OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

V současné době je úloha rozpoznávání řeči na jednom z předních míst pozornosti. Na jedné straně jde o snahu zjednodušit komunikaci mezi člověkem a strojem, na druhé jde o využití v oborech, které se řečí přímo nebo nepřímo zabývají.

V současnosti je dokonalé rozpoznávání řeči nemožné především kvůli výpočetním obtížím a problémům s rozpoznáváním plynulé řeči. Jde o potíž s chybovou posloupností řeči a růzností slov pokaždé jinak vyslovenými. Nemluvě ještě o různorodosti řečníků lišících se přízvukem, jazykem atd.

Tato práce obsahuje základní poznatky z oboru, seznamuje čtenáře s biometrickými systémy a jejich použitím, a dále se zaměřuje na jejich dílčí část, a tou je řeč a její rozpoznávání. První kapitola se zabývá biometrickými systémy a jejich metodami. Ve druhé kapitole je objasněn vznik řeči a její charakteristika. Poté následuje popis rozpoznávání řeči a obvyklých přístupů k tomuto problému. Třetí kapitola se tedy zabývá základními poznatky v oboru rozpoznávání řeči. V kapitole čtvrté je pozornost zaměřena na konkrétní metodu rozpoznávání řeči, a tou je využití skrytých Markovových modelů. Je zde vysvětlena teorie skrytých Markovových modelů, použití metody při rozpoznávání řeči a dále popis některých funkcí systému Matlab využitelných pro návrh vlastního klasifikátoru.

Ve všech prvních čtyřech kapitolách práce je čtenář seznámen s teorií potřebnou k pochopení principů řeči, klasifikátorů řeči a využití skrytých Markovových modelů v této úloze. Takový teoretický základ poté usnadňuje pochopení přístupu k návrhu vlastního systému rozpoznávání řeči. Samotný návrh je zařazen v kapitole číslo pět, která je také kapitolou nejrozsáhlejší.

Problematika rozpoznávání řeči je stále ve výzkumu a k úplnému, bezproblémovému rozpoznání jakékoliv promluvy jakéhokoliv řečníka je stále ještě dlouhý kus cesty.

1. BIOMETRICKÉ SYSTÉMY

1.1 BIOMETRIE

Jedná se o vědní obor zkoumající živé organismy. V jeho pozornosti stojí měřitelné vlastnosti těchto organismů, především však člověka. Měřitelné vlastnosti jsou v tomto případě tři druhy charakteristik, fyziologické, anatomické a behaviorální. Nejpodstatnější využití oboru biometrie lze spatřovat v potřebě rychlé a pokud možno neomylné identifikace osob. Typická je vysoká úroveň zabezpečení. Naléhavost rozvoje podobných disciplín je v dnešní době více než zjevná. Zrychlování životního tempa a nárůst počtu obyvatel využívajících prostředky bezprostředně spjatých s osobní identitou, si vyžadují praktické využívání možností biometrie, v neposlední řadě jde i o potírání zločinu nebo čím dál víc aktuální boj s terorismem. Zavedení biometrických systémů slibuje snížení počtu „krádeží identity“. Možnosti aplikace jsou prakticky kdekoliv v oblastech našich všedních dnů, počínaje přístupem do budov, dopravních prostředků, manipulací s financemi a účty atd. Jednodušeji řečeno, možnosti biometrie nám mohou pomoci „zbavit se peněženky se všemi kartami, i kilogramů klíčů v kapsách“, a kromě jiného tak odpustit naší zapomnětlivosti a znemožnit jejich zneužití. Pro ilustraci: jde o občanské průkazy, id karty, kreditní a debetní karty, pasy, identifikaci osob bez nutnosti fyzické přítomnosti v místě ověření (např.: nákupy přes internet).

Biometrické systémy jsou využívány třemi způsoby a to pro účely:

- Identifikace
- Verifikace
- Autentizace

V případě **identifikace** jde o porovnávání určitého vzorku s referenčními v databázi, jde tedy o vyhledání co možná nejpodobnějšího vzorku. U **verifikace** jde o jiný přístup, tím je porovnávání vzorku s jedním referenčním, výsledkem je tedy

údaj o tom, jak moc jsou si vzorek aktuální a referenční podobny. **Autentizace** potvrzuje nebo vyvrací autentičnost vzorku [1].

Praktické využití biometrie při aplikaci na lidského jedince:

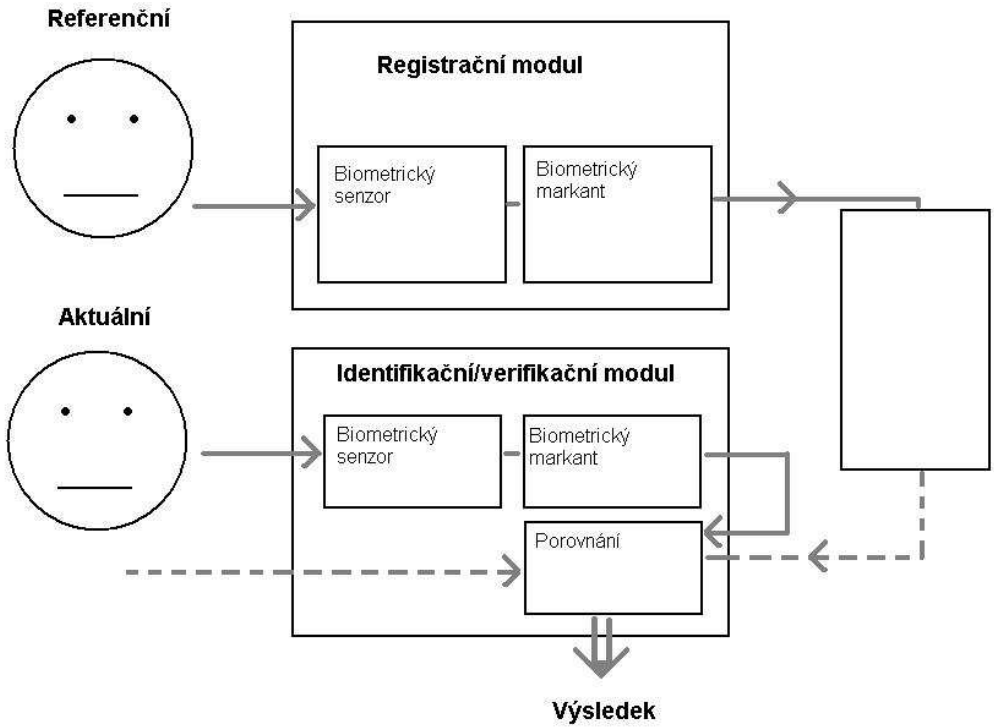
1. prostřednictvím DNA
2. prostřednictvím otisku prstu, popř. vlastností ruky
3. prostřednictvím oční duhovky
4. prostřednictvím tvaru obličeje
5. prostřednictvím podpisu
6. prostřednictvím hlasu

Výčet je samozřejmě mnohem obsáhlejší, v centru pozornosti však stojí těchto šest zaměření. Jako nejspolehlivější se jeví první tři možnosti, neboť každý jedinec má unikátní DNA, otisk prstu a stavbu oční duhovky. Pro identifikaci tvarem obličeje je výhodou užití v davu. Pro hlasovou identifikaci je uplatnění např. v telefonických aplikacích jako jsou transakce přes telefon apod. Princip biometrické identifikace spočívá v porovnávání okamžitého biometrického měření s referenčním uloženým v databázi. Ještě je dobré zmínit, že se biometrické vlastnosti ve výčtu dělí na statické a dynamické, poslední dvě jsou tudíž dynamické a zbývající ve většině případů statické. U dynamických vlastností bývá zpracování obtížnější. Pro představu: při rozpoznávání hlasu (měří se kmitočet, výška, tón hlasu) může hrát podstatnou roli to, v jakém stavu se člověk nachází, zda je pod nátlakem, ve stresu, nachlazený apod., protože některé charakteristické rysy se mohou odlišovat od těch v normálním stavu.

Pro zvýšení zabezpečení je možné pracovat s více vlastnostmi jedince najednou. Například sledováním obličeje i podpisu v rámci jedné identifikace/verifikace.

1.2 BIOMETRICKÉ SYSTÉMY A JEJICH METODY

Obecný model úloh biometrických systémů je tvořen modulem registračním a verifikačním/identifikačním. Blokové schéma na obr. 1.1



Obr. 1.1 Blokové schéma identifikace/verifikace biometrickým systémem

Samotný proces pak spočívá v rozpoznávání vzorů. Třída je tvořena charakteristickým znakem.

Obr. 1.2. Obecné blokové schéma systému rozpoznávání vzorů

V rámci předzpracování nám jde i o potlačení veškerých nežádoucích jevů znemožňujících či zkreslujících požadované výstupy. Například použití filtru pro odstranění šumu a nadbytečného akustického pozadí ze zkoumaného vzorku řeči.

Pro přesnost vyhodnocování biometrickými systémy existují parametry udávající jejich chybovost:

FAR – míra chybného přijetí je pravděpodobnost, že dva odlišné biometrické vzorky budou klasifikovány jako shodné, a systém tak způsobí selhání při odmítnutí

FRR – míra chybného odmítnutí je pravděpodobnost, že dva biometrické vzorky od stejné osoby budou klasifikovány jako neshodné, a systém tak způsobí selhání při přijetí

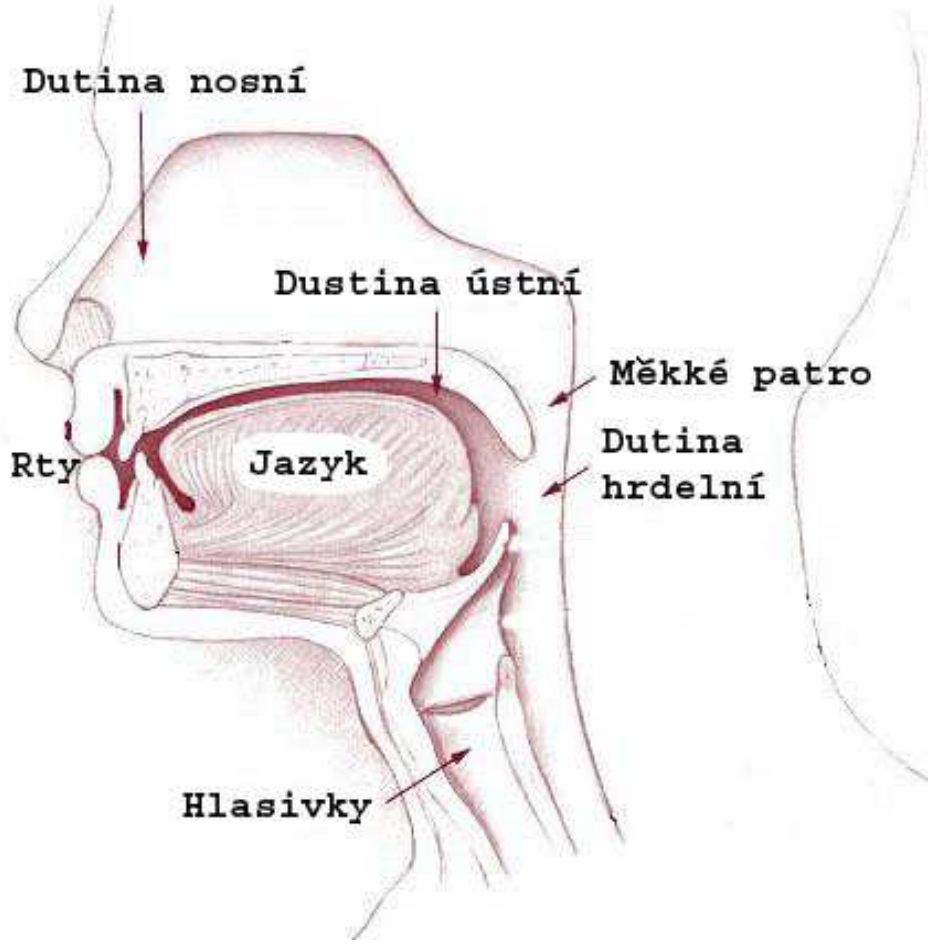
FMR – míra chybné shody je podíl chybně akceptovaných osob

FNMR – míra chybné neshody je podíl chybně neakceptovaných osob [1]

2. ŘEČ

2.1 VYTVÁŘENÍ ŘEČI

Fyzikální podstata vzniku řeči tkví v produkci řečových kmitů prostřednictvím řečových orgánů. Jejich struktura je znázorněna na obr. 2.1. Zdrojem energie jsou plíce s dýchacími svaly.V horní části hrtanu se nacházejí hlasivky, jsou orgánem, jehož kmity vyvolané prouděním vzduchu z plic jsou původcem všech znělých zvuků (samohlásek, souhlásek). Frekvence kmitů hlasivek se u lidí pohybuje v rozmezí 150 – 400 Hz a charakterizuje tzv. **základní tón lidského hlasu**, který je obsažen ve všech znělých zvucích.



Obr. 2.1 Struktura řečových orgánů

Samohlásky (vokály) – při jejich artikulaci je žádoucí co nejvolnější průchod vzduchu hlasovým ústrojím. Kromě základního hlasivkového tónu jsou obsaženy i vyšší, zesílené tóny, které nazýváme formanty. Formanty ($F_1, F_2 \ldots$) jsou nejvíce ovlivňovány rty, čelistmi a měkkým patrem.

	A	E	I	O	U
F_1	700 - 1100	480 - 700	300 - 500	500 - 700	300 - 500
F_2	1100 - 1500	1560 - 2100	2000 - 2800	850 - 1200	600 - 1000
F_3	2500 - 3000	2500 - 3000	2600 - 3500	2500 - 3000	2400 - 2900

Tab. 2.1 Přibližné hodnoty formantů českých samohlásek

Souhlásky (konsonanty) – oproti samohláskám je charakterizuje šum přítomný v akustickém spektru hlásek. Jsou vytvářeny vzduchovou turbulencí vznikající třením vydechovaného vzduchu o artikulační orgány (např: rty, jazyk, zuby). Artikulační orgány mohou tvořit přepážky závěrové, úžinové a polozávěrové dle způsobu jakým je s přepážkami pracováno.

Souhlásky		Závěrové				úžinové				polozávěrové	
Párové	neznělé	p	t	ť	k	s	š	f	ch	c	č
	znělé	b	d	ď	g	z	ž	v	h	dz	dž
Nepárové	znělé	m	n	ň		l	j	r	ř		

Tab. 2.2 Přehled a dělení českých souhlásek

Pro analýzu řeči nastává problém při přechodu mezi jednotlivými vyslovenými písmeny. Nastává totiž prodleva, kdy se artikulační orgány přizpůsobují nově vyslovovanému písmenu. Délka a charakter prodlevy závisí na hmotnosti orgánů a zapojených svalech. Problém je o to složitější, že parametry zvuku při prodlevě se mění v závislosti na kontextu dvojice písmen, intonaci i tempu řeči. Tuto závislost označujeme jako **koartikulaci**.

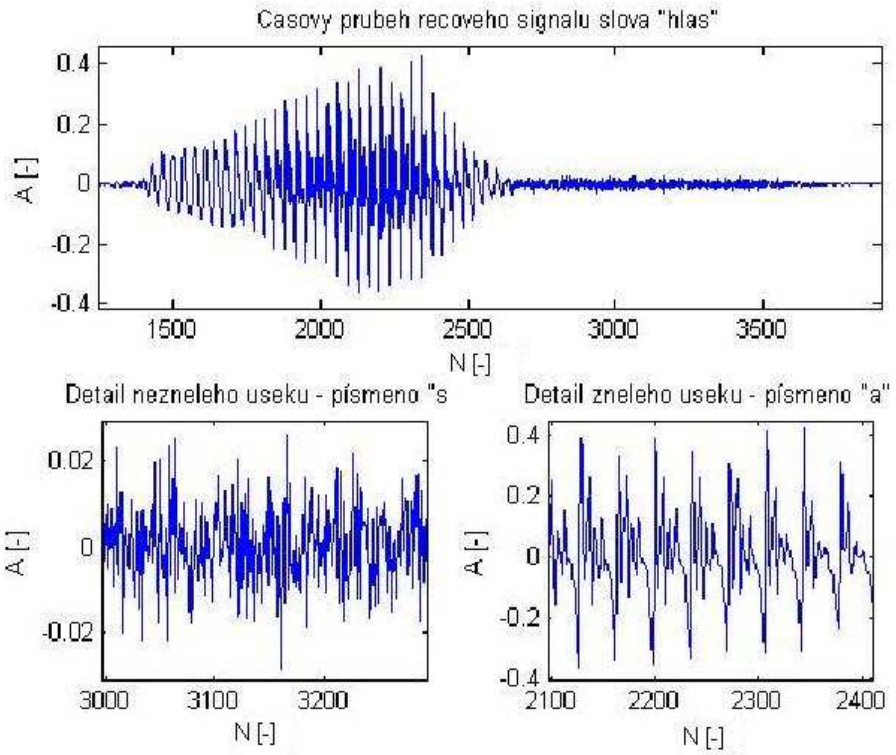
2.2 CHARAKTERISTIKA ŘEČI

Vlivy a mechanismy charakterizující lidskou řeč lze shrnout ve třech kategoriích a to:

- **Akustická struktura** – amplitudově frekvenční spektrum
- **Lingvistická struktura** – gramatika a skladba
- **Subjektivní vlivy osobnosti řečníka** – intonace, rytmus, barva hlasu . . .

Řeč je souvislý, časově proměnný proces, reprezentovaný zvukovými vlnami. Zobrazení řečového signálu je na obr.2.2. Řečový signál je posloupnost diskrétních vzorků signálu vzniklého hlasovým ústrojím člověka. Při bližším prohlédnutí řečového signálu lze nalézt oblasti, které se jeví jako periodické, jde o tzv. znělé části řeči, a ty které se jeví jako neperiodické, tedy logicky neznělé části řeči (dáno skutečností, že v té chvíli výdechový proud vzduchu z plic nerozkmitá hlasivkovou štěrbinu). Perioda znělých částí řeči se nazývá „**základní perioda řeči**“. Pro úplnost je třeba zmínit, že v reálném světě nelze chápat části řeči jako *absolutně* znělé či neznělé. Detaily, těchto průběhů jsou také zobrazeny na obr.2.2.

Jiné dělení v souvislosti s rozpoznáváním řeči rozlišuje charakteristiky nižší úrovně a charakteristiky vyšší úrovně. Do první zmíněné kategorie lze zařadit parametry, které lze snadno měřit. Jde například o frekvence základního tónu nebo amplitudu signálu. Do kategorie charakteristik vyšší úrovně spadají styl vyjadřování řečníka nebo jeho přízvuk. Pro praxi se prozatím využívají jen charakteristiky nižší úrovně, protože jsou mnohem snadněji číselně vyjádřitelné. V budoucnosti se však počítá i s využitím vyšších charakteristik, především pro získání informací o osobním pozadí řečníka. Lze z nich usuzovat odkud řečník pochází, jaké má hlasové dispozice apod.



Obr. 2.2 Časový průběhu signálu slova „hlas“ s detaily znělých a neznělých úseků

3. ROZPOZNÁVÁNÍ SLOV

Klasifikátory na rozpoznávání izolovaných slov jsou ve středu pozornosti výzkumu od poloviny 20. století. Od začátků práce na této problematice byl učiněn velký pokrok, ale cesta k ideálnímu klasifikátoru, který by byl schopen bezpečně rozpoznat libovolného řečníka v libovolném stavu, s libovolným projevem je stále daleká. Jednou z největších překážek na tomto poli je výpočetní náročnost zpracování a s tím související omezené výpočetní prostředky, obzvláště v případech, kdy je očekávána okamžitá odpověď systému v reálném čase s dobou zpoždění maximálně do půl vteřiny od přijetí řečového signálu. Tato podmínka zásadním způsobem omezuje rozmezí použitelných metod. Faktory ztěžující zpracování a následné rozpoznání plynoucí ze strany přijímaného signálu jsou:

- Různorodost hlasu řečníků (tempo, barva hlasu, přízvuk, vada řeči . . .).
- Stav řečníka a situace, v níž se nachází (hlasitost, rychlost promluvy, nálada . . .).
- Šum, rušení a jiné zvukové jevy přítomné v pozadí řečového signálu.

Obecný model rozpoznávání řeči v praxi bývá následující. Řečový signál je většinou nejprve zpracován některou metodou krátkodobé analýzy, poté je vyjádřena posloupností krátkodobých charaktristik (např. LPC, PARCOR, autokorelačních, kepstrálních koeficientů či výstupů pásmového filtru). Jednotlivé číselné údaje jsou nazývány **příznaky**, a jejich množiny **obrazy**. Samotné rozpoznávání probíhá rozřazováním obrazů do **tříd**, které představují slova ve slovníku (referenční).
Aplikované metody rozpoznávání izolovaných slov lze shrnout do tří nejpreferovanějších přístupů:

- Slovo je zpracováváno jako celek. Srovnávání probíhá hledáním nejmenší vzdáleností mezi obrazem rozpoznávaného slova a vzorovým obrazem. Vzdálenost bývá vyhodnocena metodou **dynamického programování**. Tato metoda spočívá v nalezení takové nelineární transformace časové osy, ve které je zmíněná výsledná vzdálenost nejmenší.

- Slovo je modelováno pomocí tzv. **skrytých Markovových modelů**. Slova jimi lze modelovat buď jako celky jediným modelem, anebo jsou modelovány subslovní jednotky, z nichž je celistvé slovo složeno. Rozřazení do tříd probíhá způsobem, kdy jsou stanoveny parametry každé třídy a slovo je umístěno do té, jejíž model je generován nejpravděpodobněji. Jde o statistické metody.
- Slovo je zpracováno ve dvou stupních. Akustickou analýzu následuje rozdělení řečového signálu na úseky, které jsou foneticky dekódovány.Druhým stupněm je rozpoznání podle posloupnosti úseků. Tato metoda bývá upřednostňována pro rozpoznávání souvislé řeči. Aplikovány bývají **strukturální metody rozpoznávání** [2].

V následujícím textu bude blíže popsána metoda modelování pomocí tzv. skrytých Markovových modelů.

4. ROZPOZNÁVÁNÍ POMOCÍ SKRYTÝCH MARKOVOVÝCH MODELŮ

4.1 SKRYTÉ MARKOVOVY MODELY

Skryté Markovovy modely jsou statistické modely s konečným počtem stavů, které jsou vhodné po popis stacionárních úseků signálu[{1}].

Markovův proces G se skrytým Markovovým modelem je definován parametry:

$$G = (Q, V, N, M, \pi) \qquad (4.1)$$

kde

- $Q = \{q_1, \ldots, q_N\}$ je soubor N individuálních stavů Markovova modelu
- $V = \{v_1, \ldots, v_N\}$ je abeceda L výstupních symbolů vektorového kvantizéru, (v našem případě odpovídají jednotlivé výstupní symboly v_1 indexům příslušných spektrálních vzorů v kódové knize V)
- $N = [n_{ij}]$ je **matice přechodu**; prvky určují s jakou pravděpodobností přechází systém ze stavu q_i v čase t do stavu q_j t + 1. Platí tedy :

$$n_{ij} = P(q(t+1) = q_j | q(t) = q_i), \qquad 1 \le i, j \le N \qquad (4.2)$$

- $M = [m_{jl}] = [m_j(l)]$ je **matice pravděpodobností generovaných vzorů**, její prvky určují s jakou pravděpodobností je v kterémkoliv čase t generována l-tá položka konečného souboru spektrálních vzorů, je-li systém ve stavu q_i. Platí tedy:

$$m_{jl} = m_j(l) = P(v(t) = v_l | q(t) = q_j), \quad 1 \le j \le N \qquad 1 \le l \le L \qquad (4.3)$$

- $\pi = [\pi_i]$ je sloupcový vektor **pravděpodobností počátečního stavu**. Platí tedy:

$$\pi_i = P(q(1) = q_i), \qquad 1 \le i \le N \qquad (4.4)$$

[{1}] V případě promluvy lze považovat za stacionární úsek, v němž se signál výrazně nemění. Např. mikrosegment fonému.

Soubor parametrů Markovova modelu je definován jako:

$$\lambda = (N, M, \pi) \tag{4.5}$$

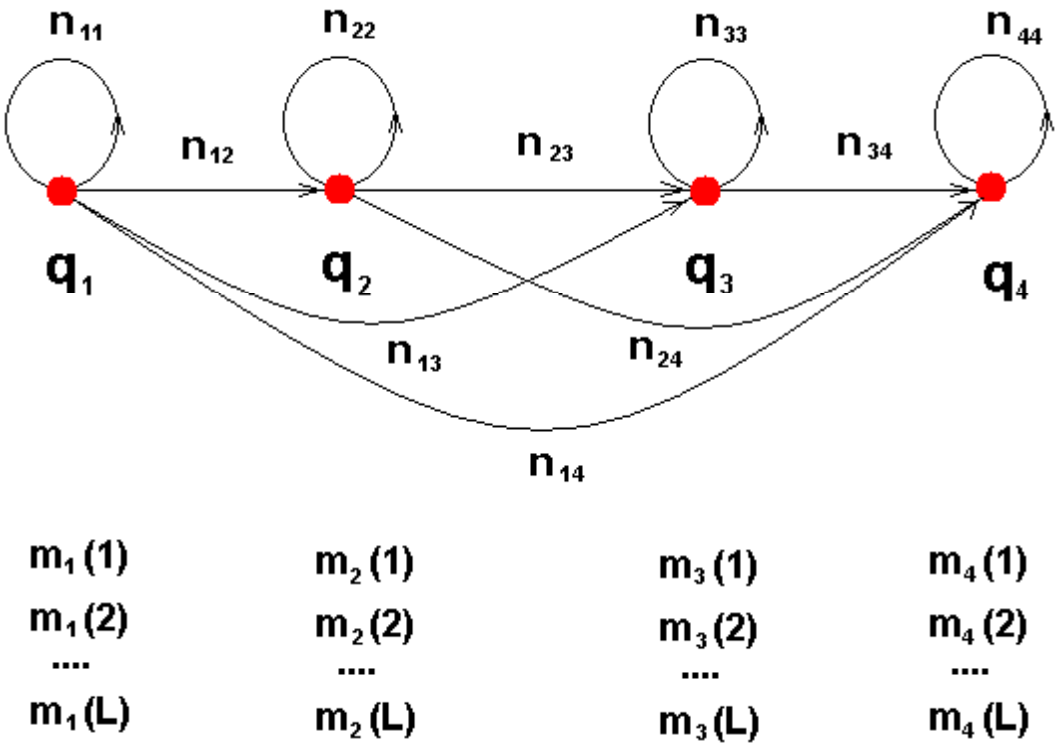
Podmínky pro parametry π_i, n_{ij}, $m_j(l)$:

$$\sum_{i=1}^{N} \pi_i = 1 \tag{4.6}$$

$$\sum_{j=1}^{N} n_{ij} = 1 \qquad \text{pro } i = 1, \ldots, N \tag{4.7}$$

$$\sum_{l=1}^{L} m_j(l) = 1 \qquad \text{pro } j = 1, \ldots, N \tag{4.8}$$

Pro příklad modelu na obr. 4.1 platí: $n_{ij} > 0$ *pro* $i \leq j$, $n_{ij} = 0$ *pro* $i > j$



Obr. 4.1 Ilustrace jednoho typu 4-stavového skrytého Markovova modelu

Hlavní parametry charakterizující model jsou pravděpodobnosti přechodů mezi stavy q_n v matici N, a matice pravděpodobností generovaných vzorů M. Pravděpodobnosti přechodů vyjadřují různou délku trvání řečových segmentů, jejichž statistické parametry lze přiřadit jednotlivým stavům. Na druhé straně matice pravděpodobností generovaných vzorů vyjadřuje změny ve spektrálním obsahu mikrosegmentů náležících danému stavu. Jak již bylo zmíněno, přechody mohou nastat jen směrem vpravo od stávajícího stavu, tudíž jsou označeny jako levo-pravé.

Ještě je vhodné zmínit, že pozorovatel vidí jen výstup náhodných funkcí, stavy podpůrného Markovova řetězce sledovat nemůže, proto jsou tyto Markovovy modely nazývány skryté.

4.2 PRINCIP ROZPOZNÁVÁNÍ

Skryté Markovovy modely jsou využívány pro rozpoznávání slov již od sedmdesátých let minulého století. Má velikou úspěšnost při rozpoznávání souvislé řeči. Princip metody modelování řeči Markovovými modely je založen na představě o vytváření řeči. Představa je taková, že se lidské hlasové ústrojí nachází v určitém artikulačním stavu pro každý mikrosegment, ze kterého se promluva sestává. V takovém mikrosegmentu je ústrojím produkován krátký signál, který lze popsat jednou z konečného počtu spektrálních charakteristik. Je vytvořena kódová kniha typových spektrálních vzorů a každá spektrální charakteristika je nahrazena indexem nejpodobnějšího typového spektrálního vzoru z kódové knihy. Vytvoření kódové knihy lze dosáhnout vektorovou kvantizací.

Klasifikátory řeči založené na modelování řečového signálu pomocí Markovových modelů generují pomocí Markovova procesu dvě svázané časové posloupnosti náhodných proměnných. Jedním je podpůrný Markovův řetězec, ten je posloupností konečného počtu stavů, a druhým je řetězec konečného počtu spektrálních vzorů. Všem spektrálním vzorům jsou vytvořeny náhodné funkce, které pravděpodobnostně hodnotí vztah mezi vzory a stavy. Podpůrný Markovův řetězec pak mění stavy podle své matice pravděpodobností přechodu.

Pro modelování řeči jsou prioritně užívány tzv. levo-pravé Markovovy modely, které jsou vhodné pro modelování procesů rozvíjejících se s postupujícím časem. Proces začíná příchodem prvního spektrálního vzoru z počátečního stavu modelu a s postupujícím časem přechází stavy s nižšími indexy do stavu s vyššími indexy, anebo setrvávají ve stávajícím stavu. Konec nastává s posledním spektrálním vzorem.

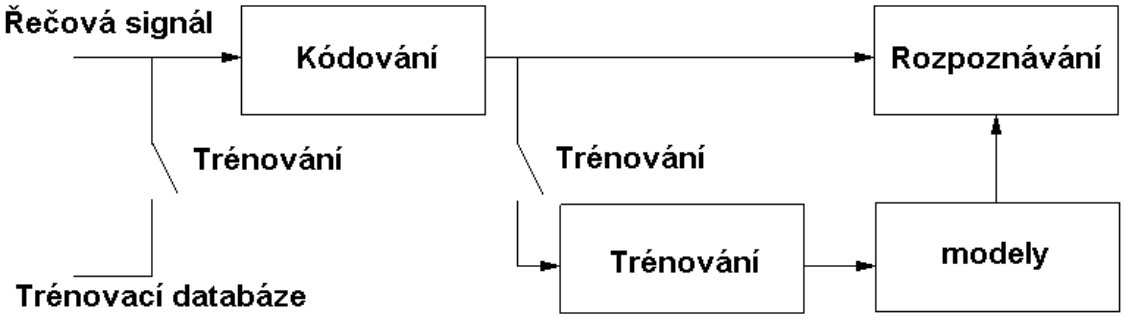
Počet stavů závisí na délce mikrosegmentů a odpovídá tedy průměrnému počtu mikrosegmentů. Experimentálně bylo zjištěno, že redukovaný počet stavů z původních desítek na asi pět, umožnil redukci parametrů při zachování dostatečné přesnosti rozpoznávání. Při příliš vysokém počtu mikrosegmentů si byly sousední stavy navzájem velmi podobné a jejich vypovídací hodnota se tedy nijak výrazně nelišila, proto byl jejich počet snížen na hodnotu přibližně odpovídající počtu písmen ve slově, popřípadě počtu subslovních jednotek, které v sobě nesou i informaci o koartikulačních přechodech. Toto zjištění umožnilo významné snížení počtu parametrů modelu a tudíž mnohem snazší trénování a menší objem slovníku, s čímž také souvisí rychlejší práce s ním.

Pro rozměrnější slovníky se setkáváme při trénování každého samostatného slova s těžkopádností a zdlouhavostí. Proto je pro zjednodušení a urychlení procesu trénování skrytých Markovových modelů vhodné užít menší jednotky řeči než jsou slova, například slabiky, **fonémy** atd. **Fonémem** rozumíme nejmenší součást zvukové stránky řeči s rozlišovací funkcí.

Trénování je režim, kdy řečník nebo řečníci vysloví postupně všechna slova, která mají být obsažena ve slovníku, lépe vícekrát, a tato slova jsou popsána příznaky. Slovo lze reprezentovat posloupností fonémů a každému fonému odpovídá určitý skrytý Markovův model. Z toho vyplývá, že modely slov lze vytvářet zřetězováním modelů jednotlivých fonémů. Zavedení fonémů namísto slov usnadnilo také flexibilitu slovníku, neboť není potřeba zařazovat celá nová slova. Přímý průběh modelem odpovídá průměrné délce slova. Pokud nastane přechod, který setrvává v předchozím stavu, jde o prodloužení slova, pokud nastane přechod do stavu s vyšším indexem, jedná se ve výsledku o zkrácení slova. Toto nám dovoluje identifikovat slovo, nezávisle na délce jeho vyslovení.

Díky uplatnění fonémů namísto slov lze zvýšit univerzálnost systému, ale naproti tomu je přesnost klasifikace slov nižší než u použití Markovových modelů celých slov, a to z toho důvodu, že při použití fonémů je nutno obejít se bez detailů, jakými jsou koartikulační informace. Pro zachování detailů jako jsou koartikulační přechody bylo navrženo vytvořit slovníky, které obsahují subslovní jednotky menší než jsou fonémy. Takový postup nám zvýší úspěšnost rozpoznávání se zachovalou flexibilitou na úkor rychlosti, ale to už je otázka kompromisu pro splnění preferencí a potřeb uživatele [2].

Rozpoznávače řeči jsou obecně tvořeny částmi, které jsou znázorněny na Obr. 4.2 ve zjednodušeném blokovém schématu. Prvotní operací s řečovým signálem bývá kódování, které v sobě zahrnuje segmentaci na úseky stejných délek, z nichž je vypočítáno spektrum, popřípadě jiné charakteristické parametry promluvy. Pro umožnění rozpoznávání promluvy je dále potřeba trénování jejího modelu pomocí trénovací databáze, kde jsou uloženy různé záznamy stejných promluv pro slovník klasifikátoru. Poslední částí je rozpoznávání, které spočívá ve vyhodnocování pravděpodobnosti, že promluva byla generována daným modelem.



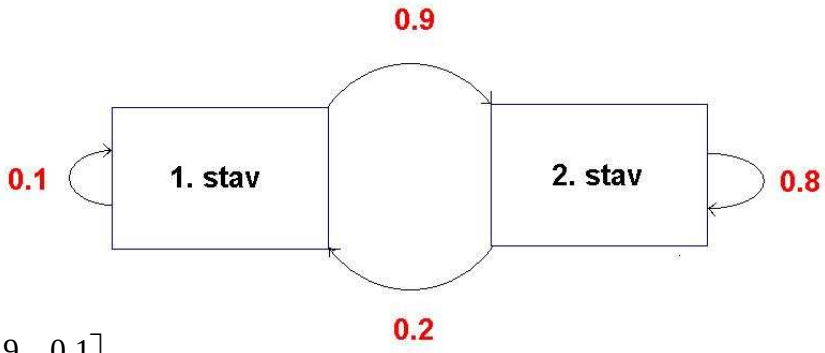
Obr. 4.2 Blokové schéma rozpoznávače řeči

4.3 SKRYTÉ MARKOVY MODELY V SYSTÉMU MATLAB

4.3.1 Markovovy řetězce:

Markovovy řetězce jsou matematickým popisem Markovových modelů, jsou charakterizovány:

- Souborem stavů {1, 2, ... N}
- Maticí T s rozměry N *X* N popisující pravděpodobnosti přechodů mezi stavy vyjádřené indexy prvků matice. Vztah mezi stavovým diagramem a maticí T je znázorněn na obr. 4.3. Je zřejmé, že součet prvků v každém řádku je roven 1.
- Souborem možných výstupů {s_1, s_2, . . . s_N}
- Maticí výstupu E s rozměry N *X* L, jejíž indexy udávají pravděpodobnost výstupu symbolu s_K



$$T = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$$

Obr. 4.3 Blokové Vztah mezi stavovým diagramem a maticí T

4.3.2 Analýza skrytých Markovových modelů

4.3.2.1Nastavení modelu a generování dat

Prvním krokem je vytvoření matic přechodu (transition) a výstupů (emission) například zadáním následujících příkazů:

```
TRANS =    [.9    .1;
            .2    .8;];
```

EMIS = [1/6, 1/6, 1/6, 1/6, 1/6, 1/6;
7/12, 1/12, 1/12, 1/12, 1/12, 1/12]; {1}

Poté náhodné vytvoření sekvence výstupů modelu **`seq`** o délce 1000 užitím funkce **`hmmgenerate`**. Je možné také nastavit, aby byla vrácena odpovídající náhodná sekvence stavů modelu jako druhý výstup ve vektoru **`states`**. Příkaz je zadán následovně:

```
[seq, states] = hmmgenerate(1000, TRANS, EMIS);
```

Pozn.: při generování sekvence **`seq`** a **`states`** funkcí **`hmmgenerate`** se začíná s modelem ve stavu i0 = 1 v kroku 0. Poté model vytvoří přechod do stavu i1 v kroku 1 a vrátí i1 jako první vstup stavu.

4.3.2.2Výpočet nejbližší sekvence stavů

Za předpokladu, že jsou známy matice přechodů a výstupů (TRANS, EMIS), můžeme vypočítat nejbližší sekvenci stavů, kterou generuje pozorovaná sekvence výstupů **`seq`**. To je možné pomocí příkazu hmmviterbi, který využívá Viterbiho algoritmus k výpočtu nejbližší sekvence stavů, kterými model prochází při generování dané sekvence výstupů. Příkaz je zadán následujícím způsobem:

```
likelystates = hmmviterbi(seq, TRANS, EMIS);
```

Pozn.:**`likelystates`** a **`seq`** mají stejnou délku. Přesnost výsledku můžeme ověřit podílem shodných hodnot obou vektorů jejich délkou.

{1} V *případě matice TRANS jde o přechody stavů z obr. 4.3 a EMIS je maticí výstupů, kde v prvním řádku jsou možnosti hodů 6-stranou hrací kostkou s hodnotami 1-6 a ve druhém řádku o 12-stranou kostku se sedmi stěnami označenými hodnotou 1 a zbývajícími 2-6. Dle přechodů je vrhána buď jedna nebo druhá kostka*

4.3.2.3Odhad matic přechodů a výstupů

Za předpokladu, že neznáme matice přechodů a výstupů v modelu, ale pozorujeme sekvenci výstupů **seq**, můžeme užít dvou funkcí k přibližnému výpočtu těchto neznámých matic: hmmestimate nebo hmmtrain

hmmestimate

K užití této funkce je potřeba znát odpovídající sekvenci stavů, kterými model prochází při generování sekvence **seq**. Následující příkaz ze sekvencí stavů (**states**) a výstupů (**seq**) přibližně vypočítá matice přechodů (TRANS_EST) a výstupů (EMIS_EST).

```
[TRANS_EST, EMIS_EST] = hmmestimate(seq, states)

TRANS_EST =   0.8989   0.1011
              0.0585   0.9415

EMIS_EST =    0.1721   0.1721   0.1749   0.1612   0.1803   0.1393
              0.5836   0.0741   0.0804   0.0789   0.0726   0.1104
```

hmmtrain

Neznáme-li sekvenci stavů jako v předchozí funkci, ale máme počáteční odhad hodnot matic TRANS a EMIS, můžeme k přibližnému vypočítání těchto matic využít funkci hmmtrain.

Příklad počátečního odhadu:

```
TRANS_GUESS = [.85    .15;
               .1     .9];

EMIS_GUESS =  [.17    .16    .17    .16    .17    .17;
               .6     .08    .08    .08    .08    .08];
```

Zadání příkazu pro výpočet:

[TRANS_EST2, EMIS_EST2] = hmmtrain(seq, TRANS_GUESS, EMIS_GUESS)

TRANS_EST2 = 0.2286 0.7714
0.0032 0.9968

EMIS_EST2 = 0.1436 0.2348 0.1837 0.1963 0.2350 0.0066
0.4355 0.1089 0.1144 0.1082 0.1109 0.1220

Tato funkce využívá iterativní algoritmus, to znamená, že s každým následujícím krokem je výpočet matic blíže výsledku než v kroku předchozím. Konec výpočtu nastává buď s určenou tolerancí mezi dvěma kroky iterace, popřípadě zastaví po určeném počtu kroků, není-li tolerance dosaženo. Implicitní počet kroků je 100.

4.3.2.4Výpočet pravděpodobností následujících stavů

Pravděpodobnosti následujících stavů výstupní sekvence **seq** jsou pravděpodobnosti, které předpokládají, že se model nachází v daném stavu, když generuje hodnotu v sekvenci **seq** danou jejím voláním. Výpočet pravděpodobností následujících stavů lze provést funkcí hmmdecode:

PSTATES = hmmdecode(seq, TRANS, EMIS)

Výstup PSTATES je matice M *x* L, kde M je počet stavů a L je délka sekvence. Jinými slovy PSTATES(i,j) je předpokládaná pravděpodobnost, že se model nachází ve stavu *i*, když generuje *j*-tý symbol **seq** daný voláním **seq**.

Pozn.: Funkce hmmdecode začíná s modelem ve stavu 1 v kroku 0 předcházejícím první výstup. PSTATES(i,1) je pravděpodobnost, že model bude ve stavu *i* v následujícím kroku 1.

4.4 KEPSTRÁLNÍ ANALÝZA V SYSTÉMU MATLAB

Kepstrální analýza je nelineární technika zpracování signálů užívaná v mnoha aplikacích, především pak právě ke zpracování řečových signálů.

Komplexní kepstrum sekvence *x* je vypočítáno nalezením přirozeného komplexního logaritmu Fourierovy transformace *x* a následné aplikace inverzní Fourierovy transformace výsledné sekvence.

Tuto operaci lze v systému Matlab provést prostřednictvím funkce cceps. Přibližně vypočte komplexní spektrum vstupní sekvence. Výsledkem je reálná sekvence o stejné délce jako sekvence vstupní.

```
c = cceps(x);
```

Reálné kepstrum signálu *y*, je vypočítáno určením přirozeného logaritmu magnitudy Fourierovy transformace y, poté získáním inverzní Fourierovy transformace výsledné sekvence.

```
r = rceps(y);
```

5. NÁVRH VLASTNÍHO ROZPOZNÁVACÍHO SYSTÉMU

5.1 PŘÍPRAVA

Návrh systému rozpoznávajícího izolovaná slova je složitý problém, jehož řešení je vhodné si rozdělit na více dílčích částí. Pro systematický postup při řešení byly navrženy následující kroky:

- Pořízení a předzpracování řečového signálu
- Vytvoření signálových charakteristik
- Vektorová kvantizace a kódová kniha
- Trénování modelu slova
- Vyhodnocení pravděpodobnosti shody promluvy

Řešení každé z těchto částí je podrobně popsáno v následujících podkapitolách.

5.2 POŘÍZENÍ A PŘEDZPRACOVÁNÍ ŘEČOVÉHO SIGNÁLU

První praktickou částí projektu, kterou bylo potřeba se zabývat bylo vytvoření databáze se zaznamenanými referenčními vzorky. Nejdříve bylo nutné nahrát požadovaná slova. V tomto kroku bylo použito běžného řečnického mikrofonu. Pro záznam do digitální podoby dostatečně posloužil program „Záznam zvuku“, který je standardní součástí operačního systému Windows XP. Záznam má tato specifika: vzorkovací frekvence - 44,1 kHz, rozlišení - 16 bitů, stereo, formát wav. V matlabu lze tyto vzorky načíst s pomocí příkazu wavread jehož parametrem je název wav souboru. Tímto způsobem bylo zaznamenáno 9 různých slov s 10 vysloveními pro každé z nich.

Následně je potřeba pro snadnější práci převést stereo signál na mono signál. Vzhledem k tomu, že zvukový stereo signál je v matlabu reprezentován vektorem hodnot pro každý kanál, je nejjednodušší cestou k získání mono signálu vytvoření jediného vektoru, jehož prvky jsou aritmetickým průměrem původních hodnot se stejným indexem.

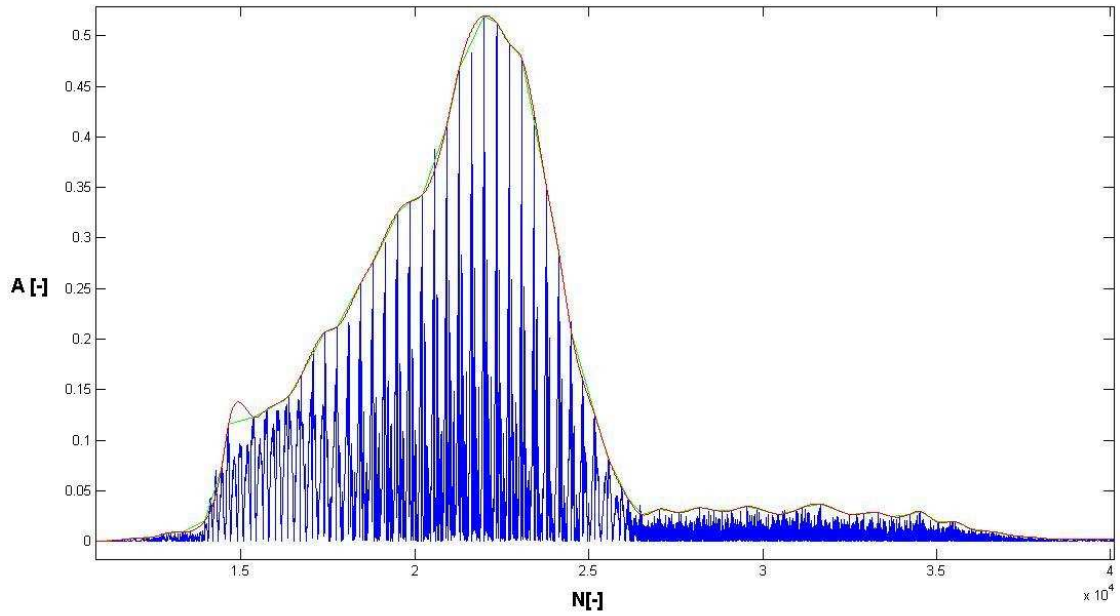
Po této úpravě je vhodné nahradit stávající vektor vzorku novým vektorem, který obsahuje pouze hodnotný signál, kterým je samotné slovo od počátku do konce bez hraničních prodlev, které vznikly během zaznamenávání signálu. Prakticky je tento proces uskutečněn tak, že jsou ze signálu vybrány pouze hodnoty v intervalu mezi body, kde poprvé a poté naposled amplituda přesáhne 15% svého maxima. Mez 15% maximální amplitudy byla nalezena empiricky tak, aby zůstaly zachovány veškeré užitečné informace o signálu, ztráty byly minimální, a zároveň došlo k ignorování případného šumu. Pro účely nalezení optimální meze byla vytvořena obálka signálu. Prakticky je tento proces uskutečněn tak, že jsou vytvořeny druhé mocniny ze všech hodnot původního vektoru, poté rozdělen signál na dostatečně krátké úseky (pokusně zjištěná délka 50ms), a nalezeno maximum každého úseku. Je tak získán vektor bodů, jejichž počet odpovídá počtu úseků. Těmito body je proložena křivka interpolací pomocí kubického splajnu. Na obr. 5.1 je zpracovaný průběh slova „hlas“, modře je vyznačena absolutní hodnota signálu, zeleně spojnice maxim a červeně aproximace.

Následuje ukázka výše popsaného algoritmu pro výpočet obálky.

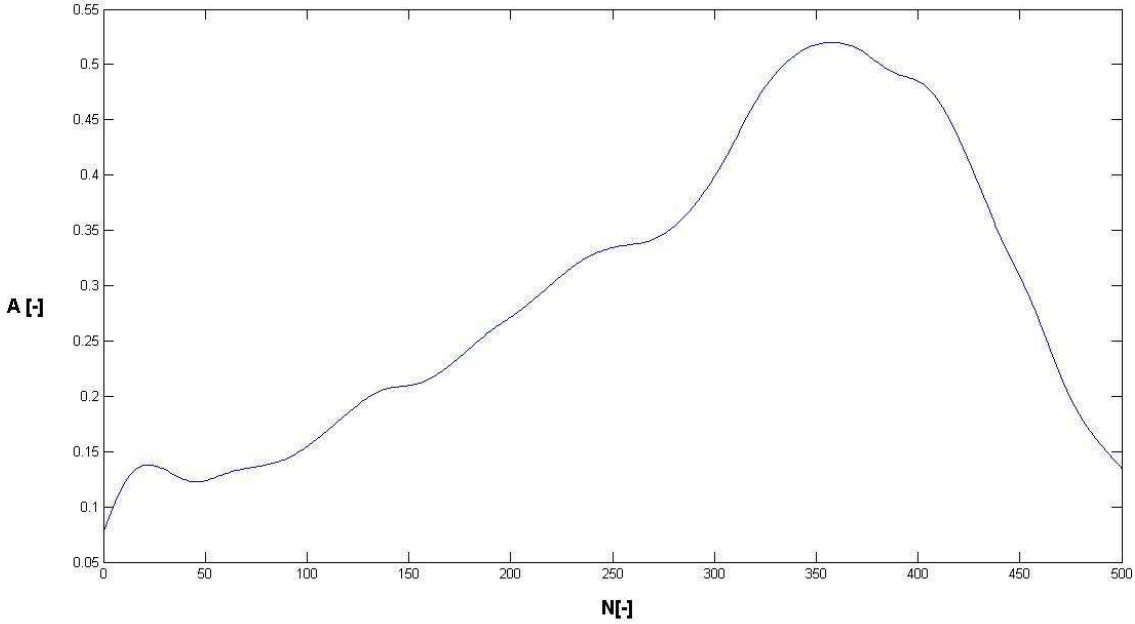
```
% cyklus pro vyhledani vrcholovych hodnot a jejich indexu

for N=1:floor(VZ01size/delka)
 vysek(1:delka) = VZ01mono(((N-1)*delka + 1):((N-1)*delka + delka));
 [A,B] = max(vysek);
 obal(2,N) = (N-1)*delka + B;
 obal(1,N) = A;
end

indexy = obal(2,:); % indexy maxim výseků pro obal
hodnoty = obal(1,:); % konkrétní hodnoty na nalezených indexech

% interpolace kubickym splinem
rozliseni_x = 0:(delka/400):VZ01size;
interpolant = spline(indexy,hodnoty,rozliseni_x);
```

`VZ01size` je délka vektrou zpracovávaného signálu, `delka` je počet úseků, na kterých hledáme maximum. Cyklus `for` hledá maxima s indexy postupně na celém rozsahu slova a ukládá je do matice `obal`. Příkazem `spline` je vytvořena aproximace funkce přes nalezená maxima s jemností zadanou vektorem `rozliseni_x`.

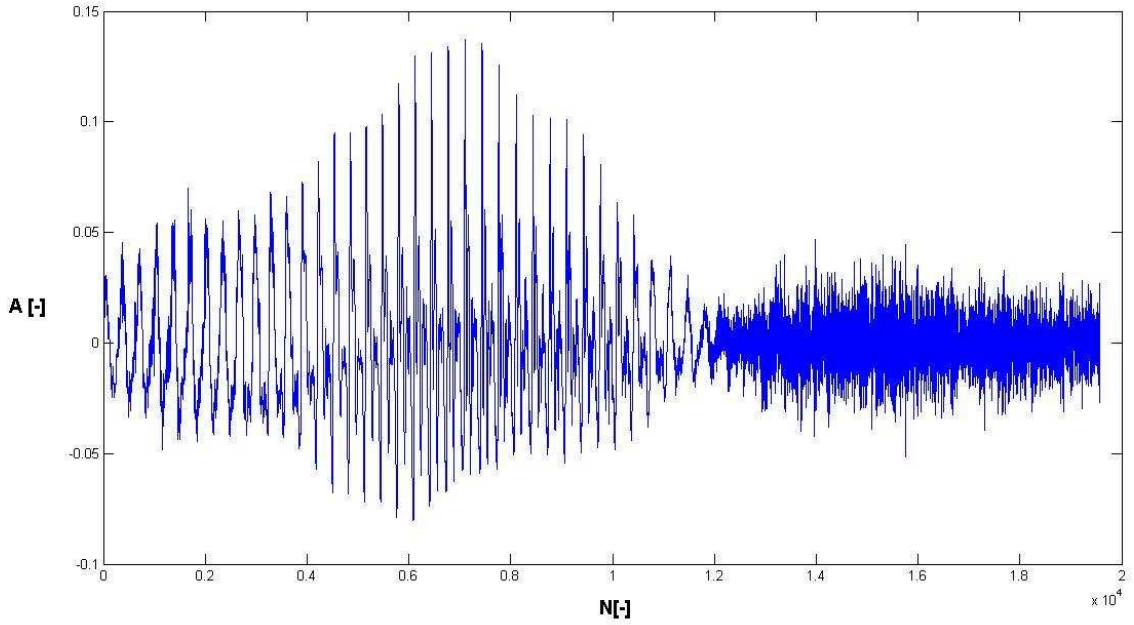
Jak již bylo zmíněno, problémem jsou právě ony počáteční nebo konečné úseky, kde s příliš vysokou mezí dojde k oříznutí části slova, a naopak při příliš nízké může dojít chybnému určení počátku či konce vlivem nechtěného rušení a šumu, jako například u zpracovávaného slova hlas. Na následujících grafech je možné porovnat signály tohoto slova, které je „problematické“ právě na konci vysloveným písmenem „s“, jehož amplituda je v poměru ke zbývajícímu průběhu signálu velmi malá. Obr. 5.2 představuje obálku chybně oříznutého slova, „ochuzeného“ o „s“. Na obr. 5.3 je potom již požadovaný výsledek, u nějž díky vhodné mezi proběhla úprava, tak jak je vyžadováno.



Obr. 5.1 Absolutní obálka řečového signálu slova „hlas“



Obr. 5.2 Chybný ořez slova „hlas“ provedený prostřednictvím obálky



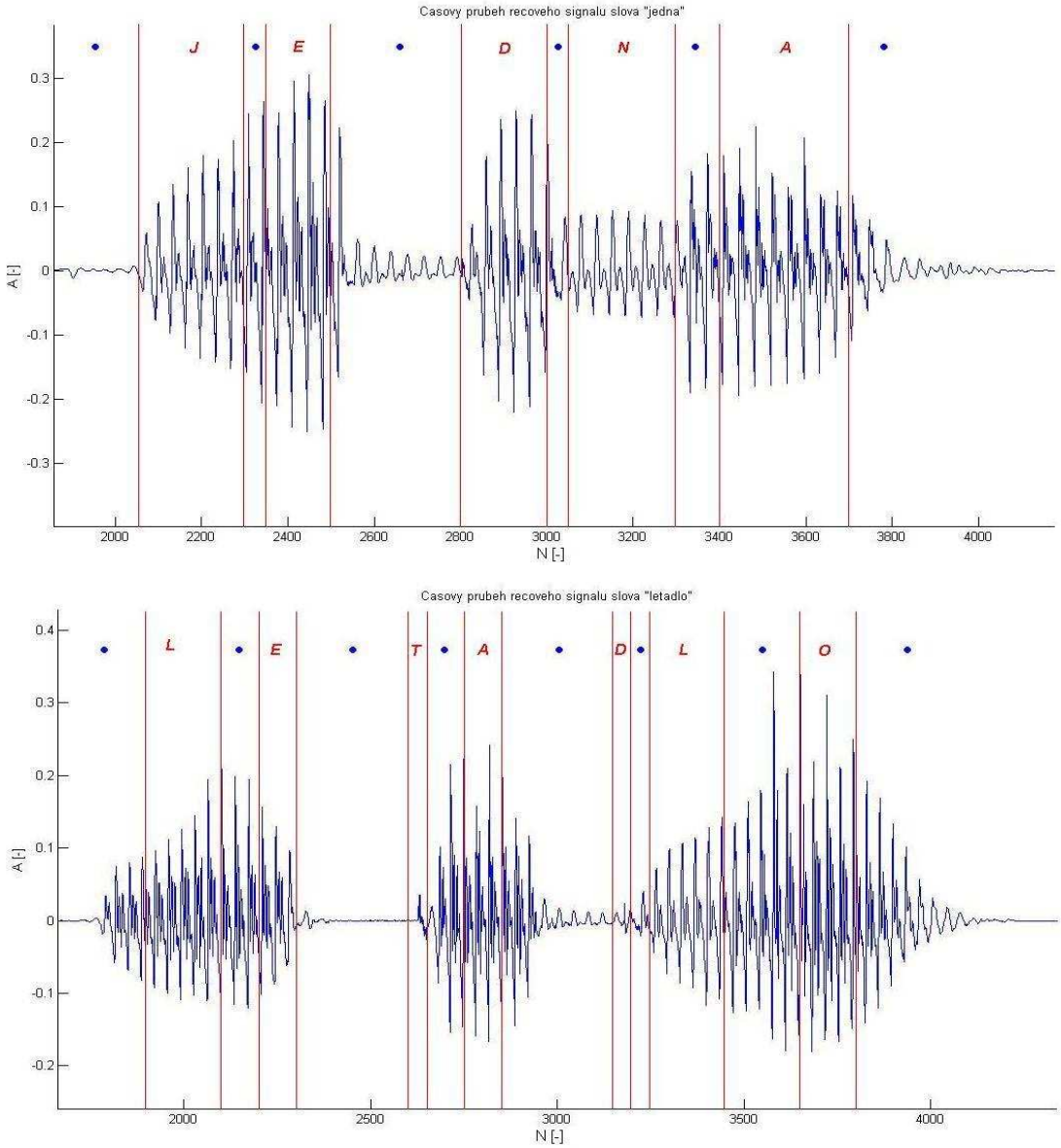
Obr. 5.3 Výsledný oříznutý signál slova „hlas“ připravený k další práci

Dalším krokem je rozdělení předzpracovaného slova na mikrosegmenty. Pro návrh rozpoznávacího systému byl zvolen jejich počet pro rozpoznávaná slova na

sedm. Byly provedeny i experimenty s počtem pět, ale pro vyšší spolehlivost a flexibilnější práci se nakonec vyšší počet jevil jako vhodnější. V praxi používané klasifikátory jsou sestrojovány právě pro počet čtyř až sedmi mikrosegmentů. Realizace tohoto požadavku probíhá rozdělením předzpracovaného vzorku na sedm stejně dlouhých úseků.

5.3 VYTVOŘENÍ SIGNÁLOVÝCH CHARAKTERISTIK

Pro zdokonalení představy o průběhu řečového signálu byly u dvou testovaných slov, „jedna“ a „letadlo“, nalezeny přibližné hranice mezi jednotlivě vyslovenými písmeny. Nalézání hranic bylo provedeno vložením náhodných úseků do smyčky a rozlišeno sluchem. Mezi samostatně znějícími „písmeny“ se nacházejí nezanedbatelné koartikulační přechody (na obr. 5.4 znázorněny modrými puntíky).

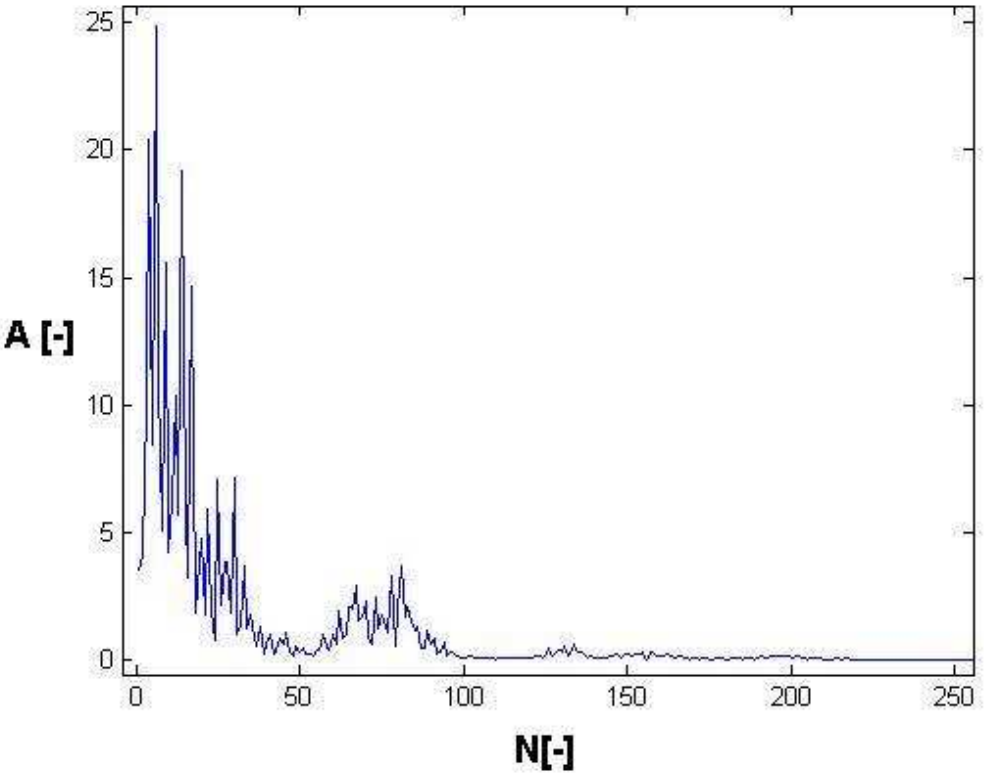


Obr. 5.4 Časový průběhu signálu slov s vyznačenými písmeny a koartikulačními úseky

Při pohledu na členění těchto dvou slov si lze povšimnout některých rozdílů u stejných písmen. V různých slovech jsou jejich průběhy jiné. Například u písmen „A“ a „E“, jsou na první pohled zřejmé rozdíly v délce jejich vyslovení a amplitudě. Z toho lze usoudit, že prvotní strojové dělení slova na jednotlivá písmena pro následné rozpoznávání by bylo velmi obtížné, nespolehlivé a výpočetně náročné. Výhodnější a rychlejší, i když méně přesné se jeví dělení na předem daný počet

vhodně dlouhých mikrosegmentů. Také výkony a amplitudy různých záznamů signálů stejných slov se liší, a proto je obtížné využít je pro srovnávání, obzvlášť při chybném určení a počátku slova. Normalizace délek signálu je v takovém případě bezúčelná.

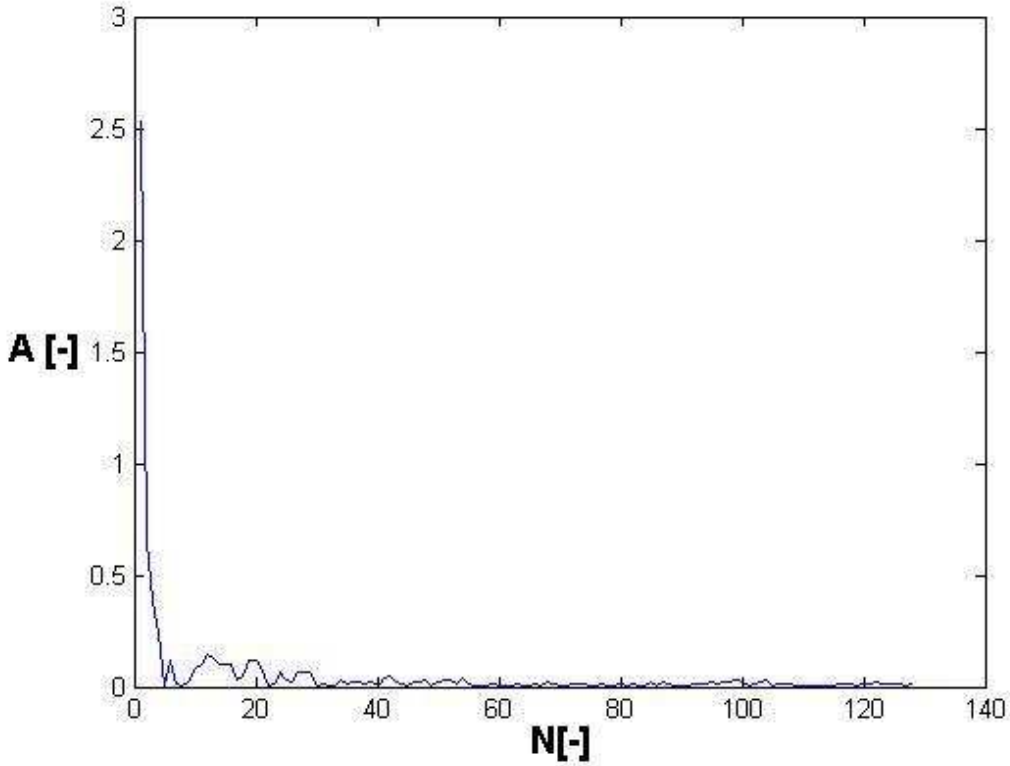
Unikátním prvkem, který může posloužit při rozpoznávání, jsou frekvenční charakteristiky obsažené v mikrosegmentech slov. K jejich nalezení slouží krátkodobá Fourierova transformace, pro jejíž aplikaci slouží v Matlabu příkaz fft. Výsledek aplikace Fourierovy transformace na mikrosegment odpovídající písmenu A ze slova „jedna“ je vidět na Obr 5.5.



Obr. 5.5 Fourierova transformace části řečového signálu písmene „A“

V praxi je využívána pro rozpoznávání řeči kepstrální analýza, jde o tzv. homomorfické, obecně nelineární zpracování signálů. Na Obr. 5.6 je vykresleno reálné kepstrum mikrosegmentu slova „hlas“, ve kterém je obsaženo písmeno „A“

Za povšimnutí stojí také fakt, že charakteristiky nesou užitečnou informaci pouze na svém počátku. Jak je na grafech 5.5 a 5.6 zřetelně vidět, jde o počáteční část, v níž jsou amplitudy mnohonásobně vyšší, než u zbytku signálu. Pro nižší výpočetní náročnost i jednoznačnější určení charakteristiky bude tato skutečnost brána v potaz během následující práce.



Obr. 5.6 Reálné kepstrum písmene „A“

5.4 VEKTOROVÁ KVANTIZACE A KÓDOVÁ KNIHA

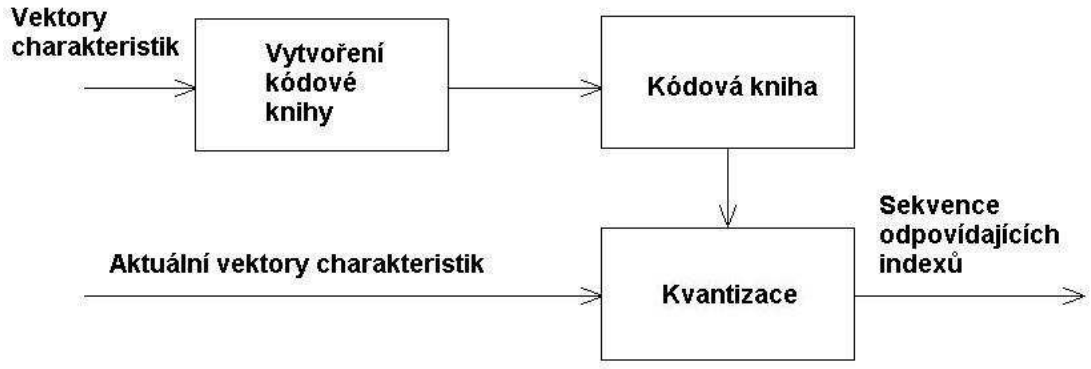
Základní částí, bez které není možno systém rozpoznávání aplikovat, je referenční soubor dat, pomocí něhož lze hodnotit míru shody pozorovaných prvků. V případě rozpoznávání slov je tímto souborem, jak již bylo zmíněno tzv. kódová kniha obsahující spektrální vzory řečových segmentů.

Vytvoření kódové knihy probíhá prostřednictvím tzv. vektorové kvantizace. Je to proces, při němž jsou vybrány charakteristiky, které jsou schopny jednoznačně

popsat požadovanou subslovní jednotku a při další práci zastoupit mikrosegment v rozpoznávaném slově.

S předzpracovanými vzorky slov je možné začít kódovou knihu tvořit. Ze všech slov, která mají být rozpoznávána jsou vyhledány jedinečné mikrosegmenty a vypočteno jejich reálné kepstrum. Jedinečnými mikrosegmenty se rozumí fonémy, vybrané tak, aby pokryly skladbu všech požadovaných slov. Ty fonémy, které jsou v různých slovech stejné jsou tedy zastoupeny pouze jedním vzorovým. V kódové knize, jíž je matice, jsou kepstra jednotlivých fonémů představována sloupcovými vektory, které jsou řazeny ve sloupcích za sebou. Index sloupce tedy odpovídá právě jednomu záznamu v kódové knize. Pro účely navrhovaného klasifikátoru byl počet vzorů v kódové knize stanoven pro všech devět slov na devatenáct.

S vytvořenou kódovou knihou lze popsat slovo sekvencí indexů, které odpovídají právě charakteristikám nejpodobnějších úseků, z nichž se slovo skládá. Samotné porovnávání mezi charkteristikami v kódové knize a charakteristikami aktuálně popisovaného slova probíhá stanovením absolutně nejmenšího rozdílu obsahů ploch pod křivkami charakteristik. Pro různé nahrávky stejného slova jsou pak vytvořeny sekvence tvořící vstup pro trénování Markovových modelů. Proces vektorové kvantizace je zobrazen v blokovém schématu na Obr. 5.7.

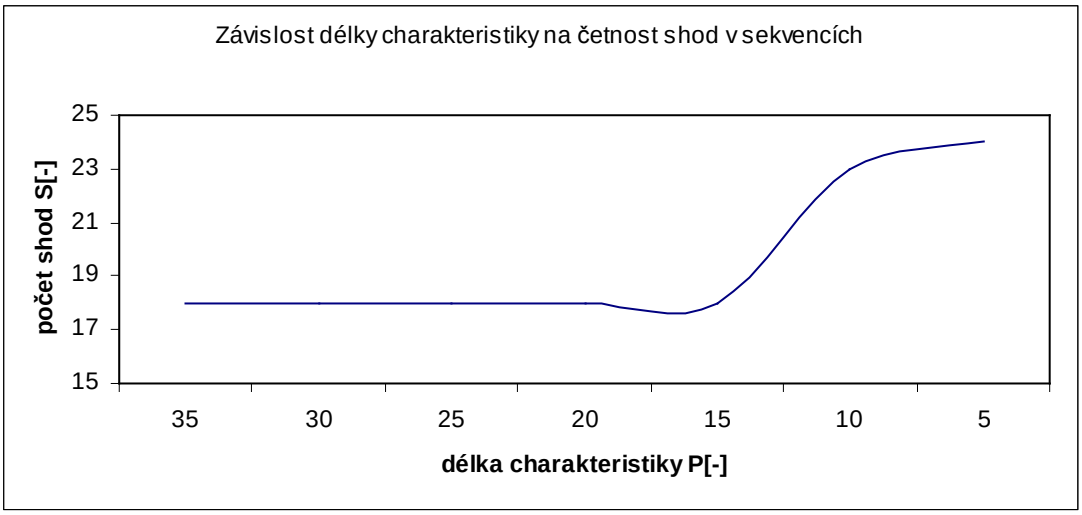


Obr. 5.7 Vektorová kvantizace [2]

Jednotlivé sekvence se mohou odlišovat, neboť ve dvou různých nahrávkách jednoho slova může například jeden vzor z kódové knihy odpovídat jednomu, dvěma

či více mikrosegmentům za sebou, v závislosti na tom, jak dlouze byl úsek odpovídajícího fonému vysloven nebo zda odpovídá v koartikulačním přechodu předchozímu nebo již následujícímu fonému.

Během procesu vytváření sekvencí výstupů bylo zároveň testováno, jaký vliv má na spolehlivost sekvencí forma porovnávaných charakteristik mezi kódovou knihou a mikrosegmenty aktuálně zpracovávaného slova. Charakteristiky jsou reálným kepstrem částí signálů. Prvním bodem testování bylo, jaký vliv má na spolehlivost porovnávání délka této charakteristiky od počátku. Jak již bylo zmíněno, porovnávání probíhá na základě rozdílu mezi obsahy ploch pod křivkami charakteristik. Se zkracující se délkou charakteristiky od počátku vykazují výsledné sekvence vyšší četnost shod pro různé nahrávky stejných slov (v tomto případě 10), což svědčí o zvýšení spolehlivosti stanovování sekvencí. Při přílišném zkrácení však může dojít ke zkreslení. Při příliš malém úseku obsahu totiž výsledky konvergují a splývají, čímž může dojít k takovému zjednodušení, které má neblahý vliv na přesnost indexování. Ze srovnávání sekvencí určených pomocí obsahů zkracujících se obsahů kepster lze vyčíst, že nejvyšší četnost lze nabýt mezi 5-15 počátečními hodnotami. Při nižším počtu dochází již ke zmíněnému zkreslení. Pro návrh je tedy zvolen počet 10 hodnot.Výsledky testované závislosti délky kepstra jsou znázorněny na Obr. 5.8.



Obr. 5.8 Závislost mezi délkou charakteristik a četností shod v sekvencích

Samotné shody mezi sekvencemi všech záznamů stejného slova jsou vyznačené na Tab. 5.1. V tomto případě jde o slovo „rozpoznat“, majoritní shody jsou vyznačeny sytě oranžově, ostatní pak světlejšími odstíny. Čísla v buňkách jsou indexy vzorů z kódové knihy. Bílá políčka jsou nepřesně určené indexy, vlivem koartikulačních přechodů a nepředvídatelných anomálií v promluvě (např. vadně vyslovený foném). Každý řádek odpovídá sekvenci pro jedno slovo.

3	15	19	6	1	10	19
5	15	13	19	10	3	15
6	15	19	5	5	3	15
3	15	10	15	15	15	19
1	15	5	19	19	5	19
10	15	19	13	5	3	5
10	15	19	3	6	10	19
19	15	5	6	15	19	19
19	15	10	3	10	10	19
3	15	19	10	6	3	19

Tab 5.1 Četnosti shod mezi sekvencemi stejného slova

Kromě zkrácení porovnávaných charakteristik je velkým přínosem pro spolehlivé „indexování“ také druhá mocnina všech hodnot uvažovaných charakteristik. Pro navrhovaný systém je při výpočtu obsahu plochy pod křivkou charakteristiky použit postup, kdy je nejprve z hodnot vypočtena druhá mocnina, poté jsou první tři trojice hodnot (tedy dohromady devět hodnot) aritmeticky zprůměrovány a následně sečteny. Při tomto způsobu výpočtu obsahů pro následný rozdíl, bylo dosaženo vyšší přesnosti při vytváření výsledných sekvencí stavů. Z několika testovaných postupů[{1}] se tento jevil jako nejefektivnější.

[{1}] V jiných postupech byly namísto aritmetických průměrů užity výpočty mediánu. Dále pak namísto prvních trojic hodnot signálu dvojice či čtveřice.

5.5 TRÉNOVÁNÍ MODELU SLOVA

S vytvořenou kódovou knihou pomocí níž jsme schopni určovat sekvence výstupů je možno započít trénování modelů slov. Pro tento účel nejlépe poslouží „Matlabovský“ příkaz **hmmtrain**, jehož vstupy jsou:

- sekvence stavů (indexů popisujících mikrosegmenty rozpoznávaných slov, dle jim nejbližších vzorů z kódové knihy)
- počáteční odhad matice přechodů mezi stavy
- počáteční odhad matice výstupů stavů

Počáteční odhad matice přechodů je volitelně stanoven s přihlédnutím k několika podmínkám. Rozměry matice musejí odpovídat rozměru N *x* N, kde N je počet prvků sekvence indexů. Dále je nutné uvažovat fakt, že jde o levopravé modely, z toho vyplývá, že všechny prvky matice přechodů pod hlavní diagonálou jsou nulové, neboť průchod modelem začíná okamžitým vstupem do prvního prvku sekvence a následně může pouze setrvávat na aktuální pozici, nebo se přesunout směrem vpravo. Prvky hlavní diagonály budou mít logicky nejvyšší hodnoty, které se v jim odpovídajících řádcích vyskytnou a na každé další pozici bude hodnota buď stejná nebo nižší. Součet prvků v řádcích vyjma posledního pak musí být roven jedné (podmínka 4.7).

Počáteční odhad matice výstupů musí mít rozměry N x L, kde L je počet sloupců (záznamů) v kódové knize. Počáteční odhad bude homogenní maticí, v níž součet všech prvků v každém řádku bude rovněž odpovídat jedné. Prvek bude tedy nabývat hodnoty 1/L.

Na Obr. 5.9 a.,b. jsou znázorněny matice přechodů, v případě a. jde o hrubý odhad, v případě b. o poněkud podrobnější. c. představuje matici výstupů.

a.

$$TRANS = \begin{bmatrix} 1 & 0 & .. & 0 \\ 0 & 1 & .. & 0 \\ : & : & . & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

c.

$$EMIS = \begin{bmatrix} 1/L & 1/L & .. & 1/L \\ 1/L & 1/L & .. & 1/L \\ : & : & . & 1/L \\ 1/L & 1/L & 1/L & 1/L \end{bmatrix}$$

b.

$$TRANS = \begin{bmatrix} 0{,}333 & 0{,}333 & 0{,}333 & 0 & .. & 0 \\ 0 & 0{,}333 & 0{,}333 & 0{,}333 & .. & 0 \\ 0 & 0 & 0{,}333 & 0{,}333 & .. & 0 \\ : & : & : & . & .. & 0{,}333 \\ : & : & : & : & . & 0{,}5 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Obr. 5.9 Matice přechodů TRANS a matice výstupů EMIS

Trénování probíhá v cyklu, kdy jsou všechna stejná slova popsaná sekvencí indexů o počtu N [{2}] (každé slovo je rozděleno na N mikrosegmentů) použita postupně jako vstupní sekvence pro funkci hmmtrain. Je-li tedy pořízeno P [{2}] záznamů stejných slov, proběhne P cyklů, kdy v každém cyklu je vstupem sekvence jiného záznamu stejného slova a zároveň je použita matice přechodů a výstupů vypočtená v předchozím cyklu s tím, že pro první krok je použit výše zmíněný počáteční odhad. Jinými slovy pro každý aktuální cyklus jsou pro funkci hmmtrain kromě nové sekvence vstupem matice přechodů a vstupů vypočtené v předchozím cyklu a v tom aktuálním jsou přepočteny právě pro novou sekvenci a stávají se vstupem funkce pro následující cyklus, není-li ten aktuální posledním. Schéma procesu je vyobrazeno na Obr. 5.10.

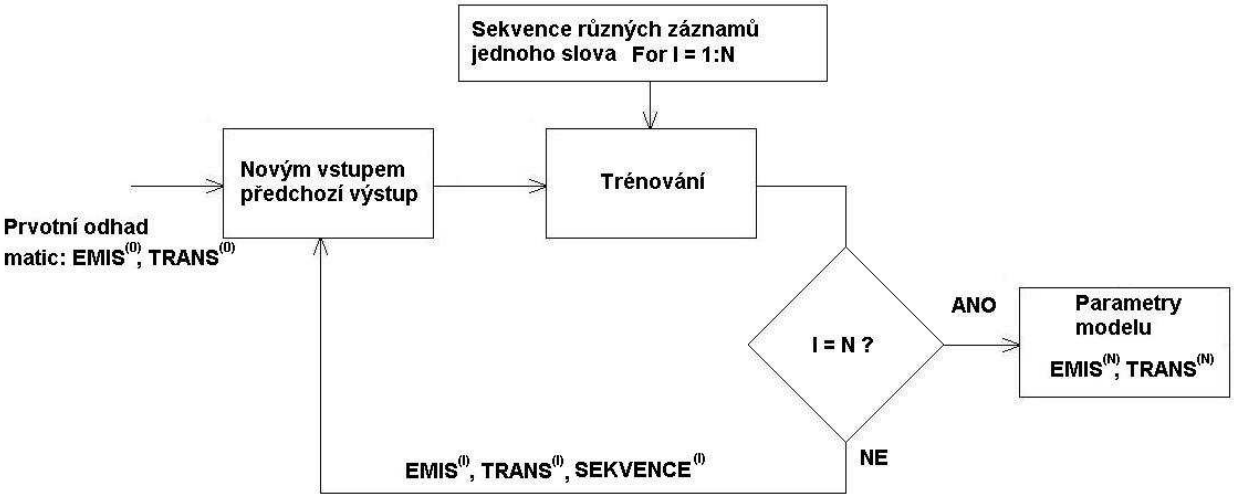
Aplikací výše popsaného procesu jsou tedy natrénovány modely různých slov, a tím získány jejich parametry. Model slova je složen z prvků matic N, M, π

[{2}] V navrhovaném systému N = 7, P = 10

(vztah 4.5). V představovaném systému jsou N a M natrénovány jako TRANS a EMIS. π je pro všechny modely stejná, π = [1,0,0,0,0,0,0]. Počet prvků vektoru π je N, tedy stejný jako počet mikrosegmentů slova. Podle podmínky 4.6 je součet prvků vektoru rovněž roven jedné. V případě modelů slov pro klasifikátor je pouze jedna možnost. Každý průchod modelem začíná vždy první pozicí v sekvenci, tudíž je pro první prvek vektoru π 100% pravděpodobnost startovací pozice průchodu modelem.

Skript s algoritmem pro trénování je součástí přílohy.

Nyní jsou stanoveny veškeré potřebné parametry modelu a je možno přejít k samotnému rozpoznávaní.



Obr. 5.10 Proces trénování modelu slova

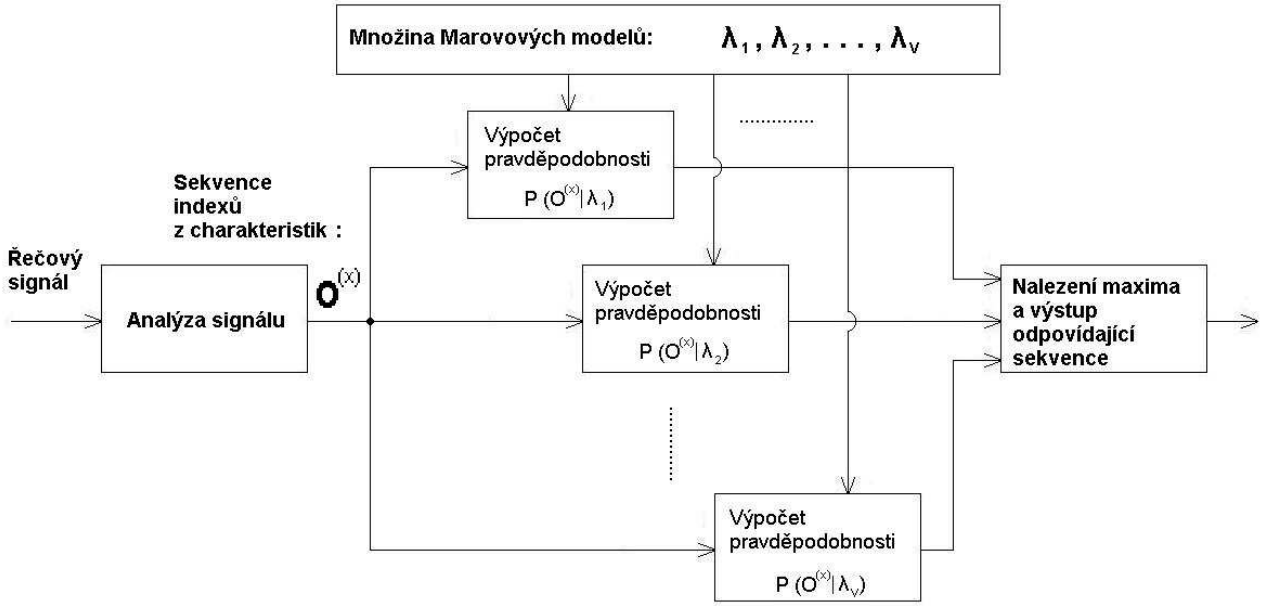
5.6 VYHODNOCENÍ PRAVDĚPODOBNOSTI SHODY PROMLUVY

S připravenými modely slov je již vyhodnocení pravděpodobnosti promluvy sekvence O rozpoznávaného slova P(O| π) snadným úkolem. Pro potřeby zjištění nejvyšší pravděpodobnosti, zda je pozorovaná sekvence stavů generována určitým modelem, je nejvhodnější použít funkci systému Matlab hmmviterbi (viz 4.3.2.2).

Jako vstupní proměnné do funkce hmmviterbi je sekvence indexů vyhodnocovaného slova a matice přechodů a výstupů modelu slova. Výstup funkce hmmviterbi je maximálně pravděpodobná posloupnost stavů, kterou je vstupní model schopen vygenerovat. Pravděpodobnost shody promluvy je tedy vypočtena tak, že je stanoven počet shodných prvků mezi vypočtenou posloupností a původní sekvencí stavů, a tento počet je následně dělen celkovým počtem prvků posloupnosti. Při provedení tohoto postupu pro všechny modely, je pak nalezena nejvyšší vypočtená pravděpodobnost, a rozpoznávané slovo je stanoveno jako nejpodobnější tomu, jehož model vygeneroval onu posloupnost s největší pravděpodobností shody s původní sekvencí stavů.

Aktuální rozpoznávaná sekvence je zadávána jako parametr jednotlivě pro všechny natrénované modely a zároveň s každým modelem vypočtena pravděpodobnost. Z pravděpodobností všech modelů je při vyhodnocení nalezeno maximum, a modelu slova, jemuž maximum náleží je přiřazeno rozpoznávané slovo jako nejbližší.

Celkové schéma principu klasifikátoru slov je znázorněno na Obr. 5.11 [3]



Obr 5.11 Princip systému pro rozpoznávání izolovaných slov

ZÁVĚR

Účelem této práce bylo představit biometrické systémy a dále se podrobněji zabývat problematikou rozpoznávání řeči. S první zmíněnou částí, biometrickými systémy se mohl čtenář seznámit ve druhé kapitole, kde je popsáno jejich členění, využití a dále metody a principy, na jakých fungují. Druhá řečená oblast, rozpoznávání řeči, je pak hlavní téma, jímž se tato práce zabývala.

Znalost mechanismu vzniku řeči a jejího fungování z fyzikálního hlediska je pro pochopení principu klasifikátorů nezbytná. Základním poznatkům z oblasti řeči byla proto věnována druhá kapitola. Při návrhu klasifikátoru je pozornost zaměřena na získání číselně vyjádřitelných charakteristik, naměřených ze vzorků slov či jejich částí, tzv. mikrosegmentů. U takovýchto charakteristik jde o to, aby nesly co nejjednoznačnější informaci o slově či mikrosegmentu využitelnou při rozpoznávání. Z těchto informací lze sestavit tzv. vektory příznaků. Pomocí systému Matlab lze s těmito vektory, tzv. sekvencemi natrénovat modely, jejichž prostřednictvím lze úspěšně navrhnout systém pro rozpoznávání izolovaných slov. Tato metoda tzv. skrytých Markovových modelů byla podrobně popsána ve čtvrté kapitole i se zmíněnými funkcemi systému Matlab, které byly při rozpoznávání využity. Samotná realizace a návrh klasifikátoru, včetně podrobně popsaných souvislostí s metodou skrytých Markovových modelů byla rozebrána v poslední, tedy páté kapitole.

Zmíněná pátá kapitola práce prezentuje průběh návrhu vlastního klasifikátoru a výsledky, které při něm byly dosaženy. Pro účely návrhu systému byly pořízeny 9 různých pracovních řečových signálů. Po opracování řečových signálů byla ověřena využitelnost některých příznaků pro rozpoznávání, jako je obálka signálu, amplituda, signálová charakteristika mikrosegmentů pomocí Fourierovy transformace a kepstra. V konečném řešení navrhovaného klasifikátoru byla využita právě ona poslední kepstrální analýza, díky vysoké spolehlivosti příznaků, dostatečně charakterizujícím řečové úseky. Funkčnost takto navrženého systému byla úspěšně ověřena pro jednoho řečníka s přesností 40%. Pokud by byla požadována úspěšnost vyšší, bylo by možné rozšířit rozpoznávací systém o další charakteristické příznaky v kódové knize, které by mohly přesněji specifikovat slova při popisu sekvencí, ovšem za předpokladu zvýšení výpočetní náročnosti.

LITERATURA

[1] Drahanský M: Přehled biometrických systémů a testování jejich spolehlivosti, Praha, CZ, 2007

[2] Psutka J: Komunikace s počítačem mluvenou řečí, Academia, 1995

[3] Rabiner LR: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of the IEEE, 77:2, p. 257-286, February 1979

PŘÍLOHY

Skript s trénovacím algoritmem **trenovani.m** :

```
% v tomto skriptu probíhá načtení a zpracování vzorků jednoho slova
% a natrénování modelu pro toto slovo
clear all

%% Načtení kodove knihy
kodova_kniha_struct = load('kodova_kniha_pismena.mat');
kodova_kniha = kodova_kniha_struct.kodova_kniha_pismena;

sloupce = size(kodova_kniha);
nr = 7; %% počet mikrosegmentů defaultně 7

%% Vstupní matice přechodů a výstupů
TRANS =     [0.333  0.333   0.333   0       0       0       0    ;
             0      0.333   0.333   0.333   0       0       0    ;
             0      0       0.333   0.333   0.333   0       0    ;
             0      0       0       0.333   0.333   0.333   0    ;
             0      0       0       0       0.333   0.333   0.333;
             0      0       0       0       0       0.5     0.5  ;
             0      0       0       0       0       0       1    ;
            ];

%% matice výstupů pro kodovou knihu useku
for sem = 1:sloupce(2)
    for tam = 1:nr
        EMIS(tam, sem) = 1/sloupce(2);
    end
end

%% Načtení jednotlivých vzorků a transpozice
a0 = wavread('slova\01_jedna\VZ01.wav')';
.
.
.
a9 = wavread('slova\01_jedna\VZ01i.wav')';
%% Vytvoření sjednocené matice a následné vkládání různě dlouhých
%%matic načtených vzorků
VZOREK(1:2, 1: length(a0)) = a0;
.
.
.
VZOREK(19:20, 1: length(a9)) = a9;

%% ZPRACOVÁNÍ VŠECH VZORKŮ JEDNOHO SLOVA %%

for i = 0:9 %% zde defaultně 0:9
            %% aplikace uprav a nulování konstant pro dalsi praci
          clear VZ;
          clear nad_mez;
          clear cut_sig;
          clear micr_ceps;
```

```
%% vybírání stereo dvojic párů
VZ = VZOREK((2*i+1):(2*i+2),:);
VZ_M(:,:) = zeros;
%% vytvoření mono signálu
VZ_M = ( VZ(1,:) + VZ(2,:) ) / 2;

%% NOVÝ SIGNÁL - ořezaný monosignál %%

%%nalezení oriznuti
%% mez, vyjadrena procenty z maxima
mez = 0.15 * max(VZ_M);
%% nalezení hodnot nad mezí pro určení začátku a konce
%% slova
nad_mez = find(VZ_M > mez);

%% konečný oříznutý mono signál
    %% délka ořezaného monosignálu
cut_sig_length = (max(nad_mez) - nad_mez(1) + 1);
    %% ořezaný monosignál
cut_sig(1:cut_sig_length) =
VZ_M(nad_mez(1):max(nad_mez));

%% Mikrosegmenty
%%%% délka mikrosegmentu
micr_length = floor(cut_sig_length/nr);

%% matice mikrosegmentů - souslednost po sloupcích
for k = 1:nr
      micr(1:micr_length, k) = (cut_sig(((k - 1) *
      micr_length + 1):(k * micr_length)));
end

%% Charakteristické parametry %%

%% REAL CEPSTRUM %%
%% výpočet reálného kepstra pro všechny mikrosegmenty
for j = 1:nr
      micr_ceps(:, j) = rceps(cut_sig(((j - 1) *
      micr_length + 1):(j * micr_length)));
end

%% odchyleky od mikrosegmentů a pozice v kódové knize %%
%% vyhodí rozdíly charakteristik každého mikrosegmentu
se vzory v kódové knize
    for p = 1:nr % počet mikrosegmentů (7)
        for o = 1:(sloupce(2))%sloupceů v kódové knize
            odchylky((p - 1) * sloupce(2) + o) =
            abs(
            sum((mean((kodova_kniha(1:3, o))) +
            mean((kodova_kniha(4:6, o))) +
            mean((kodova_kniha(7:9, o)))).^2)
            -
            sum( (mean((micr_ceps(1:3, p))) +
            mean((micr_ceps(4:6, p))) +
            mean((micr_ceps(7:9, p)))).^2)
            );
```

```
                    end
                        [min_kod(p), sekvence(p)] =
        min(odchylky(((p - 1) * sloupce(2) + 1):(p * sloupce(2))));
                        %% nalezení nejmenšího rozdílu a jeho indexu
                end

            %% Trénování modelu
            [TRANS, EMIS] = hmmtrain(sekvence, TRANS, EMIS);
end

%% Uložení natrénovaného modelu do souboru
savefile = 'model.mat';
model = [TRANS EMIS];
save(savefile, 'model')
```