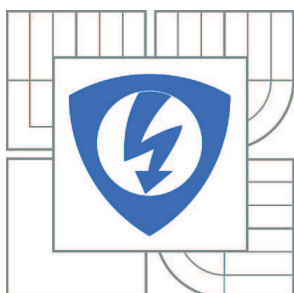


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

DETEKTOR ŘEČOVÉ AKTIVITY V SIGNÁLOVÉM PROCESORU

SPEECH ACTIVITY DETECTOR IN DIGITAL SIGNAL PROCESSOR

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

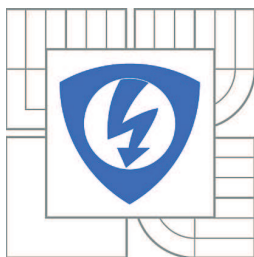
Bc. JIŘÍ KOVAŘÍK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PETR SYSEL, Ph.D.

BRNO 2012



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Jiří Kovařík

ID: 106553

Ročník: 2

Akademický rok: 2011/2012

NÁZEV TÉMATU:

Detektor řečové aktivity v signálovém procesoru

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s metodami detekce řečové aktivity používanými např. v kódecích GSM Half Rate, Full Rate, G.723, apod. Funkci detektorů ověřte v prostředí Matlab a porovnejte výsledky s výsledky originálních kódů. Dále se seznamte s vývojovým kitem TMS320C6416 a vývojovým prostředím Code Composer Studio. Dílčí části zvoleného detektor upravte pro implementaci v signálovém procesoru a otestujte jeho funkci.

DOPORUČENÁ LITERATURA:

- [1] Digital cellular telecommunications systém (Phase 2+); Half rate speech; Half rate speech transcoding (GSM 06.20 version 8.0.0 Release 1999). Draft ETSI EN 300 969.
- [2] Karban, P. Výpočty a simulace v programech Matlab a Simulink. 1. vydání. Brno: Computer Press, 2006. 220 s. ISBN 978-80-251-1448-3
- [3] Psutka, J., Müller, L., Matoušek, J., Radová, V. Mluvíme s počítačem mluvenou řečí. 1. vydání. Praha: Academia, 2006. 752 s. ISBN 80-200-1309-1

Termín zadání: 6.2.2012

Termín odevzdání: 24.5.2012

Vedoucí práce: Ing. Petr Sysel, Ph.D.

Konzultanti diplomové práce:

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Výzkum popsáný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma „Detektor řečové aktivity v signálovém procesoru“ jsem vypracoval samostatně pod vedením vedoucího semestrálního projektu 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne 24.5.2012

.....

podpis autora

Poděkování

Děkuji vedoucímu práce Ing. Petru Syslovi, Ph.D. za ochotu pomoci a za cenné rady při zpracování diplomové práce.

V Brně dne 24.5.2012

.....

podpis autora

Abstrakt

V rámci diplomové práce byly vytvořeny detektory řečové aktivity podle standardu ITU-T G.729 a G.723.1. Byly implementovány v signálovém procesoru TMS320C6416 firmy Texas Instruments. Zároveň byly detektory vytvořeny i v programovacím jazyku MATLAB. Diplomovou práci lze rozdělit na dvě části. Teoretickou a implementační. V teoretické části jsou uvedeny informace o detektorech jak jsou uváděny ve standardech ITU-T G.729 a G.723.1. V části implementační jsou popsány jednotlivé kroky implementace detektorů v signálovém procesoru TMS320C6416 a jsou zde diskutovány různé odlišnosti oproti dokumentaci.

Klíčová slova

Detektor řečové aktivity, kmitočet spektrálních párů, střední počet průchodu signálu nulou, G.729, G.723.1, TMS320C6416.

Abstract

In this diploma thesis were created voice activity detectors according to the standard ITU-T G.729 and G.723.1. The voice activity detectors were implemented in the digital signal processor TMS320C6416 made by Texas Instruments. At the same time detectors were designed using by MATLAB programming language. The diploma thesis can be divided into two parts. In the theoretical section provides information on how to report detectors in the standard ITU-T G.729 and G.723.1. In the implementation part is described steps in the implementation of the detector in signal processor TMS320C6416 and there are discussed various differences compared to the documentation.

Keywords

Voice activity detector, VAD, line spectral frequencies, LSF, zero crossing rate, G.729, G.723.1, TMS320C6416.

KOVAŘÍK, J. *Detektor řečové aktivity v signálovém procesoru*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 64 s. Vedoucí diplomové práce Ing. Petr Sysel, Ph.D.

OBSAH

| | |
|--|-----------|
| ÚVOD | 11 |
| 1 DETEKTOR ŘEČOVÉ AKTIVITY PODLE ITU-T G.729 | 12 |
| 1.1 Předzpracování signálu | 14 |
| 1.2 Extrakce parametrů | 15 |
| 1.2.1 Váhovací funkce a výpočet autokorelace | 15 |
| 1.2.2 Kmitočet spektrálních párů (LSF) | 16 |
| 1.2.3 Širokopásmová energie | 18 |
| 1.2.4 Úzkopásmová energie | 18 |
| 1.2.5 Střední počet průchodu nulou | 18 |
| 1.3 Inicializace parametrů šumu na pozadí | 18 |
| 1.4 Výpočet dlouhodobé minimální energie | 19 |
| 1.5 Výpočet rozdílových parametrů | 19 |
| 1.5.1 Spektrální zkreslení | 19 |
| 1.5.2 Rozdíl širokopásmové energie | 20 |
| 1.5.3 Rozdíl úzkopásmové energie | 20 |
| 1.5.4 Rozdíl středního počtu průchodu nulou | 20 |
| 1.6 Více hraniční VAD rozhodnutí | 20 |
| 1.7 Vyhazení VAD rozhodnutí | 21 |
| 1.8 Aktualizace parametrů šumu na pozadí | 22 |
| 2 DETEKTOR ŘEČOVÉ AKTIVITY PODLE ITU-T G.723.1 | 24 |
| 2.1 Rozdělení dat | 24 |
| 2.2 Filtr typu horní propust | 24 |
| 2.3 LPC analýza | 25 |
| 2.4 Váhovací filtr pro vnímání formantů | 26 |
| 2.5 Odhad základní periody řeči | 26 |
| 2.6 Výpočet příznaku povolení adaptace | 27 |
| 2.7 Inverzní filtrace | 27 |
| 2.8 Výpočet filtrované energie | 28 |
| 2.9 Výpočet úrovně šumu | 29 |
| 2.10 Výpočet prahu | 29 |
| 2.11 Rozhodnutí VAD | 29 |
| 2.12 VAD vyhlazení | 30 |
| 3 IMPLEMENTACE VAD PODLE ITU-T G.729 V SIGNÁLOVÉM PROCESORU | 31 |
| 3.1 Preemfáze | 31 |

| | | |
|----------|---|-----------|
| 3.2 | Váhovací funkce a výpočet autokorelace | 33 |
| 3.3 | Kmitočet spektrálních párů | 34 |
| 3.4 | Širokopásmová energie | 36 |
| 3.5 | Střední počet průchodu nulou | 37 |
| 3.6 | Úzkopásmová energie | 37 |
| 3.7 | Výpočet dlouhodobé minimální energie | 41 |
| 3.8 | Výpočet rozdílových parametrů | 41 |
| 3.9 | Vícehraniční VAD rozhodnutí | 42 |
| 3.10 | Vyhlazení VAD rozhodnutí | 43 |
| 3.11 | Aktualizace parametrů šumu na pozadí | 44 |
| 4 | IMPLEMENTACE VAD PODLE ITU-T G.723.1V SIGNÁLOVÉM PROCESORU | 45 |
| 4.1 | Předzpracování signálu | 45 |
| 4.2 | LPC analýza | 46 |
| 4.3 | Váhovací filtr pro vnímání formantů | 46 |
| 4.4 | Odhad základní periody řeči | 47 |
| 4.5 | Výpočet příznaku povolení adaptace | 48 |
| 4.6 | Inverzní filtrace a výpočet filtrované energie | 48 |
| 4.7 | Výpočet úrovně šumu | 48 |
| 4.8 | Výpočet prahu | 49 |
| 4.9 | Rozhodnutí VAD | 50 |
| 4.10 | VAD vyhlazení | 50 |
| 5 | ZÁVĚR | 51 |
| | LITERATURA | 52 |
| | SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK | 53 |
| | SEZNAM PŘÍLOH | 55 |
| A | Texty testovacích nahrávek | 56 |
| A.1 | Testovací nahrávka sny.wav | 56 |
| A.2 | Testovací nahrávka sny_sum.wav | 56 |
| A.3 | Testovací nahrávka alkoholik8k.wav | 56 |
| B | Výsledky VAD podle ITU-T G.729 | 57 |
| B.1 | Výsledky pro nahrávku sny.wav | 57 |
| B.2 | Výsledky pro nahrávku sny_sum.wav | 58 |
| B.3 | Výsledky pro nahrávku alkoholik8k.wav | 59 |
| C | Výsledky VAD podle ITU-T G.723.1 | 60 |

| | | |
|----------|---------------------------------------|-----------|
| C.1 | Výsledky pro nahrávku sny.wav | 60 |
| C.2 | Výsledky pro nahrávku sny_sum.wav | 61 |
| C.3 | Výsledky pro nahrávku alkoholik8k.wav | 62 |
| D | Obsah přiloženého CD | 63 |

Seznam obrázků

| | |
|---|----|
| OBRÁZEK 1.1: SCHÉMA VAD PODLE ITU-T G.729. | 13 |
| OBRÁZEK 1.2: KMITOČTOVÁ CHARAKTERISTIKA PREEMFÁZE. | 14 |
| OBRÁZEK 2.1: SCHÉMA DETEKTORU ŘEČOVÉ AKTIVITY PODLE ITU-T G.723.1. | 30 |
| OBRÁZEK 3.1: PRVNÍ KANONICKÁ FORMA PRO IMPLEMENTACI PREEMFÁZE VAD G.729. | 31 |
| OBRÁZEK 3.2: MODULOVÉ KMITOČTOVÉ CHARAKTERISTIKY STAVOVÝCH PROMĚNNÝCH v_2 (NAHOŘE) A v_1 (DOLE). | 32 |
| OBRÁZEK 3.3: PŘEKRÝVÁNÍ RÁMCŮ PODLE ITU-T G.729. | 33 |
| OBRÁZEK 4.1: PŘÍMÁ FORMA PRO IMPLEMENTACI PREEMFÁZE VAD G.723.1. | 45 |
| OBRÁZEK B.1: ZOBRAZENÍ VÝSLEDKU VAD PODLE ITU-T G.729 PRO NAHRÁVKU SNY.WAV. | 57 |
| OBRÁZEK B.2: ZOBRAZENÍ VÝSLEDKU VAD PODLE ITU-T G.729 PRO NAHRÁVKU SNY_SUM.WAV. | 58 |
| OBRÁZEK B.3: ZOBRAZENÍ VÝSLEDKU VAD PODLE ITU-T G.729 PRO NAHRÁVKU ALKOHOLIK8K.WAV. | 59 |
| OBRÁZEK C.1: ZOBRAZENÍ VÝSLEDKU VAD PODLE ITU-T G.723.1 PRO NAHRÁVKU SNY.WAV. | 56 |
| OBRÁZEK C.2: ZOBRAZENÍ VÝSLEDKU VAD PODLE ITU-T G.723.1 PRO NAHRÁVKU SNY_SUM.WAV. | 57 |
| OBRÁZEK C.3: ZOBRAZENÍ VÝSLEDKU VAD PODLE ITU-T G.723.1 PRO NAHRÁVKU ALKOHOLIK8K.WAV | 58 |

Seznam tabulek

| | |
|--|----|
| TABULKA 1.1: KONSTANTY POUŽITÉ V DETEKTORU ŘEČOVÉ AKTIVITY PODLE ITU-T G.729 [1]. | 23 |
| TABULKA 3.1: KONSTANTY VÍCEHRANIČNÍHO VAD ROZHODNUTÍ PODLE ORIGINÁLNÍHO KÓDU. | 43 |
| TABULKA 3.2: KONSTANTY PRO VYHLAZENÍ VAD ROZHODNUTÍ PODLE ORIGINÁLNÍHO KÓDU. | 43 |
| TABULKA 3.3: KONSTANTY PRO AKTUALIZACI ŠUMU NA POZADÍ PODLE ORIGINÁLNÍHO KÓDU. | 44 |

Úvod

Tato diplomová práce je zaměřena na detektory řečové aktivity a jejich implementaci na signálovém procesoru. Práce je rozdělena na čtyři části.

V první části je podrobně popsán detektor řečové aktivity podle ITU-T G.729. V první fázi dochází k extrakci následujících parametrů. Širokopásmová energie, úzkopásmová energie, kmitočet spektrálních párů a střední počet průchodu nulou. Pokud je číslo rámce menší než 32, tak se rozhodnutí, zda se jedná o řečový rámec či nikoliv, provádí na základě širokopásmové energie. Pokud je energie větší než 15 dB, jedná se o řečový rámec. Když je číslo rámce rovno 32, tak se provede inicializace charakteristických energií šumu na pozadí. V dalším kroku jsou vypočítány rozdílové parametry, které jsou rozdílem mezi hodnotou parametru aktuálního rámce a průměrem charakteristik šumu na pozadí. Rozdílové parametry jsou použity pro vícehraniční rozhodnutí detektoru řečové aktivity. V poslední fázi je rozhodnutí dále vyhlazeno a dochází k aktualizaci parametrů šumu na pozadí pouze za podmínky, jedná-li se o neřečový rámec.

V druhé části je popsán detektor řečové aktivity podle ITU-T G.723.1. Pracuje s délkou rámce 30 ms a v podstatě se jedná o energetický detektor s adaptivní úrovní prahu. Společné součásti s detektorem řečové aktivity podle ITU-T G.729 jsou LPC analýza desátého řádu a předzpracování signálu pomocí filtrace horní propustí (preemfáze). Rozhodování se zde provádí pouze na základě energie na rozdíl od detektoru řečové aktivity podle ITU-T G.729, kde je použito vícehraniční rozhodnutí.

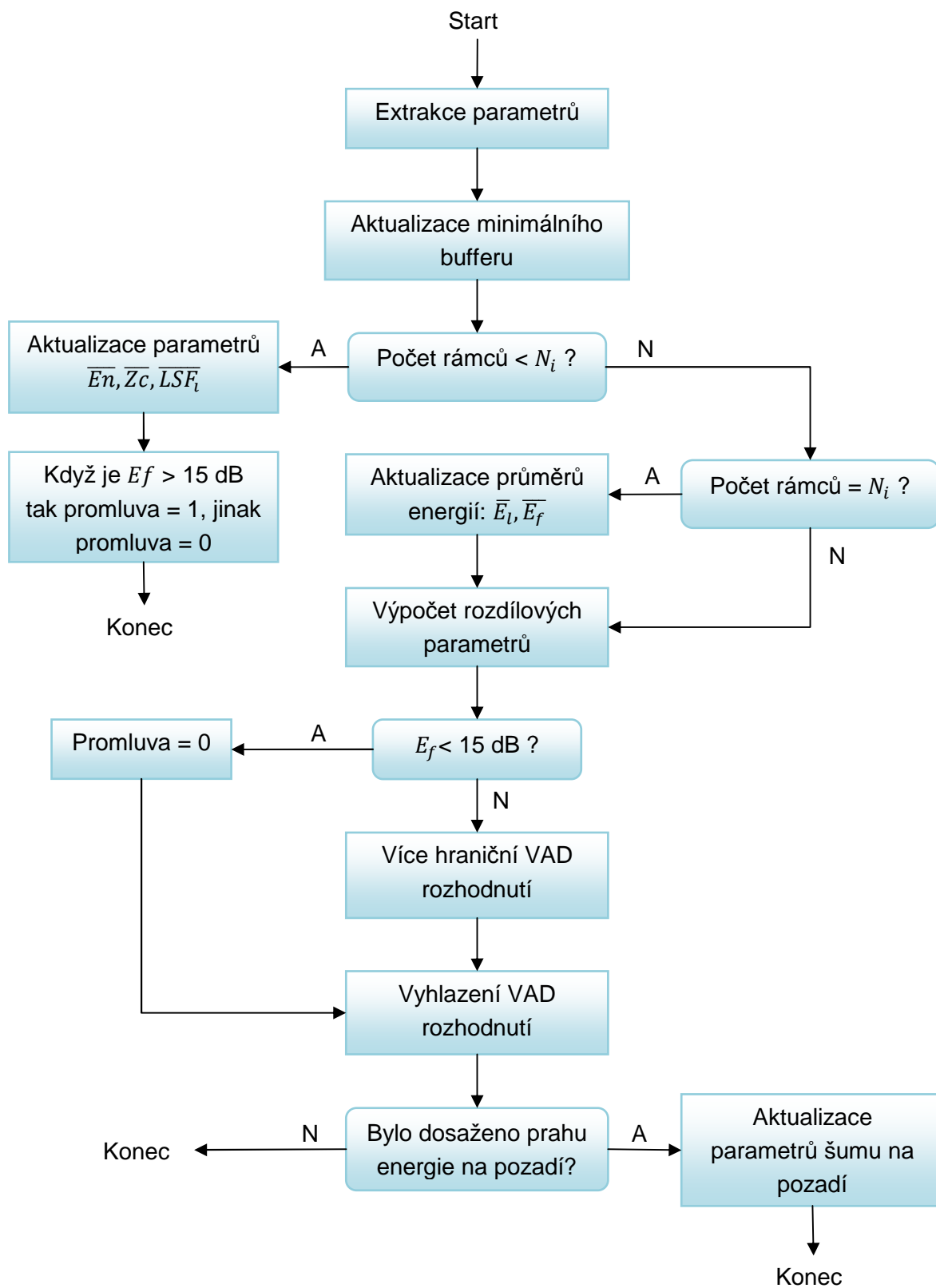
Třetí část práce se zabývá implementací detektoru řečové aktivity podle ITU-T G.729 v signálovém procesoru TMS320C6416 a jsou zde uvedeny některé rozdíly mezi dokumentací standardu ITU-T G.729 [1] a kódem, který dokumentace popisuje.

Ve čtvrté části je popsána implementace detektoru řečové aktivity podle ITU-T G.723.1 v signálovém procesoru. Podobně jako u předchozího detektoru jsou zde nějaké odlišnosti mezi dokumentací [2] a kódem, které jsou uvedeny v této kapitole.

1 Detektor řečové aktivity podle ITU-T G.729

Doporučení ITU-T G.720 - 729 se zabývají kódováním analogového signálu jinými metodami než pulzní kódovou modulací (PCM) a ve standardu ITU-T G. 729 je popsáno kódování řečového signálu pro přenosovou rychlost 8 kbit/s pomocí kodeku CS-ACELP (conjugate structure algebraic code excited linear prediction). Hlavním důvodem proč použít doporučení ITU-T G.729, je snížení zatížení linky při přenosu řečového signálu. Při zachování téměř stejné kvality řeči se sníží požadavky na kapacitu linky osmkrát oproti PCM definovanému v ITU-T G.711 a zároveň kvalita řeči zůstane téměř totožná. Nejčastěji se kodek G.729 používá ve VoIP (voice over internet protocol). Společnost Cisco provedla subjektivní testy hodnocení kvality řeči na jejich zařízeních a kodek G.729 na stupnici MOS, kde 5 znamená hodnocení vynikající a 1 špatné, dosáhl hodnoty 3,92. Pro srovnání G.711 (PCM) dosáhl skóre 4,1 [3].

Kodek G.729 přistupuje rozdílně k datům které obsahují řečový signál a jinak k datům získaných během pauzy v promluvě [1]. Právě rozhodování, který rámec obsahuje jaká data, provádí detektor řečové aktivity a příslušná data jsou potom kódována/dekódována odlišnými způsoby. Rozhodnutí o jaký rámec se jedná, se provádí každých 10 ms. Takže při vzorkovací frekvenci 8000 Hz máme k dispozici 80 vzorků signálu. Hlavními rozhodovacími parametry jsou širokopásmová energie, úzkopásmová energie, počet průchodů nulou a kmitočet spektrálních párů. Pokud je číslo rámce N_i menší než 32, tak VAD se rozhoduje na základě energie chyby predikce získané z LPC analýzy a uchovávají se data pro výpočet dlouhodobých průměrů parametrů. Pokud je energie chyby predikce menší než 15 dB tak je výstup VAD 1 tzn. jedná se o rámec obsahující řeč. V opačném případě je výstup z VAD 0. Když je N_i rovno 32, tak jsou k dispozici všechny data pro výpočty dlouhodobých průměrů parametrů a je získaná charakteristická energie šumu na pozadí. Další rozhodování probíhá pomocí souboru rozdílových parametrů a dalších jejich úprav, což bude podrobněji popsáno dále. Rozdílové parametry jsou získány jako rozdíl mezi aktuálním parametrem a jeho dlouhodobým průměrem. V rámcích neobsahujících řeč se dlouhodobé průměry parametrů přizpůsobují charakteru šumu na pozadí a jsou aktualizovány pouze, jedná-li se o rámec bez řečové aktivity.



Obrázek 1.1: Schéma VAD podle ITU-T G.729.

1.1 Předzpracování signálu

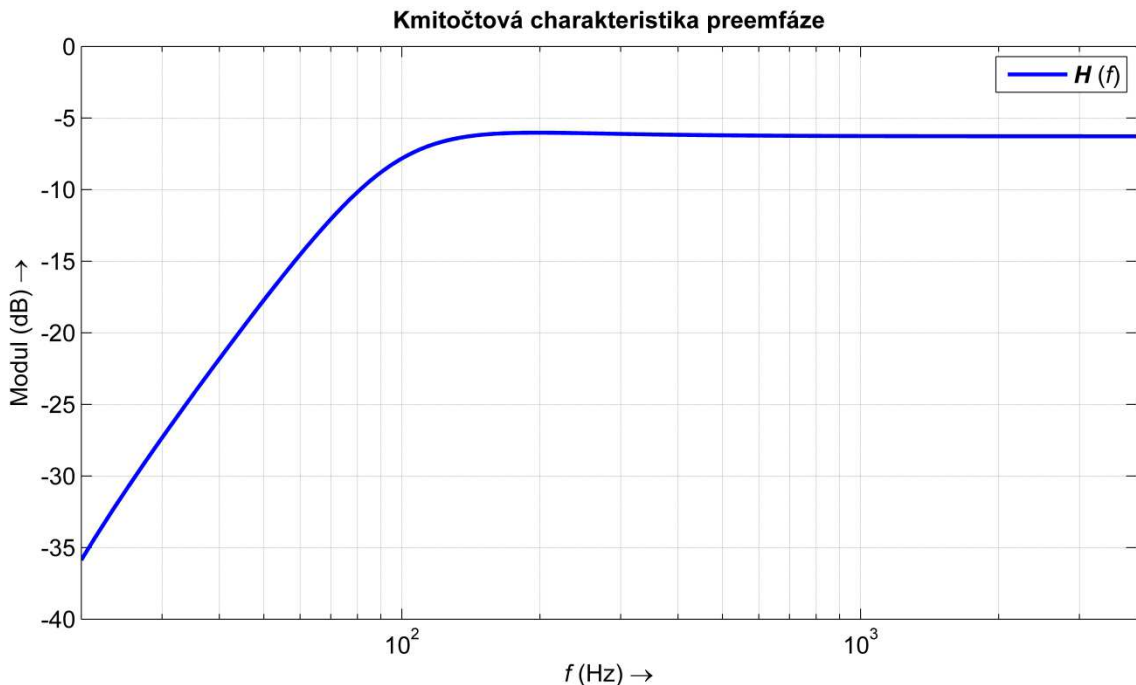
Jako vstupní signál předpokládáme PCM signál s rozlišením 16 bitů. Předzpracování signálu se skládá ze dvou částí:

- Změna měřítka signálu
- Filtrace horní propustí

Změna měřítka signálu se skládá z podělení vstupního signálu dvěma. Tímto se sníží možnost přetečení. Filtrace horní propustí zase omezí frekvence nižší než 140 Hz. Tato operace bývá často označována jako preemfáze. V pásmu do 140 Hz se velmi často vyskytují rušivé složky (např. dýchání mluvčího nebo síťové rušení 50 Hz) a pro srozumitelnost nám jeho odfiltrování nijak nevadí. Uvedené operace jsou sloučeny do jediného IIR filtru druhého řádu s přenosovou funkcí:

$$H(z) = \frac{0,46363718 - 0,92724705z^{-1} + 0,46363718z^{-2}}{1 - 1,9059465z^{-1} + 0,911402z^{-2}} \quad (1.1)$$

Frekvenční charakteristiku můžeme vidět na obrázku 1.2 a také si můžeme ověřit, že vstupní signál je opravdu zmenšen dvakrát, jak bylo uvedeno výše.



Obrázek 1.2: Kmitočtová charakteristika preemfáze.

1.2 Extrakce parametrů

Tato kapitola popisuje blok extrakce parametrů uvedený v obrázku 1.1. Parametry jsou vypočítány pro každý řečový rámeček a nejdůležitější informací pro výpočet parametrů jsou autokorelační koeficienty. Do bloku extrakce parametrů vstupuje předzpracovaný signál získaný pulzní kódovou modulací.

1.2.1 Váhovací funkce a výpočet autokorelace

Váhovací funkce se skládá ze dvou částí. První část reprezentuje polovina Hammingova okna a druhou část jedna čtvrtina periody harmonické funkce kosinus. Výsledné okno je dáno:

$$w_{lp}(n) = \begin{cases} 0,54 - 0,46 \cos\left(\frac{2\pi n}{399}\right) & n = 0, \dots, 199 \\ \cos\left(\frac{2\pi(n - 200)}{159}\right) & n = 200, \dots, 239 \end{cases} \quad (1.2)$$

Váhovací funkce využívá 120 vzorků signálu z minulého rámečku, 80 vzorků ze současného rámečku a 40 vzorků z rámečku budoucího. Tím, že používáme 40 vzorků z budoucího rámečku, tak zavádíme zpoždění 5 ms než budoucí vzorky načteme a můžeme je použít pro další výpočty.

Pro výpočet autokorelačních koeficientů si musíme definovat signál $s(n)$, který získáme vyfiltrováním vstupního signálu IIR filtrem s přenosovou funkcí uvedenou v 1.1. Takto získaný signál je vynásoben váhovací funkcí a vzniklý signál je popsán:

$$s'(n) = w_{lp}(n) \cdot s(n) \quad (1.3)$$

Teď už máme definovaný signál pro výpočet autokorelačních koeficientů a vzorec pro jejich výpočet je následující:

$$r(k) = \sum_{n=k}^{239} s'(n) \cdot s'(n - k) \quad (1.4)$$

Kde: k je rovno 0 až 12.

Pro vyřešení problémů s aritmetikou u malých vstupních signálů se autokorelační koeficienty dále násobí konstantou w_{lag} která je definována:

$$w_{lag}(k) = e^{\left[-\frac{1}{2}\left(\frac{2\pi f_0 k}{f_s}\right)^2\right]} \quad (1.5)$$

Kde: f_0 je 60 Hz,
 f_s je vzorkovací kmitočet a je roven 8000 Hz,
 k je rovno 1 až 12.

Modifikované autokorelační koeficienty jsou potom dány:

$$\begin{aligned} r'(0) &= 1,0001 \cdot r(0) \\ r'(k) &= w_{lag}(k) \cdot r(k) \end{aligned} \quad (1.6)$$

1.2.2 Kmitočet spektrálních párů (LSF)

Modifikované autokorelační koeficienty $r'(k)$ jsou použity k získání LPC koeficientů a_i vyřešením soustavy rovnic:

$$\sum_{i=1}^{10} a_i \cdot r'(|i - k|) = -r'(k) \quad (1.7)$$

Soustava rovnic je vyřešena pomocí rekurzivního Levinson-Durbinova algoritmu, který využívá toho, že matice autokorelačních koeficientů je symetrická a je v Töplitzově tvaru. Algoritmus je definován následovně:

$$\begin{aligned} E^{[0]} &= r'(0) \\ \text{for } i &= 1 \text{ to } 10 \\ & a_i^{[i-1]} = 1 \\ & k_i = - \left\{ \sum_{j=0}^{i-1} a_j^{[i-1]} r'(i-j) \right\} / E^{[i-1]} \\ & a_i^{[i]} = k_i \\ & \text{for } j = 1 \text{ to } i - 1 \\ & \quad a_j^{[i]} = a_j^{[i-1]} + k_i a_{i-j}^{[i-1]} \\ & \text{end} \\ E^{[i]} &= (1 - k_i^2) E^{[i-1]} \\ \text{end} \end{aligned} \quad (1.8)$$

Po získání koeficientů LP filtru a_i pomocí Levinson-Durbinova algoritmu jsou dále tyto koeficienty konvertovány na spektrální páry (LSP). Pro LP filtr desátého řádu jsou LSP koeficienty definovány jako kořeny následujících polynomů [5]:

$$F'_1(z) = A(z) + z^{-11}A(z^{-1}) \quad (1.9)$$

$$F'_2(z) = A(z) - z^{-11}A(z^{-1})$$

Vyřešením rovnic 1.9, by jsme přišli na to, že $F'_1(z)$ má kořen $z = -1$ a $F'_2(z)$ kořen $z = 1$. Tyto kořeny jsou eliminovány zavedením nových polynomů:

$$F_1(z) = F'_1(z)/(1 + z^{-1}) \quad (1.10)$$

$$F_2(z) = F'_2(z)/(1 - z^{-1})$$

Každý polynom má pět kořenů na jednotkové kružnici $e^{\pm j\omega_i}$ a mohou být zapsány jako [5]:

$$F_1(z) = \prod_{i=1,3,\dots,9} (1 - 2q_i z^{-1} + z^{-2}) \quad (1.11)$$

$$F_2(z) = \prod_{i=2,4,\dots,10} (1 - 2q_i z^{-1} + z^{-2})$$

Kde: q_i se rovná $\cos(\omega_i)$.

Koeficient ω_i reprezentuje kmitočet spektrálních párů a $\omega_i \in (0, \pi)$. Koeficienty LSF lze získat pomocí následujícího vzorce:

$$\omega_i = \cos^{-1}(q_i) \quad (1.12)$$

Kde: i je 1 až 10.

1.2.3 Širokopásmová energie

Širokopásmová energie E_f je získána podle následujícího vzorce:

$$E_f = 10 \cdot \log_{10} \left[\frac{1}{N} \cdot r(0) \right] \quad (1.13)$$

Kde: N je velikost okna a je rovna 240.

1.2.4 Úzkopásmová energie

Úzkopásmová energie E_l je spočítána podle vzorce:

$$E_l = 10 \cdot \log_{10} \left[\frac{1}{N} \mathbf{h}^T \mathbf{R} \mathbf{h} \right] \quad (1.14)$$

Kde: N je velikost okna a je rovna 240,
 \mathbf{R} je matice autokorelačních koeficientů v Töeplitzově tvaru,
 \mathbf{h} je vektor impulzní odezvy FIR filtru.

1.2.5 Střední počet průchodu nulou

Střední počet průchodu nulou Z_c je vypočítán pro každý rámeček jako:

$$Z_c = \frac{1}{2M} \sum_{i=0}^{M-1} \{ |\text{sgn}[x(i)] - \text{sgn}[x(i-1)]| \} \quad (1.15)$$

Kde: M je rovno 80,
 $x(i)$ je předzpracovaný vstupní signál.

1.3 Inicializace parametrů šumu na pozadí

Pro prvních N_i rámečků je průměr spektrálních parametrů šumu na pozadí označován jako $\{\overline{LSF}_i\}_{i=1}^p$ inicializován jako průměr $\{LSF_i\}_{i=1}^p$ všech počátečních rámečků. Průměr středního počtu průchodu nulou šumu na pozadí je označen jako \overline{Zc} a je inicializován jako průměr středního počtu průchodu nulou Zc z rámečků.

Průběžný průměr energie šumu na pozadí, označován jako $\overline{E_f}$ a úzkopásmové energie šumu na pozadí označované jako $\overline{E_l}$, jsou inicializovány následovně.

První inicializační procedura používá \overline{En} , definovanou jako průměrná energie E_f v prvních N_i rámcích. Tyto tři průměry ($\overline{En}, \overline{Zc}, \{LSF_i\}_{i=1}^p$) zahrnují pouze rámce, které mají energii E větší než 15 dB.

Druhá inicializační procedura pokračuje následovně:

if $\overline{En} \leq T_1$ then

$$\overline{E_f} = \overline{En} + K_0$$

$$\overline{E_l} = \overline{En} + K_1$$

else if $T_1 < \overline{En} < T_2$ then

$$\overline{E_f} = \overline{En} + K_2$$

$$\overline{E_l} = \overline{En} + K_3$$

else

$$\overline{E_f} = \overline{En} + K_4$$

$$\overline{E_l} = \overline{En} + K_5$$

Konstanty $K_0, K_1, K_2, K_3, K_4, K_5$ jsou zobrazeny v tabulce 1.1.

1.4 Výpočet dlouhodobé minimální energie

Dlouhodobá minimální energie E_{min} je vypočítána jako minimum E_f z předchozích N_0 rámců. Jakmile je N_0 poměrně velké, tak jako E_{min} se použije minimální hodnota E_f získaná v předchozích výpočtech.

1.5 Výpočet rozdílových parametrů

Čtyři rozdílové parametry jsou vypočítány z aktuálních průměrů šumu na pozadí pro daný rámec a z průběžných průměrů šumu na pozadí.

1.5.1 Spektrální zkreslení

Hodnota spektrálního zkreslení je vypočtena jako součet druhých mocnin rozdílů mezi aktuálním rámcem $\{LSF_i\}_{i=1}^p$ a průběžným šumem na pozadí jako $\{\overline{LSF}_i\}_{i=1}^p$:

$$\Delta S = \sum_{i=1}^p (LSF_i - \overline{LSF}_i)^2 \quad (1.16)$$

1.5.2 Rozdíl širokopásmové energie

Rozdíl širokopásmové energie je vypočítán jako rozdíl mezi širokopásmovou energií E_f platnou pro aktuální rámeček a průběžným průměrem širokopásmové energie šumu pozadí \overline{E}_f :

$$\Delta E_f = \overline{E}_f - E_f \quad (1.17)$$

1.5.3 Rozdíl úzkopásmové energie

Rozdíl úzkopásmové energie je vypočítán jako rozdíl mezi úzkopásmovou energií E_l platnou pro aktuální rámeček a průběžným průměrem úzkopásmové energie šumu na pozadí \overline{E}_l :

$$\Delta E_l = \overline{E}_l - E_l \quad (1.18)$$

1.5.4 Rozdíl středního počtu průchodu nulou

Rozdíl středního počtu průchodu nulou je vypočítán jako rozdíl mezi středním počtem průchodů nulou Z_c platným pro aktuální rámeček a průběžným průměrem středního počtu průchodu nulou šumu na pozadí \overline{Z}_c :

$$\Delta Z_c = \overline{Z}_c - Z_c \quad (1.19)$$

1.6 Více hraniční VAD rozhodnutí

Počáteční VAD rozhodnutí je označeno jako I_{VD} a je nastaveno na 0 ("FALSE") pokud vektor rozdílových parametrů leží uvnitř neřečové oblasti. Jinak je počáteční VAD rozhodnutí nastaveno na 1 ("TRUE"). Čtrnáct více hraničních rozhodnutí ve čtyřrozměrném prostoru je definováno:

```
if  $\Delta S > a_1 \cdot \Delta Z_c + b_1$  then  $I_{VD} = 1$ 
if  $\Delta S > a_2 \cdot \Delta Z_c + b_2$  then  $I_{VD} = 1$ 
if  $\Delta E_f < a_3 \cdot \Delta Z_c + b_3$  then  $I_{VD} = 1$ 
if  $\Delta E_f < a_4 \cdot \Delta Z_c + b_4$  then  $I_{VD} = 1$ 
if  $\Delta E_f < b_5$  then  $I_{VD} = 1$ 
if  $\Delta E_f < a_6 \cdot \Delta S + b_6$  then  $I_{VD} = 1$ 
if  $\Delta S > b_7$  then  $I_{VD} = 1$ 
if  $\Delta E_l < a_8 \cdot \Delta Z_c + b_8$  then  $I_{VD} = 1$ 
if  $\Delta E_l < a_9 \cdot \Delta Z_c + b_9$  then  $I_{VD} = 1$ 
```

if $\Delta E_l < b_{10}$ then $I_{VD} = 1$
 if $\Delta E_l < a_{11} \cdot \Delta S + b_{11}$ then $I_{VD} = 1$
 if $\Delta E_l > a_{12} \cdot \Delta E_f + b_{12}$ then $I_{VD} = 1$
 if $\Delta E_l < a_{13} \cdot \Delta E_f + b_{13}$ then $I_{VD} = 1$
 if $\Delta E_l < a_{14} \cdot \Delta E_f + b_{14}$ then $I_{VD} = 1$

Pokud není žádná ze čtrnácti podmínek pravdivá tak $I_{VD} = 0$.

1.7 Vyhlazení VAD rozhodnutí

Počáteční VAD rozhodnutí je vyhlazeno, aby respektovalo vlastnost řečového signálu, kterou je dlouhodobá stacionarita. Vyhlazení je provedeno ve čtyřech krocích.

Označení, že k vyhlazení došlo je definováno jako v_flag . Je nastaveno na nulu pokaždé, když je provedeno VAD vyhlazení. Samotné VAD vyhlazení je označeno jako S_{VD}^0 . Horní index značí, o který rámec se jedná. VAD vyhlazení se provádí v aktuálním, minulém a předminulém rámci a označuje se jako $S_{VD}^0, S_{VD}^{-1}, S_{VD}^{-2}$. Na začátku je S_{VD}^{-1} nastaveno na 1 a S_{VD}^{-2} taktéž na 1. VAD vyhlazení v aktuálním rámci S_{VD}^0 přebírá hodnotu z více hraničního VAD rozhodnutí, tzn. $S_{VD}^0 = I_{VD}$. První krok vyhlazení je:

if $(I_{VD} = 0)$ and $(S_{VD}^{-1} = 1)$ and $(E > \bar{E}_f + T_3)$ then $S_{VD}^0 = 1$ and $v_flag = 1$

Pro druhý krok vyhlazení je definován parametr F_{VD}^{-1} a vyhlazovací čítač C_e . Počáteční hodnota F_{VD}^{-1} je 1 a vyhlazovací čítač C_e začíná čítat od 0. Energie předchozího rámce je označena jako E_{-1} . Druhý krok vyhlazení je:

if $(F_{VD}^{-1} = 1)$ and $(I_{VD} = 0)$ and $(S_{VD}^{-1} = 1)$ and $(S_{VD}^{-2} = 1)$ and $(|E_f - E_{-1}| \leq T_4)$ {
 $S_{VD}^0 = 1$
 $v_flag = 1$
 $C_e = C_e + 1$
 if $(C_e \leq N_1)$ {
 $F_{VD}^{-1} = 1$
 }
 else {
 $F_{VD}^{-1} = 0$
 }
 }

```

}
else
 $F_{VD}^{-1} = 1$ 

```

Pro třetí krok vyhlazení definujeme čítač spojitosti šumu C_s , který je zpočátku nastaven na 0. Když je $S_{VD}^0 = 0$ tak čítač C_s je zvýšen o 1. Třetí krok vyhlazení je:

```

if ( $S_{VD}^0 = 1$ ) and ( $C_s > N_2$ ) and ( $E_f - E_{-1} \leq T_5$ ) {
     $S_{VD}^0 = 0$ 
     $C_s = 0$ 
}
if ( $S_{VD}^0 = 1$ )
     $C_s = 0$ 

```

Ve čtvrtém kroku je provedeno VAD rozhodnutí, pokud je splněna následující podmínka:

```

if [ $(E_f < \overline{E}_f + T_6)$  and ( $frm_{count} > N_0$ ) and ( $v\_flag = 0$ )] then  $S_{VD}^0 = 0$ 

```

1.8 Aktualizace parametrů šumu na pozadí

Nakonec je provedena aktualizace parametrů šumu na pozadí. Následující podmínky jsou testovány, a pokud jsou splněny, tak je provedena aktualizace.

```

if ( $E_f < \overline{E}_f + T_6$ ) then update

```

Šum na pozadí je aktualizován pomocí autoregresního modelu prvního řádu. Rozdílové AR koeficienty jsou použity pro různé parametry. Jsou použity, pokud se jedná o začátek promluvy, nahrávky nebo pokud je detekována velká změna šumových charakteristik.

Nechť β_{E_f} je AR koeficient pro aktualizaci \overline{E}_f , β_{E_l} je AR koeficient pro aktualizaci \overline{E}_l , β_{ZC} je AR koeficient pro aktualizaci \overline{ZC} a β_{LSF} je AR koeficient pro aktualizaci $\{\overline{LSF}_i\}_{i=1}^p$. C_n označuje součet všech rámců, v nichž se provádí aktualizace. AR aktualizace se provádí následovně.

$$\begin{aligned}\bar{E}_f &= \beta_{E_f} \cdot \bar{E}_f + (1 - \beta_{E_f}) \cdot E_f \\ \bar{E}_l &= \beta_{E_l} \cdot \bar{E}_l + (1 - \beta_{E_l}) \cdot E_l \\ \bar{ZC} &= \beta_{ZC} \cdot \bar{ZC} + (1 - \beta_{ZC}) \cdot ZC \\ \bar{LSF}_i &= \beta_{LSF} \cdot \bar{LSF}_i + (1 - \beta_{LSF}) \cdot LSF_i \quad i = 1, \dots, 10\end{aligned}$$

Potom budú aktualizované E_f a C_n .

if [(frame count > N_0) and ($\bar{E}_f < E_{min}$)] then

```
{
 $\bar{E}_f = E_{min}$ 
 $C_n = 0$ 
}
```

Tabulka 1.1: Konstanty použité v detektoru řečové aktivity podle ITU-T G.729 [1].

| Název | Konstanta | Název | Konstanta |
|----------|------------|----------|-----------|
| N_i | 32 | N_1 | 4 |
| N_0 | 128 | N_2 | 10 |
| K_0 | 0 | T_1 | 671088640 |
| K_1 | -53687091 | T_2 | 738197504 |
| K_2 | -67108864 | T_3 | 26843546 |
| K_3 | -93952410 | T_4 | 40265318 |
| K_4 | -134217728 | T_5 | 40265318 |
| K_5 | -161061274 | T_6 | 40265318 |
| a_1 | 23488 | b_1 | 28521 |
| a_2 | -30504 | b_2 | 19446 |
| a_3 | -32768 | b_3 | -32768 |
| a_4 | 26214 | b_4 | -19661 |
| a_5 | 0 | b_5 | -30802 |
| a_6 | 28160 | b_6 | -19661 |
| a_7 | 0 | b_7 | 30199 |
| a_8 | 16384 | b_8 | -22938 |
| a_9 | -19065 | b_9 | -31576 |
| a_{10} | 0 | b_{10} | -17367 |
| a_{11} | 22400 | b_{11} | -27034 |
| a_{12} | 30427 | b_{12} | 29959 |
| a_{13} | -24576 | b_{13} | -29491 |
| a_{14} | 23406 | b_{14} | -28087 |

2 Detektor řečové aktivity podle ITU-T G.723.1

Detektor řečové aktivity je součástí celého kodeku G.723.1 vytvořeného organizací ITU-T. Kodek se může použít pro komprimování jak řečového signálu, tak ostatního audio signálu multimediálních služeb, které vyžadují velmi nízkou přenosovou rychlost. Přenosová rychlost dat po komprimaci je 6,3 kbit/s nebo 5,3 kbit/s. Tuto rychlost můžeme měnit pro každý rámeček. Rámeček má dobu trvání 30 ms a při vzorkovací frekvenci 8 kHz obsahuje 240 vzorků. Kodér je navržen pro práci se signálem získaným pomocí PCM s rozlišením 16 bitů a vzorkovacím kmitočtem 8 kHz.

Při měření subjektivní kvality řeči, dosáhl kodek G.723.1 na stupnici MOS, skóre 3,9 pro přenosovou rychlost 6,3 kbit/s a 3,65 pro přenosovou rychlost 5,3 kbit/s [3].

Funkcí detektoru řečové aktivity podle ITU-T G.723.1, je každých 30 ms rozhodnout zda se jedná o rámeček s řečovou aktivitou či ne. Detektor je v podstatě energetický detektor. Energie inverzně filtrovaného signálu je porovnávána s nastaveným prahem. Řeč je indikována kdykoli je dosaženo prahu. Velikost prahu je vypočítána ve dvou krocích. V prvním kroku je úroveň hluku aktualizovaná na základě předchozí hodnoty a energie filtrovaného signálu. V druhém kroku je velikost prahu vypočítána z úrovně hluku přes logaritmickou aproximaci.

2.1 Rozdělení dat

Vstupní signál $y(n)$ je rozdělen do rámečků o 240 vzorcích označených $s(n)$. Každý rámeček je dále rozdělen na dvě části o 120 vzorcích. Tyto rámečky se používají pro výpočet odhadu základní periody řeči. Každá část je znovu rozdělena na dvě takže vzniknou čtyři subramečky o 60-ti vzorcích. Subramečky se používají pro LPC analýzu.

2.2 Filtr typu horní propust

Tento blok odfiltrává stejnosměrnou složku ze vstupního signálu $s(n)$. Výstupní signál je označen jako $x(n)$. Přenosová funkce filtru je.

$$H(z) = \frac{1 - z^{-1}}{1 - \frac{127}{128}z^{-1}} \quad (2.1)$$

2.3 LPC analýza

LPC analýza desátého řádu je uplatňována na vstupní signál $x(n)$. Rámec pro výpočet LPC má velikost 180 vzorků a aktuální subrámec se nachází uprostřed. Každý LPC rámec je váhován Hammingovým oknem, které je vyjádřeno.

$$w(n) = 0,54 - 0,46 \cdot \cos\left(\frac{2\pi n}{N-1}\right) \quad (2.2)$$

Kde: N je velikost okna a je rovna 180.

Váhovaný signál je potom získán pouhým násobením podle následujícího vztahu.

$$x'(n) = w(n) \cdot x(n) \quad (2.3)$$

Z takto získaného váhovaného signálu je potom v každém rámci vypočítáno jedenáct autokorelačních koeficientů $r(k)$ podle vztahu.

$$r(k) = \sum_{n=k}^{239} x'(n) \cdot x'(n-k) \quad (2.4)$$

Kde: k je rovno 0 až 10.

Podobně jako u detektoru řečové aktivity podle ITU-T G.729 se pro výpočet LPC koeficientů používají modifikované autokorelační koeficienty, které se získají jako.

$$\begin{aligned} R(0) &= 1,001 \cdot r(0) \\ R(k) &= w_{lag}(k) \cdot r(k) \end{aligned} \quad (2.5)$$

Kde: k je rovno 0 až 10,
 w_{lag} je konstanta uvedená v [2].

A z modifikovaných autokorelačních koeficientů jsou vypočteny LPC koeficienty a_i pomocí Levinson - Durbinova algoritmu uvedeného v (1.8).

2.4 Váhovací filtr pro vnímání formantů

Vstupním signálem tohoto bloku je signál $x(n)$. Tento signál je rozdělen do čtyř subrámců o délce 60-ti vzorků a každý subrámec je filtrován filtrem s přenosovou funkcí $W_i(z)$.

$$W_i(z) = \frac{1 - \sum_{j=1}^{10} a_{ij} \cdot z^{-j} \cdot \gamma_1^j}{1 - \sum_{j=1}^{10} a_{ij} \cdot z^{-j} \cdot \gamma_2^j} \quad (2.6)$$

Kde: a_{ij} je j -tý LPC koeficient pro i -tý rámeček,
 i je rovno 0 až 3,
 γ_1 je konstanta, která je rovna 0,9,
 γ_2 je konstanta, která je rovna 0,5.

Takto vyfiltrovaný signál je potom označen jako vážený signál $\{f(n)\}_{n=0..239}$.

2.5 Odhad základní periody řeči

Každý rámeček se provádí odhad základní periody řeči L_{OL} dvakrát. Jeden odhad pro první dva subrámce a druhý pro zbylé dva subrámce ze vstupního signálu $f(n)$. Používá se maximalizační metoda pro odhad základní periody řeči. Nejprve se vypočítá autokorelační funkce.

$$C_{OL}(j) = \frac{\left[\sum_{n=0}^{119} f(n) \cdot f(n-j) \right]^2}{\sum_{n=0}^{119} f(n-j) \cdot f(n-j)} \quad (2.7)$$

Kde: j je rovno 18 až 142.

A potom je vybráno maximum $C_{OL}(j)$, kde index j je základní perioda řeči L_{OL} pro dva subrámce. Při hledání nejlepšího indexu, je dána přednost menším indexům, abychom se vyhnuli jejím násobkům. Maximum z $C_{OL}(j)$ je vyhledáváno s počáteční hodnotou $j = 18$. Při každém nalezení maxima je hodnota porovnávána s předchozím maximem $C_{OL}(j')$. Pokud je rozdíl mezi indexy j a j' menší než 18 a $C_{OL}(j) > C_{OL}(j')$ je jako nové maximum vybrán index j . Jestliže je rozdíl mezi indexy větší nebo roven 18 je nové maximum vybráno pouze tehdy když $C_{OL}(j') > C_{OL}(j)$ o 1,25 dB.

2.6 Výpočet příznaku povolení adaptace

Příznak povolení adaptace, označen jako Aen_t pro aktuální rámec t , je použit, aby detektor řečové aktivity si byl jist, že úroveň hluku je vypočítána pouze, když v rámci není přítomná řeč. Je to založeno na skutečnosti, že hluk v pozadí, nebo ticho není řečový ani sinusový signál.

- Detekce znělého signálu:
Pro zjištění zda se jedná o řečový či neřečový signál je použito posledních čtyř hodnot L_{OL} . To znamená, že pracuje s aktuálním a předešlým rámcem. Jako první zjistíme minimální hodnotu L_{OL}^{min} . Čítač pc označuje kolik hodnot L_{OL} leží v intervalu $\langle L_{OL}^{min} - 3; L_{OL}^{min} + 3 \rangle$. Pokud čítač $pc = 4$, tak je signál označen jako znělý.
- Detekce sinusového signálu
Označme druhý koeficient odrazu vypočtený pomocí Levinson-Durbinova algoritmu pro každý subrámec $i = 0,1,2,3$ rámce t jako $k_i^t[2]$. Potom pokud je $k_i^t[2] \geq 0,95$ pro minimálně 14 z posledních 15 hodnot, tak je sinusový signál detekován a $SinD = 1$.

Výpočet příznaku povolení adaptace je potom následující.

$$\begin{aligned} Aen_t &= Aen_{t-1} + 2 && \text{if } pc = 4 \text{ or } SinD = 1 \\ Aen_t &= Aen_{t-1} - 1 && \text{otherwise} \end{aligned} \quad (2.8)$$

Aen_t může nabývat hodnot 0 až 6.

2.7 Inverzní filtrace

Vstupní signál $\{s(n)\}_{n=60\dots239}$ je inverzně filtrován FIR filtrem $A_{no}(z)$ s koeficienty $\{a_{no}(j)\}_{j=1\dots10}$. Tento filtr poskytuje odhad LPC filtru spojeného s aktuálním šumem na pozadí [2]. Vyfiltrovaný signál $e_t(n)$ je dán rovnicí.

$$e_t(n) = s(n) + \sum_{j=1}^{10} a_{no}(j) \cdot s(n-j) \quad (2.9)$$

Kde: n je rovno 60 až 239.

Pro výpočet koeficientů $\{a_{no}(j)\}_{j=1\dots10}$ je nutné vypočítat zprůměrovaný LPC filtr $\bar{A}_p(z)$, který se vypočítá z autokorelačních koeficientů $\bar{R}_p(j)$ použitím Levinson - Durbinova algoritmu (1.8).

Autokorelační koeficienty $\bar{R}_p(j)$ jsou vypočteny z posledních třech rámců podle následujícího vztahu.

$$\bar{R}_p(j) = \sum_{k=t-3}^{t-1} R^k(j) \quad (2.10)$$

Kde: $R^k(j)$ je kumulovaná autokorelační funkce,
 j je rovno 0 až 10.

Pro každý subrámec jsou vypočítány autokorelační koeficienty podle vzorce (2.5). Pro aktuální rámeček t označíme autokorelační koeficienty jako $R_i(j)$, $j = 0$ až 10 a index i značí subrámeček. Potom kumulovaná autokorelační funkce je dána vztahem.

$$R^t(j) = \sum_{i=0}^3 R_i(j) \quad (2.11)$$

Kde: i je index subrámečku,
 j je rovno 0 až 10.

Koeficienty $a_{no}(j)$ jsou získány z koeficientů $\bar{a}_p(j)$ podle následujícího vztahu.

$$\text{if } Aen_t = 0 \text{ then } a_{no}(j) = \bar{a}_p(j) \quad (2.12)$$

Kde: Aen_t je příznak povolení adaptace pro aktuální rámeček t ,
 $a_{no}(j)$ jsou koeficienty FIR filtru $A_{no}(z)$,
 $\bar{a}_p(j)$ jsou koeficienty zprůměrovaného LPC filtru $\bar{A}_p(z)$,
 j je rovno 1 až 10.

2.8 Výpočet filtrované energie

Energie Enr_t je vypočtena z inverzně filtrovaného signálu podle vzorce (2.9) pro aktuální rámeček pomocí.

$$Enr_t = \frac{1}{80} \sum_{n=60}^{239} e_t^2(n) \quad (2.13)$$

2.9 Výpočet úrovně šumu

Úroveň šumu $Nlev_t$ je aktualizována na základě předchozí hodnoty energie Enr_{t-1} , a aktuální hodnoty příznaku povolení adaptace Aen_t . Dynamický rozsah úrovně šumu v rámci t je limitován do rozsahu $[Nlev_{min}, Nlev_{max}]$. Jestliže $Nlev_{t-1} > Enr_{t-1}$, potom je úroveň šumu poprvé oříznuta.

$$Nlev_t = \begin{cases} 0,25 \cdot Nlev_{t-1} + 0,75 \cdot Enr_{t-1} & \text{if } Nlev_{t-1} > Enr_{t-1} \\ Nlev_{t-1} & \text{otherwise} \end{cases} \quad (2.14)$$

Potom je $Nlev_t$ zvýšena jestliže to příznak povolení adaptace povoluje, jinak je trochu snížena.

$$Nlev_t = \begin{cases} 1,03125 \cdot Nlev_t & \text{if } Aen_t = 0 \\ 0,9995 \cdot Nlev_t & \text{otherwise} \end{cases} \quad (2.15)$$

Nakonec je úroveň šumu $Nlev_t$ limitována do rozsahu $Nlev_{min} = 128$, $Nlev_{max} = 131071$.

2.10 Výpočet prahu

Vztah mezi úrovní šumu $Nlev_t$ pro aktuální rámec t a hodnotou prahu je definován logaritmickou aproximací podle následujícího vzorce.

$$Thr = \begin{cases} 5,012 & \text{if } Nlev = 128 \\ 10^{0,7 - 0,005 \log_2 \frac{Nlev}{128}} & \text{if } 128 < Nlev < 16384 \\ 2,239 & \text{if } Nlev \geq 16384 \end{cases} \quad (2.16)$$

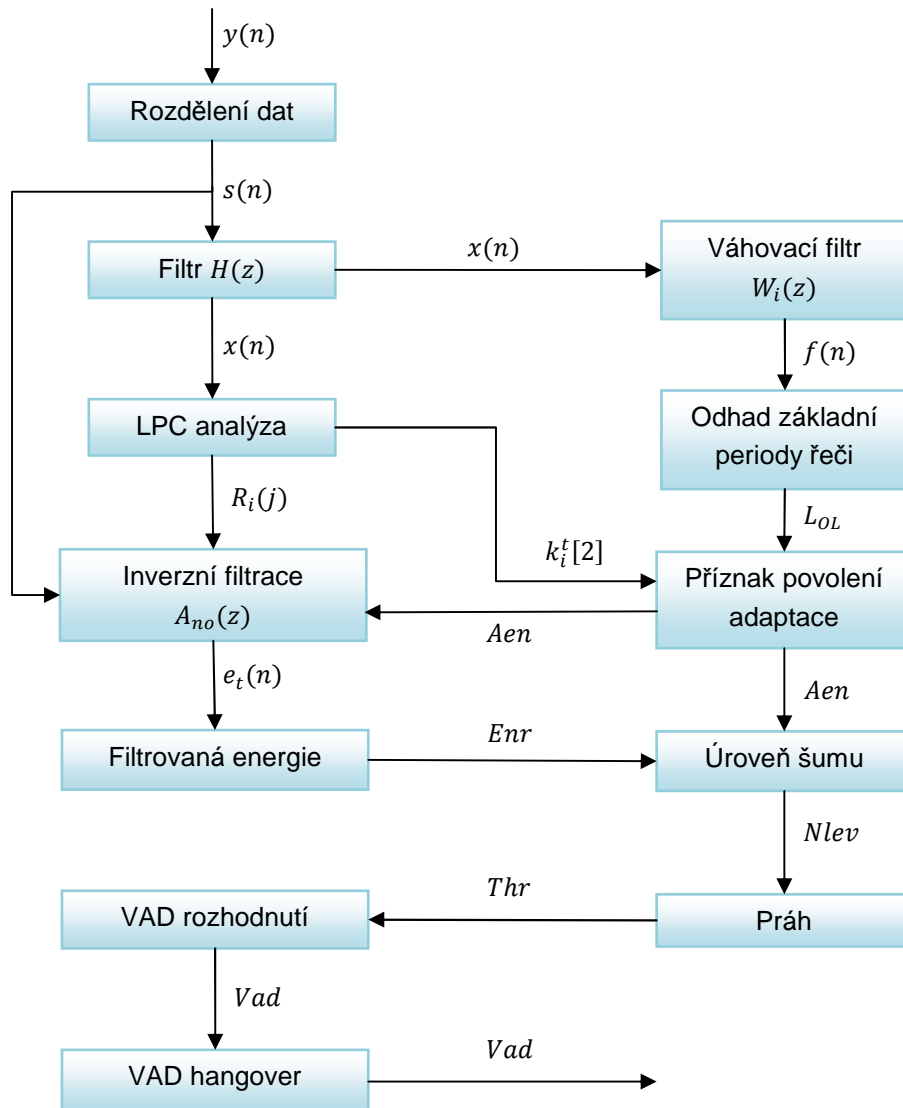
2.11 Rozhodnutí VAD

Rozhodnutí detektoru řečové aktivity je na základě porovnání mezi prahem Thr a aktuální energií Enr_t . Výsledná hodnota je potom dána následujícím vztahem.

$$Vad_t = \begin{cases} 1 & \text{if } Enr_t \geq Thr \\ 0 & \text{if } Enr_t < Thr \end{cases} \quad (2.17)$$

2.12 VAD vyhlazení

Prvních pár rámců je původně označeno jako neřečové, ale ve skutečnosti měly být označeny jako řečové. To znamená, že VAD nezachytil hlásky s nízkou energií. Tento problém právě řeší VAD vyhlazení. Vyhlazení probíhá tak, že jakmile je detekován shluk větší nebo roven dvěma rámcům, tak minulých šest rámců označíme jako řečové [2].



Obrázek 2.1: Schéma detektoru řečové aktivity podle ITU-T G.723.1.

3 Implementace VAD podle ITU-T G.729 v signálovém procesoru

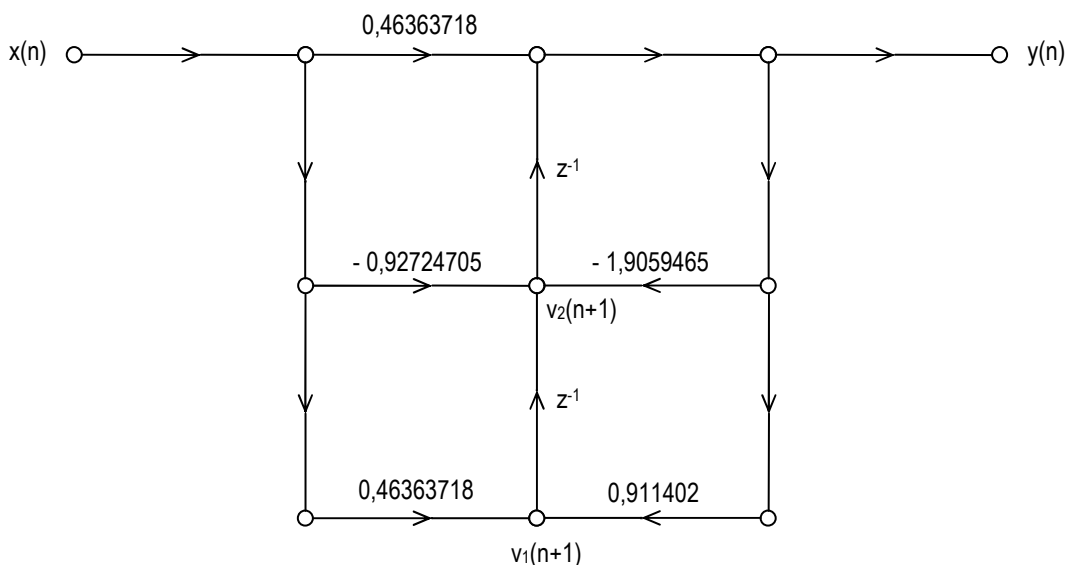
V této kapitole bude popsána implementace VAD podle ITU-T G.729 na signálovém procesoru TMS320C6416. Jedná se o signálový procesor s pevnou řádovou čárkou, typu VLIW, založený na architektuře VelocityTI.2, kterou vyvinula firma Texas Instruments. Při kmitočtu hodinového signálu 1 GHz dosahuje signálový procesor výpočetního výkonu až 8000 MIPS.

3.1 Preemfáze

Této problematice se podrobněji věnujeme v kapitole 1.1, zde se budeme zabývat pouze implementací. Přenosová funkce IIR filtru je uvedena ve vzorci (3.1).

Při návrhu IIR filtru, bychom měli zkontrolovat jeho stabilitu, ale zde vycházím ze standardu a filtr zde by měl být navržen tak, aby nestabilita nemohla vzniknout. Jen pro zajímavost jsem si zobrazil nulové body a póly v MATLABU funkcí `zplane(b, a)` a ujistil se, že filtr je stabilní.

Dalším krokem v návrhu je výběr vhodné struktury. Zvolil jsem první kanonickou strukturu, taktéž označovanou jako II. Direct form transposed. Tuhle strukturu jsem vybral proto, že u ní nedochází tak často k saturaci stavových proměnných. Graf signálových toků první kanonické formy je zobrazen na obr. 3.1.



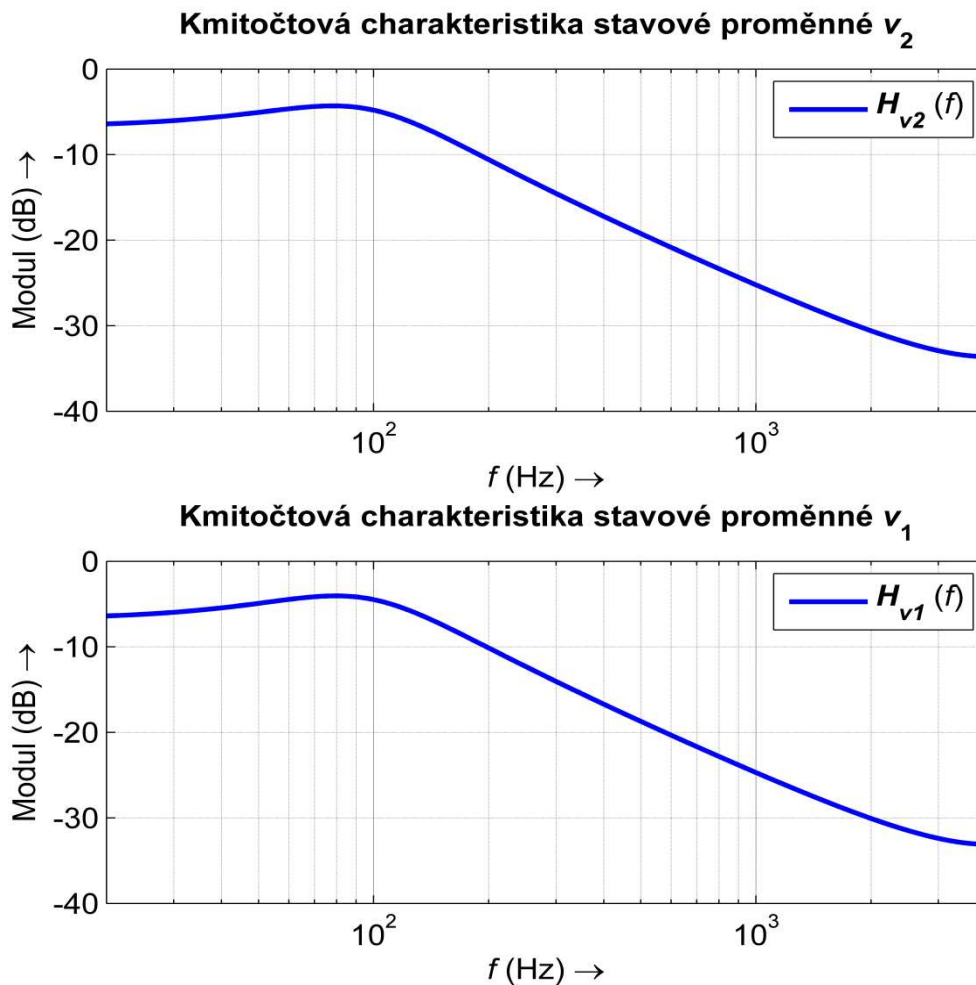
Obrázek 3.1: První kanonická forma pro implementaci preemfáze VAD G.729.

Před samotnou implementací filtru na signálový procesor jsem si ověřil, jestli může dojít k saturaci stavových proměnných. Pomocí Masonova pravidla jsem si vypočítal přenosové funkce pro stavové proměnné $v_1(n)$ a $v_2(n)$, které vyšly následovně.

$$H_{v_1}(z) = \frac{0,043579 - 0,041077z^{-1}}{1 - 1,9059465z^{-1} + 0,911402z^{-2}} \quad (3.1)$$

$$H_{v_2}(z) = \frac{0,041077 - 0,038572z^{-1}}{1 - 1,9059465z^{-1} + 0,911402z^{-2}} \quad (3.2)$$

Z přenosových funkcí (3.1) a (3.2) jsem vypočítal modulovou kmitočtovou charakteristiku zobrazenou na obr. 3.2. Z této charakteristiky je vidět, že stavové proměnné nemohou přetéct pro harmonický signál, protože přenos na všech kmitočtech je menší než 0 dB.



Obrázek 3.2: Modulové kmitočtové charakteristiky stavových proměnných v_2 (nahore) a v_1 (dole).

Tím pádem můžeme přistoupit k implementaci na signálový procesor. První kanonická forma lze vyjádřit podle diferenčních rovnic jako.

$$\begin{aligned} v_1(n+1) &= b_1x(n) - a_1y(n) + v_2(n) \\ v_2(n+1) &= b_0x(n) - a_0y(n) \\ y(n) &= \frac{b_2}{a_2}x(n) + \frac{1}{a_2}v_1(n) \end{aligned} \tag{3.3}$$

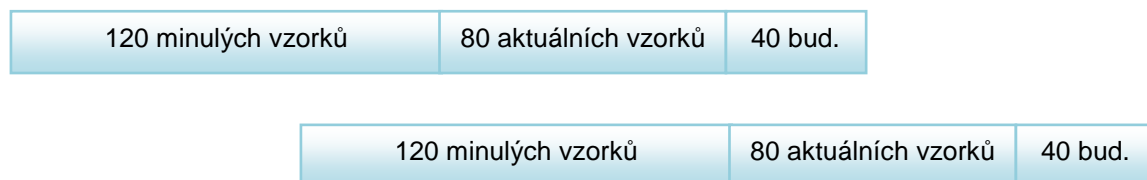
Abychom se vyhnuli dělení, které je na signálovém procesoru pomalé, tak musíme zajistit, aby koeficient a_2 byl roven jedné. Koeficient a_1 je větší než jedna a tudíž nelze vyjádřit ve dvojkovém doplňku. Z toho důvodu upravíme diferenční rovnice do následujících tvarů.

$$\begin{aligned} y(n) &= \frac{2}{a_2} \left(\frac{b_2}{2}x(n) + \frac{1}{2}v_1(n) \right) \\ v_1(n+1) &= 2 \left(\frac{b_1}{2}x(n) - \frac{a_1}{2}y(n) + \frac{1}{2}v_2(n) \right) \\ v_2(n+1) &= 2 \left(\frac{b_0}{2}x(n) - \frac{a_0}{2}y(n) \right) \end{aligned} \tag{3.4}$$

Takto upravené rovnice použijeme pro implementaci IIR filtru na signálovém procesoru. V zápisu koeficientů, které jsou uloženy v poli koef, si můžeme dovolit vynechat koeficient b_0 resp. b_2 , protože nabývají stejné hodnoty. Taktéž koeficient a_2 nemá smysl uchovávat. Výsledné pole je potom délky 4 místo 6.

3.2 Váhovací funkce a výpočet autokorelace

Výstupní signál z IIR filtru je potom rozdělen do rámců o délce 240 vzorků. Překrývání rámců je zobrazeno na obr.3.3.



Obrázek 3.3: Překrývání rámců podle ITU-T G.729.

Rozdělení do rámců je provedeno pomocí kruhového bufferu. Dále je na celý rámeček uplatněn vzorec (1.3). Hodnoty proměnné $w_{lp}(n)$ jsou získány pomocí skriptu `okno.m`. Tyto hodnoty jsou potom uloženy do pole `okno` délky 240. Aplikováním vzorce (1.3) získáme data potřebná pro výpočet korelace, která je základní veličinou pro výpočet dalších parametrů. Tyto data jsou uložena v poli `poleKorelace` délky 240.

Samotný výpočet korelace je realizován pomocí funkce `korelace`. Vstupním parametrem je proměnná `poleKorelace` a výstupní hodnoty jsou ukládány do pole `r` a `corr32` délky 13. V poli `r` jsou uloženy autokorelační koeficienty v 16b rozlišení a v poli `corr32` jsou koeficienty ve 32b rozlišení. Ve funkci `korelace` jsou postupně použity vzorce (1.4), (1.5). Protože koeficienty $r(k)$ mohou nabývat vyšších hodnot než jedna, tak je musíme posunout o správný počet bitů doprava a tento posun si zapamatovat. Jakmile budeme potřebovat sčítat autokorelační koeficienty s další proměnnou, musíme napřed posunout proměnnou o daný počet bitů doprava. Tímto zajistíme, že se sčítají bity se stejnou váhou.

Výpočet modifikovaných autokorelačních koeficientů $r'(k)$ se provádí pomocí funkce `korelaceWLAG` ve které je implementován vzorec (1.6). Tyto koeficienty jsou dále použity pro výpočet úzkopásmové a širokopásmové energie.

3.3 Kmitočet spektrálních párů

Při výpočtu kmitočtu spektrální párů (LSF) jsme postupovali podle kapitoly 1.2.2 a výpočet je proveden ve třech základních krocích:

- Výpočet LPC koeficientů
- Výpočet LSP
- Výpočet LSF

Pro výpočet LPC koeficientů desátého řádu je použita funkce Levinson. Tato funkce aplikuje Levinson - Durbinův algoritmus popsáný ve vzorci (1.8). Vstupními hodnotami je vektor modifikovaných autokorelačních koeficientů $r'(k)$. Výstupními hodnotami jsou LPC koeficienty a_j . Vstupní vektor normovaných autokorelačních koeficientů je délky 13, ale použity jsou pouze koeficienty $r'(0)$ až $r'(9)$. Použitý rozsah udává řád LPC analýzy. Pokud by byl řád LPC označen jako Q , potom by byl použitý rozsah $r'(0)$ až $r'(Q - 1)$. Ve funkci `korelace` byly vypočítány koeficienty $r'(0)$ až $r'(12)$, protože se budou používat pro výpočet úzkopásmové energie. LPC koeficienty a_j jsou uloženy ve formátu Q2.13 v poli `a_lpc` délky 11.

Výpočet LSP koeficientů se provádí pomocí funkce Az_1sp, kde vstupními parametry jsou LPC koeficienty a výstupními koeficienty LSP ve formátu Q0.15. V první části jsou nalezeny koeficienty polynomů $F_1(z)$ a $F_2(z)$ definované ve vzorci (1.10). Protože koeficienty jsou symetrické, tak stačí nelézt prvních pět koeficientů podle následujícího rekurzivního vztahu [1].

$$\begin{aligned} f_1(i+1) &= a_{i+1} + a_{10-i} - f_1(i) \\ f_2(i+1) &= a_{i+1} - a_{10-i} + f_2(i) \end{aligned} \quad (3.5)$$

Kde: i je rovno 0 až 4,
 $f_1(0)$ je rovno 1,
 $f_2(0)$ je rovno 1.

Nalezení LSP koeficientů se provádí na intervalu 0 až π . Tento interval obsahuje 60 rovnoměrně rozložených bodů. Změna znaménka znamená existenci kořenu [1] a poté je tento interval podělen čtyřikrát pro lepší rozlišení a lineárně interpolován. Kořeny jsou hledány přímo v kmitočtové oblasti a polynomy $F_1(z)$ nebo $F_2(z)$ mohou být napsány jako [1].

$$\begin{aligned} F(\omega) &= 2e^{-j5\omega} C(x) \\ C(x) &= T_5(x) + f(1)T_4(x) + f(2)T_3(x) + f(3)T_2(x) + f(4)T_1(x) + \frac{1}{2}f(5) \end{aligned} \quad (3.6)$$

Ve vzorci (3.6) $T_m(x) = \cos(m\omega)$ a m je řád Chebysheva polynomu. $f(i)$ jsou koeficienty $F_1(z)$ nebo $F_2(z)$ vypočítané pomocí vztahu (3.5). Polynom $C(x)$ je vyčíslen jako určitá hodnota $x = \cos(\omega)$ a je nalezen pomocí funkce Chebps_11. Vstupními hodnotami jsou $x = \cos(\omega)$ ve formátu Q0.15, koeficienty $f(i)$ ve formátu Q4.11 a řád Chebysheva polynomu. Výstupní hodnota $C(x)$ je vrácena ve formátu Q1.14. Funkce Chebps_11 vykonává následující vztah.

$$\begin{aligned} &\text{for } k = 4 \text{ down to } 1 \\ &\quad b_k = 2xb_{k+1} - b_{k+2} + f(5 - k) \\ &\text{end} \end{aligned} \quad (3.7)$$

$$C(x) = xb_1 - b_2 + \frac{1}{2}f(5)$$

Kde: b_5 je rovno 1,
 b_6 je rovno 0.

Funkce, která převádí LSP na LSF se nazývá `Lsp_1sf`. Vstupními parametry jsou koeficienty LSP ve formátu Q0.15 a výstupními normalizované hodnoty LSF, které mohou nabývat hodnot 0 až 0,5. Funkce provádí následující.

$$lsf_i = \cos^{-1}(lsp_i) \quad (3.8)$$

3.4 Širokopásmová energie

Funkce, která provádí výpočet širokopásmové energie E_f se nazývá `fullBandEnergy`. Vstupními parametry jsou autokorelační koeficienty $r(k)$ popsané ve vzorci (1.4) a hodnota o kolik bitů byly posunuty. Všechny vstupní parametry jsou získány pomocí funkce `korelace` popsané v kapitole 3.2. Ve funkci `fullBandEnergy` byl v prvotní implementaci aplikován vzorec (1.13), který je vyjádřen jako

$$\begin{aligned} E_f &= 10 \cdot \log_{10} \left[\frac{1}{N} \cdot r(0) \right] = 10 \left(\log_{10}[r(0)] + \log_{10} \left[\frac{1}{N} \right] \right) = \\ &= 10(\log_{10}[r(0)] - 2,380211241711606) . \end{aligned} \quad (3.9)$$

Výstupní hodnota E_f byla ve formátu Q6.9. Později se ale ukázalo, že takto aplikovaný vzorec dává odlišné výsledky od originálního kódu. Tento rozdíl byl způsoben tím, že logaritmus v originálním kódu se vypočítává z celého čísla. Pro lepší znázornění co se myslí logaritmem z celého čísla si uvedeme příklad.

$$\begin{aligned} r(0) &= 1.2496059384779 \cdot 10^{-4} \\ \log_{10}(1.249605938477912 \cdot 10^{-4}) &= -3.903226919572681 \end{aligned}$$

Vyjádříme-li číslo $1.2496059384779 \cdot 10^{-4}$ ve formátu Q2.29 se zaokrouhlením, potom se náš příklad změní následovně.

$$\begin{aligned} r(0) &= 1.249605938477912 \cdot 10^{-4} \cdot 2^{29} = 67088 \\ \log_{10}(67088) &= 4.826644845059493 \end{aligned}$$

Výsledná hodnota logaritmu čísla vyjádřeného ve formátu Q2.29 je větší o výsledek $\log_{10}(2^{29})$, a kdyby se použil vzorec (1.13) uváděný v literatuře [1], dopustili bychom se chyby. Pro výpočet širokopásmové energie E_f jsme ve funkci `fullBandEnergy` použili následující vzorec.

$$E_f = \log_{10} \left[\frac{1}{N} \cdot r(0) \cdot 2^{29} \right] \quad (3.10)$$

Výstupní hodnota E_f je ve formátu Q4.11 oproti prvotní implementaci a to hlavně z důvodu odstranění nutnosti přepočítávání konstant mezi mnou vytvořeným kódem a kódem originálním.

3.5 Střední počet průchodu nulou

Střední počet průchodu nulou Z_c popsán ve vzorci (1.15), je vypočten pomocí funkce `zeroCrossingRate`. Vstupním signálem je signál $s'(n)$, ve formátu Q0.15, popsán ve vzorci (1.3). Výstupní hodnota je taktéž ve formátu Q0.15. Při implementaci jsme nahradili rozdíl funkcí `sign` v absolutní hodnotě pouhým násobením. Pokud je výsledek záporný, tak se jedná o změnu znaménka, kterou indikuje taktéž rozdíl funkcí `sign`. Pokud narazíme na změnu znaménka, přičítáme hodnotu $1/2M$ kde $M = 80$ stejně jako ve vzorci (1.15). V originálním kódu podle ITU-T G.729 přičítají hodnotu $1/M$ nebo výsledek mají ve formátu Q1.14 i když uvádějí formát Q15.

3.6 Úzkopásmová energie

Výpočet úzkopásmové energie provádí funkce `lowBandEnergy`. Vstupními parametry jsou autokorelační koeficienty $r(k)$ a hodnota o kolik bitů byly posunuty. Všechny vstupní parametry jsou získány pomocí funkce `korelace` popsané v kapitole 3.2. V literatuře [1] se uvádí, že výpočet úzkopásmové energie se provádí podle vzorce (1.14).

$$E_l = 10 \cdot \log_{10} \left[\frac{1}{N} \mathbf{h}^T \mathbf{R} \mathbf{h} \right] \quad (1.14)$$

Při detailnější analýze originálního kódu ale nikde žádnou matici respektive dvourozměrné pole, které by ji reprezentovalo, nenajdeme. Vysvětlení a odvození jsem v žádné literatuře nenašel, proto jsem se výpočtu úzkopásmové energie věnoval detailněji. Podíváme-li se na výpočet úzkopásmové energie podle originálního kódu, nalezneme pouze násobení dvou vektorů a logaritmus o základu dvě.

```

acc0 = 0;
for (i=1; i<=NP; i++)
    acc0 = L_mac(acc0, r_h[i], lbf_corr[i]);
acc0 = L_shl(acc0, 1);
acc0 = L_mac(acc0, r_h[0], lbf_corr[0]);
Log2(acc0, &exp, &frac);

```

Později v kódu je převeden logaritmus o základu dvě na logaritmus o základu deset a proveden odečet konstanty, která reprezentuje $\log_{10} \frac{1}{N}$. Takže vzorec (1.14) lze přepsat na následující vzorec, který se používá v originálním kódu.

$$E_l = \log_{10}(\mathbf{R} \cdot \mathbf{a}) - 2,38021124 = \log_{10} \left(\frac{1}{N} \cdot \mathbf{R} \cdot \mathbf{a} \right) \quad (3.11)$$

Kde: \mathbf{R} je vektor autokorelačních koeficientů,
 \mathbf{a} je vektor `lbf_corr` v originálním kódu.

V následujícím textu bude vysvětleno odvození vztahu mezi vzorci (3.11) a (1.14). Jak je vidět ve vzorci (3.11), úzkopásmová energie v originálním kódu vychází desetkrát menší než úzkopásmová energie popsaná v literatuře [1]. Tento rozdíl je ale eliminován tím, že v originálním kódu jsou konstanty pro práci s energiemi desetkrát menší, takže uspořili jedno násobení a výsledek je stejný. Abychom mohli uplatnit vzorec (1.14) musíme z vektoru \mathbf{a} vypočítat vektor \mathbf{h} . Roznásobením výrazu $\mathbf{h}^T \mathbf{R} \mathbf{h}$ dostaneme.

$$\begin{aligned}
& R(0) \cdot [h(0)^2 + h(1)^2 + h(2)^2 + h(3)^2 + h(4)^2 + h(5)^2 + h(6)^2 + h(7)^2 + h(8)^2 + \\
& \quad h(9)^2 + h(10)^2 + h(11)^2 + h(12)^2] + \\
& R(1) \cdot [h(0)h(1) + h(1)h(0) + h(1)h(2) + h(2)h(1) + h(2)h(3) + h(3)h(2) + \\
& \quad h(3)h(4) + h(4)h(3) + h(4)h(5) + h(5)h(4) + h(5)h(6) + h(6)h(5) + \\
& \quad h(6)h(7) + h(7)h(6) + h(7)h(8) + h(8)h(7) + h(8)h(9) + h(9)h(8) + \\
& \quad h(9)h(10) + h(10)h(9) + h(10)h(11) + h(11)h(10) + h(11)h(12) + \\
& \quad h(12)h(11)] + \\
& R(2) \cdot [h(0)h(2) + h(1)h(3) + h(2)h(0) + h(2)h(4) + h(3)h(1) + h(3)h(5) + \\
& \quad h(4)h(2) + h(4)h(6) + h(5)h(3) + h(5)h(7) + h(6)h(4) + h(6)h(8) + \\
& \quad h(7)h(5) + h(7)h(9) + h(8)h(6) + h(8)h(10) + h(9)h(7) + h(9)h(11) + \\
& \quad h(10)h(8) + h(10)h(12) + h(11)h(9) + h(12)h(10)] + \\
& R(3) \cdot [h(0)h(3) + h(1)h(4) + h(2)h(5) + h(3)h(0) + h(3)h(6) + h(4)h(1) + \\
& \quad h(4)h(7) + h(5)h(2) + h(5)h(8) + h(6)h(3) + h(6)h(9) + h(7)h(4) + \\
& \quad h(7)h(10) + h(8)h(5) + h(8)h(11) + h(9)h(6) + h(9)h(12) + \\
& \quad h(10)h(7) + h(11)h(8) + h(12)h(9)] +
\end{aligned}$$

$$\begin{aligned}
& R(4) \cdot [h(0)h(4) + h(1)h(5) + h(2)h(6) + h(3)h(7) + h(4)h(0) + h(4)h(8) + \\
& \quad h(5)h(1) + h(5)h(9) + h(6)h(2) + h(6)h(10) + h(7)h(3) + h(7)h(11) + \\
& \quad h(8)h(4) + h(8)h(12) + h(9)h(5) + h(10)h(6) + h(11)h(7) + \\
& \quad h(12)h(8)] + \\
& R(5) \cdot [h(0)h(5) + h(1)h(6) + h(2)h(7) + h(3)h(8) + h(4)h(9) + h(5)h(0) + \\
& \quad h(5)h(10) + h(6)h(1) + h(6)h(11) + h(7)h(2) + h(7)h(12) + h(8)h(3) + \\
& \quad h(9)h(4) + h(10)h(5) + h(11)h(6) + h(12)h(7)] + \\
& R(6) \cdot [h(0)h(6) + h(1)h(7) + h(2)h(8) + h(3)h(9) + h(4)h(10) + h(5)h(11) + \\
& \quad h(6)h(0) + h(6)h(12) + h(7)h(1) + h(8)h(2) + h(9)h(3) + h(10)h(4) + \\
& \quad h(11)h(5) + h(12)h(6)] + \\
& R(7) \cdot [h(0)h(7) + h(1)h(8) + h(2)h(9) + h(3)h(10) + h(4)h(11) + h(5)h(12) + \\
& \quad h(7)h(0) + h(8)h(1) + h(9)h(2) + h(10)h(3) + h(11)h(4) + h(12)h(5)] + \\
& R(8) \cdot [h(0)h(8) + h(1)h(9) + h(2)h(10) + h(3)h(11) + h(4)h(12) + h(8)h(0) + \\
& \quad h(9)h(1) + h(10)h(2) + h(11)h(3) + h(12)h(4)] + \\
& R(9) \cdot [h(0)h(9) + h(1)h(10) + h(2)h(11) + h(3)h(12) + h(9)h(0) + \\
& \quad h(10)h(1) + h(11)h(2) + h(12)h(3)] + \\
& R(10) \cdot [h(0)h(10) + h(1)h(11) + h(2)h(12) + h(10)h(0) + h(11)h(1) + \\
& \quad h(12)h(2)] + \\
& R(11) \cdot [h(0)h(11) + h(1)h(12) + h(11)h(0) + h(12)h(1)] + \\
& R(12) \cdot [h(0)h(12) + h(12)h(0)]
\end{aligned}$$

Nyní lze vidět souvislost mezi výrazy $\mathbf{h}^T \mathbf{R} \mathbf{h}$ a $\mathbf{R} \mathbf{a}$. Je zřejmé, že použitím vzorce (3.11) rapidně klesá výpočetní náročnost úzkopásmové energie. Impulsní odezvu FIR filtru \mathbf{h} lze tedy vypočítat ze soustavy rovnic o třinácti neznámých.

$$h(0)^2 + h(1)^2 + h(2)^2 + h(3)^2 + h(4)^2 + h(5)^2 + h(6)^2 + h(7)^2 + h(8)^2 + h(9)^2 + h(10)^2 + h(11)^2 + h(12)^2 = 7869$$

$$\begin{aligned}
& h(0)h(1) + h(1)h(0) + h(1)h(2) + h(2)h(1) + h(2)h(3) + h(3)h(2) + \\
& h(3)h(4) + h(4)h(3) + h(4)h(5) + h(5)h(4) + h(5)h(6) + h(6)h(5) + \\
& h(6)h(7) + h(7)h(6) + h(7)h(8) + h(8)h(7) + h(8)h(9) + h(9)h(8) + \\
& h(9)h(10) + h(10)h(9) + h(10)h(11) + h(11)h(10) + h(11)h(12) + \\
& h(12)h(11) = 7011
\end{aligned}$$

$$\begin{aligned}
& h(0)h(2) + h(1)h(3) + h(2)h(0) + h(2)h(4) + h(3)h(1) + h(3)h(5) + \\
& h(4)h(2) + h(4)h(6) + h(5)h(3) + h(5)h(7) + h(6)h(4) + h(6)h(8) + \\
& h(7)h(5) + h(7)h(9) + h(8)h(6) + h(8)h(10) + h(9)h(7) + h(9)h(11) + \\
& h(10)h(8) + h(10)h(12) + h(11)h(9) + h(12)h(10) = 4838
\end{aligned}$$

$$h(0)h(3) + h(1)h(4) + h(2)h(5) + h(3)h(0) + h(3)h(6) + h(4)h(1) + h(4)h(7) + h(5)h(2) + h(5)h(8) + h(6)h(3) + h(6)h(9) + h(7)h(4) + h(7)h(10) + h(8)h(5) + h(8)h(11) + h(9)h(6) + h(9)h(12) + h(10)h(7) + h(11)h(8) + h(12)h(9) = 2299$$

$$h(0)h(4) + h(1)h(5) + h(2)h(6) + h(3)h(7) + h(4)h(0) + h(4)h(8) + h(5)h(1) + h(5)h(9) + h(6)h(2) + h(6)h(10) + h(7)h(3) + h(7)h(11) + h(8)h(4) + h(8)h(12) + h(9)h(5) + h(10)h(6) + h(11)h(7) + h(12)h(8) = 321$$

$$h(0)h(5) + h(1)h(6) + h(2)h(7) + h(3)h(8) + h(4)h(9) + h(5)h(0) + h(5)h(10) + h(6)h(1) + h(6)h(11) + h(7)h(2) + h(7)h(12) + h(8)h(3) + h(9)h(4) + h(10)h(5) + h(11)h(6) + h(12)h(7) = -660$$

$$h(0)h(6) + h(1)h(7) + h(2)h(8) + h(3)h(9) + h(4)h(10) + h(5)h(11) + h(6)h(0) + h(6)h(12) + h(7)h(1) + h(8)h(2) + h(9)h(3) + h(10)h(4) + h(11)h(5) + h(12)h(6) = -782$$

$$h(0)h(7) + h(1)h(8) + h(2)h(9) + h(3)h(10) + h(4)h(11) + h(5)h(12) + h(7)h(0) + h(8)h(1) + h(9)h(2) + h(10)h(3) + h(11)h(4) + h(12)h(5) = -484$$

$$h(0)h(8) + h(1)h(9) + h(2)h(10) + h(3)h(11) + h(4)h(12) + h(8)h(0) + h(9)h(1) + h(10)h(2) + h(11)h(3) + h(12)h(4) = -164$$

$$h(0)h(9) + h(1)h(10) + h(2)h(11) + h(3)h(12) + h(9)h(0) + h(10)h(1) + h(11)h(2) + h(12)h(3) = 3$$

$$h(0)h(10) + h(1)h(11) + h(2)h(12) + h(10)h(0) + h(11)h(1) + h(12)h(2) = 39$$

$$h(0)h(11) + h(1)h(12) + h(11)h(0) + h(12)h(1) = 21$$

$$h(0)h(12) + h(12)h(0) = 4$$

Vyřešením soustavy rovnic dostáváme impulsní odezvu FIR filtru \mathbf{h} použitého při výpočtu úzkopásmové energie E_l . Pro ověření správnosti mých úvah a odvozené soustavy rovnic odkazují na skript `overeniLowBandEnergy.m`, kde si lze porovnat výsledky vzorců (1.14), (3.11) a zobrazit si kmitočtovou charakteristiku FIR filtru. Z této charakteristiky jsme určili mezní kmitočet FIR filtru $f_m = 1021$ Hz. Pro výpočet soustavy rovnic byla využita funkce Matlabu `fsolve` [4].

3.7 Výpočet dlouhodobé minimální energie

Výpočet dlouhodobé minimální energie je realizován funkcí `InitializeAndUpdatesMins`, kde vstupními parametry je číslo rámce a širokopásmová energie. Výsledná minimální dlouhodobá energie je uložena v proměnné `Min`.

3.8 Výpočet rozdílových parametrů

Rozdílovým parametrům je věnována kapitola 1.5 a jejich výpočet je popsán ve vzorcích (1.16), (1.17), (1.18), (1.19). Aby bylo možné vypočítat rozdílové parametry, je nutné napřed vypočítat průběžné parametry šumu na pozadí. Těmito parametry jsou $\{\overline{LSF}_i\}_{i=1}^p$, \overline{E}_f , \overline{E}_l , \overline{Zc} a jsou vypočteny pouze tehdy, když je širokopásmová energie větší než 15 dB (na DSP konstanta 3072), tzn. při řečovém rámci.

Nyní si ukážeme výpočet průběžného parametru šumu na pozadí \overline{E}_f . Vždy když je $E_f > 15$ dB přičteme $\frac{1}{32}$ z celkové hodnoty E_f . Pokud by byla ve všech 32 rámcích $E_f > 15$ dB, tak bychom opravdu dostali průměrnou hodnotu. Problém nastává, když v některém rámci je $E_f < 15$ dB. Např. kdyby pouze v jednom rámci byla $E_f < 15$ dB, tak bychom museli přičítat už od začátku $\frac{1}{31}$ z celkové hodnoty E_f . My ale dopředu nevíme hodnotu E_f a tím pádem je pro nás neznámá hodnota kterou máme přičítat. Z tohoto důvodu je zaveden čítač `less_count`. Je inkrementován pokaždé, když je v prvních 32 rámcích $E_f < 15$ dB. Abychom tedy dostali správný výsledek, tak po uplynutí 32 rámců vynásobíme \overline{E}_f hodnotou $\frac{32}{32-\text{less_count}}$. V poli `factor_fx` jsou uloženy hodnoty $\frac{32}{32}, \frac{32}{31}, \dots, \frac{32}{2}, \frac{32}{1}$. Tyto hodnoty jsou ve formátu Q15 takže můžeme vyjádřit pouze číslo v rozsahu $\langle 1 - 2^{-15}; -1 \rangle$, proto je nutné vytvořit ještě další pole `shift_fx`, které nám udává, o kolik bitů doleva musíme posunout číslo uložené v poli `factor_fx` abychom dostali správnou hodnotu.

Zbývající průběžné parametry šumu na pozadí jsou vypočteny stejným způsobem a nyní můžeme vypočítat rozdílové parametry podle vzorců (1.16), (1.17), (1.18), (1.19).

Pro výpočet spektrálního zkreslení jsem vytvořil funkci `spectralDistortion`. Vstupními parametry jsou kmitočet spektrálních párů a průměrná hodnota kmitočtu spektrálních párů ve formátu Q15. Výstupním parametrem je spektrální zkreslení taktéž ve formátu Q15. Pro další rozdílové parametry jsem funkce nevytvářel a jsou vypočteny v hlavní funkci detektoru řečové aktivity

podle ITU-T G.729. Pro lepší orientaci v kódu, už jen zmíním, že ΔE_f i ΔE_l je ve formátu Q4.11 a ΔZc ve formátu Q15.

3.9 Vícehraniční VAD rozhodnutí

Soustava podmínek zda se jedná o řečový rámec či nikoliv je uvedena v kapitole 1.6 a je implementována ve funkci MultiboundaryDecision.

Vstupní parametry:

- Rozdílová širokopásmová energie ve formátu Q4.11
- Rozdílová úzkopásmová energie ve formátu Q4.11
- Rozdílový střední počet průchodu signálu nulou ve formátu Q15
- Spektrální zkreslení ve formátu Q15

V této části jsem narazil na problém, že konstanty uváděné v [1] se liší od konstant v originálním kódu. Na první podmínce vícehraničního VAD rozhodnutí ukážu, proč jsem si to dovolil tvrdit. První podmínka je následující.

if $\Delta S > a_1 \cdot \Delta Zc + b_1$ then $I_{VD} = 1$

A v originálním kódu je implementována jako.

```
acc0 = L_mult(dSZC, -14680);          /* Q15*Q23*2 = Q39 */
acc0 = L_mac(acc0, 8192, -28521);    /* Q15*Q23*2 = Q39 */
acc0 = L_shr(acc0, 8);               /* Q39 -> Q31 */
acc0 = L_add(acc0, L_deposit_h(SD));
if (acc0 > 0){
    return(VOICE);
}
```

To lze přepsat a upravit na podmínku.

if $\Delta Zc \cdot a_1 + b_1 + \Delta S > 0$ then $I_{VD} = 1 \Rightarrow$ if $\Delta S > -a_1 \cdot \Delta Zc - b_1$ then $I_{VD} = 1$

Takže potom výpočet konstant a_1 , b_1 je následující.

$$a_1 = -\left(\frac{-14680}{2^{23}}\right) = 0,00174999237$$
$$b_1 = -\left(\frac{8192}{2^{15}} \cdot \frac{-28521}{2^{23}}\right) = 8,499920368 \cdot 10^{-4}$$

Poslední nesrovnalostí v této kapitole, kterou jsem našel je, že u podmínek kde vystupují konstanty a_8 , a_9 není v originálním kódu ΔE_l , ale ΔE_f .

Tabulka 3.1: Konstanty vícehraničního VAD rozhodnutí podle originálního kódu.

| Název | Hodnota | Název | Hodnota |
|----------|----------------|----------|-----------------------------|
| a_1 | 0,00174999237 | b_1 | $8,499920368 \cdot 10^{-4}$ |
| a_2 | -0,00454545021 | b_2 | 0,00115907192 |
| a_3 | -2,5 | b_3 | -0,5 |
| a_4 | 2,0 | b_4 | -0,59997558593 |
| a_5 | 0 | b_5 | -0,47000122070 |
| a_6 | 880 | b_6 | -1,21997070312 |
| a_7 | 0 | b_7 | $9,000003337 \cdot 10^{-4}$ |
| a_8 | 2,5 | b_8 | -0,70001220703 |
| a_9 | -2,90905761718 | b_9 | -0,48181152343 |
| a_{10} | 0 | b_{10} | -0,52999877929 |
| a_{11} | 1400 | b_{11} | -1,54998779296 |
| a_{12} | 0,92855834960 | b_{12} | 0,11428451538 |
| a_{13} | 0,71429443359 | b_{13} | $-2,09264457 \cdot 10^{-4}$ |
| a_{14} | -1,5 | b_{14} | -0,89999389648 |

3.10 Vyhazení VAD rozhodnutí

Při implementaci v signálovém procesoru jsme pouze přepsali podmínky uvedené v kapitole 1.7. Podobně jako u vícehraničního VAD rozhodnutí je zde složitější zjistit správné hodnoty konstant uvedené v [1]. Z toho důvodu zde uvedu hodnoty, které jsem zjistil z originálního kódu.

Tabulka 3.2: Konstanty pro vyhlazení VAD rozhodnutí podle originálního kódu.

| Název | Hodnota | Název | Hodnota |
|-------|--------------|-------|---------|
| T_3 | 0,2001953125 | N_1 | 4 |
| T_4 | 0,2998046875 | N_2 | 10 |
| T_5 | 0,2998046875 | N_0 | 128 |
| T_6 | 0,2998046875 | | - |

3.11 Aktualizace parametrů šumu na pozadí

Po detailnějším studiu originálního kódu musím doplnit podmínku, kterou jsme si uvedli v kapitole 1.8. Podmínka v originálním kódu byla rozšířena a je následující.

if ($E_f < \bar{E}_f + T_6$) and ($rc[2] < T_7$) and ($\Delta S < T_8$) then update

Tabulka 3.3: Konstanty pro aktualizaci šumu na pozadí podle originálního kódu.

| Název | Hodnota | Název | Hodnota |
|-------|---------|-------|----------------|
| T_7 | 0,75 | T_8 | 0,002532958984 |

4 Implementace VAD podle ITU-T G.723.1v signálovém procesoru

V následující kapitole bude popsána implementace v signálovém procesoru TMS320C6416 a případně uvedeny rozdíly mezi doporučením ITU-T a samotným kódem, který je k dispozici.

4.1 Předzpracování signálu

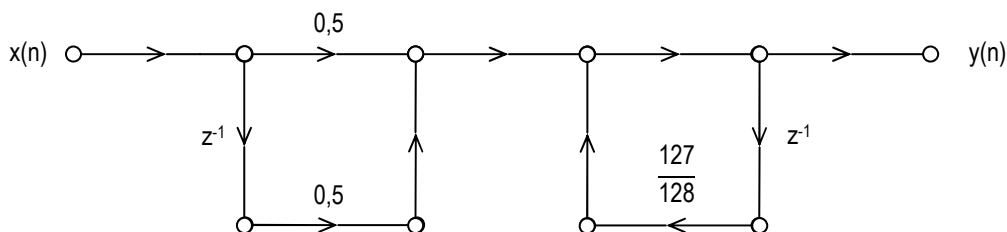
Vstupní signál je filtrován IIR filtrem typu horní propust s přenosovou funkcí uvedenou v (2.1). Z přenosové funkce si vyjádříme diferenční rovnici.

$$y(n) = x(n) - x(n - 1) + \frac{127}{128} \cdot y(n - 1) \quad (4.1)$$

Vstupní signál je podělen dvěma a tak nám vznikne nová diferenční rovnice v následujícím tvaru.

$$y(n) = \frac{1}{2} \cdot x(n) - \frac{1}{2} \cdot x(n - 1) + \frac{127}{128} \cdot y(n - 1) \quad (4.2)$$

Rovnici (4.2) implementujeme ve funkci Preemfaze. Vstupním a zároveň výstupním parametrem funkce je ukazatel na pole. Po dokončení filtrace je pole naplněné filtrovaným signálem ve formátu Q15. Jak lze vidět z diferenční rovnice, číslicový filtr je implementován v přímé formě. Jedná se o velmi jednoduchý filtr a tak si nemusíme dávat práci s implementací v některé, z kanonických forem, kde bychom ušetřili jeden zpožďovací článek.



Obrázek 4.1: Přímá forma pro implementaci preemfáze VAD G.723.1.

4.2 LPC analýza

Pro výpočet LPC koeficientů byla vytvořena funkce `comp_LPC`. Parametry, s kterými je funkce volána, jsou následující: ukazatel na pole, do kterého se uloží LPC koeficienty, ukazatel na minulá a aktuální data, ze kterých je sestaven LPC rámec a poté jsou vypočteny autokorelační koeficienty, LPC koeficienty a nakonec je upraven detektor sinusového signálu.

Při výpočtu autokorelačních koeficientů se v prvním kroku zjistí nultý autokorelační koeficient, který odpovídá energii signálu. Výsledná hodnota je normována a uchová se údaj, o kolik bitů bylo potřeba autokorelační koeficient posunout. V druhém kroku jsou vypočteny zbylé autokorelační koeficienty a jsou posunuty o stejný počet bitů jako nultý autokorelační koeficient. A v poslední řadě jsou autokorelační koeficienty vynásobeny korekčním faktorem a binomiálním oknem.

Pro lepší orientaci v kódu, byla pro samotný výpočet LPC koeficientů pomocí Levinson-Durbinova algoritmu vytvořena funkce `Durbin`. Vstupními parametry jsou autokorelační koeficienty a nultý autokorelační koeficient, který v první iteraci vystupuje jako chyba predikce. Výstupními parametry jsou LPC koeficienty ve formátu Q2.13 a druhý koeficient odrazu ve formátu Q15.

Poslední částí kódu je detektor sinusového signálu. Ten je detekován tehdy, když jsou koeficienty odrazu pro posledních 14 rámců větších jak 0,95. Detekce je dosaženo tím, že napřed proměnou `SinDet` posuneme o jeden bit doleva a potom, pokud je koeficient odrazu vyšší než 0,95 zvýšíme hodnotu `SinDet` o jedničku. Splnění podmínky, že je posledních 14 koeficientů odrazu větších než 0,95 se zjistí tak, že se v cyklu proměnnou `SinDet` posouváme o jeden bit doprava a zjišťujeme, zda je hodnota lichá nebo sudá. Pokud je hodnota lichá, tak koeficient odrazu byl větší než 0,95 a zaznamenáme si to. Po dokončení cyklu zjistíme, kolikrát byl koeficient odrazu větší než 0,95 a pokud byl větší čtrnáctkrát, tak provedeme logický součet proměnné `SinDet` s hodnotou 32768. Tímto jsme zajistili, že `SinDet` nabývá záporných hodnot pouze tehdy když, je v posledních 14-ti rámcích koeficient odrazu větší než 0,95.

4.3 Váhovací filtr pro vnímání formantů

Pro výpočet váhovacího filtru byla vytvořena funkce `Comp_Wght_Lpc`. Vstupním parametrem jsou LPC koeficienty a výstupním parametrem váhované LPC koeficienty. Vstupní i výstupní data jsou ve formátu Q2.13. Ve funkci jsou vypočteny váhované koeficienty podle vzorce (2.6). Protože umocňování je na signálovém procesoru pomalá operace jsou hodnoty $0,9^j$ a $0,5^j$ již vypočteny a uloženy v poli s názvem `PerFiltZeroTable` a `PerFiltPoleTable`.

Pro filtraci signálu váhovacím filtrem slouží funkce `Filter_Wght`. Vstupním parametrem jsou váhované koeficienty a pole `Dpnt`, ve kterém jsou vstupní vzorky a po filtraci vzorky výstupní. Filtr je realizován v přímé formě a zpožďovací články zde jsou buffery `WghtFirD1` a `WghtIirD1`. IIR filtr je desátého řádu, takže oba buffery jsou taktéž délky deset.

4.4 Odhad základní periody řeči

Odhad základní periody řeči se provádí ve funkci `Estim_Pitch`. Vstupními parametry jsou ukazatel na pole a startovní pozice v poli, ze které se odhadne základní perioda řeči. Protože podle vzorce (2.7) je patrné, že při odhadu základní periody řeči používáme i minulé vzorky, musíme si sestavit buffer, který obsahuje 142 vzorků minulých a 240 vzorků aktuálních. Potom z takto sestaveného bufferu je možné odhadnout základní periodu řeči.

Aplikace vzorce (2.7) je ale trochu odlišná, protože operace dělení je na signálovém procesoru náročná. Vypočteme si zvlášť čitatel a jmenovatel funkce a podle toho upravíme podmínky pro získání indexu základní periody řeči. První podmínka je následující.

```
Acc0 = _smpy( Ccr, Mnr );
Acc0 = _ssub( Acc0, _smpy(Enr, Tmp ) );
if ( Acc0 > (int) 0 )
```

Když si rozepíšeme výpočet `Acc0`, tak dostáváme.

$$Acc0 = Ccr \cdot Mnr - Enr \cdot Mcr \Rightarrow \frac{Ccr}{Enr} > \frac{Mcr}{Mnr}$$

A z výše uvedeného vztahu je patrné, že jsme se vyhnuli operaci dělení pouze upravením podmínky. Druhá podmínka pro získání indexu základní periody řeči je sestavena na stejné myšlence.

```
Acc0 = _smpy( Ccr, Mnr );
Acc0 = -( Acc0 >> (short) 2 );
Acc0 = _sadd( Acc0, _smpy(Ccr, Mnr ) );
Acc0 = _ssub( Acc0, _smpy(Enr, Tmp ) );
if ( Acc0 > (int) 0 )
```

A po vyjádření `Acc0` dostáváme.

$$\frac{Ccr}{Enr} > \frac{Mcr}{Mnr} + \frac{1}{4} \cdot Ccr \cdot Enr$$

4.5 Výpočet příznaku povolení adaptace

Výpočet příznaku povolení adaptace se provádí ve funkci `Comp_Vad` jako jedna z částí výpočtu VAD rozhodnutí. Při implementaci jsem nenarazil na žádný problém a výpočet probíhá přesně tak, jak je popsáno v doporučení a také v kapitole 2.6. Příznak povolení adaptace v signálovém procesoru reprezentuje proměnná `Aen`.

4.6 Inverzní filtrace a výpočet filtrované energie

Blok, který provádí inverzní filtraci a výpočet filtrované energie, lze nalézt ve funkci `Comp_Vad`. Je aplikován vzorec (2.9) a rovnou každý vzorek je umocněn dvěma a přičten do proměnné `Acc0`. Tím jsme snížili paměťové nároky na signálový procesor. Jinak bychom museli uchovávat všech 240 vzorků vyfiltrovaného signálu a potom z nich vypočítat filtrovanou energii. V posledním kroku je filtrovaná energie násobená konstantou 0,088897705. Tuto konstantu jsem zjistil z originálního kódu, protože se liší od konstanty $\frac{1}{80}$ uvedené ve vzorci (2.13), který jsem zjistil přímo z doporučení [2].

Pro inverzní filtraci je nutné znát zprůměrované LPC koeficienty, které jsou vypočteny ve funkci `Comp_Ano` resp. ve funkci `Comp_PastAvFilter`. Ukazatel `ptr_Acf` ukazuje na pole autokorelačních koeficientů o délce 44. Tzn. že v tomto poli jsou uloženy autokorelační koeficienty z posledních čtyřech rámců a vyjadřují kumulované autokorelační koeficienty. Zprůměrované autokorelační koeficienty jsou uchovány v poli `L_sumAcf` o délce jedenáct. Dále jsou vypočteny zprůměrované LPC koeficienty pomocí Levinson-Durbinova algoritmu a jsou aktualizovány pouze tehdy, když to příznak adaptace umožňuje. Zprůměrované LPC koeficienty jsou uloženy v proměnné `NLpc` a jsou ve formátu Q2.13.

4.7 Výpočet úrovně šumu

Výpočet úrovně šumu se provádí jako jedna z částí funkce `Comp_Vad`. V prvním kroku je aplikován vzorec (2.14) a pro násobení a dělení se využívá bitových posunů, protože se jedná o násobení a dělení mocninou dvou. Takto provedené násobení je nejrychlejší možné. Ve druhém kroku je použit vzorec (2.15) a když se podíváme na použité konstanty, tak můžeme zase použít bitových posunů, jelikož se jedná o násobení mocninou dvou. Násobení konstantou 1,03125 lze provést jako

```
Nlev = _sadd( Nlev, Nlev >> 5 );
```

To lze vyjádřit jako.

$$Nlev = Nlev + \frac{Nlev}{2^5} = Nlev + 0,03125 \cdot Nlev = 1,03125 \cdot Nlev$$

Pro násobení konstantou 0,9995 je použito také bitového posunu a k výsledku, bychom se dopracovali podobně, jak je uvedeno výše. V posledním kroku je úroveň šumu oříznuta do rozsahu 128 až 131071. Úroveň šumu je uchovávána v proměnné *Nlev* ve formátu Q4.27.

4.8 Výpočet prahu

Stejně jako výpočet úrovně šumu, tak i výpočet prahu nalezneme ve funkci *Comp_Vad*. Při implementaci jsem narazil na problém, že vzorec (2.16) uvedený v [2] se odlišuje od originálního kódu. Proto zde uvedu vzorec odpovídající zdrojovému kódu pro výpočet prahu.

$$Thr = \begin{cases} Nlev \cdot 5,012 & \text{if } Nlev = 128 \\ Nlev \cdot 10^{0,7-0,005 \log_2 \frac{Nlev}{128}} & \text{if } 128 < Nlev < 16384 \\ Nlev \cdot 2,239 & \text{if } Nlev \geq 16384 \end{cases} \quad (4.3)$$

Protože výpočet logaritmu a mocniny deseti jsou na signálovém procesoru výpočetně náročné operace jsou výsledky vzorce $10^{0,7-0,005 \log_2 \frac{Nlev}{128}}$ vypočteny pro hodnoty 128, 256, 512, ..., 131072 a uloženy do pole *ScfTab* délky jedenáct. Nyní si ukážeme, jak jsme číselné hodnoty v poli *ScfTab* získali. Pro hodnotu 128 je výpočet následující.

$$Thr = 10^{0,7-0,005 \log_2 \frac{128}{128}} = 5,012$$
$$ScfTab[10] = 5,012 \cdot 4096 = 20529$$

Pro ostatní hodnoty bychom postupovali obdobně. Z výše uvedeného výpočtu vyplývá, že hodnoty v poli *ScfTab* jsou ve formátu Q3.12. Poslední věcí, kterou by bylo vhodné zmínit je, že hodnoty v poli jsou v obráceném pořadí. Tzn. že, výsledek pro hodnotu 128 se nachází v *ScfTab[10]* a pro hodnotu 131072 v *ScfTab[0]*. Pro získání správné hodnoty, nacházející se mezi hodnotami uloženými v poli *ScfTab*, je prováděna lineární interpolace.

4.9 Rozhodnutí VAD

VAD rozhodnutí se nachází ve funkci `Comp_Vad` a tuto hodnotu funkce i vrací. Pokud je filtrovaná energie větší než práh, tak je rámeček označen jako řečový a funkce vrací 1. V opačném případě je rámeček označen jako neřečový a funkce vrací hodnotu 0.

4.10 VAD vyhlazení

V posledním kroku funkce `Comp_Vad` se provede VAD vyhlazení. K tomu je zapotřebí dvou čítačů `Vcnt` a `Hcnt`. Jak bylo uvedeno v kapitole 2.12, pokud byly poslední dva rámečky označeny jako řečové, tak bude i následujících šest rámečků označených jako řečových.

5 Závěr

Cílem této práce bylo se seznámit a metodami detekce řečové aktivity. Byly vybrány detektory řečové aktivity podle ITU-T G.729 a G.723.1. Oba vybrané detektory byly implementovány v signálovém procesoru TMS320C6416 a v programovém prostředí MATLAB. Při implementaci byly výsledky jednotlivých funkcí porovnávány s výsledky kódů podle standardů ITU-T.

U detektoru řečové aktivity podle ITU-T G.729 byly výsledky LPC analýzy v signálovém procesoru zatíženy chybou. Výsledky zbylých funkcí se téměř nelišily. Domnívám se, že chyba je způsobena tím, že při výpočtu dochází k chybám při zaokrouhlování, které se dále akumulují. K akumulaci dochází proto, že k výpočtu LPC koeficientů se používá Levinson-Durbinův algoritmus který je iterační. Vyzoroval jsem, že tato chyba se nejvíce uplatňuje a narůstá při operaci dělení.

Při implementaci jsem byl nemile překvapen, že standardy se mohou tolik lišit oproti přiloženým kódům v jazyce C. Např. rozdílné konstanty u detektoru řečové aktivity podle ITU-T G.729 či neúplný vzorec pro určení prahu u detektoru řečové aktivity podle ITU-T G.723.1. Všechny rozdíly, které jsem objevil, byly diskutovány v příslušných kapitolách zabývajících se implementací.

Na výsledcích uvedených v přílohách je patrné, že oba detektory řečové aktivity jsou konstruovány tak, aby raději rámeček označily jako řečový, i když se jedná o neřečový. Toto je typické nastavení u detektorů řečové aktivity používaných v kodecích. Tato vlastnost je nejvíce patrná na výsledcích zarušené nahrávky *sny_sum.wav*, kde byl přidán k nahrávce *sny.wav* bílý šum.

Literatura

- [1] ITU-T Rec. G.729, Series G: *Coding of speech at 8 kbit/s using conjugate structure algebraic code excited linear prediction*, 1996.
- [2] ITU-T Rec. G.723.1, Series G: *Dual rate speech coder for multimedia communications transmitting at 5,3 and 6,3 kbit/s*, 1996.
- [3] Cisco Systems. *Understanding Codecs: Complexity, Hardware Support, MOS, and Negotiation* [online]. Poslední revize 6.2.2006 [citováno 24. Listopadu 2011].
<http://www.cisco.com/en/US/tech/tk1077/technologies_tech_note09186a00800b6710.shtml>.
- [4] MathWorks. *Solve system of nonlinear equations – MATLAB* [online].
<<http://www.mathworks.com/help/techdoc/>> [citováno 8.2.2012].
- [5] KABAL, P. a R.P. RAMACHANDRAN. The computation of line spectral frequencies using Chebyshev polynomials. *IEEE Transactions on Acoustics, Speech, and Signal Processing*. 1986, roč. 34, č. 6, s. 1419-1426. ISSN 0096-3518. DOI: 10.1109/TASSP.1986.1164983.
- [6] PSUTKA, Josef. *Mluvíme s počítačem česky*. Vyd. 1. Praha: Academia, 2006, 746 s. ISBN 80-200-1309-1.
- [7] *TMS320C6000 Programmer's Guide* [online]. Texas Instruments Inc., 2011. Poslední aktualizace 11.7.2011.
<<http://www.ti.com/lit/ug/spru198k/spru198k.pdf>>. ID: SPRU198K.

Seznam symbolů, veličin a zkratk

| | |
|----------------|---|
| a_i | i -tý LPC koeficient |
| Aen_t | příznak povolení adaptace |
| $a_{no}(j)$ | koeficienty filtru $A_{no}(z)$ |
| $\bar{a}_p(j)$ | koeficienty zprůměrovaného LPC filtru |
| $C_{OL}(j)$ | autokorelační funkce |
| ΔE_f | rozdíl širokopásmové energie |
| ΔE_l | rozdíl úzkopásmové energie |
| ΔS | spektrální zkreslení |
| ΔZc | rozdíl středního počtu průchodu nulou |
| E_f | širokopásmová energie |
| \bar{E}_f | průběžný průměr širokopásmové energie |
| E_l | úzkopásmová energie |
| \bar{E}_l | průběžný průměr úzkopásmové energie |
| \bar{E}_n | průměrná energie E_f v prvních N_i rámcích |
| Enr_t | energie inverzně filtrovaného signálu |
| $e_t(n)$ | signál po průchodu inverzním filtrem |
| γ_1 | konstanta |
| γ_2 | konstanta |
| h | vektor impulzní odezvy |
| I_{VD} | rozhodnutí detektoru podle ITU-T G.729 |
| $k_i^t[2]$ | druhý koeficient odrazu |
| L_{OL} | základní perioda řeči |
| LPC | lineární prediktivní analýza – linear predictive coding |
| N | velikost okna |
| $Nlev_t$ | úroveň šumu |
| pc | čítač hodnot L_{OL} |
| $Qm.n$ | číselný formát, kde m značí počet bitů celočíselné části a n počet bitů desetinné části |
| $r(k)$ | autokorelační koeficienty |
| $r'(k)$ | modifikované autokorelační koeficienty (G.729) |
| $R(k)$ | modifikované autokorelační koeficienty (G.723.1) |
| R | autokorelační matice v Toeplitzově tvaru |
| $\bar{R}_p(j)$ | zprůměrované autokorelační koeficienty |
| $R^t(j)$ | kumulovaná autokorelační funkce |
| $s(n)$ | vstupní signál |
| $s'(n)$ | vážený signál |

| | |
|--------------------------------------|--|
| $SinD$ | detektor sinusového signálu |
| $S_{VD}^0, S_{VD}^{-1}, S_{VD}^{-2}$ | VAD vyhlazení v aktuálním, minulém a předminulém rámci |
| Thr | práh |
| VAD | detektor řečové aktivity – voice activity detector |
| Vad_t | rozhodnutí detektoru podle ITU-T G.723.1 |
| $w(n)$ | váhovací Hammingovo okno |
| $W_i(z)$ | přenosová funkce váhovacího filtru |
| $w_{lag}(k)$ | váhovací konstanta pro autokorelační koeficienty |
| $w_{lp}(n)$ | váhovací funkce |
| Zc | střední počet průchodu nulou |

Seznam příloh

| | | |
|----------|---|-----------|
| A | TEXTY TESTOVACÍCH NAHRÁVEK | 56 |
| A.1 | Testovací nahrávka sny.wav | 56 |
| A.2 | Testovací nahrávka sny_sum.wav | 56 |
| A.3 | Testovací nahrávka alkoholik8k.wav | 56 |
| B | VÝSLEDKY VAD PODLE ITU-T G.729 | 57 |
| B.1 | Výsledky pro nahrávku sny.wav | 57 |
| B.2 | Výsledky pro nahrávku sny_sum.wav | 58 |
| B.3 | Výsledky pro nahrávku alkoholik8k.wav | 59 |
| C | VÝSLEDKY VAD PODLE ITU-T G.723.1 | 60 |
| C.1 | Výsledky pro nahrávku sny.wav | 60 |
| C.2 | Výsledky pro nahrávku sny_sum.wav | 61 |
| C.3 | Výsledky pro nahrávku alkoholik8k.wav | 62 |
| D | OBSAH PŘILOŽENÉHO CD | 63 |

A Texty testovacích nahrávek

A.1 Testovací nahrávka sny.wav

Text: Na začátek trochu čísel. Člověk prospí průměrně šest hodin denně, spánek tak zabere téměř třetinu jeho života. Snům se oddává přibližně dvě hodiny spánku každou noc. Ty mu umožňují být druhý den ve fyzické i psychické kondici. Není proto pochyb, že spánek a sny k živočichům neodmyslitelně patří. Jsou biologickou nutností stejně tak jako jídlo a pohyb, pokud není člověku umožněno snít v průběhu noci, dohání deficit ve dne, kdy je prostě pořád mimo. Takového jedince asi zrovna nebudou zajímat novinky z vašeho intimního života a o to míň látka nové přednášky, bude si prostě snít svůj život.

A.2 Testovací nahrávka sny_sum.wav

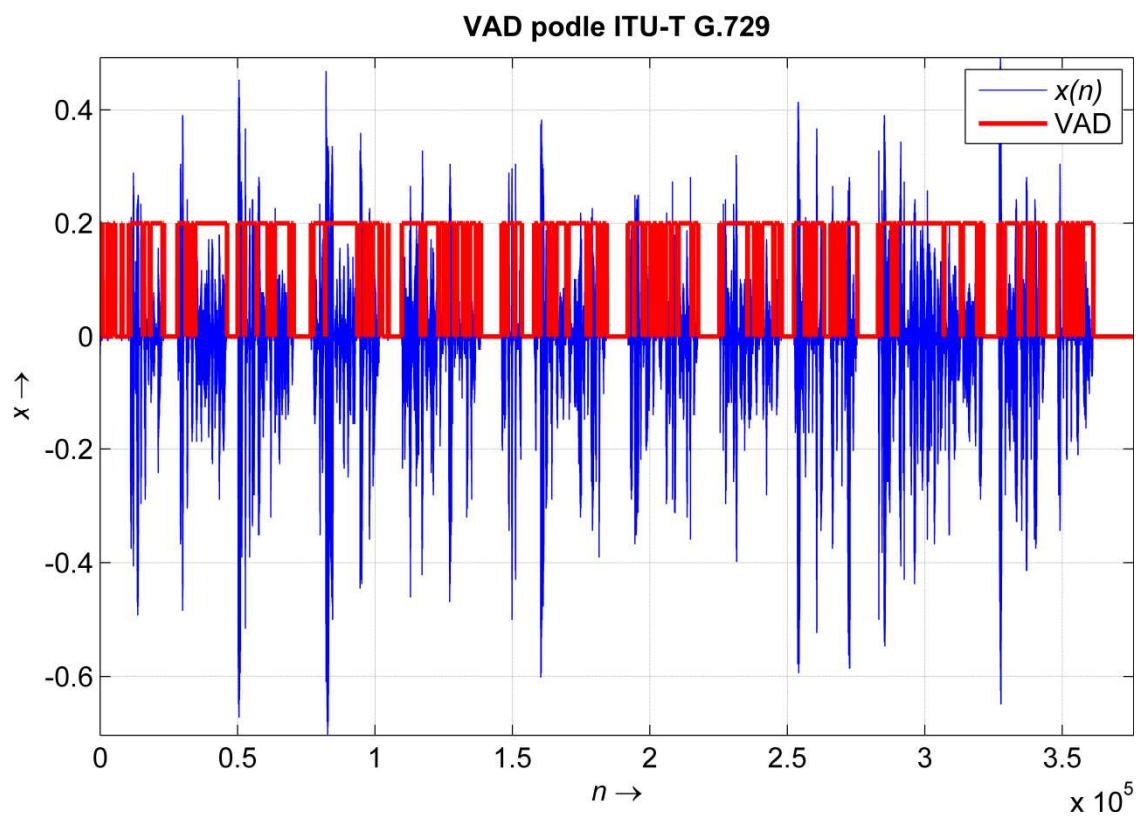
Text: Na začátek trochu čísel. Člověk prospí průměrně šest hodin denně, spánek tak zabere téměř třetinu jeho života. Snům se oddává přibližně dvě hodiny spánku každou noc. Ty mu umožňují být druhý den ve fyzické i psychické kondici. Není proto pochyb, že spánek a sny k živočichům neodmyslitelně patří. Jsou biologickou nutností stejně tak jako jídlo a pohyb, pokud není člověku umožněno snít v průběhu noci, dohání deficit ve dne, kdy je prostě pořád mimo. Takového jedince asi zrovna nebudou zajímat novinky z vašeho intimního života a o to míň látka nové přednášky, bude si prostě snít svůj život.

A.3 Testovací nahrávka alkoholik8k.wav

Text: Alkoholik je člověk, který pije víc než jeho lékař.

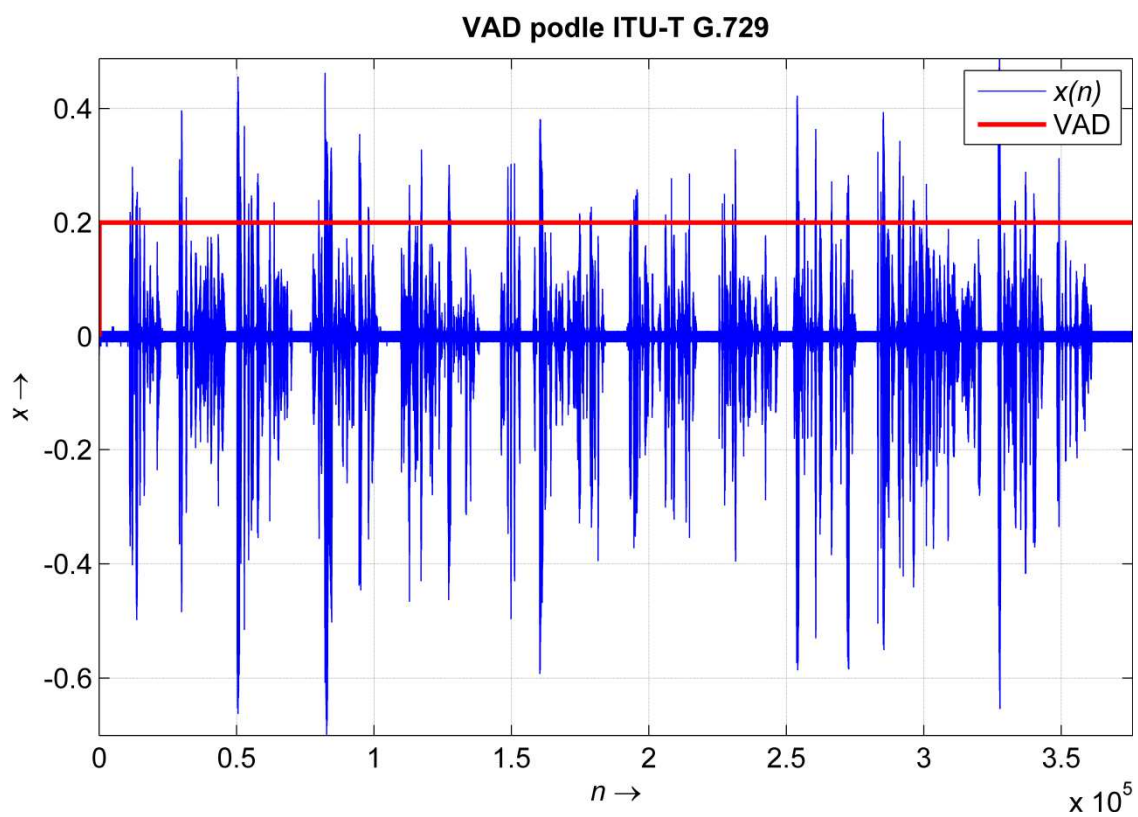
B Výsledky VAD podle ITU-T G.729

B.1 Výsledky pro nahrávku sny.wav



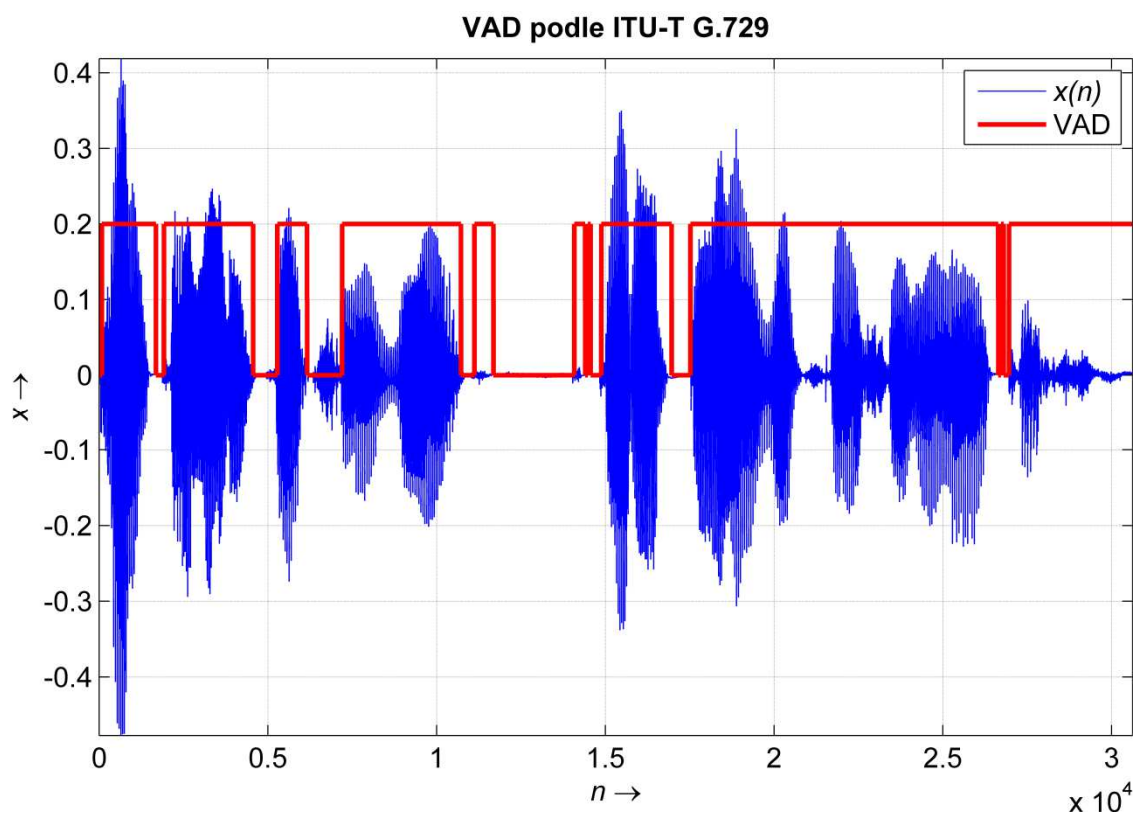
Obrázek B.1: Zobrazení výsledku VAD podle ITU-T G.729 pro nahrávku sny.wav.

B.2 Výsledky pro nahrávku sny_sum.wav



Obrázek B.2: Zobrazení výsledku VAD podle ITU-T G.729 pro nahrávku sny_sum.wav.

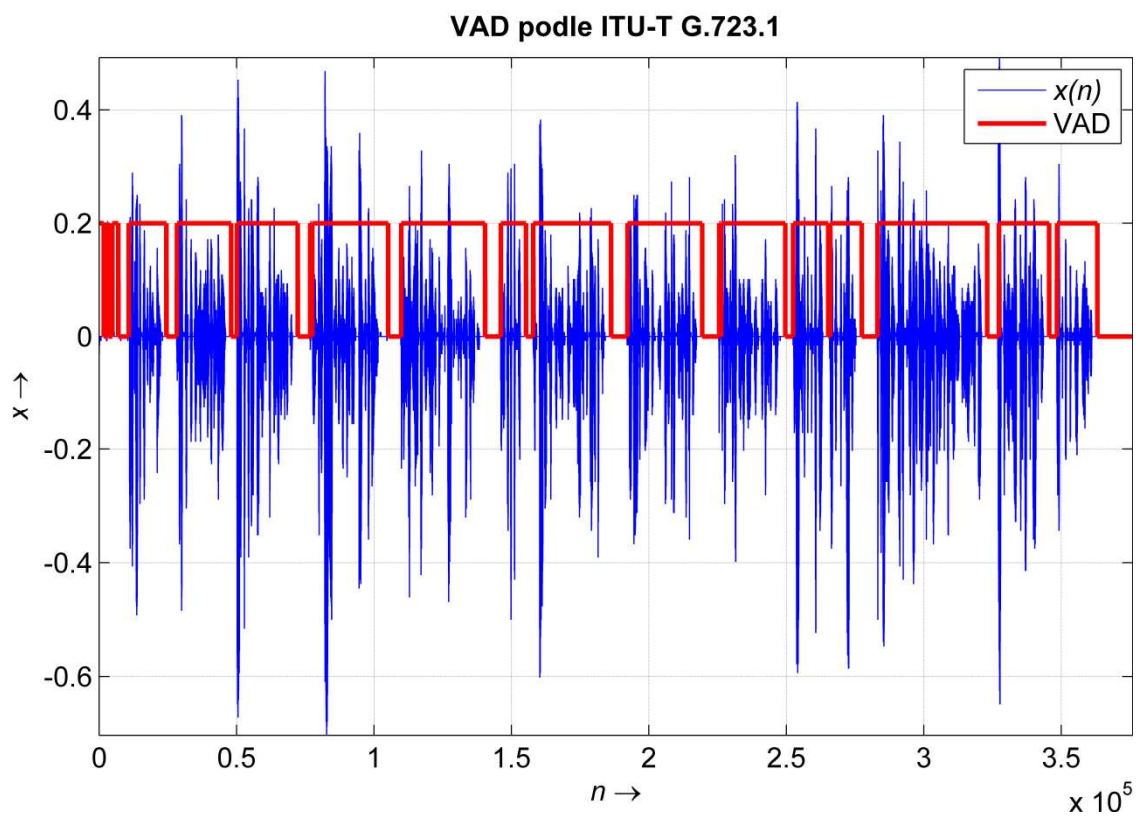
B.3 Výsledky pro nahrávku alkoholik8k.wav



Obrázek B.3: Zobrazení výsledku VAD podle ITU-T G.729 pro nahrávku alkoholik8k.wav.

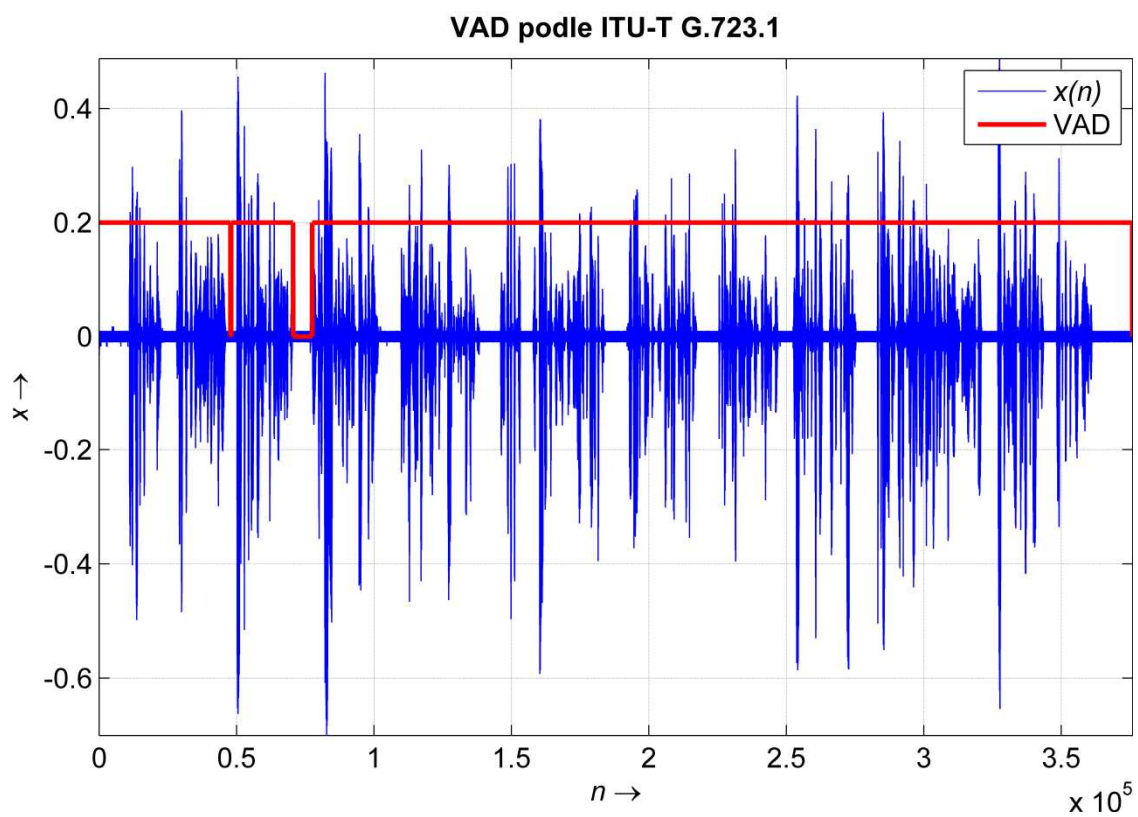
C Výsledky VAD podle ITU-T G.723.1

C.1 Výsledky pro nahrávku sny.wav



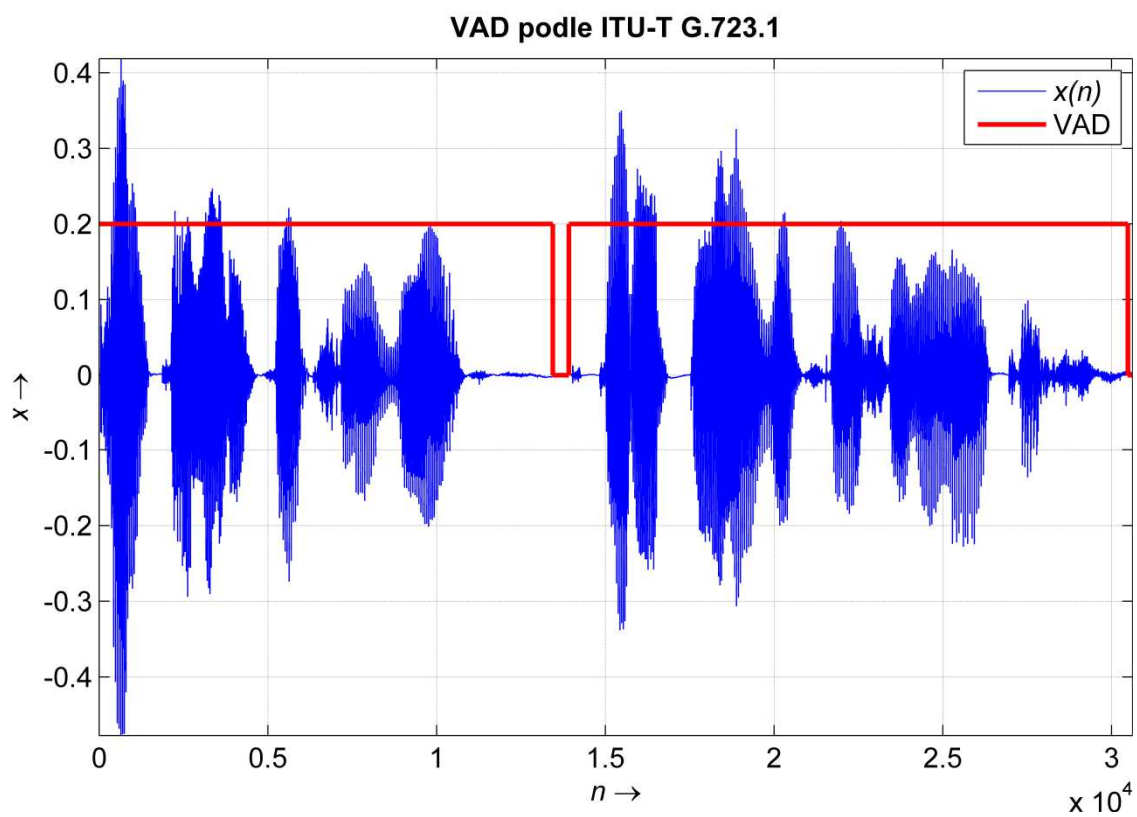
Obrázek C.1: Zobrazení výsledku VAD podle ITU-T G.723.1 pro nahrávku sny.wav.

C.2 Výsledky pro nahrávku sny_sum.wav



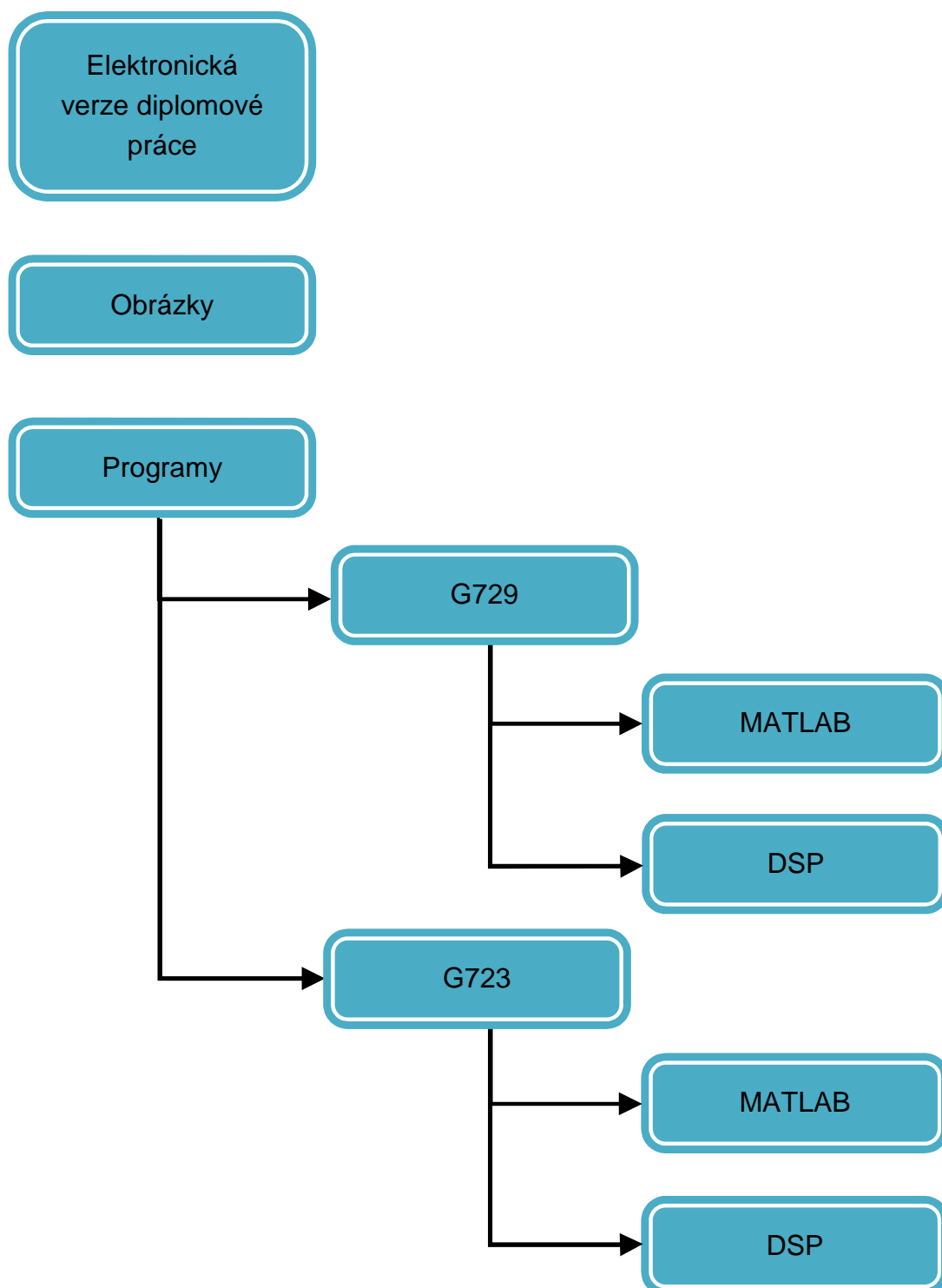
Obrázek C.2: Zobrazení výsledku VAD podle ITU-T G.723.1 pro nahrávku sny_sum.wav.

C.3 Výsledky pro nahrávku alkoholik8k.wav



Obrázek C.3: Zobrazení výsledku VAD podle ITU-T G.723.1 pro nahrávku alkoholik8k.wav.

D Obsah přiloženého CD



Spuštění jednotlivých detektorů řečové aktivity

- VAD podle ITU-T G.729

V programovém prostředí MATLAB otevřete soubor VAD_G729.m a stiskem klávesy F5 spustíte program. Změnu nahrávky lze provádět na řádku

```
[y,Fvz] = wavread('wav\alkoholik8k.wav');
```

změnou alkoholik8k.wav za sny.wav nebo sny_sum.wav.

V signálovém procesoru lze detektor řečové aktivity podle ITU-T G.729 spustit pomocí naimportování projektu do prostředí *Eclipse* a provedením úkonů build a debug. Hlavní program se nachází v souboru detektorG729.c. Pomocí programu *Multisine* se na vstup LINE IN vývojového kitu přivede signál zvolené nahrávky.

- VAD podle ITU-T G.723.1

V programovém prostředí MATLAB otevřete soubor VAD_G723.m a stiskem klávesy F5 spustíte program. Změnu nahrávky lze provádět na řádku

```
[y,Fvz] = wavread('wav\alkoholik8k.wav');
```

změnou alkoholik8k.wav za sny.wav nebo sny_sum.wav.

V signálovém procesoru lze detektor řečové aktivity podle ITU-T G.723.1 spustit pomocí naimportování projektu do prostředí *Eclipse* a provedením úkonů build a debug. Hlavní program se nachází v souboru detektorG723.c. Pomocí programu *Multisine*, se na vstup LINE IN vývojového kitu přivede signál zvolené nahrávky.