Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra kybernetiky

BAKALÁŘSKÁ PRÁCE

PLZEŇ, 2011 **Pavel Ptáček**

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta aplikovaných věd
Akademický rok: 2010/2011

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Pavel PTÁČEK**
Osobní číslo:	**A09B0883P**
Studijní program:	**B3918 Aplikované vědy a informatika**
Studijní obor:	**Kybernetika a řídicí technika**
Název tématu:	**Automatická detekce řeči v řečových nahrávkách pro korpusově orientovanou syntézu řeči**
Zadávající katedra:	**Katedra kybernetiky**

Zásady pro vypracování:

1. Seznamte se s problematikou automatické detekce řeči (voice activity detection).
2. Navrhněte algoritmus pro automatickou detekci začátku a konce "užitečného" řečového signálu ve studiových nahrávkách pořízených pro účely korpusově orientované syntézy řeči.
3. V algoritmu pro detekci použijte vhodný klasifikátor a navrhněte pro něj vhodné příznaky.
4. Navržený algoritmus otestujte na reálných řečových datech.

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 19. srpna 2011 …….

vlastnoruční podpis

Poděkování

Velmi rád bych touto cestou poděkoval vedoucímu bakalářské práce panu Doc. Ing. Jindřichu Matouškovi, Ph.D. za vedení práce, poskytnutí cenných rad, hodnotných podnětu a užitečných připomínek při vypracování předkládané bakalářské práce.

Pavel Ptáček

Abstrakt

Tato bakalářská práce se zabývá vytvořením podpůrného nástroje, který automaticky detekuje začátek a konec řeči v řečových nahrávkách. Úloha je řešena pomocí klasifikátoru s využitím knihoven SVM pro programové prostředí Matlab. Cílem práce je přesná detekce začátku a konce řeči. Čtenář se v práci dočte, jak funguje klasifikace a uvidí popsání příznaků, podle kterých klasifikování probíhá. Výsledné vyhodnocení nalezne v přehledných tabulkách, které znázorňují úspěšnost celé klasifikace. Práce také pojednává, jaké se používají příznaky pro detekci řeči v nahrávkách a také pro syntézu řeči. Při vypracování práce zabralo mnoho času testování nejvhodnějších příznaku pro detekci řeči. Řešení úlohy proběhlo pomocí klasifikátoru s použitím SVM knihoven.

Klíčová slova: klasifikátor, syntéza řeči, SVM knihovny, Matlab, detekce, příznak, řečové nahrávky

Abstract

This thesis deals with the creation of support tool, which automatically detects the beginning and ending of speech in speech recordings. The task is solved by using a classifier with SVM libraries for the programming environment Matlab. The goal is to detect precise beginning and ending of speech. The reader can find in this thesis, how it works and see description of symptoms, according to which classification takes place. The resulting evaluation finds in tables that illustrate the success of the entire classification. This work also discusses, which symptoms are used for detection of speech in speech recordings and synthesis of speech. During the work took a lot of time testing the most appropriate symptoms for the detection of speech. Solution of this task was carried out by the SVM classificator using SVM libraries.

Keywords: classificator, synthesis of speech, SVM libraries, Matlab, detection, symptoms, speech recordings

Obsah

1 Úvod

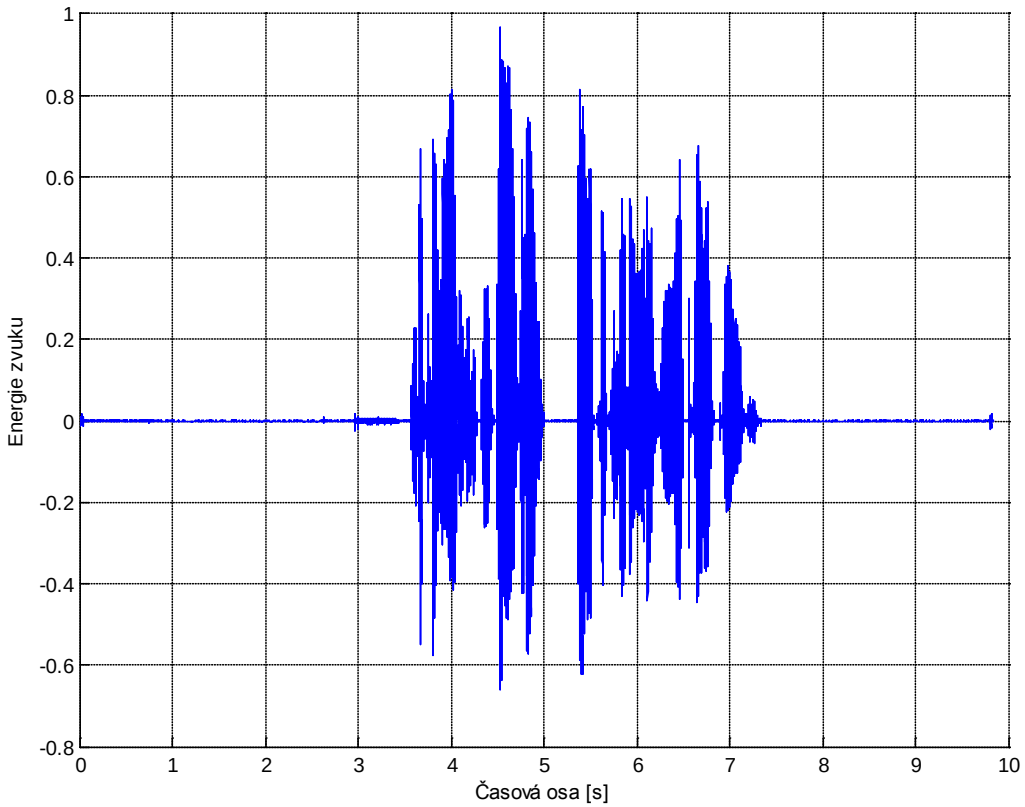
Komunikace prostřednictvím mluvené řeči je základním a nejpoužívanějším přenosem informace mezi lidmi. Je proto pochopitelné, že při současných zvyšujících se možnostech výpočetní techniky usilují vědci a technici o to, aby se rovnocenným partnerem člověka v mluveném dialogu stal počítač. Vyřešení tohoto úkolu je žádané hlavně z toho důvodu, že takový způsob komunikace může být pro člověka velmi prospěšný a často mu může i hodně usnadnit život. Chceme-li, aby se partnerem člověka v mluvené řeči stal počítač, musí se algoritmicky a technicky vyřešit několik celkem komplikovaných úloh, které se zabývají zejména zpracováním řečového signálu, počítačové syntézy a automatického rozpoznávání řeči, včetně „strojového“ porozumění významu rozpoznaných vět.

Bohužel je třeba říct, že plnohodnotný dialogový režim člověka s počítačem prostřednictvím přirozené plynule promlouvané řeči bez jakýchkoli omezení je v současnosti stále ještě nedostupný. Je to zapříčiněno zejména stálými obtížemi s rozpoznáváním spontánní řeči a dále i omezenými možnostmi klasifikace řečového signálu a procesu porozumění smyslu klasifikovaných slov a vět. Tento stav má kořeny v minulosti, kdy se řešení problematiky porozumění přirozenému jazyku rozvíjelo relativně nezávisle a odlišně od cílů konstruovaných řečových klasifikátorů.

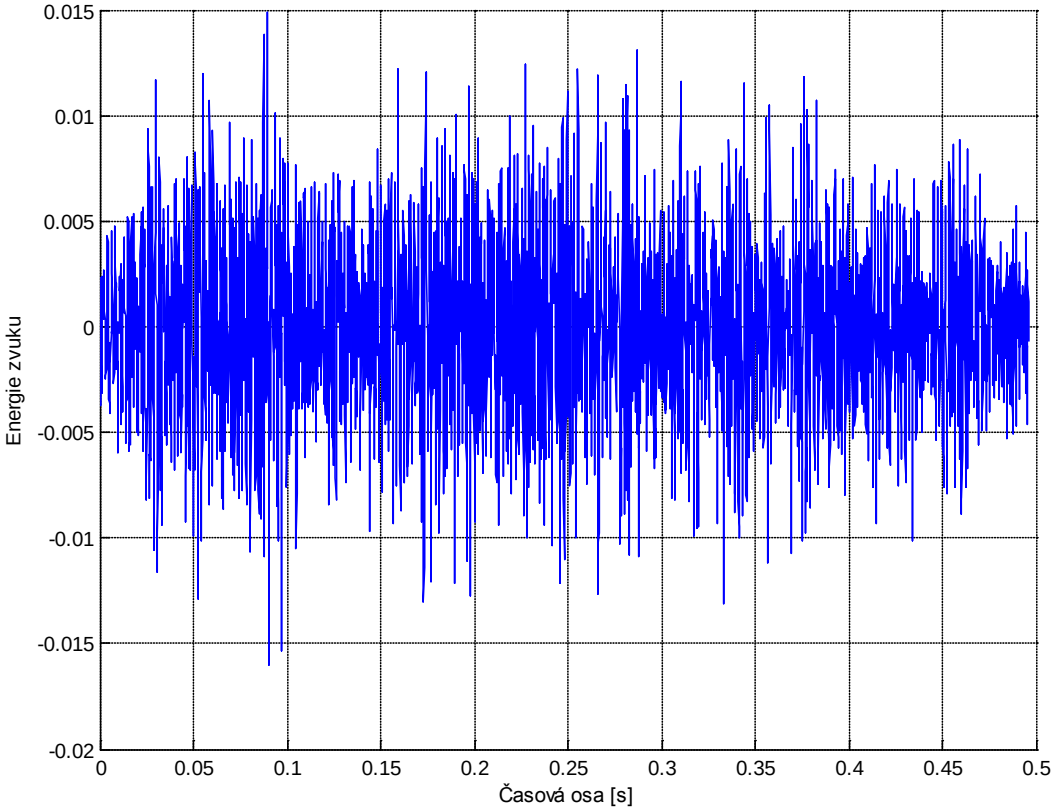
V současné době jsou široce používány komponenty hlasových dialogových systémů, to jsou moduly syntézy a rozpoznávání řeči. Všeobecné využití nacházejí i různé systémy ovládání strojů a zařízení hlasovými povely. Uplatňují se, když člověk nemá k dispozici ruce, které má zaměstnány jinou činností, nebo mohou najít využití pro tělesně hendikepované lidi. Sluchově hendikepovaní lidé určitě ocení automatické on-line titulkování televizních pořadů, pro které není předem připravena textová podoba dané promluvy. Rozsáhlé uplatnění nacházejí také v oblasti automatického převodu psaného textu na mluvenou řeč. To ocení zejména zrakově postižení a lidé s poruchami hlasu [1].

2 Charakter řečového signálu

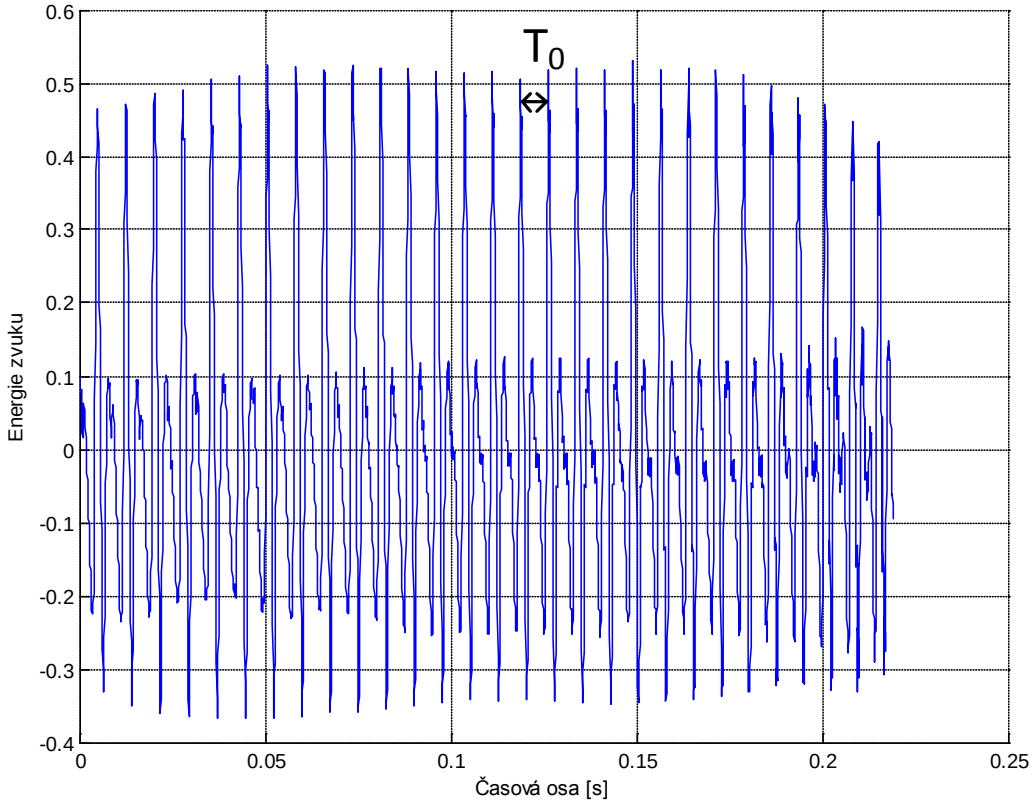
Lidská řeč je souvislý a časové proměnný proces, kterým se zabýváme v řečové syntéze. Je používána i v mé práci, kdy určuji její začátek a konec. Stává se nositelem užitečné informace od řečníka k posluchači. Zároveň je přenášena pomocí akustického vlnění. Řeč je vytvářena ovlivňováním výdechového proudu vzduchu z plic hlasovým ústrojím člověka, začínajícího hlasivkami a končícího rty. Na (obr.2.1) je zobrazen řečový signál při promluvě. Řečovým signálem rozumíme posloupnost diskrétních vzorků signálu, který většinou obdržíme z mikrofonu. Pokud si tento signál pozorně prohlédneme, najdeme zde oblasti (obr.2.3), které jsou více periodické. Jde o znělé části řeči, periodu označujeme T_0 (*základní perioda řeči*, *pitch period*). V řečovém signálu najdeme také oblasti (obr.2.2), které mají charakter šumu, tyto části řeči jsou neznělé. Znělost, popř. neznělost je způsobena tím, jestli výdechový proud vzduchu z plic rozkmitá hlasivkovou štěrbinu, nebo ne [2].



Obr.2.1 – Řečový signál v promluvě



Obr.2.2 – Detail neznělého úseku



Obr.2.3 – Detail znělého úseku

3 Příznaky pro metody detekce řeči

V zadané práci jsem navrhnul algoritmus, který automaticky detekuje začátek a konec „užitečného“ řečového signálu ve studiových nahrávkách pořízených pro účely korpusově orientované syntézy řeči. Výsledky jsou zhodnoceny v kapitole 7.

3.1 Algoritmus detekce řeči

Tento algoritmus je pro naši řešenou úlohu velmi důležitý, protože algoritmy detekce řeč/ticho využívají různých přístupů, jako jsou například: úroveň energie v signálu, počet průchodů nulovou osou, informace o znělosti/neznělosti a mnoho dalších. Lze však modelovat tuto úlohu dvěma základními bloky, a to akustickou analýzou řečového signálu, která vybere vhodné příznaky popisující řečový signál, a klasifikujícím algoritmem, jenž rozlišuje mezi řečovými a neřečovými úseky. Příznaky, které se používají k akustické analýze, jsou výkon (energie) signálu, intenzita, počet průchodů nulou, entropie, spektrální vzdálenost od pozadí, průměrná koherence. V naší řešené úloze jsem použil energii signálu a počet průchodů nulou [6].

3.1.1. Segmentace

Segmentace signálu je velmi důležitá, proto při splňování požadavku na délku rámce je třeba zvolit kompromis mezi stabilním popisem stacionárních úseků a přesným popisem nestacionárních úseků. Segment totiž musí být dostatečně krátký, aby bylo možno co nejpřesněji odhadnout parametry. Obvykle je délka rámce volena mezi 10 a 30 milisekund. V mém případě jsem zvolil délku rámce 25 milisekund, což bývá obvyklá hodnota při parametrizaci mužského hlasu v úlohách rozpoznávání řeči. Výsledky této metody jsou popsány v kapitole 8.4.4.

Při segmentaci je vhodné volit určitou míru překryvu, protože hodnoty parametrů se mohou mezi jednotlivými rámci značně měnit a při váhováním okénkem jsou potlačovány okrajové hodnoty v rámci. Tento úkon však zvyšuje výpočetní a paměťové nároky, takže je třeba vzít tuto skutečnost v potaz při volbě procenta překryvu. Na 25 milisekund rámci jsem zvolil překryv 5 milisekund.

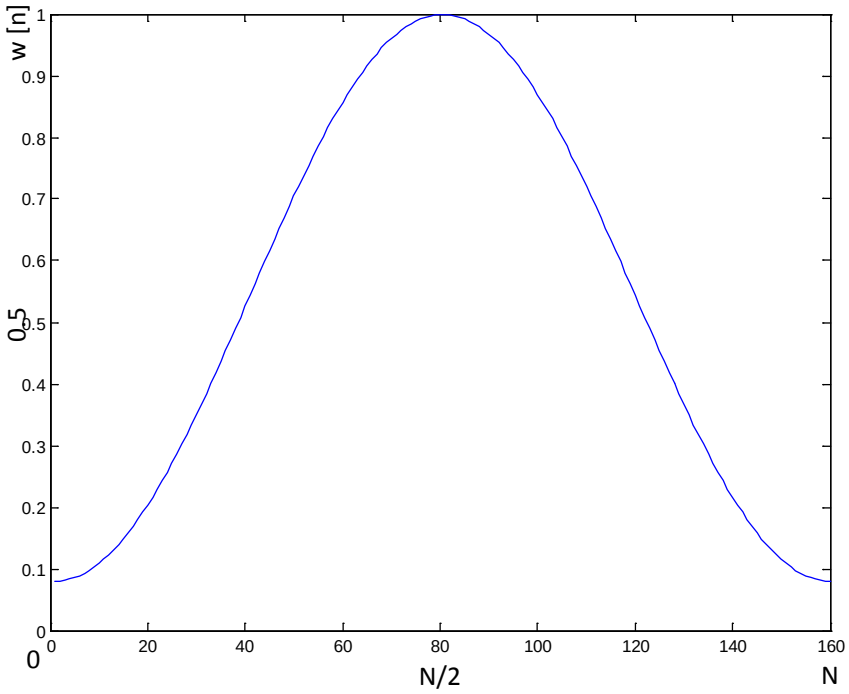
Pro váhování okénkem jsem zvolil Hammingovo okénko, které používám i u ostatních příznaků. Ostatní parametry jsem zvolil v „základním“ nastavení, které jsem chápal jako nejlepší pro svou úlohu [12].

Základní operací pro rozdělení zvukového souboru na rámce je váhování signálu okénkem. V případě pravoúhlého okénka je toto váhování rovno rozdělení na rámce. Proto jsem použil *Hammingovo okénko* (obr. 3.1), které „utlumí“ signál na okrajích rámce a zabrání tak rušivým přechodovým jevům. *Hammingovo okénko* $w(n)$ délky N je definováno takto (3.1):

$$w(n) = 0.54 - 0.46\cos(\frac{2\pi n}{N}), \qquad (3.1)$$

kde n je řečový vzorek $0 \leq n < N$.

Existuje velké množství váhových okének, jako například obdélníkové, Hanningovo, trojúhelníkové nebo Hammingovo. Pro náš případ bylo použito Hammingovo okénko.



Obr. 3.1 – Hammingovo okénko

3.2 Detekce energie v signálu

V této práci jsem použil energii signálu jako parametr, který popisuje každý rámec v promluvě, na jeho základě určujeme začátek a konec „užitečné“ řeči. Pro každý rámec dostáváme jeho úroveň energie. K vypočtení této hodnoty existuje několik metod výpočtu, jsou to:

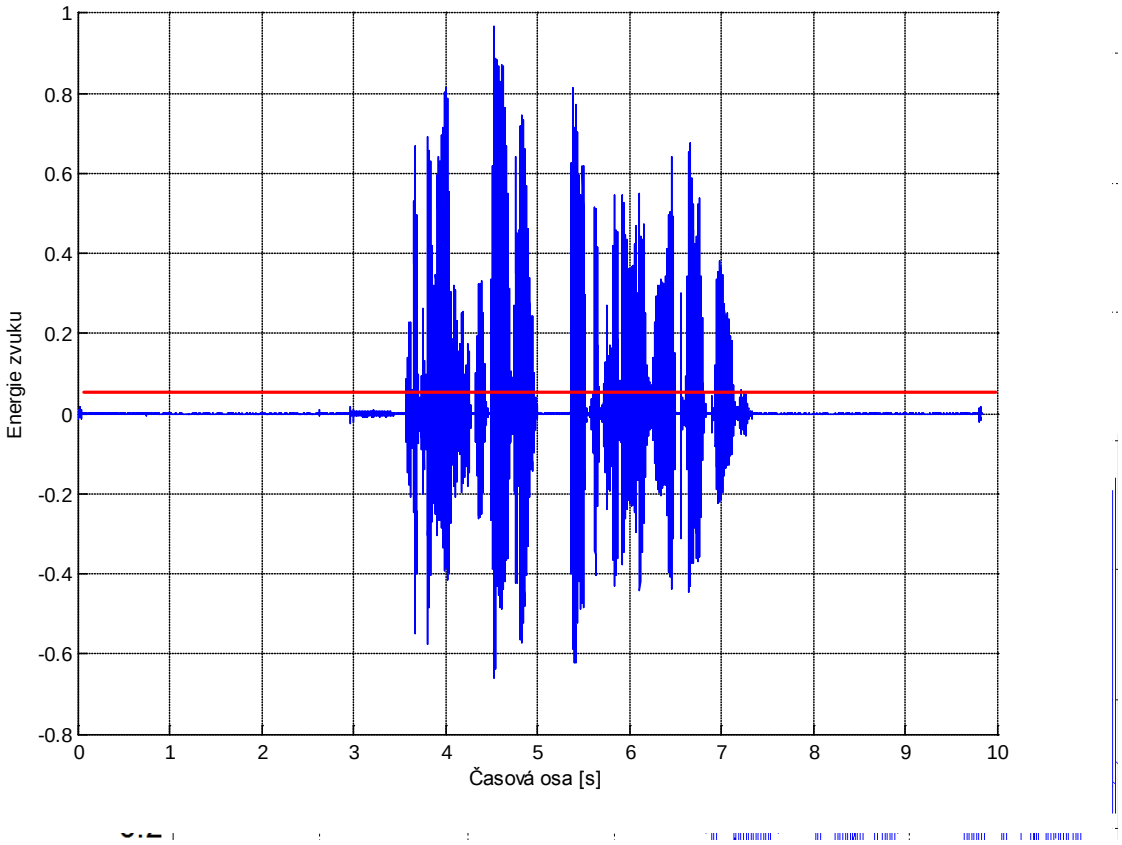
- krátkodobá energie signálu,
- logaritmus energie signálu,
- nultý kepstrální koeficient c_0.

Způsob vypočtení krátkodobé energie je uveden ve vzorci (3.2) Výpočet je prováděn v celém segmentu délky N vzorků [5].

$$E = \sum_{k=1}^{N} [s(k)w(n-k)]^2 \ , \tag{3.2}$$

kde *s(k)* je vzorek signálu v čase *k* a *w(n)* reprezentuje Hammingovo okénko.

Již v základním zadání Projektu 5 [8] jsem řešil otázky detekce začátku řeči v promluvách. Pro detekci byla použita metoda využívající jednoduchou prahovou detekci energie, jak je znázorněno na obr. 3.2. Na tomto obrázku je práh detekce znázorněn červenou čarou. Výsledky této metody jsou popsány v kapitole 7.



Obr. 3.2 – Detekce prahové energie

3.3 Krátkodobá funkce středního počtu průchodů signálu nulou

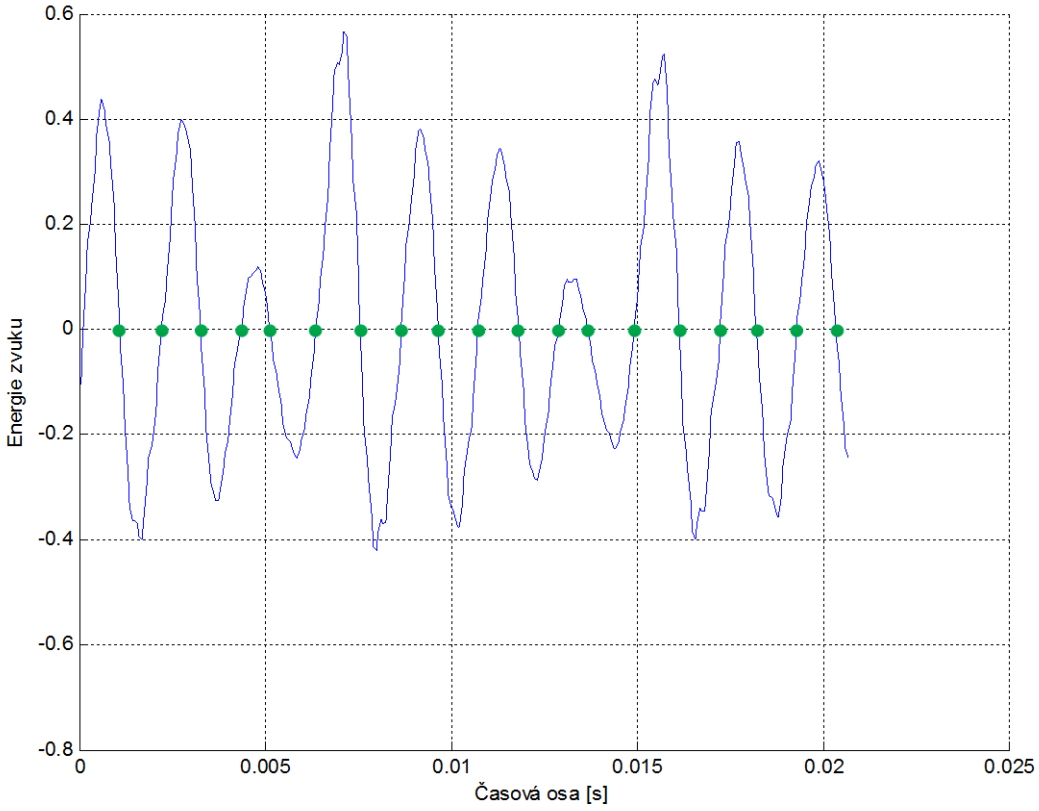
Frekvenci průchodů signálu nulovou úrovní můžeme chápat jako jednoduchou charakteristiku popisující spektrální vlastnosti signálu. Většinou se jedná o doplňkovou charakteristiku energetických detektorů a přesně takovéto využití našla i v Projektu 5, kde jsme jí využívali pro zlepšení vyhodnocování v detekci řeči [8]. Výsledky této metody jsou zobrazeny v kapitole 7. Hodnoty počtu průchodů nulou (zero crossing rate, ZCR) ovšem velmi závisí na úrovni šumu v nahrávce. Zde dochází ke sbližování hodnot pro šum a pro neznělé hlásky. Je proto potřeba využít dalších doplňkových příznaků pro lepší výsledky [6]. Krátkodobou funkci středního počtu průchodů signálu nulou lze definovat jako

$$Z_n = \sum_{k=-\infty}^{\infty} |sgn[s(k)] - sgn[s(k)]|\, w(n-k) \ , \tag{3.4}$$

kde

$$sgn[s(k)] = \begin{cases} 1 & pro \quad s(k) \geq 0 \\ -1 & pro \quad s(k) < 0 \end{cases} \tag{3.5}$$

a w(n) je Hammingovo okénko [1].

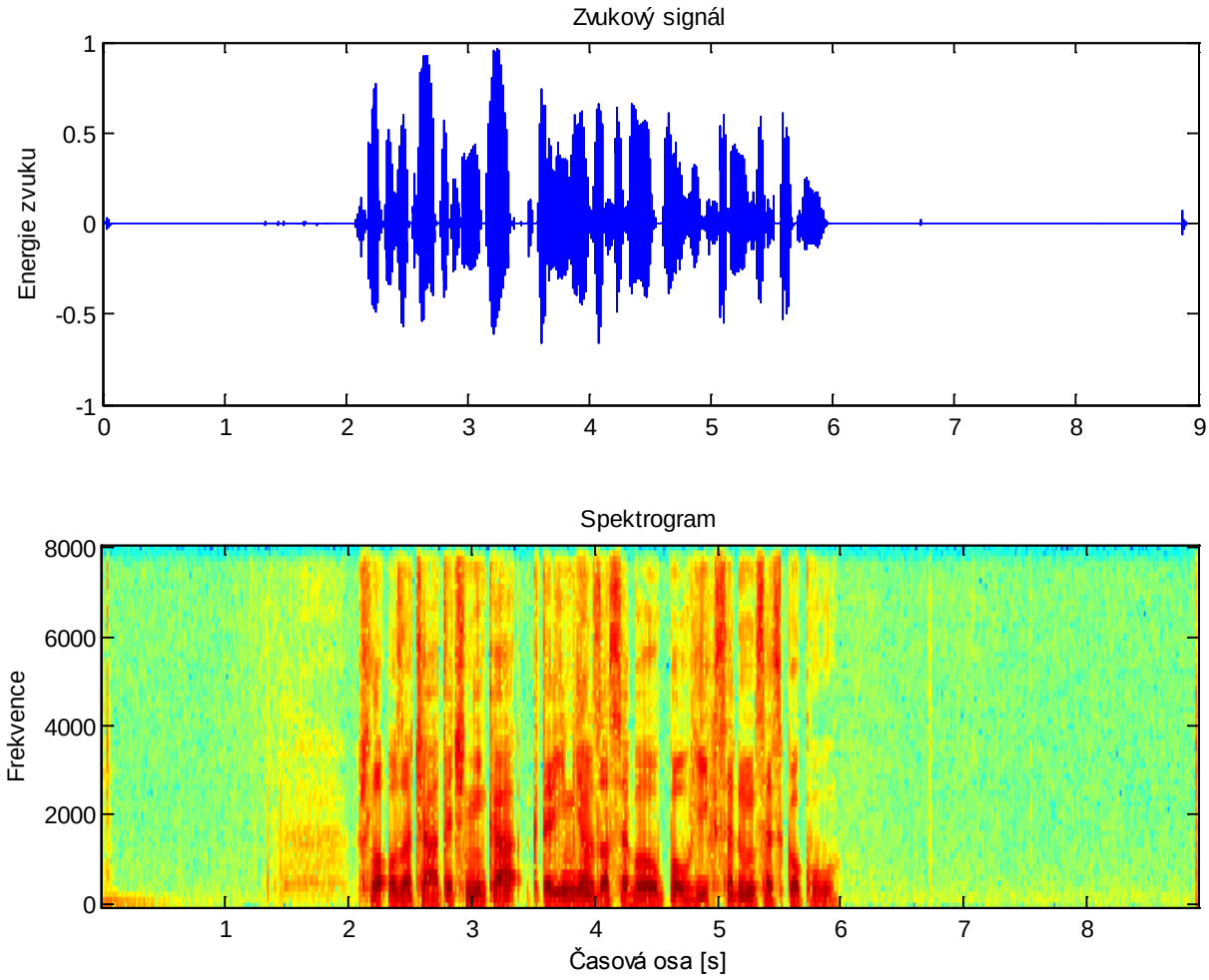


Obr. 3.3 – Zobrazení průchodů nulou v signálu zvuku

3.4 Informace o znělosti a neznělosti

Znělost je fonetická a fonologická vlastnost hlásek. Z akustického hlediska se jedná o znázornění základní frekvence zvuku F_0, která je dána aktivní činností hlasivek neboli fonací. Aktivní činnost hlasivek se požívá při vytváření znělých hlásek. U znělých hlásek jsou hlasivky napjaté a kmitají. U neznělých hlásek jsou hlasivky v klidu, tj. neúčastní se tvoření hlásky. Znělost je nejčastějším rozlišovacím fonologickým znakem u souhlásek, samohlásky jsou ve většině jazyků (včetně češtiny) pouze znělé, což není považováno za samostatný

foném. Neznělé samohlásky se vyskytují jen v některých jazycích (např. v japonštině) [7]. V zadané práci jsem znělost a neznělost využil jako příznak, podle kterého byla vyhodnocována data a zpřesňovány získané výsledné hodnoty počátečního a koncového času v promluvách.



Obr. 3.4 – V horním grafu je znázorněn zvukový signál a ve spodním grafu je spektrogram, který ukazuje „množství energie“. V červených oblastech s nižší frekvencí je znázorněn harmonický zvuk, který je tvořen samohláskami, jež jsou znělé.

3.5 Spektrální parametry LSF

V mojí práci jsem použil spektrální parametry LSF (Line Spectral Frequencies), kterými jsem rozšířil příznaky pro klasifikaci; jsou popsané v kapitole 8.4.3. Před získáním parametrů LSF musíme nejdříve projít kroky, jež jsou popsány v kapitolách 3.5.1, 3.5.2 a 3.5.3.

Popis hlasového signálu pomocí LSP (Line Spectral Pairs) je založen na válcovém modelu hlasového traktu, který můžeme modelovat pomocí sady válců stejné délky, ale různého průměru. Válce jsou do sebe zasunuté. Pokud bude těmito válci proudit vzduch, bude docházet k různým rezonancím v závislosti na tom, zda bude tento model na konci otevřený nebo uzavřený. Toto bude simulovat otevření a uzavření hlasivek, ale i úst atp. Počet rezonančních frekvencí závisí na počtu válců, kterými je hlasový trakt modelován, tedy na řádu modelu. Tento model je často popisován pomocí lineární prediktivní analýzy (Linear Predictive Coding - LPC). Pozice a šířky rezonančních frekvencí je možné popsat právě pomocí LSF, které přímo souvisí s LSP.

LSP je polynom *k*-tého řádu, jehož komplexní kořeny Θ_k jsou LSF. Pokud tyto kořeny seřadíme, dostaneme páry čísel (sudý a lichý kořen) – frekvencí, které popisují umístění a šířku rezonančních frekvencí hlasového traktu [11].

3.5.1 Výpočet LPC

Vlastnosti LPC vychází z toho, že je možné aproximovat *n*-tý vzorek signálu *x* jako lineární kombinaci *M* předchozích vzorků

$$\hat{x}[n] = \sum_{m=1}^{M}(a_m x[n-m]) , \tag{3.6}$$

kde $\hat{x}[n]$ je odhad *n*-tého vzorku a a_m je váha daného předchozího vzorku. Výsledný LPC model je popsán polynomem

$$A(z) = 1 + a_1 z + a_2 z^2 + \cdots + a_M z^M , \tag{3.7}$$

což je model *M*-tého řádu [11].

3.5.2 Výpočet LSP

Pokud zavedeme dva polynomy *P(z)* a *Q(z)* řádu *(M+1)*, které jsou asymetrické a mají k polynomu *A(z)* vztah

$$A(z) = \frac{\mathrm{P(z)+Q(z)}}{2}, \tag{3.8}$$

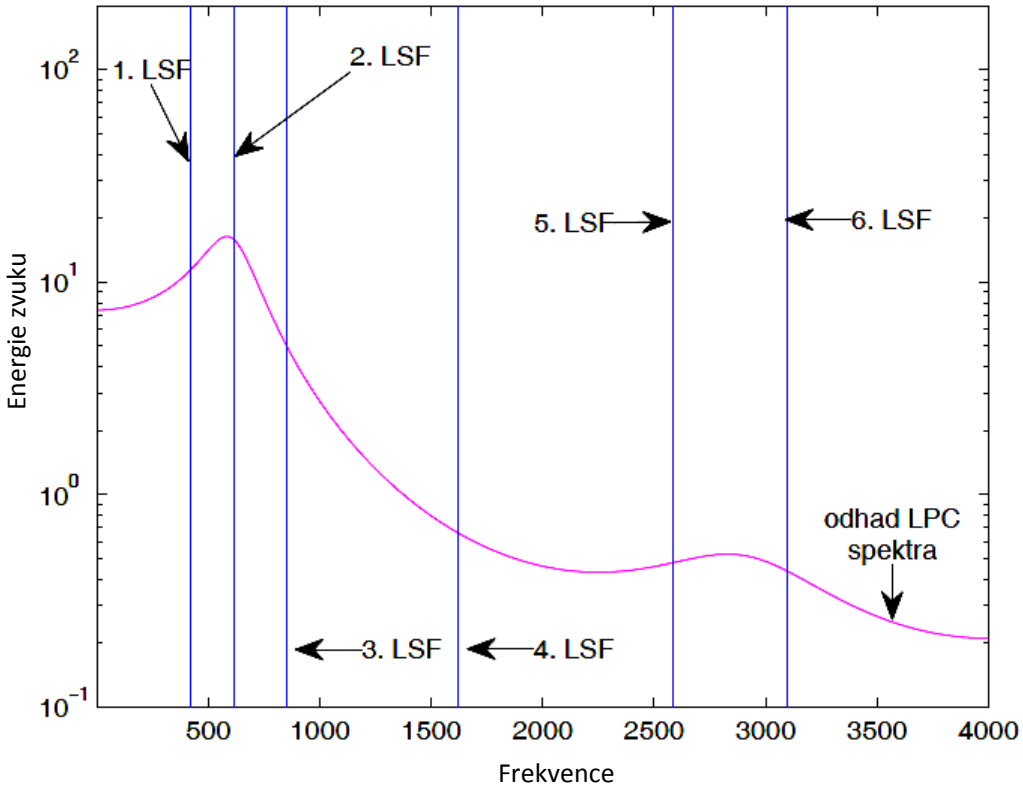
získáme LSP polynomy *P(z)* a *Q(z)*. Koeficienty těchto polynomů lze získat ze vztahů

$$\mathrm{P(z)} = A(z) - z^{-(M+1)}A(z^{-1}), \tag{3.9}$$

$$\mathrm{Q(z)} = A(z) + z^{-(M+1)}A(z^{-1}), \tag{3.10}$$

3.5.3 Výpočet LSF

Komplexní kořeny Θ_k LSP polynomů, tedy LSF, leží v *z* rovině na jednotkové kružnici a jsou navzájem proložené. LSF jsou obvykle v LPC spektru zobrazeny svislými čarami, viz obrázek 3.5 [11].



Obrázek 3.5 – Ukázka odhadu spektra pomocí LPC modelu 6. řádu pro hlásku *a* s vyznačenými LSF [11]

3.6 Spektrální parametry MFCC

Při rozpoznávání řeči pomocí spektrálních parametrů MFCC (Mel Frequency Cepstral Coefficients) se nepoužívá řečový signál ve své původní podobě, ale převádí se na méně redundantní popis, který je obvykle reprezentován vektory parametrů. V implementaci pro příznaky klasifikátoru jsem použil melovské kepstrální koeficienty (MFCC). Tato parametrizace se často využívá v aplikacích s rozpoznáváním řeči. MFCC modelují spektrální rozlišení u člověka a mají dobrou odolnost proti kvantizačnímu zkreslení. Kromě toho je tu možnost zotavení z vlivu přenosového kanálu. Vložení takového kanálu do cesty se projeví ve frekvenční oblasti násobením spektra signálu přenosovou funkcí kanálu. V kepstrální oblasti se násobení transformuje na sčítání. Tento vliv může být odstraněn odečtením průměrného kepstrálního vektoru získaného ze vstupních vektorů např. klouzavým průměrováním. To je v praxi použitelné při kompenzaci vlivů dlouhodobého charakteru, např. změna mikrofonu.

Celý výpočet od signálu po výsledné parametry se skládá z následujících kroků: segmentace, preemfáze, váhování okénkem, rychlá Fourierova transformace (FFT), filtrace melovskou bankou filtrů, logaritmus, diskrétní kosinová transformace (DCT) [12].

3.7 Formantové frekvence

Formantové frekvence rovněž našly uplatnění v mé práci, když bylo potřeba další příznaky pro klasifikátor, aby se zpřesnila jeho funkčnost. Je známo, že první tři formantové frekvence nesou důležitou informaci o charakteru samohlásek a znělých souhlásek a podle průběhu formantových frekvencích lze v čase určit i místo artikulace sousedních hlásek. Informace o formantech je nejprokazatelněji obsažena ve spektrální obálce analyzovaného úseku řeči. Většina postupů identifikace frekvencí formantů buď implicitně, nebo explicitně využívá právě spektrální obálky. I když se na první pohled zdá tato úloha velice jednoduchá, objevují se i zde problémy, které značně znesnadňují její řešení.

Dva hlavní problémy jsou:

- Výskyt nepravých vrcholů ve spektrální obálce – Maxima ve spektrální obálce jsou normálně způsobena formanty. Nicméně se zde mohou objevit i další nepravé vrcholy, které jsou vyvolány pouze různými poruchami. V případě, že spektrální obálka je určována metodou LPC, jsou tyto poruchy obyčejně způsobeny tím,

že řád prediktoru bývá předimenzován a prediktor může u nadbytečných pólů v určitých případech nahodile vytvořit nepravé vrcholy.

- Splývání formantů – Jeden z nejobtížnějších případů při identifikaci formantů nastává, když dvě formantové frekvence jsou tak těsně blízko vedle sebe, že individuální špičky ve spektrální obálce splývají a nelze je od sebe jednoduše odlišit.

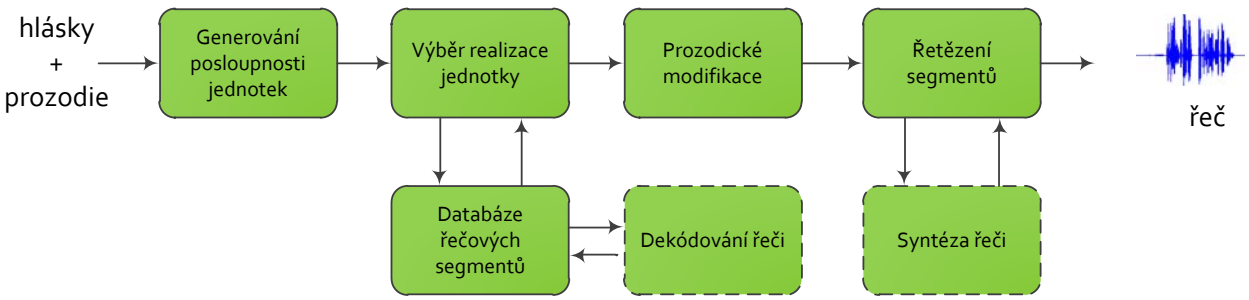
Většina postupů pro identifikaci formantových kmitočtů pracuje ve frekvenční oblasti a vychází z analýzy spektrální obálky stanovené metodou LPC [1].

4 Akustická syntéza řeči

Akustická syntéza řeči je proces či technika vytváření řečového signálu, který je velice příbuzný zadání mé práce. Jde o ještě důkladnější a důmyslnější rozpoznávání řeči, kde lze rozlišit i samotná písmena, která následně můžeme strojově spojit a vytvořit z nich řeč. Systémy akustické syntézy řeči samy o sobě nabízejí široké pole uplatnění, ať už v oblastech, kde jiný než hlasový způsob komunikace nepřichází v úvahu, ale i tam, kde možnost hlasové komunikace výrazně obohatí kvalitu dané lidské činnosti. Syntetizovaná řeč může nahradit skutečného lidského řečníka na širokém spektru různých pozic - od rutinního oznamování opakujících se informací (zastávky MHD, nádraží apod.), přes hlasový monitoring údajů (řídicí střediska), informační a dialogové systémy (automatická spojovatelka, telefonní klientské či informační linky), až po vysoce propracované a přirozené čtení libovolných textů (e-maily, SMS, ale i celé knihy). V současné době nelze též syntéze řeči upřít stoupající uplatnění v zábavním průmyslu.

Cílem akustické syntézy řeči je vytvářet řeč, a to v takové formě a kvalitě, aby obvykle co nejvěrněji kopírovala řečové charakteristiky konkrétního člověka; tedy nejen samotný hlas a jeho kvalitu, ale i styl mluvení atd. K automatickému vytváření řeči se využívá technologie syntézy řeči z textu (z anglického text-to-speech, TTS) - nejobecnější a také nejtěžší úloha syntézy řeči, jejímž úkolem je převést libovolný text na odpovídající řeč. Jde o sadu speciálních modulů a algoritmů, které zajišťují automatický převod psaného textu na mluvenou řeč. Zahrnují zpracování textu (např. analýza a normalizace), převod textu do výslovnostní podoby (tj. fonetickou transkripci a generování průběhů prozodických vlastností řeči), tvorbu inventáře akustických jednotek a vlastní metodu vytváření řeči [3].

4.1 Vytváření řeči



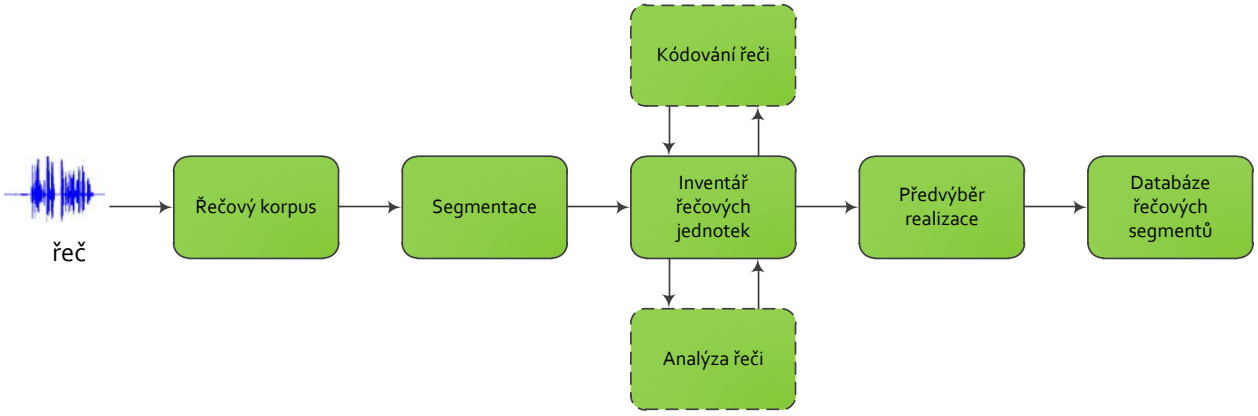
Obr. 4.1 – Schéma vytváření řeči

Pro potřeby českého systému syntézy řeči, který se používá dle schématu na obr. 4.1, je na katedře kybernetiky Západočeské Univerzity v Plzni vyvíjena moderní metoda vysoce kvalitní syntézy řeči. Systém je založen na tzv. konkatenační syntéze řeči. Stručně řečeno, základním principem tohoto přístupu je reprezentace důležitých akustických událostí lidské řeči pomocí tzv. řečových jednotek či segmentů řeči. Výsledná řeč pak vzniká konkatenací, tj. řetězením těchto řečových jednotek. Vhodnými řečovými jednotkami jsou přitom jednotky subslovní, např. hlásky nejčastěji posazené do kontextu okolních hlásek - tzv. trifony nebo difony (zjednodušeně řečeno jde o jednotky začínající v polovině jedné hlásky a končící v polovině hlásky následující) [3].

4.2 Příprava databáze řečových jednotek

Úspěšná konkatenační syntéza spočívá v pečlivé přípravě inventáře řečových jednotek – tj. segmentů řeči, se kterými syntetizér řeči pracuje. Jelikož kvalita výsledné syntetické řeči do značné míry závisí na bohatosti řečových segmentů obsažených v inventáři a na přesnosti, s jakou jsou tyto segmenty extrahovány z řečových promluv, používá se metoda automatické konstrukce inventáře na základě velkého množství reálných řečových promluv. Automatizace je důležitým aspektem systému, neboť umožňuje v krátkém časovém horizontu (řádově několik dnů) vytvořit velice precizní a akusticky a lingvisticky „bohaté“ inventáře akustických jednotek, které pak do značné míry přispívají k vysoké kvalitě vytvářené řeči. Jde o tzv. korpusově orientovanou konkatenační syntézu řeči, neboť právě řečový korpus (tj. sada

reálných řečových promluv vyslovených jedním řečníkem, jehož hlasem pak syntetizér řeči mluví, a jejich reprezentace v ortografické, fonetické, spektrální či prozodické oblasti) je základním materiálem pro vytvoření inventáře řečových jednotek [3].



Obr. 4.3 - Schéma vytváření databáze řečových segmentů

Výrazným kritériem kvality je přirozenost vytvářené syntetické řeči. Přirozenost řeči přitom do značné míry závisí na kvalitě modelování, tj. na melodii promluvy, hlasitosti a trvání jednotlivých segmentů řeči.

5 Strojové učení

Strojové učení je důležité pro objasnění práce s klasifikátorem, který s mojí prací velmi úzce souvisí a objasním ho v následující šesté kapitole. Strojové učení je podoblastí umělé inteligence, zabývající se algoritmy a technikami, které umožňují počítačovému systému učit se. Zde je učení myšleno jako změna vnitřního stavu systému, která zefektivní schopnost přizpůsobení se změnám okolního prostředí. Strojové učení je dovednost inteligentního systému měnit svoje znalosti tak, že příště bude vykonávat stejný nebo podobný úkol efektivněji.

Klasifikátor vytváří inteligentnější rozhodnutí založené na přijatých datech. Tato data bývají v rámci učení obsahem trénovací množiny dat. Po natrénování inteligentního systému je nutné ověřit, jestli je tento systém dobře zkonstruovaný a jestli vytváří uspokojivé výsledky. Tento proces se nazývá testování a děje se na testovacích datech, která ještě klasifikátor neviděl. Na testovací množině si ověříme správnost a úspěšnost našeho učícího algoritmu.

Díky učení by se měla zvyšovat výkonnost našeho systému. Strojové učení lze rozdělit do čtyř základních kategorií [4]:

- učení s učitelem (*Supervised learning*)
- učení bez učitele (*Unsupervised learning*)
- učení posilováním (*Reinforcement learning*)
- kombinace učení s učitelem a bez učitele (*Semi-supervised learning*)

6 Klasifikace

Jednou z důležitých částí mé práce byl vývoj návrhu klasifikátoru pro detekci řeči. Pro algoritmus detekce řeči využívám klasifikátor s vhodnými příznaky, které jsem navrhnul a aplikoval. Použití klasifikátoru bude popsáno v kapitole 8. Pod pojmem klasifikace se nerozumí jenom samotný proces třídění dat. Jedná se o sofistikovaný systém několika různých procesů, jež na sebe navazují. Základní klasifikační systém se skládá z částí, kterými jsou získávání dat, extrakce příznaků a samotná klasifikace. Klasifikace se využívá napříč veškerou lidskou činností [4].

6.1 Klasifikátor

Klasifikátor je algoritmus, kterému jsou poskytnuty ukázky, jak řešený problém vypadá, respektive jak vypadají jednotlivé třídy, které potřebujeme rozlišit. Snažíme se klasifikátor naučit, jak příchozí data rozdělit do jednotlivých tříd. Fázi učení také jinak nazýváme trénování a v této fázi se snažíme najít rozdělující přímku, která co nejlépe od sebe oddělí prvky dvou a více odlišných tříd. Rozdělující přímka musí být zvolena tak, aby klasifikátor dobře rozhodoval na datech, která ještě neviděl.

Fáze testování se provádí na odlišných datech než trénování. Testovací data jsou taková data, která ještě nebyla klasifikátorem viděna a slouží k ověření toho, že klasifikátor na vstupních datech, která budou přicházet, bude dobře fungovat. V procesu testování klasifikátoru musíme dosažené výsledky analyzovat, a pokud jsou správné, můžeme být s prací klasifikátoru spokojeni [4].

6.2 Generalizace

Jde o schopnost klasifikátoru správně zpracovat data, která nebyla použita v procesu učení. Když máme špatně určenou rozdělující přímku, může sice dobře rozdělovat trénovací data, ale nová testovací data nebudou dobře klasifikována. Špatně zvolená rozdělující přímka nám prakticky znemožní nebo drasticky sníží generalizaci. Když klasifikátor dobře generalizuje, správně rozpoznává to, co ještě neviděl [4].

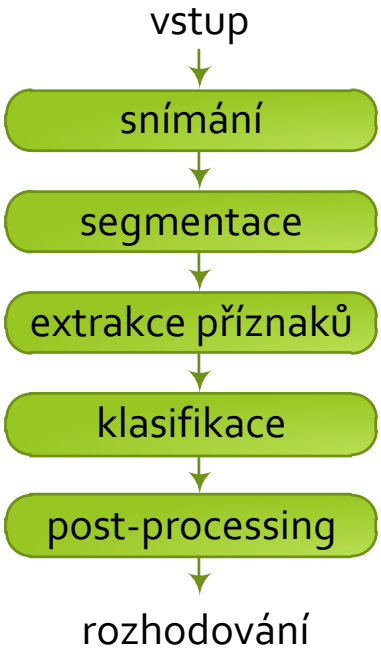
6.3 Přetrénování

Přetrénování nebo také přeučení. Při příliš velkém množství podobných trénovacích dat klesá chyba na trénovacích, ale roste chyba na testovacích datech. Přetrénování může být také způsobeno příliš přesnými příznaky, které byly získány ze vstupních dat v rámci trénování. Systém potom velmi dobře nebo dokonale rozpozná již jednou viděná data, ale hůře klasifikuje data, která ještě nikdy neviděl [4].

6.4 Práce klasifikátoru

Činnost klasifikátoru není jenom o tom, jak nějakým způsobem klasifikovat data, ale na celkovém průběhu klasifikace se podílejí ještě další procesy mimo samotné klasifikace. Data musíme nejprve získat a upravit, aby mohla být zpracována samotným klasifikátorem. Postup klasifikace je znázorněn na obrázku 6.4 a jednotlivé části celého procesu jsou [4]:

- vstup dat
- snímání
- segmentace
- extrakce příznaků
- klasifikace
- post-processing
- rozhodování



Obr. 6.4 – Znázornění práce klasifikátoru

6.4.1 Vstup

Vstupem do klasifikátoru jsou data, která chceme roztřídit do požadovaných tříd, pokud se jedná o klasifikaci, nebo shluků, pokud se jedná o shlukování. Vstupní data můžeme rozdělit na trénovací, testovací.

Trénovací data by měla dostatečně (komplexně) reprezentovat řešenou problematiku (tj. „reálný svět“). Slouží k „naučení“ klasifikátoru. Na těchto datech klasifikátor vidí, jak by asi mohla vypadat data, která dostane v rámci testování. Klasifikátor si zapamatuje vyznačené vlastnosti těchto dat a tyto informace potom využívá k rozřazování ještě neviděných dat. Toto mohou být jen data testovací.

6.4.2 Snímání

Snímání slouží pro vlastní získání dat. Data, která dále pokračují do dalších částí klasifikátoru, musí být nějak získána. Způsob, kterým data získáváme, je závislý na tom, jaká data budeme klasifikovat. Když chceme klasifikovat obrazová data, snímacím zařízením bude kamera nebo fotoaparát. Pro záznam zvukových dat použijeme mikrofon. Existuje ještě mnoho speciálních detektorů všemožných veličin pro různá použití.

6.4.3 Segmentace

Segmentace nám z celkových dat vyřeže jenom takové části, které potřebujeme a které jsou důležité.

6.4.4 Extrakce příznaků

Pokud chceme klasifikovat zvuk nebo i obrázky, potřebujeme z jednotlivých souborů získat takové informace, díky kterým budeme schopni udělat vlastní klasifikaci. Z velkého objemu dat se snažíme vybrat co nejmenší počet parametrů, které budou pořád dobře popisovat to, co naše data reprezentují. Jde vlastně o redukci vlastností daného problému. Příznaky nám umožní rozlišovat mezi jednotlivými třídami. V našem případě jsme za příznaky zvolili energii (v kapitole 3.2), ZCR (v kapitole 3.3) a informaci o znělosti a neznělosti (v kapitole 3.4), závislost rámce na předchůdci a následovníku (v kapitole 8.4.1), dynamické koeficienty (v kapitole 8.4.2), spektrální koeficienty (v kapitole 8.4.3 a 8.4.4) a formantové frekvence (v kapitole 8.4.5). Toto je proces předzpracování vstupů do samotného klasifikátoru.

6.4.5 Klasifikace

Klasifikace je proces zařazování neznámých objektů do tříd. Objekty klasifikátor rozřazuje s určitou úspěšností na základě poznatků, které získal ve fázi trénování.

6.4.6 Post-processing

Post-processing je jednou z fází, kde můžeme dolaďovat výsledky klasifikace v rámci kontextu využití dalších informací. Je to konečná úprava výsledků, které přicházejí z procesu klasifikace. I když máme data klasifikována a rozdělena do skupin, jejich zařazení ještě nemusí být naprosto finální. V tomto procesu můžeme dát našim výsledkům pravou váhu, podle které se nakonec budeme rozhodovat. Používáme různé způsoby ohodnocení výsledků, které jsme získali z klasifikací, například může jít o pravděpodobnostní vyjádření. V našem řešení post-processing využíváme, až když klasifikátor rozhodne, že první část už je užitečný zvuk - pomocí další úpravy výsledků určíme, že zvuk začíná až po pěti po sobě jdoucích částech.

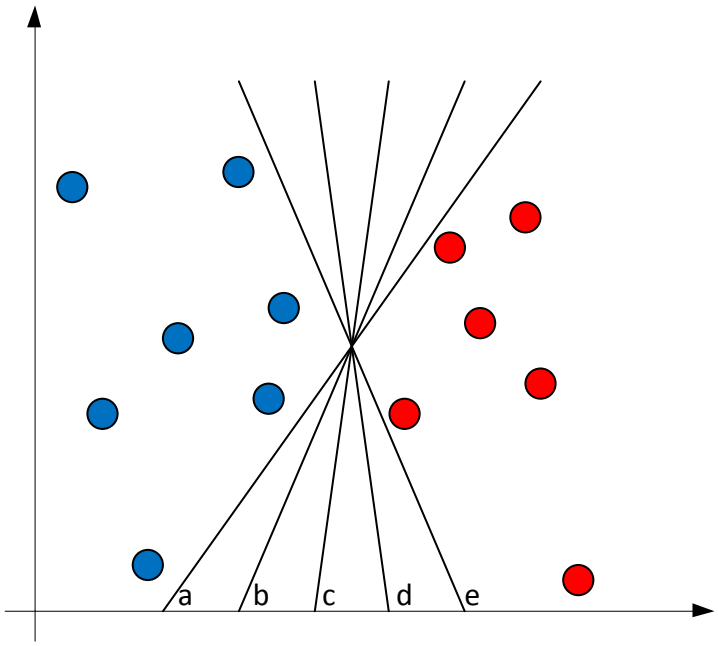
6.4.7 Rozhodování

V tomto procesu již konečným způsobem rozhodneme, kam daný vzorek zařadíme a jak s ním budeme dále pracovat.

6.5 Metoda podpůrných vektorů

Jedná se o skupinu příbuzných metod strojového učení s učitelem, které se používají pro klasifikaci a regresi, kterou jsme právě zvolili jako nejvhodnější pro detekci zvuku v promluvách. Následná ukázka je znázorněna v kapitole 8. Jako další metody můžeme zmínit Bayesovo kritérium, lineární diskriminační funkci, klasifikaci podle minimální vzdálenosti, klasifikaci podle nejbližšího souseda a další. V této práci jsem zvolil metodu podpůrných vektorů (support vector machine, SVM), která konstruuje nadrovinu nebo množinu nadrovin (obr. 6.5) ve vícedimenzionálním prostoru, který může být použit pro klasifikaci, regresi nebo jiné úlohy. Klasifikace se snaží zařadit daný objekt do určité výstupní třídy na základě poznatků, jež byly získány ve fázi učení. Regrese určuje hodnotu jedné proměnné v závislosti na jedné nebo více dalších proměnných. Metoda SVM se využívá pro kategorizaci textů, rozpoznávání obrazů nebo také v lékařství.

Základní princip SVM je převod klasifikovaných prvků původního prostoru do vícedimenzionálního prostoru, ve kterém už je možné jednotlivé třídy prvků oddělit lineárně. Tomuto principu se říká „*Jádrová funkce*“ [4].

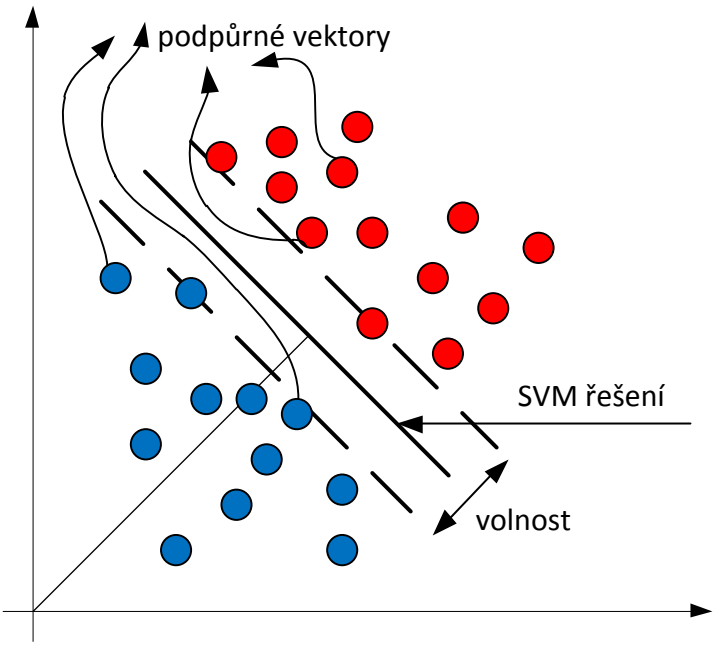


Obr. 6.5: - Určení dělících rovin, podle kterých se mohou data rozdělit do více dimenzí (přímky a,b,c,d,e znázorňují možné dělící roviny)

Díky použití dalšího prostoru přibude každému prvku další souřadnice, která tyto prvky posune. Tímto posunem se oddělí klasifikované prvky, které už lze lineárně oddělit pomocí nadroviny, jak je zobrazeno na obrázku 6.5.1. Při transformaci např. dvojrozměrného prostoru do trojrozměrného je třetí souřadnice prvků závislá na prvních dvou souřadnicích.

6.5.1 Jádrová funkce

Jádrová funkce umožňuje mapování prvků základního prostoru a následný přepočet do prostoru s více dimenzemi. V původním prostoru nejsme schopni data od sebe oddělit dělící přímkou, proto je musíme pomocí této metody převést do vícerozměrného vektorového prostoru, kde již lineární oddělovač existuje a kterým může být například rovina nebo nadrovina [4].



Obr. 6.5.1 – Princip Support Vector Machine

Využívá se v případech, kdy data nejsou dobře oddělitelná. Právě s lineárně neoddělitelnými daty si poradíme pomocí tzv. jádrové funkce. Případnou neoddělitelnost dat řeší zavedením tzv. relaxačních proměnných, které jsou nulové pro správně klasifikované vzory a nenulové pro špatně klasifikované vzory [9].

7 Detekce řeči v řečových promluvách pomocí prahové energie a počtu průchodů nulovou osou

Cílem této úlohy bylo co nejpřesnější určení začátku a konce „užitečného“ řečového signálu pomocí jednoduchých příznaků, a to podle energie řečového signálu a počtu průchodů nulovou osou. Tyto metody jsou popsány v kapitolách 7.1 a 7.2. V tabulce 7.2 vidíme, že odchylky začátku a konce mají různá znaménka. Detekování začátku s kladným znaménkem znamená, že klasifikátor detekoval začátek promluvy ještě před reálným začátkem. Se záporným znaménkem při detekci začátku znamená, že klasifikátor určil začátek promluvy příliš pozdě. Kladné znaménko u odchylky při určení konce promluvy naopak znamená, že se klasifikátor zpozdil oproti přesné reálné hodnotě.

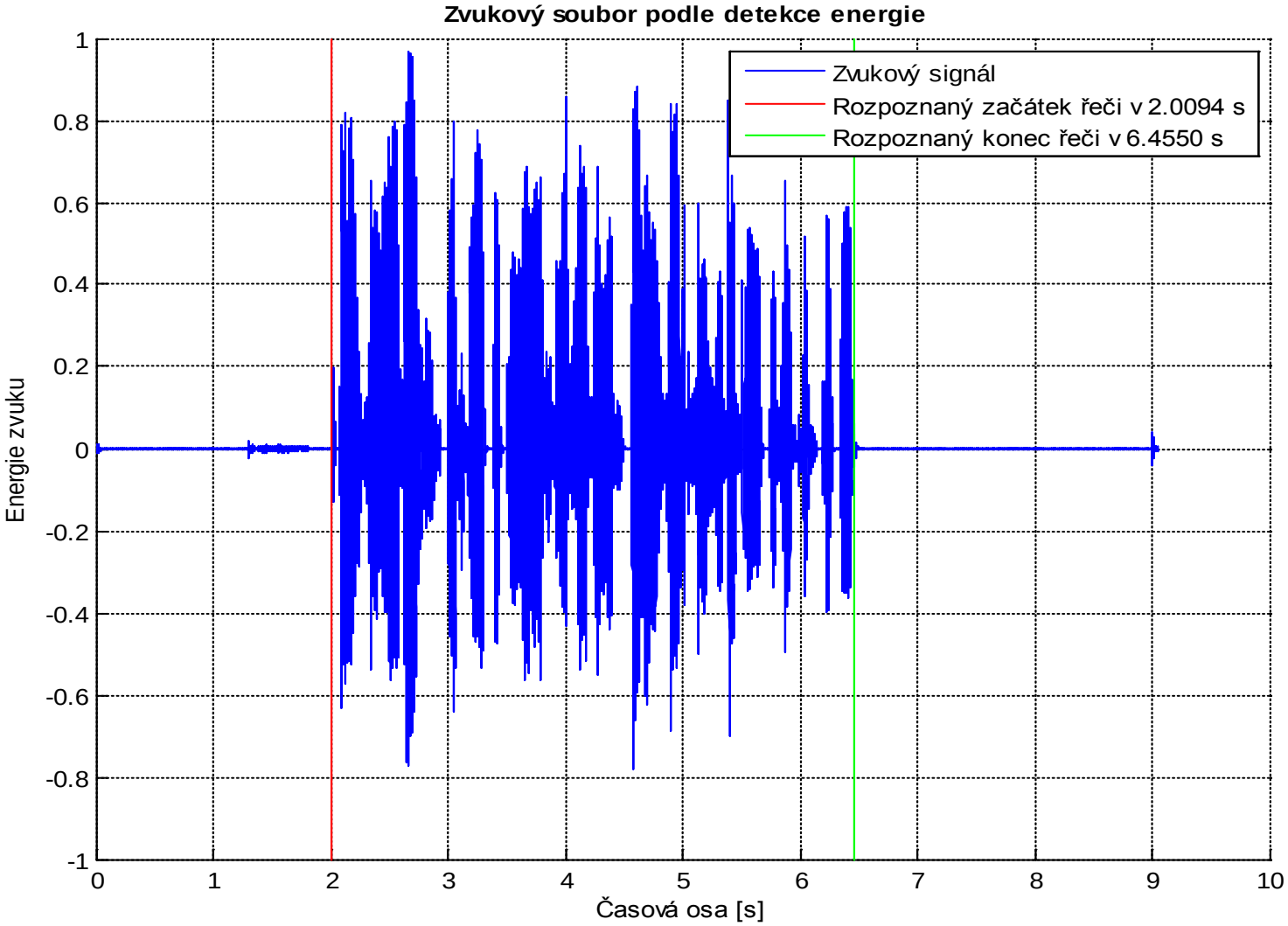
7.1 Algoritmus detekce energie

Algoritmus byl vytvořen v programovém prostředí Matlab, kde jsem úlohu zpracovával pomocí detekce prahování energie (kapitola 3.2). Nejdříve jsem načetl celý zvukový soubor, který bude následně zpracováván.

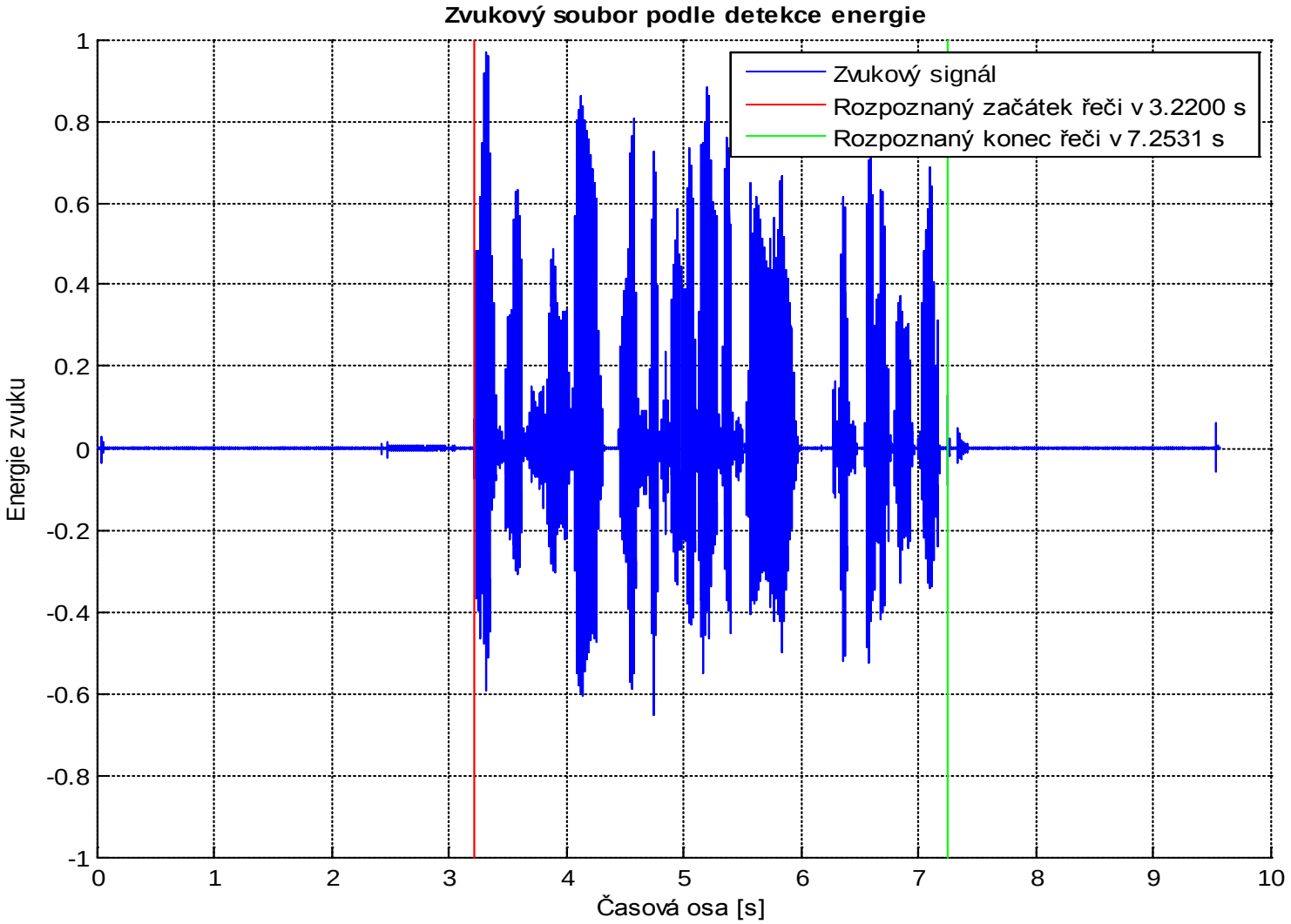
Zvukový soubor analyzujeme po částech, které se nazývají rámce. Velikost rámce jsem zvolil 10 milisekund. Tento rámec nesmí být moc krátký, protože pak by se nejednalo o spojitou detekci, ale o bodovou detekci s nepřesným určováním, velmi náchylnou na náhodný šum s velkou intenzitou. Zároveň rámec nesmí být moc dlouhý, protože pak bychom dostávali nepřesně detekované časy.

Při hledání počátku „užitečného“ signálu procházíme od začátku každý rámec a zjišťujeme jeho energii, jež nám říká, jestli je v daném rámci nějaká energetická aktivita, která může znamenat řeč nebo například šum. Když prohledávaný rámec překoná námi zadaný práh energie, je určen jako počátek hledaného signálu. Stejná detekce probíhá i od posledního rámce směrem k počátku. Tímto způsobem hledáme i koncový čas hledaného signálu. Tato detekce byla pro některé signály naprosto úspěšná, jak je vidět na obrázku 7.1.2. Většinou se jednalo o promluvy, které neměly v signálu žádné velké rušivé zvuky a hluky. Nejčastějšími nežádoucími zvuky byly zapínání/vypínání nahrávacího přístroje a nadechování řečníka. Zároveň existovaly promluvy, na kterých detekce fungovala velmi špatně a nebyla úspěšná, to nám znázorňuje obrázek 7.1.3. V tomto případě byla úroveň prahové energie příliš

vysoká a už nezachytila neznělé souhlásky na konci promluvy. Bylo potřeba zvolit rozumný kompromis mezi vysokým prahem energie, který nezachytí neznělé začátky a konce promluv s nízkou energií, a mezi nízkým prahem energie, který je hodně náchylný na rušivé zvuky a hluky, to znamená, že bude například detekovat začátek řeči už při nadechování řečníka. To byl velmi častý problém v algoritmu detekující energii v signálu. A tak jsem navrhl do tohoto algoritmu použít další příznak pro zpřesnění požadovaných výstupních časů. Toto rozšíření je popsané v následující kapitole 7.2.



Obr. 7.1.2 – Detekce pomocí energie se správnými výstupními hodnotami



Obr. 7.1.3 – Detekce pomocí energie s nesprávnými výstupními hodnotami: chyba je viditelná v detekci koncového času, prahová energie je malá a nedetekuje správně neznělé hlásky na konci promluvy. V tomto případě jde o neznělé *st*, v případě nastavení menšího prahu energie bychom správnou hodnotu ale opět nedostali, protože by byl zachycen šum na konci promluvy a následně detekován další chybný čas.

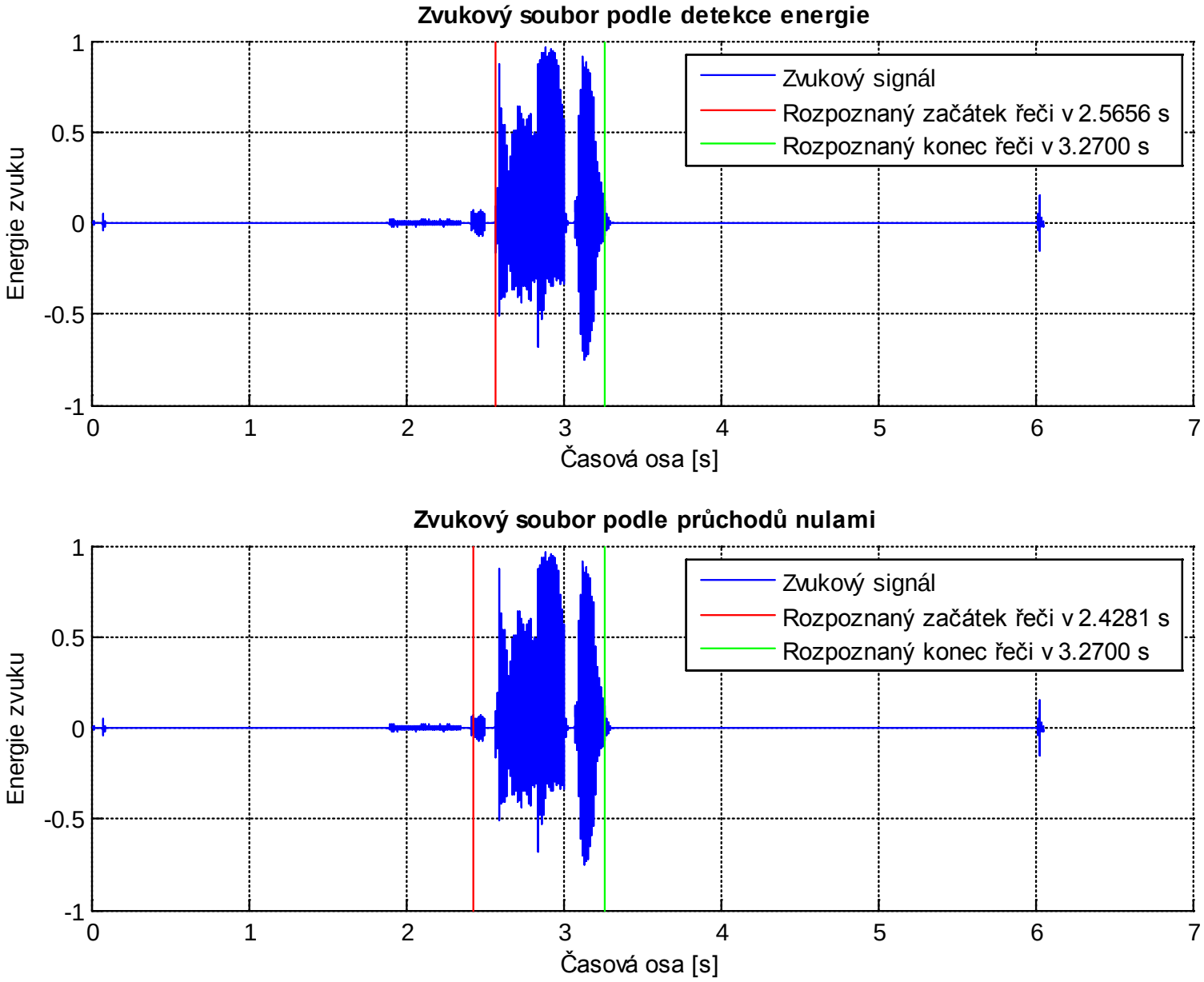
7.2 Detekce pomocí průchodů nulovou osou

Detekci pomocí průchodů nulovou osou jsem popsal v kapitole 3.3, jde o doplňkovou informaci k prahovým energiím pro automatickou detekci, která bude v ideálním případě vylepšovat přesnost detekce a bude získávat přesnější detekované hodnoty. V prostředí Matlab jsem využil předchozí prahovou detekci energie, tu jsem dále rozšířil o průchody nulovou osou.

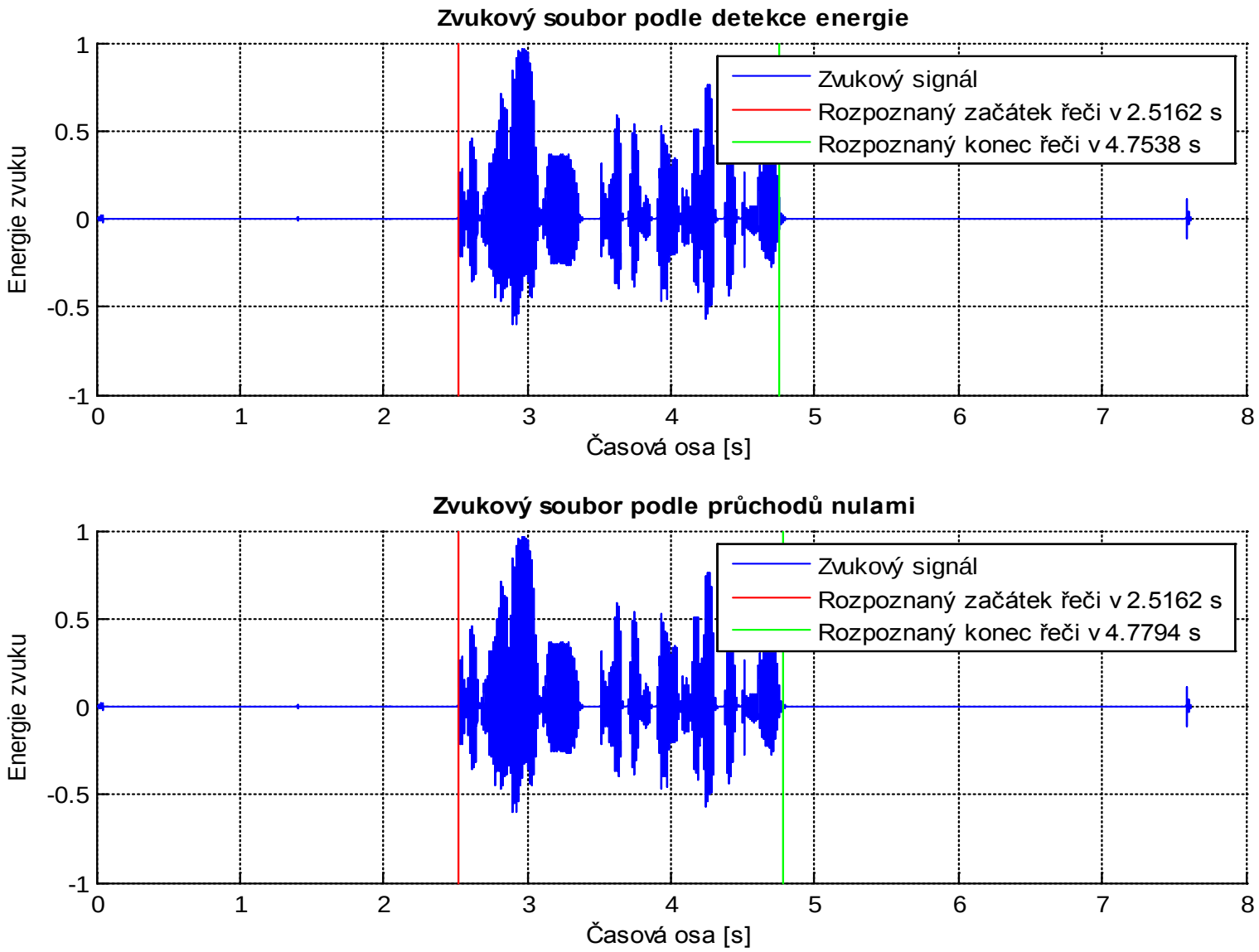
Pomocí testování jsem určil optimální počet průchodů pro detekci zvuku. Ovšem v nahrávkách se objevoval častý problém, který spočíval v nadechnutí řečníka. Při tomto nádechu došlo obrovskému nárůstu počtu průchodů nulovou osou a tím ke „znehodnocení“ této metody. Musel jsem opět nalézt funkční detekci průchodů nulovou osou, která bude správně hledat vyšší počet průchodů nulovou osou v začátku užitečné řeči a bude přibližně na podobném místě, jako začíná správná detekce podle prahové energie. Výsledkem mělo být zpřesnění detekce požadovaných časů začátku a konce „užitečného“ signálu v nahraných promluvách.

Detekce se zlepšila v případech, kde detekce podle prahové energie selhávala (obr. 7.2.1). Jednalo se většinou o případy, kde počáteční hlásku, kterou prahová detekce nezachytila, protože nebyla energeticky výrazná, ale byla znělá a měla relativně velký počet průchodů nulovou osou. Ovšem byly zaznamenány i případy, kde detekce podle energie a průchody nulovou osou vyhodnocovaly v obou případech správně. Kombinace těchto dvou metod měla přinést zlepšení, jaká jsou zobrazena na obrázcích 7.2.1 a 7.2.2. Takto by bylo dosaženo uspokojivého výsledku detekce těmito dvěma metodami. Vyhodnocení s takovýmto výsledkem pro celou množinu testovaných dat by bylo přínosné a využitelné v systémech s detekcí řeči. Správnou a nezměněnou detekci, jak podle prahové energie, tak podle průchodů nulovou osou si můžeme prohlédnout na obrázku 7.2.2.

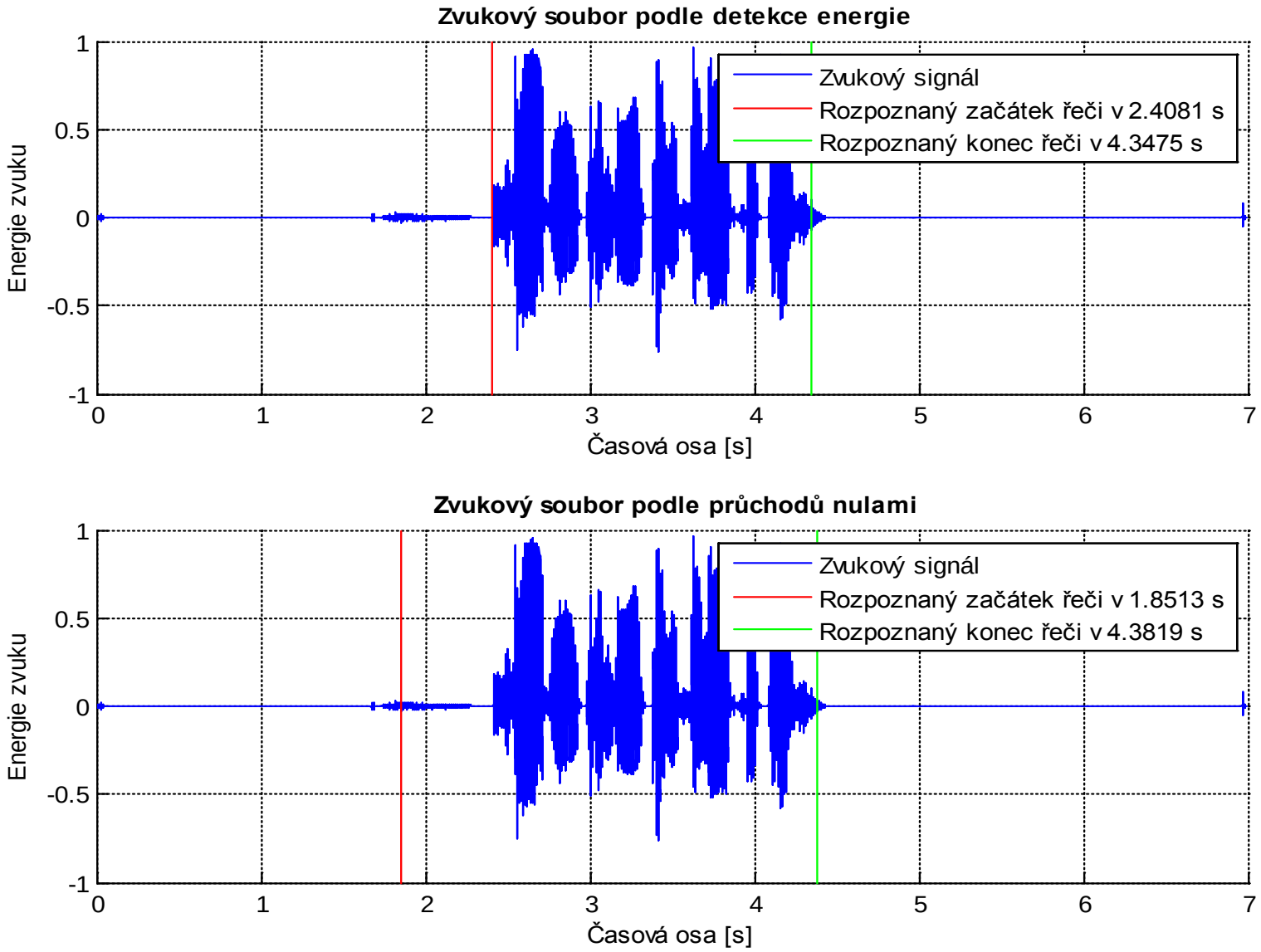
Bohužel detekce, která kombinuje prahovou energii a průchody nulovou osou, v některých případech nefungovala správně. Detekce podle průchodů nulovou osou má tu nevýhodu, že detekuje i šumy a ruchy, které mají velký počet průchodů nulovou osou. Proto kombinace detekce podle prahové energie a průchodů nulovou osou přináší špatné výstupní hodnoty. To je dobře vidět na obrázku 7.2.3. Jde většinou o případy, kde je v promluvách silný a dost slyšitelný nádech řečníka. Takový nádech má velký počet průchodů nulovou osou a občas i velkou energetickou hodnotu. Řešení, které kombinuje detekci podle prahové energie a počtu průchodů nulovou osou, nepodává zcela přijatelné výsledky.



Obr. 7.2.1 - Případ, kde došlo k chybné detekci podle energie, a po implementaci průchodů nulou jsme získali optimální výsledek



Obr. 7.2.2 - Případ, kde se správnost detekce nezměnila po implementaci průchodů nulou



Obr. 7.2.3. - Případ, kde došlo k chybě v detekci po implementaci průchodů nulou

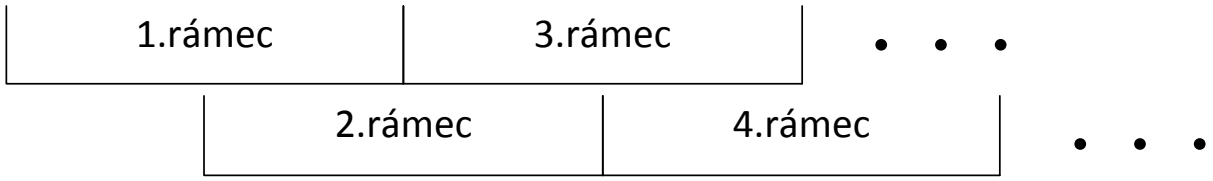
Promluvy	Délka věty [s]	Odchylka v určení začátku při použití prahové energie [ms]	Odchylka v určení konce při použití prahové energie [ms]	Odchylka v určení začátku při použití prahové energie a průchodů nulou [ms]	Odchylka v určení konce při použití prahové energie a průchodů nulou [ms]
věta 1	7,616	+1	+38	+1	+13
věta 2	9,845	-28	+119	-25	+75
věta 3	12,234	-23	+44	-23	+26
věta 4	13,557	-16	+61	-16	+46
věta 5	12,810	-32	+53	-25	+29
věta 6	10,261	-2	+27	-2	+19
věta 7	12,405	-3	+29	-3	+13
věta 8	11,018	-1	+38	-1	+18
věta 9	9,002	-116	+40	-12	+28
věta 10	8,906	-96	+41	-36	+16
věta 11	6,357	+3	+38	+3	+11
věta 12	8,394	-45	+41	-34	+31
věta 13	13,653	-12	+38	+904	+38
věta 14	10,592	-7	+56	-7	+49
věta 15	13,344	-20	+36	-18	+32
Průměrně	10,666	-26	+46	+47	+29
Celkem	159,994	-	-	-	-

Tabulka 7.2 – Detekované časy s použitím prahování energie a kombinace prahování energie a průchodů nulovou osou

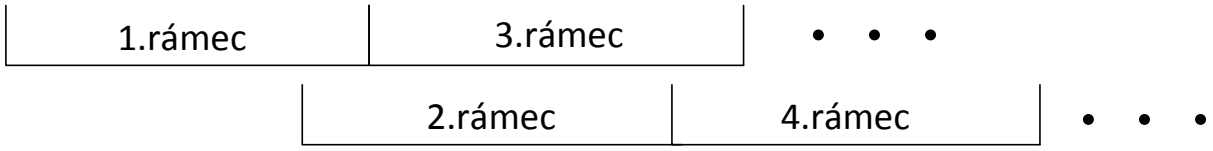
8 Detekce řeči pomocí klasifikátoru SVM

8.1 Segmentace

Při zjišťování vlivu úspěšnosti klasifikace v závislosti na délce rámce jsem v první fázi testování se základními příznaky zvolil rámec o velikosti 10 milisekund. Překryv segmentů byl zvolen 50% délky předchozího rámce (obr. 8.1.1). Při aplikaci dalších parametrů jsem rozšířil rámec na 25 milisekund s posunem 5 milisekund (obr. 8.1.2).



Obrázek 8.1.1 – Znázornění posuvu 10 ms rámců o 5 ms při vzorkování signálu



Obrázek 8.1.2 – Znázornění posuvu 25 ms rámců o 5 ms při vzorkování signálu

8.2 Trénovací data

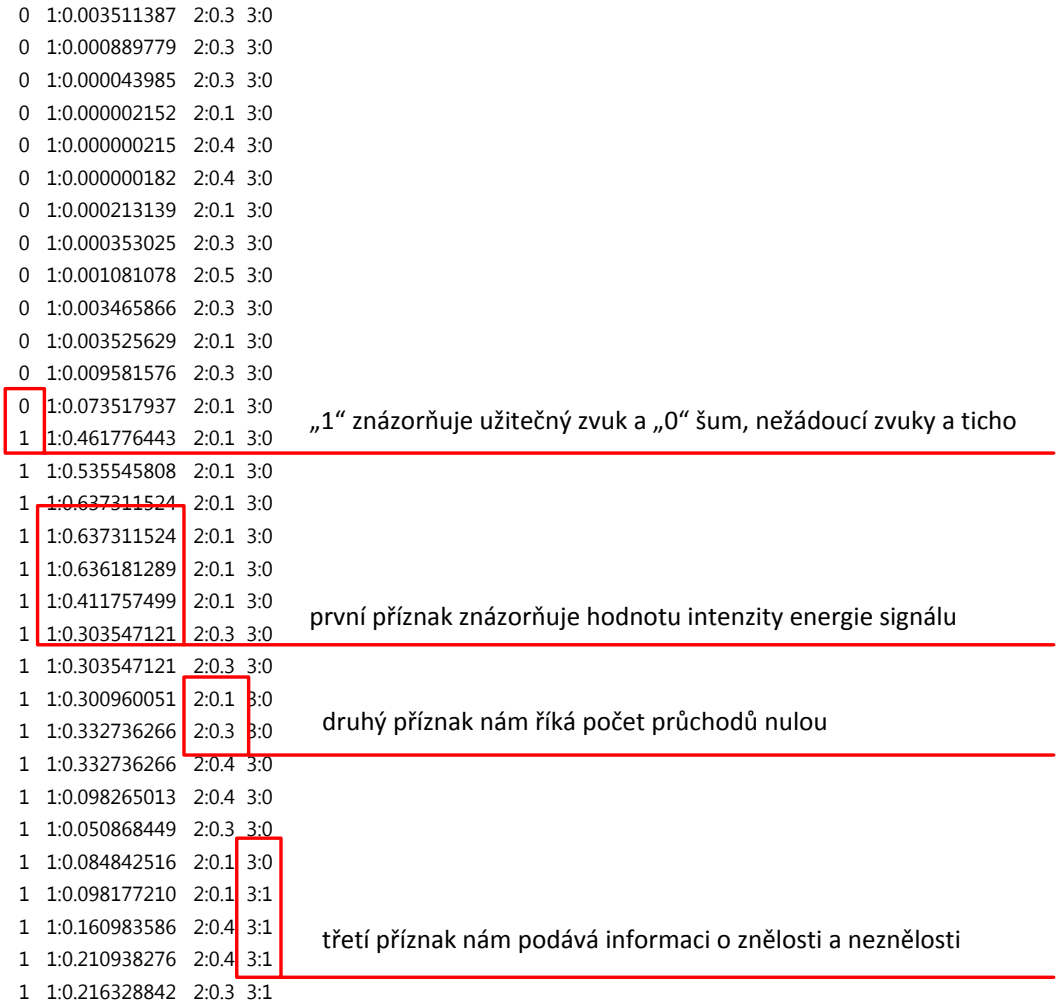
Jako trénovací data jsem si připravil 68 promluv, kde byly zastoupeny mluvená řeč, ticho, šumy a ruchy. Při vytváření trénovacích dat jsem dodal informaci od učitele, kterým jsem byl já. Klasifikátor se tak naučil v kterých případech se jedná o užitečný signál. Ticho, šumy a ruchy jsem proto označil nulou („0“). Jedničkou („1“) jsem označil řeč, která musela být označena s přesností na milisekundy, aby následně nedocházelo k chybné detekci. Těchto 68 promluv o průměrné délce asi 10 sekund jsem rozdělil na rámce dlouhé 25 milisekund. Pro každých 25 milisekund, které vytvářejí jeden rámec, jsem vypočítal různé příznaky, jež jsou popsány v kapitolách 8.3 a 8.4. Trénovací data měla z celkového počtu 68 rozličných promluv 200 880 rámců, které zhruba odpovídají časovému úseku 16 minut a 25 sekund. Tyto rozličné promluvy musejí dostatečně (komplexně) popsat řešenou problematiku, jak uvádím

v kapitole 6.4.1. Podle těchto trénovacích dat následně klasifikátor vytvoří *model,* podle kterého bude klasifikovat *testovací data*.

8.3 Základní příznaky

Z počátku práce s klasifikátorem jsem použil naprosto stejné příznaky jako v kapitole 7, kde jsem použil detekci podle energie a počtů průchodů nulovou osou. Důležitým úkolem bylo vytvoření trénovacích a následně i testovacích dat. Nejdříve jsem jako příznaky zvolil energii (kapitola 3.2), počet průchodů nulovou osou (kapitola 3.3) a informaci o znělosti a neznělosti (kapitola 3.4). Při použití těchto parametrů trénovací a testovací data měla tvar, jaký je zobrazený na obrázku 8.3.1. Trénovací i testovací data ve stejném formátu musí být správně normalizována, o tuto úlohu se postaraly pomocné knihovny SVM. Tyto knihovny si parametry samy správně normalizovaly a následně je využily pro klasifikaci. Hodnoty příznaků musely být přepočteny do intervalu 0 až 1.

Zobrazení výsledků klasifikátoru s použitím základních příznaků je zachyceno v tabulce 8.1. Tabulka ukazuje, o kterou promluvu se jedná, ukazuje její časovou délku v sekundách, počet rámců v jednotlivých promluvách, úspěšnost určení počátečního a koncového času, správně určené procento rámců a počet správně detekovaných rámců z celkového počtu. V tabulce 8.1 vidíme, že odchylky začátku a konce mají různá znaménka. Detekování začátku s kladným znaménkem znamená, že klasifikátor detekoval začátek promluvy ještě před reálným začátkem. Se záporným znaménkem při detekci začátku znamená, že klasifikátor určil začátek promluvy příliš pozdě. Kladné znaménko u odchylky při určení konce promluvy naopak znamená, že se klasifikátor zpozdil oproti přesné reálné hodnotě. Záporné hodnoty odchylek znamenají určení konce příliš brzy.



Obr. 8.3.1. – Část dat připravených pro klasifikaci

Promluvy	Délka věty [s]	Délka věty [rámce]	Odchylka v určení začátku [ms]	Odchylka v určení konce [ms]	Správně určené procento rámců [%]	Počet správně detekovaných rámců z celkového počtu
věta 1	7,616	1513	27	42	90,7395	1362 / 1501
věta 2	9,845	1959	-119	222	72,2137	1406 / 1947
věta 3	12,234	2435	15	59	85,1011	2062 / 2423
věta 4	13,557	2701	14	206	68,5757	1844 / 2689
věta 5	12,810	2551	11	60	86,7664	2203 / 2539
věta 6	10,261	2041	4	44	91,0793	1848 / 2029
věta 7	12,405	2471	25	36	86,5392	2128 / 2459
věta 8	11,018	2193	21	78	87,0243	1898 / 2181
věta 9	9,002	1789	-77	55	84,8621	1508 / 1777
věta 10	8,906	1771	-61	46	87,2655	1535 / 1759
věta 11	6,357	1261	30	53	93,5949	1169 / 1249
věta 12	8,394	1667	-125	74	87,9154	1455 / 1655
věta 13	13,653	2719	18	63	82,0835	2222 / 2707
věta 14	10,592	2107	21	218	85,7279	1796 / 2095
věta 15	13,344	2657	3	86	85,1796	2253 / 2645
Průměrně	10,666	2122	44	50	84,3128	1779 / 2110
Celkem	159,994	31835	-	-	84,3121	26689 / 31655

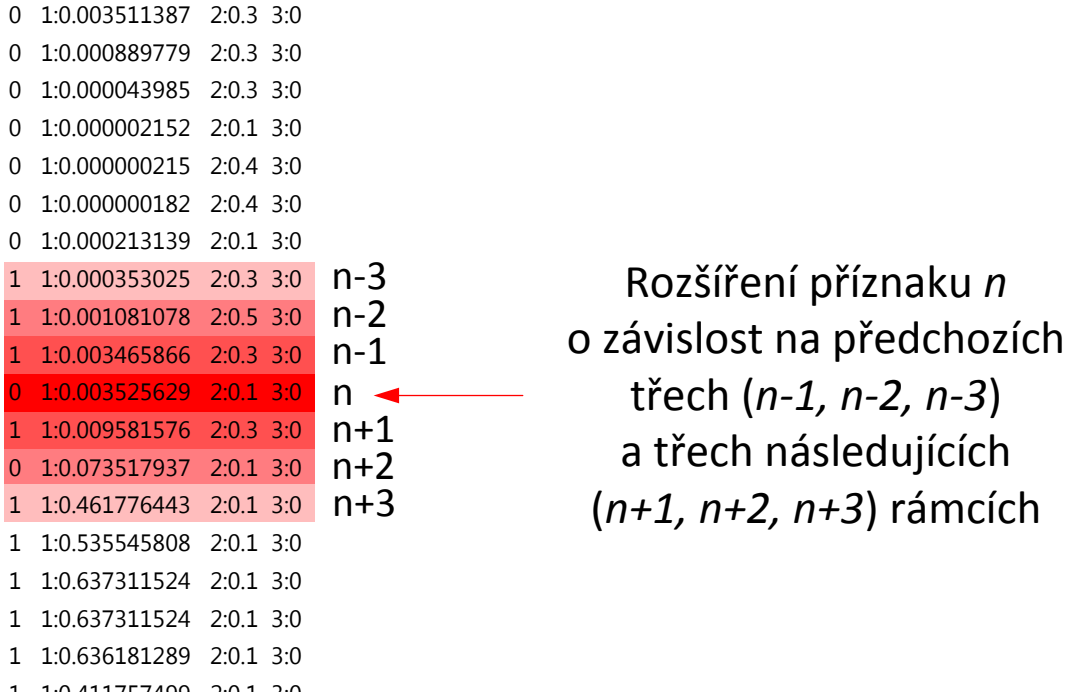
Tabulka 8.1 – Zobrazuje výsledky klasifikátoru s využitím jen základních příznaků

8.4 Rozšířené příznaky

Rozšiřování trénovacích i testovacích dat přinášelo zlepšení, a tak jsem rozšířil jeden rámec na 750 příznaků a čekal výborné výsledky. Rozšiřování bylo bohužel příliš paměťově náročné a trénovací data, při takovémto počtu příznaků na jediný řádek byla příliš velká, asi 1 GB. Po načtení takto velkých dat do operační paměti a provedení pár operací Matlabu a SVM knihoven jsem zjistil, že operační paměti není dostatek.

8.4.1 Závislost rámce na předchůdci a následovníku

Pro další zpřesnění práce klasifikátoru jsem použil větší množství příznaků (obr. 8.2.1). V jednom rámci jsem tak použil informaci o energii, průchodech nulou a o znělosti/neznělosti a také o *x* předchozích a *x* následujících rámcích. Z původních 3 příznaků tedy klasifikátor pracoval s *(2x+1)*3* příznaky na jeden rámec. Maximálně jsem aplikoval 12 předchůdců a 12 následovníků, ale podstatně se zvýšil počet příznaků na rámec a hrozil nedostatek paměti při trénování a vytváření *modelu*. S vyšším počtem předchůdců a následníků se zvyšuje i přesnost detekovaných časů klasifikátorem. Zobrazení třech následníků s třemi předchůdci oproti devíti následníkům s devíti předchůdci si můžeme porovnat v tabulkách 8.2 a 8.3.



Obr. 8.2.1 – Příklad rozšíření příznaků o závislost na předchozích a následujících rámcích

8.4.2 Dynamické koeficienty

Další příznaky jsem vytvořil pomocí takzvaných dynamických koeficientů, které jsem vypočetl pomocí vzorce (8.2.1)

$$d_t = \frac{\sum_{\theta=1}^{\Theta} \theta(c_{t+\theta} - c_{t-\theta})}{2\sum_{\theta=1}^{\Theta} \theta^2} \tag{8.2.1}$$

kde d_t je dynamický koeficient, který odpovídá danému rámci v čase t (resp. indexu času). Vypočítává se ze statických koeficientů od $c_{t-\Theta}$ do $c_{t+\Theta}$. Hodnota Θ je velikost okolí. Pro náš případ jsem zvolil velikost okolí 2.

Aplikací závislosti na předchozí a následující rámce, společně s použitím dynamických koeficientů se detekce zpřesnila. Je to dáno tím, že místo původních 3 příznaků, jich nově klasifikátor pracuje *(2x+1)*3*2*. Využitím této metody získáme jednou takové množství příznaků, takže dynamické koeficienty také zvyšují paměťovou náročnost.

8.4.3 Spektrální koeficienty LSF

Spektrální koeficienty LSF jsme získali pomocí příkazů v Matlabu, kde jsem nejprve získal LPC koeficienty a následně je převedl na LSF koeficienty. Tento postup je popsaný níže. LSF koeficienty jsem dále využíval jako příznaky pro klasifikaci. Výsledky klasifikátoru s využití především LSF koeficientů jsou zobrazeny v tabulce 8.4.

```
LPC koeficienty
===============

a = lpc(x,P)

x - řečové vzorky
P - řád prediktoru (12)


Převod na LSF koeficienty
=========================

lsf = poly2lsf(a)
```

Řád prediktoru jsem zvolil 12, tedy 12 kořenů, které jsou pro moji úlohu obecně nejpoužívanější.

8.4.4 Spektrální koeficienty MFCC

V mé práci jsem zvolil i použití MFCC koeficientů, které také pomáhaly zpřesňovat výslednou klasifikaci.

Ovšem získání MFCC koeficientů není tak jednoduché jako u LSF. Pro výpočet MFCC jste použil příkaz HCopy z programového balíku HTK (Hidden Markov Model Toolkit) [13]. Tento program vytvořil textový dokument s 12-ti parametry, které jsem načetl do Matlabu a dále je používal pro klasifikátor. Výsledky klasifikátoru s využití především MFCC koeficientů jsou zobrazeny v tabulce 8.5.

8.4.5 Formantové frekvence

Jako další příznaky pro zpřesnění klasifikace jsem použil formantové frekvence, které jsem získal z programu na zpracování zvuku a následně je uložil do textového souboru. A opět jsem je jen pohodlně načetl do Matlabu, kde jsem je s pomocí klasifikátoru nadále využíval. Výsledky klasifikátoru s využití především formantových frekvencí jsou zobrazeny v tabulce 8.6.

8.5 Klasifikace

Pro realizaci klasifikace byly použity knihovny libSVM, vytvořené autory Chin-Chung Changem a Chin-Jen Lin [10]. Obsahují prostředky pro tvorbu *modelu* ze vstupních trénovacích dat. Data připravená pro klasifikaci jsou zobrazena na obrázku 8.1.1, v kterých tvoří jeden rámec (řádek) pětimilisekundový úsek. Připravené knihovny pro prostředí Matlab společně s klasifikátorem na předem vytvořených trénovacích datech vytvoří *model* z trénovacích dat. Tento *model* bude dále využíván při klasifikaci testovacích dat s využitím technik Support Vector Machines.

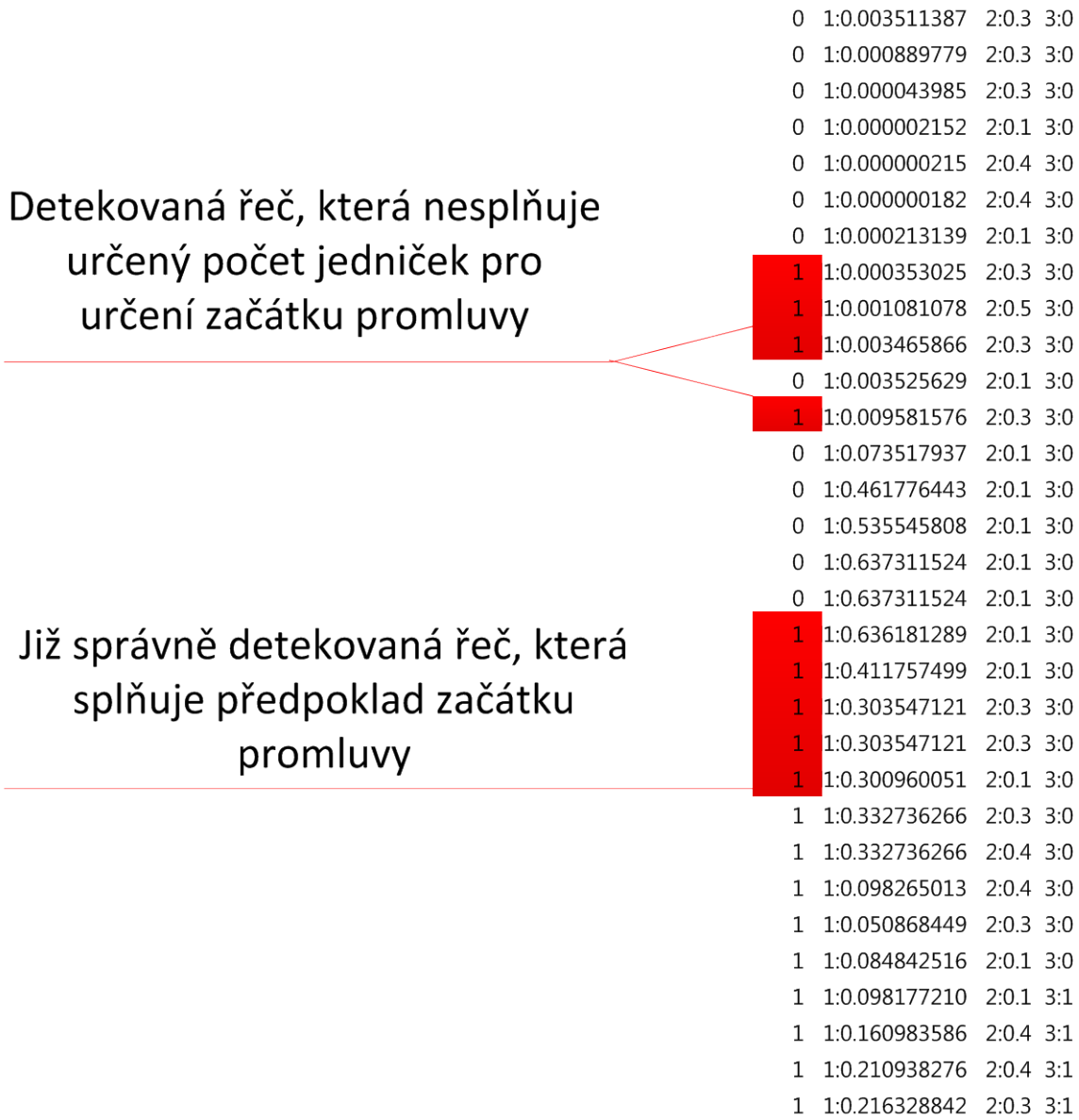
Vyhodnocení úspěšnosti klasifikátoru bylo provedeno na testovacích datech, které tvořilo 15 promluv a jejich součástí nebyla množina trénovacích dat. A také byla vytvořena trénovací data, které tvořilo 68 promluv.

Pomocí SVM klasifikátoru pro využití v prostředí Matlab dostaneme klasifikací výstupní soubor, který je nazvaný *predict_label.* Z tohoto výstupního souboru získáme důležitou informaci o tom, kde podle klasifikace začíná a končí užitečný signál.

V *predict_label* máme uloženu tuto informaci v podobě jedniček a nul. Jedničky značí, že daný rámec už je řeč a nuly znázorňují hluk, šum a ticho. Pomocí jednoduchého algoritmu zjistíme číslo rámce, kde začínají jedničky, pravděpodobně i užitečný signál. V některých případech ale může jít i o šum nebo ruchy. Z těchto důvodů provádíme dolaďování výstupní hodnot (kapitola 6.7). Ve chvíli, kdy nalezneme prvních pět po sobě jdoucích předpovídaných jedniček, víme, že zde začíná námi požadovaný počáteční čas a tím i začátek řečového signálu (obr. 8.2.3). Podobně se postupuje při detekování konce řeči v promluvě, kde hledáme pět jedniček za sebou, které budou označovat, že zde je konec řeči.

8.6 Post-processing

Toto upravování výsledků zastupuje v kapitole 6.4.6 zmíněný *post-processing*, který se stará o dolaďování získaných výsledků. V tomto případě je vidět na obrázku 8.6.1, že hledáme sérii pěti po sobě jdoucích jedniček, které určíme jako začátek řeči. Když nalezneme menší počet jedniček, které byly klasifikovány jako řeč, většinou řečí nejsou. Ve většině případů se jedná o nežádoucí šum.



Obr. 8.6.1 – Rozhodnutí o začátku řečového signálu. Detekované jedničky ve výstupním souboru musí splňovat předpoklad začátku promluvy, kterým je pět po sobě jdoucích jedniček

8.7 Korekce systémové chyby

Ve výstupu klasifikátoru jsem našel konstantní chybu. V počátečním času se klasifikátor o 10 milisekund předbíhal před reálným začátkem. Podobná chyba se objevila i na konci detekovaného času, který byl předpovídaný příliš brzo, a to o 30 milisekund. Domnívám se, že tato chyba nastala v důsledku nepatrného zaokrouhlení v určité části klasifikace. Tato systémová chyba se projevuje ve výsledcích klasifikace na trénovacích i testovacích datech.

8.8 Výstupní hodnoty a úspěšnost klasifikace

V množině testovacích dat, které nebyly z množiny trénovacích, jsem dosáhl uspokojivých výsledků. Odchylka klasifikátoru od reálného začátku promluvy byla maximálně 48 milisekund, ale v průměrné absolutní odchylce byla chyba jen 15 milisekund v detekci počátečního času a 23 milisekund v detekci koncového času. Jako příznaky pro klasifikátor s nejlepším dosaženým výsledkem, který je uvedený v tabulce 8.7, jsem použil všechny výše uvedené parametry, jedná se tak o energii, počet průchodů nulovou osou, znělost a neznělost, dynamické koeficienty, spektrální koeficienty LSF a MFCC, závislost na třech předchůdcích i třech následovnících a formantové frekvence. Rozšíření o tři předchůdce a následovníky bylo přínosné, ale nejlepší by bylo následovníků a předchůdců alespoň 12. Tuto myšlenku jsem bohužel nemohl provést z důvodu, že by vznikla chyba z důvodu nedostatku paměti. Výsledky s nejlepším výsledkem jsou zobrazeny v tabulce 8.7. Tabulka ukazuje, o kterou promluvu se jedná, ukazuje její časovou délku v sekundách, počet rámců v jednotlivých promluvách, úspěšnost určení počátečního a koncového času, správně určené procento rámců a počet správně detekovaných rámců z celkového počtu.

V tabulkách vidíme postupný vývoj práce, kde jsem přidával příznaky jako je závislost na předchůdcích a následovnících (tabulka 8.2 a 8.3), spektrální koeficienty LSF (tabulka 8.4), spektrální koeficienty MFCC (tabulka 8.5) a také formantové frekvence (tabulka 8.6). V tabulkách také vidíme, že odchylky začátku a konce mají různá znaménka. Detekování začátku s kladným znaménkem znamená, že klasifikátor detekoval začátek promluvy ještě před reálným začátkem. Se záporným znaménkem při detekci začátku znamená, že klasifikátor určil začátek promluvy příliš pozdě. Kladné znaménko u odchylky

při určení konce promluvy naopak znamená, že se klasifikátor zpozdil oproti přesné reálné hodnotě. Záporné hodnoty odchylek znamenají určení konce příliš brzy.

Z tabulky 8.7 vyplývá, že průměrná doba řečových promluv byla 10,666 sekundy a z tohoto úseku jsem získal průměrně 2094 rámců, které nesly informaci v podobě příznaků zmíněných v kapitolách 8.3 a 8.4. Klasifikátor rozhodoval s odchylkou, která u patnácti testovaných promluv byla dosti rozdílná, občas klasifikátor určil promluvu velmi přesně a na druhé straně byly i případy, kdy byla detekce nepřesná a lišila se od reálného začátku nebo konce až skoro o 50 milisekund.

Promluvy	Délka věty [s]	Délka věty [rámce]	Odchylka v určení začátku [ms]	Odchylka v určení konce [ms]	Správně určené procento rámců [%]	Počet správně detekovaných rámců z celkového počtu
věta 1	7,616	1513	-3	12	90,1520	1364 / 1513
věta 2	9,845	1959	-234	192	73,2006	1434 / 1959
věta 3	12,234	2435	-15	34	84,4764	2057 / 2435
věta 4	13,557	2701	-16	166	68,4932	1850 / 2701
věta 5	12,810	2551	-19	30	85,2999	2176 / 2551
věta 6	10,261	2041	-26	14	90,2009	1841 / 2041
věta 7	12,405	2471	-5	6	86,1999	2130 / 2471
věta 8	11,018	2193	-9	48	86,0009	1886 / 2193
věta 9	9,002	1789	-112	25	85,0196	1521 / 1789
věta 10	8,906	1771	-96	16	86,0531	1524 / 1771
věta 11	6,357	1261	-5	23	93,4179	1178 / 1261
věta 12	8,394	1667	-160	44	86,8626	1448 / 1667
věta 13	13,653	2719	-12	38	81,2431	2209 / 2719
věta 14	10,592	2107	-9	188	84,3379	1777 / 2107
věta 15	13,344	2657	-22	56	83,9669	2231 / 2657
Průměrně	10,666	2122	53,85	49,01	83,6475	1775 / 2122
Celkem	159,994	31835	-	-	83,6375	26626 / 31835

Tabulka 8.2 – zobrazující výsledky klasifikace pomocí energie, ZCR, znělosti a neznělosti, dynamických koeficientů a závislosti na třech následovnících a třech předchůdcích

Promluvy	Délka věty [s]	Délka věty [rámce]	Odchylka v určení začátku [ms]	Odchylka v určení konce [ms]	Správně určené procento rámců [%]	Počet správně detekovaných rámců z celkového počtu
věta 1	7,616	1501	27	42	90,7395	1362 / 1501
věta 2	9,845	1947	-119	222	72,2137	1406 / 1947
věta 3	12,234	2423	15	59	85,1011	2062 / 2423
věta 4	13,557	2689	14	206	68,5757	1844 / 2689
věta 5	12,810	2539	11	60	86,7664	2203 / 2539
věta 6	10,261	2029	4	44	91,0793	1848 / 2029
věta 7	12,405	2459	25	36	86,5392	2128 / 2459
věta 8	11,018	2181	21	78	87,0243	1898 / 2181
věta 9	9,002	1777	-77	55	84,8621	1508 / 1777
věta 10	8,906	1759	-61	46	87,2655	1535 / 1759
věta 11	6,357	1249	30	53	93,5949	1169 / 1249
věta 12	8,394	1655	-125	74	87,9154	1455 / 1655
věta 13	13,653	2707	18	63	82,0835	2222 / 2707
věta 14	10,592	2095	21	218	85,7279	1796 / 2095
věta 15	13,344	2645	3	86	85,1796	2253 / 2645
Průměrně	10,666	2110	44,07	50,35	84,3128	1779 / 2110
Celkem	159,994	31655	-	-	84,3121	26689 / 31655

Tabulka 8.3 – zobrazující výsledky klasifikace pomocí energie, ZCR, znělosti a neznělosti, dynamických koeficientů a závislosti na devíti následovnících a devíti předchůdcích

Promluvy	Délka věty [s]	Délka věty [rámce]	Odchylka v určení začátku [ms]	Odchylka v určení konce [ms]	Správně určené procento rámců [%]	Počet správně detekovaných rámců z celkového počtu
věta 1	7,616	1485	45	54	90,5724	1345 / 1485
věta 2	9,845	1931	56	25	80,9943	1564 / 1931
věta 3	12,234	2407	33	-11	81,2630	1956 / 2407
věta 4	13,557	2673	5	146	80,7707	2159 / 2673
věta 5	12,810	2523	91	26	87,7130	2213 / 2523
věta 6	10,261	2013	22	-36	82,2156	1655 / 2013
věta 7	12,405	2443	15	-33	84,2407	2058 / 2443
věta 8	11,018	2165	11	-46	81,7090	1769 / 2165
věta 9	9,002	1761	9	-10	84,9517	1496 / 1761
věta 10	8,906	1743	4	36	77,5100	1351 / 1743
věta 11	6,357	1233	-10	33	89,1322	1099 / 1233
věta 12	8,394	1639	-35	-11	82,5503	1353 / 1639
věta 13	13,653	2691	63	47	76,4400	2057 / 2691
věta 14	10,592	2079	15	138	79,3170	1649 / 2079
věta 15	13,344	2629	-7	51	82,3127	2164 / 2629
Průměrně	10,666	2094	24,43	41,88	82,4197	1725,87 / 2094
Celkem	159,994	31415	-	-	82,4065	25888 / 31415

Tabulka 8.4 – zobrazující výsledky klasifikace pomocí spektrálních koeficientů LSF, třech předchůdců a třech následovníků

Promluvy	Délka věty [s]	Délka věty [rámce]	Odchylka v určení začátku [ms]	Odchylka v určení konce [ms]	Správně určené procento rámců [%]	Počet správně detekovaných rámců z celkového počtu
věta 1	7,616	1511	-45	55	89,7419	1356 / 1511
věta 2	9,845	1961	-49	292	82,4069	1616 / 1961
věta 3	12,234	2437	-20	29	88,5105	2157 / 2437
věta 4	13,557	2703	4	196	82,2789	2224 / 2703
věta 5	12,810	2553	-9	25	90,7951	2318 / 2553
věta 6	10,261	2043	-11	-1	93,6368	1913 / 2043
věta 7	12,405	2473	15	51	89,9313	2224 / 2473
věta 8	11,018	2195	-9	48	89,3394	1961 / 2195
věta 9	9,002	1791	-102	20	91,8481	1645 / 1791
věta 10	8,906	1773	-41	6	92,5550	1641 / 1773
věta 11	6,357	1263	15	43	94,4576	1193 / 1263
věta 12	8,394	1669	-445	74	90,1138	1504 / 1669
věta 13	13,653	2721	-7	33	84,7483	2306 / 2721
věta 14	10,592	2109	6	183	87,8141	1852 / 2109
věta 15	13,344	2659	-12	76	88,6800	2358 / 2659
Průměrně	10,666	2124	61	59	89,1238	1885 / 2124
Celkem	159,994	31861	-	-	88,7229	28268 / 31861

Tabulka 8.5 – zobrazující výsledky klasifikace pomocí spektrálních koeficientů MFCC, třech předchůdců a třech následovníků

Promluvy	Délka věty [s]	Délka věty [rámce]	Odchylka v určení začátku [ms]	Odchylka v určení konce [ms]	Správně určené procento rámců [%]	Počet správně detekovaných rámců z celkového počtu
věta 1	7,616	1479	-213	-8	88,3705	1307 / 1479
věta 2	9,845	1925	-389	-13	88,7273	1708 / 1925
věta 3	12,234	2402	-215	-16	89,3838	2147 / 2402
věta 4	13,557	2667	304	-29	86,7642	2314 / 2667
věta 5	12,810	2518	-189	5	90,2303	2272 / 2518
věta 6	10,261	2008	99	-26	89,5418	1798 / 2008
věta 7	12,405	2437	-40	-9	89,4132	2179 / 2437
věta 8	11,018	2159	296	3	85,4562	1845 / 2159
věta 9	9,002	1756	-12	0	87,7563	1541 / 1756
věta 10	8,906	1737	974	1	90,2130	1567 / 1737
věta 11	6,357	1227	965	18	89,2421	1095 / 1227
věta 12	8,394	1634	102	42	84,5165	1381 / 1634
věta 13	13,653	2686	13	-7	87,8630	2360 / 2686
věta 14	10,592	2074	-299	-27	88,4764	1835 / 2074
věta 15	13,344	2624	48	16	88,0716	2311 / 2624
Průměrně	10,666	2089	288,32	14,44	88,2684	1844 / 2089
Celkem	159,994	31333	-	-	88,2775	27660 / 31333

Tabulka 8.6 – zobrazující výsledky klasifikace pomocí formantových frekvencí, dynamických koeficientů, pěti předchůdců a pěti následovníků

Promluvy	Délka věty [s]	Délka věty [rámce]	Odchylka v určení začátku [ms]	Odchylka v určení konce [ms]	Správně určené procento rámců [%]	Počet správně detekovaných rámců z celkového počtu
věta 1	7,616	1485	33	41	91,2458	1355 / 1485
věta 2	9,845	1931	44	16	90,2123	1742 / 1931
věta 3	12,234	2407	21	-5	82,7171	1991 / 2407
věta 4	13,557	2673	5	46	87,8788	2349 / 2673
věta 5	12,810	2523	48	17	93,6583	2363 / 2523
věta 6	10,261	2013	10	-18	86,1898	1735 / 2013
věta 7	12,405	2443	10	-13	94,0647	2298 / 2443
věta 8	11,018	2165	11	-36	85,8661	1859 / 2165
věta 9	9,002	1761	9	-5	94,6053	1666 / 1761
věta 10	8,906	1743	4	26	90,7057	1581 / 1743
věta 11	6,357	1233	-5	23	91,5653	1129 / 1233
věta 12	8,394	1639	-25	-11	86,8212	1423 / 1639
věta 13	13,653	2691	34	27	87,5883	2357 / 2691
věta 14	10,592	2079	15	58	85,0890	1769 / 2079
věta 15	13,344	2629	-7	38	89,9201	2364 / 2629
Průměrně	10,666	2094	14,96	22,61	89,0831	1865,40 / 2094
Celkem	159,994	31415	-	-	89,0689	27981 / 31415

Tabulka 8.7 – nejlepší výsledné hodnoty pomocí SVM klasifikátor

9 Závěr a shrnutí

V prvních kapitolách jsme popsali problematiku syntézy řeči, která se používá pro automatické čtení textů. Velice důležitým přínosem těchto technologii je využití pro nevidomé nebo jinak hendikepované.

Syntéza řeči představuje rozsáhlou problematiku a tato práce měla za úkol řešit jen podpůrnou úlohu detekce začátků a konců ve studiových nahrávkách, které se poté používají v samotné syntéze řeči. Korpusově orientovaná syntéza řeči většinou spoléhá na velké množství nahrávek. Přesné určení počátečního a koncového času v promluvě je pro syntézu řeči velmi důležité. Tato úloha však může najít využití také například tehdy, když budeme chtít ušetřit paměťový prostor a odříznout z nahrávky nepotřebné a místo zabírající ticho. Vzhledem k tomu, že se pro syntézu řeči používají velmi kvalitní studiové nahrávky, nemusí se jednat o zanedbatelný paměťový prostor.

V kapitole 7 byla vyzkoušena detekce začátků a konců realizována pomocí prahové energie a následně i pomocí průchodů nulovou osou. Tato detekce nedosahovala představovaných kvalit, a tak jí muselo rozšířit použití klasifikátoru.

Detekce začátku a konce užitečného signálu nakonec byla provedena pomocí klasifikátoru, který pracoval v prostředí Matlab se SVM knihovnami. Klasifikátor měl za úkol ze zadaných příznaků určit jaký rámec v promluvě je ticho a šum, a ten ohodnotit nulou. A také správně určit, které rámce jsou rámce užitečného zvuku, neboli řeči a ty ohodnotit jedničkou. Pro klasifikátor samozřejmě byla vytvořena trénovací data, které tvořila řada příznaků. Mezi prvními testovanými pokusy byla klasifikace podle prahové energie a průchodů nulovou osou. Dále tyto příznaky byly rozšířeny o informaci o znělosti a neznělosti, o dynamické koeficienty, o závislosti na předchozích a následujících rámcích, o spektrální parametry LSF a dále o spektrální parametry MFCC a formantové frekvence.

Testovací data měla ty samé příznaky jako data trénovací, až na informaci od učitele. Klasifikátor z rámců v testovacích datech vyhodnotil, který rámec je zvuk a který šum nebo ticho. Toto vyhodnocení vytvořilo vektor jedniček a nul. Následným zpracováním jsem pomocí testování rozhodl, že po pěti po sobě jdoucích jedničkách už většinou začíná řeč. Tento postup jsem opakoval stejným způsobem i od konce. Pomocí tohoto vyhodnocení výsledků jsem určil počáteční a koncový čas řečového signálu. A následně jsem vypočetl odchylky od reálného času, které jsou uvedeny v tabulkách 8.1, 8.2, 8.3, 8.4, 8.5, 8.6 a 8.7.

Jako další pokračování na této práci by bylo možné vylepšit výsledky pomocí detailnějšího nastudování nastavení v SVM klasifikátoru. To znamená vyzkoušení různých jádrových funkcí v SVM. Dále by byla určitě prospěšná optimalizace nastavení všech parametrů SVM a v neposlední řadě by určitě klasifikátoru pomohlo rozšíření o další příznaky.

Literatura

[1] Psutka Josef, Müller Luděk, Matoušek Jindřich, Radová Vlasta. *Mluvíme s počítačem česky.* Praha: Academia, 2006.

[2] Brunnhofer Václav, *Kepstrální analýza řečového signálu,* Semestrální práce, ČVUT v Praze, nedat.

[3] http://www.kky.zcu.cz/cs/research-fields/acoustic-speech-synthesis [online], cit. 19.5.2011

[4] Kubernát Tomáš, *Snížení náročnosti výpočtů v libSVM s použitím řetězcových funkcí,* Bakalářská práce, VUT v Brně 2010

[5] Běhůnek Martin, *Rozpoznávání řeči při různé kvalitě vstupního signálu,* Diplomová práce, ČVUT v Praze 2010

[6] *Akustické listy*, říjen 2010, www.czakustika.cz [online], cit. 19.5.2011

[7] *Znělost*, http://cs.wikipedia.org/wiki/Znělost [online], cit. 19.5.2011

[8] Ptáček Pavel, *Detekce začátku řeči v řečových signálech*, Projekt 5 (KKY/PRJ5), ZČU v Plzni 2011

[9] Mahdal Jakub, *Srovnání klasifikátorů*, Systémy zpracování řeči, VUT v Brně 2006

[10] Chih-Chung Chang and Chih-Jen Lin, A library for support vector machines, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[11] *Akustické listy*, prosinec 2010, www.czakustika.cz [online], cit. 19.5.2011

[12] Martin Žemlička, *Rozeznávání izolovaných slov závislých na řečníkovi v reálném čase,* Diplomová práce, ČVUT v Praze 2007

[13] Young, S.; Evermann, G.; Gales, M.; Hain, T.; Kershaw, D.; Liu, X.; Moore, G.; Odell, J.; Ollason, D.; Povey, D.; Valtchev, V., Woodland, P.: The HTK Book (for HTK Version 3.4), Cambridge University, 2006