VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

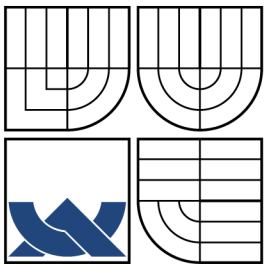
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

LINEÁRNÍ PREDIKČNÍ A KEPSTRÁLNÍ SYNTÉZA ŘEČOVÉHO SIGNÁLU V SYSTÉMU TTS

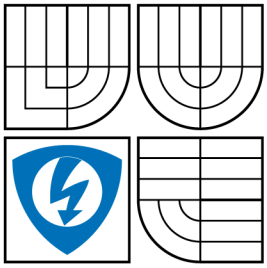
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE JIŘÍ MEKYSKA
AUTHOR

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

LINEÁRNÍ PREDIKČNÍ A KEPSTRÁLNÍ SYNTÉZA ŘEČOVÉHO SIGNÁLU V SYSTÉMU TTS

LINEAR PREDICTION AND CEPSTRAL SYNTHESIS OF SPEECH SIGNAL IN THE TTS SYSTEM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — JIŘÍ MEKYSKA
AUTHOR

VEDOUCÍ PRÁCE — PROF. ING. ZDENĚK SMÉKAL, CSC.
SUPERVISOR

BRNO 2008

ZDE VLOŽIT LIST ZADÁNÍ

Z důvodu správného číslování stránek

ZDE VLOŽIT PRVNÍ LIST LICENČNÍ SMOUVY

Z důvodu správného číslování stránek

ZDE VLOŽIT DRUHÝ LIST LICENČNÍ SMOUVY

Z důvodu správného číslování stránek

ABSTRAKT

Práce se zabývá lineární predikční a kepstrální syntézou řečového signálu v systémech TTS (Text-to-Speech) s možností modelování prozodie. Je zde uveden popis řečového signálu v akustické a fonetické rovině, princip tvorby řeči a způsob znázornění řečového signálu v časové a kmitočtové oblasti. Dále je zde uvedena bloková stavba TTS systémů, přičemž každý blok je zvlášť detailně popsán. V práci je také popsána problematika modelování prozodie pomocí tří nejdůležitějších suprasegmentálních rysů (základní tón, trvání a intenzita řeči). Na konci je proveden návrh a realizace univerzálního českého TTS systému, který je založen na syntéze řeči v kmitočtové oblasti. Tento systém je implementován v programu MATLAB.

KLÍČOVÁ SLOVA

Řeč, kmitočet, formant, hlasový trakt, lineární predikce, kepstrum, syntéza, analýza, TTS systém, filtr, prozodie, emoce, fonetická transkripce, MATLAB.

ABSTRACT

This work deals with a linear prediction and cepstral synthesis of speech signal in the TTS (Text-to-Speech) systems with the opportunity of modeling the prosody. The work contains a description of speech signal in acoustic and phonetic plane, the principle of speech production and the way we can figure the speech signal in time and frequency domain. Next, there is the TTS block structure mentioned, whereas each block has its own detailed description. In the work, the modeling of prosody using the three most important suprasegmental features (fundamental tone, continuation and speech intensity) is also described. At the end of this work, there is a design and realization of universal Czech TTS system which is based on the speech synthesis in frequency domain. This system is implemented in program MATLAB.

KEYWORDS

Speech, frequency, formant, vocal tract, linear prediction, cepstrum, synthesis, analysis, TTS system, filter, prosody, emotion, phonetic transcription, MATLAB.

MEKYSKA, J. *Lineární predikční a kepstrální syntéza řečového signálu v systému TTS*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav telekomunikací, 2008. 81 s. Bakalářská práce. Vedoucí práce byl prof. Ing. Zdeněk Smékal, CSc.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Lineární predikční a kepstrální syntéza řečového signálu v systému TTS“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne ....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce prof. Ing. Zdeňku Smékalovi, CSc. za velmi užitečnou pomoc, vstřícný přístup a cenné rady při zpracování bakalářské práce. Děkuji Ing. Martinu Vondrovi, Ph.D., Dr. Ing. Robertu Víchovi, DrSc. a Ing. Ivanu Koulovi za cenné rady z oblasti zpracování řečových signálů. Také děkuji Ing. Petrovi Syslovi za poskytnutí nahrávek řeči.

V Brně dne .....

.....

(podpis autora)

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

1 ÚVOD

Jedním z dorozumívacích prostředků je pro celé lidstvo již několik tisíciletí řeč, přičemž samotná řeč má obecně dvě podoby: mluvenou a psanou. Díky řeči je člověk schopen vyjádřit různé myšlenky, nápady, pocity, emoce a obsahy sdělení. Mluvená řeč se pro lidstvo stala mocným nástrojem k dorozumívání, proto také není divu, že se člověk snažil vytvořit stroj, který by tuto řeč produkoval. První úspěchy je možné datovat do 18. století, kdy dánský vědec Christian Kratzenstein vytvořil aparaturu složenou z pěti píšťal, která dokázala produkovat pět dlouhých samohlásek [7]. Ve 20. století byly mechanické syntetizátory nahrazeny elektrickými. Jeden z prvních a nejznámějších elektronických syntetizátorů pocházel z Bellovi laboratoře a nesl jméno VOCODER [32]. Během druhé poloviny 20. století se hlasové syntetizátory stále zdokonalovaly, přičemž byl tento vývoj v 80. a 90. letech umocněn příchodem osobních počítačů.

Hlasové syntezátory zaznamenaly od 18. století značný pokrok a je nutno říci, že dnes již pracují s dostatečně vyspělými technologiemi a metodami, takže výsledná syntetická řeč je přirozená, srozumitelná a velmi podobná klasické lidské řeči. Příkladem takto dokonalých syntezátorů mohou být dva komerční programy RealSpeak™ od firmy Nuance a Voice Reader od firmy Linguatec [27], [29]. Díky uspokojivé přirozenosti a srozumitelnosti výsledného řečového signálu se systémy produkující lidskou řeč začaly používat v různých oblastech, jako jsou například telekomunikační či informační systémy. Jeden ze systémů, kde se syntéza lidské řeči využívá, je systém TTS (Text-to-Speech). Jednoduše můžeme říci, že je to systém, který převádí psanou řeč na mluvenou. Tyto systémy mají dnes velké uplatnění, používají se k automatickému čtení internetových stránek a emailů, nahrazují hlasatele na nádražích nebo pomáhají hlasově i zrakově postiženým lidem.

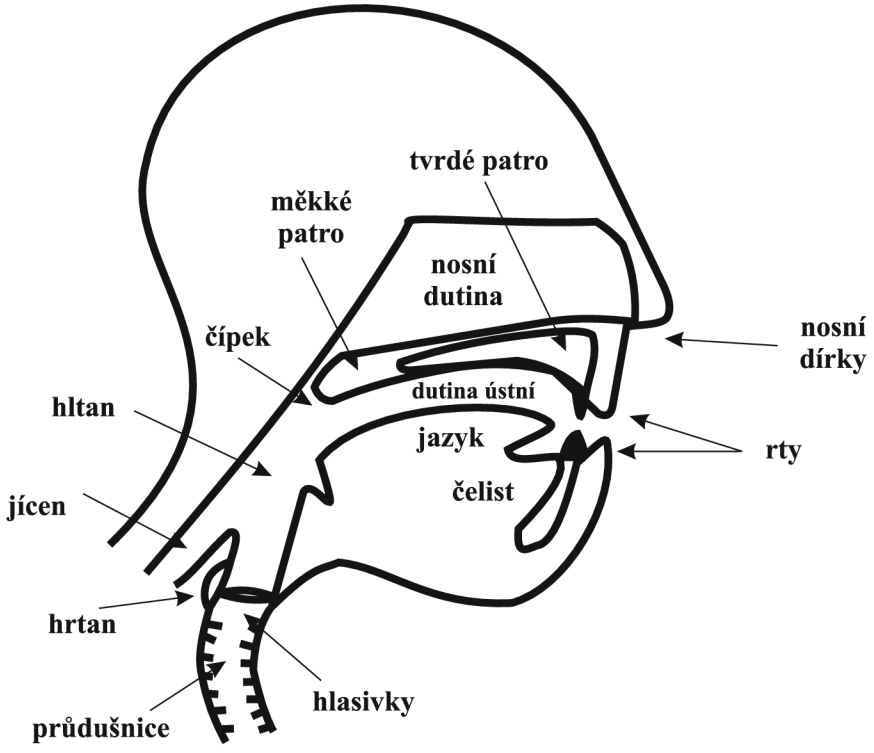
Cílem této bakalářské práce s názvem „Lineární predikční a kepstrální syntéza řečového signálu v systému TTS“ je teoretický rozbor zpracování a syntézy řečového signálu, rozbor TTS systémů, návrh univerzálního českého TTS systému a následná realizace v grafickém prostředí programu MATLAB.

2 ŘEČ

Řeč, jejíž centrum je v levé hemisféře lidského mozku, je produktem vyšší nervové činnosti. Tento nejstarší a nejpřirozenější prostředek komunikace mezi lidmi nese informace, které můžeme analyzovat hned několika způsoby. Existuje několik navzájem překrývajících se úrovní, které popisují řeč. Je to úroveň: akustická, fonetická, lexikální, syntaktická, sémantická a pragmatická. Pro potřebu syntézy řeči nás bude nejvíce zajímat úroveň akustická a fonetická.

2.1 Akustická rovina

Když člověk mluví, jeho plíce se chovají jako zdroj budicího signálu, který prochází hlasovým traktem, kde je různě modifikován a poté vychází ústy a nosem ven. Hlasový trakt funguje podobně jako hudební nástroj, který produkuje různé zvuky. Modifikací tvaru hlasového traktu modifikujeme také výsledný zvuk. Schéma hlasového traktu můžeme vidět na obr. 2.1.

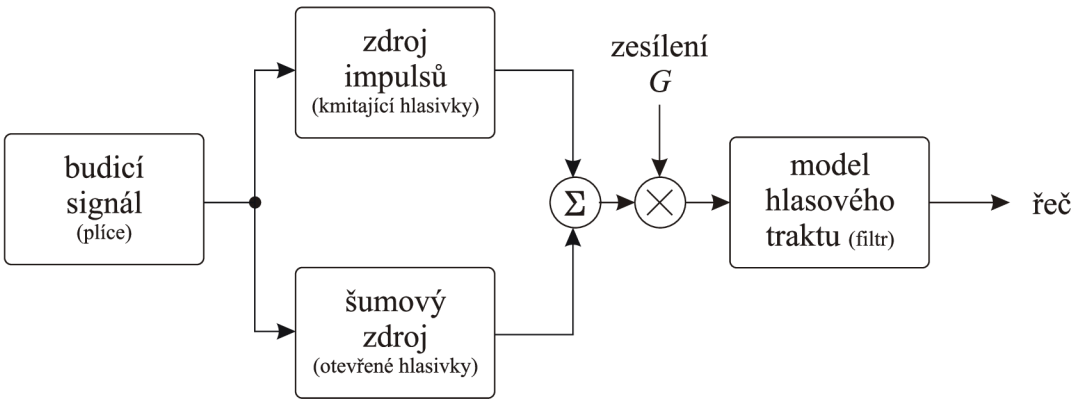


Obr. 2.1: Hlasový trakt

Hlasové ústrojí můžeme chápat jako rozvětvený zvukovod složený ze dvou větví: nazální a orální. Větev nazální má prakticky stálý tvar, mění se pouze v oblasti

vstupu, kde ji můžeme buď uzavřít, částečně otevřít nebo úplně otevřít. Orální větev může svůj tvar v určitých mezích měnit. Obě větve pak vyúsťují do prostoru v těsné blízkosti, a jejich výstupní signály se skládají do jediného signálu. Původní spektra signálů ze zvukových zdrojů se tak filtrují hlasovým traktem.

Akustickou soustavu vokálního traktu můžeme budit dvěma způsoby, a to periodickým signálem s proměnnou periodou, nebo šumem. Při periodickém buzení je budicí signál (proudění vzduchu) periodicky přerušován hlasivkami a vzniká tak signál, jehož kmitočet se označuje F_0 a nazýváme ho kmitočet základního tónu. Tento kmitočet se u různých lidí liší. U dětí může dosahovat hodnoty až 450 Hz, u dospělé ženy se pohybuje v rozmezí od 200 Hz do 300 Hz a u dospělého muže klesá až na 80 Hz. Při průchodu periodického signálu, o určitém kmitočtu, hlasovým traktem se může stát, že tento signál v některých místech hlasového traktu rezonuje (je zesílen) a naopak některými místy prochází téměř beze změny. Ty složky o kmitočtech, na kterých je signál zesílen, se nazývají formanty a ty, na kterých nedochází ke změnám, se nazývají antiformanty. Je-li zvukový zdroj šumový, dochází k turbulentnímu proudění vzduchu úžinami, které mění svou polohu na základě artikulované hlásky. Zjednodušený technický model hlasového ústrojí člověka je zobrazen na obr. 2.2.



Obr. 2.2: Model hlasového ústrojí člověka

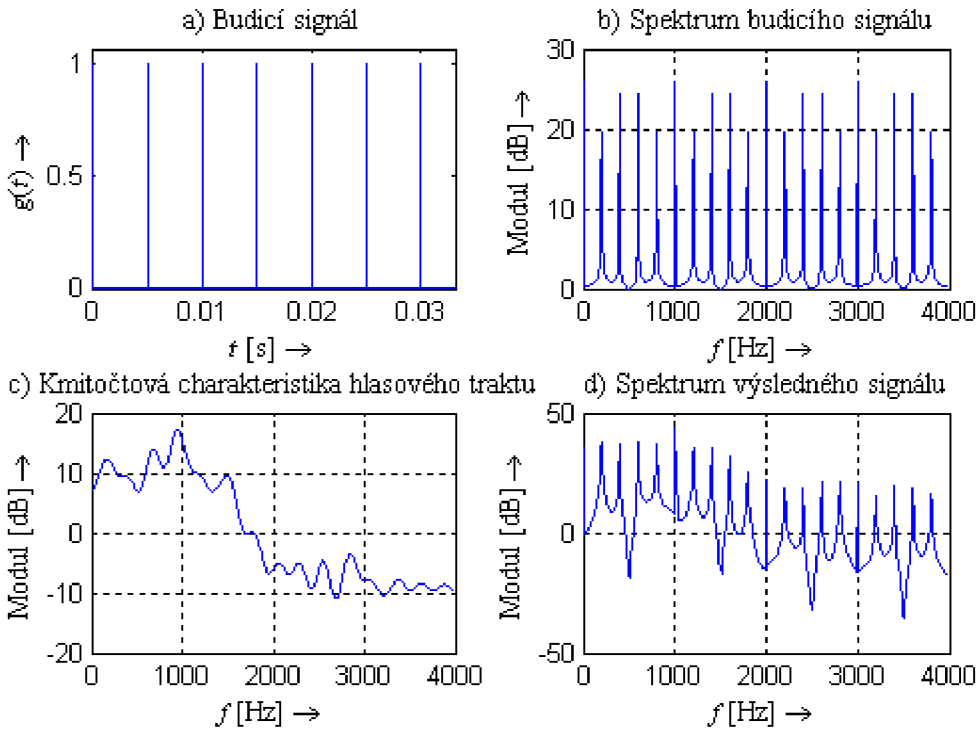
V obou případech je výsledný řečový signál dán konvolucí budicího signálu a impulsní charakteristiky hlasového filtru (traktu): [18]

$$s(t) = g(t) * h(t) = \int_{-\infty}^{\infty} g(\tau) \cdot h(t-\tau)\,\mathrm{d}\tau, \tag{2.1}$$

kde $s(t)$ je výsledný řečový signál, $g(t)$ budicí signál a $h(t)$ impulsní charakteristika. V kmitočtové oblasti můžeme tuto konvoluci zapsat jako součin spektra budicího signálu a kmitočtové charakteristiky filtru [18].

$$S(\omega) = G(\omega) \cdot H(\omega). \tag{2.2}$$

Obecně se tento postup označuje jako teorie zdroje a filtru. Princip tvorby znělé hlásky je zobrazen na obr. 2.3.



Obr. 2.3: Princip tvorby znělé hlásky ($f_{\mathrm{vz}} = 8\,\mathrm{kHz}$)

Skutečná řeč probíhá jako kontinuální proces a mezi jeho jednotlivými složkami (hláskami) dochází ke koartikulaci, to znamená, že při společném vyslovení sousedících zvuků, splývají jejich okrajové fáze, mění se a různě se vyslovují. Akustická skladba jednotlivých hlásek je dále, v rámci delších úseků (slovo, věta), modifikována modulačními faktory, které vytvářejí prozodické vlastnosti řeči. Jde o modifikace časové, intenzitní a melodické.

2.2 Fonetická rovina

Základní jednotkou řeči je foném (hláska). Skládáním fonému pak vznikají jednotlivá slova. Obecně rozeznáváme dvě základní skupiny fonémů:

- vokály (samohlásky) – ustálená poloha hlasového traktu
- konsonanty (souhlásky) – přechodové stavy hlasového ústrojí

2.2.1 Vokály (samohlásky)

Všechny vokály jsou znělé a vznikají v ustálené poloze hlasového traktu. Jsou buzeny periodickým signálem, který má relativně vysokou energii. Délka vokálů je důležitým rysem, který záleží na dialektu, individuálním stylu, emocích atd. Krátké vokály mají přibližně poloviční dobu trvání než vokály dlouhé. Přibližné doby trvání českých vokálů v milisekundách jsou uvedeny v tab. 2.1 [15].

Tab. 2.1: Délka trvání českých samohlásek

samohláska	typ. délka [ms]	rozmezí [ms]	samohláska	typ. délka [ms]	rozmezí [ms]
a	120	90–160	á	240	190–300
e	90	60–120	é	190	160–220
i	80	50–100	í	170	140–200
o	100	70–130	ó	200	160–250
u	90	60–120	ú	180	120–240

V průběhu kmitočtového spektra vokálů jsou zřetelně pozorovatelné jednotlivé formanty, jako např. na obr. 2.3 v grafu c). Jestliže kmitočet základního tónu F_0 udává výšku tónu, potom formanty udávají výsledný sluchový dojem. V analýze nebo syntéze řeči se většinou pracuje se třemi nejvýznamnějšími formanty. Každý formant udává kmitočet, na kterém procházející signál rezonuje v určité dutině hlasového traktu. Platí následující přiřazení formantů a dutin:

F_1 — dutina hrdelní
F_2 — dutina ústní
F_3 — dutina nosní

Kmitočty, na kterých se formanty vyskytují, jsou pro jednotlivé vokály uvedeny v tab. 2.2 [12].

Tab. 2.2: Kmitočtová pásma prvních třech formantů pro české vokály

hláska	F_1 [Hz]	F_2 [Hz]	F_3 [Hz]
i í	300—500	2000—2800	2300—3500
e é	480—700	1560—2100	2000—3000
a á	700—1100	1100—1500	1500—3000
o ó	500—700	850—1200	1500—3000
u ú	300—500	600—1000	1900—2900

2.2.2 Konsonanty (souhlásky)

Oproti vokálům jsou konsonanty buzeny šumem, což zapříčiňuje především podstatně těžší identifikaci. Vznikají postavením překážky do průchodu turbulentního vzduchu. Při buzení konsonant se hlasový trakt v určitých mezích mění. Konsonanty můžeme dělit dle několika způsobů. Níže jsou uvedeny příklady některých dělení.

Podle hlasnosti

- znělé — hlasivky vibrují
- neznělé — hlasivky jsou v klidu, hlasový trakt je buzen pouze vzduchem

Podle charakteru překážky

- překážka úplná — závěrové — okluzívy
- překážka neúplná (zúžení cesty výdechového proudu) — úžinové — frikativy
- polozávěrové — semiokluzívy

Podle párovosti

- párové — podobné postavením hlasového traktu, liší se znělostí
- nepárové — vždy znělé, nemají neznělý protějšek

Tab. 2.3 stručně zobrazuje dělení konsonantů podle znělosti, charakteru překážky a párovosti [12].

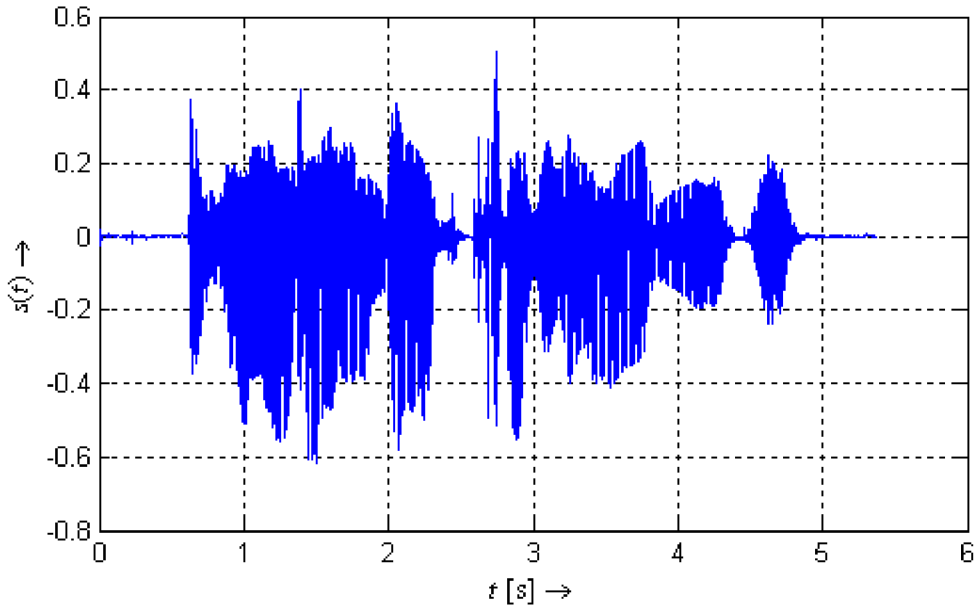
Tab. 2.3: Dělení souhlásek

souhlásky		závěrové (okluzívy)	úžinové (frikativy)	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	

2.3 Znázornění řečových signálů

2.3.1 Časový průběh

Abychom mohli řečový signál lépe analyzovat, je výhodné tento signál určitým způsobem vizualizovat. Asi nejzákladnějším znázorněním řečového signálu je zobrazení v časové rovině, kde na horizontální osu vynášíme dobu trvání signálu a na vertikální úroveň signálu. Příklad znázornění řečového signálu v časové rovině je na obr. 2.4.



Obr. 2.4: Časový průběh řečového signálu ($f_{\mathrm{vz}} = 8\,\mathrm{kHz}$, věta: „Oheň řádil v křoví nejvíc.“)

2.3.2 Kmitočtové spektrum

Pro nějakou hlubší analýzu je znázornění v časové rovině nevhodné. Nemůžeme z grafu například vyčíst, jaké kmitočty jsou v řečovém signálu zastoupeny, které fonémy jsou znělé, a které jsou neznělé atd. Proto se mnohem častěji řečový signál zobrazuje v rovině kmitočtové, neboli v rovině, kde na horizontální osu vynášíme kmitočet a na vertikální úroveň, popř. modul. Tomuto zobrazení se říká kmitočtové spektrum, příklad tohoto spektra je na obr. 2.5.

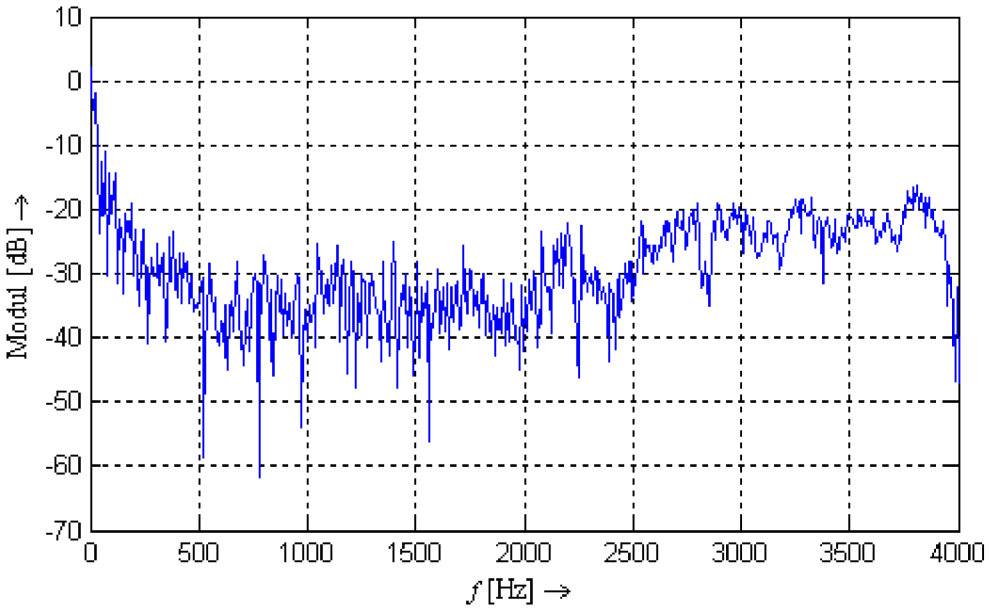
Kmitočtové spektrum řečového signálu nám znázorňuje zastoupení jednotlivých kmitočtů v daném signálu. Pro neperiodický signál je spektrum spojité, pro periodický signál naopak diskrétní. Obecně se pro znázornění signálu v kmitočtové oblasti používá Fourierova transformace: [18]

$$X(\omega) = \int_{-\infty}^{\infty} x(t) \cdot \mathrm{e}^{-\mathrm{j}\omega t}\,\mathrm{d}t, \tag{2.3}$$

kde funkci $X(\omega)$ označujeme jako spektrální funkci a ω $[\mathrm{rad} \cdot \mathrm{s}^{-1}]$ jako úhlový kmitočet. Funkce je definována $\forall\omega, -\infty < \omega < \infty$, má modul $|X(\omega)|$ a argument $\varphi(\omega) = \arg(X(\omega))$. Mluvíme tak o modulovém a argumentovém spektru. Pro reálný signál dále platí: [18]

$$X(\omega) = X^*(-\omega), \tag{2.4}$$

$$|X(\omega)| = |X(-\omega)|\,, \tag{2.5}$$



Obr. 2.5: Kmitočtové spektrum řečového signálu ($f_{\mathrm{vz}} = 8\,\mathrm{kHz}$, věta: „Oheň řádil v křoví nejvíc.“)

$$-\arg(X(\omega)) = \arg(X(-\omega)). \tag{2.6}$$

Ovšem při analýze většinou nepracujeme se spojitým řečovým signálem, ale se signálem, který je již navzorkován a omezen určitým oknem (jeho délka se stává periodou). V tom případě k výpočtu spektra nemůžeme použít klasickou Fourierovu transformaci, ale diskrétní Fourierovu transformaci (DFT). Ta je dána vztahem: [18]

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot \mathrm{e}^{-\mathrm{j}2\pi\frac{nk}{N}}, \tag{2.7}$$

kde N je délka diskrétního signálu $x[n]$ a $k = 0, 1, 2, ..., N-1$. Vynásobíme-li hodnoty $X[k]$ vzorkovací periodou T_{vz}, dostaneme aproximaci spektrální funkce v kmitočtových bodech $k\Delta f$, kde:

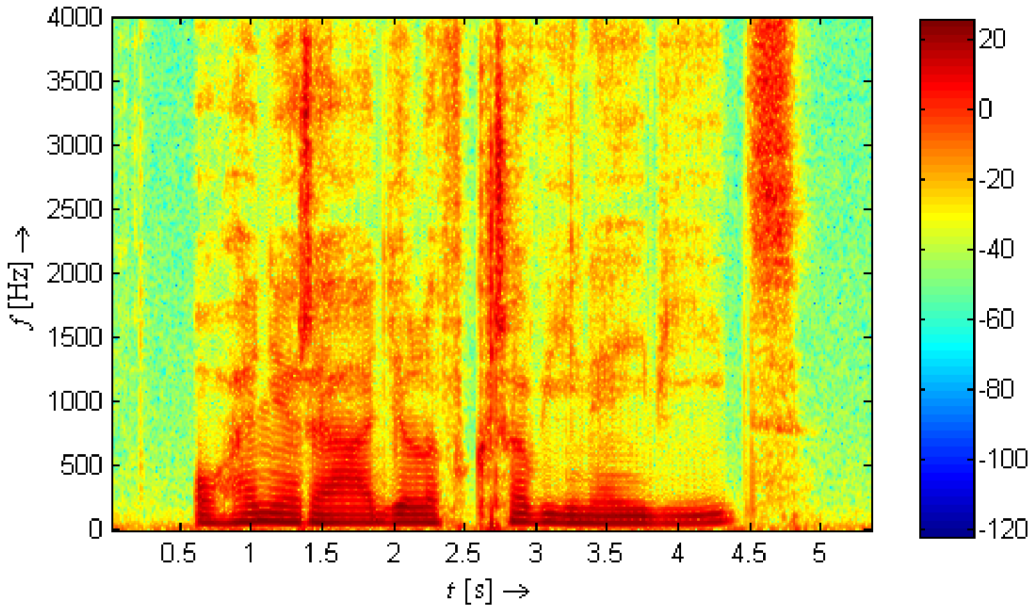
$$\Delta f = \frac{f_{\mathrm{vz}}}{N}. \tag{2.8}$$

f_{vz} je vzorkovací kmitočet signálu $x[n]$.

Efektivní algoritmus, který vypočítává diskrétní Fourierovu transformaci, se nazývá rychlá Fourierova transformace (FFT).

2.3.3 Spektrogram

Pro jednodušší a rychlejší zkoumání vlastností je výhodné vynášet průběh řečového signálu do grafu, kde na horizontální osu vynášíme čas a na vertikální kmitočet. Podle intenzity barev zastoupených v tomto zobrazení dostaneme informaci o intenzitě každého kmitočtu. Tento způsob zobrazení se nazývá spektrogram. Příklad spektrogramu je na obr. 2.6.



Obr. 2.6: Spektrogram řečového signálu ($f_{\mathrm{vz}} = 8\,\mathrm{kHz}$, věta: „Oheň řádil v křoví nejvíc.“)

V pravé části tohoto spektra lze dobře rozlišit spektrum periodického signálu od spektra šumového signálu. Periodický signál má většinu energie rozloženu na nižších kmitočtech, a navíc jsou u periodických signálů výrazné horizontální pruhy se zvýšenou intenzitou. U šumového signálu je energie rozprostřena spíše ve vyšších částech spektra a horizontální pruhy se zvýšenou energií se zde nevyskytují. Na spektrogramu jsou také dobře vidět okluzivní pauzy. Jestliže bude tato věta nahrána v souboru „`veta.wav`“, pak takový spektrogram v MATLABu vykreslíme pomoci následující posloupnosti příkazů:

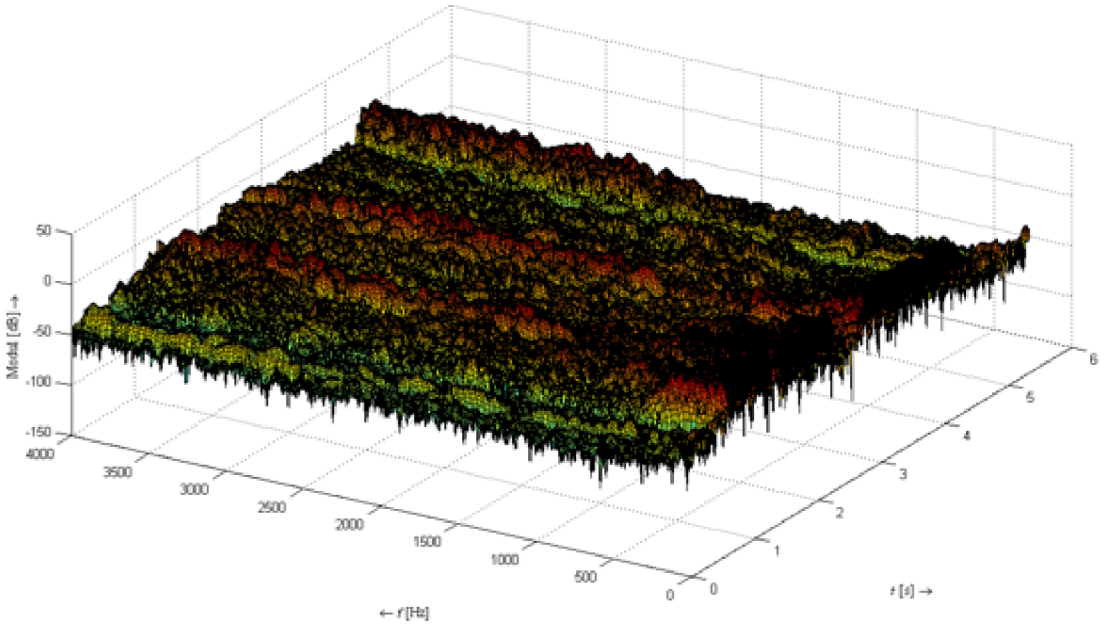
```
[y,FS,Nb] = wavread('veta.wav'); %načtení *.wav souboru

specgram(y,512,FS,hamming(345),320); %vykreslení spektrogramu
colorbar; %zapnutí barevné legendy
```

```
ylabel('{\itf} [Hz] \rightarrow'); %popis os
xlabel('{\itt} [s] \rightarrow');
```

V prvním řádku si nahrajeme do paměti samotný signál `y`, hodnotu jeho vzorkovacího kmitočtu `FS` a počet bitů na jeden vzorek `Nb`. Spektrogram pak vykreslíme pomocí funkce `specgram`.

2.3.4 3D spektrogram

3D spektrogram je forma spektrogramu, která je zobrazena v trojrozměrném prostoru. Oproti klasickému spektrogramu tak vynášíme intenzitu jednotlivých kmitočtů na osu z. Příklad spektrogramu můžeme opět vidět na obr. 2.7.



Obr. 2.7: 3D spektrogram řečového signálu ($f_{vz} = 8\,\text{kHz}$, věta: „Oheň řádil v křoví nejvíc.“)

Jestliže bychom v určitém čase udělali průřez 3D spektrogramem rovinou rovnoběžnou s kmitočtovou osou a osou z, dostali bychom spektrum signálů v tomto čase. 3D spektrogram v MATLABu vykreslíme pomoci následující posloupnosti příkazů:

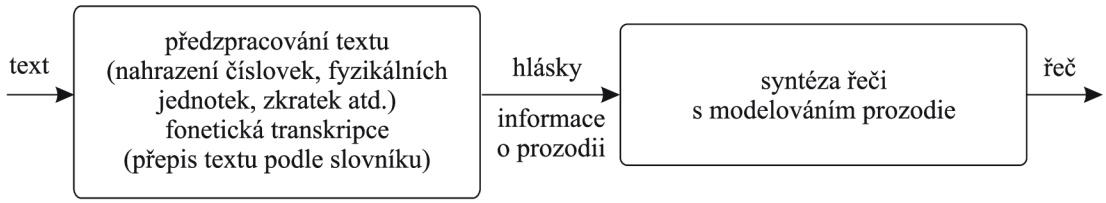
```
[y,FS,Nb] = wavread('veta.wav'); %načtení *.wav souboru

[S,F,T] = specgram(y,512,FS,hamming(345),320); %načtení vektorů pro
vytvoření 3D spektrokramu
```

```
surf(T,F,20*log10(abs(S))); %vykreslení 3D spektrogramu

zlabel('Modul [dB] \rightarrow'); %popisky os
ylabel('\leftarrow {\itf} [Hz]');
xlabel('{\itt} [s] \rightarrow');
```

Nejdříve si pomocí funkce `specgram` nahrajeme vektor časů `T`, vektor koeficientů spektra `S` a vektor kmitočtů `F`, dále pak pomocí funkce `surf` vykreslíme 3D spektrogram.

3 SYSTÉMY TTS

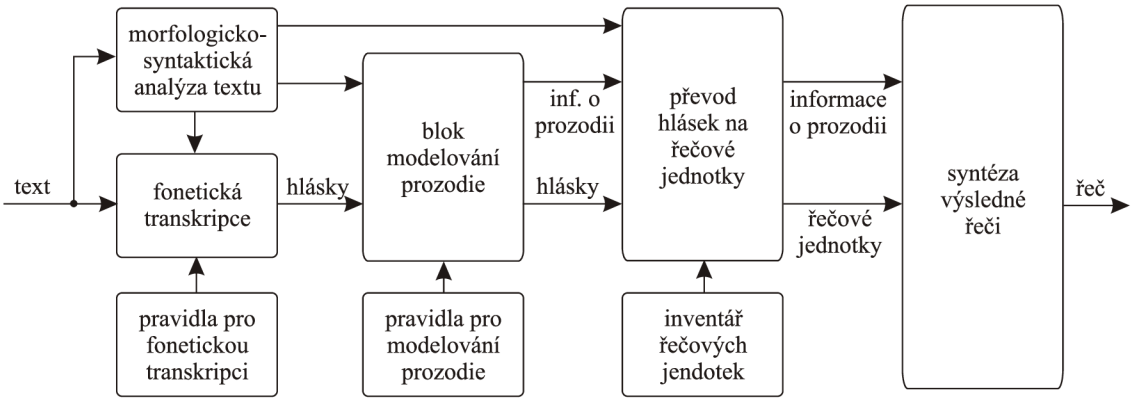
Jak už bylo v úvodu řečeno, systémy TTS zajišťují automatický překlad psaného textu do mluvené řeči, přičemž se u těchto systémů snažíme docílit co největší věrohodnosti produkované řeči. První systémy nepracovaly s prozodii, produkovaná řeč byla potom monotónní a nepříjemná. Další systémy už začaly pracovat se základním tónem, který je z prozodického hlediska nejdůležitější. Při jednoduché úvaze můžeme systém TTS rozdělit do dvou bloků, tak jak je to zobrazeno na obr. 3.1.



Obr. 3.1: Základní blokové schéma systému TTS

Z obrázku je patrné, že než dojde k samotné syntéze řeči z textu, musíme vstupní řetězec znaků předzpracovat, rozložit ho na hlásky a přidat informaci o prozodii, kterou by byl systém schopen zpracovat. Druhý blok by potom na základě hlásek provedl syntézu řeči a na základě dalších vstupních informací by modeloval prozodii. Výstupem by pak byla syntetická řeč.

Pro podrobnější zkoumání systému TTS rozložíme dva hlavní bloky na další části tak, jak je tomu na obr. 3.2.



Obr. 3.2: Podrobnější blokové schéma systému TTS

První blok základního blokového schématu jsme rozložili na dalších sedm částí a blok samotné syntézy řeči s modelováním prozodie jsme již dále nedělili. Nyní si

probereme každý blok systému TTS zvlášť, řekneme si jakou plní úlohu, popř. jaký má v systému TTS význam.

3.1 Morfologicko-syntaktická analýza textu

Než dojde k samotné fonetické transkripci, je potřeba vstupní text upravit do takové podoby, aby výsledná řeč obsahově a významově korespondovala s psaným textem. To je velmi komplexní úkol, který také závisí na jazyku, ve kterém je text psán. Abychom dosáhli požadovaných výsledků, musíme vstupní text podrobit nejdříve morfologicko-syntaktické analýze. Tato analýza odhaluje strukturu každé věty a definuje jednotlivé mluvnické vztahy mezi slovy, nebo mezi slovem a větou. Každý blok morfologicko-syntaktického analyzátoru text nějakým způsobem upraví a pošle dál ke zpracování. Tyto bloky jsou řazeny sekvenčně nebo paralelně, přičemž mohou do textu zasáhnout i vícekrát. Níže si popíšeme základní bloky morfologicko-syntaktického analyzátoru.

3.1.1 Předzpracování textu

Každý vstupní text musíme nejdříve unifikovat na takový formát, který je TTS systém schopen zpracovat. Tento blok provede detekci textu (prostý text, XML, HTML, e-mail, naskenovaný text atd.) a odfiltruje nadbytečné znaky, jako jsou např. znaky bílé a znaky formátovací. Dále může text rozdělit do různých bloků, odstavců nebo vět. Nakonec ještě detekuje místo ukončení věty. K tomu potřebuje spolupráci dalších bloků, jelikož např. tečka v textu nemusí vždy znamenat konec věty (pořadové číslice). Proto může tento blok pracovat paralelně s kontextovou analýzou, nebo může na detekci konce věty použít neuronové sítě.

3.1.2 Normalizace textu

Tento blok převádí vstupní text do úplné slovní formy. Je potřeba nahradit všechny čísla slovy. Např. číslo 1324 se musí nahradit slovy „tisíc tři sta dvacet čtyři“. Z toho je patrné, že musíme také dbát na řád, ve kterém se daná číslice vyskytuje. Dále jsou nahrazena všechna pořadová čísla. Např. větu „Podal mi 3. propisku.“ přepíšeme jako „Podal mi třetí propisku.“ Ovšem spojení „3. srpna“ přepíšeme jako „třetího srpna“. Proto musí tato komponenta také spolupracovat s komponentou kontextové analýzy, aby bylo jasné, v jakém pádě se daná číslice nachází. V textu je také potřeba slovy nahradit římské číslice, např. „Henry VIII.“ přepíšeme jako „Henry osmý“.

Dále musí být v textu nahrazeny všechny zkratky. Zkratku „např.“ přepíšeme jako „například“, zkratku „Doc.“ přepíšeme jako „docent“, zkratku „OSN“ přepíšeme jako „Organizace spojených národů“ atd. Některé zkratky se ovšem vyslovují tak, jak jsou psány, např. „SAS“ a „RAF“. Jestliže se v textu vyskytují fyzikální jednotky, je potřeba zkratky těchto jednotek opět přepsat. Např. „4 Ω“ na „čtyři ómy“. Přepsat musíme také některé speciální znaky, jako např. „+“, „%“, „€“ a „@“.

3.1.3 Morfologická analýza

Tento blok v textu detekuje slova ohebná a neohebná a určí všechny možné mluvnické kategorie, ve kterých se daná slova mohou nacházet. Důležité je to především u slov ohebných, která se mohou na základě kontextu (určuje pád, rod, čas atd.) vyskytovat s různými předponami, příponami a koncovkami. Při této analýze je potřeba detekovat slovní základ (kmen — většinou pomocí rozsáhlého slovníku) a způsob utvoření daného slova pomocí předpon, přípon a koncovek (většinou pomocí pravidel).

3.1.4 Kontextová analýza (tagging)

Tento blok redukuje seznam všech mluvnických kategorií morfologické analýzy na základě kontextu okolních slov. K tomu používá metodu pravděpodobnostní (založena na přechodových pravděpodobnostech mezi sousedními mluvnickými kategoriemi dvou slov) nebo metodu deterministickou (přijímá nebo odmítá dané možnosti na základě klasifikačních a regresních stromů). Jestliže by blok morfologické analýzy zpracoval např. slovo „pekla“, určil by dvě mluvnické kategorie.„Pekla“ jako podstatné jméno v druhém pádě a „pekla“ jako sloveso v minulém čase. Kontextový analyzátor by pak např. z věty „Zdeňka pekla buchtu.“ omezil výběr pouze na jednu mluvnickou kategorii, a to na sloveso v minulém čase.

3.1.5 Syntakticko-prozodický rozbor

Dále je potřeba nalézt ve zpracované větě větné úseky, které úzce souvisejí s očekávanou prozodickou realizací věty. K tomu se používají různé metody, jako např. metoda ručně odvozené heuristiky (fráze se určují na základě pozic čárek, středníků, pomlček, závorek atd.), metoda používající gramatiku nebo korpusově orientovaná metoda (modelování hranic se provádí pomocí Markovových řetězců).

Uvedené bloky nemusí vždy pracovat bezchybně, což se také může projevit na výsledné syntetické řeči. Systém TTS nemusí např. správně dekódovat vstupní znakovou sadu, může zaměnit některá slova za zkratky nebo může špatně odhadnout

kontext. Problém by také mohl nastat tehdy, kdyby systém text automaticky načítal z předlohy (např. při skenování). Při psaní textu do sloupců nebo jiných útvarů (např. básně či novinové články) může dojít ke špatné detekci toku, a systém by mohl zaměnit pořadí jednotlivých textových řetězců.

3.2 Fonetická transkripce

Jestliže máme vstupní text již předzpracován, můžeme přistoupit k samotné fonetické transkripci. Fonetickou transkripcí rozumíme, nahrazování znaků v textovém řetězci schematickým zápisem mluvené řeči. Před několika desítkami lety vyvstal požadavek na vytvoření univerzální fonetické abecedy, která by dokázala přepsat text v jakémkoliv jazyce. Tak vznikla nejvýznamnější mezinárodní fonetická abeceda IPA (International Phonetic Alphabeth). Ovšem tato abeceda nebyla vhodná pro účely strojového zpracování řeči, proto byla sestavena fonetická abeceda SAMPA (Speech Assessment Methods Phonetic Alphabet) pro strojový přepis. Tato abeceda byla pak ještě pro každý jazyk zvlášť upravena. Srovnání fonetických abeced češtiny nalezneme v tab. 3.1 [13]. Kromě fonetických abeced IPA a SAMPA zde máme také českou fonetickou abecedu (ČFA).

Text přepsaný pomocí fonetické abecedy již nese část prozodické informace, se kterou budeme dále pracovat. Například znakem „:“ zajistíme to, aby byl daný foném vysloven dlouze.

3.2.1 Výslovnost

Než se dostaneme k samostatné fonetické transkripci, musíme si uvědomit, že mnohá slova se vyslovují jinak, než se píší. Vytváří se proto slovníky, na základě kterých daná slova nejdříve upravíme (nahradíme některé znaky), a teprve potom podrobíme transkripci. To je například případ anglického jazyka, kde tyto slovníky dosahují velkého objemu. Naopak např. finština je jazyk, kde se většina slov čte na základě složených hlásek. V češtině se vytvářejí jen krátké slovníky, které pomáhají slova přepsat tak, jak se čtou. Přepisuje se např. „Josef“ na „Jozef“, „populistická politika“ na „populistycká polityka“ atd. V některých jazycích se také odvíjí výslovnost slova na umístění ve větě. Příkladem může být výslovnost anglického slova „lives“ ve větách „Four lives were lost.“ a „One lives to eat.“ Výslovnost cizích jmen může taktéž přinášet jisté potíže.

Tab. 3.1: Srovnání fonetických abeced češtiny (zjednodušená česká fonetická abeceda (ZČFA), IPA, SAMPA a česká fonetická abeceda (ČFA))

ZČFA	IPA	SAMPA	ČFA	Příkl.	ZČFA	IPA	SAMPA	ČFA	Příkl.
vokály					frikativy				
i	ɪ	i	i	lis	ř	r̝	P\	rzh	moře
e	ɛ	e	e	pes	r	r	r	r	rok
a	a	a	a	sad	l	l	l	l	vlak
o	o	o	o	kov	j	j	j	j	jev
u	u	u	u	sukně	plozivy				
í	i:	i:	ii	víno	p	p	p	p	pec
é	ɛ:	e:	ee	lék	b	b	b	b	bratr
á	a:	a:	aa	sál	t	t	t	t	tuk
ó	o:	o:	oo	kód	d	d	d	d	dům
ú	u:	u:	uu	růže	ť	c	c	tj	děti
diftongy					ď	ɟ	J\	dj	děti
ou	oʊ	o_u	ow	bouda	k	k	k	k	kost
au	aʊ	a_u	aw	auto	g	g	g	g	tygr
eu	eʊ	e_u	ew	euro	nazály				
frikativy					m	m	m	m	muž
f	f	f	f	fíky	n	n	n	n	víno
v	v	v	v	vítr	ň	ŋ	J	nj	laňka
s	s	c	s	sůl	afrikáty				
z	z	z	z	koza	c	ts	t_s	c	cena
š	ʃ	S	sh	škola	č	tʃ	t_S	ch	oči
ž	ʒ	Z	zh	žena	dz	dz	d_z	dz	podzim
ch	x	x	x	chata	dž	dʒ	d_Z	dzh	džbán
h	ɦ	h\	h	hůl					

3.3 Modelování prozodie

Prozodické rysy se mohou rozdělit do několika úrovní: prozodie hlásky, prozodie slova a prozodie věty (celku). Prozodické (suprasegmentální) informace jsou modelovány třemi základními parametry: základním tónem, trváním a intenzitou řeči. Mimo to mohou lidskou prozodii také ovlivňovat pauzy, artikulace, barva hlasu a dýchání apod. Vhodnou kombinací těchto faktorů pak můžeme dosáhnout simulace různého pohlaví, věku a emocí řečníka. Rozdělení suprasegmentálních účinků do třech nezávislých oblastí, jako je oblast časová, kmitočtová a intenzitní je však prakticky nemožné, jelikož v mluvené řeči jsou tyto oblasti vzájemně provázány. V TTS systémech se k modelování prozodie nejčastěji používají tři uvedené hlavní parametry. Různými pokusy bylo dokázáno, že je lidské ucho více citlivé na změnu těchto parametrů v syntetické řeči, než v řeči lidské. Dále si popíšeme vlivy těchto jednotlivých parametrů na výslednou řeč.

3.3.1 Vliv základního tónu

Variace základního tónu F_0 se jeví jako nejdůležitější parametr při modelování prozodie. Změnu F_0 vnímáme jako změnu výšky hlasu. Základní kmitočet nám udává počet hlasivkových kmitů za sekundu. Jak již bylo dříve řečeno, hodnota tohoto kmitočtu závisí na pohlaví, stáří, individuálních fyziologických vlastností a citovém stavu řečníka, takže může nabývat hodnot od 80 Hz (dospělý muž) až do 450 Hz (dítě).

Průběh výšky základního tónu může ovlivnit intonaci, a tím i význam věty. U klasické oznamovací (deklarativní) věty intonace postupně klesá. K přímému poklesu intonace dochází také u věty rozkazovací. Naproti tomu u věty tázací intonace postupně roste. Intonace stoupavě-klesavá se vyskytuje např. u vět doplňujících. Zvýšením kmitočtu základního tónu můžeme také modelovat důraz na danou hlásku nebo slovo. Základní tón je úzce spjat i s dalšími parametry. Např. současný pokles F_0 a intenzity na konci prozodické a tematické jednotky označuje ukončení tématu nebo promluvy, eventuelně vybízí k převzetí slova.

3.3.2 Vliv intenzity

Síla hlasu (hlasitost) je modelována intenzitou řečového signálu. Její úroveň je spojena s funkcí dýchacího a fonačního systému a je přímo úměrná subglotálnímu (hlasivkovému) tlaku. Budeme-li u neutrálních vět považovat průměr intenzity v rámci věty za 100 %, pak poslední slabika dosahuje pouze 25 % a hodnota intenzity první

slabiky poslední rytmické skupiny (nositelky přízvuku) se pohybuje okolo 45 %. Začátek věty, tedy první, přízvučná slabika, odpovídá 115 % intenzity průměrné hodnoty [19]. Změna intenzity také odpovídá změnám citového a emocionálního stavu řečníka. I když je intenzita uvedena jako jeden ze tří hlavních parametrů prozodie, nemívá v systémech TTS velkou váhu a někdy se dokonce opomíjí, jelikož je považována za odvozený parametr z hodnoty kmitočtu základního tónu, protože je stejně jako on spjata se změnami subglotálního tlaku.

3.3.3 Vliv trvání

Celek časových složek, trvání slabik a pauz, je vnímán jako tempo řeči. Liší se trvání jednotlivých fonémů, přičemž toto trvání spolu s tempem řeči závisí na postoji a emocích mluvčího. Každý mluvčí má své individuální tempo, proto neexistuje nějaká přesně daná hodnota, kterou bychom se mohli řídit. Tab. 3.2 [19] udává tempo řeči v češtině podle studií různých autorů v počtu slabik za sekundu.

Tab. 3.2: Tempo řeči v češtině podle studií různých autorů v počtu slabik za sekundu

	maximální tempo	minimální tempo	průměrné tempo
Romportl, M. (1958)	4,83	2,58	
Bartošek, J. (1974)	6,67	3,33	
Palková, Z. (1987)			4,89
Štěpánková, O. (1985)	4,21	2,21	
Sedláková, M. (1989)	3,88	1,84	

3.3.4 Další parametry prozodie

K dalším (vedlejším) parametrům prozodie patří artikulace, což je v podstatě způsob vytváření hlásek pohyby mluvidel. Díky artikulaci jsme schopni rozlišit jednak jednotlivé hlásky, jednak jejich mírné rozdíly odpovídající buď fyzickému a psychickému stavu mluvčího, nebo jeho komunikačnímu záměru. Dalším vedlejším parametrem je dýchání, které ovlivňuje intenzitu promluvy, její plynulost a také průběh základního tónu. Neefektivní hospodaření s dechem může narušit plynulost promluvy a ta se pak může stát nesrozumitelnou. Pauzy taktéž mají v prozodii svou roli. Obecně dělíme pauzy na dvě kategorie: pauzy tiché, které postrádají jakékoliv akustické podněty[1], a pauzy vyplněné různými zvuky. Trvání pauz mezi větami je v přirozené řeči velmi proměnné. V syntezátorech se délka trvání pauzy mezi větami většinou

[1] většinou označují gramatickou nebo stylistickou hranici

odvíjí od interpunkčního znaménka. V tab. 3.3 jsou uvedeny délky trvání pauz podle interpunkčních znamének na základě literatury [14].

Tab. 3.3: Závislost trvání pauz za větami na interpunkčních znacích

Interpunkční znak za větou (intonačním úsekem)	Percepčně přijatelné trvání pauzy u syntetického signálu
. ! ?	420 ms
; : -	280 ms
, ( )	140 ms

3.3.5 Emoce

Emoce jsou komplexním jevem, který zahrnuje zážitek, projevy fyziologické i různé projevy chování [19]. Emoce se projevuje jako emocionální chování, jehož součástí je i chování řečové. Tudíž dochází při emocích ke změně prozodie mluvčího (výška základního tónu, tempo, intenzita). Kromě této změny dochází i ke změnám srdečního rytmu a krevního tlaku. Může dojít i k žaludečním stahům, bledosti nebo naopak červenání. Obecně se emoce dělí do dvou skupin:

- emoce syrová — fyziologická narušení, šok, záchvat
- socializovaná emoce — chování, chronická forma nějakého stavu, způsob bytí

Ovšem v dalším dělení se literatura již různí. Pro naši potřebu si emoce dále rozdělíme na čtyři kategorie, přičemž budeme uvažovat, že další emoce vzniknou kombinací těchto kategorií [19]. Tyto emoce tedy jsou:

- radost — reakce na zisk či úspěch
- vztek — emocionální reakce na nějakou překážku
- smutek — reakce na neúspěch či ztrátu
- strach — emocionální reakce na hrozbu

Kombinacemi těchto emocí může vzniknout euforie, rozpaky, frustrace, stud, něha, ironie, hrozba atd. Podle hlasových projevů můžeme emoce rozdělit na tyto skupiny:

- aktivní příjemné emoce — mají vyšší globální úroveň hlasu, větší intenzitu a vyšší tempo

- pasivní příjemné emoce — pomalejší tempo, pravidelný rytmus a melodické stoupání
- aktivní nepříjemné emoce — vyšší globální hodnota základního tónu, vysoká intenzita, nepravidelný rytmus
- pasivní nepříjemné emoce — nepravidelné intonační průběhy

Vyšší poloha hlasu je spjata s radostí nebo lehkostí, ale i se stydlivostí, nižší poloha charakterizuje smutek, vážnost a ujištění. Úroveň základního tónu je spojena s mírou aktivity vyjadřované emoce: veliké rozdíly mezi minimální a maximální hodnotou jsou spojené s radostí a se vztekem, malé rozdíly se smutkem a nudou. Stylotvorná funkce intonace je v řeči modelována průměrnou výškou hlasu. Distribuce průměrné hodnoty základního tónu je například v přednášce pravidelnější než v konverzaci, naopak je méně pravidelná v přednášce než ve čtené promluvě.

Jak už bylo řečeno dříve, jednotlivé suprasegmentální parametry řeči jsou vzájemně propojené a přirozená promluva je neumožňuje měnit jednotlivě. Při modelování prozodie v systému TTS budeme vycházet ze tří základních prozodických parametrů (základní tón, intenzita, doba trvání), které budeme pro jednotlivé emoce různě měnit. K přibližnému nastavení těchto parametrů budeme využívat poznatky Pierra R. Léona [19], který na základě různých experimentů sestavil prozodické charakteristiky různých postojů ve francouzském jazyce. Tyto charakteristiky doplníme prozodickými charakteristikami vybraných postojů zjištěných na základě experimentu Vlčkové-Mejvaldové, která zjišťovala hodnoty základních parametrů prozodie přímo pro český jazyk. V tab. 3.4 máme uvedeny poznatky Léona a v tab. 3.5 jsou sepsány výsledky experimentu Vlčkové-Mejvaldové [19].

3.3.6 Způsoby modelování hlasového traktu

Syntetická řeč se v systémech TTS získává dvěma základními způsoby:

- explicitně, na základě chování hlásek a jejích přechodů, či modelováním mechanických parametrů hlasového traktu
- implicitně, na základě uchování jednotlivých řečových jednotek (fonémy, difóny, trifóny atd.) a následném poskládání v takovém pořadí, které udává vstupní text

Jedna ze syntéz, které používají explicitní postup, je formantová syntéza. Tato syntéza je úspěšná a byla často používána v různých TTS systémech. Ovšem kvůli

Tab. 3.4: Prozodické charakteristiky vybraných postojů podle Pierra R. Léona (1972)

postoj	průměr F_0 [Hz]	minimální a maximální hodnoty F_0 a jejich rozdíl [Hz]	tempo (sl./s)	percepční charakteristika globální intenzity	procento identifikace skupinou posluchačů
neutralita	120	$80 \sim 160$ $E = 80$	6,82	střední	100 %
radost	150	$90 \sim 225$ $E = 135$	8,02	slabá	60 %
smutek	100	$85 \sim 140$ $E = 55$	7,25	slabá	100 %
vztek	200	$120 \sim 320$ $E = 200$	6,13	silná	70 %
stud	150	$110 \sim 235$ $E = 125$	7,21	slabá	80 %
strach	175	$120 \sim 225$ $E = 105$	6,17	střední	60 %
překvapení	175	$110 \sim 250$ $E = 140$	7,14	střední	60 %
pohoršení	125	$70 \sim 200$ $E = 130$	6,14	slabá	90 %
reklama	175	$80 \sim 270$ $E = 190$	5,63	silná	90 %

své složitosti byla většinou nahrazena konkatenační syntézou. Byla založena na modelování charakteristických kmitočtů hlasového traktu pomocí banky filtrů. Mezi její výhody patřily malé nároky na paměť počítače, snadné řízení prozodických charakteristik a plynulá řeč. Musely se však pracně nastavovat jednotlivé parametry, které byly navíc na sobě velmi závislé, takže při změně jednoho parametru se musely ostatní taktéž pozměnit. Asi nejznámějším zástupcem těchto syntetizérů je Klattův formantový syntezátor [13].

Postupem času se zvyšovaly paměti počítačů, a tím pádem se mohly aplikovat implicitní metody syntézy řeči, které požadovali část paměti pro uchování řečových jednotek. Mezi tyto metody patří dnes asi nejužívanější konkatenační syntéza. Základní princip této syntézy je ten, že jednotlivé zvuky, ze kterých se řeč skládá, lze

Tab. 3.5: Naměřené prozodické charakteristiky vybraných postojů na základě experimentu Vlčkové-Mejvaldové [19]

postoj	průměr F_0 [Hz]	minimální a maximální hodnoty F_0 a jejich rozdíl [Hz]	tempo (sl./s)
neutralita	132	83 ∼ 202 E = 119	7,8
radost	269	172 ∼ 379 E = 207	4,4
překvapení	97	96 ∼ 161 E = 68	4,4
vztek	247	133 ∼ 270 E = 137	5,4
smutek	91	75 ∼ 130 E = 55	4,4
nuda	112	89 ∼ 148 E = 59	4,4
strach	107	76 ∼ 150 E = 74	3,8
obdiv	117	103 ∼ 144 E = 41	6,9

reprezentovat pomocí konečného počtu řečových jednotek. Konkatenační syntézu můžeme dle způsobu modelování řečového signálu dělit na:

- modelování řečového signálu v časové oblasti (TD-PSOLA)
- modelování řečového signálu ve kmitočtové oblasti (LPC syntéza, kepstrální syntéza, harmonické modelování)
- hybridní systémy (LP-PSOLA, MBROLA)

Při modelování řečového signálu v časové oblasti dochází ke skládání navzorkovaných úseků řeči, přičemž tyto úseky mohou mít proměnlivou délku. Mezi nejznámější zástupce této metody patří TD-PSOLA (Time Domain Pitch Synchronous Overlap Add). Výsledkem syntézy v časové oblasti je dostatečně přirozená řeč, ovšem při požadavku na modelování prozodie není tato metoda vhodná. Další nevýhodou je nespojitost vzorků v hodnotě základního tónu a ve spektrální obálce. Proto se tyto metody různě upravovaly a vznikly hybridní systémy jako např. MBROLA (Multi Band Resynthesis Overlap Add).

Dnes se asi nejvíce používá modelování řečového signálu ve kmitočtové oblasti. Toto modelování bývá nejčastěji založeno na teorii zdroje a filtru, která je znázorněna na obr. 2.2. Při tomto postupu uvažujme, že výsledný řečový signál vznikl omezením budicího signálu (periodický sled pulsů nebo šum) filtrem. Díky syntéze v kmitočtové oblasti jsme schopni lépe modelovat prozodické vlastnosti mluvčího, především pak průběh základního tónu, který jakožto hlavní suprasegmentální parametr nejvíce ovlivňuje prozodii. Pro další popis modelování prozodie ve kmitočtové oblasti budeme hovořit o hlasovém traktu zjednodušeně jako o již zmíněném filtru, který je popsán impulsní či kmitočtovou charakteristikou. Dále se také zaměříme pouze na modelování prozodie, které využívá lineárně predikční a kepstrální analýzu k popisu hlasového traktu. Na základě těchto analýz budeme odhadovat průběh kmitočtové charakteristiky hlasového traktu v krátkém řečovém úseku. Protože obecně není řeč stacionárním signálem (signál, jehož libovolné statistické charakteristiky nejsou závislé na libovolném přemístění počátku časové osy), musíme ji rozložit na již zmíněné krátké úseky (10 ms až 30 ms), které budeme považovat za stacionární. Potom co provedeme analýzu každého řečového úseku a dostaneme popis kmitočtové charakteristiky hlasového traktu, není již problém provést zpětnou syntézu tak, jak ji popisuje teorie zdroje a filtru.

Lineární predikční analýza

Tato dnes hojně využívaná analýza je založena na autoregresivním modelu (Auto-Regressive AR), používající IIR filtr s póly, které leží uvnitř jednotkové kružnice v rovině z. Metoda poměrně dobře odhaduje parametry řeči, přičemž nevyžaduje vysoké nároky na operační paměť. Navíc docela dobře kóduje řečové úseky, proto se stala velmi oblíbenou metodou při syntéze hlasu. Lineární predikční analýza, též označovaná jako LPC (Linear Predictive Coding), odhaduje aktuální vzorek řeči $\tilde{s}[n]$ z M předešlých vzorků. Pro dopřednou lineární predikční analýzu platí vztah: [16]

$$\tilde{s}[n] = -\left(a_1 s[n-1] + a_2 s[n-2] + \ldots + a_M s[n-M]\right) = -\sum_{i=1}^{M} a_i s[n-i], \quad (3.1)$$

kde $\tilde{s}[n]$ je predikovaný vzorek, M udává počet předchozích vzorků a a_i koeficienty filtru (lineární predikční koeficienty). Označíme li $s[n]$ jako referenční vzorek, potom chyba predikce $e[n]$ je rovna: [16]

$$e[n] = s[n] - \tilde{s}[n] = s[n] + \sum_{i=1}^{M} a_i s[n-i]. \quad (3.2)$$

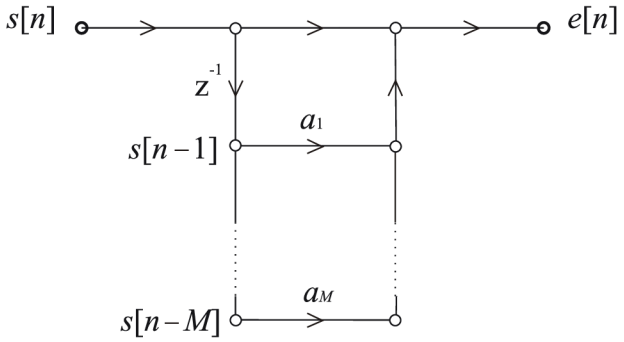
Tato chyba je výstupem inverzního lineárně predikčního filtru, jehož graf signálových toků je na obr. 3.3. Při syntéze pak rovnici (3.2) upravíme na tvar:

$$s[n] = e[n] - \sum_{i=1}^{M} a_i s[n-i]. \quad (3.3)$$

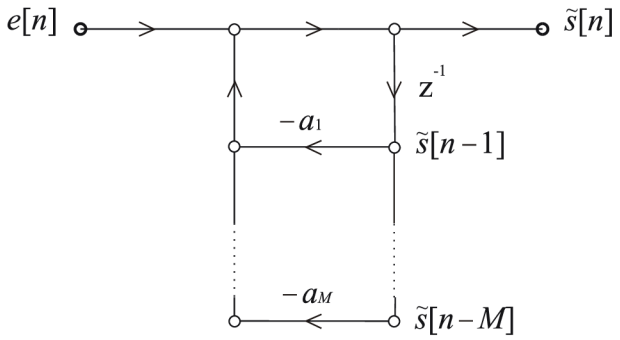
Graf signálových toků syntetizujícího filtru se změní podle obr. 3.4, přičemž jeho přenosová funkce bude ve tvaru: [16]

$$H(z) = \frac{G}{1 + \sum_{i=1}^{M} a_i \mathrm{z}^{-i}}, \tag{3.4}$$

kde G je zesílení signálu.



Obr. 3.3: Graf signálových toků inverzního lineárně predikčního filtru



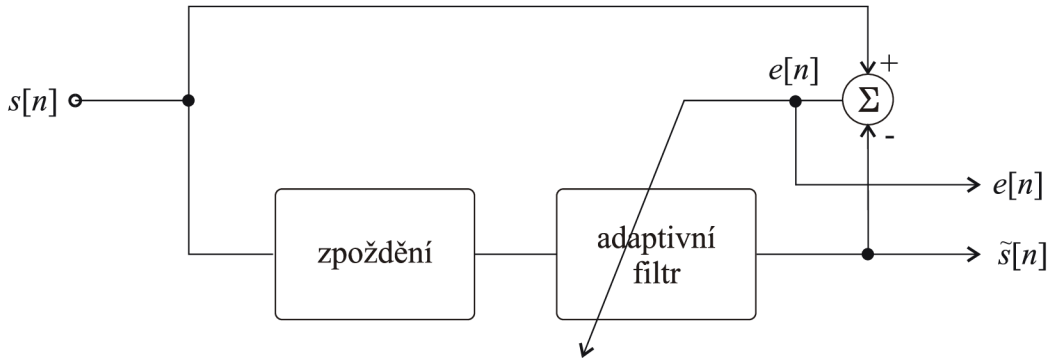
Obr. 3.4: Graf signálových toků syntetizujícího lineárně predikčního filtru

K řešení rovnice (3.2) a nalezení lineárních predikčních koeficientů filtru se používají různé autokorelační metody, přičemž jako nejefektivnější se jeví Levinson-Durbinův algoritmus založený na Levinsonově rekurzi. Postup, při kterém se tyto koeficienty zjišťují, znázorňuje blokové schéma s adaptivním filtrem na obr. 3.5.

Pro správnou volbu řádu lineární predikce M se používá poučka vycházející ze vzorkovacího kmitočtu: [15]

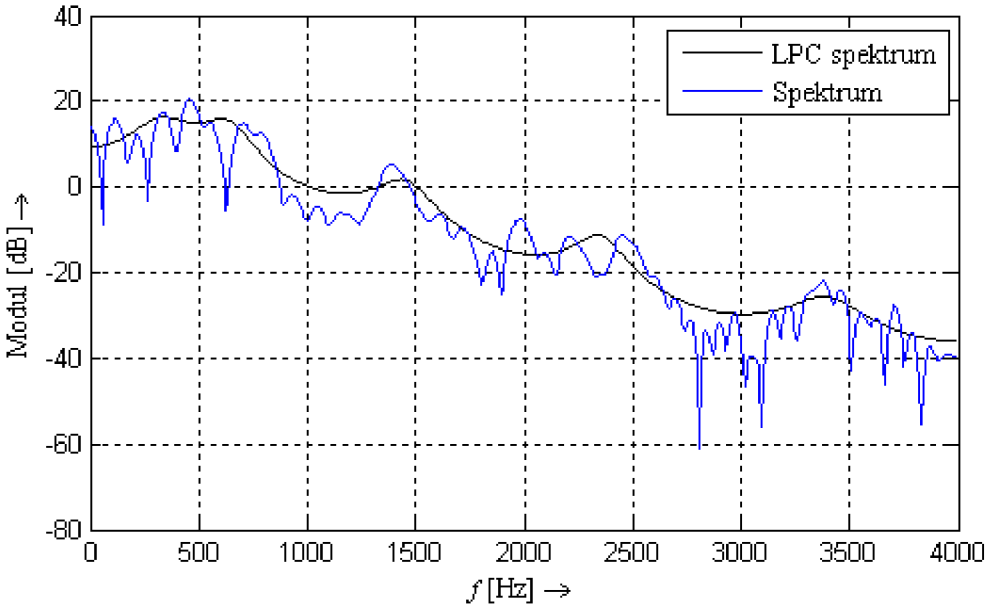
$$M = f_{\mathrm{vz}}[\mathrm{kHz}] + 4, \tag{3.5}$$

kde f_{vz} je vzorkovací kmitočet řečového signálu. Tato metoda využívající lineární predikci má nevýhodu v tom, že filtr popisující hlasový trakt obsahuje pouze póly



Obr. 3.5: Použití adaptivního filtru pro lineární predikci

a modeluje pouze formanty a ne antiformanty. Příklad LPC spektra hlásky „á“ znázorňuje obr. 3.6.

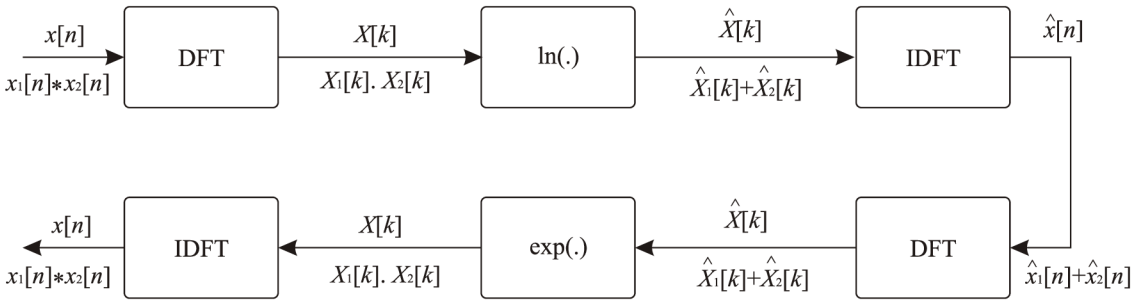


Obr. 3.6: Spektrum a LPC spektrum hlásky „á“ ($f_{\mathrm{vz}} = 8\,\mathrm{kHz}$)

Kepstrální analýza

Kepstrální analýza patří do skupiny nelineárního zpracování signálů, které jsou založeny na využití zobecněného principu superpozice. Tyto nelineární postupy jsou s výhodou používány k oddělování signálů, které vznikly konvolucí či násobením dvou a více složek, to je i případ řeči. Obecné schéma postupu nelineárního zpracování signálu je na obr. 3.7. Ze schématu je patrné, že díky přesunu z časové oblasti

do kmitočtové, jsme převedli konvoluci signálů na součin a následným logaritmováním jsme změnili operaci násobení na sčítání. Pro správnou rekonstrukci signálu pak musí být spodní část schématu inverzní k horní.



Obr. 3.7: Obecné schéma postupu nelineárního zpracování signálu

Celý postup můžeme popsat následujícími rovnicemi: [13]

$$
\begin{aligned}
x[n] &= x_1[n] * x_2[n], && (3.6)\\
X[k] &= \mathrm{DFT}\{x[n]\} = \mathrm{DFT}\{x_1[n] * x_2[n]\} = X_1[k] \cdot X_2[k], && (3.7)\\
\hat{X}[k] &= \ln(X[k]) = \ln(X_1[k] \cdot X_2[k]) = \ln(X_1[k]) + \ln(X_2[k]) = \\
&= \hat{X}_1[k] + \hat{X}_2[k], && (3.8)\\
\hat{x}[n] &= \mathrm{IDFT}\left\{\hat{X}[k]\right\} = \mathrm{IDFT}\left\{\hat{X}_1[k] + \hat{X}_2[k]\right\} = \hat{x}_1[n] + \hat{x}_2[n], && (3.9)\\
\hat{X}[k] &= \mathrm{DFT}\{\hat{x}[n]\} = \mathrm{DFT}\{\hat{x}_1[n] + \hat{x}_2[n]\} = \hat{X}_1[k] + \hat{X}_2[k], && (3.10)\\
X[k] &= \exp\left(\hat{X}[k]\right) = \exp\left(\hat{X}_1[k] + \hat{X}_2[k]\right) = \\
&= \exp\left(\hat{X}_1[k]\right) \cdot \exp\left(\hat{X}_2[k]\right), = X_1[k] \cdot X_2[k], && (3.11)\\
x[n] &= \mathrm{IDFT}\{X[k]\} = \mathrm{IDFT}\{X_1[k] \cdot X_2[k]\} = x_1[n] * x_2[n], && (3.12)
\end{aligned}
$$

kde $X[k]$ je obrazem diskrétní Fourierové transformace signálu $x[n]$ a $\hat{X}[k]$ je přirozeným logaritmem $X[k]$. Z hlediska popisu řeči bude pro nás nejdůležitější reálné kepstrum řeči $c[n]$, které vypočítáme podle vztahu (3.13): [13]

$$
c[n] = \mathrm{Re}\left\{\frac{1}{N}\sum_{k=0}^{N-1} \ln|X[k]| \cdot \mathrm{e}^{\mathrm{j}2\pi\frac{kn}{N}}\right\}, \tag{3.13}
$$

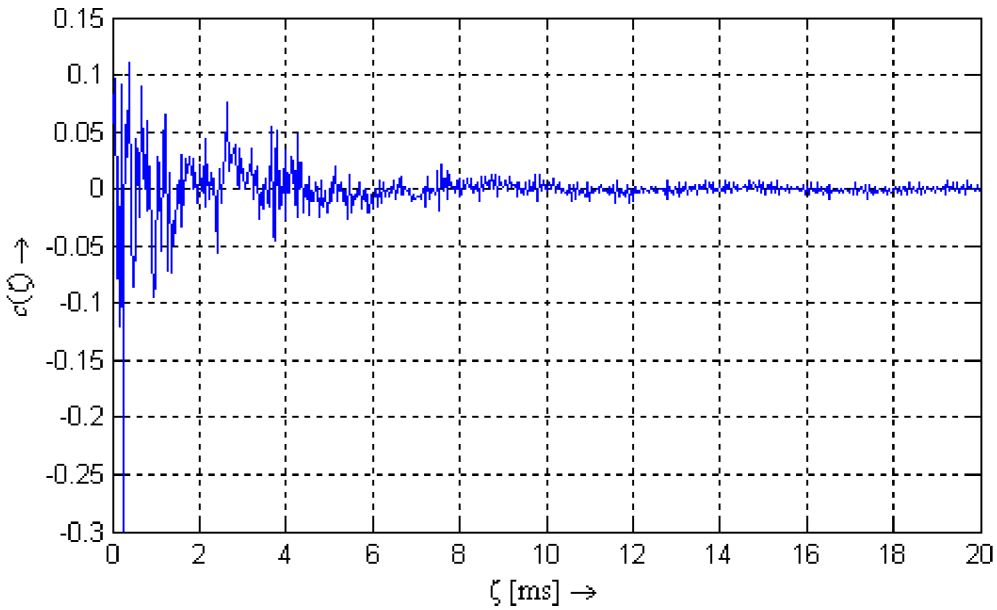
kde N je délka diskrétního signálu $x[n]$ a $k = 0, 1, 2, ..., N-1$. Toto reálné kepstrum má impulzní odezvu hlasového traktu soustředěnu kolem $n = 0$. Vynásobíme-li kepstrum vhodným kepstrálním oknem, vybereme ze signálu jen odpovídající část a při dopředné diskrétní Fourierovy transformaci a aplikaci inverzní funkce k logaritmu obdržíme modulovou kmitočtovou charakteristiku hlasového traktu. Kepstrální okno $l[n]$ by mělo tyto hodnoty: [13]

$$
l[n] = \begin{cases} 1 & \text{pro } n < n_0, \\ 0 & \text{pro } n \geq n_0, \end{cases} \tag{3.14}
$$

kde n_0 je vybráno tak, aby bylo menší než základní perioda hlasivkového tónu. Při vzorkovacím kmitočtu řečového signálu $f_{\mathrm{vz}} = 8\,\mathrm{kHz}$ se pro popis hlasového traktu volí $n_0 = 27$, pro vzorkovací kmitočet $f_{\mathrm{vz}} = 16\,\mathrm{kHz}$ se pak volí $n_0 = 53$ [21]. Budeme-li chtít naopak vybrat složky buzení, bude kepstrální okno definováno vztahem: [13]

$$l[n] = \begin{cases} 0 & \text{pro } n < n_0, \\ 1 & \text{pro } n \geq n_0. \end{cases} \tag{3.15}$$

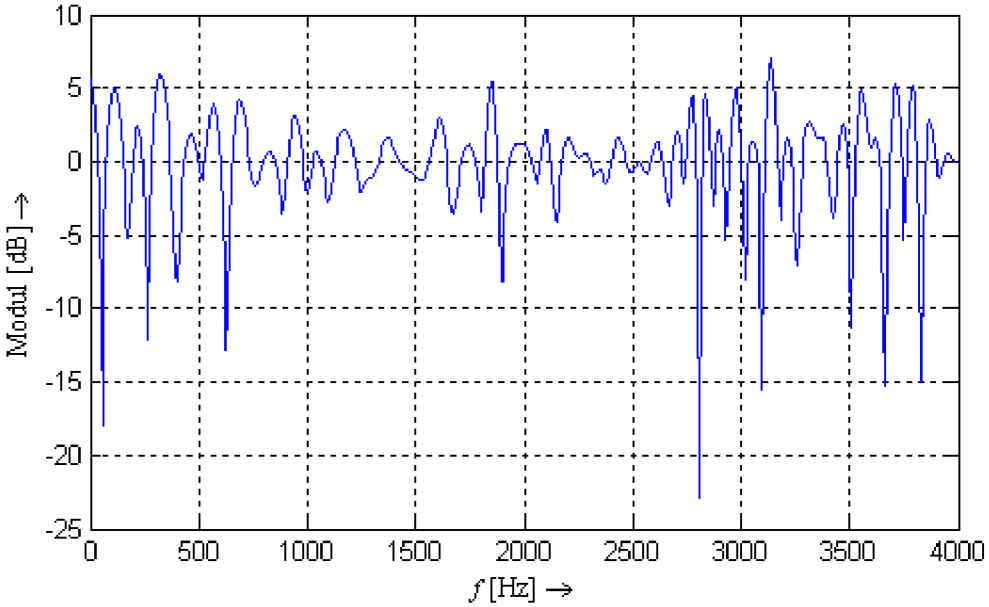
Příklad reálného kepstra úseku hlásky „á“ je na obr. 3.8, zde si můžeme všimnout, že na horizontální osu nevynášíme čas ale kvefrenci ζ. Spektrum budicího signálu úseku této hlásky je na obr. 3.9 a modulová kmitočtová charakteristika hlasového traktu modelovaná pomocí kepstra na obr. 3.10, kde je taktéž zobrazeno spektrum a LPC spektrum hlásky „á“. Na obr. 3.10 je také dobře vidět, že díky kepstru můžeme zvýraznit i antiformanty, což je důsledek toho, že filtr charakterizující hlasový trakt obsahuje póly a oproti LPC filtru také nulové body. Je zde využíván autoregresivní náhodný proces s klouzavým průměrem (ARMA) [16].



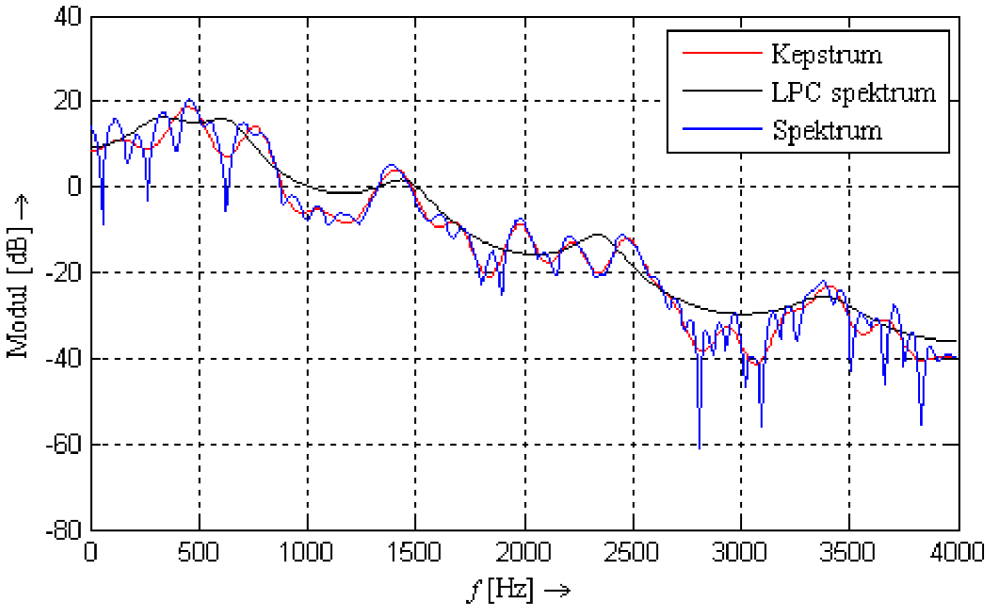
Obr. 3.8: Reálné kepstrum úseku hlásky „á“ ($f_{\mathrm{vz}} = 8\,\mathrm{kHz}$)

3.4 Převod hlásek na řečové jednotky

Výstupní tok z bloku modelování prozodie dále pokračuje do bloku převodu hlásek na řečové jednotky. Tyto jednotky mohou vznikat opět několika způsoby. Např. při



Obr. 3.9: Spektrum budicího signálu úseku hlásky „á“ ($f_{\mathrm{vz}} = 8\,\mathrm{kHz}$)



Obr. 3.10: Modulová kmitočtová charakteristika hlasového traktu modelovaná pomocí kepstra, spektrum a LPC spektrum úseku hlásky „á“ ($f_{\mathrm{vz}} = 8\,\mathrm{kHz}$)

formantové syntéze modelujeme hlasový trakt na základě jeho mechanických parametrů, které se pro jednotlivé hlásky a jejich prozodii mění. Vzniklý řečový úsek pak dále pokračuje do bloku syntézy řeči, kde se řetězí s ostatními úseky, které dohromady vytváří syntetický řečový signál. Ovšem jak už bylo dříve zmíněno, dnes se nejvíce využívá konkatenační syntéza, která pracuje na jiném principu.

Jestliže využíváme v systému TTS konkatenační syntézu, musíme si v určité databázi uchovávat jednotlivé řečové jednotky (segmenty). Tuto databázi budeme dále označovat jako inventář řečových jednotek. Jakmile máme tento inventář vytvořený, můžeme pak na základě vstupního textu vybírat jednotlivé řečové jednotky, upravovat jejich prozodii a posílat je dál k samotné syntéze. Samotnou konkatenační syntézou se budeme dále podrobněji zabývat při popisu bloku syntézy, nyní si podrobněji popíšeme volbu řečových jednotek.

3.4.1 Volba řečových jednotek

Volba úseků řeči, ze kterých se výsledný řečový signál bude skládat, je velmi důležitým postupem, který pak nakonec bude ovlivňovat hlavně kvalitu a přirozenost. Otázka však je, jak dlouhé úseky je třeba volit. Vlivem koartikulace může totiž při řetězení dojít k spektrálním či prozodickým nespojitostem. Při spektrální nespojitosti dochází k tomu, že si neodpovídají hodnoty jednotlivých formantů na hranicích úseků řeči. U prozodické nespojitosti pak dochází hlavně k tomu, že si na hranicích neodpovídají hodnoty základního tónu F_0. Bylo by potřeba tuto volbu optimalizovat, z tohoto důvodu nás budou zajímat čtyři důležitá kritéria:

- Maximální pokrytí koartikulačních jevů. Aby byla výsledná syntetická řeč co nejvíce přirozená, bylo by potřeba v ní postihnout všechny koartikulační jevy. Z tohoto hlediska je nejlepší, aby byly řečové úseky pokud možno co nejdelší.

- Bezproblémové řetězení. Aby nedocházelo ke spektrálním či prozodickým nespojitostem, bylo by opět vhodné volit řečové úseky co nejdelší, jelikož při samotném řetězení pak bude vznikat jen málo bodů (míst) s těmito nespojitostmi. Ideální by tedy bylo volit za řečové úseky celá slova, věty či fráze. K odstranění prozodických nespojitostí bychom pak využívali stejná slova, která by se lišila jen v suprasegmentálních rysech.

- Zobecnitelnost. Bylo by dobré, aby byl TTS systém schopen zpracovat jakýkoliv vstupní text. To ovšem může klást velké paměťové nároky na systém, jelikož při volení slov či vět jako řečových jednotek vzrůstá celkový počet těchto segmentů.

- Velikost databáze řečových jednotek. Aby neměl TTS systém velké nároky na paměť, je potřeba, aby byl inventář řečových jednotek co nejmenší.

Když si shrneme všechna kritéria, obdržíme dva požadavky. Potřebujeme totiž systém, který by pokud možno pracoval s co nejdelšími řečovými jednotkami, abychom vystihli všechny koartikulační jevy a co nejvíce snížili počet míst řetězení, na druhou stranu však potřebujeme, aby byl inventář těchto jednotek co nejmenší. Ovšem tím bychom porušili kritérium zobecnitelnosti, jelikož by bylo jen otázkou času, než by na vstup TTS systému přišel textový řetězec, ke kterému by v inventáři neexistovala příslušná řečová jednotka. Proto musíme při návrhu vždy volit určitý kompromis, který se bude také odvíjet od aplikace příslušného TTS systému. TTS systém, u kterého se požaduje, aby pracoval jen s omezeným slovníkem, může ve svém inventáři uchovávat delší řečové jednotky. Typickým příkladem mohou být TTS systémy, které se používají na nádražích, letištích nebo v městské hromadné dopravě. Např. struktura věty v tramvajích DPMB je: „Příští zastávka ____ . Prosíme pozor, zastávka leží v tarifní zóně ____ .“ Do volných míst se pak může například doplnit „Husitská“ a „101“. Naopak u systémů, které by měly zpracovat jakýkoliv vstupní text, volíme řečové jednotky co nejmenší. Ovšem výsledná řeč by měla být stále přirozená a srozumitelná.

Různé TTS systémy tedy používají různé řečové jednotky, popř. jejich kombinace. Níže si uvedeme ty nejpoužívanější z nich.

- Věty, fráze, slova. Jak bylo již zmíněno, tyto řečové jednotky co nejvíce vystihují koartikulační jevy a syntéza využívající tyto úseky je velmi přirozená. Na druhou stranu je však nemožné využívat tyto jednotky v univerzálním TTS systému.

- Slabiky. Tyto řečové jednotky stále zachovávají koartikulaci uvnitř slabik, ovšem databáze obsahující všechny slabiky v řeči by byla stále velmi obsáhlá.

- Fonémy. Jsou to nejmenší jednotky řeči, které rozlišují slova. Fonémů je pro češtinu 40 a tak by se zdálo, že volba těchto jednotek je, z hlediska kapacity inventáře, velmi výhodná [6]. Ovšem fonémy postrádají koartikulační jevy a řeč složená ze samostatných fonémů není dosti srozumitelná.

- Difóny. Jsou to řečové úseky, které začínají v polovině předešlé hlásky a končí v polovině hlásky následující. Tyto řečové jednotky zachovávají přechody mezi jednotlivými hláskami, a tím i částečně zachovávají koartikulační jevy. Jestliže bývá v jazyce počet hlásek N, potom počet difónů je přibližně roven N^2,

přičemž ne všechny difóny se v jazyce vyskytují [13]. Difóny se staly velmi oblíbenými řečovými úseky, které se používají v mnoha TTS systémech.

- Trifóny. Při řetězení difónů může v některých případech dojít k velkým spektrálním nespojitostem. To se dá vylepšit vhodným doplněním inventáře o trifóny. Pro trifón se v různých literaturách udávají různé definice, my budeme dále trifón uvažovat jako řečový úsek začínající v polovině jedné hlásky a pokračující přes druhou hlásku až do poloviny hlásky třetí. Jestliže je v jazyce počet hlásek N, potom počet trifónů je přibližně roven N^3, ovšem v praxi se používají jen některé [13].

3.4.2 Inventář řečových jednotek

Inventář řečových jednotek je databáze (banka) všech řečových úseků, ze kterých se může výsledný řečový signál složit. Tato databáze uchovává jejich realizace a popř. další informace, které daný úsek popisují (např. informace o prozodii). Abychom mohli inventář vytvořit, potřebujeme k tomu vhodný řečový korpus, ze kterého bychom segmentací dostali všechny požadované řečové jednotky.

Řečový korpus obsahuje vhodně zvolené nahrávky lidské řeči. Jestliže bude následující segmentace prováděna manuálně, snažíme se o to, aby byl řečový korpus co nejmenší, přičemž musí obsahovat všechny výskyty požadovaných řečových jednotek. Naopak při automatické segmentaci volíme korpus dostatečně velký tak, aby se v něm vyskytovali realizace řečových jednotek i s různými prozodickými rysy. Z tohoto hlediska můžeme korpusy dělit na:

- foneticky vyvážené řečové korpusy — četnost řečových jednotek v korpusu je stejná jako četnost těchto jednotek v přirozené řeči
- foneticky bohaté řečové korpusy — jednotlivé fonetické jednotky se v korpusu vyskytují stejně často

Namluvené nahrávky by měli být kvalitní, nejlépe nahrané ve zvukově izolované místnosti. Řečník by měl mít během nahrávání neutrální postoj a neměl by pokud možno měnit své suprasegmentální rysy (základní tón, intenzitu a tempo). Někdy je potřeba nahrát korpus citově zabarvený, řeč by však měla být stále stejně přirozená. Nahrávky řečníka většinou bývají ve formě různých vět či slov.

Během segmentace se snažíme odhadnout hranice akustických realizací jednotlivých řečových jednotek. To může být při manuálním postupu velmi pracný úkol,

navíc může být hranice odhadnuta s drobnou chybou, která se pak projeví ve výsledné syntetické řeči. Manuálním segmentacím řečových jednotek se věnují různí odborníci, přičemž i hranice udána dvěma lidmi se může lišit. Jelikož se v poslední době délka korpusů stále zvětšuje, abychom vystihli různé prozodické realizace řečových úseků, začali se používat metody automatické segmentace a tvorby jednotek založené např. na skrytých Markovových modelech (HMM), nebo na technice dynamického borcení časové osy. Tyto automatické metody jsou více popsány v literatuře [13].

3.4.3 Řečové jednotky v kmitočtové oblasti

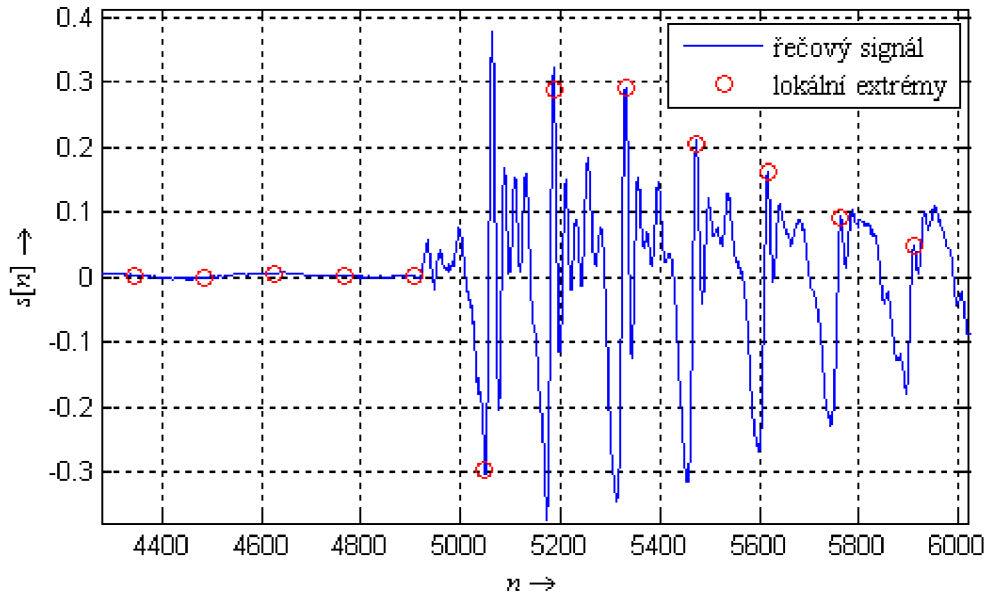
Ať už je řečový korpus jakkoliv obsáhlý, nikdy nedokáže pokrýt všechny suprasegmentální rysy řeči. Navíc při konkatenační syntéze může i při tak velkých korpusech docházet ke spektrálním nespojitostem. Jestliže uchováváme řečové jednotky v paměti jako klasické digitalizované vzorky řeči, je úprava jednotlivých suprasegmentálních rysů obtížná. Při požadavcích na změnu prozodie v TTS systémech se proto začala více uplatňovat syntéza řečového signálu v kmitočtové oblasti. V tomto případě většinou v inventáři neuchováváme samotné vzorky řeči, ale spíše koeficienty filtrů, díky kterým můžeme modelovat modulovou kmitočtovou charakteristiku hlasového traktu. Pro naši práci pak budou nejdůležitější koeficienty LPC syntetizujícího filtru, popř. vzorky reálného kepstra v okolí $n = 0$. Jak např. z lineární predikční analýzy řečového signálu vyplývá, dojde tím i k jisté kompresi a ke snížení nároků na paměť inventáře řečových jednotek.

V obou případech (jak u LPC kódování, tak u kepstrální analýzy) musíme jednotlivé úseky řeči nejdříve rozložit na menší části (rámce), abychom zajistili určitou stacionaritu řečového signálu. Teprve potom můžeme tyto rámce podrobit daným analýzám. Obecně se používají dvě metody segmentace řečového signálu na rámce: [20]

- pitch asynchronní segmentace — délka rámce je pevná, nezávislá na základní periodě
- pitch synchronní segmentace — délka rámce je variabilní, závislá na základní periodě

Pro naši další práci budeme uvažovat segmentaci pitch synchronní. Délka rámce N_{fr} závisí především na základní periodě řeči T_0, ale také na tempu řeči. Obecně by mělo platit, aby doba trvání rámce postihla alespoň dvě základní periody T_0. Jestliže je délka řečového signálu N, pak počet překrývajících rámců N_{frms}, pro dva různé

řečové signály o stejné délce N bude tedy různý. Metody pro určování délek rámců mohou být různé, my budeme vycházet z metody popsané v literatuře [4]. Tato metoda hledá v časové oblasti lokální extrémy v řečovém signálu. Pomocí těchto lokálních extrémů můžeme odhadnout periodu základního tónu, a tím i délku rámce, jelikož víme, že budou v rámci obsaženy tyto periody dvě. Na obr. 3.11 je příklad použití této metody na úsek řečového signálu. Jak je z obrázku dobře vidět, tato metoda také docela dobře odhaduje délky neznělých úseků.



Obr. 3.11: Lokální extrémy v řečovém signálu ($f_{\mathrm{vz}} = 8\,\mathrm{kHz}$)

Při segmentaci budeme dále určovat, zdali je daný úsek (rámec) řeči znělý nebo neznělý. Tuto informaci budeme pak ukládat v inventáři řečových jednotek, abychom při zpětné syntéze věděli, zdali máme provézt buzení šumem nebo periodickým signálem. Pro určení znělosti úseku budeme používat dvě metody [20]. Obecně platí, že znělé úseky řeči mají větší energii a menší průchod nulovou úrovní, než úseky neznělé. Energie E diskrétního řečového signálu na jednom segmentu řeči délce N_{fr} je dána vztahem: [16]

$$E = \sum_{n=1}^{N_{\mathrm{fr}}} |s[n]|^2 \,. \tag{3.16}$$

Počet průchodů nulovou úrovní Z definujeme: [20]

$$Z = \frac{1}{2} \sum_{n=1}^{N_{\mathrm{fr}}-1} |\mathrm{sign}(s[n]) - \mathrm{sign}(s[n+1])| \,, \tag{3.17}$$

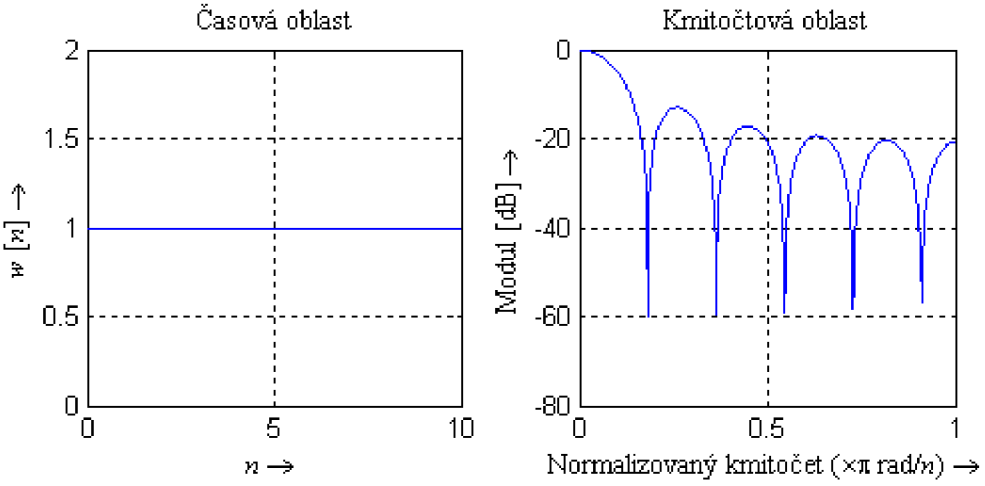
$$\text{sign}(s[n]) = \begin{cases} +1 & \text{když } s[n] > 0 \text{ nebo } s[n] = 0 \wedge s[n-1] > 0, \\ -1 & \text{když } s[n] < 0 \text{ nebo } s[n] = 0 \wedge s[n-1] < 0. \end{cases} \tag{3.18}$$

Pro daný řečový korpus, ze kterého se úseky vybírají, je pak potřeba určit prahovou úroveň E a Z. Jestliže bude energie řečového úseku nižší než zvolená prahová úroveň, pak budeme tento úsek považovat za neznělý. Jestliže bude energie větší než prahová úroveň, pak spočítáme počet průchodů nulovou úrovní. Při nižším počtu než je prahová úroveň Z budeme úsek považovat za znělý a při větším počtu za neznělý.

Při vybírání jednotlivých rámců budeme řeč vždy násobit určitým váhovacím oknem $w[n]$. Jako nejjednodušší se jeví násobení pravoúhlým oknem, které je popsáné rovnicí: [33]

$$w[n] = \begin{cases} 1 & \text{pro } 0 \leq n \leq N-1, \\ 0 & \text{jinde.} \end{cases} \tag{3.19}$$

Zobrazení pravoúhlého okna v časové a kmitočtové oblasti je na obr. 3.12.



Obr. 3.12: Zobrazení pravoúhlého okna v časové a kmitočtové oblasti

Při násobení signálu váhovacím oknem v časové oblasti dochází ke konvoluci příslušných obrazů v oblasti kmitočtové [30]. Tím dojde ke zkreslení informace.

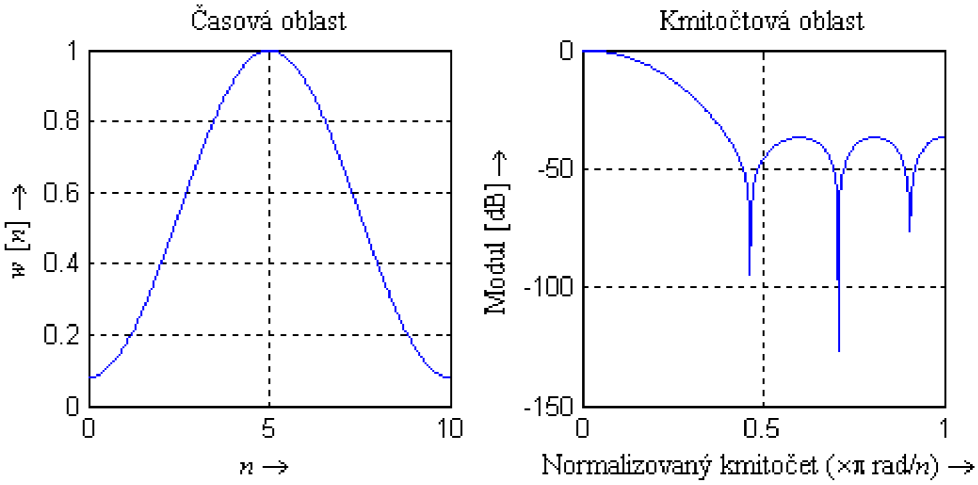
$$s'[n] = s[n] \cdot w[n], \tag{3.20}$$

$$S'[k] = S[k] * W[k], \tag{3.21}$$

kde $s'[n]$ je požadovaný rámec a $S'[k]$ jeho spektrum. Toto zkreslení můžeme částečně omezit volbou vhodnějšího okna než je pravoúhlé. Pro analýzu řečového signálu se nejčastěji používá Hammingovo okno, které je popsané následujícím vztahem: [33]

$$w[n] = \begin{cases} 0{,}53836 - 0{,}46164 \cdot \cos\left(\frac{2\pi n}{N-1}\right) & \text{pro } 0 \leq n \leq N-1, \\ 0 & \text{jinde.} \end{cases} \tag{3.22}$$

Zobrazení Hammingova okna v časové a kmitočtové oblasti je na obr. 3.13.



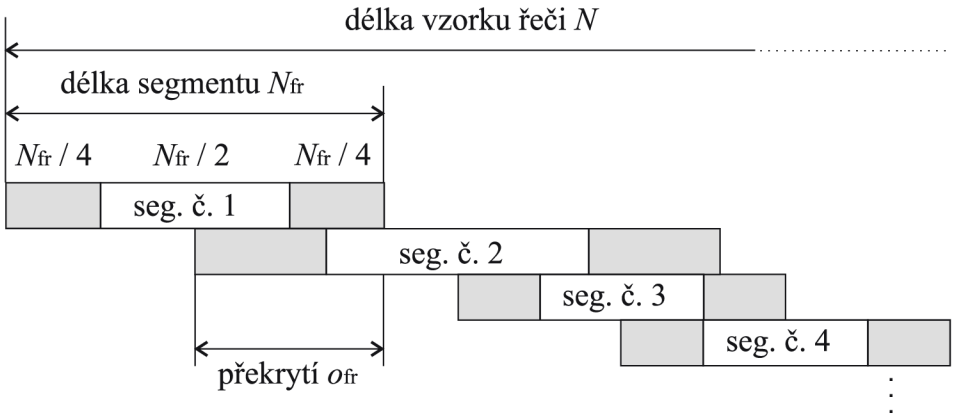
Obr. 3.13: Zobrazení Hammingova okna v časové a kmitočtové oblasti

Díky násobení Hammingovým oknem dojde k útlumu signálu na obou okrajích jednotlivých rámců. Při zpětném řetězení těchto rámců pak nedochází k tak znatelným skokovým změnám parametrů, které dva sousedící rámce popisují.

Tuto skokovou změnu můžeme také částečně eliminovat vhodným překrýváním během segmentace. Překrývání však nemůže být zase příliš dlouhé, jelikož se tím snižuje rychlost změny hlasového traktu. Většinou se volí délka překrytí o_{fr} jako polovina délky rámce, tedy $o_{\text{fr}} = N_{\text{fr}}/2$. Způsob segmentace znázorňuje obr. 3.14.

Změna periody základního tónu T_0

Při zpětné syntéze může nastat případ, že budeme měnit periodu základního tónu T_0, přičemž nebude stejná jako perioda, ze které jsme vycházeli při segmentaci. Uvedli jsme si, že budeme chtít, aby každý segment (jak při segmentaci, tak při syntéze) obsáhl alespoň dvě základní periody T_0. Označme si periodu základního tónu při segmentaci $T_{0\text{se}}$ a periodu základního tónu při syntéze $T_{0\text{sy}}$. Mohou nastat tři případy. Jestliže $T_{0\text{se}} = T_{0\text{sy}}$ pak provedeme syntézu standardním způsobem. Jestliže



Obr. 3.14: Způsob segmentace řečového signálu

$T_{0\text{se}} > T_{0\text{sy}}$, pak pro syntézu k-krát použijeme ten samý rámec (resp. parametry popisující tento rámec). Koeficient k, v tomto případě, vypočítáme dle vztahu:

$$k = \left\lceil \frac{T_{0\text{se}}}{T_{0\text{sy}}} \right\rceil , \tag{3.23}$$

kde operace $\lceil . \rceil$ značí zaokrouhlení hodnoty uvnitř závorek nahoru. Jestliže $T_{0\text{se}} < T_{0\text{sy}}$, pak provedeme syntézu z rámce a pro další syntézu vybereme z inventáře rámec v k-tém pořadí po předešlém použitém rámci. Koeficient k, v tomto případě, vypočítáme dle vztahu:

$$k = \left\lfloor \frac{T_{0\text{sy}}}{T_{0\text{se}}} \right\rfloor . \tag{3.24}$$

kde operace $\lfloor . \rfloor$ značí zaokrouhlení hodnoty uvnitř závorek dolů.

Změna trvání řeči

Při syntéze řečových jednotek se postupuje tak, že vždy s jednou sadou parametrů provedeme syntézu jednoho úseku (jedné periody). Při změně trvání řeči[2] budeme syntézu provádět podobným způsobem, ovšem s některou sadou parametrů již neprovedeme syntézu jen jednoho úseku, ale úseků dvou, přičemž počet vzorků v druhém úseku bude stejný jako počet vzorků v úseku předešlém. To, se kterou sadou parametrů budeme provádět vícenásobnou syntézu, určí procentní navýšení doby trvání nové řečové jednotky oproti době trvání staré řečové jednotky. Označme si m jako koeficient opakování sady parametrů při změně trvání řeči. Tento koeficient

[2] V našem případě se budeme zajímat o prodloužení trvání.

nám bude udávat s kolikátou sadou parametrů v pořadí provedeme dvojnásobnou syntézu. Koeficient m vypočítáme následujícím vztahem:

$$m = \left\lfloor \frac{100}{\text{procentní navýšení}} \right\rfloor . \tag{3.25}$$

Pokud budeme přímo znát původní dobu trvání řečové jednotky t_p a novou dobu trvání řečové jednotky t_n, pak koeficient m vypočítáme podle vztahu:

$$m = \left\lfloor \frac{t_\mathrm{p}}{t_\mathrm{n} - t_\mathrm{p}} \right\rfloor . \tag{3.26}$$

Například bude-li doba trvání původní řečové jednotky 100 ms a nová doba trvání 117 ms (lyší se o 17 %), pak koeficient m vypočítáme jedním z následujících vztahů:

$$\begin{aligned} m &= \left\lfloor \frac{100}{117 - 100} \right\rfloor = 5, \\ m &= \left\lfloor \frac{100}{17[\%]} \right\rfloor = 5. \end{aligned}$$

V tomto případě budeme tedy s každou pátou sadou provádět opakovanou syntézu.

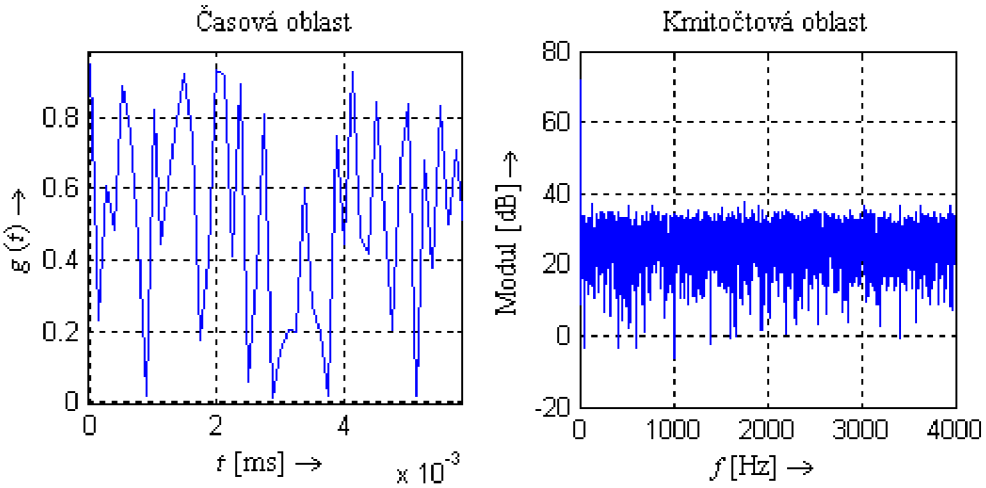
Změna intenzity řeči

Změnu úrovně řečového signálu jednoduše provedeme tak, že hodnotu vzorků vynásobíme nějakou konstantou. V případě zesílení bude konstanta větší než 1, v případě snížení úrovně tomu bude naopak. Konstanta může také pro různé vzorky nabývat různých hodnot, toho se využívá například při syntéze rozkazovacích vět, kde úroveň řečového signálu postupně roste.

Modelování budicího signálu

Jak už jsme si říkali dříve, budeme uvažovat dva druhy budicího signálu. Pro znělé úseky řeči se většinou jako budicí signál používá sled jednotkových impulzů, jejichž úroveň je v čase buď neproměnná, nebo se s časem snižuje. Perioda takto vytvořeného signálu odpovídá periodě základního tónu T_0. Příklad tohoto budicího signálu v časové a kmitočtové oblasti je na obr. 2.3, graf a) a b). Někdy se ještě mezi tyto jednotkové impulzy vkládá další posloupnost impulzů, jejichž úroveň je několikrát menší než úroveň impulzů základních, které se vyskytují s periodou T_0. Při buzení pouze jednotkovými impulzy totiž dochází k drobnému bzučení ve výsledné syntetické řeči, toto bzučení se dá částečně potlačit přidáním již zmíněné posloupnosti impulzů o nízké úrovni. Naopak pro neznělé úseky řeči se jako budicí signál používá bílý šum, který se vyznačuje tím, že je jeho obálka ve spektru rovnoběžná s kmitočtovou osou. V MATLABu můžeme generátor bílého šumu přibližně nahradit

generátorem náhodných čísel s rovnoměrným rozložením. Příklad takto vytvořeného šumu v časové a kmitočtové oblasti je na obr. 3.15.



Obr. 3.15: Bílý šum v časové a kmitočtové oblasti ($f_{\mathrm{vz}} = 8\,\mathrm{kHz}$)

Jestliže tedy budeme vědět, zdali je úsek řeči znělý či neznělý, budeme znát kmitočet základního tónu a modulovou kmitočtovou charakteristiku hlasového traktu, můžeme provést filtraci budicího signálu a výsledkem bude úsek signálu řečového. Každý úsek se bude dále řetězit s dalšími řečovými úseky, dokud nevznikne požadovaná řečová jednotka. Tuto jednotku pak ještě v případě potřeby převádíme do časové oblasti a posíláme dále do bloku samotné syntézy.

3.5 Syntéza výsledné řeči

Blok syntézy výsledné řeči je posledním důležitým blokem v TTS systému. Vstupují sem jednotlivé řečové jednotky a příslušné prozodické informace. Na základě těchto vstupů jsou pak jednotky řetězeny a výsledkem je syntetický řečový signál. Prozodické informace pak mohou ještě upravovat například hlasitost nebo trvání celé věty. Struktura nebo uspořádání tohoto bloku jsou opět závislé na zvolené metodě syntézy. Pro naše účely se budeme opět zabývat již zmíněnou konkatenační syntézou.

Konkatenační syntézu jsme si již popsali v bloku modelování prozodie a v bloku převodu hlásek na řečové jednotky. Jedná se vlastně o jakési řetězení řečových jednotek tak, aby dohromady daly požadovaný řečový signál. Mezi její hlavní výhody patří:

- Používání přirozených segmentů řeči. Konkatenační syntéza pracuje přímo s řečovými jednotkami, vyhýbáme se tím např. složitému modelování mechanický parametrů hlasového traktu atd. Navíc tím, že pracujeme přímo s jednotkami řeči, je výsledný syntetický signál přirozenější.

- Rychlý návrh syntetizéru. Oproti formantové syntéze je návrh TTS systému využívající konkatenační syntézu méně časově náročný. Většinou při návrhu nejdéle trvá samotný proces volby řečových jednotek.

- Vysoká kvalita syntetické řeči. Řeč vytvořená konkatenační syntézou se vyznačuje dobrou kvalitou a přirozeností. Navíc při dosti obsáhlém slovníku může být výsledná řeč téměř nerozeznatelná od přirozené řeči.

I když je to dnes nejpoužívanější metoda, má i své nevýhody. Mezi ně především patří:

- Výpočetní a paměťová náročnost. Při uchovávání velkých řečových korpusů a jednotek dochází k velkým nárokům na paměť. Navíc při vyhledávacích algoritmech vznikají i znatelné nároky na procesor, ovšem při dnešním vývoji počítačové techniky se tyto nároky stávají zanedbatelnými.

- Závislost na řečníkovi. I když jsme schopni změnou kmitočtu základního tónu měnit prozodické informace, a tím i částečně pohlaví řečníka, je výsledná řeč silně ovlivněna mluvčím, který řečový korpus namluvil.

- Omezený inventář řečových jednotek. Jestliže univerzální TTS systém používá jako řečové jednotky slova či věty, může se časem stát, že na vstupu se objeví takový textový řetězec, který nebude mít v inventáři řečových jednotek svůj ekvivalent.

- Místa řetězení. Samotná podstata konkatenační syntézy se může někdy stát nevýhodou. V místech řetězení řečových úseků totiž může docházet k různým prozodickým a spektrálním nespojitostem.

Výslednou syntetickou řeč umožňuje TTS systém většinou uložit v různých formátech jako např. MP3, WAV, WMA atd. Někdy bývají výstupy TTS systémů doplněny různými videosekvencemi, jako např. mluvící obličej.

3.6 Praktické použití TTS systémů

TTS systémy mají v praxi široké využití, navíc s rostoucí srozumitelností a přirozeností syntetické řeči tyto systémy v některých oblastech nahrazují samotné lidi. Dále si uvedeme nejdůležitější oblasti, ve kterých se TTS systémy používají:

- Automatické čtení předlohy. TTS systémy se s výhodou používají pro čtení z nějakých předloh, kterými mohou být e-maily, SMS zprávy, faxy, internetové stránky atd.

- Pomůcka pro handicapované lidi. Velmi velký význam mají TTS systémy pro lidi s poruchami řeči. Díky speciálně upraveným klávesnicím mohou zapisovat text, který je následně převeden na řeč, tyto systémy se také začínají implementovat do mobilních telefonů. Díky předčítání z předlohy jsou TTS systémy využívány i nevidomými lidmi. Kvalitní systémy by pak mohly být použity i v různých výukových systémech pro lidi s logopedickými vadami.

- Komunikace člověk-počítač. U různých dialogových systémů, kde člověk mluví s počítačem je kromě rozpoznávání řeči použita také syntéza. Toho se může využívat například v různých mobilních zařízeních nebo GPS systémech, kde člověk např. zadá slovně jméno požadované destinace a systém ho na toto místo slovně navádí.

- Výuka cizích jazyků. Zejména různí samouci využívají kvalitních TTS systémů ke správnému učení výslovnosti cizích slov. Tyto systémy také bývají součástí některých multimediálních slovníků.

4 NÁVRH TTS SYSTÉMU

V této části práce provedeme návrh jednoduchého TTS systému, který využívá lineární predikční a kepstrální syntézu. Návrh je proveden tak, aby bylo možné tento systém implementovat do grafického prostředí (GUIDE) programu MATLAB. Návrh se řídí teoretickým rozborem, který byl proveden v kapitole č. 3.

Navrhovaný TTS systém je univerzální, to znamená, že by měl převést jakýkoliv text napsaný v českém jazyce do řečového signálu. Je zde možnost volit řeč vyslovenou mužem, ženou nebo dítětem a také simulovat emoce řečníka. Uživatel si může sám vybrat, zdali se provede lineární predikční nebo kepstrální syntéza. Jako řečové jednotky použijeme jednotlivé hlásky, které se doplní některými dvojhláskami. Použití samotných hlásek není zcela ideální, jelikož nevystihneme koartikulaci, ovšem při použití jiných řečových jednotek, jako např. difónů či trifónů, bychom museli vytvořit rozsáhlý inventář, který by čítal až tisíce těchto řečových úseků, což by bylo časově velmi náročné. Proto inventář doplníme některými dvojhláskami, abychom vystihli alespoň tu koartikulaci, která je pro sluch nejvíce znatelná, např. přechod mezi hláskami o_u, a_u atd. Jako řečový korpus použijeme různé nahrávky zaznamenané ve zvukově izolované místnosti na Ústavu telekomunikací Fakulty elektrotechniky a komunikačních technologií Vysokého učení technického v Brně a také na Ústavu fotoniky a elektroniky Akademie věd ČR. Součásti TTS systému je také program, určený k vytváření vzorků řeči z korpusu a následnému uložení těchto vzorků do inventáře řečových jednotek. Každý tak může zvětšit inventář dle své libosti. Vstupní textový řetězec se nahrává buď z nějakého textového souboru (např. `*.txt`), nebo ručně vepisuje do příslušného okna. Výsledná syntetická řeč je navzorkována vzorkovacím kmitočtem $f_{vz} = 16\,\text{kHz}$ a každý vzorek je vyjádřen 16 bity. Mohli bychom použít také $f_{vz} = 8\,\text{kHz}$, ovšem řeč by zněla příliš uměle. Syntetickou řeč můžeme přehrát nebo uložit ve formátu `*.wav`.

Další práce se řídí blokovým schématem TTS systému na obr. 3.2. Každý blok je zvlášť navrhnut tak, aby dohromady s ostatními dával jeden celek. Nejdříve se však budeme věnovat pomocnému programu, který slouží k segmentaci a následné analýze řečových vzorků z nahraného korpusu.

4.1 Program analýzy

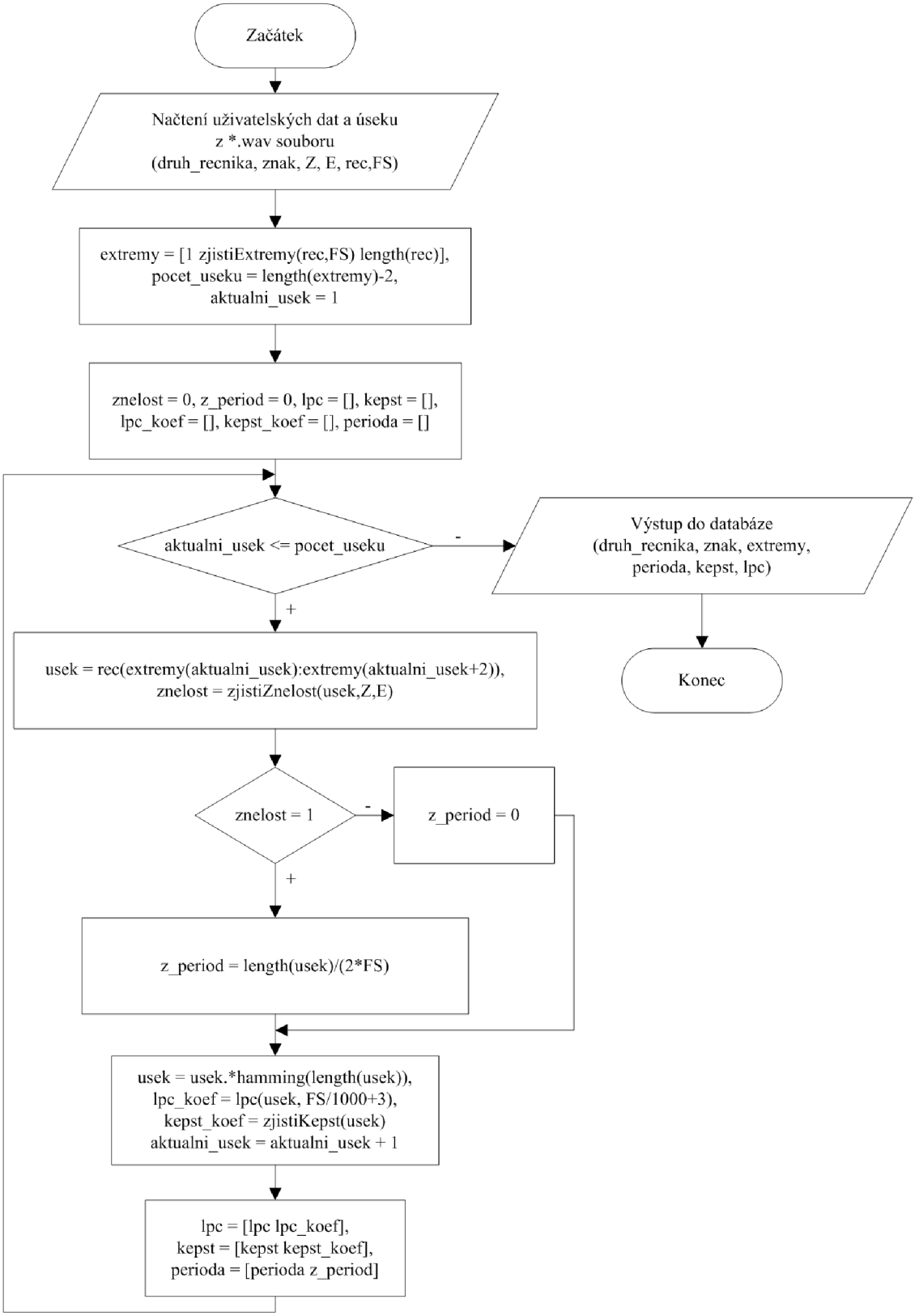
Tento program umožní nahrát řečový korpus, jehož vzorkovací kmitočet je $f_{vz} = 16\,\text{kHz}$ a formát `*.wav`. Načítaní zvukových souborů v tomto formátu se v prostředí MATLAB provádí funkcí `[Y,FS,NBITS] = wavread(FILE)`, kde pole `Y` je samotný

signál, `FS` hodnota vzorkovacího kmitočtu a `NBITS` počet bitů na jeden vzorek signálu. Tento signál je vykreslen v časové oblasti, přičemž je toto zobrazení doplněno také spektrogramem[1]. Pomocí tlačítek si v grafu vyznačíme požadovanou oblast, nad kterou chceme provádět následnou analýzu. U grafu jsou také pole, pro nastavení prahových úrovní E a Z, na základě kterých se rozhoduje, zdali se při segmentaci jedná o znělý či neznělý úsek. Dále budeme ještě volit druh řečníka (muž, žena, dítě) a hlásky či dvojhlásky, kterým označený úsek odpovídá. Algoritmus řízení pitch synchronní analýzy lze znázornit vývojovým diagramem na obr. 4.1.

Na začátku si do paměti nahrajeme označený úsek řečového signálu, informace o druhu řečníka, příslušný znak, vzorkovací kmitočet a hodnoty prahových úrovní E a Z. Pomocí funkce `zjistiExtremy(rec,FS)` zjistíme pozice lokálních extrémů v řečovém signálu. K těmto pozicím přidáme na začátek hodnotu 1, na konec číslo posledního vzorku řečového úseku a uložíme do pole `extremy`. Díky tomuto poli můžeme tedy vstupní řečový signál rozložit na několik úseků, jejichž počet udává proměnná `pocet_useku`. Dále budeme až po poslední úsek provádět segmentaci. Při každém cyklu si do pole `usek` nahrajeme dva úseky z řečového signálu (každý úsek reprezentuje jednu periodu základního tónu, proto vybíráme vždy dva), přičemž dodržujeme to, aby se dva sousední segmenty překrývaly. Pomocí funkce `zjistiZnelost(usek,Z,E)` zjistíme znělost segmentu. Vstupem do této funkce je samotný segment, a také prahové úrovně. Jestliže je úsek znělý, vypočítáme přibližně hodnotu periody základního tónu a uložíme do proměnné `z_period`. Jestliže je daný úsek neznělý, uložíme do proměnné `z_period` nulu (díky tomu pak při zpětné syntéze poznáme, že se jedná o úsek neznělý). Dále zjistíme pro daný úsek pomocí funkcí `lpc(usek,FS/1000+3)` a `zjistiKepst(usek)` hodnoty LP koeficientů a kepstrálních koeficientů. Počet těchto koeficientů je vždy stejný, pro $f_{\mathrm{vz}} = 16\,\mathrm{kHz}$ je počet LP koeficientů 20, navíc uchováme zesílení G^2. Kepstrálních koeficientů je pro každý úsek vždy 52. Jakmile budeme znát LP koeficienty, kepstrální koeficienty a hodnotu základní periody, přidáme vše do příslušných polí a vrátíme se na začátek cyklu. Po posledním cyklu pak nahrajeme zjištěná data do inventáře (databáze). Databáze má v MATLABu charakter buňky, která má na každém řádku odpovídající počet matic, uchovávající data. Použití buňky nám velmi usnadní práci s databází. Lineární predikční a kepstrální koeficienty jsou nahrané v samostatných maticích, ke kterým je snadný přístup. Takto vzniklá databáze je setříděna podle názvu znaků fonetické abecedy a dále podle mluvčího. Databázi uložíme ve formátu `databaze.mat`, což je druh souboru, který může program MATLAB zpětně nahrát a uložit tak databázi do paměti.

[1] Spektrogram zde slouží k přesnějšímu určení hranic řečových jednotek.

[2] Toto zesílení budeme pak při filtraci dosazovat do čitatele přenosové funkce.



Obr. 4.1: Algoritmus řízení pitch synchronní analýzy

Součástí programu analýzy je také stručný seznam všech uložených řečových jednotek. Navíc je možné tyto jednotky z inventáře mazat.

4.2 Analýza vstupního textu

V našem programu omezíme analýzu vstupního textu jen na některá pravidla uvedená v teoretickém rozboru TTS systémů. Uvažujme, že se text do programu přímo vepisuje, nebo načítá z předlohy ve formátu `*.txt`, to znamená, že jej nebudeme dále nějak měnit či ovlivňovat jeho tok. Zaměříme se pouze na normalizaci textu, to znamená, že nebudeme provádět kontextové, morfologické a syntaktické analýzy, které by vyžadovaly další složité algoritmy. Z toho ovšem vyplývá, že může dojít ke špatnému překladu, např. u pořadových čísel.

Normalizaci textu si rozdělíme do několika kroků. Nejdříve nahradíme některé nejpoužívanější fyzikální jednotky. Tyto jednotky se vyskytují vždy za čísly, proto detekujeme místa, kde se vyskytuje číslo, mezera a poté jednotka, nebo místa, kde detekujeme pouze číslo a hned za ním jednotku. Při nahrazování fyzikálních jednotek také záleží na velikosti uvozujícího čísla. Je-li číslo rovno 1, použijeme název zkratky v prvním pádě jednotného čísla. Je-li číslo rovno 2, 3, nebo 4, dosadíme název zkratky v prvním pádě množného čísla. Má-li číslo jinou hodnotu, dosadíme název v druhém pádě množného čísla. Stejná pravidla platí i pro záporná čísla. Potom např. 14 mA přepíšeme jako „14 miliampérů“ a −3 mA přepíšeme jako „−3 miliampéry“.

Dále nahradíme slovy samotná čísla. Opět se omezíme na ta nejvíce používaná, jako jsou čísla 0–20 s krokem 1, 20–100 s krokem 10 a nakonec 100–1000 s krokem 100, přičemž nahrazujeme čísla od nejvyšších hodnot, jelikož složená čísla, která nejsou ve slovníku, nahradíme pak alespoň jednotlivými číslicemi. Po číslech nahradíme některé zkratky jako „např.“, „aj.“, „atd.“. Ovšem musíme dávat pozor na to, aby byla zkratka osamocena a nebyla například na konci slova. Například větu „Kap, kap.“ by špatný algoritmus mohl přepsat jako „Kap ka podobně.“ Proto hledáme před zkratkou symbol, který patří do abecedy. Podobně také nahrazujeme zkratky různých titulů, jako například „Ing.“ nebo „Doc.“. V dalším kroku nahradíme některé zkratky organizací, např. OSN, NATO nebo FEKT. Nakonec nahradíme slovně některé další znaky, např. „%“, „@“, „+“.

Ještě před normalizací si však rozdělíme text na jednoduché věty (větné úseky).

To bychom mohli udělat i po normalizaci, ovšem během normalizace provedeme změnu všech velkých znaků na malé, což by ztížilo rozpoznávání konce vět. K označení větných úseků využijeme znak „#“. Ovšem může se stát, že bude text tento znak již obsahovat, proto tyto znaky nejdříve z textu odstraníme. Konec věty nebo části souvětí detekujeme na základě interpunkčních znamének uvedených v tab. 3.3. Za každé toto znaménko vložíme jeden znak „#“. Jestliže na konci textu není ani jedno z interpunkčních znamének, přidáme sem tečku a znak „#“. Speciálně pro detekci věty ukončené tečkou musíme vytvořit zvláštní algoritmus, jelikož tečka nemusí značit pouze konec věty, ale také může značit pořadovou číslici. Proto se vždy zajímáme o další dva znaky za tečkou. Jestliže za ní následuje mezera a velké písmeno, považujeme toto místo za konec věty. Podobný problém by mohl nastat při detekování čárky oddělující desetinou část čísla od celé. Problém ošetříme tím, že budeme ihned za čárkou hledat mezeru.

4.3 Fonetická transkripce

Při fonetické transkripci použijeme slovník, pomocí něhož upravíme některá slova tak, abychom poté v další fázi mohli přepisovat již samotné znaky. Tento slovník není pro český jazyk příliš obsáhlý, pomůckou pro nás může být slovník, použitý v českém TTS systému EPOS. Ten je ve verzi EPOS 2.5.37 uložen pod touto cestou: `epos-2.5.37\cfg\lng\czech\main.dic` [24]. Části slov, jsou v tomto slovníku rozděleny na čtyři skupiny. První skupina obsahuje části slov, kterými slova začínají, další skupina obsahuje části slov, kterými slova končí, třetí skupina obsahuje části slov nebo slova, která se mohou vyskytovat kdekoliv v textu a poslední skupina obsahuje samostatná slova, která se mohou nacházet kdekoliv v textu. Pro každou skupinu tak musíme vytvořit vlastní pravidlo náhrady.

Jakmile máme všechna slova patřičně upravená, provedeme transkripci založenou na fonetické abecedě SAMPA. Vyjdeme ze slovníku uvedeného v tab. 3.1, který dle případných potřeb ještě upravíme (přidáme některé dvojhlásky, popř. změníme některé znaky abecedy). Z toho, že používáme fonetickou abecedu SAMPA také vyplývá, že při nahrávání řečových jednotek do inventáře uvedeme jednotlivé znaky ve formátu SAMPA. Takto upravený text již nese některé prozodické informace.

4.4 Modelování prozodie a syntéza řeči

Na základě přídavných prozodických informací můžeme přejít k samotné syntéze řečové jednotky. Uživatel si může zvolit, zdali bude řečník muž, žena nebo dítě,

a také má na výběr z 8 postojů, jejichž prozodické charakteristiky jsou pro český jazyk uvedeny v tab. 3.5. Samozřejmě, že tyto hodnoty jsou jen orientační, proto je musíme pro každý korpus a druh řečníka pozměnit. Tempo se však neodvíjí od počtu slabik za sekundu, ale od tempa při neutrálním postoji. Udává se v procentech, což vyjadřuje, o kolik procent je promluva rychlejší nebo pomalejší oproti neutrální promluvě. Stejným způsobem volíme intenzitu, s jakou je řečový signál vysloven.

Samotnou syntézu provedeme pro každý větný úsek zakončený znakem „#“ zvlášť. Nejdřív na základě interpunkčního znaménka zjistíme, o jakou intonaci se jedná (stoupavá, klesavá) a poté si vymezíme na základě upravené tabulky 3.5 počáteční a koncovou hodnotu periody základního tónu. Mezi tyto dvě hodnoty vložíme další hodnoty tak, aby celkový počet hodnot odpovídal celkovému počtu segmentů ve větě, a hodnoty postupně přecházely od nejmenší k největší, popř. naopak (záleží na intonaci). Dále pak můžeme provést syntézu řečových jednotek a ty poskládat do věty. Algoritmus řízení syntézy věty ukončené znakem „#“ lze znázornit vývojovým diagramem na obr. 4.2.

Nejdříve si do paměti načteme řetězec, informace o druhu řečníka, o jeho postoji a nakonec druh syntézy, kterou si uživatel zvolil (lineární predikční nebo kepstrální). Poté si pomocí funkce `zjistiPeriody(retezec,druh_recnika,emoce)` zjistíme hodnoty period základních tónů pro jednotlivé segmenty. Tato funkce na základě posledního znaku v řetězci (znak „#“ již neuvažujeme) zjistí intonaci a již zmíněným způsobem vypočítá požadované hodnoty. Dále si vytvoříme proměnné `prvni_znak` a `druhy_znak`, pomocí kterých se budeme pohybovat v textovém řetězci. Dokud je hodnota proměnné `druhy_znak` menší nebo rovna délce řetězce, bude probíhat syntéza. Při pohybu v řetězci, se pro každý úsek textu testuje, zdali pro něj máme v inventáři uloženou řečovou jednotku. Jestliže hned první znak z úseku textu v inventáři není, zjistíme, zdali se nejedná o interpunkční znaménko. Jestliže ano, pak pomocí funkce `generujMezeru(znak)` vygenerujeme na základě tab. 3.3 příslušnou mezeru (posloupnost nul) a přidáme k výslednému řečovému signálu, který je uložen v poli `rec`. Jestliže tomu tak není, znak přeskočíme a pokračujeme od následujícího. Jakmile zjistíme nejdelší možný úsek textu, zastoupený v inventáři, uložíme jej do pole `znak`, se kterým budeme dále pracovat. Pomocí funkce `zjistiZBanky(znak,pohlavi,informace)`, budeme vyhledávat informace v našem inventáři. Proměnnými `znak` a `pohlavi` určíme řádek v matici, ze kterého budeme vyčítat informace dané polem `informace`. Nejdříve si do pole `perioda_p` nahrajeme hodnoty period základních tónů pro všechny segmenty daného znaku. Dále na základě uživatelské volby provedeme jednu ze dvou možných syntéz. V obou případech z inventáře vyčteme do matice příslušné koeficienty, na základě kterých provedeme

Obr. 4.2: Algoritmus řízení syntézy věty ukončené znakem „#“

buď pomocí funkce `generujLPC` nebo `generujKepst` syntézu segmentů, které budeme skládat do pole `rec`. Každému segmentu odpovídá jeden řádek koeficientů v této matici. Do obou funkcí vstupují příslušné koeficienty, pole period vyčtené z inventáře a pole period vrácené funkcí `zjistiPeriody`. U lineární predikční syntézy vstupuje do funkce `generujLPC` také zesílení G, které je vždy první hodnotou na řádku matice LP koeficientů. Jakmile provedeme syntézu celé věty, odešleme výsledný řečový signál k dalšímu zpracování.

4.5 Úprava a export řečového signálu

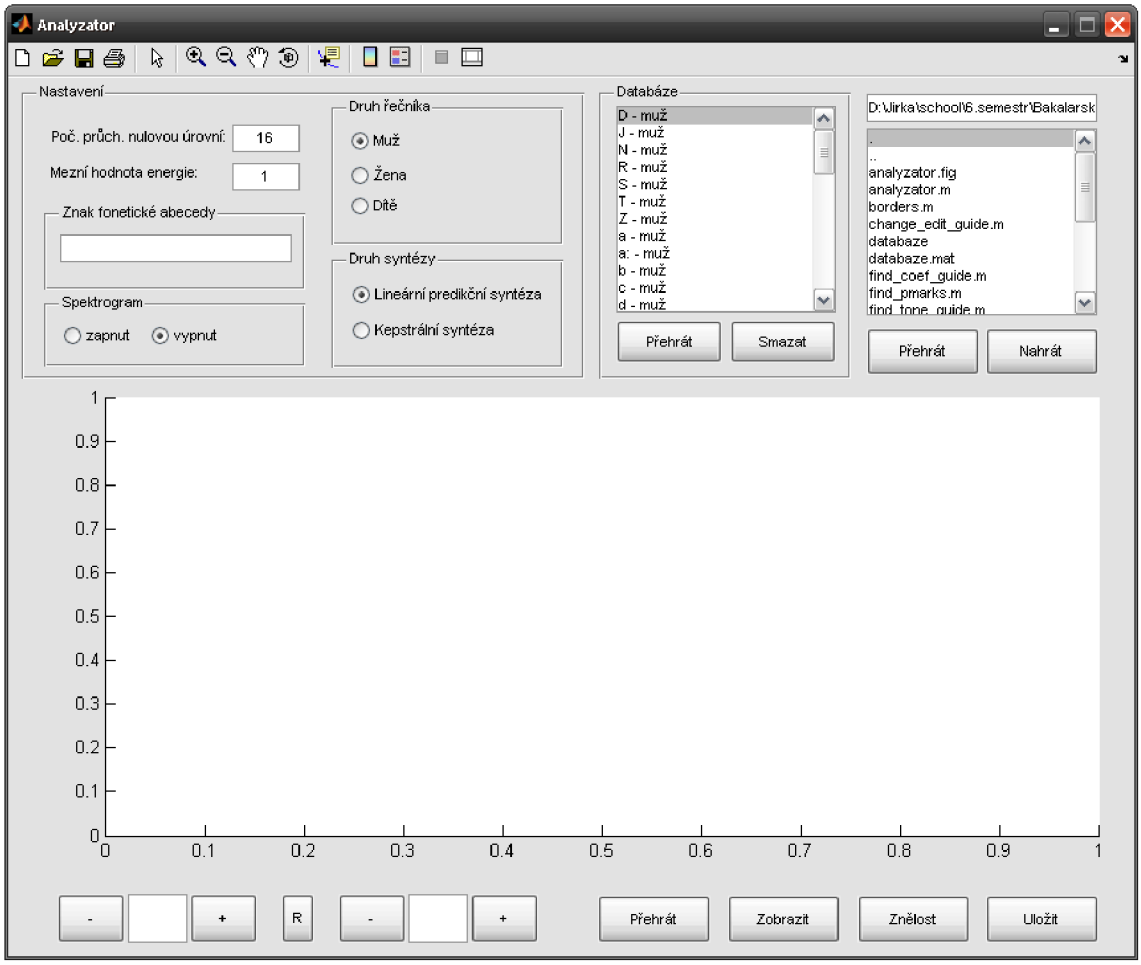
Jakmile získáme řečové signály jednotlivých vět, provedeme ještě korekci jejich intenzity (úrovně) a tempa. Úroveň se opět odvíjí od interpunkčního znaménka na konci věty a od postoje mluvčího. Jestliže budeme např. modelovat vztek, úroveň signálu po celou dobu jeho trvání zvýšíme. Jestliže se bude jednat o větu oznamovací, bude úroveň řečového signálu s dobou trvání mírně klesat atp. Velikosti změn úrovní se opět odvíjí od zvoleného řečového korpusu.

Tempo řeči je v programu závislé pouze na postoji mluvčího, a jak bylo řečeno dříve, odvíjí se od tempa neutrální řeči. Zvyšování nebo snižování tempa řeči se provede způsobem popsaným v kapitole 3.4.3.

Po dodatečných úpravách jednotlivé řečové signály poskládáme do signálu jednoho a uchováme v paměti, odkud signál buď přehrajeme, nebo uložíme na disk.

5 REALIZACE TTS SYSTÉMU

V této části práce popíšeme program, který částečně vychází z návrhu, který byl proveden v předešlé kapitole. Jedná se o jednoduchý TTS systém, který umožní syntézu založenou na fonémech nebo difónech, přičemž bude rovněž možné vybírat z řečových jednotek namluvených mužem, ženou či dítětem. Celý popis je doplněn některými příklady a ukázkami.



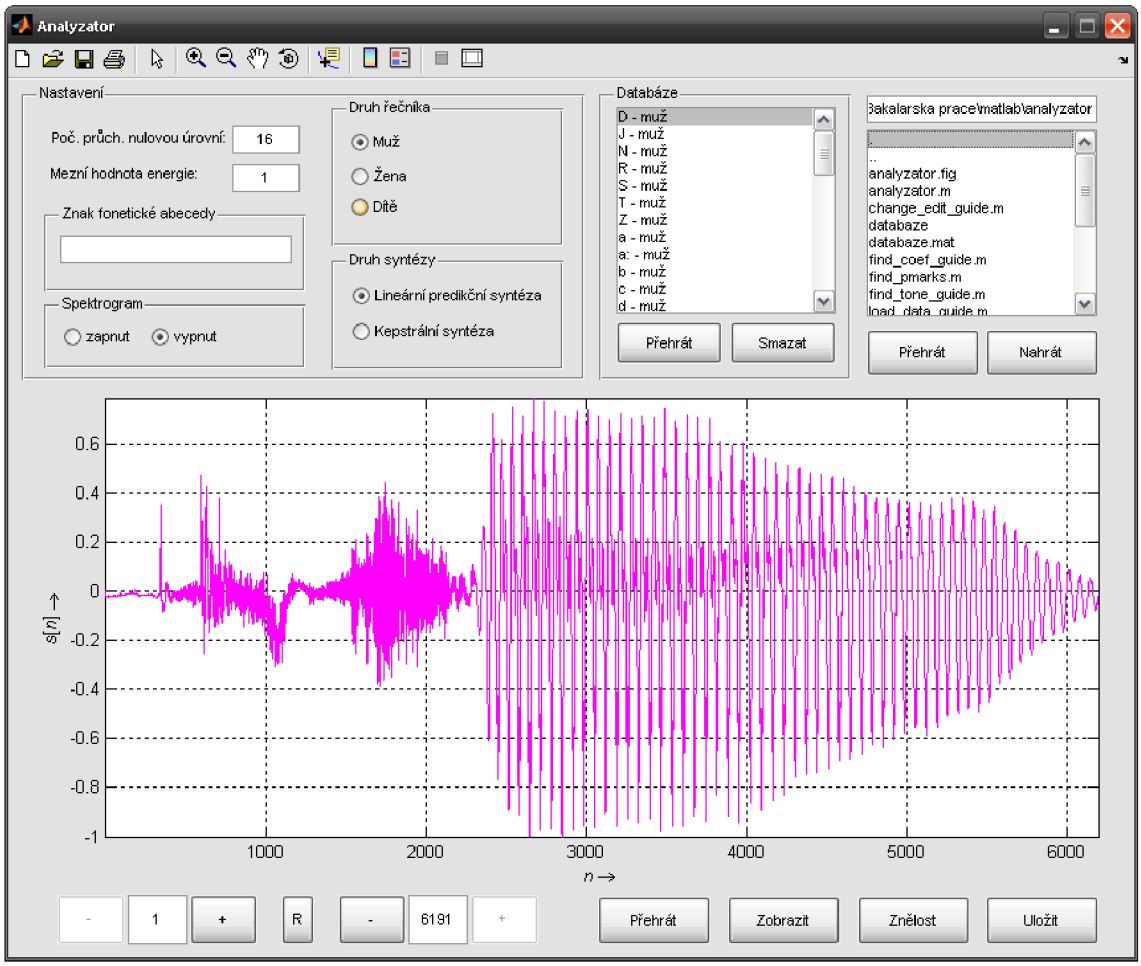
Obr. 5.1: Úvodní obrazovka analyzátoru

5.1 Program analýzy

Program analýzy je složen z několika souborů, jejichž význam je uveden v tab. 5.1.[1] Analyzátor se spouští pomocí příkazu `analyzator`, ovšem je nutné, aby byl v MATLABu namapovaný adresář, ve kterém se soubor `analyzator.m` a další přidružené

[1]Nápovědu k funkcím lze rovněž vyvolat z příkazového okna MATLABu příkazem `help nazev_funkce`.

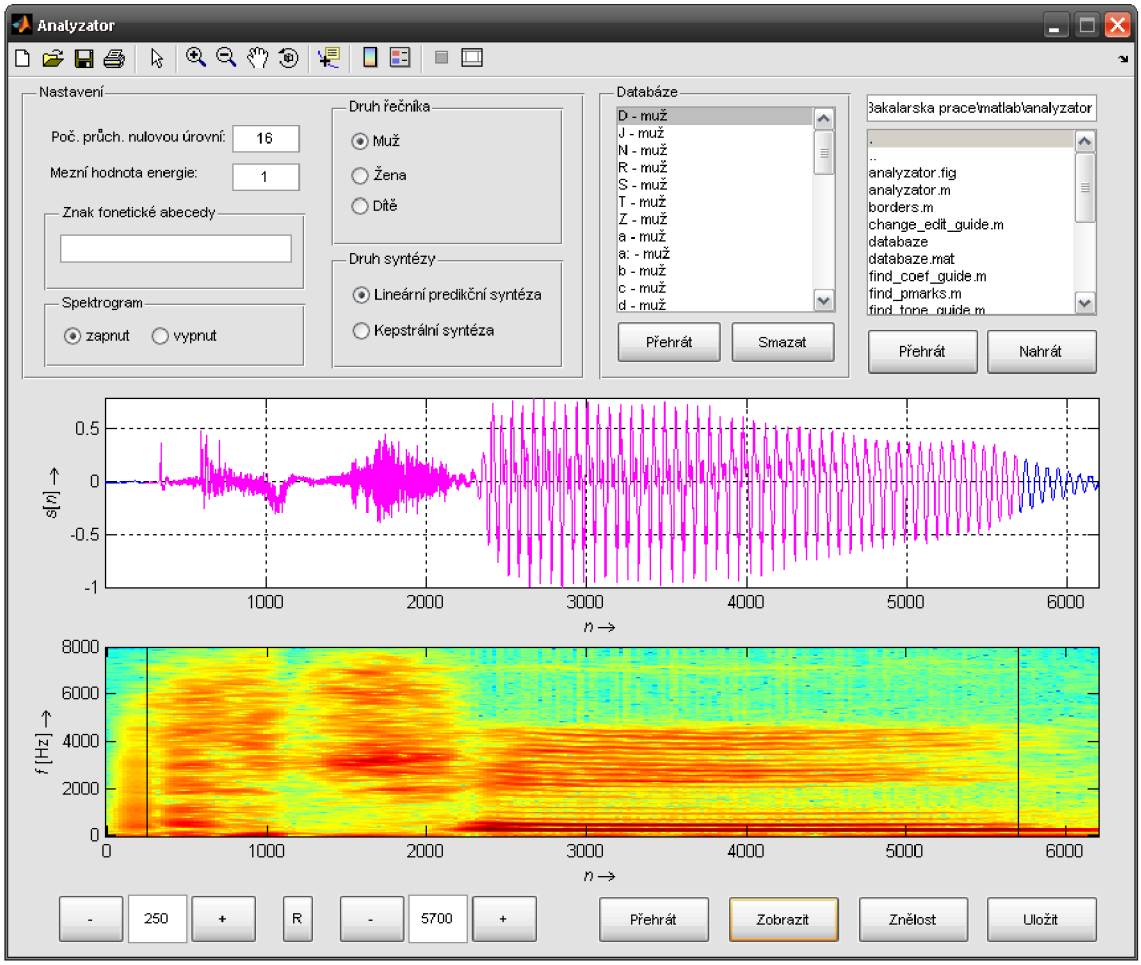
soubory nachází. Po potvrzení příkazu se objeví okno analyzátoru, zobrazené rovněž na obr. 5.1, ve kterém je již možné pracovat.



Obr. 5.2: Analyzátor s načteným řečovým signálem

V pravé části okna je listbox, který slouží k procházení adresářů a lokalizaci řečového signálu, který bude následně analyzován. Ještě před načtením signálu je možné jej přehrát, což nám umožní jednodušší identifikaci. Poté je možné signál nahrát, a to buď stisknutím tlačítka `Nahrát`, nebo dvojklikem na název souboru. Jakmile se signál načte, bude okno analyzátoru vypadat podobně jako na obr. 5.2. Signál se zobrazí v časové rovině. Spektrogram se může aktivovat volbou v nastavení, přičemž se vykreslí pod časový průběh signálu. Uživatel si nyní může vybrat část signálu, a to pomocí tlačítek + a −, umístěných vlevo dole, která omezují pravou a levou hranici signálu. Tyto hranice je možné také vepsat přímo ručně. Označená část signálu je v časové rovině vybarvena růžově, neoznačená modře, ve spektrogramu se oblast zájmu vyznačí svislými čarami stejně jako na obr. 5.3. Stisknutím tlačítka `R`

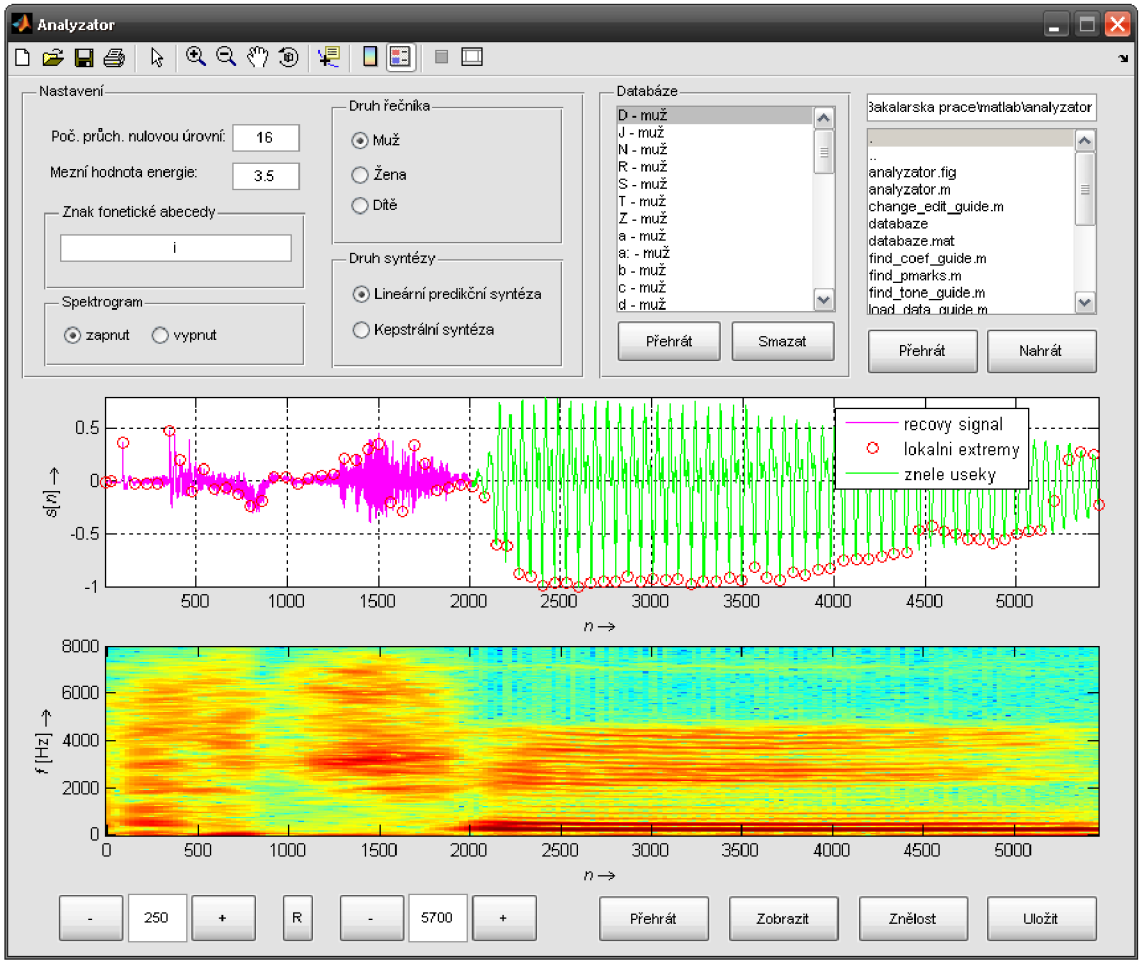
se označí znovu celý signál.



Obr. 5.3: Analyzátor s označenou částí signálu

S označenou částí je možné dále pracovat. Především je možné vyznačit v oblasti znělé a neznělé úseky, přičemž mezní hodnoty, ovlivňující rozhodování o znělosti, může uživatel sám nastavit v levé horní části okna. Po stisknutí tlačítka `Znělost` se v řečovém signálu vyznačí znělé úseky zelenou barvou. Příklad je na obr. 5.4, kde je vyznačena znělá a neznělá část slova „tři“. Z obrázku je hned zřejmé, ve kterých částech se nachází jednotlivé fonémy. Neznělé fonémy „t“ a „ř“ jsou vykresleny růžově a znělá samohláska „i“ zeleně. Červenými kolečky jsou rovněž označené lokální extrémy.

Jakmile je uživatel se správně zjištěnou znělostí spokojen, může vzorek signálu uložit do banky řečových jednotek. V levé horní části se jen zadá znak fonetické abecedy, reprezentující vzorek, určí se druh řečníka a pak se stisknutím tlačítka `Uložit` vše uloží. Obsah databáze je pro přehlednost vypsán v dalším listboxu.



Obr. 5.4: Řečový signál s vyznačenou znělou a neznělou částí

Vzorek je možné v databázi přehrát, nebo smazat. O tom, zda-li se provede lineární predikční nebo kepstrální syntéza, rozhodne opět uživatel, a to volbou v části `Druh syntézy`.

5.2 TTS systém

TTS systém využívá funkce, které jsou popsány v tab. 5.2. Spouští se z příkazového okna MATLABu příkazem `syntezator`. Nutné však je, aby měl MATLAB tento adresář namapovaný a také, aby soubor s databází `databaze.mat` byl přístupný pod cestou `../analyzator/databaze.mat`. Při práci s difóny musí být rovněž přístupná databáze pod cestou `../analyzator/databaze/databaze_difony_ufe.mat`. Po spuštění TTS systému se zobrazí úvodní obrazovka jako na obr. 5.5.

Tab. 5.1: Souborová struktura analyzátoru

Název souboru	Stručný popis
`analyzator.fig`	Soubor s informacemi o vzhledu analyzátoru.
`analyzator.m`	Tento soubor obsahuje základní zdrojový kód analyzátoru.
`databaze.mat`	Obsahuje buňku, která uchovává databázi řečových jednotek.
`find_coef_guide.m`	Funkce `find_coef_guide(handles,y,FS,E,Z)`, která ze vstupního řečového vzorku `y` vypočítá lineární predikční a kepstrální koeficienty, které spolu s dalšími informacemi (druh řečníka, znak fonetické abecedy, periody a lokální extrémy) uloží do databáze `databaze.mat`
`find_pmarks.m`	Funkce `find_pmarks(speech,FS)` obsahuje algoritmus popsaný v lit. [4]. Ve vstupním řečovém signálu `speech` nalezne lokální extrémy.
`find_tone_guide.m`	Funkce `find_tone_guide(handles,y,FS,E,Z)` barevně vyznačí v řečovém signálu `y` znělé a neznělé úseky, přičemž o znělosti rozhoduje mezní hodnota energie `E` a mezní počet průchodů nulovou úrovní `Z`. Proměnná `FS` značí vzorkovací kmitočet.
`change_edit_guide.m`	Funkce `change_edit_guide(handles)` kontroluje, zdali není v polích, určující hranice zpracovaného signálu, znak jiného charakteru, než číselného.
`load_data_guide.m`	Funkce `load_data_guide(handles)` zobrazuje v listboxu okna analyzátoru databázi řečových jednotek.
`pnt_butt_guide.m`	Funkce `pnt_butt_guide(handles)` zajišťuje aktivaci a deaktivaci tlačítek, která slouží k vymezení části načteného signálu.
`resynt_guide.m`	Funkce `resynt_guide(handles,FS)` provede zpětnou syntézu řečového signálu nahraného v databázi řečových jednotek, přičemž druh syntézy bude vybírán pomocí radio buttonu v analyzátoru.
`show_signal_guide.m`	Funkce `show_signal_guide(handles,y,left,right)` zobrazí signál `y`, omezený zleva hranicí `left` a zprava hranicí `right`, v časové rovině.

Název souboru	Stručný popis
`souhrnm5.dll`	Funkce `souhrnm5(excitation,kepst_koef,1,stcon)` provede kepstrální syntézu řečového signálu, přičemž pole `excitation` obsahuje budicí signál, pole `kepst_koef` kepstrální koeficienty a pole `stcon` obsahuje počáteční podmínky do filtru.
`spectrogram_guide.m`	`spectrogram_guide(handles,y,FS,varargin)` je funkce, která vykreslí spektrogram řečového signálu. Ve spektrogramu je také možné vyznačit oblast zájmu.

5.2.1 Předzpracování vstupního textu

Ještě před syntézou vepíše uživatel do okna `Vstupní text` řetězec, který bude vysloven. Pokud je text uložen na disku ve formátu `*.txt`, je možné tento soubor najít pomocí listboxu a otevřít buď dvojklikem, nebo tlačítkem `Načíst`. Dále je možné provést přímo syntézu, nebo pomocí tlačítka `Přepsat` upravit a zobrazit text, se kterým syntezátor dále pracuje.

Předzpracování textu probíhá v několika fázích, které si popíšeme na následujícím příkladu. Vstupní textový řetězec bude vypadat následovně:[2]

```
VUT v Brně je asi nejznámější technická univerzita na Moravě. FEKT
zastřešuje několik ústavů, ÚTKO sídlí přibližně 1km od Purkyňových
kolejí a cca. 500,8 m od Palackého kolejí. Ve vstupní hale této
budovy je možné připojit se na wi-fi 802.11g, která poskytuje
přenosovou rychlost až 54Mb za sekundu. Asi 99% (možná i více)
studentů na tomto ústavu tvoří mužská část. To už možná neplatí
pro fakultu chemickou. Vedoucí mé bakalářské práce je
Prof. Ing. Zdeněk Smékal, CSc., který sídlí v místnosti PA-342
```

Nejdříve vstupní řetězec rozdělíme na jednotlivé větné úseky pomocí znaku #. To má na starost funkce `sentence_guide(inputString)`. Tato funkce rovněž nahradí tečky neukončující větu celým slovem (v příkladě přepíše 802.11g). Podobně nahrazuje i čárky (500,8 m). Pokud na konci řetězce chybí interpunkční znaménko, je zde automaticky doplněna tečka. Po úpravě bude řetězec vypadat následovně:

```
#VUT v Brně je asi nejznámější technická univerzita na Moravě.#FEKT
zastřešuje několik ústavů,#ÚTKO sídlí přibližně 1km od Purkyňových
```

[2]Text obsahuje úmyslně některé typografické, mluvnické a gramatické chyby.

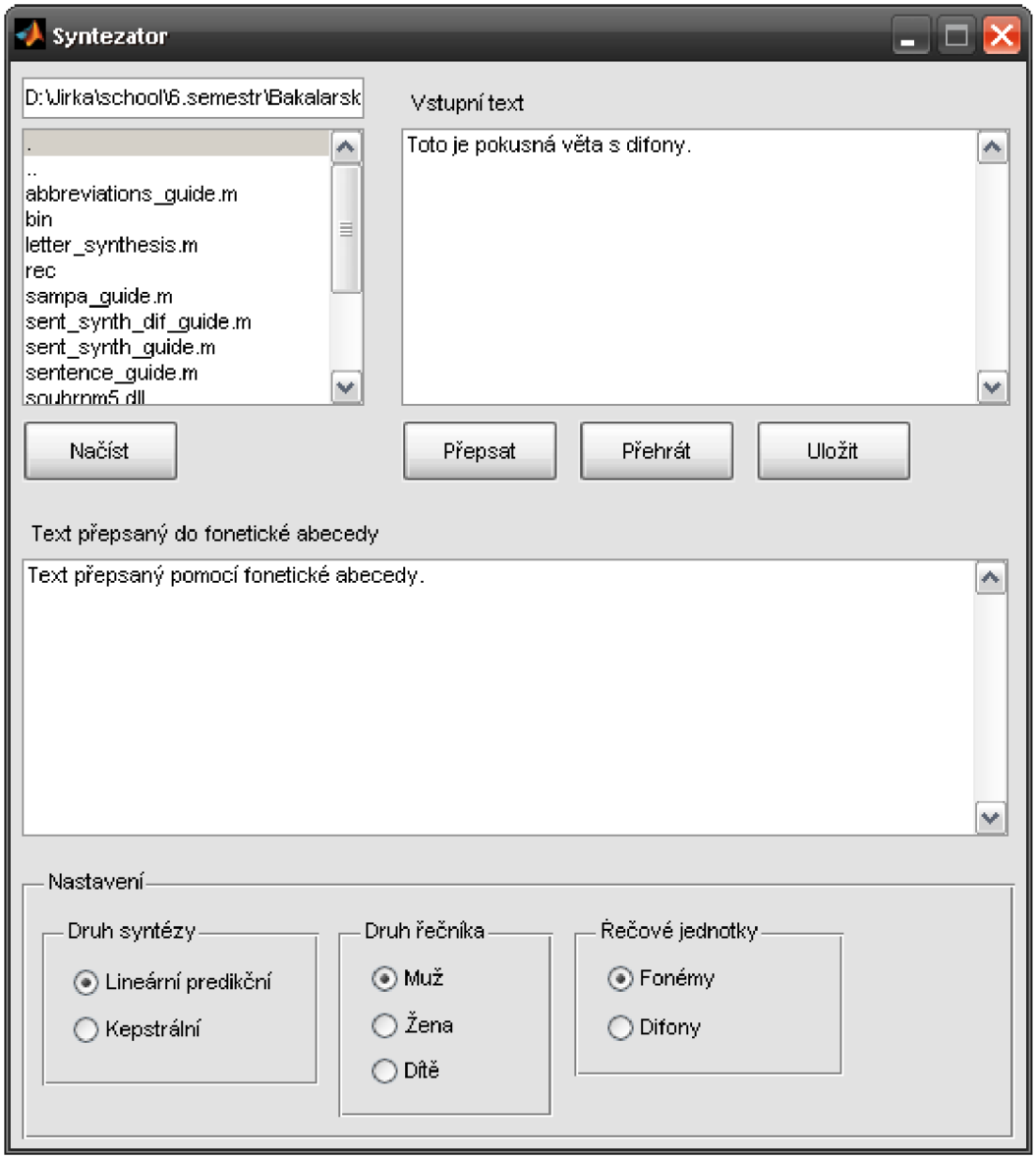
Tab. 5.2: Souborová struktura TTS systému

Název souboru	Stručný popis
`abbreviations_guide.m`	Funkce `abbreviations_guide(inputString)` nahradí některá často používaná čísla celými slovy, zbylá čísla nahradí na základě přítomných číslic. Funkce také nahrazuje některé často používané zkratky.
`letter_synthesis.m`	`letter_synthesis(synt,FS,index,stcon,data)` je funkce, která provede syntézu řečové jednotky z databáze uložené v buňce `data`, přičemž pozici této buňky v databázi určuje proměnná `index`. Druh syntézy určuje proměnná `synt`, počáteční podmínky do filtru uchovává pole `stcon`. Funkce vrací vzorek řeči a nové počáteční podmínky do filtru.
`sampa_guide.m`	Funkce `sampa_guide(inputString)` provede fonetickou transkripci vstupního textového řetězce `inputString`, která je založena na abecedě podobné fonetické abecedě SAMPA.
`sent_synth_dif_guide.m`	`sent_synth_dif_guide(handles,text,FS)` je funkce, která provede syntézu řečového signálu, který bude odpovídat vstupnímu textovému řetězci `text`. Na výběr je z lineární predikční a kepstrální syntézy. Syntéza je provedena z difónů, které jsou uloženy v databázi pod cestou `../analyzator/databaze/databaze_difony_ufe.mat`. Funkce vrací výsledný řečový signál.
`sent_synth_guide.m`	Funkce `sent_synth_guide(handles,text,FS)` provede syntézu řečového signálu, který bude odpovídat vstupnímu textovému řetězci `text`. Na výběr je z lineární predikční a kepstrální syntézy. Syntéza je provedena z fonémů, které jsou uloženy v databázi pod cestou `../analyzator/databaze.mat`. Funkce vrací výsledný řečový signál.

Název souboru	Stručný popis
`sentence_guide.m`	Funkce `sentence_guide(inputString)` rozdělí na základě interpunkčních znamének vstupní textový řetězec `inputString`, přičemž jako rozdělující znak je použit #.
`souhrnm5.dll`	`souhrnm5(excitation,kepst_koef,1,stcon)` je funkce, která provede kepstrální syntézu řečového signálu, přičemž pole `excitation` obsahuje budicí signál, pole `kepst_koef` kepstrální koeficienty a pole `stcon` obsahuje počáteční podmínky do filtru.
`space_synthesis.m`	Funkce `space_synthesis(synt,FS,time)` generuje pauzu, přičemž délka pauzy je rovna násobku proměnné `time` a `FS`. Funkce spolu s polem pauzy vrací nulové počáteční podmínky do filtru.
`syntezator.fig`	Soubor s informacemi o vzhledu TTS systému.
`syntezator.m`	Tento soubor obsahuje základní zdrojový kód TTS systému.
`units_guide.m`	Funkce `units_guide(inputString)` nahradí ve vstupním řetězci `inputString` fyzikální jednotky celými slovy. Slovy jsou rovněž nahrazena znaménka + a −.
`wordPrepr_guide.m`	Funkce `wordPrepr_guide(inputString)` nahradí některá slova novými a to na základě slovníku, který je modifikovanou podobou slovníku použitého v TTS systému EPOS verze 2.5.3. [24]

```
kolejí a cca. 500 čárka 8 m od Palackého kolejí.#Ve vstupní hale této
budovy je možné připojit se na wi-fi 802 tečka 11g,#která poskytuje
přenosovou rychlost až 54Mb za sekundu.#Asi 99% (#možná i více)#
studentů na tomto ústavu tvoří mužská část.#To už možná neplatí
pro fakultu chemickou.#Vedoucí mé bakalářské práce je
Prof.#Ing.#Zdeněk Smékal,#CSc.,#který sídlí v místnosti PA-342.#
```

Na textu je rovněž dobře vidět to, jak byl rozdělen titul. I když v tomto případě tečka nemá ukončující funkci, byl za ni vložen znak #. To způsobí mj. delší pauzy ve výčtu titulů, navíc by zde, při případné změně prozodie, klesala intonace, což je žádoucí.



Obr. 5.5: Úvodní obrazovka TTS systému

Dále v textovém řetězci nahradíme fyzikální a jiné, v technické oblasti často používané, zkratky. Slovy budou rovněž nahrazeny znaménka + a −. K tomu použijeme funkci `units_guide(inputString)`.

```
#VUT v Brně je asi nejznámější technická univerzita na Moravě.#FEKT
zastřešuje několik ústavů,#ÚTKO sídlí přibližně 1 kilometr od
Purkyňových kolejí a cca. 500 čárka 8 metrů od Palackého kolejí.#
Ve vstupní hale této budovy je možné připojit se na wi-fi 802 tečka
11g,#která poskytuje přenosovou rychlost až 54 megabitů za sekundu.#
```

```
Asi 99% (#možná i více)#studentů na tomto ústavu tvoří mužská
část.#To už možná neplatí pro fakultu chemickou.#Vedoucí mé
bakalářské práce je Prof.#Ing.#Zdeněk Smékal,#CSc.,#který sídlí
v místnosti PAmínus 342.#
```

Z textu je zřejmé, že funkce správně nahradila některé zkratky, ovšem je nutné upozornit na to, že při zpracování není zohledněn kontext, ve kterém se zkratka nachází. Kdyby se nacházely v jiných pádech, mohlo by dojít ke špatnému přepisu. Přesnější fungování algoritmu je popsáno v kapitole 4.2. Funkce vyhledává před čísly znaménka + a −, která následně nahradí celými slovy. Pak může ale dojít k chybě, která nastala v poslední větě.

Dalším krokem bude nahrazení čísel a dalších zkratek. K tomu použijeme funkci `abbreviations_guide(inputString)`.

```
#vut v brně je asi nejznámější technická univerzita na moravě.#fekt
zastřešuje několik ústavů,#útko sídlí přibližně jedna kilometr od
purkyňových kolejí a cirka pětset čárka osm metrů od palackého
kolejí.#ve vstupní hale této budovy je možné připojit se na wi-fi
osm nula dva tečka jedenáctg,#která poskytuje přenosovou rychlost
až pět čtyři megabitů za sekundu.#asi devět devětprocent (#možná
i více)#studentů na tomto ústavu tvoří mužská část.#to už možná
neplatí pro fakultu chemickou.#vedoucí mé bakalářské práce je
profesor#inženýr#zdeněk smékal,#kandidát věd,#který sídlí
v místnosti pamínus tři čtyři dva.#
```

Z textu je patrné, že funkce zmenšila všechna velká písmena na malá. Hned z prvního nahrazení čísla zjistíme, že opět nepracujeme s kontextem. Algoritmus nahrazující čísla je popsán v kapitole 4.2. Kromě čísel jsou celými slovy nahrazeny další zkratky a tituly. Znak % byl sice nahrazen celým slovem, ale jelikož před ním nebyla mezera (správně by zde měla být), došlo k nežádoucímu spojení dvou slov.

Některá slova v řetězci je potřeba před fonetickou transkripcí ještě upravit. K tomu nám slouží funkce `wordPrepr_guide(inputString)`.

```
#vysoké učení technYcké v brně je asi nejznámější technYcká
unyverzita na moravě.#fakulta elektrotechnYky a komunYkačních
technologyjí zastřešuje několyk ústavů,#ústav telekomunYkací
sídlí přibližně jedna kilometr od purkyňových kolejí a cirka
```

```
pětset čárka osm metrů od palackého kolejí.#ve vstupní hale
této budovy je možné připojit se na wi-fi osm nula dva tečka
jedenáctg,#která poskytuje přenosovou rychlost až pět čtyři
megabitů za sekundu.#asi devět devětprocent (#možná i více)#
studentů na tomto ústavu tvoří mužská část.#to už možná
neplatí pro fakultu chemickou.#vedoucí mé bakalářské práce
je profesor#inženýr#zdeněk smékal,#kandYdát věd,#který sídlí
v místnosti pamínus tři čtyři dva.#
```

Celými slovy byla nahrazena např. zkratka `VUT` a `FEKT`. Dále byla upravena ta slova, kde by syntéza po jednotlivých fonémech produkovala špatně vyslovená slova. Např. slovo `univerzita` muselo být přepsáno na `unyverzita`. Takto upravený text již necháme podrobit fonetické transkripci vycházející ze slovníku, který je uveden v tab. 5.3.

Tab. 5.3: Fonetická abeceda vytvořená pro TTS systém

ZČFA	Vlastní přepis	ZČFA	Vlastní přepis	ZČFA	Vlastní přepis
a	a	e	e	i	i
y	y	o	o	u	u
á	a:	é	e:	í	i:
ý	y:	ó	o:	ú	u:
ů	u:	au	a_u	ou	o_u
eu	e_u	f	f	v	v
s	s	z	z	š	S
ž	Z	ch	x	h	h
l	l	r	r	ř	R
j	j	p	p	b	b
t	t	d	d	ť	T
ď	D	k	k	g	g
m	m	n	n	ň	N
c	c	č	t_S	w	v
ě	J	ck	d_zk	čb	d_Z
q	kv	x	ks		

Textový řetězec necháme zpracovat funkcí `sampa_guide(inputString)`. Finální podoba našeho ukázkového textu by pak vypadala následovně:

```
#vysoke: ut_SeNi: texnyd_zke: v brNe je asi nejzna:mNejSi:
```

```
texnyd_zka: unyverzita na moravJ.#fakulta elektrotexnyky
a komunykat_SNi:x texnologyji: zastReSuje Nekolyk u:stavu:,
#u:stav telekomunykaci: si:dli: pRibliZNe jedna kilometr od
purkyNovy:x koleji: a cirka pJtset metru: od palad_zke:ho
koleji:.#ve vstupNi: hale te:to budovy je moZne: pRipojit se
na vi-fi osm nula dva tet_Ska jedena:ctg,#ktera: poskytuje
pRenosovo_u ryxlost aZ pJt t_StyRi megabitu: za sekundu.#asi
devJt devJtprocent (#moZna: i vi:ce)#studentu: na tomto
u:stavu tvoRi: muZska: t_Sa:st.#to uZ moZna: neplaTi: pro
fakultu xemid_zko_u.#vedo_uci: me: bakala:Rske: pra:ce je
profesor#inZeny:r#zdeNek sme:kal,#kandyda:t vJd,#ktery:
si:dli: v mi:stnosTi pami:nus tRi t_StyRi dva.#
```

Nyní je text již připraven k odeslání na vstup syntezátoru. Ten si řetězec rozdělí na největší možné řečové jednotky, kterými disponuje v inventáři. Pokud některé znaky v databázi nenajde, přeskočí je.

5.2.2 Syntéza řeči

Kliknutím na tlačítko `Přehrát` v okně TTS systému dojde k syntéze a následnému přehrání řečového signálu. Syntéza se řídí nastavením, které je možno měnit ve spodní části okna. Při syntéze je nejdříve vygenerován budící signál a následně je filtrován filtrem, přičemž čitatel a jmenovatel přenosové funkce je složen z koeficientů vypočítaných při analýze. Ve filtru jsou vždy nastaveny počáteční podmínky, které se po filtraci uchovávají v paměti pro další syntézu. Vzniklé řečové jednotky se dále v časové oblasti řetězí a přivádí na výstup systému. Na výstupu se signál ještě normalizuje tak, aby platilo, že $|s[n]| \leq 1$. Kliknutím na tlačítko `Uložit` je možné signál rovněž uložit na disk ve formátu `*.wav`.

6 ZÁVĚR

Tato práce se věnuje lineární predikční a kepstrální syntéze řečového signálu v systémech TTS s možností změny prozodie mluvčího. Je zde popsána řeč v akustické a fonetické rovině, dále je zde uveden princip tvorby řeči, přičemž je zmíněno i několik způsobů grafického znázornění řečového signálu. Některé způsoby zobrazení řečového signálu jsou doplněny o příklady v programu MATLAB.

Na tuto část navazuje teoretický rozbor TTS systému. V úvodu této části je uvedeno obecné a rozšířené blokové schéma těchto systémů, přičemž jsme každý blok zvlášť popsali, uvedli jeho funkci a popř. vliv na výsledný řečový signál. Bylo zde tedy popsáno celé předzpracování vstupního textu, modelování prozodie, segmentace a syntéza řeči v kmitočtové oblasti. Zvláštní pozornost byla věnována bloku modelování prozodie, kde jsme popsali tři hlavní suprasegmentální rysy, které mají největší vliv na přirozenost syntetické řeči. Dále jsme popsali některé emoce, především to, jak se v řeči projevují. Popis emocí byl navíc doplněn tabulkami, které byly experimentálně zjištěny lidmi zabývajícími se touto problematikou.

Na teoretický rozbor řeči a TTS systému navazuje návrh univerzálního českého TTS systému, který je určen k implementaci v programu MATLAB. Systém je navržený pro práci s jednotlivými fonémy, tudíž nebude produkovat zcela přirozenou řeč, což mj. způsobuje ztráta koartikulace sousedních hlásek při segmentaci. Díky tomu, že bude systém založen na syntéze v kmitočtové oblasti, bude umožňovat měnit druh řečníka, a zároveň modelovat jeho emoce. Na výběr bude ze dvou druhů syntéz: lineární predikční a kepstrální syntéza. Kepstrální syntéza produkuje přirozenější řeč, jelikož díky kepstru můžeme vystihnout jak formanty tak antiformanty jednotlivých fonémů. V návrhu jsou rovněž popsány některé algoritmy, sloužící k předzpracování vstupního textového řetězce.

V této části práce je rovněž proveden návrh jednoduchého analyzátoru řeči. Tento analyzátor umožní řečový signál nahrát, zobrazit ho v časové a kmitočtové oblasti, vyznačit znělé a neznělé části, vypočítat a uložit parametry signálu do inventáře řečových jednotek.

Poslední část práce obsahuje popis realizovaného TTS systému a analyzátoru. Realizace analyzátoru vychází zcela z návrhu, který byl proveden v předešlé kapitole. V práci je především uveden popis práce s analyzátorem, přičemž je vše vysvětleno na příkladě. Analyzátor je rovněž schopen provést zpětnou syntézu řečové jednotky z inventáře.

Tato kapitola dále pokračuje popisem realizovaného TTS systému, který částečně vychází z předešlého návrhu. Systém neumožňuje měnit prozodii, jelikož se syntéza

založena na fonémech nevyznačuje dobrou srozumitelností a přirozeností řeči. Pro příklad jsme proto vytvořili inventář řečových jednotek založených na difónech, které již produkují řeč více srozumitelnou, nicméně stále ještě nepřirozenou.

Během práce jsme přišli na to, že srozumitelnost syntetické řeči nejvíce záleží na vhodné volbě a zpracování řečových jednotek. Jak bylo řečeno, při práci s fonémy je velmi obtížné produkovat srozumitelnou řeč. Lepší výsledky produkuje řeč složená s difónů, jelikož ty už vystihují koartikulaci mezi hláskami. Nicméně řeč je stále nepřirozená. Proto se v dnešní době začíná preferovat práce s trifóny, jelikož se řeč, založená na těchto řečových jednotkách, vyznačuje přijatelnou srozumitelností. V praxi se začínají uplatňovat především tzv. kontextově závislé fonémy závislé na svém levém a pravém okolí. Těchto řečových jednotek se však v českém jazyce vyskytuje až několik tisíc, proto se upouští od manuální segmentace a přechází se k segmentaci zautomatizované, přičemž se k tomuto účelu nejvíce používají skryté Markovovy modely. Řečové jednotky se dnes navíc vzorkují vzorkovacím kmitočtem $f_{\mathrm{vz}} = 32\,\mathrm{kHz}$, takže ve výsledku nezní příliš ploše.

V některých komerčních TTS systémech se k této problematice přistupuje tak, že se v inventáři nejčastěji vyskytující slova (nebo jiné úseky) uchovávají celá a zbylá se pak již odsyntetizují pomocí kratších řečových jednotek. Takto vzniklá řeč se pak vyznačuje dobrou srozumitelností a přirozeností, ovšem na úkor toho, že inventář řečových jednotek zabírá velké místo na disku nebo v paměti.

Tato práce sloužila především k rozboru základní problematiky TTS systémů. Tyto systémy dnes dosahují velké složitosti a propracovanosti. Na každé části systému se však může ještě mnohé zlepšit, ať už se jedná o předzpracování textu, kde zatím stoprocentně nefunguje sémantická analýza, tak např. části zajišťující změnu prozodie dodnes neprodukují zcela přirozenou řeč. Nicméně největší pozornost je stále věnována volbě a přípravě vhodného inventáře řečových jednotek. Proto by tato práce měla dále pokračovat, přičemž by se měla orientovat na syntézu řeči z textu s využitím statistického přístupu k automatickému vytvoření databáze řečových jednotek. K tomu se dnes hojně využívá volně stažitelný HTK soubor nástrojů, který slouží k práci se skrytými Markovovými modely [28]. Speciálně pro syntézu řeči byl ještě navržen HTS balíček, který je nadstavbou HTK [25]. Úkolem další práce by mělo být nastudování těchto nástrojů, vytvoření foneticky bohatého řečového korpusu a následná tvorba vhodného inventáře. Potom, co bude řeč dostatečně přirozená, bude možné přistoupit k modelování prozodie.

LITERATURA

[1] BATŮŠEK, R. *Czech SAMPA* [online]. Praha: ČVUT, 2003 [cit. 31. 10. 2007]. Dostupné z URL: <http://noel.feld.cvut.cz/sampa/>.

[2] ČERNOCKÝ, J. *Zpracování řečových signálů*. Elektronická skripta FIT VUT v Brně. 2006, 129 s. Dostupné z URL: <http://www.fit.vutbr.cz/∼cernocky/speech/opora/zre_opora.pdf>.

[3] ČERMÁK, J, BALÍK, M. *Přístupy k modelování prozodie v TTS systémech* [online]. Elektrorevue: Internetový časopis. 2005, roč. 2005, č. 50 [cit. 16. 10. 2007]. Dostupné z URL: <http://www.elektrorevue.cz/clanky/05050/index.html>. ISSN 1213-1539.

[4] GONCHAROFF, V. a GRIES P. *An Algorithm for Accurately Marking Pitch Pulses in Speech Signals*. Chicago: University of Illinois, 1998. 4 s. Oborová práce.

[5] HORÁK, P. *Modelování suprasegmentálních rysů mluvené češtiny pomocí lineární predikce* [online]. ČVUT Praha, 2002. 116 s. Dizertační práce. Dostupné z URL: <http://epos.ure.cas.cz/dizertace/text/dizertace.pdf>.

[6] KRČMOVÁ, M. *Fonetika* [online]. Brno: Filosofická fakulta MU, 2007 [cit. 23. 10. 2007]. Dostupné z URL: <http://is.muni.cz/elportal/estud/ff/js07/fonetika/materialy/index.html>.

[7] LEMMETTY, S. *History and Development of Speech Synthesis* [online]. Helsinky: University of Technology, 1999 [cit. 16. 10. 2007]. Dostupné z URL: <http://www.acoustics.hut.fi/publications/files/theses/lemmetty_mst/chap2.html>.

[8] MATOUŠEK, J. *Syntéza řeči z textu s využitím statistického přístupu k automatickému vytvoření databáze řečových jednotek: ARTIC – český TTS systém*. Katedra Kybernetiky, ZČÚ, Plzeň, 2000. 164 s. Vedoucí dizertační práce prof. Ing. Josef Psutka, CSc.

[9] MATOUŠEK, J., PSUTKA, J. *ARTIC: A New Czech Text-to-Speech System Using Statistical Approach to Speech Segment Database Construction.* [online]. In The Proceedings of the International Conference on Spoken Language Processing ICSLP 2000. Peking: Čína, 2000. 4. svazek, s. 612-615. Dostupné z URL: <http://artic.zcu.cz/documents/2000_ICSLP_matousek.pdf>.

[10] MATOUŠEK, J., PSUTKA, J., KRŮTA, J. *Design of Speech Corpus for Text-to-Speech Synthesis.* [online]. In 7th European Conference on Speech Communication and Technology EUROSPEECH 2001. Aalborg: Dánsko, 2001. s. 2047-2050. Dostupné z URL: <http://artic.zcu.cz/documents/2001_EUROSPEECH_matousek.pdf>.

[11] MATOUŠEK, J., TIHELKA, D., PSUTKA, J. *Automatic Segmentation for Czech Concatenative Speech Synthesis Using Statistical Approach with Boundary-Specific Correction.* [online]. In Proceedings of the 8th European Conference on Speech Communication and Technology EUROSPEECH 2003. Ženeva: Švýcarsko, 2003. s. 301-304. Dostupné z URL: <http://artic.zcu.cz/documents/2003_EUROSPEECH_matousek.pdf>. ISSN 1018-4074.

[12] PALKOVÁ, Z. *Fonetika a fonologie češtiny s obecným úvodem do problematiky oboru*. Karoilinum, Praha 1994. ISBN 80-7066-843-1.

[13] PSUTKA, J., MÜLLER, L., MATOUŠEK, J., RADOVÁ, V. *Mluvíme s počítačem česky*. Praha: Academia, 2006. 746 s. ISBN 80-200-1309-1.

[14] PTÁČEK, M. *Vybrané statě z akustiky řeči*. Skripta Fonetického ústavu FF UK, Praha, 1993.

[15] SIGMUND, M. *Analýza řečových signálů*. Brno: FEI VUT Brno, 2000. 86 s. ISBN 80-214-1783-8.

[16] SMÉKAL, Z. *Číslicové zpracování signálů*. Brno: FEKT VUT Brno, 2007. 183 s.

[17] SMÉKAL, Z. a SYSEL, P. *Číslicové filtry*. Elektronická skripta FEKT VUT v Brně. 2004, 130 s.

[18] ŠEBESTA, V., SMÉKAL, Z. *Signály a soustavy*. Brno: UREL FEKT VUT v Brně, 2006. s. 1-167.

[19] VLČKOVÁ-MEJVALDOVÁ, J. *Prozodie, cesta i mříž porozumění*. Karolinum 2006, Praha. ISBN 80-246-1266-6.

[20] VONDRA, M. *Souhrnné kepstrální modely pro syntézu TTS*. FEI VUT v Brně, 2001. 78 s. Vedoucí diplomové práce prof. Ing. Zdeněk Smékal, CSc.

[21] VONDRA, M., SMÉKAL, Z., VÍCH, R. *State–Space Representation Of Cepstral Vocal Tract Model For DSP Implementation*. ElectronicsLetters.com – www.electronicsletters.com, 2002, roč. 2002, č. 6.9., s. 1 – 6. ISSN: 1213-161X.

[22] WON PARK, S. *Linear Predictive Speech Processing* [online]. Texas A & M University, 2007. 25 s. Oborová práce. Dostupné z URL: <http://www.engineer.tamuk.edu/SPark/chap7.pdf>.

[23] ZAPLATÍLEK, K., DOŇAR, B. *MATLAB - tvorba uživatelských aplikací.* 1. vyd. Praha: BEN - technická literatura, 2004. 216 s. ISBN 80-7300-133-0.

[24] *Epos TTS System* [online]. Sourceforge.net, 2007 [cit. 5. 11. 2007]. Dostupné z URL: <http://sourceforge.net/project/showfiles.php?group_id=82334>.

[25] *HMM-based Speech Synthesis System (HTS)* [online]. Department of Computer Science, Nagoya Institute of Technology, 2008 [cit. 23. 4. 2008]. Dostupné z URL: <http://hts.sp.nitech.ac.jp/>.

[26] *International Phonetic Alphabet* [online]. 2005 [cit. 1. 11. 2007]. Dostupné z URL: <http://www.arts.gla.ac.uk/IPA/>.

[27] *RealSpeak^{TM}: Expressive, Natural, Multi-Lingual Text-To-Speech* [online]. Nuance Communications, Inc., 2007 [cit. 28. 11. 2007]. Dostupné z URL: <http://www.nuance.com/realspeak/>.

[28] *The Hidden Markov Model Toolkit (HTK)* [online]. Cambridge University Engineering Department, 2006 [cit. 23. 4. 2008]. Dostupné z URL: <http://htk.eng.cam.ac.uk/>.

[29] *Voice Reader Online* [online]. Linguatec, 2007 [cit. 28. 11. 2007]. Dostupné z URL: <http://www.linguatec.net/onlineservices/voice_reader>.

[30] *Wikipedia: The Free Encyclopedia: Discrete Fourier transform* [online]. 2007 [cit. 8. 11. 2007]. Dostupné z URL: <http://en.wikipedia.org/wiki/Discrete_Fourier_transform>.

[31] *Wikipedia: The Free Encyclopedia: Formant* [online]. 2007 [cit. 16. 10. 2007]. Dostupné z URL: <http://en.wikipedia.org/wiki/Formants>.

[32] *Wikipedia: The Free Encyclopedia: Speech synthesis* [online]. 2007 [cit. 16. 10. 2007]. Dostupné z URL: <http://en.wikipedia.org/wiki/Speech_synthesis>.

[33] *Wikipedia: The Free Encyclopedia: Window function* [online]. 2007 [cit. 8. 11. 2007]. Dostupné z URL: <http://en.wikipedia.org/wiki/Window_function>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

a_i	lineární predikční koeficienty
ARMA	(autoregressive moving average) autoregresivní náhodný proces s klouzavým průměrem
$c[n]$	reálné kepstrum řeči
ČFA	česká fonetická abeceda
DFT	(discrete Fourier Transform) diskrétní Fourierova transformace
DPMB	Dopravní podnik města Brna
e	Eulerovo číslo
E	energie diskrétního řečového signálu
$e[n]$	chyba predikce
E-mail	(electronic mail) elektronická pošta
f	kmitočet [Hz]
F_0	základní kmitočet řečového signálu [Hz]
F_1, F_2, F_3	kmitočet prvního, druhého a třetího formantu [Hz]
FEKT	Fakulta elektrotechniky a komunikacních technologií
FFT	(fast Fourier Transform) rychlá Fourierova transformace
f_{vz}	vzorkovací kmitočet [Hz]
G	zesílení
$g(t)$	spojitý budicí signál
$G(\omega)$	spektrum spojitého budicího signálu
GPS	(Global Positioning System) navigační družicový systém
GUIDE	(Graphical User Interface Design Envrionment) grafické prostředí programu MATLAB
$h(t)$	spojitá impulsní charakteristika hlasového traktu
$H(z)$	přenosová funkce diskrétního systému

$H(\omega)$	kmitočtová charakteristika hlasového traktu
HMM	(hidden Markov model) skrytý Markovův model
HTML	(Hyper Text Markup Language) značkovací jazyk pro hypertext
IDFT	(Inverse Discrete Fourier Transform) zpětná diskrétní Fourierova transformace
IIR	(infinite impulse response) nekonečná impulzní odezva
IPA	(International Phonetic Alphabeth) Mezinárodní fonetická abeceda
j	komplexní jednotka
k	koeficient opakování rámců při syntéze
konkatenace	operace spojování řetězců
$l[n]$	kepstrální okno
LPC	(Linear Predictive Coding) lineární predikční kódování
LP-PSOLA	Linear Predictor Pitch Synchronous Overlap Add
m	koeficient opakování sady parametrů při změně trvání řeči
M	počet předchozích vzorků řeči
MATLAB	MATrix LABoratory
MBROLA	Multi Band Resynthesis Overlap Add
MP3	(MPEG-1 Layer 3) formát ztrátové komprese zvukových souborů
MPEG	(Motion Picture Experts Group) název skupiny standardů používaných na kódování audiovizuálních informací
N	délka segmentu řeči
NATO	(North Atlantic Treaty Organisation) Severoatlantická aliance
N_{fr}	délka rámce při segmentaci řeči
N_{frms}	počet překrývajících rámců
o_{fr}	délka překrytí rámce
OSN	Organizace spojených národů

RAF	(Royal Air Force) letectvo britských ozbrojených sil
$s(t)$	spojitý řečový signál
$S(\omega)$	spektrum spojitého řečového signálu
$S[k]$	diskrétní spektrum řečového signálu
$s[n]$	diskrétní řečový signál obecně
$s'[n]$	úsek diskrétního řečového signálu
$\tilde{s}[n]$	odhad vzorku řeči
$S'[k]$	úsek diskrétního řečového signálu v kmitočtové oblasti
SAMPA	(Speech Assessment Methods Phonetic Alphabet) řečové vyhodnocení metod fonetické abecedy
SAS	(Special Air Service) britské speciální jednotky
SMS	(Short Message Service) krátká textová zpráva šířená pomocí telefonu
t	čas [s]
t_{p}	původní doba trvání řečové jednotky [s]
t_{n}	nová doba trvání řečové jednotky [s]
T_0	základní perioda řečového signálu [s]
$T_{0\mathrm{se}}$	perioda základního tónu při segmentaci [s]
$T_{0\mathrm{sy}}$	perioda základního tónu při syntéze [s]
TD-PSOLA	Time Domain Pitch Synchronous Overlap Add
TTS	(Text-to-Speech) systém sloužící k převodu textu na řeč
T_{vz}	vzorkovací perioda [s]
$W[k]$	obraz váhovacího okna v kmitočtové oblasti
$w[n]$	váhovací okno
WAV	(Waveform Audio Format) audio formát
WMA	(Window Media Audio) audio formát

$x(t)$	obecný spojitý signál
$X(\omega)$	spektrum obecného signálu
$X[k]$	spektrum obecného diskrétního signálu
$\hat{X}[k]$	přirozený logaritmus spektra obecného diskrétního signálu
$x[n]$	obecný diskrétní signál
$\hat{x}[n]$	obecný diskrétní signál v nelineární časové oblasti
XML	(eXtensible Markup Language) rozšiřitelný značkovací jazyk
z	komplexní proměnná transformace $\mathcal{Z}$
Z	počet průchodů nulovou úrovní
ZČFA	zjednodušená česká fonetická abeceda
Δf	skutečný kmitočet [Hz]
ζ	kvefrence [s]
π	Ludolfovo číslo (3,1415926535...)
τ	časový úsek [s]
ω	úhlový kmitočet [$\mathrm{rad} \cdot \mathrm{s}^{-1}$]
Ω	Ohm – jednotka elektrického odporu
€	Euro – jednotka měny
$\forall$	platí pro všechny
$\lceil . \rceil$	operace zaokrouhlení nahoru
$\lfloor . \rfloor$	operace zaokrouhlení dolů