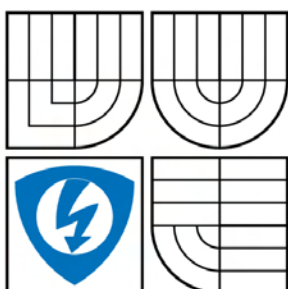


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

POTLAČOVÁNÍ ŠUMU V ŘEČI ZALOŽENÉ NA WAVELETOVÉ TRANSFORMACI A ROZEZNÁVÁNÍ ZNĚLOSTI SEGMENTŮ

SPEECH DENOISING BASED ON WAVELET TRANSFORM AND VOICE RECOGNITION IN
SEGMENTS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

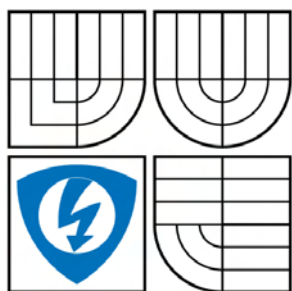
Bc. TOMÁŠ CHRÁPEK

VEDOUCÍ PRÁCE

SUPERVISOR

Mgr. PAVEL RAJMIC, Ph.D.

BRNO 2008



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Chrápek Tomáš Bc.

ID: 89797

Ročník: 2

Akademický rok: 2007/2008

NÁZEV TÉMATU:

Potlačování šumu v řeči založené na waveletové transformaci a rozeznávání znělosti segmentů

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte stávající metody pro rozlišování znělosti / neznělosti řečových úseků, neopomeňte metody využívající hodnot keprstrálních koeficientů. Nastudujte základy waveletové transformace signálů, zejména souvislost transformace v diskretním čase s rekurzivní filtrací signálu. Na základě návrhu metody potlačování šumu pomocí waveletů, která kombinuje různé přístupy pro znělé a neznělé segmenty, naprogramujte v Matlabu takový algoritmus. Vytvořte grafické uživatelské rozhraní pro pohodlné ovládání a testování.

DOPORUČENÁ LITERATURA:

- [1] DELLER, J.R.; HANSEN, H.L.; PROAKIS, J.G.: Discrete-Time Processing of Speech Signals. Macmillan, New York, 1993. ISBN 0-7803-5386-2
- [2] VARGIC, R: Wavelety a banky filtrů. STU, Bratislava, 2004. ISBN 80-227-2093-3
- [3] RAJMÍČ, P.: Využití waveletové transformace a matematické statistiky pro separaci signálu a šumu. PhD Thesis, Brno university of Technology, Brno, 2004.

Termín zadání: 11.2.2008

Termín odevzdání: 28.5.2008

Vedoucí práce: Mgr. Pavel Rajmic, Ph.D.

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 § 152 trestního zákona č. 140/1961 Sb.

LICENČNÍ SMLOUVA

POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Bc. Tomáš Chrápek
Bytem: Selská 171/3, 70030, Ostrava - Hrabůvka
Narozen/a (datum a místo): 15.7.1984, Ostrava

(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 244/53, 60200 Brno 2
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Ing. Kamil Vrba, CSc.

(dále jen "nabyvatel")

Článek 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- disertační práce
- diplomová práce
- bakalářská práce

jiná práce, jejíž druh je specifikován jako

(dále jen VŠKP nebo dílo)

Název VŠKP: Potlačování šumu v řeči založené na waveletové transformaci a rozeznávání znělosti segmentů

Vedoucí/školitel VŠKP: Mgr. Pavel Rajmic, Ph.D.

Ústav: Ústav telekomunikací

Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

- tištěné formě - počet exemplářů 1
- elektronické formě - počet exemplářů 1

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.

3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.

4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užit, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....

Nabyvatel

.....

Autor

Anotace

Waveletová transformace je moderním nástrojem pro zpracování signálů a je hojně a s úspěchem využívána hlavně pro své unikátní vlastnosti, zejména schopnost zachytit rychlé změny ve zpracovávaném signálu.

Práce pojednává obecně o waveletech, waveletové transformaci a jejím využití v aplikacích zabývajících se potlačováním šumu. Byl uveden hlavní problém spojený s potlačováním šumu v řečovém signálu, a to degradace neznělých částí spojená s potlačováním šumu. Neznělé části jsou při aplikaci standardního algoritmu na waveletové potlačení šumu potlačeny spolu s šumem. Řešením je rozdílný přístup ke znělým a neznělým segmentům řečového signálu.

Cílem diplomové práce bylo vytvořit aplikaci, jež bude implementovat waveletové potlačování aditivního gaussovského šumu v řečových signálech s obecně rozdílným přístupem ke znělým a neznělým segmentům. Tato aplikace je naprogramována jako grafické uživatelské rozhraní (GUI) ve vývojovém prostředí MATLAB a umožňuje pohodlně testovat procedury představené v diplomové práci.

Práce dále prezentuje dosažené výsledky a diskutuje jejich úspěšnost vzhledem k požadavkům na aplikaci tohoto typu kladených. Nejpodstatnějším závěrem této diplomové práce je potom fakt, že při potlačování šumu v neznělých segmentech řečového signálu musíme volit kompromis mezi dostatečným potlačením šumu a zachováním srozumitelnosti řeči.

Klíčová slova: wavelet, waveletová transformace, waveletové koeficienty, znělost, neznělost.

Abstract

The wavelet transform is a modern signal processing tool. The wavelet transform earned itself a great success mainly for its unique properties, such as the capability of recognizing very fast changes in processed signal.

The theoretical part of this work is introduction to wavelet theory, more specifically wavelet types, a wavelet transform and its application in systems dealing with signal denoising. A main problem connected to speech signals denoising was introduced. The problem is degradation of the speech signal when denoising unvoiced parts. It is because of the fact that unvoiced parts and noise itself have very similar characteristics. The solution would be to apply different attitude to voiced and unvoiced segments of the speech.

The main goal of this diploma thesis was to create an application implementing the speech signal denoising using the wavelet transform. The special attention should have been paid to applying different attitude to voiced and unvoiced segments of the speech. The demanded application is programmed as a graphical user interface (GUI) in MATLAB environment. The algorithm implemented in this form allows users to test introduced procedures with a great comfort.

This work presents achieved results and discusses them considering general requirements posed on an application of given type. The most important conclusion of this Diploma Thesis is the fact that some kind of trade-off between sufficient signal denoising and keeping the speech understandable has to be made.

Keywords: wavelet, wavelet transform, wavelet coefficients voiced segments, unvoiced segments.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma Potlačování šumu v řeči založené na waveletové transformaci a rozeznávání znělosti segmentů jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s využitím 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 tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne.....

.....
podpis autora

PODĚKOVÁNÍ

Děkuji vedoucímu své diplomové práce, panu Mgr. Pavlu Rajmicovi, Ph.D., za průběžnou podporu a podnětné připomínky při zpracovávání práce na téma „Potlačování šumu v řeči založené na waveletové transformaci a rozeznávání znělosti segmentů“.

V Brně dne.....

.....
podpis autora

Obsah

OBSAH	1
SEZNAM OBRÁZKŮ	2
SEZNAM TABULEK	3
ÚVOD	4
1 WAVELETOVÁ TRANSFORMACE	5
1.1 DEFINICE WAVELETU	5
1.2 INTEGRÁLNÍ WAVELETOVÁ TRANSFORMACE	7
1.3 DISKRÉTNÍ WAVELETOVÁ TRANSFORMACE	8
1.4 MR ANALÝZA	8
1.5 KONEČNÁ DISKRÉTNÍ WAVELETOVÁ TRANSFORMACE	9
2 KLASIFIKACE ZNĚLOSTI A NEZNĚLOSTI FONÉMŮ	11
2.1 STATISTICKÁ METODA KLASIFIKACE	11
2.2 METODA MEL-KEPSTRÁLNÍCH KOEFICIENTŮ	12
2.2.1 Mel-banka filtrů	12
2.2.2 Mel-frekvenční kepstrální koeficienty	14
3 WAVELETOVÉ POTLAČOVÁNÍ ŠUMU	15
3.1 TYPY PRAHOVACÍCH PRAVIDEL	16
4 NÁVRH SYSTÉMU PRO POTLAČOVÁNÍ ŠUMU V ŘEČI A JEHO IMPLEMENTACE	18
4.1 WAVELETOVÉ POTLAČOVÁNÍ ŠUMU V ŘEČOVÉM SIGNÁLU	18
4.2 GUI – GRAFICAL USER INTERFACE	18
4.2.1 Základní panel	21
4.2.2 Přidání šumu	22
4.2.3 UV klasifikace	23
4.2.4 Waveletové potlačení šumu	24
4.2.5 Obecně k průchodu signálu	25
4.3 POUŽITÉ MATLABOVSKÉ FUNKCE	26
5 TESTOVÁNÍ	31
5.1 ANALÝZA ZNĚLOSTI A NEZNĚLOSTI SEGMENTŮ	31
5.2 WAVELETOVÉ POTLAČOVÁNÍ ŠUMU	32
5.3 ZHODNOCENÍ TESTOVÁNÍ	36
ZÁVĚR	38
POUŽITÁ LITERATURA	39
SEZNAM PŘÍLOH	40

Seznam obrázků

OBR. 1.1: MATEŘSKÝ WAVELET TYPU HAAR	6
OBR. 1.2: MATEŘSKÝ WAVELET TYPU DAUBECHIES 2 (A), DAUBECHIES 8 (B), SYMLET 2 (C) A SYMLET 8 (D)	6
OBR. 1.3: JEDEN KROK WAVELETOVÉ DEKOMPOZICE. APROXIMAČNÍ KOEFICIENTY J-TÉ ÚROVNĚ JSOU POMOCÍ FILTRU H, RESP. G A DECIMACE ROZLOŽENY NA APROXIMAČNÍ, RESP. DETAILNÍ KOEFICIENTY ÚROVNĚ $J + 1$.	10
OBR. 1.4: JEDEN KROK WAVELETOVÉ REKONSTRUKCE. APROXIMAČNÍ KOEFICIENTY J-TÉ ÚROVNĚ JSOU VYPOČÍTÁNY Z APROXIMAČNÍCH A DETAILNÍCH KOEFICIENTŮ ÚROVNĚ O JEDNA VYŠŠÍ POMOCÍ INTERPOLACE, APLIKACÍ FILTRACÍ INVERZNÍCH K_H A G A VYSEKNUTÍ JEN UŽITEČNÉ STŘEDNÍ ČÁSTI.	10
OBR. 2.1: MEL-BANKA FILTRŮ POČÍTÁJÍCÍ S LINEÁRNÍM CHARAKTEREM ŠKÁLY V CELÉM PÁSMU. PÁSMO JE ROZDĚLENO NA 10 ÚSEKŮ (TJ. BANKA OBSAHUJE 10 FILTRŮ).	13
OBR. 2.2: MEL-BANKA FILTRŮ POČÍTÁJÍCÍ S NELINEÁRNÍM CHARAKTEREM ŠKÁLY V CELÉM PÁSMU. PÁSMO JE ROZDĚLENO NA 10 ÚSEKŮ (TJ. BANKA OBSAHUJE 10 FILTRŮ)	13
OBR. 3.1: GRAFY PRAHOVACÍCH PRAVIDEL PRO $\lambda = 1$. U POLOMĚKKÉHO PRAHOVÁNÍ JE ZVOLENO $\lambda_1 = 1$ A $\lambda_2 = 2$.	17
OBR. 4.1: ZÁKLADNÍ PODOBA GRAFICKÉHO UŽIVATELSKÉHO ROZHRAŇÍ „WAVELETOVÉ POTLAČOVÁNÍ ŠUMU V ŘEČI“	19
OBR. 4.2: DIALOG PRO OTEVŘENÍ SOUBORU.	21
OBR. 4.3: ZÁKLADNÍ PANEL GUI S NAČTENÝM SIGNÁLEM A JEDNÍM Z MOŽNÝCH NASTAVENÍ ZOBRAZENÍ	22
OBR. 4.4: PANEL PŘIDÁNÍ ŠUMU	23
OBR. 4.5: PANEL UV KLASIFIKACE	23
OBR. 4.6: ROZDÍLNÝ PŘÍSTUP K PŘECHODU Z NEZNĚLÉHO ÚSEKU NA ZNĚLÝ A NAOPAK. ČERVENĚ JE VYKRESLEN ANALYZOVANÝ SIGNÁL, OSTRÁ KLASIFIKACE JE ZNÁZORNĚNA MODŘE, POSTUPNÁ ČERNĚ.	24
OBR. 4.7: PANEL WAVELETOVÉ POTLAČENÍ ŠUMU	25
OBR. 4.8: CELÝ GUI PO PROVEDENÍ VŠECH KROKŮ OD NAČTENÍ SIGNÁLU AŽ PO POTLAČENÍ V ŠUMU V NĚM	26
OBR. 5.1: KLASIFIKACE ZNĚLÝ A NEZNĚLÝ NA SIGNÁLU BEZ ŠUMU. STATISTICKÁ METODA (A) A METODA MEL-KEPSTRÁLNÍCH KOEFICIENTŮ (B).	32
OBR. 5.2: KLASIFIKACE ZNĚLÝ A NEZNĚLÝ NA SIGNÁLU S ŠUMEM. STATISTICKÁ METODA (A) A METODA MEL-KEPSTRÁLNÍCH KOEFICIENTŮ (B).	32
OBR. 5.3: PŮVODNÍ SIGNÁL BEZ ŠUMU s_0 (A), SIGNÁL S ŠUMEM s_{SUM} (B), SIGNÁL s_V (C), SIGNÁL s_{UV} (D), SIGNÁL s_A (E).	37

Seznam tabulek

TAB. 5.1: TABULKA VÝSLEDNÉHO KOEFICIENTU, KTERÝM JE NÁSOBEN ALGORITMEM AUTOMATICKY URČENÝ KOEFICIENT PRAHU... 33

Úvod

Waveletová transformace je moderním nástrojem pro zpracování signálů. Původně byla používána především v oblasti komprese obrazu, postupně se ale její působnost a využití rozšířily i na zvukové signály. Waveletová transformace je hojně a s úspěchem využívána hlavně pro své unikátní vlastnosti, z nichž jmenujme např. schopnost zachytit rychlé změny nebo nespojitosti ve zpracovávaném signálu. V poslední době patří mezi jednu z nejvýznamnějších oblastí využití waveletové transformace aplikace zabývající se potlačováním šumu v signálech.

Cílem této diplomové práce je vytvořit aplikaci, jež bude implementovat waveletové potlačování aditivního gaussovského šumu v řečových signálech s obecně rozdílným přístupem ke znělým a neznělým segmentům.

Kapitola 1 se zabývá teorií waveletové transformace. Je uvedena definice waveletu, představena integrální waveletová transformace pro spojité signály a zejména pak diskrétní waveletová transformace pro zpracování diskrétních signálů. V této kapitole se čtenář seznámí také s mnohoměřítkovou analýzou (MR-analýza), což je analýza pro rozklad signálu do bazových funkcí různých měřítek. V neposlední řadě je představen a popsán Mallatův algoritmus – rychlý algoritmus pro výpočet waveletové transformace.

Kapitola 2 pojednává o dvou různých metodách klasifikace znělosti a neznělosti fonémů v řečovém signálu. První z nich je jednoduchá statistická metoda založená na označování znělých a neznělých segmentů na základě statistických vlastností řečového signálu, druhou – sofistikovanější – metodou je metoda využívající mel-frekvenčních kepstrálních koeficientů.

Po představení waveletovy transformace a metod klasifikace znělosti a neznělosti fonémů následuje kapitola 3, která se věnuje samotnému waveletovému potlačování šumu. Je vysvětlen princip metody prahování waveletových koeficientů a jsou představena různá prahovací pravidla.

Kapitolou 4 začíná praktická část, která obsahuje návrh a řešení zadaného cíle této diplomové práce. Je navrženo blokové schéma systému pro potlačování šumu v řeči, které je následně implementováno ve vývojovém prostředí MATLAB.

V kapitole 5 je provedeno testování metod uvedených v diplomové práci a dosažené výsledky jsou následně zhodnoceny.

1 Waveletová transformace

1.1 Definice waveletu

Funkci $\psi(t)$ můžeme nazvat (mateřským) waveletem, pokud má následující dvě vlastnosti a pokud splňuje podmínku přípustnosti.

1. Integrál dané funkce je roven nule

$$\int_{-\infty}^{\infty} \psi(t) dt = 0, \quad (1.1)$$

2. Funkce má konečnou energii, neboli

$$\int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty. \quad (1.2)$$

Podmínka přípustnosti:

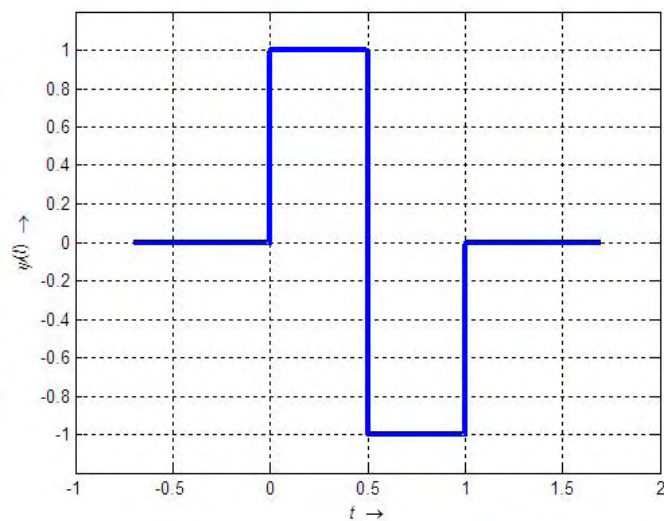
$$\int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty, \quad (1.3)$$

kde $\hat{\psi}$ představuje integrální Fourierovu transformaci funkce ψ .

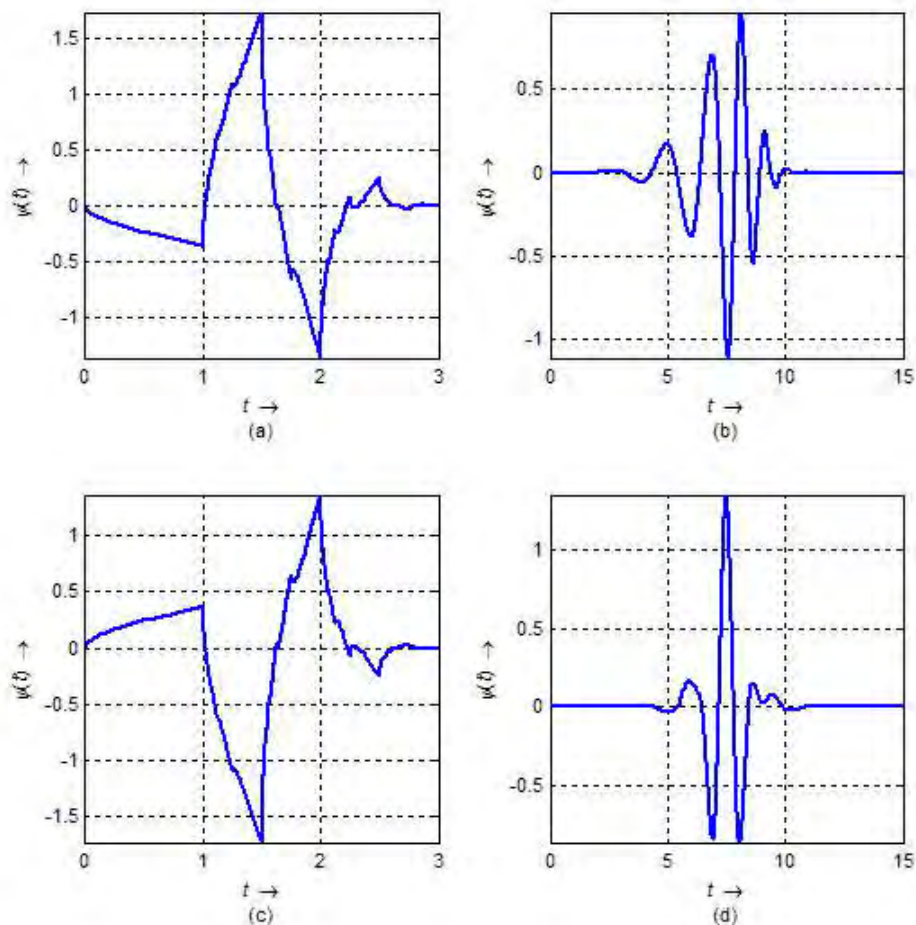
Nejjednodušším waveletem, který vyhovuje výše uvedeným podmínkám, je tzv. Haarův wavelet (viz **Obr. 1.1**): [1]

$$\psi(t) = \begin{cases} 1 & 0 \leq t < 1/2 \\ -1 & 1/2 \leq t < 1 \\ 0 & \text{jinak} \end{cases} \quad (1.4)$$

V praxi velmi rozšířenými wavelety jsou wavelety třídy Daubechies nebo tzv. Symlety. Wavelety každé z těchto dvou tříd lze ještě rozdělit do několika typů označovaných číslicí. Máme potom např. wavelety typu Daubechies 2 (viz **Obr. 1.2a**), Daubechies 8 (viz **Obr. 1.2b**), Symlet 2 (viz **Obr. 1.2c**) nebo Symlet 8 (viz **Obr. 1.2d**).



Obr. 1.1: Matěřský wavelet typu Haar



Obr. 1.2: Matěřský wavelet typu Daubechies 2 (a), Daubechies 8 (b), Symlet 2 (c) a Symlet 8 (d).

1.2 Integrovní waveletová transformace

Nechť je $f(t)$ libovolná funkce s konečnou energií. Integrovní waveletová transformace (CWT¹) funkce $f(t)$ za použití waveletu $\psi(t)$ je definována jako

$$\text{CWT}_f(a,b) = |a|^{-\frac{1}{2}} \int_{-\infty}^{\infty} f(t) \psi^* \left(\frac{t-b}{a} \right) dt, \quad (1.5)$$

kde a a b jsou reálná čísla a $*$ představuje operaci komplexního sdružení. Waveletová transformace je tedy funkce dvou proměnných. Všimněme si, že $f(t)$ i $\psi(t)$ patří do $L^2(\mathbb{R})$, což je množina všech funkcí definovaných na \mathbb{R} .

$$L^2(\mathbb{R}) = \left\{ f \mid \int_{-\infty}^{\infty} |f(t)|^2 dt < \infty \right\}. \quad (1.6)$$

Tyto funkce představují signály s konečnou energií.

Když nadefinujeme

$$\psi_{a,b}(t) = |a|^{-\frac{1}{2}} \psi \left(\frac{t-b}{a} \right), \quad (1.7)$$

můžeme rovnici (1.5) zapsat

$$\text{CWT}_f(a,b) = \int_{-\infty}^{\infty} f(t) \psi_{a,b}^*(t) dt. \quad (1.8)$$

Koeficient $|a|^{-\frac{1}{2}}$ zajišťuje, že energie zůstane pro všechna a a b stejná, čili bude platit

$$\int_{-\infty}^{\infty} |\psi_{a,b}(t)|^2 dt = \int_{-\infty}^{\infty} |\psi(t)|^2 dt \quad (1.9)$$

pro všechna a a b . Pro libovolné a je funkce $\psi_{a,b}(t)$ časové posunutí funkce $\psi_{a,0}(t)$ o parametr b . Z výrazu

$$\psi_{a,0}(t) = |a|^{-\frac{1}{2}} \psi \left(\frac{t}{a} \right) \quad (1.10)$$

¹ Continuous Wavelet Transform, někdy je možno v české literatuře najít pod názvem spojitá waveletová transformace

vychází, že $\psi_{a,0}(t)$ je časově a amplitudově upravená verze $\psi(t)$. Parametr a je tedy dilatační a parametr b translační konstantou. [7]

1.3 Diskrétní waveletová transformace

Definujeme-li spočetně mnoho funkcí

$$\psi_{j,k}(t) = 2^{-j/2} \psi(2^{-j}t - k), \quad j, k \in \mathbb{Z}, \quad (1.11)$$

potom je tzv. diskrétní waveletová transformace definována jako

$$\text{DWT}_f(j, k) = \langle f, \psi_{j,k} \rangle = \int_{-\infty}^{\infty} f(t) \psi_{j,k}^*(t) dt, \quad j, k \in \mathbb{Z}. \quad (1.12)$$

1.4 MR analýza

Pojem MR-Analýza pochází z anglického „multiresolution analysis, MRA“ a do češtiny se většinou překládá jako „mnohoměřítková analýza“. Je to analýza pro rozklad signálu do básových funkcí různých měřítek. Velké hodnoty měřítka j korespondují s reprezentací trendu v signálu a pro $j \rightarrow -\infty$ vystihují básové funkce $\psi_{j,k}$ detailnější chování signálu. Tento koncept umožní rozložit jakýkoliv signál i s nespojitostmi nebo s ostrými hroty.

Pro přípustný wavelet ψ je tento rozklad možno zapsat jako

$$L^2(\mathbb{R}) = \dots \oplus W_2 \oplus W_1 \oplus W_0 \oplus W_{-1} \oplus \dots, \quad (1.13)$$

kde \oplus představuje ortogonální součet prostorů funkcí W_i . Pokud navíc existuje tzv. měřítková funkce $\phi \in L^2(\mathbb{R})$, která generuje MR-analýzu $\{V_j\}_{j \in \mathbb{Z}}$, pak platí

$$V_j = V_{j+1} \oplus W_{j+1} \quad (1.14)$$

a rozklad (1.13) lze upravit na

$$L^2(\mathbb{R}) = V_j \oplus W_j \oplus W_{j-1} \oplus W_{j-2} \oplus \dots \quad (1.15)$$

pro libovolné pevně zvolené j .

Z rovnice (1.14) vychází dva důležité vztahy. Měřítková rovnice

$$\phi(t) = \sqrt{2} \sum_{k=-\infty}^{\infty} h_k \phi(2t - k) \quad (1.16)$$

a waveletová rovnice

$$\psi(t) = \sqrt{2} \sum_{k=-\infty}^{\infty} g_k \phi(2t - k) \quad (1.17)$$

pro dané posloupnosti \mathbf{h} a \mathbf{g} . Funkce ϕ a ψ se tedy dají vyjádřit jako lineární kombinace posunutí měřítkové funkce $\phi(2t)$. [7]

1.5 Konečná diskretní waveletová transformace

Nejčastěji se v praxi ovšem setkáváme s diskretními signály konečné délky. Pro tyto signály je možno z integrální waveletové transformace odvodit verzi pro diskretní signály. Takovou transformaci poté nazveme konečnou diskretní waveletovou transformací (DTWT). Je-li \mathbf{y} vektor délky n , pak jeho konečnou diskretní waveletovou transformací je vektor \mathbf{d}

$$\mathbf{d} = \mathbf{W}\mathbf{y}, \quad (1.18)$$

kde \mathbf{W} je ortogonální matice řádu $n \times n$ a obsahuje speciálně uspořádané prvky z vektorů \mathbf{h} a \mathbf{g} . Díky ortogonalitě \mathbf{W} je možno inverzní waveletovou transformaci vyjádřit

$$\mathbf{d} = \mathbf{W}^{-1}\mathbf{d} = \mathbf{W}^T\mathbf{d}. \quad (1.19)$$

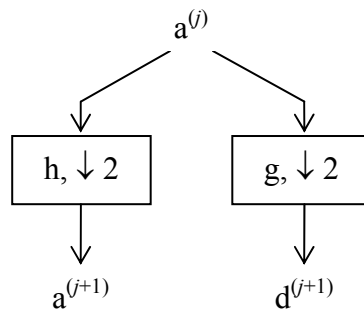
V praxi se nicméně využívá pyramidového algoritmu navrženého S.Mallatem (odtud název Mallatův algoritmus²). Mallat využil toho, že struktura matice \mathbf{W} má pevný řád a navrhl následující rekurzivní algoritmus:

Vstupní signál \mathbf{y} filtrujeme speciálními kvadraturními zrcadlovými filtry h (filtr typu dolní propust) a g (filtr typu horní propust). Každým z filtrů filtrujeme původní signál, takže dostaneme dva nové signály (vektory). Tyto vektory decimujeme (tj. vypustíme každý druhý vzorek) a získáme opět nové posloupnosti. Koeficienty, které jsou výstupem filtru g , nazveme detailními waveletovými koeficienty a koeficienty, které jsou výstupem filtru h , jsou aproximační waveletové koeficienty vstupního signálu. Jeden krok této dekompozice je znázorněn na **Obr. 1.3**. Popsaný postup je jedním krokem waveletové dekompozice. Dalším krokem je provedení výše popsaného postupu na detailních koeficientech získaných prvním krokem waveletové dekompozice.

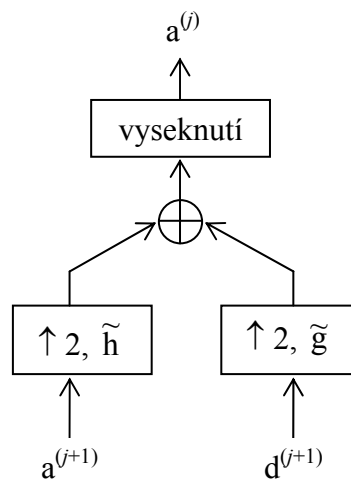
Počet opakování filtrací a decimací je volitelný a je nazýván hloubkou dekompozice. Maximální hloubka dekompozice d_{\max} je omezena vztahem $d_{\max} \leq \log_2 s$.

Algoritmus rekonstrukce je podobný algoritmu dekompozice. Nejprve do posloupností aproximačních a detailních koeficientů vkládáme nuly (provádíme interpolaci – operace opačná k decimaci) a následně filtrujeme filtry \tilde{h} , resp. \tilde{g} , které jsou inverzní k h , resp. g . Výsledky sečteme a vysekneme jen prostřední část, kterou tvoří aproximační koeficienty příslušné k úrovni o jedna nižší. Popsaný postup provádíme tolikrát, kolik je hloubka dekompozice. Po poslední interpolaci a inverzní filtraci získáme výsledný signál v časové doméně. Jeden krok waveletové rekonstrukce je znázorněn na **Obr. 1.4**. [7]

² Mallatovu algoritmu se také říká rychlá waveletová transformace (FWT, Fast Wavelet Transform)



Obr. 1.3: Jeden krok waveletové dekompozice. Aproximační koeficienty j -té úrovně jsou pomocí filtru h , resp. g a decimace rozloženy na aproximační, resp. detailní koeficienty úrovně $j + 1$.



Obr. 1.4: Jeden krok waveletové rekonstrukce. Aproximační koeficienty j -té úrovně jsou vypočítány z aproximačních a detailních koeficientů úrovně o jedna vyšší pomocí interpolace, aplikací filtrací inverzních k h a g a vyseknutí jen užitečné střední části.

2 Klasifikace znělosti a neznělosti fonémů

2.1 Statistická metoda klasifikace

Nejjednodušší metodou klasifikace znělosti a neznělosti je metoda založená na statistické analýze řečového signálu. Tato procedura analyzuje signál po malých časových úsecích (segmentech, např. 30 ms) a rozhoduje, zda je daný úsek znělý či neznělý.

Budeme uvažovat tyto parametry charakterizující jednotlivé segmenty řečového signálu y :

1. Energie signálu v jednom segmentu:

$$E = \sqrt{\frac{1}{N} \sum_{n=1}^N y^2(n)}, \quad (2.1)$$

kde n je pořadí vzorku v segmentu signálu y délky N .

2. Normalizovaný autokorelační koeficient segmentu signálu:

$$r_1 = \frac{\sum_{n=1}^N y(n)y(n-1)}{\sum_{n=1}^N y^2(n)}. \quad (2.2)$$

3. Počet průchodů nulou, neboli kolikrát přejde signál v jednom segmentu přes nulovou hodnotu. Průchod nulou je definován jako $\text{sgn}(y(n)) \neq \text{sgn}(y(n+1))$.

Teď už záleží jen na určení mezních hodnot, podle kterých budeme určovat, zda daný segment považovat za znělý či neznělý. Můžeme např. stanovit, že segment budeme považovat za neznělý, pokud bude platit alespoň jedno z následujících kritérií:

- Energie E je menší než 20% průměrné energie celého signálu (toto kritérium ve skutečnosti indikuje ticho, ne neznělost).
- Normalizovaný autokorelační koeficient $r_1 < 0,5$.
- Počet průchodů nulou v daném segmentu je 1,5krát větší než průměrný počet průchodů nulou v jednom segmentu.

Pokud nebude splněno ani jedno z výše uvedených kritérií, daný segment bude označen jako znělý. V této jednoduché metodě založené na základní statistické analýze signálu je opět problém s některými hláskami, např. se „Z“, které je sice znělé, ale touto metodou je z důvodu velkého počtu průchodu nulou vyhodnoceno jako neznělé. Právě z tohoto a podobných důvodů je třeba přistoupit k některým více sofistikovaným metodám určování znělosti a neznělosti, kterým bude věnována pozornost v diplomové práci.

2.2 Metoda mel-kepstrálních koeficientů

2.2.1 Mel-banka filtrů

Banka filtrů obecně slouží k tomu, abychom mohli získat soubor informací s co největší vypovídací hodnotou. Banka filtrů slouží k vhodnému rozdělení spektra signálu, výstupy jednotlivých filtrů pak tvoří jednotlivé příznaky signálu. Ve zkratce jde o filtraci spektra signálu bankou filtrů.

Základem mel-banky filtrů je kmitočtová stupnice s jednotkou mel (jinak řečeno mel-frekvenční stupnice). Mel je jednotkou subjektivní výšky zvuku, která byla zavedena z toho důvodu, že subjektivní vjem změny výšky zvuku neodpovídá změně kmitočtu v Hz. Je definováno, že tón o kmitočtu 1000 Hz a hlasitosti 40 Ph³ odpovídá hodnotě subjektivní výšky 1000 mel. Potom na rozdíl od tónů vyjádřených v Hz platí, že dva tóny vzdálené od sebe na mel-frekvenční stupnici o stejný počet jednotek mel jsou také vnímány jako stejně vzdálené. Např. tón o frekvenci 2000 mel má pro posluchače dvakrát vyšší výšku než tón o frekvenci 1000 mel (to samé ale nelze říct o tónech o kmitočtech 2000 Hz a 1000 Hz). Rozložení výsledné mel-frekvenční škály je do přibližně 800 Hz lineární a nad 800 Hz vnímá člověk vztah mezi výškou a skutečným kmitočtem logaritmicky. Pro přepočtení skutečného kmitočtu v Hz na frekvenci vyjádřenou v jednotkách mel je definován

$$f_{\text{mel}} = 2595 \log \left(1 + \frac{f}{700} \right), \quad (2.3)$$

kde f_{mel} je frekvence v mel a f kmitočty v Hz. Pro přepočtení opačným směrem platí inverzní vztah

$$f = 700 \cdot 10^{\left(\frac{f_{\text{mel}} - 1}{2595} \right)}. \quad (2.4)$$

Samotnou mel-banku pak zpravidla tvoří sada trojúhelníkových filtrů, které mohou mít konstantní nebo klesající modul přenosu. Tato volba záleží na spektrálních vlastnostech řeči. Jednotlivé filtry jsou v mel-frekvenční oblasti rozloženy lineárně. Za horní hranici se většinou volí Nyquistův kmitočet, ale samozřejmě ji lze umístit i níže, pokud je předpokládána oblast využití užší. Výpočet hodnot trojúhelníkových filtrů na jednotlivých kmitočtech k je následující:

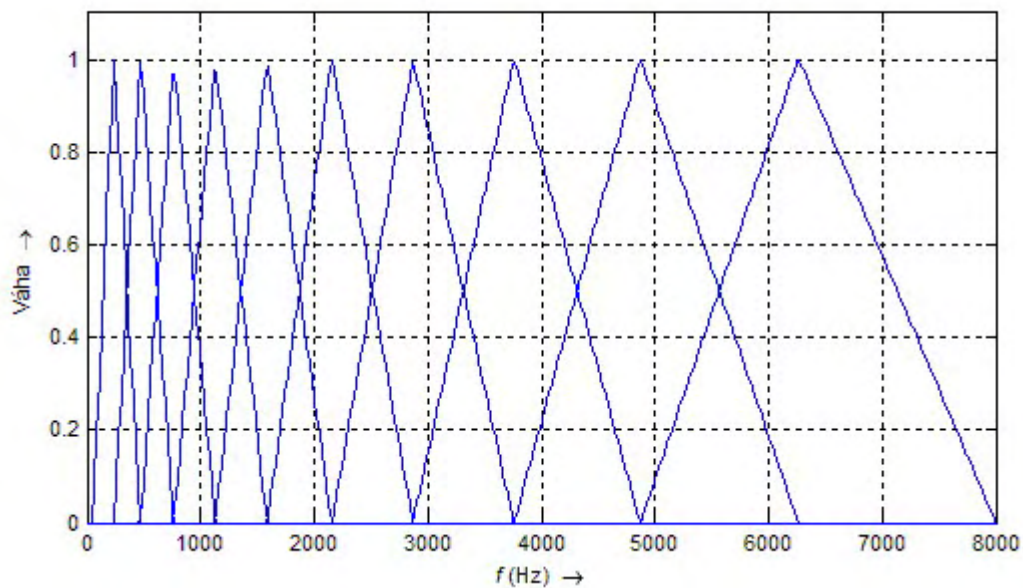
$$H_m(k) = \begin{cases} 0 & \text{pro } k < f(m-1), \\ \frac{[k-f(m-1)]}{[f(m)-f(m-1)]} & \text{pro } f(m-1) \leq k \leq f(m), \\ \frac{[f(m+1)-k]}{[f(m+1)-f(m-1)]} & \text{pro } f(m) \leq k \leq f(m+1), \\ 0 & \text{pro } k > f(m+1), \end{cases} \quad (2.5)$$

kde $H_m(k)$ reprezentuje funkci filtru, k je počet vzorků FFT, m reprezentuje daný filtr a f převod ze stupnice jednotek mel do Hz. Při přepočtu jednotlivých filtrů do kmitočtové oblasti

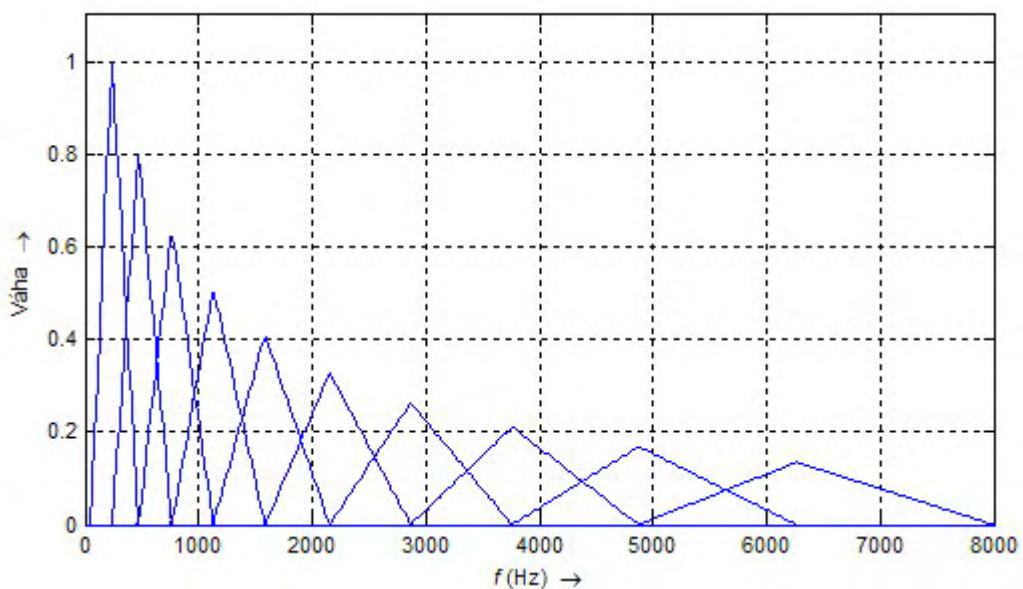
³ Fón - jednotka hladiny hlasitosti, která odpovídá tónům, které člověk vnímá stejně hlasitě. 1 fón odpovídá hladině intenzity 1 dB pro tón o frekvenci 1 kHz.

pracujeme s jejich středními kmitočty $f_{\text{mel}}(m)$ a využíváme rovnice (2.4) pro přepočítání do frekvenční stupnice v Hz. Dále je třeba nadefinovat dolní mezní frekvenci f_L a horní mezní frekvenci f_H filtrů v Hz, která je potom převedena do mel-frekvenční škály. Střední kmitočty, a tedy i hraniční body jednotlivých filtrů, se vypočítají podle vztahu:

$$f(m) = f \left(f_{L,\text{mel}} + m \frac{f_{H,\text{mel}} - f_{L,\text{mel}}}{M + 1} \right), \quad (2.6)$$



Obr. 2.1: Mel-banka filtrů počítající s lineárním charakterem škály v celém pásmu. Pásmo je rozděleno na 10 úseků (tj. banka obsahuje 10 filtrů).



Obr. 2.2: Mel-banka filtrů počítající s nelineárním charakterem škály v celém pásmu. Pásmo je rozděleno na 10 úseků (tj. banka obsahuje 10 filtrů).

kde M je počet filtrů (počet úseků, na které bude pásmo ohraničené frekvencemi f_L a f_H rozděleno). Příklad banky filtrů pro $M = 10$, $f_L = 50$ Hz a $f_H = 8$ kHz je na **Obr. 2.1**. Místo standardního rozložení filtrů podle rovnice (2.5) jsou v praxi uplatňovány především banky filtrů respektující vesměs lineární charakter na počátku pásma a nelineární charakter v jeho zbytku; nebo filtry, které předpokládají nelineární charakter v celém pásmu (viz **Obr. 2.2**). [10]

2.2.2 Mel-frekvenční keprální koeficienty

Jelikož lidské vnímání zvuku nemá lineární charakter, je třeba najít metodu, která toto zohlední a za pomoci které budeme moci extrahovat příznaky řeči. Vhodnou filtrací a transformací signálu je možno extrahovat např. mel-keprální koeficienty, které nelineární charakter lidského vnímání zvuku v úvahu berou. Jsou definovány jako diskrétní kosinová transformace logaritmu výstupů bank filtrů. Mel-keprální koeficienty získáme následujícím postupem.

Nejdříve je třeba vypočítat diskrétní spektrum úseku řečového signálu:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}nk} \quad \text{pro } 0 \leq k \leq N. \quad (2.7)$$

Poté získané spektrum rozdělíme výše popsanou mel-bankou filtrů, tzn. každý koeficient FFT vynásobíme odpovídajícím koeficientem mel-banky. Získáme tak M výstupních hodnot rovných počtu filtrů reprezentujících energii v daném filtru. Výstupy jednotlivých filtrů jsou ale vzájemně značně korelované a nejsou tedy vhodné pro parametrický popis řeči. Z tohoto důvodu se převádí na koeficienty keprální. Předposledním krokem k získání keprálních koeficientů je tedy logaritmizace energie každého filtru podle vztahu

$$e(m) = \ln \left[\sum_{k=0}^{N-1} |X(k)|^2 H_m(k) \right] \quad \text{pro } 0 \leq m \leq M., \quad (2.8)$$

Posledním krokem pak je inverzní DFT. Jelikož ale pracujeme pouze s logaritmem nezáporného reálného spektra, což je sudá funkce, lze IDFT nahradit aproximací výpočetně méně náročnější diskrétní kosinovou transformací (DCT-II)

$$c(n) = \sum_{m=0}^{M-1} e(m) \cos\left(\frac{\pi n(m - \frac{1}{2})}{M}\right). \quad (2.9)$$

Koeficient $c(0)$ představuje průměrnou hodnotu log-modulu spektra a několik prvních koeficientů $c(n)$ (zpravidla do hodnoty $n = 12$ nebo $n = 13$) představuje vlastnosti hlasového traktu. [8], [10]

Pro naše potřeby je důležitý pouze koeficient $c(0)$, který představuje energii signálu a je postačující k určení, zda je daný segment řečového signálu znělý nebo neznělý.

3 Waveletové potlačování šumu

Nejjednodušším přístupem k dosažení potlačení šumu za použití waveletové transformace je tzv. „prahování“.

Šum, který se přidá k signálu, má obvykle aditivní charakter, takže můžeme psát

$$\mathbf{X} = \mathbf{f} + \varepsilon, \quad (3.1)$$

kde \mathbf{f} je původní signál, ε je náhodný signál s normálním rozložením pravděpodobnosti a \mathbf{X} je signál výsledný.

S využitím linearity waveletové transformace (transformace je vyjádřena pomocí matice \mathbf{W}) přejde vztah (3.1) v

$$\mathbf{Y} = \mathbf{b} + \nu, \quad (3.2)$$

kde $\mathbf{Y} = \mathbf{W}\mathbf{X}$, $\mathbf{b} = \mathbf{W}\mathbf{f}$, $\nu = \mathbf{W}\varepsilon$. Rozložení pravděpodobnosti šumu zůstalo díky ortogonalitě transformace stejné. Vektory \mathbf{Y} , \mathbf{b} a ν představují waveletové koeficienty příslušné vektorům \mathbf{X} , \mathbf{f} a ε .

Cílem waveletového potlačování šumu je nalézt určitý předpis tak, aby odhad $\hat{b}_i \approx b_i$. Koeficienty b_i jsou totiž waveletové koeficienty příslušející signálu nezatíženého šumem. Po úpravě waveletových koeficientů provedeme za pomoci matice \mathbf{W}^T operaci inverzní k waveletové transformaci a signál tak navrátíme do původní domény. Výsledkem bude signál s potlačenou šumovou složkou. Výše popsaná metoda byla nazvána WaveShrink.

Tento přístup je analogický ke klasické fourierovské filtraci. Rozdíl mezi Fourierovou a waveletovou transformací je v tom, že v případě první jmenované má každý koeficient globální působnost. U waveletové transformace mají její koeficienty jen lokální vliv, takže tento typ transformace dovoluje konstruovat lokálně adaptivní filtry.

Mezi nejčastější metody konstruování odhadů \hat{b}_i patří techniky založené na prahování. Princip je velmi jednoduchý. Hodnoty signálu y_i , jejichž absolutní hodnota je menší než zvolený práh $\lambda > 0$, jsou vynulovány a ostatní jsou buď ponechány v původní podobě nebo ještě nějak upraveny.

Waveletová transformace koncentruje nejvíce energie z vektoru \mathbf{f} pouze do malého počtu waveletových koeficientů b_i . Tento fakt spolu s tím, že ν je vektorem obsahujícím koeficienty náhodného signálu (šumu) s normálním rozložením pravděpodobnosti, vysvětluje princip waveletového potlačování šumu. Když je vhodně zvolen práh λ , prahování se neprojeví na koeficientech y_i , které reprezentují především deterministický signál, ale zasáhne koeficienty y_i , které reprezentují aditivní šum.

Prahaování aplikujeme pouze na detailní koeficienty, neboť není vhodné modifikovat aproximační koeficienty, které vyjadřují trend v signálu. [7]

3.1 Typy prahovacích pravidel

V následujících příkladech bude funkce $\delta(y_i, \lambda)$ vyjadřovat odhad koeficientů \hat{b}_i .

1. Tvrdé prahování

Všechny waveletové koeficienty, které jsou pod prahovou hodnotou, jsou vynulovány. Ostatní jsou ponechány beze změny: $\hat{b}_i = \delta^h(y_i, \lambda)$, kde

$$\delta^h(x, \lambda) = \begin{cases} 0 & \text{pro } |x| \leq \lambda \\ x & \text{pro } |x| > \lambda. \end{cases}$$

Viz Obr. 3.1 (a).

2. Měkké prahování

Waveletové koeficienty pod prahovou hodnotou jsou opět vynulovány, ale velikost ostatních koeficientů je zmenšena o velikost prahu: $\hat{b}_i = \delta^s(y_i, \lambda)$, kde

$$\delta^s(x, \lambda) = \text{sgn}(x) \max\{0, |x| - \lambda\}.$$

Viz Obr. 3.1 (b).

3. Poloměkké prahování

Zde je třeba definovat dvě prahové hodnoty, a to tak, že $0 < \lambda_1 \leq \lambda_2$. $\hat{b}_i = \delta^{ss}(y_i, \lambda_1, \lambda_2)$, kde

$$\delta^{ss}(x, \lambda_1, \lambda_2) = \begin{cases} 0 & \text{pro } |x| \leq \lambda_1 \\ \text{sgn}(x) \frac{\lambda_2(|x| - \lambda_1)}{\lambda_2 - \lambda_1} & \text{pro } \lambda_1 < |x| \leq \lambda_2 \\ x & \text{pro } |x| > \lambda_2. \end{cases}$$

Viz Obr. 3.1 (c).

4. Nezáporná garota

$\hat{b}_i = \delta^{\text{ng}}(y_i, \lambda)$, kde

$$\delta^{\text{ng}}(x, \lambda) = \begin{cases} 0 & \text{pro } |x| \leq \lambda \\ x - \frac{\lambda^2}{x} & \text{pro } |x| > \lambda. \end{cases}$$

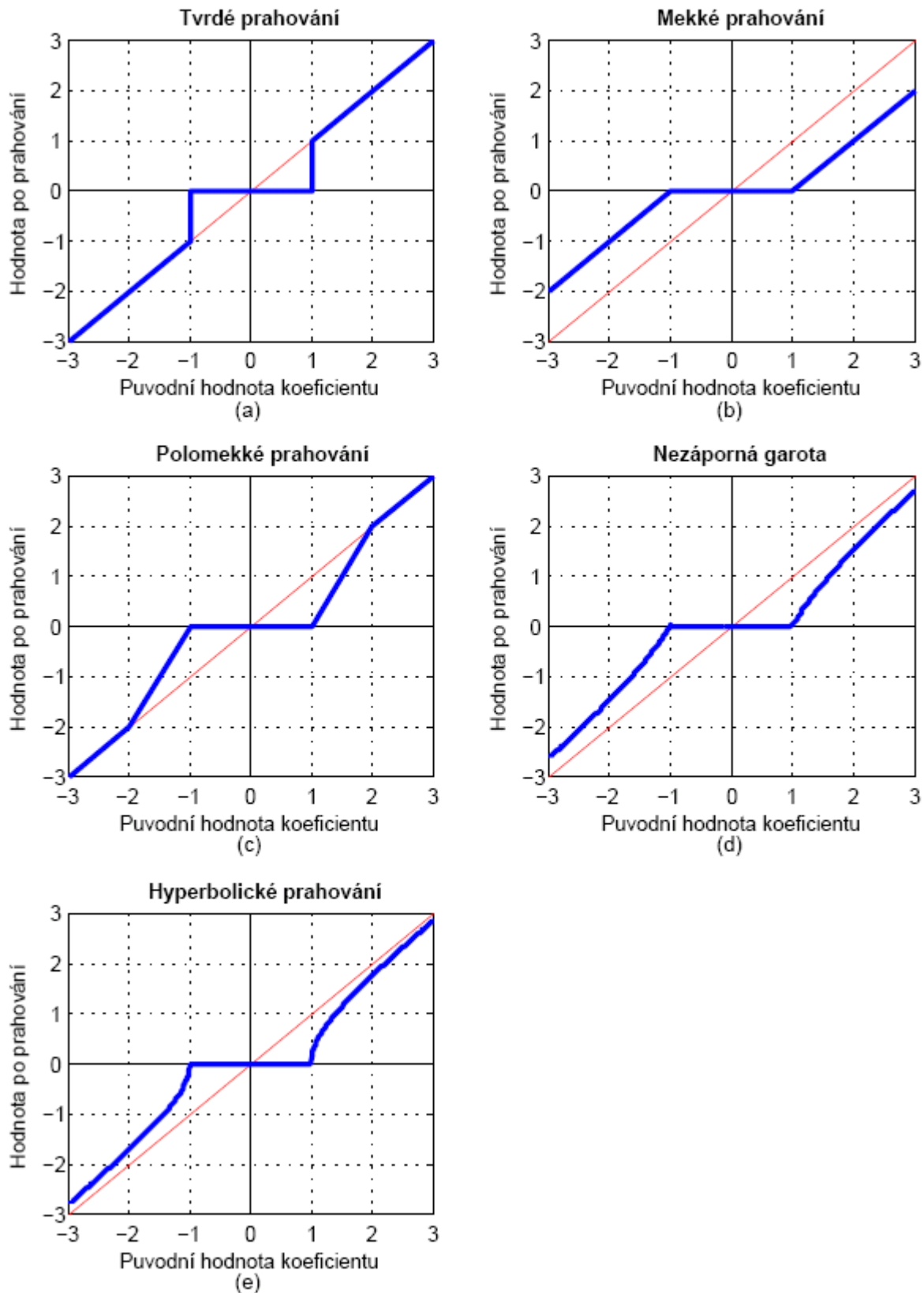
Viz Obr. 3.1 (d).

5. Hyperbolické prahování

$\hat{b}_i = \delta^{\text{hy}}(y_i, \lambda)$, kde

$$\delta^{\text{hy}}(x, \lambda) = \begin{cases} 0 & \text{pro } |x| \leq \lambda \\ \text{sgn}(x) \sqrt{x^2 - \lambda^2} & \text{pro } |x| > \lambda. \end{cases}$$

Viz Obr. 3.1(e). [7]



Obr. 3.1: Grafy prahovacích pravidel pro $\lambda = 1$. U polomekkého prahování je zvoleno $\lambda_1 = 1$ a $\lambda_2 = 2$.

4 Návrh systému pro potlačování šumu v řeči a jeho implementace

V této kapitole se budu zabývat implementací výše popsaného ve vývojovém prostředí MATLAB. V MATLABu jsem vytvořil uživatelské rozhraní (GUI), které slouží k implementaci klasifikace znělosti a neznělosti a waveletového potlačování šumu. Popíši a vysvětlím vytvořený GUI a programové kódy s ním spjaté. V příloze č.1 je zobrazeno blokové schéma systému implementovaného v GUI.

4.1 Waveletové potlačování šumu v řečovém signálu

Standardní řečový signál se skládá ze znělých částí, neznělých částí a pauz mezi jednotlivými částmi slovního projevu – ticha.

S waveletovým potlačováním šumu ve znělých částech není žádný podstatný problém, neboť znělé úseky řeči neobsahují žádný nebo jen velmi slabý šum. Waveletová transformace provedená nad těmito úseky tedy odfiltruje jen aditivní šum a vyšší frekvenční složky řečového signálu. Pokud je odstup signálu od šumu dostatečně velký, řečový signál s potlačenou šumovou složkou zní téměř stejně jako signál původní. Pokud je odstup signálu od šumu malý (tj. šum je vzhledem k signálu silný), spolu s šumovou složkou jsou výrazně potlačeny i vyšší kmitočty řečového signálu, což způsobí jeho degradaci.

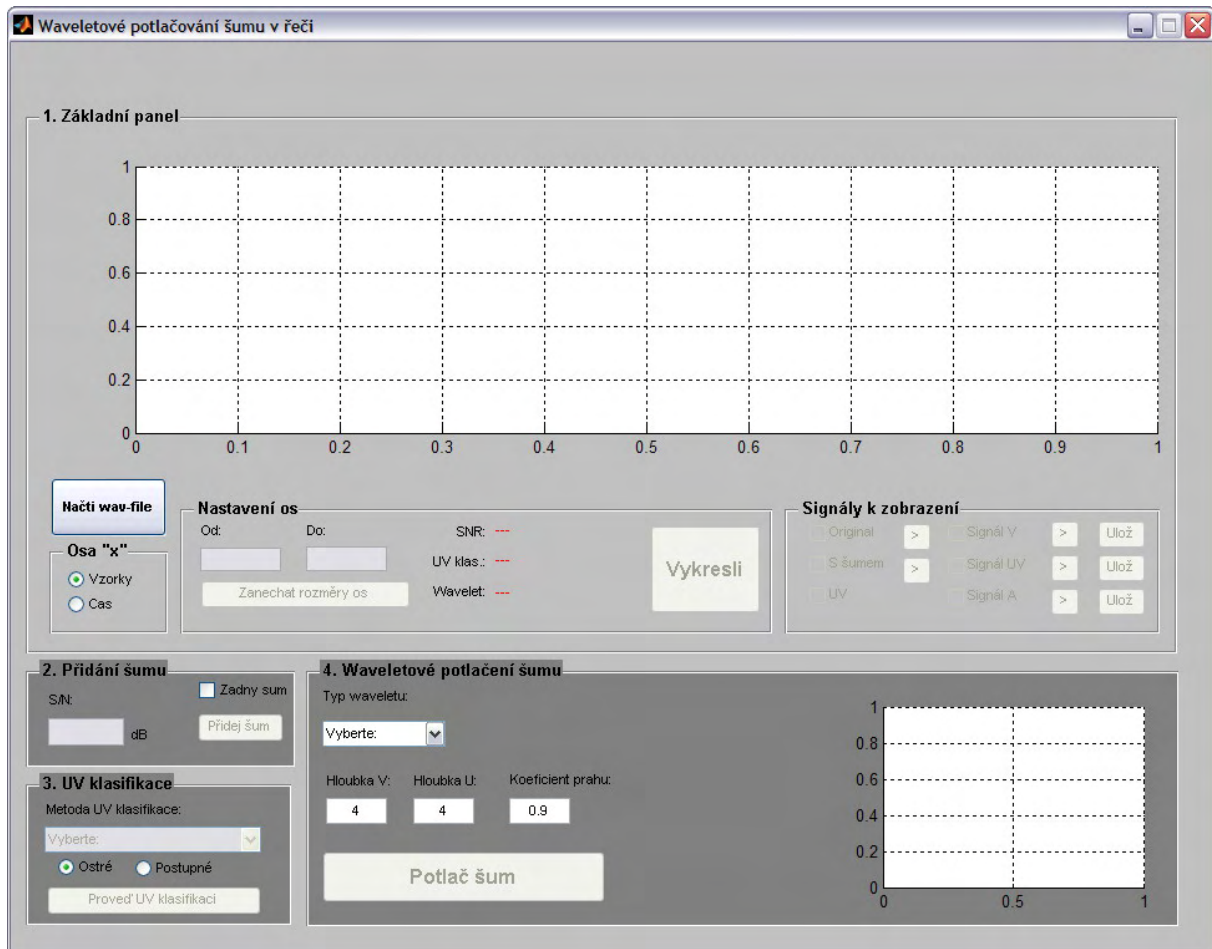
Problém nastává, když potřebujeme potlačit šum v částech neznělých a přejeme si zachovat srozumitelnosti řeči. Pokud provedeme waveletovou transformaci se stejnými parametry jako nad úseky znělými, spolu s šumem odfiltrujeme i neznělé části projevu jako např. sykavky, neboť ty mají charakter šumu. Řešením může být při waveletovém potlačování šumu aplikovat rozdílný přístup k znělým a neznělým úsekům. Nad neznělými úseky potom můžeme buď potlačování šumu vůbec neprovádět nebo lépe provádět, ale s rozdílnými parametry (např. s menší intenzitou nebo s menším prahem).

4.2 GUI – Grafical User Interface

Zkratka GUI znamená v překladu „grafické uživatelské rozhraní“. V rámci této diplomové práce jsem vytvořil GUI, jehož podobu těsně po spuštění můžeme vidět na **Obr. 4.1**.

Jak lze vidět, GUI se skládá z několika logicky od sebe oddělených částí. Každý tento oddíl se stará o specifickou část úpravy signálu od načtení, eventuálního přidání šumové složky, přes klasifikaci znělosti a neznělosti jednotlivých segmentů řeči, až po konečnou aplikaci algoritmu potlačujícího šum. V následujícím seznamu sepíši a velmi stručně vysvětlím jednotlivé části GUI. Podrobnější fungování popíši dále.

Veškeré názvy ovládacích prvků a objektů v GUI a možnosti či volby, které jsou v GUI na výběr, jsou v textu vypsány *kurzívou*. Veškeré proměnné nebo funkce, které referují přímo na proměnné nebo funkce použité v programovém kódu, jsou vypsány fontem Courier New.



Obr. 4.1: Základní podoba grafického uživatelského rozhraní „Waveletové potlačování šumu v řeči“.

1. Panel *Základní panel*

- Plocha pro vykreslení grafu
 - Tento souřadný systém os slouží jako hlavní plocha k vykreslování veškerých signálů.
- Tlačítko *Načti wav-file*
 - Tlačítko otevře dialogové okno pro načtení wav-souboru.
- Skupina tlačítek *Osa „x“*
 - Je možno zaškrtnou buď volbu *Vzorky* nebo *Čas*. Osa „x“ potom bude cejchována podle této volby.
- Panel *Nastavení os*
 - Editovací rámečky slouží k specifikaci rozpětí osy „x“.
 - Tlačítko *Zanechat rozměry os* ovládá zmrazení rozpětí osy „x“ a „y“.
 - Tlačítko *Vykresli* spouští funkci, která na základě nastavení parametrů v panelech *Osa „x“*, *Nastavení os* a *Signály k zobrazení* vykreslí graf/y signálu/ů.
- Panel *Signály k zobrazení*
 - Zaškrťovací políčka k volbě signálů, které si přejeme zobrazit.

- Vedle každého zaškrťovacího políčka kromě *UV* je tlačítko, které slouží k okamžitému zvukovému přehrání odpovídajícího signálu.
- V posledním sloupci ovládacích prvků jsou tlačítka, které provedou uložení odpovídajícího signálu s potlačenou šumovou složkou do adresáře, z něhož byl načten původní signál.

2. Panel *Přidání šumu*

- Zaškrťovací políčko *Žádný šum*
 - Je-li zaškrtnuto, nepřijeme si přidat žádný další aditivní šum.
- Do editovacího rámečku *S/N* se píše požadovaný odstup signálu od přidaného šumu do signálu.
- Tlačítko *Přidej šum* přidá do původního signálu bílý šum s definovaným odstupem signálu od šumu.

3. Panel *UV klasifikace*

- Rozbalovací menu *Metoda UV klasifikace*
 - Je možno vybrat ze dvou metod klasifikace znělosti a neznělosti. Statistickou metodu založenou na prahování nebo metodu opírající se o mel-kepstrální koeficienty.
- Upřesnění klasifikace znělosti a neznělosti
 - Volba *Ostré* značí standardní označení znělých úseků jedničkou a neznělých nulou.
 - Volba *Postupné* upraví standardní označení tak, že přechody z nuly na jedničku při přechodu z neznělého segmentu na znělý nebo naopak nebudou „ostré“, ale postupné. Blíže viz dále v textu.
- Tlačítko *Proved' UV klasifikaci* provede klasifikaci znělosti a neznělosti segmentů v celém signálu.

4. Panel *Waveletové potlačení šumu*

- Rozbalovací menu *Typ waveletu*
 - Volíme základní typ použitého waveletu.
 - Pokud má zvolený typ ještě podtypy (většinou označeny čísly), objeví se další rozbalovací menu *Řád waveletu*, ve kterém vybereme požadovaný typ.
- Editovací rámečky *Hloubka V* a *Hloubka U*
 - V obou rámečcích uvádíme požadovanou hloubku dekompozice při procesu potlačování šumu.
 - V rámečku *Hloubka V* je uvedena hlavní požadovaná hloubka dekompozice a hloubka dekompozice pro znělé segmenty u signálu s_{UV} – bude vysvětleno dále.
 - V rámečku *Hloubka U* je uvedena hloubka dekompozice použitá pro potlačení šumu v neznělých segmentech již zmíněného signálu s_{UV} .
- Editovací rámeček *Koeficient prahu*
 - Zde je možno zvolit určitý koeficient, který upraví zvolenou velikost prahu především v neznělých segmentech řečového signálu – podrobněji dále.
- Souřadný systém os pro vykreslení zvoleného waveletu.

- Tlačítko *Potlač šum* spustí na základě všech zvolených parametrů proces potlačování šumu.

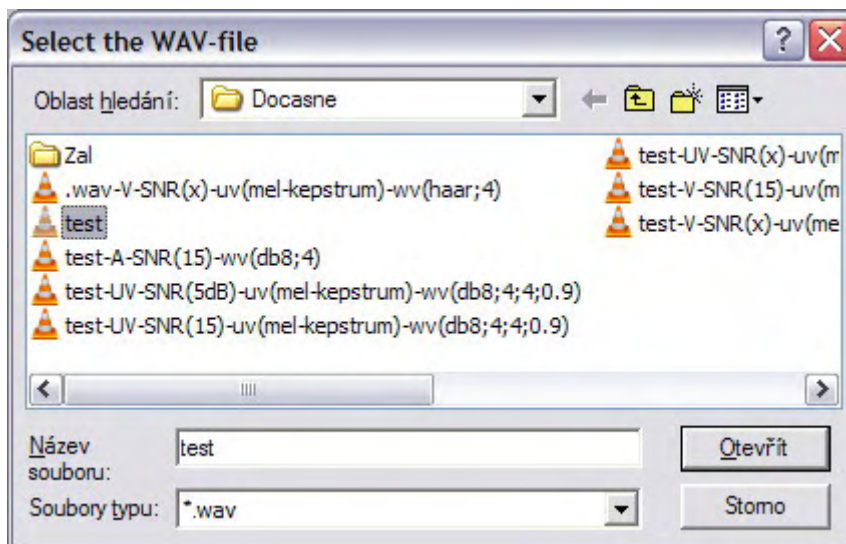
Dále krok po kroku (a blok po bloku) popíši, jak pracovat s výše představeným a popsaným grafickým uživatelským rozhraním. Zároveň budou vysvětleny jednotlivé postupy použité při návrhu či programování specifické části kódu.

4.2.1 Základní panel

Hned po spuštění GUI nemáme moc jiných možností, než si načíst signál ve formátu *.wav, se kterým budeme pracovat. Po stisknutí tlačítka *Načti wav-file* tedy vyskočí standardní dialogové okno pro otevírání souborů (viz Obr. 4.2).

Po stisknutí tlačítka *Otevřít* se zvolený soubor načte a v případě, že je to regulérní WAV-soubor, signál se uloží do proměnné, která reprezentuje původní signál (s_0), a spustí se část kódu, která je zodpovědná za vyplnění editovacích rámečků *Od* a *Do* v panelu *Nastavení os* a za zaškrtnutí signálu *Original* v panelu *Křivky k zobrazení*. Poté dojde ke spuštění funkce *vykresleni_axes_UV* (viz str.30). Také se odemkne panel *Přidání šumu*, abychom mohli pokračovat dále. V případě, že zvolený soubor nemá příponu *.wav, nad grafem se objeví chybová hláška a funkce tlačítka je předčasně ukončena. Než budeme pokračovat dále v přidávání a následném potlačování šumu, vysvětlím ještě další ovládací prvky v panelu *Základní panel*.

Jedním z logických oddílů panelu *Základní panel* je skupina tlačítek *Osa „x“*. Tento prvek lze ovládat dokonce ještě před samotným načtením zpracovávaného signálu. Volba zde provedená určuje, jestli bude osa „x“ cejchována ve vzorcích signálu nebo jestli to bude osa časová. Volbu lze provádět buď před načtením signálu, v tom případě se prakticky nic očividného dít nebude, nebo kdykoli během práce v GUI. Při každé změně (ne při každém kliknutí na jedno z tlačítek) je spuštěna funkce *vykresleni_axes_UV* (viz str.30).

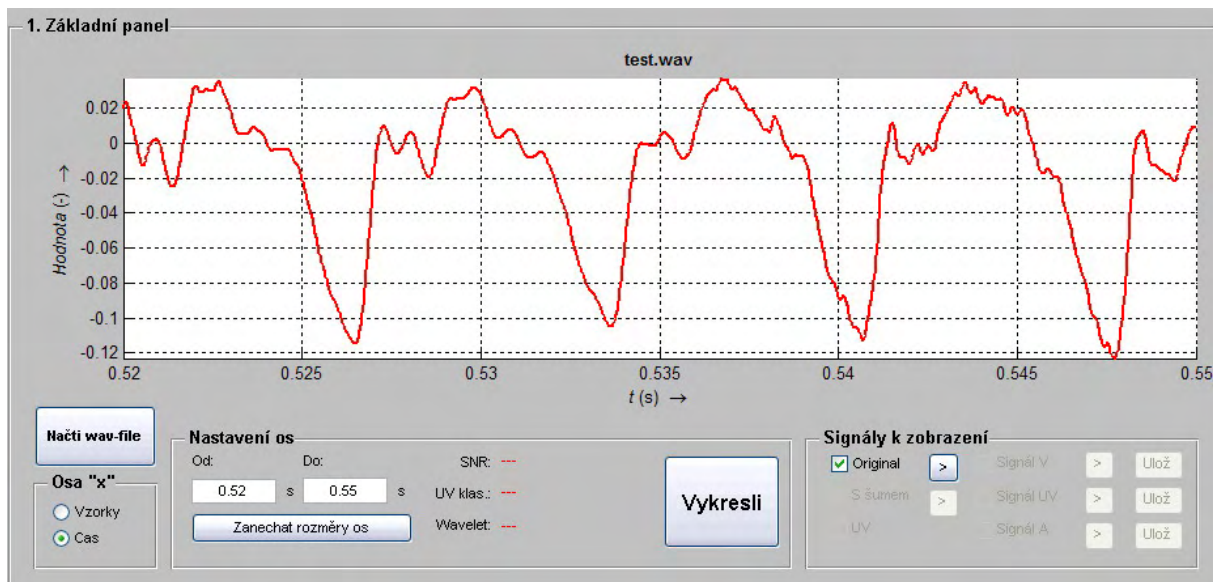


Obr. 4.2: Dialog pro otevření souboru.

Dalším z logických oddílů tohoto panelu je panel *Nastavení os*. Do editovacích rámečků *Od* a *Do* zde můžeme zadat, jaká část signálu se bude vykreslovat. Hned pod těmito rámečky je tlačítko *Zanechat rozměry os*, které má dvě polohy. Při zamáčknutí do polohy „dole“ tlačítko zamkne aktuální nastavení rozpětí osy „x“ i osy „y“, v poloze „nahore“ se

dolní a horní hranice obou os nastavují opět podle nastaveného rozměru os v editovacích rámečcích a podle velikosti zobrazeného signálu. Tlačítko *Vykresli* spouští opět funkci `vykresleni_axes_UV` (viz str.30).

Mezi tlačítkem *Vykresli* a editovacími rámečky se nachází ještě jeden prvek s řádky popsanými *SNR*, *UV klas.* a *Wavelet*. Zde se vypisuje aktuální nastavení voleb, které provedeme dále.



Obr. 4.3: Základní panel GUI s načteným signálem a jedním z možných nastavení zobrazení.

Panel *Signály k zobrazení* nám umožňuje zvolit, jaké signály chceme současně vykreslit (k tomu slouží zaškrťovací políčka před názvy jednotlivých signálů). V průběhu práce s GUI se volba jednotlivých signálů automaticky povoluje nebo zakazuje podle toho, jak se v GUI pohybujeme, co děláme a které signály v systému zrovna existují a jsou aktuální. Máme na výběr z původního signálu (*Original*), signálu s šumem (*S šumem*), klasifikace znělosti a neznělosti segmentů (*UV*) a ze tří signálů s potlačenou šumovou složkou – viz dále v textu. Vedle každého signálu je také tlačítko \triangleright , které slouží k okamžitému zvukovému přehrání odpovídajícího signálu za pomoci matlabovské funkce `soundsc`. O tlačítkách *Ulož* budu psát také až dále v sekci věnované samotnému potlačování šumu.

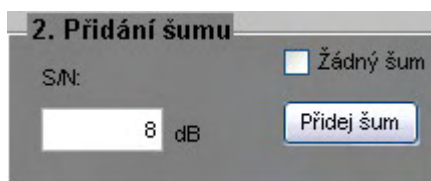
Jak může panel *Základní panel* vypadat za jednoho konkrétního nastavení zobrazovaných parametrů můžeme vidět na **Obr. 4.3**. Nyní můžeme přistoupit k další části zpracování signálu, která je implementována v panelu *Přidání šumu*.

4.2.2 Přidání šumu

Zde máme v podstatě dvě volby. První možností je, že zaškrtneme políčko *Bez šumu* a do proměnné reprezentující signál s šumem (`s_sum`) se uloží signál původní (`s0`). Tuto volbu použijeme v případě, že námi zvolený signál již šum obsahuje. Druhou možností je zvolit si požadovaný odstup signálu od šumu v jednotkách dB a vyplnit ho do editovacího rámečku *S/N*. Poté stačí stisknout tlačítko *Přidej šum* a šum vygenerovaný matlabovskou funkcí `awgn` je vložen do původního signálu a signál i s šumem je uložen do proměnné `s_sum`. Po stisknutí tlačítka *Přidej šum* se také pod grafem vypíše hodnota zvoleného odstupu signálu od šumu *S/N*, zaškrtně se políčko v panelu *Signály k zobrazení* u signálu *S šumem* a zavolá funkce

vykresleni_axes_UV (viz str.30). Ať už jsme do signálu šum přidali nebo ne, odemkne se nám k editaci panel *UV Klasifikace*.

Panel *Přidání šumu* i s nastavením parametrů můžeme vidět na **Obr. 4.4**.



Obr. 4.4: Panel *Přidání šumu*

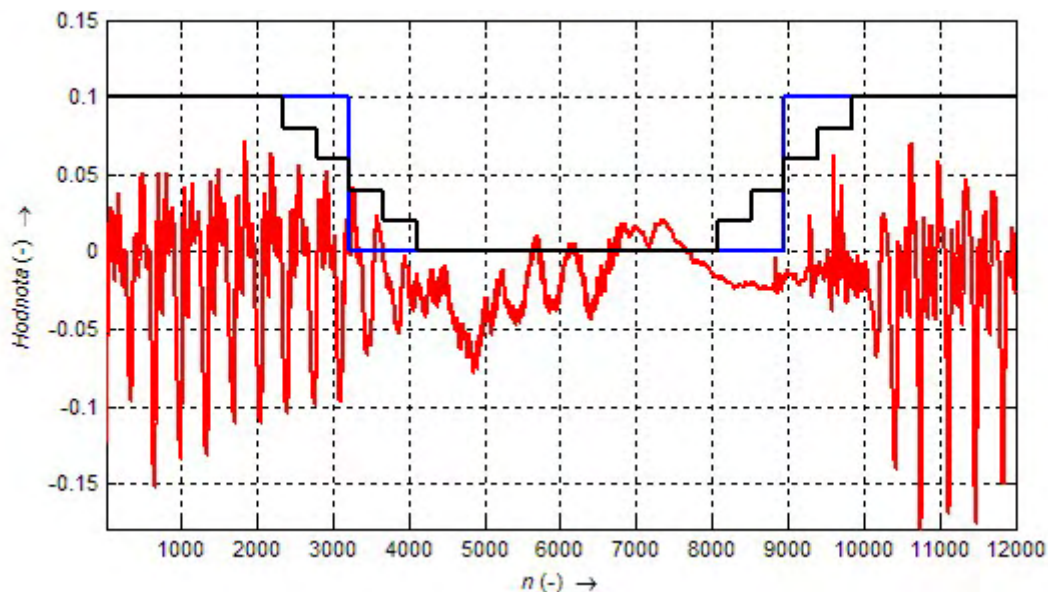
4.2.3 UV klasifikace

Zde systém provádí po segmentech klasifikaci celého signálu na znělost a neznělost. Signál je označován za znělý nebo neznělý po 10 ms intervalech, znělost a neznělost je ale počítána pro 30 ms okna, aby byl zajištěn vzájemný překryv analyzovaných segmentů. Panel *UV klasifikace* je na **Obr. 4.5**.



Obr. 4.5: Panel *UV klasifikace*

Na výběr máme ze dvou metod. Buď ke klasifikaci použijeme statistickou metodu (viz kapitola 2.1, str. 11) nebo metodu založenou na využití mel-kepstrálních koeficientů (viz kapitola 2.2, str. 12). Dále je třeba zvolit, zda výstupem bude signál s dvěma hodnotami, a sice jedničkou pro znělé a nulou pro neznělé segmenty (volba *Ostré*), nebo nebudou přechody z nuly na jedničku v rámci dvou sousedních segmentů okamžité, ale bude to chvíli trvat (volba *Postupné*). Rozdíl mezi oběma přístupy je názorně vidět na **Obr. 4.6**. V případě volby *Postupné* se výstupní klasifikující signál ve dvou bezprostředně po sobě navazujících segmentech liší maximálně o hodnotu 0,2. Po stisknutí tlačítka *Proved' UV klasifikaci* systém připraví vstupní parametry pro funkci *uv_detect.m* (viz str.28) a spustí ji. Dále je zaškrtnuto políčko *UV* v panelu *Signály k zobrazení* a pod grafem je vedle nápisu *UV klas* vypsána vybraná metoda. Nakonec je opět zavolána funkce *vykresleni_axes_UV* (viz str.30).



Obr. 4.6: Rozdílný přístup k přechodu z neznělého úseku na znělý a naopak. Červeně je vykreslen analyzovaný signál, Ostrá klasifikace je znázorněna modře, Postupná černě.

4.2.4 Waveletové potlačení šumu

Posledním panelem v řetězci zpracování signálu v tomto GUI od jeho načtení až po potlačení šumu v něm obsaženém je panel *Waveletové potlačení šumu*. Veškeré funkce tohoto panelu kromě tlačítka *Potlač šum* jsou přístupny již od spuštění GUI i bez toho, aby bylo nutno načíst do systému vstupní signál.

Vybíráme si zde typ waveletu, který následně využijeme pro potlačení šumu, hloubku dekompozice a koeficient, kterým budeme pro signál s_{UV} upravovat práh při prahování waveletových detailních koeficientů.

V rozbalovacím menu *Typ waveletu* máme na výběr ze dvou hlavních rodin waveletů. První z nich jsou wavelety typu Daubechies a druhou jsou wavelety typu Symlet. Obě tyto rodiny mají své podtypy rozlišené čísly, takže můžeme mluvit např. o waveletu typu Daubechies 2 nebo Daubechies 8. Speciálním představitelem waveletu typu Daubechies, konkrétně Daubechies 1, je wavelet typu Haar. Wavelet typu Haar můžeme vidět na Obr. 1.1, několik představitelů waveletů typu Daubechies a Symlet na Obr. 1.2.

Jestliže zvolíme položku *Daubechies* nebo *Symlets*, zviditelní se nám další rozbalovací menu *Řád waveletu*, kde si můžeme vybrat konkrétní typ. Mezi výchozí hodnoty patří čísla od 1 do 10 s tím, že pokud je vybrána položka *další*, rozbalovací menu se zneviditelní a místo něho se objeví editovací rámeček, do kterého vepíšeme číslo požadovaného typu a stiskneme Enter nebo klikneme myší mimo rámeček. Pokud je vloženo číslo korektní, vloží se do nabídky v rozbalovacím menu před položku *další* a rozbalovací menu *Řád waveletu* se opět zviditelní.

Jakmile je výběr typu waveletu kompletní, vpravo dole se daný wavelet vykreslí. Jak tento panel může vypadat vidíme na Obr. 4.7.

Než se dostaneme k zadávání hodnot do editovacích rámečků pod výběrem typu waveletu, je třeba říct, že výstupem konečného potlačování šumu není signál jeden, ale tři. Jsou to signály:

- s_V – Na znělých úsecích je šum v signálu potlačován s hloubkou dekompozice *Hloubka V*, na neznělých úsecích není aplikováno potlačování šumu vůbec.

- s_{UV} – Na znělých úsecích je šum v signálu potlačován s hloubkou dekompozice *Hloubka V*, na neznělých úsecích s hloubkou dekompozice *Hloubka U*. Při potlačování šumu v signálu s_{UV} je navíc brán v úvahu i *Koeficient prahu*, který upraví algoritmem automaticky zvolený práh následujícím způsobem:

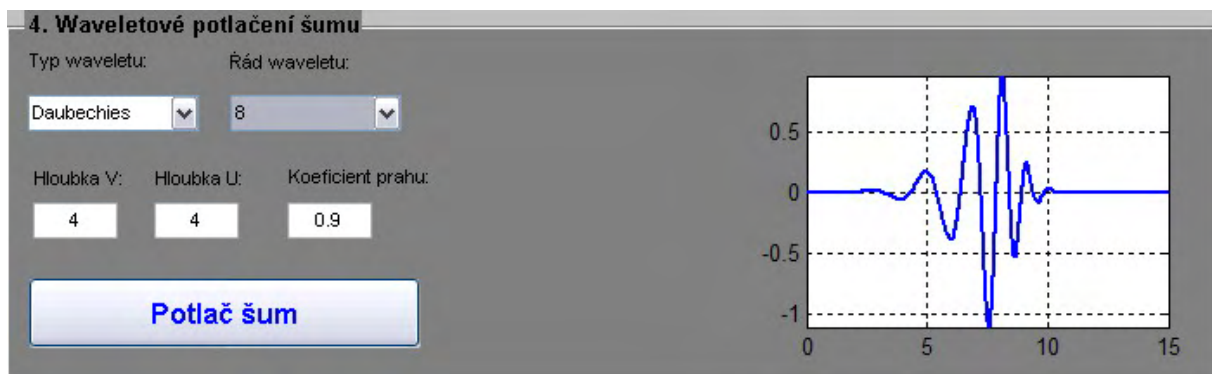
$$thr_{new} = thr_{def} \cdot \min\{1; uv + k/2\} \quad (4.1)$$

kde thr_{def} je práh určený algoritmem, uv je hodnota signálu UV pro daný segment a k je námi zvolený koeficient. GUI povolí koeficient z intervalu $\langle 0; 1 \rangle$.

Když zvolíme $k = 0$, bude se práh řídit přesně podle znělosti segmentů. Ale znamená to, že v tomto případě by se signál s_{UV} velmi podobal signálu s_V , protože by byl úplně zrušen práh pro neznělé segmenty.

Když zvolíme $k > 0,8$, pro znělé segmenty to bude znamenat, že práh nebude nikdy upraven, neboť výraz $\min\{1; uv + k/2\}$ bude v tomto případě stále 1 (hodnota uv pro znělé segmenty totiž nikdy neklesne pod 0,6).

- s_A – Tento signál vznikne potlačením šumové složky na celém svém intervalu bez ohledu na to, zda jde o segment znělý či neznělý. Je použita hloubka dekompozice *Hloubka V*.



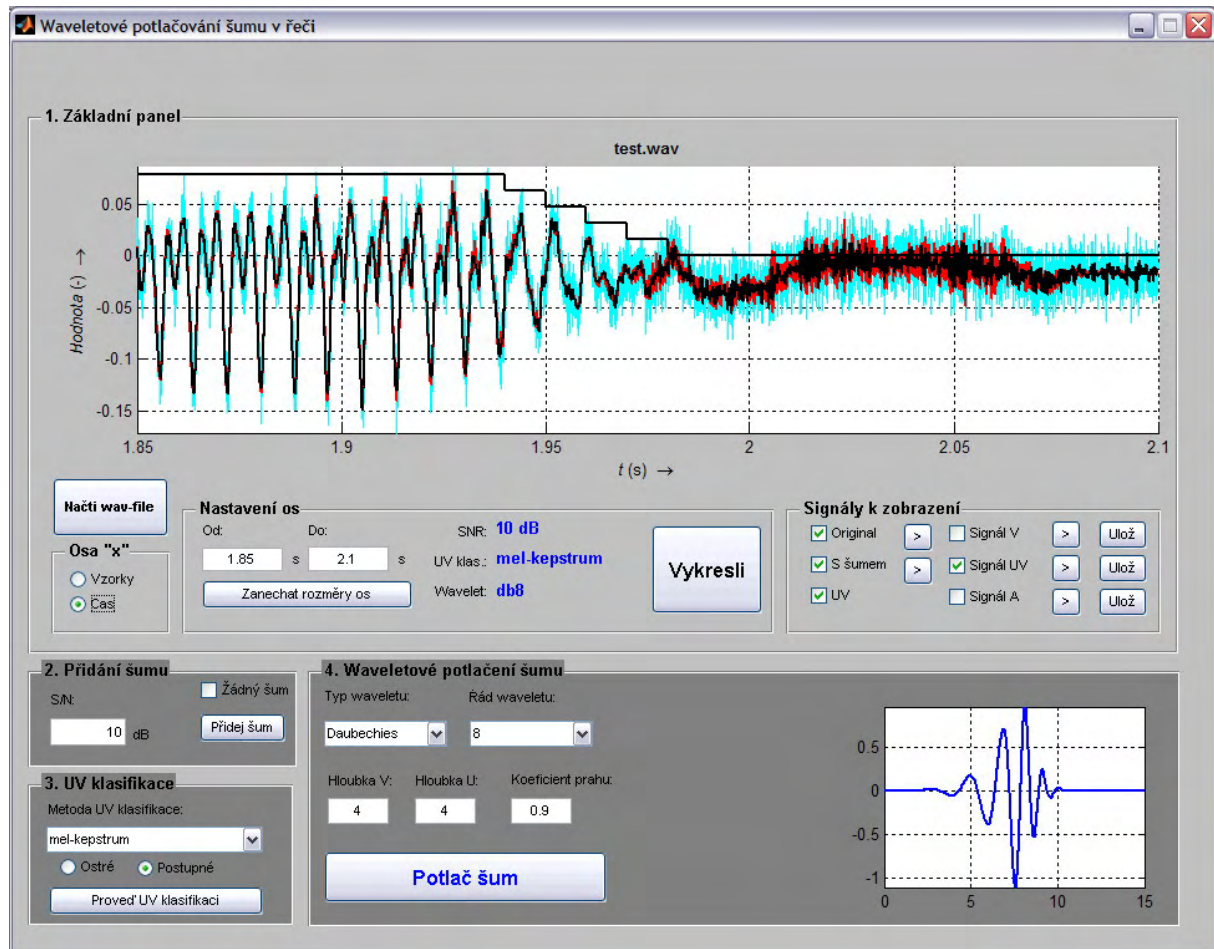
Obr. 4.7: Panel *Waveletové potlačení šumu*

Hloubku dekompozice volíme v editovacích rámečcích *Hloubka V* a *Hloubka U*, koeficient pro prahování signálu s_{UV} v rámečku *Koeficient prahu*.

Po stisknutí tlačítka *Potlač šum* jsou sebrány veškeré nutné parametry pro funkci *odsum* (viz str.30), která je následně spuštěna. Signál je opět po jednotlivých segmentech zpracován a výstupem jsou tři již dříve zmíněné signály s_V , s_{UV} a s_A . Po úspěšném běhu algoritmu pro potlačení šumu jsou zpřístupněny zaškrtačkové políčka u signálů s_V , s_{UV} a s_A v panelu *Signály k zobrazení*, stejně jako tlačítka *Ulož* vpravo od tlačítek na okamžité přehrání signálu. Také je pod graf k položce *Wavelet* dopsán použitý typ waveletu. Nakonec je opět zavolána funkce *vykresleni_axes_UV* (viz str.30).

4.2.5 Obecně k průchodu signálu

K celému postupu zbývá doplnit ještě to, že kdykoliv něco změním v některém z bloků, musíme od daného bloku opět postupovat ve zpracování signálu postupně. Např. pokud chceme změnit míru šumu, musíme provést znovu klasifikaci znělosti a neznělosti a až poté můžeme přistoupit k potlačování šumu.



Obr. 4.8: Celý GUI po provedení všech kroků od načtení signálu až po potlačení v šumu v něm.

Jak může vypadat celý GUI po provedení všech kroků lze vidět na Obr. 4.8.

4.3 Použité matlabovské funkce

V následující kapitole se budu věnovat popisu naprogramovaných funkcí a budu se snažit vysvětlit mechanismus jejich fungování.


wave_gui.m

Tento soubor je stěžejním, neboť ovládá celý GUI. Jakákoli funkce, která je volána např. stisknutím tlačítka nebo výběrem možnosti z rolovacího menu, je implementována v tomto souboru. Kostra souboru wave_gui.m byla vytvořena automaticky při prvním uložení rozpracovaného GUI a postupně jsem přidával další a další funkce. Většinu z nich vysvětlím dále. Specifikem celého systému je proměnná typu struktura handles, ve které jsou uloženy veškeré handles (ukazatele) na objekty GUI. Také jsou do ní ukládány proměnné, které je třeba využívat ve více funkcích, popř. napříč celým programem. Následuje seznam a krátký popis většiny funkcí ze souboru wave_gui.m.

- wave_gui_OpeningFcn – Tato funkce se volá hned po spuštění GUI ještě předtím, než se uživatelské rozhraní objeví. Do handles jsou prealokovány proměnné pro signály a zapnuta mřížka u obou vykreslovacích ploch.

- `pushbutton_nacti_Callback` – Funkce je zavolána při stisknutí tlačítka *Načti wav-file*. Nejprve se otevře dialogové okno pro načtení zvukového signálu. Pokud uživatel vybere soubor špatného typu, nad hlavním grafem se objeví chybová hláška a znepřístupní se veškeré ovládací prvky, které nemohou fungovat bez načteného souboru. Také jsou vymazány všechny proměnné se signály. Pokud uživatel vybere WAV-soubor, je tento načten do proměnné `s0`, jsou zpřístupněny příslušné ovládací prvky a podle nastavení panelu *Osa „x“* je signál vykreslen.
- `uipanel4_SelectionChangeFcn` – Funkce je zavolána při každé změně v panelu *Osa „x“*. Hlavní část kódu v ní obsaženého se vykoná pouze tehdy, pokud je již načten WAV-soubor. Pokud ještě načten není, jsou změněny jen jednotky zobrazované za rámečky *Od* a *Do*. Podmínkami je pak nastaveno, jak se zachovat při různém nastavení panelu *Osa „x“*. Funkce se vykoná pouze tehdy, pokud opravdu došlo ke změně v nastavení v tomto panelu. Pokud uživatel stále kliká na tu samou možnost, nic se nebude dít. V případě, že ke změně v nastavení dojde, jsou přepočteny hodnoty zadané v editovacích rámečcích *Od* a *Do* ze vzorků na sekundy nebo naopak a graf je překreslen s novým cejchováním osy „x“.
- `pushbutton_UV_Callback` – Funkce je volána při zmáčknutí tlačítka *Proveď UV klasifikaci*. Chování funkce je ovlivněno nastavením parametrů v rozbalovacím menu *Metoda UV klasifikace* a vybráním jedné z možností *Ostré* nebo *Postupné*. Veškeré zvolené parametry jsou předány funkci `uv_detect`, která provede samotnou klasifikaci. Výstup je uložen do proměnné `uv`. Následně je provedena režie přístupnosti dalších ovládacích prvků a pokud je již zvolen typ waveletu, který budeme chtít použít, je zpřístupněno i tlačítko *Potlač šum*. Jsou také vymazány všechny tři výstupní signály, které vznikají po spuštění funkce na potlačení šumu. Nakonec je vykreslen graf i s klasifikací znělosti a neznělosti.
- `pushbutton_sum_Callback` – Funkce je volána při zmáčknutí tlačítka *Přidej šum*. Nejdříve je načtena hodnota z editovacího rámečku *S/N* a je ověřena korektnost zadání. Pokud je vše v pořádku, do signálu je přidán bílý Gaussovský šum tak, aby byla splněna velikost zadané hodnoty odstupu signálu od šumu. Signál i s šumem je uložen do proměnné `s_sum`, je provedena režie přístupnosti ovládacích prvků a vymazány výstupní signály a signál s klasifikací znělosti a neznělosti. Nakonec je do grafu vykreslen původní signál a signál s šumem.
- `check_bezsumu_Callback` – Funkce je volána při zaškrtnutí nebo odškrtnutí políčka *Žádný šum*. Když je zaškrtnuto, původní signál `s0` se zkopíruje do proměnné `s_sum`, s kterou se pracuje v následujících blocích. Pokud je odškrtnuto, zpřístupní se editovací rámeček *S/N* a tlačítko *Přidej šum*. Podle stavu zaškrtačovacího políčka *Žádný šum* je také řízena režie ostatních ovládacích prvků. Nakonec jsou opět zvolené signály vykresleny do grafu.
- `menu_wavelet_Callback` – Funkce je volána při výběru položky v rozbalovacím menu *Typ waveletu*. Funkce se chová podle toho, jakou možnost v rozbalovacím menu zvolíme. Buď zvolíme možnost *Haar*, která je pro úplné určení typu waveletu postačující, nebo některou z možností *Daubechies* či *Symlets*, které potřebují ještě dodefinovat přesný typ v dalším rozbalovacím menu, které se zviditelní. Jestliže je typ waveletu nadefinován kompletně, wavelet je pomocí funkce `vykresli_wavelet` vykreslen (v tomto případě pouze možnost *Haar*). Pokud již jsou také se signálem provedeny i všechny předchozí nutné kroky, je zpřístupněno tlačítko *Potlač šum*.
- `menu_typ_Callback` – Funkce je volána při výběru položky v rozbalovacím menu *Řád waveletu*. Zvolené číslo je přidáno k identifikátoru hlavní rodiny požadovaného waveletu a wavelet je pomocí funkce `vykresli_wavelet` vykreslen. Pokud již jsou

se signálem provedeny i všechny předchozí nutné kroky, je zpřístupněno tlačítko *Potlač šum*. Pokud je v menu zvolena možnost *dalsi*, je toto menu zneviditelněno a místo něho se objeví editovací rámeček, do kterého můžeme zadat vyšší typ waveletu, než je standardně nabízen.

- `edit_typ_Callback` – Funkce je volána po stisknutí klávesy Enter po vepsání hodnoty do editovacího rámečku *Řád waveletu* nebo kliknutí vedle rámečku. Po zadání se rámeček zneviditelní a místo něj se objeví původní rozbalovací menu s aktivní námi vloženou novou hodnotou na předposledním místě v seznamu. Až do zavření GUI budeme mít tuto novou hodnotu přístupnou spolu s původními deseti čísly.
- `pushbutton_odsum_Callback` – Funkce je volána po stisknutí tlačítka *Potlač šum*. Nejprve je ověřena regulérnost zadání hodnot *Hloubka V (N1)*, *Hloubka U (N2)* a *Koeficient prahu (koef)*. V případě jakéhokoli chybného zadání je nad grafem zobrazena chybová hláška a běh funkce je ukončen. Poté je zavolána funkce `odsum`, která řídí proces odšumování. Po potlačení šumu je opět provedena režie ovládacích prvků a vykreslen graf.
- `pushbutton_s0_Callback`, `pushbutton_s_sum_Callback`,
`pushbutton_s_V_Callback`, `pushbutton_s_UV_Callback`,
`pushbutton_s_A_Callback` – Tyto funkce se spouští po stisknutí odpovídajícího tlačítka  a slouží k okamžitému zvukovému přehrání daného signálu.
- `pushbutton_uloz_s_V_Callback`, `pushbutton_uloz_s_UV_Callback`,
`pushbutton_uloz_s_A_Callback` – Tyto funkce se spouští po stisknutí odpovídajícího tlačítka *Ulož* a slouží k uložení daného výstupního do WAV-souboru. Jméno WAV souboru je generováno tak, aby obsahovalo parametry, které ovlivnily konečnou podobu daného signálu.
- `vykresli_wavelet` – Funkce získá prostřednictvím matlabovské funkce `wavefun` data pro vykreslení waveletu, který následně vykreslí do souřadného systému os nacházejícího se v GUI vpravo dole.
- `odsum` (včetně `odsum_signal`) a `vykresleni_axes_UV` – viz níže.

uv_detect.m

Pokud je v rozbalovacím menu *Metoda UV klasifikace* zvolena možnost *prahovani*, je provedena klasifikace znělosti a neznělosti segmentů statistickou metodou (viz. kap. 2.1). Signál je procházen po úsecích dlouhých 10 ms, vždy je ale zpracováván v oknech o velikosti 30 ms. V každém běhu hlavní smyčky je určena dolní (a) a horní (b) hranice zpracovávaného segmentu. Jelikož se lehce může stát, že celková délka signálu není přesným násobkem délky jednoho 30ms úseku, je nutno implementovat ošetření přestřelení horní hranice. Do proměnné `pos` jsou ukládány indexy středu každého segmentu v celkovém signálu. Pro každý segment je potom spuštěna vnitřní smyčka, pomocí které je vypočítána energie E , počet průchodů signálu nulou Z a autokorelační koeficient R . Všechny tyto tři hodnoty jsou ukládány do vektorů, aby potom mohlo být rozhodnuto o znělosti nebo neznělosti příslušných segmentů. Jakmile jsou tyto tři hodnoty spočteny pro segmenty napříč celým signálem, je přistoupeno k rozhodování o znělosti či neznělosti. Pravidla pro rozhodování byla definována v kap. 2.1. Na základě mých empirických zkušeností je ale ve funkci `uv_detect` nastaven limit pro energii segmentu k posouzení jeho znělosti místo 20 % na 16 % z průměrné energie signálu.

Pokud je v rozbalovacím menu *Metoda UV klasifikace* zvolena možnost *mel-kepstrum*, je provedena klasifikace znělosti a neznělosti segmentů metodou využívající mel-kepstrální

koeficienty (viz kap. 2.2). Algoritmus pro výpočet mel-kepstrálních koeficientů je implementován v samostatné funkci `kepstrum_det` (popsána níže). Jakmile máme mel-kepstrální koeficienty, můžeme opět stejným způsobem jako v případě statistické metody určit, zda budeme považovat segment za znělý nebo neznělý. Porovnáváme průměrnou hodnotu prvního mel-kepstrálního koeficientu v celém signálu s hodnotou mel-kepstrálního koeficientu v daném segmentu. Pokud je tato hodnota vyšší než 80 % z průměrné hodnoty prvního mel-kepstrálního koeficientu, je segment označen jako znělý.

kepstrum_det.m

Jak již bylo napsáno výše, v této funkci je prakticky implementován postup popsáný v kap. 2.2., resp. v kapitolách 2.2.1 a 2.2.2. Nejprve jsou opět deklarovány vstupní parametry, jako jsou délky oken, pro které se bude signál zpracovávat, posuv okna pro zpracování dalšího segmentu, počet filtrů v mel-bance a mezní kmitočty mel-banky. Signál je poté upraven na délku rovnou celému násobku délky okna. Signál je nasegmentován a jednotlivé rámce jsou všechny uloženy do proměnné `matice_ramcu`, s kterou je poté pracováno. Jsou vypočteny spektrální charakteristiky jednotlivých rámců a pomocí funkce `fft2melfilt` je vytvořena mel-banka filtrů. Spektrální charakteristiky jednotlivých rámců jsou poté filtrovány mel-bankou filtrů a následně vypočteny mel-kepstrální koeficienty.

odsum a odsum_signal

Funkce `odsum` je hlavní funkcí provádějící samotné odšumování signálu. Vstupní i výstupní proměnnou je globální proměnná `handles`. Nejdříve jsou z `handles` načteny všechny důležité proměnné. Poté je spuštěna hlavní smyčka této funkce. Signál je procházen po úsecích dlouhých 10 ms, vždy je ale zpracováván v oknech dlouhých 30 ms. Je tak zprostředkován vzájemný překryv zpracovávaných segmentů a zajištěna lepší návaznost a propojenost za sebou jdoucích úseků řečového signálu.

V každém běhu hlavní smyčky je určena dolní (`a`) a horní (`b`) hranice zpracovávaného segmentu. Jelikož se lehce může stát, že celková délka signálu není přesným násobkem délky jednoho 30ms úseku, je nutno implementovat ošetření přestřelení horní hranice. Následně jsou vypočítány i dolní (`a1`) a horní (`b1`) hranice středního úseku jednoho segmentu, tj. prostředních 10 ms z 30 ms dlouhé části. Do proměnné `pos` jsou ukládány indexy středu každého segmentu v celkovém signálu.

Jelikož se prvních 10 ms a posledních 10 ms celého signálu nikdy nedostane do intervalu určeného proměnnými `a1` a `b1`, je nutno tyto dva úseky zpracovat zvlášť. V těchto dvou konkrétních případech není aplikováno vzájemné překrytí oken.

Následuje spuštění samotného algoritmu pro potlačení šumu implementovaného ve funkci `odsum_signal`. Tato funkce je spouštěna pro každý typ výstupního signálu, který je definován (viz kap. 4.2.4) s rozdílnými parametry.

Funkce `odsum_signal` nejprve spočítá pomocí matlabovské funkce `wavedec` aproximační a detailní koeficienty a uloží je do proměnné `c`. Proměnná `L` potom definuje, která část proměnné `c` představuje aproximační koeficienty úrovně `N` a které části představují detailní koeficienty úrovně 1 až `N`. Matlabovská funkce `ddencmp` poté automaticky spočítá velikost (`thr`) a typ prahování (`sorh`) – možnosti jsou buď měkké ('s') nebo tvrdé ('h'). Práh je následně upraven zadanými parametry a waveletové koeficienty jsou postoupeny funkci `wdencmp`, která provádí odšumování. Výsledný signál `clean` je potom segmentem s potlačeným šumem.

vykreslení_axes_UV

Pravděpodobně nejčastěji volaná funkce. Na základě parametrů zvolených v panelech *Osa „x“*, *Nastavení os* a *Signály k zobrazení* vykresluje graf.

Funkce nejprve načte spodní hranici *a* a vrchní hranici *b* a zkontroluje, jestli jsou zadána čísla. Pokud je v editovacím rámečku *Do* napsán řetězec *max*, je za horní hranici považován konec signálu. Taktéž pokud je tlačítko *Zanechat rozměry os* zamáčknuto do polohy „dole“, je aktuální rozpětí os uloženo, aby mohlo být později nastaveno.

Další část kódu upravuje číselné zadání rozměrů os. Pokud se stane, že je jako dolní hranice uvedeno číslo menší než jedna v případě cejchování ve vzorcích, resp. číslo menší než nula v případě časové osy, je tato hranice upravena na minimální hodnotu, tedy jedna, resp. nula. Stejně je ošetřena i situace, kdy je jako vrchní hranice uvedeno číslo větší než délka signálu. V takovém případě je vrchní hranice nastavena právě na konec signálu. Když uživatel zadá hodnotu spodní hranice větší než hodnotu vrchní hranice, program nad grafem zobrazí chybovou hlášku.

Jakmile jsou vyřešeny tyto „formality“, je vykreslen každý signál, u kterého je v panelu *Signály k zobrazení* zaškrtnuto příslušné tlačítko. Výjimku tvoří signál UV, který je vykreslován až po nastavení os. Po vykreslení jsou podle polohy tlačítka *Zanechat rozměry os* nastaveny osy.

Poslední částí je vykreslení klasifikace znělosti a neznělosti, pokud je zaškrtnuto příslušné políčko. Nula v signálu UV se zobrazuje vždy jako nula, jednička se zobrazuje tak, aby hladina signálu UV byla srovnatelná s velikostí ostatních signálů v zobrazovaném úseku. Pokud by se mělo stát, že rozpětí osy „y“ je takové, že se tam signál UV nemůže řádně zobrazit, postupným procesem úpravy rozpětí osy „y“ je tato situace napravena.

5 Testování

V této kapitole budeme testovat funkčnost vytvořeného rozhraní a funkcí s ním spojených. Budeme sledovat, jak signál prochází systémem v závislosti na nejrůznějším nastavení parametrů. Jelikož nemůžeme pokrýt všechny možnosti, budeme se tedy snažit vystihnout ty nejpodstatnější a najít jakési optimální nastavení parametrů pro získání co nejkvalitnějšího výsledku.

Výstupy budeme hodnotit jak objektivně, tak subjektivně (podle poslechu). V hodnocení úspěšnosti a korektnosti rozeznávání znělosti a neznělosti bude převládat objektivní hodnocení, subjektivní hodnocení potom u posuzování kvality potlačování šumu, neboť v aplikacích na potlačování šumu záleží především na něm.

Jelikož různé přístupy ke zpracování vstupního signálu se začínají uplatňovat až při klasifikaci znělosti a následně při potlačování šumu, nejvíce pozornosti bude věnováno právě těmto dvěma částem.

Testování provedeme na mnou nahrané větě „Mohli jsme si s tím pohnout.“ Tato věta byla zvolena z toho důvodu, že obsahuje hlásky „s“ a v případě části věty „jsme si s tím“ se hodně střídají znělé a neznělé úseky. Jelikož tato diplomová práce je založena na potlačování šumu v řeči s rozdílným přístupem ke znělým a neznělým segmentům, je toto relativně rychlé střídání znělých a neznělých segmentů žádoucí.

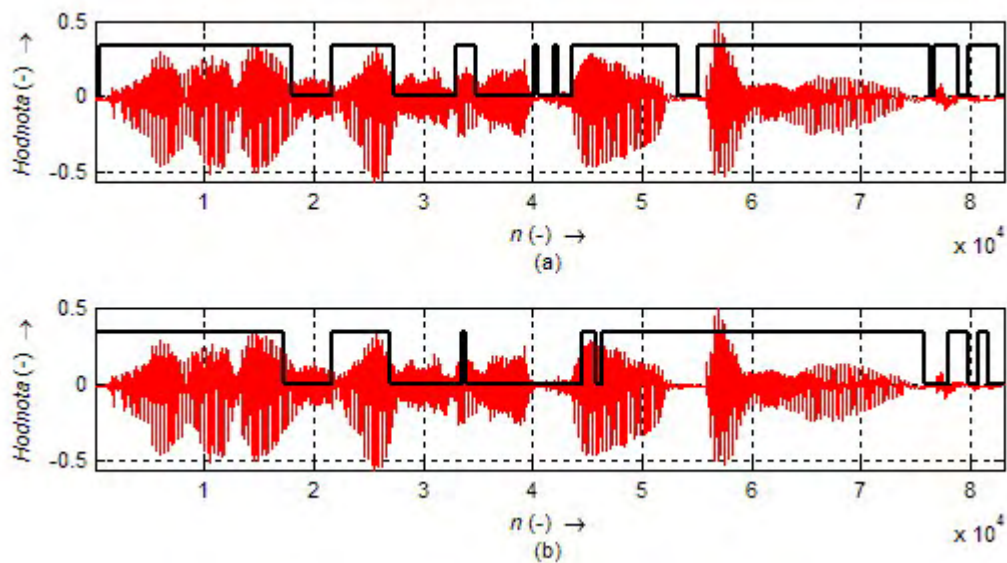
Budou vytvořeny signály s aditivním bílým šumem s odstupem signálu od šumu (SNR) o hodnotách 0 dB, 5 dB, 10 dB a 15 dB.

Různé přístupy k analýze znělosti a neznělosti budou testovány na signálu bez šumu a na signálu s šumem o $SNR = 0$ dB. Metoda, která se ukáže být jako nejlepší, bude použita v testování různých parametrů nastavovaných při potlačování šumu.

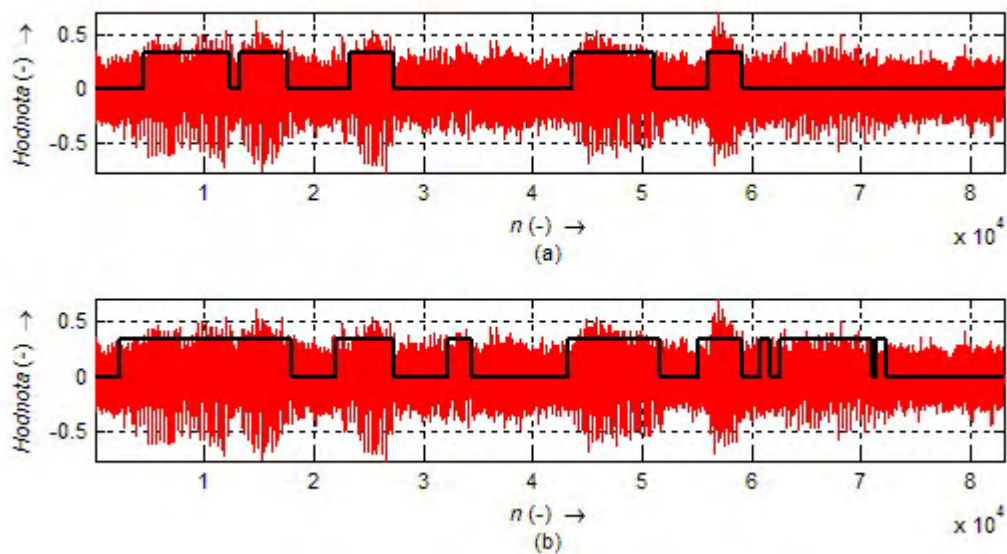
5.1 Analýza znělosti a neznělosti segmentů

Každá ze dvou implementovaných metod, tedy statistická metoda a metoda mel-kepstrálních koeficientů, dá trochu jiné výsledky. Provedeme analýzu znělosti a neznělosti segmentů na signálu bez šumu (viz Obr. 5.1) a na signálu s šumem o $SNR = 0$ dB (viz Obr. 5.2). Vyšší hladina černé křivky značí znělost, nižší neznělost. Jak lze z grafů vyčíst, při analýze signálu bez šumu (Obr. 5.1) mají obě metody větší problémy s určitými segmenty, kde je chybně určí jako znělé nebo neznělé, než při analýze signálu i s přidaným šumem. Je to zřejmě z toho důvodu, že signál bez šumu je velmi čistý a klíčové parametry, podle kterých se rozhoduje, zda je segment znělý nebo ne, jsou velmi blízko prahovým hodnotám. Často tak dojde k chybnému určení znělosti či neznělosti. Naproti tomu při klasifikaci znělosti a neznělosti segmentů na signálu s šumem (Obr. 5.2) dosahujeme mnohem stabilnějších výsledků. Jelikož přibližně víme, kde má být signál určen jako znělý a kde jako neznělý, můžeme porovnat a zhodnotit dosažené výsledky. Podívejme se, kde každá z metod vyhodnotila znělost úseku signálu s šumem jinak.

- Přibližně mezi vzorky 12000 a 13000. Tento úsek odpovídá pravděpodobně hlásce „h“ ve slově „mohli“ a měl by být klasifikován jako **znělý**.
- Přibližně mezi vzorky 32000 a 35000. Tento úsek odpovídá hlásce „i“ ve frázi „jsme si s tím“ a měl by být klasifikován jako **znělý**.
- Přibližně mezi vzorky 60000 a 73000. Tento úsek odpovídá pravděpodobně hláskám „hnou“ ve slově „pohnout“ a měl by být klasifikován jako **znělý**.



Obr. 5.1: Klasifikace znělosti a neznělosti na signálu bez šumu. Statistická metoda (a) a metoda mel-kepstrálních koeficientů (b).



Obr. 5.2: Klasifikace znělosti a neznělosti na signálu s šumem. Statistická metoda (a) a metoda mel-kepstrálních koeficientů (b).

Našli jsme tři rozdíly v klasifikaci signálu. Ve všech případech „měla pravdu“ metoda mel-kepstrálních koeficientů, můžeme tedy prohlásit, že pro klasifikaci znělosti a neznělosti řečových segmentů se tato metoda hodí více.

5.2 Waveletové potlačování šumu

Potlačování šumu budeme testovat postupně na signálech s šumem o velikosti SNR 0 dB, 5 dB, 10 dB a 15 dB. Budeme používat mel-kepstrální metodu klasifikace znělosti a neznělosti (jak metodu „ostrou“, tak „postupnou“). Nejvíce budeme testovat vliv různého

nastavení parametrů hloubek dekompozice *Hloubka V*, *Hloubka U* a koeficientu pro úpravu prahu *Koeficient prahu*.

Do editovacího rámečku *Koeficient prahu* můžeme zadat jen reálné číslo z intervalu $\langle 0 ; 1 \rangle$. Výsledný koeficient úpravy programem automaticky určeného prahu záleží na velikosti námi zadaného *Koeficientu prahu* a na hodnotě signálu UV pro daný segment. Výsledný práh pro funkci `wdencomp` získáme z automaticky určeného prahu podle vztahu (5.1). Koeficienty, kterými je násoben automaticky určený práh, lze najít v **Tab. 5.1**. V řádcích je hodnota zadávaná do editovacího rámečku *Koeficient prahu* a ve sloupcích je hodnota signálu UV pro daný segment. Pokud jsme zvolili *Ostrou* klasifikaci, je pro nás podstatný jen první a poslední sloupec tabulky. V případě *Postupné* klasifikace využijeme sloupce všechny.

Tab. 5.1: Tabulka výsledného koeficientu, kterým je násoben algoritmem automaticky určený koeficient prahu.

<i>uv</i> <i>k</i>	Neznělé segmenty			Znělé segmenty		
	0,00	0,20	0,40	0,60	0,80	1,00
0,00	0,00	0,20	0,40	0,60	0,80	1,00
0,10	0,05	0,25	0,45	0,65	0,85	1,00
0,20	0,10	0,30	0,50	0,70	0,90	1,00
0,30	0,15	0,35	0,55	0,75	0,95	1,00
0,40	0,20	0,40	0,60	0,80	1,00	1,00
0,50	0,25	0,45	0,65	0,85	1,00	1,00
0,60	0,30	0,50	0,70	0,90	1,00	1,00
0,70	0,35	0,55	0,75	0,95	1,00	1,00
0,80	0,40	0,60	0,80	1,00	1,00	1,00
0,90	0,45	0,65	0,85	1,00	1,00	1,00
1,00	0,50	0,70	0,90	1,00	1,00	1,00

Testování 1

Pevné nastavení:

- *Signál* – *s_UV*
- *Metoda UV klasifikace* – *mel-kepstrum*
- *Typ waveletu* – *Daubechies 8*
- *Hloubka V* (*N1*) = 5
- *Hloubka U* (*N2*) = 5

Měnicí se parametry:

- *SNR* = 0 dB / 5 dB / 10 dB / 15 dB
- *Ostrost UV klasifikace* – *Ostrá* / *Postupná*
- *Koeficient prahu* (*koef*) = 0,5 / 0,8 / 1,0

Vliv *Ostrosti UV klasifikace* se na výstupním signálu poslechově projeví tím více, čím je *SNR* vyšší. Ale ani pro *SNR* = 15 dB není rozdíl v kvalitě poslechu nijak velký. Proto můžeme říct, že *Postupná UV klasifikace* nám oproti *Ostré UV klasifikaci* zajistí více méně jen kosmetickou úpravu výstupního signálu v místech změny indikované znělosti a neznělosti, přičemž poslechový vjem se změní jen minimálně nebo vůbec ne.

Vliv *Koeficientu prahu* již je samozřejmě větší. Ve znělých úsecích je *Koeficient prahu* v použité funkci nastaven napevno na 0,65, tudíž se ve většině signálu neprojeví.

Jedinou výjimkou je situace v případě použití *Postupné UV* klasifikace, kdy je hodnota UV signálu pro daný segment 0,6. V tom případě je automaticky určená hodnota prahování waveletových koeficientů vynásobena koeficientem 0,925. Z tabulky **Tab. 5.1** je také patrné, že v neznělých úsecích se nastavení *Koeficientu prahu* projeví vždy. U signálů s velkou hodnotou SNR (15 dB) je lepší použít menší hodnotu *Koeficientu prahu*, neboť šum není tak silný a menší hodnota $koef$ znamená lepší zachování charakteru neznělých částí řeči. Pokud je ale hodnota SNR velmi malá (0 nebo 5 dB), je lepší použít vyšší hodnotu *Koeficientu prahu* a obětovat do určité míry srozumitelnost. Pokud zvolíme hodnotu $koef$ vysokou, odfiltrujeme i část neznělého úseku, kterou bychom si ideálně přáli zachovat. Zvolíme-li hodnotu $koef$ nízkou, zůstane nám v signálu velká porce šumu a neznělé, málo vyfiltrované úseky budou v řeči v porovnání s odfiltrovanými znělými úseky hodně intenzivní. Pro lepší poslechový vjem je lepší obětovat trochu té srozumitelnosti, než nechat v signálu takové prudké „výbuchy“ šumu.

Největší vliv na srozumitelnost výstupního signálu má hodnota SNR . Čím je menší, tím větší šum se v signálu nachází a tím je „těžší“ ho odfiltrovat. Pro signály se slabým šumem (SNR 10 nebo 15 dB) fungují naprogramované funkce relativně dobře, v pozadí je nicméně stále znatelný zbytkový šum v neznělých částech. U signálů se silným šumem (SNR 0 nebo 5 dB) už je to horší, neboť šum již degraduje charakter neznělých úseků velmi podstatně a algoritmus tedy nemá moc šancí vybrat ze dvou signálů, z nichž oba mají charakter šumu (např. hláska „s“ v neznělém segmentu a samotný šumový signál), právě ten, který má v signálu zůstat. Musíme poté volit kompromis mezi srozumitelností neznělých úseků a dostatečným potlačením šumu.

Testování 2

Pevné nastavení:

- *Signál* – s_{UV}
- *Metoda UV klasifikace* – *mel-kepstrum*
- $SNR = 5$ dB
- *Ostrost UV klasifikace* – *Postupná*
- *Typ waveletu* – *Daubechies 8*
- *Hloubka $V(N1)$* = 5
- *Koeficient prahu* ($koef$) = 0,8

Měnicí se parametry:

- *Hloubka $U(N2)$* = 3 / 4 / 5 / 6

Vliv změny hloubky dekompozice při potlačování šumu v neznělých úsecích řečového signálu je patrný (v tomto odstavci bude myšleno hloubkou dekompozice hloubka dekompozice právě v neznělých úsecích). V případě hloubky dekompozice o hodnotě 3 je v signálu ještě patrná větší dávka šumu, mezi případy s $N2 = 5$ a $N2 = 6$ není zase tak podstatný rozdíl, subjektivně ale lépe zní signál v případě $N2 = 6$.

Testování 3

Pevné nastavení:

- *Signál* – s_{UV}
- *Metoda UV klasifikace* – *mel-kepstrum*
- *SNR* = 5 dB
- *Ostrost UV klasifikace* – *Postupná*
- *Hloubka V* (N_1) = 5
- *Hloubka U* (N_2) = 6
- *Koeficient prahu* ($koef$) = 0,8

Měňící se parametry:

- *Typ waveletu* – *Haar / Daubechies 8 / Symlets 8 / Daubechies 12*

Při potlačování šumu za pomoci waveletu typu Haar má výsledný řečový signál jakýsi „kovový“ a „robotický“ charakter, jako kdyby vycházel z hodně starého speakeru. Připomíná promluvu robotů (počítačů) ve starých filmech. Použití waveletu typu Daubechies 8 dává subjektivně naprosto shodné výsledky jako při použití waveletu typu Symlets 8. Není tedy podstatné, který z těchto dvou je použit. Použití waveletu typu Daubechies 16 dává opět výsledek subjektivně na poslech totožný s předchozím případem.

Testování 4

Pevné nastavení

- *Metoda UV klasifikace* – *mel-kepstrum*
- *SNR* = 5 dB
- *Ostrost UV klasifikace* – *Postupná*
- *Typ waveletu* – *Daubechies 8*
- *Hloubka V* (N_1) = 5
- *Hloubka U* (N_2) = 6
- *Koeficient prahu* ($koef$) = 0,8

Měňící se parametry:

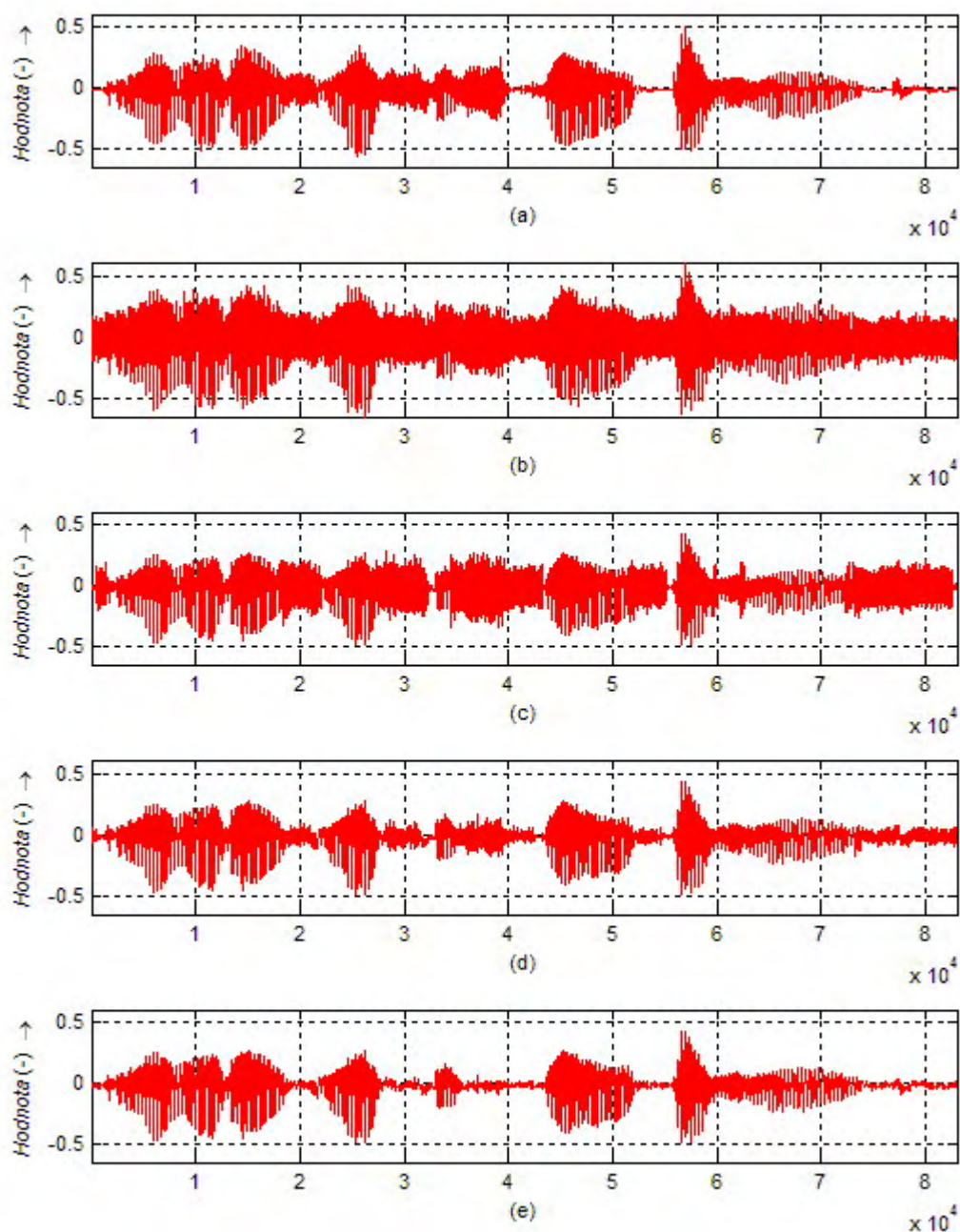
- *Signál* – s_V / s_{UV} / s_A

Rozdíl mezi signály s_V , s_{UV} a s_A je popsán v kap. 4.2.4. Krátce řečeno, signál s_A vznikne potlačením šumu se stejnou intenzitou v celém signálu bez ohledu na znělost a neznělost jednotlivých segmentů. Vznikne tak signál, který je sice v celé své délce oproti signálu s_{UV} velmi čistý (jen sem tam se objeví velmi slabé „zabublání“ vzniklé nedokonalým potlačením všech detailních koeficientů), ale zcela chybí např. již zmíněné „s“. Místo něho je prakticky tiché místo, takže takto vzniklá řeč zní dost směšně. Signál s_V vznikne tak, že neznělé segmenty nejsou upraveny vůbec. Tento signál je shodný se signálem s_{UV} , když zvolíme *Ostrý přístup* k UV klasifikaci a za *Koeficient prahu* $koef$ zvolíme nulu. Vznikne tak signál, který se moc nedá poslouchat, jelikož neznělé segmenty nejsou nijak upraveny a řečový signál je pak značně „štěkávkový“. Když zvážíme, jestli budeme chtít poslouchat řeč, kde budou sykavky úplně chybět, nebo řeč, kde sykavky budou patrné, ale signál bude v těchto neznělých segmentech více zašuměn, volba je pro většinu lidí jasná. Radši trochu více šumu a ponechat v signálu neznělé segmenty, než je úplně smazat.

5.3 Zhodnocení testování

Z testování vyplývá, že pro všechny signály je nejlepší použít metodu, z které vzejde signál s_{UV} . Pro signály s velkou hodnotou SNR je výhodnější použít „razantnější“ metody potlačování šumu v signálu, v našem případě to znamená použít větší hodnotu pro *Koeficient prahu*, který upravuje konečnou velikost samotného prahu. V neznělých segmentech tak dosáhneme kompromisu mezi srozumitelností řeči a potlačením šumu. Na signály s malou hodnotou SNR můžeme použít méně intenzivní metody potlačování šumu v neznělých segmentech, tedy menší *Koeficient prahu* *koef.* S menším *koef.* dosáhneme větší srozumitelnosti v neznělých segmentech a jelikož je SNR malé, zbylý šum nebude subjektivní vjem řeči tolik rušit. Na obrázku **Obr. 5.3** můžeme vidět příklady jednotlivých signálů, se kterými se zde pracovalo.

Abychom ověřili získané výsledky, nechal jsem stejnou větu namluvit dalším mluvčím, konkrétně mým bratrem. Větu řekl podstatně rychleji a s typickým ostravským přízvukem (krátce), takže ani jedna z metod klasifikace znělosti a neznělosti neklasifikovala ani jeden segment z části věty „jsme **si s tím** pohnout“ jako znělý. Obě samohlásky „i“ tedy byly klasifikovány jako neznělé (resp. první je klasifikována jako znělá při malé hodnotě SNR – o tomto jevu už jsem psal dříve při vysvětlení, proč dává UV klasifikace horší výsledky na signálu bez šumu). Promluva je celkově rychlejší a tak dává algoritmus o něco horší výsledky než v případě mnou namluvené věty, která byla vyslovena pomalu a zřetelně s ohledem na správnou artikulaci.



Obr. 5.3: Původní signál bez šumu s_0 (a), signál s šumem s_{sum} (b), signál s_V (c), signál s_{UV} (d), signál s_A (e).

Závěr

Diplomová práce se zabývala potlačováním šumu v řeči založené na waveletové transformaci a rozeznávání znělosti segmentů. V kapitolách 1, 2 a 3 byl uveden teoretický úvod a základ pro praktickou část v kapitole 4, tvořící spolu s kapitolou 5 stěžejní část diplomové práce.

V kapitole 4 bylo navrženo blokové schéma systému pro potlačování šumu v řeči. Byl uveden hlavní problém spojený s potlačováním šumu v řečovém signálu, a to degradace neznělých částí spojená s potlačováním šumu. Hlávky jako „s“, „š“, „c“, „č“, „z“ nebo „ž“ jsou charakteristické tím, že jejich spektrální vlastnosti připomínají do značné míry šum samotný. Při aplikaci standardního algoritmu na waveletové potlačení šumu jsou tedy potlačeny spolu s šumem. Bylo navrženo řešení tohoto problému, které spočívá v zmírnění síly waveletového potlačování šumu právě v neznělých úsecích řečového signálu.

V kapitole 4 je dále představeno grafické uživatelské rozhraní (GUI) naprogramované ve vývojovém prostředí MATLAB. GUI je spolu se všemi svými ovládacími prvky podrobně popsán a je vysvětleno, za co je která část zodpovědná. Poslední podkapitolu tvoří popis vytvořených matlabovských funkcí, aby bylo pro čtenáře snazší se v programových kódech orientovat.

Kapitola 5 se věnuje testování naprogramovaných funkcí a algoritmů. Testování probíhalo ve vytvořeném GUI, které umožňuje pohodlně vyzkoušet veškeré myslitelné kombinace nastavení určených parametrů.

Pro signály s vyšší hodnotou odstupu signálu od šumu fungují naprogramované funkce velmi uspokojivě, jen občas se ve výstupním signálu objeví určité „bublání“ způsobené nedostatečnou dokonalostí použitého algoritmu pro potlačování šumu. Znělé segmenty jsou zbaveny šumu zcela dokonale, ale je znát určitá degradace hlasu mluvčího. K neznělým segmentům je přistupováno jinak, parametrem nejvíce ovlivňující konečnou srozumitelnost řečového projevu se ukázal být koeficient, který přímo ovlivňuje velikost prahu pro prahování waveletových koeficientů. Adjustováním tohoto koeficientu je možno řídit míru intenzity potlačování šumu spolu s hláskou šumového charakteru a zachovat tak jistou míru srozumitelnosti.

Není překvapivé, že výstupy jsou mnohem kvalitnější v případech, kdy je signál degradován šumem v menší míře. Zejména se to týká opět neznělých úseků. Chceme-li totiž zachovat určitou míru z charakteru neznělého fonému, nutně musíme systémem propustit i šum. A tento šum, který systémem propustíme do výstupu, je tím výraznější, čím menší byla hodnota SNR při vkládání šumu do původního signálu. U signálů, které jsou šumem degradovány ve velké míře, dochází tedy k tomu, že po průchodu systémem jsou v neznělých částech spolu s původním signálem patrné i ostré projevy vloženého šumu.

Co se týká přímo klasifikace znělosti a neznělosti fonémů, ve velké míře záleží na čistotě původního projevu mluvčího. Pokud mluvčí mluví srozumitelně, řádně artikuluje, nevynechává hlávky a nepolyká konce slov, mají obě metody velmi vysokou úspěšnost. Jako příklad nedodržení správné artikulace při hovoření českým jazykem může být uveden např. mluva lidí pocházejících nebo žijících v ostravském regionu. Tamější mluva je charakteristická velmi krátkými samohláskami a rychlou kadencí promluvy. Metody na zjišťování znělosti nebo neznělosti tedy někdy selhávají na skutečnosti, že znělé části mezi dvěma neznělými segmenty (např. samohlávka „i“ ve větě „Co sis to udělal?“) nejsou kvůli jejich velmi krátkému trvání a menší energii klasifikovány správně.

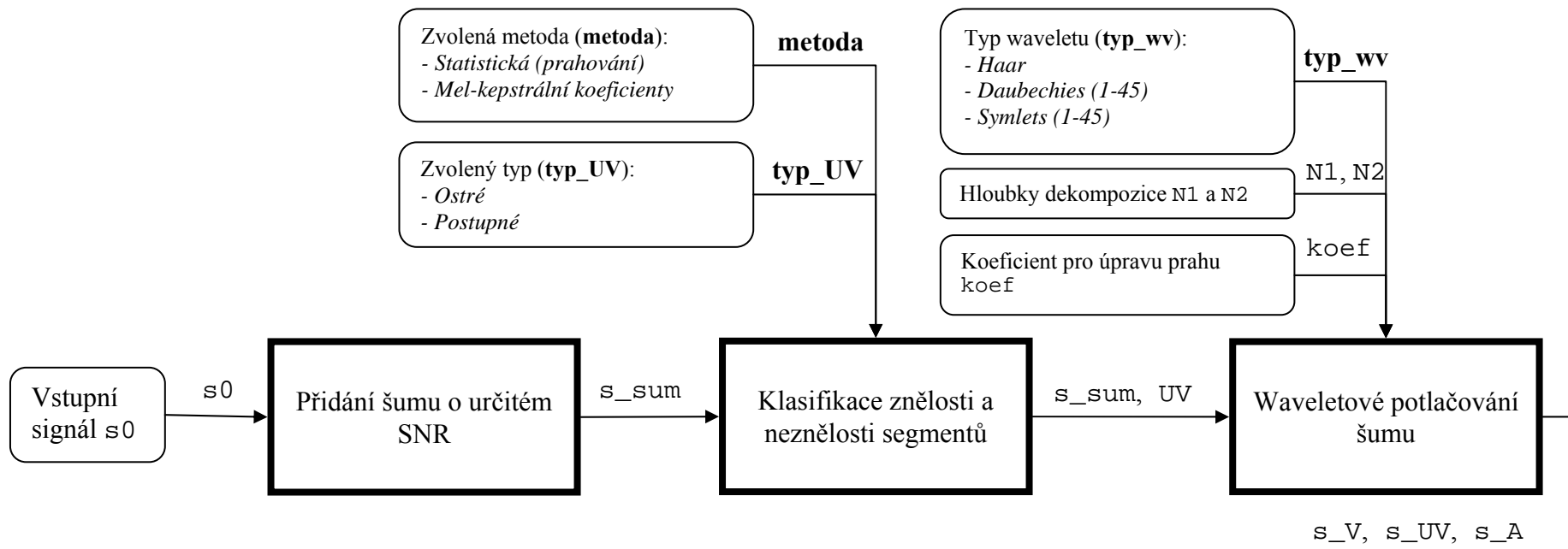
Použitá literatura

- [1] DAVIS, G.M. *Noise reduction in speech applications*, Davis G.M. editor, CRC Press LLC, 2002. ISBN 0-8493-0949-2.
- [2] DELLER, J. R., HANSEN, J. H. L., PROAKIS, J. G. *Discrete-Time Processing of Speech Signals*. Reprint edition. New York: IEEE, 2000. 936 s. ISBN 0-7803-5386-2.
- [3] CHRÁPEK, T. *Základní úlohy DSP v Matlabu a Simulinku*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2006. 41 s. Bakalářská práce.
- [4] JAN, Jiří. *Číslicová filtrace, analýza a restaurace signálů*. 2. upr. a rozš. vyd. Brno : VUTUM VUT Brno, 2002. 427 s. ISBN 80-214-2911-9.
- [5] LEE, J. K., YOO, Ch. D. [N569] Wavelet speech enhancement based on voiced/unvoiced decision. In *The 32nd International Congress and Exposition on Noise Control Engineering*. [s.l.] : [s.n.], 2003. s. 4149-4156. Dostupný z WWW: <<http://fft1.kaist.ac.kr/paperdata/Wavelet%20Speech%20Enhancement%20Based%20on%20VoiceUnvoiced%20Decision.pdf>>.
- [6] RAO, R. M., BOPARDIKAR, A. S. *Wavelet Transforms : Introduction to Theory and Applications*. 1st edition. [s.l.] : ADDISON-WESLEY, 1998. 291 s. ISBN 0-201-63463-5.
- [7] RAJMIC, P. *Využití waveletové transformace a matematické statistiky pro separaci signálu a šumu*. VUT v Brně, Brno, 2004. Dizertační práce.
- [8] SHAH, J. K., et al. Robust Voiced/Unvoiced Classification Using Novel Features And Gaussian Mixture Model. In *IEEE International Conference on Acoustics, Speech, & Signal Processing (ICASSP), May 17-21, 2004, Montreal, Canada*. [s.l.] : [s.n.], 2004. s. 4. Dostupný z WWW: <http://shahj.net/publications/icassp03_shah.pdf>.
- [9] SMÉKAL, Z. *Číslicové zpracování signálů*, VUT Brno, Brno, ČR, 2005.
- [10] STEJSKAL, V. Cvičení z předmětu Zpracování řeči 2 : Banky filtrů. In *Cvičení z předmětu Zpracování řeči*. [s.l.] : [s.n.], 2008. s. 11.
- [11] ŠEBESTA, V., SMÉKAL, Z. *Signály a soustavy*, VUT Brno, Brno, ČR, 2004.

Seznam příloh

- Příloha 1: Blokové schéma postupu signálu systémem
Příloha 2: Obsah přiloženého CD

Příloha 1: Blokové schéma postupu signálu systémem



Vysvětlivky k blokovému schématu:

- Vstupní nebo výstupní signály či proměnné jsou zobrazeny v blocích se zaoblenými rohy.
- Funkční bloky jsou zobrazeny jako obdélníky s tlustým okrajem.
- Šipky znázorňují směr průchodu signálu a proměnných systémem.
- Fontem Courier New jsou vypsány proměnné, které mají v programovém kódu stejný název.
- **Tučně** jsou vypsány proměnné, které představují určitý vstupní parametr, ale v kódu se nejmenují stejně.
- *Kurzívou* jsou vypsány možnosti nastavení, které má uživatel, pokud tyto možnosti nejsou již uvedeny ve formátu proměnných.

Výstupní signály:

- s_V
- s_UV
- s_A

Příloha č.2: Obsah příloženého CD

Příložené CD obsahuje tři adresáře:

/Matlab

Obsahuje tyto soubory:

- `kepstrum_det.m` – funkce provádějící mel-kepstrální analýzu
- `pohnout.wav` – věta „Mohli jsme si s tím pohnout.“ namluvená mnou
- `pohnout2.wav` – věta „Mohli jsme si s tím pohnout.“ namluvená mým bratrem
- `uv_detect.m` – funkce provádějící klasifikaci znělosti a neznělosti
- `wave_gui.fig` – GUI
- `wave_gui.m` – funkce ovládající GUI

Pro spuštění naprogramovaného GUI je třeba nastavit v Matlabu pracovní adresář právě na tento nebo veškeré M-soubory zkopírovat jinam (musí být pohromadě). Spuštění provedeme napsáním `wave_gui` do programové řádky a potvrzením.

/Testovani

Obsahuje některé výstupní soubory z prováděného testování. Jméno souboru vzniká automaticky při jeho tvorbě z jména testovaného souboru podle parametrů použitých při průchodu signálu systémem.

Názvu výstupního souboru je tvořen následujícím způsobem:

- jméno původního souboru bez přípony
- text „-SNR(**X**dB)“, kde **X** je numerická hodnota *SNR*
- text „-V“ pro signál s_V ; „-UV“ pro signál s_{UV} ; „-A“ pro signál s_A
- text „-uv(prahovani“, když je použita statistická metoda UV klasifikace; „-uv(mel-kepstrum“, když je použita metoda mel-kepstrálních koeficientů
- text „-ostr“, když je použita *Ostrá* UV klasifikace; „-post“, když je použita *Postupná* UV klasifikace
- text „-wv(**typ**,**N1**)“ pro signál s_V a s_A ; „-wv(**typ**,**N1**,**N2**,**koef**)“ pro signál s_{UV} , kde **typ** je typ použitého waveletu, **N1** je obecná hloubka dekompozice, **N2** je hloubka dekompozice v neznělých segmentech a **koef** je koeficient pro úpravu prahu waveletových koeficientů
- text „.wav“

Příklad názvu výstupního souboru:

- `pohnout-SNR(5dB)-UV-uv(mel-kepstrum-post)-wv(db8,5,6,0.8).wav`

Tento soubor vznikl ze vstupního signálu `pohnout.wav`, byl do něho přidán šum tak, že $SNR = 5$ dB, výstupní signál je typu s_{UV} , byla použita *Postupná* mel-kepstrální metoda UV klasifikace a potlačení šumu bylo provedeno pomocí waveletu typu Daubechies 8 s hloubkou dekompozice pro znělé segmenty $N1 = 5$, hloubkou dekompozice pro neznělé segmenty $N2 = 6$ a koeficientem pro úpravu prahu $koef = 0,8$.

/Text

Obsahuje elektronickou verzi diplomové práce ve formátu .pdf i s přílohami. Přílohy jsou dány i v samostatných souborech. Také se zde nachází soubor `metadata.pdf`.