

TECHNICKÁ UNIVERZITA V LIBERCI

FAKULTA MECHATRONIKY A MEZIOBOROVÝCH INŽENÝRSKÝCH STUDIÍ

DIPLOMOVÁ PRÁCE

Rozpoznávání řečníka na základě
hlasových charakteristik

Liberec 2002

Petr David

TECHNICKÁ UNIVERZITA V LIBERCI

FAKULTA MECHATRONIKY A MEZIOBOROVÝCH INŽENÝRSKÝCH STUDIÍ

Studijní program: 2612T-Elektrotechnika a informatika

Studijní obor: Automatické řízení a inženýrská informatika

Rozpoznávání řečníka na základě hlasových charakteristik

(Speaker recognition based on voice parameters)

Petr David

Vedoucí diplomové práce: Prof. Ing. Jan Nouza, CSc.

Technická Univerzita v Liberci

Rozsah práce a příloh

Počet stran: 67

Počet obrázků: 58

Počet tabulek: 2

Počet vzorců: 27

Počet příloh: 12

Datum: 22.5.2002

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií

Katedra elektroniky a zpracování signálů

Školní rok: 2001/2002

ZADÁNÍ DIPLOMOVÉ PRÁCE

pro: **Petra Davida**

studijní program: **2612T – Elektrotechnika a informatika**

obor: **Automatické řízení a inženýrská informatika**

Vedoucí katedry Vám ve smyslu zákona o vysokých školách č.111/1998 Sb. určuje tuto diplomovou práci:

Název tématu:

Rozpoznávání řečníka na základě hlasových charakteristik

Zásady pro vypracování:

1. Seznamte se se současným stavem v oboru rozpoznávání řečníka (Speaker Recognition).
2. Navrhněte systém pro textově nezávislé rozpoznávání řečníka založený na keprstrální reprezentaci signálu a na metodách VQ (vektorová kvantizace) a GMM (Gaussian mixture model).
3. Vytvořte rozsáhlou databázi pro ověřování výše uvedeného systému. Při ověřování se zaměřte zejména na úspěšnost správného rozpoznání mluvčího a také na možnosti verifikace mluvčího.
4. Hledejte optimální nastavení systému z hlediska řečových příznaků, velikosti kódové knihy a počtu mixtur GMM.
5. Pokuste se vyhodnotit funkčnost navržených metod i z hlediska robustnosti, zejména vzhledem k různým parametrům přenosové cesty (různý mikrofón, vliv zesílení a šumu) a také s ohledem na časový odstup mezi trénovacími a testovacími nahrávkami.

Rozsah grafických prací: dle potřeby dokumentace

Rozsah průvodní zprávy: cca 40 až 50 stran

Seznam odborné literatury:

[1] sborník konference Speaker Odyssey, Chania 2001

[2] sborníky konferencí Eurospeech, ICSLP.

[3] časopis IEEE Transactions on Audio a Speech Processing

Vedoucí diplomové práce: Prof. Ing. Jan Nouza, CSc.

Konzultant:

Zadání diplomové práce: 25.10.2001

Termín odevzdání diplomové práce: 25. 5. 2002



Nouza

.....
Vedoucí katedry

[Signature]

.....
Děkan

V Liberci dne 25.10.2001

Anotace:**Rozpoznávání řečníka na základě hlasových charakteristik**

Cílem diplomové práce je navrhnout a realizovat systém pro automatickou identifikaci osob (mluvčích), ve kterém je informace o identitě osoby získána z jejích hlasových charakteristik. Práce nejprve v jednotlivých kapitolách popisuje potřebnou teorii, vysvětluje některé principy a zároveň diskutuje praktické otázky týkající se návrhu a architektury rozpoznávacího systému.

Systém uvedený v této práci je založen jak na deterministickém, tak i na stochastickém přístupu, kde se při hledání identity mluvčího používají matice příznaků reprezentující jednotlivé promluvy. Konfigurace vektorů příznaků se v jednotlivých experimentech lišila, ale v principu bylo použito buď 8 statických MFCC kepstrálních příznaků, 8 dynamických příznaků prvního řádu (delta příznaky), nebo 8 dynamických příznaků druhého řádu (delta-delta příznaky), případně jejich kombinace. Takto zparametrizovaný signál následně vstupuje do klasifikátoru, kde je vypočtena výsledná vzdálenost (pravděpodobnost), která charakterizuje podobnost mezi testovanou promluvou a modely jednotlivých mluvčích. V této práci jsou prezentovány klasifikátory dva. První pracuje na bázi vektorové kvantizace (VQ) a druhý je založen na technice gaussovských mixturových modelů (GMM). Oba klasifikátory - a tím i systémy postavené na těchto klasifikátorech - jsou textově nezávislé.

Aby byla ověřena funkčnost zde prezentovaného systému, bylo zapotřebí otestovat jej na relativně malé (ale již ne triviální) databázi mluvčích. Tato databáze byla tvořena 29 muži a 26 ženami. Každý z mluvčích pronesl 74 vět, z nichž prvních 25 bylo použito při samotném testování (jako vstupní promluvy neznámých mluvčích) a zbylých 49 bylo využito pro tvorbu modelů. V konečném důsledku to tedy znamená, že jednotlivé konfigurace systému (počet a typ příznaků, množství elementů modelů...) byly testovány 1375 promluvami (1 nesprávně rozpoznaná promluva = chyba 0,0727%) proti databázi mluvčích čítající 55 osob.

Výsledná maximální procentuální úspěšnost, kterou se v rozpoznávacích testech podařilo dosáhnout, byla:

klasifikátor založený na technice VQ	99,49 %
při konfiguraci: 49 trénovacích promluv, model čítající 256 kódových knih, { 8 MFCC, 8 delta } příznaků	
klasifikátor založený na technice GMM	99,56 %
při konfiguraci: 49 trénovacích promluv, model čítající 256 mixtur, { 8 MFCC, 8 delta, 8 delta-delta } příznaků	

Abstract:

Speaker recognition based on voice parameters

The aim of the diploma thesis is to build and realize system for automatic speaker recognition, where information about speaker's identity is obtained from individual information included in speech waves. In several chapters the principles and architecture of recognition system is discussed. Furthermore practical experiences acquired during work on this system are mentioned, too.

At the highest level, all speaker recognition systems contain two main modules: feature extraction and pattern-matching. Feature extraction is the process that extracts a small amount of data from the voice signal that can later be used to represent each speaker. Configuration of feature vectors was in various experiments different, but principally was always used either 8 static MFCC cepstral features, 8 first order differential coefficients or 8 second order differential coefficients eventually their combinations. Pattern matching involves the actual procedure to identify the unknown speaker by comparing extracted features from his/her voice input with the models from a set of known speakers. In this thesis there are presented two pattern-matching classifiers: first classifier is based on vector quantization (VQ) and the second one is using gaussian mixture models (GMM). Both classifiers and thereby the system based on these classifiers are text independent.

To check the validity of the system that is presented here, there was built speaker database (relatively small, but not trivial). There were 26 female and 29 male speakers in this database. Each speaker uttered 74 sentences, 25 testing (simulation of unknown speakers) and all remaining sentences were used for training models. Consequently particular experiments with different configurations of the system (number and configuration of features, number of components in models...) were tested by 1375 trials (one incorrectly recognized sentence = error 0,0727 %).

Resultant maximum percentage fruitfulness, which was achieved during recognition experiments:

classifier based on VQ technique	99,49 %
with configuration: 49 training utterances, model with 256 codewords, {8 MFCC, 8 delta } features	
classifier based on GMM technique	99,56 %
with configuration: 49 training utterances, model with 256 mixtures, {8 MFCC, 8 delta, 8 delta-delta} features	

Poděkování:

Na tomto místě bych chtěl poděkovat vedoucímu své diplomové práce, Prof. Ing. Janu Nouzovi, CSc., za odborné vedení, cenné rady a podnětné připomínky, které přispěly k řešení daného problému.

Prohlášení:

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č.121/2000 o právu autorském, zejména §60 (školní dílo).

Beru na vědomí, že TU v Liberci má právo na uzavření licenční smlouvy o užití mé DP a prohlašuji, že **souhlasím** s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TU v Liberci, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce.

V Liberci dne 22.5.2002



Petr David

OBSAH

SEZNAM POUŽITÝCH SYMBOLŮ	10
---------------------------------------	-----------

1. ÚVOD

1.1. Identifikace a verifikace lidského hlasu	11
1.2. Systémy pro rozpoznávání mluvěcího	12
1.3. Faktory ovlivňující rozpoznávání.....	15
1.4. Základní části systému pro rozpoznávání mluvěcího	16

2. PŘÍSTUPY A METODY PRO ROZPOZNÁVÁNÍ MLUVČÍHO

2.1. Metody pro textově závislé rozpoznávání mluvěcího.....	17
2.1.1. Metoda DTW	17
2.1.2. Metoda HMM	20
2.2. Metody pro textově nezávislé rozpoznávání mluvěcího	22
2.2.1. Metoda založená na vektorovém kvantování	22
2.2.2. Metoda založená na gaussovských modelech.....	24
2.2.3. Metoda nejbližšího souseda.....	26
2.2.4. Umělé neuronové sítě	27

3. REALIZACE METOD PRO IDENTIFIKACI MLUVČÍHO

3.1. Parametrizace řečového signálu	32
3.1.1. Digitalizace řečového signálu.....	33
3.1.2. Segmentace signálu	34
3.1.3. Preemfáze	35
3.1.4. Aplikace okénka	35
3.1.5. Výpočet statických příznaků.....	36
3.1.6. Výpočet dynamických příznaků	39
3.2. Návrh a realizace systému založeného na metodě VQ.....	41
3.2.1. Režim trénování - vytvoření kódové knihy (LBG algoritmus)	41
3.2.2. Režim rozpoznávání	44
3.3. Návrh a realizace systému založeného na metodě GMM	47
3.3.1. Režim trénování - vytvoření modelu mluvěcího (EM algoritmus)	47

3.3.2. Režim rozpoznávání	50
4. EXPERIMENTY S ROZPOZNÁVÁNÍM MLUVČÍCH	
4.1. Databáze promluv	52
4.2. Experimenty s implementovanými metodami.....	53
4.2.1. Základní experiment - srovnání výkonnosti VQ a GMM.....	53
4.2.2. Míra ovlivnění úspěšnosti identifikace trénovacími daty	55
4.2.3. Vypovídací schopnost keprálních příznaků	56
4.2.4. Úspěšnost rozpoznávání v závislosti na způsobu výpočtu delta příznaků	57
4.2.5. Prospěšnost EM algoritmu v GMM systémech	57
4.2.6. Robustnost aplikovaných algoritmů	57
5. VERIFIKACE MLUVČÍHO	
5.1. Jednoduchý systémem pro VM	59
5.2. GMM jako základ systému pro textově nezávislou VM.....	60
6. ZÁVĚR.....	64
LITERATURA	67
PŘÍLOHY	68

SEZNAM POUŽITÝCH ZKRATEK

<u>zkratka</u>	<u>název</u>	<u>anglický název</u>
VQ	vektorová kvantizace	vector quantization
GMM	gaussovské mixturové modely	gaussian mixture models
DTW	dynamické borcení času	dynamic time warping
HMM	skryté markovské modely	hidden markov models
$\bar{\mathbf{x}}$	vektor střední hodnoty	mean vector
Σ	kovarianční matice	covariance matrix
NN	nejbližší soused	nearest neighbor
PDF	hustota pravděpodobnosti	probability density function
ANN	umělé neuronové sítě	artificial neural networks
BPNN	neuronová síť se zpětným šířením chyb	backpropagational neural network
FFT	rychlá Fourierova transformace	fast Fourier transformation
MFCC	melovské frekvenční keprstrální koeficienty	mel-frequency cepstral coefficient
DFT	diskrétní Fourierova transformace	discrete Fourier transformation
IDFT	inverzní diskrétní Fourierova transformace	inverse discrete Fourier transformation
F_s	vzorkovací frekvence	sample frequency
DCT	diskrétní kosinová transformace	discrete cosine transformation
KK	kódová kniha	code book
LBG	algoritmus k tvorbě KK	Linde, Buzo and Gray algorithm
EM	algoritmus předpoklad-maximalizace	expectation-maximization algorit.
PDF	pravděpodobnostní hustota rozdělení	probability density function
LR	hodnota věrohodnosti	likelihood ratio
UBM	univerzální model mluvčích	universal background model

1. ÚVOD

Nebudeme si nic zastírat, byla to především potřeba možné identifikace osob z důvodu jejich napojení na kriminální činnost, jež vedla k prudkému rozvoji vědní disciplíny s názvem *biometrika*. V roce 1880 přišel antropolog Alfons Bertillion se svou „bláznivou“ metodou na identifikaci usvědčených zločinců měřením určitých tělesných partií. Ačkoliv již brzy byla „Bertillonáž“ nahrazena mnohem kvalitnější metodou identifikace osob (identifikace podle otisků prstů - Richard Edward Henry), Alfons Bertillion předznamenal prudký vývoj v této oblasti.

Biometrika ve své podstatě znamená „*měření biologických faktorů člověka*“. Věda byla řadu let velmi zaneprázdněna výzkumem prostředků pro identifikaci osob pomocí měření a záznamu lidských fyzických charakteristik. Důmyslné senzory jsou v dnešní době schopné identifikovat člověka podle tvaru ruky, chodidla nebo hlavy. Jsou schopny změřit a identifikovat lidské oko a topografii otisků vašich prstů. Je zde rovněž typ vaší krve a váš jedinečný kód DNA. Taktéž byly vynalezeny přístroje, které vás mohou identifikovat podle zabarvení vašeho hlasu.

1.1. Identifikace a verifikace lidského hlasu

Identifikace a verifikace lidského hlasu jsou definovány jako elektronické metody pozitivní identifikace osoby pomocí rozšířené analýzy digitálního „otisku hlasu“. Tvar hlasivek, ústní dutiny, jazyka a zubů způsobují, že rezonance vokálního traktu je u různých osob dostatečně odlišná. Abychom správně rozeznali konkrétního mluvčího od ostatních, můžeme také využít různé individuální návyky, které si člověk osvojí během svého života. Metoda ověřování hlasu je také známa pod jinými názvy, například jako autentizace pomocí hlasu, ověřování identity mluvčího či otisk hlasu.

Jednou z nejúspěšnějších technik pro identifikaci mluvčího je porovnávání vzorků pomocí analýzy řečového signálu. Dynamické, stejně jako okamžité spektrální znaky zřejmě hrají významnou roli ve vnímání řeči. Některé ověřovací technologie zakládají své autentizační rozhodnutí na analýze vět. Věta má v sobě více akustické informace než jednoduché slovo; více informace umožňuje vyšší kvalitu srovnávacího procesu pro absolutní shodu. Slova bývají krátká a neobsahují dostatečnou akustickou informaci, která by spolehlivě odlišila mluvčího. Věty zná pouze autentický mluvčí a mohou jimi být i množiny slov, které je mluvčí schopen vyslovit opakovaně test za testem. Uživatelé si často vytvářejí své vlastní tajné autentizační věty, čímž je bezpečnost systému částečně rozšířena, protože neoprávnění uživatelé (narušitelé) neví, kterou větu použít, natož jakým hlasem ji vyslovit. Testy ukazují, že narušitel, který nezná příslušnou autentizační větu autorizovaného uživatele, je systémem odmítnut ve více než 99 % případů.

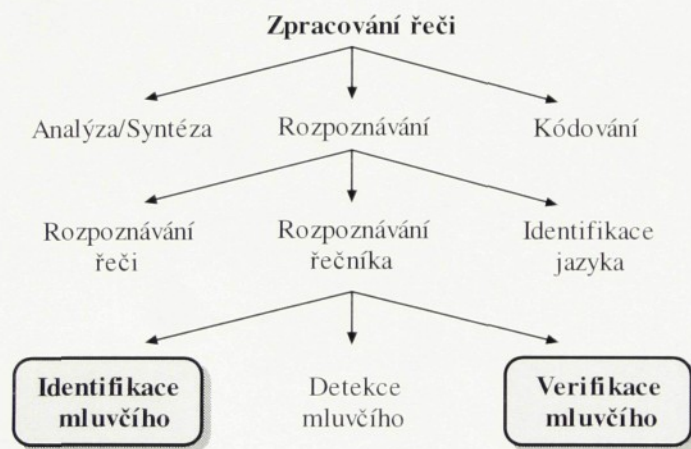
Ačkoliv výzkum v této oblasti začal již v 70. letech tohoto století, komerční využití rozpoznávání lidského hlasu jako biometrické metody sloužící k jednoznačné identifikaci

osob nastává právě v současné době. Uvědomíme-li si, jak častá je hlasová komunikace v každodenní činnosti každého z nás, je identifikace mluvčího na základě jeho hlasu poměrně zajímavá biometrická technika. Významnost ověřování hlasu mezi biometrickými technikami spočívá také v její sociální přijatelnosti. Dalšími důležitými vlastnostmi jsou pak rychlost, spolehlivost, jednoduchost použití a nízké pořizovací i provozní náklady.

Charakteristickým znakem současných systémů pro ověřování hlasu je, že verifikace může být za určitých okolností (nastudnutí, okolní šum, atd.) mnohem komplikovanější než u jiných biometrických disciplín, což vede k názoru, že verifikace hlasu je v některých případech pro uživatele méně příjemná. Avšak vzhledem k již zmiňovaným kladům této biometrické techniky lze předpokládat její významný rozvoj.

1.2. Systémy pro rozpoznávání mluvčího

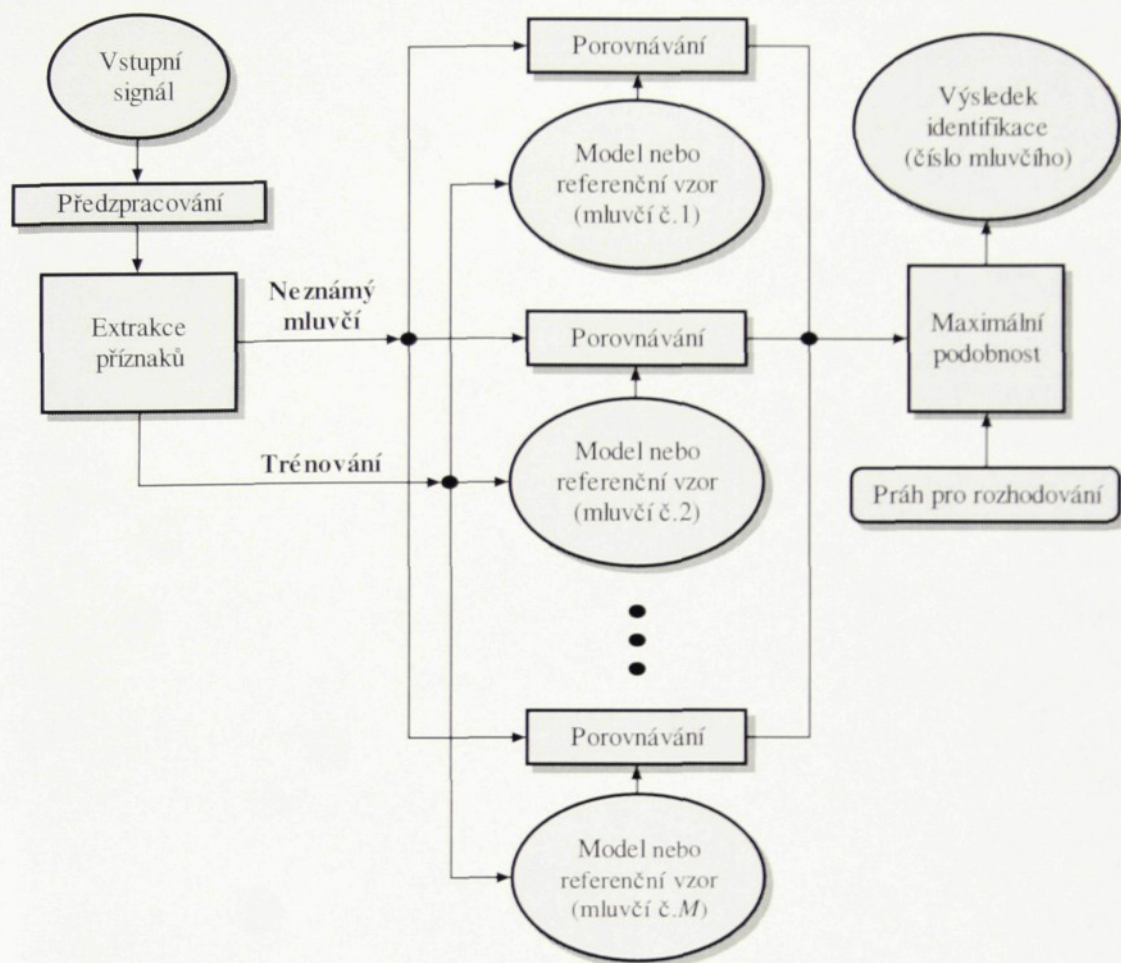
Zpracování řeči má mnoho různých aplikací. Na obr.1.1 jsou zobrazeny některé z nich.



Obr.1.1 Zpracování řeči

(naznačení některých možných aplikací zpracování řeči a zvýraznění dvou oblastí, které jsou popsány v této diplomové práci)

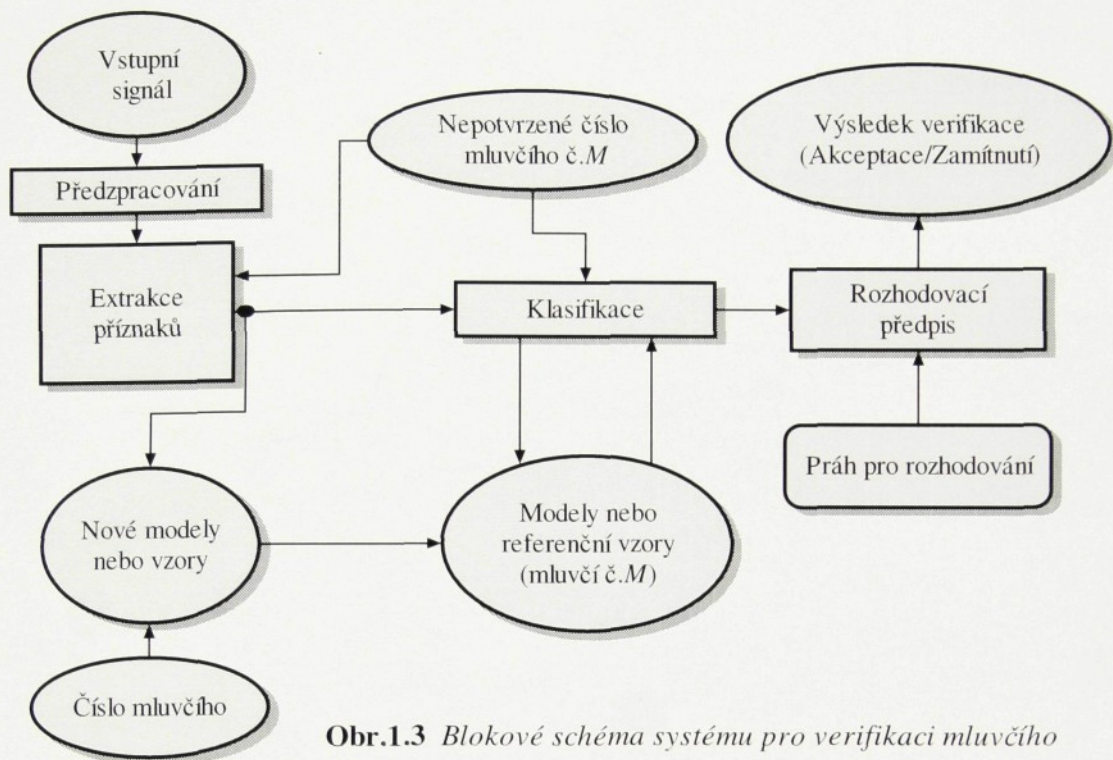
Pomineme-li detekci mluvčího, můžeme rozpoznávání řečníka rozdělit do dvou základních úloh. První úlohou je *Identifikace mluvčího* (IM – speaker identification), což je proces, při kterém není předem známa identita mluvčího a systém musí rozhodnout, kdo daná osoba je, případně do které skupiny mluvčích náleží. Na obr.1.2 můžeme vidět příklad jednoduchého systému pro identifikaci mluvčího s vyznačením základních částí, které by měl takový systém obsahovat. Speciálním případem IM je *identifikace v otevřené množině* (open set identification). Při této identifikaci pro rozpoznávaného mluvčího nemusí existovat referenční model (řečník nenáleží do žádné skupiny). To je většinou případ "soudních" aplikací. Zde by bylo vhodné, aby systém uměl rozpoznat neexistenci reference a vypsál, že testovaná promluva nepřísluší k žádnému referenčnímu modelu. Je tedy třeba nejdříve provést identifikaci a najít pro hlas neznámého člověka nejpravděpodobnější referenční řečník a pak rozhodnout, je-li zvolený referenční řečník skutečným autorem promluvy. Identifikaci v otevřené množině lze tedy chápat jako kombinaci identifikace a verifikace v uzavřené množině. Druhou úlohou je *verifikace mluvčího* (VM – speaker verification). Proces VM je



Obr.1.2 Blokové schéma systému pro identifikaci mluvčího

používaný pro poskytnutí, či zamítnutí požadavků pro přístup k určitým zdrojům na základě mluveného slova. Většina aplikací, ve kterých je použit hlas jako klíč pro ověření totožnosti mluvčího, je klasifikována jako VM. Na rozdíl od IM je v případě VM identita mluvčího známa (na základě jeho tvrzení, případně nějakého kódu atd.) a systém musí ověřit, zda mluvčí je skutečně tím, kým tvrdí, že je. Jednoduché blokové schéma systému pro VM je zobrazeno na obr.1.3. Jako kritický problém, který se při aplikaci VM může vyskytnout, můžeme brát pokus neoprávněné osoby o neautorizovaný vstup (např. přehrání pásky s hlasem registrovaného mluvčího říkajícího klíčová slova nebo věty). K překlenutí tohoto problému existují řešení, která pro každého mluvčího generují náhodnou sadu klíčových slov (mohou to být například číslice jdoucí v určitém pořadí za sebou).

Metody rozpoznávání řečníka lze dále rozdělit do dvou skupin podle toho, jaké typy promluv se při rozpoznávání používají. V případě rozpoznávání řečníka z daného textu jde o *textově závislé rozpoznávání* (text dependent recognition). Zde je vyžadováno, aby neznámý řečník vyslovil předem definovaný text. Je nutné, aby tento text byl identický při trénování i při rozpoznávání. Mohou to být například stejná klíčová slova, jejich různé kombinace a nebo



Obr.1.3 Blokové schéma systému pro verifikaci mluvčího

tajný kód příslušející mluvčímu. Tento kód může být kupříkladu sestaven z číslic oddělených pauzami. Při tomto způsobu rozpoznávání mluvčího se samozřejmě předpokládá, že řečník si přeje být rozpoznán a je ochotný vyžadovanou promluvu vyslovit. Naproti tomu u rozpoznávání řečníka z libovolného textu (*textově nezávislé rozpoznávání* – text independent recognition) není na text kladeno naprosto žádné omezení a neznámý řečník může vyslovit v podstatě to, co ho právě napadne. V tomto případě je rozhodovací proces postaven pouze na charakteristikách řečového traktu mluvčího a na charakteristikách buzení hlasu. Tento způsob rozpoznávání nachází využití hlavně v soudních vědách a kriminalistice (souhrnně *forensní aplikace*), kdy si řečník většinou nepřeje být rozpoznán, a proto není ochoten vyslovit požadovanou promluvu tak, jak je to vyžadováno u textově závislých systémů. Další možné využití této metody je v případě, že rozpoznání má proběhnout bez vědomí mluvčího.

Různé oblasti použití jednotlivých úloh rozpoznávání řečníka jsou uvedeny v tab.1.1.

	IDENTIFIKACE		VERIFIKACE	
	v uzavřené množině	v otevřené množině	nezávislá na textu	závislá na textu
BEZPEČNOSTNÍ SYSTÉMY				ANO
				ANO
			ANO	ANO
SODNÍ VĚDY	ANO	ANO	ANO	ANO
KRIMINALISTIKA	ANO	ANO	ANO	ANO

Tab.1.1 Oblasti použití úloh rozpoznávání řečníka

V bezpečnostních systémech, které zajišťují např. fyzický vstup osob do objektů (hlasové zámky), přístup k databázím s tajnými nebo důvěrnými informacemi nebo provádění bankovních operací po telefonu, se obvykle využívá VM. U těchto systémů se předpokládá, že řečník si přeje být poznán, a proto je ochoten prokázat svou totožnost. V soudních vědách a kriminalistice může najít své uplatnění jak verifikace, tak i identifikace řečníka. Verifikaci lze využít např. tehdy, existuje-li záznam hlasu pachatele z místa činu a je-li třeba ověřit, zda hlas na záznamu je shodný s hlasem určité podezřelé osoby. Identifikaci je pak možné využít v případech, že podezřelých je více a je přitom třeba určit konkrétního pachatele.

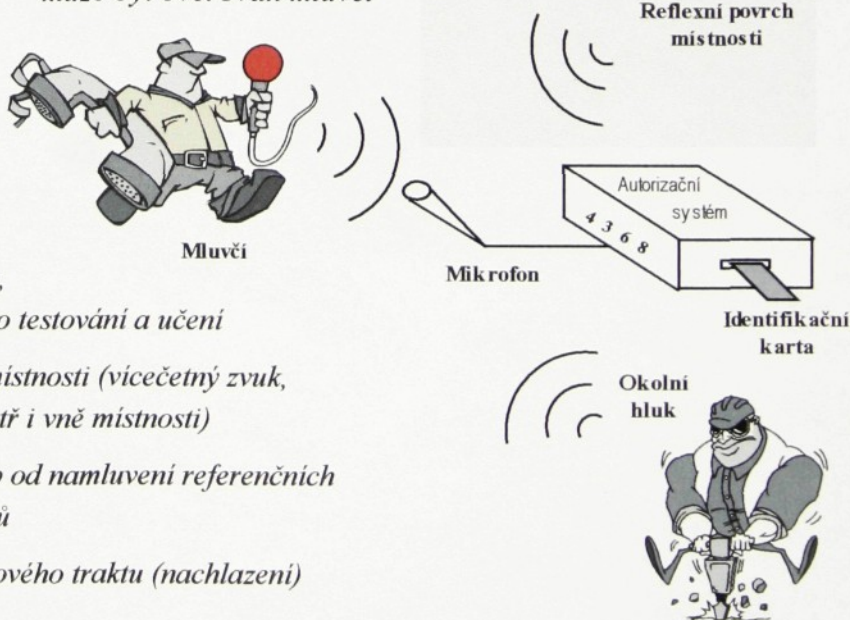
Fonoskopické expertizy (audioexpertizy) poskytované soudním znalcem jsou příkladem praktického použití metod pro rozpoznávání řečníka v soudní praxi. Tyto expertizy mohou být použity k identifikaci „anonymního“ mluvčího na základě porovnání se záznamem mluvčího, který může přicházet v úvahu jako pachatel anonymního záznamu. Identifikace lze provádět ze (v rámci možností co nejkvalitnějšího) záznamu telefonního hovoru, z magnetofonové či jiné přímé nahrávky hlasu „anonymní“ osoby, anebo typováním neznámého mluvčího ze zvukového záznamu. Z dostatečně dlouhé nahrávky je možné s větší či menší přesností určit pohlaví mluvčího, věk, regionální příslušnost (eventuelně mateřský jazyk, hovoří-li mluvčí se zřetelně „cizím akcentem“), vzdělání, popř. sociální zázemí [1]. Podrobnější popis úloh rozpoznávání řečníka a metod jejich řešení lze nalézt v [2].

1.3. Faktory ovlivňující rozpoznávání

Při provozu systému pro IM nebo VM je nutné vypořádat se s faktory, které jsou mimo působnost vnitřních algoritmů programu a mohou v konečném důsledku způsobit chyby při rozpoznávání. Těmito faktory mohou být, jak je zobrazeno i na obr.1.4:

- *nesprávně vyslovené či přečtené slovo*
- *extrémní emoční stres*
- *různé umístění mikrofonu, různé mikrofony pro testování a učení*
- *špatná akustika v místnosti (vícečetný zvuk, hluk vznikající uvnitř i vně místnosti)*
- *velký časový odstup od namluvení referenčních a testovacích vzorků*
- *dočasné změny řečového traktu (nachlazení)*

Obr.1.4 Příklad prostředí, ve kterém může být ověřován mluvčí



Pokud bychom chtěli zkonstruovat robustní systém, je nutné vzít v úvahu všechny tyto faktory, protože právě ty přímo ovlivňují výkonnost rozpoznávacího algoritmu.

1.4. Základní části systému pro rozpoznávání mluvěcího

Jak vyplývá ze schémat na obr.1.2 a obr.1.3, systém pro IM nebo VM by měl obsahovat níže uvedené části. Prvním úkolem je převod akustické vlny na signál zpracovatelný v počítači. O to se stará blok pro *digitalizaci řečového signálu*. Velkou výhodou úlohy rozpoznávání mluvěcího na počítači je její finanční nenáročnost. Pro nahrání řečového signálu do počítače je potřeba pouze mikrofón a zvuková karta, což je v současné době standardní výbava každého osobního počítače. Potřebný software pro nahrávání je dodáván s operačním systémem nebo také se zvukovou kartou. O náležité převedení analogového signálu na digitální se postará hardware zvukové karty. Dalším krokem je *extrakce příznaků*, jenž spočívá v rozčlenění signálu na intervaly o délce 10-30ms. Každý tento interval je reprezentován vícerozměrným polem příznaků.

Pokud již máme signál vyjádřený pomocí vhodných příznaků, můžeme přejít k samotnému hledání správné referenční promluvy, která je také vyjádřena příznaky. Tato fáze se nazývá *rozpoznávání (klasifikace)*. Pro jednoduchost zde budeme uvažovat systém pracující s referenčními vzory slov. U těchto systémů je sekvence příznakových vektorů porovnávána s referenčními modely (ty musí být reprezentovány stejnými příznaky). K porovnání použijeme metody pro rozpoznávání obrazů. Klasifikace pro danou promluvu a referenci vyústí ve *výslednou kumulovanou vzdálenost* (suma vzdáleností mezi jednotlivými vektory příznaků testované a referenční promluvy). Tato vzdálenost nám pak poskytuje informaci o podobnosti testované a referenční promluvy. Protože je nutné do systému průběžně ukládat nové referenční promluvy, musíme do našeho systému zařadit smyčku, která umožní *učení a ukládání nových referenčních obrazů*.

Jako poslední přichází na řadu *rozhodovací člen* (v nejjednodušším případě nějaké pravidlo), jímž vyhodnotíme provedená porovnání. U základní úlohy IM (uzavřená sada, textově závislý systém) je to minimální kumulovaná vzdálenost. Pokud bychom pracovali na úloze VM, tak by se jednalo o rozhodnutí, zda mluvěcí je opravdu tím, kým tvrdí, že je. Zde je to již o něco komplikovanější, ale v podstatě se jedná o ověření dvou hypotéz: mluvěcí je skutečně oním člověkem, za kterého se prohlašuje, a nebo jím není.

Snahu o tvorbu systému pro rozpoznávání mluvěcího lze tedy charakterizovat takto: získat rozpoznávací systém vykazující *vysokou rozlišovací schopnost, vysokou adaptabilitu* na různé mluvěcí a *nízkou variabilitu* v rámci jednoho mluvěcího při *příjemných výpočetních nárocích* (pokud možno systém pracující v reálném čase). V neposlední řadě budeme po takovém systému požadovat, aby byl co nejrobustnější (pokud možno co nejméně závislý na rušivých vlivech okolí).

2. PŘÍSTUPY A METODY PRO ROZPOZNÁVÁNÍ MLUVČÍHO

Pro oblast rozpoznávání mluvěcích můžeme v principu použít všechny metody navržené pro rozpoznávání řeči s tím, že se v nich budeme snažit o zvýraznění těch rysů, které od sebe odliší jednotlivé mluvěčí. Jde tedy o pravý opak cíle, kterého chceme dosáhnout při návrhu systému rozpoznávání řeči, u něhož se většinou vyžaduje, aby byl nezávislý na hovořící osobě. U rozpoznávání mluvěcího je kladen hlavní důraz na zachycení a modelování řečových charakteristik jednotlivých řečníků, naopak potlačeny mohou být (a v některých případech dokonce musí - viz. textově nezávislé rozpoznávání) informace o vlastním obsahu sdělení i o jeho časovém průběhu. V následujícím přehledu si ukážeme možnosti využití základních technik známých z oblasti rozpoznávání řeči a představíme si jejich modifikace pro úlohu rozpoznávání mluvěcího. V [3] se lze dozvědět, jak probíhal rozvoj automatických metod rozpoznávání řečníka od úplných prvopočátků.

2.1. Metody pro textově závislé rozpoznávání mluvěcího

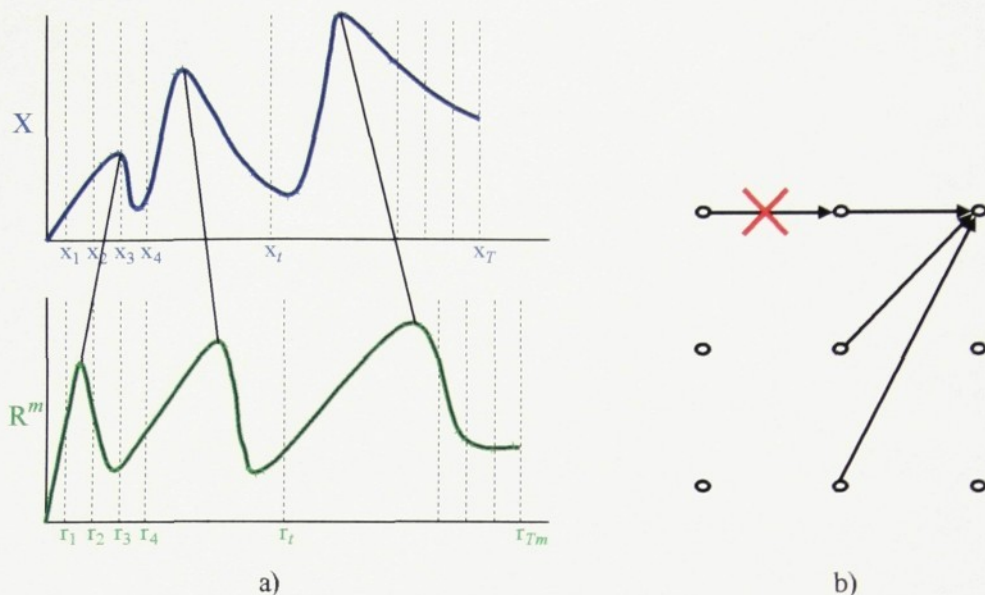
Textově závislé rozpoznávání většinou bývá základem úlohy verifikace totožnosti řečníka, ale může být i součástí obecného systému rozpoznávání řečníka. Osoba, která má být „hlasově prověřena“, má říci „heslo“, kterým může být slovo či celá věta. Systém zanalyzuje signál, porovná ho se vzory uloženými v paměti a na základě zjištěné míry podobnosti rozhodne o identitě osoby. K řešení takto definované úlohy lze v zásadě použít nejjednodušší metodu pro porovnávání dvou řečových signálů, tj. *dynamické borcení času* (DTW - Dynamic Time Warping). Její princip je detailně vysvětlen v [4].

2.1.1. Metoda DTW

DTW je nejpopulárnější metoda pro kompenzaci časové variability řečového signálu v systémech používajících referenční modely. Předpokládejme, že promluva řečníka, kterého chceme identifikovat, byla zparametrizována a je popsána časovou posloupností vektorů příznaků $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T)$. V paměti systému se nacházejí referenční vzory odpovídající stejným způsobem zparametrizovaným záznamům hesel, které předtím nahrály všechny osoby s přístupem do systému. Předpokládejme, že počet těchto osob je M a referenčním vzorem osoby s indexem m je sekvence $\mathbf{R}^m = (\mathbf{r}_1^m, \mathbf{r}_2^m, \dots, \mathbf{r}_{T_m}^m)$. Míru podobnosti mezi posloupnostmi $\mathbf{X}$ a $\mathbf{R}^m$ určíme jako vzdálenost měřenou pomocí algoritmu DTW:

$$D(\mathbf{X}, \mathbf{R}^m) = \min_f \sum_{t=1}^T d(\mathbf{x}_t, \mathbf{r}_{f(t)}^m), \quad (2.1)$$

kde $f(t)$ je transformační funkce splňující podmínky metody DTW. Při výpočtu se míra podobnosti průběžně optimalizuje vzhledem k nelineárnímu zobrazení f zavedenému mezi prvky posloupnostmi $\mathbf{X}$ a $\mathbf{R}^m$ tak, aby celková vzdálenost byla minimální. Na obr.2.1a je naznačeno, jak by měla transformační funkce pracovat. Promluva by měla být zkrácena nebo

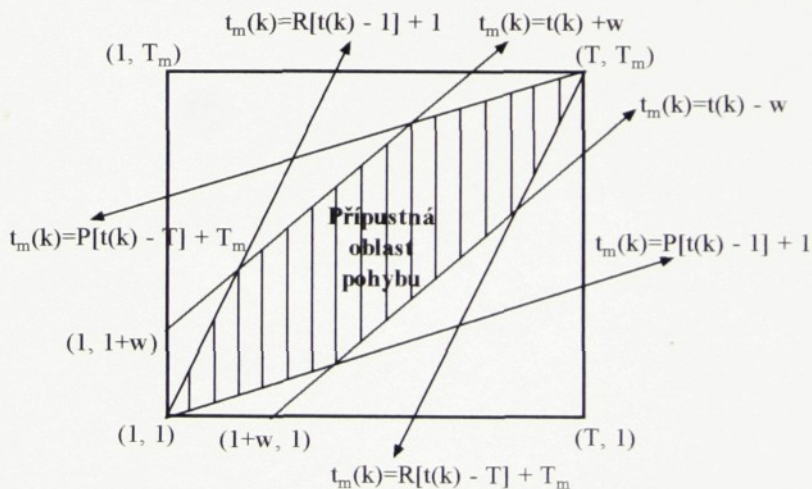


Obr.2.1 DTW – dynamické borcení času

a) znázornění, jak by měl pracovat algoritmus DTW

b) znázornění lokálních podmínek spojitosti definovaných podle Itakury

prodloužena tak, aby si oba signály (mohou to být například energie signálů), po provedení zásahů do časové osy, co nejvíce odpovídaly. Nejdůležitějším prvkem celého algoritmu je transformační funkce f , která musí splňovat řadu podmínek vycházejících ze struktury řečových obrazů v časové ose. Tyto podmínky je potom třeba respektovat při hledání optimální cesty funkce DTW v rovině (T, T_m) . Některé z těchto podmínek však lze modifikovat a vytvořit tak různé varianty algoritmu DTW. Varianta ITA (Itakura, [5]) je znázorněna na obr.2.1b.



Obr.2.2 Vymezení přípustné oblasti pohybu metody DTW

Kromě lokálních podmínek se zavádějí ještě tzv. globální podmínky, z nichž některé jsou dále vysvětleny a graficky znázorněny na obr.2.2.

- *Omezení na hraniční body* – v případě zpracování izolovaných slov, kdy jsou přesně určeny počáteční i koncové hraniční body jak testovaného, tak i referenčního obrazu, lze omezení na hraniční body funkce DTW vyjádřit podmínkami:

$$t(1)=1 \qquad t_m(1)=1 \qquad t(K)=T \qquad t_m(K)=T_m$$

- *Omezení na lokální souvislost a strmost* – aby se zamezilo nadměrné kompresi či expanzi časového měřítka, aplikují se na funkci DTW omezení na monotónnost a souvislost. Nejčastější typy lokálního omezení funkce DTW používané při rozpoznávání izolovaných slov jsou uvedeny v práci [6].
- *Globální vymezení oblasti pohybu funkce DTW* - DTW algoritmus je poměrně výpočetně náročný, proto je vhodné zobecněním podmínek lokálního omezení vymezit přípustnou oblast průchodu funkce DTW. Koeficienty P (resp. R) jsou minimální (resp. maximální) směrnice přímky vymezující přípustnou oblast a lze je odvodit z [6].

Algoritmus DTW se zatím většinou realizuje programově, i když existují i hardwarová provedení [7].

Máme-li tedy určit, které z možných osob nejvíce odpovídá daná promluva, musíme najít nejmenší z výše uvedených vzdáleností, tedy:

$$m^* = \underset{m}{\operatorname{ArgMin}} D(\mathbf{X}, \mathbf{R}^m) = \underset{m}{\operatorname{ArgMin}} \left[\underset{f}{\operatorname{Min}} \sum_{t=1}^T d(\mathbf{x}_t, \mathbf{r}_{f(t)}^m) \right] \quad (2.2)$$

Vztahem (2.2) lze určit index osoby m^* , která je nejpravděpodobnějším kandidátem z daného okruhu osob.

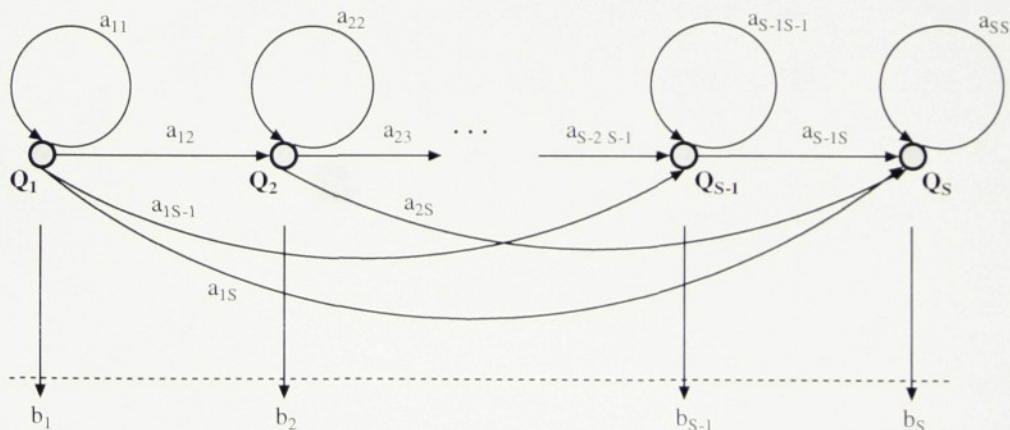
Úspěšnost a tudíž i použitelnost základní metody můžeme zvýšit dalšími faktory, např.:

- Každá osoba má jiné přístupové heslo, které je známo pouze jí a systému. Rozpoznávání se tedy opírá především o fonetickou podobu hesla a jen částečně o hlasovou charakteristiku řečníka. Heslem nemusí být jen jediné slovo, ale může to být i celá věta, což dále zvyšuje možnost lepšího odlišení různých osob.
- V přípravné (trénovací) fázi každá osoba vysloví přístupové heslo vícekrát (ideálně s časovým odstupem mezi jednotlivými promluvami). Podle vztahu (2.1) se z nahrávek určí vzor nejlépe reprezentující danou osobu a zároveň se změří míra odchylky mezi jednotlivými promluvami téže osoby. Tuto míru pak lze využít pro nastavení prahu pro odmítnutí.
- V přípravné fázi je několik dalších osob požádáno, aby též vyslovily stejná hesla jako oprávněné osoby. Reprezentace těchto promluv mohou být zařazeny do databáze vzorů, s nimiž se provádí klasifikace podle vztahu (2.2). Zároveň mohou být použity pro

upřesnění hodnoty prahu podle bodu b). Systém je pak schopen lépe odolávat pokusům o neoprávněný přístup.

2.1.2. Metoda HMM

Princip metody modelování řeči *skrytými markovskými modely* (HMM – Hidden Markov Models) vychází z představy o vytváření řeči. Při generování řeči člověkem si lze představit, že hlasové ústrojí je během krátkého časového intervalu (např. segmentu) v jednom z konečného počtu stavů artikulačních konfigurací (např. generuje určitý foném). V uvažovaném segmentu je pak hlasovým ústrojím produkován krátký signál, který závisí na stavu artikulačního ústrojí a může být popsán určitými spektrálními charakteristikami. Toho lze například dosáhnout vektorovou kvantizací, kdy je vytvořena kódová kniha typových spektrálních vzorů a každá spektrální charakteristika je nahrazena indexem „nejbližšího“ typového spektrálního vzoru z kódové knihy. Při konstrukci klasifikátoru, který je založen na modelování řečového signálu pomocí markovského modelu, vycházíme také z představy o vytváření řeči. Při modelování jsou generovány dvě vzájemně svázané časové posloupnosti náhodných proměnných, a to podpůrný markovský řetězec, který je posloupností konečného počtu stavů, a řetězec konečného počtu spektrálních vzorů. Pro jednotlivé spektrální vzory jsou vytvořeny náhodné funkce, které pravděpodobnostně ohodnocují vztah těchto vzorů ke všem stavům. Předpokládá se, že v jednotlivých časových diskretních okamžicích je proces v jediném stavu a lze jej pozorovat prostřednictvím náhodné funkce korespondující s běžným stavem. Podpůrný markovský řetězec pak mění stavy podle své matice pravděpodobností přechodu. Pravděpodobnost přechodu pro rozpoznávaného mluvěcího může být odhadnuta ze sady trénovacích vektorů a následně pak určí pravděpodobnost, že daná promluva byla vyprodukována rozpoznávaným mluvěcím.



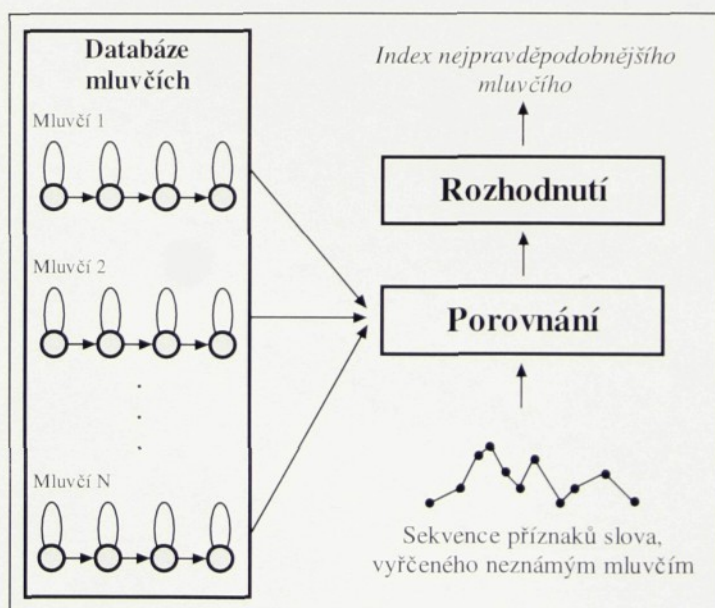
Obr.2.3 HMM – struktura modelu, která je obvyklá pro reprezentaci promluvy

Při modelování mluvené řeči se prioritně využívají tzv. levo-pravé markovské modely (obr.2.3), které jsou použity tam, kde probíhá nějaký vývoj s postupujícím časem. Základní vlastností uvedených modelů je, že proces začíná příchodem prvního spektrálního vzoru

z počátečního stavu modelu a se vzrůstajícím časem dochází k přechodům ze stavů s nižšími indexy do stavů s vyššími indexy nebo setrvání ve stejném stavu. Proces končí příchodem posledního spektrálního vzoru, přičemž model se v tom okamžiku nachází v koncovém stavu. Přímá cesta modelem přitom reprezentuje průměrné trvání slova. Přechody do stejného stavu berou v úvahu prodloužení slova a přechody do příštího stavu dovolují zkrácení slova. Model promluvy je tedy tvořen S stavy uspořádanými v lineární struktuře. Přechod je možný (ale také nemusí nastat – viz obr.2.3) pouze mezi sousedními stavy a pouze ve směru zleva doprava. Pravděpodobnost setrvání modelu ve stavu s je dána hodnotou a_{ss} , pravděpodobnost přechodu do sousedního stavu hodnotou $a_{s,s+1}$. Dále musí platit, že $a_{ss} + a_{s,s+1} = 1$. Každý stav b_s je charakterizován pravděpodobnostní výstupní funkcí s normálním rozložením vyjádřenou hodnotami $\bar{x}_s$ a Σ_s .

Metodu skrytých markovských modelů lze použít prakticky stejným způsobem jako v předešlém případě. Každá osoba má svůj model natrénovaný na základě několika nahrávek svého přístupového hesla. Při rozpoznávání se pak hledá osoba, jejíž model dosahuje nejvyšší pravděpodobnosti. Vzhledem k faktu, že k natrénování modelu je potřeba více nahrávek téhož hesla, má model v sobě již schopnost lépe určovat míru věrohodnosti, že osoba s nejvyšším skórem je ta pravá. I zde je však dobré využít další možnosti naznačené v předchozí kapitole.

Klasifikátor HMM pracuje na podobném principu jako klasifikátor zmíněný v předchozí kapitole – viz obr.2.4. Databáze mluvěcích v tomto případě není složena z referenčních vzorů promluv, ale z modelů mluvěcích. Ty jsou natrénovány vícekrát vyřčenými promluvami jednoho slova (textově závislé rozpoznávání). Další podrobnosti o metodě HMM se lze dočíst v [4].



Obr.2.4 Klasifikátor založený na technice skrytých markovských modelů (HMM).
Mluvčí ve slovníku jsou reprezentováni modely stejných slov.

2.2. Metody pro textově nezávislé rozpoznávání mluvího

K rozpoznávání mluvího, které má být nezávislé na obsahu promluvy, již nelze použít metodu DTW a ani výše uvedenou verzi HMM. Teoreticky by bylo možné použít hláskově orientovaný rozpoznávač pracující metodou HMM. Každý mluví by měl v paměti systému sadu modelů všech hlásek a systém by pro každou takovou sadu vyhodnotil nejpravděpodobnější sekvenci hlásek (s využitím Viterbiho dekodéru známého z úlohy rozpoznávání souvislé řeči). Mluví, jehož sekvence by získala nejvyšší skóre, by byl prohlášen za hlavního kandidáta při rozpoznávání. Tento způsob by však byl velmi náročný na množství řeči, jenž by každá osoba musela dodat pro natrénování modelů. Prakticky použitelný by byl pouze v případech, kdy jsou k dispozici rozsáhlé záznamy promluv konkrétní osoby a kdy se spíše jedná o ověření hypotézy, že jiné hlasové záznamy patří téže osobě.

Metody DTW a HMM využívají informaci o časovém vývoji parametrů signálu řeči. Při rozpoznávání obsahu řeči je to nutné, v úloze identifikace mluvího se však bez této časové informace můžeme obejít. To je základem metod založených na *vektorovém kvantování řeči* (VQ – Vector Quantization) a na *gaussovských mixturových modelech* (GMM – Gaussian Mixture Models).

2.2.1. Metoda založená na vektorovém kvantování

Vektorové kvantování je metoda, která spojité vektorové hodnoty transformuje na konečný počet diskrétních vektorových hodnot. Představuje zobecnění klasického skalárního kvantování, dobře známého z procesu digitalizace, kdy jsou spojité hodnoty analogového signálu přiřazována čísla z omezeného souboru. Např. v osmibitovém AD převodníku jsou všechna napětí převedena do škály 256 hodnot.

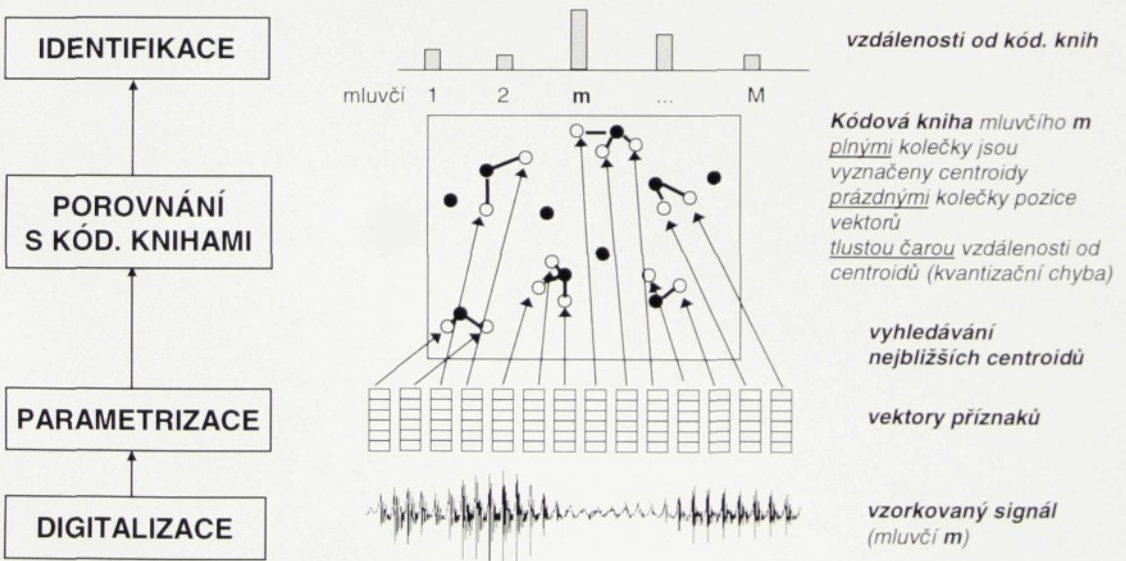
Princip vektorového kvantování spočívá v tom, že prostor, v němž jsou definovány P -dimenzionální vektory, je rozdělen do L podprostorů a každý podprostor je reprezentován jediným vektorem, který se nazývá *centroid*. Rozdělení prostoru většinou není rovnoměrné, ale řídí se skutečným prostorovým rozmístěním vektorů v dané úloze. Tyto vektory často tvoří nepravidelné shluky a právě tyto shluky jsou jádry zmíněných podprostorů, z nichž se metodami založenými na průměrování stanovují centroidy. Centroidy lze proto považovat za jakási těžiště shluků a nejlepší reprezentanty vektorů v daném prostoru. Metoda vektorového kvantování se často používá pro úsporné kódování a přenos vektorových signálů. Z této skutečnosti vyplývá i terminologie VQ. Centroidy se běžně označují jako *kódová slova*, která tvoří *kódovou knihu* o velikosti L . Princip úsporného kódování spočívá v tom, že místo aby se přenosovým kanálem přenášely původní vektory, vysílá se pouze index nejbližšího kódového slova. Redukce datového toku může být značná. Např. k přenosu 10-složkového vektoru složeného z čísel typu *float* by bylo třeba $10 \times 4 = 40$ bytů. Při úsporném přenosu s využitím kódové knihy o 256 slovech by stačil pouze jediný byte; došlo by tedy ke čtyřicetinasobné úspoře.

Aplikaci VQ v úloze rozpoznávání mluvěcího lze popsat následujícím způsobem. Každý mluvěcí je požádán, aby vyslovil několik trénovacích promluv. Jejich počet nemusí být velký, ale vychází z požadavku, aby věty pokryly většinu hlásek vyskytujících se v daném jazyce. Signál těchto nahrávek je zparametrizován s použitím P příznaků. Pro každého mluvěcího je vytvořena vlastní kódová kniha, nejlépe reprezentující všechny takto získané vektory P -dimenzionálního prostoru. Kódovou knihu mluvěcího s indexem m tvoří L vektorů $\{\mathbf{V}_1^m \dots \mathbf{V}_L^m\}$. V procesu identifikace promluvy neznámého mluvěcího se vypočítá vzdálenost sekvence příznakových vektorů $\mathbf{X}$ od kódových knih jednotlivých osob:

$$D(\mathbf{X}, \mathbf{V}^m) = \sum_{t=1}^T \text{Min}(\mathbf{x}_t, \mathbf{V}_i^m), \quad (2.3)$$

kde $\mathbf{V}_i^m$ je kódové slovo nejbližší vektoru $\mathbf{x}$ v čase t . Tato vzdálenost se nazývá *celková kvantizační chyba* (nebo také *kvantizační zkreslení*). Mluvěcí, jehož kódová kniha má nejnížší celkovou vzdálenost stanovenou podle (2.3), se stává vítězem klasifikace - viz obr.2.5. Na tomto obrázku je ukázáno, jak by mohlo vypadat rozdělení ve 2-dimenzionálním vektorovém prostoru. Každý shluk příznaků je reprezentován jedním centroidem.

Vztah (2.3) se podobá vztahu (2.1). Významný rozdíl však spočívá ve skutečnosti, že vektory kódové knihy $\mathbf{V}$ nejsou časově uspořádány tak jako vektory referencí $\mathbf{R}$. Nehledá se zde proto časové, nýbrž prosté vzdálenostní přiřazení. Náročnost výpočtu rozpoznávání lze odhadnout z následující úvahy. Pro nalezení nejbližšího kódového slova v čase t je třeba L výpočtů vzdáleností v P -dimenzionálním prostoru. Je-li promluva popsána T příznakovými vektory a je-li v databázi M mluvěcích, musíme provést celkem $L \times T \times M$ výpočtů vzdáleností.



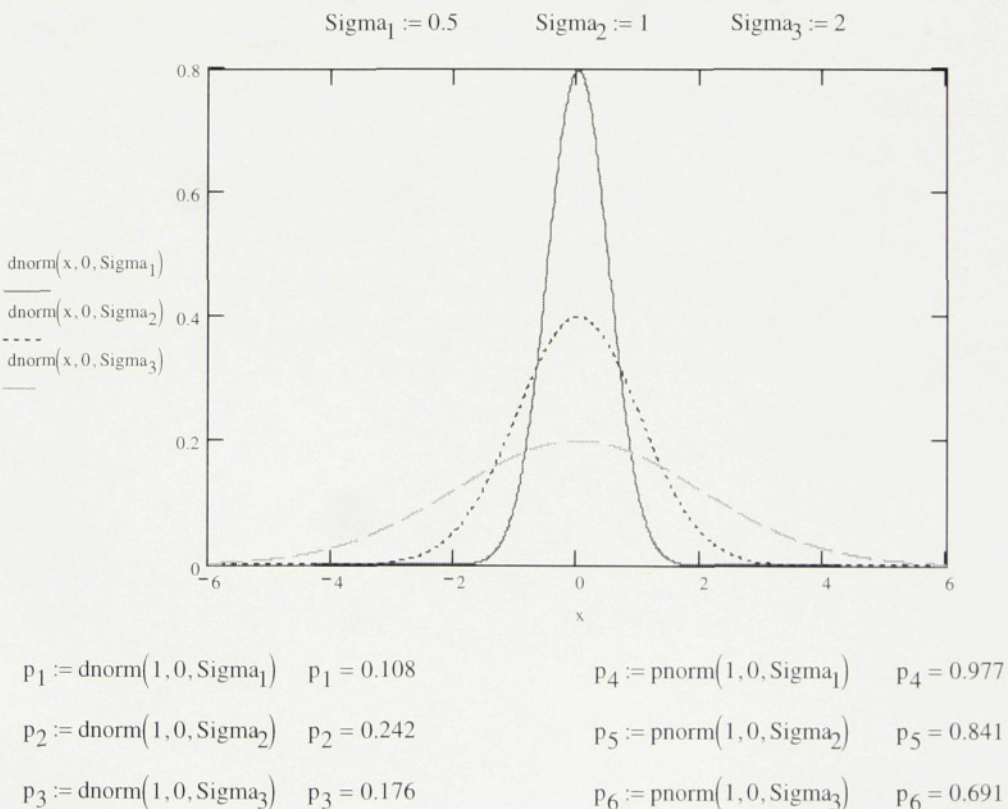
Obr.2.5 Proces identifikace mluvěcího metodou vektorové kvantizace

Metoda tvorby kódové knihy je časově i algoritmicky náročnější a je popsána v následujících kapitolách.

2.2.2. Metoda založená na gaussovských modelech

Nejprve si řekněme něco málo o gaussovských rozloženích. Proč jsou vlastně gaussovská rozložení tak hojně využívána? Mnoha studiemi a mnoha různými fyzikálními experimenty (i na řečových signálech) bylo prokázáno, že gaussovská rozdělení jsou prakticky všude kolem nás. Proto se také gaussovské rozdělení někdy nazývá „normální rozdělení“. Pro příklady výskytu gaussovského rozdělení nemusíme chodit daleko. Například to může být měření v úhlových nebo lineárních rozměrech, měření složek rychlosti, anebo měření výšky, případně váhy lidí. Rozložení náhodné proměnné x můžeme reprezentovat gaussovským rozdělení tehdy, pokud hustota pravděpodobnosti jejího rozdělení o střední hodnotě $\bar{x}$ a směrodatné odchylce σ^2 je dána vztahem:

$$f(x) = N(\bar{x}, \sigma^2) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot \exp \left[-\frac{(x - \bar{x})^2}{2\sigma^2} \right] \quad (2.4)$$

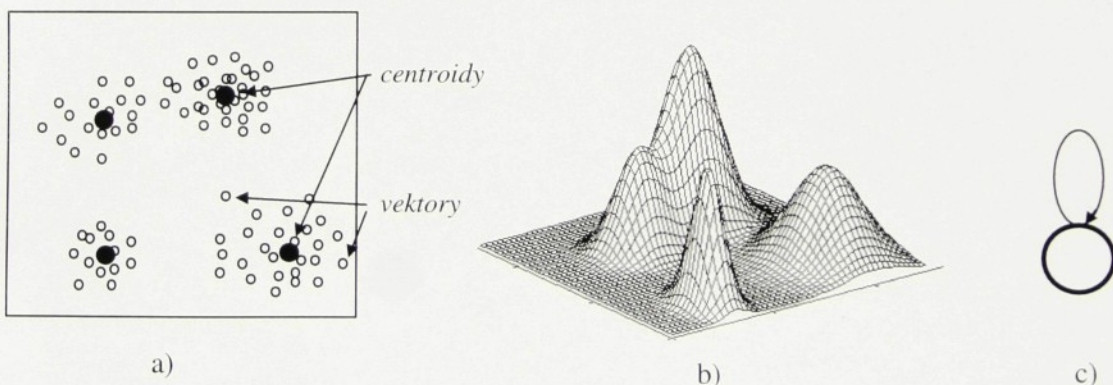


Obr.2.6 Tři gaussovská rozdělení se stejnou hodnotou $\bar{x}$, ale každé s jinou směrodatnou odchylkou. Rozdělení s nejmenší směrodatnou odchylkou má nestrmější průběh

Na obr.2.6 vidíme, že gaussovské rozdělení je symetrické kolem $\bar{x}$ a největší hodnotu má také v $\bar{x}$. Funkcí $dnorm$ vypočítáme hustotu pravděpodobnosti, kde prvním parametrem této funkce je bod, pro který hodnotu počítáme. Druhým a třetím parametrem jsou již zmíněné hodnoty $\bar{x}$ a σ^2 . Funkce $pnorm$ má stejné parametry, avšak na rozdíl od $dnorm$ vypočítává distribuční funkci.

Pokud bychom chtěli co nejjednodušeji vysvětlit princip metody GMM v systémech rozpoznávajících řečníka, je možná jistá analogie s VQ. Obě metody mají společné to, že pracují s jistým druhem kódové knihy.

Při vektorové kvantizaci jsou shluky příznakových vektorů reprezentovány pouze centroidy, tedy jakýmsi průměrnými vektory - viz obr.2.7a. Mnohem úplnější informaci o poloze a tvaru shluků získáme, jestliže se pokusíme stanovit pravděpodobnostní hustotu rozložení vektorů v prostoru. Vzhledem k existenci shluků nelze aplikovat jednomodální gaussovské rozložení, ale je potřeba použít lineární kombinaci více gaussovských rozložení. Tento způsob modelování nazýváme gaussovské mixturové modely. Zkombinováním více gaussovských rozložení můžeme reprezentovat jakékoliv rozdělení hustoty pravděpodobnosti. A právě to je velmi vhodné pro aplikace textově nezávislého rozpoznávání mluvěho. Ukázka funkce hustoty pravděpodobnosti složené ze 4 mixtur je na obr.2.7b a odpovídá přibližně rozložení vektorů znázorněných na obr.2.7a. Hlavní rozdíl mezi oběma přístupy je tedy v tom, že jednotlivé gaussovské funkce jsou popsány hustotou pravděpodobnosti rozložení *všech* příznaků (každý příznak nějakým způsobem ovlivní gaussovskou funkci), kdežto jednotlivé příznaky v případě vektorové kvantizace jsou přiřazeny pouze jednomu centroidu.



Obr.2.7 GMM - gaussovský mixturový model

a) znázornění prostorového rozložení vektorů a centroidů

b) odpovídající 4-mixturový model, c) GMM jako zvláštní případ jednostavového HMM

GMM je definován jako vážená směs gaussovských (normálních) rozložení:

$$b(\mathbf{x}) = \sum_{m=1}^M c_m N(\bar{\mathbf{x}}_m, \Sigma_m) = \sum_{m=1}^M c_m \frac{1}{\sqrt{(2\pi)^p \det \Sigma_m}} \cdot \exp\left[-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}}_m)^T \Sigma_m^{-1}(\mathbf{x} - \bar{\mathbf{x}}_m)\right], \quad (2.5)$$

kde M je počet mixtur, c_m je váhový koeficient m -té mixtury, $\bar{\mathbf{x}}_m$ a Σ_m jsou střední hodnoty a kovarianční matice jednotlivých mixtur. Vztah (2.4) se velmi podobá vztahu pro výstupní funkci u vícemixturových HMM (skrytých markovských modelů), používaných běžně při rozpoznávání řeči. GMM lze skutečně považovat za zvláštní případ HMM, jak naznačuje obr.2.7c. Jde o HMM, v němž je potlačena časová informace, neboť model zůstává stále ve svém jediném stavu. Tuto podobnost lze s výhodou využít při trénování GMM. Máme-li k dispozici software pro trénování HMM, můžeme jej aplikovat též na případ GMM vhodným nastavením parametrů. Jinak je nutné naprogramovat vlastní iterační proceduru založenou na principu EM (Expectation - Maximization).

Použití metody GMM v úloze rozpoznávání mluvího je podobné jako v případě metody VQ. Nejprve pro každého mluvího natrénujeme jeho vlastní GMM, tj. parametry pravděpodobnostní funkce (2.4). Při rozpoznávání řečníka, jehož promluva je reprezentována sekvencí příznakových vektorů $\mathbf{X}$, hledáme model m^* dosahující nejvyšší pravděpodobnosti:

$$m^* = \underset{m}{\operatorname{ArgMax}} P(\mathbf{X}, \mathbf{M}^m) = \underset{m}{\operatorname{ArgMax}} \left[\prod_{t=1}^T b_m(x_t) \right] \quad (2.6)$$

V praktických aplikacích se používají GMM s 16 až 256 mixturami v závislosti na množství trénovacího materiálu. Bližší informace o metodě GMM lze nalézt v [8]. Výsledky nasazení bývají o něco málo lepší v porovnání s metodou VQ, a to díky věrnějšímu popisu prostorového rozložení příznakových vektorů jednotlivých osob [9].

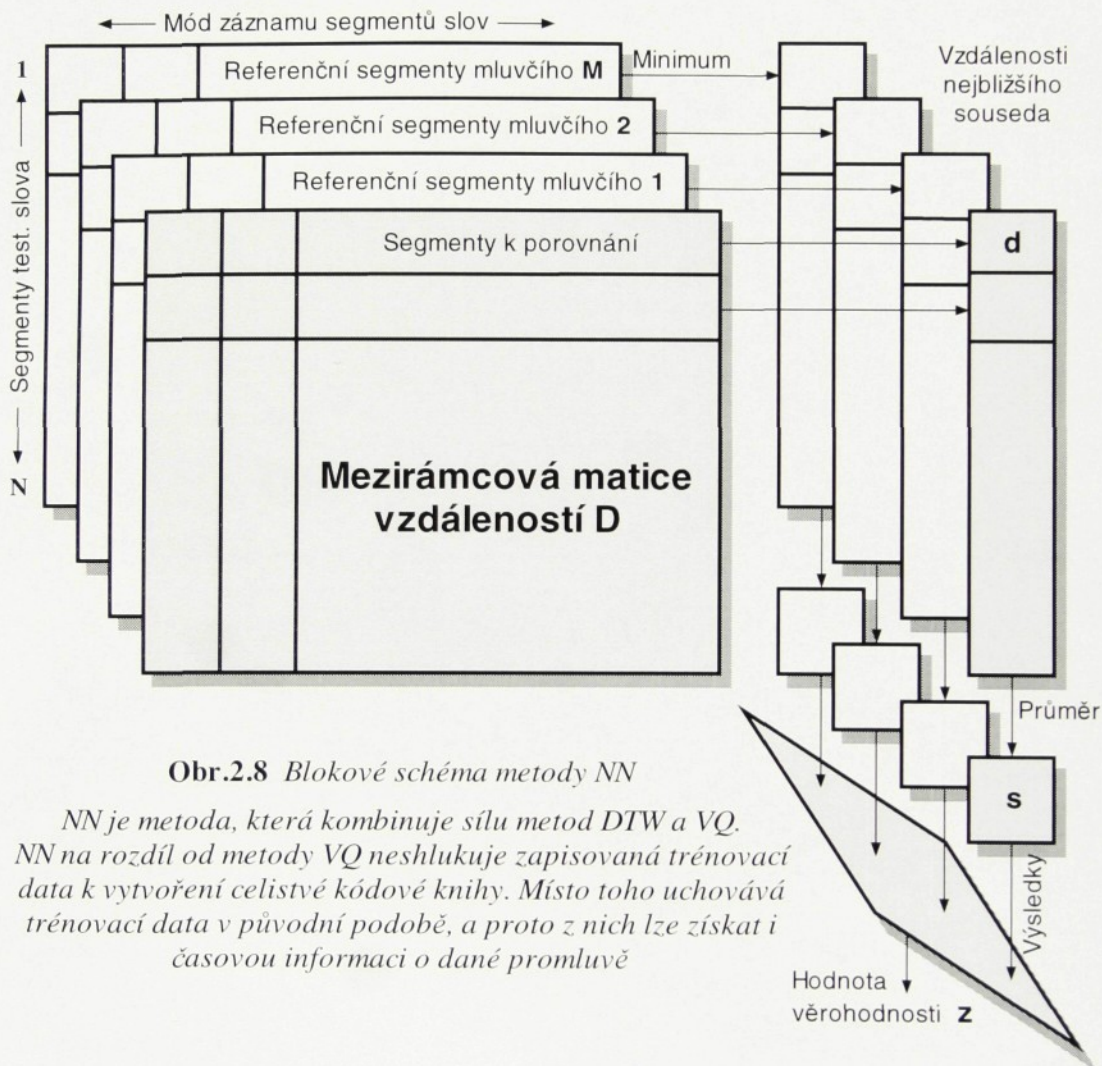
2.2.3. Metoda nejbližšího souseda

Metoda *nejbližšího souseda* (NN - Nearest Neighbor) nebo obecněji metoda k NN, může být použita k vypočítání *hustoty pravděpodobnosti* (PDF – probability density function) dat v průběhu třídění nebo může být sama o sobě použita jako klasifikátor. Nejjednodušším k NN klasifikátorem je NN klasifikátor (k NN, kde $k = 1$). Jeho práce probíhá takto: jsou uložena veškerá trénovací data, tzn. všechny příznakové vektory s korespondujícími popisy. Dále jsou vypočteny vzdálenosti mezi testovanými vektory příznaků a všemi trénovacími daty od všech mluvích. Testovaným vektorům jsou následně přiřazeny trénovací vektory (respektive jejich jmenovky) s nejmenší vzdáleností, tj. nejbližší sousedé. Jsou spočtena skóre pro všechny mluví a ten s nejlepším skóre je vybrán jako autor promluvy.

Obr.2.8 znázorňuje metodu NN aplikovanou na úlohu VM. Jak můžeme vidět na schématu, mezirámcová matice vzdáleností je spočtena měřením vzdálenosti mezi testovanými segmenty (vstup) a žadatelovými referenčními segmenty (segmenty byly uloženy v režimu trénování). NN vzdálenost je minimální vzdálenost mezi testovanými a zaznamenanými segmenty. Pro vypočtení konečného výsledku je třeba všechny NN vzdálenosti testovaných segmentů zprůměrovat. Jak je vidět na zadní části schématu, testované segmenty jsou ještě porovnány s referencemi uloženými od dalších mluvích.

Výsledky jsou pak zkombinovány a je z nich vypočten výsledný pravděpodobnostní koeficient z .

Klasifikátor založený na metodě nejbližšího souseda uchovává všechna trénovací data a pracuje tak, že hledá nejbližšího souseda mezi všemi daty. Tento postup je ovšem velmi náročný jak na operační paměť, tak i na výpočetní sílu počítače.



Obr.2.8 Blokové schéma metody NN

NN je metoda, která kombinuje sílu metod DTW a VQ. NN na rozdíl od metody VQ neshlukuje zapisovaná trénovací data k vytvoření celistvé kódové knihy. Místo toho uchovává trénovací data v původní podobě, a proto z nich lze získat i časovou informaci o dané promluvě

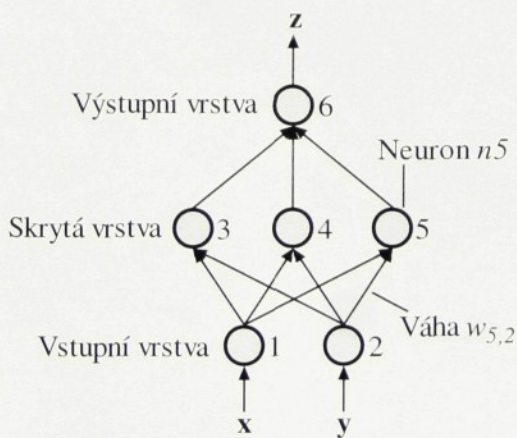
2.2.4. Umělé neuronové sítě

Biologické neuronové sítě jsou základem nervových soustav v živé přírodě. Tyto sítě se skládají z nervových buněk, *neuronů*. Každý neuron je má spojen s ostatními neurony velkým množstvím vstupů, tzv. *dendritů*. Počet těchto dendritů dosahuje několik desítek až stovek tisíc. Neuron reaguje na součet signálů na svých vstupech. Pokud součet těchto signálů přesáhne určitou prahovou hodnotu, neuron vygeneruje výstupní signál, který transformuje k ostatním neuronům. Každý neuron má pouze jeden výstup, který se nazývá *axon*. Axon je

spojen se vstupy ostatních neuronů takzvanými *synapsemi*. Synapse, které fungují jako rozhraní, zesilují nebo naopak zeslabují výstupní signál neuronu. Tato rozhraní fungují pouze jako jednosměrné brány, tj. přenášejí signál pouze ve směru axon-dendrit.

Neuronová síť se vyznačuje schopností učení, tj. schopností správně reagovat na opakovaný podnět. Pokud se někde objeví korelace mezi určitými výstupy neuronů a nějakou akcí, pak je možné brát výstupy těchto neuronů jako indikátory této akce. Při učení se aktivují některé neurony a nastavují synapse mezi sebou. Opakováním učení se vytvoří v mozku dendritická cesta mezi neurony sloužícími jako detektory a mezi řídicími neurony, které na daný podnět reagují. Proto pokud se daný podnět objeví opakovaně, neuronová síť vygeneruje příslušnou odezvu.

Neurony a neuronové sítě lze také simulovat matematicky. Prostředky pro tuto simulaci se nazývají *umělé neuronové sítě* (ANN - Artificial Neural Networks). Hlavní síla umělých neuronových sítí spočívá v jejich schopnosti najít (aproximovat) závislost mezi vstupními proměnnými a mezi proměnnými výstupními. V matematické simulaci je neuron objektem, který má svoje vstupy a prahovou hodnotu definovanou reálným číslem, synapse jsou nahrazeny váhami. Neurony v umělé neuronové síti jsou uspořádány do vrstev. Každý neuron (kromě neuronů ve vstupní vrstvě) přijímá výstupy od každého neuronu z předcházející vrstvy a je spojen výstupem s každým neuronem v následující vrstvě (kromě neuronu v poslední vrstvě). Počet vrstev neuronové sítě a počty neuronů v jednotlivých vrstvách závisí na problému, který má být zpracován pomocí neuronové sítě.



Obr.2.8 Schéma neuronové sítě

První vrstva tvořená neurony 1 a 2 se nazývá vstupní vrstva. Vrstva tvořená neurony 3, 4 a 5 se nazývá skrytá, nebo také vnitřní vrstva. V poslední, tzv. výstupní vrstvě se nachází jediný neuron. Počet neuronů ve vstupní a ve výstupní vrstvě je dán počtem vstupních a výstupních proměnných. Počet neuronů ve skryté vrstvě je dán složitostí aproximované funkce.

O principu funkce umělých neuronových sítí pojednává obr.2.8, kde je zobrazeno schéma umělé neuronové sítě. Vidíme, že umělá neuronová síť se skládá z neuronů (uzlové body), které jsou navzájem propojeny váhami. Každý neuron je reprezentován racionálním číslem, které můžeme nazvat hodnotou neuronu. Jednotlivé neurony jsou uspořádány do vrstev. Na obr.2.8 je zobrazena síť tvořená třemi vrstvami: první vrstva, tvořená neurony 1 a 2, se nazývá vstupní vrstva, druhá, tvořená neurony 3, 4, 5, se nazývá skrytá, nebo také vnitřní vrstva. V poslední (tzv. výstupní) vrstvě se v tomto konkrétním případě nachází jediný

neuron. Počet neuronů ve vstupní a ve výstupní vrstvě je dán řešeným problémem (resp. počtem vstupních a výstupních proměnných). Počet neuronů ve skryté vrstvě je dán složitostí aproximované funkce a bývá předmětem optimalizace neuronové sítě pro konkrétní problém.

Spojení mezi neurony se nazývá váha a je podobně jako neurony samotné reprezentováno racionálním číslem. Všechny neurony dvou sousedních vrstev jsou navzájem propojeny váhami.

Úkolem neuronové sítě je vypočítat hodnoty výstupních proměnných. Postup výpočtu výstupních proměnných neuronovou sítí je následující :

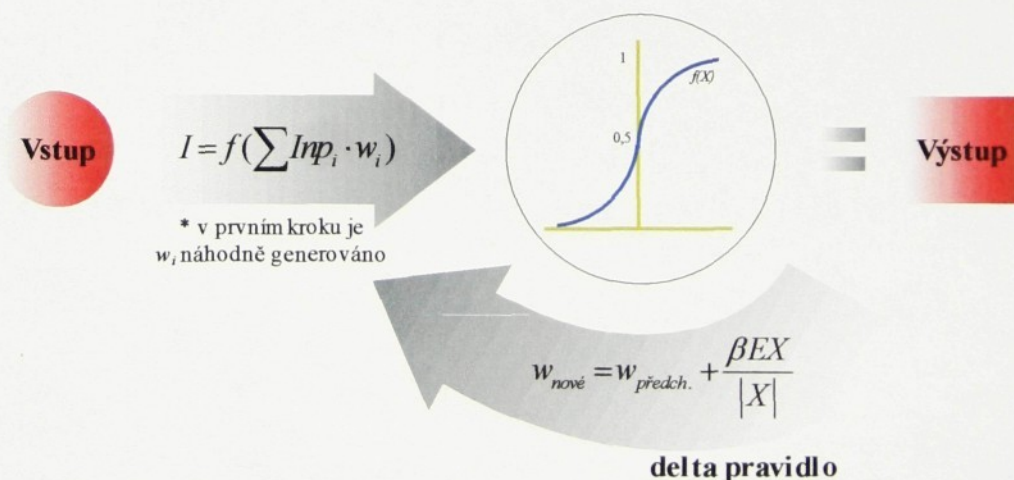
- hodnoty vstupních neuronů jsou nastaveny na hodnoty vstupních proměnných
- v dalším kroku jsou vypočítány hodnoty neuronů nacházejících se v mezivrstvě
- následně jsou podle stejného schématu vypočteny hodnoty neuronů ve vrstvě výstupní
- hodnota neuronů v poslední vrstvě je přímo rovna hodnotě výstupní proměnné

Hodnoty jednotlivých neuronů jsou počítány postupně, počínaje prvním neuronem ve skryté vrstvě, konče posledním neuronem ve vrstvě poslední. Hodnota každého neuronu je vypočtena jako součet hodnot neuronů v předcházející vrstvě násobených hodnotami jednotlivých vah:

$$Out = f(\sum(Inp_i \cdot w_i) + bias_i), \quad (2.7)$$

kde Inp_i je vstup neuronu (hodnota neuronu v předchozí vrstvě), w_i je váha spojení s příslušným neuronem, $bias_i$ je prahová hodnota neuronu, $f()$ je sigmoidní funkce, která má tvar $y = 1/(1 + \exp(-X))$.

Jak vyplývá z rovnice (2.6), veškerá výpočetní síla neuronové sítě spočívá pouze v hodnotách vah a biasů jednotlivých neuronů v neuronové síti. Hodnoty těchto vah a biasů



Obr.2.9 Příklad trénování jednoho uzlu tak, jak by to bylo provedeno v případě BPNN

jsou získávány tzv. trénováním neuronové sítě. Jde o proces analogický učení u biologických neuronových sítí, které se naučí reagovat na určitý podnět definovaným způsobem.

Trénování neuronové sítě vyžaduje tzv. trénovací soubor (nebo také trénovací množina), který se skládá ze vstupních a výstupních hodnot. Při trénování umělé neuronové sítě jsou pomocí aktuálně nastavených hodnot vah (na počátku inicializovaných generátorem náhodných čísel) vypočítány hodnoty výstupních neuronů (tedy hodnoty výstupních proměnných) pro vstupní hodnoty obsažené v trénovacím souboru. Tyto vypočtené hodnoty jsou porovnány se skutečnými hodnotami výstupních proměnných. Na základě srovnání vypočtených a skutečných hodnot výstupních proměnných je provedena korekce vah a biasů jednotlivých neuronů tak, aby při dalším výpočtu výstupů bylo dosaženo lepší shody mezi hodnotami vypočtenými a hodnotami skutečnými (teoretickými). Tento proces se nazývá trénovací cyklus a pro jeden uzel je zobrazen na obr.2.9. Trénování neuronové sítě se opakuje buď až do dosažení zvoleného počtu trénovacích cyklů, do dosažení zvolené hodnoty tss (2.7), případně dokud celý proces konverguje, tj. významně se zmenšuje hodnota tss .

$$tss = \sum (Out_i - Trg_i)^2 \quad (2.8)$$

Out_i je výstup sítě (hodnota i -tého výstupního neuronu) a Trg_i je očekávaná hodnota i -tého výstupního neuronu.

Existuje několik způsobů učení, nejpoužívanějším je tzv. metoda trénování se *zpětným šířením chyb*, odtud je také odvozen název *neuronové sítě se zpětným šířením chyb* (BPNN - Backpropagational Neural Network). Síť s tímto způsobem trénování jsou v dnešní době nejvíce používané. Tato metoda patří do skupiny *učení s učitelem* (supervised learning), kdy učení probíhá tak, že síti jsou postupně předkládány dvojice vstup – požadovaný výstup. Po předložení vstupu je síť nechána, aby vypočítala výstup, který je porovnán s požadovanou odezvou a trénovací algoritmus automaticky upraví hodnoty vah tak, aby shoda výsledků vypočtených neuronovou sítí se skutečnými výsledky byla maximální. Trénování se provádí tak dlouho, dokud není dosažen limitní počet cyklů (během jednoho cyklu se spočítá výstup sítě a nastaví se váhy), nebo dokud hodnota tss dosáhne určité maximální hodnoty. Neuronová síť pomocí učení je schopna najít přibližné matematické závislosti a použít je pro výpočet výsledků i ze vstupů, které nebyly známy během trénování sítě. Jakmile je tedy síť jednou natrénovaná, je schopna (s určitou chybou) předpovědět výsledky pro neznámá data. Další metodou, jak je možné neuronovou síť trénovat je takzvané *učení bez učitele* (unsupervised learning). V tomto případě jsou síti předkládány pouze vstupy, protože informace o výstupech (informace od učitele) úplně chybí. Síť musí sama hledat zákonitosti ve vstupních datech a provádět nastavování vah. Cílem je taková konfigurace vah, při které síť na podobné vstupy reaguje podobnými výstupy.

Systémy založené na ANN nepochybně řeší některé problémy lépe než jakékoliv jiné metody. Například přesnost v rozpoznávání fonémů je o několik procent vyšší než u všech ostatních systémů. Výsledky získané z ANN jsou typicky kompaktní, což znamená že mají

malý počet parametrů a nabízejí rychlejší dekodování v porovnání se standardními HMM systémy. Masivnějšímu rozšíření však brání obtížné trénování sítě, kdy je třeba účinně a robustně určit parametry rozsáhlé sítě. To má za následek velké množství dat nutných pro režim trénování. Síť je také možné přetrénovat, což je příčinou poklesu procenta úspěšného rozpoznání.

3. REALIZACE METOD PRO IDENTIFIKACI MLUVČÍHO

Při tvorbě systému pro IM se v první řadě musíme rozhodnout, na jakém principu bude pracovat klasifikátor a zda se bude jednat o systém textově závislý nebo textově nezávislý. Dále je třeba rozhodnout, zda identifikace bude probíhat v režimu otevřené nebo uzavřené množiny. Jak již bylo uvedeno v kapitole 1.2, je to situace, kdy chceme rozpoznat osobu, pro kterou nemusí být ve slovníku uložena žádná reference a systém na to musí odpovídajícím způsobem reagovat. Po provedení těchto úvah může být zvolena některá z dostupných metod (přehled metod je uveden v kapitole 2). Protože pro testování našeho systému byly použity nahrávky od mluvěcích, kteří vyslovovali mimo jiné i věty z denního tisku (různé texty), bylo nutné rozhodnout se pro textově nezávislý systém.

Jako první, vzhledem k tomu, že je to nejjednodušší a nejintuitivnější metoda pro textově nezávislé rozpoznávání mluvěcího, byla pro realizaci systému použita metoda vektorového kvantování. Navíc v rámci ročníkového projektu [10] již systém využívající vektorové kvantování vyvíjen byl, a tak bylo možné navázat na zkušenosti získané při vývoji tohoto systému. Ze stejných důvodů, proč jsme použili VQ, použijeme klasifikátor pracující s gaussovskými mixturovými modely pro druhý systém. Jak bylo již uvedeno v kapitole 2.2.2, pro GMM dále hovoří skutečnost, že jsou logickým vylepšením VQ. Další ne nepodstatnou výhodou je to, že obě metody používají stejný způsob testování, tj. kromě klasifikátoru a způsobu trénování se princip činnosti obou systémů neliší. Z toho vyplývá, že to, co bylo použito pro VQ, je možné použít i pro GMM (celá fáze předzpracování signálu, výpočet a následná práce s příznaky, uživatelské prostředí, vyhodnocování výsledků).

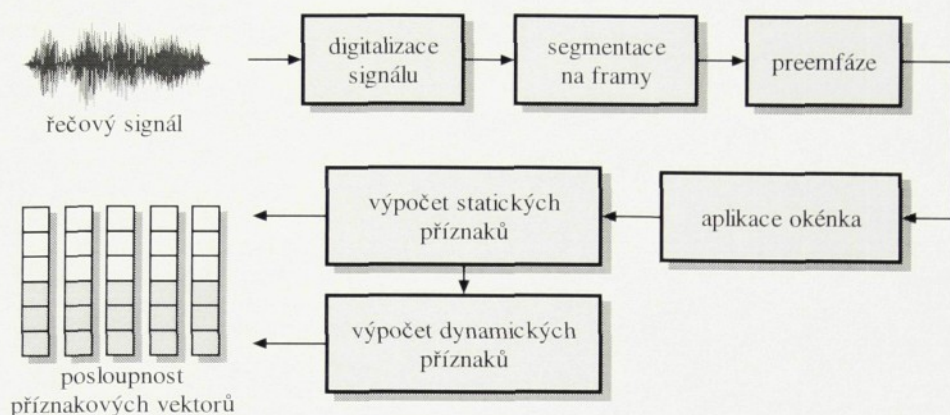
Dalším velmi důležitým elementem při konstrukci systému pro identifikaci mluvěcího je proces, kdy je řečový signál parametrizován pro použití v počítači. Jak bude ukázáno v experimentální části této diplomové práce, správná volba počtu a druhu příznaků má na úspěšnost rozpoznávání stejně významný vliv jako volba klasifikátoru. V tomto směru byla volba poměrně jednoduchá, protože stejně jako je tomu v případě rozpoznávání řeči, tak i v oblasti rozpoznávání mluvěcích je nejpoužívanější kepstrální reprezentace signálu. V našem systému budou tedy použity MFCC kepstrální příznaky.

V této kapitole bude naznačen postup konstrukce textově nezávislého systému pro rozpoznávání řečníka, který bude pracovat v uzavřené množině, a bude založen na kepstrální reprezentaci signálu a na metodách VQ a GMM.

3.1. Parametrizace řečového signálu

Hlavním cílem parametrizace řečového signálu je snaha o odstranění nadbytečné (redundantní) informace, která je v řečovém signálu v poměrně hojné míře zastoupena. Parametrizaci provádíme tak, abychom získali příznaky, které budou co nejvěrněji reprezentovat originální signál. Po příznacích na druhou stranu požadujeme, aby parametrizovaný signál byl pokud možno co nejmenší. Samotná parametrizace signálu je

prvním krokem v celém řetězci kroků, jež je nutné učinit pro realizaci rozpoznávání. Tyto operace jsou schématicky znázorněny na obr.3.1 a v následujícím textu budou ještě podrobněji vysvětleny.



Obr.3.1 Základní schéma postupu parametrizace řečového signálu

3.1.1. Digitalizace řečového signálu

Abychom byli schopni řečový signál zpracovávat v počítači, je nejprve nutné převést jej na jeho číslíkovou reprezentaci, tj. provést digitalizaci. Kvalitu digitalizovaného signálu nejvíce ovlivňují dva základní parametry. Těmito parametry jsou *kvantizační krok* a *vzorkovací frekvence*. Abychom určili tyto parametry, je třeba nejprve analyzovat lidskou řeč. Obecně platí, že čím vyšší je použitá vzorkovací frekvence a čím větší je rozlišení převodu (menší kvantizační krok => menší kvantizační chyba), tím jsou výsledky rozpoznávání lepší. V systémech pro identifikaci nebo verifikaci mluvěcích jsme ale většinou limitováni parametry přenosové cesty (rozpoznávaný signál je většinou přenášen po telefonní lince), a proto nemá smysl používat vzorkovací frekvenci 44 kHz, když přicházející signál je degradován přenosem po telefonní lince a má vzorkovací frekvenci pouze 8 kHz.

Frekvence lidského hlasu závisí na konfiguraci hlasového traktu (jakou hlásku mluví právě vyslovuje) a na konkrétním mluvěcím (věk, pohlaví, psychické rozpoložení atd.). Ve většině případů frekvence lidského hlasu nepřesahuje 3 kHz. Na základě experimentů s lidským hlasem bylo dokázáno, že naprostá většina informace v lidské řeči je obsažena ve frekvencích řádově stovek Hz u znělých hlásek a maximálně několika kHz u hlásek neznělých. Proto pro účely rozpoznávání je naprosto postačující vzorkovací frekvence 16 kHz, ale je možné úspěšně pracovat i s frekvencí 8 kHz. Abychom byli schopni pokrýt poměrně velkou dynamiku lidského hlasu (hodnota kolem 60 dB), měli bychom použít minimálně 12 bitové rozlišení. Z praktických důvodů se používá rozlišení 16 bitů, v případě telefonního signálu pak 8 bitů s nelineárním kvantizačním krokem.

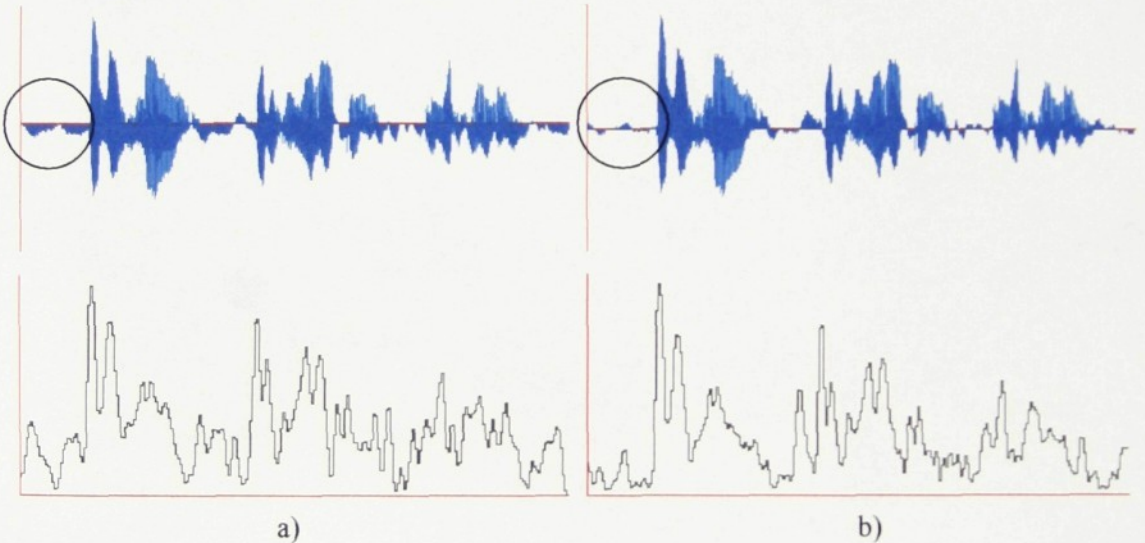
Do fáze digitalizace signálu je možné zařadit i *kalibraci signálu*. Kalibrace signálu je ve své podstatě kompenzací (odstraněním) stejnosměrné složky, která je závislá na hardwaru

používáním pro nahrávání signálu do počítače. Tato složka by nám potom mohla dělat problémy při rozpoznávání. Stejnosemnnou složku (3.1) odstraníme tak, že její vypočtenou hodnotu ze všech N vzorků odečteme od každého z nahrávaných vzorků.

$$ss = \frac{\sum_{n=1}^N vzorek(n)}{N} \quad (3.1)$$

Tato normalizace by také mohla probíhat přímo při nahrávání. Několik prvních *mikrosegmentů* (například čtyři) využijeme pro určení stejnosměrné složky zvukové karty a při nahrávání dalších vzorků je tato nežádoucí hodnota již odstraňována.

Na obr.3.2a je zobrazen řečový signál, jehož stejnosměrná složka má hodnotu -1080 při 16 bitovém kódování (přibližně 4% maximální hodnoty). Na časovém průběhu signálu je posun směrem dolů zcela zřetelný, což může být příčinou špatného rozpoznání začátku promluvy (pokud ji určujeme z intenzity signálu).



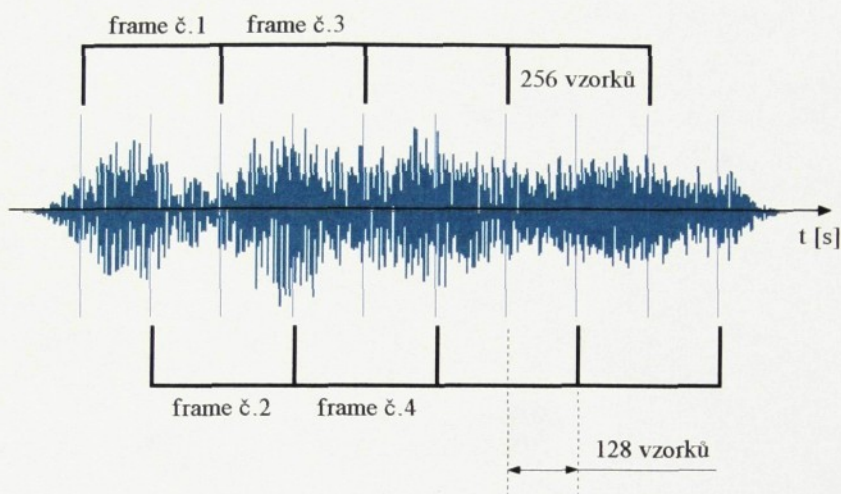
Obr.3.2 Řečový signál a jeho intenzita

a) před odstraněním stejnosměrné složky, b) po odstranění stejnosměrné složky

3.1.2. Segmentace signálu

Při parametrizaci je signál většinou popisován v jiné než časové oblasti. Obvykle se využívají charakteristiky signálu vycházející z frekvenční oblasti. Ve frekvenční oblasti lze signál při stejné kvalitě popsat menším počtem složek a dále se také snažíme o odstranění redundantní informace. Ve frekvenční oblasti lze totiž předpokládat, že parametry řečového signálu zůstávají v průběhu několika desítek milisekund téměř konstantní (to je způsobeno omezenou rychlostí přestavby hlasového traktu člověka). Vzhledem k velké dynamice řečového signálu se tyto části volí v rozmezí od 10 do 35 ms. Proto je tedy kontinuální řečový signál rozdělen do bloků (dále již „*frame*“) o velikosti N vzorků s tím, že každý další blok je

předsazen o M ($M < N$) vzorků. První frame je složen z prvních N vzorků. Druhý frame začíná M vzorků za prvním framem a přesahuje ho o $N-M$ vzorků. Stejně tak třetí frame začíná $2M$ vzorků za prvním (nebo M vzorků za druhým) a přesahuje jej o $N-2M$ vzorků viz. obr.3.3. Typicky je velikost N rovna 256 (to je 32 ms při 8 kHz vzorkovací frekvenci a navíc 256 je mocninou čísla 2 a s framem je tedy možné provádět rychlou variantu FFT) a M je rovna 128. Tyto hodnoty byly použity i v systémech vytvořených na TU v Liberci.



Obr.3.3 Segmentace řečového signálu

3.1.3. Preemfáze

Preemfáze je metoda sloužící ke zvýraznění vyšších kmitočtů v signálu, protože mají obvykle nižší úroveň. Většinou se realizuje pomocí jednoduchého číslicového filtru:

$$y(n) = x(n) - a \cdot x(n-1), \quad (3.2)$$

kde $y(n)$ je a -tý vzorek signálu po preemfázi a $x(n)$ je n -tý vzorek původního signálu. Konstanta a se běžně volí v rozsahu 0,95 – 0,98. Použití preemfáze není nutné, a tak ani v našem systému použita nebyla.

3.1.4. Aplikace okénka

Rozdělíme-li signál na framy, vzniká na jejich okrajích náhlý přechod (uříznutí), který je nežádoucí při dalším zpracování segmentů (frekvenční analýza). To, že odstraníme nespojitosti na začátku a konci framu, má za důsledek vyhlazení průběhu spektra. Aplikaci okénka konkrétně provádíme potlačením váhy vzorků na začátku a konci segmentu a to tak, že jednotlivé vzorky segmentu násobíme *váhou* nebo též *okénkovací funkcí*. Okénkovacích funkcí je celá řada, ale v oblasti zpracování řeči dominuje jediná a tou je Hammingova okénkovací funkce. Pokud si označíme funkci pro Hammingovo okénko jako $w(n)$, $0 \leq n \leq N-1$, kde N je opět počet vzorků v každém framu, tak funkce pro vážení jednotlivých framů bude mít tvar:

$$w(n) = 0,54 + 0,46 \cdot \cos \left[\left(\frac{1}{2} \cdot N - n \right) \frac{2 \cdot \pi}{N} \right] \quad (3.3)$$

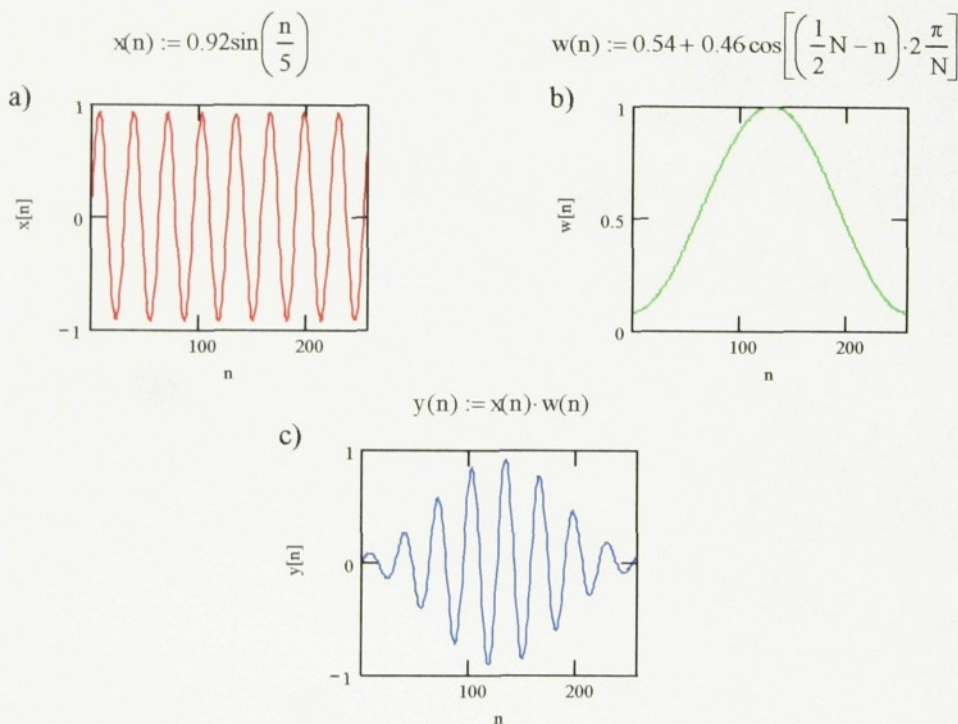
Použijeme-li tedy okénkovací funkci uvedenou ve vztahu (3.3), n -tý vzorek signálu vypočteme takto:

$$y(n) = x(n) \cdot w(n), \quad (3.4)$$

kde $x(n)$ je původní a $y(n)$ nově získaný vzorek signálu.

Na obr.3.4a je vykreslen jeden segment signálu před aplikací Hammingova okénka, na obr.3.4b je samotná váhová funkce a na obr.3.4c je segment již po vynásobení váhovou funkcí.

$$n := 1..256 \quad N := 256$$



Obr.3.4 Signál jednoho framu před a po aplikaci Hammingova okénka

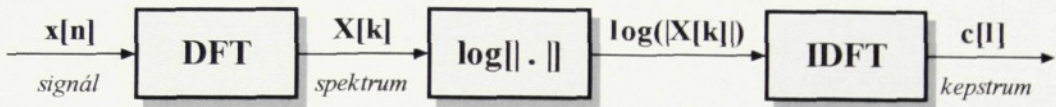
3.1.5. Výpočet statických příznaků

V mnoha studiích bylo prokázáno, že nejvhodnějšími příznaky pro aplikace automatického rozpoznávání mluvího jsou keprální koeficienty. Dále bylo prokázáno, že keprální příznaky jsou velmi dobře použitelné v textově závislé i nezávislé identifikaci, pokud byly nahrávky pořízeny za relativně vhodných podmínek (dobrý odstup signálu od šumu, stacionární podmínky při nahrávání).

Kepstrální analýzu používáme k oddělování složek signálu, který vznikl konvolucí několika složek (řečový signál je výsledkem konvoluce buzení hlasového traktu s impulzní

odezvou hlasového ústrojí). Obecně se při výpočtu kepstra postupuje podle vztahu (3.5), kde kepstrum $c[l]$ vypočítané ze vstupní posloupnosti vzorků signálu $x[n]$ je definováno jako inverzní Fourierova transformace logaritmu absolutní hodnoty spektra signálu. Tento postup je naznačen i v blokové verzi viz. obr.3.5.

$$c[l] = DFT^{-1} \{ \log(|DFT \{x[n]\}|) \} \quad (3.5)$$

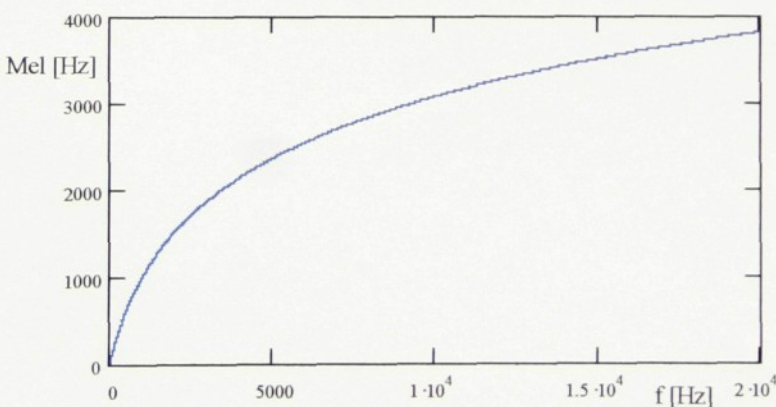


Obr.3.5 Výpočet kepstra signálu

MFCC kepstrální příznaky

Pro výpočet kepstrálních příznaků existuje mnoho postupů. Pro náš systém jsme zvolili *MFCC kepstrum* (MFCC - Mel-Frequency Cepstral Coefficient), jež je v současných aplikacích hojně využívanou metodou výpočtu kepstrálních koeficientů. Důvodem velmi dobré výkonnosti příznaků vypočítaných touto metodou je věrohodná aproximace charakteristiky lidského ucha, které při vnímání řeči sleduje logaritmickou (Melovskou) stupnici. Melovská stupnice je zobrazena na obr.3.6. Tuto stupnici můžeme v rozmezí 0 až 1000 Hz považovat za přibližně lineární a v mezích nad 1000 Hz ji lze nahradit logaritmickým průběhem. Převodní vztah, který slouží k převodu mezi frekvencí a mel-frekvencí, je následující:

$$Mel(f) = 2595 \cdot \log_{10} \left(1 + \frac{f}{700} \right) \quad (3.6)$$



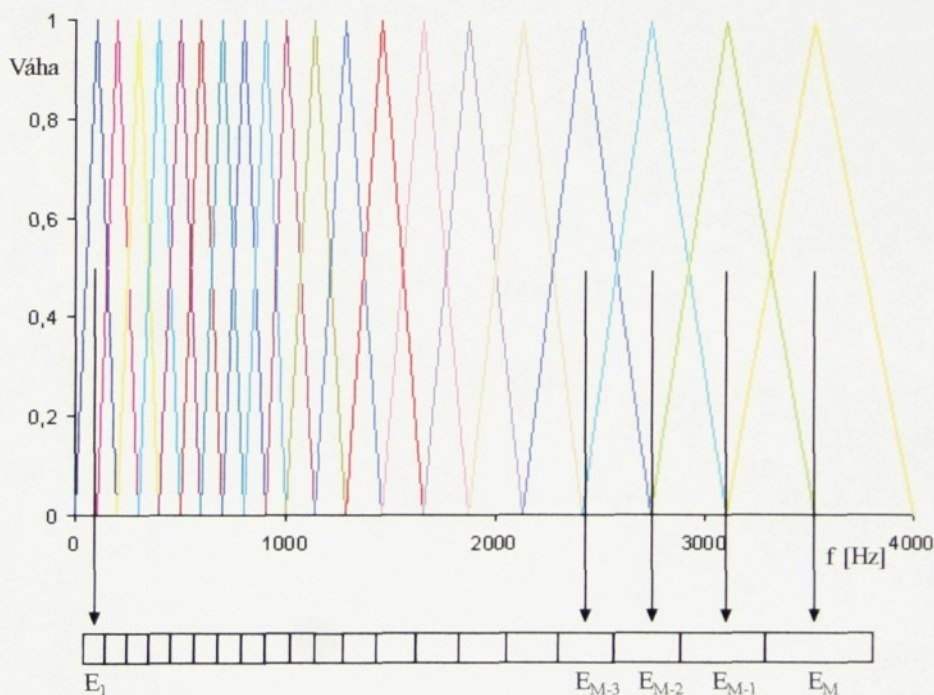
Obr.3.6 Graf převodní funkce mezi mel-frekvencí a frekvencí

V praxi provádíme výpočet MFCC kepstrálních příznaků tímto způsobem:

- Máme-li již digitalizovaný signál (viz. předchozí kapitoly), převedeme jej vhodným způsobem do frekvenční oblasti. Nejvhodnějším způsobem výpočtu amplitudového

spektra signálu je algoritmus rychlé *diskrétní Fourierovy transformace* (DFT – discrete Fourier transformation). Jak již bylo dříve uvedeno, spektrum signálu počítáme z jednotlivých framů. Nyní můžeme využít toho, že velikost framu je 256 vzorků (mocnina dvou) a použít rychlou (klasickou) variantu FFT algoritmu. Více informací o algoritmu FFT je možné se dozvědět v [6]. Výstupem FFT je stejný počet vzorků jako na vstupu tj. 256, ale využitím souměrnosti spektra okolo $F_s/2$ bylo možné po provedení normalizace použít pouze prvních 128 vzorků (jednostranné spektrum).

- Dalším krokem ve výpočtu kepra je převod amplitudového spektra na mel-spektrum. Počet koeficientů mel-spektra se obvykle volí okolo 20. Jeden koeficient mel-spektra obdržíme tak, že všechny hodnoty spektra filtrujeme (vážíme) bankou M trojúhelníkových filtrů (průběhy hodnot jednotlivých filtrů jsou uvedeny na obr.3.7). Zjednodušeně jde o 20 pásmových propustí. Pro jednotlivé filtry všechny vážené vzorky sečteme a dostaneme tak energie signálu v každém pásmu filtru.



Obr.3.7 Melovská banka filtrů

Uspořádání filtrů v uvedeném Melovském měřítku ovlivňuje způsob jejich rozestavení na frekvenční ose. Rozestupy mezi *centrálními frekvencemi* Δ_m a šířky pásem filtrů jsou do frekvence 1 KHz konstantní, počet filtrů v této lineární oblasti volíme poloviční proti počtu všech filtrů v celém frekvenčním pásmu (v našem případě je to 10 filtrů). Směrem k vyšším frekvencím se rozestupy mezi jednotlivými filtry zvětšují, stejně tak se zvětšují i šířky pásem (toto je ilustrováno na obr.3.7). Rozestupy se zvětšují tak, že $\Delta_m = 1,2 \cdot \Delta_{m-1}$ a centrální frekvence $b_m = b_{m-1} + \Delta_m$. Výpočet energie spektra v m -tém pásmu je tedy:

$$E(m) = \sum_{k=b_m-\Delta_m}^{b_m+\Delta_m} X(k) \cdot U_{\Delta m}(k+b_m), \quad (3.7)$$

kde $X(k)$ je k -tý vzorek spektra signálu framu získaný pomocí FFT a $U_{\Delta m}(k)$ je váhová funkce trojúhelníkového filtru získaná výpočtem ze vztahu:

$$U_{\Delta m}(k) = \begin{cases} |k| < \Delta_m \rightarrow 1 - \frac{|k|}{\Delta_m} \\ |k| \geq \Delta_m \rightarrow 0 \end{cases} \quad (3.8)$$

- Výsledek ve formě vektoru energie ($E_1, E_2, E_3, \dots, E_M$) je dále transformován ve vektor MFCC keprstrálních příznaků. Tato transformace je uskutečněna *diskrétní kosinovou transformací* (DCT – Discrete Cosine Transformation) logaritmu vektoru energie mel-spektra. Vztah pro výpočet l -tého keprstrálního koeficientu je uveden zde:

$$c(l) = \sum_{m=1}^M \log(E_m) \cdot \cos \left[l \cdot \left(m - \frac{1}{2} \right) \cdot \frac{\pi}{M} \right] \quad (3.9)$$

Počet vypočítaných keprstrálních koeficientů si můžeme zvolit, ale nikdy nesmí překročit počet energií mel-spektra M . V našem systému z 20 filtrů vypočítáme 8 MFCC keprstrálních koeficientů. Bylo ověřeno, že první člen z DCT reprezentuje pouze průměrnou hodnotu vstupního signálu a nenese skoro žádnou informaci, která by nám pomohla při rozpoznávání. Proto se tento první koeficient nepoužívá. Aplikací DCT vlastně převádíme signál zpět do časové oblasti. Pro převod zparametrizovaného signálu zpět do časové oblasti lze použít DCT díky tomu, že transformované spektrum je reálné a symetrické.

Takto vypočítaná keprstrální reprezentace signálu nám dává dobrou informaci o lokálních spektrálních vlastnostech tohoto signálu, a tím i o identitě mluvího.

3.1.6. Výpočet dynamických příznaků

Dynamickými příznaky rozumíme derivace statických příznaků (o nichž pojednávala předchozí kapitola) v čase, tzn. že dynamické příznaky vyjadřují změnu statických příznaků v čase. V reálných systémech pro rozpoznávání mluvího jsou společně se statickými příznaky používány dynamické příznaky prvního (tzv. *delta příznaky*) nebo též i druhého řádu (tzv. *delta-delta příznaky* nebo také *akcelerační příznaky*).

V systémech rozpoznávajících řeč platí, že některé dynamické příznaky mohou mít vyšší vypovídající schopnost než jejich statické varianty, viz např.[11]. V experimentech, které budou následovat v další kapitole, se budeme snažit dokázat platnost této skutečnosti i v systémech pro rozpoznávání mluvího.

Dynamické příznaky počítáme pomocí numerické derivace průběhu statických příznaků v čase. Jelikož se pro výpočet dynamických příznaků používají různé konfigurace vstupních

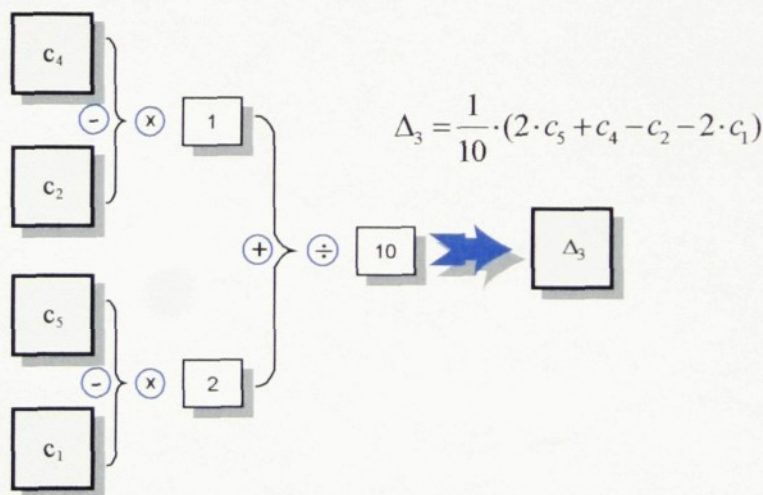
statických příznaků (od prosté difference až po složité vícebodové metody), tak v této podkapitole bude uvedena obecná metoda, jak tyto dynamické příznaky vypočítat. V následujícím textu má $x(n)$ význam vstupu a $y(n)$ je výstupem, tj. derivací tohoto vstupu.

$$y(n) = \frac{\sum_{i=1}^W i \cdot [x(n+i) - x(n-i)]}{2 \cdot \sum_{i=1}^W i^2} \quad (3.10)$$

Z regresního vztahu (3.10) je možné vypočítat libovolné derivace vstupu určením požadované délky okna W (udáme jeho poloměr). Poloměrem okna rozumíme počet koeficientů na každou stranu od aktuální pozice, již je index vypočítávaného příznaku. Položíme-li W rovno jedné, dostaneme vztah pro jednoduchou nekauzální diferenci:

$$y(n) = \frac{x(n+1) - x(n-1)}{2} \quad (3.11)$$

Tento vztah lze také použít, ale pokud chceme obdržet vyhlazenější průběh derivace, je lepší za W zvolit číslo větší než jedna. Vztah (3.10) je graficky znázorněn na obr.3.8. Pokud chceme vypočítat akcelerační příznaky, tak postup bude naprosto totožný pouze s tím rozdílem, že jako vstup budeme brát delta příznaky a výstupem budou delta-delta příznaky. U obou systémů, které jsou v této práci prezentovány, byly použity dynamické příznaky s délkou okna 2. V experimentální části bude vyzkoušena i jiná varianta poloměru okna a dosažené výsledky budou porovnány.



Obr.3.8 Výpočet koeficientu Δ_3 dvoubodovou metodou

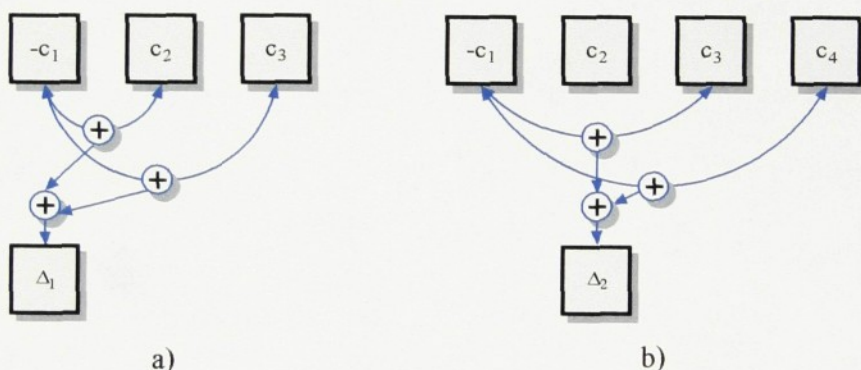
Při praktické realizaci algoritmu pro výpočet dynamických příznaků je třeba dát pozor na okrajové podmínky (především počítáme-li difference z více bodů), které nastávají na okrajích zpracovávaného signálu. Tyto okrajové podmínky je nutné rozumným způsobem ošetřit. Pracujeme-li (tak jako v našich systémech) s velikostí okna 2, jeden ze způsobů, kterým můžeme okrajové podmínky ošetřit, je pro začátek signálu uveden vztahem:

$$y(n) = \frac{\sum_{i=1}^2 i \cdot [x(n+i) - x(0)]}{2 \cdot \sum_{i=1}^2 i^2}, \quad n < W \quad (3.12)$$

Pro opačnou stranu signálu platí vztah (3.13), který je velmi podobný předchozímu vztahu (3.12), jen s tím rozdílem, že se prohodilo pořadí krajních podmínek. N je počet vstupních vektorů a W je délka okna.

$$y(n) = \frac{\sum_{i=1}^2 i \cdot [x(n-1) - x(n-i)]}{2 \cdot \sum_{i=1}^2 i^2}, \quad n \geq N - W \quad (3.13)$$

Praktický příklad výpočtu prvního a druhého příznaku (opět je délka okna 2) je na obr.3.9. Naznačeno je pouze to, jakým způsobem jsou k sobě vstupní příznaky přidruženy.



Obr.3.9 Způsob výpočtu okrajových dynamických koeficientů
a) první delta koeficient b) druhý delta koeficient

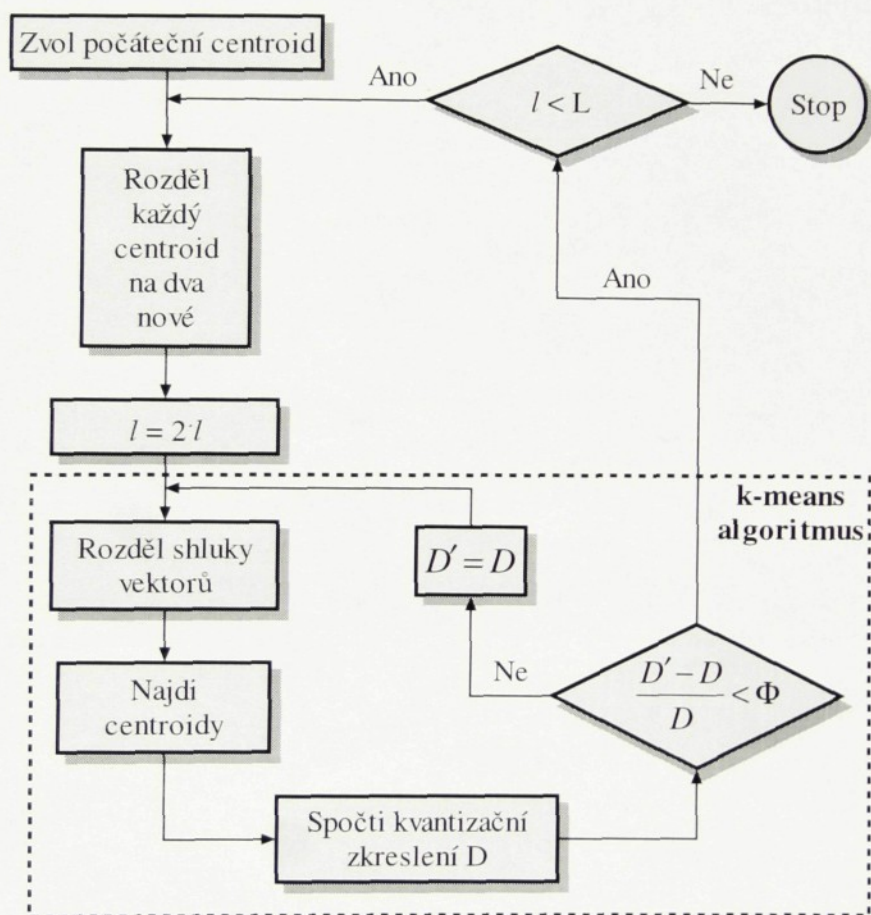
3.2. Návrh a realizace systému založeného na metodě VQ

V této podkapitole je podrobněji vysvětlen princip činnosti, realizace a způsob testování textově nezávislého systému pro rozpoznávání mluvěcích, který byl implementován a ověřován na TU v Liberci. Tento systém je postaven na základě vektorového kvantování.

3.2.1. Režim trénování - vytvoření kódové knihy (LBG algoritmus)

Na rozdíl od textově závislého rozpoznávače je nutné u VQ ještě před procesem testování projít trénovacím režimem, tedy fází, v níž se pro každého mluvěcího vytvoří kódové knihy. Jelikož inicializační kódová kniha (základní odhad, ze kterého vycházejí další iterace) má bezprostřední vliv na kvalitu výsledné kódové knihy, došlo časem k vylepšení *k-means algoritmu* [12], který byl do té doby používán pro tvorbu kódových knih. Bylo dokázáno [13], že přístup, ve kterém je L -rozměrná kódová kniha konstruována po částech, je globálně

optimální (k-means algoritmus může konvergovat pouze k lokálnímu optimu). Tento vylepšený k-means algoritmus dostal jméno po svých tvůrcích a nazývá se *LBG* (Linde, Buzo a Gray [14]). Algoritmus je implementován jako rekursivní procedura znázorněná vývojovým diagramem na obr.3.10. Velikost výsledné kódové knihy vypočítané tímto algoritmem je mocninou čísla 2. Tato skutečnost vyplývá z již zmíněného principu práce LBG algoritmu, který výslednou kódovou knihu vytváří postupným zdvojnásobáním předchozích kódových knih. Tyto knihy mohou mít různé velikosti a právě na počtu elementů do značné míry závisí výsledek i rychlost rozpoznávání.



Obr.3.10 Vývojový diagram LBG algoritmu pro tvorbu kódové knihy

LBG algoritmus pracuje následujícím způsobem :

Krok 1: Vytvoření kódové knihy pouze s jedním kódovým slovem

To je centroid z úplné sady trénovacích vektorů (zde není nutná žádná iterace, pouze se nalezne vektor, který nejlépe reprezentuje všechny vektory - průměr).

Krok 2: Zdvojnásobení kódové knihy rozdělením současných shluků a vytvořením nových kódových slov (centroidů) podle pravidla:

$$y_n^+ = y_n * (I + \Omega)$$

$$y_n^- = y_n * (I - \Omega),$$

kde y_n je n -tý centroid z kódové knihy vzniklé v předešlé iteraci,

y_n^+ a y_n^- jsou nově určené centroidy ($n = 1, \dots, l$),

Ω je vektor iteračních parametrů (obvykle 0,01 , 0,001).

Krok 3: Hledání nejbližšího souseda:

Pro každý trénovací vektor najdi kódové slovo v současné kódové knize, které má nejmenší kvantizační zkreslení, a přiřaď tento vektor shluku asociovanému s nejbližším kódovým slovem.

Krok 4: Aktualizace kódového slova:

Vypočti kódové slovo v každém shluku jako průměr z trénovacích vektorů asociovaných k tomuto shluku.

Krok 5: Iterace 1:

Urči celkovou kvantizační chybu D (přes všechny trénovací vektory) a porovnej ji s chybou v předchozím kroku D' , opakuj kroky 3 a 4, pokud je relativní změna chyby větší než parametr Φ .

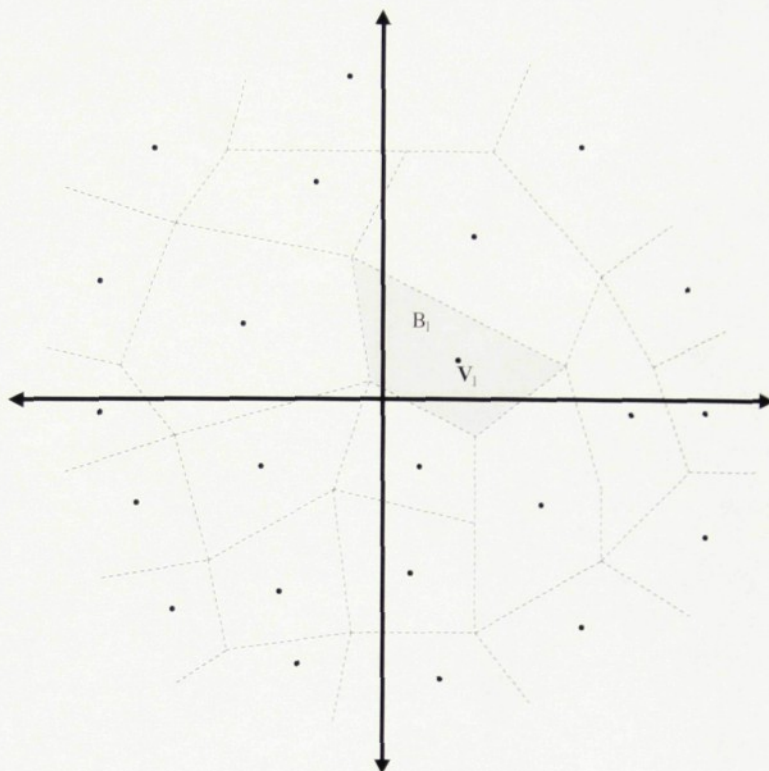
Krok 6: Iterace 2:

Opakuj kroky 2, 3 a 4 tvořením l -rozměrných kódových knih, dokud nevznikne kódová kniha s L kódovými slovy.

LBG algoritmus vytváří L -rozměrnou kódovou knihu postupně. Začíná vytvořením 1-rozměrné kódové knihy, pak použije rozdělovací techniku pro inicializaci 2-rozměrné kódové knihy, atd. V rozdělovacím procesu se pokračuje tak dlouho, dokud není vytvořena požadovaná L -rozměrná kódová kniha. V každém iteračním kroku se určuje celková kvantizační chyba a sleduje se její konvergence. Kroky 3, 4 a 5 tvoří dohromady k-means algoritmus, který zajišťuje optimální rozdělení P -dimenzionálního prostoru do L buněk. Toto optimum je bráno jako minimální celková distorze všech vstupních vektorů. Ukázka rozdělení dvojdimenzionálního prostoru do 23 buněk je na obr.3.11. Je-li prostor takto rozdělen, VQ provádí tvrdé přiřazení (rozdělení) vstupního vektoru $\mathbf{X}$ kódovému slovu $\mathbf{V}_l$, pokud $\mathbf{X}$ leží v buňce B_l . Platí tedy vztah :

$$\mathbf{V}_l = q(\mathbf{X}), \quad \text{pokud } \mathbf{X} \in B_l \quad (3.14)$$

Tvary a velikosti buněk mohou být různé.



Obr.3.11 Rozdělení dvojrozměrného prostoru do 23 buněk

3.2.2. Režim rozpoznávání

Cílem rozpoznávání je určení identity mluvčího. VQ toho dosahuje tak, že určuje kvantizační chybu, což je vzdálenost porovnávaného vektoru příznaků k nejbližšímu kódovému slovu kódové knihy. Ve fázi rozpoznávání je tedy vstupní promluva vektorově kvantována za použití všech vektorů z dané kódové knihy. Kvantizační chyby se pro všechny vstupní vektory sečtou a výsledkem je celková kvantizační chyba. Ten, kdo má se svou kódovou knihou nejmenší vzdálenost (kvantizační chybu) od zparametrizované testované promluvy, je prohlášen za testovaného mluvčího, viz. vztah (2.3). Na obr.3.12 je tento postup naznačen i graficky.

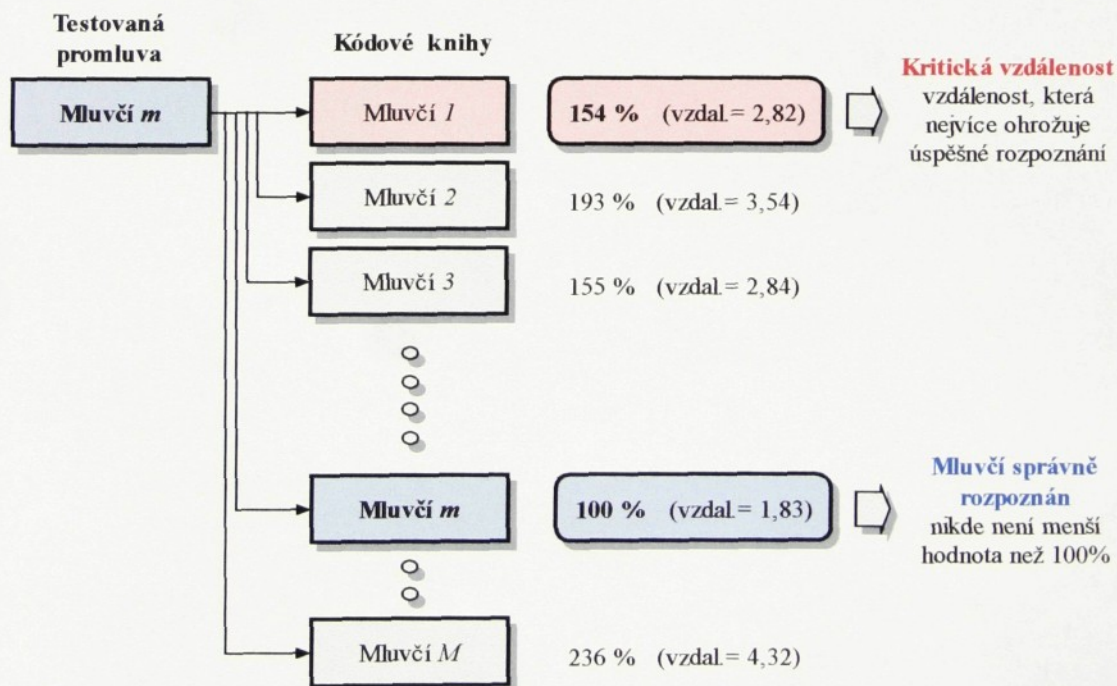


Obr.3.12 Postup porovnávání neznámého mluvčího s jednou KK

Jak tedy probíhá vlastní rozpoznávání?

V reálném životě by náš systém pro identifikování totožnosti mluvčího pracoval následujícím způsobem :

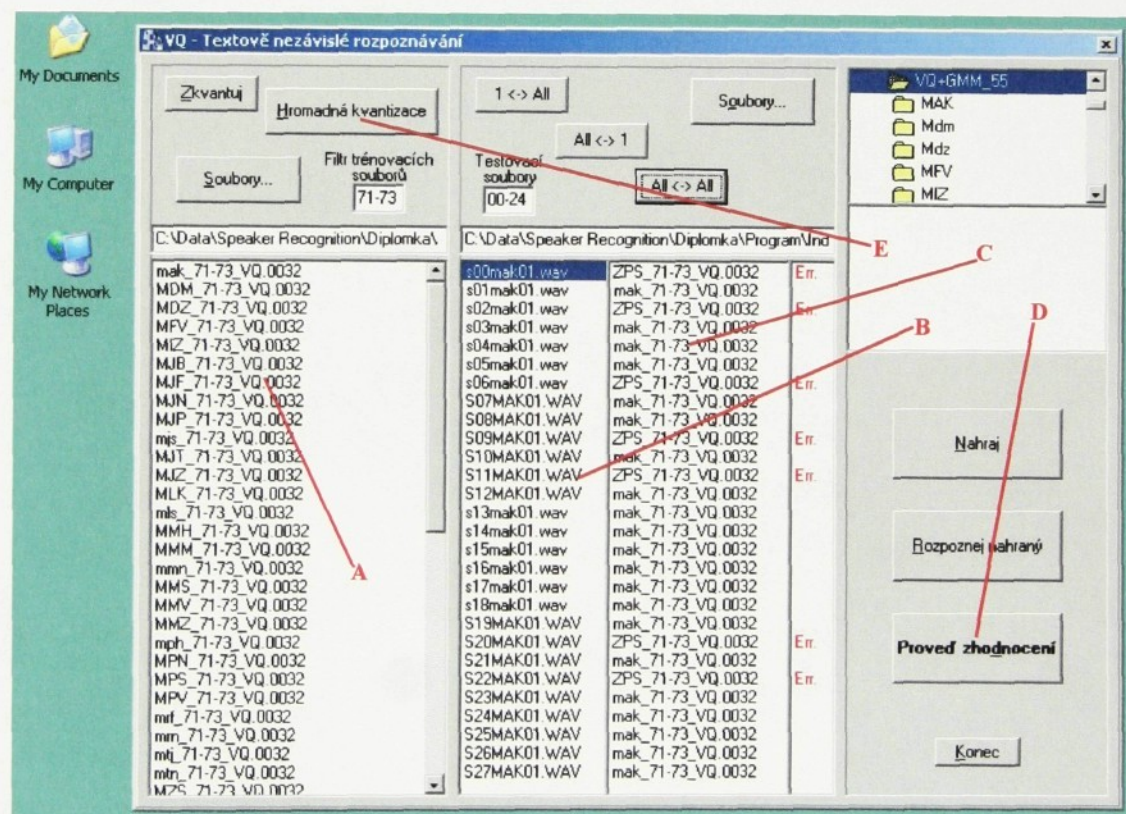
0. Na začátku je potřeba vytvořit databázi mluvčích, kteří budou identifikační systém (dále již jen IS) používat. Tito mluvčí budou reprezentováni kódovými knihami, které vytvoříme postupem uvedeným v kapitole 3.2.1. Takto vytvořené kódové knihy budou uloženy v IS a používány při identifikaci.
-
1. Osoba, která chce být identifikována, přistoupí k našemu systému pro identifikaci mluvčího a vysloví jakoukoliv promluvu o délce 5 až 10 sekund.
 2. IS v části určené k extrakci příznaků tuto promluvu převede na posloupnost vektorů příznaků.
 3. IS tyto příznakové vektory porovná s první kódovou knihou, kterou má uloženou ve své databázi. Výsledkem porovnání bude vzdálenost, která charakterizuje podobnost testované promluvy s kódovou knihou (čím menší je vzdálenost, tím si jsou promluvy podobnější).
 4. Krok 3 opakujeme tak dlouho, dokud nejsou vypočítány vzdálenosti testované promluvy proti všem kódovým knihám v databázi (viz. obr.3.13). Jediné, co je třeba si zapamatovat, je vzdálenost a index kódové knihy, ke které tato vzdálenost náleží.



Obr.3.13 Rozpoznávání proti celé databázi mluvčích

- IS nalezne nejmenší vzdálenost. Index spojený s touto vzdáleností ukazuje na konkrétního mluvčího, jehož hlasem byla kódová kniha natrénována. Tento mluvčí je určen jako identifikovaná osoba.

Pro účely testování výkonnosti systému je ovšem tento přístup nepřijatelný. Při testování systému (různé konfigurace vektorů příznaků, různé velikosti KK, různé trénovací nahrávky, atd.) vyžadujeme neměnná vstupní data, to proto, abychom mohli srovnávat výsledky jednotlivých testů. Proto bod 1, kdy jsme získávali testovanou promluvu přímo od mluvčího, nahradíme nahráním testované promluvy ze souboru uloženého v počítači. Tyto testovací nahrávky byly nahrány při trénování systému. V počítači jsou uloženy v souborech typu „wav“ (to, že nahrávky nejsou „živé“, systém nijak neovlivní). Další postup testování je již naprosto totožný s postupem u živého mluvčího. Jak vypadá tento systém z uživatelského pohledu je ukázáno na obr.3.14.



Obr.3.14 Pohled na program v konfiguraci pro hromadné testování promluv uložených v počítači na pevném disku

Databáze kódových knih jsou na obrázku označeny jako (A), konkrétně se jedná o kódové knihy s 32 kódovými slovy (určeny příponou souborů *.0032). V ListBoxu (B) jsou nahrány všechny testované promluvy a v (C) jsou k nim určeni rozpoznání mluvčí. Napravo od tohoto ListBoxu jsou vyhodnocovány názvy rozpoznávaných mluvčích a pokud si názvy

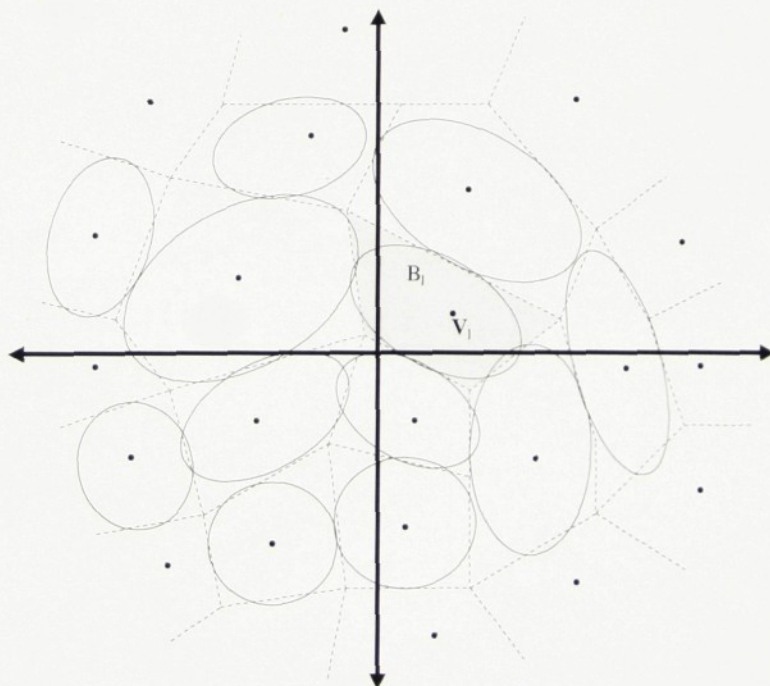
neodpovídají, je vypsána chyba. Tlačítko (D) je určeno ke spuštění hromadného testování, tj. jsou testovány všechny promluvy ze zadané cesty a výstup je uložen do souboru pro další zpracování výsledků. A konečně tlačítko (E) spouští režim trénování pro zvolené soubory, tj. vytváříme soubory, které mohou být načteny do (A).

3.3. Návrh a realizace systému založeného na metodě GMM

Tato podkapitola se zabývá principem činnosti, realizací a způsobem testování GMM systému pro rozpoznávání mluvčích, který byl implementován a ověřován na TU v Liberci. Teoretický základ metody GMM je uveden v podkapitole 2.2.2. Tento systém má základy ve vektorovém kvantování, jež používá pro tvorbu počátečních odhadů svých modelů.

3.3.1. Režim trénování - vytvoření modelu mluvčího (EM algoritmus)

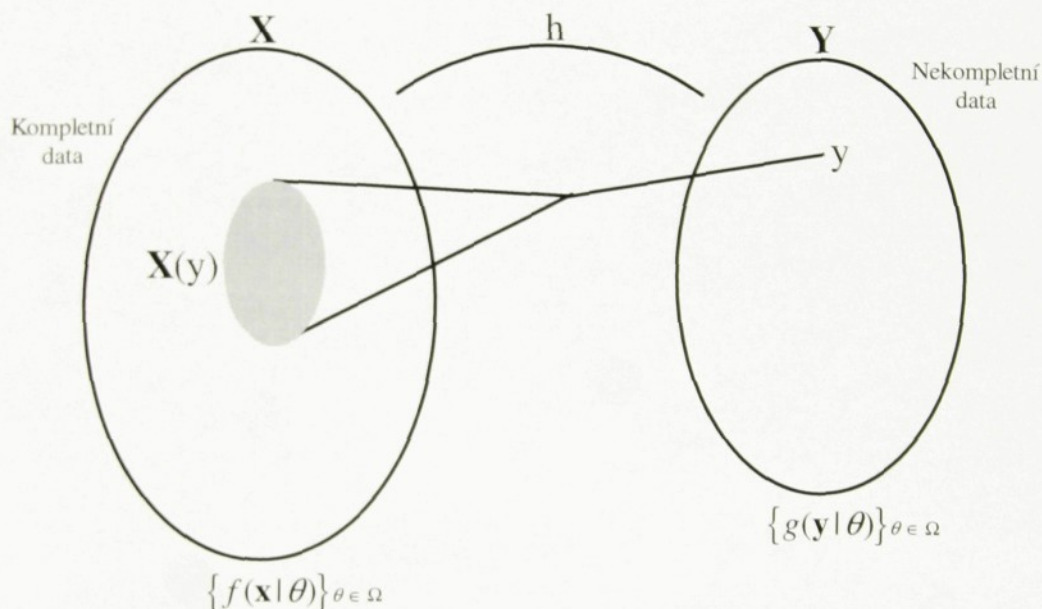
Před vlastním vysvětlením principů EM algoritmu by bylo zřejmě vhodné ujasnit si ještě jednou, jakým způsobem GMM reprezentují data v prostoru. VQ má bohužel tu vlastnost, že rozděluje data v prostoru do určitých regionů podle měření určité vzdálenosti, ale nerespektuje již pravděpodobnostní rozdělení vstupních dat. To může zapříčinit chybné vymezení hranic a následně i poškození originální struktury dat. Alternativní cestou jak nahradit toto „tvrdé“ přiřazení jsou gaussovska pravděpodobnostní rozložení. Ukázka takového rozložení v dvojrozměrném prostoru je na obr.3.15.



Obr.3.15 Rozdělení dvojrozměrného prostoru 16 gaussovskými hustotami pravděpodobnosti

Pro tyto funkce již není problém překročit hranice rozdělující prostor a mohou tak lépe reprezentovat strukturu dat. Gaussovske pravděpodobnostní hustotní funkce (PDF – probability density function) jsou funkcemi dvou parametrů. Je to střední hodnota a matice kovariancí. A nyní již k samotnému EM algoritmu.

Při tvorbě modelu se potýkáme s jedním zásadním problémem. Chceme zjistit jeho parametry, ale nemáme k dispozici veškerá data.



Obr.3.16 Tvorba modelu z nekompletních dat

Na obr.3.16 je ilustrován hypotetický příklad, kde jsou dva prostory (X a Y) mapovány funkcí $h : X \rightarrow Y$.

- Přiřazení $\mathbf{x} \in X$ implikuje, že $\mathbf{y} = h(\mathbf{x}) \in Y$.
- Bohužel pouze $\mathbf{y} = h(\mathbf{x})$ je nám známo. Víme jen, že $\mathbf{x} \in X(\mathbf{y})$.

Princip činnosti EM algoritmu

Mějme populační hustotu $p(\mathbf{x} | \lambda)$, kde λ jsou parametry které určují rozdělení. My se snažíme o modelování této hustoty určením parametrů λ . To znamená, že p může být množina gaussovských funkcí a λ budou jejich střední hodnoty a kovariance. Vzorek dat má velikost N . L je věrohodnost parametrů daných daty. Předpokládejme vektory dat s rozdělením p . A nyní si představme funkci, ve které parametrů λ , ve které jsou data X konstantní. My tedy chceme nalézt λ takové, aby maximalizovalo L , nebo (a to častěji) $\log(L)$. A právě toto iterativně vykonává EM algoritmus. Pokud si shrneme poznatky o EM algoritmu, tak je to metoda jak nalézt nejpravděpodobnější odhad parametrů λ z dostupných, ale neúplných vzorků dat.

EM algoritmus začíná odhadem parametrů λ_i , poté iteračním způsobem hledáme optimální parametry λ . λ_i jsou vždy odhadnuté parametry, λ jsou optimální parametry.

Pro iterace k a $k+1$ platí: $p(\mathbf{X} | \lambda^{(k+1)}) > p(\mathbf{X} | \lambda^{(k)})$. Pro konvergenci parametrů obecně postačuje pět až deset iterací.

EM algoritmus pracuje následujícím způsobem :
Krok 1: Inicializace Zvol počáteční odhad λ. $\rightarrow$ V naší implementaci EM algoritmu jsme k volbě počátečního odhadu používali kódová slova z vektorové kvantizace. To znamená LBG algoritmus. Ten má o něco lepší výsledky než běžně používaný k-means algoritmus.
Krok 2: „Expectation“ Vypočítej pomocnou Q-funkci $Q(\lambda, \bar{\lambda})$, což je vlastně také „expectation“ logaritmické věrohodnosti získané z kompletních dat. Tato funkce je závislá na $\bar{\lambda}$
Krok 3: „Maximalization“ Vypočítej $\hat{\lambda} = \underset{\lambda}{\text{ArgMax}} Q(\lambda, \bar{\lambda})$, tak budeme maximalizovat pomocnou funkci Q
Krok 4: Iterace $\lambda = \hat{\lambda}$, opakuj od kroku 2 do konvergence

V tomto algoritmu jsou tyto proměnné:

- λ vektor parametrů, který maximalizuje počáteční odhad,
- $\bar{\lambda}$ nový odhad, vypočtený z λ ,
- $\hat{\lambda}$ maximalizuje $p(Y = y | \lambda)$,
- y reálná data,
- x skrytá data.

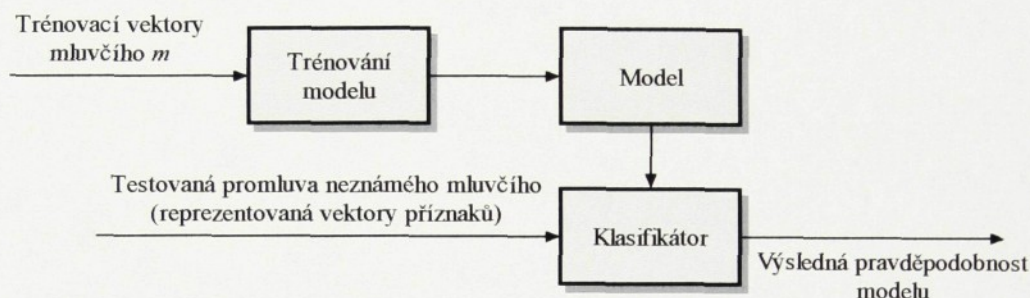
Protože vektory příznaků $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$ pokládáme za nezávislé, logaritmická věrohodnost modelu λ se vypočte jako:

$$\log p(\mathbf{X} | \lambda) = \sum_{t=1}^T \log p(\mathbf{x}_t | \lambda) \quad (3.15)$$

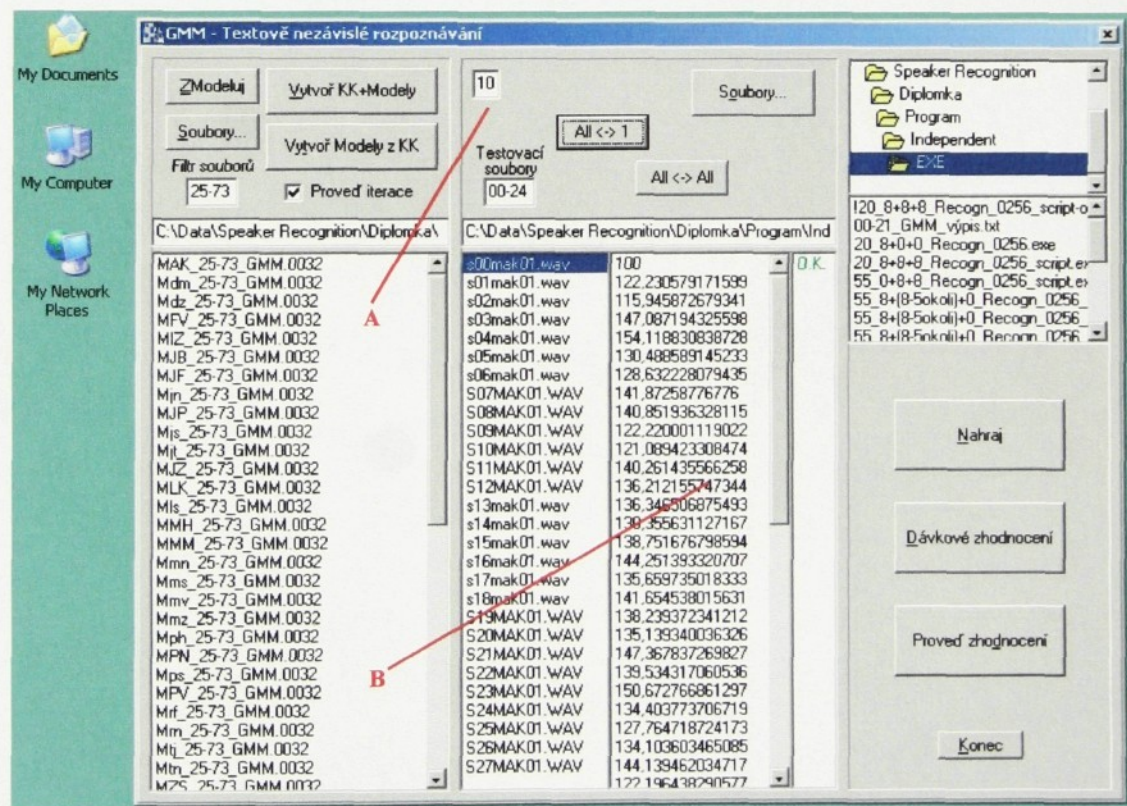
Velmi často se provádí normalizace této věrohodnosti vydělením počtem příznakových vektorů T [12].

3.3.2. Režim rozpoznávání

Z vnějšího pohledu je přístup k identifikaci mluvčího pomocí gaussovských mixturových modelů totožný s přístupem, který používáme u systému pracujícího s vektorovou kvantizací. Na obr.3.17 je upravená verze obr.3.12. Schémata se liší pouze v druhu používaného modelu a ve výstupu klasifikátoru. Výstupem GMM klasifikátoru je pravděpodobnost modelu, u VQ je výstupem celkové kvantizační zkreslení.



Obr.3.17 Postup porovnávání neznámého mluvčího s jedním modelem GMM systému



Obr.3.18 Uživatelské rozhraní GMM systému

Vítězem v procesu rozpoznávání je mluvčí s největší pravděpodobností, jinak postup testování proti databázi je naprosto shodný s VQ. Na obr.3.13 je tento postup naznačen graficky. Vítěz rozpoznávání se určí podle vztahu (2.6). Uživatelské rozhraní GMM systému je zobrazeno na obr.3.18. Je opět shodné s uživatelským rozhraním VQ, pouze přibyl EditBox (A). V tomto EditBoxu určujeme maximální počet iterací EM algoritmu. Položka (B) ukazuje na příklad výpisu porovnávání, kde hodnoty v ListBoxu značí kritické pravděpodobnosti (pojem byl odvozen od kritických vzdáleností).

4. EXPERIMENTY S ROZPOZNÁVÁNÍM MLUVČÍCH

4.1. Databáze promluv

Protože stěžejním úkolem této práce byla realizace skutečných systémů pro IM, bylo žádoucí navrhnout a vytvořit databázi záznamů promluv vhodnou pro trénování a testování těchto systémů.

Nakonec byly při testování systémů použity dvě databáze promluv.

- Pro experimenty s oběma systémy (VQ a GMM) byla vytvořena databáze skládající se z promluv 20 mluvěch - 10 mužů a 10 žen. Všichni byli přibližně stejného věku (od 22 do 26 let), což ve své podstatě ztížilo úlohu obou rozpoznávačů. Pokud by například mezi nahrávky bylo zařazeno několik dětských mluvěch, pak by se úspěšnost rozpoznávání o něco zvýšila. Důvodem by byly velké odlišnosti základní frekvence dětského hlasu od základní frekvence hlasu dospělých lidí. Tato databáze byla použita pouze pro testování robustnosti námi navrženého systému. Pro testování robustnosti systémů musela databáze splňovat dvě základní podmínky. První podmínkou bylo nahrávání promluv s časovým odstupem alespoň jednoho roku, druhou podmínkou byla změna nahrávacího hardwaru (jiná zvuková karta anebo alespoň výměna mikrofonu). Bohužel u druhé databáze tyto podmínky splněny nebyly (databáze byla stará několik let a my jsme již neměli přístup k mluvěch), a proto vznikly tyto nahrávky.

Při nahrávání byly použity náhodně vybrané věty z denního tisku. V průměru měla jedna věta přibližně devět slov. Ve větách se vyskytovala převážně česká slova a žádná věta nebyla nikdy použita dvakrát.

První série vět byla nahrána zhruba před rokem a obsahovala 40 vět od každého mluvěho:

- a) 20 vět ... použito jako trénovací věty pro tvorbu kódových knih (v souhrnu 400 testovacích promluv)
- b) 20 vět ... použito pro testování (získání referenčních hodnot úspěšnosti rozpoznávání)

Druhá série nahrávek byla pořízena asi před měsícem. Nahrávání se zúčastnilo dvanáct z původních dvaceti mluvěch (ostatní mluvě již nebyli dostupní). Nahráváno bylo pouze dvacet vět, a to ty věty, které byly použity pro testování. Jednalo se tedy o naprosto identické věty jako v první sérii. Aby testování robustnosti bylo co nejvěrnější, potřebovali jsme stejné délky promluv u starých i u nových nahrávek (úspěšnost rozpoznávání měla být ovlivněna pouze časovým odstupem a jiným nahrávacím hardwarem, ne délkou promluv). Nejjednodušší způsob, jak toho dosáhnout, bylo nahrávání stejných vět. Takže v druhé sérii bylo nahráno:

c) 20 vět ... věty určené pro testování, stejné jako b)

Všechny věty byly pořizeny v podstatě v laboratorních podmínkách. To znamená, že nahrávání probíhalo za statických podmínek (bez vnějšího akustického rušení). Dále byl při nahrávání použit jiný mikrofon a jiná zvuková karta. Signál byl ukládán do souboru formátu wav, a to v 16 bitovém kódování, vzorkovací frekvence byla 8 kHz a signál byl uložen v jednom mono kanálu.

- Pro provádění všech ostatních experimentů byla využita druhá databáze.

Tato data jsou podmnožinou databáze sloužící k experimentům s rozpoznáváním řeči v češtině. Proto také všichni mluvčí vyslovovali stejné věty. Jednotlivé věty byly navíc vybrány tak, aby byly pokud možno co nejbohatší na množství obsažených fonémů. Databáze se skládala z celkového počtu 55 mluvčích - 29 mužů a 26 žen z různých věkových skupin. Každý z mluvčích pronesl celkem 74 vět:

a) 25 vět ... určeny pro testovací účely (v souhrnu 1375 testovacích promluv)

b) 49 vět ... z těchto vět byly tvořeny modely mluvčích (nepoužívali jsme vždy plný počet trénovacích vět)

Mluvčí byli nahráváni v průběhu jednoho či dvou sezení, ale doba mezi sezeními nebyla nikdy delší než jeden měsíc. Nahrávky byly opět čisté, pořizené v laboratorních podmínkách bez vnějšího akustického rušení. Signál byl opět uložen v souboru formátu wav. 16 bitové kódování, vzorkovací frekvence 8 kHz a jeden mono kanál jsou v této oblasti nejpoužívanějšími hodnotami.

4.2. Experimenty s implementovanými metodami

Zde se pokusíme o praktické ověření funkčnosti implementovaných metod identifikujících řečníka, dále se pokusíme najít optimální konfiguraci systému z hlediska nejlepšího rozpoznávacího skóre a nakonec provedeme experimenty, kterými se budeme snažit zjistit robustnost implementovaných metod.

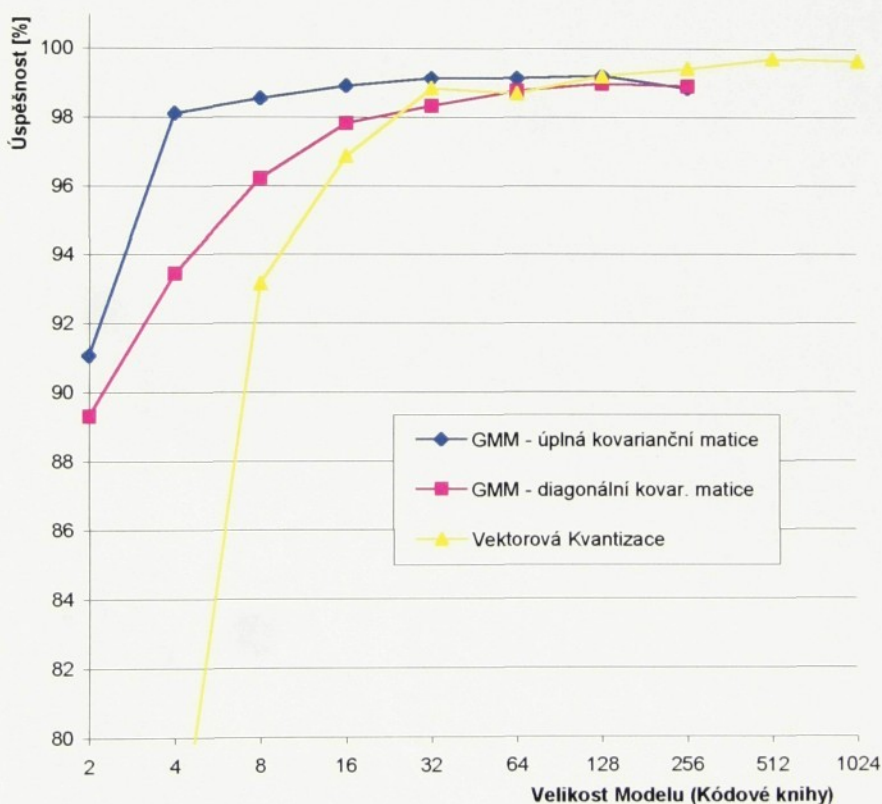
4.2.1. Základní experiment - srovnání výkonnosti VQ a GMM

Vektorová kvantizace pracuje s kódovou knihou, která je sestavena pouze z centroidů. Gaussovske mixturové modely jsou závislé na třech parametrech. Jsou jimi váhové koeficienty mixtur c_m , vektory centroidů (středních hodnot) $\bar{\mathbf{x}}_m$ a kovarianční matice jednotlivých mixtur Σ_m .

Ačkoli obecný model pracuje s úplnou kovarianční maticí, v praxi většinou pracujeme pouze s diagonálními kovariančními maticemi. Vedou nás k tomu tři hlavní důvody: 1) výpočty jsou efektivnější s diagonálními maticemi $\rightarrow$ snadný výpočet inverzní matice a determinantu, 2) empiricky bylo zjištěno, že GMM systémy s diagonálními maticemi mají

vcelku srovnatelné výsledky jako GMM systémy s úplnými maticemi, 3) nižší paměťová náročnost → pracujeme-li s modelem využívajícím úplnou kovarianční matici P -tého řádu a skládá-li se tento model z M mixtur, pak kovarianční matice mají $P \cdot M$ řádků, u diagonálních matic si vystačíme pouze s M řádky.

Srovnání těchto tří systémů je zobrazeno na obr.4.1. Pokud se zaměříme na oblast s menším počtem elementů v modelu (kódové knize), tak v této konfiguraci jasně kraluje GMM systém s úplnou kovarianční maticí. Zjednodušená varianta GMM má o něco horší výsledky, ale přesto stále překonává VQ systém. V případě, kdy systémy pracují s velkým počtem elementů, je výkonnost systémů již daleko vyváženější a pokud VQ pracuje s extrémně velkým počtem kódových slov (1024), vítězí dokonce nad systémy s gaussovskými modely. GMM systémy s více než 128 mixturami (úplná kovarianční matice) respektive 256 mixturami (diagonální matice) jeví známky podtrénování. Proto již nemá žádný význam provádět testování většího počtu elementů.



Obr.4.1 Graf porovnání výkonnosti tří systémů identifikujících mluvčího

Mimo tyto experimenty byly ještě zjišťovány rozdíly v rozpoznatelnosti mužského a ženského hlasu. Při rozpoznávání řeči jsou lepší výsledky dosahovány ženami, proto nás zajímalo, zda je tomu tak i při rozpoznávání mluvčího. U VQ tomu tak je. Ženy jsou rozpoznávány asi o půl procenta lépe než muži. V systému pracujícím s GMM nebylo

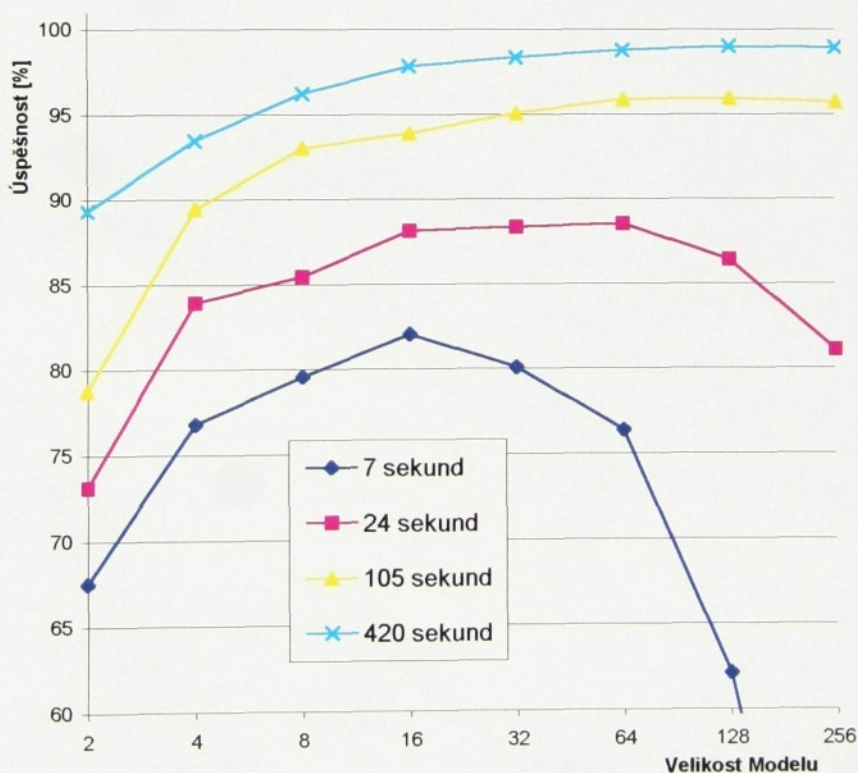
dosaženo přesvědčivého výsledku. Se změnou počtu mixtur v modelu se většinou změnil i poměr úspěšnosti mezi muži a ženami (na jednu či na druhou stranu). Obdržené výsledky nebyly příliš vypovídající, proto zde nejsou uvedeny v grafické podobě.

4.2.2. Míra ovlivnění úspěšnosti identifikace trénovacími daty

Kvalitní a vhodně zvolená data použitá v režimu trénování jsou velmi důležitou podmínkou úspěšné identifikace. Pro ověření míry ovlivnění úspěšnosti identifikace množstvím trénovacích dat bylo k dispozici 49 trénovacích nahrávek od každého mluvčího. Tyto nahrávky byly rozděleny následujícím způsobem:

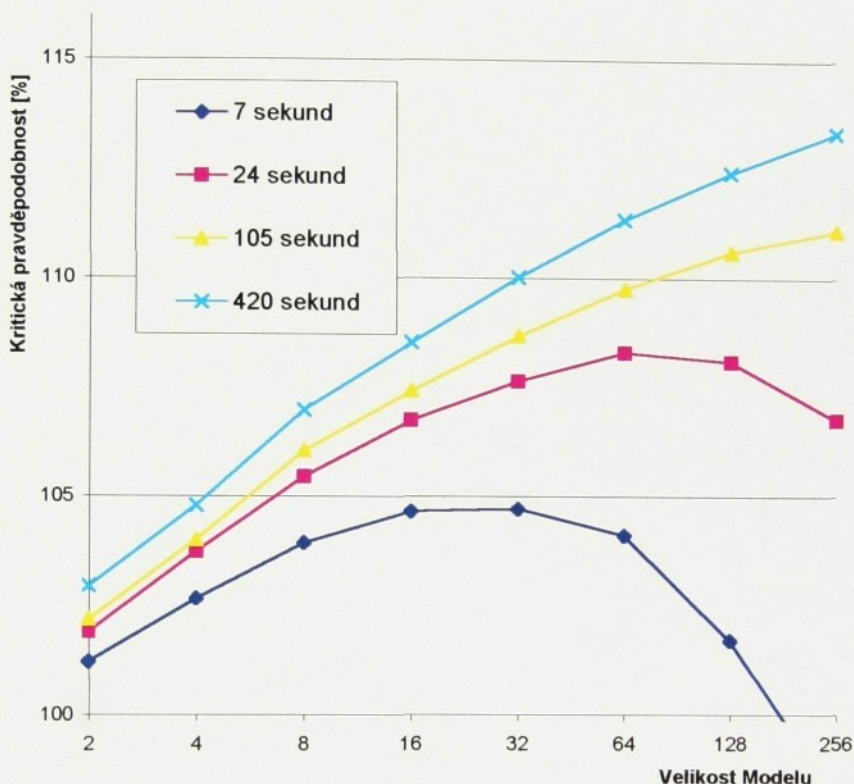
Trénovací soubory	73 - 73	-> 7.18 sekundy => 0.12 minuty
	71 - 73	-> 23.6 sekundy => 0.39 minuty
	68 - 73	-> 48.5 sekundy => 0.81 minuty
	62 - 73	-> 104 sekund => 1.73 minuty
	52 - 73	-> 210 sekund => 3.5 minuty
	25 - 73	-> 419 sekund => 7 minut

GMM systém s diagonální kovarianční maticí jehož modely byly trénovány uvedenými délkami promluv je zobrazen na obr.4.2.



Obr.4.2 Graf zobrazující jednotlivé křivky GMM systému získané po natrénování modelů různými délkami promluv

U promluvy s délkou trvání 7 sekund je nejmarkantnější velmi rychlý pokles rozpoznávacího skóre. Pokud se pohybujeme v oblasti s větším počtem mixtur, nestačí tak krátká promluva pro statisticky věrohodné natrénování modelu. Naproti tomu modely trénované 420 sekundami řeči mají stoupavou tendenci i při 256 mixturách.



Obr.4.3 Graf znázorňující kritické vzdálenosti

Na obr.4.3 je vidět pokles kritických vzdáleností (kritické vzdálenosti byly blíže vysvětleny na obr.3.13) u systémů trénovaných různými délkami promluv. Můžeme vidět, že kritické vzdálenosti přímo souvisí s rozpoznávacím skóre systémů. Tam kde stoupají kritické vzdálenosti, tam roste i úspěšnost rozpoznávání.

4.2.3. Vypovídací schopnost keprálních příznaků

O dynamických příznacích jsme blíže hovořili v kapitole 3.1.6. Zajímalo nás, zda jsou dynamické příznaky při rozpoznávání mluvčích stejně vypovídající jako je tomu při rozpoznávání řeči. V příloze č.1 na obr.4.4 je zobrazen graf vlivu dynamických příznaků na úspěšnost rozpoznávání GMM systému. proti základní konfiguraci, kdy byly použity pouze MFCC příznaky je vidět podstatné (až 4%) zlepšení. Snad nejlépe dopadla konfigurace MFCC + delta-delta příznaků. Delta-delta příznaky viditelně předčily delta příznaky. Spornější je situace kolem konfigurace MFCC + delta + delta-delta příznaků. Při menším

počtu komponent vedou, ale později jejich výkonnost poklesne pod výkonnost MFCC + delta-delta příznaků. Může to být způsobeno krátkým režimem trénování (k tvorbě modelů bylo použito pouze 25 sekund řeči). V příloze č.1 na obr.4.5 je již lépe vidět, jak konkrétní počty mixtur v modelech ovlivňují i přírůstky rozpoznávacího skóre.

V příloze č.2 jsou oba grafy zopakovány pro vektorovou kvantizaci. U VQ systému jsme objevili jednu velmi zajímavou skutečnost. Jeho výkonnost se po aplikaci různých konfigurací dynamických příznaků téměř vůbec nezlepšila.

4.2.4. Úspěšnost rozpoznávání v závislosti na způsobu výpočtu delta příznaků

Výpočet delta příznaků se většinou provádí s délkou okna 2 (delta okolí o velikosti 2 framy). Otázkou bylo, co se stane pokud délku okna zvětšíme na 5. Teoreticky by jsme měli dostat o něco lepší výsledky, protože derivace spočítaná s větší délkou okna by měla mít vyhlazenější průběh. Výsledek těchto experimentů je zaznamenán v příloze č.3. Na obr.4.8 jsou průběhy GMM systému, který byl trénován 25 a 420 sekundami řeči. Na obr.4.9 můžeme vidět stejné průběhy pro vektorovou kvantizaci.

Pro oba systémy zvětšení délky okna nepřineslo téměř žádné zlepšení. Průběhy jak u VQ tak i u GMM systému jsou v podstatě totožné. Na základě těchto výsledků bylo rozhodnuto, že v dalších experimentech již bude používána pouze délka okna 2.

4.2.5. Prospěšnost EM algoritmu v GMM systémech

V tomto experimentu byla zjišťována účinnost EM algoritmu. Experimenty byly prováděny s GMM systémem s úplnou kovarianční maticí a s GMM systémem s diagonální kovarianční maticí. V literatuře je uváděno, že pro konvergenci algoritmu by mělo postačovat 10 iterací. Naším cílem bylo experimentálně ověřit, zda tomu tak opravdu je. Výsledky obou experimentů jsou uvedeny v příloze č.4.

EM algoritmus pro GMM systém s úplnou kovarianční maticí dosáhl již při první iteraci 15% zlepšení úspěšnosti rozpoznávání. Tento přírůstek úspěšnosti byl dosažen u modelu s dvěma mixturami. GMM systém s diagonální kovarianční maticí zaznamenal pro dvě mixtury také výrazné zlepšení. Toto zlepšení však již nebylo tak zřetelné. Od 16 mixtur dále se řádově pohybovalo pouze v jednotkách procent.

Závěrem těchto experimentů bylo, že EM algoritmus je neúčinnější pro malé počty mixtur. Platí tedy, že čím menší je počet mixtur tím jsou výsledky EM algoritmu lepší.

4.2.6. Robustnost aplikovaných algoritmů

Posledním experimentem bylo ověření robustnosti naprogramovaných metod. K ověřování robustnosti systémů byla použita speciální databáze. Struktura této databáze je podrobněji popsána v kapitole 4.1.

Počet testovacích promluv nakonec klesl na 216. Původně bylo plánováno 240 experimentů, ale vzhledem k velmi špatným prvním výsledkům bylo rozhodnuto, že 2 nově nahrané věty budou vyhrazeny pro experiment s dotrénováním systémů. K dispozici jsme tedy měli 216 testovacích vět. To znamená, že jedna špatně rozpoznaná věta nám zhorší celkové rozpoznávací skóre o 0,46 %. V databázi bylo uloženo 20 mluvčích.

Výsledky experimentů jsou uvedeny v přílohách č.5, 6, 7 a 8. Referenční hodnoty jsme získali testováním 1 rok starých nahrávek. Tyto nahrávky byly pořízeny v průběhu jednoho sezení. Pro digitalizaci řečového signálu byl použit stejný hardware. Tyto referenční hodnoty jsou v grafech značeny bez horních indexů.

Jednou hvězdičkou v horním indexu jsou označeny výsledky obdržené při testování promluv pořízených s jednoletým odstupem. Pro nahrání těchto vět byl použit i odlišný hardware (jiný mikrofon, jiná zvuková karta). Dvě hvězdičky v horním indexu označují testovací věty s jednou hvězdičkou, přičemž ale modely mluvčích prošli znovu trénovacím režimem. Při tomto novém trénování byly použity i dvě věty z nových nahrávek. Výsledky testované na GMM a VQ systému jsou znázorněny na obr.4.12 respektive na obr.4.13. Všimněte si prosím velkého poklesu úspěšnosti u nových nahrávek. Pokles téměř o 60 % je krajně nepříjemný a je nutné jej nějakým způsobem řešit. Tímto řešením může být „dotrénování“ respektive nové natrénování řečovým signálem, které již bude obsahovat i část nových nahrávek. Systém s těmito modely je označován právě dvěma hvězdičkami. V našem experimentu byl systém, který byl původně trénován 30 sekundami řeči (čtyři věty), natrénován znovu čtyřmi původními větami a dvěma větami novými. Druhým (ověřovacím) experimentem bylo testování systémů trénovaných 110 sekundami řeči.

Systémy v těchto experimentech pracovali pouze s MFCC kepstrálními koeficienty. Poslední sada testů (snažili jsme se o co možná největší robustnost) využívala i delta a delta-delta koeficientů.

S výše uvedenými vylepšeními jsme byli schopni snížit propastný rozdíl (již uvedených 60 %) na rozdíl o poznání menší. Tento rozdíl činil přibližně 5 až 15 %.

Do přílohy č.9 je zařazena analýza časového a frekvenčního průběhu řečového signálu jednoho mluvčího. Nahrávka p20mfj00.wav byla pořízena před rokem a nahrávka p20mfj01.wav nyní. Jedná se o sadu nahrávek, které byly při původním experimentu správně rozpoznány a po ročním časovém odstupu se při rozpoznávání objevila chyba. Příloha č.10 obsahuje tytéž signály, které jsou tentokrát analyzované spektrogramy. Přílohy č.11 a č.12 obsahují zvětšené spektrogramy.

Analýzou těchto dvou signálů se nám podařilo odhalit odlišnosti ve spektru signálů. Dovolujeme si tvrdit, že tyto změny spíše než časovým odstupem, byly způsobeny změnou nahrávacího hardwaru. Proto by jedním z řešení jak zlepšit robustnost systémů mohlo být natrénování jakéhosi obecného modelu. Tento model by se následně dal použít pro odstranění změn způsobených hardwarem. V této oblasti vidíme asi největší výzvu do budoucna.

5. VERIFIKACE MLUVČÍHO

Verifikace mluvího nebyla hlavním tématem této diplomové práce. Z tohoto důvodu budou v této práci naznačeny jen některé základní myšlenky, které vycházejí z metod pro identifikaci mluvího. Tyto metody se budeme snažit upravit tak, aby úspěšně pracovaly v systémech pro verifikaci mluvího.

5.1. Jednoduchý systémem pro VM

Naším cílem bylo otestovat jednu variantu systému pro verifikaci mluvího, kde by verifikace bylo dosaženo jednoduchou úpravou stávajícího systému pro identifikaci mluvího. Hlavní myšlenka byla taková, že po analýze dat mluvích, kteří byli rozpoznávání systémem pro identifikaci mluvího, obdržíme prahové hodnoty celkového kvantizačního zkreslení těchto mluvích. Určení prahových hodnot je zcela nezbytné pro potvrzení či vyvrácení totožnosti verifikovaného mluvího. Analyzovaná data jednoho náhodně vybraného mluvího jsou uvedena v tab.5.1. Konfigurace VQ systému byla tato: KK o 256 kódových slovech, tré nování proběhlo 420 sekundami řeči, vektor příznaků se skládal z 8 MFCC koeficientů, 8 delta a 8 delta-delta koeficientů. Jednalo se tedy o nejúspěšnější konfiguraci systému, která byla testována.

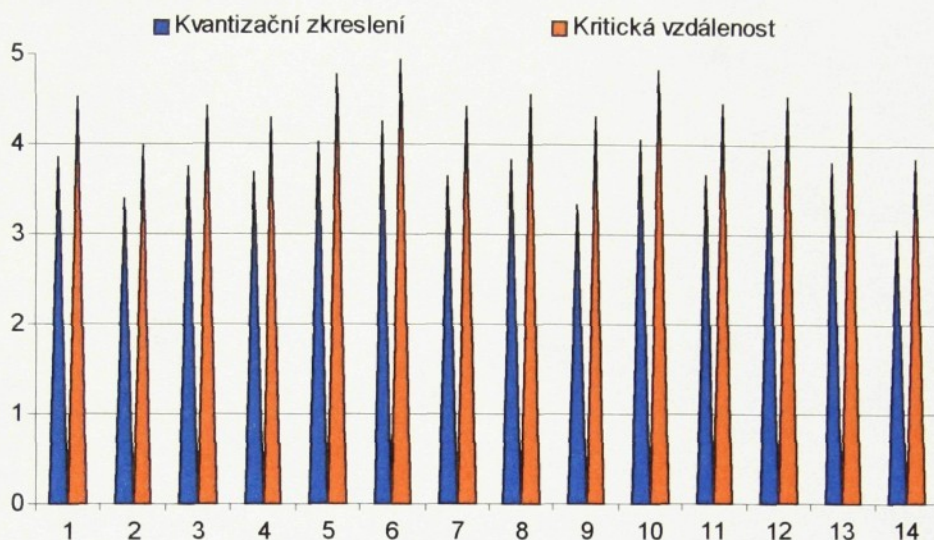
0	0	0	0	0	0	0	0	0	0	0	0	0	0
3,84	3,39	3,75	3,69	3,99	4,24	3,64	3,83	3,33	4,06	3,66	3,95	3,81	3,06
117,32	120,93	117,73	116,33	119,32	115,94	121,34	118,91	129,21	118,71	121,77	115,07	120,71	125,42

Tab.5.1 Analyzovaná data jednoho mluvího vystupující z VQ klasifikátoru

Sloupce v tabulce reprezentují jednotlivé testované promluvy mluvího. První řádek obsahuje hodnoty úspěšnosti, respektive neúspěšnosti procesu rozpoznávání. Nula znamená úspěšně rozpoznanou promluvu, jednička označuje chybu v rozpoznávání. V druhém řádku jsou zapsány hodnoty celkového kvantizačního zkreslení po normalizaci. Normalizace je provedena vydělením celkového kvantizačního zkreslení počtem příznakových vektorů reprezentujících promluvu. Obecně platí, že při rozpoznávání mluvího metodou VQ normalizaci provádět nemusíme, ale v případě verifikace je to nezbytné. A konečně třetí řádek obsahuje hodnoty kritických vzdáleností (pojem kritické vzdálenosti byl vysvětlen na obr.3.13).

Hodnoty kvantizačního zkreslení se pohybují v rozmezí $3,06 \div 4,24$. Hodnoty kritických vzdáleností jsou naproti tomu v rozmezí $115 \div 129$ % hodnot kvantizačního zkreslení. Názorněji jsou tyto hodnoty uvedeny na obr.5.1. Na tomto příkladu můžeme pozorovat, že je skoro nemožné uspokojivě určit univerzální prahovou hodnotu pro posouzení pravosti či nepravosti mluvího. Pokud bychom za prahovou hodnotu určili například 4, což je asi nejrozumnější hodnota, pravý mluví by byl třikrát zamítnut a narušitel by byl dvakrát

vpuštěn. V porovnání s identifikací, která s těmito testovacími daty měla úspěšnost 100 %, to není příliš uspokojivý výsledek.



Obr.5.1 Určení prahové hodnoty pro verifikaci

Při provedení této analýzy u metody GMM jsme dospěli k obdobným výsledkům. Opět se nepodařilo uspokojivě oddělit správného mluvího od narušitelů. Z těchto důvodů bylo od dalších pokusů s tímto způsobem verifikace mluvího upuštěno. Další možná metoda verifikace mluvího je uvedena v následující podkapitole.

5.2. GMM jako základ systému pro textově nezávislou VM

GMM je použito jako všeobecný pravděpodobnostní model pro vícerozměrné rozdělení hustoty pravděpodobnosti, které je schopné reprezentovat jakékoliv rozdělení hustoty pravděpodobnosti. Právě tato schopnost je velmi vhodná pro aplikace textově nezávislé verifikace mluvího.

Jak již bylo uvedeno dříve, verifikace mluvího je úloha, při níž ověřujeme, zda promluva reprezentovaná posloupností vektorů příznaků $\mathbf{X}$ byla opravdu vyslovena proklamovaným mluvího m . Tento problém můžeme přeformulovat na testování dvou základních hypotéz:

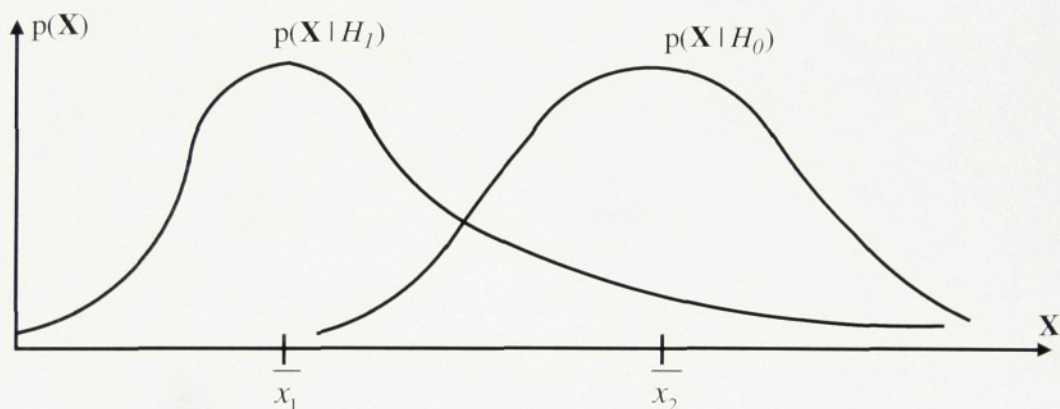
$$\begin{aligned}
 H_0: & \quad \mathbf{X} \text{ je promluva od mluvího } m \\
 & \quad \text{a} \\
 H_1: & \quad \mathbf{X} \text{ není promluva od mluvího } m.
 \end{aligned}$$

Optimální způsob, jak rozhodnout, která z daných hypotéz je správná, je tzv. test hodnoty věrohodnosti (LR - Likelihood Ratio test). Zjišťování správnosti hypotéz je optimální

pouze tehdy, pokud přesně známe pravděpodobnostní funkce. Bohužel v praxi je to jen velmi zřídka. LR test je definován takto:

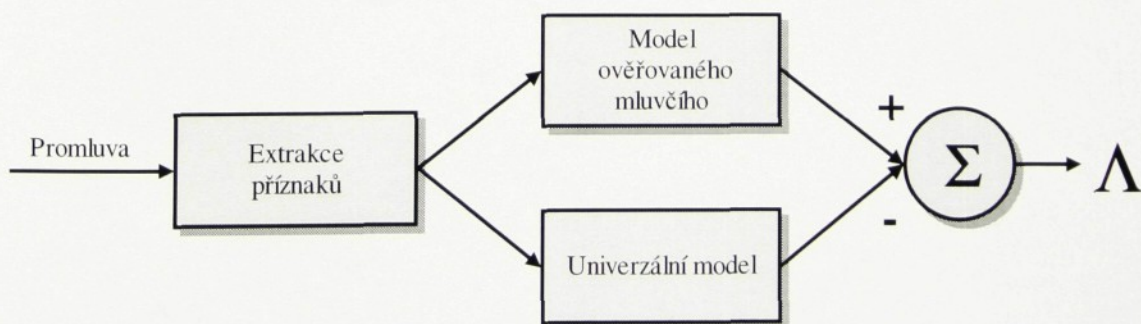
$$\frac{p(\mathbf{X} | H_0)}{p(\mathbf{X} | H_1)} \begin{cases} \geq \theta & \text{akceptování hypotézy } H_0 \\ < \theta & \text{zamítnutí hypotézy } H_0, \end{cases} \quad (5.1)$$

kde $p(\mathbf{X} | H_i)$, $i=0, 1$ je hustota pravděpodobnosti pro hypotézu H_i vypočtená pro sledovaný segment řeči $\mathbf{X}$ ($p(A|B)$ bereme jako pravděpodobnost, pokud B je v této funkci nezávislá proměnná). Rozhodovací práh pro akceptování nebo zamítnutí H_0 je θ . Na obr.5.2 jsou vyobrazeny dvě věrohodnostní funkce $p(\mathbf{X} | H_0)$ a $p(\mathbf{X} | H_1)$. Hlavním úkolem systému pro verifikaci mluvěho je stanovení technik pro přesné určení těchto dvou pravděpodobností, $p(\mathbf{X} | H_0)$ a $p(\mathbf{X} | H_1)$.



Obr.5.2 Věrohodnostní funkce $p(\mathbf{X} | H_0)$ a $p(\mathbf{X} | H_1)$

Na obr.5.3 jsou zobrazeny základní komponenty systému pro verifikaci mluvěho, který pracuje s hodnotami věrohodnosti. Výstupem člena extrahujícího příznaky je sekvence vektorů příznaků, které reprezentují danou promluvu (mohou to být opět MFCC keprstrální příznaky) $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$, kde $\mathbf{x}_t$ je vektor příznaků v diskrétním čase $t \in [1, 2, \dots, T]$.



Obr.5.3 GMM systém pro verifikaci mluvěho, pracující s UBM

Tyto vektory příznaků jsou použity pro výpočet pravděpodobností H_0 a H_1 . Z matematického hlediska je hypotéza H_0 reprezentována modelem označeným λ_{hyp} , který charakterizuje testovaného mluvčího m popsáného vektory příznaků $\mathbf{X}$. Můžeme například říci, že gaussovské rozdělení nejlépe reprezentuje rozložení vektorů příznaků pro H_0 . Pak λ_{hyp} je dána střední hodnotou a kovarianční maticí gaussovského rozdělení.

Alternativní hypotéza H_1 je reprezentována modelem λ_{hyp} . Výsledná hodnota věrohodnosti je tedy $p(\mathbf{X} | \lambda_{hyp}) / p(\mathbf{X} | \lambda_{hyp})$. Tato míra je často udávána jako logaritmická hodnota věrohodnosti:

$$\Lambda(\mathbf{X}) = \log p(\mathbf{X} | \lambda_{hyp}) - \log p(\mathbf{X} | \lambda_{hyp}) \quad (5.2)$$

Zatímco model pro reprezentaci H_0 je velmi dobře popsán a lze ho poměrně snadno získat z trénovacích promluv mluvčího m , model pro λ_{hyp} je vymezen daleko hůře, protože musí reprezentovat veškeré ostatní alternativy k danému mluvčímu. Pro modelování této druhé hypotézy existují dva základní přístupy. Prvním přístupem pro pokrytí druhé hypotézy je používání souboru modelů ostatních mluvčích. V různých kontextech je tento přístup nazýván „Likelihood Ratio Sets“, „Cohorts“ a „Background Speakers“. Množina modelů M mluvčích pracujících na pozadí $\{\lambda_1, \lambda_2, \dots, \lambda_M\}$ je využita při tvorbě modelu pro druhou hypotézu:

$$p(\mathbf{X} | \lambda_{hyp}) = \vartheta(p(\mathbf{X} | \lambda_1), \dots, p(\mathbf{X} | \lambda_M)), \quad (5.3)$$

kde $\vartheta()$ je funkce (průměr, maximum ...) hodnot věrohodnosti získaných z množiny mluvčích na pozadí.

Druhý hlavní přístup jak získat informace o $p(\mathbf{X} | H_1)$ je založen na spojení promluv několika mluvčích a provedení trénování jediného modelu. Tento model je pak v literatuře nazýván „General Model“, „World Model“, „Universal Background Model“ (UBM) a můžeme ho označit λ_{UBM} . Je velmi důležité, aby počet promluv mluvčích byl co největší a aby tyto mluvčí dobře reprezentovali osoby, se kterými se počítá při rozpoznávání. Velkou výhodou tohoto přístupu je fakt, že po pečlivém natrénování univerzálního modelu je možno tento model použít pro všechny rozpoznávané mluvčí.

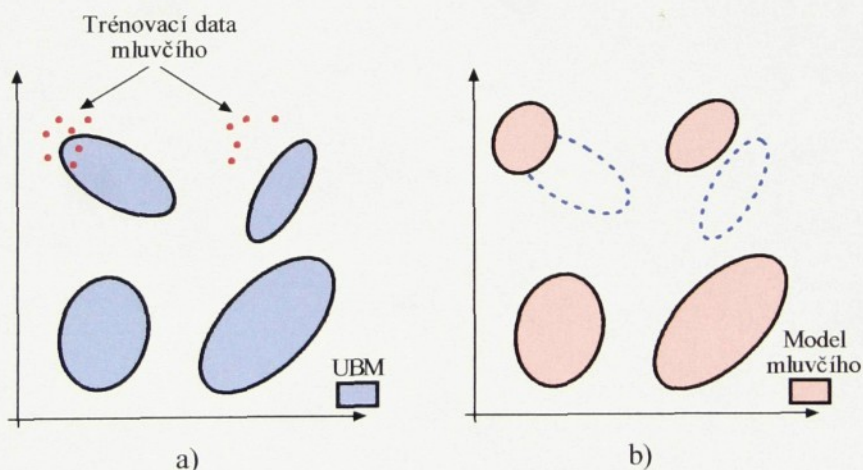
UBM je trénován EM algoritmem, pro trénování modelů mluvčích je použita forma Bayesovské adaptace (BA). Pokud tedy hovoříme o systém GMM-UBM, odvozujeme model testovaného mluvčího úpravou parametrů UBM na základě trénovací promluvy mluvčího a Bayesovské adaptace (občas uváděna jako „Maximum A Posterior“ - MAP analýza). Hlavní myšlenkou při adaptaci je odvození modelu mluvčího pozměněním dobře natrénovaných parametrů UBM. Stejně jako EM algoritmus je adaptace dvoustupňový analytický proces, který můžeme zhruba popsat následujícím algoritmem.

Odvození modelu mluvího z UBM

Krok 1: Je identický s prvním krokem EM algoritmu. Provádíme v něm odhadování vhodných statistických přehledů trénovacích dat pro každé rozdělení UBM.

Krok 2: Na rozdíl od EM algoritmu jsou nové statistické odhady zkombinovány se starými odhady získanými z UBM. K tomu jsou použity směšovací koeficienty, které jsou závislé na datech.

Postup při adaptaci je graficky naznačen na obr.5.4. Nejprve je určeno pravděpodobnostní uspořádání trénovacích vektorů příznaků vůči komponentám UBM (obr.5.4a). S ohledem na toto uspořádání určíme váhové koeficienty c_m , střední hodnoty $\bar{\mathbf{x}}_m$ a kovarianční matice Σ_m všech mixtur. To je vlastně stejný postup jako u „Expectation“ kroku EM algoritmu. Nakonec se tyto nové parametry použijí pro aktualizaci statistických údajů starého UBM (viz. obr.5.4b).



Obr.5.4 Grafické znázornění dvou kroků, které je třeba učinit při transformaci UBM na model testovaného mluvího

Logaritmické pravděpodobnostní skóre pro sekvenci vstupních vektorů příznaků verifikovaného mluvího je nakonec vypočítáno podle vztahu:

$$\Lambda(\mathbf{X}) = \log p(\mathbf{X} | \lambda_{hyp}) - \log p(\mathbf{X} | \lambda_{UBM}) \quad (5.4)$$

Podrobnější informace o verifikaci mluvího, při které je použita metoda GMM, jsou uvedeny v [15].

6. ZÁVĚR

Automatické rozpoznávání mluvěcího je založeno na používání výpočetní techniky pro rozpoznávání osob na základě řeči. Systémy pro rozpoznávání mluvěcího se používají ve dvou základních módech: k identifikaci jednotlivých osob a k ověřování tvrzení o jejich identitě. Pro obě tyto třídy existují podle použité metody rozpoznávání obrazů v podstatě tři základní přístupy. Prvním z nich je porovnávání s referenčními vzory slov, kde jako dominantní metoda pro textově závislé rozpoznávání stále vystupuje algoritmus DTW. U textově nezávislých systémů to může být buď vektorová kvantizace nebo metoda nejbližšího souseda, která kombinuje přednosti obou předchozích. Tyto metody se také dají charakterizovat deterministickým přístupem. Statistický přístup je další možný způsob rozpoznávání mluvěcích. Rozpoznávání je zde dosaženo porovnáváním se statistickými modely mluvěcích. Prim v této kategorii hraje modelování slov za užití skrytých markovských modelů (GMM). Jako poslední můžeme uvést přístup opírající se o umělou inteligenci. Zde se experimentuje s neuronovými sítěmi, jejichž schopnost učit se je založena na vytváření nových propojení (cest). Stále více se dostávají do popředí tzv. hybridní, nebo také víceúrovňové systémy. Ty jsou složeny z více vrstev, z nichž každá je postavena na některém z uvedených přístupů. Dále se rozpoznávání mluvěcího může rozlišovat podle toho, zda je závislé nebo nezávislé na textu promlouvaného slova.

Úkolem této diplomové práce bylo nejen seznámení se s těmito metodami, ale dvě z těchto metod i implementovat a experimentálně ověřit v pokud možno reálné aplikaci. První implementovanou metodou byla vektorová kvantizace, druhou pak gaussovské mixturové modely.

Testování obou systémů probíhalo proti stejné databázi. Tato databáze byla tvořena 29 muži a 26 ženami. Každý z mluvěcích pronesl 74 vět, z nichž prvních 25 bylo použito při samotném testování (jako vstupní promluvy neznámých mluvěcích), zbylých 49 bylo využito pro tvorbu modelů. V konečném důsledku to tedy znamená, že jednotlivé konfigurace systému (počet a typ příznaků, množství elementů modelů, atd.) byly testovány 1375 promluvami (1 nesprávně rozpoznaná promluva = chyba 0,0727%) proti databázi mluvěcích čítající 55 osob. Robustnost byla ověřována na speciální databázi. Tato databáze měla menší počet mluvěcích (20), ale na druhou stranu splňovala kritéria, která jsme na ni kladly, tj. větší časový odstup mezi nahrávkami, nahrávky pořízené na odlišném hardwaru.

Vektorová kvantizace a gaussovské mixturové modely jsou metody textově nezávislé, a proto je možné pro identifikaci použít jakékoliv věty. Prakticky ve všech experimentech byla na osu x osu vynášena velikost modelu (kódové knihy) a tím byl zjišťován vliv tohoto parametru na výsledky identifikace. Jako první byl proveden experiment na zjištění výkonnosti obou metod. Přitom byly ověřovány i jejich parametry pro různé velikosti KK. Ukázalo se, že pokud jsou vhodně zvoleny konfigurace obou systémů, je VQ dokonce o něco málo výkonnostně lepším systémem než GMM (VQ k tomu ale potřebuje daleko více

komponent). V tomto základním experimentu dosáhla VQ úspěšnosti celých 99,71 % (1024 kódových slov), GMM systém s úplnou kovarianční maticí 99,2 % (128 mixtur) a GMM systém s diagonální maticí úspěšně rozpoznal 98,98 % (128 mixtur) testovacích nahrávek. Tyto experimenty byly ovšem prováděny se základní konfigurací kepstrálních příznaků, takže po vylepšení těchto systémů jsme mohli očekávat o něco lepší výsledky.

Další experimenty si probereme pouze v rychlosti. Jedním z těchto experimentů bylo zjišťování, jak množství trénovacích dat může ovlivnit úspěšnost rozpoznávání. Po vyhodnocení výsledků bylo jasné, že je to jeden z nejdůležitějších faktorů, které ovlivňují úspěšnost rozpoznávání. Jednoznačně platí, že čím více je kvalitních trénovacích dat, tím větší je úspěšnost rozpoznávání. Dalším experimentem jsme ověřovali vypovídací schopnost dynamických příznaků. K dispozici jsme měli 8 statických (MFCC) a 16 dynamických (delta a delta-delta) příznaků. Z výsledků jsme vyvodili, že dynamické příznaky v systémech pro rozpoznávání mluvčího nemají takovou vypovídací schopnost jako příznaky statické, přesto však významnou měrou přispívají ke zmenšení chybovosti. Toto ovšem platí pro GMM. VQ na dynamické příznaky prakticky nereaguje a úspěšnost rozpoznávání je identická jak s dynamickými příznaky, tak i bez nich. Modely GMM systémů jsou vytvářeny EM algoritmem. Zajímalo nás, jakým způsobem počet iterací ovlivní výkonnost. V podstatě se dá říci, že EM algoritmus zlepšuje přiměřeným způsobem parametry modelů. Ovšem vzhledem k velmi dobrému počátečnímu odhadu není zlepšení až tak výrazné.

Posledním experimentem bylo ověření robustnosti obou systémů. Jak jsem již uvedl, tento experiment byl uskutečněn na speciálně k tomuto účelu vytvořené databázi. První hrozivé zhoršení výkonnosti (zhoršení až o 50 %) se nám podařilo snížit až na přijatelných 10 %. Toto zlepšení bylo zapříčiněno dotrénováním modelů a přidáním dynamických příznaků k původním statickým.

V průběhu experimentů se nám podařilo sestavit GMM systém využívající diagonální kovarianční matici, který po natrénování 49 promluvami dosahoval maximální rozpoznávací skóre 99,56 %. Příznakový vektor se skládal z 8 MFCC, 8 delta a 8 delta-delta příznaků. Model se skládal z 256 mixtur. Maximální úspěšnost VQ systému byla 99,49 %. Tento systém měl naprosto stejnou konfiguraci jako GMM systém, kromě příznakového vektoru, který se skládal pouze z 8 MFCC a 8 delta příznaků.

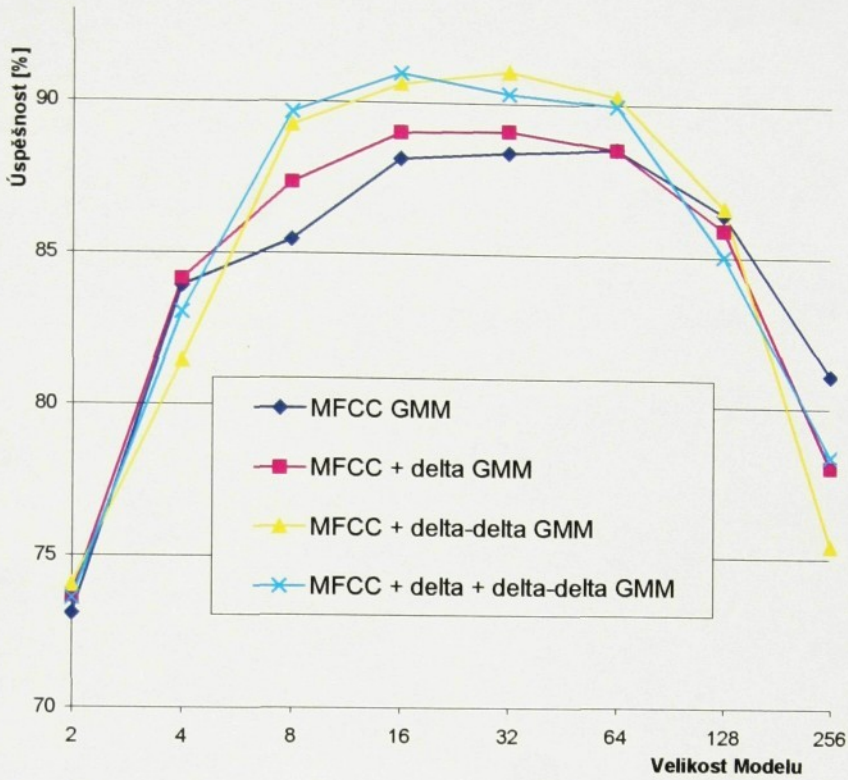
Takto vytvořené systémy jsou sice funkční, ale do praktické využitelnosti mají ještě daleko. Stávající systémy je třeba vylepšit především z hlediska robustnosti. Experimenty prokázaly, že v této oblasti je rozhodně ještě co zlepšovat. Uvědomme si, že stále pracujeme s velmi kvalitním signálem a přesto změna nahrávacího hardwaru a větší časový odstup nahrávek zhoršily výkonnost systémů téměř o 20 %. Určité procento chyb se sice dá odstranit dotrénováním systému a použitím dynamických příznaků, přesto po systému požadujeme větší robustnost. Dále by se mohla vylepšit výkonnost systému trochu jiným uspořádáním Mel-spektrálních filtrů při výpočtu MFCC kepstrálních příznaků. Logaritmické rozmístění je

sice velmi účinné, ale pokud pokryjeme nejvíce reprezentativní frekvence větším počtem Melovských filtrů, můžeme účinnost keprálních příznaků ještě vylepšit.

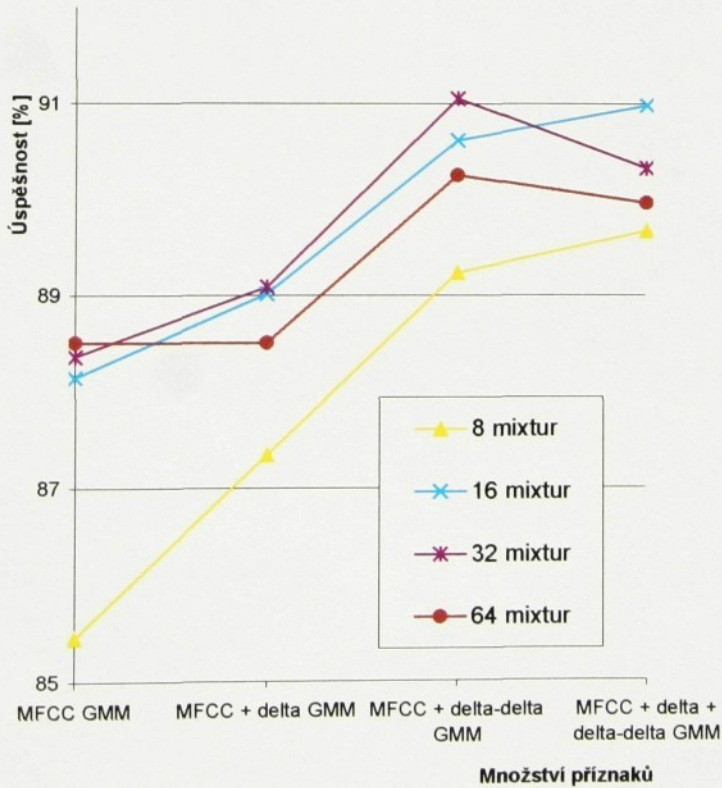
Verifikace mluvího je v dnešní době prakticky využitelnější než identifikace mluvího. Z tohoto důvodu je potřeba zaměřit snahu o tvorbu „real time“ systému tímto směrem. Především u GMM systému, který ještě nebyl optimalizován z hlediska rychlosti (chtěli jsme především otestovat vlastnosti a výkonnost), je prostor k algoritmické optimalizaci.

LITERATURA

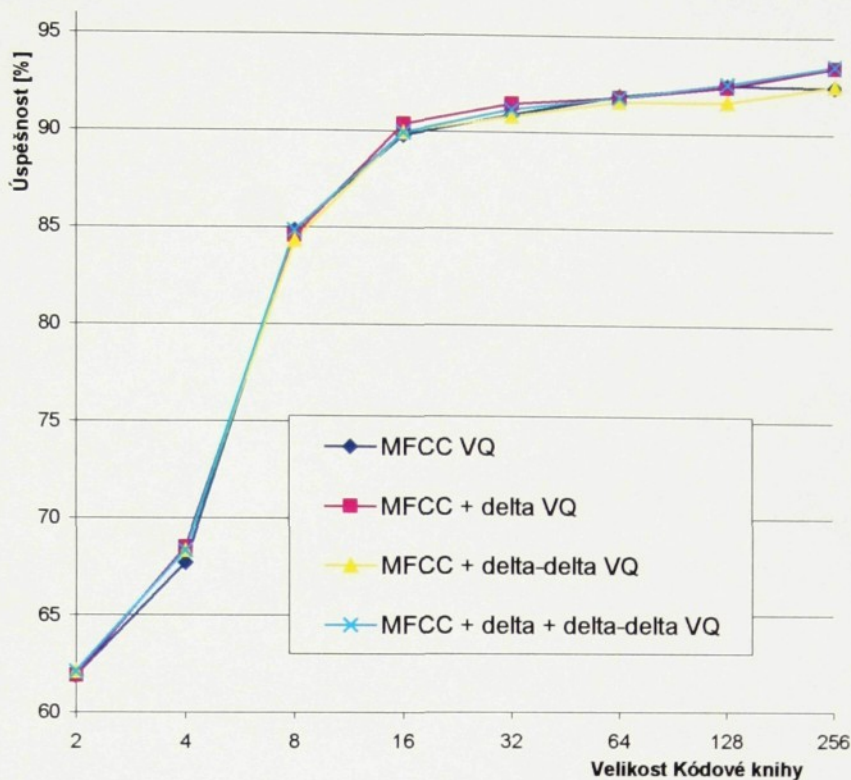
- [1] Svobodová, M.: "*Identifikace mluvíčího v češtině na základě akustického spektra hlásek*", konference: "Przemiany fonetyki slowianskiej w latach 1944–2000 Toruń, 26–27.2.2000", Fonetický ústav FF UK Praha.
- [2] Campbell, J.P. jr.: "*Speaker Recognition: A Tutorial*", Proceedings of the IEEE, vol. 85, no. 9, September 1997, pp. 1437-1462.
- [3] Radová, V.: "*Rozpoznávání řečníka*", pp. 1-7, ZČU Plzeň.
- [4] Nouza, J.: "*Základní principy a metody rozpoznávání řeči*", "Počítačové zpracování řeči – cíle, problémy, metody a aplikace", Liberec 2001, pp. 57-65.
- [5] Itakura, F.: "*Minimum Prediction Residual Principle Applied to Speech Recognition*", IEEE Trans. on ASSP, vol.23, no.1, 1975.
- [6] Psutka, J.: "*Komunikace s počítačem mluvenou řečí*", Academia, Praha 1995, pp. 39-57, 127-200, 254-259.
- [7] Nouza, J.: "*Implementace algoritmu DTW pro rozpoznávání řeči v reálném čase*"
- [8] Reynolds, D.A.: "*A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification*", Ph.D. thesis, Georgia Institute of Technology, September 1992.
- [9] Tanprasert, Ch., Achariyakulporn, V.: "*Comparative study of GMM, DTW and ANN on Thai Speaker Identification System*".
- [10] David, P.: "*Ročníkový projekt - Rozpoznávání osob na základě hlasového projevu*", TU Liberec, Liberec 2001.
- [11] Nouza, J.: "*On the Speech Feature Selection Problem: Are Dynamic Features More Important Than the Static Ones?*", Proc. of EUROSPEECH'95 Conference, Madrid, Spain, September 1995, pp. 919-923.
- [12] Huang, X., Acero, A., Hon, H.W.: "*Unsupervised Estimation Methods*", "Spoken Language processing – a guide to theory, algorithm, and system development", New Jersey 2001, pp. 166-169, pp. 170-175.
- [13] Gray, R.M., Karnin, E.D.: "*Multiple Local Optima in Vector Quantizers*", IEEE Trans. on Information Theory, 1982, IT-28, pp. 256-261.
- [14] Linde, Y., Buzo, A., Gray, R.M.: "*An Algorithm for Vector Quantizer Design*", IEEE Trans. on Communication, 1980, COM-28(1), pp. 84-95.
- [15] Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: "*Speaker Verification Using Adapted Gaussian Mixture Models*", Digital Signal Processing, 2000, vol. 10, pp. 19-41.



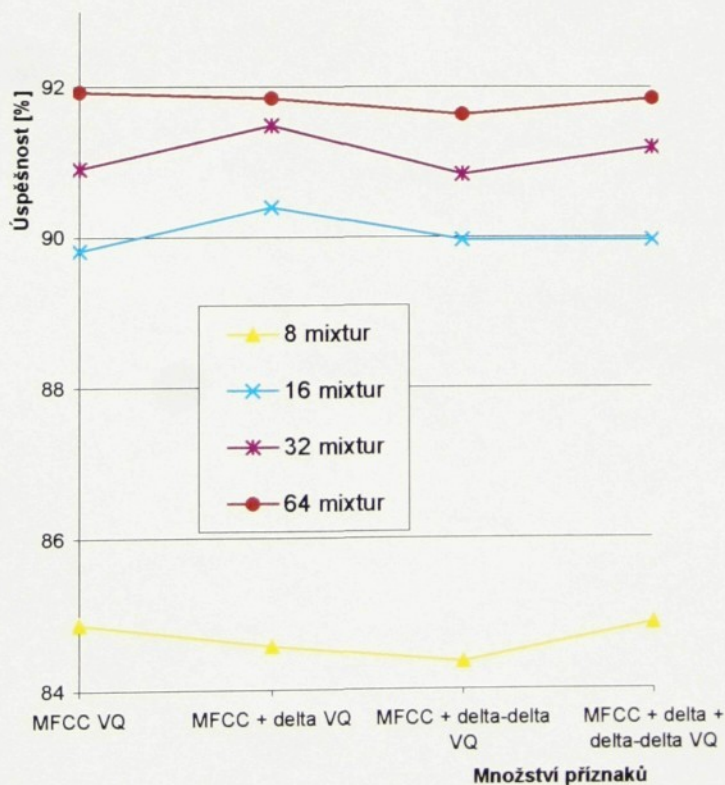
Obr.4.4 Graf vlivu dynamických příznaků na výkonnost GMM systému



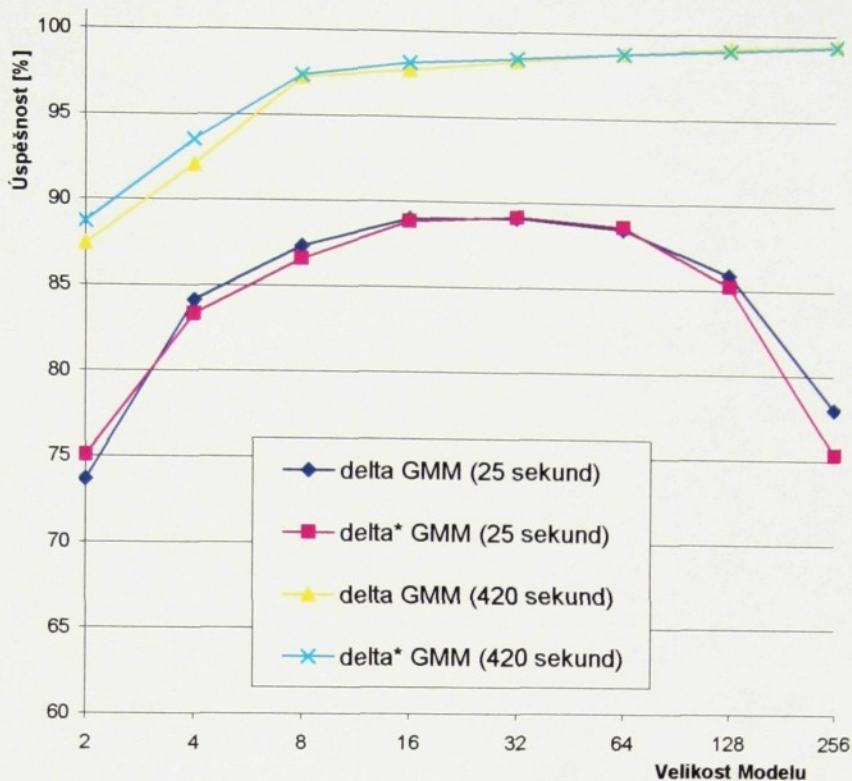
Obr.4.5 Graf reakce GMM systému na dynamické příznaky



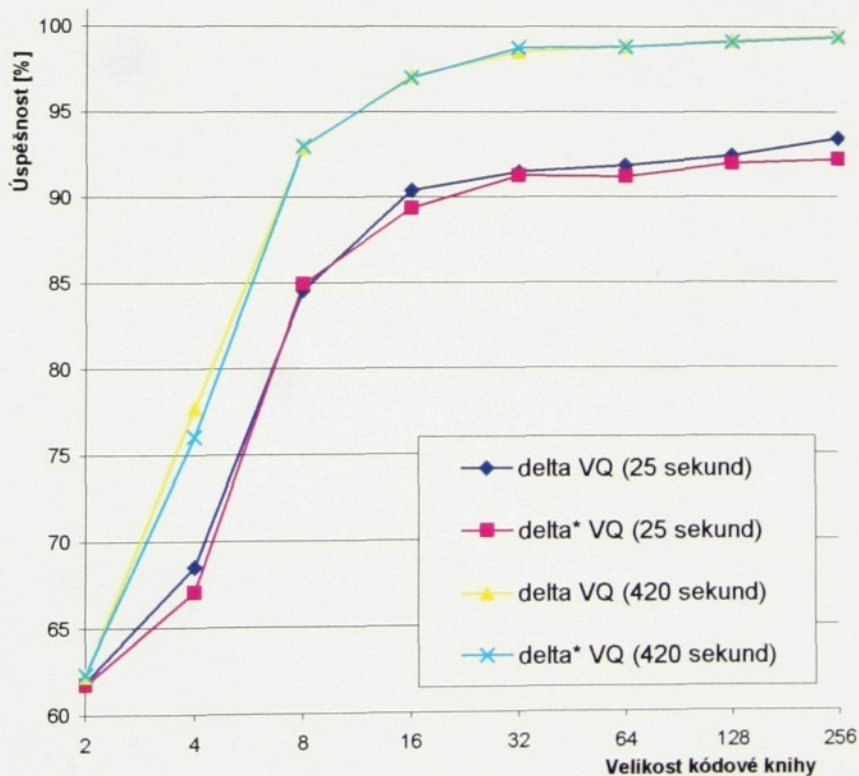
Obr.4.6 Graf vlivu dynamických příznaků na výkonnost VQ systému



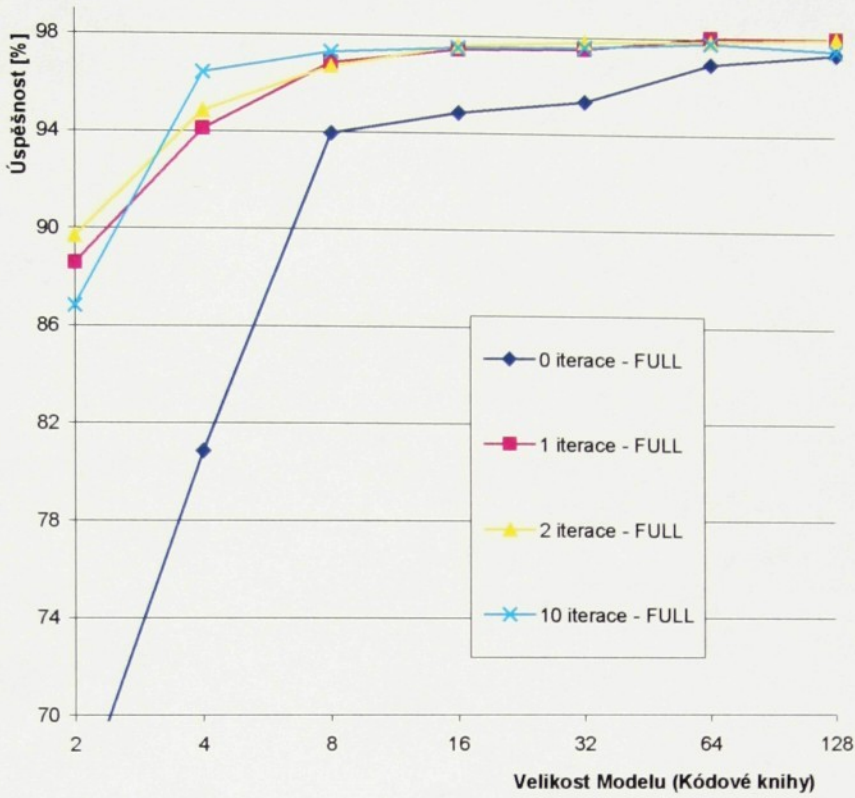
Obr.4.7 Graf reakce VQ systému na dynamické příznaky



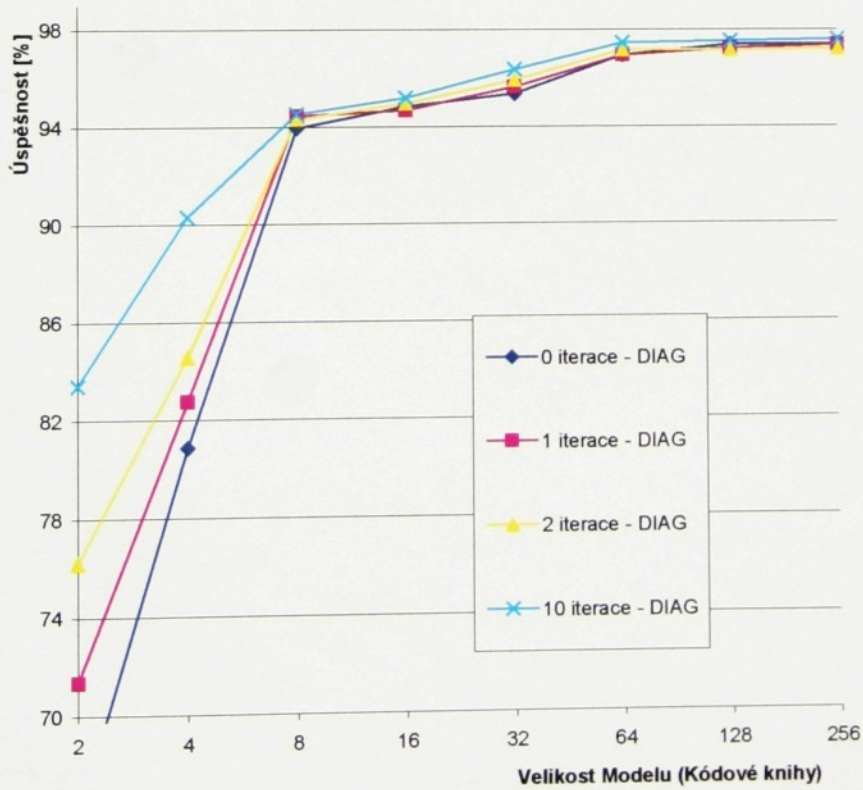
Obr.4.8 Graf vlivu různých způsobů výpočtu delta příznaků na GMM systém



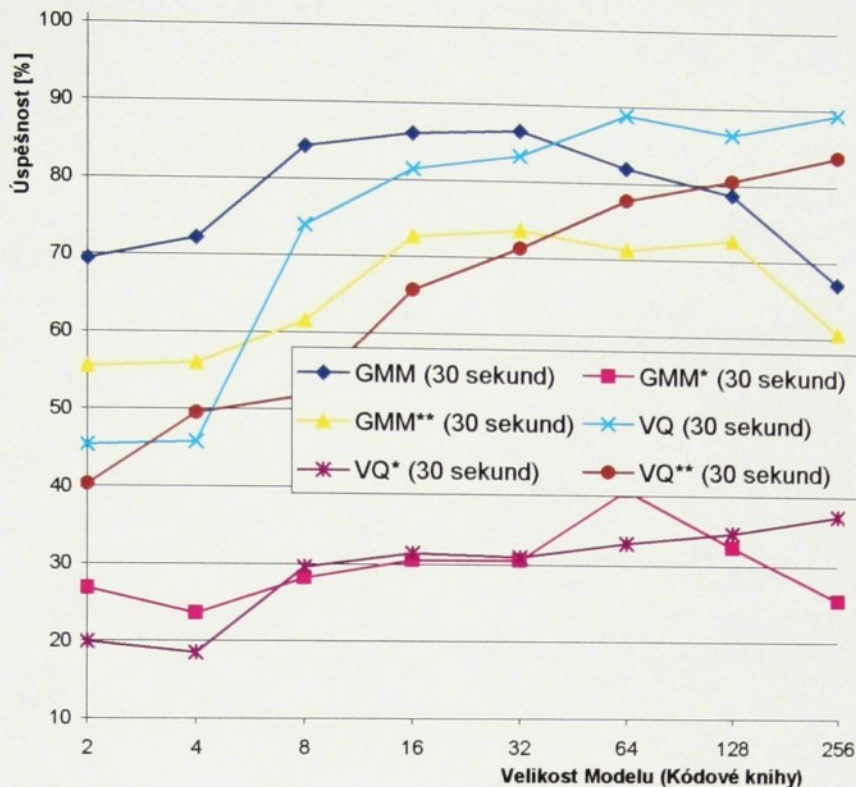
Obr.4.9 Graf vlivu různých způsobů výpočtu delta příznaků na VQ systém



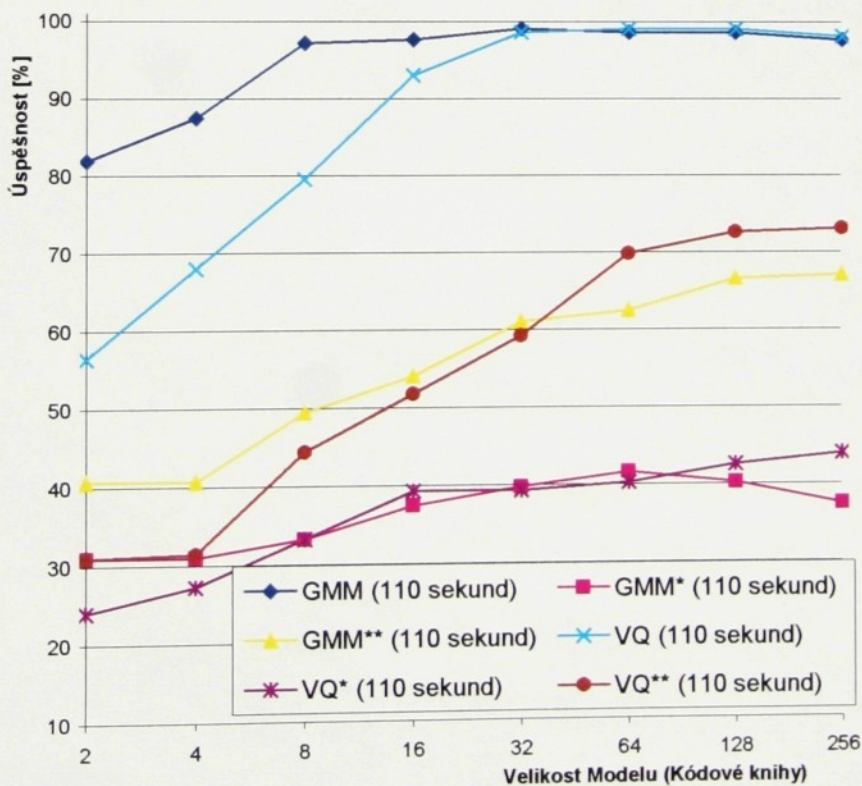
Obr.4.10 Graf vlivu iterací EM algoritmu na GMM systém s FULL kov. maticí



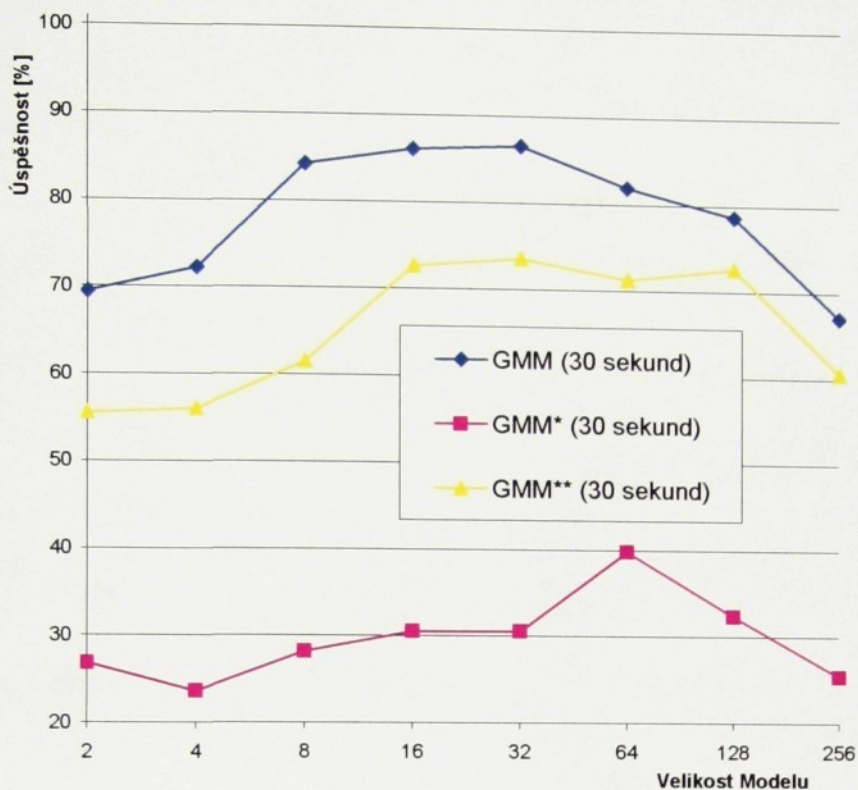
Obr.4.11 Graf vlivu iterací EM algoritmu na GMM systém s DIAG kov. maticí



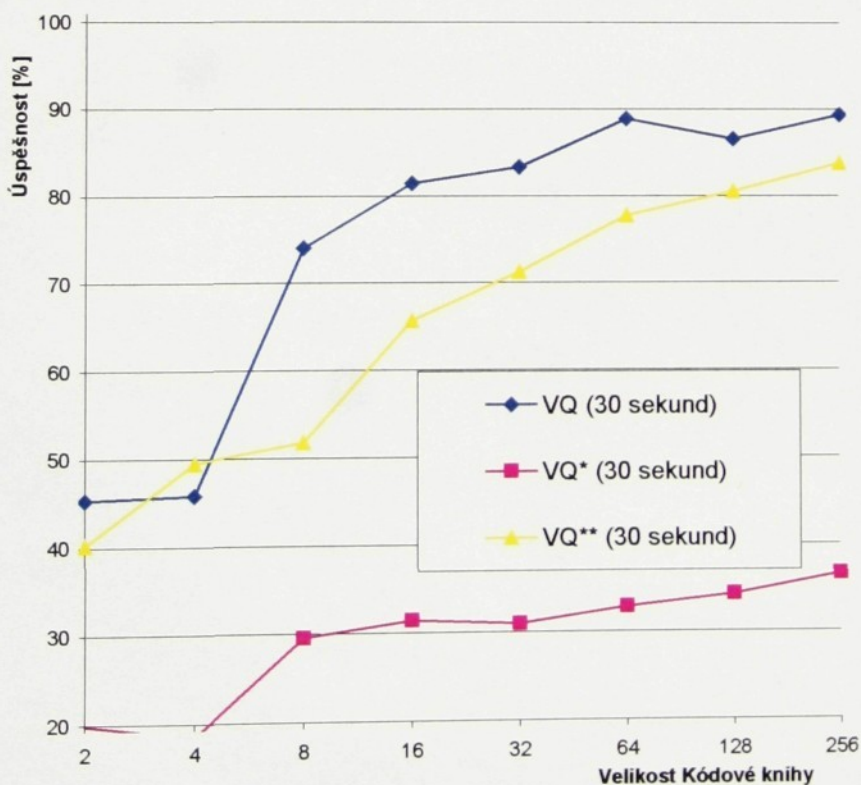
Obr.4.12 Graf robustnosti VQ a GMM systémů trénovaných 30 sekundami řeči



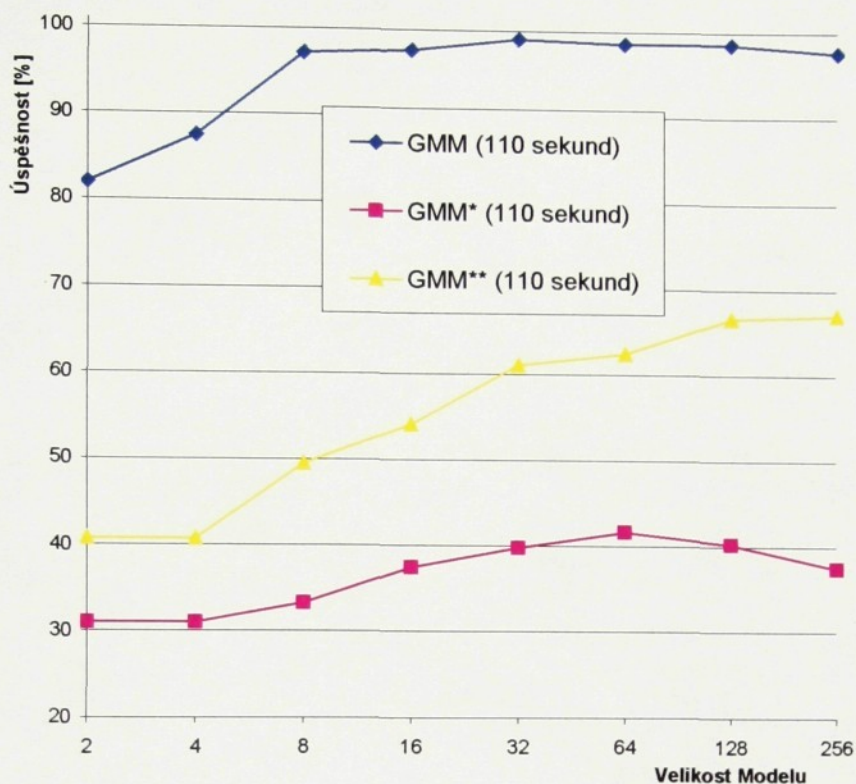
Obr.4.13 Graf robustnosti VQ a GMM systémů trénovaných 110 sekundami řeči



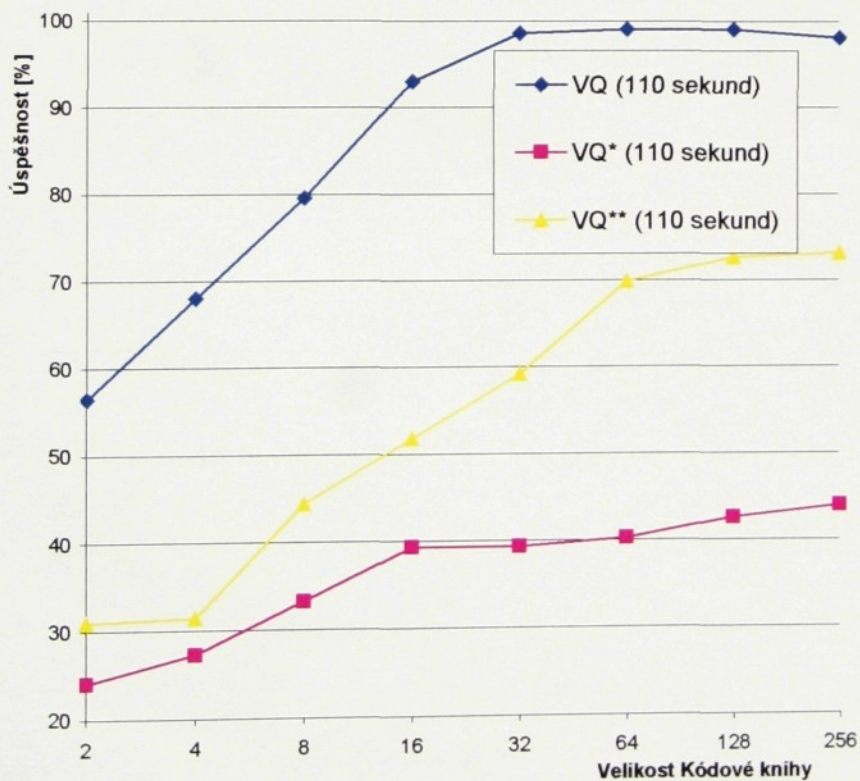
Obr.4.14 Detailní graf robustnosti GMM systému trénovaného 30 sekundami řeči



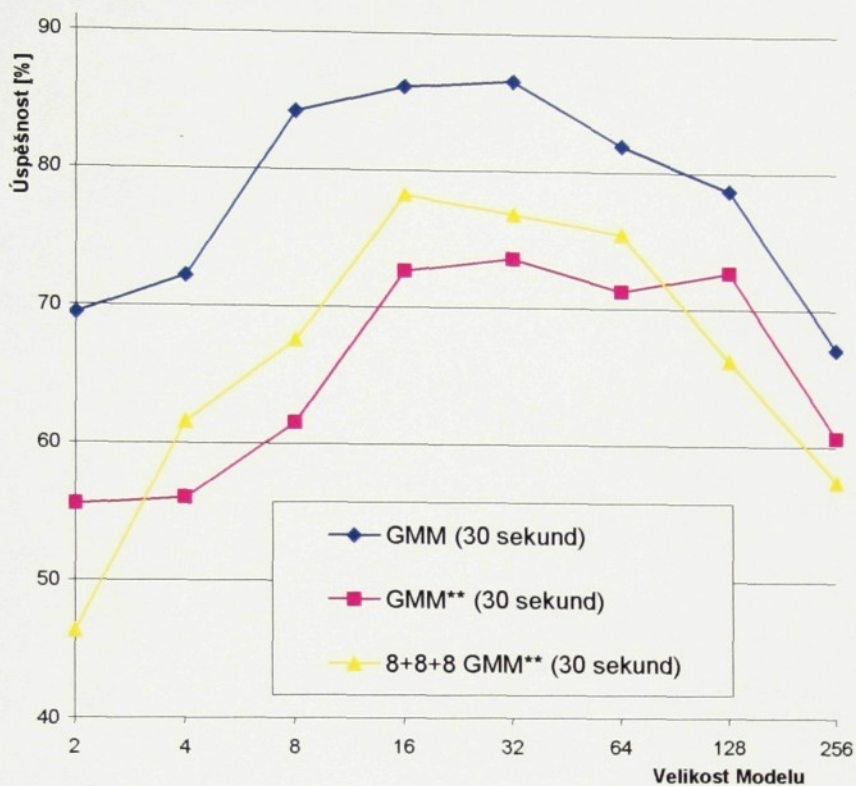
Obr.4.15 Detailní graf robustnosti VQ systému trénovaného 30 sekundami řeči



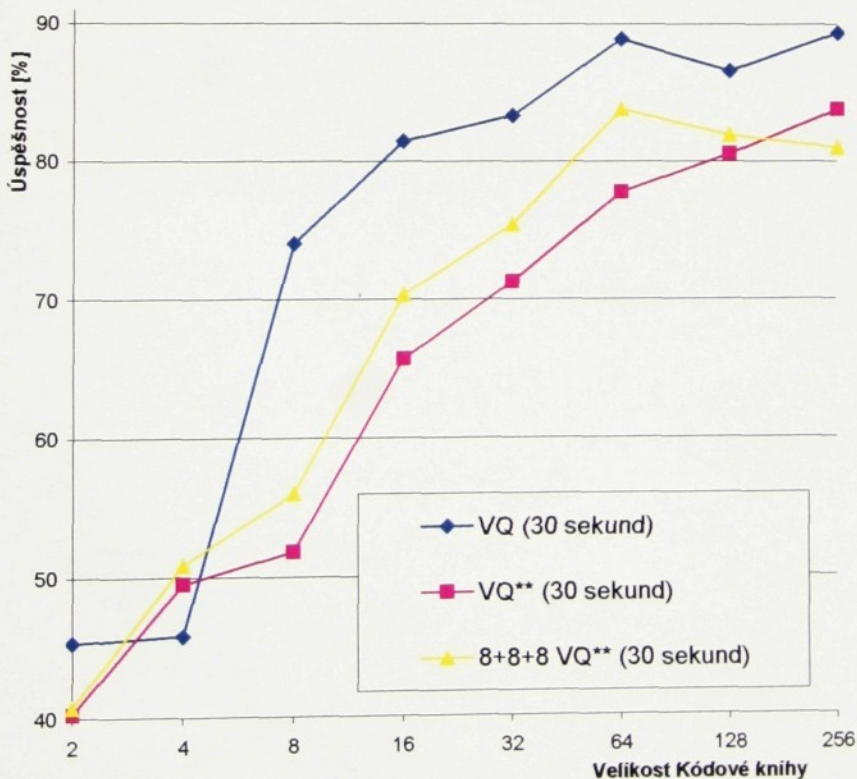
Obr.4.16 Detailní graf robustnosti GMM systému trénovaného 110 sekundami řeči



Obr.4.17 Detailní graf robustnosti VQ systému trénovaného 110 sekundami řeči

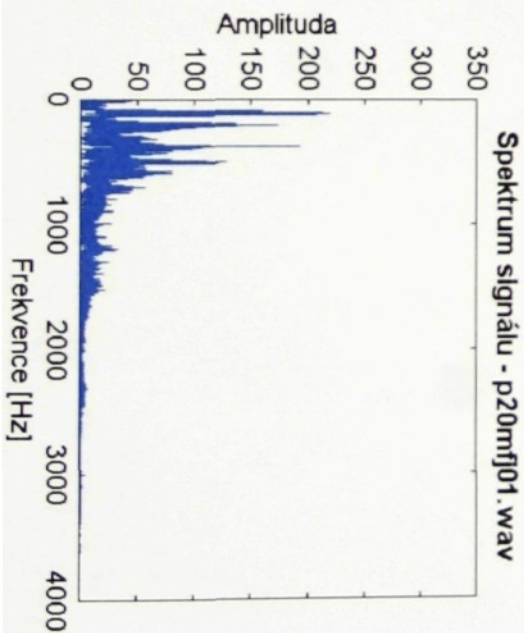
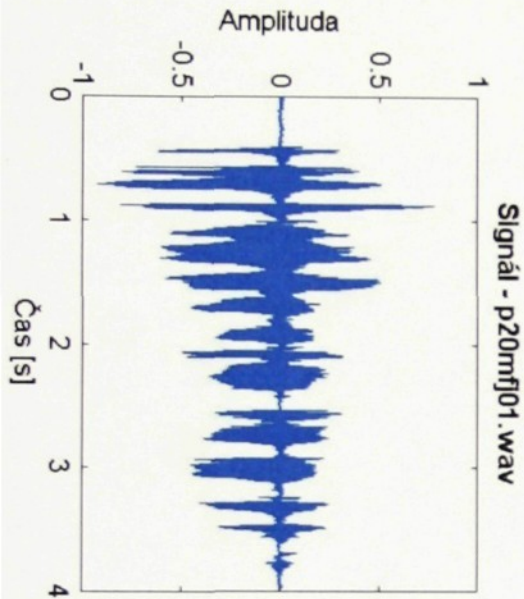
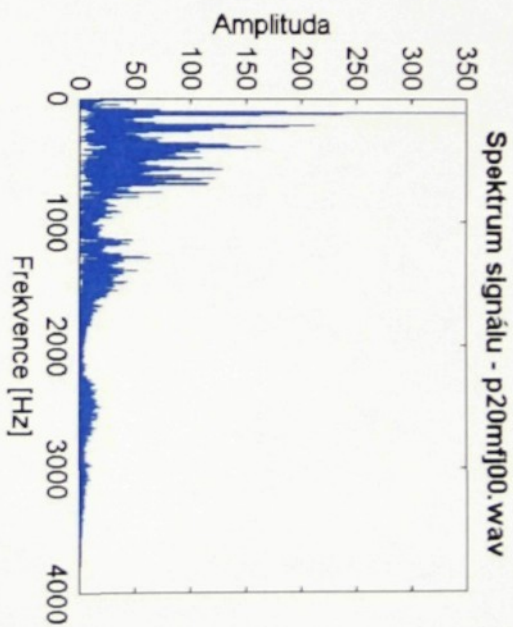
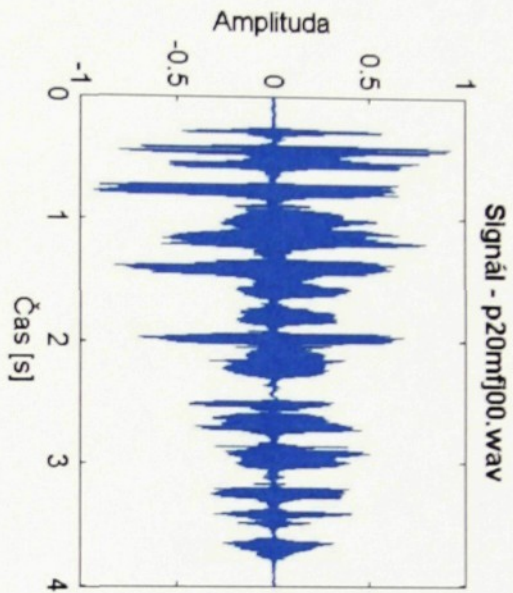


Obr.4.18 Graf výsledku pokusu o zvýšení robustnosti GMM systému

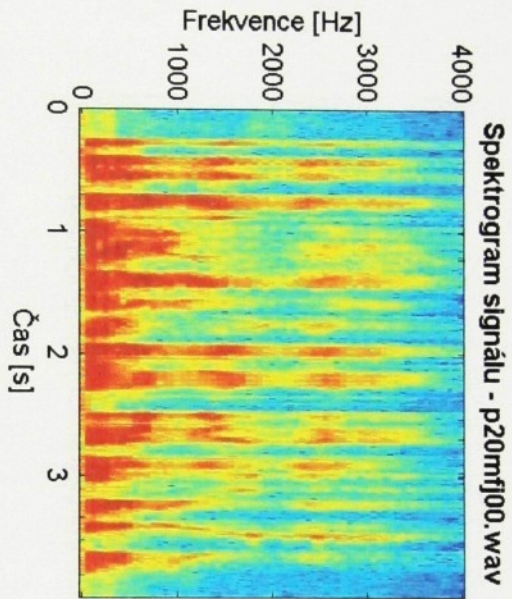


Obr.4.19 Graf výsledku pokusu o zvýšení robustnosti VQ systému

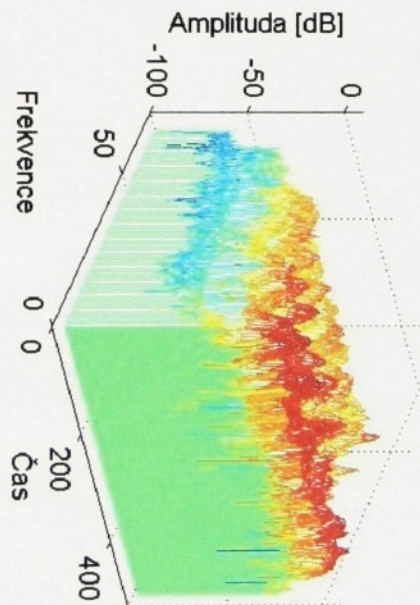
- Analýza časového průběhu řečového signálu



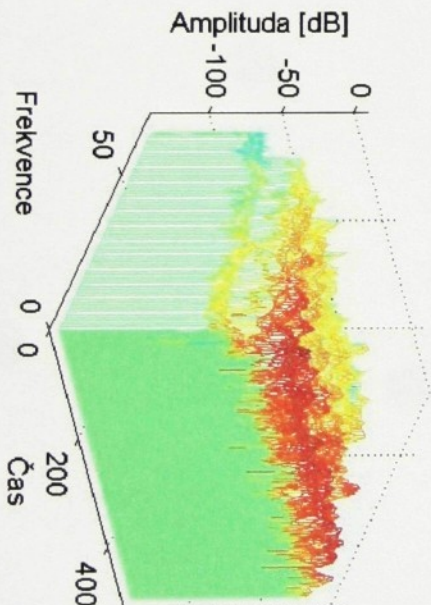
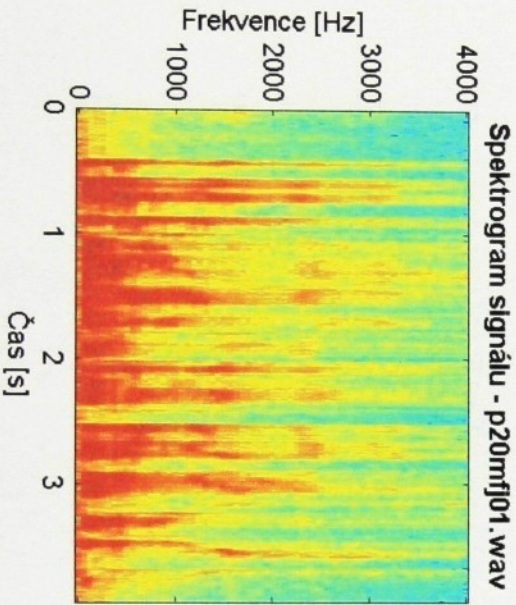
- Analýza spektra řečového signálu



Síťový spektrogram signálu - p20mfj00.wav

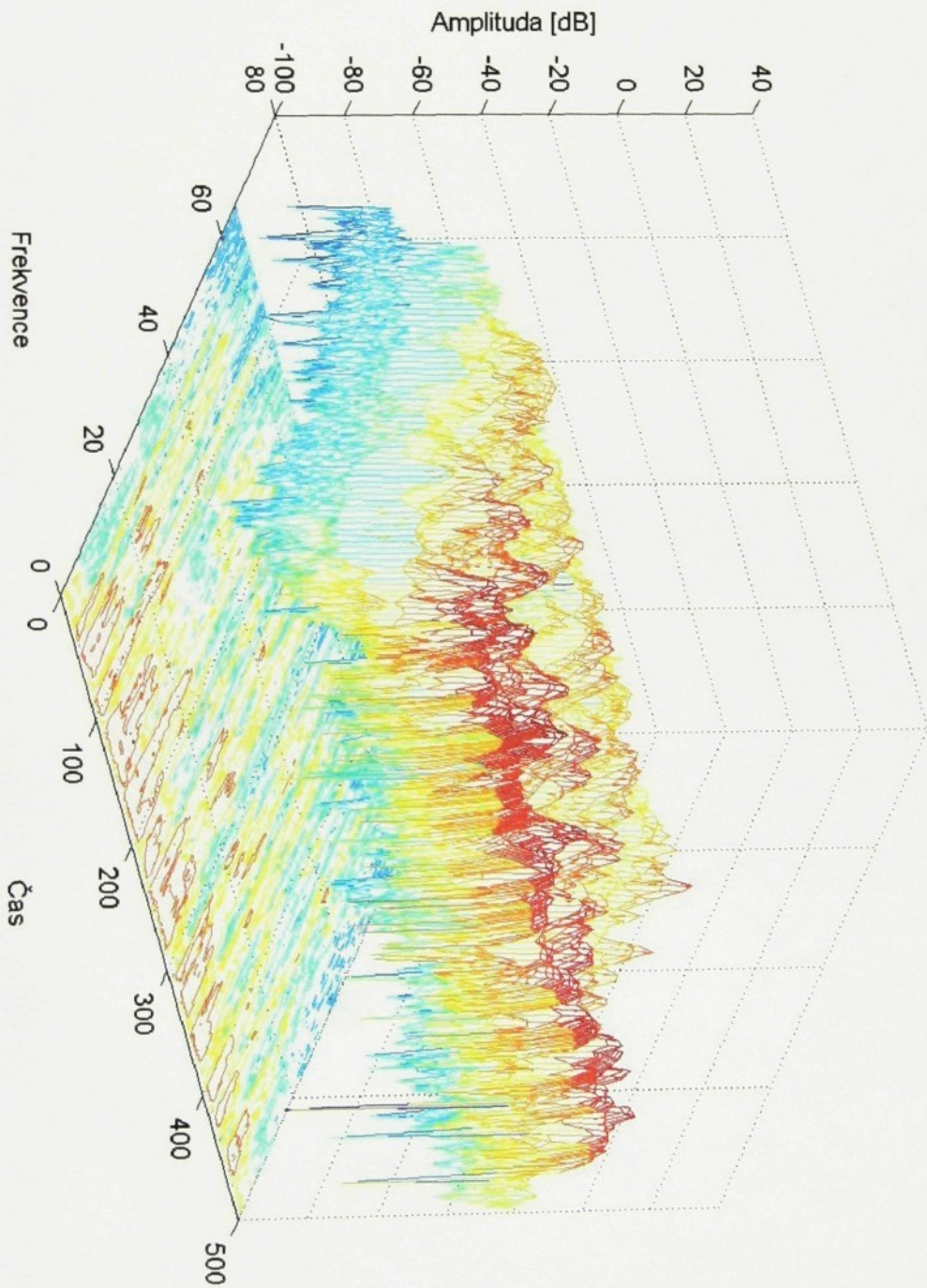


Síťový spektrogram signálu - p20mfj01.wav



- Spektrogram signálu p20mfj00.wav

Síťový spektrogram signálu - p20mfj00.wav



- Spektrogram signálu p20mfj01.wav

