České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra radioelektroniky

Implementace rozpoznávače na bázi GMM-HMM v programovém systému MATLAB

Diplomová práce

Bc. Kristýna Žáková

Magisterský program: Elektronika a komunikace
Specializace: Audiovizuální technika a zpracování signálů
Vedoucí práce: Doc. Ing. Petr Pollák, CSc.

Praha, Květen 2020

Vedoucí práce:
Doc. Ing. Petr Pollák, CSc.
Katedra teorie obvodů
Fakulta elektrotechnická
České vysoké učení technické v Praze
Technická 2
160 00 Praha 6
Česká republika

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Žáková** Jméno: **Kristýna** Osobní číslo: **456918**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra radioelektroniky**

Studijní program: **Elektronika a komunikace**

Specializace: **Audiovizuální technika a zpracování signálů**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Implementace rozpoznávače na bázi GMM-HMM v programovém systému MATLAB

Název diplomové práce anglicky:

Implementation of GMM-HMM based Recognizer in Program System MATLAB

Pokyny pro vypracování:

1. Seznamte se s problematikou rozpoznávání řeči na bázi GMM-HMM s užším zaměřením na základní principy a algoritmy konstrukce rozpoznávače s malým slovníkem.
2. Implementujte zjednodušenou variantu rozpoznávače s malým slovníkem v programovém systému MATLAB použitelnou především pro demonstraci základních vlastností a principů funkčnosti systému na bázi GMM-HMM pro výukové účely.
3. S ohledem na využití pro výukové účely se při implementaci zaměřte také na vizualici finálních i dílčích výsledků v rozpoznávací i trénovací fázi.
4. Výsledný rozpoznávač otestujte na signálech z dostupných databázích.

Seznam doporučené literatury:

[1] X. Huang, A. Acero, H.-W. Hon. Spoken Language Processing. Prentice Hall, 2001.
[2] J. Psutka, L. Müller, J. Matoušek, V. Radová. Mluvíme s počítačem česky. Academia 2006.
[3] J. Uhlíř a kol.: Technologie hlasových komunikací. Nakladatelství ČVUT, Praha, 2007.
[4] I. McLoughlin.: Applied Speech and Audio Processing with MATLAB Examples. Cambridge University Press, 2009.

Jméno a pracoviště vedoucí(ho) diplomové práce:

doc. Ing. Petr Pollák, CSc., katedra teorie obvodů FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **24.01.2020** Termín odevzdání diplomové práce: _____

Platnost zadání diplomové práce: **30.09.2021**

doc. Ing. Petr Pollák, CSc. podpis vedoucí(ho) práce	doc. Ing. Josef Dobeš, CSc. podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomantka bere na vědomí, že je povinna vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

. _____
Datum převzetí zadání

_____
Podpis studentky

Prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

21. května 2020, Praha

.....
Bc. Kristýna Žáková

Abstrakt

Hlavním cílem této práce bylo vytvořit demonstrační implementaci rozpoznávače izolovaných slov na bázi GMM-HMM v programovém prostředí MATLAB. Praktické využití takové realizace tkví zejména v možnosti představení materiálu pro výukové účely, důraz je tak kladen zvláště na popis principu a postupů tvorby rozpoznávacího systému se skrytými Markovovými modely. Vytvořený rozpoznávač pokrývá český a anglický jazyk, k trénování modelů pro tyto jazyky byly využity databáze TIMIT a SPEECON. Systém je však sestaven univerzálně a je možné na vstupu využít jiných dostatečně bohatých řečových korpusů s fonetickým přepisem na úrovni hlásek. Implementace umožňuje vytváření akustických modelů monofónů a slov z dostupné databáze, představuje proces dekódování a prezentuje uživateli výsledky i mezivýsledky rozpoznávacího procesu. Vytvořená implementace byla testována na signálech z již uvedených řečových databází, ale také na řečových signálech nahraných v rámci této práce.

Klíčová slova: HMM, GMM, GMM-HMM, rozpoznávání slov, zpracování řeči, MATLAB

Abstract

The main goal of this work was to create a demonstration implementation of an isolated word recognizer based on GMM-HMM in the MATLAB programming environment. The practical use of such realization lies mainly in the possibility of presenting the material for educational purposes. The emphasis is placed on the description of the principle and procedures of creating a recognition system with Hidden Markov Models. The created recognizer covers the Czech and English languages; the TIMIT and SPEECON databases were used to train the respective models. However, the system is compiled universally, and it is possible to use other sufficiently rich speech corpora with phonetic transcription at the level of phones as the input. The implementation enables the creation of acoustic models of monophones and words based on available speech corpora, presents the decoding process and shows the final and intermediate results of the recognition process. The created implementation was tested on signals from the already mentioned speech databases, but also on speech signals recorded as a part of this thesis.

Key words: HMM, GMM, GMM-HMM, word recognition, speech processing, MATLAB

Poděkování

Tímto bych ráda poděkovala vedoucímu své diplomové práce, Doc. Ing. Petru Pollákovi, CSc. za všestrannou pomoc, množství cenných rad, podnětů a doporučení, zároveň také za velkou trpělivost a vstřícnost při konzultacích. Stejně tak velké díky patří všem mým přátelům a rodině za ochotu, se kterou se podíleli na vytvoření vlastního souboru testovacích nahrávek.

Seznam tabulek

Seznam obrázků

Seznam použitých zkratek

ANN Artificial Neural Network. 7

CNN Convolutional Neural Network. 7

DARPA-ISTO Defense Advanced Research Projects Agency - Information Science and Technology Office. 13

DCT Discrete Cosine Transform. 19

DFT Discrete Fourier Transform. 8

DTW Dynamic Time Warping. 9

GMM Gaussian Mixture Model. 7, 12, 15, 16, 21–24, 27, 36, 38, 46

GUI Graphic User Interface. 33

HMM Hidden Markov Model. 9–12, 15, 16, 20, 21, 24, 26, 27, 29, 36–38

LPC Linear Predictive Coding. 4, 8

LSTM Long Short-Term Memory. 7

MFCC Mel-Frequency Cepstrum Coefficient. 5, 8, 20, 29, 46

MIT Massachusetts Institute of Technology. 13

NIST National Institute of Standards and Technology. 13

PLP Perceptual Linear Prediction. 5, 8, 19

SpeeCon Speech Driven Interfaces for Consumer Devices. 13

VUT Vysoké učení technické v Brně. 14

ČVUT České vysoké učení technické v Praze. 13, 14

Obsah

Kapitola 1

Úvod

Svět techniky se v současné době rozvíjí neuvěřitelnou rychlostí nad hranice představivosti obyčejného člověka. Auta i lodě začínají jezdit zcela samy, úrodu na polích zemědělci kontrolují družicemi z vesmíru, výrobní linky jsou automatizované a ovládané roboty, neurochirurgové mluví o možné transplantaci hlavy v dalším desetiletí. Vše kolem nás se vyvíjí a lidstvo je v každodenním bytí závislé na pokroku, zjednodušování a zpříjemňování aktivit, jež jsou pro nás nezbytné.

Právě rozpoznávání řeči, jako jeden z mnoha rozvíjejících se oborů, je pro mnoho z těchto moderních systémů velmi častou aplikací. Například automobilový průmysl je v rozpoznávání řeči velkým tahounem. Automatická identifikace, vytáčení a asistence v autech představují pro řidiče nejen usnadnění jízdy, ale také komfort s udržením stávajících návyků při mnohem větším bezpečí [1], [2].

Ani telekomunikace nejsou v tomto oboru pozadu - na zákaznických informačních linkách již potkáváme hlasy tzv. „virtuálních asistentek“, které zvládnou poradit a pomoci volajícímu, a to nejen za hranicemi naší země - tyto systémy již zvládají poslouchat a komunikovat v českém i slovenském jazyce [3]. Volající se tak spojí s živou bytostí na informační lince až v případě, že jeho problém či žádost není zcela běžná.

Velký potenciál se pak skrývá v příležitosti umožnit komfortnější fungování zdravotně postiženým či odstraňovat jazykové bariéry mezi lidmi napříč státy a kontinenty - s tím pomáhají aplikace jako je Skype Translator, překládající aktivně 8 mluvených řečí, schopný se jako systém dále učit s četností užívání a s pomocí strojového učení (machine learning), nebo také služby startupů jako jsou Voiceitt či Talkitt. Zajímavá je také headstarter kampaň na Waverly Labs, plánující překlad mezi jazyky v reálném čase přes sluchátka uživatele [4]. I toto řešení je však i přes velké množství objednávek limitováno - je stavěno na běžnou řeč, tedy vynechává jedince s nářečími či řečovými vadami. Obsluhové kiosky a hlasoví asistenti jsou pak nejen pro sluchově postižené stěžejní [5].

Hlasové systémy však mohou v mnohém ulehčit každému a zvýšit pohodlnost života. Pomocí hlasové asistentky Siri od společnosti Apple je možné potvrdit žádané bankovní převody [6], stejně lze přistupovat ke svým bankovním informacím, transakcím a zůstatkům na účtech [7] - pouze svým hlasem. Ve finančnictví je však do budoucna pro zlepšení uživatelského zážitku nutné trénovat modely znalé bankovní terminologie, která je od běžné mluvy odlišná - to může být podobné pro větší množství rozličných odvětví.

V oblasti nákupů, tzv. „retailu“, však hlasová ovládání fungují i bez nutnosti trénování speciálních řečových modelů - už teď můžeme jen hlasovým příkazem objednávat různé zboží. Do hlasově ovládaných nákupních procesů dosud nejvíce investoval Amazon (Amazon Echo), ale očekává se, že v této oblasti dojde k dalšímu velkému nárůstu, viz [8].

Je jasné, že problematika rozpoznávání řeči je obecně velice složitá - v současné době se zjednodušuje a rozšiřuje mezi uživatele zejména díky užití technologií hlubokého učení (deep learning) [9]. Je to doména důležitá nejen pro průmysl, ale také v akademické a vědecké sféře. Jedná se o oblast širokého uplatnění pro mnohé absolventy studia slaboproudé elektrotechniky,

zejména pak komunikačních a informatických oborů. Cílem této práce je tedy vytvořit vhodné demonstrační prostředí pro výukové účely, přibližující principy systémů rozpoznávačů řeči na bázi skrytých Markovových modelů (HMM), které tvoří jádro aktuálně používaných řešení v praktických aplikacích. Pro demonstraci základních vlastností a principů systému založeného na GMM-HMM byla tedy realizována zjednodušená implementace v programovacím prostředí MATLAB. To je zároveň častým prostředkem výuky, se kterým se studenti souvisejících oborů při studiu setkávají.

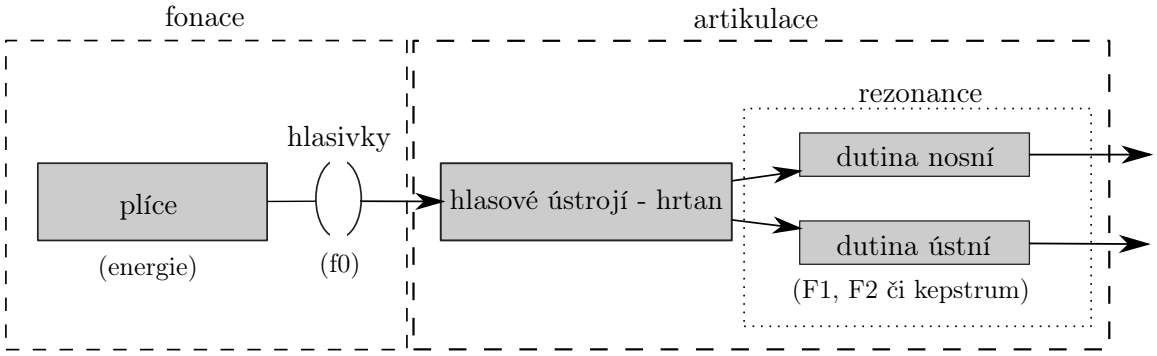
Kapitola s číslem 2 stručně seznamuje s problematikou rozpoznávání řeči a představuje základní principy a algoritmy s ní spjaté. Ve třetí části je pak představena samotná implementace, vysvětleny jednotlivé kroky a postupy, jichž bylo při realizace práce využito. Čtvrtá část práce obsahuje diskuzi nad dosaženými výsledky a některé z výsledků prezentuje s ohledem na využití pro výukové účely. Vychází jak z trénovací, tak z rozpoznávací fáze procesu - jak by bylo pro prezentaci při výuce záhodno - avšak zaměřuje se také na testování výsledného rozpoznávače na dostupných signálech. Zároveň seznamuje s využitými datovými zdroji, databázemi a samotnými daty, jelikož nejen ty jsou pro rozpoznávač klíčové.

Kapitola 2

Řeč a rozpoznávání řeči

Tato kapitola stručně shrnuje základní principy rozpoznávání řeči. Úvodem zmiňuje základní vlastnosti řeči, jak dochází k její produkci a transformaci řečového signálu na data nesoucí informační obsah, který je důležitý pro následné zpracování.

2.1 Produkce řeči



Obrázek 2.1: Akustický model řeči

Princip tvorby mluvené řeči v lidském hlasovém traktu je naznačen ve schématu na obr. 2.1. Vznikající řeč, řečové akustické kmity, pochází z hlasového ústrojí uloženého v hrtanu a jejím zdrojem je vzduch hnaný z plic skrz hlasivky (hlasovou štěrbinu a vokální trakt).

Vzduch je modulován kmitáním hlasivek, jež se mohou pomocí svalů přibližovat či oddalovat. Základní tón lidského hlasu, tedy frekvence kmitů hlasivek, částečně způsobuje jedinečnost hlasu každého jednotlivce a odvozují se od něj všechny tónové složky řeči. Změny v rychlosti kmitání hlasivek vnímáme jako změny v základní frekvenci f0 (resp. v periodě hlasivkového tónu). Výška hlasu (základní tón) je ovlivněna fyzikálními vlastnostmi hlasivek - jejich délkou, hmotností a pružností.

Hlasivky významně ovlivňují tvorbu řeči vlastním chováním. Kmitají-li, vytváří se v hlasovém ústrojí hlásky mající tónové složky (samohlásky či znělé konsonanty). Pokud hlasivky nekmitají, tvoří pouze štěrbinu pro procházející vzduch, který třením o artikulační orgány - jazyk, zuby, rty - tvoří šum, a tedy neznělé souhlásky.

Některé svrchní harmonické tóny jsou v dutinách vokálního traktu zesilovány (dutiny plní funkci rezonátoru). Frekvence rezonátorů vokálního traktu se nazývají formanty, jsou nejvíce ovlivňovány čelistmi, měkkým patrem a rty. Právě na základě různého frekvenčního obsahu rozlišujeme jednotlivé hlásky, tj. obecné subslovní elementy.

Pro simulaci generování a přenosu řeči ve vokálním traktu je možné využít **diskrétní signálový model** zdroj-filtr, který modeluje řeč kombinací zdroje zvuku a lineárního akustického filtru

(zastupujícího funkci hlasivek) [10]. Díky relativní jednoduchosti je tento model využíván v řadě aplikací, ať už syntéze řeči [11], [12], kódování [13] či například v oblasti telekomunikací [14].

Jako zdroj zvuku se užívá generátor pulsů (pro modelování znělých hlásek) a generátor šumu (pro modelování neznělých hlásek), produkující excitační signál. Excitační signál pak vstupuje do lineárního systému, kde je zesilován a tvarován dle parametrů hlasového ústrojí. V nejjednodušším případě je vokální trakt (filtr) aproximován all-pole filtrem s koeficienty počítanými na bázi LPC. Konvolucí excitačního signálu s impulsovou odezvou filtru pak vzniká syntetizovaná řeč.

Pro rozpoznávání řeči pracujeme především se spektrálními vlastnostmi řeči, jež jsou ovlivňovány ve vokálním traktu. Informaci o vokálním traktu lze modelovat kepstrem, jehož první koeficienty nesou informace o tvaru amplitudového spektra.

2.2 Fonetická reprezentace řeči

Akustický obsah řeči je možné reprezentovat konečnou množinou elementů zvaných fonémy, které mají v konkrétních jazycích rozlišovací funkci. Foném sám o sobě nenese žádný význam, avšak umožňuje od sebe rozlišit jednotlivé významové jednotky. Většina jazyků má mezi 32 a 64 fonémy, spisovná čeština v současné době 39 (obecně přijímaným počtem je 36 fonémů). Foném je hlavním zájmem zkoumání fonologie, nauky o funkci hlásek (schopnosti rozlišovat význam).

Oproti tomu fonetika je věda zkoumající zvukovou stránku řeči, způsob tvorby zvuků, jejich akustické vlastnosti a vnímání. Základním elementem fonetiky je fón (hláska). Hláska je tedy obecnou základní jednotkou řeči, jíž je možné chápat jako konkrétní zvuk (fón) realizující abstraktní jazykovou funkční jednotku (foném). Při realizaci rozpoznávače řeči jsou právě hlásky základními modelovanými akustickými elementy. Jednotlivé jazyky mají kolem 40-100 hlásek, česká fonetická abeceda rozlišuje 44 (resp. 42) hlásek v závislosti na množství rozlišovaných alofónů (variant hlavních hlásek). Anglická fonetická abeceda pak rozlišuje 40-44 hlásek (v některých transkripcích ale až 61), po redukci alofónů je možné pracovat i jen s 39 hláskami.

Při přechodu mezi jednotlivými hláskami nastává prodleva, kdy se artikulační orgány přizpůsobují nově vyslovené hlásce. Délka prodlevy závisí nejen na fyziologických vlastnostech hlasového orgánu, ale také na intonaci, tempu řeči, dokonce i na kontextu po sobě jdoucích hlásek. Tato závislost je označována jako koartikulace. V souvislosti se vzájemným ovlivňováním sousedních hlásek pak hovoříme také například o difónu či trifónu, subslovních jednotkách, které tato ovlivnění popisují [15]. Tyto kontextově závislé hlásky, a to zejména trifóny, se používají pro modelování ve složitějších úlohách, např. při rozpoznávání spojité řeči.

2.3 Vnímání řeči

Zvuk je mechanické vlnění vyskytující se ve formě podélné vlny. Tato zvuková vlna je zachycena ušním boltcem a bez jakýchkoliv změn vedena zevním zvukovodem k bubínku. Ten funguje jako rezonátor. Bubínek a sluchové kůstky (kladívko, kovadlinka a třmínek) přemění zvukovou vlnu na mechanické vibrace, jež jsou z plynného prostředí vnějšího a středního ucha převedeny do kapalného prostředí ucha vnitřního. Můžeme tedy říci, že podstatou sluchu je transformace mechanických zvukových vln na elektrické akční potenciály.

2.3.1 Vnímání frekvence

Frekvenční analýza, ke které dochází na bazilární membráně, může být teoreticky popsána řadou pásmových filtrů (banka filtrů), jejichž kmitočtové odezvy v šířce rostou spolu s frekvencí zvuku a subjektivní vnímání zvuku je tedy nelineární (logaritmicky závislé na frekvenci). Právě nelineární frekvenční analýza, založená na analogii s vnímáním řeči lidským sluchem, je jádrem technik

používaných pro výpočet řečových příznaků. Ty jsou potom vstupem následných klasifikačních algoritmů.

Frekvenční odezvy jednotlivých filtrů se však prolínají, což vychází z faktu, že jednotlivé části bazilární membrány nemohou kmitat nezávisle na sobě. Toto nelineární borcení frekvenční osy se často označuje jako rozklad na kritická pásma. Pro zvuky frekvencí 0-20 kHz většinou stačí rozklad na 25 pásem. Koncept kritických pásem je důležitý pro vnímání hlasitosti, výšky tónu či maskování - je tedy motivací pro digitální reprezentace řečového signálu, které jsou založeny na frekvenčním rozkladu.

V souvislosti s kritickými pásmy hovoříme o stupnicích navržených na základě experimentálního výzkumu, jako je barková, erbová nebo melová. Ve spojitosti s poslední zmíněnou se setkáváme s tzv. mel-frekvenčními kepstrálními koeficienty (MFCC), které představují nejčastější způsob parametrizace spektra při rozpoznávání řeči - sama o sobě se ve fonetice melová stupnice však prakticky již nepoužívá [16]. Právě výpočet melovských kepstrálních koeficientů kompenzuje zmíněné nelineární vnímání frekvencí, a to použitím banky trojúhelníkových pásmových filtrů s lineárním frekvenčním rozložením (mluví se o tzv. melovské frekvenční škále [17]).

2.3.2 Vnímání hlasitosti zvuku

Hlasitost, klíčová vlastnost ve vnímání zvuku, je vyjádřena jako skutečná hladina akustického tlaku tónu (v dB) k vnímané hlasitosti stejného tónu (v jednotkách zvaných fóny) v rozsahu lidského sluchu (což je 20 Hz - 20 kHz). Tento vztah se dá graficky vyjádřit tzv. křivkami stejné hlasitosti, které ukazují fakt, že vnímaná hlasitost souvisí s frekvencí zvuku. Nízké frekvence tedy musí být výrazně intenzivnější než frekvence ve středním či vyšším rozsahu, aby byly vůbec vnímány.

Z křivek stejné hlasitosti je také jasné, že lidský sluch je nejcitlivější na kmitočty v rozsahu cca 100 Hz až 6 kHz s nejvyšší citlivostí kolem 3-4 kHz. Evolučně se tedy sluchový orgán vyvíjel tak, aby byl nejcitlivější na kmitočty obsazené většinou mluveného projevu.

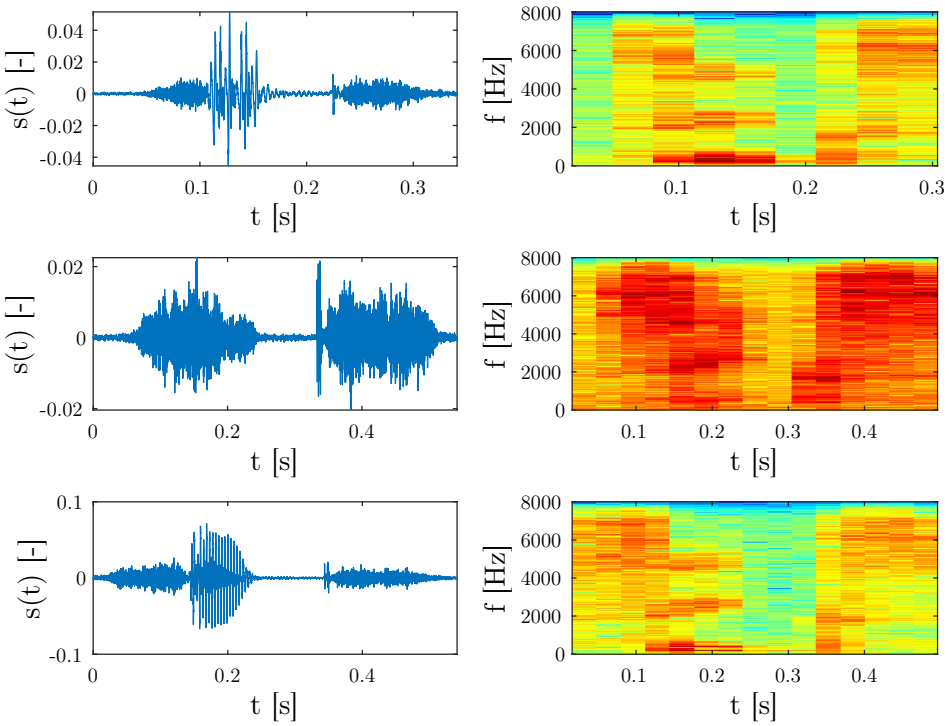
Při mnohých úlohách je důležité mít na paměti, že při šíření vlny z artikulačního ústrojí do volného prostoru dochází k útlumu intenzity zvuku pro vyšší frekvenční složky.

Pro účely rozpoznávání řeči je při modelování třeba vyřazovat irelevantní informace řečového signálu. Může být vhodné použít transformaci signálových spektrálních charakteristik tak, aby odpovídaly charakteristikám lidského sluchového ústrojí, a k redukci oblastí s velmi vysokými a nízkými frekvencemi (převodu do frekvenčního pásma, kde je ucho citlivější) [18]. Tohoto přístupu je využíváno při výpočtu kepstrálních koeficientů na bázi Perceptual Linear Prediction (PLP).

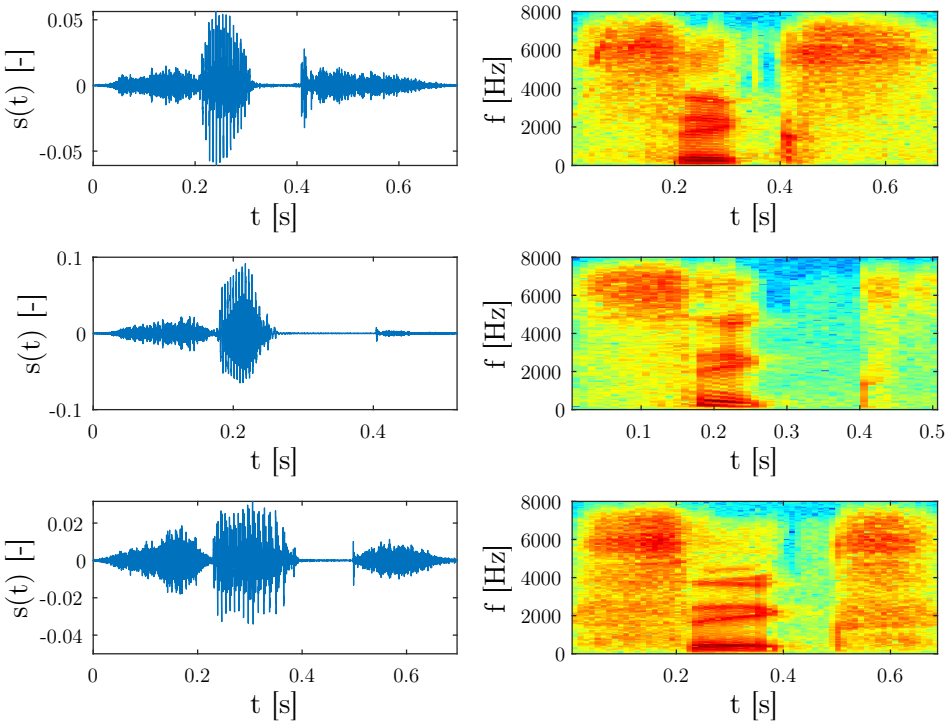
2.4 Strojové rozpoznávání řeči

Rozpoznávání řeči je mezioborovou disciplínou představující a zprostředkovávající převod mluvené řeči automaticky, bez nutnosti lidského přepisu, do textové podoby. Byť tento obor není světově nový a dosahuje dobrých výsledků, nedá se říci, že lze sestrojit zcela bezchybný a ideální rozpoznávač překládající libovolná slova z rozsáhlého (resp. téměř neomezeného) slovníku. Existuje několik důvodů, proč tomu tak je - hlasy různých řečníků jsou odlišné (odlišné uzpůsobení hlasového ústrojí), stejně jako mluva jednoho a téhož řečníka je rozdílná v různých situacích (obr. 2.2, resp. 2.3) - hlasitost, zdraví či emoce řečníka se mohou negativně projevovat na úspěšnosti rozpoznávání. Další příčinou, která může rozpoznávání značně ztížit, resp. velmi nepříznivě ovlivnit, je měnící se akustické pozadí řeči. To bylo demonstrováno například v [19], kde se úspěšnost rozpoznávání v hlučném prostředí pohybuje na 31 % z téměř 100 %, které jsou hlášeny po kompenzaci.

Publikace [20] uvádí, že nejtěžší úlohou je rozpoznávat řečové signály, které přicházejí od zcela různých mluvčích (takový rozpoznávač je pak tzv. nezávislý na mluvčím) a že rozpoznávače, které obecně dosahují nejvyšší úspěšnosti, jsou naopak na mluvčím závislé. Existuje však několik



Obrázek 2.2: Variability řeči: slovo „six“, jeden mluvčí - řečové signály a spektrogramy



Obrázek 2.3: Variability řeči: slovo „six“, 3 mluvčí - řečové signály a spektrogramy

přístupů, se kterými je možné efektivní nezávislý rozpoznávací systém vytvořit. Jedním z nich je použití většího množství reprezentací každé promluvy k zachycení variací mezi mluvčími[1] – promluvy jsou vysloveny mnoha řečníky a po jejich analýze je vytvořen jakýsi „prototyp“, který se pro rozpoznávání používá.

Rozpoznávací systémy jako takové je možné rozdělit do dvou skupin – rozpoznávače izolovaných slov a rozpoznávače plynulé řeči. Pod rozpoznáváním izolovaných slov myslíme rozeznávání povelů, příkazů a slov, které jsou zřetelně odděleny mezerou na začátku a na konci promluvy. Je proto možné daný řečový úsek jednoznačně detekovat a rozpoznávat jako celek. Jedná se o nejjednodušší formu rozpoznávání, jelikož nalezení počátků a konců slov je relativně snadnou úlohou, výslovnost slova zároveň nijak není ovlivňována okolními výrazy. Délka takto odděleně vyslovených slov je často až o 50 % delší, než když jsou vyslovena v plynulé řeči [21], a tedy řečové parametry jsou v promluvě lépe a rovnoměrněji rozloženy, začátky ani konce slov nebývají polknuty atd. Rozpoznávání plynulé řeči je značně složitější úlohou. Souvislá promluva není oddělena jasnými pauzami, je obtížné najít počátky a konce jednotlivých slov (ty jsou zároveň ovlivněny slovy předchozími a následujícími, stejně jako u koartikulace hlásek je produkce daného fonému ovlivněna produkcí těch okolních) a důraz při výslovnosti slov ve větě se různí. Výrazy, jež nesou obsah (jako slovesa či podstatná jména), jsou často vysloveny mnohem důrazněji, než ta slova, která jsou spíše funkční – předložky, zájmena. Ty jsou tak často ve spojité promluvě vysloveny ne zcela správně a je obtížné je korektně rozeznávat [20].

Systémy pro rozpoznávání řeči jsou definovány mimo jiné také slovníkem výrazů, podle nichž jsou natrénovány a s nimiž následně provádějí úlohu samotného rozpoznávání – „mapování“ mezi sekvencemi analyzované řeči a mezi sekvencemi symbolů ve slovníku. Obsahuje-li užitý slovník 1000 či více slov, označuje se jako velký slovník. Takto objemná slovní zásoba však s sebou nese řadu problémů, jako je například zaměnitelnost slovníkových výrazů, která v takovém množství podstatně roste. Oproti tomu s malými slovníky je možné modelovat každé slovo individuálně a ukládat separátně také jejich parametry. To ale přestává být se zvětšujícím se objemem slovníků možné – místo toho dochází k definicím elementárních subslovních jednotek, ze kterých poté modely slov bývají složeny. Takový přístup sice obvykle vede ke zhoršené úspěšnosti rozpoznávání, avšak jedná se o optimální řešení vzhledem k dostupným výpočetním kapacitám a úložištím [20]. Složitost vyhledávání (rozpoznávání, mapování sekvencí řeči a slovníku) bývá dalším parametrem, jež je velikostí slovníku ovlivněn.

Každý systém rozpoznávání řeči může být složen z jednotlivých modulů, které implementují dílčí stěžejní úlohy, které jsou popsány v následujících podkapitolách.

Díky technologickým pokrokům a větším možnostem výpočetní síly se v současné době některé z těchto stěžejních úloh (ne-li většina) řeší pomocí neuronových sítí (ANN). Obecně lze říci, že všechny současně vytvářené a efektivně používané rozpoznávací systémy právě výpočtů s neuronovými sítěmi využívají. V praxi se používají zejména pro efektivní výpočty pravděpodobnostních rozdělení velmi rozsáhlých databází, tj. nahrazují výpočet pravděpodobnosti stavu pomocí hustoty pravděpodobnosti na bázi GMM. Výhodou pak je, že umožňují hromadný výpočet pravděpodobností pro všechny subslovní elementy v daném okamžiku, resp. řetězení vstupních dat pro zvýšení kontextové informace. Pro trénování ANN je však potřeba velkého množství trénovacích dat.

V současné době jsou trendem v rozpoznávání plynulé řeči pomocí neuronových sítí tzv. *End-to-End* rozpoznávací systémy. Ty používají výhradně neuronovou síť pro přímé mapování snímaného zvuku na znaky či slova. Zjednodušeně lze říci, že jednotlivé vrstvy různého typu (CNN, LSTM, full-connected) realizují dílčí moduly jako výpočet příznaků či akustické a jazykové modelování. Díky speciálním algoritmům hlubokého učení není potřeba přílišné (resp. žádné) odborné znalosti a jejich vytváření a trénování je tedy jednodušší [22]. Předpokladem je opět dostupnost velkého (výrazně většího) množství trénovacích dat. Obecně využitím ANN pro rozpoznávání řeči se do hloubky zabývají například publikace [23], [24], [25] a další.

[1] Tohoto přístupu bylo využito v této práci.

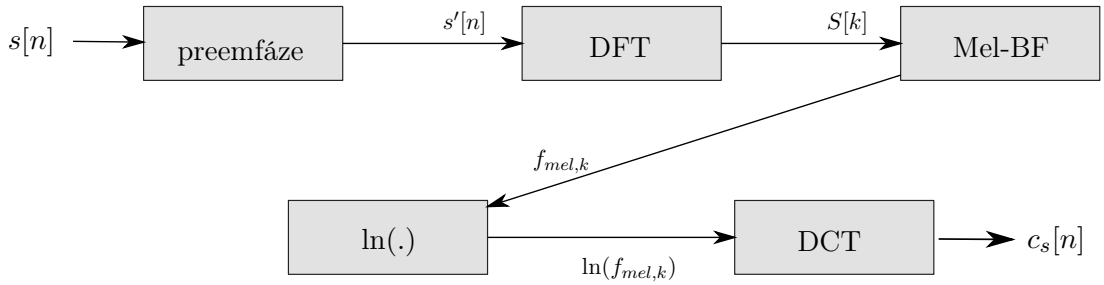
2.4.1 Parametrizace

Pro rozpoznávání není důležitý celý řečový signál, práce s ním v celém procesu rozpoznávání by zároveň byla výkonově i časově náročná. V úlohách rozpoznávání řeči se řečové signály nejprve předzpracovávají a extrahují se takové příznaky, které jsou pro rozpoznávání vhodné. Získání příznakových vektorů pomocí metod krátkodobé signálové analýzy (jinými slovy parametrizace vstupního signálu) je prvním důležitým modulem každého rozpoznávače řeči.

Parametrizace vychází z již uvedeného modelu produkce lidské řeči a z faktu, že řeč je kvazistacionární signál, jež je možné považovat za stacionární pro časové úseky 10-30 ms. Díky tomu je možné řeč rozdělit na menší jednotky a s nimi dále parametricky pracovat. Nejpoužívanější metodou reprezentace řečového signálu je tzv. kepstrum, definované jako inverzní DFT logaritmu amplitudy DFT signálu, viz. rovnice 2.1.

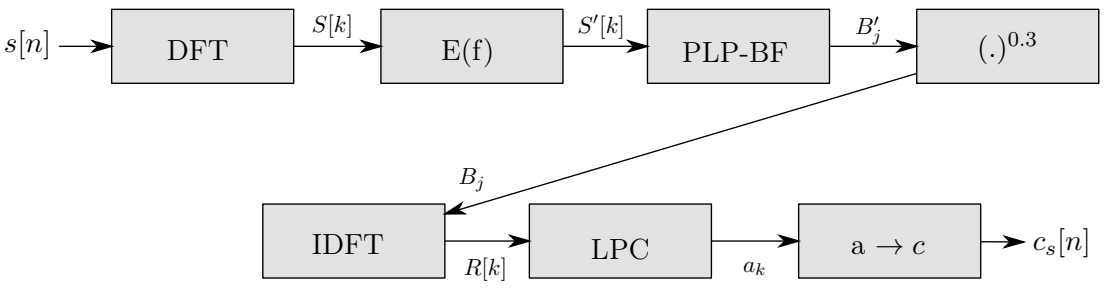
$$c_s[n] = DFT^{-1}\{ln|S[k]|\} = \sum_{k=0}^{N-1} ln|S[k]|e^{jkn\frac{2\pi}{N}} \tag{2.1}$$

Při výpočtu kepstra se odděluje podstatná spektrální informace, tj. frekvenční odezva hlasového traktu, která je komprimovaná v několika prvních kepstrálních koeficientech (typicky cca 12), od informace nepodstatné pro rozlišení hlásek (buzení). Do výpočtu je možné zahrnout také nelineární vnímání frekvence, což se děje při výpočtu Mel-Frequency Cepstrum Coefficient (MFCC), kdy se nepracuje s lineární frekvenční osou, ale převádíme frekvenční osu na osu v jednotkách mel. K celému výpočtu MFCC koeficientů dochází dle schématu na obr. 2.4.



Obrázek 2.4: Schéma výpočtu MFCC příznaků

Další možností výpočtu řečových příznaků je metoda Perceptual Linear Prediction (PLP) (blokové schéma viz obr. 2.5), která představuje rozšíření LPC o práci s nelineární spektrální analýzou (kompenzace z křivek stejné hlasitosti, umocnění energie v jednotlivých pásmech pro aplikaci zákona slyšení, představující nelineární závislost mezi skutečnou intenzitou a vnímanou hlasitostí, a o převod lineární frekvenční osy do Barkovy frekvenční osy). Díky tomu, že k výpočtu spektra dochází pomocí lineární predikce, má takový výpočet menší robustnost.



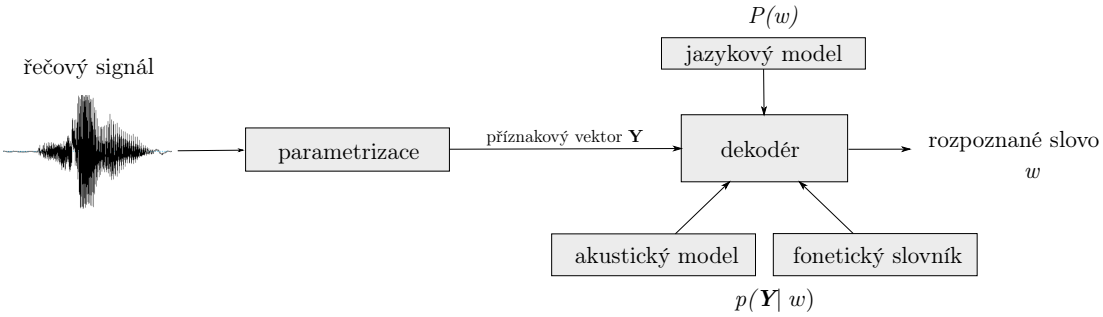
Obrázek 2.5: Schéma výpočtu PLP příznaků

2.4.2 Klasifikace

Algoritmus klasifikace je klíčovým prvkem rozpoznávacího systému. Dochází v něm k rozhodnutí o začlenění rozpoznávaného slova (či analyzovaného řečového segmentu) do jedné ze tříd referenční databáze zdrojů, tedy slovníku. Rozpoznáváme-li kupříkladu izolovaná slova, klasifikátor pracuje se sekvencí příznakových vektorů odpovídajících rozpoznávanému slovu a porovnává, které ze tříd (resp. kterému ze vzorů) ze slovníku se nejvíce sekvence podobají, případně rozhodne, že rozpoznávané slovo nepřiřadí žádnému slovu ze slovníku.

Volba algoritmu klasifikace pak zcela ovlivňuje volbu referencí pro rozpoznávání. Lidská promluva je výrazně závislá na řečníkovi samotném - je ovlivňována mnoha faktory - a vždy se tak liší nejen délkou promluvy, ale také nářečím, intonací aj. S přihlédnutím k tomu se nabízí zejména využití skrytých Markovových modelů (Hidden Markov Model (HMM)) či dynamického borcení času (Dynamic Time Warping (DTW)).

DTW hledá minimální vzdálenost (zpravidla euklidovskou) mezi posloupností příznaků neznámé promluvy a posloupností příznaků referenčního promluvy, které jsou na sebe mapovány. Reference s nejkratší vzdáleností je pak považována za výsledek rozpoznávání. Tento algoritmus je uplatnitelný jen v nejjednodušších aplikacích, pro rozpoznávání řeči se dnes využívá již jen sporadicky.



Obrázek 2.6: Zjednodušené schéma rozpoznávače založeného na HMM

Dnešním standardem pro aplikace rozpoznávání řeči je modelování na bázi HMM. Základní schéma rozpoznávače řeči pracujícího s pomocí metody skrytých Markovových modelů je vyjádřeno na obr. 2.6. Analyzované signály, po parametrizaci vyjádřené příznakovými vektory $\mathbf{Y} = y_1,y_2,...,y_T$, vstupují do dekodéru. Cílem je vzhledem k daným řečovým signálům $\mathbf{Y}$ nalézt takové slovo w ze slovníku, pro které je maximální $\mathrm{P}(w|\mathbf{Y})$ - pravděpodobnost, že vysloveným slovem je slovo w, když jsou na vstupu pozorované řečové signály $\mathbf{Y}$. To je vyjádřeno následujícím výpočtem:

$$\hat{\boldsymbol{w}} = \underset{w}{\operatorname{argmax}}\{P(w \mid \mathbf{Y})\} \tag{2.2}$$

Tuto pravděpodobnost však není snadné modelovat přímo (ačkoliv některé systémy se o to snaží, viz [26]), a tak se upravuje dle Bayesova pravidla pro podmíněnou pravděpodobnost do tvaru

$$\hat{\boldsymbol{w}} = \underset{w}{\operatorname{argmax}}\{p(\mathbf{Y} \mid w)P(w)\}, \tag{2.3}$$

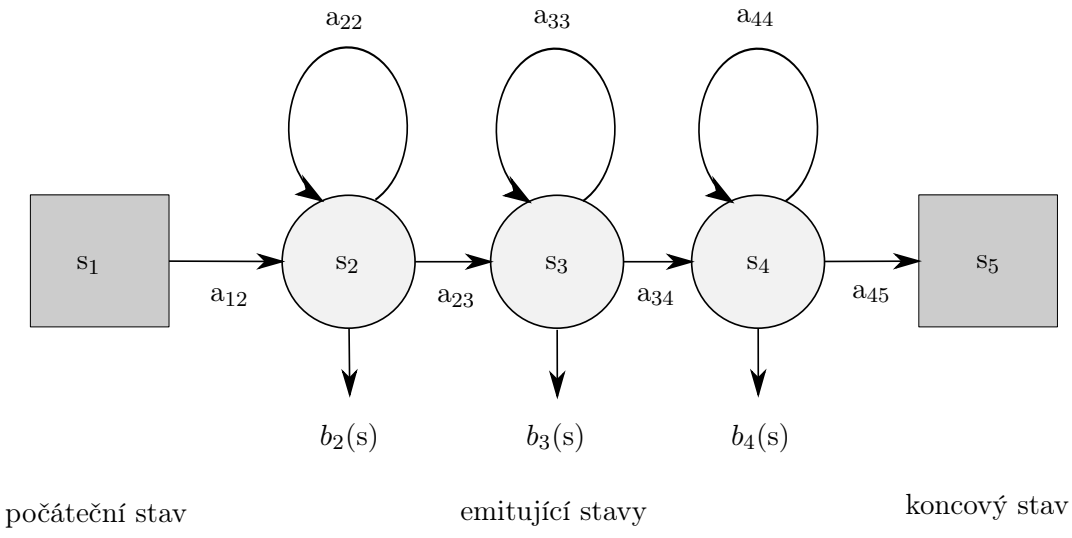
kde je pravděpodobnost $p(\boldsymbol{Y}/w)$ determinována akustickým modelem a $P(w)$ jazykovým modelem, známa ze slovníku. Přímý výpočet sdružené podmíněné pravděpodobnosti vektoru $\mathbf{Y}$, $p(\boldsymbol{Y}/w)$, resp. $p(y_1,y_2,...,y_T/w)$ není téměř možný, využívá se místo ní právě akustických modelů, skrytých Markovových modelů, a odhadují se jejich parametry.

Předpokládáme, že posloupnost řečových vektorů $\mathbf{Y}$ řeči, po částech stacionárního signálu, je generována skrytým Markovovým modelem (řetězcem skládájícím se z N stavů), přičemž každý z nich má jiný soubor statických charakteristik. Základem N-stavového HMM je konečný stavový automat, tj. pravděpodobnostní model popisující posloupnosti stavů a přechody mezi nimi.

Vztahy (vnitřní vazby) mezi stavy (uzly) jsou popsány pomocí matice pravděpodobnosti přechodů **A**. Každému uzlu (stavu) je příslušný vždy jeden příznakový vektor. To je možné zjednodušeně vyjádřit následujícími parametry:

- počtem stavů modelu,
- hustotní funkci výstupních vektorů pozorování $\mathbf{B} = b_j(y)$, jejíž hodnoty určují, s jakou pravděpodobností je generován vektor y příslušný stavu j,
- maticí přechodů mezi jednotlivými stavy $\mathbf{A} = (a_{i,j})$, jejíž prvky určují, s jakou pravděpodobností přechází systém ze stavu i v libovolném čase t do stavu j v čase $t+1$.

Obecné HMM uvažují všechny možné přechody mezi uzly, pro signály s časově postupujícím průběhem (kde nepředpokládáme zpětnou vazbu) se pak používají levo-pravé modely, které říkají, že je možný přechod z jednoho uzlu do téhož uzlu či do jakéhokoliv následujícího, což odpovídá toku času.



Obrázek 2.7: Jednoduchý pětistavový levo-pravý HMM model

Pro modelování jednotlivých hlásek se v praxi nejčastěji využívá pětistavový levo-pravý dopředný HMM model, mající tři emitující stavy a dva neemitující (první a poslední). Neemitující stavy jsou ty, které slouží zejména pro navazování modelů za sebe, negenerují při přechodu do jiného stavu vektor pozorování. Jednoduché schéma takového modelu je znázorněno na obr. 2.7. Tento model bývá nejvhodnější - je vcelku jednoduchý, robustní a jeho stavy dobře reprezentují části hlásek.

Modely se musí vytvořit pro všechna slova či subslovní elementy, které se budou při rozpoznávání využívat, pro modelování slov lze řetězit modely hlásek. Pro rozpoznávání řečových sekvencí se pak hledá model, který generuje toto slovo (resp. jeho příznakové vektory **Y**) s největší pravděpodobností. Tuto pravděpodobnost je možné vypočítat například **Viterbiho algoritmem** [27], [28].

Aby byly všechny modely připraveny pro použití k rozpoznávání, je nutné pro každý existující model určit všechny jeho parametry - matici pravděpodobnosti přechodů **A** i parametry hustot pravděpodobností jednotlivých stavů **B**. Je možné tyto parametry pouze incializovat (a to buď s použitím náhodného obsahu, či v závislosti na znalostech o trénovací množině dat), nebo je po inicializaci upravovat pro dosažení přesnějších a lepších výsledků, resp. přetrénovávat je. Pro to se využívá například cyklicky opakovaného odhadu, **Baum-Welchova algoritmu** [29], [30].

Metoda HMM je rozsáhlá a vhodná také pro rozpoznávače s velkým slovníkem. V této práci je metoda aplikována na problematiku rozpoznávání izolovaných slov, její rozšíření a zobecnění na rozpoznávání plynulé řeči je však možné - a je také široce používané.

Mimo aplikace v oblasti rozpoznávání řeči se HMM využívá také v bioinformatice [31], [32], [33], pro modelování jazyků [34], prostorových [35] a obrazových dat [36]. Struktury využitých HMM modelů však mohou být jiné.

Kapitola 3

Demonstrační implementace v MATLABu

Realizovaná implementace byla vytvořena zejména s cílem využití pro výukové účely a je velmi zjednodušená. Neklade si za cíl prezentovat ideální realizaci či nejpřesnější výsledky rozpoznávání, ale zejména popis principu HMM a jednotlivých kroků spojených s dílčími úlohami pro sestavení rozpoznávacího systému na těchto modelech založených. Úplná implementace komplexního rozpoznávače, který by mohl být využíván v praktických aplikacích, je složitá, v systému MATLAB nerealizovatelná a pro účely získání základního přehledu nepříliš vhodná. V rámci implementace byl vytvořen rozpoznávací systém pro izolovaná slova, a to zejména z důvodu časové a datové náročnosti složitějších a komplexnějších rozpoznávačů, ale také z hlediska snadnější demonstrace principů rozpoznávání na bázi GMM-HMM.

3.1 Řečové databáze pro trénování

Existence a příprava trénovacích dat je pro konstrukci rozpoznávacích systémů zcela zásadní a jedná se o základní framework pro další práci s rozpoznáváním. Trénovacími daty je myšlen takový korpus řečových nahrávek, který je dostatečně obsáhlý a foneticky bohatý, případně rozmanitý (pokud je cílem rozpoznávací systém nezávislý na mluvčím). Jelikož byl v rámci práce realizován demonstrační rozpoznávací systém ve dvou jazycích, pracuje se také se dvěma řečovými databázemi.[1]

3.1.1 TIMIT

V práci byl pro realizaci anglické mutace rozpoznávače použit řečový korpus TIMIT, který obsahuje řečové signály namluvené mužskými i ženskými řečníky americkou angličtinou (v 8 dialektech). Kromě běžné anotace obsahu promluv (ortografické transkripce) však obsahuje také fonetický přepis s přesným vymezením hranic jednotlivých hlásek. Každý z 630 řečníků nahrál 10 foneticky bohatých vět do nahrávky kódované 16-bit se vzorkovací frekvencí 16 kHz, jedná se tedy o vcelku kvalitní zvukové záznamy. Celý soubor je rozdělen na trénovací a testovací sadu dostatečně bohatou pro příslušné fáze (viz tab. 3.1). Trénovací set (dle dokumentace přibližně 70 až 80 % celého korpusu) je vybrán z nahrávek TIMIT tak, aby se každá z hlásek objevila v počtu dostatečném pro zpracování. V ideálním případě by se žádná z trénovacích vět neměla objevit v testování (k zaručení nulové duplicity a nezkreslených výsledků). Stejně tak jsou v trénovacím i testovacím setu dostatečně reprezentovány jak mužské, tak ženské hlasy různých dialektů.

Pro využití korpusu v realizované demonstrační implementaci bylo mimo nahrávek samotných (ve formátu *.wav*) důležité také použití jsou jejich přepisů:

[1] Tyto databáze byly v rámci práce použity zejména pro trénovací, ale také pro testovací fázi.

Tabulka 3.1: Testovací a trénovací set korpusu TIMIT

Set	Počet řečníků	Počet namluvených vět
Trénování	462	3696
Hlavní testovací	24	192
Kompletní testovací	168	1344

Tabulka 3.2: Ukázka přepisu TIMIT nahrávek

.wrd	**.phn**
2090 9080 elderly	0 2090 h#
9080 14726 people	2090 2783 q
14726 17039 are	2783 4010 eh
17039 23960 often	4010 4929 l
23960 36379 excluded	4929 5560 d
	5560 6851 axr
	6851 7477 l
	7477 9080 iy
	...
.txt: 0 37888 Elderly people are often excluded.	

- ortografická transkripce (*.txt*) řečené promluvy,
- časově uspořádaná transkripce slov (*.wrd*) s definovanými hranicemi jednotlivých slov,
- časově uspořádaná fonetická transkripce (*.phn*).

Rozdíl v těchto souborech je patrný dle uvedeného příkladu v tab. 3.2, který patří k nahrávce **test\dr5\fhew0\sx43** (soubor je formátován jako *<číslo počátečního vzorku> <číslo posledního vzorku> <přepis>*). Zatímco ortografická transkripce a transkripce slov byly důležité až ve fázi testování systému (pro vyhledání a vyjmutí celých slov, které mají být rozpoznávány, z jednotlivých vět), fonetická transkripce byla velice důležitá pro parametrizaci a tvorbu modelů hlásek.

Databáze byla nahrána pod záštitou iniciativy DARPA-ISTO, a to společností Texas Instruments. Massachusetts Institute of Technology (MIT) pak pracoval na transkripci a výsledná práce je vydána, spravována a distribuována NIST, Národním institutem standardů a technologie. Dalším účastníkem při tvorbě korpusu byla společnost SRI International. Databáze byla na CD-ROMu vydána v roce 1988 a je dostupná na základě zakoupené licence.[2]

Více informací, podrobností a technických specifikací o řečovém korpusu je možné najít například v [37].

3.1.2 SPEECON

Pro českou variantu rozpoznávače bylo v práci využito české databáze řečových signálů vytvořených v rámci projektu Speech Driven Interfaces for Consumer Devices (SpeeCon). Ty byly nahrány za pomoci celkem 550 dospělých a 50 dětských mluvčích různého věku, ženského i mužského pohlaví, v 5 odlišných prostředích. Všechny nahrávky byly nahrány ve 4 kanálech, s 16-bit kódováním a 16 kHz vzorkovací frekvencí.

Korpus SPEECON je velmi rozsáhlý, v práci bylo využito pouze jeho částí z prostředí *office* a *entertainment* (obsahují menší množství šumu na pozadí) - a to konkrétně zvukových souborů s extenzí *.CS0* (které byly nahrávány z největší blízkosti mikrofonem v headsetu). Potřebný fonetický přepis pro účely této práce není v databázi dostupný, ta obsahuje pouze ortografickou a

[2] ČVUT je držitelem této licence.

fonetickou transkripci každé promluvy, avšak bez určení hranic hlásek. Fonetický přepis ve formátu *.phn*, viz tab. 3.3[3], byl k databázi vytvořen pomocí automatické segmentace. Není tedy extrémně přesný, což mírně jednotlivé modely hlásek ovlivňuje a má zároveň vliv na výsledky rozpoznávání.

Tabulka 3.3: Ukázka přepisu SPEECON nahrávek

.phn
0 19840 sil
19840 23200 o
23200 24320 t
24320 25600 a
25600 26560 z
26560 28480 nn
28480 30080 ii
30080 33280 k
33280 42720 sil
Celá promluva: 0 42720 Otazník

Databáze byla vytvořena Ústavem počítačové grafiky a multimédií na brněnském VUT, Fakultou elektrotechnickou na ČVUT a německou firmou Castel, nezávislou validaci měla na starosti společnost SPEX (Holandsko). Byla vydána na celkem 23 DVD nosičích v roce 2009 a je dostupná po zaplacení licenčního poplatku.[4]

3.2 Programové prostředí MATLAB

MATLAB je interaktivním programovým prostředím a skriptovacím jazykem, který byl původně určen zejména pro matematické účely, avšak dnes je díky široké paletě funkcí a rozšíření používán pro řadu aplikací. Je využíván pro vědecké a výzkumné účely (ve veřejném i soukromém sektoru), ale také v akademických sférách[5], zejména v oblasti technických oborů a ekonomie.

Základní strukturou při výpočtech jsou v tomto prostředí matice, prvky pole – nejen čísla, ale také proměnné a složitější struktury, znaky či symbolické proměnné. Jednotlivé proměnné se nedeklarují. Programování probíhá v uživatelsky ovladatelném prostředí a zahrnuje jak výpočty, tak vizualizaci a mnoho podpůrných funkcí z toolboxů.

Pro realizaci rozpoznávačů řeči v reálném prostředí by však MATLAB nikdy nebyl využit. Pro tyto systémy je třeba zpracovávat obrovské množství dat (stovky až tisíce hodin řečových záznamů), využívat složitých specializovaných funkcí, vytvořit robustní prostředí připravené k rozvíjení, využití neuronových sítí. Využít programového prostředí MATLAB, které umožňuje postupné výpočty, vizualizaci celkových i dílčích výsledků a implementaci jednodušších funkcí (které by pro praktické využití nebyly ideální, naopak – spíše krkolomné) je pak vhodné právě pro demonstrační účely a studium problematiky rozpoznávání řeči.

Celá demonstrační implementace byla vytvořena v prostředí MATLAB R2019b[6] a je v něm zcela funkční. Při úplném dodržení struktury využívaných proměnných a modelů je však možné použít vstupy vytvořené z jiných programovacích prostředí – systém je na načtení takových souborů připraven.

[3] Tento přepis patří k nahrávce **BLOCK01_SES013_SA013CK**.

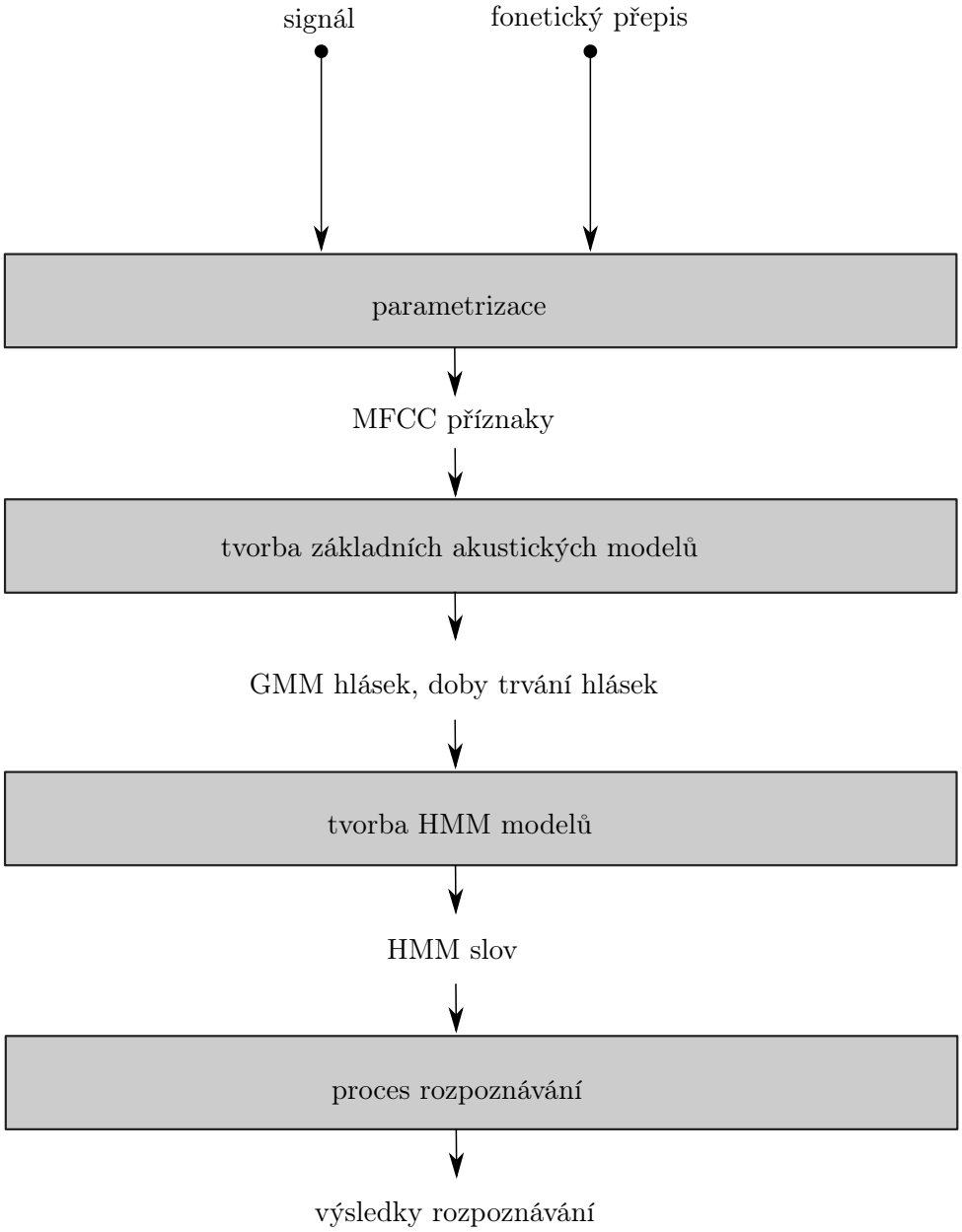
[4] ČVUT je držitelem této licence.

[5] Prostředí MATLAB je světovým standardem pro výuku technických a inženýrských oborů na univerzitách i v průmyslu a je ve vysoké míře využíváno také pro výuku na řadě fakult ČVUT, elektrotechnice nevyjímaje.

[6] ČVUT je držitelem multilicence.

3.3 GMM-HMM systém

Rozpoznávání řeči probíhá ve třech generických fázích – přípravě základních GMM modelů hlásek, vytvoření akustických HMM modelů hlásek a slov (s možným přetrénováním) a nakonec v samotném rozpoznávacím procesu, viz následující zjednodušené schéma na obr. 3.1.



Obrázek 3.1: Zjednodušené schéma rozpoznávacího systému

Každá z fází potřebuje pro své fungování jiné vstupní proměnné. Pro parametrizaci a vytvoření akustických modelů hlásek je obecně nutné systému dodat trénovací databázi promluv s fonetickým přepisem na úrovni hlásek (pro co nejjednodušší proces trénování), k tvorbě akustických modelů slov je pak třeba slovník s fonetickým přepisem na stejné úrovni. Z těchto souborů je pak v implementaci vytvořen zbytek klíčových proměnných – parametrických vektorů hlásek, GMM a HMM hlásek, HMM slov. Ze schématu 3.1 je patrné, že jednotlivé bloky rozpoznávače na sebe navazují a postupně si tyto proměnné předávají. V programové realizaci jsou takové milníky jasně odděleny a je možné načíst proměnné různého obsahu (i z externích zdrojů či ze separátní přípravy) tak, aby byla demonstrace co nejvíce univerzální a nebylo nutné dodržo-

vat pouze ty vstupy, které byly použity v rámci této práce.

Praktická implementace po prvotní inicializaci parametrů[7] a adresových cest začíná importem seznamu trénovacích dat. Z tohoto seznamu se dále zpracovávají postupně všechny nahrávky (s možností akustického přizpůsobení, viz 3.4.2). Dochází k parametrizaci (sekce 3.4.3), případným úpravám (jako je redukce hlásek pro anglickou fonetickou abecedu, sekce 3.4.3), tvorbě akustických modelů (GMM, sekce 3.4.4).

Dále jsou vytvořeny zjednodušené akustické modely (HMM hlásek, sekce 3.5) - pracuje se s jednostavovým modelem (resp. modelem, který má jeden emitující stav a dva neemitující). Na základě takového zjednodušení je možné dále vytvořit jednoduché HMM modely slov (viz sekce 3.5.2) bez nutnosti složitějšího přetrénovávání. Samotné rozpoznávání (včetně celkového zpracování signálu), tedy porovnávání signálu se všemi existujícími modely, následuje v další fázi (sekce 3.6).

Konečný rozpoznávací proces má vždy jen dva vstupy – řečový signál, který chceme rozpoznávat, a modely slov, na které signál mapujeme (a ve kterých se hledá největší podobnost s rozpoznávanou řečí). Tyto modely jsou reprezentovány skrytými Markovovými modely, jejichž hustota pravděpodobnosti je na bázi směsí Gaussových funkcí. Ty představují lepší modelování variability příznaků (například pro různé mluvčí). Taková realizace je označována jako GMM-HMM systém.

3.4 Parametrizace a tvorba GMM modelů hlásek

První důležitou fází tvorby celého systému je zpracovat trénovací databázi nahrávek pro vytvoření základních hláskových GMM modelů – vytvořit parametrické vektory hlásek a z nich pak připravit akustické modely určené k další fázi procesu. Obecně je možné pracovat s různými trénovacími databázemi, pro účely této práce bylo použito zejména databáze TIMIT (pro anglický jazyk) a SPEECON (pro jazyk český). Zpracování těchto dat se od sebe významně neliší, proto je dál pro uvedení příkladů využita a zmíněna zejména anglická verze, se kterou byla realizace vytvořena dříve.

3.4.1 Použitý řečový korpus

Práce maximálně využívá řečových korpusů podrobněji popsaných v kap. 3.1. Pro potřeby využití v samotné tvorbě rozpoznávacího systému byly důležité zejména soubory určené pro trénovací fáze. Jako vstup do implementace je třeba mít připravený soubor ve formátu *.txt*, který kumuluje na jedno místo názvy všech trénovacích souborů tak, aby se s nimi při skriptování dalo automatizovaně pracovat. Tento seznam se do systému nahrává a odkazuje tak na samotné soubory nahrávek, ale také na jejich fonetické přepisy (na úrovni hlásek s odkazy na počáteční a koncové vzorky, viz tab. 3.4).

V kódu je uvedeno nahrávání souboru *timit_list_train_wavs.txt* – ten představuje pro demonstrační implementaci zmíněný připravený seznam všech trénovacích souborů korpusu TIMIT. Pro SPEECON se pak jedná o soubor *list_train_phrich_phns.txt* – seznam všech trénovacích promluv obsahujících foneticky bohaté věty a slova. Nejedná se tedy o seznam úplně všech promluv z korpusu určeného k trénování (pro užití v demu se zdálo jako zcela dostatečné).[8]

3.4.2 Počáteční úpravy

Databáze sloužící pro trénování modelů rozpoznávacího systému byly nahrány se vzorkovací frekvencí 16 kHz (hloubkou 16 bit), stejně jako všechny testovací signály v práci použité (viz část 4.1).

[7] Všechny tyto parametry jsou definovány na úplném počátku skriptu.

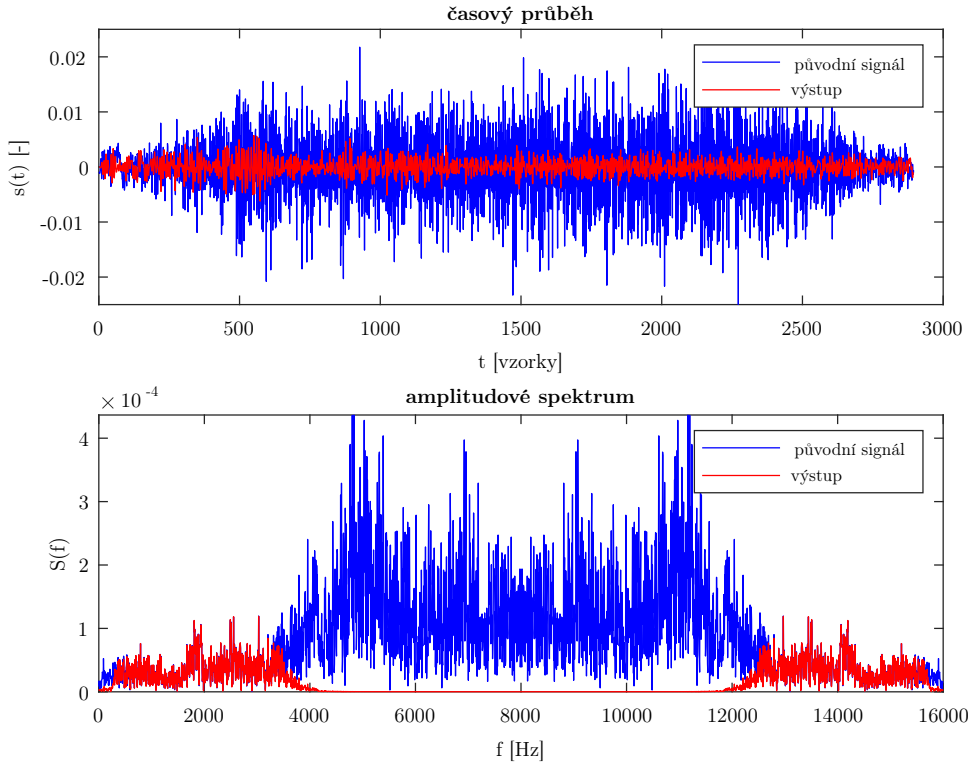
[8] Každý ze seznamů však odkazuje na jiný typ souborů – anglický na audio nahrávky, český na fonetické přepisy. To z principu představuje nutné drobné formální rozdíly v kódu při jejich nahrávání a dalším volání ve funkcích.

Tabulka 3.4: Ukázka použitého přepisu TIMIT nahrávek na úrovni hlásek

.phn
0 2090 h#
2090 2783 q
2783 4010 eh
4010 4929 l
4929 5560 d
5560 6851 axr
6851 7477 l
7477 9080 iy
Přepis slova „elderly“

Pokud by však měl být rozpoznávací systém použit na nahrávkách, které tento předpoklad zcela nesplňují, nedosahoval by správných výsledků.

Pro taková data by bylo potřeba akusticky přizpůsobit trénovací a testovací set dat, resp. sjednotit akustické podmínky pro všechny zpracovávané signály. Nejjednodušší a nejuniverzálnější možností je oba sety transformovat do telefonního pásma (taková změna, aplikovaná na nahrávku TIMIT, je demonstrována na obr. 3.2) a sjednotit jejich vzorkovací frekvenci na 8 kHz. V připravené realizaci nebyla taková úprava vzhledem k charakteru dat nutná, systém je na tuto možnost ale připraven a nabízí využití funkce *phoneband_filtering*. Ta filtruje a decimuje signál, ale také předá dále všechny upravené výpočetní parametry, kterých je poté v implementaci využíváno.

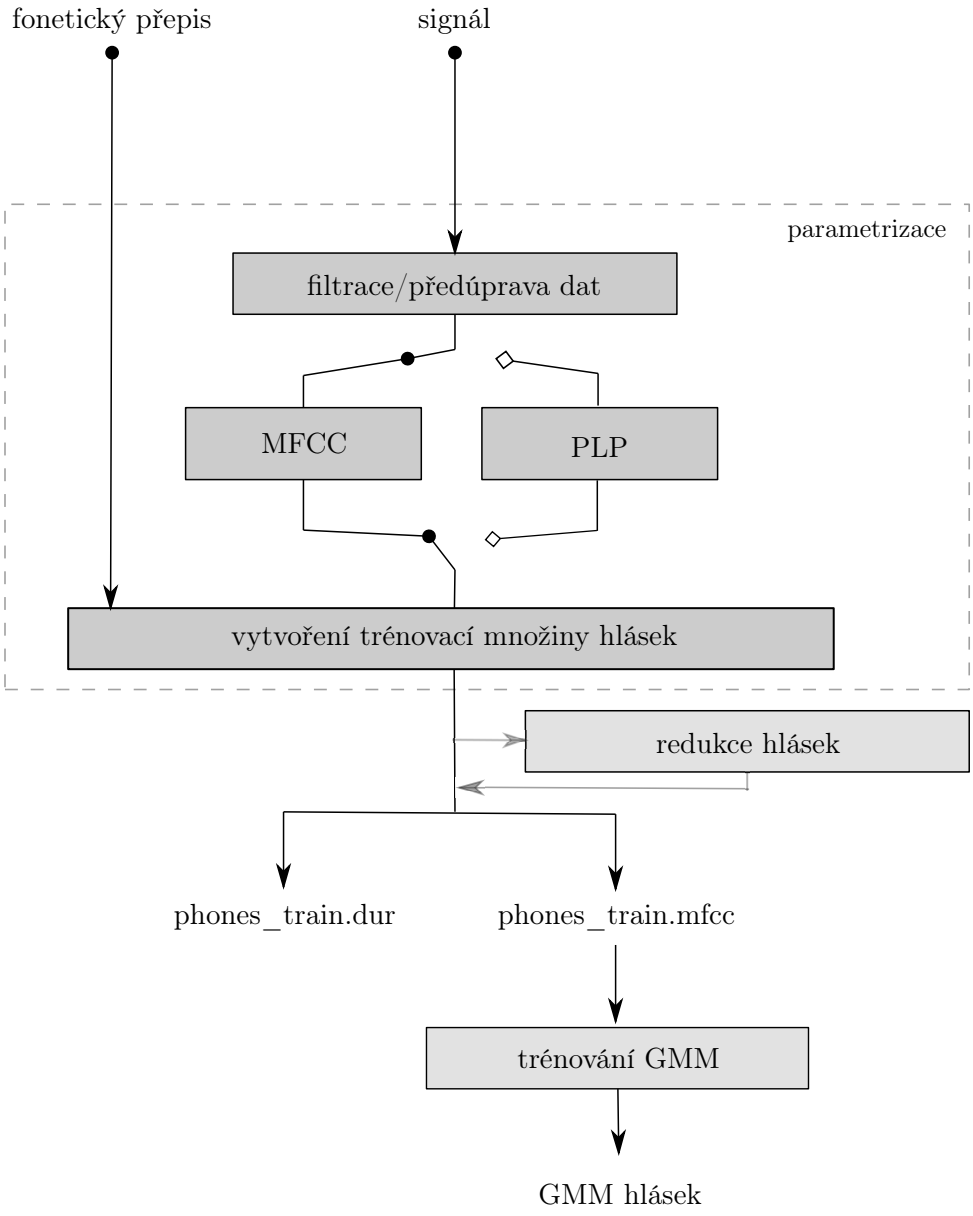


Obrázek 3.2: Změny v signálu po filtraci do telefonního pásma

K filtraci signálu byla zvolena realizace IIR filtrem, a to kvůli nižší výpočetní složitosti, dosahování nižších zpoždění mezi vstupy a výstupy, ale také díky strmějším přechodům mezi propustným a nepropustným pásmem. Pro udržení stability byl zvolen IIR filtr Butterworth

řádu 12. Propustné pásmo bylo definováno mezi 300 - 3400 Hz. Dvakrát nižší vzorkovací frekvence signálů (tedy 8 kHz) byla získána decimací (MATLAB příkazem *decimate*) v poměru 1:2.

Jak vychází z popisu, tato funkce slouží k počátečním úpravám trénovacích dat, a tedy k sestavení správně fungujícího systému.[9] Je možné ji využít také k úpravě testovacích dat, ale s nutnou úpravou parametrů (vzhledem k originální vzorkovací frekvenci testovaných záznamů).



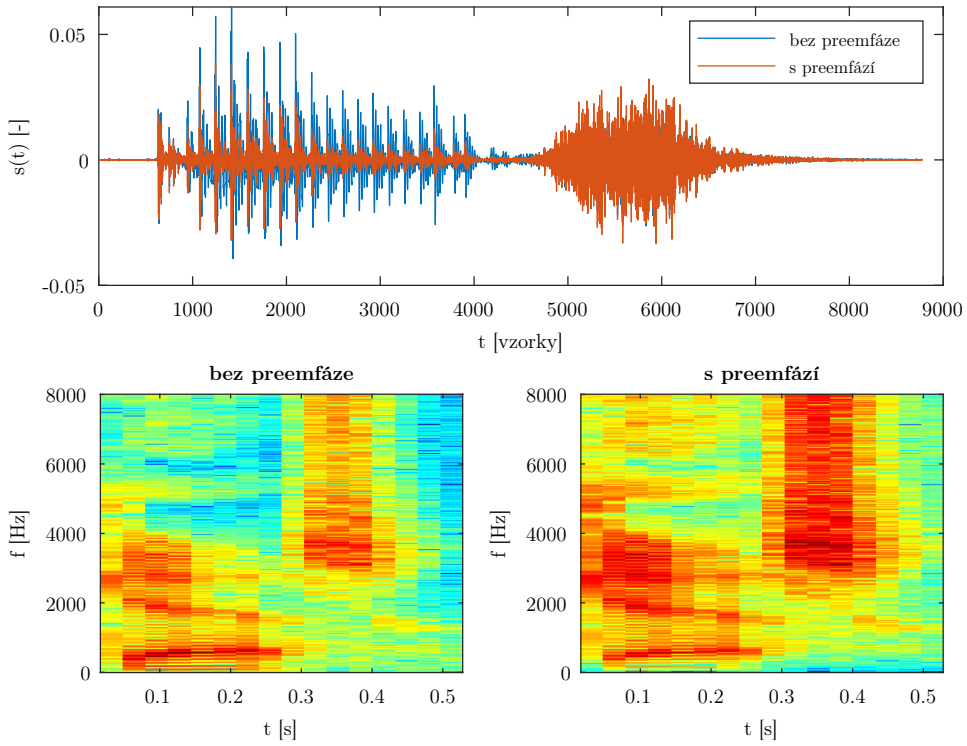
Obrázek 3.3: Schéma parametrizace a vytvoření GMM modelu

3.4.3 Parametrizace

Pokud jsou potřebná data připravena a máme k dispozici jejich fonetické přepisy, první velkou fází systému je počítání parametrických vektorů pro každou hlásku, která se v trénovací množině vyskytuje. V realizované implementaci se pracuje s výpočtem parametrů MFCC (dle schématu na obr. 2.4) a je tak spuštěna navržená funkce *mfcc_computing_phones*.

[9] Pouze v případě, že chce uživatel testovat na jiných datech, než pro které je systém připraven. Jinak je tento krok zcela přeskočen a systém funguje jak má.

V ní dochází k načtení každé trénovací promluvy i její fonetické transkripce. Průchodem volným prostorem je akustický signál reprezentující řeč utlumován a složky s vyšší frekvencí jsou postupně o 20 dB na dekádu zkreslovány. Jelikož některé složky řeči důležité pro analýzu jsou právě v oblastech vyšších kmitočtů, je třeba takový útlum vyrovnat filtrací, resp. aplikací preemfáze na nahraný signál.[10] Na obr. 3.4 je možné vidět, jakým způsobem je aplikací preemfáze signál ovlivněn - vyšší kmitočty jsou jasně zvýrazněny.



Obrázek 3.4: Vliv preemfáze na řečový signál, časové průběhy a spektrogramy

Dále pak dochází k segmentaci signálu a aplikaci Hammingova váhovacího okna (pro potlačení prosakování ve spektru). Rychlou Fourierovou transformací je získáno amplitudové spektrum, ze kterého umocněním absolutní hodnoty získáme výkonové spektrum jednotlivých rámců signálu.

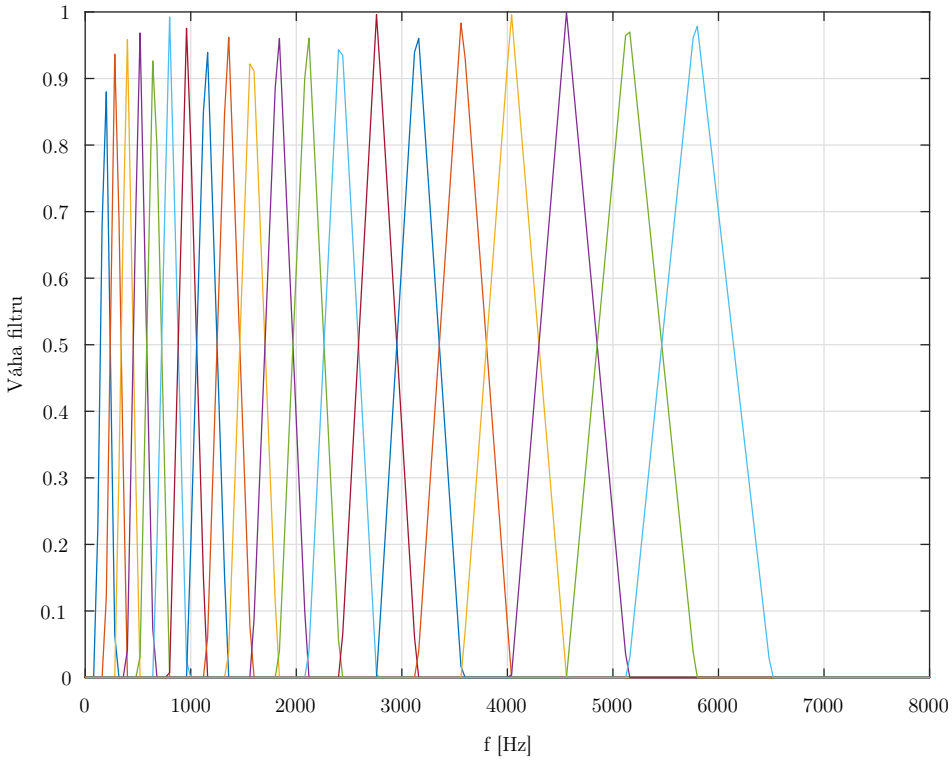
V dalším kroku prochází vypočítané frekvenční spektrum pásmovou filtrací, reprezentovanou bankou trojúhelníkových filtrů (obr. 3.5). Ty jsou lineárně rozmístěny na tzv. melovské frekvenční škále (již zmíněno v části 2.4.1) s M = 20 filtry. Počet těchto filtrů závisí na počtu kritických pásem, které jsou ve zkoumaném rozsahu obsaženy (nejčastěji se volí v závislosti na hodnotě použité vzorkovací frekvence) [17]. Pásmová filtrace sama o sobě je v závěru násobením bodů výkonového spektra s odpovídajícím ziskem filtru na dané frekvenci (a tyto hodnoty se pro každý filtr sčítají).

Na výstupy jednotlivých filtrů (výkony, resp. energie v jednotlivých pásmech) se aplikuje přirozený logaritmus a převod do kepstrální oblasti je dokončen pomocí DCT (díky symetrii výkonového spektra lze nahradit zpětnou diskrétní Fourierovu transformaci reálnou transformací kosinovou).

Ze zmíněných kroků, k jejichž výpočtu dochází v dostupné funkci *vmfcc*[11], vychází 13 me-

[10] Oproti tomu při parametrizaci PLP koeficienty se preemfáze nepočítá, tento útlum je zohledněn křivkami stejné hlasitosti.

[11] S použitými parametry odpovídajícími inicializaci v počátku skriptu a počtu pásem banky filtrů M = 20.



Obrázek 3.5: Mel-banka filtrů použitá v implementaci na každý segment signálu

lovských kepstrálních koeficientů (resp. 12+1, nulový koeficient c[0] odpovídá logaritmu energie signálu a nemusí s ním být v rozpoznávání nutně počítáno[12]).

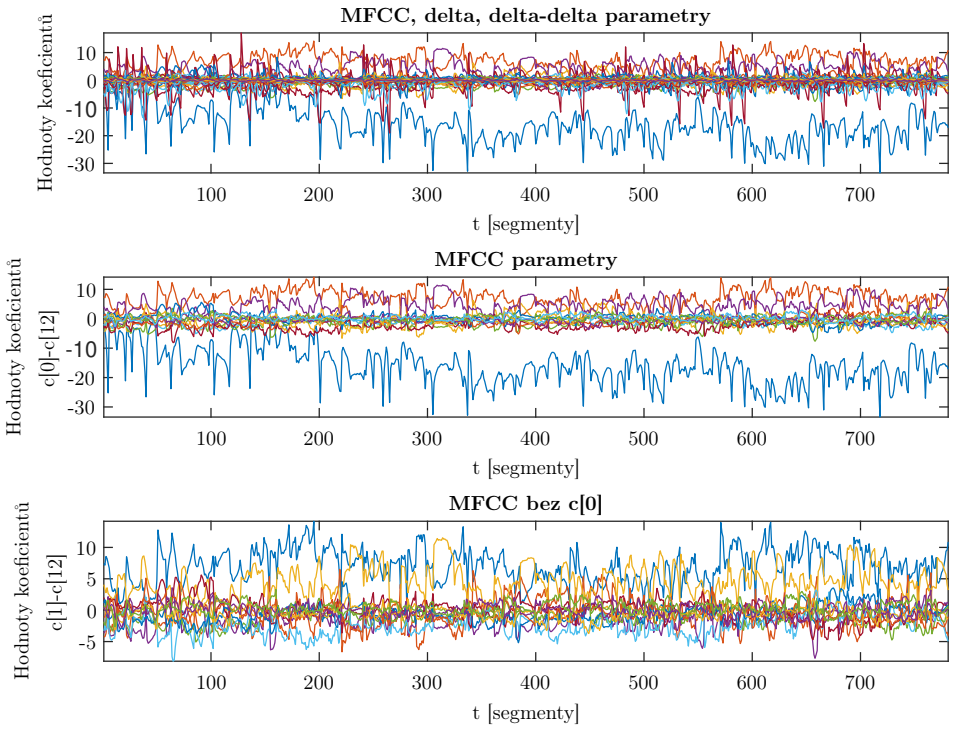
Pro bohatší informační obsah, který by rozpoznávacímu systému přinesl větší přesnost, se příznakový vektor MFCC obohacuje o delta a delta-delta parametry (tzv. dynamické a akcelerační koeficienty, vzájemně nekorelované). Jejich účelem je k vektoru statických příznaků (MFCC) připojit další příznaky, které obsahují dynamickou informaci o signálu (časové změny pro každý z analyzovaných rámců). Vektor delta příznaků je derivací vektorů příznaků pro každý ze segmentů, obdobně delta-delta příznaky jsou druhou derivací původních statických MFCC příznaků.[13] Výsledný parametrický vektor tak čítá celkem 39 příznaků.

Díky načtené fonetické transkripci je ze spočítaného příznakového vektoru možné vyjmout části, které přísluší jednotlivým hláskám (dle počátečních a koncových vzorků). Opakovaným průchodem všech testovacích sekvencí se tak kumulují parametry pro všechny hlásky a proces parametrizace končí až po zpracování všech trénovacích souborů, jež jsou k dispozici. Výstupem z funkce *mfcc_computing_phones*, jež počítá příznakové vektory jednotlivých hlásek, je pak struktura *phones_train*, kde jsou všechny napočítané parametry pro jednotlivé hlásky uloženy.

Parametrické vektory (které je možné v implementaci po napočítání zobrazit pro libovolné hlásky, viz obr. 3.6) nejsou však jedinou informací, která se z popsaného zpracovávání uchovává. Délka trvání každé hlásky v jednotlivých promluvách (vyjádřená v počtu krátkodobých segmentů) se ukládá pro pozdější inicializaci HMM modelů (dále v 3.5.1).

[12] S jeho využitím je ale dosaženo lepších výsledků.

[13] Využití bohatších informací o kontextu rámců není příliš neobvyklé. Představuje zvýšení výkonu rozpoznávače přibližně o 20 % [38], [39].



Obrázek 3.6: Průběh kepstrálních koeficientů pro hlásku IH

Redukce hlásek

Pro anglický jazyk je k trénování použitý řečový korpus velice obsáhlý a velmi podrobně přepsaný na fonetické úrovni (zohledňující také různé alofóny hlásek). V důsledku je ve fonetickém přepisu větší množství hlásek. Pro zjednodušené trénování akustických modelů je využití takto bohaté množiny nevhodné. Pro eliminaci příliš podobných a specifických úseků řeči, které jsou v transkripci popsány jako samostatná hláska a které se v korpusu neobjevují v dostatečném množství pro trénování, je nutné fonetickou sadu redukovat.

Tento krok však v pracích věnujících se práci s TIMIT databází a rozpoznáváním řeči není neobvyklý, byl představen několikrát [40], [41]. Byť tyto práce specifikují, jakým způsobem v nich došlo k redukci hlásek, konečné úpravy byly jimi jen inspirovány, v realizované implementaci dochází k úpravám viz tab. 3.5.

V praktické implementaci není tato redukce nijak zvlášť složitá. Napočítané příznaky pro hlásky, jež se odstraňují, jsou spojeny s těmi, které jsou jim foneticky nejblíže, a pak z proměnných zcela odstraněny.

Tento krok pro český korpus dat není využit, není proto považován za pevný a nutný blok realizace rozpoznávače. V práci je však pro redukci alofónů anglického jazyka na místě.

Realizace demonstračního rozpoznávače tedy pracuje s celkem 39 hláskami pro anglický jazyk a 44 hláskami pro jazyk český – všechny jsou přehledně uvedeny v tab. 3.6.[14]

3.4.4 Modely GMM

Když jsou parametrické vektory pro všechny hlásky napočítány, je třeba připravit základní GMM modely jednotlivých hlásek, které budou reprezentovat hustoty pravděpodobnosti příslušné emitujícím stavům HMM modelu hlásky (tj. akustickému modelu). GMM model věrohodně a spo-

[14] V prostředí MATLAB je pak seznam používaných hlásek uložen v proměnné *list_of_all_phonems*.

Tabulka 3.5: Mapování na menší počet tříd hlásek, hlásky z levého sloupce jsou sloučeny s hláskou v pravém sloupci, hláska 'q' byla zcela odstraněna

Hláska z TIMIT přepisu	Sloučeno s hláskou
ao	aa
ax, ax_h	ah
axr	er
hv	hh
ix	ih
el	l
em	m
en, nx	n
eng	ng
zh	sh
ux	uw
pcl	p
tcl	t
kcl	k
bcl	b
gcl	g
pau, epi, h#	sil

Tabulka 3.6: Použité fonetické sady

Hlásky použité pro korpus v anglickém jazyce	Hlásky použité pro korpus v českém jazyce
sil, eh, n, ih, f, sh, l, d, ay, k, aa, b, iy, p, ow, s, m, ey, dh, t, aw, w, ah, er, g, z, ae, v, uh, r, dx, y, oy, th, ch, hh, ng, uw, jh	sil, v, i, e, s, ii, ch, m, ss, l, ng, k, aa, dd, t, zz, d, c, a, z, r, o, f, u, n, h, p, cc, dz, j, tt, ou, rrr, rr, ee, dzz, b, mv, nn, g, eu, uu, au, oo

lehlivě reprezentuje vlastnosti celé množiny podobných dat (hlásek), modeluje je, resp. dává informace o popisu jednotlivých fonémů (ze sekvence příznaků).[15] Díky bohatému korpusu trénovacích dat a jejich variabilitě pak bude i model dostatečně univerzální (zohledňující mužské, resp. ženské hlasy, nářečí, hlasová zbarvení). Pro modelování takového vzoru je používána právě diskutovaná Gaussova distribuce, která je nejběžnější (a nejlépe analyzovatelnou) spojitou distribucí a je již četně využívána v systémech rozpoznávání řeči [42], [43], [44].[16]

Pro výpočet parametrů GMM modelu je v implementaci použito příkazu *fitgmdist*. Tvar distribuce se ovlivňuje parametry vektorů střední hodnoty (*mean*, ovlivňující umístění) a odchylky (*variance*, ovlivňující rozptyl). Spolu s informací o vahách směsí a podobě kovarianční matice udávají celkovou podobu GMM modelu. Odhad těchto parametrů spadá obecně do třídy problémů chybějících dat (missing data problem). Pokud jsou známé, je možné je v MATLAB příkazu definovat přímo, avšak nejčastěji se odhadují. Jelikož máme k dispozici dostatečný počet parametrů řeči, jež se rozložením snažíme vyjádřit, jako vstup do funkce se využívá všech příznakových vektorů. Z nich jsou pak tyto parametry dopočítány automatickým odhadem funkce *fitgmdist* tak, aby distribuci ideálně modelovaly. Tvar modelu je však možné ovlivnit i jinak, a to zvláště počtem komponent (směsí), se kterými má počítat. Počet komponent představuje počet vrcholů modelu (mající vlastní *mean* i *variance*). V implementaci byly vytvořeny GMM se čtyřmi komponenty, a to zejména pro větší detail ovlivňující úspěšnost rozpoznávání.[17] Zároveň je počítáno s diagonální kovarianční maticí, s počtem použitých Gaussových funkcí s ní pak

[15] Pro trénování vyžadují akustické modely velké množství anotovaných trénovacích dat. Má-li být rozpoznávací systém nezávislý na řečníkovi, je potřeba zpracovat stovky hodin řečových dat namluvených různými řečníky. U systému pouze pro jednoho mluvčího je taková trénovací sada namluvena pouze jedním řečníkem a délka jejího trvání může být klidně i desetkrát menší.

[16] Další možností akustického modelování je využít techniky vektorové kvantizace.

[17] Vliv počtu směsí na výsledný systém byl testován, tab. 4.1, 4.2, 4.3.

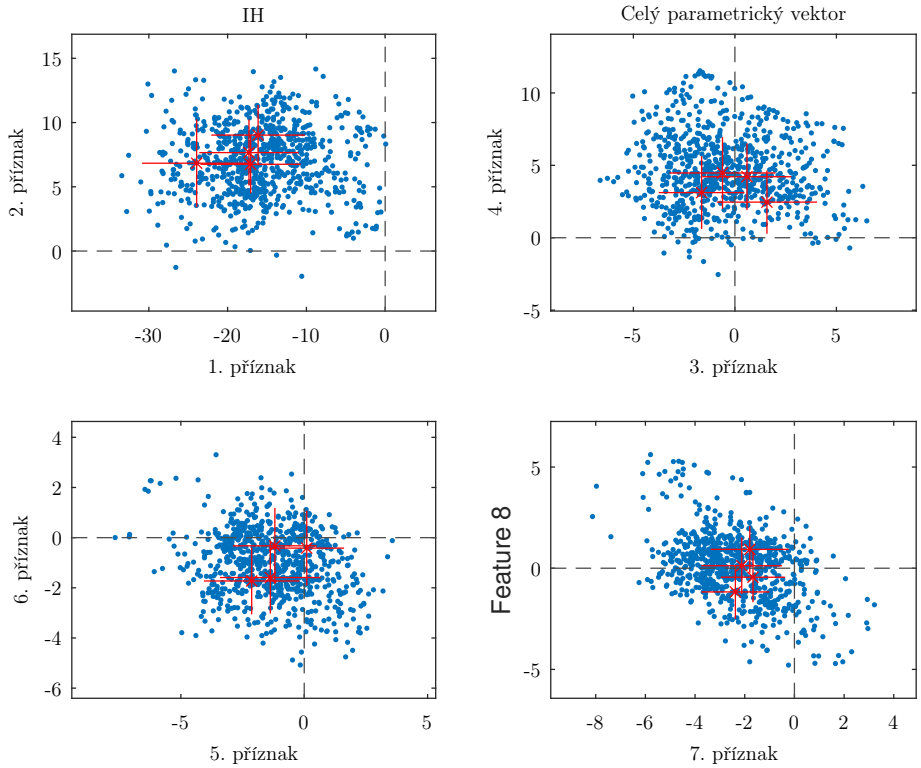
roste úspěšnost rozpoznávání. Proces tvorby GMM modelů v MATLAB implementaci je možné zjednodušeně vyjádřit následovně:

```
mixnum = 4;
% užité nastavení počtu směsí

mfcc_ae = [];
for i = 1:length(phones_train.ae)
mfcc_ae = [mfcc_ae; phones_train(i).ae];
end
% řetězení jednotlivých MFCC struktur
% vstup do funkce fitgmdist musí být proměnná typu double (single)

gmm.ae = fitgmdist(mfcc_ae, mixnum, 'CovType', 'Diagonal');
% tvorba modelu GMM pro hlásku AE s použitím napočítaných MFCC parametrů
```

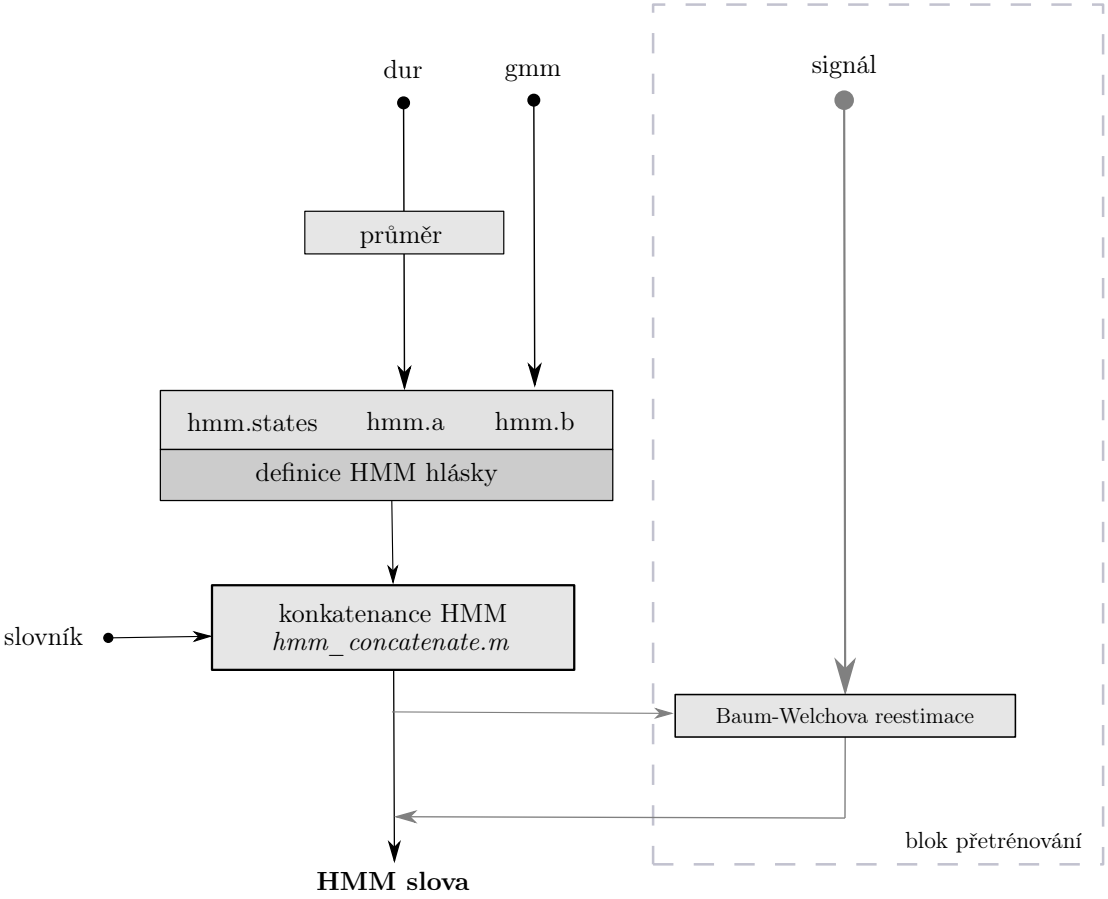
Pro vizualizaci mezivýsledku (resp. vytvořeného GMM) implementace napočítaný akustický model zobrazuje přes prvních 100 kepstrálních příznaků. Jelikož kepstra podobných segmentů tvoří shluky, je možné model překrytý přes vzorky zobrazovat následovně (obr. 3.7):



Obrázek 3.7: GMM model pro hlásku IH v červeném zobrazení přes kepstrální příznaky

3.5 Vytvoření HMM modelů slov

Jakmile dojde k vytvoření GMM modelů hlásek, je první důležitý krok vytvoření rozpoznávacího systému hotov. S napočítanými parametry (resp. modely) se dále pracuje, neboť hlavním vstupem do konečného procesu rozpoznávání jsou HMM modely slov ze slovníku, z nějž k rozpoznávání dochází.



Obrázek 3.8: Schéma vytváření HMM hlásek a slov

3.5.1 HMM modely monofónů

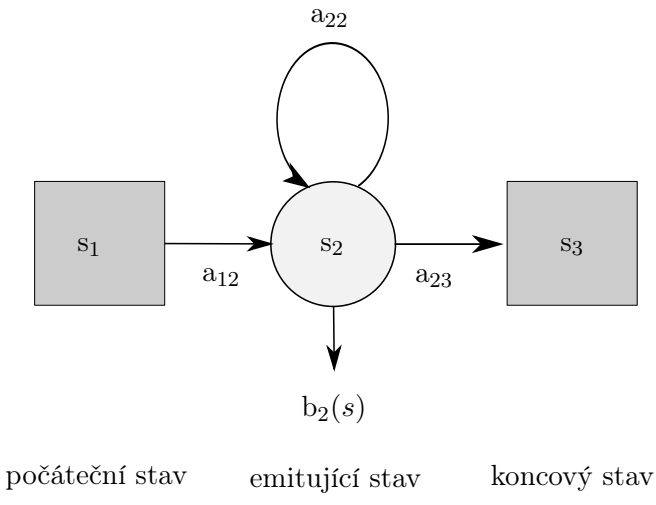
Jsou-li vytvořeny GMM modely hlásek, je prvním krokem tvorby skrytých Markovových modelů pro účely kýžené implementace vytvoření HMM modelu každé z hlásek. Byť je pro rozpoznávání často pracováno s trifóny (viz sekce 2.2), vytvoření modelů pro takto kontextově závislé jednotky je velice složité a vyžaduje opravdu rozsáhlé a důsledně popsané databáze, jež tento kontext zohledňují. K tomu v této práci tedy nedošlo. Modely hlásek, se kterými je v systému pracováno, jsou monofónové a navíc jen s jedním emitujícím stavem (tj. celkem třístavové s ještě dvěma neemitujícími stavy), viz obr. 3.9.

Všechny HMM v implementaci vytvořené (nehledě na to, zda se jedná o model hlásky nebo dále slova) zachovávají stejnou strukturu (obr. 3.10), jsou popsány:

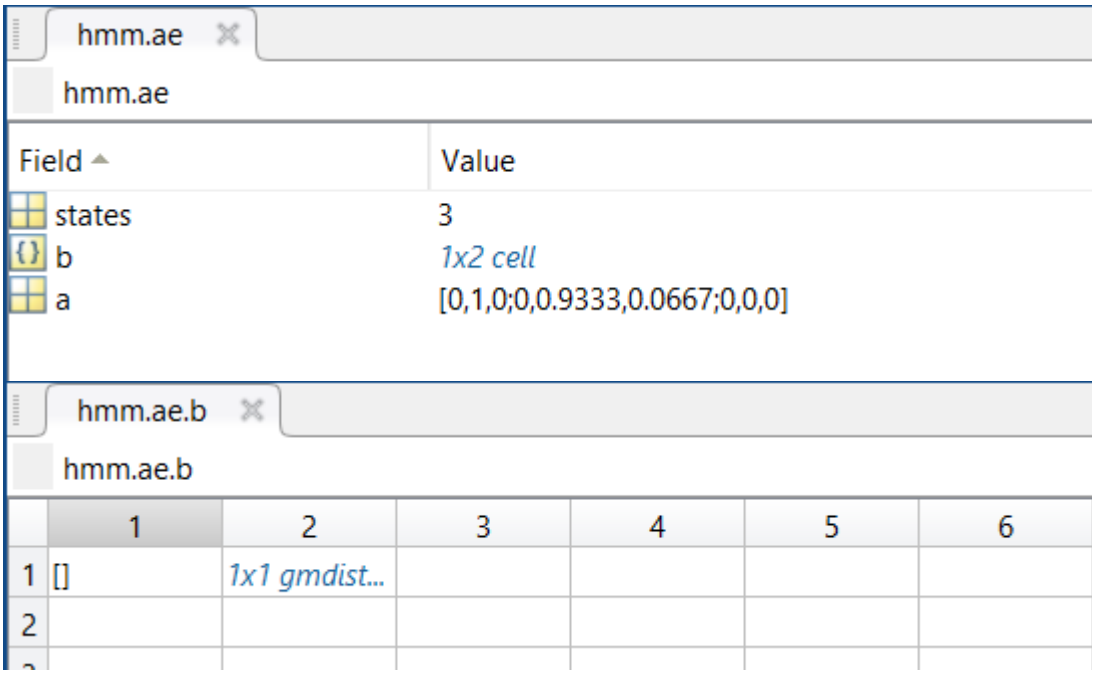
- počtem stavů,
- maticí pravděpodobnosti přechodů mezi stavy **A**,
- sadou hustot pravděpodobnosti pro jednotlivé stavy **B**.

Hustoty výstupních pravděpodobností **B** pro emitující stav je určen vypočítaným GMM modelem, pro neemitující stavy je **B** prázdné (těmto stavům není přiřazen žádný příznakový vektor). Matice přechodů **A**, vytvořená pro každou hlásku, není nijak složitá (pro 3 stavy má rozměry 3x3) a vychází z následujících předpokladů:

- že k přechodu z prvního neemitujícího do emitujícího stavu dojde vždy ($a_{1,2} = 1$),
- že pravděpodobnost setrvání ve stavu (hlásce) či přechodu do dalšího stavu (konec hlásky či další zřetězená hláska) vychází z průměrné délky trvání hlásky v trénovací sadě promluv,



Obrázek 3.9: Použitý třístavový levo-pravý HMM model hlásky



Obrázek 3.10: Struktura modelu HMM hlásky AE v MATLABu

- a že poslední stav (neemitující) ukončuje model a může být k němu zřetězen další z modelů (resp. první neemitující stav dalšího modelu).

Druhý bod se tedy odkazuje na proměnnou, do níž se v implementaci ukládá délka trvání jednotlivých hlásek ve všech promluvách (bylo zmíněno v části 3.4.3). Množství takto uložených informací o délce trvání se odvíjí od četnosti výskytu hlásky v trénovacích datech. V našem případě však předpokládáme, že bylo využito dostatečně objemného řečového korpusu, a tedy že je tato informace dostatečně četná pro spolehlivou generalizaci. Je vypočítána průměrná délka trvání hlásky v promluvách a pro výpočet pravděpodobnosti přechodu $a_{2,2}$ je využito jednoduchého výpočtu:

$$a_{2,2} = \frac{phones\_train.X.avg - 1}{phones\_train.X.avg} \tag{3.1}$$

Pravděpodobnost $a_{2,3}$ - pravděpodobnost přechodu z emitujícího stavu do neemitujícího (konec hlásky) je pak doplňkem do 1 (součet pravděpodobností v řádku musí být vždy roven jedné).

Celá matice **A** pro skrytý Markovův model jedné hlásky (pro příklad hlásky ae s průměrnou délkou trvání 15 segmentů) vypadá pro představu následovně:

$$hmm.ae.a = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0.9333 & 0.0667 \\ 0 & 0 & 0 \end{bmatrix}$$

V MATLAB implementaci je pak HMM model principiálně definován dle následujícího zjednodušeného vzoru:

```
hmm.ae.states = 3;
% počet stavů

hmm.ae.b = {[] gmm.ae}
% poslední neemitující stav se nevyužívá

hmm.ae.a = zeros(hmm.ae.states, hmm.ae.states);

hmm.ae.a(1,2) = 1;
% první neemitující stav vždy přejde do prvního emitujícího

for i = 2:hmm.ae.states-1
  hmm.ae.a(i,i) = (phones_train.ae.(1).dur_avg - 1) / phones_train.ae.(1).dur_avg;
  % výpočet z průměrné doby trvání

  hmm.ae.a(i,i+1) = 1 - hmm.ae.a(i,i);
  % doplněk do 1
end
```

3.5.2 Inicializace HMM modelů slov

Dále je pro každé slovo ze slovníku nutné vytvořit unikátní HMM. Jednotlivé modely jsou vytvořeny zřetězením všech hlásek, které jsou určeny fonetickou transkripcí slova (dle slovníku, sekce 3.5.3). Všechny vytvořené modely jsou levo-pravé a dopředné, což vychází z charakteru řeči jako signálu plynoucího v čase.

HMM slov mají stejnou strukturu jako HMM hlásek, musí být tedy dány informací o počtu stavů, maticí pravděpodobnosti přechodů **A** a hustotami výstupních pravděpodobností **B**. Jelikož jsou všechny tyto informace již obsaženy v modelech hlásek, ze kterých má být model slova tvořen, definice těchto dílčích informací nepředstavuje složitou úlohu – jsou přebírány z již vytvořených akustických modelů subslovních elementů.

Pro řetězení modelů, resp. hlásek, bylo využito funkce *hmm_ concatenate*, která byla postupně laděna od spojování jednostavových modelů hlásek až po složitější - je tedy vhodná pro spojování modelů hlásek do modelů jednotlivých slov. Funkce je tvořena pro libovolný počet vstupních parametrů (*varargin*), a tedy je schopna najednou řetězit libovolné množství připravených HMM s obecnou délkou, resp. strukturou.[18]

Metoda počítá s tím, že všechny modely začínají a končí neemitujícím stavem. Informace o celkovém počtu stavů vytvářené entity je snadno spočitatelná. K „připojovanému“ modelu se přičte počet stavů „připojeného“ modelu, ponížený o 2 (tedy přičítáme pouze počet emitujících stavů připojeného modelu). Podobně pak také emitující pravděpodobnosti **B** - ke stávajícím se do dalších buněk přiřadí emitující pravděpodobnosti připojeného modelu. Jelikož konečný neemitující stav je nepotřebný a není v HMM implementaci zahrnut, stačí připojovat **B** vždy od druhého indexu. První je prázdný a je již ponechán z „kořenového“ modelu. Matice přechodů **A** se vždy jen rozšíří o pravděpodobnosti patřící k emitujícím stavům připojeného modelu. Z obecné definice matice přechodů vyplývá, že je třeba jednotlivé pravděpodobnosti přenásobovat. Jelikož

[18] Funkce tedy umožňuje řetězit mimo hlásek také např. modely slov do modelů sousloví či vět atd.

však nepočítáme s modely s přeskoky, je možné k řetězení přistoupit zmíněným rozšířením - v podstatě vždy násobíme jedničkou.

Počáteční hustoty výstupních pravděpodobností **B** byly tedy určeny jako GMM modely hlásek (s diagonální kovarianční maticí a počtem směsí rovné 4), prvky v matici pravděpodobnosti přechodů **A** pak byly napočítány z průměrné doby trvání jednotlivých hlásek v řečovém korpusu (s hodnotou normalizovanou vůči 1, jak již bylo popsáno v sekci 3.5.1). Z podstaty HMM je matice **A** nenulová na diagonále, pro inicializaci modelů nebylo počítáno s možností přeskakování jednotlivých stavů.

Výstupem řetězení za pomoci zmíněné funkce je tedy celistvý model s vlastní **A** a **B**, informacemi o počtu stavů. Pro příklad lze uvést model slova „gas“, který se skládá z modelů hlásek g, ae, s a počátečního a koncového sil s maticemi přechodů

$$hmm\_sil.a = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0.9412 & 0.0588 \\ 0 & 0 & 0 \end{bmatrix}, hmm\_g.a = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0.7500 & 0.2500 \\ 0 & 0 & 0 \end{bmatrix},$$

$$hmm\_ae.a = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0.9333 & 0.0667 \\ 0 & 0 & 0 \end{bmatrix}, hmm\_s.a = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0.9091 & 0.909 \\ 0 & 0 & 0 \end{bmatrix}.$$

Matice **A** pro slovo složené z uvedených dílčích hlásek tak vypadá následovně:

$$hmm\_gas.a = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.9412 & 0.0588 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.7500 & 0.2500 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.9333 & 0.0667 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.9091 & 0.0909 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.9412 & 0.0588 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Má celkem 7 stavů (2 neemitující a 5 emitujících) a **B** je tak složen z 5 GMM (první a poslední neemitující stav žádný model nemají, ty slouží pouze pro řetezění) - v HMM slova tak **B** pro poslední stav ani není. Je tedy možné zjednodušeně vyjádřit jako:

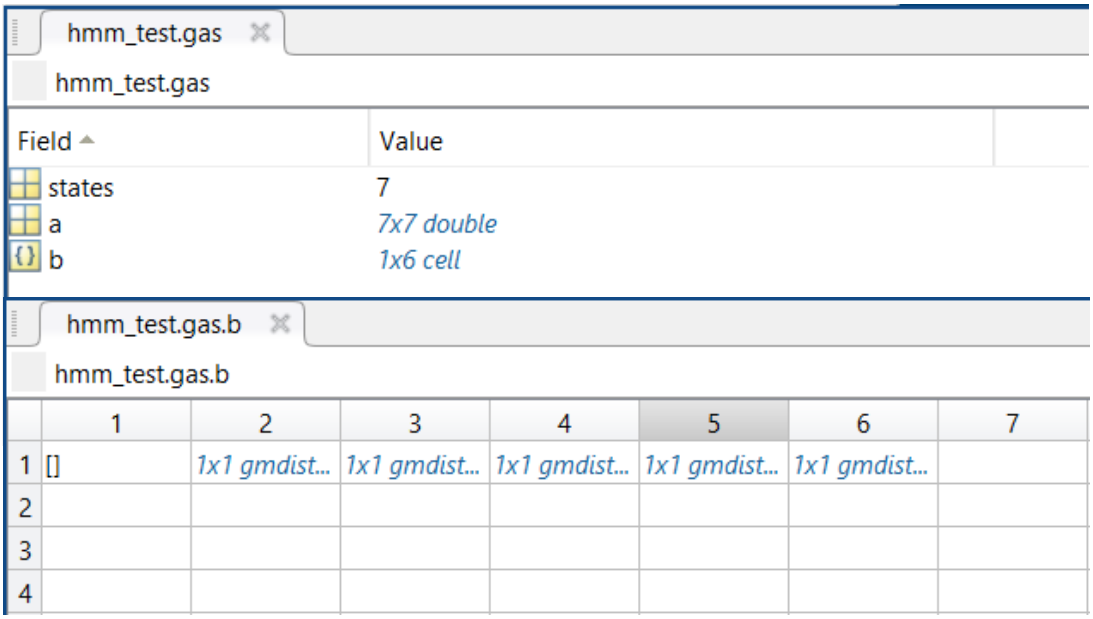
```
hmm_gas.b = {[] gmm.sil gmm.g gmm.ae gmm.s gmm.sil};
% poslední neemitující stav se nevyužívá, první je prázdný
```

3.5.3 Slovníky

Pro rozpoznávání konkrétních slov musíme znát slovník, z nějž budou slova vybírána a podle nějž jsou tvořeny modely slov. Slovník musí dále obsahovat informaci o výslovnosti. Realizovaný demonstrační rozpoznávač je schopen pracovat se slovníkem definovaným v *.txt* souboru s velmi jednoduchou strukturou naznačenou následovně:[19]

```
one sil w ah n sil
two sil t uw sil
three sil th r iy sil
four sil f aa r sil
five sil f ay v sil
six sil s ih k s sil
seven sil s eh v ah n sil
eight sil ey t sil
nine sil n ay n sil
zero sil z ih r ow sil
```

[19] Jedná se zároveň o jeden ze slovníků použitých v implementaci.



Obrázek 3.11: Struktura modelu HMM slova „gas“ v MATLABu

Tedy uvést slovo a za ním fonetický přepis, všechny hlásky v něm, pokaždé odděleny mezerou. Jedná se o nejrychlejší, nejpřehlednější a pro uživatele také nejsnazší způsob, jak vytvořit nový slovník či upravit stávající, jeho zpracování v prostředí MATLAB zároveň nevyžaduje složitou algoritmizaci. Je to rovněž standardně používaný formát i v jiných aplikacích.

K fonetickému přepisu je na začátek a konec slova přidána hláska sil. Jedná se o hlásku představující ticho (mlčení, ukončené slovo, pauza) - ta sice dále v modelech není nutná (rozpoznávač funguje i bez ní), ale jelikož není možné dopředu říci, jakou řečovou nahrávku bude případný uživatel v systému testovat (a jak dobře bude na začátcích oříznuta tak, aby opravdu obsahovala jen hlásky[20], ale zároveň aby nebyl oříznutý jejich případný „náběh“ či „útlum“), je dobré sil zahrnout. Právě proto, že toto není pro funkci rozpoznávače nutné (a záleží zcela na uživateli, jakým způsobem a s jakými daty se rozhodne pracovat), nebylo přidání „hluchých“ počátků a konců slov zahrnuto pevně v MATLAB kódu pro vytváření modelů.[21]

Pro účely práce byly vytvořeny slovníky tři (dva pro anglický jazyk a jeden pro český). První, anglický, (*timit_ words.txt*) obsahuje několik vzájemně nesouvisejících slov a sousloví, o kterých z dokumentace řečového korpusu víme, že jsou obsaženy v testovací množině. Zároveň jsou však vybrána tak, aby si některé z nich byly foneticky podobné, některé zcela unikátní, a také aby ve slovníku nebyly jen jednotlivá slova, ale také sousloví.

Druhý slovník (*digits.txt*) obsahuje číslovky 0-9 v anglickém jazyce a je tak ukázkou možného praktického využití rozpoznávačů izolovaných slov například pro účely diktování telefonních čísel, čísla účtu a jiných praktických aplikací, jež vycházejí z úvodu této práce.

Pro vytvoření těchto slovníků bylo použito doprovodných souborů k databázi TIMIT (zejména slovníku všech řečených slov s jejich kvazi-fonetickým přepisem). Předpisy ve slovníku TIMIT (*timitdic.txt*) spolu s on-line nástrojem CMUdict[22] sloužily jako podklad pro vytvoření slovníku ve zpracované implementaci – jednotlivá slova ve slovnících jsou sepsána dle transkripcí uvedených v těchto zdrojích.

Slovník vytvořený pro český jazyk se skládá z číslovek 0-9. Mimo důvody pro takový slovník

[20] Zejména při rozpoznávání na nahrávkách snímaných v reálném čase je tento ořez prakticky nemožný.

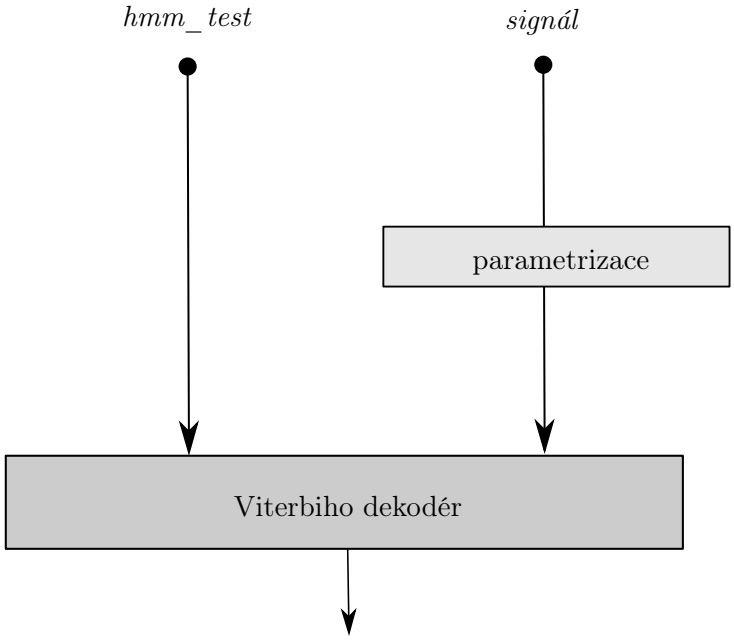
[21] Pokud jsou však ve slovníku slovní spojení o větším počtu slov, je třeba je hláskou *sil* oddělovat. Stejně tak, pokud z časového průběhu signálu vidíme, že je v něm mezi hláskami odmlka, je možné *sil* do fonetického přepisu zahrnout a zlepšit tak přesnost modelu - a tedy i úspěšnost rozpoznávání.

[22] **The CMU Pronouncing Dictionary**, version 0.7b [online], Carnegie Mellon University

spojené s využitím v praxi byl zvolen zejména kvůli snadnému procesu testování (použitá databáze pro české promluvy již obsahuje nahrávky izolovaných číslovek). Fonetické přepisy promluv databáze SPEECON byly zdrojem pro sestavení předpisů slov slovníku.

3.6 Vyhledávací funkce a dekódování

Pro samotný proces rozpoznávání je třeba jen dvou vstupů – řečového signálu, který chceme rozpoznávat, a HMM slov, jež jsou obsažena v použitém slovníku. Ve zkratce je možné říct, že při rozpoznávání dochází k mapování příznaků analyzovaného signálu s připraveným modelem. Model, jež na výstupu tohoto mapovacího procesu získá nejvyšší skóre, je pak považován za rozpoznané slovo.



Obrázek 3.12: Schéma procesu dekódování

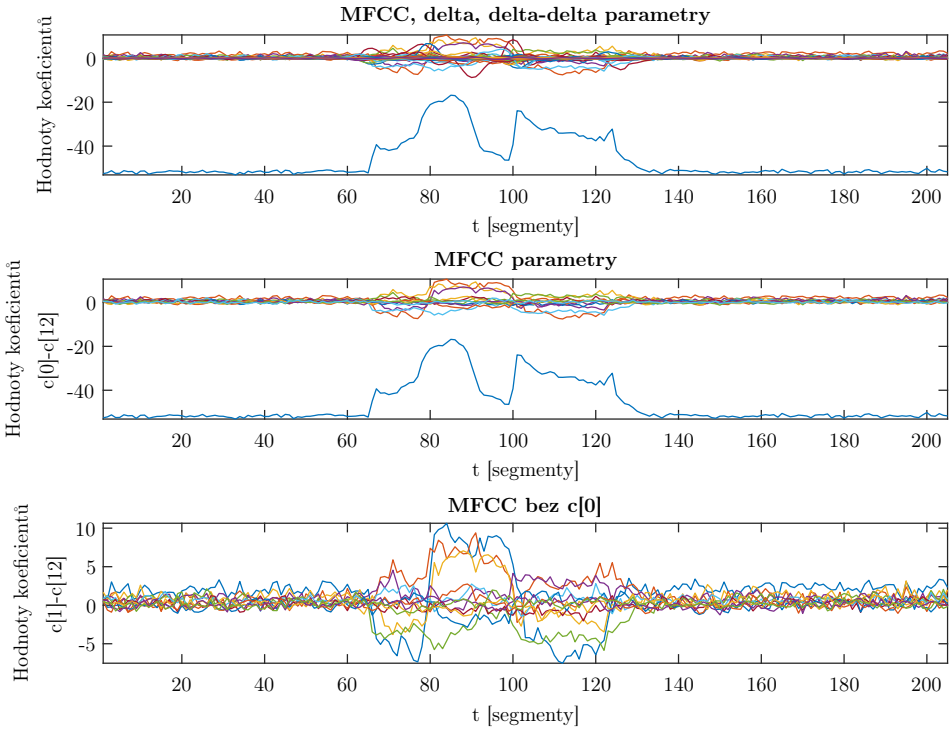
Z toho tedy vyplývá, že analyzovaný signál musí do dekodéru vstoupit ve formátu stejných spektrálních příznaků, jako byly použity pro trénování modelů. V našem případě je tedy potřeba vypočítat MFCC, delta, delta-delta příznaky a s nimi pak dále pracovat (náhled na napočítané příznakové vektory je možné v implementaci zobrazit, viz obr. 3.13).

Proces dekódování je výpočetně náročný. Dekódovacích algoritmů existuje několik (s různou složitostí). Jelikož demonstrační implementace představuje proces rozpoznávání relativně malého počtu izolovaných slov či sousloví, která jsou vytvořena stejným procesem zřetězení patřičných modelů, je možné využít jednoduchý Viterbiho dekodér pro výpočet celkové pravděpodobnosti P založený na výpočtu dopředných pravděpodobností a pro jednotlivé modely všech alternativ. Pokud by však byl slovník rozsáhlejší a alternativ pro rozpoznání bylo větší množství, byl by postup takového dekodéru spíše neefektivní [45].

Viterbiho dekodérem se počítá pravděpodobnost pro optimální průchod modelem (výpočet je urychlen hledáním pouze jednoho nejpravděpodobnějšího průchodu). Výpočet celkové pravděpodobnosti průchodu modelem je skriptován rekurentně podle následujícího vzorce:

$$ln(\alpha_j[t])) = ln(b_j(\mathbf{s}_t)) + \max_{2 \leq i \leq N-1} \{ln(\alpha_i[t-1]) + ln(a_{ij})\} \tag{3.2}$$

a využívá tak efektivního výpočtu s logaritmy pravděpodobností. Inicializace pro $t = 1$ pak dle vzorce 3.3. Konečná pravděpodobnost je počítána dle vzorce 3.4 a normována vůči celkové



Obrázek 3.13: Průběh kepstrálních koeficientů pro slovo „six“, včetně krajních pauz

délce (N).

$$ln(\alpha_j[1])) = ln(b_j(\mathbf{s}_1)) + ln(a_{1j}) \tag{3.3}$$

$$ln(\alpha_N[T])) = \max_{2 \leq i \leq N-1} \{ln(\alpha_i[T-1]) + ln(a_{iN})\} \tag{3.4}$$

Optimální průchod modelem (cesta průchodu jednotlivými stavy) byl získán zpětným trasováním dle vzorce 3.7, kde *trace* je pro inicializaci *(t = 1)* roven jedné, v průběhu průchodu dle vzorce 3.5 a pro *(t = T)* (konec průchodu) pak dle vzorce 3.6.

$$trace_j(t) = \underset{2 \leq i \leq N-1}{\operatorname{argmax}} \{ln(\alpha_i[t-1]) + ln(a_{ij})\} \tag{3.5}$$

$$trace_N(T) = \underset{2 \leq i \leq N-1}{\operatorname{argmax}} \{ln(\alpha_i[T-1]) + ln(a_{iN})\} \tag{3.6}$$

$$i*_t = trace_{t+1}(i*_{t+1}) \; pro \; t = T-1, T-2 \ldots 1 \tag{3.7}$$

Výstupem Viterbiho algoritmu není tedy jen výsledná pravděpodobnost, dopředná pravděpodobnost, ale také pravděpodobnost průchodu modelem a cesta průchodu modelem (cesta přes jednotlivé stavy). Testování (resp. rozpoznávání) probíhá v implementaci ve funkci *datatesting* a výstupem jsou vždy výsledky vztažené ke všem slovům ve slovníku (rozpoznávaný řečový signál je srovnáván se všemi modely slov v databázi). Hrubý náhled na strukturu výsledků rozpoznávání v MATLABu je vidět na obr. 3.14, je stejná pro všechny výstupní proměnné.

probab.perfume

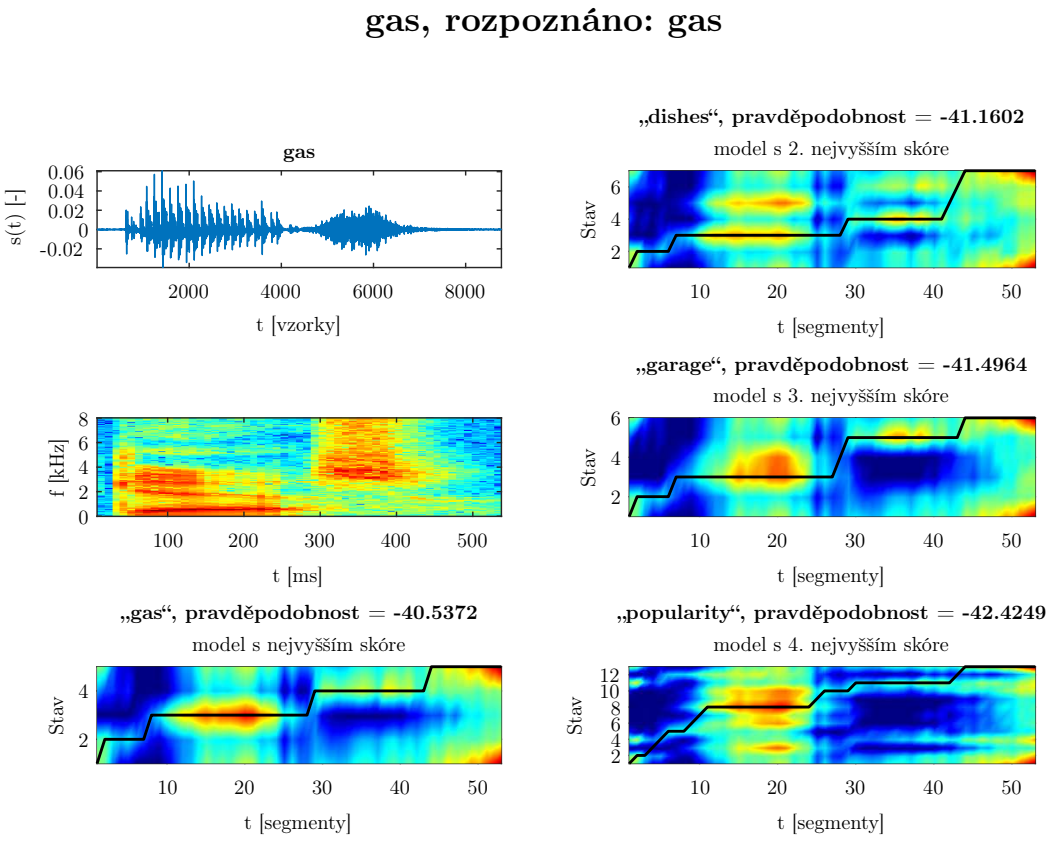
probab.perfume

Field	Value
back	-45.8693
beautiful	-44.4333
beforefrost	-45.2203
bleachers	-45.0498
blouses	-45.2553
blow	-45.6013
bonfire	-45.2298
businessmergers	-45.0905
careful	-43.9837
dishes	-44.8854
garage	-44.7180
garbage	-45.4319
gas	-45.6865
marvelously	-44.6369
overalls	-45.3615
peanutoil	-43.6434
people	-43.9389
perfume	-42.1129

Obrázek 3.14: Struktura výsledné pravděpodobnosti (výstupu rozpoznávání) v MATLABu, resp. pravděpodobnost přiřazení vysloveného slova „perfume“ k příslušnému modelu

3.7 Rozhraní a prezentace výsledků

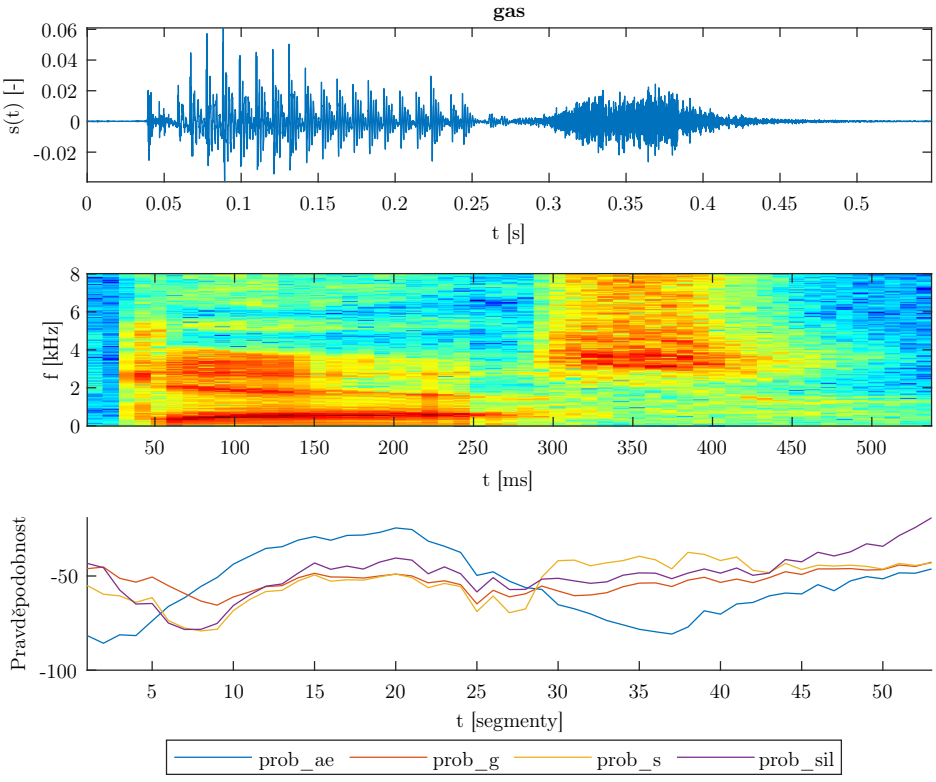
Byť jsou výstupem funkce *datatesting* všechny výsledky rozpoznávacího procesu, které se dají vyjádřit numericky, pro prezentaci výsledků je klíčová jejich vizualizace a podrobnější vyhodnocení. K tomu je v implementaci připravena funkce *plottesting*, prezentující několik výsledků.



Obrázek 3.15: Výsledky rozpoznávání slova „gas“ (1)

V první řadě je uživateli zobrazen časový průběh analyzované nahrávky spolu s jejím spektrogramem, dále pak logaritmická pravděpodobnost a cesta průchodu modelem na barevné škále s informací o celkové pravděpodobnosti rozpoznání. Prezentovány jsou čtyři výrazy s nejvyšším skóre a je také explicitně zmíněn model, který by byl rozpoznávačem vyhodnocen jako správný. Z takových výsledků je jasně vidět, že správným modelem cesta (na obr. 3.15 vyobrazena jako černá linka) prochází postupně od prvního emitujícího stavu až po ten poslední, a to přibližně po diagonále. Při pozorování časového průběhu signálu a cesty modelem je viditelné, že časový průběh (příslušná vyslovená hláska) odpovídá stavu, ve kterém se cesta zrovna nachází. Vykreslená cesta zároveň, dá se říci, ve správném modelu kopíruje nejvyšší logaritmickou pravděpodobnost (červené hodnoty v barevném vykreslení). Tyto vysoké pravděpodobnosti se vyskytují také v jiných místech modelu, ale neodpovídají předpokladu levo-pravého dopředného posunu z prvního do posledního stavu při průchodu. Nijak tedy nepřispívají ke správné detekci, resp. rozpoznání.

Dalším výsledkem, který je pro analýzu výsledku zajímavý, je výstup z funkce *plotoneresult*. Ta vykresluje analyzovaný signál i s pravděpodobností jednotlivých hlásek, které jsou v modelu slova obsaženy (obr. 3.16). Je tedy jasně vidět, jaká hláska je v jaké části analyzované řeči s nejvyšší pravděpodobností zastoupena. Na začátku je pro krátký úsek (cca 2 segmenty) maximální fialová křivka, představující model sil (ticho, neřečový úsek), po něm oranžová hláska g, na delší dobu nejvyšší modrá křivka pro ae a žlutá pro hlásku s. Na posledních segmentech opět převládá model sil - takový průběh maxim tedy zcela odpovídá fonetickému přepisu pro rozpoznávané slovo[23] i časovému průběhu signálu.



Obrázek 3.16: Výsledky rozpoznávání slova „gas“ (2)

Všechny postupy (fáze realizace) jsou dostatečně jasně komentovány, uživatel je v průběhu procesu informován prostřednictvím příkazového okna tak, aby nejen věděl, v jakém stavu se aktuální výpočet nachází, ale případně mohl zakročit a provést kýžené změny například v defi-

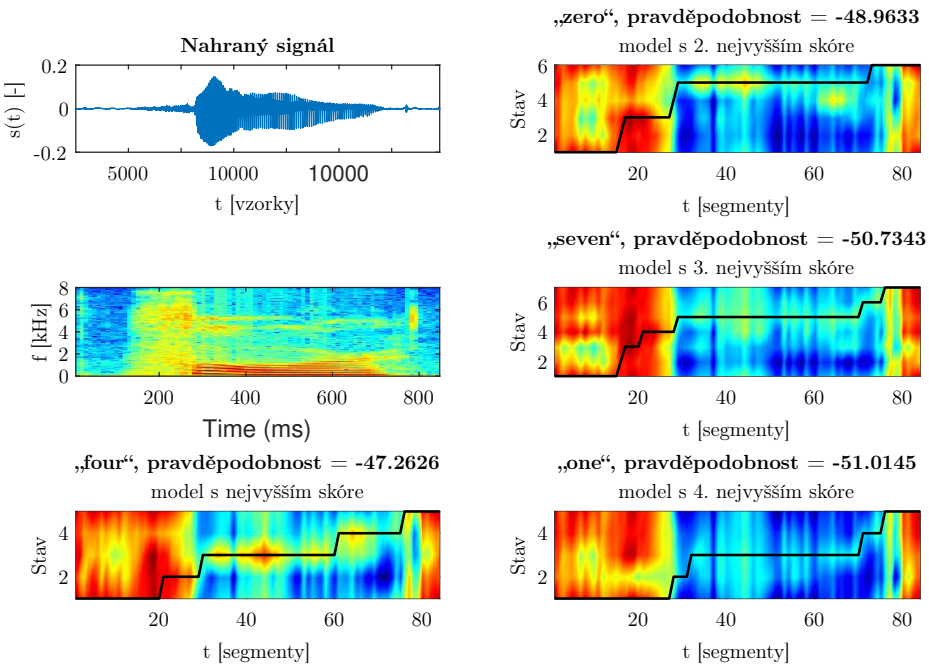
[23] gas sil g ae s sil

novaných parametrech. Nebylo použito žádného pokročilého GUI, a to zejména proto, že implementace byla od počátku zamýšlena jako prostředek pro výukové účely s názorným průchodem všemi fázemi a vizualizací průběžných fází rozpoznávání. V konečné části je uživateli zobrazena také úspěšnost rozpoznávání.

3.8 Online rozpoznávač izolovaných slov

Kromě rozpoznávání slov z předem připravených nahrávek v úložišti (které můžeme nazvat *offline* či *pasivním* rozpoznáváním) je v rámci práce připraven také rychlý online rozpoznávač řeči, který umí pracovat s libovolným vytvořeným slovníkem (stejně jako ten pasivní). Uživatel je po jeho spuštění vyzván k promluvě (nahrávání je spuštěno a zastaveno stisknutím *Enter* na klávesnici), která je pak automaticky porovnána se všemi modely v databázi, resp. v daném slovníku. Vrátí pak uživateli - obdobně jako u pasivního rozpoznávače - označení rozpoznaného výrazu spolu s grafickou reprezentací. V té je uživatel seznámen se čtyřmi výrazy, jež z rozpoznávání vyšly s největší pravděpodobností, viz obr. 3.17, kde je jasně vidět, že tento způsob práce s online nahrávanými řečovými signály je úspěšný, byť správný model se od špatných (ve většině případů) neliší příliš - pravděpodobně zejména kvůli způsobu nahrávání (odlišné akustické pozadí, které v běžné situaci není tiché apod.).

Tato funkcionalita slouží zejména k rychlému testování systému, demonstraci fungování a prezentaci výsledků. Nebyla součástí zadání práce a byť používá stejných metod a funkcí, neprovází uživatele celým procesem rozpoznávání nijak názorně, nenabízí žádné mezivýsledky.



Obrázek 3.17: Výsledek online rozpoznávání vysloveného výrazu „four“

3.9 Dostupnost řešení

Byť je demonstrační implementace vytvořena univerzálně - tedy je možné na vstup nahrát libovolnou řečovou databázi s dostupným fonetickým přepisem - pro počítání, vytváření modelů a celkové splnění zadání bylo využito korpusů, jež jsou chráněny licenční smlouvou. Aby bylo možné i přesto skripty v prostředí MATLAB libovolně spouštět po ukončení diplomové práce i uživateli, kteří nejsou držiteli licenčního oprávnění - a to buď blok po bloku, nebo například jen pro testování či online demonstraci - byly připraveny soubory ve formátu *.mat*, které obsahují napočítaná řešení pro každý z dílčích bloků, slovníků a jazykových mutací. V hlavním skriptu *main.m* je pak možné je na příslušných milnících načíst pomocí připraveného příkazu *load*. Tyto soubory není možné k práci přiložit ve webovém rozhraní ani na CD nosiči (jsou příliš objemné), jsou však k dispozici na fakultním webovém serveru Laboratoře zpracování řeči http://noel.feld.cvut.cz/speechlab v sekci *Ke stažení (Downloads)*.

Kapitola 4

Analýza signálů z dostupných databází

Jednou ze zadaných úloh práce bylo vytvořený rozpoznávací systém otestovat na signálech z dostupných databází. Tato část shrnuje konečné i dílčí výsledky provedených testů a seznamuje s daty, které byly pro testovací (potažmo dříve trénovací) fáze rozpoznávacího systému využity, a také s celkovou úspěšností demonstrační implementace.

4.1 Dostupná data

Testovací fáze práce s sebou nese potřebu přípravy dat, na kterých se připravený systém otestuje. Tento krok je úzce spjatý se slovníkem, na který je rozpoznávací systém natrénován - slova, která nejsou ve slovníku, nemohou být rozpoznávána.

Každá z nahrávek, jež byla použita pro testování, je ve formátu *.wav* a nese název shodný s heslem ve slovníku. Stejně jako jsou vytvořeny fonetické přepisy pro slovníková slova s počátečním a koncovým modelem sil, také nahrávky jsou připraveny tak, aby na začátku a na konci existovala řečová pauza (resp. neřečová část).

4.1.1 Testované signály z TIMIT

Pro testování promluv bylo z nahrávek databáze vyjmuto několik signálů, jejichž obsahem jsou promluvy izolovaných slov dle použitého slovníku (*timit_words* či *digits*). Aby testování probíhalo korektně, signály by měly být odlišné od těch, které byly využity v testování. Proto byly řečové signály patřící do *timit_words* vyjmuty z definované testovací množiny. Pro sběr signálů vyslovených číslovek nebylo možné dat TIMIT korektně využít, kompletní slovník není pokryt testovacím setem - některá slova se vyskytují pouze v trénovací množině.[1]

4.1.2 Testované signály ze SPEECON

Pro testování řeči dle slovníku *cislovky*, vytvořeného pro český jazyk, bohužel nebylo možné využít testovacích promluv řečového korpusu. Tato sada byla tvořena pro jiné účely testování, konkrétně pro zkoumání přesnosti fonetické segmentace. Signály obsahující vyslovené číslovky 0-9 tak byly vyjmuty ze sady trénovacích nahrávek, konkrétně ze souborů *SA***CI*.CS0* (tj. s kódy CI1, CI2, CI3 a CI4), jelikož se jedná o promluvy s izolovanými číslovkami (a jedna promluva odpovídá jedné řečené číslovce).

4.1.3 Vlastní databáze nahrávek

Mimo řečového korpusu TIMIT a SPEECON byla pro testování vytvořené demonstrační implementace nahrána malá databáze nahrávek, obsahující izolovaná slova a sousloví totožná s promluvami z anglického řečového korpusu (resp. slovníku, který byl testován). To zejména proto,

[1] Konkrétně jde o slova „four“ a „seven“, která by byla vyňata z jiných sousloví.

aby bylo analyzováno několik nahrávek, různících se nejen akustickým prostředím, ve kterém byly pořízeny, ale také výslovností. Rodným jazykem všech mluvčích, jež do nahrávání přispěli svými nahrávkami, je totiž čeština či slovenština.

Soubor nahraných dat byl pořízen v průběhu realizace této diplomové práce a bohužel, z důvodu omezení pojících se s nouzovým stavem v České republice spojeného s epidemií koronaviru (SARS-CoV-2), není příliš rozsáhlá. Obsahuje však promluvy od celkem 16 ženských i mužských mluvčích s věkem mezi 18-54 lety. Nahrávky byly pořízeny v programu Praat[2], a to se vzorkovací frekvencí 16 kHz a kódováním PCM 16-bit.[3] Všechny řečové signály byly nahrávány v domácím prostředí všesměrovým kondenzátorovým stolním mikrofonem o frekvenčním rozsahu 50 Hz až 16 kHz, citlivostí -30 dB/mW. Celá vytvořená databáze je dostupná na webovém serveru Laboratoře zpracování řeči, viz sekce 3.9.

4.2 Optimalizace nastavení

Při diskuzi nastavení počtu pásem banky filtrů pro parametrizaci, resp. výpočet melovských kepstrálních koeficientů, se v práci vycházelo z praktických zkušeností a teoretického předpokladu uvedeného v doporučené literatuře [17]. V rámci práce došlo k otestování různých nastavení a vzhledem k dosaženým výsledkům, jež s těmito předpoklady korelovaly, bylo pro trénovací data (se vzorkovací frekvencí 16 kHz) použito M = 20.

Při trénování GMM modelů jednotlivých hlásek je možné nastavit různý počet směsí Gaussových rozdělení pro zpřesnění trénovaného modelu. V ideálním případě je nastaven počet směsí mixnum = 4, avšak rozdíl v případě užití nižšího počtu není natolik markantní, viz tab. 4.1, 4.2 a 4.3. Znatelnější by rozdíl byl v případě, že by v práci bylo využito ladění modelů HMM například Baum-Welchovou reestimací, kde by se počítalo nejdříve s jednou směsí, přetrénovalo, zvýšilo na dvě a dále až po finální čtyři směsi.

Nižší počet směsí je však vzhledem k přihlédnutí k výsledným hodnotám vhodné zvolit v případě, kdy by se rozpoznávač využíval na menším setu trénovacích dat a neměl by dostatečné zastoupení všech hlásek mezi vzorky. Jednotlivé modely pak spíše konvergují a není třeba je při procesu přípravy rozpoznávače tolik kontrolovat. V případě vytvořené implementace je využito čtyř směsí, ale je nutné při trénování modelů hlídat konvergenci pro jednotlivé hlásky, případně pro některé z nich trénování zopakovat.[4]

Tabulka 4.1: Vliv počtu směsí GMM na výsledek rozpoznávání, na vstupu výraz SUGAR

výraz / počet směsí GMM	mixnum = 1	mixnum = 2	mixnum = 3	mixnum = 4
perfume	-47,7399	-48,5853	-48,0593	-48,6443
bonfire	-48,1893	-48,6872	-48,9883	-48,6362
business mergers	-48,6998	-48,2374	-48,2867	-48,1823
popularity	-49,2250	-48,8991	-48,9790	-49,3920
peanut oil	-49,8507	-49,3874	-49,1545	-49,5843
sugar	**-45,1464**	**-44,9706**	**-44,4243**	**-44,3188**
potatoes	-49,6771	-49,6447	-49,2070	-49,1967
dishes	-48,0588	-47,6769	-47,2371	-47,1498
promote birth control	-48,3080	-48,6863	-48,0492	-48,5320
Úspěšnost	100 %	100 %	100 %	100 %

[2] **Praat: doing phonetics by computer**, verze 6.1.08 [volně dostupné online], Praat.org, autoři: Paul Boersma, David Weenink (Phonetic Sciences, University of Amsterdam)

[3] Řečové signály byly nahrávány jako standardní 16 bitové PCM soubory bez hlavičky (little-endian) s koncovkou *.CS0*, po editaci (oříznutí jednotlivých výrazů) uloženy do formátu *.wav*.

[4] Faktem však je, že užití vyššího počtu směsí je vždy lepší. V praktických realizacích rozpoznávačů spojité i nespojité řeči se používá i řádově větší počet směsí, což s sebou přináší nutnost většího množství trénovacích dat.

Tabulka 4.2: Vliv počtu směsí GMM na výsledek rozpoznávání, na vstupu výraz POPULARITY

výraz / počet směsí GMM	mixnum = 1	mixnum = 2	mixnum = 3	mixnum = 4
perfume	-50,3438	-49,0710	-48,6334	-48,3376
bonfire	-50,7542	-49,8994	-49,1276	-48,7222
business mergers	-51,3250	-50,2011	-49,4590	-49,2190
popularity	**-47,4463**	**-46,7469**	**-46,1338**	**-46,0353**
peanut oil	-50,0081	-49,1920	-48,6262	-48,1738
sugar	-50,5315	-49,8540	-49,4286	-48,9404
potatoes	-50,3227	-50,2059	-49,6739	-49,3386
dishes	-51,4657	-51,2051	-51,0478	-50,4929
promote birth control	-50,5416	-49,6427	-49,0260	-48,5917
Úspěšnost	100 %	100 %	100 %	100 %

Tabulka 4.3: Vliv počtu směsí GMM na výsledek rozpoznávání, na vstupu výraz PROMOTE BIRTH CONTROL

výraz / počet směsí GMM	mixnum = 1	mixnum = 2	mixnum = 3	mixnum = 4
perfume	-48,4211	-47,6858	-47,2811	-46,8356
bonfire	-47,5886	-47,1292	-46,7682	-46,2315
business mergers	-47,4661	-46,9499	-46,6894	-46,1910
popularity	-47,5750	-46,5582	-46,2400	-45,6133
peanut oil	-47,3682	-46,1895	-45,5348	-45,1465
sugar	-48,0655	-46,8690	-46,4704	-46,1539
potatoes	-47,6388	-45,9234	-45,6843	-45,2533
dishes	-49,7371	-48,0932	-47,6589	-47,1105
promote birth control	**-44,0694**	**-42,8156**	**-42,2594**	**-42,0245**
Úspěšnost	100 %	100 %	100 %	100 %

4.3 Úspěšnost rozpoznávání

V rámci práce bylo provedeno testování rozpoznávacího systému a vytvořených HMM modelů pro dostupné (resp. vytvořené) nahrávky, a to pro všechny slovníky - v českém i anglickém jazyce. Výsledky testování nejsou považovány za stěžejní výsledek demonstračního prostředí, vzhledem k tomu, že cílem bylo vytvořit spíše názornou implementaci, než efektivní s vysokou úspěšností (pro jejíž vytvoření by nebylo vhodné pracovat v prostředí MATLAB a pravděpodobně by zahrnovalo zejména práci s neuronovými sítěmi). Jedná se však o plnohodnotnou aktivitu v rámci zadání práce samotné.

Pro anglický jazyk byly uskutečněny tři skupiny testování, a to:

- testování modelů slovníku *timit_ words* a nahrávek vyjmutých z řečového korpusu TIMIT,
- testování modelů slovníku *timit_ words* a nahrávek z vlastní pořízené databáze,
- testování modelů slovníku *digits* a nahrávek z vlastní pořízené databáze.[5]

Pro názorné zhodnocení výsledků je nejefektivnější zobrazení průběhu pravděpodobností, jež jsou v MATLAB prostředí prezentovány (pro slovník *timit_ words* ukázka viz obr. 3.15 a 3.16). Máme-li však hodnotit čistě jen úspěšnost, je možné ji prezentovat výslednými pravděpodobnostmi testování několika slov ze slovníkové množiny. Na následujících testech jsou prezentovány výsledky pro vybrané (odlišné) výrazy, resp. pravděpodobnost, s jakou je určeno, že řečený výraz přísluší danému modelu. Nejvyšší pravděpodobnost odpovídá modelu, jež by byl vybrán jako správný.

[5] V promluvách korpusu TIMIT se všechny číslovky nevyskytují a nebylo by tedy možné je pro testování korektně připravit.

Rozdíl mezi výslednou pravděpodobností přiřazení ke správnému modelu slova a mezi přiřazením ke špatnému modelu slova není příliš velký. Je tomu tak zejména proto, že v implementaci je pracováno s jednostavovými HMM monofónů. Lepších výsledků by bylo dosaženo vytvořením třístavových modelů, případně přetrénováním těchto vícestavových modelů např. Baum-Welchovou reestimací (či jiným trénovacím, reestimačním algoritmem).

Výsledky testování slovníku *timit_words* jsou prezentovány pro 6 výrazů. Úspěšnost demonstračního prostředí dosáhla na nahrávkách pocházejících z korpusu TIMIT 100 % (viz tab. 4.4, resp. B.1). To lze vysvětlit nejen podobností akustického prostředí, ve kterém byly nahrávky pořízeny, ale také podobně diverzifikovanou skupinou mluvčích a faktem, že vytvořené GMM pocházejí z trénovacích dat, která byla velmi správně foneticky přepsána a zdokumentována.

Tabulka 4.4: Výsledky rozpoznávání pro slovník *timit_words* s nahrávkami z korpusu TIMIT

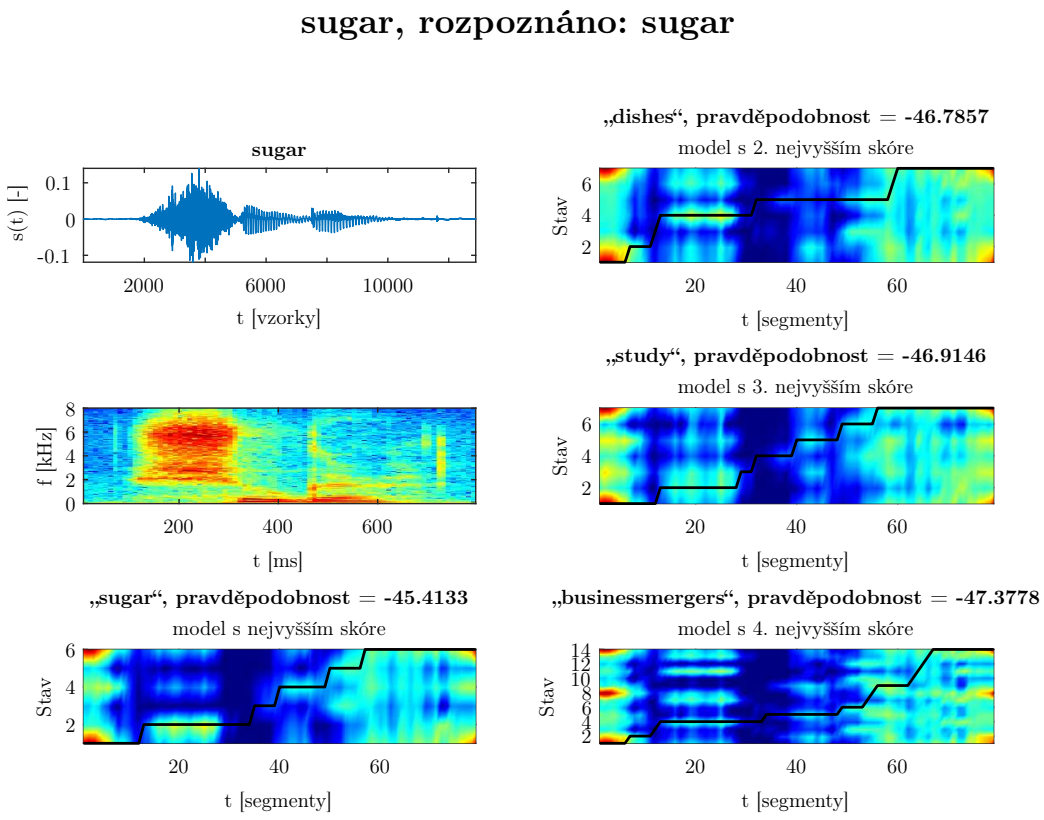
model / nahrávka	bonfire	peanut oil	sugar	potatoes	dishes	promote birth control
bonfire	**-42,2851**	-47,2920	-48,6362	-42,3370	-42,8317	-46,2315
peanut oil	-43,9995	**-42,8882**	-49,5843	-41,2922	-43,1268	-45,1465
sugar	-45,8020	-47,2155	**-44,3188**	-43,2839	-43,9159	-46,1539
potatoes	-44,5759	-45,4149	-49,1967	**-40,3603**	-41,5986	-45,2533
dishes	-46,2362	-45,7295	-47,1498	-42,3731	**-39,1232**	-47,1105
promote birth control	-44,1549	-45,1902	-48,5320	-41,6654	-44,4945	**-42,0245**

Pouze z numerických výsledků je viditelné, že delší slova a sousloví se výslednou pravděpodobností odlišují výrazněji než krátké výrazy. Zároveň také vidíme, že slova, jež obsahují podobně výrazné hlásky (například slova, která začínají explozivní hláskou p) mají velice podobnou výslednou pravděpodobnost.

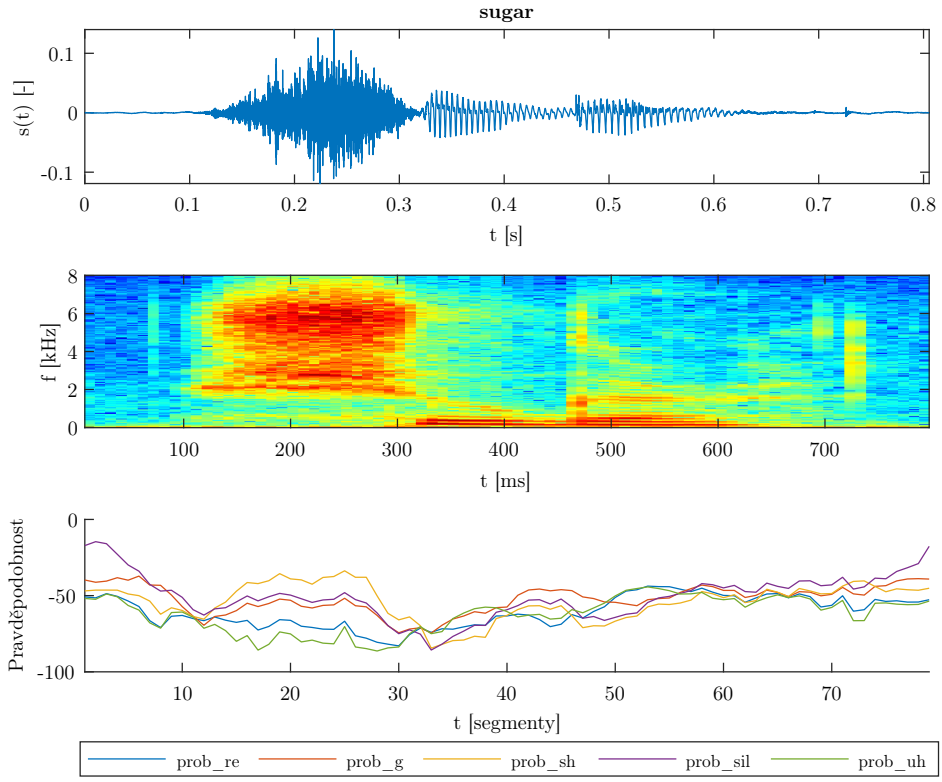
Obdobně tomu je při testování vlastních nahrávek k slovníkům *timit_words* a *digits*. Pro prezentaci v této práci bylo náhodně vybráno výsledků pro dva mluvčí - jeden ženský a jeden mužský hlas, mluvčí **K1** (tab. 4.5, resp. B.2 a 4.7, resp. B.4) a **J1** (tab. 4.6, resp. B.3 a 4.8, resp. B.5) - v grafické prezentaci pro jedno vybrané slovo pak na obr. 4.1, 4.2 (4.3, 4.4) a 4.5, 4.6 (4.7, 4.8). U obou mluvčích systém vykazuje 100% úspěšnost pro slovník *timit_words*. Celkově jsou trendy velmi podobné výsledkům se záznamy z databáze TIMIT, je očividné, že pokud na vstup rozpoznávacího systému vstupují kvalitní zvukové nahrávky s nízkým okolním šumem a ruchem ze zdroje, jež je od mikrofonu v běžné vzdálenosti, funguje i vytvořená implementace, byť je demonstrační, velmi uspokojivě.

Tabulka 4.5: Výsledky rozpoznávání pro slovník *timit_words* s nahrávkami z vlastní databáze, mluvčí K1

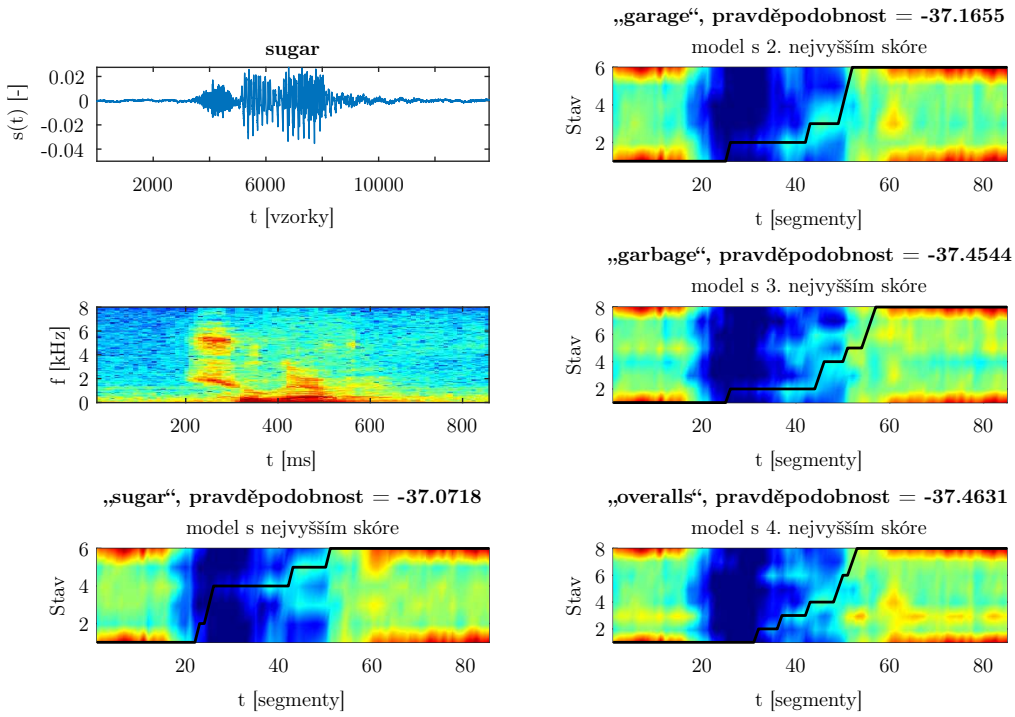
model / nahrávka	bonfire	peanut oil	sugar	potatoes	dishes	promote birth control
bonfire	**-39,0163**	-44,8945	-48,4244	-41,2450	-46,6952	-45,9585
peanut oil	-41,3065	**-41,5985**	-47,9605	-40,5579	-45,1534	-45,0850
sugar	-41,5105	-46,0102	**-45,4133**	-40,9826	-44,2114	-46,0624
potatoes	-41,0039	-44,8529	-48,3206	**-39,5260**	-42,6990	-45,4431
dishes	-41,1552	-45,4779	-46,7857	-40,0736	**-40,9992**	-47,1624
promote birth control	-40,2680	-43,8817	-48,3434	-41,4056	-46,4356	**-42,3245**



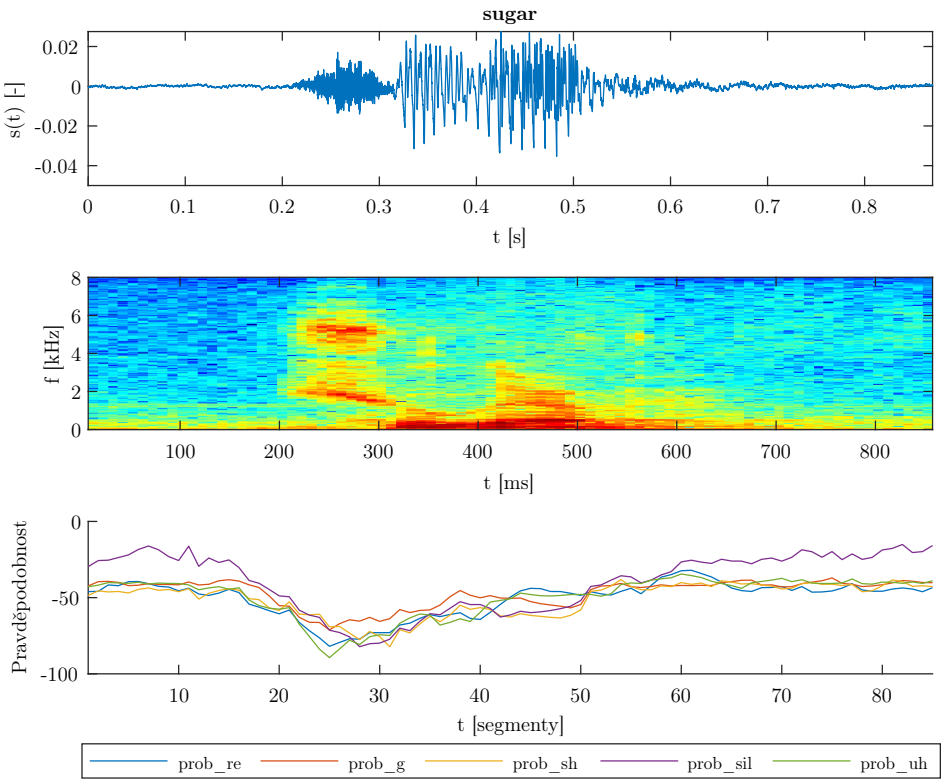
Obrázek 4.1: Výsledky rozpoznávání slova „sugar“ (1, K1)



Obrázek 4.2: Výsledky rozpoznávání slova „sugar“ (2, K1)



Obrázek 4.3: Výsledky rozpoznávání slova „sugar“ (1, J1)



Obrázek 4.4: Výsledky rozpoznávání slova „sugar“ (2, J1)

Tabulka 4.6: Výsledky rozpoznávání pro slovník *timit_ words* s nahrávkami z vlastní databáze, mluvčí J1

model / nahrávka	bonfire	peanut oil	sugar	potatoes	dishes	promote birth control
bonfire	**-38,7524**	-46,7906	-37,6523	-46,3152	-47,0698	-46,3115
peanut oil	-40,2803	**-42,5405**	-37,9554	-43,8525	-45,4162	-45,7917
sugar	-42,1797	-46,6797	**-37,0718**	-46,4925	-44,9758	-47,2855
potatoes	-41,6339	-45,5781	-38,2670	**-41,4276**	-43,4836	-46,6540
dishes	-40,5848	-45,5776	-38,0909	-44,6048	**-41,3678**	-47,7115
promote birth control	-40,9497	-45,3402	-38,3023	-44,5082	-46,8286	**-43,8891**

Tabulka 4.7: Výsledky rozpoznávání pro slovník *digits* s nahrávkami z vlastní databáze, mluvčí K1

model / nahrávka	one	two	five	seven	nine	zero
one	**-25,7884**	-23,9211	-30,5827	-24,3794	-24,5289	-30,7841
two	-26,6794	**-23,0838**	-31,2079	-25,0916	-25,2092	-30,2588
five	-26,4915	-24,9401	-30,4957	-24,9433	-24,8356	-31,1249
seven	-26,2191	-23,6399	-30,5538	**-23,9018**	-24,5993	-30,5050
nine	-25,9013	-23,8514	**-30,1539**	-24,4417	**-23,9913**	-30,5577
zero	-26,6800	-23,4860	-30,9769	-24,4348	-25,0786	**-29,8856**

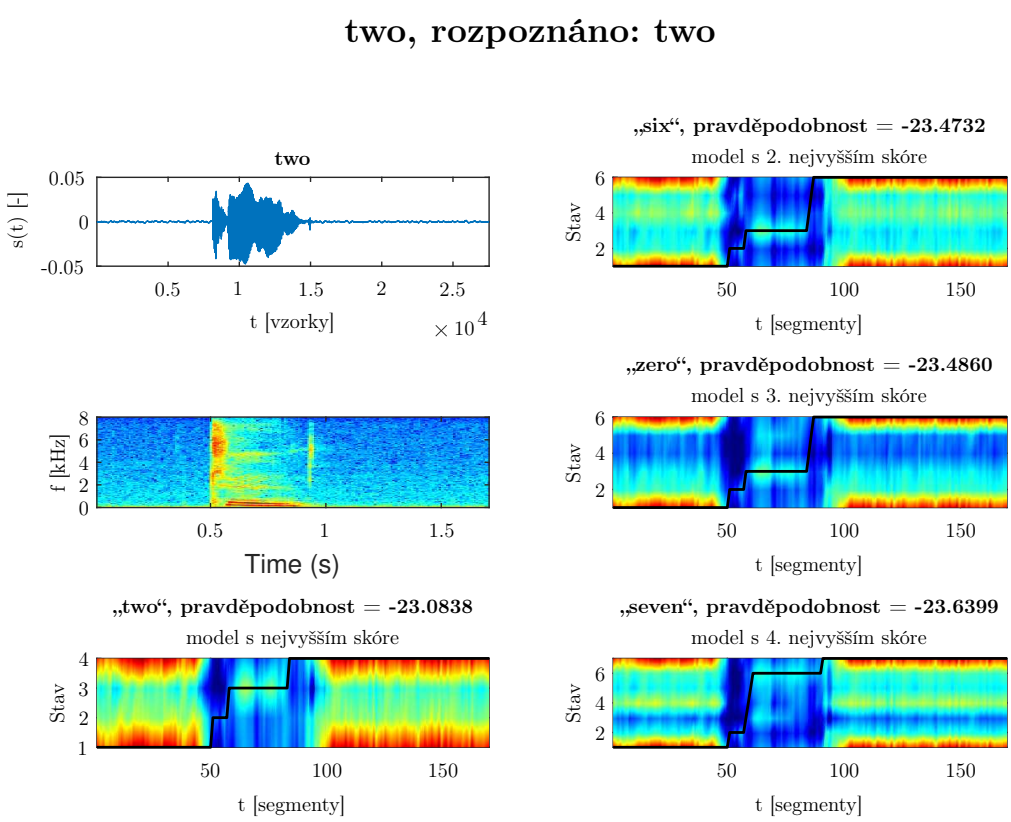
Tabulka 4.8: Výsledky rozpoznávání pro slovník *digits* s nahrávkami z vlastní databáze, mluvčí J1

model / nahrávka	one	two	five	seven	nine	zero
one	**-25,7884**	-23,9211	-30,5827	-24,3794	-24,5289	-30,7841
two	-26,6794	**-23,0838**	-31,2079	-25,0916	-25,2092	-30,2588
five	-26,4915	-24,9401	-30,4957	-24,9433	-24,8356	-31,1249
seven	-26,2191	-23,6399	-30,5538	**-23,9018**	-24,5993	-30,5050
nine	-25,9013	-23,8514	**-30,1539**	-24,4417	**-23,9913**	-30,5577
zero	-26,6800	-23,4860	-30,9769	-24,4348	-25,0786	**-29,8856**

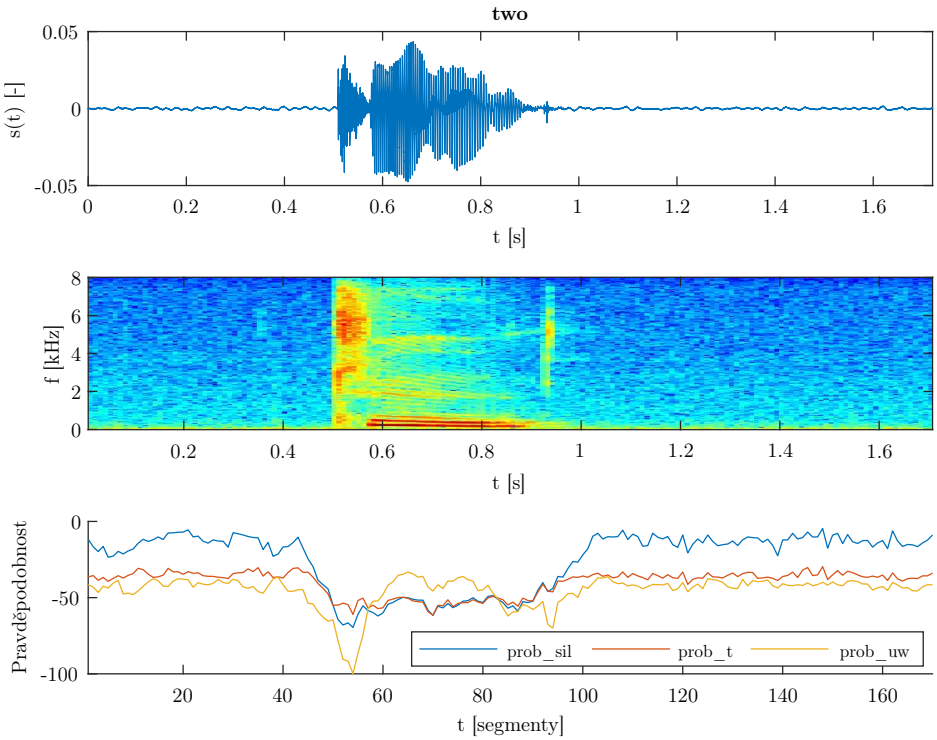
Na nahrávkách odpovídajících slovníku *digits* je u obou mluvčích úspěšnost na daných výrazech 83,3 % (přes celý slovník, tedy při testování všech nahrávek ze slovníku - číslic 0-9 - pak jen 70 %). Nižší úspěšnost je následkem nejen fonetické podobnosti některých výrazů (five, nine), ale také velmi krátké délky slov. Tím pádem i rozdíly mezi správnými a špatnými modely nejsou tak markantní. Dalším možným důvodem pro takový výsledek je odlišná výslovnost mluvčích oproti těm z trénovací množiny - vlastní nahrávky nebyly pořízeny s rodilými mluvčími s daným americkým akcentem.[6]

Výsledky testování na nahrávkách vyjmutých z databáze SPEECON je prezentováno pro všechna slova ze slovníku (resp. pro číslovky 0-9) v tab. B.6, ve zkrácené množině v tab. 4.9. Jak se dalo očekávat, jelikož všechny nahrávky jsou vyjmuty z trénovací sady, úspěšnost rozpoznávání je 100 %. Negativně se zde neprojevil ani vliv automatické segmentace.

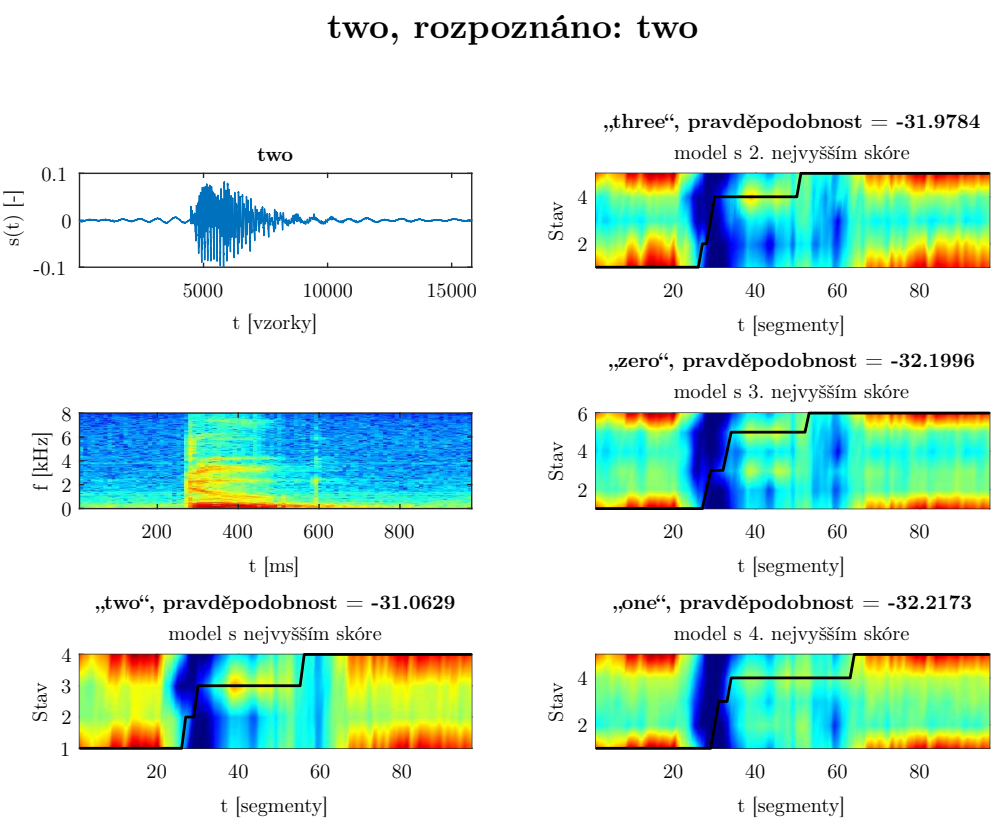
[6]Toto je potřeba vzít v úvahu ve všech testovacích scénářích, kde je testováno s vlastními nahrávkami tohoto charakteru



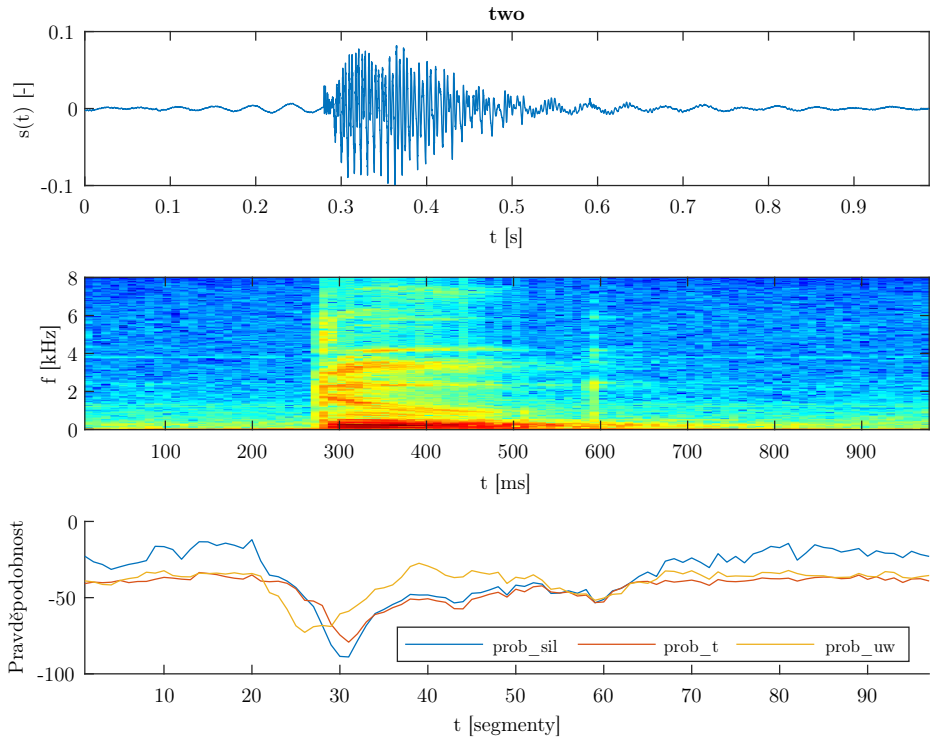
Obrázek 4.5: Výsledky rozpoznávání slova „two“ (1, K1)



Obrázek 4.6: Výsledky rozpoznávání slova „two“ (2, K1)



Obrázek 4.7: Výsledky rozpoznávání slova „two“ (1, J1)

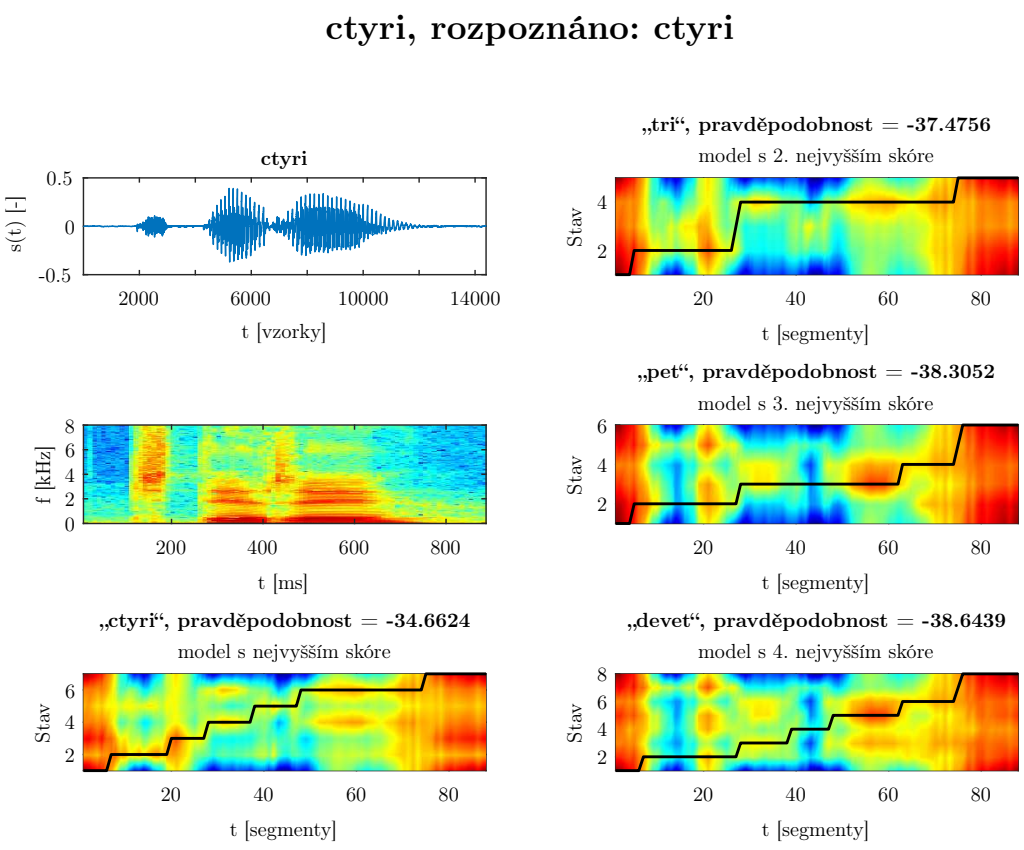


Obrázek 4.8: Výsledky rozpoznávání slova „two“ (2, J1)

Tabulka 4.9: Výsledky rozpoznávání pro slovník *cislovky* s nahrávkami z databáze SPEECON, všechny modely a signály

model / nahrávka	nula	dva	pet	sest	sedm	osm
nula	**-35,6468**	-35,7165	-31,8498	-37,0650	-36,8973	-39,0226
dva	-36,9249	**-35,6863**	-33,0019	-37,3342	-37,0835	-41,7575
pet	-38,0749	-42,3564	**-28,5476**	-35,4022	-35,6938	-40,5619
sest	-39,1806	-43,6504	-31,1440	**-32,3088**	-35,6902	-39,0412
sedm	-37,1581	-40,2775	-31,7355	-35,0652	**-33,8497**	-39,4493
osm	-37,6000	-40,0288	-32,2164	-36,4210	-35,2073	**-36,2872**

Pro názornou ukázku grafických výstupů testování z databáze SPEECON (jelikož nebyla na těchto datech v práci dosud zobrazena) jsou prezentovány výstupy rozpoznávacího procesu pro nahrávku „ctyri“. Celková a logaritmická pravděpodobnost spolu s pravděpodobností průchodu modelem pro čtyři modely s nejvyšší pravděpodobností na obr. 4.9 a pravděpodobnost jednotlivých hlásek v průběhu nahrávky pak na obr. 4.10.

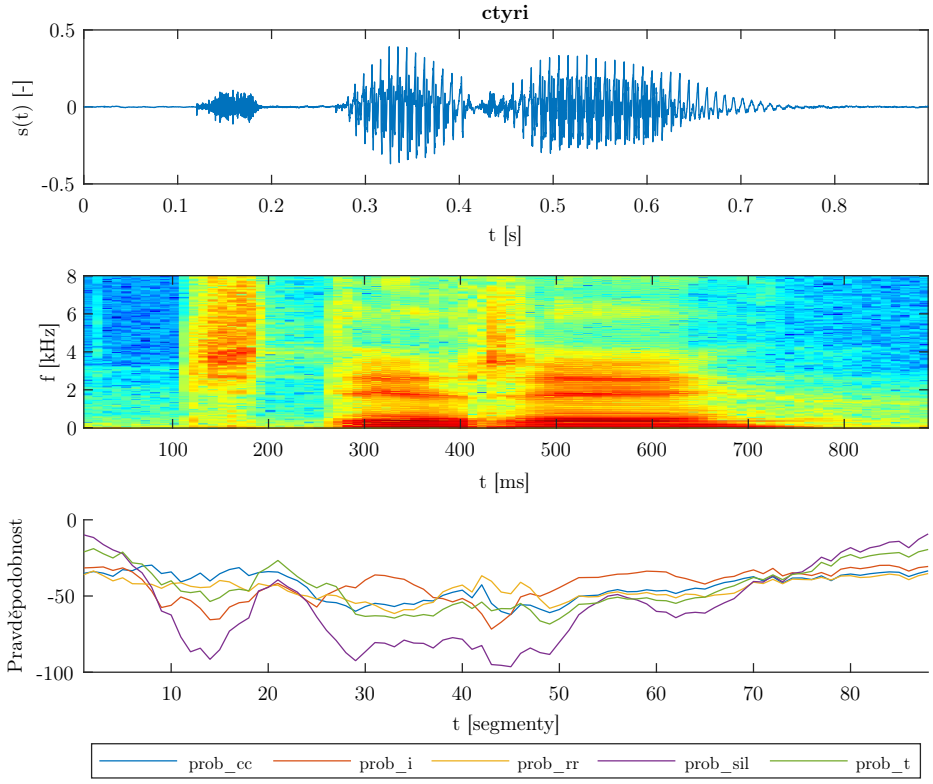


Obrázek 4.9: Výsledky rozpoznávání slova „ctyri“ (1)

4.3.1 Hodnocení

I přes cíl práce, rozcházející se z podstaty věci s přesností a efektivitou realizovaného rozpoznávacícho systému, jsou výsledky rozpoznávání relativně dobré.[7] Byť je implementované řešení

[7] U online rozpoznávače úspěšnost v této míře testována nebyla, velmi závisí na použitém zařízení pro nahrávání zvuku atd.



Obrázek 4.10: Výsledky rozpoznávání slova „ctyri“ (2)

demonstrační, ukazuje zároveň, že princip řešení je doopravdy funkční a použitelný v jednoduchých úlohách rozpoznávání řeči.

Lepších výsledků by bylo možné dosáhnout rozšířením modelu monofónů z jednoho stavu na více (ideálně tři), jejich přetrénováváním, případně modelováním trifónů namísto monofónů.[8] Případné přetrénování by však probíhalo s rozsáhlými trénovacími daty odpovídajícími modelům slovníkových slov.

[8]Tato varianta však není možná s daty TIMIT ani SPEECON, zároveň by asi nebylo možné ji realizovat v prostředí MATLAB.

Kapitola 5

Závěr

Cílem práce bylo realizovat demonstrační systém na bázi GMM-HMM pro rozpoznávání izolovaných slov v prostředí MATLAB, který by bylo možné využívat zejména pro výukové účely a demonstraci základních principů rozpoznávání řečových signálů pomocí skrytých Markovových modelů. Byla navržena implementace takového rozpoznávacího systému pro anglický a český jazyk a zároveň testována na signálech z dostupných - a také v rámci práce vytvořených - databázích. Pro přípravu rozpoznávacího systému se v práci využívaly dva řečové korpusy - TIMIT a SPEECON, avšak systém je natolik univerzálně vytvořen, že je možné zpracovávat data z jiných zdrojů, je-li dostupná jejich fonetická transkripce. Pro trénování bylo využito celé trénovací množiny korpusu TIMIT a těch trénovacích souborů SPEECON, jež jsou označeny jako foneticky bohaté a byly nahrávány v prostředí s nízkým okolním šumem. Systém je tak zároveň nezávislý na mluvčím.

Při parametrizaci dat docházelo k výpočtu MFCC, dynamických a akceleračních parametrů. Akustické modely vycházely z GMM se čtyřmi směsmi a diagonální kovarianční maticí pro každou hlásku zvlášť. Stejně tak byl pro každou elementární subslovní jednotku, resp. monofón, vytvořen akustický HMM model s jedním emitujícím a dvěma neemitujícími stavy. HMM modely slov pak byly vytvořeny řetězením modelů monofónů dle definovaných slovníků s fonetickými předpisy zohledňujícími krajní tichá místa v nahrávkách (tedy náběhy a dozvuky krajních hlásek). Pro proces rozpoznávání, mapování příznaků analyzovaných nahrávek na připravené modely slov, bylo využito jednoduchého Viterbiho dekódovacího algoritmu se zpětným trasováním pro získání optimální cesty průchodu modelem. Rozpoznávaný řečový signál je porovnáván se všemi slovy ze slovníku a dále vyhodnocen. Kromě samotného systému s jasnými bloky a možností nahrávat dílčí vstupy z jiných zdrojů dat byl vytvořen jednoduchý rozpoznávač umožňující analyzované signály přímo nahrávat. Tento blok však slouží pouze jako interaktivní nástroj pro okamžité ověření funkčnosti systému a pro prezentační účely. Jeho postupy nejsou postupně analyzovány, uživateli jsou prezentovány jen výsledky rozpoznávání.

Implementace v rámci jednotlivých trénovacích i testovacích bloků prezentuje dílčí i finální výsledky s ohledem na snadnější pochopení procesů, problematiky i charakteristik analyzovaných signálů a modelů, tedy s ohledem na edukativní cíl práce. Použité postupy kladou důraz na zřetelnost a pochopení problematiky GMM-HMM rozpoznávačů, ne příliš tolik na vytvoření efektivního a ideálního rozpoznávacího systému, k jehož realizaci by bylo využito zcela jiného programového prostředí a rozsáhlejších korpusů trénovacích dat. I přesto byla realizovaná demonstrace otestována.

Testovací množiny dat byly vybrány nejen ze zmíněných databází TIMIT a SPEECON, ale také z databáze řečových nahrávek, která byla pořízena v rámci této práce. Prezentovaný systém i přes svou jednoduchost dosahuje uspokojivých výsledků s úspěšností pohybující se mezi 70 až 100 %. Pro zlepšení těchto výsledků by bylo možné rozvinout vytvořené modely monofónních hlásek na větší počet emitujících stavů a dále je přetrénovávat na rozsáhlejší trénovací množině dat odpovídající slovům ve slovníku, a to například algoritmem Baum-Welchovy rees-

timace.

Skripty a soubory, které byly vytvořeny, mohou sloužit pro podobné experimenty či další práci s GMM-HMM systémem rozpoznávání řeči a jsou součástí CD přílohy této diplomové práce. Řešení i vytvořené nahrávky jsou také dostupné na webovém serveru Laboratoře zpracování řeči http://noel.feld.cvut.cz/speechlab v sekci *Ke stažení (Downloads)*.

Příloha A

Obsah přiloženého CD

Na CD nosiči, jež je přílohou k odevzdané práci, je uloženo následující:

- tato práce ve formátu *.pdf*,
- seznam souborů použitých k trénování rozpoznávače,
- slovníky pro rozpoznávač využité v implementaci,
  - cislovky.txt,
  - digits.txt,
  - timit_words.txt,
- MATLAB skripty a metody implementace,
- řečové signály pořízené v rámci práce pro testování.

Příloha B

Tabulky s výsledky rozpoznávání

Tabulka B.1: Výsledky rozpoznávání pro slovník *timit_words* s nahrávkami z korpusu TIMIT, vyhodnocení pro všechny modely

model / nahrávka	bonfire	peanut oil	sugar	potatoes	dishes	promote birth control
back	-47,3820	-49,1826	-50,6416	-42,8064	-45,2973	-47,7264
beautiful	-45,7056	-44,8378	-49,5147	-41,6655	-40,9197	-45,8146
before frost	-44,0471	-47,0349	-48,5943	-42,6513	-41,1347	-45,2833
bleachers	-46,1020	-46,8811	-45,1599	-42,1880	-40,4940	-46,8753
blouses	-46,5464	-46,0193	-50,8219	-41,9503	-41,8113	-47,6165
blow	-46,9478	-45,8676	-50,4001	-42,5339	-45,9361	-47,8475
bonfire	**-42,2851**	-47,2920	-48,6362	-42,3370	-42,8317	-46,2315
business mergers	-46,7974	-48,5186	-48,1823	-42,6779	-41,6125	-46,1910
careful	-46,2782	-44,6850	-47,6855	-40,9231	-41,9830	-46,5635
dishes	-46,2362	-45,7295	-47,1498	-42,3731	**-39,1232**	-47,1105
garage	-47,9444	-46,1793	-48,3181	-43,7147	-42,8897	-47,2905
garbage	-45,1676	-46,6940	-48,7402	-42,8103	-44,0448	-46,5308
gas	-48,3946	-48,7401	-50,6255	-42,3801	-42,4610	-47,8951
marvelously	-44,6992	-45,1125	-49,1819	-42,9258	-41,2805	-45,4753
overalls	-45,4499	-45,3738	-47,4975	-42,3411	-41,3559	-45,2957
peanut oil	-43,9995	**-42,8882**	-49,5843	-41,2922	-43,1268	-45,1465
people	-46,2911	-44,9856	-49,7136	-41,1905	-43,8417	-46,9808
perfume	-45,9718	-46,5557	-48,6443	-42,1279	-41,2865	-46,8356
poor	-45,7554	-47,0415	-47,8990	-41,5991	-45,2124	-46,9355
popularity	-43,1398	-45,0395	-49,3920	-41,0887	-44,2206	-45,6133
potatoes	-44,5759	-45,4149	-49,1967	**-40,3603**	-41,5986	-45,2533
pretty in autumn	-46,2545	-46,1274	-49,2585	-41,3441	-43,0187	-45,9781
promote birth control	-44,1549	-45,1902	-48,5320	-41,6654	-44,4945	**-42,0245**
real people	-46,5607	-45,4988	-49,0877	-41,9045	-43,9433	-46,0541
study	-46,0623	-46,1443	-47,8382	-41,6592	-44,3164	-47,2737
sugar	-45,8020	-47,2155	**-44,3188**	-43,2839	-43,9159	-46,1539

Tabulka B.2: Výsledky rozpoznávání pro slovník *timit_words* s nahrávkami z vlastní databáze, mluvčí K1, vyhodnocení pro všechny modely

model / nahrávka	bonfire	peanut oil	sugar	potatoes	dishes	promote birth control
back	-41,6034	-46,6626	-48,7631	-42,0321	-46,4413	-47,6694
beautiful	-40,3464	-42,4638	-47,6502	-40,8503	-44,5632	-45,6713
before frost	-40,1053	-45,1859	-49,2485	-41,0099	-43,8180	-45,8107
bleachers	-40,7851	-44,8836	-48,3093	-39,6939	-42,0153	-46,9200
blouses	-39,5849	-44,7881	-48,2257	-40,1949	-42,0831	-47,2427
blow	-40,8964	-44,5840	-48,3990	-42,5743	-47,0098	-46,8958
bonfire	**-39,0163**	-44,8945	-48,4244	-41,2450	-46,6952	-45,9585
business mergers	-40,2144	-45,1686	-47,3778	-39,6711	-42,6341	-45,7313
careful	-41,8498	-44,3849	-48,0398	-41,5305	-46,8783	-46,9083
dishes	-41,1552	-45,4779	-46,7857	-40,0736	**-40,9992**	-47,1624
garage	-41,3936	-45,4222	-48,0426	-41,8608	-44,3435	-47,1291
garbage	-40,8350	-45,1910	-48,2871	-41,5823	-45,1570	-46,3507
gas	-41,6001	-46,2779	-48,3642	-40,0179	-43,4291	-47,3481
marvelously	-40,2017	-43,9342	-47,8450	-40,7938	-44,2095	-44,8401
overalls	-41,2444	-43,9322	-48,3222	-41,5100	-43,2464	-45,4893
peanut oil	-41,3065	**-41,5985**	-47,9605	-40,5579	-45,1534	-45,0850
people	-41,3422	-42,7387	-48,5629	-41,0078	-45,8092	-46,9536
perfume	-40,8775	-45,0988	-47,9870	-40,3870	-46,3546	-46,9579
poor	-41,1517	-45,6824	-49,0268	-41,7009	-46,7230	-47,0496
popularity	-40,1340	-43,9238	-48,4481	-41,0048	-44,5844	-46,4355
potatoes	-41,0039	-44,8529	-48,3206	**-39,5260**	-42,6990	-45,4431
pretty in autumn	-40,7853	-43,6581	-47,6071	-39,8387	-44,9611	-45,3427
promote birth control	-40,2680	-43,8817	-48,3434	-41,4056	-46,4356	**-42,3245**
real people	-40,8251	-42,7596	-48,3559	-40,9009	-45,0258	-45,1109
study	-40,8960	-45,7670	-46,9146	-40,7840	-44,3499	-46,7063
sugar	-41,5105	-46,0102	**-45,4133**	-40,9826	-44,2114	-46,0624

Tabulka B.3: Výsledky rozpoznávání pro slovník *timit_words* s nahrávkami z vlastní databáze, mluvčí J1, vyhodnocení pro všechny modely

model / nahrávka	bonfire	peanut oil	sugar	potatoes	dishes	promote birth control
back	-42,2334	-47,9488	-38,0083	-47,5127	-46,6893	-48,1476
beautiful	-41,2961	-43,6952	-37,9258	-44,1429	-44,7571	-46,3106
before frost	-40,0563	-46,3753	-37,9258	-46,8999	-46,1655	-46,9542
bleachers	-41,1215	-45,8079	-38,2960	-44,5796	-41,7117	-47,4310
blouses	-41,1516	-45,9152	-37,5090	-45,0022	-44,0472	-47,3716
blow	-41,9795	-45,9996	-37,7830	-46,1516	-47,6184	-47,2173
bonfire	**-38,7524**	-46,7906	-37,6523	-46,3152	-47,0698	-46,3115
business mergers	-40,4462	-45,1391	-37,7449	-44,2740	-42,7629	-46,1589
careful	-42,3325	-44,9914	-37,5719	-46,3148	-47,0654	-47,2617
dishes	-40,5848	-45,5776	-38,0909	-44,6048	**-41,3678**	-47,7115
garage	-42,6211	-46,5834	-37,1655	-46,1872	-43,9560	-47,5439
garbage	-42,0401	-45,8821	-37,4544	-45,1758	-44,1481	-47,0308
gas	-42,4110	-47,3334	-37,5148	-46,2431	-44,8166	-47,7157
marvelously	-40,3889	-45,3332	-37,6693	-45,5126	-45,4029	-45,0539
overalls	-41,3517	-45,1689	-37,4631	-44,7421	-45,2102	-45,9646
peanut oil	-40,2803	**-42,5405**	-37,9554	-43,8525	-45,4162	-45,7917
people	-42,3385	-44,2311	-37,8776	-44,1728	-45,4021	-47,1655
perfume	-41,5544	-44,6769	-37,8567	-44,6911	-46,1290	-46,7834
poor	-42,1557	-46,5529	-38,1966	-46,1582	-47,4738	-47,2597
popularity	-41,4831	-44,7714	-37,8132	-44,5654	-45,3914	-46,8435
potatoes	-41,6339	-45,5781	-38,2670	**-41,4276**	-43,4836	-46,6540
pretty in autumn	-41,3526	-44,6056	-38,0885	-43,9984	-45,2457	-45,7832
promote birth control	-40,9497	-45,3402	-38,3023	-44,5082	-46,8286	**-43,8891**
real people	-42,6481	-43,9553	-37,8711	-44,3047	-45,0266	-46,2625
study	-42,0111	-45,7649	-37,8444	-45,1332	-44,7512	-47,5607
sugar	-42,1797	-46,6797	**-37,0718**	-46,4925	-44,9758	-47,2855

Tabulka B.4: Výsledky rozpoznávání pro slovník *digits* s nahrávkami z vlastní databáze, mluvčí K1, vyhodnocení pro všechny modely

model / nahrávka	one	two	five	seven	nine	zero
one	**-25,7884**	-23,9211	-30,5827	-24,3794	-24,5289	-30,7841
two	-26,6794	**-23,0838**	-31,2079	-25,0916	-25,2092	-30,2588
three	-26,6049	-23,6402	-30,9045	-25,2026	-24,3962	-30,9658
four	-26,2351	-25,1871	-30,6270	-24,7460	-24,9380	-31,2339
five	-26,4915	-24,9401	-30,4957	-24,9433	-24,8356	-31,1249
six	-27,5650	-23,4732	-31,5814	-24,3891	-25,6000	-30,9649
seven	-26,2191	-23,6399	-30,5538	**-23,9018**	-24,5993	-30,5050
eight	-27,4469	-23,8822	-31,4490	-25,0605	-25,1551	-31,2238
nine	-25,9013	-23,8514	***-30,1539***	-24,4417	**-23,9913**	-30,5577
zero	-26,6800	-23,4860	-30,9769	-24,4348	-25,0786	**-29,8856**

Tabulka B.5: Výsledky rozpoznávání pro slovník *digits* s nahrávkami z vlastní databáze, mluvčí J1, vyhodnocení pro všechny modely

model / nahrávka	one	two	five	seven	nine	zero
one	**-25,7884**	-23,9211	-30,5827	-24,3794	-24,5289	-30,7841
two	-26,6794	**-23,0838**	-31,2079	-25,0916	-25,2092	-30,2588
three	-26,6049	-23,6402	-30,9045	-25,2026	-24,3962	-30,9658
four	-26,2351	-25,1871	-30,6270	-24,7460	-24,9380	-31,2339
five	-26,4915	-24,9401	-30,4957	-24,9433	-24,8356	-31,1249
six	-27,5650	-23,4732	-31,5814	-24,3891	-25,6000	-30,9649
seven	-26,2191	-23,6399	-30,5538	**-23,9018**	-24,5993	-30,5050
eight	-27,4469	-23,8822	-31,4490	-25,0605	-25,1551	-31,2238
nine	-25,9013	-23,8514	***-30,1539***	-24,4417	**-23,9913**	-30,5577
zero	-26,6800	-23,4860	-30,9769	-24,4348	-25,0786	**-29,8856**

Tabulka B.6: Výsledky rozpoznávání pro slovník *cislovky* s nahrávkami z databáze SPEECON, všechny modely a signály

model / nahrávka	nula	jedna	dva	tri	ctyri	pet	sest	sedm	osm	devet
nula	**-35,6468**	-40,0210	-35,7165	-44,6011	-41,3047	-31,8498	-37,0650	-36,8973	-39,0226	-36,6444
jedna	-37,5799	**-37,1355**	-36,1877	-43,8123	-38,8927	-29,4957	-36,0570	-35,5042	-40,8338	-36,5360
dva	-36,9249	-39,0208	**-35,6863**	-46,2862	-41,5592	-33,0019	-37,3342	-37,0835	-41,7575	-36,6559
tri	-39,1970	-43,1102	-45,7339	**-37,5374**	-37,4756	-30,7179	-34,7289	-35,8352	-38,4808	-37,2385
ctyri	-39,2850	-43,1106	-45,7552	-37,8162	**-34,6624**	-30,7639	-35,0637	-35,7319	-38,6878	-37,3593
pet	-38,0749	-41,0407	-42,3564	-43,5337	-38,3052	**-28,5476**	-35,4022	-35,6938	-40,5619	-35,4169
sest	-39,1806	-42,9870	-43,6504	-42,4355	-38,9403	-31,1440	**-32,3088**	-35,6902	-39,0412	-36,7635
sedm	-37,1581	-39,3302	-40,2775	-42,4557	-39,1417	-31,7355	-35,0652	**-33,8497**	-39,4493	-37,2370
osm	-37,6000	-41,4483	-40,0288	-43,8477	-41,2418	-32,2164	-36,4210	-35,2073	**-36,2872**	-38,9137
devet	-37,3810	-40,0450	-39,4397	-44,0067	-38,6439	-29,0278	-35,9191	-35,5394	-39,9574	**-33,5529**

Bibliografie

[1] C. Gartenberg, "You can now control your Hyundai through Google Home", *The Verge*, led. 2017, www: https://www.theverge.com/circuitbreaker/2017/1/3/14149836/hyundai-google-home-remote-car-start-blue-link-ces-2017.

[2] E. de Vries, "In-Car Speech Recognition: The Past, Present, and Future", *Globalme*, čvc 2019, www: https://www.globalme.net/blog/the-present-and-future-of-in-car-speech-recognition.

[3] CGI IT Czech Republic s.r.o., "Na českých infolinkách uslyšíte stále častěji roboty", *Euro24*, břez. 2020, www: https://www.euro.cz/byznys/na-ceskych-infolinkach-uslysite-stale-casteji-roboty.

[4] *Waverly Labs: No More Language Barriers*, www: https://www.waverlylabs.com/, 2019.

[5] Łukasz Brocki, D. Koržinek a K. Marasek, "Voice Portal for Public City Transportation", *Interfejs użytkownika - Kansei w praktyce*, led. 2009.

[6] *RBC first bank in Canada to enable bill payments using Siri*, www: http://www.rbc.com/newsroom/news/2017/20170801-siri.html, srp. 2017.

[7] "USAA Launches Pilot of New Skill for Amazon Alexa", *USAA Community*, čvc 2017, www: http://communities.usaa.com/t5/Press-Releases/USAA-Launches-Pilot-of-New-Skill-for-Amazon-Alexa-USAALabs-com/ba-p/132810.

[8] S. Halzack, *Listen Up, Retailers: Don't Ignore Voice Shopping.*

[9] Open Data Science, *Deep Learning for Speech Recognition*, www: https://medium.com/@ODSC/deep-learning-for-speech-recognition-cbbebab15f0d", srp. 2019.

[10] G. Fant, *Acoustic Theory of Speech Production: With Calculations Based on X-ray Studies of Russian Articulations.* Hague, The Netherlands: Mouton: Mouton de Gruyter, 1971, sv. 2, ISBN: 3111869458.

[11] Y. L. Tian a J. Jing-cheng, "Applying source-filter model in Chinese speech synthesis", 2002.

[12] G. Kouroupetroglou a G. Chrysochoidis, "Formant Tuning in Byzantine Chanting", čvc 2014.

[13] *A Free Codec For Free Speech*, www: https://speex.org/.

[14] G. Gandhimathi a S. Jayakumar, "An analysis on source-filter model based artificial banwidth extension system", 2014.

[15] J. Rajnoha, "Rozpoznávání řeči v reálných podmínkách na platformě standardního PC", dipl, ČVUT Praha, Fakulta elektrotechnická, Technnická 2, Praha, led. 2006.

[16] R. Skarnitzl, P. Šturm a J. Volín, *Zvuková báze řečové komunikace - Fonetický a fonologický popis řeči*, 1. vyd. Karolinum, 2016, ISBN: 9788024632728.

[17] J. Psutka, L. Müller, J. Matoušek a V. Radová, *Mluvíme s počítačem česky.* Prague: Academia, 2006, s. 752, ISBN: 80-200-1309-1.

[18] A. Garg a P. Sharma, "Survey on acoustic modeling and feature extraction for speech recognition", in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2016, s. 2291–2295.

[19] P. Lockwood a J. Boudy, "Experiments with a non-linear spectral subtractor (NSS), hidden Markov models and the projection, for robust speech recognition in cars", in *Second European Conference on Speech Communication and Technology*, 1991.

[20] K.-F. Lee a R. Reddy, *Automatic Speech Recognition: The Development of the Sphinx Recognition System*. USA: Kluwer Academic Publishers, 1988, ISBN: 0898382963.

[21] L. R. Rabiner a R. W. Schafer, *Introduction to Digital Speech Processing, Foundations and Trends ®in Signal Processing*, 1. now Publishers Inc., 2007, sv. 1.

[22] D. Wang, X. Wang a S. Lv, "An Overview of End-to-End Automatic Speech Recognition", *Symmetry*, roč. 11, s. 1018, srp. 2019. DOI: 10.3390/sym11081018.

[23] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath a B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups", *IEEE Signal Processing Magazine*, roč. 29, č. 6, s. 82–97, lis. 2012, ISSN: 1558-0792. DOI: 10.1109/MSP.2012.2205597.

[24] F. Richardson, D. Reynolds a N. Dehak, "Deep Neural Network Approaches to Speaker and Language Recognition", *IEEE Signal Processing Letters*, roč. 22, č. 10, s. 1671–1675, říj. 2015, ISSN: 1558-2361. DOI: 10.1109/LSP.2015.2420092.

[25] G. Dede a M. H. Sazli, "Speech recognition with artificial neural networks", *Digital Signal Processing*, roč. 20, s. 763–768, květ. 2010. DOI: 10.1016/j.dsp.2009.10.004.

[26] M. Gales, "Discriminative Models for Speech Recognition", ITA Workshop, Cambridge University Engineering Department, ún. 2007.

[27] S. Bhardwaj, S. Pathania a R. Akela, "Baum-Welch training for segment-based speech recognition", in *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)*, sv. 4, květ. 2015, s. 1179–1183.

[28] Y. R. Kumar, A. V. Babu, K. A. N. Kumar a J. S. R. Alex, "Modified Viterbi decoder for HMM based speech recognition system", in *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, čvc 2014, s. 470–474. DOI: 10.1109/ICCICCT.2014.6993008.

[29] H. Shu, I. L. Hetherington a J. Glass, "Baum-Welch training for segment-based speech recognition", in *2003 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE Cat. No.03EX721)*, lis. 2003, s. 43–48. DOI: 10.1109/ASRU.2003.1318401.

[30] L. J. Rodriguez-Fuentes a M. I. Torres, "Comparative Study of the Baum-Welch and Viterbi Training Algorithms Applied to Read and Spontaneous Speech Recognition", in *IbPRIA*, 2003.

[31] A. Krogh, "An introduction to hidden Markov models for biological sequences", English, in *Computational methods in molecular biology*, S. Salzberg, D. Searls a S. Kasif, ed., United Kingdom: Elsevier, 1998, s. 45–63.

[32] R. Durbin, S. R. Eddy, A. Krogh a G. J. Mitchison, "Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids", 1998.

[33] S. Eddy, "What Is a Hidden Markov Model?", *Nature biotechnology*, roč. 22, s. 1315–6, lis. 2004. DOI: 10.1038/nbt1004-1315.

[34] F. J. Och a H. Ney, "Improved Statistical Alignment Models", in *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ř. ACL '00, Hong Kong: Association for Computational Linguistics, 2000, 440–447. DOI: 10.3115/1075218.1075274.

[35] P. J. Green a S. Richardson, "Hidden Markov Models and Disease Mapping", *Journal of the American Statistical Association*, roč. 97, č. 460, s. 1055–1070, 2002. DOI: 10.1198/016214502388618870.

[36] Jia Li, R. M. Gray a R. A. Olshen, "Multiresolution image classification by hierarchical modeling with two-dimensional hidden Markov models", *IEEE Transactions on Information Theory*, roč. 46, č. 5, s. 1826–1841, 2000.

[37] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett a N. L. Dahlgren, *DARPA TIMIT Acoustic Phonetic Continuous Speech Corpus CDROM*, 1993.

[38] B. A. Hanson a T. H. Applebaum, "Robust speaker-independent word recognition using static, dynamic and acceleration features: experiments with Lombard and noisy speech", *International Conference on Acoustics, Speech, and Signal Processing*, 857–860 vol.2, 1990.

[39] S. Furui, "Speaker-independent isolated word recognition using dynamic features of speech spectrum", *IEEE Trans. Acoustics, Speech, and Signal Processing*, roč. 34, s. 52–59, 1986.

[40] C. Lopes a F. Perdigao, "Phoneme Recognition on the TIMIT Database", in *Speech Technologies*, I. Ipsic, ed., Rijeka: IntechOpen, 2011, kap. 14. DOI: 10.5772/17600.

[41] K.-F. Lee a H.-W. Hon, "Speaker-independent phone recognition using hidden Markov models", *IEEE Trans. Acoustics, Speech, and Signal Processing*, roč. 37, s. 1641–1648, 1988.

[42] E. Rodríguez, B. Ruíz, Á. García-Crespo a F. García, "Speech/speaker recognition using a HMM/GMM hybrid model", in *Audio- and Video-based Biometric Person Authentication*, J. Bigün, G. Chollet a G. Borgefors, ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, s. 227–234, ISBN: 978-3-540-68425-1.

[43] M. M. Goodarzi a F. Almasganj, "A GMM/HMM Model for Reconstruction of Missing Speech Spectral Components for Continuous Speech Recognition", *Int. J. Speech Technol.*, roč. 19, č. 4, 769–777, pros. 2016, ISSN: 1381-2416. DOI: 10.1007/s10772-016-9369-x.

[44] P. Bansal, A. Kant, S. Kumar, A. Sharda a S. Gupta, "Improved hybrid model of HMM/GMM for speech recognition", *Intell. Technol. Appl*, led. 2008.

[45] J. Uhlíř, P. Sovka, P. Pollák, V. Hanžl a R. Čmejla, *Technologie hlasových komunikací*, 1. vyd. Nakladatelství ČVUT, čvc 2007, ISBN: 978-80-01-03888-8.