní rámec
ramec = Fveta (:,100); % vybereme ramec
f = (0:((Lram/2)-1)) / Lram * Fs; % nastavíme frekvenční osu
F_T = fft(ramec); % Fourierova transformace
F_T = F_T(1:(Lram/2)); % stačí vykreslit polovina spektra
f_zp = (0:1023) / 2048 * Fs; % zero padding
F_T_zp = fft([F_T' zeros(1,2048-Lram)]); % Fourierova transformace, kde
musíme zbytek do počtu vzorku doplnit nulami
F_T_zp = F_T_zp (1:1024); % stačí vykreslit polovina spektra
%% spektrální hustota
Gdft = (abs(F_T_zp).^2)/Lram; % spektrální hustota
```

```
Gdftlog = 10*log10(Gdft); % spektrální hustota v logaritmech
%% rozpoznání znělých hlásek,nebo detekce řeči
Detekce = hlaska(Fveta);
%% Detekce řečové aktivity ve větě pomocí krátkodobé intenzity M.
[veta,vel_veta] = slova(Detekce(3,:), Fveta);
%% porovnání referenčních slov se slovy v promluvě
[sl_duben,sl_budem] = obsahuje_duben(veta,vel_veta,Fs);
[sl_brezen,sl_vlezem] = obsahuje_brezen(veta,vel_veta,Fs);
[sl_jeste,sl_kamna] = obsahuje_jestekamna(veta,vel_veta,Fs);
%% Vyslovit větu
 rekni(Fveta,Fs);
```

10 Praktické vyhodnocení práce

V této kapitole jsou vyhodnoceny výsledky, ke kterým jsem při aplikaci vytvořeného softwarového programu na získané nahrávky dospěl. V práci jsou od všech řečníků namluveny dvě věty, každá z vět mimo jiné obsahuje dvě slova, která si jsou akusticky podobná. Věty použité v práci jsou: „Březen za kamna vlezem“ a „Duben ještě tam budem“. Řečníků bylo celkem osm, 4 ženy a 4 muži. Každý z nich řekl obě věty dvakrát. Úkolem bylo namluvit referenční slova jediným řečníkem a zjistit s jakou úspěšností jsme schopni tato slova v promluvách ostatních řečníků najít. Všechna referenční slova jsou namluvená mužem. Tabulky s výsledky jsou rozděleny podle použitých typů lineárního omezení funkce DTW. Pro lepší orientaci je znázorněna tabulka 3, která popisuje jednotlivé části použitých výsledných tabulek.

Tabulka 3 - Popis výsledných tabulek.

Referenční slovo					
Typ funkce DTW	Mluvčí (muž/žena)	testované slovo	testované slovo	testované slovo	testované slovo
I	1				
	2				
	3				
	4				
Detekováno referenční slovo v %					

V prvním řádku tabulky je uvedeno referenční slovo, s kterým jednotlivá testovaná slova porovnáváme. Testovaná slova jsou uvedena v řádku pod ním. Každému testovanému slovu náleží příslušný sloupec, ve kterém budou uvedeny hodnoty vzdáleností referenčního slova s tímto testovaným pro obě věty od každého mluvčího. Mluvčí jsou uvedeni ve sloupci na levé straně. Každému mluvčímu jsou přiřazeny dva řádky pro dvě nezávisle namluvené věty. V každé tabulce je jedno z testovaných slov shodné se slovem referenčním. Ze sloupce hodnot pro toto testované slovo vypočítáme medián. Z empirických zkušeností je medián pro každý typ funkce DTW ještě násoben konstantou a užíván jako hranice pro detekci slova. Pokud nějaká hodnota vzdáleností v tabulce bude menší než hodnota detekční, bude se považovat toto slovo řečené v dané větě určitým řečníkem za stejné jako je slovo referenční. Počet správně detekovaných slov pro každé testované slovo je uveden v procentech v posledním řádku tabulky. Pro ideální detekci by měla být hodnota stoprocentní u testovaného slova, o kterém víme, že je shodné s referenčním. U všech ostatních by byla nulová. Výsledné vzdálenosti testovaných slov od referenčního jsou pro každou tabulku vizuálně rozlišeny. Nejmenší vzdálenosti mají modrou barvu a postupně přecházejí k červené, která značí největší vzdálenost testovaného slova od referenčního.

V práci jsou rozpoznávána 4 referenční slova. Dvě jsou vybrána pro analýzu detekce slov akusticky podobných a zbylá dvě slouží k analýze detekce slov odlišných. Všechna referenční slova jsou řečníkem namluvena 18 krát.

10.1 Rozpoznání ženského hlasu

V systémech používaných v praxi se doporučuje stejný řečník pro referenční i testovaná slova. Bude tedy zajímavé zjistit, jak bude vytvořený program rozpoznávat mužem namluvené referenční slova s promluvou ženy. Podle typu funkce DTW byl pro výpočet detekční hranice medián násoben konstantou.

Tabulka 4 - Tabulka konstant pro rozpoznávání ženského hlasu.

Typ funkce DTW	konstanta
I	1
II	1,05
III	1,05

10.1.1 Rozpoznání akusticky podobných slov

V této podkapitole jsou uvedeny výsledky rozpoznání akusticky si podobných slov.

Tabulka 5 - Rozpoznání slova „březen“ v promluvě ženy I. typem funkce DTW.

březen					
Typ funkce DTW	mluvčí	březen	za	kamna	vlezem
I	žena 1	15,455	14,324	16,395	13,052
		13,998	15,368	17,477	13,427
	žena 2	18,691	16,48	20,194	14,299
		16,725	15,809	19,47	14,96
	žena 3	15,981	15,723	17,717	14,961
		16,794	16,858	17,589	15,444
	žena 4	16,642	20,55	20,627	19,079
		16,206	17,76	20,388	15,946
Detekováno referenční slovo v %		50,00%	50,00%	12,50%	87,50%

Při použití prvního typu lokálního omezení funkce DTW je na výsledné tabulce vidět, že testované slovo „březen“ je ze všech případů správně rozpoznáno v 50%. Akusticky podobné slovo „vlezem“ je ale chybně detekováno v 87,5%. Bohužel i ostatní slova, která si jsou s referenčním slovem méně akusticky podobná, byla chybně detekována.

Tabulka 6 - Rozpoznání slova „březen“ v promluvě ženy II. typem funkce DTW.

březen					
Typ funkce DTW	mluvčí	březen	za	kamna	vlezem
II	žena 1	17,224	17,732	19,549	14,473
		15,321	18,793	20,173	14,828
	žena 2	20,385	19,331	23,554	16,442
		18,73	18,727	22,424	17,021
	žena 3	17,944	19,398	21,117	16,545
		18,309	21,083	21,526	18,154
	žena 4	18,22	24,88	25,404	25,027
		17,581	21,323	23,184	18,625
Detekováno referenční slovo v %		87,50%	37,50%	0,00%	87,50%

Druhý typ omezení dosahuje správného rozpoznání referenčního slova v 87,5%, ovšem stejné hodnoty dosáhla i chybná detekce slova akusticky podobného. Oproti první metodě jsou ale výsledky lepší, když testované slovo „kamna“ nebylo správně detekováno ani jednou.

Tabulka 7 - Rozpoznání slova „březen“ v promluvě ženy III. typem funkce DTW.

březen					
Typ funkce DTW	mluvčí	březen	za	kamna	vlezem
III	žena 1	17,428	17,253	18,875	14,209
		15,428	18,498	19,602	14,694
	žena 2	20,407	18,58	23,122	15,872
		19,136	18,092	21,764	16,327
	žena 3	18,133	18,602	20,328	16,15
		18,258	20,907	21,388	17,612
	žena 4	18,64	24,383	25,509	24,517
		17,875	20,605	22,653	18,708
Detekováno referenční slovo v %		75,00%	62,50%	12,50%	87,50%

U třetího typu omezení jsou výsledky rozpoznání akusticky si podobných slov stejné jako u druhého typu. Zde jsou ale mnohem větší hodnoty chybné detekce ostatních slov věty.

Tabulka 8 - Rozpoznání slova „duben“ v promluvě ženy I. typem funkce DTW.

duben					
Typ funkce DTW	mluvčí	duben	ještě	tam	budem
I	žena 1	14,308	20,64	15,949	13,72
		13,509	19,863	15,893	12,229
	žena 2	13,701	19,41	14,561	11,541
		13,531	19,291	15,033	12,474
	žena 3	12,138	18,953	13,844	13,383
		12,808	19,537	14,225	13,629
	žena 4	14,132	20,686	16,114	11,892
		15,19	23,6	17,529	13,712
Detekováno referenční slovo v %		50,00%	0,00%	0,00%	62,50%

U rozpoznávání dalšího referenčního slova „duben“ prvním typem omezení bylo správné testované slovo detekováno v 50 % výskytu. Pro akusticky podobné slovo „budem“ je ale hodnota o 12,5% větší. Další testovaná slova věty nebyla detekována ani jednou.

Tabulka 9 - Rozpoznání slova „duben“ v promluvě ženy II. typem funkce DTW.

Duben					
Typ funkce DTW	mluvčí	duben	ještě	tam	budem
II	žena 1	15,946	25,658	20,069	15,315
		14,736	24,362	19,631	14,043
	žena 2	15,087	23,139	17,683	12,592
		14,75	23,951	18,149	13,785
	žena 3	13,413	23,045	17,772	15,012
		14,484	23,343	19,063	15,356
	žena 4	16,217	24,468	20,994	13,424
		17,043	27,897	23,131	16,492
Detekováno referenční slovo v %		62,50%	0,00%	0,00%	87,50%

U druhého typu omezení funkce jsou hodnoty takřka podobné s hodnotami z prvního typu omezení. Opět je detekce akusticky si podobných slov téměř stejná a ostatních slov nulová.

Tabulka 10 - Rozpoznání slova „duben“ v promluvě ženy III. typem funkce DTW.

Duben					
Typ funkce DTW	mluvčí	duben	ještě	tam	budem
III	žena 1	16,217	27,476	19,032	15,287
		14,902	25,606	18,442	14,142
	žena 2	15,294	24,401	16,667	12,541
		14,963	25,53	17,271	13,47
	žena 3	13,453	23,4	16,861	14,339
		14,565	24,075	17,386	14,639
	žena 4	16,866	25,85	19,771	13,44
		17,681	29,58	22,014	16,989
Detekováno referenční slovo v %		62,50%	0,00%	0,00%	87,50%

Třetí typ omezení funkce dosahuje naprosto stejných výsledků jako druhý typ omezení.

Na základě dosažených výsledků můžeme tedy konstatovat, že program není vhodný pro rozpoznávání referenčních slov řečených mužem v promluvě ženy. Jediná možnost použití je v případě, že by slova v promluvě byla od sebe akusticky hodně odlišná, což se ve většině případů zaručit nedá.

10.1.2 Rozpoznávání akusticky odlišných slov

V této podkapitole budou uvedeny výsledky pro rozpoznání referenčního slova mezi slovy, která si nejsou akusticky podobná. Výsledné tabulky pro jiné referenční slovo „ještě“ jsou uvedeny v příloze D, kde není dosaženo tak dobré detekce jako u referenčního slova „kamna“ uvedeného zde.

Tabulka 11 - Rozpoznání slova „kamna“ v promluvě ženy I. typem funkce DTW.

Kamna									
Typ funkce DTW	mluvčí	březen	za	kamna	vlezem	duben	ještě	Tam	budem
I	žena 1	19,696	16,338	13,176	18,583	20,031	24,504	14,47	19,685
		18,891	16,663	13,099	18,229	20,272	23,803	15,614	19,714
	žena 2	21,559	17,422	16,206	17,602	19,207	23,914	17,359	19,44
		21,509	16,365	16,405	18,366	19,689	23,677	16,781	19,421
	žena 3	21,144	16,95	16,16	18,799	20,766	25,442	15,05	21,661
		21,245	17,6	16,511	17,861	20,666	25,537	16,536	20,075
	žena 4	20,553	21,241	16,472	19,824	19,867	23,66	15,607	19,419
		20,684	18,695	17,821	18,549	20,16	26,372	16,747	19,723
Detekováno ref. slovo v %		0,00%	0,00%	50,00%	0,00%	0,00%	0,00%	50,00%	0,00%

U prvního typu omezení je správné testované slovo detekováno v 50 % výskytu. V 50% je ale i chybně detekované jiné testované slovo „tam“. U všech ostatních testovaných slov z obou vět se chybná detekce neuskutečnila.

Tabulka 12 - Rozpoznání slova „kamna“ v promluvě ženy II. typem funkce DTW.

Kamna									
Typ funkce DTW	mluvčí	březen	za	kamna	vlezem	duben	ještě	Tam	budem
II	žena 1	23,989	19,46	14,154	22,847	23,125	31,202	17,464	23,474
		22,247	19,331	13,977	21,929	23,444	29,779	19,407	26,904
	žena 2	24,76	20,413	17,346	21,382	21,756	30,174	22,776	25,455
		26,271	19,324	17,488	23,499	22,255	30,819	21,216	24,238
	žena 3	27,788	21,174	18,244	23,122	24,624	31,945	20,961	25,878
		25,116	21,387	18,487	20,602	24,624	32,523	22,87	24,301
	žena 4	23,788	24,523	18,238	22,879	23,099	28,35	19,919	23,288
		23,381	21,67	19,023	20,359	23,373	31,034	21,068	22,984
Detekováno ref. slovo v %		0,00%	0,00%	87,50%	0,00%	0,00%	0,00%	12,50%	0,00%

Druhý typ lokálního omezení už přináší značně lepší výsledky. U akusticky si nepodobných slov je správné testované slovo detekováno v 87,5 %, testované slovo „tam“ v 12,5 % případů a pro všechna další slova je detekce nulová.

Tabulka 13 - Rozpoznání slova „kamna“ v promluvě ženy III. typem funkce DTW.

kamna									
Typ funkce DTW	mluvčí	březen	za	kamna	vlezem	duben	ještě	tam	budem
III	žena 1	24,308	19,322	14,064	23,112	22,969	32,334	17,308	22,551
		22,688	18,898	13,797	22,068	23,304	31,069	19,329	27,501
	žena 2	24,702	19,918	17,393	21,126	21,748	30,93	22,306	24,825
		26,577	18,415	17,499	23,675	22,241	31,674	20,901	23,405
	žena 3	28,012	20,816	17,975	22,934	24,17	31,98	20,396	24,806
		25,353	21,628	18,103	20,643	23,976	32,398	22,298	23,149
	žena 4	24,038	23,365	18,733	22,916	22,69	28,963	19,013	22,479
		23,553	20,958	18,787	20,28	23,522	32,195	19,852	22,503
Detekováno ref. slovo v %		0,00%	12,50%	75,00%	0,00%	0,00%	0,00%	12,50%	0,00%

Třetí typ omezení již zlepšení výsledků nepřináší. Správné testované slovo je odhaleno v 75 % a další dvě testovaná slova jsou chybně detekována v 12,5 % případů výskytu.

10.2 Rozpoznání mužského hlasu

V této podkapitole jsou porovnávány promluvy mužů s referenčními slovy muže. Stejně jako u promluv žen musí být medián pro každý typ omezení funkce násoben konstantou.

Tabulka 14 - Tabulka konstant pro rozpoznávání mužského hlasu.

Typ funkce DTW	konstanta
I	1,05
II	1,1
III	1,1

10.2.1 Rozpoznávání akusticky podobných slov

V této podkapitole jsou uvedeny výsledky rozpoznávání pro akusticky si podobná slova.

Tabulka 15 - Rozpoznání slova „březen“ v promluvě muže I. typem funkce DTW.

březen					
Typ funkce DTW	mluvčí	březen	za	kamna	vlezem
I	muž 1	12,298	15,241	16,103	11,635
		11,984	15,217	16,351	13,626
	muž 2	11,931	15,416	14,123	12,674
		11,319	14,419	14,074	12,502
	muž 3	12,561	15,67	15,268	14,273
		12,381	14,26	16,199	13,596
	muž 4	12,649	14,467	14,743	13,477
		11,63	14,626	14,641	13,478
Detekováno referenční slovo v %		100,00%	0,00%	0,00%	37,50%

U prvního typu omezení bylo na 100 % správně detekováno testované slovo „březen“. Z 37,5 % bylo chybně detekováno testované slovo „vlezem“. Všechna ostatní testovaná slova detekována nebyla ani jednou.

Tabulka 16 - Rozpoznání slova „březen“ v promluvě muže II. typem funkce DTW.

Březen					
Typ funkce DTW	mluvčí	březen	za	kamna	vlezem
II	muž 1	15,3	17,676	18,825	13,376
		13,233	17,916	18,895	14,833
	muž 2	13,305	17,374	17,27	14,87
		12,602	16,786	16,71	14,681
	muž 3	13,582	18,515	18,081	16,325
		14,101	16,676	19,549	16,064
	muž 4	14,214	17,814	17,907	15,81
		12,952	17,072	17,27	15,501
Detekováno referenční slovo v %		87,50%	0,00%	0,00%	25,00%

Při aplikaci druhého typu omezení se z namluvených vět v 87,5 % případů výskytu správně detekovalo referenční slovo. V 25 % však došlo k chybné detekci u slova „vlezem“. Všechny ostatní testovaná slova opět detekována nebyla ani jednou.

Tabulka 17 - Rozpoznání slova „březen“ v promluvě muže III. typem funkce DTW.

Březen					
Typ funkce DTW	mluvčí	březen	za	kamna	vlezem
III	muž 1	16,265	17,506	18,604	12,992
		13,3	17,625	18,247	14,591
	muž 2	13,625	17,129	16,365	14,221
		12,66	16,387	16,103	14,161
	muž 3	13,726	18,227	17,478	16,088
		14,452	16,166	18,861	15,833
	muž 4	14,27	17,193	17,177	15,109
		13,129	16,397	16,878	15,229
Detekováno referenční slovo v %		87,50%	0,00%	0,00%	50,00%

Pomocí třetího typu omezení bylo referenční slovo ve větě správně rozpoznáno z 87,5 %. Akusticky podobné slovo „vlezem“ bylo chybně rozpoznáno v 50 % případů. U všech ostatních testovaných slov zůstaly vzdálenosti nad detekční hranicí.

Tabulka 18 - Rozpoznání slova „duben“ v promluvě muže I. typem funkce DTW.

duben					
Typ funkce DTW	mluvčí	duben	ještě	tam	budem
I	muž 1	12,707	19,378	16,992	12,886
		11,854	17,708	15,155	11,232
	muž 2	10,01	16,526	14,277	10,435
		9,7028	16,182	13,124	10,59
	muž 3	11,458	17,929	13,401	11,671
		10,982	17,437	13,806	11,58
	muž 4	12,487	16,869	15,929	11,381
		11,356	14,816	12,933	10,038
Detekováno referenční slovo v %		75,00%	0,00%	0,00%	87,50%

Při rozpoznávání dalšího referenčního slova prvním typem omezení program správně detekoval v promluvách referenční slovo na 75 % a chybně detekoval slovo akusticky podobné v 87,5 % případů. Ve všech ostatních testovaných slovech detekce nenastala.

Tabulka 19 - Rozpoznání slova „duben“ v promluvě muže II. typem funkce DTW.

duben					
Typ funkce DTW	mluvčí	duben	ještě	tam	budem
II	muž 1	14,934	24,475	24,383	15,428
		13,732	22,083	22,149	12,602
	muž 2	11,217	19,028	18,599	11,578
		11,162	19,276	17,912	11,504
	muž 3	13,118	22,247	17,452	12,959
		12,206	21,719	17,461	12,814
	muž 4	14,575	19,972	20,352	12,867
		12,838	17,493	17,355	11,186
Detekováno referenční slovo v %		75,00%	0,00%	0,00%	87,50%

Druhý typ omezení přináší pro referenční slovo „duben“ naprosto stejné výsledky jako první typ.

Tabulka 20 - Rozpoznání slova „budem“ v promluvě muže III. typem funkce DTW.

duben					
Typ funkce DTW	mluvčí	duben	ještě	tam	budem
III	muž 1	15,479	25,859	23,723	16,1
		14,273	24,029	21,576	12,888
	muž 2	11,573	20,188	18,005	11,355
		11,363	20,169	17,007	11,343
	muž 3	13,546	23,657	16,163	12,689
		12,522	22,788	15,948	12,49
	muž 4	14,959	21,461	19,57	12,714
		13,149	18,521	16,361	11,156
Detekováno referenční slovo v %		75,00%	0,00%	0,00%	87,50%

Pro třetí typ lokálního omezení funkce DTW jsou výsledky shodné s oběma předešlými typy. Referenční slovo „duben“ se tedy se slovem „budem“ v promluvě hůře rozpoznávalo než referenční slovo „březen“ se slovem „vlezem“. Stačí tedy menší akustická podobnost a slova se rozpoznávají lépe.

10.2.2 Rozpoznávání akusticky odlišných slov

V této kapitole budou uvedeny výsledky porovnání referenčního slova s testovanými slovy, která si nejsou akusticky podobná. Obdobné výsledky pro jiné referenční slovo „ještě“ jsou uvedeny v příloze C. Na rozdíl od rozpoznávání slov u žen, výsledky rozpoznávání odlišných slov u mužů vyšly pro obě referenční slova podobně a s lepšími výsledky.

Tabulka 21 - Rozpoznání slova „kamna“ v promluvě muže I. typem funkce DTW.

kamna									
Typ funkce DTW	mluvčí	březen	za	kamna	vlezem	duben	ještě	tam	budem
I	muž 1	17,256	15,258	11,953	15,176	17,5	23,204	13,59	17,077
		15,661	16,773	13,721	17,229	15,379	20,853	13,035	17,46
	muž 2	17,884	17,013	12,421	17,066	15,802	22,041	14,045	18,303
		16,687	16,242	12,541	17,447	16,177	21,702	13,682	18,823
	muž 3	17,943	17,138	12,439	17,955	19,467	24,934	15,036	20,474
		19,001	16,523	13,196	18,101	19,812	25,601	15,137	20,166
	muž 4	13,898	14,564	10,863	15,29	16,487	20,407	12,619	16,207
		15,034	13,167	11,696	15,006	16,003	19,213	13,331	16,129
Detekováno ref. slovo v %		0,00%	0,00%	75,00%	0,00%	0,00%	0,00%	25,00%	0,00%

Aplikací prvního typu omezení se referenční slovo podařilo správně detekovat v 75 % případů. V 25 % případů výskytu se ovšem chybně detekovalo testované slovo „tam“. Ve všech ostatních slovech správně detekce nenastala.

Tabulka 22 - Rozpoznání slova „kamna“ v promluvě muže II. typem funkce DTW.

kamna									
Typ funkce DTW	mluvčí	březen	za	kamna	vlezem	duben	ještě	tam	budem
II	muž 1	22,007	17,786	12,702	19,094	21,522	27,562	18,553	19,21
		20,091	19,68	15,311	19,584	17,894	25,35	17,48	20,969
	muž 2	23,249	20,761	15,336	21,497	20,675	27,71	17,372	23,072
		22,002	19,912	14,548	22,273	21,228	27,852	17,537	22,685
	muž 3	22,192	20,348	13,87	22,18	25,075	30,58	22,03	24,463
		25,173	19,773	15,23	22,107	24,427	31,812	22,365	24,944
	muž 4	17,478	17,391	12,804	19,125	20,927	25,061	18,1	19,388
		19,121	15,317	13,803	18,84	19,477	24,699	19,983	20,096
Detekováno ref. slovo v %		0,00%	12,50%	100,00%	0,00%	0,00%	0,00%	0,00%	0,00%

Při druhém typu lokálního omezení, se referenční slovo podařilo detekovat v každém případě. Krom testovaného slova „za“, které bylo chybně detekováno v 12,5 % případů, se detekce nikde neuskutečnila.

Tabulka 23 - Rozpoznání slova „kamna“ v promluvě muže III. typem funkce DTW.

kamna									
Typ funkce DTW	mluvčí	březen	za	kamna	vlezem	duben	ještě	tam	budem
III	muž 1	22,538	17,135	12,657	19,109	21,395	28,374	17,482	19,258
		20,323	18,875	15,633	19,518	17,647	25,945	16,71	20,701
	muž 2	23,973	20,636	15,527	20,949	20,191	28,395	16,539	21,888
		22,372	19,753	14,232	21,443	20,841	28,25	16,559	21,913
	muž 3	22,335	19,81	13,662	21,831	24,543	30,684	20,934	23,415
		24,996	19,271	15,134	21,293	23,545	31,736	21,227	23,899
	muž 4	17,961	16,956	12,64	18,409	20,982	25,94	17,512	19,103
		19,123	14,78	13,53	18,149	19,156	25,545	19,154	19,354
Detekováno ref. slovo v %		0,00%	12,50%	75,00%	0,00%	0,00%	0,00%	0,00%	0,00%

U posledního třetího typu omezení funkce DTW bylo referenční slovo správně detekováno v 75 % případů. V 12,5 % bylo chybně detekováno slovo „za“, pro všechna ostatní testovaná slova detekce nenastala.

10.3 Rozpoznání referenčního hlasu

V této podkapitole jsou uvedeny výsledky, kdy referenční slova i testovaná promluva pochází od stejného řečníka. Ukázány zde budou pouze tabulky pro rozpoznávání akusticky odlišných slov. Výsledné tabulky pro rozpoznávání akusticky si podobných slov jsou uvedeny v příloze B. Konstanty, kterými násobíme mediány, jsou stejné jako u rozpoznávání mužského hlasu a jsou uvedeny v tabulce 14.

10.3.1 Rozpoznávání akusticky odlišných slov

Rozpoznávání akusticky odlišných slov v případě, že referenční i testovaná slova jsou od stejného řečníka, je ideální případ, který je doporučován u podobných systémů v praxi.

Tabulka 24 - Rozpoznání slova „kamna“ v promluvě referenčního řečníka I. typem funkce DTW.

kamna									
Typ funkce DTW	mluvčí	březen	za	kamna	vlezem	duben	ještě	tam	budem
I	muž	15,803	15,289	10,009	16,399	15,792	21,758	13,34	18,073
		15,875	15,243	10,116	15,433	17,119	22,043	13,865	17,623
		13,676	14,022	10,036	15,358	15,862	21,181	13,316	18,776
Detekováno ref. slovo v %		0,00%	0,00%	100,00%	0,00%	0,00%	0,00%	0,00%	0,00%

První typ lokálního omezení detekoval referenční slovo ve všech správných případech. Pro všechna testovaná slova neproběhla ani jedna chybná detekce. Na tabulkách je vidět velký rozdíl mezi hodnotami testovaného slova shodného s referenčním a hodnotami ostatních slov.

Tabulka 25 - Rozpoznání slova „kamna“ v promluvě referenčního řečníka II. typem funkce DTW.

kamna									
Typ funkce DTW	mluvčí	březen	za	kamna	vlezem	duben	ještě	tam	budem
II	muž	19,822	18,708	10,968	19,895	18,606	25,39	17,751	21,018
		20,503	18,172	10,871	17,928	20,513	25,786	18,643	20,031
		18,209	17,023	10,86	18,674	19,003	25,371	18,302	22,058
Detekováno ref. slovo v %		0,00%	0,00%	100,00%	0,00%	0,00%	0,00%	0,00%	0,00%

U druhého typu lokálního omezení opět neproběhla u žádného slova chybná detekce a všechna testovaná slova shodná s referenčním byla detekována. Rozdíl mezi hodnotami testovaného slova shodného s referenčním a hodnotami ostatních slov je ještě větší než u prvního typu omezení.

Tabulka 26 - Rozpoznání slova „kamna“ v promluvě referenčního řečníka III. typem funkce DTW.

Kamna									
Typ funkce DTW	mluvčí	březen	za	Kamna	vlezem	duben	ještě	tam	budem
III	muž	19,389	17,142	10,858	19,646	18,887	25,899	16,332	20,069
		20,434	16,569	10,667	17,911	20,007	26,516	17,585	19,468
		17,961	16,029	10,69	18,448	18,602	25,532	17,38	21,093
Detekováno ref. slovo v %		0,00%	0,00%	100,00%	0,00%	0,00%	0,00%	0,00%	0,00%

I poslední typ lokálního omezení má výborné výsledky. Referenční slovo se podařilo správně detekovat v každém případě výskytu a žádné jiné testované slovo chybně detekováno nebylo.

11 Závěr

Cílem práce bylo vytvořit softwarovou aplikaci v prostředí Matlab, která bude schopná analyzovat rozpoznávání referenčních slov, namluvených jedním mužem, v diskrétní promluvě různých řečníků. Referenční slova byla rozpoznávána mezi slovy akusticky podobnými i odlišnými.

Vytvořená aplikace získává hodnoty pro rozpoznání třemi různými typy lokálního omezení funkce DTW. Výsledky všech tří typů se dají objektivně porovnávat pouze u rozpoznávání odlišných slov. První typ je zcela základní a tomu odpovídají i výsledky, které jsou ze všech typů nejhorší. Naopak nejlepších výsledků ve všech skupinách dosáhl druhý typ lokálního omezení použitého v práci. Pro urychlení výpočtů v aplikaci je tedy možné výpočty pomocí prvního a třetího typu lokálního omezení vynechat.

První skupinu řečníků tvořily ženy. Při rozpoznávání referenčního slova v promluvě, kde se vyskytuje i slovo referenčnímu slovu podobné, se aplikace příliš často projevovala chybnými detekcemi, jejichž počet byl vždy vyšší než správná detekce referenčního slova. Pro rozpoznávání mezi slovy akusticky odlišnými, již výsledky u druhé a třetí metody lokálního omezení dosahují 75% respektive 87,5% úspěšnosti odhalení referenčního slova oproti 12,5% chybné detekce slova jiného.

Druhou testovanou skupinu tvořili muži. Výsledky rozpoznávání dvou akusticky si podobných slov jsou obdobné jako u skupiny žen. Opět se tedy potvrdil předpoklad, že softwarová aplikace od sebe nedokáže dobře rozlišit dvě slova, která jsou si akusticky podobná. Ovšem v případě, kdy jsme se snažili najít referenční slovo mezi slovy akusticky odlišnými, se projevil fakt, že referenční slova byla namluvena mužem a výsledky správné detekce dosahovaly úspěšnosti 75%-100%. Chybná detekce se projevovala výrazněji jen u první metody lokálního omezení funkce DTW, u druhé a třetí metody omezení byla hodnota chybné detekce 12,5%.

Ve třetí skupině je testován mužský hlas s vlastním referenčním hlasem. I pro stejný hlas řečníka aplikace nedokáže přesně rozlišit podobná slova a výsledky jsou tedy shodné s předchozími skupinami. Ovšem pro rozpoznávání referenčního slova mezi slovy odlišnými dosahují hodnoty detekce výborných výsledků. Pro každý typ lokálního omezení funkce je detekce referenčního slova stoprocentní, při žádném výskytu nenastanou chybné detekce. Hodnoty vzdáleností mezi referenčním slovem a správnými testovanými slovy jsou mnohem menší než hodnoty vzdáleností referenčního slova se všemi ostatními slovy.

Pokud by se měla vytvořená softwarová aplikace využít na systémy s jedním referenčním hlasem, které budou rozpoznávat příkazy od více řečníků, musí se počítat s horší kvalitou rozpoznávání příkazů. Pokud by byly požadavky na kvalitu rozpoznávání vysoké, doporučoval bych, aby byl systém ovládán jenom tím, kdo poskytnul referenční slova. V tom případě budou výsledky rozpoznávání na velmi vysoké úrovni.

Literatura:

BAZIKA, Pavel. 2008. Rozpoznávač hlasu na procesoru Cell. *Diplomová práce.* Praha : Fakulta elektrotechnická ČVUT, 2008.

ČERNOCKÝ, Jan. 2006. Zpracování řečových signálů - studijní opora. [Online] 6. 12 2006. [Citace: 28. 1 2010.] http://www.fit.vutbr.cz/study/courses/ZRE/public/opora/zre_opora.pdf.

KODYS, spol. s r.o. Hlasem řízený sklad K.voice - Kodys. *Kodys.* [Online] [Citace: 15. 5 2010.] http://www.kodys.cz/reseni/logistika-skladovani-a-preprava/hlasem-rizeny-sklad-k.voice.html.

LAVICKÝ, Martin. 2003. Demonstrativní systém na rozpoznávání jednotlivých slov. *Diplomová práce.* Brno : Fakulta elektrotechniky a informačních technologií VUT, 2003.

NÝVLT, Václav. 2003. Ostre068.pdf. [Online] 9. 5 2003. [Citace: 15. 5 2010.] http://topreklama.cz/voice/068ostre.pdf.

Pollák, Petr a Rajnoha, Josef. 2008. Detektory řečové aktivity na bázi perceptivní kepstrální analýzy. *článek.* Praha : Fakulta elektrotechnická ČVUT, 2008.

PSUTKA, Josef. 1995. *Komunikace s počítačem mluvenou řečí.* Praha : Academia Praha, 1995. ISBN 80-200-0203-0.

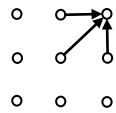
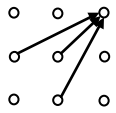
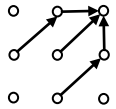
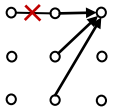
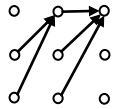
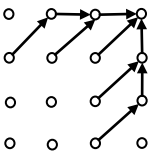
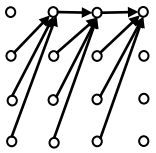
PSUTKA, Josef, a další. 2006. *Mluvíme s počítačem česky.* Praha : Academia Praha, 2006. ISBN-80-200-1309-1.

STEJSKAL, Vojtěch. Zpracovávání řeči. *cvičení.* Brno : Fakulta informačních technologií, VUT.

TopReklama. Voice Me - hlasový ovladač. *Top reklama.* [Online] AG TOP TIP - s. r. o. [Citace: 15. 05 2010.] http://topreklama.cz/voice/.

VUT, Fakulta informační technologie. Zpracování řečových signálů. [Online] Fakulta informační technologie VUT. [Citace: 15. 2 2010.] http://www.fit.vutbr.cz/study/courses/ZRE/public/.

Příloha A - Tabulka nejpoužívanějších lokálních omezení

Typ funkce DTW		A	β	Typ W(k)	g(n,m)
I		0	∞	a	$min\begin{Bmatrix} g(n,m-1) + d(n,m) \\ g(n-1,m-1) + 2d(n,m) \\ g(n-1,m) + d(n,m)) \end{Bmatrix}$
				d	$min\begin{Bmatrix} g(n,m-1) + d(n,m) \\ g(n-1,m-1) + d(n,m) \\ g(n-1,m) + d(n,m)) \end{Bmatrix}$
II		1/2	2	a	$min\begin{Bmatrix} g(n-1,m-2) + 3d(n,m) \\ g(n-1,m-1) + 2d(n,m \\ g(n-2,m-1) + 3d(n,m) \end{Bmatrix}$
				c	$min\begin{Bmatrix} g(n-1,m-2) + d(n,m) \\ g(n-1,m-1) + d(n,m) \\ g(n-2,m-1) + d(n,m) \end{Bmatrix}$
III		1/2	2	a	$min\begin{Bmatrix} g(n-1,m-2) + 2d(n,m-1) + d(n,m) \\ g(n-1,m-1) + 2d(n,m) \\ g(n-2,m-1) + 2d(n-1,m) + d(n,m) \end{Bmatrix}$
IV		1/2	2	b1	$min\begin{Bmatrix} g(n-1,m) + \kappa d(n,m) \\ g(n-1,m-1) + d(n,m) \\ g(n-1,m-2) + d(n,m) \end{Bmatrix}$ $\kappa = 1 \text{ pro j } (k-1) \neq j\ (k-2)$ $\kappa = 1 \text{ pro j } (k-1) \neq j\ (k-2)$
V		1/2	2	d	$min\begin{Bmatrix} g(n-1,m-2) + 2d(n,m) \\ g(n-1,m-1) + d(n,m) \\ g(n-2,m-2) + 2d(n-1,m) + d(n,m) \\ g(n-2,m-1) + d(n-1,m) + d(n,m) \end{Bmatrix}$
VI		1/3	3	a	$min\begin{Bmatrix} g(n-1,m-3) + 2d(n,m-2) + d(n,m-1) + d(n,m) \\ g(n-1,m-2) + 2d(n,m-1) + d(n,m) \\ g(n-1,m-2) + 2d(n,m) \\ g(n-2,m-1) + 2d(n-1,m) + d(n,m) \\ g(n-3,m-1) + 2d(n-2,m) + d(n-1,m) + d(n,m) \end{Bmatrix}$
VII		1/3	3	a	$min\begin{Bmatrix} g(n-1,m-3) + 4d(n,m) \\ g(n-1,m-2) + 3d(n,m) \\ g(n-1,m-1) + 2d(n,m) \\ g(n-2,m-3) + 4d(n-1,m) + d(n,m) \\ g(n-2,m-2) + 3d(n-1,m) + d(n,m) \\ g(n-2,m-1) + 2d(n-1,m) + d(n,m) \\ g(n-3,m-3) + 4d(n-2,m) + d(n-1,m) + d(n,m) \\ g(n-3,m-3) + 3d(n-2,m) + d(n-1,m) + d(n,m) \\ g(n-3,m-1) + 2d(n-2,m) + d(n-1,m) + d(n,m) \end{Bmatrix}$

Příloha B - Tabulky výsledků rozpoznání akusticky si podobných slov v promluvě referenčního řečníka.

březen					
Typ funkce DTW	mluvčí	březen	za	kamna	vlezem
I	muž	11,03	14,753	16,323	12,073
		12,184	14,605	15,395	11,278
		10,78	14,026	15,85	12,298
Detekováno referenční slovo v %		66,67%	0,00%	0,00%	33,33%
Typ funkce DTW	mluvčí	březen	za	kamna	vlezem
II	muž	12,255	18,57	19,306	13,538
		14,204	17,688	18,61	12,473
		12,572	18,23	19,585	13,793
Detekováno referenční slovo v %		66,67%	0,00%	0,00%	100,00%
Typ funkce DTW	mluvčí	březen	za	kamna	vlezem
III	muž	12,444	18,07	18,137	13,144
		14,709	17,221	17,805	12,156
		12,69	17,015	18,905	13,35
Detekováno referenční slovo v %		66,67%	0,00%	0,00%	100,00%

duben					
Typ funkce DTW	mluvčí	duben	ještě	tam	budem
I	muž	10,405	15,541	14,728	11,591
		11,323	17,988	15,72	12,958
		11,244	17,112	14,084	12,432
Detekováno referenční slovo v %		100,00%	0,00%	0,00%	33,33%
Typ funkce DTW	mluvčí	duben	ještě	tam	budem
II	muž	11,325	19,134	19,687	13,191
		12,669	21,5	18,966	15,46
		12,858	20,671	18,675	13,754
Detekováno referenční slovo v %		100,00%	0,00%	0,00%	66,67%
Typ funkce DTW	mluvčí	duben	ještě	tam	budem
III	muž	11,257	20,47	17,811	13,19
		12,67	22,818	17,46	15,236
		13,046	21,794	17,176	13,616
Detekováno referenční slovo v %		100,00%	0,00%	0,00%	66,67%

Příloha C – Tabulky výsledků rozpoznání akusticky odlišného slova „ještě“ v promluvě muže.

ještě									
Typ funkce DTW	mluvčí	březen	za	kamna	vlezem	duben	ještě	tam	budem
I	muž 1	16,837	21,491	19,867	17,904	16,105	12,764	20,122	15,128
		14,848	19,748	18,151	18,15	14,531	12,148	19,145	14,378
	muž 2	17,259	21,375	18,27	17,251	15,275	13,524	17,813	16,296
		15,918	20,467	18,664	17,764	17,017	14,44	18,525	15,87
	muž 3	17,162	21,82	19,585	18,452	15,74	16,129	20,191	15,361
		17,045	21,096	20,934	19,08	15,638	16,501	20,425	15,131
	muž 4	17,575	21,535	18,608	20,213	18,322	12,585	20,116	17,885
		17,453	20,981	19,979	19,32	17,497	12,931	19,832	18,033
Detekováno ref. slovo v %		0,00%	0,00%	0,00%	0,00%	0,00%	62,50%	0,00%	0,00%

ještě									
Typ funkce DTW	mluvčí	březen	za	kamna	vlezem	duben	ještě	tam	budem
II	muž 1	21,703	24,288	23,693	20,661	19,842	13,59	28,687	18,963
		17,766	23,197	21,61	20,544	17,728	13,204	26,518	18,175
	muž 2	21,664	25,616	22,273	21,086	18,426	15,225	25,373	21,154
		19,164	23,703	22,394	22,729	20,104	15,867	25,95	20,678
	muž 3	20,791	24,677	24,418	22,794	19,291	19,119	26,871	20,955
		20,835	25,045	25,642	23,976	18,872	20,247	27,43	20,878
	muž 4	21,255	26,541	23,826	24,106	20,949	13,627	27,384	21,764
		21,138	25,249	23,731	22,739	20,273	14,882	27,528	22,449
Detekováno ref. slovo v %		0,00%	0,00%	0,00%	0,00%	0,00%	75,00%	0,00%	0,00%

ještě									
Typ funkce DTW	mluvčí	březen	za	kamna	vlezem	duben	ještě	tam	budem
III	muž 1	22,107	23,341	22,951	19,42	19,445	13,897	27,32	17,185
		16,933	22,23	20,877	19,78	16,209	13,005	24,631	16,601
	muž 2	22,133	24,464	21,01	20,096	17,937	15,272	24,123	20,055
		18,722	22,829	21,507	21,436	19,533	15,688	24,435	19,349
	muž 3	20,171	23,81	23,953	21,472	19,272	18,853	24,905	19,719
		20,637	23,882	25,022	22,76	18,79	19,969	24,91	19,256
	muž 4	20,55	25,397	22,982	22,478	19,72	13,736	25,305	20,435
		20,836	24,407	22,814	21,496	19,124	15,083	25,422	20,595
Detekováno ref. slovo v %		0,00%	0,00%	0,00%	0,00%	12,50%	75,00%	0,00%	12,50%

Příloha D - Tabulky výsledků rozpoznání akusticky odlišného slova „ještě“ v promluvě ženy.

ještě									
Typ funkce DTW	mluvčí	březen	za	kamna	vlezem	duben	ještě	tam	budem
I	žena 1	16,488	20,929	19,388	17,314	17,853	15,763	19,469	15,678
		16,395	20,779	19,976	17,518	17,708	16,451	21,348	15,306
	žena 2	20,37	20,629	20,905	19,085	18,186	18,4	20,465	16,634
		19,749	20,811	21,133	19,127	19,773	18,779	20,591	16,969
	žena 3	19,202	20,266	18,847	17,42	17,9	19,637	20,263	17,575
		19,161	22,186	20,255	18,186	18,275	18,778	19,149	15,382
	žena 4	18,854	23,775	22,441	21,508	18,07	16,072	20,08	16,334
		19,395	21,889	22,737	20,109	19,165	17,949	20,787	16,326
Detekováno ref. slovo v %		25,00%	0,00%	0,00%	37,50%	50,00%	50,00%	0,00%	100,00%

ještě									
Typ funkce DTW	mluvčí	březen	za	kamna	vlezem	duben	ještě	tam	budem
II	žena 1	20,571	25,607	23,613	21,25	21,748	18,051	25,945	20,145
		20,51	25,841	24,4	21,755	20,702	18,543	26,648	19,34
	žena 2	22,952	24,169	25,994	23,139	21,274	20,711	26,917	20,986
		24,667	25,461	25,467	22,146	22,98	22,061	26,675	23,41
	žena 3	22,737	24,832	22,985	20,33	20,933	22,84	27,693	22,506
		21,778	27,698	26,082	21,311	22,174	22,343	26,299	19,975
	žena 4	23,138	30,064	27,637	26,313	21,153	17,315	26,245	20,513
		21,861	25,624	26,606	22,291	22,368	19,617	27,838	20,391
Detekováno ref. slovo v %		25,00%	0,00%	0,00%	12,50%	37,50%	62,50%	0,00%	75,00%

ještě									
Typ funkce DTW	mluvčí	březen	za	kamna	vlezem	duben	ještě	tam	budem
III	žena 1	20,129	24,811	22,856	20,014	21,368	18,363	24,786	18,4
		20,232	25,085	24,015	21,532	20,317	18,915	24,941	17,841
	žena 2	22,706	23,282	25,411	21,984	20,652	20,636	24,915	19,172
		25,012	24,455	24,783	20,317	22,6	22,163	25,24	21,954
	žena 3	22,515	23,885	22,111	20,009	20,203	22,807	26,06	20,677
		21,742	26,932	25,681	21,129	21,467	22,064	23,828	18,217
	žena 4	22,716	29,231	27,818	25,554	20,431	17,302	24,134	18,94
		21,603	24,643	26,201	21,749	21,89	19,957	25,934	19,387
Detekováno ref. slovo v %		25,00%	0,00%	0,00%	50,00%	50,00%	62,50%	0,00%	87,50%